Collibra

Collibra Platform

Data Quality & Observability Classic

Version 2025.06

Collibra

Data Quality & Observability Classic guide

Release date: June 08, 2025

Revision date: June 05, 2025

You can find the most up-to-date technical documentation on our documentation site at

https://productresources.collibra.com/docs/collibra/latest/#cshid=data-quality

Contents

Contents ..... **ii**

Release Notes xii

Release 2025.06 xii

Release 2025.05 xix

Release 2025.04 xxvii

Release 2025.03 xxxiv

Release 2025.02 xlii

Builds cccl

Installation ccclxiv

Connections ..... dxiii

Before you begin ..... dlxxxix

Prerequisites ..... dlxxxix

Steps ..... dlxxxix

Workload Identity ..... dxc

Prerequisites ..... dxci

1. Authenticate to API ..... dxci

2. Create a connection ..... dxcii

3. Verify the connection ..... dxciii

4. Use Workload Identity to authenticate your connection ..... dxciv

Core Concepts ..... dccxxvi

Prerequisites ..... dccxxxiii

How does SQL Assistant for Data Quality work? ..... dccxxxv

Overview of the SQL Assistant for Data Quality components ..... dccxxxviii

Requirements and permissions ..... dccxli
Steps ..... dccxliii
Requirements ..... dcclix
Configuring Pushdown for Databricks ..... dcclx
What's next? ..... dcclxii
Running Snowflake setup scripts ..... dcclxiii
Setting up the Snowflake Connection template ..... dcclxvii
Known limitations ..... dcclxix
What's next? ..... dcclxx
Requirements ..... dcclxx
Configuring Pushdown for BigQuery ..... dcclxxi
Limitations ..... dcclxxiii
What's next? ..... dcclxxiii
Requirements ..... dcclxxiv
Configuring Pushdown for Athena ..... dcclxxiv
What's next? ..... dcclxxvi
Known limitations ..... dcclxxvi
Requirements ..... dcclxxvii
Configuring Pushdown for Redshift ..... dcclxxvii
What's next? ..... dcclxxix
Known limitations ..... dcclxxix
Requirements ..... dcclxxix
Configuring Pushdown for Trino ..... dcclxxx
What's next? ..... dcclxxxii
Requirements ..... dcclxxxii
Configuring Pushdown for SQL Server ..... dcclxxxiii

What's next? ..... dcclxxxv
Known limitations ..... dcclxxxvi
Requirements ..... dcclxxxvi
Configuring Pushdown for SAP HANA ..... dcclxxxvii
What's next? ..... dcclxxxix
Prerequisites ..... dcclxxxix
Configuring Pushdown for Oracle ..... dcclxxxix
What's next? ..... dccxcii
Steps ..... dcccxli
Known limitations ..... dcccxliv
JDBC vs Local Data ..... dcccxlviii
Prerequisites ..... dccclxxi
Using Parallel JDBC ..... dccclxxi
Known limitations ..... dccclxxiii
Break record details ..... dccclxxiv
Prerequisites ..... dccclxxv
Steps ..... dccclxxvii
Setting up break record archival ..... dccclxxix
Viewing break record details ..... dccclxxxiii
Limitations ..... dcccxc
Collibra DQ Log Stages ..... dcccxcvii
Retrieve Stage 2 and 3 Logs Using a Managed Hadoop Spark Cluster ..... dcccxcix
Explorer Web Audit Logs ..... cm
Connections Web Audit Logs ..... cm
Rules Web Audit Logs ..... cm
Steps ..... cmiii

Managing Spark resources from the command line ..... cmxlii
Command Line Reference ..... cmxlv
DQ Job JDBC ..... **cmlxviii**
Parallel JDBC ..... cmlxx
DQ Job Hive ..... **cmlxxi**
DQ Job Files ..... **cmlxxvii**
DQChecks with file ..... cmlxxviii
DQChecks with multiple files ..... cmlxxxi
DQ Job JSON ..... **cmlxxxvi**
Files ..... cmlxxxvi
Using Spark SQL ..... cmlxxxvi
Using DQ Libraries ..... cmlxxxviii
Known Limitations ..... cmlxxxix
DQ Job S3 ..... **cmxc**
Advanced ..... **cmxciii**
Monthly Data ..... cmxciii
Monthly BackRun (Using Collibra Data Quality's built-in Monthly) ..... cmxciv
Daily Data ..... cmxciv
Daily Data (Using Collibra Data Quality's built-in Daily) ..... cmxcv
Daily Data with Timestamp instead of Date ..... cmxcv
OR Timestamp using $ ..... cmxcv
Hourly Data ..... cmxcvi
DQ Check Template with Service Hook ..... cmxcvi
REST API End Point ..... cmxcvii
DQ Job Back Run ..... **cmxcix**
DQ Job Cron ..... **mi**

Working with Link IDs ..... **miv**

Importance of uniqueness ..... miv

Link ID and the Metastore ..... mvi

Set up link IDs ..... mvi

View break records ..... mvi

DQ Job Validate Source ..... **mix**

Add Date Column ..... **mxiii**

Example ..... mxiii

Cloudera CLASSPATH ..... **mxiv**

What is a CLASSPATH? ..... mxiv

What is a JAR? ..... mxxi

Common issues that can occur with CLASSPATH's ..... mxxii

Column Matching ..... **mxxiii**

Date Time Variable Options ..... **mxxv**

Deploy Mode ..... **mxxvi**

Lookback ..... **mxxvii**

Lookback options ..... mxxvii

Understanding lookback command line flags ..... mxxix

Filter & Filter Not ..... **mxxx**

Example ..... mxxx

The inverse ..... mxxx

Header Check ..... **mxxxi**

Configuring with the Check Header checkbox ..... mxxxi

Configuring from the command line ..... mxxxi

Multiple Pattern Relationships ..... **mxxxiii**

Example ..... mxxxiii

Nulls in Datasets ..... **mxxxiv**

Example ..... mxxxiv

Spark-shell Sample ..... **mxxxv**

Transform ..... **mxxxvii**

Example ..... mxxxviii

Known Limitations ..... mxxxix

Prerequisites ..... mcxiii

Steps ..... mcxiii

Configuring Rule Details ..... mcxxi

Managing a Livy Session ..... mcxxiii

Working with COUNT and GROUP BY ..... mcxxiv

Prerequisites ..... mcxxv

Using a filter query ..... mcxxv

Data steward use case ..... mcxxvi

Filter query scoring impact ..... mcxxviii

Adding a rule filter ..... mcxxxi

Limitations ..... mcxxxii

Prerequisites ..... mcxxxii

Adding a rule tolerance threshold ..... mcxxxiii

Viewing rule tolerance results ..... mcxxxiv

Requirements and permissions ..... mcxxxix

Steps ..... mcxli

Rules settings ..... mcli

Example of the Percent scoring method ..... mcliii

Example of the Absolute scoring method ..... mclv

Add Secondary Datasets ..... **mclix**

Adding a secondary dataset from the same connection ..... mclx
Adding a secondary dataset from a different connection ..... mclxi
Referencing secondary datasets ..... mclxi
Dataset-level stat rules ..... mclxv
Column-level stat rules ..... mclxv
Known Limitations ..... mclxviii
Data type ..... mclxviii
Score-based conditions ..... mccxc
Row count condition ..... mccxci
Run time conditions ..... mccxci
Stale data conditions ..... mccxci
Rule conditions ..... mccxcii
Where are timestamps and time zones in the UI? ..... mcccviii
Setting the time zone ..... mcccx
Admin Configuration ..... mcccxxiii
Prerequisites ..... mcccxxviii
Steps ..... mcccxxviii
What's next? ..... mcccxxix
Prerequisites ..... mcccxxxi
Steps ..... mcccxxxi
What's next? ..... mcccxxxii
Steps ..... mcccxlviii
Steps ..... mcccxlix
Steps ..... mccclii
Steps ..... mccclv
Steps ..... mccclxxxv

Prerequisites ..... mccclxxxviii
Steps ..... mccclxxxix
Steps ..... mcdiii
Steps ..... mcdiv
Prerequisites ..... mcdxvi
Configure Security Settings for Active Directory and LDAP ..... mcdxvi
Required script settings to configure LDAP ..... mcdxxiv
Script Settings for Search Configuration ..... mcdxxv
AD Group to Role Mapping Overview ..... mcdxxvi
What's next? ..... mcdxxvi
Mapping AD security groups to roles ..... mcdxxvii
Steps ..... mcdxxvii
Mapping security groups to roles when using SSO with SAML ..... mcdxxx
Configuring user search filter ..... mcdxxxiv
Steps ..... mcdxxxiv
Known limitations ..... mcdxxxvii
What's next? ..... mcdxxxviii
Steps ..... mcdxxxix
Known limitations ..... mcdxli
What's next? ..... mcdxli
Steps ..... mcdxlii
Set the SAML authentication properties ..... mcdxlv
Download service provider metadata for the IDP ..... mcdli
Steps ..... mcdli
Enable the SAML sign in option ..... mcdlii
Steps ..... mcdlii

Steps ..... mcdliii
Steps ..... mcdlv
Steps ..... mcdlxi
Configure the Azure AD B2C portal ..... mcdlxv
Configure the DQ Application ..... mcdlxvii
Restart the DQ Application ..... mcdlxxii
Test basic login functionality ..... mcdlxxii
Basic role mapping in DQ ..... mcdlxxiv
What's next? ..... mcdlxxiv
Populating user roles in Azure AD B2C via Microsoft Graph API ..... mcdlxxv
Test login with populated roles ..... mcdlxxix
Security Configuration ..... mcdxcviii
Integrations ..... mdxlvii
Solutions ..... mdlxxiv
APIs ..... mdlxxvii
Rule types ..... mdcli
Simple rule ..... **mdclii**
Example #1 ..... mdclii
Example #2 ..... mdcliv
Freeform SQL ..... **mdclix**
Individual statement ..... mdclix
Join statements ..... mdclxi
Known Limitations ..... mdclxiii
Data type rules ..... **mdclxv**
Rule types ..... mdclxv
Syntax ..... mdclxvi

Outliers ..... **mdclxvii**

Architecture ..... mdcclviii

Documentation Archive ..... mdcclxv

Release Notes

Important

- **Failure to upgrade to the most recent release of the Collibra Service and/or Software may adversely impact the security, reliability, availability, integrity, performance or support (including Collibra's ability to meet its service levels) of the Service and/or Software.** For more information, read our Collibra supported versions policy.
- Some items included in this release may require an additional cost. Please contact your Collibra representative or Collibra Account Team with any questions.

Release 2025.06

- Announcement
- New and improved
- Fixes

Release Information

- Expected release date of Data Quality & Observability Classic 2025.06: June 30, 2025
- Release notes publication date: June 3, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.06), Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.06, you must upgrade to Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.05 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
  - We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

In the 2025.06 and 2025.07 releases, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic. For a breakdown of Java and Spark availability in current and upcoming releases, click

"See what is changing" below.

1 See what is changing[1]

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

For

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

New and improved

Platform

- This application, which was previously known as Collibra Data Quality & Observability, is now called Data Quality & Observability Classic. This change reflects the introduction of the new Data Quality & Observability as part of the unified Collibra Platform. For more information about Data Quality & Observability, go to About Data Quality & Observability.

Jobs

- When archiving break records for Pushdown jobs, link IDs are now optional. This change provides greater flexibility in storing and accessing break records and offers a more scalable solution for remediating records outside of Data Quality & Observability Classic.
    - If Data Quality & Observability Classic does not have permission to write to tables in your source database, Pushdown jobs with or without link IDs selected require you to run an `ALTER` statement on the **COLLIBRA_DQ_RULES** table to add the new results column for JSON data. Data Quality & Observability Classic will attempt this alteration automatically, which will cause the rule to fail because it will attempt to write to a column that doesn't exist in your source database.

      Important If you don't use a link ID, the results column in the **COLLIBRA_DQ_RULES** table will become exponentially larger, leading to unintended storage and compute costs in your source database. To minimize these costs, you can use a link ID or move these results out of

> your source database and into a remote file storage system, such as Amazon S3.

- Data Quality & Observability Classic now checks that a link ID is set in the settings dialog on Explorer when creating a DQ job, to ensure the archive break records feature works when enabled. If one or more link ID columns are selected, a checkbox and drop-down list appear next to the **Archive Breaking Records** option in the settings dialog. If no link ID columns are selected, the checkbox and drop-down list are unavailable, and a tooltip appears when you hover over the option.
- Oracle Pushdown is now generally available.
- Oracle Pushdown connections do not support the following data types:
  - LONG RAW
  - NCHAR
  - NVARCHAR2
  - ROWID
  - BYTES
- Trino Pushdown jobs now support source-to-target analysis for datasets within the same Trino Pushdown connection.
- SAP HANA, Snowflake, SQL Server, and Trino Pushdown connections do not support the VARBINARY data type.
- The **valsrclimitui** admin limit, which controls the number of Validate Source records shown in the UI, now works as expected. For example, if **valsrclimitui** is set to 3, the UI shows 3 records from the Column Schema and 3 records from the Cell Panels.

> **Note** The **fpglimitui**, **histlimitui**, and **categoricallimitui** admin limits are not working as expected today. Further, any admin limit ending "ui", such as **valsrclimitui**, is intended only for updates made via the Admin Limits page of the Admin Console. Changes made to these admin limits via the command line will not be reflected.

Rules

- Validation is no longer performed automatically for the following data type rules:
  - DATATYPECHECK
  - DATECHECK
  - DOUBLECHECK
  - EMPTYCHECK
  - INTCHECK
  - NULLCHECK
  - STRINGCHECK
- The Rule Workbench now has a warning message when the preview limit exceeds the recommended maximum of 500 records.
- The maximum length of the name of a rule is now 250 characters.
- The Preview Breaks modal now indicates the name of the rule.
- The **Low**, **Medium**, and **High** buttons now correctly reflect your selection when you reopen the Rule Details dialog box after automatically changing a rule's points. Additionally, if you manually update the score, the button automatically adjusts to match the corresponding scoring range:
  - **Low**: 1-4 points
  - **Medium**: 5-24
  - **High**: 25-100

Admin Console

- Business unit names on the Business Unit page in the Admin Console cannot contain the special characters / (forward slash), ` (backtick), | (pipe), or $ (dollar sign). If you use any of these special characters, an error message appears, and the Edit Business Unit dialog box remains open until the special characters are removed.
- Data retention policies now apply to the following metastore tables:
  - alert_output
  - behavior
  - dataset_activity
  - dataset_schema_source

    - hint
    - validate_source
- The **Title** and **Story** fields of the User Story section of the User Profile page cannot contain the following special characters:
    - ‘ (single quote)
    - “ (double quote)
    - -- (double dash)
    - < (less than)
    - > (greater than)
    - & (ampersand)
    - / (forward slash)
    - ` (backtick)
    - | (pipe)
    - $ (dollar sign)

      If you use any of these special characters, an error message appears, and the Edit User Story dialog box remains open until the special characters are removed.

Fixes

Rules

- You can now apply rule filter queries to rules created using rule templates on Pushdown datasets.
- Duplicated data quality rules no longer appear after you import data assets using CMA.
- Rules now run without errors when you use a single link ID to archive break records with SQL Server Pushdown.

Profile

- Divide-by-zero errors and issues with empty histograms are now resolved. Calculations and visualizations now work as expected without interruptions.

Findings

- You can now export rule break records with accurate counts. The total number of rule break records for a rule now matches the number in the **Breaking Records** column of the exported file.

Integration

- When you rename a rule in Data Quality & Observability Classic that is part of a dataset integrated with Collibra Platform, additional Data Quality Rule assets are no longer created during the next integration. The existing Data Quality Rule asset is also no longer unintentionally set to a "suppressed" state during the next integration, and its name is updated to match the name in Data Quality & Observability Classic.

Release 2025.05

- Announcement
- New and improved
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.05: June 2, 2025
- Release notes publication date: May 6, 2025

Announcement

> **Important**
>
> As a security measure, we are announcing the end of life of the Java 8 and 11

versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.05), Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.05, you must upgrade to Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.05 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
  - We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic. For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing[1]

1 For

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

New and improved

Warning We identified a known security vulnerability (CVE-2025-48734) in this release. We will address this issue in the upcoming 2025.05.1 patch.

Platform

- You can now authenticate users via Microsoft Azure Active Directory B2C when signing into Data Quality & Observability Classic.
- Only users with the ROLE_ADMIN role can now see the username and connection string of JDBC connections on the Connections page and in API responses, improving connection security.
- Metastore credentials are no longer stored in plain text when you create a DQ Job via a notebook. This improvement increases the security of your credentials.
- We enhanced the security of Data Category validation.
- We improved the security of our application.

Jobs

- Trino Pushdown jobs now support validate source.
- The Run Job buttons now have two enhancements:
  - The Run Job with Date button on the Jobs tab of the Findings page is now labeled **Select Run Date**.
  - The **Run Job** button on the metadata bar now includes a helpful tooltip on hover.
- SAP HANA connections now support the operators `=>`, `||`, and `&&` for scoped queries, the reserved word `'LIKE_REGEXTR'`, and the function `REPLACE` in scoped queries.

- The assignments queue is now disabled for the validate source activity on new installations of Data Quality & Observability Classic. To enable it, an admin can set **valsrcdisableaq** to "false" on the Admin Limits page.

Rules

- When Archive Break Records is disabled, you can now use NULL link IDs with Athena, BigQuery, Hive, Redshift, and SQL Server connections.
- When Archive Break Records is enabled, and a rule doesn't specify a link ID column but the job does, you can no longer archive break records in these scenarios, as Data Quality & Observability Classic no longer appends the link ID to the rule. When a rule does not contain the selected link ID, the break records for that rule will not be archived, and an exception will occur. The job will execute successfully, but the rule will not produce any archived break records, and the exception message will explain that the link ID is not part of the rule.
- Trino Pushdown jobs now support profiling and custom rules on columns of the ARRAY data type.
- The Rule Details dialog on the Rule Workbench page now has an optional **Purpose** field. This allows you to add details about Collibra Platform assets associated with your rule and see which assets should relate to your Data Quality Rule asset upon integration.

Alerts

- You can now set up Data Quality & Observability Classic to send data quality alerts to one or more webhooks, eliminating the dependency on SMTP for email alerts.
- You can now rename alerts on the Dataset Alerts page. When you rename an alert, the updated name is reflected in the alert_nm column of the alert_cond Metastore table. The updated name applies only to alerts generated in future job runs and does not affect historical data.
- In addition to the existing rule name condition variable for percentages, you can now include a rule name with a value identifier, such as `test_rule.score > 0`, to create condition alerts based on rule scores.

Profile

- When you retrain a breaking adaptive rule, negative values are now supported for lower and upper bounds, min, max, and mean.

Findings

- To enhance usability, we moved the data preview from the Rules results table into a dedicated modal, making it easier to navigate and view data.
    - The Actions button is now always visible on the right side of the Rules tab, and includes the following options:
        - **Archived Breaks**, previously called Rule Breaks, retains the same content and functionality.
        - **Preview Breaks**, previously embedded under the in the first column, shows all breaking records for a given rule. This option provides the same information that was previously available in the Rules tab and still requires the ROLE_DATASET_ACCESS role. Additionally, the Rule Break Preview modal reflects the preview limit of the rule.
    - Tooltips were added to various components on the Rules tab and Rule Break Preview modal.

Integration

- You can now use the **Map to primary column only** switch in the Connections step of the integration setup wizard. This allows you to map a rule to only the Column asset in Collibra Platform that corresponds to the primary column in Data Quality & Observability Classic.
- When you rename a custom rule in Data Quality & Observability Classic with an active integration with Collibra Platform, a new asset is no longer created in Collibra Platform. Instead, renamed rules reuse the Collibra Platform asset IDs associated with the previous rule name when their metadata is integrated.

APIs

- The **/v3/datasetDefs/template** API endpoint now returns the nested key-value pairs for shape settings and job schedule in the datasetDefs JSON object.

Admin Console

- You can now use the **previewlimit** setting on the Admin Limits page to define the maximum number of preview results shown in the Dataset Overview and Rule Workbench. The default value is 250.

  > **Warning** Large **previewlimit** values can negatively impact performance.

Fixes

Platform

- All datasets are now available when you run a dataset rule fetch with dataset security disabled for the admin user account.
- FIPS-enabled standalone installations of Data Quality & Observability Classic now support SAML authentication.

Jobs

- When you remove a table from a manually triggered or scheduled job, you now receive a more descriptive error stating that the table does not exist instead of a generic syntax error.
- You can now use Amazon S3 as a secondary dataset name for a rule query.
- You can now rerun a job for a selected date using the **Run DQ Job** button on the metadata bar of the Findings page.
- The command line now retains the created query, including the `${rd}` parameter, when you run a job using the Run DQ Job button on the metadata bar.

- The controller response for queries to the alert_cond table in the Metastore now maps internal objects correctly.
- The Profile page now correctly shows the top 5 TopN and BottomN shape results for Pullup jobs.
- When you add a transform to a job with histogram enabled or set to auto, the job processes as expected, and aliased columns display correctly on the Profile page.

Rules

- You can again save rules on Pushdown jobs that exceed ⅓ of a buffer page.
- Pushdown rules that use stat variables now run with the correct information from the current job, rather than using data from previous job runs.
- The DOUBLECHECK rule on PostgreSQL and Redshift Pullup jobs now flags rows with negative double values as valid.
- The Result Preview in the Rule Workbench no longer produces an error when you use out-of-the-box templates.

Profile

- You can now compare the Baseline and Run values for the "Filled rate" when the Quality switch is enabled for a column on the Profile report.

Findings

- Pushdown jobs run on Snowflake connections now show as failed on the Findings page if a password retrieval issue occurs.

Dataset Manager

- When you apply an option from the Actions drop-down list, such as delete a dataset, the correct dataset now has the action applied to it.

Admin Console

- When you select a submenu option from the Admin Console menu, the submenu section now remains open.

Release 2025.04

- Announcement
- New and improved
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.04: April 28, 2025
- Release notes publication date: April 2, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.04), only the Java 17 build profile of Data Quality & Observability Classic contains all new and improved features and bug fixes listed in the release notes. The Java 8 and 11 build profiles for Standalone installations contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases.

Depending on your installation of Data Quality & Observability Classic, you can

expect the following in this release for Java 17 build profiles:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.04, you must upgrade to Java 17 and Spark 3.5.3 if you did not already do so in the 2025.02 or 2025.03 release.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.04 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
  - We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

While this release contains Java 8, 11, and 17 builds of Data Quality & Observability Classic for Standalone installations, it is the final release to contain Java 8 and 11 builds and Spark versions older than 3.5.3. Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic.

For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing[1]

1 For

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

New and improved

Platform

- The Platform Path of the SQL Assistant for Data Quality feature now uses the Gemini AI model version **gemini-1.5-pro-002**.

  Important To continue using the Platform Path of the SQL Assistant for Data Quality feature, you must upgrade to Data Quality & Observability Classic 2025.04.

- FIPS-enabled standalone installations of Data Quality & Observability Classic now support SAML authentication.
- The DQ agent now automatically restores itself when the Metastore reconnects after a temporary disconnection due to maintenance or a glitch.
- We implemented robots.txt to prevent all search engine crawlers from indexing any Data Quality & Observability Classic content.

Jobs

- We are pleased to announce that Oracle Pushdown is now available for preview testing.
- When you add a timeslice to a timeUUID data type column in a Cassandra dataset, an unsupported data type error message now appears.
- Dremio, Snowflake, and Trino Pullup jobs now support common table expressions (CTE) for parallel JDBC processing.
- You can now archive the break records of shapes from Trino Pushdown jobs.
- Link IDs for exact match duplicates are no longer displayed on the findings page.
- We improved the insertion procedure of validate source findings into the Metastore.

Rules

- When you add or edit a rule with an active status, SQL syntax validation now runs automatically when you save it. If the rule passes validation, it saves as expected. If validation fails, a dialog box appears, asking whether you want to continue saving with errors. Rules with an inactive status save without validation checks.
- When a rule condition exceeds 2 lines, only the first 2 lines are shown in the Condition column of the Rules tab on the Findings page. You can click "more..." or hover over the cell to show the full condition.

Profile

- The Profile page now correctly shows the top 5 TopN and BottomN shape results for Pullup jobs.
- When there are only two unique string values, the histogram on the Profile page now shows them correctly.

Findings

- To improve page alignment across the application, the Findings page now has a page title.

Alerts

- Rule score-based alerts now respect rule tolerance settings. Alerts are suppressed when the rule break percentage falls within the tolerance threshold.

Dataset Manager

- If you don't have the required permissions to perform certain tasks in the Dataset Manager, such as updating a business unit, an error message now appears when you attempt the action.

Scorecards

- You can now delete scorecards with names that contain trailing empty spaces, such as "scorecard " and "test scorecard ".

Integration

- You can now search with trailing spaces when looking up a community for tenant mapping in the Data Quality & Observability Classic and Collibra Platform integration configuration.

APIs

- You now need the **ROLE_ADMIN** or **ROLE_ADMIN_VIEWER** role to access the **/v2/getdatasetusagecount** API endpoint.

Fixes

Platform

- We have improved the security of our application.
- The data retention process now works as expected for all tenants in multi-tenant instances.
- Permissions errors for SAML users without the **ROLE_PUBLIC** role are now resolved. You no longer need to assign **ROLE_PUBLIC** to users who already have other valid roles.

Jobs

- Native rules, secondary dataset rules, and scheduled jobs on Databricks Pullup connections authenticated via EntraID Service Principal now run as expected.

- Pushdown jobs scheduled to run concurrently now process the correlation activity correctly.
- Pushdown jobs with queries that contain new line characters now process correctly, and the primary table from the query is shown in the Dataset Manager.

Findings

- The confidence calculations for numerical outliers in Pullup jobs have been updated for negative values. Positive value confidence calculations and Pushdown calculations remain unchanged.

Alerts

- Conditional alerts now work as expected when based on rules with names that start with numbers.

Integration

- The default setting for the integration schema, table, and column recalculation service (**DQ_DGC_MAPPING_STATS_SCHEDULER_ENABLED**) is now **false**, reducing unnecessary database activity. You can enable the service or call it through the API when needed.

  > **Important** From versions 2024.11 to 2025.03 of Data Quality & Observability Classic, if you don't want queries from the recalculation of mapped and unmapped stats of total entities to run, set the **DQ_DGC_MAPPING_STATS_SCHEDULER_ENABLED** to **false** in the owl-env.sh or Web ConfigMap. You can keep **DQ_INTEGRATION_SCHEDULER_ENABLED** set to **true**.
  > For more information, go to the Collibra Support center.

- The Quality tab for database assets is now supported in out-of-the-box aggregation path configurations.
- The auto map feature now correctly maps schemas that contain only views.

- Dimension configuration for integration mapping no longer shows duplicate Data Quality & Observability Classic dimensions from the dq_dimension table.

API

- When you copy a rule with an incorrect or non-existent dataset name using the **v3/rules/copy** API, an error message now specifies the invalid dataset or rule reference. This prevents invalid references in the new dataset.

Release 2025.03

- Announcement
- Enhancements
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.03: March 31, 2025
- Release notes publication date: March 4, 2025

Announcement

> **Important**
>
> As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.
>
> **In this release (2025.03)**, Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality

& Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.03, you must upgrade to Java 17 and Spark 3.5.3 if you have not already done so in the 2025.02 release.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.03 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
  - We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

The April 2025 (2025.04) release will contain Java 8, 11, and 17 versions of Data Quality & Observability Classic. This will be the final release to contain Java 8 and 11 builds and Spark versions older than 3.5.3, and will include feature enhancements and bug fixes from the 2025.02 release and critical bug fixes from the 2025.03 and 2025.04 releases. Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic. For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing[1]

1 For

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

Enhancements

Platform

- On April 9, 2025, Google will deprecate the Vertex text-bison AI model, which SQL Assistant for Data Quality uses for the "preview path" option. To continue using SQL Assistant for Data Quality, you must switch to the "platform path," which requires an integration with Collibra Platform. For more information about how to configure the platform path, go to Prerequisites.
- We removed support for Kafka streaming.

Connections

- Private endpoints are now supported for Azure Data Lake Storage (Gen2) (ABFSS) key- and service principal-based authentication and Azure Blob Storage (WASBS) key-based authentication using the **cloud.endpoint=<endpoint>** driver property. To do this, add **cloud.endpoint=<endpoint>** to the Driver Properties field on the Properties tab of an ABFSS or WASBS connection template. For example, **cloud.endpoint=microsoftonline.us**.
- Trino now supports parallel processing in Pullup mode. To enable this enhancement, the Trino driver has been upgraded to version **1.0.50**.

Jobs

- The **Jobs** tab on the Findings page now includes two button options:
  - **Run Job from CMD/JSON** allows you to run a job with updates made in the job's command line or JSON. This option is **Run Job from JSON** in Pushdown mode since the command line option is not available.

    - **Run Job with Date** allows you to select a specific run date.

      > **Note** The **Run DQ Job** button on the metadata bar retains its functionality, allowing you to rerun a job for the selected date.

- The order of link IDs in the **rule_breaks** and **opt_owl** Metastore tables for Pushdown jobs is now aligned.
- The options to archive the break records of associated monitors in the Explorer settings dialog box of a Pushdown job are now disabled when the Archive Break Records option is disabled at the connection-level.
- We updated the logic of the maximum global job count to ensure it only increases, rather than fluctuating based on the maximum count of the last run job's tenant. This change allows tenants with lower maximum job counts to potentially run more total jobs while still enforcing the maximum connections for individual jobs. Over time, the global job count will align with the highest limit among all tenants.
- You can now archive the break records of shapes from SAP HANA and Trino Pushdown jobs.
- You can now use the new `"behaviorShiftCheck"` element in the JSON payload of jobs on Pullup connections. This allows you to enable or disable the shift metric results of Pullup jobs, helping you avoid misleading mixed data type results in string columns. By default, the `"behaviorShiftCheck"` element is enabled (set to `true`). To disable it, use the following configuration: `"behaviorShiftCheck": false`.

Rules

- You can now set the **MANDATORY_PRIMARY_RULE_COLUMN** setting to **TRUE** from the Application Configuration Settings page of the Admin Console to require users to select a primary column when creating a rule. This requirement is enforced when a user creates a new rule or saves an existing rule for the first time after the setting is enabled. Existing rules are not affected automatically.
- The names of Template rules can no longer include spaces.
- CSV rule export files now include a **Filter Query** column when a rule filter is defined. If no filter is used, the column remains empty. The **Condition** column has been

renamed to **Rule Query** to better distinguish between rule and filter queries. Additionally, the **Passing Records** column now shows the correct values.

- You can now apply custom dimensions added to the **dq_dimension** table in the metastore to rules from the Rule Details dialog box on the Rule Workbench. These custom dimensions are also included in the Column Dimension Report.
- Livy caching now uses a combination of username and connection type instead of just the username. This improvement allows you to seamlessly switch between connections to access features such as retrieving the run results previews for rules or creating new jobs for remote file connections, without manually terminating sessions.

> **Note** Manually terminating a Livy session will still end all sessions associated with that user.

Findings

- You can now work with the Findings page in full-screen view.

Scorecards

- You now receive a helpful error message in the following scenarios:
  - Create a scorecard without including any datasets.
  - Update a scorecard and remove all its datasets.
  - Add or update a scorecard page with a name that already exists.

Fixes

Platform

- We have improved the security of our application.
- We mitigated the risk of SQL injection vulnerabilities in our application.

- Helm Charts now include the external JWT properties required to configure an externally managed JWT.

Jobs

- Google BigQuery jobs no longer fail during concurrent runs.
- When you add a `-conf` setting to the agent configuration of an existing job and rerun it, the command line no longer includes duplicate `-conf` parameters.
- When you expand a Snowflake connection in Explorer, the schema is now passed as a parameter in the query. This ensures the Generate Report function loads correctly.
- Record change detection now works as expected with Databricks Pushdown datasets.
- When you select a Parquet file in the Job Creator workflow, the **Formatted view** tab now shows the file's formatted data.
- When you edit a Pullup job from the **Command Line**, **JSON**, or **Query** tab, the changes initially appear only on the tab where you made the edits. After you rerun the job, the changes are reflected across all three tabs.
- The Dataset Overview now performs additional checks to validate queries that don't include a `SELECT` statement.

Rules

- When you update the adaptive level or pass value option in the Change Detection dialog box of an adaptive rule, you must now retrain it by clicking **Retrain** on the Behaviors tab of the Findings page.
- @t1 rules on file-based datasets with a row filter now return only the rows included in the filter.
- @t1 rules on Databricks datasets no longer return a NullPointerException error.
- When you run Rule Discovery on a dataset with the "Money" Data Class in the Data Category, the job no longer returns a syntax error when it runs.

Findings

- We updated the time zone library. As a result, some time zone options, such as "US/Eastern," have been updated to their new format. Scheduled jobs are fully compatible with the corresponding time zones in the new library. If you need to adjust a time zone, you must use the updated format. For example, "US/Eastern" is now "America/New_York."
- Labels under the data quality score meter are now highlighted correctly according to the selected time zone of the dataset.

Alerts

- You no longer receive erroneous job failure alerts for successful runs. Additional checks now help determine whether a job failed, improving the accuracy of job status notifications.
- You can now consistently select or deselect the **Add Rule Details** option in the Condition Alert dialog box.

Reports

- The link to the Dataset Findings documentation topic on the Dataset Findings report now works as expected.

Connections

- Editing an existing remote file job no longer results in an error.
- Teradata connections now function properly without requiring you to manually add the **STRICT_NAMES** driver property.

APIs

- When you run a job using the **/v3/jobs/run** API that was previously exported and imported with **/v3/datasetDefs**, the Shape settings from the original job now persist

in the new job.
- Bearer tokens generated in one environment using the **/v3/auth/signin** endpoint (for local users) or the **/v3/auth/oauth/signin** endpoint (for OAuth users) are now restricted to that specific Data Quality & Observability Classic environment and cannot be used across other environments.
- We improved the security of our API endpoints.

Integration

- You can now use the automapping option to map schemas, tables, and columns when setting up an integration between Data Quality & Observability Classic and Collibra Platform in single-tenant Data Quality & Observability Classic environments.
- The Quality tab now correctly shows the data quality score when the head asset of the starting relation type in the aggregation path is a generic asset or when the starting relation type is based on the co-role instead of the role of the relation type.
- Parentheses in column names are no longer replaced with double quotes when mapped to Collibra Platform assets. This change allows automatic relations to be created between Data Quality Rule and Column assets in Collibra Platform.

Release 2025.02

- Announcement
- Enhancements
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.02: February 24, 2025
- Release notes publication date: January 20, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.02) and the March (2025.03) release, Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - You must upgrade to Java 17 and Spark 3.5.3 to install Data Quality & Observability Classic 2025.02.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.02 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to

configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.

- We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

The April 2025 (2025.04) release will contain Java 8, 11, and 17 versions of Data Quality & Observability Classic. This will be the final release to contain Java 8 and 11 builds and Spark versions older than 3.5.3, and will include feature enhancements and bug fixes from the 2025.02 release and critical bug fixes from the 2025.03 and 2025.04 releases. Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic. For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing[1]

For

[1]

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

Enhancements

Platform

- When an externally managed user enables assignment alert notifications and enters an email address on their user profile, findings assigned to that email address now receive alert notifications.
- **SAML_ENTITY_BASEURL** is now a required property for SAML authentication.
- To improve the security of our application, we upgraded SAML. As part of the upgrade, the following SAML Authentication are no longer used:
  - **SAML_TENANT_PROP_FROM_URL_ALIAS**
  - **SAML_METADATA_USE_URL**
  - **SAML_METADATA_TRUST_CHECK**
  - **SAML_INCLUDE_DISCOVERY_EXTENSION**
  - **SAML_SUB_DOMAIN_REPLACE_SOURCE**
  - **SAML_SUB_DOMAIN_REPLACE_TARGET**
  - **SAML_MAX_AUTH_AGE**

    Note If the **SAML_LB_EXISTS** property is set to **true** and **SAML_LB_INCLUDE_PORT_IN_REQUEST** set to **false**, you may need to update **SAML_ENTITY_BASEURL** to include the port in the URL. The **SAML_ENTITY_BASEURL** should match the IdP's ACS URL.

    Additionally, we recommend using the Collibra DQ UI when signing in with SAML, as SP-initiated sign-on is not fully supported in Collibra DQ.

> **Tip** We recommend synchronizing the clocks of the Identity Provider (IdP) and Service Provider (Collibra DQ) using Network Time Protocol (NTP) to prevent authentication failures caused by significant clock skew.

Connections

- You can now limit the schemas that are displayed in a JDBC connection in Explorer. This enhancement helps you manage usage and maintain security by restricting access to only the necessary schemas. (idea #CDQ-I-152)

  > **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.

- Hive connections now use the driver class name, `com.cloudera.hive.jdbc.HS2Driver`. Additionally, only Hive versions 2.6.25 and newer are supported.

  > **Warning** If you have a Standalone installation of Data Quality & Observability Classic and leverage Hive connections, you must upgrade to the latest 2.6.25 Hive driver to use Data Quality & Observability Classic 2025.02.

Jobs

- DQ Jobs on Trino Pushdown connections now allow you to select multiple link ID columns when setting up the dupes monitor.

Rules

- We are delighted to announce that rule filtering is now generally available. To use rule filtering, an admin needs to set RULE_FILTER to TRUE in the Application

Configuration.
- We are also pleased to announce that rule tolerance is generally available. However, the RULE_TOLERANCE_BETA setting must remain set to TRUE in the Application Configuration until the 2025.04 version.
- You can now define a DQ Dimension when you create a new template or edit an existing one to be applied to all new custom rules created using this template. Additionally, a Dimension column is now shown on the Templates page.
- There is now a dedicated break_msg column in the rule_output Metastore table, which shows the break message when a rule break occurs.

Dataset Manager

- You can now add meta tags up to 100 characters long from the Edit option under the Actions menu on the Dataset Manager. (idea #CDQ-I-74)

Fixes

Platform

- We have improved the security of our application.
- To improve the security of our application, Data Quality & Observability Classic now supports a Content Security Policy (CSP).

Jobs

- When a scheduled job runs, only the time it is scheduled to run displays in the Scheduled Time column on the Jobs page.
- When you edit and re-run a DQ Job with a source-to-target mapping of data from two different data sources, the `-srcds` parameter on the command line now correctly contains the `src_` prefix before the source dataset, and the DQ Job re-runs successfully.

- When you include a custom column via the source query of a source-to-target mapping, the validate overlay function no longer fails with an error message.
- When you clone a dataset from Dataset Manager, Explorer, or the Job tab on Findings, the Job Name field and the Command Line in the Review step of Explorer now correctly reflect the "temp_cloned_dataset" naming convention for cloned datasets.
- DQ Jobs with the same run date can no longer run concurrently.
- Pullup DQ Jobs with names that include dashes, such as "sample-dataset," no longer remain stalled in the "staged" activity when you rerun them.
- The correlation and histogram activities no longer cause DQ Jobs on Databricks Pushdown connections to fail sporadically.
- The correlation activity now displays correctly in the UI and no longer causes DQ Jobs on Pushdown connections to fail.

Rules

- When you use Freeform SQL queries with `LIKE %` conditions, for example, `SELECT * FROM @public.test where testtext LIKE '%a, c%'`, they now return the expected results.
- SQL Server job queries where the table name is escaped with brackets, for example, `select * from dbo.[Table]`, now process correctly when the job runs.
- The Rule Results Preview button is no longer disabled when the API call to gather the Livy status fails due to an invalid or non-existent session. The API call now correctly manages the cache for Livy sessions terminated due to idle timeout.
- The contents of the Export Rules with Details now include data from the new Tolerance rule setting.
- Data type rules now evaluate the contents of each column, not just the data type of each column to ensure the correct breaks are returned.
- SQL reserved keywords included in string parsing are now correctly preserved in their original case.
- When you update the name of a rule and an alert is configured on it, the alert will now show the updated name when sent.

Alerts

- When you set a breaking rule as passing, a rule status alert for rules with breaking statuses no longer sends erroneous error messages for that rule.

Connections

- When you substitute a PWD value as a sensitive variable on the Variables tab of a Databricks connection template, the sensitive variable in the connection URL is now set correctly for source-to-target mappings where Databricks is the source dataset.

APIs

- The **/v3/jobs/<jobId>/breaks/rules** endpoint no longer returns a 500 error when using a valid jobId. Instead, it now returns empty files when no results are found for exports without findings.
- When you schedule a Job to run monthly or quarterly, the `jobSchedule` object in the **/v3/datasetDefs** endpoint now reflects your selection.
- When you run the **/v3/rules/{dataset}/{ruleName}/{runId}/breaks** endpoint when Archive Rules Break Records is enabled, break records are now retrieved from the source system instead of the Metastore.
- Meta tags are now correctly applied to new datasets that are created with the **/v3/datasetDefs** endpoint.
- When an integration import job fails, the **/v3/dgc/integrations/jobs** endpoint now returns the correct “failed” status. Additionally, the integration job status “ignored” is now available.

Integration

- An invalid RunID no longer returns a successful response when using a Pushdown dataset with the **/v3/jobs/run** endpoint.

Data Quality & Observability Classic release archive

Release 2025.06

- Announcement
- New and improved
- Fixes

Release Information

- Expected release date of Data Quality & Observability Classic 2025.06: June 30, 2025
- Release notes publication date: June 3, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.06), Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file.

Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.

- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.06, you must upgrade to Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.05 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
  - We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

In the 2025.06 and 2025.07 releases, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic. For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing[1]

For

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✕ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✕ No	✕ No	✓ Yes	3.5.3 only	
2025.03	✕ No	✕ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

> more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

New and improved

Platform

- This application, which was previously known as Collibra Data Quality & Observability, is now called Data Quality & Observability Classic. This change reflects the introduction of the new Data Quality & Observability as part of the unified Collibra Platform. For more information about Data Quality & Observability, go to About Data Quality & Observability.

Jobs

- When archiving break records for Pushdown jobs, link IDs are now optional. This change provides greater flexibility in storing and accessing break records and offers a more scalable solution for remediating records outside of Data Quality & Observability Classic.
    - If Data Quality & Observability Classic does not have permission to write to tables in your source database, Pushdown jobs with or without link IDs selected require you to run an `ALTER` statement on the **COLLIBRA_DQ_RULES** table to add the new results column for JSON data. Data Quality & Observability Classic will attempt this alteration automatically, which will cause the rule to fail because it will attempt to write to a column that doesn't exist in your source database.

      > **Important** If you don't use a link ID, the results column in the **COLLIBRA_DQ_RULES** table will become exponentially larger, leading to unintended storage and compute costs in your source database. To minimize these costs, you can use a link ID or move these results out of your source database and into a remote file storage system, such as Amazon S3.

- Data Quality & Observability Classic now checks that a link ID is set in the settings dialog on Explorer when creating a DQ job, to ensure the archive break records feature works when enabled. If one or more link ID columns are selected, a checkbox and drop-down list appear next to the **Archive Breaking Records** option in the settings dialog. If no link ID

columns are selected, the checkbox and drop-down list are unavailable, and a tooltip appears when you hover over the option.
- Oracle Pushdown is now generally available.
- Oracle Pushdown connections do not support the following data types:
  - LONG RAW
  - NCHAR
  - NVARCHAR2
  - ROWID
  - BYTES
- Trino Pushdown jobs now support source-to-target analysis for datasets within the same Trino Pushdown connection.
- SAP HANA, Snowflake, SQL Server, and Trino Pushdown connections do not support the VARBINARY data type.
- The **valsrclimitui** admin limit, which controls the number of Validate Source records shown in the UI, now works as expected. For example, if **valsrclimitui** is set to 3, the UI shows 3 records from the Column Schema and 3 records from the Cell Panels.

> **Note** The **fpglimitui**, **histlimitui**, and **categoricallimitui** admin limits are not working as expected today. Further, any admin limit ending "ui", such as **valsrclimitui**, is intended only for updates made via the Admin Limits page of the Admin Console. Changes made to these admin limits via the command line will not be reflected.

Rules

- Validation is no longer performed automatically for the following data type rules:
  - DATATYPECHECK
  - DATECHECK
  - DOUBLECHECK
  - EMPTYCHECK
  - INTCHECK
  - NULLCHECK
  - STRINGCHECK
- The Rule Workbench now has a warning message when the preview limit exceeds the recommended maximum of 500 records.
- The maximum length of the name of a rule is now 250 characters.
- The Preview Breaks modal now indicates the name of the rule.

- The **Low**, **Medium**, and **High** buttons now correctly reflect your selection when you reopen the Rule Details dialog box after automatically changing a rule's points. Additionally, if you manually update the score, the button automatically adjusts to match the corresponding scoring range:
    - **Low**: 1-4 points
    - **Medium**: 5-24
    - **High**: 25-100

Admin Console

- Business unit names on the Business Unit page in the Admin Console cannot contain the special characters / (forward slash), ` (backtick), | (pipe), or $ (dollar sign). If you use any of these special characters, an error message appears, and the Edit Business Unit dialog box remains open until the special characters are removed.
- Data retention policies now apply to the following metastore tables:
    - alert_output
    - behavior
    - dataset_activity
    - dataset_schema_source
    - hint
    - validate_source
- The **Title** and **Story** fields of the User Story section of the User Profile page cannot contain the following special characters:
    - ‘ (single quote)
    - “ (double quote)
    - -- (double dash)
    - < (less than)
    - > (greater than)
    - & (ampersand)
    - / (forward slash)
    - ` (backtick)
    - | (pipe)
    - $ (dollar sign)

      If you use any of these special characters, an error message appears, and the Edit User Story dialog box remains open until the special characters are removed.

Fixes

Rules

- You can now apply rule filter queries to rules created using rule templates on Pushdown datasets.
- Duplicated data quality rules no longer appear after you import data assets using CMA.
- Rules now run without errors when you use a single link ID to archive break records with SQL Server Pushdown.

Profile

- Divide-by-zero errors and issues with empty histograms are now resolved. Calculations and visualizations now work as expected without interruptions.

Findings

- You can now export rule break records with accurate counts. The total number of rule break records for a rule now matches the number in the **Breaking Records** column of the exported file.

Integration

- When you rename a rule in Data Quality & Observability Classic that is part of a dataset integrated with Collibra Platform, additional Data Quality Rule assets are no longer created during the next integration. The existing Data Quality Rule asset is also no longer unintentionally set to a “suppressed” state during the next integration, and its name is updated to match the name in Data Quality & Observability Classic.

Release 2025.05

- Announcement
- New and improved
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.05: June 2, 2025
- Release notes publication date: May 6, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.05), Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.05, you must upgrade to Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.05 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.

- We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

1 Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic. For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

See what is changing[1]

For

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

New and improved

Warning We identified a known security vulnerability (CVE-2025-48734) in this release. We will address this issue in the upcoming 2025.05.1 patch.

Platform

- You can now authenticate users via Microsoft Azure Active Directory B2C when signing into Data Quality & Observability Classic.
- Only users with the ROLE_ADMIN role can now see the username and connection string of JDBC connections on the Connections page and in API responses, improving connection security.
- Metastore credentials are no longer stored in plain text when you create a DQ Job via a notebook. This improvement increases the security of your credentials.
- We enhanced the security of Data Category validation.
- We improved the security of our application.

Jobs

- Trino Pushdown jobs now support validate source.
- The Run Job buttons now have two enhancements:
  - The Run Job with Date button on the Jobs tab of the Findings page is now labeled **Select Run Date**.
  - The **Run Job** button on the metadata bar now includes a helpful tooltip on hover.
- SAP HANA connections now support the operators `=>`, `||`, and `&&` for scoped queries, the reserved word `'LIKE_REGEXTR'`, and the function `REPLACE` in scoped queries.
- The assignments queue is now disabled for the validate source activity on new installations of Data Quality & Observability Classic. To enable it, an admin can set **valsrcdisableaq** to "false" on the Admin Limits page.

Rules

- When Archive Break Records is disabled, you can now use NULL link IDs with Athena, BigQuery, Hive, Redshift, and SQL Server connections.
- When Archive Break Records is enabled, and a rule doesn't specify a link ID column but the job does, you can no longer archive break records in these scenarios, as Data Quality & Observability Classic no longer appends the link ID to the rule. When a rule does not contain the selected link ID, the break records for that rule will not be archived, and an exception will occur. The job will execute successfully, but the rule will not produce any archived break records, and the exception message will explain that the link ID is not part of the rule.
- Trino Pushdown jobs now support profiling and custom rules on columns of the ARRAY data type.
- The Rule Details dialog on the Rule Workbench page now has an optional **Purpose** field. This allows you to add details about Collibra Platform assets associated with your rule and see which assets should relate to your Data Quality Rule asset upon integration.

Alerts

- You can now set up Data Quality & Observability Classic to send data quality alerts to one or more webhooks, eliminating the dependency on SMTP for email alerts.
- You can now rename alerts on the Dataset Alerts page. When you rename an alert, the updated name is reflected in the alert_nm column of the alert_cond Metastore table. The updated name applies only to alerts generated in future job runs and does not affect historical data.
- In addition to the existing rule name condition variable for percentages, you can now include a rule name with a value identifier, such as `test_rule.score > 0`, to create condition alerts based on rule scores.

Profile

- When you retrain a breaking adaptive rule, negative values are now supported for lower and upper bounds, min, max, and mean.

Findings

- To enhance usability, we moved the data preview from the Rules results table into a dedicated modal, making it easier to navigate and view data.
    - The Actions button is now always visible on the right side of the Rules tab, and includes the following options:
        - **Archived Breaks**, previously called Rule Breaks, retains the same content and functionality.
        - **Preview Breaks**, previously embedded under the in the first column, shows all breaking records for a given rule. This option provides the same information that was previously available in the Rules tab and still requires the ROLE_DATASET_ACCESS role. Additionally, the Rule Break Preview modal reflects the preview limit of the rule.
    - Tooltips were added to various components on the Rules tab and Rule Break Preview modal.

Integration

- You can now use the **Map to primary column only** switch in the Connections step of the integration setup wizard. This allows you to map a rule to only the Column asset in Collibra Platform that corresponds to the primary column in Data Quality & Observability Classic.
- When you rename a custom rule in Data Quality & Observability Classic with an active integration with Collibra Platform, a new asset is no longer created in Collibra Platform. Instead, renamed rules reuse the Collibra Platform asset IDs associated with the previous rule name when their metadata is integrated.

APIs

- The **/v3/datasetDefs/template** API endpoint now returns the nested key-value pairs for shape settings and job schedule in the datasetDefs JSON object.

Admin Console

- You can now use the **previewlimit** setting on the Admin Limits page to define the maximum number of preview results shown in the Dataset Overview and Rule Workbench. The default value is 250.

  > **Warning** Large **previewlimit** values can negatively impact performance.

Fixes

Platform

- All datasets are now available when you run a dataset rule fetch with dataset security disabled for the admin user account.
- FIPS-enabled standalone installations of Data Quality & Observability Classic now support SAML authentication.

Jobs

- When you remove a table from a manually triggered or scheduled job, you now receive a more descriptive error stating that the table does not exist instead of a generic syntax error.
- You can now use Amazon S3 as a secondary dataset name for a rule query.
- You can now rerun a job for a selected date using the **Run DQ Job** button on the metadata bar of the Findings page.
- The command line now retains the created query, including the `${rd}` parameter, when you run a job using the Run DQ Job button on the metadata bar.
- The controller response for queries to the alert_cond table in the Metastore now maps internal objects correctly.
- The Profile page now correctly shows the top 5 TopN and BottomN shape results for Pullup jobs.
- When you add a transform to a job with histogram enabled or set to auto, the job processes as expected, and aliased columns display correctly on the Profile page.

Rules

- You can again save rules on Pushdown jobs that exceed ⅓ of a buffer page.
- Pushdown rules that use stat variables now run with the correct information from the current job, rather than using data from previous job runs.
- The DOUBLECHECK rule on PostgreSQL and Redshift Pullup jobs now flags rows with negative double values as valid.
- The Result Preview in the Rule Workbench no longer produces an error when you use out-of-the-box templates.

Profile

- You can now compare the Baseline and Run values for the "Filled rate" when the Quality switch is enabled for a column on the Profile report.

Findings

- Pushdown jobs run on Snowflake connections now show as failed on the Findings page if a password retrieval issue occurs.

Dataset Manager

- When you apply an option from the Actions drop-down list, such as delete a dataset, the correct dataset now has the action applied to it.

Admin Console

- When you select a submenu option from the Admin Console menu, the submenu section now remains open.

Release 2025.04

- Announcement
- New and improved

- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.04: April 28, 2025
- Release notes publication date: April 2, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.04), only the Java 17 build profile of Data Quality & Observability Classic contains all new and improved features and bug fixes listed in the release notes. The Java 8 and 11 build profiles for Standalone installations contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases.

Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release for Java 17 build profiles:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.04, you must upgrade to Java 17 and Spark 3.5.3 if you did not already do so in the 2025.02 or 2025.03 release.

- If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
- Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.04 with Java 17.
- If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

While this release contains Java 8, 11, and 17 builds of Data Quality & Observability Classic for Standalone installations, it is the final release to contain Java 8 and 11 builds and Spark versions older than 3.5.3. Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic.

For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing[1]

For

[1]

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

New and improved

Platform

- The Platform Path of the SQL Assistant for Data Quality feature now uses the Gemini AI model version **gemini-1.5-pro-002**.

  Important To continue using the Platform Path of the SQL Assistant for Data Quality feature, you must upgrade to Data Quality & Observability Classic 2025.04.

- FIPS-enabled standalone installations of Data Quality & Observability Classic now support SAML authentication.
- The DQ agent now automatically restores itself when the Metastore reconnects after a temporary disconnection due to maintenance or a glitch.
- We implemented robots.txt to prevent all search engine crawlers from indexing any Data Quality & Observability Classic content.

Jobs

- We are pleased to announce that Oracle Pushdown is now available for preview testing.
- When you add a timeslice to a timeUUID data type column in a Cassandra dataset, an unsupported data type error message now appears.
- Dremio, Snowflake, and Trino Pullup jobs now support common table expressions (CTE) for parallel JDBC processing.
- You can now archive the break records of shapes from Trino Pushdown jobs.
- Link IDs for exact match duplicates are no longer displayed on the findings page.
- We improved the insertion procedure of validate source findings into the Metastore.

Rules

- When you add or edit a rule with an active status, SQL syntax validation now runs automatically when you save it. If the rule passes validation, it saves as expected. If validation fails, a dialog box appears, asking whether you want to continue saving with errors. Rules with an inactive status save without validation checks.
- When a rule condition exceeds 2 lines, only the first 2 lines are shown in the Condition column of the Rules tab on the Findings page. You can click "more..." or hover over the cell to show the full condition.

Profile

- The Profile page now correctly shows the top 5 TopN and BottomN shape results for Pullup jobs.
- When there are only two unique string values, the histogram on the Profile page now shows them correctly.

Findings

- To improve page alignment across the application, the Findings page now has a page title.

Alerts

- Rule score-based alerts now respect rule tolerance settings. Alerts are suppressed when the rule break percentage falls within the tolerance threshold.

Dataset Manager

- If you don't have the required permissions to perform certain tasks in the Dataset Manager, such as updating a business unit, an error message now appears when you attempt the action.

Scorecards

- You can now delete scorecards with names that contain trailing empty spaces, such as "scorecard " and "test scorecard ".

Integration

- You can now search with trailing spaces when looking up a community for tenant mapping in the Data Quality & Observability Classic and Collibra Platform integration configuration.

APIs

- You now need the **ROLE_ADMIN** or **ROLE_ADMIN_VIEWER** role to access the **/v2/getdatasetusagecount** API endpoint.

Fixes

Platform

- We have improved the security of our application.
- The data retention process now works as expected for all tenants in multi-tenant instances.
- Permissions errors for SAML users without the **ROLE_PUBLIC** role are now resolved. You no longer need to assign **ROLE_PUBLIC** to users who already have other valid roles.

Jobs

- Native rules, secondary dataset rules, and scheduled jobs on Databricks Pullup connections authenticated via EntraID Service Principal now run as expected.
- Pushdown jobs scheduled to run concurrently now process the correlation activity correctly.
- Pushdown jobs with queries that contain new line characters now process correctly, and the primary table from the query is shown in the Dataset Manager.

Findings

- The confidence calculations for numerical outliers in Pullup jobs have been updated for negative values. Positive value confidence calculations and Pushdown calculations remain unchanged.

Alerts

- Conditional alerts now work as expected when based on rules with names that start with numbers.

Integration

- The default setting for the integration schema, table, and column recalculation service (**DQ_DGC_MAPPING_STATS_SCHEDULER_ENABLED**) is now **false**, reducing unnecessary database activity. You can enable the service or call it through the API when needed.

  > **Important** From versions 2024.11 to 2025.03 of Data Quality & Observability Classic, if you don't want queries from the recalculation of mapped and unmapped stats of total entities to run, set the **DQ_DGC_MAPPING_STATS_SCHEDULER_ENABLED** to **false** in the owl-env.sh or Web ConfigMap. You can keep **DQ_INTEGRATION_SCHEDULER_ENABLED** set to **true**.
  > For more information, go to the Collibra Support center.

- The Quality tab for database assets is now supported in out-of-the-box aggregation path configurations.
- The auto map feature now correctly maps schemas that contain only views.
- Dimension configuration for integration mapping no longer shows duplicate Data Quality & Observability Classic dimensions from the dq_dimension table.

API

- When you copy a rule with an incorrect or non-existent dataset name using the **v3/rules/copy** API, an error message now specifies the invalid dataset or rule reference. This prevents invalid references in the new dataset.

Release 2025.04 (Java 8 and 11 only)

- Announcement
- Enhancements
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.04: April 28, 2025
- Release notes publication date: April 2, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.04), only the Java 17 build profile of Data Quality & Observability Classic contains all new and improved features and bug fixes listed in the 2025.04 release notes. The Java 8 and 11 build profiles for Standalone installations contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases.

While this release contains Java 8, 11, and 17 builds of Data Quality & Observability Classic for Standalone installations, it is the final release to contain Java 8 and 11 builds and Spark versions older than 3.5.3. Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic.

For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing[1]

1

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

Enhancements

Platform

- When an externally managed user enables assignment alert notifications and enters an email address on their user profile, findings assigned to that email address now receive alert notifications.
- **SAML_ENTITY_BASEURL** is now a required property for SAML authentication.
- To improve the security of our application, we upgraded SAML. As part of the upgrade, the following SAML Authentication are no longer used:
  - **SAML_TENANT_PROP_FROM_URL_ALIAS**
  - **SAML_METADATA_USE_URL**
  - **SAML_METADATA_TRUST_CHECK**
  - **SAML_INCLUDE_DISCOVERY_EXTENSION**
  - **SAML_SUB_DOMAIN_REPLACE_SOURCE**
  - **SAML_SUB_DOMAIN_REPLACE_TARGET**
  - **SAML_MAX_AUTH_AGE**

    > **Note** If the **SAML_LB_EXISTS** property is set to **true** and **SAML_LB_INCLUDE_PORT_IN_REQUEST** set to **false**, you may need to update **SAML_ENTITY_BASEURL** to include the port in the URL. The **SAML_ENTITY_BASEURL** should match the IdP's ACS URL.
    >
    > Additionally, we recommend using the Collibra DQ UI when signing in with SAML, as SP-initiated sign-on is not fully supported in Collibra DQ.

    > **Tip** We recommend synchronizing the clocks of the Identity Provider (IdP) and Service Provider (Collibra DQ) using Network Time Protocol (NTP) to prevent authentication failures caused by significant clock skew.

Connections

- Private endpoints are now supported for Azure Data Lake Storage (Gen2) (ABFSS) key- and service principal-based authentication and Azure Blob Storage (WASBS) key-based authentication using the **cloud.endpoint=<endpoint>** driver property. To do this, add

cloud.endpoint=<endpoint> to the Driver Properties field on the Properties tab of an ABFSS or WASBS connection template. For example, **cloud.endpoint=microsoftonline.us**.

- You can now limit the schemas that are displayed in a JDBC connection in Explorer. This enhancement helps you manage usage and maintain security by restricting access to only the necessary schemas. (idea #CDQ-I-152)

  > **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.

- Hive connections now use the driver class name, `com.cloudera.hive.jdbc.HS2Driver`. Additionally, only Hive versions 2.6.25 and newer are supported.

  > **Warning** If you have a Standalone installation of Data Quality & Observability Classic and leverage Hive connections, you must upgrade to the latest 2.6.25 Hive driver to use Data Quality & Observability Classic 2025.02.

Jobs

- The **Jobs** tab on the Findings page now includes two button options:
  - **Run Job from CMD/JSON** allows you to run a job with updates made in the job's command line or JSON. This option is **Run Job from JSON** in Pushdown mode since the command line option is not available.
  - **Run Job with Date** allows you to select a specific run date.

    > **Note** The **Run DQ Job** button on the metadata bar retains its functionality, allowing you to rerun a job for the selected date.

- DQ Jobs on Trino Pushdown connections now allow you to select multiple link ID columns when setting up the dupes monitor.

Rules

- We are delighted to announce that rule filtering is now generally available. To use rule filtering, an admin needs to set RULE_FILTER to TRUE in the Application Configuration.
- We are also pleased to announce that rule tolerance is generally available.
- You can now define a DQ Dimension when you create a new template or edit an existing one to be applied to all new custom rules created using this template. Additionally, a Dimension column is now shown on the Templates page.
- There is now a dedicated break_msg column in the rule_output Metastore table, which shows the break message when a rule break occurs.

Dataset Manager

- You can now add meta tags up to 100 characters long from the Edit option under the Actions menu on the Dataset Manager. (idea #CDQ-I-74)

Fixes

Platform

- You can now use the Bouncy Castle FIPS provider for Standalone deployments of Data Quality & Observability Classic.
- To improve the security of our application, Data Quality & Observability Classic now supports a Content Security Policy (CSP).
- We have improved the security of our application.

Jobs

- When a scheduled job runs, only the time it is scheduled to run displays in the Scheduled Time column on the Jobs page.
- When you add a `-conf` setting to the agent configuration of an existing job and rerun it, the command line no longer includes duplicate `-conf` parameters.
- When you edit and re-run a DQ Job with a source-to-target mapping of data from two different data sources, the `-srcds` parameter on the command line now correctly

contains the `src_` prefix before the source dataset, and the DQ Job re-runs successfully.
- When you include a custom column via the source query of a source-to-target mapping, the validate overlay function no longer fails with an error message.
- When you clone a dataset from Dataset Manager, Explorer, or the Job tab on Findings, the Job Name field and the Command Line in the Review step of Explorer now correctly reflect the "temp_cloned_dataset" naming convention for cloned datasets.
- DQ Jobs with the same run date can no longer run concurrently.
- Pullup DQ Jobs with names that include dashes, such as "sample-dataset," no longer remain stalled in the "staged" activity when you rerun them.
- The correlation and histogram activities no longer cause DQ Jobs on Databricks Pushdown connections to fail sporadically.
- The correlation activity now displays correctly in the UI and no longer causes DQ Jobs on Pushdown connections to fail.

Rules

- When you use Freeform SQL queries with `LIKE %` conditions, for example, `SELECT * FROM @public.test where testtext LIKE '%a, c%'`, they now return the expected results.
- SQL Server job queries where the table name is escaped with brackets, for example, `select * from dbo.[Table]`, now process correctly when the job runs.
- The Rule Results Preview button is no longer disabled when the API call to gather the Livy status fails due to an invalid or non-existent session. The API call now correctly manages the cache for Livy sessions terminated due to idle timeout.
- @t1 rules on file-based datasets with a row filter now return only the rows included in the filter.
- @t1 rules on Databricks datasets no longer return a NullPointerException error.
- The contents of the Export Rules with Details now include data from the new Tolerance rule setting.
- Data type rules now evaluate the contents of each column, not just the data type of each column to ensure the correct breaks are returned.
- When you run Rule Discovery on a dataset with the “Money” Data Class in the Data Category, the job no longer returns a syntax error when it runs.
- SQL reserved keywords included in string parsing are now correctly preserved in their original case.

- When you update the name of a rule and an alert is configured on it, the alert will now show the updated name when sent.
- When you update the adaptive level or pass value option in the Change Detection dialog box of an adaptive rule, you must now retrain it by clicking **Retrain** on the Behaviors tab of the Findings page.

Findings

- Labels under the data quality score meter are now highlighted correctly according to the selected time zone of the dataset.

Alerts

- When you set a breaking rule as passing, a rule status alert for rules with breaking statuses no longer sends erroneous error messages for that rule.

Connections

- When you substitute a PWD value as a sensitive variable on the Variables tab of a Databricks connection template, the sensitive variable in the connection URL is now set correctly for source-to-target mappings where Databricks is the source dataset.
- Editing an existing remote file job no longer results in an error.
- Teradata connections now function properly without requiring you to manually add the **STRICT_NAMES** driver property.

APIs

- The **/v3/jobs/<jobId>/breaks/rules** endpoint no longer returns a 500 error when using a valid jobId. Instead, it now returns empty files when no results are found for exports without findings.
- When you run a job using the **/v3/jobs/run** API that was previously exported and imported with **/v3/datasetDefs**, the Shape settings from the original job now persist in the new job.
- When you schedule a Job to run monthly or quarterly, the `jobSchedule` object in the **/v3/datasetDefs** endpoint now reflects your selection.

- When you run the **/v3/rules/{dataset}/{ruleName}/{runId}/breaks** endpoint when Archive Rules Break Records is enabled, break records are now retrieved from the source system instead of the Metastore.
- Meta tags are now correctly applied to new datasets that are created with the **/v3/datasetDefs** endpoint.
- When an integration import job fails, the **/v3/dgc/integrations/jobs** endpoint now returns the correct "failed" status. Additionally, the integration job status "ignored" is now available.

Integration

- You can now use the automapping option to map schemas, tables, and columns when setting up an integration between Data Quality & Observability Classic and Collibra Platform in single-tenant Data Quality & Observability Classic environments.
- An invalid RunID no longer returns a successful response when using a Pushdown dataset with the **/v3/jobs/run** endpoint.
- Parentheses in column names are no longer replaced with double quotes when mapped to Collibra Platform assets. This change allows automatic relations to be created between Data Quality Rule and Column assets in Collibra Platform.

Release 2025.03

- Announcement
- Enhancements
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.03: March 31, 2025
- Release notes publication date: March 4, 2025

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In this release (2025.03), Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - To install Data Quality & Observability Classic 2025.03, you must upgrade to Java 17 and Spark 3.5.3 if you have not already done so in the 2025.02 release.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.03 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
  - We encourage you to migrate to a Kubernetes installation to improve the scalability and ease of future maintenance.

The April 2025 (2025.04) release will contain Java 8, 11, and 17 versions of Data Quality & Observability Classic. This will be the final release to contain Java 8 and 11 builds and

Spark versions older than 3.5.3, and will include feature enhancements and bug fixes from the 2025.02 release and critical bug fixes from the 2025.03 and 2025.04 releases. Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be made for Java 8 and 11 versions of Data Quality & Observability Classic. For a breakdown of Java and Spark availability in current and upcoming releases, click "See what is changing" below.

See what is changing

For

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes[1]	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

> more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

Enhancements

Platform

- On April 9, 2025, Google will deprecate the Vertex text-bison AI model, which SQL Assistant for Data Quality uses for the "preview path" option. To continue using SQL Assistant for Data Quality, you must switch to the "platform path," which requires an integration with Collibra Platform. For more information about how to configure the platform path, go to Prerequisites.
- We removed support for Kafka streaming.

Connections

- Private endpoints are now supported for Azure Data Lake Storage (Gen2) (ABFSS) key- and service principal-based authentication and Azure Blob Storage (WASBS) key-based authentication using the **cloud.endpoint=<endpoint>** driver property. To do this, add **cloud.endpoint=<endpoint>** to the Driver Properties field on the Properties tab of an ABFSS or WASBS connection template. For example, **cloud.endpoint=microsoftonline.us**.
- Trino now supports parallel processing in Pullup mode. To enable this enhancement, the Trino driver has been upgraded to version **1.0.50**.

Jobs

- The **Jobs** tab on the Findings page now includes two button options:
  - **Run Job from CMD/JSON** allows you to run a job with updates made in the job's command line or JSON. This option is **Run Job from JSON** in Pushdown mode since the command line option is not available.
  - **Run Job with Date** allows you to select a specific run date.

> **Note** The **Run DQ Job** button on the metadata bar retains its functionality, allowing you to rerun a job for the selected date.

- The order of link IDs in the **rule_breaks** and **opt_owl** Metastore tables for Pushdown jobs is now aligned.
- The options to archive the break records of associated monitors in the Explorer settings dialog box of a Pushdown job are now disabled when the Archive Break Records option is disabled at the connection-level.
- We updated the logic of the maximum global job count to ensure it only increases, rather than fluctuating based on the maximum count of the last run job's tenant. This change allows tenants with lower maximum job counts to potentially run more total jobs while still enforcing the maximum connections for individual jobs. Over time, the global job count will align with the highest limit among all tenants.
- You can now archive the break records of shapes from SAP HANA and Trino Pushdown jobs.
- You can now use the new `"behaviorShiftCheck"` element in the JSON payload of jobs on Pullup connections. This allows you to enable or disable the shift metric results of Pullup jobs, helping you avoid misleading mixed data type results in string columns. By default, the `"behaviorShiftCheck"` element is enabled (set to `true`). To disable it, use the following configuration: `"behaviorShiftCheck": false`.

Rules

- You can now set the **MANDATORY_PRIMARY_RULE_COLUMN** setting to **TRUE** from the Application Configuration Settings page of the Admin Console to require users to select a primary column when creating a rule. This requirement is enforced when a user creates a new rule or saves an existing rule for the first time after the setting is enabled. Existing rules are not affected automatically.
- The names of Template rules can no longer include spaces.
- CSV rule export files now include a **Filter Query** column when a rule filter is defined. If no filter is used, the column remains empty. The **Condition** column has been renamed to **Rule Query** to better distinguish between rule and filter queries. Additionally, the **Passing Records** column now shows the correct values.

- You can now apply custom dimensions added to the **dq_dimension** table in the metastore to rules from the Rule Details dialog box on the Rule Workbench. These custom dimensions are also included in the Column Dimension Report.
- Livy caching now uses a combination of username and connection type instead of just the username. This improvement allows you to seamlessly switch between connections to access features such as retrieving the run results previews for rules or creating new jobs for remote file connections, without manually terminating sessions.

  > **Note** Manually terminating a Livy session will still end all sessions associated with that user.

Findings

- You can now work with the Findings page in full-screen view.

Scorecards

- You now receive a helpful error message in the following scenarios:
  - Create a scorecard without including any datasets.
  - Update a scorecard and remove all its datasets.
  - Add or update a scorecard page with a name that already exists.

Fixes

Platform

- We have improved the security of our application.
- We mitigated the risk of SQL injection vulnerabilities in our application.
- Helm Charts now include the external JWT properties required to configure an externally managed JWT.

Jobs

- Google BigQuery jobs no longer fail during concurrent runs.
- When you add a `-conf` setting to the agent configuration of an existing job and rerun it, the command line no longer includes duplicate `-conf` parameters.
- When you expand a Snowflake connection in Explorer, the schema is now passed as a parameter in the query. This ensures the Generate Report function loads correctly.
- Record change detection now works as expected with Databricks Pushdown datasets.
- When you select a Parquet file in the Job Creator workflow, the **Formatted view** tab now shows the file's formatted data.
- When you edit a Pullup job from the **Command Line**, **JSON**, or **Query** tab, the changes initially appear only on the tab where you made the edits. After you rerun the job, the changes are reflected across all three tabs.
- The Dataset Overview now performs additional checks to validate queries that don't include a `SELECT` statement.

Rules

- When you update the adaptive level or pass value option in the Change Detection dialog box of an adaptive rule, you must now retrain it by clicking **Retrain** on the Behaviors tab of the Findings page.
- @t1 rules on file-based datasets with a row filter now return only the rows included in the filter.
- @t1 rules on Databricks datasets no longer return a NullPointerException error.
- When you run Rule Discovery on a dataset with the "Money" Data Class in the Data Category, the job no longer returns a syntax error when it runs.

Findings

- We updated the time zone library. As a result, some time zone options, such as "US/Eastern," have been updated to their new format. Scheduled jobs are fully compatible with the corresponding time zones in the new library. If you need to adjust a time zone, you must use the updated format. For example, "US/Eastern" is now "America/New_York."

- Labels under the data quality score meter are now highlighted correctly according to the selected time zone of the dataset.

Alerts

- You no longer receive erroneous job failure alerts for successful runs. Additional checks now help determine whether a job failed, improving the accuracy of job status notifications.
- You can now consistently select or deselect the **Add Rule Details** option in the Condition Alert dialog box.

Reports

- The link to the Dataset Findings documentation topic on the Dataset Findings report now works as expected.

Connections

- Editing an existing remote file job no longer results in an error.
- Teradata connections now function properly without requiring you to manually add the **STRICT_NAMES** driver property.

APIs

- When you run a job using the **/v3/jobs/run** API that was previously exported and imported with **/v3/datasetDefs**, the Shape settings from the original job now persist in the new job.
- Bearer tokens generated in one environment using the **/v3/auth/signin** endpoint (for local users) or the **/v3/auth/oauth/signin** endpoint (for OAuth users) are now restricted to that specific Data Quality & Observability Classic environment and cannot be used across other environments.
- We improved the security of our API endpoints.

Integration

- You can now use the automapping option to map schemas, tables, and columns when setting up an integration between Data Quality & Observability Classic and Collibra Platform in single-tenant Data Quality & Observability Classic environments.
- The Quality tab now correctly shows the data quality score when the head asset of the starting relation type in the aggregation path is a generic asset or when the starting relation type is based on the co-role instead of the role of the relation type.
- Parentheses in column names are no longer replaced with double quotes when mapped to Collibra Platform assets. This change allows automatic relations to be created between Data Quality Rule and Column assets in Collibra Platform.

Release 2025.02

- Announcement
- Enhancements
- Fixes

Release Information

- Release date of Data Quality & Observability Classic 2025.02: February 24, 2025
- Release notes publication date: January 20, 2025

Announcement

> **Important**
>
> As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.
>
> **In this release (2025.02) and the March (2025.03) release**, Data Quality & Observability Classic is only available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in this release:

- **Kubernetes installations**
  - Kubernetes containers automatically contain Java 17 and Spark 3.5.3.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
- **Standalone installations**
  - You must upgrade to Java 17 and Spark 3.5.3 to install Data Quality & Observability Classic 2025.02.
  - If you use custom drivers, ensure they are compatible with Java 17 and Spark 3.5.3.
  - Follow the latest steps to upgrade to Data Quality & Observability Classic 2025.02 with Java 17.
  - If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `false` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.
  - We encourage you to migrate to a Kubernetes instal scalability and ease of future maintenance.

The April 2025 (2025.04) release will contain Java 8, 11, and 17 v & Observability Classic. This will be the final release to contain Ja Spark versions older than 3.5.3, and will include feature enhance from the 2025.02 release and critical bug fixes from the 2025.0 Between 2025.05 and 2025.07, only critical and high-priority bu Java 8 and 11 versions of Data Quality & Observability Classic. Fo and Spark availability in current and upcoming releases, click "Se below.

See what is changing[1]

For

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	Standalone only
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes		

Important
The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases.

Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

[1]

more information, go to the Data Quality & Observability Classic Java Upgrade FAQ.

Enhancements

Platform

- When an externally managed user enables assignment alert notifications and enters an email address on their user profile, findings assigned to that email address now receive alert notifications.
- **SAML_ENTITY_BASEURL** is now a required property for SAML authentication.
- To improve the security of our application, we upgraded SAML. As part of the upgrade, the following SAML Authentication are no longer used:
  - **SAML_TENANT_PROP_FROM_URL_ALIAS**
  - **SAML_METADATA_USE_URL**
  - **SAML_METADATA_TRUST_CHECK**
  - **SAML_INCLUDE_DISCOVERY_EXTENSION**
  - **SAML_SUB_DOMAIN_REPLACE_SOURCE**
  - **SAML_SUB_DOMAIN_REPLACE_TARGET**
  - **SAML_MAX_AUTH_AGE**

    Note If the **SAML_LB_EXISTS** property is set to **true** and **SAML_LB_INCLUDE_PORT_IN_REQUEST** set to **false**, you may need to update **SAML_ENTITY_BASEURL** to include the port in the URL. The **SAML_ENTITY_BASEURL** should match the IdP's ACS URL.

    Additionally, we recommend using the Collibra DQ UI when signing in with SAML, as SP-initiated sign-on is not fully supported in Collibra DQ.

    Tip We recommend synchronizing the clocks of the Identity Provider (IdP) and Service Provider (Collibra DQ) using Network Time Protocol (NTP) to prevent authentication failures caused by significant clock skew.

Connections

- You can now limit the schemas that are displayed in a JDBC connection in Explorer. This enhancement helps you manage usage and maintain security by restricting access to only the necessary schemas. (idea #CDQ-I-152)

  > **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.

- Hive connections now use the driver class name, `com.cloudera.hive.jdbc.HS2Driver`. Additionally, only Hive versions 2.6.25 and newer are supported.

  > **Warning** If you have a Standalone installation of Data Quality & Observability Classic and leverage Hive connections, you must upgrade to the latest 2.6.25 Hive driver to use Data Quality & Observability Classic 2025.02.

Jobs

- DQ Jobs on Trino Pushdown connections now allow you to select multiple link ID columns when setting up the dupes monitor.

Rules

- We are delighted to announce that rule filtering is now generally available. To use rule filtering, an admin needs to set RULE_FILTER to TRUE in the Application Configuration.
- We are also pleased to announce that rule tolerance is generally available. However, the RULE_TOLERANCE_BETA setting must remain set to TRUE in the Application Configuration until the 2025.04 version.
- You can now define a DQ Dimension when you create a new template or edit an existing one to be applied to all new custom rules created using this template. Additionally, a Dimension column is now shown on the Templates page.
- There is now a dedicated break_msg column in the rule_output Metastore table, which shows the break message when a rule break occurs.

Dataset Manager

- You can now add meta tags up to 100 characters long from the Edit option under the Actions menu on the Dataset Manager. (idea #CDQ-I-74)

Fixes

Platform

- We have improved the security of our application.
- To improve the security of our application, Data Quality & Observability Classic now supports a Content Security Policy (CSP).

Jobs

- When a scheduled job runs, only the time it is scheduled to run displays in the Scheduled Time column on the Jobs page.
- When you edit and re-run a DQ Job with a source-to-target mapping of data from two different data sources, the `-srcds` parameter on the command line now correctly contains the `src_` prefix before the source dataset, and the DQ Job re-runs successfully.
- When you include a custom column via the source query of a source-to-target mapping, the validate overlay function no longer fails with an error message.
- When you clone a dataset from Dataset Manager, Explorer, or the Job tab on Findings, the Job Name field and the Command Line in the Review step of Explorer now correctly reflect the "temp_cloned_dataset" naming convention for cloned datasets.
- DQ Jobs with the same run date can no longer run concurrently.
- Pullup DQ Jobs with names that include dashes, such as "sample-dataset," no longer remain stalled in the "staged" activity when you rerun them.
- The correlation and histogram activities no longer cause DQ Jobs on Databricks Pushdown connections to fail sporadically.
- The correlation activity now displays correctly in the UI and no longer causes DQ Jobs on Pushdown connections to fail.

Rules

- When you use Freeform SQL queries with `LIKE %` conditions, for example, `SELECT * FROM @public.test where testtext LIKE '%a, c%'`, they now return the expected results.
- SQL Server job queries where the table name is escaped with brackets, for example, `select * from dbo.[Table]`, now process correctly when the job runs.
- The Rule Results Preview button is no longer disabled when the API call to gather the Livy status fails due to an invalid or non-existent session. The API call now correctly manages the cache for Livy sessions terminated due to idle timeout.
- The contents of the Export Rules with Details now include data from the new Tolerance rule setting.
- Data type rules now evaluate the contents of each column, not just the data type of each column to ensure the correct breaks are returned.
- SQL reserved keywords included in string parsing are now correctly preserved in their original case.
- When you update the name of a rule and an alert is configured on it, the alert will now show the updated name when sent.

Alerts

- When you set a breaking rule as passing, a rule status alert for rules with breaking statuses no longer sends erroneous error messages for that rule.

Connections

- When you substitute a PWD value as a sensitive variable on the Variables tab of a Databricks connection template, the sensitive variable in the connection URL is now set correctly for source-to-target mappings where Databricks is the source dataset.

APIs

- The **/v3/jobs/<jobId>/breaks/rules** endpoint no longer returns a 500 error when using a valid jobId. Instead, it now returns empty files when no results are found for exports without findings.

- When you schedule a Job to run monthly or quarterly, the `jobSchedule` object in the **/v3/datasetDefs** endpoint now reflects your selection.
- When you run the **/v3/rules/{dataset}/{ruleName}/{runId}/breaks** endpoint when Archive Rules Break Records is enabled, break records are now retrieved from the source system instead of the Metastore.
- Meta tags are now correctly applied to new datasets that are created with the **/v3/datasetDefs** endpoint.
- When an integration import job fails, the **/v3/dgc/integrations/jobs** endpoint now returns the correct "failed" status. Additionally, the integration job status "ignored" is now available.

Integration

- An invalid RunID no longer returns a successful response when using a Pushdown dataset with the **/v3/jobs/run** endpoint.

Release 2025.01

- Announcement
- Enhancements
- Fixes
- Features in preview

Release Information

- Release date of Data Quality & Observability Classic 2025.01: January 27, 2025
- Release notes publication date: December 31, 2024

Announcement

> **Important**
>
> As a security measure, we are announcing the end of life of the Java 8 and 11 versions of

Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

1 In the **February 2025 (2025.02)** release, Data Quality & Observability Classic will only be available on Java 17 and Spark 3.5.3. Depending on your installation of Data Quality & Observability Classic, you can expect the following in the 2025.02 release:

- **Kubernetes installations**: Kubernetes containers will automatically contain Java 17 and Spark 3.5.3. You may need to update your custom drivers and SAML keystore to maintain compatibility with Java 17.
- **Standalone installations**: You must upgrade to Java 17 and Spark 3.5.3 to install Data Quality & Observability Classic 2025.02. Additional upgrade guidance will be provided upon the release date. We encourage you to migrate to a Kubernetes installation, to improve the scalability and ease of future maintenance.

The April 2025 (2025.04) release will have Java 8, 11, and 17 versions of Data Quality & Observability Classic and will be the final release to contain new features on Java 8 and 11. Between 2025.05 and 2025.07, only critical and high-priority bug fixes will be included in the Java 8 and 11 versions of Data Quality & Observability Classic.

See what is changing

Additional

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

> details on driver compatibility, SAML upgrade procedures, and more will be available alongside the 2025.02 release.
>
> For more information, visit the Data Quality & Observability Classic Java Upgrade FAQ.

Enhancements

Platform

- When you access Swagger from the Admin Console or the ⓘ in the upper right corner of any Data Quality & Observability Classic page, it now opens in a separate tab or window.

Connections

- You can now authenticate Amazon S3 connections using OAuth2 with an Okta principal as a service account. This enhancement simplifies the authentication process and improves security.
- When you add a connection, you can now use commas and equals characters in values in the Values option on the Variables tab. For example, **val1=,val2** is now a supported variable value.
    - Additionally, the Driver Properties on the Properties tab now use a JSON array format instead of a comma-separated string. For example, **[{"name":"prop1","value":"val1"},{"name":"prop2","value":"val2"}]** is now the supported format for properties.

Jobs

- Snowflake Pushdown connections now support source-to-target validation for datasets from the same Snowflake Pushdown connection.
- Snowflake Pushdown connections now support all regex special characters in column names except for square brackets **[ ]**.

Rules

- When you hover over the break percentage (Perc) value on the Rules tab of the Findings page, a new popover message now shows the full value. This update improves visibility and makes it easier to view detailed information.
- Tooltips are now available for the Points and Perc columns on the Rules tab of the Findings page.
- You now see a descriptive error message if your data class rule fails to save because of missing required fields or a duplicate name when using the data class rule creator.
- You no longer need to rerun a DQ Job after renaming a rule for its new name to be reflected across all relevant pages.

APIs

- The **POST /v3/datasetdefs** endpoint now supports creating outlier and/or pattern records directly by including the outliers or patterns element in the JSON payload (dataset name is required).
    - If an ID is not provided, a new record is created.
    - If an ID is provided and matches an existing record, the existing record will be updated.
    - Alternatively, you can use the existing **/v2/upsertPatternOpt** and **/v2/upsertOutlierOpt** endpoints to create records and then reference their IDs when you use the **PUT /v3/datasetdefs** endpoints for updates.
- The ControllerAdmin API now requires **ROLE_ADMIN** or **ROLE_ADMIN_VIEWER** to use the endpoints:
    - /gettotalcost
    - /gettotalbytes
- The ControllerSecurity API now requires **ROLE_ADMIN**, **ROLE_USER_MANAGER**, or **ROLE_OWL_ROLE_MANAGER** to use the endpoints:
    - /getalldbuserdetails
    - /getdbuserwithroles
    - getAllAdGroupsandRoles
- The ControllerUsageStats API now requires **ROLE_ADMIN** or **ROLE_ADMIN_VIEWER** to use the endpoints:

    - /getuserprofilecount
    - /getcolumnscount
    - /getusercount
    - /getdatasetcount
    - /getalertcount
    - /getemailcount
    - /getfailingdataset
    - /getpassingdataset
    - /getowlcheckcount
    - /getowlusagelast30days
    - /getowlusagemonthly
    - /getrulecount
    - /gettotalowlchecks
- We deprecated the **GET /deleteappconfig**. As an alternative, you can use **DELETE /deleteappconfig** instead.
- We deprecated the **GET /deleteadminconfig**. As an alternative, you can use **DELETE /deleteadminconfig** instead.

Fixes

Jobs

- You can again preview columns on SAP HANA tables with periods (**.**) in their names.
- When you run DQ Jobs from the command line, the command `-srcinferschemaoff`, which prevents Data Quality & Observability Classic from inferring the schema in the validate source activity, now works as expected when files are in the source.
- When **DB_VIEWS_ON** is set to **TRUE** on the Application Configuration page of the Admin Console, the **Include Views** option is now also applied when using the Mapping step for a Database connection.

Findings

- The data preview of DQ Jobs created on file data sources that use pipe delimiters (**|**) now displays correctly on the Profile and Shapes tab of the Findings page.
- DQ Jobs with the validate source layer enabled no longer return an error when you open their Findings pages.

- When you edit the scheduled run time of a DQ Job from the Findings page, the updated scheduled time now saves correctly.
- We removed the sort icon for the Breaking Records and Passing Records columns on the Rules tab of the Findings page, because it was not functional.
- When you click the Run ID link of a dataset on the Assignments Queue, the Findings page now opens to the finding type associated with the Run ID. For example, a Rule finding links to the Rules tab on the Findings page.

Profile

- When you change an Adaptive Rule from automatic to manual scoring on the Profile page, the points value now saves correctly.
- When you manually set the lower and upper bounds of an Adaptive Rule, you can now use decimal values, such as 10.8.
- When you use the search function within the Profile page to search for a partial string of a column name, such as "an" to return results for the columns "exchange" and "company," the correct number of results are now returned. Previously, a maximum of 10 results were returned.
- When a column contains special characters, the column stats section on the Profile page now formats the results correctly, ensuring better readability.

Rules

- When you use the Archive Break Records feature and the link ID column is an alias column, Data Quality & Observability Classic now properly identifies the link ID column to ensure the rule processes correctly.
- Complex DQ Job query joins containing SAP HANA reserved words in SQL queries are now supported. This ensures that the query compilation processes successfully. The following SAP HANA reserved words are supported in SQL queries:
  - `"union", "all", "case", "when", "then", "else", "end", "in", "current_date", "current_timestamp", "current_time", "weekday", "add_days", "add_months", "add_years", "days_between", "to_timestamp", "quarter", "year", "month", "to_char"`

- User-defined rules against DQ Jobs on Snowflake Pushdown connections no longer return a syntax error when "`@`" is present anywhere in the query.
- A rule on a DQ Job with Archive Breaking Records enabled now processes successfully without returning an exception message, even when it does not reference all link IDs in its `SELECT` statement.
- We temporarily removed the ability to view rule results in fullscreen on the Findings page to address an issue where rule results were missing in fullscreen. The fullscreen view will be re-added in the upcoming Data Quality & Observability Classic 2025.03 release.
- Int Check rules using string column types no longer flag valid integers as breaking.

  > **Note** When displaying preview break records, if a column in the dataset has an inferred data type that differs from the defined type, the data type check is performed based on the defined type rather than the inferred type.

- Break records are no longer shown for rules with an exception status. Additionally, an error message now provides details about the exception.

Dataset Manager

- When you rename a dataset from the Dataset Manager, you can now use a name that exists in a different tenant but not in your current one.
- Datasets with spaces in their names no longer lose their rules when you rename them from the Dataset Manager.

Reports

- The time zone configurations of certain Data Quality & Observability Classic environments no longer prevent you from properly viewing the Dataset Findings Report.
- When you use the search filter on the DQ Check Summary Report, you no longer encounter an error when you reach records for Check Type - Rule.

Connections

- The CDATA driver is now supported for Athena Pushdown connections.

Admin Console

- The Security Audit Trail now correctly shows the username of the user who deleted a dataset.

APIs

- The **/v3/jobs/<jobId>/breaks/rules** endpoint no longer returns a 500 error when using a valid jobId. Instead, it now returns empty files when no results are found for exports without findings.
- The **/v3/jobs/{jobID}/findings** and **/v3/jobs/{dataset}/{runDate}/findings** endpoints now return the correct value for the passFail parameter.

Integration

- When you integrate a dataset into Collibra Platform, then add extra columns to it and re-run the integration, the exported JSON file now correctly shows the additional columns.
- The score tile on the Quality tab of Data Quality Rule and Data Quality Job assets now works as expected.
- Dates on the Overview - History table and chart on the Quality tab are now arranged in chronological order.
- The tooltip for Last Run Status on the Summary tab of Data Quality Job assets now contains the correct message.

Features in preview

Rules

- When you edit an existing rule from the Rules tab of the Findings page, you can now edit the tolerance value.
- Rule Tolerance is now available for Pushdown connections.
- Rule filter queries applied to rules on Pullup DQ Job now support secondary datasets. Filter queries on `@t1` rules are not supported at this time.

Release 2024.11

- Announcement
- Enhancements
- Fixes
- Features in preview
- Maintenance Updates

Release Information

- Release dates of Data Quality & Observability Classic:
  - November 25, 2024: Data Quality & Observability Classic 2024.11
  - December 11, 2024: Data Quality & Observability Classic 2024.11.1
- Release notes publication date: October 31, 2024

Announcement

Important

As a security measure, we are announcing the end of life of the Java 8 and 11 versions of Data Quality & Observability Classic, effective in the August 2025 (2025.08) release.

In the February 2025 (2025.02) release, Data Quality & Observability Classic will only be available on Java 17. Depending on your installation of Data Quality & Observability Classic, you can expect the following in the 2025.02 release:

- **Kubernetes installations**: Kubernetes containers will automatically contain Java 17. You may need to update your custom drivers and SAML keystore to maintain compatibility with Java 17.
- **Standalone installations**: You must upgrade to Java 17 to install Data Quality & Observability Classic 2025.02. Additional upgrade guidance will be provided upon the release date. We encourage you to migrate to a Kubernetes installation, to improve the scalability and ease of future maintenance.

The March 2025 (2025.03) release will have Java 8 and 11 versions of Data Quality & Observability Classic and will be the final release to contain new features on those

Java versions. Between 2025.04 and 2025.07, only critical and high-priority bug fixes will be included in the Java 8 and 11 versions of Data Quality & Observability Classic.

Additional details on driver compatibility, SAML upgrade procedures, and more will be available alongside the 2025.02 release.

For more information, visit the Data Quality & Observability Classic Java Upgrade FAQ.

Enhancements

Platform

- When querying the rule_output table in the Metastore, the rows_breaking and total_count columns now populate the correct values for each assignment_id. When a rule filter is used, the total_count column reflects the filtered number of total rows.

Integration

- We automated connection mapping by introducing:
  - Automapping of schemas, tables, and columns.
  - The ability to view table statistics for troubleshooting unmapped or partially mapped connections.

  > **Note** Automapping is currently only available in multi-tenant environments. If your organization has a single-tenant environment, continue to use the manual mapping option until automapping is available in single-tenant environments in the 2025.03 Data Quality & Observability Classic release.

- The Quality tab is now hidden when an aggregation path is not available. (idea #DCC-I-3252)

Pushdown

- Snowflake Pushdown connections now support source to target analysis for datasets from the same Snowflake Pushdown connection.

- You can now monitor advanced data quality layers for SAP HANA Pushdown connections, including categorical and numerical outliers and records.
- Trino Pushdown connections now support multiple link IDs for dupes scans.

Connections

- We now provide out of the box support for Cassandra and Denodo data source connections. You can authenticate both connection types with the basic username and password combination and password manager method.
- You can now authenticate SQL Server connections with NTLM.
- We upgraded the Snowflake JDBC driver to 3.20.0.

Jobs

- You can now set a new variable, `-rdAdj`, in the command line to dynamically calculate and substitute the run date for the `-rd` variable at the run time of your DQ Job.
- The metadata bar now displays the schema and table name.

Findings

- If you assign multiple Link IDs in a Dupes configuration, each Link ID is now present in the break record preview.
- When there are rule findings, the Breaking Records column on the Rules tab displays the number of rows that do not pass the conditions of a rule. In the Metastore, the values from the Breaking Records column are included in the rows_breaking column of the rule_output table. However, after initially upgrading to 2024.11, values in the rows_breaking column remain `[NULL]` until you re-run your DQ Job.

  > **Important** To include data from the rows_breaking column in a dashboard or report, you first need to re-run your DQ Job to populate the column with data.

Alerts

- There are now 8 new variables that allow you to create condition alerts for the scores of findings that meet their criteria. These condition variables include:

    - behaviorscore
    - outlierscore
    - patternscore
    - sourcescore
    - recordscore
    - schemascore
    - dupescore
    - shapescore

> **Example** To create an alert for shapes scores above 25, you can set the condition to **shapescore > 25**.

- Job failure alerts now send when a DQ Job fails in the Staged or Initiation activities.

Dataset Manager

- You can now edit and clone DQ Jobs from the Actions button in the far right column on the Dataset Manager.

Fixes

Integration

- Data Quality Job assets now display a “No data quality score available” message when an invalid rule is selected.
- When Data Quality & Observability Classic cannot retrieve the columns from a table or view during the column mapping process, the column UUIDs in Collibra Platform are now used by default.

Pushdown

- You can now run Pushdown Jobs using OAuth Tokens generated by the /v3/auth/Oauth/signin endpoint.
- Unique adaptive rules for Pushdown Jobs with columns that contain null values no longer fail when a scheduled run occurs.

- When turning behavioral scoring off in the JSON definition of DQ Job created on Pushdown connections, behavior scores are no longer displayed.
- When DQ Job created on Pushdown connections with Archive Break Records enabled run, references to link IDs in the rule query are now checked and added automatically if they are missing. This also allows you to add your own `CONCAT()` when using complex rules.
- We improved the performance of DQ Jobs created on Snowflake Pushdown connections that use `LIMIT 1` for data type queries.

Connections

- We fixed a critical issue that prevented DQ Jobs on temp files from running because of a missing temp file bucket error.

Jobs

- Backrun DQ Jobs are now included in the Stage 3 Job Logs.
- Data Preview now works correctly when the source in the Mapping (source to target) activity is a remote file storage connection, such as Amazon S3.
- DQ Jobs on Oracle datasets now run without errors when Parallel JDBC is enabled.
- When using Dataset Overview to query an Oracle dataset, you no longer receive a generic "Error occurred. Please try again." error message when the source data contains a column with a "TIMESTAMP" data type.
- When including any combination of the conformity options (Min, Mean, or Max) from the Adaptive Rules tab, the column of reference on the Shapes tab is no longer incorrectly marked “N/A” instead of “Auto.”
- Shapes can now be detected after enabling additional Adaptive Rules beyond the default Adaptive Rules settings for file-based DQ Jobs.
- After setting up a source to target mapping in the Mapping step of Explorer where both source and target are temp files, you no longer encounter a “Leave this Mapping” message when you click one of the arrow on the right side of the page to proceed to the next step.

Findings

- After suppressing a behavior score for a dataset that you then use to create a scorecard, the scorecard and Findings page now reflect the same score.
- After suppressing a behavior score and the total score is over 100, the new score is calculated correctly.

Rules

- Rules with spaces in the link ID column now load successfully in the Rule Breaks preview. For example, the link ID column `~|ABC ~|`, where `~|` is the column delimiter, now loads successfully in the Rule Breaks preview.
- When changing a rule type from a non-Native to Native rule, the Livy banner no longer displays and the Run Result Preview button is enabled. When changing any rule type to any other rule type that is non-Native, Livy checks run and the appropriate banner displays or the Run Result Preview button is enabled.

Alerts

- When a single rule is passing after adding 3 distinct alerts for each Rule Status trigger (Breaking, Exception, and Passing) and one alert with all 3, unexpected alerts no longer send when the DQ Job runs.
- Batch alerts now use the same alerts queue to process as all other alert emails.

APIs

- The /v2/getdatapreview API is now crossed out and marked as deprecated in Swagger. While this API is now deprecated, it continues to function to allow backward compatibility and functionality in legacy workflows.
- The Swagger UI response array now includes the 204 status code, which means that a request has been successfully completed, but no response payload body will be present.

Latest UI

- When using Dataset Overview to query an Oracle dataset, you no longer receive a generic "Error occurred. Please try again." error message when the source data contains a column with a "TIMESTAMP" data type.
- The Adaptive Rules modal on the Findings page now allows you to filter the results to display only Adaptive or Manual Rules or display both.
- We re-added the ability to expand the results portion of the Findings page to full screen.
- There is now an enhanced warning message when you create an invalid Distribution Rules from the Profile page.
- The Select Rows step of Explorer now has a tooltip next to the Standard View option to explain why it is not always available.
- The Actions button on the Dataset Manager now includes options to edit and clone DQ Jobs.
- The Rule Details dialog now has a tooltip next to the "Score" buttons to explain the downscoring options.
- We consolidated the individual login buttons on the Tenant Manager page to a single button that returns you to the main login page.
- Table headers in exported Job Logs generated from the Jobs page now display correctly.

Features in preview

Rules

- You can now apply a rule tolerance value to indicate the threshold above which your rule breaks require the most urgent attention. Because alerts associated with rules can generate many alert notifications, this helps to declutter your inbox and allows you to focus on the rule breaks that matter most to you.
- Rule filtering is now available for Pushdown DQ Jobs.

Maintenance Updates

- We added a new check on the flyway library to resolve issues upon upgrade to Data Quality & Observability Classic 2024.10.

- Denodo connections now support OAuth2 authentication.
- You can now Configure AWS passwordless authentication using Amazon RDS PostgreSQL as the Metastore using Amazon RDS PostgreSQL as the Metastore.

> **Note** AWS passwordless authentication is currently only supported for EC2 Instance Profile-based authentication with an Amazon RDS Metastore for Data Quality & Observability Classic standalone and cluster-based deployments. IAM pod role-based authentication support will be available in a future release.

Release 2024.10

- Enhancements
- Fixes
- Features in preview
- DQ security

Release Information

- Release date of Data Quality & Observability Classic 2024.10: October 29, 2024
- Release notes publication date: September 23, 2024

> **Warning**
>
> Some customers have encountered issues while upgrading to Data Quality & Observability Classic 2024.10 due to a change in our flyway library that is not backwards compatible. The fix for this issue is included in the Data Quality & Observability Classic 2024.11.1 patch. As always, we recommend backing up and restoring your Metastore before upgrading Data Quality & Observability Classic versions.

> **Note** The above issue only impacts upgrades to Data Quality & Observability Classic 2024.10. New installations will not encounter this issue.

Enhancements

Pushdown

- SAP HANA Pushdown is now generally available.
- When creating and running DQ Jobs on SQL Server Pushdown connections, you can now perform schema, profile, and rules checks.
- You can now scan for fuzzy match duplicates in DQ Jobs created on BigQuery Pushdown connections.
- You can now scan for numerical outliers in DQ Jobs created on Trino Pushdown connections.
- DQ Jobs created on Snowflake Pushdown connections now support union lookback for advanced outlier configurations.
- DQ Jobs created on Snowflake Pushdown connections now support source to target validation to ensure data moves consistently through your data pipeline and identify changes when they occur.

Integration

- You can now define custom integration data quality rules, which are also known as aggregation paths, in the Collibra Platform operating model setting to allow you to view data quality scores for assets other than databases, schemas, tables, and columns.

  > **Important** You need a minimum of Collibra Platform 2024.10 and Data Quality & Observability Classic 2024.07.

- To allow you to manage the scope of the DGC resources to which OAuth can grant access during the integration, a new OAuth parameter in the Web ConfigMap is now set to `DQ_DGC_OAUTH_DEFAULT_REQUEST_SCOPE: "dgc.global-manage-all-resources"` by default. This configuration grants Data Quality & Observability Classic access via OAuth to all DGC resources during the integration. For more granular control over the DGC resources to which Data Quality & Observability Classic is granted access via OAuth, we plan to introduce additional allowable values in a future release.

- Users of the Quality tab in Collibra Platform who do not have a Data Quality & Observability Classic account can now view the history table to track the evolution of the quality score of a given asset.
- When using Replay to run DQ Jobs over a defined historical period, for example 5 days in the past, the metrics from each backrun DQ Job is included in the DQ History table and the quality calculation.
- After integrating a dataset from Data Quality & Observability Classic and Collibra Platform, you can now see the number of passing and failing rows for a given rule on the Data Quality Rule asset page.
- The JSON of a dataset integration between Data Quality & Observability Classic and Collibra Platform now shows the number of passing and breaking records.

Jobs

- You can now edit the schedule details of DQ Jobs from the Jobs Schedule page.
- A banner now appears when a data type is not supported.

Rules

- Data Class Rules now have a maximum character length of 64 characters for the Name option and 256 for the Description.
- Email Data Classes where the email contains a single character domain name now pass validation. For example, user@x.com
- The Rule Definitions page now has a Break Record Preview Available column to make it easier to see when a rule is eligible for previewing break records.
- You can now use the search field to search for Rule Descriptions on the Rule Definitions page.

Alerts

- You can now toggle individual alerts from the Active column on the Alert Builder page to improve control over when you want alerts to send. This can prevent unnecessary alerts from being sent during certain occasions, such as setup and debugging.

Dataset Manager

- The Dataset Manager table now contains a searchable and sortable column called Connection Name to help identify your datasets more easily.
- We aligned the roles and permissions requirements for the Dataset Manager API.
    - PUT /v2/updatecatalogobj requires its users to have ROLE_ADMIN, ROLE_DATA_GOVERNANCE_MANAGER, or ROLE_DATASET_ACTIONS.
    - PUT /v2/updatecatalog requires its users to be the dataset owner or have ROLE_ADMIN or ROLE_DATASET_ACTIONS.
    - DELETE /v2/deletedataset requires its users to be the dataset owner or have ROLE_ADMIN or ROLE_DATASET_MANAGER.
    - PATCH /v2/renameDataset requires its users to be the owner of the source dataset or have ROLE_ADMIN, ROLE_DATASET_MANAGER, or ROLE_DATASET_ACTIONS.
    - POST /v2/update-run-mode requires its users to have ROLE_DATASET_TRAIN, ROLE_DATASET_ACTIONS, or dataset access.
    - POST /v2/update-catalog-data-category requires its users to have ROLE_DATASET_TRAIN, ROLE_DATASET_ACTIONS, or dataset access.
    - PUT /v2/business-unit-to-dataset requires its users to have ROLE_DATASET_ACTIONS or dataset access.
    - POST /v2/business-unit-to-dataset requires its users to have ROLE_DATASET_ACTIONS or dataset access.
    - POST /dgc/integrations/trigger/integration requires its users to have ROLE_ADMIN, ROLE_DATASET_MANAGER, or ROLE_DATASET_ACTIONS.
    - POST /v2/postjobschedule requires its users to have ROLE_OWL_CHECK and either ROLE_DATASET_ACTIONS or dataset access.

Connections

- The authentication dropdown menu for any given connection now displays only its supported authentication types.
- We've upgraded the following drivers to the versions listed:
    - Db2 4.27.25
    - Snowflake 3.19.0

> **Note**
> If you use additional encryption algorithms for JWT authentication, you must set one of the following parameters during your deployment of Data Quality & Observability Classic, depending on your deployment type:
>
> **Helm-based deployments**
> Set the following parameter in the Helm Chart: `--set global.web.extraJvmOptions="-Dnet.snowflake.jdbc.enableBouncyCastle=true"`
>
> **Standalone deployments**
> Set the following environment variable in the owl-env.sh: `-export EXTRA_JVM_OPTIONS="-Dnet.snowflake.jdbc.enableBouncyCastle=true"`

- SQL Server 12.6.4 (Java 11 only)

> **Note**
> While the Java 8 version is not officially supported, you can replace the SQL Server driver in the /opt/owl/drivers/mssql folder with the Java 8 version of the supported driver. You can find Java 8 versions of the supported SQL Server on the Maven Repository.

Admin Console

- A new access control layer, **Require DATASET_ACTIONS role for dataset management actions**, is available from the Security Settings page. When enabled, a new out of the box role, **ROLE_DATASET_ACTIONS**, is required to allow its users to edit, rename, publish, assign data categories and business units, and enable integrations from the Dataset Manager.
- A new out of the box role, **ROLE_ADMIN_VIEWER**, allows users who are assigned to it to access the following Admin Console pages, but restricts access to all others:
  - Actions
  - All Audit Trail subpages
  - Dashboard

> **Note** Users with ROLE_ADMIN_VIEWER cannot access the pages to which the quick access buttons on the Dashboards page are linked.

    - Inventory
    - Schedule Restrictions

> **Note** Users with ROLE_ADMIN_VIEWER cannot add or delete schedule restrictions.

    - Usage
- You can now set both size- and time-based data purges from the Data Retention Policy page of the Admin Console. Previously, you could only set size-based data retention policies.
- The Data Retention Policy now supports auto-clean in both single- and multi-tenant environments.

APIs

- We've made several changes to the API documentation. First, we aligned the role checks between the Product APIs (V3 endpoints) and the Data Quality & Observability Classic UI. We've also enhanced the documentation in Swagger to include more detailed descriptions of endpoints. Lastly, we reproduced the Swagger documentation of the Product API in the Collibra Developer Portal to ensure a more unified user experience with the broader Collibra platform and allow for easier scalability of API documentation in the future.

Fixes

Integration

- The Overview - History section of Data Quality Job assets now displays the correct month of historical data when an integration job run occurs near the end of a given month.

Jobs

- DQ Jobs on SAP HANA connections with SQL referencing table names containing semicolons ; now run successfully when you escape the table name with quotation marks "". For example, the SQL query `select * from TEST."SPECIAL_CHAR_TABLE::$/-;@#%^&*?!{}~\+="` now runs successfully.
- You can now run DQ Jobs that use Amazon S3 as the secondary dataset with Instance Profile authentication.

Rules

- Native rules on DQ Jobs created on connections authenticated by password manager now run successfully and return all related break records when their conditions are met.

Alerts

- The Assignee column is no longer included in the alert email for Rule Status and Condition alerts with rule details enabled.

APIs

- When using the POST /v2/controller-db-export call on a dataset with an alert condition, then using the POST /v2/controller-db-import call, now returns a successful 200 response instead of an unexpected JSON parsing error.

Latest UI

- When running DQ Jobs on NFS connections, data files with the date format **${yyyy}${MM}${dd}** within their file name are now supported.
- Native Rules now display the variable name of parameters such as @runId and @dataset with the actual value in the Condition column of the Rules tab on the Findings page.
- The Jobs Schedule page now shows the time zone offset (+ or - a number of hours) in the Last Updated column. Additionally, the TimeZone column is now directly to the right of the Scheduled Time column to improve its visibility.
- You can now sort columns on the Job Schedule page.

- The Agent Configuration, Role Management - Connections, Business Units, Inventory pages of the Admin Console now have fixed column headers and the Actions button and horizontal scrollbar are now visible at all times.
- After adding or deleting rules, the rule count on the metadata bar now reflect any updates.

Features in preview

- The Rule Workbench now contains an additional query input field called “Filter,” which allows you to narrow the scope of your rule query so that only the rows you specify are considered when calculating the rule score. A filter query not only helps to provide a better representation of your quality score but improves the relevance of your rule results, saving both time and operational costs by reducing the need to create multiple datasets for each filter.

  > **Important** This feature is currently available as a public preview option. For more information about preview features, see Previews at Collibra.

Known limitations

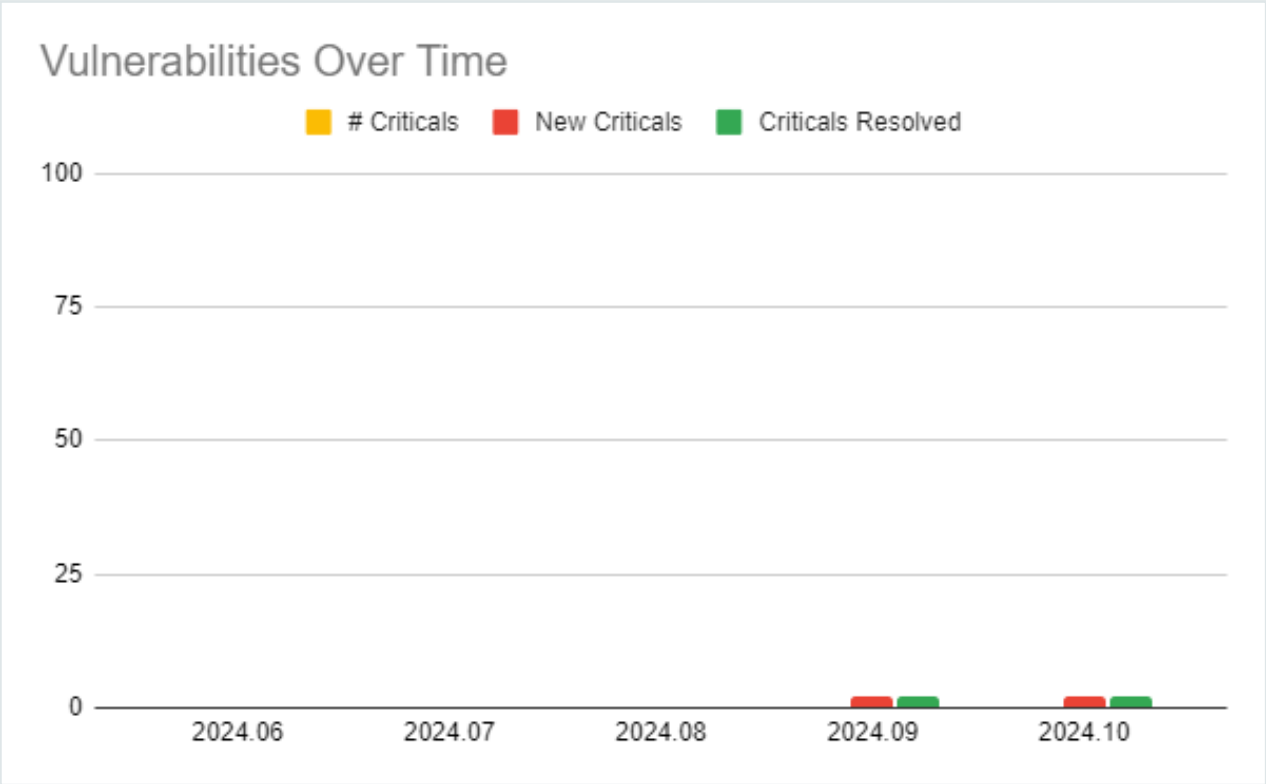
- In the this release, table names with spaces are not supported because of dataset name validation during the creation of a dataset. This will be addressed in an upcoming release.

DQ Security

Show security metrics
The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs
DQ	2024-06-21	2024.06	0	0	0	0
DQ	2024-07-29	2024.07	0	0	0	0
DQ	2024-08-19	2024.08	0	0	0	0
DQ	2024-09-26	2024.09	0	2	2	0
DQ	2024-10-29	2024.10	0	2	2	0

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2024.09

- Enhancements
- Fixes
- DQ Security

Release Information

- Release date of Data Quality & Observability Classic 2024.09: September 30, 2024
- Release notes publication date: September 5, 2024

Enhancements

Platform

- We added support for FIPS-compliant algorithms.

Integration

- Users of the Quality tab in Collibra Platform who do not have a Data Quality & Observability Classic account can now view the 7-day history of data quality scores, allowing you to monitor the health of your data over time.
- When running a DQ Job with back run and an active integration, the results of the back run are now sent to Collibra Platform where they are stored in the DQ services history table.

Pushdown

- You can now scan for shapes in DQ Jobs created on Trino Pushdown connections.

Jobs

- You can now create DQ Jobs on SAP HANA tables that contain the special characters **::$/-;@#%^&*?!{}~+=**

  > **Note** The special characters **.()** are not supported.

Findings

- When the **dupelimit** and **dupelimiui** limits on the Admin Limits page are both set to 30, the Findings page now limits the number of dupes findings marked on the Dupes tab to 30.

Alerts

- If your organization uses multiple web pods on a Cloud Native deployment of Data Quality & Observability Classic, you now receive only one alert email when an alert condition is met.

APIs

- When dataset security is enabled on a tenant and a user whose roles meet the requirements of the Dataset Def API and has a their role assigned to the dataset, the API returns the expected results.

Fixes

Integration

- When setting up an integration, the Connections step now has a search component and pagination to prevent table load failure when the tables per schema size exceeds browser memory.
- The dimension type of Adaptive Rules (NULL, EMPTY, MIN, and so on) now correctly maps to the type and sub-type from the DQ dimension table.
- The predicate of custom rules is again included in rule asset attributes.

Pushdown

- You can again download CSV and JSON files containing rule break records of DQ Jobs created on Pushdown connections.
- The rule name no longer appears in the data preview and downloaded rule breaks of rule findings for DQ Jobs created on Pushdown connections.

> **Note** For consistency with rule break downloads in Pullup mode, we plan to separate rule breaks by rule in a future release. As of this release, Pullup mode still includes the rule name and runId in rule break downloads

- DQ Jobs created on Pushdown connections where columns with shape values that contain **$** or **‘** now run successfully. Previously, such Jobs failed with an unexpected exception message.
- When running DQ Jobs created on Pushdown connections that scan multiple columns for duplicate values, the data of the columns now appears under the correct column name.
- When a DQ Job created on Pushdown connection contains 0 rows of data and one or more rules are enabled, the rules are now included in the Job run and displayed on the Findings page.

Jobs

- The Explorer connection tree now loads successfully when a schema contains tables that contain unsupported column types.
- Dataset Overview on the Explorer page can now process the `select *` part of the SQL statement if there is an unsupported column type.
- We fixed an issue on the Jobs page where Data Quality & Observability Classic was unable to retrieve the Yarn Job ID.
- When re-running a DQ Job from the metadata bar on DQ Job that previously ran with backrun (`-br`), the DQ Job that you re-run will no longer incorrectly initiate a backrun.

> **Note** If the `-br` option is included in the beginning of your command line, your DQ Job will perform a backrun and `-br` will be removed from the command line when the DQ Job completes.

Findings

- After retraining a behavioral finding to pass a value for a blindspot, the score now correctly reflects the retrained scoring model.

Profile

- When adding a stat rule for distribution from the **+Add Rule** option on the Profile page, the computed boundaries of categorical variables in the distribution rule now display correctly.

Rules

- When rules on Pullup datasets time out, the rule output record now displays the out-of-memory (OOM) exception message on the Findings page.
- Run Result Preview on the Rule Workbench now works as expected for custom rules that use the simple rule template.
- When creating a rule for an existing DQ Job created on a Pushdown connection, Run Result Preview now runs without errors.
- You can now use rules that contain a **$** (not stat rules) with profiling off for DQ Jobs created on Pushdown connections.

Admin Console

- Admin Limits now require values to be -1, 0, or positive numbers. An inline message appears below the Value field when a limit does not meet the allowed values.

Latest UI

- We improved the performance of the Scorecards page when there are many datasets to load.
- We reduced the number of backend calls on the Profile and Findings pages to improve the load time performance.
- You can now create, edit, and delete alerts for datasets with 0 rows. When you run a job on a dataset with 0 rows, the alerts function as expected.
- Dataset Manager no longer crashes due to slow network calls when you click the Filter icon as the page loads.
- We resolved multiple scenarios where the metadata bar did not display any rows or columns.
- When hovering over a row on the Scheduler page, the row has a gray highlight, but the days of week cells remain green or white, depending on whether a DQ Job is scheduled to run on a given day.
- The date picker on the Job tab of the Findings page is now available for DQ Jobs created on Pushdown connections and you can successfully run DQ Jobs with the dates you select.
- The Findings and other pages now load correctly in Safari browsers.

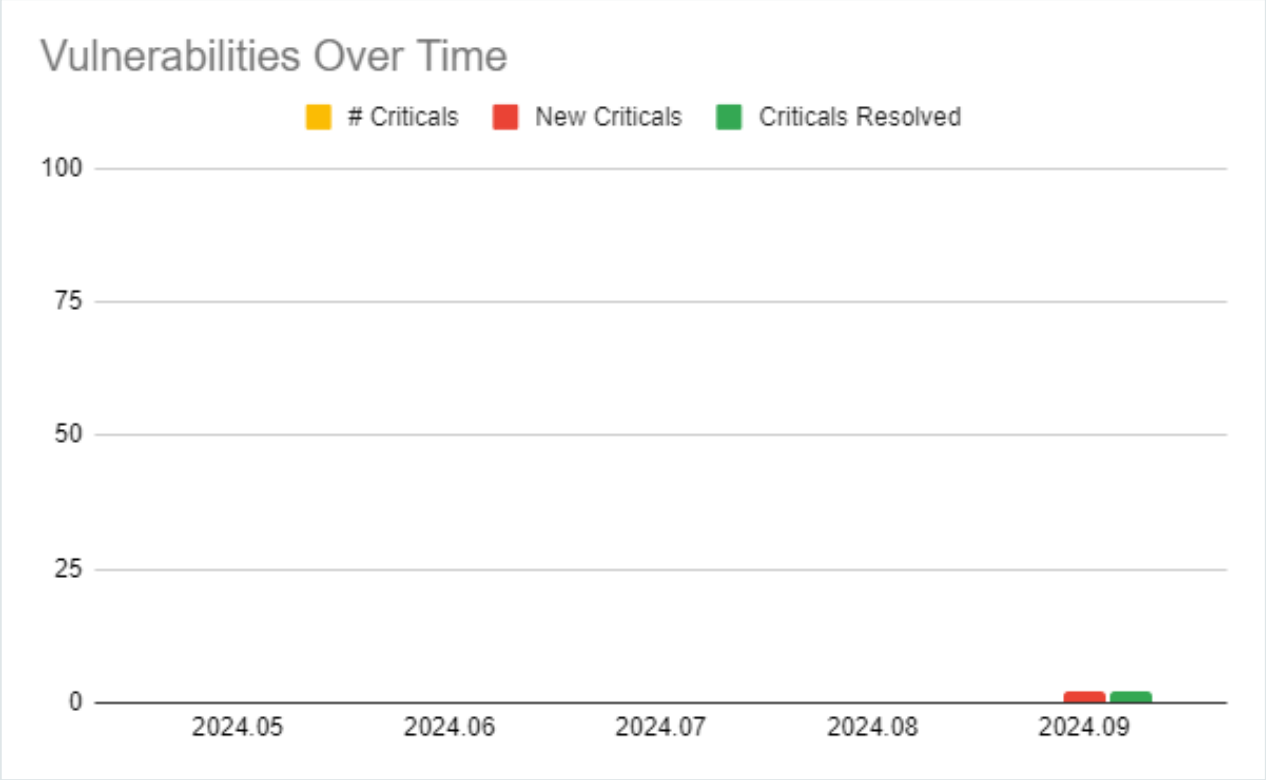
- DQ Jobs created on Pushdown connections no longer generate duplicate user-defined job failure alerts.
- The donut charts in the database and schema reports from Explorer now consistently display the correct stats.

DQ Security

Show security metrics
The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs
DQ	2024-05-09	2024.05	0	0	0	0
DQ	2024-06-21	2024.06	0	0	0	0
DQ	2024-07-29	2024.07	0	0	0	0
DQ	2024-08-19	2024.08	0	0	0	0
DQ	2024-09-26	2024.09	0	2	2	0

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2024.08

- Enhancements
- Fixes
- DQ Security

Release Information

- Release date of Data Quality & Observability Classic 2024.08: August 26, 2024
- Publication dates:
  - Release notes: August 5, 2024
  - Documentation Center: August 9, 2024

Enhancements

Integration

- When a job with an active integration with Collibra Platform runs, the Job Log on the Jobs page in Data Quality & Observability Classic reflects the details of each step of the integration.

Pushdown

- You can now scan for both fuzzy and exact match duplicate records in Trino Pushdown jobs.
- All Pushdown-compatible data sources now support the use of temporal datasets in stat rule statements, for example, "`SELECT @t1.$rowcount AS yesterday, @dataset.$rowcount AS today WHERE yesterday <> today`"

Connections

- You can now create BigQuery Pullup jobs on a cross-project connection without manually updating the command line to prepend the projectId in the source query.

- NetApp now has a dedicated connection tile under the Remote File Connections tab of the Add New Connection modal. Previously, to connect to a NetApp data source, you had to follow the Amazon S3 path and add the NetApp connection properties to the Properties tab.
- You can now archive breaking records to NetApp locations.

Jobs

- You can now execute command line queries that end with double quotes around the table name, for example, `select * from "<SCHEMA>"."<TABLE>"`

Profile

- When you hover your cursor over the histogram on a dataset profile page, the upper quartile, median, and lower quartile statistics now display.

Findings

- NULL values are now excluded from the calculation of duplicate values for Pullup jobs.
- The indicator representing the number of findings for a given layer are now in the upper right corner of the associated layer.
- The Actions button is now always visible at the far right side of the Adaptive Rules modal. Additionally, the Adaptive Rule types are now color-coded.
- The chips in the Observations column of the Records tab are now color-coded.

Rules

- The Copy Results and Download Results buttons from the Dataset Overview are now available on the Rule Workbench.

Alerts

- We cleaned up the mailTemplate.html file within the dq-webapp to improve user experience.

Dataset Overview

- The ability to preview data on the Dataset Overview now requires access to the connection upon which your job is based and at least one of the following roles:
    - ROLE_VIEW_DATA
    - ROLE_ADMIN
    - ROLE_CONNECTION_MANAGER

Scorecards

- We shortened the height of the scorecard blocks to reduce the amount of time it takes to scroll the Scorecards page when multiple scorecard blocks are present.

Assignments

- You can now use the Date Range filter on the Assignments Queue to sort and define a range of dates in the Update Ts (timestamps) column.

Dataset Manager

- Admins can now bulk update the agent and Spark settings of Pullup datasets.

SQL Assistant for Data Quality

- You can now see details of the Vertex AI model in the 'About' modal in the upper right corner of your Data Quality & Observability Classic instance.

APIs

- We aligned the role requirements for the Jobs and Alerts V3 APIs.
    - When dataset security and DB connection security are disabled, users have full access to the Jobs and Alerts endpoints.
    - When dataset security is enabled and DB connection security is disabled, users without a role assignment to a dataset cannot use the following endpoints referencing that dataset:

        - GET v3/jobs/{dataset}/{rundate}/logs
        - GET v3/jobs/{jobId}
        - GET v3/jobs/{jobId}/waitForCompletion
        - GET v3/jobs/{jobId}/logs
        - GET v3/jobs/{jobId}/findings
        - GET v3/jobs/{jobId}/breaks/shapes
        - GET v3/jobs/{jobId}/breaks/rules
        - GET v3/jobs/{jobId}/breaks/outliers
        - GET v3/jobs/{jobId}/breaks/dupes
        - GET v3/alerts/{dataset}
        - GET v3/alerts/{dataset}/{alertname}
        - GET v3/alerts/notifications
        - DELETE v3/alerts/{dataset}/{alertname}
    - When dataset security is enabled and DB connection security is disabled, users without a role assignment to a dataset can use the following endpoints:
        - GET v3/jobs
        - GET v3/alerts
    - The Job and Alert APIs honor dataset security by preventing access to the alert or job details when the user making the request does not have role access to the related dataset.
    - By design, the Job APIs only honor connection security related to job creation or execution actions.
    - Currently, connection security is not enforced for the Alert APIs.

Fixes

Connections

- You can again edit datasets created on Remote File Connections.

Jobs

- When the Case Sensitive and Exact Match options are not selected in the Dupes layer, jobs that run in Pullup mode now scan for all case-insensitive fuzzy match duplicates.

- When using the Partial Scan option in the latest UI, you can now use the 'select all' checkbox option in the column header to select all columns when they contain unsupported data types, such as CLOB.
- After setting up a partial scan of an Oracle dataset in the latest UI, the job now runs without error.
- You can again run jobs on Oracle datasets where source-to-target mappings to Databricks connections are configured.
- Columns now display on the Profile page when the Profile activity fails.
- When adding a distribution rule on a column from the Profile page, the percentages are now correctly calculated based on the total number or rows.
- You can again edit jobs based on temp files in Standalone deployments of Data Quality & Observability Classic.

Reports

- The Coverage Report now returns a maximum of 1 calendar year of the statistics of database-, schema-, and table-level jobs. If each level does not have an existing structure, the report returns a helpful error message.

Agent

- When you select an option from the Master Default dropdown menu on the Edit Agent dialog of the Agent Configuration page, the correct value now displays based on your selection.

Integration

- Schemas and tables now correctly map to Collibra Platform assets when you automap them from the connection mapping step of the Integration Admin Console page
- The total row count now correctly displays in the Loaded Rows field on the asset page after the integration of a Pushdown dataset.
- The Run Job Again option is no longer visible on the View Monitoring modal of the At a glance sidebar for table assets of scheduled and non-scheduled jobs.

- The scheduler in Data Quality & Observability Classic that previously monitored for triggers sent from Collibra Platform to run a job in Data Quality & Observability Classic is now disabled and no longer scans for these inputs.

Pushdown

- You can now see the histogram of BigQuery Pushdown jobs.
- Snowflake Pushdown jobs with rules that reference secondary datasets with identical column names no longer return exceptions.
- The Profile page now shows TopN and BottomN shape results for Snowflake Pushdown jobs.
- The timestamp portion of the Run ID is now supported in Pushdown jobs that are configured to run on a schedule.
- When Archive Break Records for Rules is enabled and the rules of a Pullup job includes at least one DATATYPECHECK rule, the Rules page now shows the correct statuses when the rules are copied to a new Pushdown job. Additionally, the DATATYPECHECK rules from the Pullup job do not copy to the Pushdown job, as DATATYPECHECK rules are not supported in Pushdown mode.

APIs

- When using the POST v3/rules endpoint to add an inactive rule (isActive option set to 0) to a dataset, the rule is now added to the dataset correctly.

Latest UI

- We improved the performance of the Explorer page so that schemas with many tables no longer lock the page in an unresponsive state while they load.
- The errors that occurred when compiling a dataset source query using a date variable in Explorer are now resolved.
- Dataset pages now load correctly when an invalid run ID is applied to a Pushdown job.
- When editing a DQ Job on a Remote File Connection, the Compile button is now disabled and includes a note instructing you to edit the query from the command line instead.

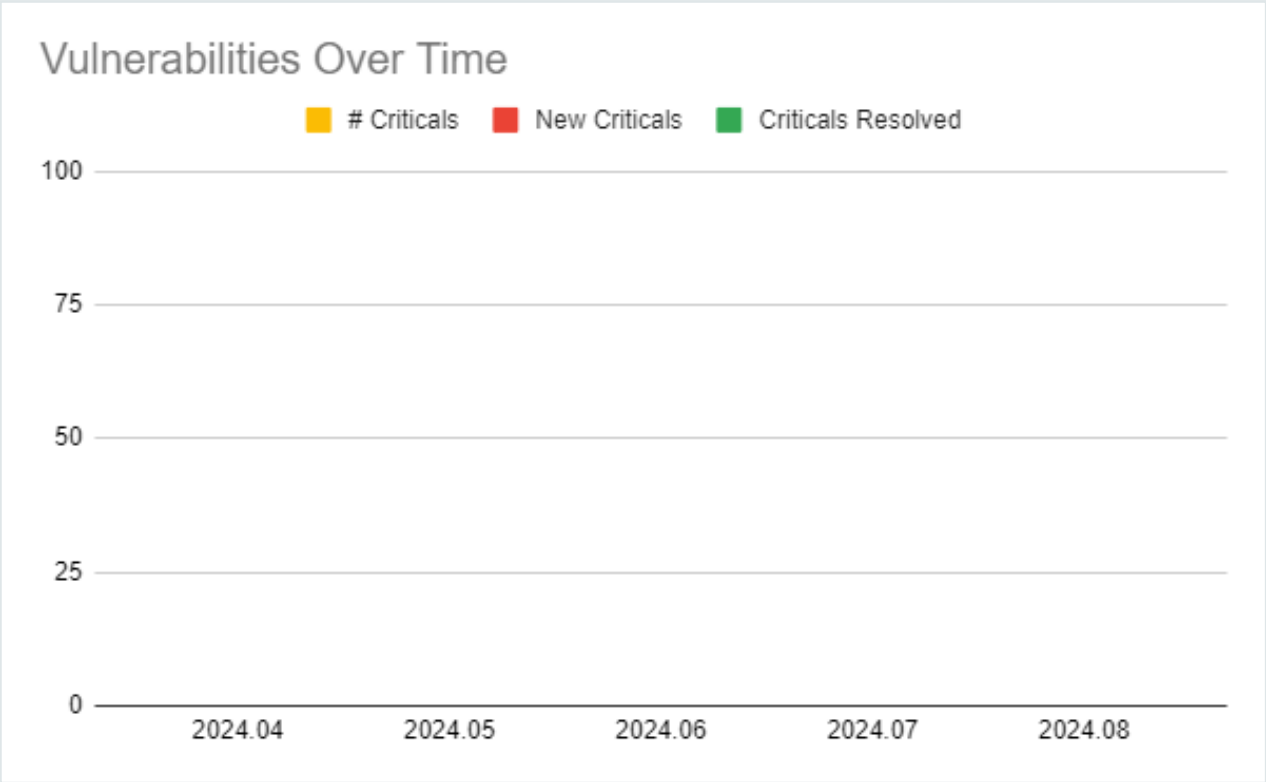
- The issues that prevented some users from editing certain datasets created from Remote File Connections are now resolved.
- You can again edit DQ Jobs created on Temp File connections.
- The Agent Master Default option on the Agent Configuration page now displays correctly.
- You can now deselect unsupported column types when performing a partial scan in Explorer.
- The row count in the job estimate now reflects the source query row count.
- When using Validate Source, JDBC source connections mapped to Databricks connections no longer return errors.
- We added labels to the Histogram portion of the Profile page that are available when you hover your cursor over the histogram.
- Percentages for Quick Rules (Distribution) on the Profile page now display correctly.
- Observations on the Record tab of the Findings page are now color-coded.
- We fixed an issue with the Rule Workbench where the rule body became uneditable after loading a data type rule.
- Unintended changes to out-of-the-box Sensitive Labels are now prevented.
- File names of exports from the Schedule page now include date and timestamps.
- The Coverage Report now returns connection-level metrics from the latest UI and APIs.
- We cleaned up typos on the Completeness Report page.

DQ Security

Show security metrics
The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs
DQ	2024-04-15	2024.04	0	0	0	0
DQ	2024-05-09	2024.05	0	0	0	0
DQ	2024-06-21	2024.06	0	0	0	0
DQ	2024-07-29	2024.07	0	0	0	0
DQ	2024-08-19	2024.08	0	0	0	0

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2024.07

- Highlights
- Enhancements
- Fixes
- Security Metrics

Release Information

- Release date of Data Quality & Observability Classic 2024.07: July 30, 2024
- Publication dates:
  - Release notes: June 24, 2024
  - Documentation Center: July 4, 2024

Highlights

Important
As of this release, the classic UI is no longer available.

Important
To improve the security of Data Quality & Observability Classic, we removed the default keystore password from the installation packages in this release. In general, if your organization uses a SAML or SSL keystore, we recommend that you provide a custom keystore file. However, if your organization has a Standalone installation of Data Quality & Observability Classic and plans to continue using the default keystore, please contact Support to receive the default password to allow you to install and successfully use Data Quality & Observability Classic versions 2024.07 and newer. If your organization has a containerized (Cloud Native) installation of Data Quality & Observability Classic, you can continue to leverage the default keystore file, as the latest Helm Charts have the default password set within the values.yaml file.

Integration

You can now select the data quality layers that will have corresponding assets created in Collibra Platform either automatically upon a successful integration or only when a layer contains breaking records. By selecting individual layers instead of including all of them by default, this can help prevent an overwhelming number of assets from being created automatically.

Admins can configure this on the Integration Setup wizard of the Integrations page in the Admin Console.

Admins can also map file and database views from Data Quality & Observability Classic to corresponding assets in Collibra Platform Catalog. This allows for out-of-the-box relations to be created between their file- or view-based Data Quality & Observability Classic datasets and the file table and database view assets (and their columns) in Collibra

Platform.

> **Note** Collibra Platform 2024.07 or newer is required for view support.

Pushdown

We're delighted to announce that Pushdown for SQL Server is now generally available!

Pushdown is an alternative compute method for running DQ Jobs, where Data Quality & Observability Classic submits all of the job's processing directly to a SQL data warehouse. When all of your data resides in the SQL data warehouse, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirements of a DQ Job.

Enhancements

Explorer

- From the Sizing step in Explorer, you can now change the agent of a Pullup job by clicking the Agent Details field and selecting from the available agents listed in the Agent Status modal.
- You can now use custom delimiters for remote files in the latest UI.
- When using the Mapping layer for remote file connections, you can now add –`srcmultiline` to the command line to remove empty rows from the source to target analysis.
- We added a Schema tab to the Dataset Overview to display table schema details after clicking Run in the query box. (ticket #145266)
- When the number of rows in a dataset exceeds 999, you can now hover your cursor over the abbreviated number to view the precise number of rows. (ticket #125753)

Rules

- Break record previews for out-of-the-box Data Type rules now display under the Rules tab on the Findings page when present. (idea #DCC-I-2155)
- When a dataset is renamed from the Dataset Manager, any explicit references to the dataset name for all primary and secondary rules are updated. (idea #DCC-I-2624)
- You can now rename rules from the Actions dropdown menu on the Dataset Rules page. Updated rule names cannot match the name of an existing rule and must contain only alphanumeric characters without spaces.

Alerts

- We added a new Rule Status alert to allow you to track whether your rule condition is breaking, throwing an exception, or passing.
- When configuring Condition-type alerts, you now have the option to Add Rule Details, which includes the following details in the email alert (idea #DCC-I-732):
    - Rule name
    - Rule condition
    - Total number of points to deduct from the quality score when breaking
    - Percentage of records that are breaking
    - Number of breaking records
    - Assignment Status
- Admins can now set a custom alert email signature from the Alerts page in the Admin Console. (idea #DCC-I-2400)

Profile

- Completeness percentage is now listed on the dataset Profile page in the latest UI.

Findings

- We added a Passing Records column to the rule break table under the Rules tab on the Findings page to show the number of records that passed a rule. This enhancement simplifies the calculation of the total number of records, both passing and breaking. Additionally, we renamed the Records column, Breaking Records. (idea #DCC-I-2223)

- You can now download CSV files and copy signed links of rule break records on secure S3 connections, including NetApp, MinIO, and Amazon S3.

Scorecards

- To improve page navigation, we added a dedicated Add Page button to the top right corner of the Scorecards page and moved the search field from the bottom of the scorecards dropdown menu to the top.

Reports

- When Dataset Security is turned on, the only datasets and their associated data that display on the Dataset Dimension and Column Dimension dashboards are the ones to which users have explicit access. This is only applicable when Dataset Security is enabled by an admin in Security Settings.

Jobs

- When exporting job logs, the names of the export files of job logs exported in bulk from the Jobs page begin with a timestamp representing when the file was downloaded. Additionally, the names of the export files of job logs of individual jobs begin with a timestamp representing the UpdateTimestamp and includes the first 25 characters of the dataset name.

Dataset Manager

- Admins can now update dataset hosts in bulk by selecting Bulk Manage Host from the Bulk Actions dropdown menu and specifying a new host URL.

Integration

- When datasets do not have an active Collibra Platform integration, you can now enable dataset integrations in bulk from the Dataset Manager page by clicking the checkbox option next to the datasets you wish to integrate, then selecting Bulk Enable Integration from the Bulk Actions dropdown menu.

    - Additionally, when datasets have an active integration, you can now submit multiple jobs to run by selecting Bulk Submit Integration Jobs from the Bulk Actions dropdown menu.
- We improved the score calculation logic of Data Quality Rule assets.
- When viewing Rule assets and assets of data quality layers in Collibra Platform, the Rule Status now displays either Passing, Breaking, Learning, or Suppressed. Previously, rules and data quality layers without any breaks displayed an Active status, but that is now listed as Passing. (ticket #137526)

Pushdown

- You can now scan for exact match duplicates in BigQuery Pushdown jobs.

  > **Note** An enhancement to enable scanning for fuzzy match duplicates in BigQuery Pushdown jobs is planned for Data Quality & Observability Classic 2024.10.

- You can now scan for shapes and exact match duplicates in SAP HANA Pushdown jobs. Additionally, we've added the ability to archive duplicates and rules break records to the source SAP HANA database to allow you to easily identify and take action on data that requires remediation.

SQL Assistant for Data Quality

- We added AI_PLATFORM_PATH to the Application Configuration Settings to allow Data Quality & Observability Classic users who do not have a Collibra Platform integration to bypass the integration path when this flag is set to FALSE.
    - When set to TRUE (default), code will hit the integration or public proxy layer endpoint.
    - When set to FALSE, code will bypass the integration path.

Identity Management

- We removed the Add Mapping button from the AD Security Settings page in the Admin Console.

- When dataset security is enabled and a dataset does not have a previous successful job run, users without explicit access to it will not see it when they use the global search to look it up.

APIs

- You must have **ROLE_ADMIN** to update the host of one or many datasets using the PATCH /v3/datasetDefs/batch/host call. Any updates to the hosts of datasets are logged in the Dataset Audit Trail.
- By using the Dataset Definitions API, admins can now manage the following in bulk:
  - Agent
    - PATCH /v3/datasetDefs/batch/agent
  - Host
    - PATCH /v3/datasetDefs/batch/host
  - Spark settings
    - PATCH /v3/datasetDefs/batch/spark

Platform

- To ensure security compliance for Data Quality & Observability Classic deployments on Azure Kubernetes Service (AKS), we now support the ability to pass sensitive PostgreSQL Metastore credentials to the Helm application through the Kubernetes secret, `--set global.configMap.data.metastore_secret_name`. Further, you can also pass sensitive PostgreSQL Metastore credentials to the Helm application through the Azure Key Vault secret provider class object, `--set global.vault.enabled=true --set global.vault.provider=akv --set global.vault.metastore_secret_name`.

Fixes

Connections

- You can now create a Snowflake JDBC connection with a connection URL containing either a double quote or curly bracket, or a URL encoded version of the same, for example, **{"tenant":"foo","product":"bar","application":"baz"}**. (ticket #148348)

Findings

- We fixed an issue which prevented sorting in the Records column for datasets with multiple rule outputs and different row count values. (ticket #142362)
- We resolved a misalignment of Outlier column values in the latest UI. (ticket #146163)

Reports

- The Column Dimension and Dataset Dimension reports now display an error message when users who do not have **ROLE_ADMIN** attempt to access them. (ticket #141397)

Integration

- We improved the error handling when retrieving RunId for datasets. (ticket #145040, 146724, 148232)

Latest UI

- When a user does not have any assigned roles, an empty chip no longer displays in the Roles column on the User Management page in the Admin Console.
- We removed the Enabled and Locked columns from the AD Mapping page in the Admin Console, because they do not apply to AD users.
- A warning message now appears when you attempt to load a temp file and the Temp File Upload option is disabled on the Security Settings page in the Admin Console.
- You can now sort the Records column of the Rules tab on the Findings page.
- Outlier findings now display in the correct column.
- The section of the application containing Scorecards, List View, Assignments, and Pulse View is now called Views.
- The links on the metadata bar now appear in a new order to better reflect the progression of how and when they are used.
- You can now change the agent of a dataset during the edit and clone process.
- If schemas or tables fail to load in Explorer, an error message now appears.

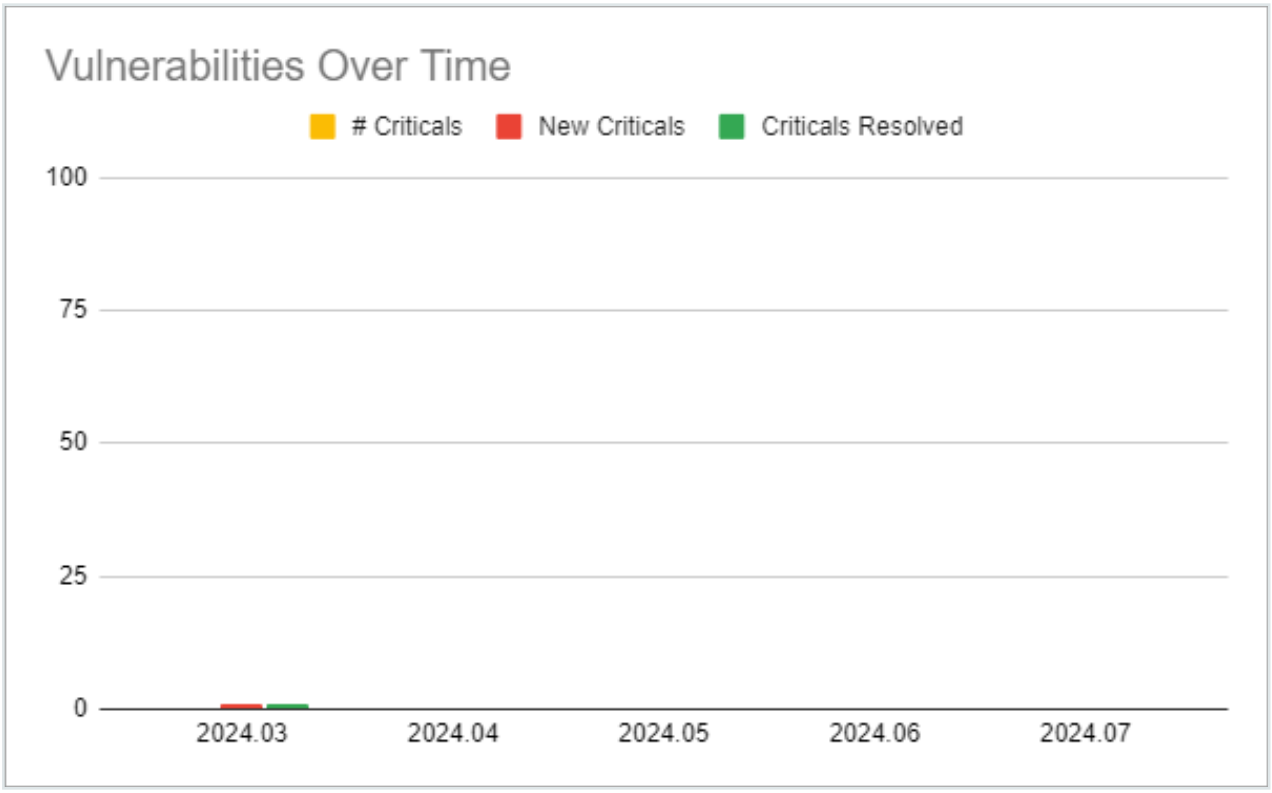
DQ Security

Show security metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

A	B	C	D	E	F	G
Product SKU	**Date**	**Release**	**# Criticals**	**New Criticals**	**Criticals Resolved**	**# Highs**
DQ	2024-03-28	2024.03	0	1	1	0
DQ	2024-04-15	2024.04	0	0	0	0
DQ	2024-05-09	2024.05	0	0	0	0
DQ	2024-06-21	2024.06	0	0	0	0
DQ	2024-07-29	2024.07	0	0	0	0

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2024.06

- Highlights
- Enhancements
- Fixes

- Limitations
- DQ Security
- Maintenance Updates

Release Information

- Release date of Data Quality & Observability Classic 2024.06: July 1, 2024
- Publication dates:
  - Release notes: June 6, 2024
  - Documentation Center: June 14, 2024

Highlights

Important
In the upcoming Data Quality & Observability Classic 2024.07 (July 2024) release, the classic UI will no longer be available.

Integration

Users without a Data Quality & Observability Classic license can now use the Quality tab on asset pages in the latest UI of Collibra Platform. Before Data Quality & Observability Classic 2024.06, unless you created data quality rules in Collibra Platform using the Data Quality & Observability Classic integration, the Quality tab would not populate and you could not aggregate data quality across any assets.

> **Note** To enable this Collibra Platform functionality, contact a Collibra Customer Success Manager or open a support ticket.

Additionally, we improved our security standards by adding support for OAuth 2.0 authentication when setting up an integration with Collibra Platform.

Important
Default SAML and SSL keystore are not supported. If you use a SAML or SSL keystore

> to manage and store keys and certificates, you must provide your own keystore file for both. When using both a SAML and SSL keystore, you only need to provide a single keystore file.

Enhancements

Connections

- When configuring an Amazon S3 connection and setting it as an Archive Breaking Records location, you can now use Instance Profile to authenticate it.
- When setting up a MongoDB connection, you can now use Kerberos TGT Cache to authenticate it.
- You can now use EntraID Service Principal to authenticate Databricks connections.
- Trino Pushdown connections now support Access Token Manager authentication.
- We upgraded the Teradata driver to version 20.0.0.20.

Explorer

- Explorer now fetches a new authentication token after the previous token expires to ensure seamless connectivity to your data source when using Access Token Manager or Password Manager to authenticate Pullup or Pushdown connections.

Jobs

- When using **DB connection security** and **DQ job security**, we added the security setting **Require Connection Access**, which requires users with ROLE_OWL_CHECK to have access to the connection they intend to run jobs on. When DB connection security and DQ job security are enabled, but Require Connection Access is not, users with ROLE_OWL_CHECK can run jobs to which they have dataset access.

Findings

- When exporting Outlier break records containing large values that were previously represented with scientific notation, the file generated from the **Export with Details**

option now exports the true format of these values to match the unshortened, raw source data.

APIs

- Admins and user managers can now leverage the POST /v2/deleteexternaluser call to remove external users.
- You can now add template and data class rules to a dataset with the POST /v3/rules/{dataset} call.
    - When you add template and data class rules to a dataset, the templates and data classes must already exist.
    - You can use the GET /v3/rules/{dataset} call to return all rules from a datase, then use the POST /v3/rules/{dataset} to add them to the dataset you specify. If you add these rules to a different dataset, you must update the dataset name in the POST call and any references of the dataset name in the rules.

Integration

- You can now map Data Quality & Observability Classic connections containing database views to their corresponding database view assets in Collibra Platform.
- The Details table on the Quality tab of asset pages is now keyboard navigable.

Pushdown

- When rule breaks are stored in the PostgreSQLMetastore with link IDs assigned, you can now download a CSV file containing the details of the rule breaks and link ID columns via the Findings page Rules tab Actions Rule Breaks modal.
- You can now archive rule break records for Trino Pushdown connections.
- You can now download a CSV file from the Findings page containing break records of rule breaks in the Metastore.

Platform

- Data Quality & Observability Classic now supports CentOS 9 and RedHat Enterprise Linux 9.

Latest UI

- All pages within the Data Quality & Observability Classic application with a blue banner are now set to the latest UI by default. Upon upgrade to this version, any REACT application configuration settings from previous versions will be overridden. The following pages are now set to the latest UI by default:
    - Login
    - Registration
    - Tenant Manager Login
    - Explorer
    - Profile
    - Findings
    - Rule Builder
    - Admin Connections

Fixes

Connections

- We enhanced the **-conf spark.driver.extraClassPath={driver jar}** Spark config to allow you to run jobs against Sybase datasets that reference secondary Oracle datasets. (ticket #129397)

Explorer

- When using Temp Files in the latest UI, you can now load table entries. (ticket #144174)
- Encrypted data columns with the BYTES datatype are now deselected and disabled in the Select Columns step, and all data displays correctly in Dataset Overview. (ticket #137738)
- When mapping source to target, we fixed an issue with the data type comparison, which previously caused incorrect Column Order Passing results. (ticket #139814, 140349)
- Preview data for remote file connections now displays throughout the application as expected. (ticket #139582,142538, 143876)

- We aligned the /v3/getsqlresult and /v2/getlistdataschemapreviewdbtablebycols endpoints so that Google BigQuery jobs with large numbers of rows do not throw errors when they are queried in Dataset Overview. (ticket #140730, 140915, 141515)

Rules

- We fixed an issue where rules did not display break records because extra spaces were added around the parentheses.
- When the file path of the S3 bucket used for break record archival has a timestamp from a previous run, the second run with the same runId no longer fails with an "exception while inserting break records" error message. (ticket #145702)
- We fixed an issue which resulted in an exception message when "limit" was present in a column name included in the query, for example, `select column_limit_test from public.abc`. (ticket #138356)

Alerts

- Dataset-level alerts with multiple conditions no longer send multiple alerts when only one of the conditions is met. (ticket #144655, 146177)

Scheduling

- After scheduling a job to run monthly in the latest UI, the new job schedule now saves correctly. (ticket #143484)

Jobs

- We fixed an issue where only the first page of job logs with multiple pages sorted in ascending or descending order. (ticket #139876)
- The Update Ts (update timestamp) on the Dataset Manager and Jobs page now match after rerunning a job. (ticket #141511)

Agent

- We fixed an issue that caused the agent to fail upon start-up when the SSL keystore password was encrypted. (ticket #140899)

Integration

- We fixed an issue where renaming an integrated dataset, then re-integrating it, caused the integration to fail because an additional job asset was incorrectly added to the object table. (ticket #140286, 140667, 140936, 143281, 143697, 144857)
- After editing an integration with a custom dimension that was previously inserted into the **dq_dimension** Metastore table, you can now select the custom dimension from the dropdown menu of the Dimensions tab of the Integrations page of the Admin Console. (ticket #137450, 145377)
- The Quality tab is now available for standard assets irrespective of the language. Previously, Collibra Platform instances in other languages, such as French, did not support the Quality tab. (ticket #140433)
- We fixed an issue where rules that reference a column with a name that partially matches another column, for example, "cell" and "cell_phone", were incorrectly mapped to both columns in Collibra Platform. (ticket #84983)
- The integration URL sent to Collibra Platform no longer references legacy Data Quality & Observability Classic URLs. (ticket #139764)

Pushdown

- When the `SELECT` statement of rules created on Snowflake Pushdown datasets uses mixed casing (for example, `Select`) instead of uppercasing, breaking records now generate in the rule break tables as expected. (ticket #143619, 147953)
- We fixed an issue where the username and password credentials for authenticating Azure Blob Storage connections did not properly save in the Metastore, resulting in job failure at runtime. (ticket #131026, 138844, 140793, 142635,145201)
- When a rule includes an @ symbol in its query without referring to a dataset, for example, `select * from @dataset where column rlike '@'`, the rule now passes syntax validation and no longer returns an error. (ticket #139670)

APIs

- When dataset security is enabled, users cannot call GET /v3/datasetdef or POST /v3/datasetdef. (ticket #138684)
- When `-profoff` is added to the command line and the job executes, `-datashapeoff` is no longer removed from the command line flags when `-profoff` is removed later. (ticket #140424)

Identity Management

- Users who have dataset access but not connection access can no longer access any dataset Explorer pages. (ticket #138684)

Latest UI

- We resolved an error when creating jobs with Patterns and Outlier checks with custom column references.
- When editing Dupes, columns are no longer deselected when you select a new one.
- Scorecards now support text wrapping so that scorecards with long names fit within UI elements in the latest UI. Additionally, Scorecards now have a character limit of 60 and an error message will display if a scorecard name exceeds it. (ticket #139208)
- Long meta tag names that exceed the width of the column on the Dataset Manager page now have a tooltip to display the full name when you hover your cursor over them.
- We resolved errors modifying existing mapping settings.
- We resolved an error when saving the Data Class when the Column Type is Timestamp.

Limitations

Platform

- Due to a change to the **datashapelimitui** admin limit in Data Quality & Observability Classic 2024.04, you might notice significant changes to the number of Shapes marked on the Shapes tab of the Findings page. While this will be fixed in Data Quality & Observability Classic 2024.06, if you observe this issue in your Data Quality & Observability Classic environment, a temporary workaround is to set the

datashapelimit admin limit on the Admin Console > Admin Limits page to a significantly higher value, such as 1000. This will allow all Shapes findings to appear on the Shapes tab.

- When Archive Break Records is enabled for Azure Databricks Pushdown connections authenticated over EntraID, the data preview does not display column names correctly and shows 0 columns in the metadata bar. Therefore, Archive Break Records is not supported for Azure Databricks Pushdown connections that use EntraID authentication.

Integration

- With the latest enhancement to column mapping, you can now successfully map columns containing uppercase letters and special characters, but columns containing periods cannot be mapped.

DQ Security

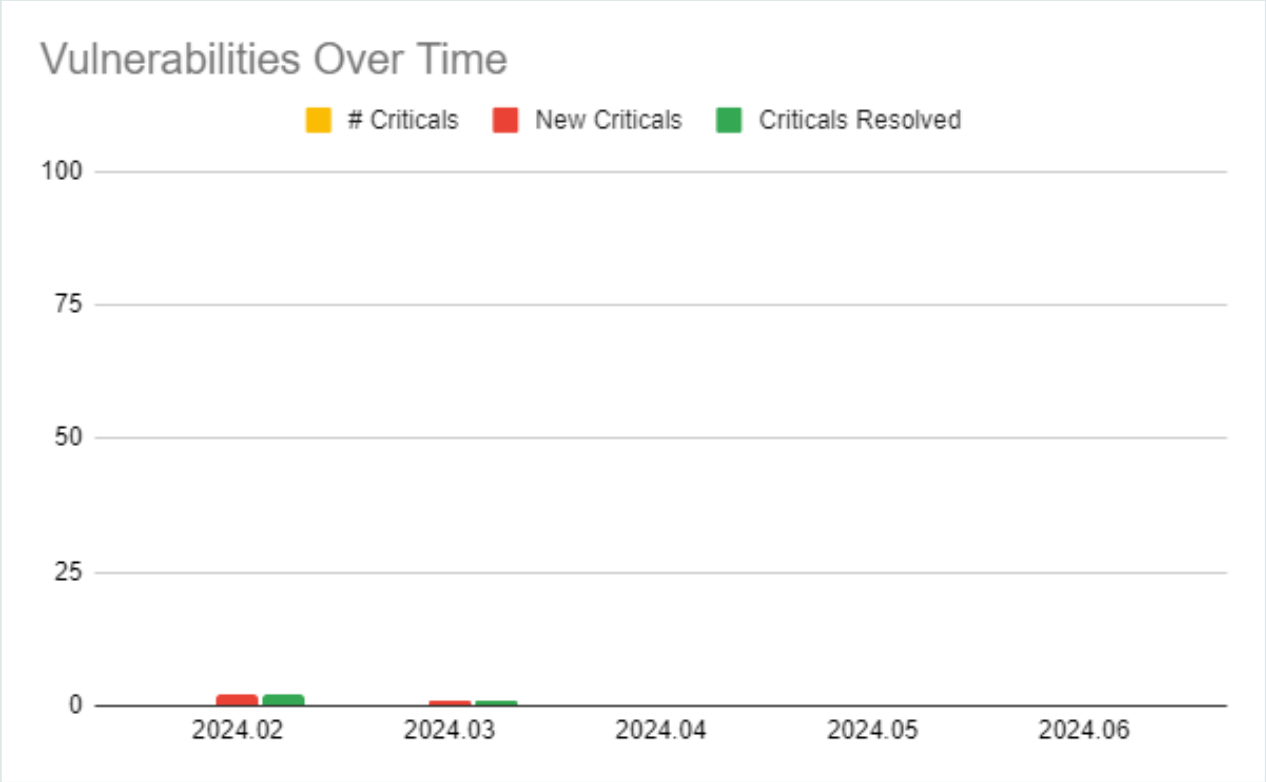
> **Important** A high vulnerability, CVE-2024-2961, was recently reported and is still under analysis by NVD. A fix is not available as of now. However, after investigating this vulnerability internally and confirming that we are impacted, we have removed the vulnerable character set, **ISO-2022-CN-EXT**, from our images so that it cannot be exploited using the `iconv` function. Therefore, we are releasing Data Quality & Observability Classic 2024.06 with this known CVE without an available fix, and we have confirmed that Data Quality & Observability Classic 2024.06 is not vulnerable.
>
> Additionally, a new vulnerability, CVE-2024-33599, was recently reported and is still under analysis by NVD. Name Service Cache Daemon (nscd) is a daemon that caches name service lookups, such as hostnames, user and group names, and other information obtained through services like DNS, NIS, and LDAP. Because nscd inherently relies on glibc to provide the necessary system calls, data structures, and functions required for its operation our scanning tool reported this CVE under glibc vulnerabilities. Since this vulnerability is only possible when ncsd is present and nscd is neither enabled nor available in our base image, we consider this vulnerability a false positive that cannot be exploited.

Show security metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2024-01-22	2024.02	0	2	2	0	6	JFrog Artifact
DQ	2024-03-28	2024.03	0	1	1	0	13	JFrog Artifact
DQ	2024-04-15	2024.04	0	0	0	0	13	JFrog Artifact
DQ	2024-05-09	2024.05	0	0	0	0	16	JFrog Artifact
DQ	2024-06-21	2024.06	0	0	0	0	12	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Maintenance Updates

2024.06.1

- The following functions are now tenant-agnostic:
  - The Rule Definitions page now loads correctly in Cloud Native and Standalone deployments.
  - The Rule Definitions page only shows the rules from the tenant you are signed into.

Release 2024.05

- Highlights
- Enhancements
- Fixes
- Limitations
- DQ Security
- Maintenance Updates

Release Information

- Release date of Data Quality & Observability Classic 2024.05: June 3, 2024
- Publication dates:
  - Release notes: April 22, 2024
  - Documentation Center: May 2, 2024

Highlights

> **Important**
> In the upcoming Data Quality & Observability Classic 2024.07 (July 2024) release, the classic UI will no longer be available.

Integration

For a more comprehensive bi-directional integration of Data Quality & Observability Classic and Collibra Platform, you can now view data quality scores and run jobs from the Data Quality Jobs modal on asset pages. You can find this modal via the **View Monitoring** link located on the **At a glance** pane to the right of the Quality tab on asset pages.

This significant enhancement strengthens the connection between Data Quality & Observability Classic and Collibra Platform, allowing you to compare data quality relations seamlessly without leaving the asset page. Whether you are a data steward, data

engineer, or another role in between, this enhanced integration breaks down barriers, empowering you with the ability to unlock data quality and observability insights directly within Collibra Platform.

> **Note** A fix for the issue that has prevented the use of the Quality tab on asset pages for users who do not have a Data Quality & Observability Classic license is scheduled for the third quarter (Q3) of 2024.

Pushdown

We are delighted to announce that Pushdown is now generally available, for three new data sources, including:

- Amazon Athena
- Amazon Redshift
- Trino

Additionally, Pushdown for SAP HANA and Microsoft SQL Server are now available for preview testing. Contact a Collibra CSM or apply directly to participate in private preview testing for SAP HANA Pushdown.

Pushdown is an alternative compute method for running DQ Jobs, where Data Quality & Observability Classic submits all of the job's processing directly to a SQL data warehouse. When all of your data resides in the SQL data warehouse, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirements of a DQ Job.

SQL Assistant for Data Quality

SQL Assistant for Data Quality is now generally available! This exciting tool allows you to automate SQL rule writing and troubleshooting to help you accelerate the discovery, curation, and visualization of your data. By leveraging SQL Assistant for Data Quality powered by Collibra AI, beginner and advanced SQL users alike can quickly discover key data points and insights and then convert them into rules.

Anywhere Dataset Overview is, so is SQL Assistant for Data Quality. This means you can unlock the power of Collibra AI from the following pages:

- Explorer
- Profile
- Findings
- Alert Builder

For the most robust SQL rule building experience, you can also find SQL Assistant for Data Quality when adding or editing a rule on the Rule Workbench page.

Further, we've added the ability to create an AI prompt for the frequency distribution of all values within a column. From the **Collibra AI** dropdown menu, select **Frequency Distribution** in the Advanced section and specify a column for Collibra AI to create a frequency distribution query. (idea #DCC-I-2639)

Enhancements

Capabilities

- We added a JSON tab to the Review step in Explorer and the Findings page to allow you to analyze, copy, and run the JSON payload of jobs.
- When exporting rule breaks with details, the column order in the .xlsx file now matches the arrangement in the Data Preview table on the Findings page. (idea #DCC-I-1656, DCC-I-2400)
    - Specifically, the columns in the export file are organized from left to right, following the same sequence as in the Data Preview table. The columns are sorted with the following priority:
        i. Column names starting with numbers.
        ii. Columns names starting with letters.
        iii. Column names starting with special characters.
- When a pullup has multiple link IDs in a break record export file, the link IDs will be parsed into multiple columns.
- To improve user experience when using the Rules table on the Findings page, we've locked the column headers and added a horizontal scrollbar to the rule breaks sub-table.

- You can now configure your user account to receive email notifications for your assignments by clicking your user avatar in the upper right corner of the Data Quality & Observability Classic application and selecting "Send me email notifications for my assignments" in the Notifications section.
- When a user who is not the dataset owner or has ROLE_ADMIN or ROLE_DATASET_MANAGER attempts to delete one or multiple datasets, they are prevented from a successful dataset deletion, and an error message displays to help inform them of the role requirements needed to delete datasets. (idea #DCC-I-1938)
- We added a new Attributes section with two filter options to the Dataset Manager. (idea #DCC-I-2155)
    - The **Rules Defined** filter option displays the datasets in your environment that contain rules only (not alerts).
    - The **Alerts Defined** filter option displays the datasets in your environment that contain alerts only (not rules).
    - When both filter options are selected, datasets that contain both rules and alerts display.
- With this release, we made several additional enhancements to SQL Assistant for Data Quality:
    - The Collibra AI dropdown menu now has improved organization. We split the available options into two sections:
        - **Basic**: For standard rule generation and troubleshooting suggestions.
        - **Advanced**: For targeted or otherwise more complex SQL operations.
    - You can now click and drag your cursor to highlight and copy specific rows and columns of the results table, click column headers to sort or highlight the entire column, and access multiple results pages through new pagination.
    - We improved the UI and error handling.
- You can now authenticate SQL Server connections using an Active Directory MSI client ID. This enhancement, available in both Java 11/Spark 3.4.1 and Java 8/Spark 3.2.2, better enables your team to follow Azure authentication best practices and InfoSec policies. For more information about configuration details, see the Authentication documentation for SQL Server.
- We added an automatic cleaner to clear the `alert_q` table of stale alerts marked as `email_sent = true` in the Metastore.
- We removed the license key from job logs.

Platform

- By specifying additional projects (AdditionalProjects) in the Connection URL, we now support multiple GCP projects in Google BigQuery connections. You can now specify the project ID in the Connection URL (limited to 1 additional project ID). With this enhancement, you no longer need to append the project ID in the command line.
- When running Google BigQuery jobs via the /v3/jobs/run API, the dataDef updates with the correct `-lib` and `-srclib` parameters, and the jobs run successfully.
- The names of all out-of-the-box sensitive labels now begin with "OOTB_". This enhancement allows you to define your own sensitive labels with names that were previously reserved, such as PII, PHI, and CUI.

  > **Important** If you upgrade to Data Quality & Observability Classic 2024.05 and then roll back to a previous version, you will receive a unique constraint conflict error, as the sensitive label enhancement required a change to the Metastore.

- The following Jobs APIs now return the source rule breaks file containing the SQL statement for breaks jobs in JSON, CSV, or SQL:
  - /v3/jobs/{dataset}/{runDate}/breaks/rules
  - /v3/jobs/{dataset}/{runDate}/breaks/outliers
  - /v3/jobs/{dataset}/{runDate}/breaks/dupes
  - /v3/jobs/{dataset}/{runDate}/breaks/shapes
  - /v3/jobs/{jobId}/breaks/rules
  - /v3/jobs/{jobId}/breaks/outliers
  - /v3/jobs/{jobId}/breaks/dupes
  - /v3/jobs/{jobId}/breaks/shapes
- We've updated or enhanced the following API endpoints:

Method	Endpoint	Controller Name	Description
POST	/v3/rules/{dataset}	rule-api	After using GET /v3/rules to return all rules in your environment, you can now use POST /v3/rules/{dataset} to migrate them to another environment. When settings are changed and you use POST /v3/rules/{dataset} again, those rules (with the same name) are updated.
GET	/v3/datasetDefs/{dataset}	dataset-def-api	We've made the following enhancements to the GET /v3/datasetDefs/{dataset} API: • We've added job scheduling information to the dataset def to allow you to GET and POST this information along with the rest of the dataset definition. • We've added the outlier weight configs to the dataset def. • You can now use the GET /v3/datasetDefs/{dataset} API to return a dataset's meta tags. • We've restructured the JobScheduleDTO to make job scheduling more intuitive when using the /v3/datasetDefs/{dataset} API.

Method	Endpoint	Controller Name	Description
POST	/v3/datasetDefs/find	dataset-def-api	We've updated the following parameter names for consistency with the latest Collibra DQ UI: • **connectiontype** is now **connectiontypes** • **dataclass** is now **dataClasses** • **datacategory** no longer displays • **businessUnitIds** is now **businessUnitNames** • **dataConceptIds** is now **dataCategoryNames** • **sensitivityIds** is now **sensitivityLabels** Additionally, this API returns specific filtered arrays of datasetDefs. Parameter descriptions: • "limit": 0, = The maximum number of records returned • "offset": 0, = The number of records that should be skipped from the beginning and can be used to return the next 'pages' or number of results after calling the API in sequence
POST	/v3/datasetDefs	dataset-def-api	You can now use the POST /v3/datasetDefs/{dataset} API to add meta tags to a dataset.

Method	Endpoint	Controller Name	Description
DELETE	/v3/datasetDefs	dataset-def-api	When removing a dataset using the DELETE /v3/datasetdef API, you can now successfully rename another dataset to the name of the deleted dataset.
POST	/v2/datasetDefs/migrate	controller-dataset	You can now add a Pullup dataset def to create a dataset record in the Dataset Manager **without** running the job or setting a job schedule. This is useful when migrating from a source environment to a target environment. **Note** This API endpoint is supported only in Pullup mode. It is not supported in Pushdown mode.
GET	/v2/assignment-q/find-all-paging-datatables	controller-assignment-q	We've added an updateTimestampRange parameter to the GET /v2/assignment-q/find-all-paging-datatables API to allow for the filtering of assignments records based on timestamp updates.

Integration

- We improved the connection mapping when configuring the integration by introducing pagination for tables, columns, and schemas.
- For improved security when sharing data between applications, we have temporarily removed the Score Details attribute from the Collibra Platform integration and the JSON payload.

Fixes

Capabilities

- The Dataset Overview, Findings, Profile, and Rules pages in the latest UI now correctly display the number of rows in your dataset. Previously, the rows displayed correctly in the job logs but did not appear on the aforementioned pages. (ticket #137230, 137979, 140203)
- When using remote file connections with Livy enabled in the latest UI, files with the same name load data content correctly. We fixed an issue where data from the first file persisted in the second file of the same name.
- We fixed an issue where renaming a dataset using the same characters with different casing returned a success message upon saving, but still reflected the old dataset name. For example, an existing dataset renamed "EXAMPLE_DATASET" from "example_ dataset" now updates correctly. (ticket #139384)
- When creating jobs on S3 datasets based on data from CSV files with pipe delimited values, the delimiter no longer reverts from **Pipe (|)** to **Comma (,)** when you run the job. (ticket #132097)
- We fixed an issue with the Edit Schedule modal on the latest UI where both **Enabled** and **Disabled** displayed at once. (ticket #139207)

Platform

- We fixed an issue where the username and password credentials for authenticating Azure Blob Storage connections did not properly save in the Metastore, resulting in job failure at runtime. (ticket #131026, 138844, 140793, 142635,145201)
- When a rule includes an @ symbol in its query without referring to a dataset, for example, `select * from @dataset where column rlike '@'`, the rule now passes syntax validation and no longer returns an error. (ticket #139670)

Integration

- You can now map columns containing uppercase letters or special characters from Google BigQuery, Amazon Athena, Amazon Redshift, Snowflake, and PostgreSQL datasets created in Data Quality & Observability Classic to column relations in Collibra

Platform. (ticket #133280)
- We fixed an issue where integrated datasets did not load correctly on the Dataset Manager page. Instead, a generic error message appeared on the Dataset Manager without loading any datasets. (ticket #136303, 140286)
- We fixed an issue where the dimension cards did not display when using the Quality tab on Column and Rule Asset pages. (ticket #122949)

Pushdown

- We updated some of the backend logic to allow the Archive Break Records option in the Connections modal to disable the Archive Break Records options on the Settings modal on the Explorer page. (ticket #137396)
- We added support for special characters in column names. (ticket #135383)

Latest UI

- We added upper and lower bound columns to the export with details file for Outliers.
- We fixed the ability to clear values in the Sizing step when manually updating job estimation fields during job creation.
- We improved the ability to update configuration settings for specific layers in the job creation process.
- We fixed intermittent errors when loading text and Parquet files in the job creation process.
- We added the correct values to the Day of Month dropdown menu in the Scheduler modal.

Limitations

Platform

- Due to a change to the **datashapelimitui** admin limit in Data Quality & Observability Classic 2024.04, you might notice significant changes to the number of Shapes marked on the Shapes tab of the Findings page. While this will be fixed in Data Quality & Observability Classic 2024.06, if you observe this issue in your Data Quality & Observability Classic environment, a temporary workaround is to set the

datashapelimit admin limit on the Admin Console > Admin Limits page to a significantly higher value, such as 1000. This will allow all Shapes findings to appear on the Shapes tab.

Integration

- With the latest enhancement to column mapping, you can now successfully map columns containing uppercase letters and special characters, but columns containing periods cannot be mapped.

DQ Security

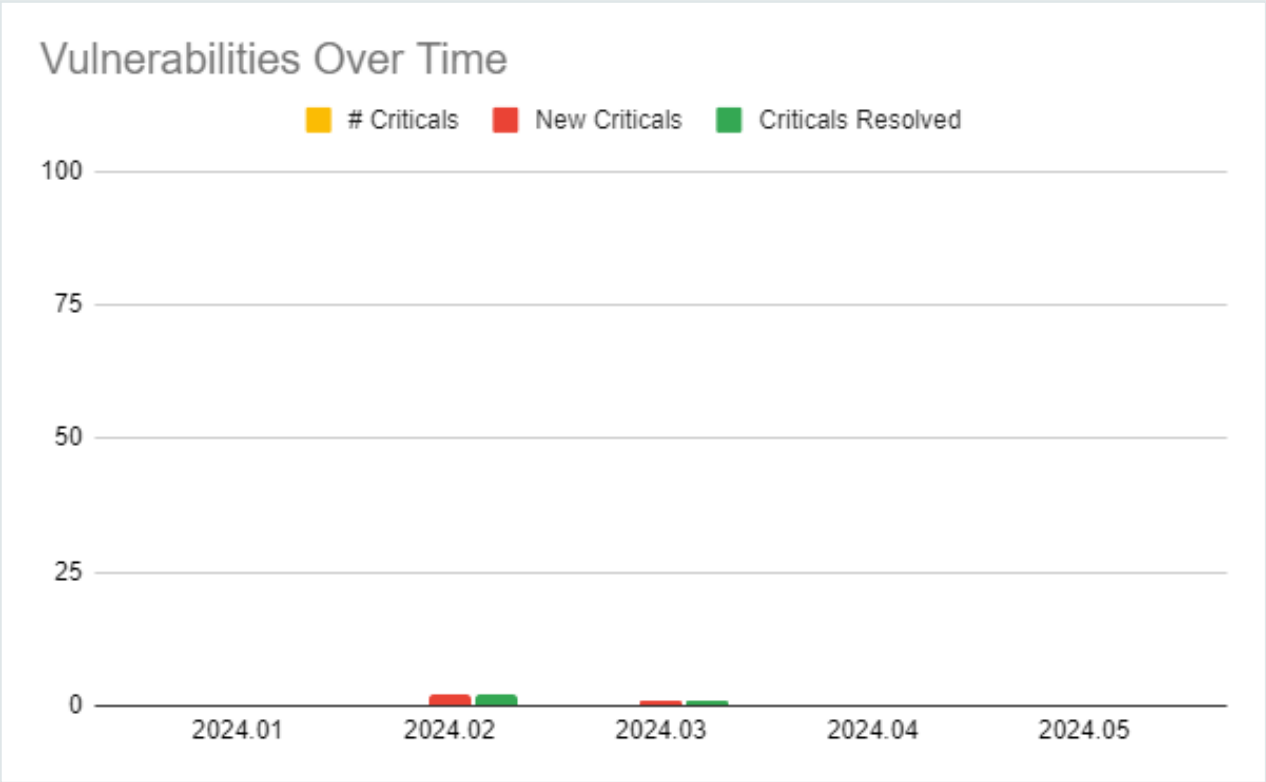
> **Important** A high vulnerability, CVE-2024-2961, was recently reported and is still under analysis by NVD. A fix is not available as of now. However, after investigating this vulnerability internally and confirming that we are impacted, we have removed the vulnerable character set, **ISO-2022-CN-EXT**, from our images so that it cannot be exploited using the `iconv` function. Therefore, we are releasing Data Quality & Observability Classic 2024.05 with this known CVE without an available fix, and we have confirmed that Data Quality & Observability Classic 2024.05 is not vulnerable.
>
> Additionally, a new vulnerability, CVE-2024-33599, was recently reported and is still under analysis by NVD. Name Service Cache Daemon (nscd) is a daemon that caches name service lookups, such as hostnames, user and group names, and other information obtained through services like DNS, NIS, and LDAP. Because nscd inherently relies on glibc to provide the necessary system calls, data structures, and functions required for its operation our scanning tool reported this CVE under glibc vulnerabilities. Since this vulnerability is only possible when ncsd is present and nscd is neither enabled nor available in our base image, we consider this vulnerability a false positive that cannot be exploited.

Show security metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2024-01-03	2024.01	0	0	0	0	3	JFrog Artifact
DQ	2024-01-22	2024.02	0	2	2	0	6	JFrog Artifact
DQ	2024-03-28	2024.03	0	1	1	0	13	JFrog Artifact
DQ	2024-04-15	2024.04	0	0	0	0	13	JFrog Artifact
DQ	2024-05-09	2024.05	0	0	0	0	16	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Maintenance Updates

2024.05.2

- When editing an existing scheduled dataset and re-running it from Explorer, the job no longer fails with an "Invalid timeslot selected" error. (ticket #149549)
- When using the GET /v3/datasetDefs/{dataset} call to return a dataset with a scheduled run, then update it with the POST /v3/datasetDefs call or modify the name of the dataset in the same POST call, you no longer need to manually remove the `"jobSchedule": {}` element and the API calls are successful.

Release 2024.04

- Enhancements
- Fixes

- Limitations
- DQ Security

Release Information

- Release date of Data Quality & Observability Classic 2024.04: April 29, 2024
- Publication dates:
  - Release notes: April 4, 2024
  - Documentation Center: April 4, 2024

Enhancements

Capabilities

- You can now authenticate SQL Server connections using Active Directory Password and Active Directory Service Principal. This enhancement, available in both Java 11/Spark 3.4.1 and Java 8/Spark 3.2.2, allows you to use Azure AD-based Synapse SQL and Azure Service Principal authentication in Data Quality & Observability Classic, to further enable your team to follow Azure authentication best practices and InfoSec policies.
- We improved some of the ways you can work with the Scorecards page:
  - The Page dropdown menu is now sorted alphabetically.
  - You can now rename scorecard pages without the need to recreate them from scratch.
  - You can now change the order in which scorecards are displayed on a scorecards page by clicking the up or down arrows to the left of a scorecard.
- Archive Break Records for both Pushdown and Pullup jobs now parses linkId values in the Sample File Preview and Rule Breaks Preview on the Findings page and in the downloadable CSV file.
- Running a job with a SQLF rule that references a secondary dataset or uses @t1 no longer flashes an extra record in the job log or on the Jobs page with an Agent Id of 0. With this enhancement, only a single job record displays on the Jobs page. (idea #DCC-I-2413)
- When using the Dataset Overview, you can now click Download Results to download a CSV file containing the contents of the Results table.

Platform

- We improved some of the ways you can create or edit alerts:
  - To enhance the fluidity of batch alerting from the Status Alert and Condition Alert modals, you can now select the new **Batch** option to display the **Batch Name** field, where you can search, select, or create alert batches.
  - With the **Batch** option selected, the **Alert Recipient** field locks. To edit this field, you can click the 🔒 to unlock it. When the **Batch** option is not selected, the **Alert Recipient** field remains unlocked and editable.

Fixes

Capabilities

- We improved the Data Shape Granular setting to include more string length values in Shapes findings. (ticket #135351)
- When joining datasets created from two different data source connections with Kerberos keytab as the authentication type, we fixed an issue that prevented the secondary dataset from loading because its Kerberos authentication was incorrectly passed. (ticket #131309)
  - To ensure that the secondary dataset is passed correctly during Kerberos authentication, add `-jdbcprinc` and `-jdbckeytab` to the Agent configuration's Free Form (Appended) section. For example, `-jdbckeytab /tmp/keytab/dq-user.keytab -jdbcprinc dq_user@example.collibra.com`
- We added the ability to export all job schedule records from the Jobs Schedule page, using the Export All option. Previously, the ability to export job schedule records was limited to up to 20 records per page. (ticket #136970)
- We added the following enhancements to the Findings page when data quality findings exceed certain thresholds. (ticket #136922)
  - When there are more than 9,999 findings of any data quality layer, the value displayed in the badge on the corresponding findings tab will round to the nearest thousand with a +. For example, 12,345 will display as 12K+.

    - When there are more than 999,000 findings of any data quality layer, the value displayed in the badge on the corresponding findings tab will always display as 1M+. For example, 1,234,567 will display as 1M+.
    - When a value is truncated, you can hover your cursor over the badge to display the exact number of findings.
- When using SAML SSO to sign into multi-tenant Data Quality & Observability Classic environments, the SAML Enabled Tenants dropdown menu no longer shows the Tenant Name. Instead, the dropdown menu now shows the Tenant Display Name. (ticket #137865)
- We removed the runId from the "Findings - Most Recent Run" link in alert emails to correctly take you to the most recent run of your job when you click the link.
- When the "Findings - Run which Produced the Alert" link in alert emails contains a runId and a timestamp in the URL, you will be taken to that specific job runId and timestamp when you click the link.

Platform

- We introduced pagination to the Rule Summary page while limiting each page to displaying a maximum of 25 records. Previously, the Rule Summary page only displayed 25 records on a single page, even when the number of records should have exceeded 25. (ticket #140229)
- When an admin sets a limit for the **datashapelimitui** setting from Admin Limits, the Findings page no longer displays Shapes findings beyond that limit. (ticket #129091)
- The Role Management page in the latest UI now allows admins to set the access rights of roles and users when associating them with specific datasets (ticket #138489)
- The Dataset Manager page now loads correctly when the alias name of a dataset is null. (ticket #133400)
- When filtering by row count on the Dataset Manager page, the results included in the filter no longer include daily row counts that exceed the range you select. (ticket #133453)
- JSON files from Amazon S3 connections no longer fail to load when Livy is not enabled.
- When large datasets (for example, one with more than 100 million records) timeout with a 504 error response when loading the table in Explorer, an error message appears in the Explorer UI with details about the error. (ticket #133530)

Pushdown

- Databricks Pushdown now supports ANSI-compliant SQL on the server side. (ticket #136562)
- The out-of-the-box Data Category, Currency_CD, no longer counts the number of null and empty values as part of the underlying SQL query. (ticket #133578)

Latest UI

- We added support for multiple pages of results on the Rule Summary page.
- We improved the performance of the Pulse View when loading large amounts of data.
- We updated the display of the Pulse View when scrolling.
- The Command Line input field on the Findings page now supports vertical scrolling.
- The Source tab on the Findings page now displays all labels for Cell results.
- We updated the Job Schedule export function to include all jobs instead of limiting them to 20.
- We added the ability to assign ACL to datasets from the Role Management page in the Admin Console.
- We added Copy and Export buttons to the View AR modal on the Findings page for Behaviors.
- We fixed the ability to change the upper and lower bound values for AdaptiveRules.
- We now display the exact number of results on the tabs for each of the layers on the Findings page instead of "99+" for all values over 99.
- The Rules tab now updates automatically when you add Quick Rules from the Data Preview section.
- Success messages now appear on the Findings page when you use the Validate, Invalidate, and Resolve functions.

Limitations

Platform

- Due to a change to the **datashapelimitui** admin limit in this release, you might notice significant changes to the number of Shapes marked on the Shapes tab of the Findings page. While this will be fixed in Data Quality & Observability Classic 2024.06, if you

observe this issue in your Data Quality & Observability Classic environment, a temporary workaround is to set the **datashapelimit** admin limit on the Admin Console > Admin Limits page to a significantly higher value, such as 1000. This will allow all Shapes findings to appear on the Shapes tab.

DQ Security

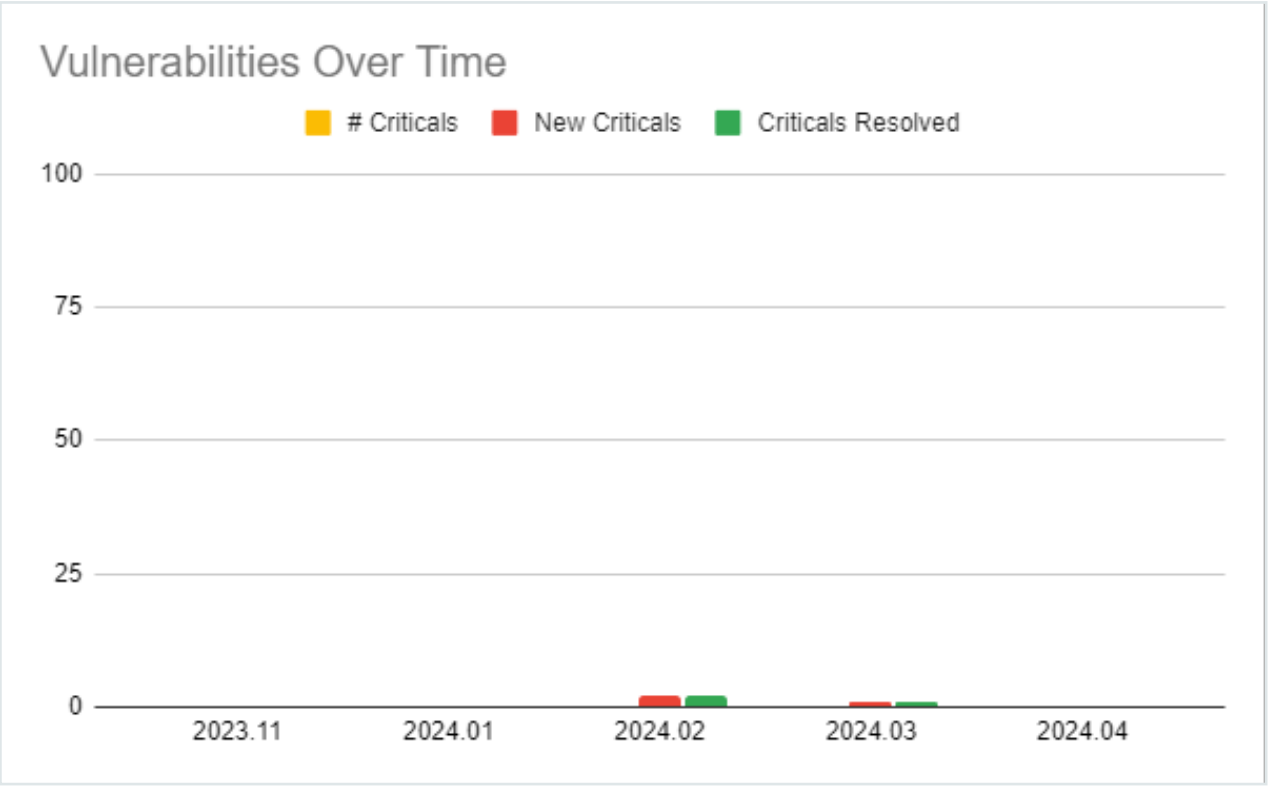
> **Important** A new high vulnerability, CVE-2024-2961, was recently reported and is still under analysis by NVD. A fix is not available as of now. However, after investigating this vulnerability internally and confirming that we are impacted, we have removed the vulnerable character set, **ISO-2022-CN-EXT**, from our images so that it cannot be exploited using the `iconv` function. Therefore, we are releasing Data Quality & Observability Classic 2024.04 with this known CVE without an available fix, and we have confirmed that Data Quality & Observability Classic 2024.04 is not vulnerable.

Show security metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-11-13	2023.11	0	0	0	0	3	JFrog Artifact
DQ	2024-01-03	2024.01	0	0	0	0	3	JFrog Artifact
DQ	2024-01-22	2024.02	0	2	2	0	6	JFrog Artifact
DQ	2024-03-28	2024.03	0	1	1	0	13	JFrog Artifact
DQ	2024-04-15	2024.04	0	0	0	0	13	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2024.03

- Enhancements
- Fixes
- Limitations
- DQ Security

Release Information

- Release date of Data Quality & Observability Classic 2024.03: April 1, 2024
- Publication dates:
  - Release notes: March 8, 2024
  - Documentation Center: March 11, 2024

Enhancements

Capabilities

- Admins can now view monthly snapshots of the total number of active datasets in the new Datasets column on the Admin Console ▸ Usage page. Additionally, Columns statistics are no longer counted twice when you edit and re-run a dataset with different columns.
- Admins can now optionally remove “Data Quality & Observability Classic” from alert email subjects. When removing the Data Quality & Observability Classic from the subject, you must fill in all the alert SMTP details to use the alert configuration checkboxes on the screen. By removing Data Quality & Observability Classic from the alert subject, you can now set up your own services to automatically crawl Collibra DQ email alerts.
- Dataset-level alerts, such as Job Completion, Job Failure, and Condition Alerts, as well as global-level alerts for both Pullup and Pushdown Job Failure now send incrementally from the auto-enabled alert queue.

  > **Important** As a result of this enhancement, following an upgrade to Data Quality & Observability Classic 2024.03, any old, unsent alerts in the alert queue will be sent automatically. This is a one-time event and these alerts can be safely ignored.

  - We've also added new functionality where, when an alert fails to send, it will still be marked as `email_sent = true` in the `alert_q` Metastore table. However, no email alerts will be sent as a result of this. An enhancement to automatically clean the `alert_q` table of stale alerts marked as `email_sent = true` is scheduled for the upcoming Data Quality & Observability Classic 2024.05 release.
- We've optimized internal processing when querying Trino connections by passing the catalog name in the Connection URL. The catalog name can be set by creating a Trino connection from Admin Console ▸ Connections and adding `ConnCatalog=$catalogName` to the Connection URL.
- We've added a generic placeholder in the Collibra DQ Helm Charts to allow you to bring in additional external mount volumes to the DQ Web pod. Additionally, when enabled in the DQ check logs and external mount volumes, the persistent volumes provisioned to set the storage class of the persistent volume claims for Collibra DQ now include a

placeholder value to allow you to specify the storage class type. This provides an option to bring Azure vault secrets as external mount volumes into the DQ Web pod.
Show steps
The following shows the Helm **values.yaml** file to mount the Kubernetes secret "extra-secret" at the DQ Web pod location "/mnt":

```
...
                web:
                extraStorage:
                volumes:
                - name: extra-secret
                secret:
                defaultMode: 493
                secretName: extra-secret
                volumeMounts:
                - mountPath: /mnt
                name: extra-secret
            ...
```

You can set the storage class for the DQ Web PVC with the following Helm value:

```
--set global.persistence.web.storageClassName=gp3
```

Platform

- We now support Cloud Native Collibra DQ deployments on OpenShift Container Platform 4.x.
- When using a proxy, SAML does not support URL-based metadata. It can only support file-based metadata. To ensure this works properly, set the property `SAML_METADATA_USER_URL=false` in the **owl-env.sh** file for Standalone deployments or **DQ-web ConfigMap** for Cloud Native.
- We removed the Profile Report, as all of its information is also available on the Dataset Findings Report.

Fixes

Capabilities

- We fixed an issue that caused Redshift datasets that referenced secondary Amazon S3 datasets to fail with a 403 error. (ticket #132975)
- On the Profile page of a dataset, the number of TopN shapes now matches the total number of occurrences of such patterns. (ticket #133817)
- When deleting and renaming datasets from the Dataset Manager, you can now rename a dataset using the name of a previously deleted one. (ticket #132799)
- When renaming a dataset from the Dataset Manager and configuring it to run on a schedule, the scheduled job no longer combines the old and new names of the dataset when it runs. (ticket #132798)
- On the Role Management page in the latest UI, roles must now pass validation to prevent ones that do not adhere to the naming requirements from being created. (ticket #133497)
- When creating a Trino dataset with Pattern detection enabled, date type columns are no longer cast incorrectly as varchar type columns during the historical data load process. (ticket #132478)
- On the Connections page in the latest UI, the input field now automatically populates when you click Assume Role for Amazon S3 connections. (ticket #132323, 132423)
- On the Findings page, when reviewing the Outliers tab, the Conf column now has a tooltip to clarify the purpose of the confidence score. (ticket #129768)
- When DQ Job Security is enabled and a user does not have ROLE_OWL_CHECK assigned to them, both Pullup and Pushdown jobs now show an error message “Failed to load job to the queue: DQ Job Security is enabled. You do not have the authority to run a DQ Job, you must be an Admin or have the role Role_Owl_Check and be mapped to this dataset.” (ticket #133623)
- When creating a DQ job on a root or nested Amazon S3 folder without any files in it, the system now returns a more elegant error message. (ticket #134187)
- On the Profile page in the latest UI, the WHERE query is now correctly formed when adding a valid values quick rule. (ticket #133455)

Platform

- When viewing TopN values, the previously encrypted valid values now decrypt correctly. (ticket #131951)
- When a user with ROLE_DATA_GOVERNANCE_MANAGER edits a dataset from the Dataset Manager, the Metatags field is the only field such a user can edit and have their updates pushed to the Metastore. (ticket #132468, 135889)
- We fixed an issue with the /v3/datasetDefs/{dataset} and /v3/datasetDefs/{dataset}/cmdLine APIs that caused the -lib, -srclib, and -addlib parameters to revert in the command line. (ticket #131281)

Pushdown

- When casting from one data type to another on a Redshift dataset, the job now runs successfully without returning an exception message. (ticket #128718)
- When running a job with outliers enabled, we now parse the source query for new line characters to prevent carriage returns from causing the job to fail. (ticket #132322)
- The character limit for varchar columns is now 256. Additionally, to prevent jobs from failing when a varchar column exceeds the 256-character limit. (ticket #131355)
- BigQuery Pushdown jobs on huge datasets of more than 10GB no longer fail with a "Response too large" error. (ticket #134643, 135504)
- When running a Pushdown job with archive break records enabled without a link ID assigned to a column, a helpful warning message now highlights the requirements for proper break record archival. (ticket #132545)
- The Source Name parameter on the Connections template for Pushdown connections now persists to the /v2/getcatalogandconnsrcnamebydataset API call as intended. (ticket #132334)

Limitations

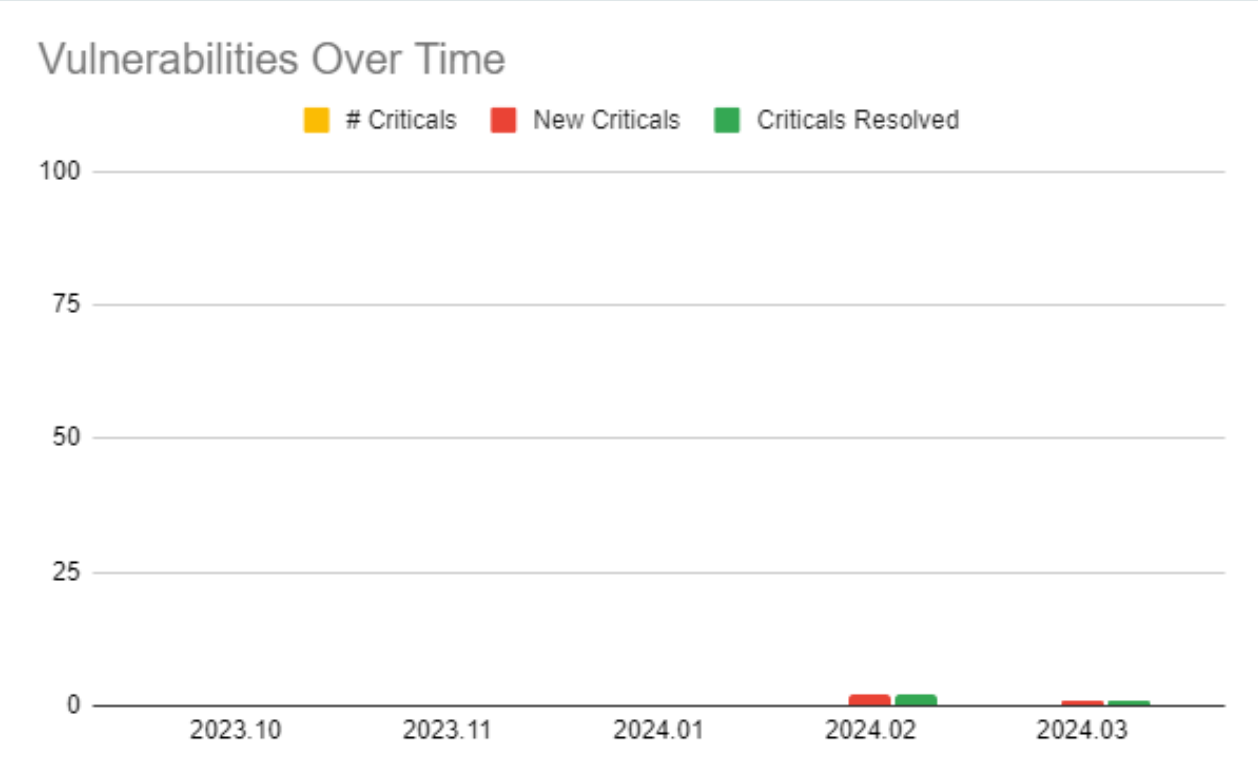
- TopN values from jobs that ran before enabling encryption on the Collibra DQ instance are not decrypted. To decrypt TopN values after enabling encryption, re-run the job once encryption is enabled on your Collibra DQ instance.

DQ Security

Show security metrics
The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-10-30	2023.10	0	0	0	0	1	JFrog Artifact
DQ	2023-11-13	2023.11	0	0	0	0	3	JFrog Artifact
DQ	2024-01-03	2024.01	0	0	0	0	3	JFrog Artifact
DQ	2024-01-22	2024.02	0	2	2	0	6	JFrog Artifact
DQ	2024-03-28	2024.03	0	1	1	0	13	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2024.02

- Highlights
- Migration Updates

- Enhancements
- Features in preview
- Fixes
- Known Limitations
- DQ Security

Release Information

- Release date of Data Quality & Observability Classic 2024.02: February 26, 2024
- Publication dates:
  - Release notes: January 22, 2024
  - Documentation Center: February 4, 2024

Highlights

Archive Break Records

Pullup
When rule breaks are stored in the PostgreSQL Metastore with link IDs assigned, you can now download a CSV file containing the details of the rule breaks and link ID columns via the Findings page ▸ Rules tab ▸ Actions ▸ Rule Breaks modal.

Pushdown
In order to completely remove sensitive data from the PostgreSQL Metastore, you can now enable **Data Preview from Source** in the Archive Break Records section of the Explorer Settings. When you enable Data Preview from Source, data preview records do not store in the PostgreSQL Metastore.

Previews of break records associated with Rules, Outliers, Dupes, and Shapes breaks on the Findings page reflect the current state of the records as they appear in your data source. With this option disabled, the preview records that display in the web app are snapshots of the PostgreSQL Metastore records at runtime. This option is disabled by default.

Additionally, with Archive Break Records enabled and a link ID column assigned, you can

now download a CSV or JSON file containing the details of the breaks and link ID columns via the Findings page ▸ Rules, Outliers, Dupes, or Shapes tab ▸ Actions ▸ Rule Breaks modal.

Lastly, when Archive Break Records is enabled, you can now optionally enter an alternative dataset-level schema name to store source break records, instead of the schema provided in the connection.

Important
Changes for Kubernetes Deployments
As of Collibra DQ version 2023.11, we've updated the Helm Chart name from `owldq` to `dq`. For Helm-based upgrades, point to the new Helm Chart while maintaining the same release name. Please update your Helm install command by referring to the renamed parameters in the values.yaml file. It is also important to note that the pull secret has changed from `owldq-pull-secret` to `dq-pull-secret`.

Further, following deployment, your existing remote agent name will change. For example, if your agent name is owldq-owl-agent-collibra-dq, the new agent name will be dq-agent-collibra-dq. If your organization uses APIs for development, ensure that you upgrade AGENT name configurations in your environments.

Lastly, when you deploy using the new Helm Charts, new service (Ingress/Load Balancer) names are created. This changes the IP address of the service and requires you to reconfigure your Load Balancer with the new IP.

Please see the expandable sections below for more details about specific changes.

Show renamed keys

Old Key	Renamed Key
global.version.owl	global.version.dq
global.image.owlweb	global.image.web
global.image.owlagent	global.image.agent

Show default value changes

Parameter	Old Default Value	New Default Value
global.mainChart	owldq	dq
global.image.pullSecret.name	owldq-pull-secret	dq-pull-secret
global.web.key.secretName	owldq-ssl-secret	dq-ssl-secret
global.image.web.name	owl-web	dq-web
global.image.agent.name	owl-agent	dq-agent
global.image.livy.name	owl-livy	dq-livy
global.image.spark.name	spark	dq-spark

> **Note**
> If your organization has a standalone deployment of Collibra DQ with SSL enabled for DQ Web, and both DQ Web and DQ Agent are on the same VM or server, we recommend upgrading directly to Collibra DQ 2023.11.3 patch version instead of 2023.11. For more information, see the Maintenance Updates section below.

Migration Updates

> **Important** This section only applies if you are upgrading from a version older than Collibra DQ 2023.09 on Spark Standalone. If you have already followed these steps during a previous upgrade, you do not have to do this again.

We have migrated our code to a new repository for improved internal procedures and security. Because owl-env.sh jar files are now prepended with `dq-*` instead of `owl-*`, if you have automation procedures in place to upgrade Collibra DQ versions, you can use the RegEx replace `regex=r"owl-.*-202.*-SPARK.*\.jar|dq-.*-202.*-SPARK.*\.jar"` to update the jars.

Additionally, please note the following:

- **Standalone Upgrade Steps**
  When upgrading from a Collibra DQ version before 2023.09 to a Collibra DQ version

2023.09 or later on Spark Standalone, the upgrade steps have changed.

Show steps

1. Open a terminal session.
2. Move the old jars from the owl/bin folder with the following commands.

```
mv owl-webapp-<oldversion>-<spark301>.jar /tmp
mv owl-agent-<oldversion>-<spark301>.jar /tmp
mv owl-core-<oldversion>-<spark301>.jar /tmp
```

3. Copy the new jars into the owl/bin folder from the extracted package.

```
mv dq-webapp-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-agent-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-core-<newversion>-<spark301>.jar
/home/owldq/owl/bin
```

4. Copy the latest `owlcheck` and `owlmanage.sh` to /opt/owl/bin directory.

> **Tip** You may also need to run `chmod +x owlcheck owlmanage.sh` to add execute permission to `owlcheck` and `owlmanage.sh`.

5. Start the Collibra DQ Web application.

```
./owlmanage.sh start=owlweb
```

6. Start the Collibra DQ Agent.

```
./owlmanage.sh start=owlagent
```

7. Validate the number of active services.

```
ps -ef | grep owl
```

Enhancements

Capabilities

- When using the Dataset Overview, you can now click the **-q** button to load the contents of the dataset source query into the SQL editor.
- When using the Dataset Overview, you can now use Find and Replace to find any string in the SQL editor and replace it with another.
- When a finding is assigned to a ServiceNow incident and the ServiceNow connection has Publish Only enabled on the ServiceNow Configuration modal in the Admin screens, this finding record is pushed to ServiceNow as it was in previous versions, but the status is no longer linked. This means that you can adjust the statuses on the ServiceNow Incident as you wish and the finding in DQ. Whereas before, the ServiceNow Incident had to be closed in order for the DQ finding to be resolved.
- From the Settings page in Explorer, you can now select the **Core Fetch Mode** option to allow SQL queries with spaces to run successfully. When selected, this option adds `-corefetchmode` to the command line to enable the core to fetch the query from the load options table and override the `-q`.
- When attempting to connect to NetApp or Amazon S3 endpoints in URI format with the **HTTPS** option selected, you can now add the following properties to the Properties tab on Amazon S3 connection templates to successfully create connections:
  - For Amazon S3 endpoint URI: **s3-endpoint=s3**
  - For NetApp: **s3-endpoint=netapp**
- When using the Pulse View, you can now select a few new options from the Show Failed dropdown menu, including Failed Job Runs and Failing Scores. Previously, the Show Failed option only displayed job runs that previously failed.
- You can now use uppercasing in secondary datasets and rule references.
- You can now configure arbitrary users as part of the root user group for DQ pod deployment.
- Due to security concerns, we have removed the license key from the job logs.

Platform

- We've upgraded the following drivers to their latest versions:

Driver	Version
Databricks	2.6.36
Google BigQuery	1.5.2.1005
Dremio	24.3.0
Snowflake	3.14.4

- You can now enable multi-tenancy for a notebook API.
- We now apply the same Spark CVE fixes that are applied to Cloud Native deployments of Collibra DQ to Standalone deployments.

Pushdown

- From the Settings page on Explorer, you can now select **Date** or **DateTime (TimeStamp)** from the Date Format dropdown menu to substitute the runDate and runDateEnd at runtime.
- To conserve memory and processing resources, the results query now rolls up outliers and shapes, and the link IDs no longer persist to the Metastore.
- All rules from the legacy Rule Library function correctly for Snowflake and Databricks Pushdown except for Having_Count_Greater_Than_One and Two_Decimal_Places when Link ID is enabled. See the Known Limitations section below for more information.
- You can now use cross-dataset rules that traverse across connections on the same data source.

Features in preview

Collibra AI

- SQL Assistant for Data Quality (in preview) now allows you to select between four new options to generate prompts for:

    - **Categorical**: Writes a SQL query to detect categorical outliers.
    - **Dupe**: Writes a SQL query to detect duplicate values.
    - **Record**: Writes a SQL query to find values that appear on a previous day but not for the next day.
    - **Pattern**: Writes a SQL query to find infrequent combinations that appear less than 5 percent of the time in the columns you specify.

DQ Integration

- The new Quality tab is now available as part of the latest UI updates for Asset pages in Collibra Platform for private preview participants, giving you at-a-glance insights into the quality of your assets. These insights include:
    - Score and dimension roll-ups.
    - Column, data quality rule, data quality metric, and row overviews.
    - Details about the data elements of an asset.
- You can now see the Overview DQ Score on an Asset when searching via Data Marketplace. This improves your ability to browse the data quality scores of Assets without opening their Asset Pages.

Pushdown

- Pushdown processing for SQL Server is now available for preview testing.

Fixes

Capabilities

- While editing the command line of a job containing an outlier by replacing `-by HOUR` with `-tbin HOUR`, the command line no longer reverts to its original state after profiling completes. (ticket #126764)
- When exporting the job log details to CSV, Excel, PDF, or Print from the Jobs page, the exported data now contains all rows of data. (ticket #129832)
    - Additionally, when exporting the job log details to PDF from the Jobs page, the PDF file now contains the correct column headers and data. (ticket #129832)

- When working with the Alert Builder, you no longer see a “No Email Servers Configured” message despite having correctly configured SMTP settings. (ticket #127520)

DQ Integration

- When integrating data from an Athena connection, you can now use the dropdown menu in rules to map an individual column to a Rule in Collibra Platform. (ticket #125152, 126150)

Pushdown

- When archive breaking records is enabled, statements containing backticks ` or new lines are properly inserted into the source system. (ticket #130122)
- Snowflake Pushdown jobs with many outlier records either dropped or added, new limits to memory usage now prevent out-of-memory issues. (ticket #126284)

Known Limitations

- When Link ID is enabled for a Snowflake or Databricks Pushdown job, Having_Count_Greater_Than_One and Two_Decimal_Places do not function properly.
    - The workaround for Having_Count_Greater_Than_One is to manually add the Link ID to the group by clause in the rule query.
    - The workaround for Two_Decimal_Places is to add a * to the inner query.

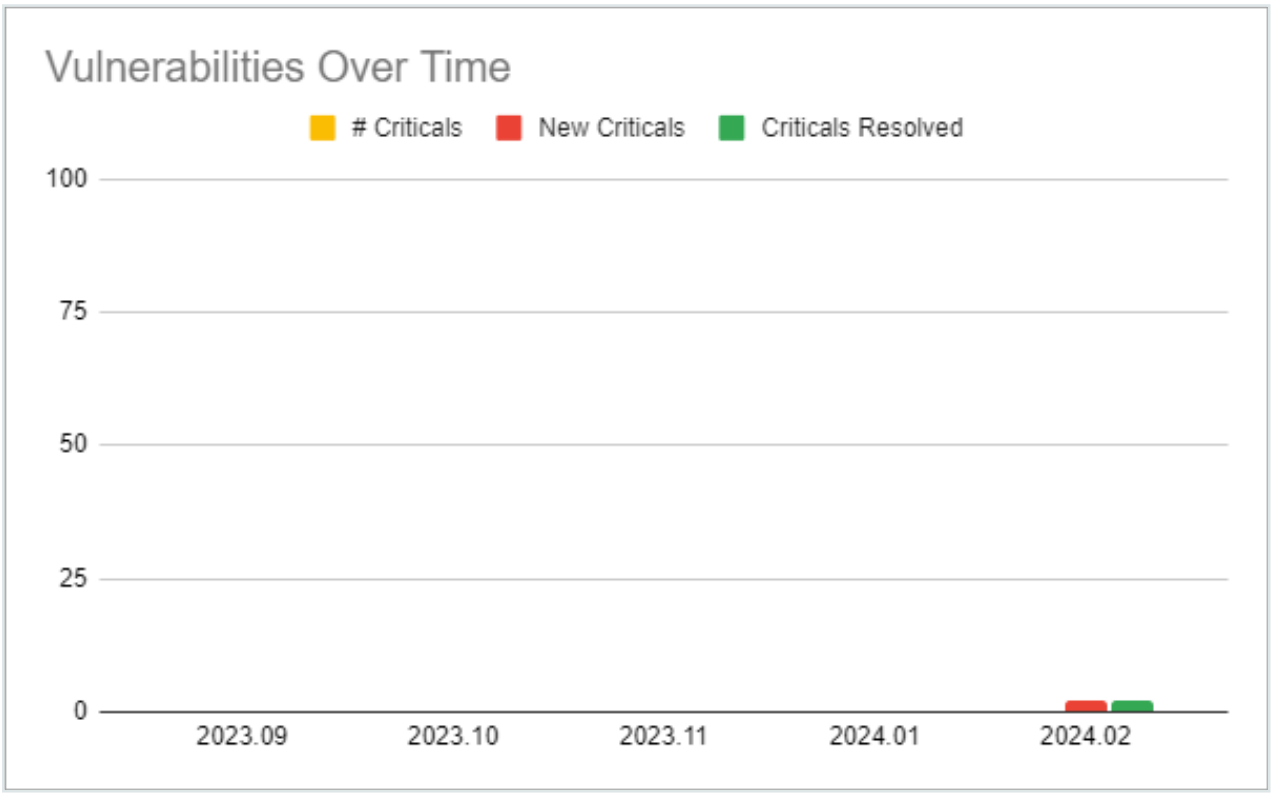
DQ Security

> **Note** If your current Spark version is 3.2.2 or older, we recommend upgrading to Spark 3.4.1 to address various critical vulnerabilities present in the Spark core library, including Log4j.

Show security metrics
The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-09-18	2023.09	0	0	0	0	1	JFrog Artifact
DQ	2023-10-30	2023.10	0	0	0	0	1	JFrog Artifact
DQ	2023-11-13	2023.11	0	0	0	0	3	JFrog Artifact
DQ	2024-01-03	2024.01	0	0	0	0	3	JFrog Artifact
DQ	2024-01-22	2024.02	0	2	2	0	6	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2024.01

- Highlights
- Migration Updates
- New Features
- Enhancements
- Fixes
- Known Limitations
- Security Metrics

Release Information

- Release date of Data Quality & Observability Classic 2024.01: January 29, 2024
- Publication dates:
  - Release notes: January 4, 2024
  - Documentation Center: January 29, 2024

Highlights

Integration

We've introduced several new features and enhancements to significantly improve the integration experience.

- Aggregation paths on tables are now set by default, simplifying the configuration within the Collibra DQ Admin Console.
- The new Quality tab is now available as part of the latest UI updates for Asset pages in Collibra Platform for private preview participants, giving you at-a-glance insights into the quality of your assets. These insights include:
  - Score and dimension roll-ups.
  - Column, data quality rule, data quality metric, and row overviews.
  - Details about the data elements of an asset.
- When multiple jobs are attached to a table, the Quality tab on an Asset page shows an average similar to a scorecard in Collibra DQ.

> **Note** **Table** Assets roll up to the **DQ Job** Data Asset. Best practice is to roll up DQ Job to Table to align the dedication score and the Quality tab Asset score for a Table Asset.

Spark Version Update

As of Collibra DQ version 2023.11, we've upgraded our out-of-the-box Apache Spark version from 3.2.0 to 3.4.1. We strongly encourage organizations on Standalone deployments of Collibra DQ to upgrade to the latest Spark package to utilize of the new

features and address some of the major vulnerabilities with Spark 3.2 or earlier versions. Additionally, Collibra DQ support for Spark 2.x is limited as of Collibra DQ 2024.01, as Spark 2.x has reached its end of life.

If you use Spark 3.2.2 or lower, we recommend upgrading to 3.4.1 to address various critical vulnerabilities present in the Spark core library, including log4J.

Important
Changes for Kubernetes Deployments
As of Collibra DQ version 2023.11, we've updated the Helm Chart name from `owldq` to `dq`. For Helm-based upgrades, point to the new Helm Chart while maintaining the same release name. Please update your Helm install command by referring to the renamed parameters in the values.yaml file. It is also important to note that the pull secret has changed from `owldq-pull-secret` to `dq-pull-secret`.

Further, following deployment, your existing remote agent name will change. For example, if your agent name is owldq-owl-agent-collibra-dq, the new agent name will be dq-agent-collibra-dq. If your organization uses APIs for development, ensure that you upgrade AGENT name configurations in your environments.

Lastly, when you deploy using the new Helm Charts, new service (Ingress/Load Balancer) names are created. This changes the IP address of the service and requires you to reconfigure your Load Balancer with the new IP.

Please see the expandable sections below for more details about specific changes.

Show renamed keys

Old Key	Renamed Key
global.version.owl	global.version.dq
global.image.owlweb	global.image.web
global.image.owlagent	global.image.agent

Show default value changes

Parameter	Old Default Value	New Default Value
global.mainChart	owldq	dq
global.image.pullSecret.name	owldq-pull-secret	dq-pull-secret
global.web.key.secretName	owldq-ssl-secret	dq-ssl-secret
global.image.web.name	owl-web	dq-web
global.image.agent.name	owl-agent	dq-agent
global.image.livy.name	owl-livy	dq-livy
global.image.spark.name	spark	dq-spark

> **Note**
> If your organization has a standalone deployment of Collibra DQ with SSL enabled for DQ Web, and both DQ Web and DQ Agent are on the same VM or server, we recommend upgrading directly to Collibra DQ 2023.11.3 patch version or 2024.01.

Migration Updates

> **Important** This section only applies if you are upgrading from a version older than Collibra DQ 2023.09 on Spark Standalone. If you have already followed these steps during a previous upgrade, you do not have to do this again.

We have migrated our code to a new repository for improved internal procedures and security. Because owl-env.sh jar files are now prepended with `dq-*` instead of `owl-*`, if you have automation procedures in place to upgrade Collibra DQ versions, you can use the RegEx replace `regex=r"owl-.*-202.*-SPARK.*\.jar|dq-.*-202.*-SPARK.*\.jar"` to update the jars.

Additionally, please note the following:

- **Standalone Upgrade Steps**
  When upgrading from a Collibra DQ version before 2023.09 to a Collibra DQ version 2023.09 or later on Spark Standalone, the upgrade steps have changed.

Show steps

1. Open a terminal session.
2. Move the old jars from the owl/bin folder with the following commands.

```
mv owl-webapp-<oldversion>-<spark301>.jar /tmp
mv owl-agent-<oldversion>-<spark301>.jar /tmp
mv owl-core-<oldversion>-<spark301>.jar /tmp
```

3. Copy the new jars into the owl/bin folder from the extracted package.

```
mv dq-webapp-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-agent-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-core-<newversion>-<spark301>.jar
/home/owldq/owl/bin
```

4. Copy the latest `owlcheck` and `owlmanage.sh` to /opt/owl/bin directory.

> **Tip** You may also need to run `chmod +x owlcheck owlmanage.sh` to add execute permission to `owlcheck` and `owlmanage.sh`.

5. Start the Collibra DQ Web application.

```
./owlmanage.sh start=owlweb
```

6. Start the Collibra DQ Agent.

```
./owlmanage.sh start=owlagent
```

7. Validate the number of active services.

```
ps -ef | grep owl
```

New Features

Integration

- You can now see the Overview DQ Score on an Asset when searching via Data Marketplace. This improves your ability to browse the data quality scores of Assets without opening their Asset Pages.

Enhancements

Capabilities

- When setting up a connection to a Google BigQuery data source, you can now use Workload Identity to authenticate your connection. By using Workload Identity to authenticate your BigQuery connection, you can now access data stored in BigQuery across GCP projects without relying on JSON credential files or metadata obfuscation.
- When using the Alert Builder, you can now create an alert for when a job run fails. You can configure this by selecting the Job Failure option from the Status Alert modal on the Alert Builder page.
    - Additionally, the alert types on the Alert Builder page have changed from Dataset Run Alerts and Job Status to **Condition** and **Status**, respectively.
- When you receive an email alert, the body of the alert now includes the Alert Type as either **Condition** or **Status**. When the Alert Type is **Condition**, the query upon which the condition is based also displays.
- When using the /v3/rules/{dataset}/ruleBreaksSelect and you select INTERNAL for the storageType parameter, the API generates a SQL query that returns the break records included in the PostgreSQL Metastore.
- The /v2/getbreakingrecords API is now filtered by runId and limited to 100 records by default.
- When using multi-tenancy with SAML enabled, you can now set showsamltenantmetadatalabel=false from the ConfigMap to hide tenant metadata labels from SAML-enabled tenants to match the names of non-SAML-authenticated tenants.
- When signing into Collibra DQ, you are now required to select a tenant from the dropdown menu before proceeding.
- The Scheduler page is now powered by the v2/getallscheduledjobs API.

- What was previously the DQ Job button next to the search bar at the top of the Collibra DQ application is now called **Findings**.

Platform

- Standalone installations now come packaged with the PostgreSQL 12 installer instead of PostgreSQL 11.
- Helm Charts are now versioned according to their corresponding release version. You can find version details in the Charts.yaml file.
- You can now use the default truststore by setting the `global.web.tls.trust.default` flag to `true`.
- With SAML configured, you can now use the default keystore without the need for a custom keystore.
- We've introduced pattern validation for some commonly used IDs and names in Collibra DQ. You can override these patterns in the owl-env.sh file for Standalone or the Web ConfigMap for Cloud Native by replacing the default values using a RegEx for the following env variables:

Env variable	Variable name	Default	Usage
VALIDATION_PATTERN_NAME_ID	Name ID pattern	Alphanumeric characters (letters and digits) and underscores (_).	For overriding defaults of Name IDs in Collibra DQ.
VALIDATION_PATTERN_COMMON_NAME	Common name	Alphanumeric characters (letters and digits) and underscores (_).	For overriding defaults of common names used in Collibra DQ.
VALIDATION_PATTERN_SCHEMA_NAME	Schema name	Alphanumeric characters (letters and digits) and underscores (_).	For overriding defaults of connection schema names in Collibra DQ.
VALIDATION_PATTERN_CONN_NAME	Connection name	Alphanumeric characters (letters and digits), underscores (_), and hyphens (-).	For overriding defaults of connection names in Collibra DQ.

Env variable	Variable name	Default	Usage
VALIDATION_PATTERN_DATASET_NAME	Dataset name	Alphanumeric characters (letters and digits), underscores (_), and hyphens (-).	For overriding defaults of dataset names in Collibra DQ.
VALIDATION_PATTERN_FILE_NAME	File name	Alphanumeric characters (letters and digits), underscores (_), hyphens (-), periods (.), backslashes (/), and spaces ( ).	For overriding defaults of file name usages in Collibra DQ.
VALIDATION_PATTERN_LDAP_DN	DN	Comma separated key value pair with both the key and value consisting of lowercase letters, digits, and hyphens.	For overriding defaults of LDAP DN usages in Collibra DQ.

> **Example** If you want to override the default that does not allow spaces in the connection name, set the connection name variable to the following RegEx: `VALIDATION_PATTERN_CONN_NAME=^[a-zA-Z0-9_- ]+$`

- In order to promote compatibility with various software delivery platforms, the leading '0' in the defaultMode value of 0493 has been removed from 4 YAML files for Kubernetes deployments. The new defaultMode is now 493 for the following YAML files:
  - k8s/charts/dq/charts/dq-agent/templates/dq-agent-statefulset.yaml
  - k8s/charts/dq/charts/dq-livy/templates/dq-livy-deployment.yaml
  - k8s/charts/dq/charts/dq-web/templates/dq-web-statefulset.yaml
  - k8s/charts/dq/charts/spark-history-server/templates/deployment.yaml
- Configurable PBE encryption is now supported for Kubernetes and Standalone deployments of Collibra DQ.
- When validating a finding and assigning it to a ServiceNow user or group, you can now reassign this finding from ServiceNow to a Collibra DQ user. (ticket #122476, 133165)

Pushdown

- When changes are made to the schema based on the source query, the dataset_schema table now contains any changes made.

Fixes

Capabilities

- When creating a rule on a Pullup dataset that uses “in” or “rlike” where the condition ends on wrapped parentheses (ie.))), the string replace logic now works as expected. Previously, rules that contained rlike() or in() would throw exceptions when the job ran. (ticket #128759)

  > **Important** In order to keep rules as close to the original input as possible, some of the padding that was originally appended to certain characters has been removed.

- When archiving rule breaks, break records now export to S3 buckets as expected. (ticket #125411)
- When changing the business unit on the Metadata Bar or Dataset Manager, the updated business unit now replaces the old one in the Metastore instead of creating an additional entry. (ticket #127823, 130086)
- When editing a dataset from Dataset Manager in the new UI, the Schema/Parent Folder and Table/File Name fields are no longer swapped. (ticket #129103)
- When running a job against a BigQuery dataset where the underlying query uses “from” before the schema/table, Collibra DQ now parses the “from” correctly. (ticket #129798)
- When attempting to create a Pullup job against a Trino schema whose name contains the escape character `\`, you can now view the table details in Explorer. (ticket #131065)
- The configs `dupelimit` and `dupelimitui` now work as expected where `dupelimit` restricts the number of dupe findings stored in the Metastore and of those findings, `dupelimitui` restricts the number of dupes shown on the Dupes tab on the Findings page. (ticket #127748)
- When using the Rule Builder in the new UI, freeform rules using RLIKE now successfully pass validation. (ticket #127779)
- When using the Alert Builder in the new UI, you can now update alert batch names as expected. (ticket #129079)
- When a scheduled job runs with zero rows and Parallel JDBC enabled, it no longer fails. (ticket #128385)

- When using Kerberos-based authentication for PostgreSQL or CockroachDB data sources, jobs no longer time out before authenticating against KDC. (ticket #128540)

Platform

- While configuring SSL after a fresh Standalone install of or upgrade to Collibra DQ version 2023.11, the DQ agent now starts as expected. (ticket #131078)

  > **Note** With this update, any system that has `export SERVER_PORT` set will impact the default value of -1. You need to comment out `export SERVER_PORT=9000` as `PORT=9000 #owl-web port NUMBER` to define the port.

- We fixed an issue that caused unexpected license name requests. (ticket #126731)

DQ Integration

- When mapping columns from rules in Collibra DQ, column names are now parsed correctly in Collibra Platform.
- When a User Defined Rule is deactivated or deleted from a dataset in Collibra DQ, then the Rule and Score status in Collibra Platformwill be set to “Suppressed” when the dataset runs again in Collibra DQ. (ticket #124414)

Pushdown

- When running a Redshift Pushdown job to create a profile of a Redshift dataset, you can now view TopN Shapes for columns containing NULL values. (ticket # 129221)
- When casting from one data type to another on a Redshift dataset, the job now runs successfully without throwing an exception message. (ticket #128718)
- When creating a Pushdown job on a BigQuery dataset that contains a DATE column time slice, you can now create the job without receiving a command line error. (ticket #129283)
- When using the SQL Query option on a Snowflake table with a mixed case name, you can now create the job without receiving an error. (ticket #126118)

DQ Cloud

- When running a job on a Snowflake table containing a timestamp column, timestamps now translate correctly as date, time, and timestamp types. (ticket #120975)

  > **Note** While this fix was originally intended to be limited to DQ Cloud, timestamps also translate correctly on Standalone deployments.

- When using the Dupes tab on the Findings page, dupe observations now display correctly when more dupes are discovered than the preview limit of 30. (ticket #127625)

Known Limitations

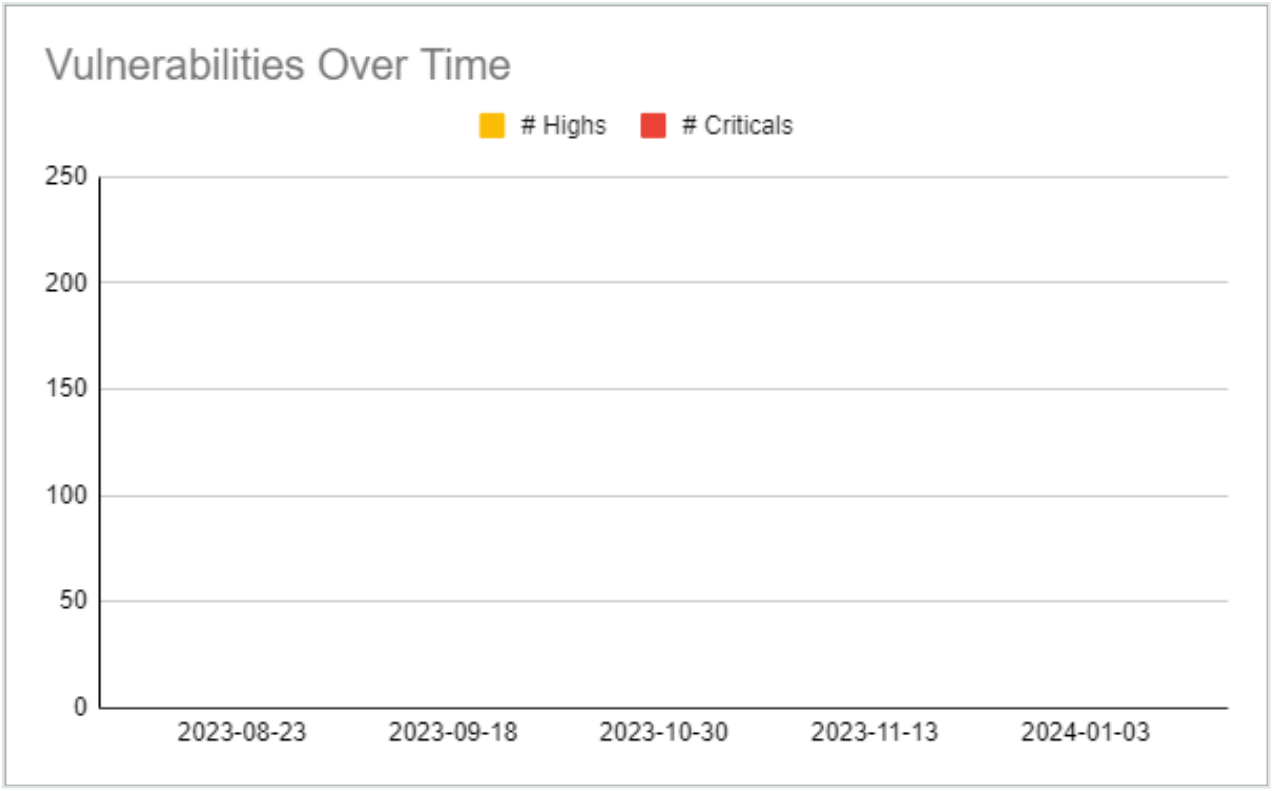
- When configuring `dupelimitui`, values above 2000 result in some UI lag. Because of this, we recommend using `dupelimitui` values under 2000.

DQ Security

Show security metrics
The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-08-23	2023.08	0	0	0	0	1	JFrog Artifact
DQ	2023-09-18	2023.09	0	0	0	0	1	JFrog Artifact
DQ	2023-10-30	2023.10	0	0	0	0	1	JFrog Artifact
DQ	2023-11-13	2023.11	0	0	0	0	3	JFrog Artifact
DQ	2024-01-03	2024.01	0	0	0	0	3	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Release 2023.11

- Highlights
- Migration Updates
- Liveness Probe Updates
- New Features
- Enhancements
- Fixes
- DQ Security
- Updated UI
- Updated UI Limitations
- Maintenance Updates

Release Information

- Release date of Data Quality & Observability Classic 2023.11: November 20, 2023
- Publication dates:
  - Release notes: November 8, 2023
  - Documentation Center: November 13, 2023

Highlights

Pushdown

We're excited to announce that Pushdown for BigQuery is now generally available! Pushdown is an alternative compute method for running DQ jobs, where Collibra DQ submits all of the job's processing directly to a SQL data warehouse, such as BigQuery. When all of your data resides in BigQuery, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirements of a DQ job.

UI Redesign

New installs of Collibra DQ come with REACT_MUI and UX_REACT_ON admin flags set to TRUE by default. Additionally, if a pre-existing install of Collibra DQ had these flags set to FALSE, they are now set to TRUE. While you can still modify these flags from the Admin Console ▸ Configuration Settings ▸ Application Config page, we recommend keeping them set to TRUE to allow for full feature functionality and an elevated user experience. See the Updated UI table below for more details on what's changed.

Spark Version Update

We've upgraded our out-of-the-box Apache Spark version from 3.2.0 to 3.4.1. We strongly encourage organizations on Standalone deployments of Collibra DQ to upgrade to the latest Spark package to utilize of the new features and address some of the major vulnerabilities with Spark 3.2 or earlier versions. Additionally, Collibra DQ support for Spark 2.x will be limited as of Collibra DQ 2024.01, as Spark 2.x has reached its end of life.

Collibra AI

We're delighted to announce that Collibra AI is now available for private preview testing! Collibra AI introduces automated SQL rule writing capabilities on the Rule Workbench and Dataset Overview that help you accelerate the discovery, curation, and visualization of your data. Contact your Collibra CSM for more details about participating in this exciting private preview.

Important

Changes for Kubernetes Deployments

We've updated the Helm Chart name from `owldq` to `dq`. For Helm-based upgrades, point to the new Helm chart while maintaining the same release name. Please update your Helm install command by referring to the renamed parameters in the values.yaml file. It is also important to note that the pull secret has changed from `owldq-pull-secret` to `dq-pull-secret`.

Further, following deployment, your existing remote agent name will change. For example, if your agent name is owldq-owl-agent-collibra-dq, the new agent name will be dq-agent-collibra-dq. If your organization uses APIs for development, ensure that you upgrade AGENT name configurations in your environments.

Lastly, when you deploy using the new Helm Charts, new service (Ingress/Load Balancer) names are created. This changes the IP address of the service and requires you to reconfigure your Load Balancer with the new IP.

Please see the expandable sections below for more details about specific changes.

Show renamed keys

Old Key	Renamed Key
global.version.owl	global.version.dq
global.image.owlweb	global.image.web
global.image.owlagent	global.image.agent

Show default value changes

Parameter	Old Default Value	New Default Value
global.mainChart	owldq	dq
global.image.pullSecret.name	owldq-pull-secret	dq-pull-secret
global.web.key.secretName	owldq-ssl-secret	dq-ssl-secret
global.image.web.name	owl-web	dq-web
global.image.agent.name	owl-agent	dq-agent
global.image.livy.name	owl-livy	dq-livy
global.image.spark.name	spark	dq-spark

> **Note**
> If your organization has a standalone deployment of Collibra DQ with SSL enabled for DQ Web, and both DQ Web and DQ Agent are on the same VM or server, we recommend upgrading directly to Collibra DQ 2023.11.3 patch version instead of 2023.11. For more information, see the Maintenance Updates section below.

Migration Updates

> **Important** This section only applies if you are upgrading from a version older than Collibra DQ 2023.09 on Spark Standalone. If you have already followed these steps during a previous upgrade, you do not have to do this again.

We have migrated our code to a new repository for improved internal procedures and security. Because owl-env.sh jar files are now prepended with `dq-*` instead of `owl-*`, if you have automation procedures in place to upgrade Collibra DQ versions, you can use the RegEx replace `regex=r"owl-.*-202.*-SPARK.*\.jar|dq-.*-202.*-SPARK.*\.jar"` to update the jars.

Additionally, please note the following:

- **Standalone Upgrade Steps**
  When upgrading from a Collibra DQ version before 2023.09 to a Collibra DQ version

2023.09 or later on Spark Standalone, the upgrade steps have changed.

Show steps

1. Open a terminal session.
2. Move the old jars from the owl/bin folder with the following commands.

```
mv owl-webapp-<oldversion>-<spark301>.jar /tmp
mv owl-agent-<oldversion>-<spark301>.jar /tmp
mv owl-core-<oldversion>-<spark301>.jar /tmp
```

3. Copy the new jars into the owl/bin folder from the extracted package.

```
mv dq-webapp-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-agent-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-core-<newversion>-<spark301>.jar
/home/owldq/owl/bin
```

4. Copy the latest `owlcheck` and `owlmanage.sh` to /opt/owl/bin directory.

> **Tip** You may also need to run `chmod +x owlcheck owlmanage.sh` to add execute permission to `owlcheck` and `owlmanage.sh`.

5. Start the Collibra DQ Web application.

```
./owlmanage.sh start=owlweb
```

6. Start the Collibra DQ Agent.

```
./owlmanage.sh start=owlagent
```

7. Validate the number of active services.

```
ps -ef | grep owl
```

Liveness Probe Updates

- **Cloud Native**
  - When a Kubernetes pod service becomes unstable, a new liveness probe automatically deletes the pod to ensure the DQ agent stays alive and running. No further action is necessary for Cloud Native deployments; this note is strictly for informational purposes only.
- **Standalone**
  - Because the implementation of the liveness probe for Kubernetes required a change in the `owlmanage.sh` file for Standalone installations, you need to follow the steps below to upgrade a Standalone deployment.

    > **Important**
    > If your organization has a Standalone installation of Collibra DQ, you must copy the latest `owlmanage.sh` to /opt/owl/bin directory, as the file has changed.

    Show steps

    1. Open a terminal session.
    2. Move the old dq-agent jar from the owl/bin folder with the following command.

       ```
       mv dq-agent-<oldversion>-<spark301>.jar /tmp
       ```

    3. Copy the new dq-agent jar into the owl/bin folder from the extracted package.

       ```
       mv dq-agent-<newversion>-<spark301>.jar
       /home/owldq/owl/bin
       ```

    4. Copy the latest `owlmanage.sh` to /opt/owl/bin directory.

       > **Tip** You may also need to run `chmod +x owlmanage.sh` to add execute permission to `owlmanage.sh`.

5. Start the Collibra DQ Agent.

```
./owlmanage.sh start=owlagent
```

6. Validate the number of active services.

```
ps -ef | grep owl
```

New Features

Pushdown

- When running Profiling on Pushdown jobs, advanced level profiling is now an opt-in feature and does not run by default. Advanced Profile determines whether a string field contains various string numerics, calculates TopN, BottomN, and TopN Shapes, and detects the scale and precision of double fields. We've also included the Profile String Length setting in the Advanced Profile option on the Explorer ▸ Settings modal.

Enhancements

Capabilities

- When using the Alert Builder, you can now create an alert for when a job run completes successfully. When you add an alert, you can now choose from two options:
  - **Dataset Run Alerts** let you set an alert for when a job run meets a certain condition.
  - **Job Status** lets you set an alert for when a job run completes.
- You can now configure arbitrary users as part of root user groups for cloud native deployments of Collibra DQ on Openshift.
- You can now use OAuth 2.0 to authenticate Trino connections.
- We've moved the /v2/getprofiledeltasbyrunid API to the V3 Profile API GET /v3/profile/deltas.

Pushdown

- You can now archive source break records from Redshift Pushdown jobs.

Integration

- The Unassigned DQ Job domain now has the Parent Community name appended to it. Given the unique community name constraint when using more than one tenant or instance, this change allows for the unique naming of Business Units. The new Community structure is as follows:
    - Parent Community
        - Business Unit Community
          OR
        - **Unassigned DQ Job + Parent Community**
            - DQ Job Domain
                - DQ Job Asset
            - Rulebook (definitions)
                - DQ Rule Assets
            - Rulebook (scores)
                - DQ Metric Assets
- When on a dataset-level page, such as Findings or Profile, you can now select "Reset Integration" from the Integration dropdown menu on the metadata bar to realign mappings that may have changed in Collibra Platform. If you get an error message when your mappings are aligned, this action can reset the integration and allow you to proceed with Collibra DQ metadata ingestion.
- When a dataset integration with Collibra Platform is enabled, you can now view the Community, Sub-Community, and Domain hierarchy to which the integration is mapped in Collibra Platform from the metadata bar on all dataset-level pages. This increases the transparency of where Assets are created in Collibra Platform without needing to navigate away from Collibra DQ.
    - Additionally, you can click any of the breadcrumbs to open the Community, Sub-Community, or Domain in Collibra Platform.
- When running Pushdown jobs with integrations enabled, the results now integrate into Collibra Platform successfully. Previously, you had to manually enable or disable the integration each time a Pushdown job ran.
- After integrating a dataset from the Admin Console, Findings or Dataset Manager pages, you can now click the "Data Quality Job" link in the new View in DIC column to open its corresponding Asset page in Collibra Platform.

- The GET /dgcjson endpoint is now included in the main Integrations API as GET /dgc/integrations/getdgcjson. Previously, this was located under the "UI Internal" section of Swagger.

> **Tip** When an integration error occurs, check that the Community, Domain, and Asset names in Collibra Platform don't already exist from a previous integration.

DQ Cloud

- Collibra Edge now reflects Collibra DQ's new default Spark version 3.4.1.

Fixes

Capabilities

- When running a job to check for outliers where the lookback value is set to something other than the default 5, the minhistory now updates to the correct value in the metastore. (ticket #124063)

> **Note** Manual overrides of -dlminhist on the command line do not save in the metastore.

- When using the new Collibra DQ UI, run date information now displays correctly. Previously, the job configuration of manual DQ job runs would override with hardcoded dates, which caused the upcoming scheduling date to only reflect the hardcoded date. (ticket #126345)
- When changing the assignment for a dataset with the new UI turned on for an SSO instance of Collibra DQ, existing SAML assignments now load correctly on the Findings and Assignments pages. (ticket 128063)
- When assigning users to the following roles, they can now access all appropriate Admin screens (ticket #128115):
  - ROLE_CONNECTION_MANAGER
  - ROLE_DATA_GOVERNANCE_MANAGER
  - ROLE_DATASET_MANAGER
  - ROLE_OWL_ROLE_MANAGER
  - ROLE_USER_MANAGER

Platform

- When editing the Batch Name field of a Dataset Run Alert, the batch name distribution list no longer updates if the Batch Name field is empty or blank `" "`. (ticket #126763, 126796)
- When reviewing the Outlier tab on the Findings page, outlier findings now expand correctly when you drill down into them. (ticket #126065)
- When creating an alert, you can now enter the special characters `! # $ % & ' * + - / = ? ^ _ ` . { | } ~` in an email address for the alert recipient. (ticket #126763)

DQ Integration

- When viewing user-defined or adaptive rules, Passing Fraction now reflects the points deducted from an individual rule, rather than the total rule score. Previously, the total breaks for the rule type of user-defined rules were used to generate the Passing Fraction, rather than the individual rule breaks. (tickets #124217, 127700)
- When using the configuration wizard to map your single-tenant Collibra DQ environment, you can now link your integration to an existing community and create new communities as part of your integration. (tickets #122948, 123426, 126227, 127044, 128345)
- When integrating a Collibra DQ dataset and setting up dimensions in Collibra Platform, the columns now display correctly in Collibra Platform after running the dataset in Collibra DQ. (ticket #126379)

Pushdown

- When adding outliers to a Pushdown dataset and running a job, the outlier configurations now render properly. Previously, when editing a job, one or more of the outliers did not display as expected. (ticket #124736)

DQ Cloud

- Fixed an issue that caused the Collibra DGC MDL Proxy to run out of memory under certain conditions.

DQ Security

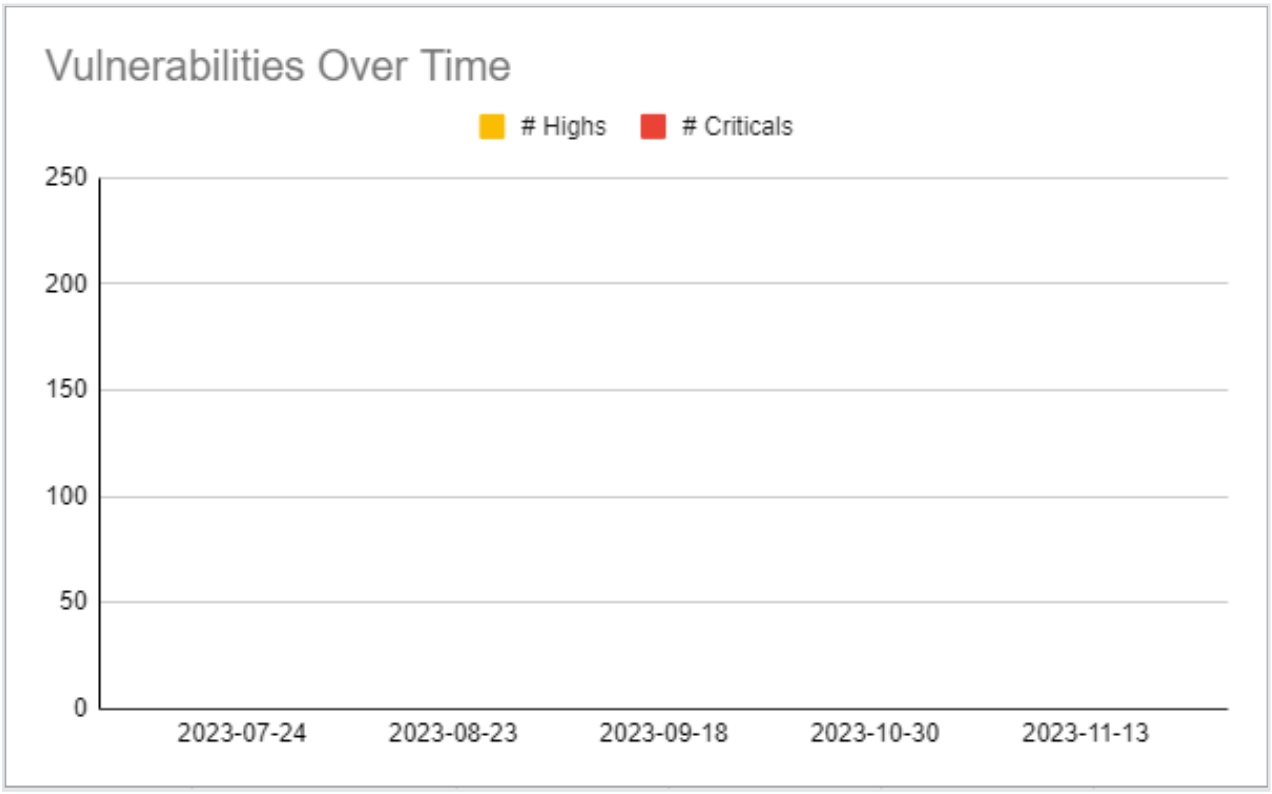
> **Note**
> We've removed all existing classic UI JS libraries and their references from the Updated UI to address and prevent any potential security vulnerabilities.

Show security metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-07-24	2023.07	0	0	0	0	4	JFrog Artifact
DQ	2023-08-23	2023.08	0	0	0	0	1	JFrog Artifact
DQ	2023-09-18	2023.09	0	0	0	0	1	JFrog Artifact
DQ	2023-10-30	2023.10	0	0	0	0	1	JFrog Artifact
DQ	2023-11-13	2023.11	0	0	0	0	3	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Updated UI

In addition to broader user interface and user experience enhancements, we've also added some impressive new features! The following table showcases some of the highlights.

Component	Description	Available in Classic
Metadata Bar	The Metadata Bar is a dataset anchor that simplifies the navigation to some of your most frequently used pages, such as Dataset Overview, Profile, Rules, and Findings. It also provides quick insight into your dataset, such as the number of active rules, the data source from which the dataset was created, and whether or not your job is scheduled to run automatically. When an integration is set up, the metabar also allows you to easily enable or disable dataset metadata integrations into Collibra Platform. You can access the Metadata Bar on any dataset-level page, including: • Findings • Profile • Dataset Rules • Alert Builder • Dataset Overview	No
Dataset Overview	Dataset Overview lets you query your dataset to discover key data points and insights, which you can convert to data quality rules entirely within the Dataset Overview modal. With the power to write SQL to query your dataset, you can accelerate the process of data discovery and reveal important insights in real time. Dataset Overview also allows private preview participants to leverage Collibra AI to automatically write and troubleshoot SQL for faster rule writing and advanced exploration of your dataset. See the Collibra AI private preview documentation to learn more.	No
Explorer	The new workflow simplifies the process of creating a DQ job to run against a dataset. With just a few clicks, you can create a basic profile job in a matter of seconds instead of minutes. For a more advanced scan, the step-by-step guide walks you through the process, eliminating many of the more tedious elements of the classic Explorer.	Yes

Component	Description	Available in Classic
Findings	The new Findings page will feel similar to the classic page, but with a few important changes: • The daily score, pulse view, row count, and pass/fail charts are now broken out into individual tabs for an improved display. • The dataset metadata that previously resided next to the chart views is now anchored to the top of the page within the Metadata Bar for quicker analysis of your dataset's key data points. • The findings tabs of the various data quality dimensions now have enhanced readability and more clearly displayed actions.	Yes
Profile	While many of the same column- and dataset-level insights are unchanged from the classic UI, the presentation of information is now modernized for a crisper experience.	Yes
Rules	Dataset Rules lists all previously saved rules for a given dataset and provides an overview of their details, such as the definitions of SQL conditions, rule types, and whether or not rules pass validation checks. From here, you can access the Rule Workbench to create or edit a rule. The Rule Workbench replaces the classic Rule Builder, fusing an elegant SQL command line interface with the preview and advanced setting capabilities you expect from a modern SQL builder. Like the Dataset Overview, you can also use Collibra AI generated SQL to write and troubleshoot rules on the Rule Workbench. See the Collibra AI private preview documentation to learn more. We've also split Data Class and Template rules into their own pages to emphasize that they are independent of jobs and datasets and improve their overall organization.	No
Alerts	The new Alert Builder gives you an at-a-glance overview of all alerts for a particular dataset and simplifies adding new alerts and editing existing ones.	Yes
Dataset Manager	Dataset Manager provides a list of all datasets in your Collibra DQ environment, as well as a variety of management options, such as bulk actions, assigning datasets to data categories, and the ability to filter datasets by a variety of criteria.	Yes

Component	Description	Available in Classic
Column Manager	Column Manager is a detailed breakdown of all the columns in the datasets in your Collibra DQ environment that shows key data points like data type, various ratios, and the Pass/Fail status of a given column. You can also bulk apply rules, data classes, and sensitive labels to selected columns.	No
Report Dashboards	The Reports section now has two new dashboards available in the updated UI: • Column Dimension is an overview to view the data quality dimensions of the columns of all or specific datasets. By filtering by business unit, dataset, column, and monthly periods, you can make more informed decisions and increase the value of your data. Column Dimension also provides insight into the total current dimension scores, dimension scores across time, and DQ scores (along with other metadata) for each column. • Dataset Dimension is an overview to view data quality dimensions with different filters to make more informed decisions and increase the value of your data. You can filter by specific datasets or all datasets in your Collibra DQ environment, by business unit, and by monthly periods.	No
Connections	The updated Connections page in the Admin Console does away with the connection tiles of the classic page in favor of a highly searchable and sortable paginated table format. The new page also features two tabs for Connections and the Drivers stored in your Collibra DQ environment. Additionally, when you add or edit a connection, the connection template is now organized in three tabs for Connection Details, Driver Properties, and Connection Variables. With these sections now clearly delineated, the process of creating or updating a connection is now much cleaner.	Yes
Admin Console	The Admin Console now lists each admin activity for better organization and simpler navigation than the tiles of the classic UI.	Yes

Maintenance Updates

Explorer

- When using the SQL compiler on the dataset overview for remote files, the Compile button is disabled because the execution of data files at the Spark layer is unsupported.
- You cannot currently upload temp files from the new File Explorer page. This may be addressed in a future release.
- The Formatted view tab on the File Explorer page only supports CSV files.
- When creating a job, the Estimate Job step from the classic Explorer is no longer a required step. However, if incorrect parameters are set, the job may fail when you run it. If this is the case, return to the Sizing step and click Estimate next to Job Size before you Run the job.

Connections

- When adding a driver, if you enter the name of a folder that does not exist, a permission issue prevents the creation of a new folder.
    - A workaround is to use an existing folder.

Admin

- When adding another external assignment queue from the Assignment Queue page, if an external assignment is already configured, the Test Connection and Submit buttons are disabled for the new connection. Only one external assignment queue can be configured at the same time.
- Due to security requirements, we've removed the ability for application administrators to add new local users from the User Management page in the Admin Console. All new users must use the Register link on the Collibra DQ sign in screen.
    - When auto-approve is not configured, admin users can still manually approve new user requests and add roles to the new user from the User Management page.

Profile

- When adding a distribution rule from the Profile page of a dataset, the Combined and Individual options incorrectly have "OR" and "AND" after them.
- When using the Profile page, Min Length and Max Length does not display the correct string length. This will be addressed in an upcoming release.

Rules

- When creating a quick rule from the Data Preview tab of the Findings, Profile, or Rules pages, the Preview Limit and Run Time Limit do not honor the application default limits of 6 and 30, respectively. Instead, the Preview Limit and Run Time Limit are both incorrectly set to 0.
    - While this will be addressed in the January (2024.01) release, a workaround is to manually edit these fields from the Rule Workbench ▸ Settings modal.

Alerts

- Batch email updates are not currently working in the preview UI. This will be addressed in the January (2024.01) release.
- When editing the Batch Name of a job alert, there is a limitation that prevents you from editing the email address field associated with the batch alert.

Scorecards

- When creating a new scorecard from the Page dropdown menu, because of a missing function, you cannot currently create a scorecard.
    - While a fix for this is planned for the September (2023.09) release, a workaround is to select the Create Scorecard workflow from the three dots menu instead.

Navigation

- The Dataset Overview function on the Metadata Bar is not available for remote files.
- The Dataset Overview modal throws errors for the following connection types:
  - BigQuery (Pushdown and Pullup)
  - Athena CDATA
  - Oracle
  - SAP HANA
- The Dataset Overview function throws errors when you run SQL queries on datasets from S3 and BigQuery connections.

Maintenance Updates

2023.11.3

- While configuring SSL after a fresh Standalone install or upgrade to Collibra DQ version 2023.11, the DQ Agent and Web now start as expected. (ticket #131078)
  - With this fix, the DQ Agent now uses port 9101 by default to expose the Health Check API.

> **Note** Ensure you select the latest corresponding Helm Chart when taking a maintenance update for Cloud Native deployments.

2023.11.4

- When synchronizing DQ rules without business units configured in Collibra DQ, you can now synchronize them to both root and sub-communities. (ticket #127044, 128138)

Release 2023.10

- Highlights
- Migration Updates
- Enhancements
- Fixes

- Known Limitations
- DQ Security
- Preview UI
- Preview UI Limitations

Release Information

- Expected release date of Data Quality & Observability Classic 2023.10: October 29, 2023
- Publication dates:
  - Release notes: October 22, 2023
  - Documentation Center: October 27, 2023

Highlights

Pushdown
We're excited to announce that Pushdown for Trino is now available as a public preview offering! Pushdown is an alternative compute method for running DQ jobs, where Collibra DQ submits all of the job's processing directly to a SQL data warehouse, such as Trino. When all of your data resides in Trino, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirements of a DQ job.

Migration Updates

> **Important** This section only applies if you are upgrading from a version older than Collibra DQ 2023.09 on Spark Standalone. If you have already followed these steps during a previous upgrade, you do not have to do this again.

We have migrated our code to a new repository for improved internal procedures and security. Because owl-env.sh jar files are now prepended with `dq-*` instead of `owl-*`, if you have automation procedures in place to upgrade Collibra DQ versions, you can use the RegEx

replace `regex=r"owl-.*-202.*-SPARK.*\.jar|dq-.*-202.*-SPARK.*\.jar"` to update the jars.

Additionally, please note the following:

- **Standalone Upgrade Steps**
  When upgrading from a Collibra DQ version before 2023.09 to a Collibra DQ version 2023.09 or later on Spark Standalone, the upgrade steps have changed.
  Show steps
  1. Open a terminal session.
  2. Move the old jars from the owl/bin folder with the following commands.

     ```
     mv owl-webapp-<oldversion>-<spark301>.jar /tmp
     mv owl-agent-<oldversion>-<spark301>.jar /tmp
     mv owl-core-<oldversion>-<spark301>.jar /tmp
     ```

  3. Copy the new jars into the owl/bin folder from the extracted package.

     ```
     mv dq-webapp-<newversion>-<spark301>.jar
     /home/owldq/owl/bin
     mv dq-agent-<newversion>-<spark301>.jar
     /home/owldq/owl/bin
     mv dq-core-<newversion>-<spark301>.jar
     /home/owldq/owl/bin
     ```

  4. Copy the latest `owlcheck` and `owlmanage.sh` to /opt/owl/bin directory.

     > **Tip** You may also need to run `chmod +x owlcheck owlmanage.sh` to add execute permission to `owlcheck` and `owlmanage.sh`.

  5. Start the Collibra DQ Web application.

     ```
     ./owlmanage.sh start=owlweb
     ```

  6. Start the Collibra DQ Agent.

     ```
     ./owlmanage.sh start=owlagent
     ```

  7. Validate the number of active services.

```
ps -ef | grep owl
```

Enhancements

Capabilities

- When using the Mapping (Validate Source in classic mode) activity, we've introduced the following options:
  - The Skip Lines (`srcskiplines` from the command line) option instructs Collibra DQ to skip the number of lines you specify in CSV source datasets.
  - The Multi Lines (`srcmultilines` from the command line) option instructs Collibra DQ to read JSON source files formatted in multi-line mode.
- When using the Dataset Manager, you can now filter by Pushdown and/or Pullup connection type.
- When using Pulse View, the Lookback column is now called Last X Days.

Pushdown

- Profiling on Pushdown jobs now uses a tiered approach to determine if a string field contains various string numerics, calculate TopN, BottomN, and TopN Shapes, and detect the scale and precision of double fields.
- When running a job with profiling and other layers enabled, the entire allocated connection pool is now used from the beginning to the end of the job to extract the maximum allowed parallelism. Previously, profiling ran first and had to finish before the activities of any other layers began.
- You can now use the Alert Builder to set up alerts for Athena Pushdown jobs.

DQ Cloud

- DQ Cloud now supports the Collibra DQ to Collibra Platform API-based integration with the same functionality as on-premises deployments of Collibra DQ.

Fixes

Capabilities

- Native SQL rules on connections using Password Manager authentication types now run successfully on Cloud Native deployments of Collibra DQ. (ticket #111493)
- When running a job with a join of 2 datasets, the job no longer incorrectly shows a "Finished" status when the secondary dataset is still in the Spark loading process. (ticket #116004)
- When using Calibrate to modify the start and end dates for jobs with outliers, the dates now save properly. Previously, the calibration dates did not save in the calibration modal. (ticket #120283)
- When creating a job on an Oracle connection, Collibra DQ now blocks SDO_GEOMETRY and other geospatial data types from processing, allowing you to create the job. Previously, these data types prevented the creation of jobs. (ticket #122200)

Platform

- When using a SAML sign-in in a multi-tenant environment, user authentication is now successful when RelayState is unavailable. (ticket #123238)
- When signing into Collibra DQ using the SAML SSO option, you now see all configured tenants. Previously, only one tenant displayed on the SAML SSO dropdown menu. (ticket #124323)
- When specifying database properties containing spaces (' ') in the connection string field of a JDBC connection, the connection URL now properly transcribes string spaces. (ticket #124397)
- When you select the Completeness Report from the Reports page, it now opens as intended. (ticket #123602)

DQ Cloud

- When deleting datasets from Catalog, there is no longer a discrepancy between the number of datasets displayed on the Collibra DQ UI and in PostgreSQL metastore. (ticket #125069)

- Eliminated a spurious error message logged during some normal operations when in Standalone mode.
- You can now use the GET tenant/v3/edges/agents API to retrieve information about your Edge site agents and their statuses.
- OwlCheckQ now synchronizes without error.

Pushdown

- When creating a rule against a Snowflake Pushdown job that references the same dataset twice, the syntax now passes validation. Previously, a Snowflake Pullup rule that referenced the same dataset twice and used the same syntax passed validation, but the Snowflake Pushdown rule did not. (tickets #122364, 122923, 123748)
- When archiving break records to Snowflake, the break records are now properly archived and you no longer receive an error. (tickets #123760, 123987)

Known Limitations

Capabilities

- The Assignments page does not currently filter by date range. However, this functionality is planned for an upcoming release.

Platform

- When using parquet or txt files on the Mapping activity, you must select "parquet" or ".txt," respectively, as the extension. If you use the default "auto" extension option, these file types cannot be used.

DQ Security

> **Important**
> We've modified the OS image by removing some of the OS utils, such as curl, to address major vulnerabilities. If you use any of these OS utils in your custom scripts within

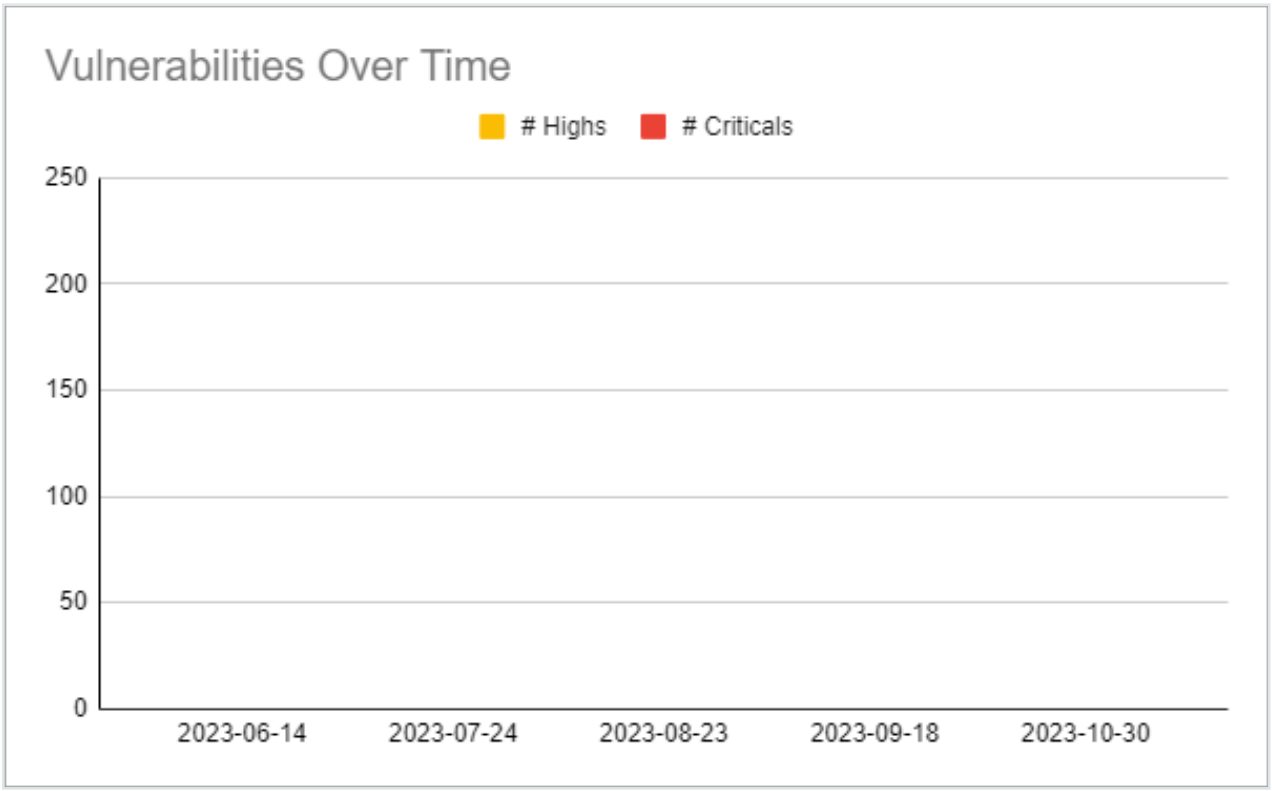
containers, you need to modify them to use different mechanisms, such as /dev/tcp socket, for the same functions.

Show security metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-06-14	2023.06	0	0	0	0	0	JFrog Artifact
DQ	2023-07-24	2023.07	0	0	0	0	4	JFrog Artifact
DQ	2023-08-23	2023.08	0	0	0	0	1	JFrog Artifact
DQ	2023-09-18	2023.09	0	0	0	0	1	JFrog Artifact
DQ	2023-10-30	2023.10	0	0	0	0	3	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Preview UI

Preview UI Status

The following table shows the status of the preview redesign of Collibra DQ pages as of this release.

Page	Location	Status
Homepage	Homepage	✓ Done
Sidebar navigation	Sidebar navigation	✓ Done
User Profile	User Profile	✓ Done
List View	Views	✓ Done
Assignments	Views	✓ Done
Pulse View	Views	✓ Done
Catalog by Column (Column Manager)	Catalog (Column Manager)	✓ Done
Dataset Manager	Dataset Manager	✓ Done
Alert Definition	Alerts	✓ Done
Alert Notification	Alerts	✓ Done
View Alerts	Alerts	✓ Done
Jobs	Jobs	✓ Done
Jobs Schedule	Jobs Schedule	✓ Done
Rule Definitions	Rules	✓ Done
Rule Summary	Rules	✓ Done

Page	Location	Status
Rule Templates	Rules	✓ Done
Rule Workbench	Rules	✓ Done
Data Classes	Rules	✓ Done
Explorer	Explorer	✓ Done
Reports	Reports	✓ Done
Dataset Profile	Profile	✓ Done
Dataset Findings	Findings	✓ Done
Sign-in Page	Sign-in Page	✓ Done

> **Note** Admin pages are not yet fully available with the new preview UI.

Preview UI Limitations

Explorer

- When using the SQL compiler on the dataset overview for remote files, the Compile button is disabled because the execution of data files at the Spark layer is unsupported.
- You cannot currently upload temp files from the new File Explorer page. This may be addressed in a future release.
- The Formatted view tab on the File Explorer page only supports CSV files.
- When creating a job, the Estimate Job step from the classic Explorer is no longer a required step. However, if incorrect parameters are set, the job may fail when you run it. If this is the case, return to the Sizing step and click Estimate next to Job Size before you Run the job.

Connections

- When adding a driver, if you enter the name of a folder that does not exist, a permission issue prevents the creation of a new folder.
    - A workaround is to use an existing folder.

Admin

- When adding another external assignment queue from the Assignment Queue page, if an external assignment is already configured, the Test Connection and Submit buttons are disabled for the new connection. Only one external assignment queue can be configured at the same time.

Profile

- When adding a distribution rule from the Profile page of a dataset, the Combined and Individual options incorrectly have "OR" and "AND" after them.
- When using the Profile page, Min Length and Max Length does not display the correct string length. This will be addressed in an upcoming release.

Navigation

- The Dataset Overview function on the Metadata Bar is not available for remote files.
- The Dataset Overview modal throws errors for the following connection types:
    - BigQuery (Pushdown and Pullup)
    - Athena CDATA
    - Oracle
    - SAP HANA
- The Dataset Overview function throws errors when you run SQL queries on datasets from S3 and BigQuery connections.

Release 2023.09

- Highlights
- Migration Updates
- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics
- Preview UI
- Preview UI Limitations

Release Information

- Expected release date of Data Quality & Observability Classic 2023.09: October 8, 2023
- Publication dates:
  - Release notes: September 24, 2023
  - Documentation Center: September 29, 2023

Highlights

Pushdown

We're delighted to announce that Pushdown processing for Amazon Athena and Redshift is now available as public previews! Pushdown is an alternative compute method for running DQ jobs, where Collibra DQ submits all of the job's processing directly to a SQL data warehouse, such as Athena and Redshift. When all of your data resides in Athena or Redshift, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirements of a DQ job.

Job Estimator

Collibra DQ utilizes Spark's ability to break large datasets into smaller, more manageable segments called partitions. When you run large Pullup jobs, you can now leverage the job

estimator to automatically calculate and update the number of partition columns required to optimally run and write rules against them. Previously, the only way to know when a job required the scaling of resources was when it failed.

Important
We have migrated our code to a new repository. Consequently, Collibra DQ owl-env.sh jar files are no longer prepended with `owl-*`. Instead, they are now prepended with `dq-*`. For more details, it's crucial that you review the Migration Updates section below.

Migration Updates

We have migrated our code to a new repository for improved internal procedures and security. Because owl-env.sh jar files are now prepended with `dq-*` instead of `owl-*`, if you have automation procedures in place to upgrade Collibra DQ versions, you can use the RegEx replace `regex=r"owl-.*-202.*-SPARK.*\.jar|dq-.*-202.*-SPARK.*\.jar"` to update the jars.

Additionally, please note the following:

- **Standalone Upgrade Steps**
  When upgrading to Collibra DQ 2023.09 on Spark Standalone, the upgrade steps have changed.
  Show steps
  1. Open a terminal session.
  2. Move the old jars from the owl/bin folder with the following commands.

     ```
     mv owl-webapp-<oldversion>-<spark301>.jar /tmp
     mv owl-agent-<oldversion>-<spark301>.jar /tmp
     mv owl-core-<oldversion>-<spark301>.jar /tmp
     ```

  3. Copy the new jars into the owl/bin folder from the extracted package.

```
mv dq-webapp-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-agent-<newversion>-<spark301>.jar
/home/owldq/owl/bin
mv dq-core-<newversion>-<spark301>.jar
/home/owldq/owl/bin
```

4. Copy the latest `owlcheck` and `owlmanage.sh` to /opt/owl/bin directory.

> **Tip** You may also need to run `chmod +x owlcheck owlmanage.sh` to add execute permission to `owlcheck` and `owlmanage.sh`.

5. Start the Collibra DQ Web application.

```
./owlmanage.sh start=owlweb
```

6. Start the Collibra DQ Agent.

```
./owlmanage.sh start=owlagent
```

7. Validate the number of active services.

```
ps -ef | grep owl
```

Enhancements

Capabilities

- When running rules that reference secondary datasets, you now have the option to use serial rule processing to reduce operational costs.
  - Set `-serialrule` to `true` to leverage the Spark cache for the secondary dataset.
- When authenticating your connection to CockroachDB with a PostgreSQL driver, you can now leverage Kerberos TGT without errors.

- When creating a DQ job to run against a remote file data source, you can now select BEL as a delimiter.
- When adding a name to a rule on the Rule Workbench, a helpful message displays if you use an invalid special character.
    - Rule names can only contain alphanumerical characters, underscores, and hyphens.
- When reviewing Rules findings, the default number of rows available to preview is now 6. Previously, the Rules tab only displayed 5 preview rows.
- When creating a Pullup job from Explorer, the Mapping step now automatically maps source columns to target columns.
- We've updated the connection icons on the Explorer, Pulse View, and Admin Connections pages.
    - When you add a new connection from the Admin Connections page, the icon will also update accordingly.
- When monitoring the Jobs page with React on, you can now right-click to open a dataset in a new tab.
- When assigning or validating a finding to an external user whose first name, last name, and external user ID cannot be found or do not exist, you can now set a backup display name in the ConfigMap to ensure you can still validate or assign that finding to the external user.
    - Set `SAML_USE_EXTERNAL_USER_ID_FOR_DISPLAY` to `true`.

Platform

- When deleting a user, the user is now removed from both the user and user_profile metastore tables.
- When loading a large remote file into Explorer, a progress bar now tracks its loading status.

DQ Integration

- When using the configuration wizard in Collibra DQ to set up an integration, your Collibra Platform credentials are now encrypted in the metastore to ensure that your information is always secure.

DQ Cloud

- We've introduced a new endpoint to retrieve aggregated WAL (write-ahead logs) stats.
- When deploying a new Edge site, the TenantAlignmentService no longer stops checking for new tenants in DQ Cloud after 100 attempts.

Pushdown

- When using Archive Break Records for Databricks Pushdown, the 'seqno' column for all break records tables created in Databricks is no longer designated as an identity column. Instead, its default value is now NULL. We've made this adjustment because Databricks does not support concurrent transactions for Delta tables with identity columns.
    - If you already created these tables in your Databricks environment, you need to delete them. Subsequently, allow the Collibra DQ application to re-create these tables for you, ensuring compatibility with the latest changes. To do this, you can run the following SQL commands on your Databricks target schema dedicated to maintaining records of source breaks:

      ```
      drop table collibra_dq_outliers
      drop table collibra_dq_duplicates
      drop table collibra_dq_rules
      drop table collibra_dq_breaks
      drop table collibra_dq_shapes
      ```

    - After you run a DQ job, the tables will be re-created on your Databricks schema.
- We've improved the memory usage to prevent large quantities of rule break records from causing out-of-memory errors.
- When running a Pushdown job, the entire allocated connection pool is now used to extract the maximum allowed parallelism to allow profiling to run in parallel with other layers and reduce the latency of the job.
    - Only the required number of connection threads are used for an activity.
- When creating rules to run against Pushdown datasets, you can now use cross-join queries.
- We've added a Pendo tracking event to track the number of Pushdown jobs and columns in an environment.

Fixes

Capabilities

- When editing DQ jobs for KDB (PostgreSQL) connections, you can now successfully execute a query with a large number of records. (ticket #113493, #116740)
- When creating a BigQuery job, you can now create a dataset for a destination table without throwing an error. (ticket #118534, #122761)
- When archiving break records from Pullup jobs, you can again write break records to S3 storage buckets. Previously, an invalid rule error returned which stated "Exception while inserting break records into S3: No FileSystem for scheme s3". (ticket #121509)
- When you open the Oversized Job Report, you can again see the reports without any errors. (ticket #121752)

Platform

- When reviewing the configuration after running a Validate Source job, you no longer receive a validation error due to lost database, schema, table, field, and query information. (ticket #113977)
- Oracle dataset host strings no longer parse incorrectly. Previously, Oracle dataset host strings were parsed as "jdbc" instead of displaying the correct host string. To see the updated and correct host string for Oracle datasets, rerun your jobs manually via the scheduler or API. (ticket #124846)

DQ Integration

- When completing the connection mapping for your Collibra DQ to Collibra Platform integration, you now correctly see database views from Collibra DQ to the tables and columns to which they relate in Collibra Platform. (ticket #124191, #124213, #125676)

DQ Cloud

- When upgrading to Collibra DQ version 2023.06, you can now see entries in your List View scorecards. Previously, there was a discrepancy between Edge and the Cloud

metastore. (ticket #121624)

Pushdown

- When running a Pushdown job with the /v3/jobs/run API, the username now correctly updates to the authenticated user. (ticket #121192)
- When upgrading to Collibra DQ version 2023.07.2, you can now see the Data Preview for breaking record count for a freeform SQL rule against a Snowflake Pushdown dataset. (ticket #122585)

Known Limitations

Capabilities

- There is a limitation with Validate Source where source columns containing white spaces do not map properly to the target columns.
    - A workaround is to remove the white spaces from the command line and then copy/paste the command line into a new DQ job.
- When using the Pulse View page after adding a new connection, there is a limitation where the icon of the connection does not automatically appear on the Pulse View page. Instead, it appears as a generic JDBC icon.

DQ Security Metrics

> **Note** The medium, high, and critical vulnerabilities of the DQ Connector are now resolved.

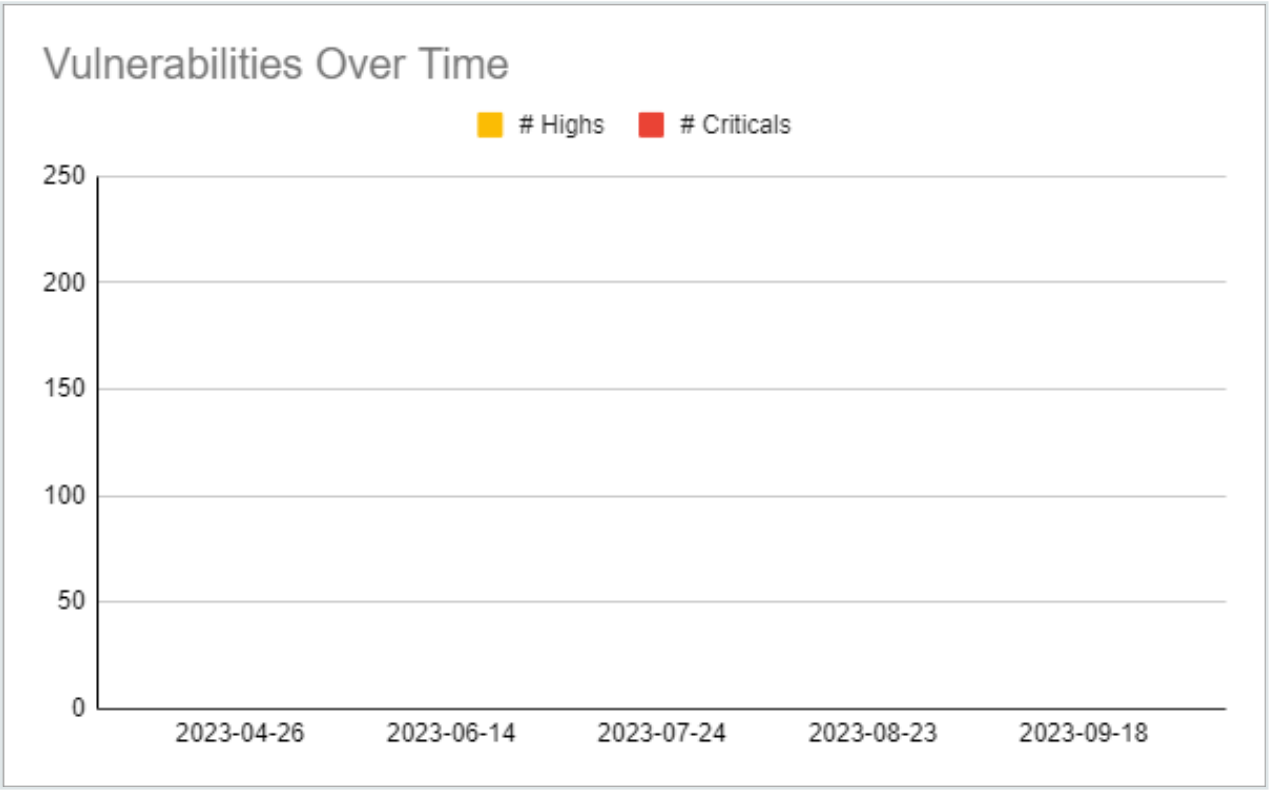
> **Warning** We found 1 critical and 1 high CVE in our JFrog scan. Upon investigation, these CVEs are disputed by Red Hat and no fix is available. For more information, see the official statements from Red Hat:
> https://access.redhat.com/security/cve/cve-2023-0687 (Critical)
> https://access.redhat.com/security/cve/cve-2023-27534 (High)

Show details

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-04-26	2023.05	0	1	1	0	1	JFrog Artifact
DQ	2023-06-14	2023.06	0	0	0	0	0	JFrog Artifact
DQ	2023-07-24	2023.07	0	0	0	0	4	JFrog Artifact
DQ	2023-08-23	2023.08	0	0	0	0	1	JFrog Artifact
DQ	2023-09-18	2023.09	0	0	0	0	1	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Preview UI

Preview UI Status

The following table shows the status of the preview redesign of Collibra DQ pages as of this release.

Page	Location	Status
Homepage	Homepage	✓ Done
Sidebar navigation	Sidebar navigation	✓ Done
User Profile	User Profile	✓ Done
List View	Views	✓ Done
Assignments	Views	✓ Done
Pulse View	Views	✓ Done
Catalog by Column (Column Manager)	Catalog (Column Manager)	✓ Done
Dataset Manager	Dataset Manager	✓ Done
Alert Definition	Alerts	✓ Done
Alert Notification	Alerts	✓ Done
View Alerts	Alerts	✓ Done
Jobs	Jobs	✓ Done
Jobs Schedule	Jobs Schedule	✓ Done
Rule Definitions	Rules	✓ Done
Rule Summary	Rules	✓ Done
Rule Templates	Rules	✓ Done
Rule Workbench	Rules	✓ Done
Data Classes	Rules	✓ Done
Explorer	Explorer	✓ Done
Reports	Reports	✓ Done

Page	Location	Status
Dataset Profile	Profile	✓ Done
Dataset Findings	Findings	✓ Done
Sign-in Page	Sign-in Page	✓ Done

> **Note** Admin pages are not yet fully available with the new preview UI.

Preview UI Limitations

Explorer

- When using the SQL compiler on the dataset overview for remote files, the Compile button is disabled because the execution of data files at the Spark layer is unsupported.
- You cannot currently upload temp files from the new File Explorer page. This may be addressed in a future release.
- The Formatted view tab on the File Explorer page only supports CSV files.

Connections

- When adding a driver, if you enter the name of a folder that does not exist, a permission issue prevents the creation of a new folder.
    - A workaround is to use an existing folder.

Admin

- When adding another external assignment queue from the Assignment Queue page, if an external assignment is already configured, the Test Connection and Submit buttons are disabled for the new connection. Only one external assignment queue can be configured at the same time.

Profile

- When adding a distribution rule from the Profile page of a dataset, the Combined and Individual options incorrectly have "OR" and "AND" after them.
- When using the Profile page, Min Length and Max Length does not display the correct string length. This will be addressed in an upcoming release.

Scorecards

- When creating a new scorecard from the Page dropdown menu, because of a missing function, you cannot currently create a scorecard.
    - While a fix for this is planned for the September (2023.09) release, a workaround is to select the Create Scorecard workflow from the three dots menu instead.

Navigation

- The Dataset Overview function on the Metadata Bar is not available for remote files.
- The Dataset Overview modal throws errors for the following connection types:
    - BigQuery (Pushdown and Pullup)
    - Athena CDATA
    - Oracle
    - SAP HANA
- The Dataset Overview function throws errors when you run SQL queries on datasets from S3 and BigQuery connections.

Release 2023.08

- Highlights
- New Features
- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics
- Preview UI Redesign
- Preview UI Limitations

Highlights

Pushdown

We're delighted to announce that Pushdown processing for Databricks is now generally available! Pushdown is an alternative compute method for running DQ jobs, where Collibra DQ submits all of the job's processing directly to a SQL data warehouse, such as Databricks. When all of your data resides in Databricks, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirements of a DQ job.

Pushdown

We're also excited to announce that Pushdown processing for Google BigQuery is now available as a public preview option!

> **Note** The legacy documentation hosted at dq-docs.collibra.com has reached its end-of-life period and now has a redirect link to the official Data Quality & Observability Classic documentation.

New Features

Capabilities

- When reviewing outlier findings, you can now use the Invalidate All option to invalidate all outliers from a given job run in bulk.
- When configuring rule details on the Rule Workbench, you can now define the Scoring Type as either a Percent, which is the default scoring type, or Absolute, which deducts points for breaking rules where the percentage is greater than 0.
- When reviewing rule break findings, you can now select Rule Breaks from the Actions dropdown menu to preview the rule break export file and copy a signed link to the external storage location, giving you more control over how you use and share break records.

DQ Cloud

- When upgrades of DQ Edge sites are required, you can now leverage a utility script to update the Edge DQ version without reinstalling the Edge site.
- We've added the config parameter `licenseSource` to the Collibra DQ Helm chart to make it easier for our internal teams to update DQ Cloud licenses.
    - `"config"` is the default value for DQ Cloud deployments.

Pushdown

- When archiving break records from Databricks Pushdown jobs, you can now write them directly to a database or schema in Databricks.
- When you archive break records to the source warehouse, records are now pruned according to specified parameters to prevent tables or schemas from growing to unreasonable sizes. When a job gets pruned from the jobs table, the source break records from the datasource get pruned as well.

Enhancements

Capabilities

- When scheduling jobs to run automatically, jobs now reflect the runId of the time and timezone you set. Previously, the runId of scheduled jobs reflected the default UTC server time, irrespective of the timezone you set.
- When setting up a connection to a Google BigQuery data source, you can now use the Service Account Credential option to upload a GCP JSON credential file to authenticate your connection. This enhancement means you no longer need to use the workaround of uploading a JSON credential file as a base64 encoded Kerberos secret.

Platform

- The endpoints for the controller-email API have changed. The following endpoints are now available:
    - GET /v2/email/server
    - POST /v2/email/server

  - POST /v2/email/server/validate
  - POST /v2/email/group
  - GET /v2/email/server/status
  - GET /v2/email/groups

> **Note** For more information about the new endpoints, refer to the UI Internal option in Swagger.

Integration

- When using either the Update Integration Credentials or Add New Integration modal to map a connection, the Connections tab now only displays the full database mapping when you click **Show Full Mapping**, which improves the loading time and enhances the overall experience of the Connections tab.
  - Additionally, there is now a **Save and Continue** button on the Connections tab to ensure your mappings save before proceeding to the next step.

DQ Cloud

- Pendo tracking events no longer contain license_key information when log files are sent to Collibra Console.
- We've improved the performance and resilience of Collibra DQ on Edge sites.

Pushdown

- If Archive Break Records Location is not selected when setting up archive break records from the Connections page, the default schema is now the default schema of the database platform .yaml file. Previously, when a break records output location was not specified, the default location, PUBLIC, would be used.
- When writing custom rules with rlike (regex) operators against Pushdown datasets, exception messages no longer throw when the job runs.
- When running a Pushdown job from the Collibra DQ app, not via API, the correct column count displays on the Findings page. Previously, the v2/run-job-json returned empty columns, which resulted in the total number of columns displayed on the Findings page as 0.

Fixes

Capabilities

- When adding a rule with a string match to a dataset where the string contains parentheses, extra spaces around the parentheses are no longer mistakenly added. (ticket #117055, 118319)
- When selecting a region for an Amazon S3 connection, you can now use AP_SOUTHEAST_3 and AP_SOUTHEAST_4. (ticket #119535)
- When assessing outlier percent change calculations on Findings, the percentage now displays correctly. (ticket #114045)
- When using the out-of-the-box Template rule, Not_In_Current_Run as a dataset rule, an exception no longer throws when the job runs. (ticket #118401)

Platform

- When your Collibra DQ session expires, you are now redirected to the sign-in page. (ticket #111578)
- When migrating DQ jobs from one environment to another, columns that were selected to be included in an outlier check in the source environment now persist to the target environment. Previously, some columns that were selected in the source environment did not persist to the target. (ticket #115224)
- When attempting to edit a completed job on a Redshift connection, the preview limit is now set to 30 rows. Previously, larger datasets experienced long load times or timed out, which prevented you from editing them from Explorer. (ticket #119831, 120245)
- Fixed the Critical CVE CVE-2023-34034 by upgrading the spring library. (ticket #122280)
- When running a job, you no longer receive SQL grammar errors. (ticket #120691)

Known Limitations

Capabilities

- When using the rule breaks capability on the classic Rules Findings tab and rule break records from native rules do not exist in the metastore, the preview modal displays a

blank preview and sample file.
- When using the rule breaks capability and the remote archive location does not have write permissions, the exception details of the rule being archived are only visible on the Rules Findings tab.
- After updating the timezone from the default UTC timezone to a different one on a dataset with multiple days of data, the dates on the Findings page charts and Metadata Bar still reflect the default UTC timezone.
    - A fix will be included in Collibra DQ version 2023.11.

Pushdown

- The archive break records capability cannot be configured from the settings modal on the Explorer page for BigQuery Pushdown connections.
- When using the archive break records capability, BigQuery Pushdown currently only supports rule break records.
    - Additional support is planned for an upcoming release.
- When using the archive break records capability to archive rule breaks generated from freeform rules with explicitly selected columns, and not `SELECT *`, you must include the Link ID column in the rule query for break records to archive correctly.
- When you select a date column as the column of reference in the time slice filter of a BigQuery dataset, an unsupported data type message displays. While this will be resolved in an upcoming release, a temporary workaround is to use the SQL View option to manually update the source query to reference a date column. For `example,`

```
select * from example.nyse where trade_date = safe_cast('${rd}' as DATE)
```

DQ Security Metrics

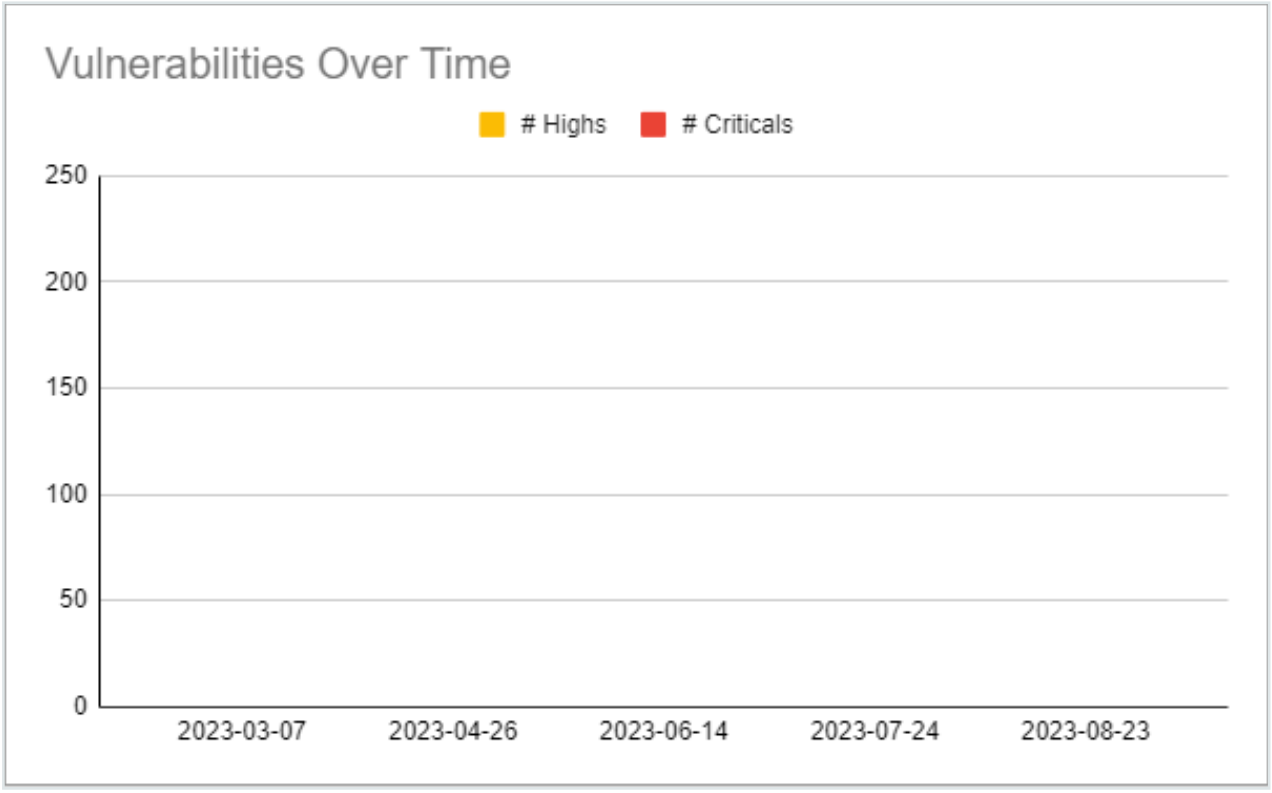
> **Note** The medium, high, and critical vulnerabilities of the DQ Connector are now resolved.

Warning We found 1 critical and 1 high CVE in our JFrog scan. Upon investigation, these CVEs are disputed by Red Hat and no fix is available. For more information, see the official statements from Red Hat:
https://access.redhat.com/security/cve/cve-2023-0687 (Critical)
https://access.redhat.com/security/cve/cve-2023-27534 (High)

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-03-07	2023.03	0	1	1	0	0	JFrog Artifact
DQ	2023-04-26	2023.05	0	1	1	0	1	JFrog Artifact
DQ	2023-06-14	2023.06	0	0	0	0	0	JFrog Artifact
DQ	2023-07-24	2023.07	0	0	0	0	4	JFrog Artifact
DQ	2023-08-23	2023.08	0	0	0	0	1	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



Preview UI Redesign

The following table shows the status of the preview redesign of Collibra DQ pages as of this release. Because the status of these pages only reflects Collibra DQ's internal test environment and completed engineering work, pages marked as "Done" are not necessarily available externally. Full availability of the new preview pages is planned for an upcoming release.

Page	Location	Status
Homepage	Homepage	✓ Done
Sidebar navigation	Sidebar navigation	✓ Done
User Profile	User Profile	✓ Done
List View	Views	✓ Done
Assignments	Views	✓ Done
Pulse View	Views	✓ Done
Catalog by Column (Column Manager)	Catalog (Column Manager)	✓ Done
Dataset Manager	Dataset Manager	✓ Done
Alert Definition	Alerts	✓ Done
Alert Notification	Alerts	✓ Done
View Alerts	Alerts	✓ Done
Jobs	Jobs	✓ Done
Jobs Schedule	Jobs Schedule	✓ Done
Rule Definitions	Rules	✓ Done
Rule Summary	Rules	✓ Done

Page	Location	Status
Rule Templates	Rules	✓ Done
Rule Workbench	Rules	✓ Done
Data Classes	Rules	✓ Done
Explorer	Explorer	✕ In Progress
Reports	Reports	✓ Done
Dataset Profile	Profile	✓ Done
Dataset Findings	Findings	✓ Done
Sign-in Page	Sign-in Page	✓ Done

> **Note** Admin pages are not yet fully available with the new preview UI.

Preview UI Limitations

Explorer

- When using the SQL compiler on the dataset overview for remote files, the Compile button is disabled because the execution of data files at the Spark layer is unsupported.
- You cannot currently upload temp files from the new File Explorer page. This may be addressed in a future release.
- The Formatted view tab on the File Explorer page only supports CSV files.

Connections

- When adding a driver, if you enter the name of a folder that does not exist, a permission issue prevents the creation of a new folder.
    - A workaround is to use an existing folder.

Admin

- When adding another external assignment queue from the Assignment Queue page, if an external assignment is already configured, the Test Connection and Submit buttons are disabled for the new connection. Only one external assignment queue can be configured at the same time.

Scorecards

- When creating a new scorecard from the Page dropdown menu, because of a missing function, you cannot currently create a scorecard.
    - While a fix for this is planned for the September (2023.09) release, a workaround is to select the Create Scorecard workflow from the three dots menu instead.

Navigation

- The Dataset Overview function on the metabar is not available for remote files.

Release 2023.07

- Highlights
- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics
- MUI Redesign
- React Limitations
- Maintenance Updates

Highlights

Integrations

When integrating metadata from Collibra DQ with Collibra Platform, you can now access the Integration page from the Admin Console in Collibra DQ. The Integration page features an integration configuration wizard, which leads you through a guided mapping, and a dataset status table to monitor and edit existing integrations.

The integration configuration wizard lets you:

- Seamlessly map data quality objects, such as tables, schemas, and columns to resources in Collibra Platform.
- Align DQ tenants with Collibra Platform communities.
- Map DQ dimensions to their corresponding dimensions in Collibra Platform.

By using these new capabilities, you can now efficiently synchronize Collibra DQ metadata with Collibra DGC with confidence and ease.

Pushdown

When storing break records from Snowflake Pushdown jobs, you can now write them directly to a database or schema in your data source. By removing the reliance on the PostgreSQL metastore for the archival of break records, you can:

- Eliminate network traffic costs for better performance.
- Reduce storage limit restrictions.
- Improve security by generating break records that never leave Snowflake.
- Enhance your reporting abilities by removing the need to move break records from the PostgreSQL metastore to your data source.
- Join datasets against your break records.

> **Note** The legacy documentation hosted at dq-docs.collibra.com will reach its end-of-life alongside the upcoming August (2023.08) release of Data Quality & Observability Classic.

Enhancements

Capabilities

- The Reports landing page now has improved organization, helpful descriptions of each report and dashboard type, and links to the documentation.
- When using the v3/jobs API, you can now specify a time range to filter the list of jobs with the following parameters:
    - startTimeBeginning
    - startTimeEnd
    - updateTimeBeginning
    - updateTimeEnd
- When reviewing the Rule Definitions dashboard, the Dataset column now includes links to the Findings page of the most recent run of a given dataset.
- The labels no longer overlap when using the Rule Summary charts.
- When using React, we renamed Catalog Dataset to Dataset Manager, and Column Manager, formerly called Catalog by Column, is now generally available.

Platform

- When adding or modifying a connection and a Connection URL string or a property in the Driver Properties field is invalid, the error message that returns now includes helpful information to help you troubleshoot the error.

Pushdown

- When including a shapes check in a Pushdown job, the shapes query now has a limit of the maximum number of shapes per column plus one to ignore any columns that exceed the maximum number of shapes per column.
- When scanning for outliers in a Databricks Pushdown job, you can now select the Categorical option on the Outliers Configuration modal to check for categorical outliers.

Fixes

Capabilities

- When profiling a PostgreSQL table that contains infinity ∞, negative infinity −∞, NaN values in numeric columns, you can once again run jobs successfully. (ticket #112693)
- When using the % symbol as a wildcard in queries against Db2 datasets, the % symbol no longer converts to an unreadable character, and the query runs correctly. (ticket #113618)
- When running a scheduled job during a restricted time in a multi-tenant environment, the schedule restriction functionality again works correctly. Previously, jobs on certain tenants could still run during restricted times. (ticket #113794)
- When setting up an SAP HANA connection, you can again successfully connect. Previously, a Java incompatibility prevented successful SAP HANA connections. (ticket #114883)

  > **Note** When using a JDK 8 build of Collibra DQ, you must use the SAP HANA driver provided in the Collibra DQ build package.

Platform

- When reviewing the Breaks tab of a dataset's associated rules on CockroachDB connections, you can once again use the search bar to filter break records. (ticket #105410)
- When configuring multi-tenancy on CockroachDB connections, you can once again add new tenants because the V193 migration script is now compatible with CockroachDB environments V22.1.x or later. (ticket #119528)
- When reviewing the web logs for Explorer, all potential personally identifiable information (PII) is now removed from the logging message. (ticket #119375)
- When deploying a Kubernetes pod, the security admission and RBAC clusternode warnings are now resolved. (ticket #113204)
- When upgrading a multi-tenant deployment of Collibra DQ to a newer version, the agent display name no longer reverts to the agent name if you previously set the agent display name to something else. (ticket #114355, 119704)

- When using the Alert Definitions page without an email server configured, an SMTP configuration error message incorrectly displayed based on a user's role. (ticket #114086)
- When a user without a role mapped to a given connection attempts to access a dataset from Explorer via the Edit Dataset button on the Findings page, their access is now restricted until a role is assigned. Previously, role restrictions were not checked from Explorer. (ticket #115217)
- When using AssumeRole on an S3 connection, the STS server now defaults to the same region that the connection is configured in. (ticket #118448)

Pushdown

- When running a Snowflake Pushdown job that includes the profile activity, you no longer get an error message when a binary data type column exists in your dataset. (ticket #111977)

Known Limitations

DQ Cloud

- Because Livy is not yet available for DQ Cloud, in order to display the data preview of remote files in the UI, you need to use temporary Spark scratch disk space to momentarily cache the preview. Until Livy is available for DQ Cloud, the caching of preview data occurs within DQ Cloud instead of Edge.
- After upgrading a DQ Cloud instance, previously updated agent display names revert to the agent name. Any previously saved agent display names need to be manually re-entered after upgrading.

DQ Security Metrics

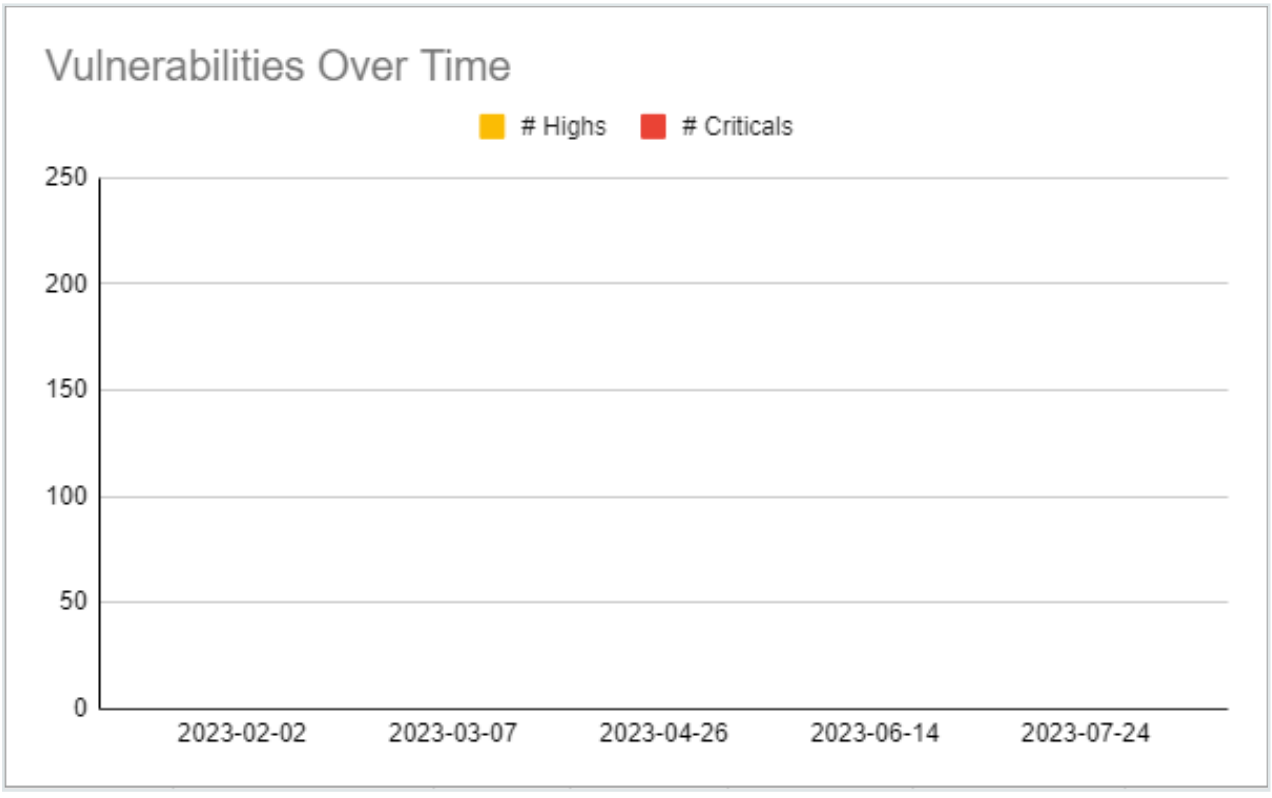
> **Note** The medium, high, and critical vulnerabilities of the DQ Connector are now resolved.

Warning We found 1 critical and 1 high CVE in our JFrog scan. Upon investigation, these CVEs are disputed by Red Hat and no fix is available. For more information, see the official statements from Red Hat:
https://access.redhat.com/security/cve/cve-2023-0687 (Critical)
https://access.redhat.com/security/cve/cve-2023-27534 (High)

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-02-02	2023.02	0	0	0	0	0	JFrog Artifact
DQ	2023-03-07	2023.03	0	1	1	0	0	JFrog Artifact
DQ	2023-04-26	2023.05	0	1	1	0	1	JFrog Artifact
DQ	2023-06-14	2023.06	0	0	0	0	0	JFrog Artifact
DQ	2023-07-24	2023.07	0	0	0	0	4	JFrog Artifact

The following image shows a table of Collibra DQ security metrics arranged by release version.



MUI Redesign

The following table shows the status of the MUI redesign of Collibra DQ pages as of this release. Because the status of these pages only reflects Collibra DQ's internal test environment and completed engineering work, pages marked as "Done" are not necessarily available externally. Full availability of the new MUI pages is planned for an upcoming release.

Page	Location	Status
Homepage	Homepage	✓ Done
Sidebar navigation	Sidebar navigation	✓ Done
User Profile	User Profile	✓ Done
List View	Views	✓ Done
Assignments	Views	✓ Done
Pulse View	Views	✓ Done
Catalog by Column (Column Manager)	Catalog (Column Manager)	✓ Done
Dataset Manager	Dataset Manager	✓ Done
Alert Definition	Alerts	✓ Done
Alert Notification	Alerts	✓ Done
View Alerts	Alerts	✓ Done
Jobs	Jobs	✓ Done
Jobs Schedule	Jobs Schedule	✓ Done
Rule Definitions	Rules	✓ Done
Rule Summary	Rules	✓ Done

Page	Location	Status
Rule Templates	Rules	✓ Done
Rule Workbench	Rules	✕ In Progress
Data Classes	Rules	✓ Done
Explorer	Explorer	✕ In Progress
Reports	Reports	✕ In Progress
Dataset Profile	Profile	✕ In Progress
Dataset Findings	Findings	✓ Done
Sign-in Page	Sign-in Page	✓ Done

> **Note** Admin pages are not yet fully available with the new React MUI.

React Limitations

Explorer

- When using the SQL compiler on the dataset overview for remote files, the Compile button is disabled because the execution of data files at the Spark layer is unsupported.
- You cannot currently upload temp files from the new File Explorer page. This may be addressed in a future release.

Connections

- When adding a driver, if you enter the name of a folder that does not exist, a permission issue prevents the creation of a new folder.
    - A workaround is to use an existing folder.

Admin

- When adding another external assignment queue from the Assignment Queue page, if an external assignment is already configured, the Test Connection and Submit buttons are disabled for the new connection. Only one external assignment queue can be configured at the same time.

Scorecards

- When creating a new scorecard from the Page dropdown menu, because of a missing function, you cannot currently create a scorecard.
    - While a fix for this is planned for the September (2023.09) release, a workaround is to select the Create Scorecard workflow from the three dots menu instead.

Navigation

- While the breadcrumbs at the top of all legacy pages are no longer available in the new preview pages, as of the upcoming August (2023.08) release, a metadata bar will be anchored near the top of the Finding, Profile, Dataset Preview, and Alert Builder for quick access to all essential pages.

Maintenance Updates

2023.07.2

- Fixed an issue where users who previously used the API integration on Collibra DQ version 2023.05 to import Collibra DQ metadata into Collibra Platform received an error message for Flyway scripts when migrating records from the integration status table to the integration history table after upgrading to Collibra DQ version 2023.07.

- Fixed an issue where preview data on the Findings page did not display fully when column values used scientific notation.

2023.07.4

- When using the integration API, the DQ Rule governs DQ Job relation of the DQ Rule asset no longer incorrectly identifies the "SOURCE" and "TARGET" properties in the integration JSON file.
- Fixed a critical CVE (CVE-2023-34034) in Data Quality & Observability Classic version 2023.07 that was identified after the release date.

Release 2023.06

- Highlights
- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics
- MUI Redesign

Highlights

- When running rules against datasets that return many breaking records, you can now optionally export the records to an external Amazon S3 or Azure storage bucket instead of writing them to the PostgreSQL Metastore. By offloading break records to cloud storage, you have more control over how you manage the results of your DQ jobs, and you can store data in the supported remote connection of your choice.
- When building SQL rules to run against a dataset, you can now use the new Rule Workbench to create, edit, and preview SQL conditions to help verify that your data adheres to your organization's business requirements.

Enhancements

Capabilities

- You can now scan data stored in Trino using the Starburst Trino JDBC connection.
    - Ensure that your connection URL contains the following substring: &source-e=jdbc:presto. For example, `jdbc:trino://example-host.trino.galaxy.star-burst.io:443/sample?SSL=true&source=jdbc:presto`
- When working with remote file connections, you can now leverage the Hudi Spark connector to scan Hudi files.

    > **Note** The Hudi Spark connector requires a separate package to work properly due to a security vulnerability in the Hudi bundle jar. Please reach out to your CSM for more information about accessing this package.

- When using the Estimate Job feature to gauge the resource requirements of a Spark job, you can now set `partitionautocal` to true on the Admin Limits settings page to automatically calculate Spark partitioning for optimal performance results.
- The Estimate Job feature now estimates the overall memory requirements instead of the memory per executor.
- You can now use the Data Class and Template Rules pages to manage out-of-the-box and custom Data Class and Template rules.
- The dayswithoutData and runswithoutData stat rules now count Day 0 and Run 0 as the first day and run without data.
- The dayswithoutData stat rule now counts days with stale data for 365 days instead of 100.
- When using Source to check a column for mismatched data, you can now select the Strict Source Downscoring option to include `-validatevaluesthresholdstrictdownscore` in your query during Source setup and base the downscore value on the number of cell mismatches found across all columns. When using this option, you must also assign a key column for source validation.
- When a large number of Dupes are found in a DQ job run, the getDataPreviewByObsType API now has improved performance and no longer times out.

Platform

- When adding a new dataset from a Google BigQuery connection, you can now successfully import, query, and identify EXTERNAL table types.
- To address the conflict of JAASConf secret deletion with DQ jobs having longer Spark launch times for Driver and Executor pods, we adjusted the DQ agent's clean-up routine. (ticket #109256)

Pushdown

- When configuring the settings of a Pushdown job, you can now choose to log all SQL queries for a dataset by turning SQL Logging on or off from the Settings modal.
- When reviewing the Value column for Outliers on the Findings page, low-precision values now truncate after 2 decimal places when all the trailing values are 0. For example, 3780.0000000 now truncates to 3780.00. High-precision values, such as 162.9583789, still display all decimal values.
- When adding ML layers for Databricks Pushdown jobs, you can now configure numerical outliers. Support for categorical outliers is planned for an upcoming release.
- Because Pushdown is not yet available for DQ Cloud, the Pushdown checkbox option on the Databricks and Snowflake connection templates is temporarily unavailable when setting up a connection on a DQ Cloud deployment.

DQ Cloud

- When admin or tenant admin users need to view the state of an Edge site identifier graph, you can now use the following APIs:
    - As admin
        - GET /v3/diagnostics/edgeSiteIdCache returns a text/plain body with a graph when successful.
        - DELETE /v3/diagnostics/edgeSiteIdCache returns no response body when successful.
    - As tenant admin
        - GET /tenant/v3/diagnostics/edgeSiteIdCache/{tenant} returns a text/plain body with a graph when successful.
        - DELETE /tenant/v3/diagnostics/edgeSiteIdCache/{tenant} returns no response body when successful.

> **Note** The tenant admin API works only for the {tenant} where the user is tenant admin.

Fixes

Capabilities

- When viewing Outlier records on the Findings page, you can again drill into individual records to view the associated graph. (ticket #110281)
- Rules with null characters in the data again return rule breaks. Previously, null characters caused rule break inserts into the Metastore to return exception messages. (ticket #103402)
- When exporting rule breaking records to Excel, integer values from Collibra DQ no longer display as decimal values in the Excel export file. (ticket #112830)
- Fixed an issue where Shapes detection incorrectly assigned non-numeric text as numeric scientific notation. (ticket #113704)

Platform

- When using an SSO configuration to sign into Collibra DQ, you no longer get errors when you attempt to sign in. The refactoring in this fix initially looks up a user profile by user-name, then by external user ID, before attempting to lookup by externalUserId until a user is found and returned. (ticket #109171)
- When drilling into a column on the Profile page, Min and Max string length again displays the correct values. (ticket #110044)
- When reviewing the Data Summary Report, the Table/File Name column now displays the correct name. (ticket #111508)
- Fixed the builds for the 2023.05 Collibra DQ version. Previously, jobs failed with an "Unknown" status during the Alerts activity and showed a "NoClassDefFoundError" in the job log. (ticket #115956, 116052)
- When running jobs on existing datasets with alerts configured, jobs no longer fail with an error when the ALERT_SCHEDULE_ENABLED option is set to TRUE. (ticket #112837)

DQ Cloud

- When drilling into rule break records on the Findings page, data preview records again display correctly. (ticket #114162)
- When reviewing the Completeness Report, data now displays correctly when you load the page and update as the inputs change. (ticket #115312)

Pushdown

- When using query wrapping in Snowflake Pushdown profiling, you can now successfully run a profile scan without errors. (ticket #111977)

Known Limitations

Capabilities

- When using BigQuery datasets and tables with names that begin with a number, there is a limitation that causes DQ jobs to fail.
    - A workaround is to wrap the dataset or table name with back quotes ` and then update the scope before you run the DQ job. For example, `select * from `321_apples.example`` and `select * from `samples.311_service_requests``
- When using the Rule Template to build SQL rules, there is a limitation with SQLF type rules where an extra select clause is added into the query when:
    - SQL keywords, such as `select`, `from`, and `where` are not lowercase.
        - A workaround is to write all keywords in lowercase.
    - "@DATASET", "@dataset", and "DATASET" are used to reference a dataset.
        - Instead, use "dataset" in all lowercase letters without an "@" symbol.
- When exporting break records to external storage buckets, there is a limitation that prevents support for the feature on Amazon S3 and Azure connections that use SAML and Instance Profile authentication.
    - A workaround is to use Access Key authentication instead.
- When using the getRuleBreaksCreateTable API to create a table statement for break records exported to an external storage bucket, there is currently a limitation that prevents the return of the proper response. While this issue will be addressed in the 2023.07 release, this API is not intended to be used at this time.

Platform

- When leveraging the Spark 3.2.2 standalone installation that comes with the AWS marketplace installation, there is a limitation due to a jar mismatch for the AWS S3 Archive feature. The issue is that the existing hadoop-aws.3.2.1.jar file is incompatible with the feature.
    - A workaround is to update the hadoop-aws-3.2.1.jar to hadoop-aws-3.3.1.jar in the spark/jars directory. The necessary .jar file can be obtained from the following link: Apache Downloads.

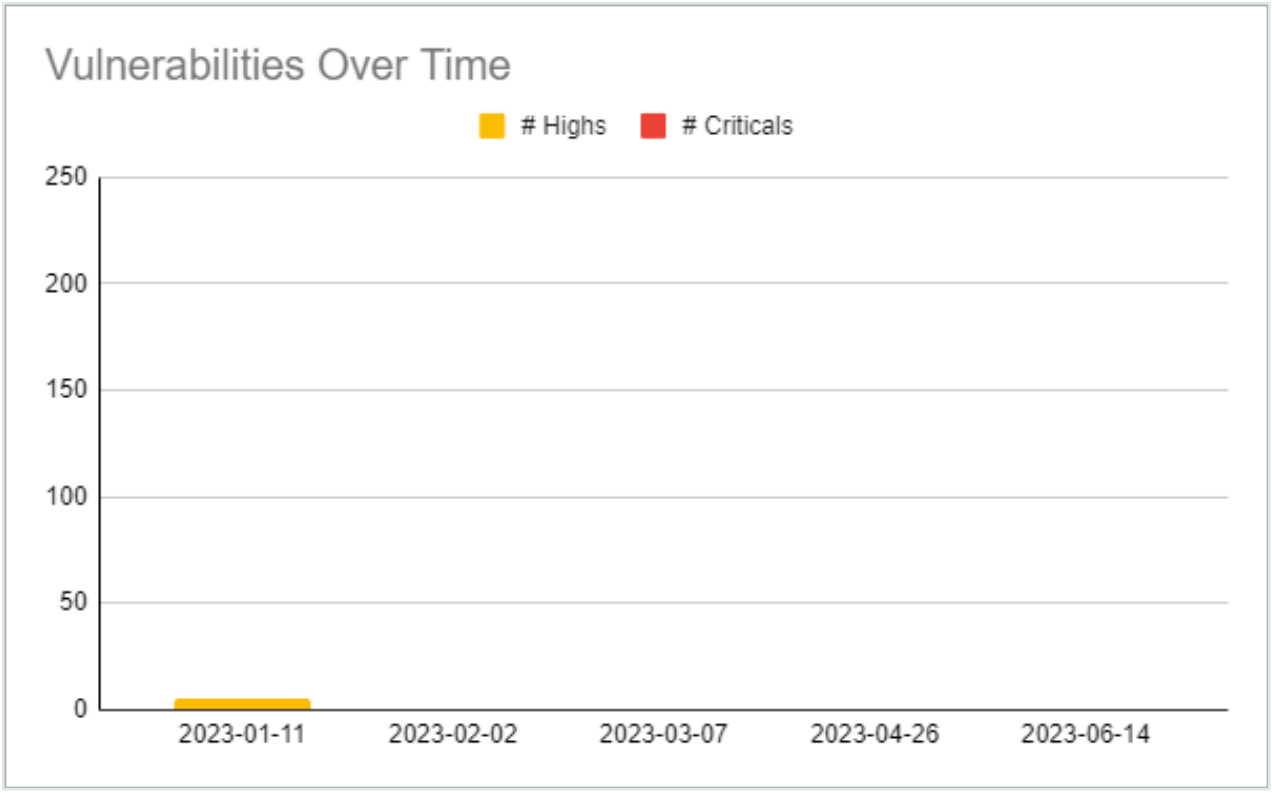
      > **Note** If you encounter any difficulties locating the necessary jar file on the Apache Downloads page, contact your CS or SE for assistance.

DQ Cloud

- DQ Cloud does not currently support SAML configuration.
- When reviewing the Completeness Report, new data only displays correctly after you upgrade your Collibra DQ Cloud instance to version 2023.05.2 or later.
- If an Edge site needs to be reinstalled, you must use the original PostgreSQL metastore database or metastore corruption may occur in Collibra DQ. If necessary, restore the metastore database from a backup before reinstalling. Ensure that the installation command line parameter `collibra_edge.collibra.dq.metastoreUrl` points to the correct database.

DQ Security Metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.



The following image shows a table of Collibra DQ security metrics arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2023-01-11	2023.01	0	1	1	5	2	JFrog Artifact
DQ	2023-02-02	2023.02	0	0	0	0	0	JFrog Artifact
DQ	2023-03-07	2023.03	0	1	1	0	0	JFrog Artifact
DQ	2023-04-26	2023.05	0	1	1	0	1	JFrog Artifact
DQ	2023-06-14	2023.06	0	0	0	0	0	JFrog Artifact

MUI Redesign

The following table shows the status of the MUI redesign of Collibra DQ pages as of this release. Because the status of these pages only reflects Collibra DQ's internal test environment and completed engineering work, pages marked as "Done" are not necessarily available externally. Full availability of the new MUI pages is planned for an upcoming release.

Page	Location	Status
Homepage	Homepage	✓ Done
Sidebar navigation	Sidebar navigation	✓ Done

Page	Location	Status
User Profile	User Profile	✓ Done
List View	Views	✓ Done
Assignments	Views	✓ Done
Pulse View	Views	✓ Done
Catalog by Column (Column Manager)	Catalog (Column Manager)	✓ Done
Dataset Manager	Dataset Manager	✓ Done
Alert Definition	Alerts	✓ Done
Alert Notification	Alerts	✓ Done
View Alerts	Alerts	✓ Done
Jobs	Jobs	✓ Done
Jobs Schedule	Jobs Schedule	✓ Done
Rule Definitions	Rules	✓ Done
Rule Summary	Rules	✓ Done
Rule Templates	Rules	✓ Done
Rule Workbench	Rules	✕ In Progress
Data Classes	Rules	✓ Done
Explorer	Explorer	✕ In Progress
Reports	Reports	✕ In Progress

Page	Location	Status
Dataset Profile	Profile	✕ In Progress
Dataset Findings	Findings	✓ Done
Sign-in Page	Sign-in Page	✓ Done

Release 2023.05

- Highlights
- New Features
- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics
- Maintenance Updates

Highlights

- We are excited to announce that Pushdown processing for Snowflake is now generally available! Pushdown is an alternative compute method for running DQ jobs, where Collibra DQ submits all of the job's processing directly to a SQL data warehouse, such as Snowflake. When all of your data resides in Snowflake, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirement of a DQ Job.
- We are also excited to announce that Pushdown processing for Databricks is now available as a public preview! When you use Pushdown for Databricks ensure that PUSHDOWN_FOR_DATABRICKS is set to `TRUE` in the Application Configuration page of the Admin Console.
- You can now use Swagger to leverage the new Integration API to send Collibra DQ rules, ML layer findings, and associated data quality scores to Collibra Platform. With data quality details in Collibra Platform, you can develop a more robust understanding of the health and performance of your organization's data.

New Features

Capabilities

- When you create or edit a connection with REACT_MUI off, you can now define variables from the Connections template to securely enter sensitive properties, such as user credentials.
- When you enable Include Links for recent job runs, the email alert now includes multiple contextual links to help you access the affected areas of the application.

Platform

- You can now use a serverless agent to submit jobs from the UI. This lets you have a Collibra DQ installation without a native DQ agent, and allows for extensive agent customization and lightweight deployment options against new compute engines. This is currently only supported for Dataproc.

Pushdown

- You can now set the maxconcurrentjobs config on the Admin Limits page to specify a maximum number of concurrent Pushdown jobs to run at once to prevent the Job service from overloading.
- You can now use the new /v3/rules/{dataset}/validatePushdown endpoint to validate the syntax of Pushdown rules.

DQ Cloud

- You can now edit and delete DQ Cloud connections.
    - When you delete a connection all of the links to its agent are also deleted. When you create a new connection in DQ Cloud, ensure that in the connections template your Connection URL is unique and select an agent from the Target Agent dropdown menu. This step is required and you cannot change your agent after the connection is created.

Enhancements

Capabilities

- When you use a key comparison for source validation of decimal values that include scales of 0, you can now select a new Ignore Precision option from the Source configuration tab or set the flag, `-validatevaluesignoreprecision`, to true from the command line to ignore scales of 0 for integers before decimals and fractional values after decimals.

> **Note** `-validatevaluesignoreprecision` is only available for Validate Source when you include a key.

Platform

- The connection template is now updated to the correct Impala JDBC driver, `com.cloudera.impala.jdbc.Driver`
- The Kafka Streaming Connection is no longer available from the Connections Management page because it has finished its preview cycle and has not been promoted to becoming generally available.
- The reactor-core package now supports Spark245 for successful Azure connections.
- When you configure AdaptiveRules, Data Type Check is now included in the Schema detection activity.
- When you select the Include Links option on the Alerts Configuration page, you must also enter your host address from the URL of your application into the HOST_NAME input field on the Application Configuration page. For example, if your Collibra DQ URL is http://dq.collibra.com, you need to enter http://dq.collibra.com into the HOST_NAME input field.
    - The Include Links feature for alerts now has logging to show when the Include Links option is selected but HOST_NAME is not set.
        - Informational text is now available on the Alerts Configuration page to help with the configuration of alerts.

Pushdown

- When you run a Pushdown Job with Replay on, the start time of child Jobs in the Jobs queue now reflects their actual start times instead of the start time of the parent job from which the Replay initiated.
- The Outliers option of the Add Layers step is now disabled for Databricks Pushdown job configuration.

DQ Cloud

- DQ Cloud is now upgraded to the Collibra DQ 2023.05 version.

Fixes

Capabilities

- Fixed an issue with the email lookup query of password reset requests that prevented users from resetting passwords. (ticket #110075)
- Added a new admin limit flag, `valsrcdisableaq`, to fix a slow performance issue for record insertion into the assignment_q metastore table. (ticket #108547)
    - `valsrcdisableaq` lets you turn the connection between the assignmentQ feature and Source activity on or off. When you set it to true, you cannot invalidate or retrain any Source findings.
- Removed the **scan** button from Explorer ▸ Connections because it was no longer supported from the UI. You can still profile multiple tables at the same time with the /v3/dataSetDefs/ or /v2/run-catalog-scan-json/ endpoint. (ticket #107687)
- Fixed an issue where, when DQ jobs were created from a BigQuery source and a Hive source was added, the suffix '/core' was appended to the end of the driver name of the secondary dataset driver path, which resulted in long run times and an unavailable agent. (ticket #109796, #111432)
- Fixed an issue with where loading a CSV file that use Custom delimiters on S3 remote file connections caused the file to hang in the load activity. (ticket #108890)
- Fixed an issue with an incorrect mismatch in the validate source results, when both the source and target had values of zero. (ticket #106788)
- Fixed an issue where attempts to invalidate findings in bulk hung on the "Updating Dataset" message because of an unsupported request method. (ticket #106826, 113881)

- Fixed a permission issue on SQL Server Kerberos-based connections for queries run in Collibra DQ on system tables in SQL Server. (ticket #110773)

Platform

- Fixed issues where external users could not successfully run jobs. (ticket #110773, 111988, 112417, 112857)
- Fixed an issue with Collibra DQ on AWS Cloud where Instance Profile authentication caused scheduled S3 jobs to fail. (ticket #110570, 110574)
- Addressed all the vulnerabilities of the optional drivers from Collibra DQ version 2023.03. You can now take the images with the optional drivers, as they now pass security scans. (ticket #112617)

DQ Cloud

- Fixed an issue where scheduled jobs did not run because they became stuck in "Staged" status. (ticket #112802)

Known Limitations

Capabilities

- A limitation with S3 remote file connections that use an escape character as a delimiter causes columns to parse incorrectly. For example: /a
- When you edit an existing connection and its details, another connection is created and any sensitive properties from the old connection do not persist in the new connection.
    - A workaround is to recreate the sensitive properties from the old connection on the new connection.
- When you create or edit connections, the new sensitive properties feature is only available with React turned off.
    - This only works for datasets created before 2023.05. Permalinks to specific Job runs in alert emails only work for datasets that run after the introduction of this feature in Collibra DQ version 2023.05. If a dataset was created before this implementation and runs on a schedule, the permalinks will not work.
- Native SQL is not currently supported on the Rules Workbench.

Platform

- If you are an external user of Collibra DQ versions 2023.03, 2023.04, and their patches, you may experience issues where completed jobs do not send alerts. If this occurs, set ALERT_SCHEDULE_ENABLED=false and then restart web and agent.
- If a license key value is damaged or deleted in a current app session you may experience job failures with license key errors.
    - If you know that your deployment of Collibra DQ has a valid license key, you can attempt to refresh the page, sign out of the app and then sign back in. This will sync your license key with the metastore, which will allow you to once again run jobs without license key expiration job failures.

DQ Cloud

- Instance Profile is not supported for S3 connections.
- When using the Completeness Report, data only appears after upgrading to 2023.06 or later.
- When using the Findings page, you currently cannot drill into a rule break record. While there is no workaround for this limitation, a fix is planned for the 2023.06 release.
- When using the Findings page, you currently cannot tag job runs as off-peak. This will be fixed in the 2023.07 release.

Pushdown

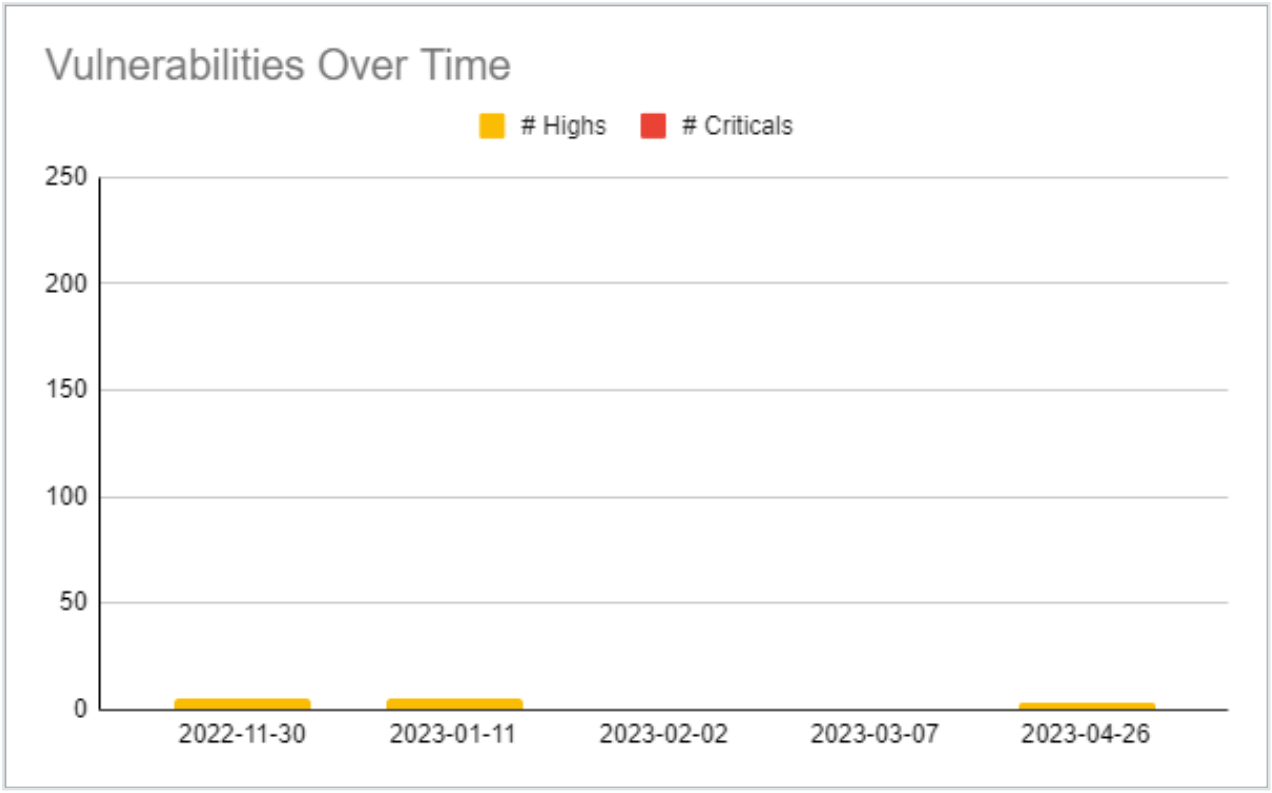
- When scanning for dupes, fuzzy match duplicates display incorrectly as "NULL" values on the Findings page. This will be fixed in the 2023.06 release.

DQ Connector

- When navigating the Admin Console menu with React turned on, an "Integrations" option leads to a non-functional page. This will be fixed in the 2023.06 release.

DQ Security Metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.



The following image shows a table of Collibra DQ security metrics arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-11-30	2022.12	0	0	0	5	2	JFrog Artifact
DQ	2023-01-11	2023.01	0	1	1	5	2	JFrog Artifact
DQ	2023-02-02	2023.02	0	0	0	0	0	JFrog Artifact
DQ	2023-03-07	2023.03	0	1	1	0	0	JFrog Artifact
DQ	2023-04-26	2023.05	0	1	1	0	1	JFrog Artifact

MUI Redesign

The following table shows the status of the MUI redesign of Collibra DQ pages as of this release. Because the status of these pages only reflects Collibra DQ's internal test environment and completed engineering work, pages marked as "Done" are not necessarily available externally. Full availability of the new MUI pages is planned for an upcoming release.

Page	Location	Status
Homepage	Homepage	✓ Done
Sidebar navigation	Sidebar navigation	✓ Done

Page	Location	Status
User Profile	User Profile	✓ Done
List View	Views	✓ Done
Assignments	Views	✓ Done
Pulse View	Views	✓ Done
Catalog by Column (Column Manager)	Catalog (Column Manager)	✓ Done
Dataset Manager	Dataset Manager	✓ Done
Alert Definition	Alerts	✓ Done
Alert Notification	Alerts	✓ Done
View Alerts	Alerts	✓ Done
Jobs	Jobs	✓ Done
Jobs Schedule	Jobs Schedule	✓ Done
Rule Definitions	Rules	✓ Done
Rule Summary	Rules	✓ Done
Rule Templates	Rules	✓ Done
Rule Workbench	Rules	✕ In Progress
Rule Builder	Rules	✕ In Progress
Data Classes	Rules	✓ Done
Explorer	Explorer	✕ In Progress

Page	Location	Status
Reports	Reports	✕ In Progress
Dataset Profile	Profile	✕ In Progress
Dataset Findings	Findings	✓ Done
Sign-in Page	Sign-in Page	✓ Done

Maintenance Updates

2023.05.1

- Fixed the builds for the 2023.05 Collibra DQ version. Previously, jobs failed with an "Unknown" status during the Alerts activity and showed a "NoClassDefFoundError" in the job log. (ticket #115956, 116052)

Release 2023.04

- Highlights
- New Features
- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics
- Maintenance Updates

Highlights

- We are excited to announce that Pushdown for Snowflake is now generally available! Pushdown is an alternative compute method for running a DQ Job, where Collibra DQ submits all of the Job's processing directly to a SQL data warehouse, such as Snowflake.

When all of your data resides in Snowflake, Pushdown reduces the amount of data transfer, eliminates egress latency, and removes the Spark compute requirement of a DQ Job.

- Six new out-of-the-box reports are now available:
  - User Activity shows user click activity in Collibra DQ.
  - Row Count by Dataset shows the number of rows in a given dataset.
  - Alert Details shows alert details and the method by which they are sent, for example email.
  - Job Runs per User shows the number of Jobs run per user over a certain period of time.
  - Daily Job Runs per User shows the number of Jobs run per user on a particular day.
  - Rule Updates per User shows when a user-created rule is inserted, updated, or deleted, and applied to the dataset by a user over a certain period of time.

> **Note**
> With the development of the newly designed Template Rules page in the upcoming the May release, we will use the newly built v2/createcustomrule API for custom rules and will deprecate the old v2/insertcustomrule.

> **Warning**
> When you enable **Include Links** on the Alerts Configuration page of the Admin Console, you must also enter your host address in the **HOST_NAME** input field on the Application Configuration page. If the HOST_NAME input field does not contain a host address when Include Links is enabled, your DQ Jobs will fail.

New Features

Reports

- The User Activity, Row Count, Alert Details, Job Runs per User, Daily Job Runs per User, and Rule Updates per User reports are now available.

Enhancements

Rules

- You can now view rows scanned and rows breaking into the rule_output table.

Reports

- You can now view and filter reports on column dimensions with the Column Dimension Dashboard. This dashboard provides an overview of all columns, datasets, and business units, and provides insights about total current DQ Dimension scores, Dimension scores over time, and DQ scores (along with other metadata) for each column.
    - This dashboard is the first version and design adjustments are planned for a future release.
    - We also plan to redesign the Reports page in an upcoming release to better list all charts and dashboards and make it easier for you to know what we offer and potential use cases for the reports and dashboards.
    - Contact the Collibra DQ Product team or your CSM to learn more.

Platform

- In order to further enable Kubernetes-deploying users to more easily configure certain parameters upon deployment, the following variables are now available as owl-web-configmap properties:
    - SAML_MAX_AUTH_AGE
    - SAML_ENABLED
    - SAML_ENTITY_ID
    - CORS_ALLOWED_ORIGINS
    - SAML_ENTITY_BASEURL
- With the MUI turned on, the DQ Score Cards tab is now called Views.

Pushdown

- Record change observations are now included on the Findings page for Pushdown Jobs.
- The functionality of the v3/rules/{source}/copy API to work across Pull-Up and Push-down datasets is now extended.

- You can now add a description of your dataset from the Review page before you create a Pushdown Job.

Fixes

Rules

- Fixed an issue with the Rule Builder that prevented rules from properly validating. (ticket #106100)
- Fixed an issue with the Rule Builder that prevented the selection of available columns and limited the ability to scroll through the list of options on the Primary Column drop-down list. (ticket #110276)
- Fixed an issue with the rule breaks table within the Breaks tab where sorting the table by the Rule Name column caused an error to throw. (ticket #105410)
- Fixed an issue where, when a rule had sub-select at the beginning of the WHERE clause, the rule displayed an Exception status when it ran. (ticket #109812)

Explorer

- Fixed an issue with the Transform option on Athena connections, the DATE() function was incorrectly included when the job was executed with an unlocked Run CMD. (ticket #105316)

Platform

- Fixed an issue where, while editing an existing job in Explorer, Collibra DQ became stuck in the loading stage. (ticket #105573)
- Fixed an issue that incorrectly allowed the creation of local users even with LOCAL_REGISTRATION_ENABLED set to `false`. (ticket #110325)
- Fixed an issue where a role-less user was able to access the Edit Rule modal from the Rule table on the Findings page. (ticket #104129)
- Fixed an issue where the install packages for Collibra DQ versions 2023.02 and 2023.03 did not contain the owlmanage.sh script. (ticket #110786)

Security

- Fixed a potential security issue where the introduction of a smart pre-admin approval message on the sign-in screen in the 2023.03 release version is now reverted to the previous sign-in error message.

APIs

- Fixed an issue the /v2/get-job-log API where the Job Log did not load and threw an error when more than two datasets were included in a rule. (ticket # 105498)

Connections

- Fixed an issue where too many open connections prevented new connections from being opened. (ticket #107034)
- Fixed an issue where a database connection remained idle and did not close when the /v2/getsqlresult API was called. (ticket #108365)

Source

- Fixed an issue where Redshift source queries failed when -profilePushdown count option was used. (ticket #106085)
- Fixed an issue with the Cell Threshold invalidation logic which caused the issue count to remain the same after the issue was invalidated and the job re-ran.

Known Limitations

Connections

- A limitation with Hive JDBC was observed where having both JDBC41 and JDBC42 drivers caused a conflict that prevents table data from loading. As a result, this release includes only the JDBC41 driver. This driver will be upgraded in a future release.
- Hive Direct job execution is currently not supported due to a technical issue that we plan to fix in an upcoming release.

Capabilities

- Sub-selecting columns of targets in the Source activity results in a failed Job when the Job runs.
    - A possible workaround is to keep the query as a select all (`select *`) operation.
- When you enable **Include Links** on the Alerts Configuration page of the Admin Console, you must also enter your host address in the **HOST_NAME** input field on the Application Configuration page.
    - A fix for this is planned for the May release.

Pushdown

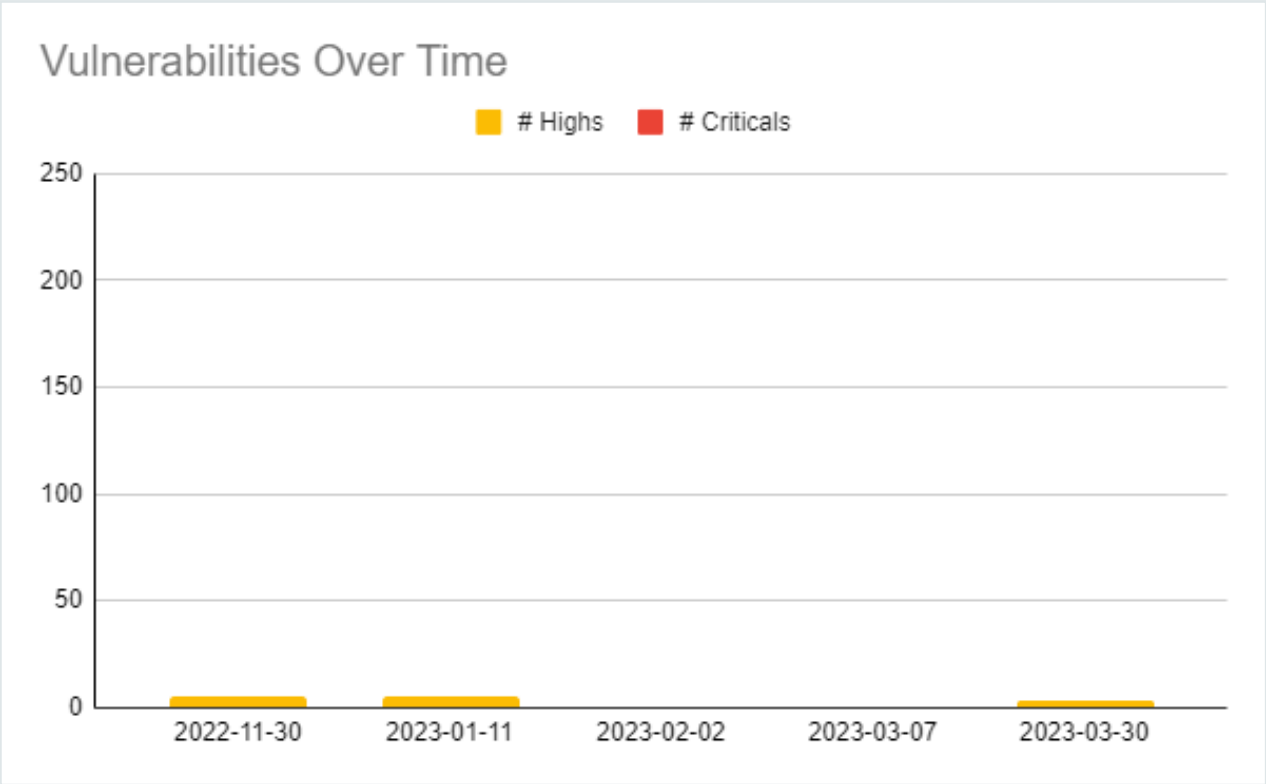
- Pushdown is currently only available for on-premises deployments of Collibra DQ.
    - Support for Pushdown in DQ Cloud is planned for a future release.

DQ Connector

- There is an issue where Rest API calls from Collibra DGC to Collibra DQ result in a 403 Forbidden Error because Collibra DGC passes a cookie header which causes Collibra DQ to reject the request due to recent updates to CSRF token requirements. This limits Collibra DQ's ability to connect to Collibra Platform.
    - A possible workaround until a fix is available is to update `export CSRF_TOKEN_ENABLED=false` in the owl-env.sh file for Standalone deployments or update `dq.security.csrf.token.enabled=${CSRF_TOKEN_ENABLED:false}` in the Web ConfigMap for Kubernetes deployments.

DQ Security Metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.



The following image shows a table of Collibra DQ security metrics arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolv	# Highs	# Mediums	Artifact(s)
DQ	2022-11-30	2022.12	0	0	0	5	2	JFrog Artifact
DQ	2023-01-11	2023.01	0	1	1	5	2	JFrog Artifact
DQ	2023-02-02	2023.02	0	0	0	0	0	JFrog Artifact
DQ	2023-03-07	2023.03	0	1	1	0	0	JFrog Artifact
DQ	2023-03-30	2023.04	0	1	1	3	2	JFrog Artifact

MUI Redesign

The following table shows the status of the MUI redesign of Collibra DQ pages as of this release. Because the status of these pages only reflects Collibra DQ's internal test environment and completed engineering work, pages marked as "Done" are not necessarily available externally. Full availability of the new MUI pages is planned for an upcoming release.

Page	Location	Status
Homepage	Homepage	✓ Done
Sidebar navigation	Sidebar navigation	✓ Done

Page	Location	Status
User Profile	User Profile	✓ Done
List View	Views	✓ Done
Assignments	Views	✓ Done
Pulse View	Views	✓ Done
Catalog by Column (Column Manager)	Catalog (Column Manager)	✓ Done
Dataset Manager	Dataset Manager	✓ Done
Alert Definition	Alerts	✓ Done
Alert Notification	Alerts	✓ Done
View Alerts	Alerts	✓ Done
Jobs	Jobs	✓ Done
Jobs Schedule	Jobs Schedule	✓ Done
Rule Definitions	Rules	✓ Done
Rule Summary	Rules	✓ Done
Rule Templates	Rules	✕ In Progress
Rule Workbench	Rules	✕ In Progress
Rule Builder	Rules	✕ In Progress
Data Classes	Rules	✕ To Do

Page	Location	Status
Explorer	Explorer	✕ In Progress
Reports	Reports	✕ In Progress
Dataset Profile	Profile	✕ In Progress
Dataset Findings	Findings	✓ Done
Sign-in Page	Sign-in Page	✓ Done

Maintenance updates

2023.04.1

- External users can now successfully run Jobs that include alert links.
    - The email address of the owner of a Job with alert links is no longer included in alert emails for both local and external users.

2023.04.2

- External users can now successfully run Jobs that include alert links.
    - The email address of the owner of a Job with alert links is no longer included in alert emails for both local and external users.
- Fixed an issue with Collibra DQ on AWS Cloud where Instance Profile authentication caused scheduled S3 Jobs to fail.

Release 2023.03

- Highlights
- New Features

- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics
- Maintenance Updates

Highlights

- When your Collibra DQ license is about to expire, a counter in a warning banner on the Collibra DQ home page now counts down from 30 days until your license expires. You can update your license key at any time to avoid any disruptions.
    - A message now appears in the Description field of the Jobs activity logs and the Job Exception field of the Jobs log table view when a DQ Job fails because of an expired or invalid license key.
    - An error message now appears on the Findings page when a job fails because of an expired or invalid license key.
    - When admin users update a license with a valid license key on the License page of the Admin Console, a success message appears.
    - When admin users update a license with an invalid license key on the License page of the Admin Console, an error message appears.
        - DQ Jobs will again run normally once a valid license key is saved.

> **Warning**
> Any DQ Jobs that run on accounts with expired or invalid license keys will no longer run until a valid license key is saved.

- To improve how Collibra DQ handles Jobs concurrently at lower memory agent configurations, the default agent batch size for concurrent Jobs is now 5. For example, when you run more than 5 Jobs concurrently, only 5 Jobs run at once. All additional Jobs remain in Staged status until the initial 5 Jobs are complete, which allows new Jobs to continue to enter the queue on the Jobs page.
    - The following are the default values in the Helm owl-agent-configmap.yaml file:
        - `DEFAULT_BATCH_ID: <base64-encoded-string>`
        - `DEFAULT_BATCH_SIZE: 5`
    - The following are the default values in the Helm values.yaml and values_test.yaml files:

        - `agent_batch_id: "default?batchSize=5"`
        - `agent_batch_size: 5`
    - It may be necessary to override the default values for high concurrency requirements. Increase the `DEFAULT_BATCH_ID` and `DEFAULT_BATCH_SIZE` values and update the corresponding Helm values for DQ Agent configuration accordingly.
- You can now convert log outputs to JSON formatting for ingestion by a third-party log consumer.
- You can now optionally include additional fields and hyperlinks to the environments from which alerts originate in email alerts.
    - Admin users can turn this option on or off from the Alerts page within the Admin Console.

> **Note**
> If you only install owlweb or owlagent, you need to copy the owlmanage.sh file from the installation package into the bin directory with the following command:
>
> ```
> cp $PACKAGE_LOCATION/owlmanage.sh $INSTALL_
> PATH/bin/owlmanage.sh
> ```

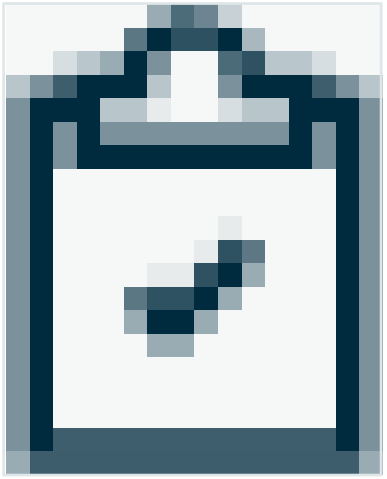
New Features

Platform

- Collibra DQ now displays in-app messages about expired, expiring, and invalid licenses, including a banner message on the Collibra DQ home screen that counts down from 30 days until your license expires.
    - You can update or edit your license from the License Management page in the Admin Console.

Enhancements

Jobs

- The icon for the Jobs page is now the  clipboard icon.

Rules

- Export LinkIds now has the same export capabilities as Export and Export with Details.
- You can now use keyboard shortcuts on the Rule Builder page to display dataset-level stat rules.

Alerts

- You can now optionally include additional fields and hyperlinks to the environments from which alerts originate in email alerts.

Agent

- To improve how Collibra DQ handles Jobs concurrently at lower memory agent configurations, the default agent batch size for concurrent jobs is now 5.

Admin

- You can now set default values on the Add External Service Configuration form on the Assignment Queues page when you want to override the values for individual finding

validate action.

Security

- Collibra DQ now enforces a Spring CSRF token, as well as CORS and SameSite, as standards to mitigate against CSRF attacks.
    - Any cookie-based auth requests without this token are rejected with a 401 error.
    - This feature is active by default, but can be deactivated if necessary. Deactivating this feature does not compromise the security of the Collibra DQ app because we have other mitigation standards in place for CSRF.
    - All non-GET HTTP requests need to carry a CSRF token header when they call the API. This happens automatically when you use the app within Collibra DQ.
    - This implementation only impacts cookie-based authentication. Additionally, Swagger UI and APIs that use bearer token authentication are not impacted by this change.
- Server-side validation for username, first name, last name, and email is now included in the /v2/createlocaljdbcuser internal registration API.

Platform

- New User Request emails now include additional fields and hyperlinks to the environment from which the request originates for easier access.
- You can now convert log outputs to JSON formatting for ingestion by a third-party log consumer.

Snowflake Pushdown (in preview)

- You can now scan for exact and fuzzy matches as part of the Dupes check layer.
- You can now use the Correlation tab on the Profile page to understand correlation levels between numerical columns.
- Minimum and maximum string length profiling is now available.

Fixes

Explorer

- Fixed an issue where a custom Transform statement was corrupted after running a DQ Job and clicking the Edit button. (ticket #104992)

Rules

- Fixed an issue where custom rules with primary and secondary datasets that shared similar naming conventions failed because of incorrect replacement of @Dataset or @secondary with the name of the dataset by Collibra DQ's replace logic. (tickets #105812, 108490)

Source

- Fixed an issue that resulted in an Authentication error when the source data loaded because both the source and target used the same Kerberos TGT-based connection. (ticket #102639)

Platform

- Fixed an issue where incorrect syntax in the owldq/templates/rbac.yaml manifest caused errors for some Kubernetes deployments on AKS.
- Fixed an issue where an upgrade to Release 2023.01 caused errors, which referred to the agent_displayname column on the agent_q table when the web pods launched.
- Fixed an issue where the Flyway script V188 did not run successfully after an upgrade from Release Version 2022.08 to 2023.01. (ticket #107222)

Connections

- Fixed an issue that prevented Collibra DQ from establishing a connection to Azure Data Lake after an upgrade to Release Version 2022.11. (tickets #104271, 107451)

Outliers

- Fixed an issue where jobs that used union lookback and whose findings were marked as ignored, off-peak, or removed, were incorrectly included in the outlier calculation. (ticket #104746)

Known Limitations

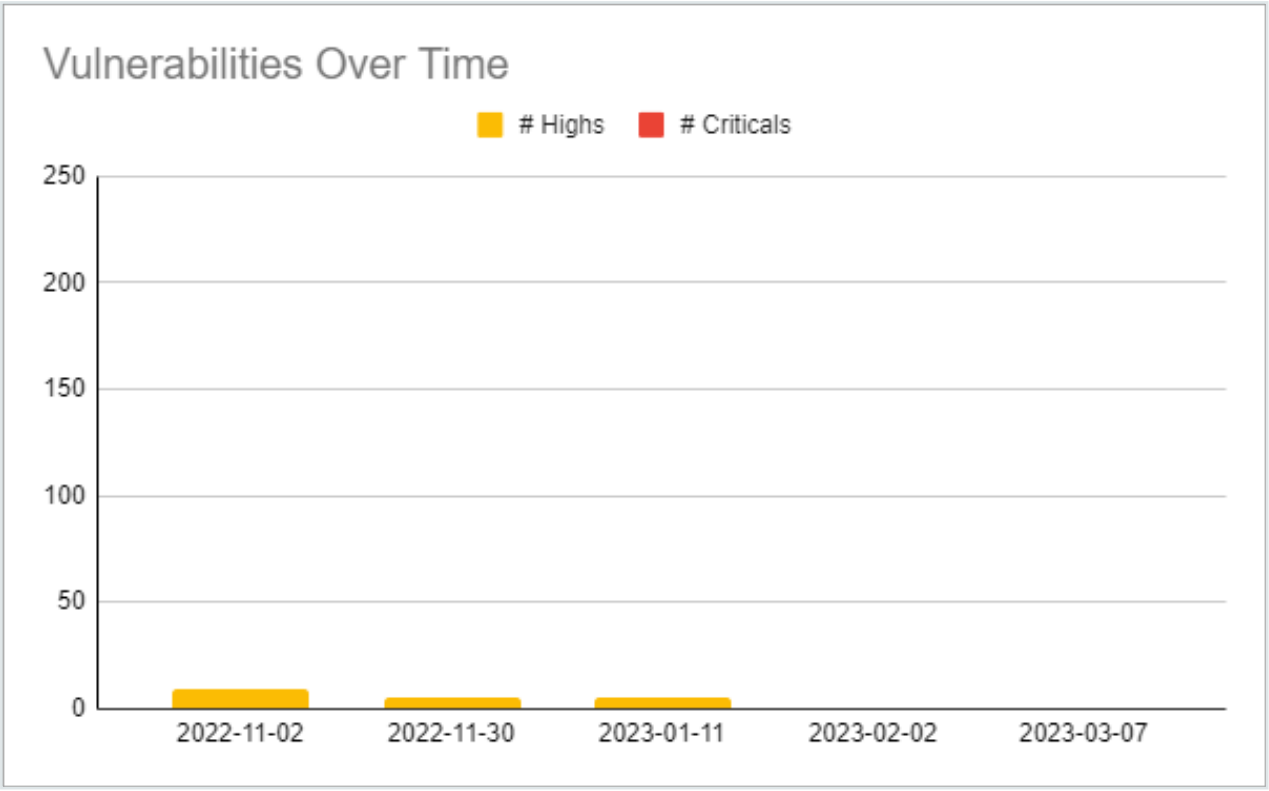
Snowflake Pushdown (in preview)

- Mixed datatype rules are not supported in Pushdown.

DQ Connector

- There is an issue where Rest API calls from Collibra DGC to Collibra DQ result in a 403 Forbidden Error because Collibra DGC passes a cookie header which causes Collibra DQ to reject the request due to recent updates to CSRF token requirements. This limits Collibra DQ's ability to connect to Collibra Platform.
    - A possible workaround until a fix is available is to update `export CSRF_TOKEN_ENABLED=false` in the owl-env.sh file for Standalone deployments or update `dq.security.csrf.token.enabled=${CSRF_TOKEN_ENABLED:false}` in the Web ConfigMap for Kubernetes deployments.

DQ Security Metrics



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-11-02	2022.11	0	0	0	9	6	JFrog Artifact
DQ	2022-11-30	2022.12	0	0	0	5	2	JFrog Artifact
DQ	2023-01-11	2023.01	0	1	1	5	2	JFrog Artifact
DQ	2023-02-02	2023.02	0	0	0	0	0	JFrog Artifact
DQ	2023-03-07	2023.03	0	1	1	0	0	JFrog Artifact

Maintenance updates

- External users can now successfully run Jobs that include alert links.
    - The email address of the owner of a Job with alert links is no longer included in alert emails for both local and external users.

Release 2023.02

- Highlights
- New Features

- Enhancements
- Fixes
- Known Limitations
- DQ Security Metrics

Highlights

- To enhance user experience, the React MUI is now on by default across the Collibra DQ app. You can restore the legacy experience by following the steps on the React page.
- You can now use the new /v3/rules/{source}/copy API to bulk copy all rules from an existing dataset to another target dataset. With this new API, you can input a source dataset from which to copy rules, then choose one or more target datasets to which your rules copy.
- You can now leverage the Spring Framework to access controller-level logs and debug API calls.
- The new Oversized Job Report is now available by default in the Reports section. The Oversized Job Report lets you review usage analytics for jobs that use more hardware than what is required for their workloads.
- You can now view and edit your Agent Display Name from the Agent Management screen. This lets you customize your Agent Display Name to make it easier to identify your agent throughout Collibra DQ.
- When running large DQ Spark jobs, you can now bring in on-demand Persistent Volume Claims to use as Spark scratch disk space on Kubernetes deployments.

> **Note**
> If you only install owlweb or owlagent, you need to copy the owlmanage.sh file from the installation package into the bin directory with the following command:
>
> ```
> cp $PACKAGE_LOCATION/owlmanage.sh $INSTALL_
> PATH/bin/owlmanage.sh
> ```

New Features

Reports

- The new Oversized Job Report is now available by default.

Agent

- You can now view and edit your Agent Display Name from the Agent Management screen.
- When running large DQ Spark jobs, you can now bring in on-demand Persistent Volume Claims to use as Spark scratch disk space on Kubernetes deployments.

APIs

- You can now use the new /v3/rules/{source}/copy API to bulk copy all rules from an existing dataset to another target dataset.
- You can now leverage the Spring Framework to access controller-level logs and debug API calls.

Enhancements

Rules

- Native rules with invalid SQL queries now display an exception message instead of incorrectly displaying a passing status.
- You can now drill into a pulse view chart on the $daysWithoutData stat rules on the Findings page.

Platform

- React MUI is now on by default.
- To ensure proper compliance with Collibra's security protocol, new installations now require you to set up a password that meets Collibra's password policy and a default email address associated with the admin user account.
    - Instructions for providing these credentials are interactive as part of the setup script for Standalone installs.
    - For new installs in non-interactive mode, such as automated deployment, export the following variables, as they are required for the script to execute correctly:
        - `export DQ_ADMIN_USER_PASSWORD`
        - `export DQ_ADMIN_USER_EMAIL`

    - For Cloud native installs, you must use the following Helm variables to set the password and default admin email:
        - `--set global.web.admin.email=${email} \`
        - `--set global.web.admin.password=${password} \`
    - For multi-tenant instances, admin users are created via the UI, not with this process.

Security

- To ensure proper compliance with Collibra's security protocol, all deployments of Collibra DQ on Kubernetes now use a read-only root filesystem.
    - All ephemeral storage directories, such as /opt/owl/log, /tmp/scratch, and /opt/owl/cache, are still writable for Collibra DQ to function correctly.
    - The read-only root filesystem is enforced on all long-running pods except for Spark jobs and Livy sessions.
    - Ensure that you pick the corresponding Helm charts for a successful upgrade.
        - Any customizations must be ported to the corresponding Helm charts. For additional assistance, please contact Collibra Professional Services.
        - You can no longer upload custom JDBC drivers under the /opt/owl/drivers folder. Refer to the documentation for more details on how to configure custom JDBC drivers.

> **Note** Standalone deployments of Collibra DQ are not affected by these updates.

Snowflake Pushdown (in preview)

- You can now scan for categorical outliers.
- You can now apply weighting and boundaries as part of the Outlier layer.

Fixes

Rules

- Fixed an issue where regex rules, which worked in previous DQ versions, appeared as exceptions after upgrading to the 2022.09 version. (ticket #102615)

Dupes

- Fixed an issue that prevented jobs from running with Dupes check switched on for tables with only one column. (ticket #104585)

Platform

- Fixed an issue with the print statement in the stage 3 logs where an excessive number of log records were written due to too many observations. (ticket #104081)

Security

- Fixed an issue with role mapping where invalid values for `LDAP_GROUP_RESULT_DN_ATTRIBUTE` and `LDAP_GROUP_RESULT_CONTAINER_BASE` set to the base OU of the AD groups caused the role mapping to fail.

Connections

- Fixed an issue with connections using Kerberos authentication that prevented jobs from running because of IT restrictions around writing to Openshift containers. (ticket #102964)
- Fixed an issue with Databricks cluster connections to allow `PWD={}` input in the connection string URL. (ticket #104845)
    - You are still required to enter `PWD={your-text-here}` in the URL.
    - This fix only applies to Databricks cluster connections, not to Databricks connections that use SQL-endpoints.
- The logging of entire ServiceNow response messages has been added to help identify errors.

Agent

- Fixed an issue where associating or disassociating connections from a remote Agent resulted in a PreparedStatementCallback error, which prevented connections from being deleted. (tickets #102322, 102763)

Snowflake Pushdown (in preview)

- Fixed an issue with Min and Max on the Profile page that caused strings with numerical values to display invalid results. (ticket #105935)

Known Limitations

Rules

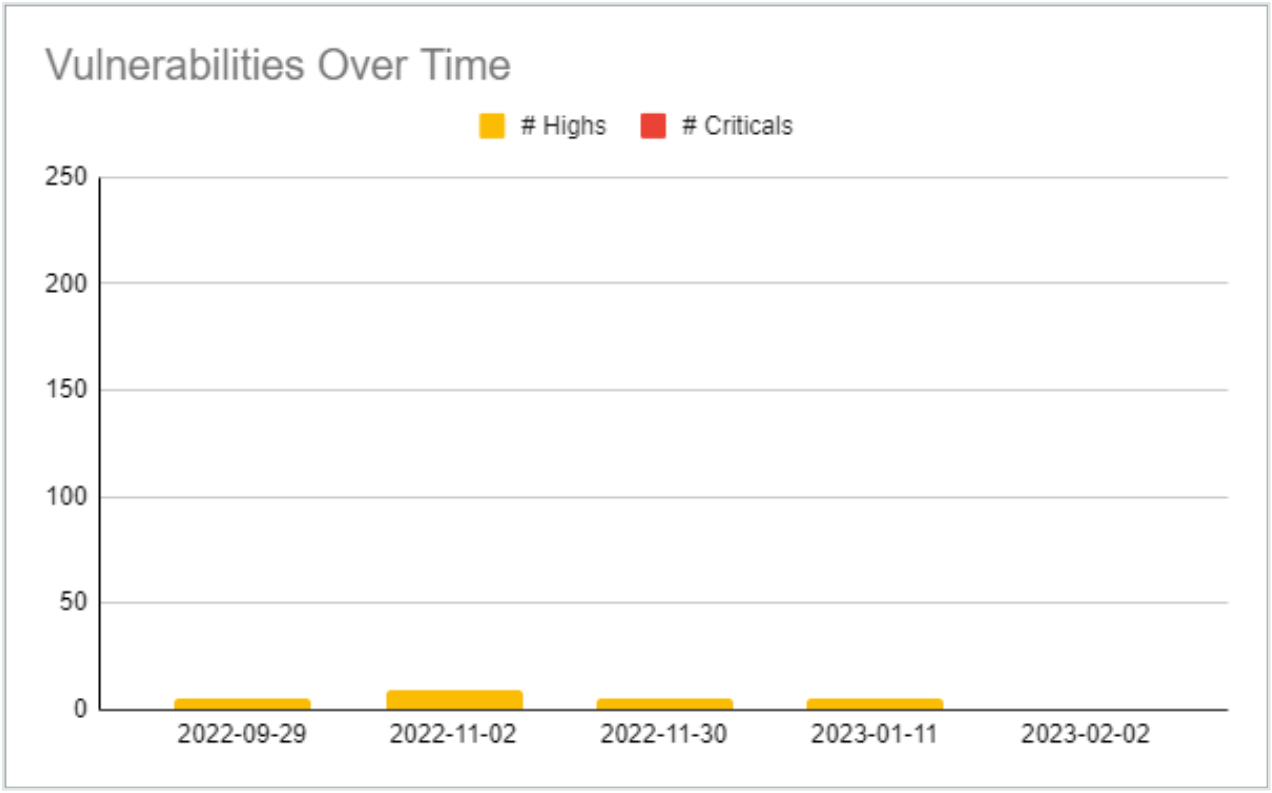
- The $daysWithoutData stat rule has a limitation where rules that are not named "daysWithoutData" do not display their Pulse View charts when you drill into the rule on the Findings page. A workaround for this is to name your stat rule exactly as it appears here: `daysWithoutData`.

Platform

The dollar sign `$` is a special character that must be escaped with a backslash `\` when running the setup script or Helm via the command line.

DQ Security Metrics

The following image shows a chart of Collibra DQ security vulnerabilities arranged by release version.



The following image shows a table of Collibra DQ security metrics arranged by release version.

Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-09-29	2022.10	0	0	0	5	13	JFrog Artifact
DQ	2022-11-02	2022.11	0	0	0	9	6	JFrog Artifact
DQ	2022-11-30	2022.12	0	0	0	5	2	JFrog Artifact
DQ	2023-01-11	2023.01	0	1	1	5	2	JFrog Artifact
DQ	2023-02-02	2023.02	0	0	0	0	0	JFrog Artifact

Release 2023.01

New Features

Admin

- Admin users can now view and modify the DQ license key and license name from the new License page.

Platform

- You can now deploy new Helm charts on new and existing releases of Collibra DQ Cloud Native deployments without encountering character size limitations.

Reports

- Four new reports are now available by default:
  - The Missing Jobs Report shows jobs that were expected to run but didn't run as scheduled.
  - The Hardware Usage Report shows the datasets that required the most total cores to run and more general hardware usage statistics.
  - The Observability Score Roll-Up Report shows the aggregated scores of all AdaptiveRules (all datasets + all columns) and averages passing and breaking for all columns over 30 days.
  - The Rules Passing Fraction Roll-Up Report shows all the passing rows and total rows scanned for user-defined rules aggregated by dimensions over the past 30 days.

Enhancements

Rules

- The Export LinkIds button is now displayed within the Rules tab of the Findings page. This button was previously only available under the Breaks tab on the Rule Builder page.
  - You can enable LinkIDs from the Scope workflow on the Explorer page to export LinkIds for rule break records.
- The Copy Rules API (/v3/rules/copy) now has the following enhancements:
  - Copied rules now copy to a new dataset correctly regardless of their rule type.
  - Security logs for every rule copy request are now available for admin users to review in the Audit Trail section of the Admin Console.

Explorer

- When creating a DQ Job for a table with no rows, the columns are now shown in the Scope section.

Platform

- A new property, `LOCAL_REGISTRATION_ENABLED`, in the owl-env.sh script and K8s config map is now available to display or hide the registration link on the Sign in page for local users.
    - For owl-env.sh:
        - The command `export LOCAL_REGISTRATION_ENABLED=true` allows the registration link to display on the Sign in page.

          > **Note** Because the registration link is visible by default, this property is also set to true by default.

        - The command `export LOCAL_REGISTRATION_ENABLED=false` hides the registration link from the Sign in page.
    - For K8s:
        - The configuration `LOCAL_REGISTRATION_ENABLED:"true"` allows the registration link to display on the Sign in page.

          > **Note** Because the registration link is visible by default, this property is also set to true by default.

        - The configuration `LOCAL_REGISTRATION_ENABLED:"false"` hides the registration link from the Sign in page.

Connections

- SPARK322 and SPARK320 are now shipped with a Spark JDBC connection provider for Standalone deployments.
- When you bring Collibra DQ jars into Databricks, you are now required to set the property `spark.sql.-sources.disabledJdbcConnProviderList='basic,oracle,mssql'` at either the Spark Cluster-level or the SparkSession-level before using Collibra DQ's set of functions for Spark profiles 3.2.1 and onwards.

Admin

- You can now sort the Date column on the Usage page. Dates now appear in descending order by default.

Fixes

Rules

- Fixed an issue that prevented the descriptions of saved rules from being edited when the rule name contained a greater than symbol after a single quote. (ticket #100114)
- Fixed an issue with rule builder validation that caused a rule syntax exception message to throw. (tickets #99735, 101165)
- Fixed an issue with rules with complex conditions (multiple rlike strings) for Freeform SQL rules that resulted in an exception message. (ticket #100116)

DQ Job

- Fixed an issue where behavioral observations made for a dataset did not subtract points from the data quality score. (ticket #98539)

Alerts

- Fixed an issue that limited the ability to edit or delete alerts with names containing apostrophes from the UI. (ticket #98864)

Outliers

- Fixed an issue where recalibrating an outlier would bulk apply downtrain labeling to it. (ticket #100085)

Reports

- Fixed an issue where Completeness Reports were not generated when the Custom Range filter was used. (ticket #99786)

DQ Connector

- Fixed an issue with the Collibra DQ - Collibra Platform integration that prevented Rules and Charts from importing. (ticket #104872)

Admin

- Fixed an issue with time-based data retention when using linkId that caused too many break records to store in the metastore. (ticket #99072, 102900)

Known Limitations

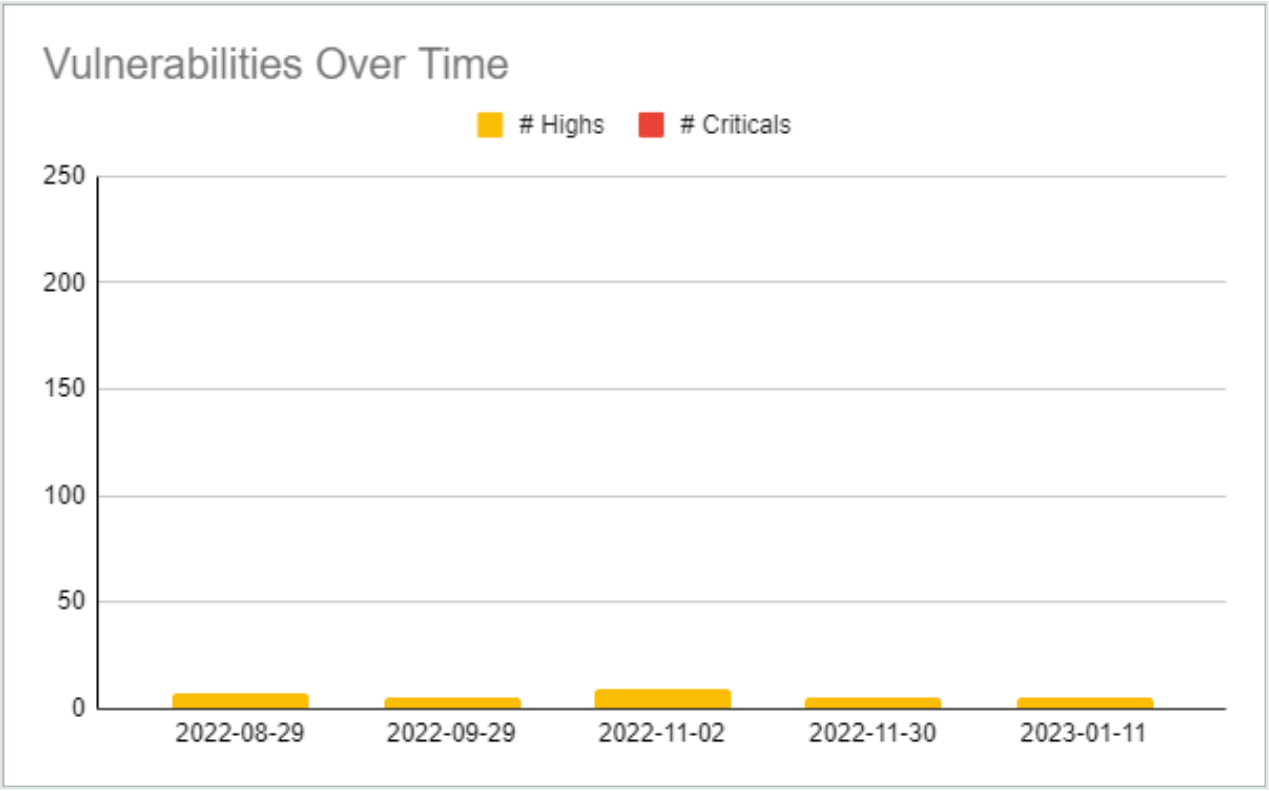
Rules

- The new Export LinkIds button generates a CSV file limited to viewing only via a spreadsheet program, like Excel.
    - A workaround is to Save/Export the CSV file from the spreadsheet program in order to allow viewing in general text editors.

DQ Job

- Remote file jobs with headers containing white spaces fail with a requirement failed exception message.
    - A workaround is to edit the DQ Job command line in the Run CMD tab and place single quotes around the column name in `-q` and double quotes around the entire `-header` flag.

DQ Security Metrics



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-08-29	2022.09	0	0	0	7	12	JFrog Artifact
DQ	2022-09-29	2022.10	0	0	0	5	13	JFrog Artifact
DQ	2022-11-02	2022.11	0	0	0	9	6	JFrog Artifact
DQ	2022-11-30	2022.12	0	0	0	5	2	JFrog Artifact
DQ	2023-01-11	2023.01	0	1	1	5	2	JFrog Artifact

Hotfixes

Data Quality & Observability Classic 2023.01.1

- Fixed an issue with Databricks cluster connections to allow `PWD={}` input in the connection string URL. (ticket #104845)
    - You are still required to enter `PWD={your-text-here}` in the URL.
    - This fix only applies to Databricks cluster connections, not to Databricks connections that use SQL-endpoints.

Release 2022.12

New Features

Explorer

- The types of queries that can run from the View Data page are now restricted to read-only queries only.

APIs

- You can now copy SQLG- and SQLF-type rules from an existing dataset to another existing dataset with the /v3/rules/copy API call.

Connections

- You can now create a MongoDB connection with a CDATA driver.

Snowflake Pushdown (in preview)

- You can now detect outliers when running a Pushdown job.

Enhancements

DQ Job

- All tables on the Jobs page now include pagination, dropdown filters, and the ability to export.

Rules

- Rules associated with datasets with zero rows now execute successfully.
- Stat rule evaluation on secondary datasets is now supported for SQLF rules.

Profile

- You can now view run execution details and stale data by toggling the box chart on the findings page.

APIs

- The getRecords notebook API function is now updated and the getGeneric query is renamed getDupesPreview.
- You can now obtain rules from a dataset and reassign them to another dataset with the following Databricks notebook API functions:
    - `def addRules(rules: List[Rule], dataset: String): Owl`
    - `def getRulesDfByDataset(dataset: String): DataFrame`
    - `def getRulesByDataset(dataset: String): List[Rule]`
    - `def getRuleNamesByDataset(dataset: String): DataFrame`

Platform

- TechPreview (TP) labels are now removed from the UI.

Connections

> **Warning** As of September 2022, Databricks JDBC driver version 2.6.27 is packaged as part of both standalone and Kubernetes download packages. The Databricks Simba driver (version 2.6.22) is no longer packaged for Kubernetes. As a result of this change, the Databricks connection template has changed, and any existing connection using the old driver (2.6.22) must be updated. For more information on updating your drivers, refer to Standalone Upgrade.

- The Databricks SQL endpoint is now supported for JDBC connections.
- 3DES and DES encryption cipher for Kerberos authentication types are no longer supported because of recent Red Hat OS (RHEL 8.7) cipher deprecation.

Fixes

Explorer

- Fixed an issue with the `-rdEnd` variable in the command line the variable in the query to be improperly escaped. (ticket #98702)

Profile

- Fixed an issue where the confidence score (Conf) displayed values greater than the threshold of 100. (ticket #99636)
- Fixed an issue where HTML in data fields was rendered on the Data Preview section of the findings page. (ticket #97883)
- Fixed an issue with Data Preview that resulted in an OOM error when the data_preview table contained a large number of records.

Rules

- Fixed an issue where values on the Rules tab did not correctly display in scientific notation format. (ticket #89738)
- Fixed an issue when using a secondary dataset that prevented @dataset for primary dataset from being supported.

Scorecards

- Profile is now removed from the DQ Scorecards submenu.

Security

- Fixed an issue with LDAP external groups to role mappings when there was no fully qualified path for the LDAP group that caused malformed API calls and did not save properly.
- Fixed an issue with the Dataset Security feature. (ticket #100317)
    - When the following security settings are configured, the system fully restricts access to the findings page for admin users:
        - Dataset security is turned on.
        - Default owner access is unchecked.

- Dataset belongs to no roles or no roles to which the user has access.

APIs

- Fixed an issue with the /v2/gethints endpoint that prevented the Hints table from displaying correctly on the findings page. (ticket #98941)
- Fixed an issue with the getRecords and getGenerics APIs that prevented any information from being returned. (ticket #98820)

Alerts

- Fixed an issue with the SQL on the Alert Notifications page that prevented data from appearing in the DataTables error message.

Agent

- Fixed an issue with GKP deployments where job scans failed because the driver pod could not create connections to the metastore. (ticket #102175)

Validate Source

- Fixed an issue where the Source to Target scorecard incorrectly displayed a mismatch because of an unexpected column type checked during a schema order check. (ticket #98300)

Connections

- Resolved connection issues in certain cases by upgrading the Athena driver to version 2.0.33. (ticket #100340)
- Fixed an issue where HDFS connections could not rerun a job successfully because certain parameters were automatically appended to the Free Form (Appended) field of the Agent configuration. (ticket #95810)
- Fixed an issue with Dremio connection timeouts on Kubernetes deployments. (ticket #101221)
    - To prevent Dremio connection issues, set the following value in the Free Form (Appended) field of the Agent configuration:
      `-conf spark.driver.extraJavaOptions=-Dcd-jd.io.netty.tryReflectionSetAccessible=true`

Known Limitations

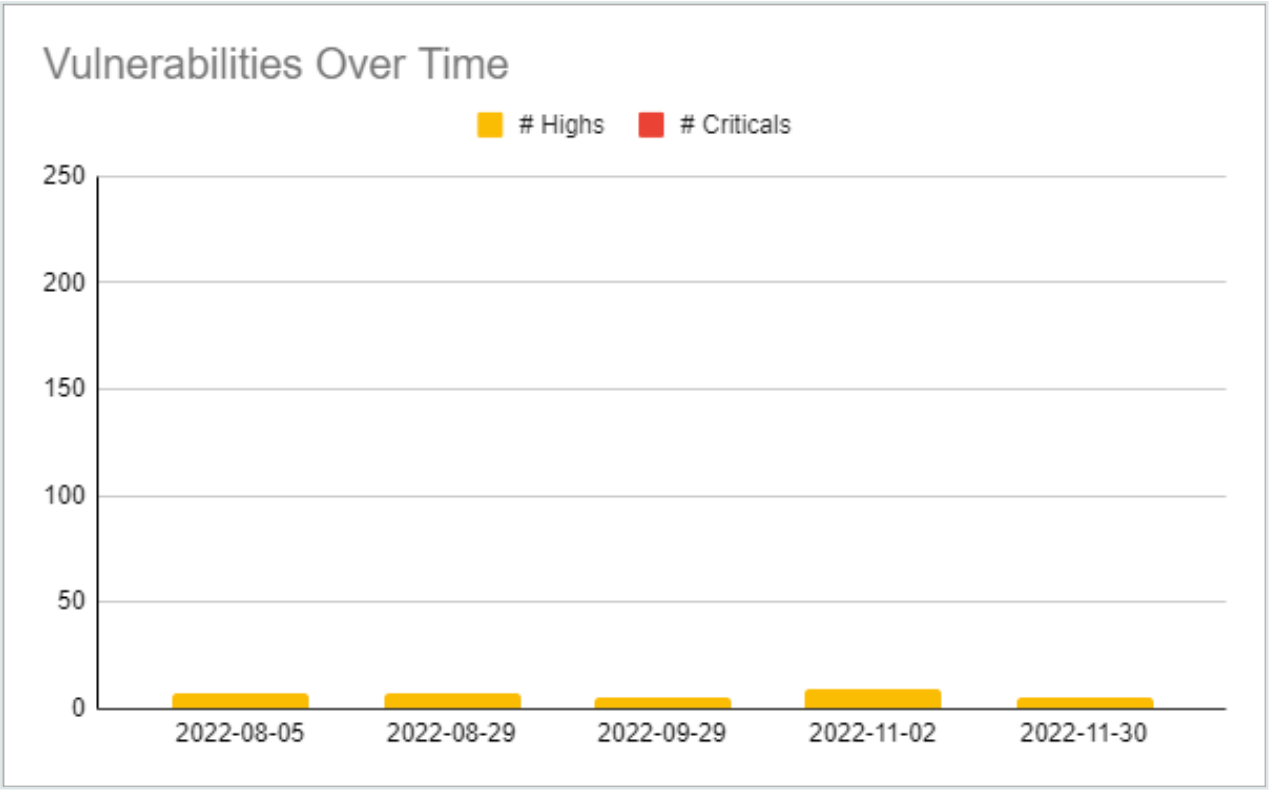
Rules

- Freeform rules with fully qualified column names are currently unsupported when they use the following syntax:
  `select <column name> FROM @<dataset name> WHERE @<dataset name>.<column name> condition`
    - A workaround to this limitation is to use aliasing instead.

APIs

- When using the new /v3/rules/copy API, the copied rule automatically appends "copied" to the rule name. After copying a rule, you may need to manually update the rule name.
    - If the copied rule is performed on a target dataset that does not conform to the compatible columns, then you need to manually update the rule to ensure the columns are compatible across datasets.
    - Dataset Security is not enforced when using the /v3/rules/copy API.

DQ Security Metrics



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-08-05	2022.08	0	0	0	7	14	JFrog Artifact
DQ	2022-08-29	2022.09	0	0	0	7	12	JFrog Artifact
DQ	2022-09-29	2022.10	0	0	0	5	13	JFrog Artifact
DQ	2022-11-02	2022.11	0	0	0	9	6	JFrog Artifact
DQ	2022-11-30	2022.12	0	0	0	5	2	JFrog Artifact

Hotfixes

Data Quality & Observability Classic 2022.12.3

- Fixed an issue with Databricks cluster connections to allow `PWD={}` input in the connection string URL. (ticket #104845)
    - You are still required to enter `PWD={your-text-here}` in the URL.
    - This fix only applies to Databricks cluster connections, not to Databricks connections that use SQL-endpoints.

Release 2022.11

Warning
The MS SQL driver that comes with JDK11 standalone packages does not currently work in the JDK11 environment. MSSQL requires a separate JAR for JDK11. Please contact your Customer Success Manager for the compatible driver.

Dremio is not currently supported for JDK11 standalone packages. If you plan to run JDK11, add `-Dcdjd.io.netty.tryReflectionSetAccessible=true` to owlmanage.sh as a JVM option for your Web and Spark instances. Please contact your Customer Success Manager for assistance.

Dremio jobs currently fail on both K8s and standalone JDK11 deployments. Add the following config to the Free Form (Appended) field of the Agent Configuration template: `-conf spark.driver.extraJavaOptions=-Dcdjd.io.netty.tryReflectionSetAccessible=true`.

As of October 18, 2022, all images for the 2022.10 release have a Critical CVE (CVE-2022-42889). If you picked up the 2022.10 release before October 18, 2022, there should be no issue with your scans. If issues persist, please contact your Customer Success Manager for a new build.

Note
After you complete an upgrade or a new installation of Collibra DQ, you are now required to enter a license name by following either a one-time prompt on the login page, entering the `LICENSE_NAME` environment variable in the environment variable file (owl-env.sh), or by entering the `global.configMap.data.license_name` Helm chart variable. Your license name is the value after `YOUR NAME IS =` found in the license provision email sent to you by Collibra. Customers who do not have this information due to being issued a license before March 2022 should input license information following the format below.

For a single instance: <yourcompanyname>
For multiple instances: <yourcompanyname>-dev, <yourcompanyname>-test, <yourcompanyname>-prod
No spaces or special characters are permitted except for hyphens –.

> **Note** Before you execute the setup.sh script, update the `SPARK_PACKAGE` variable to the desired Spark package, for example `spark-3.2.2-bin-hadoop3.2.tgz`. You can check the `../packages` folder for the Spark package that was previously downloaded. You do not need to update the setup.sh script if you use the following export statement:
> `export SPARK_PACKAGE="spark-3.2.2-bin-hadoop3.2.tgz"`

New Features

Platform

- The following pages now support the new React MUI:
  - Scorecards
  - List View
  - Assignments
  - Pulse View
  - Alerts

> **Note** React is turned off by default for the 2022.11 release. If you would like to try the new React pages, you can toggle it on from the Admin Console, or contact your Customer Success Manager for assistance.

DQ Job

- You can now terminate jobs from the Jobs page if they are in progress, incorrectly submitted, or stuck in Staged status. When you terminate a job, two alerts are generated.
  - Jobs in the Spark UI display Finished statuses, even though they are terminated from the DQ UI.

Alerts

- You can now generate alerts for the following stale data stat rules:
  - `$daysWithoutData`
  - `$runsWithoutData`
  - `$daysSinceLastRun`
- You can now generate alerts for jobs stuck in Staged status for more than one hour.

Admin

- You can now configure LDAP for user access in multi-tenant environments.

Connections

- You can now use key-pair authentication for Snowflake connections.
    - When you append to the Connection URL string, your entry must be comma separated.
    - When you manually modify the Driver Properties field, your entry must be semicolon separated.
- CDATA connections are now supported in standalone deployments.
    - CDATA drivers are now included in the release package.

Cloud Storage

- Azure Blob Storage is now a supported target storage system.

Snowflake Pushdown (in preview)

- Schema Change monitoring from the AdaptiveRules tab is now enabled by default.
    - Schema is now separated from basic profiling.
- The new DatasetDefDTO API now returns Pushdown information.
- Dataset security checks are now implemented for Pushdown jobs.

Enhancements

Explorer

- The Job Estimate dialogue now has improved guidance on executors and cores. The Job Estimate now estimates when a max core, max executor, and max memory is reached.

DQ Job

- Job schedule time zone is now a read-only field and can no longer be configured. Existing scheduled jobs reflect their current settings, but all other scheduled jobs are now based on the time zone of the DQ server (UTC). (ticket #88797, 89736, 92611, 95231)

Dupes

- A new warning message now displays when increasing the duplicate check limit from the UI. (ticket #95604)

Security

- Kubernetes service accounts associated with AWS IAM pod roles for controlling access to AWS services for cloud native DQ deployments on AWS EKS are now supported.
- When DATASET SECURITY is enabled, DATASET ACCESS is now required to edit, map, or retrieve datasets or business units. (ticket #92934)

Fixes

Rules

- Fixed an issue that prevented freeform rules containing double backslashes from saving. (ticket #96636, 96640)
- Fixed an issue that caused rules containing open brackets ([ ) to display break records incorrectly. (ticket #94399)
- Fixed an issue that caused rules containing regex to throw out of range exceptions. (ticket #98435)

DQ Job

- Fixed an issue where run time was not displayed on the findings page because run_id column type in the metastore did not include time zone. (ticket #96050)
- Fixed an issue that caused Parquet files to fail during the LOAD activity. (ticket #96191)
    - Other NFS file types, including ORC, CSV, and Avro, also run successfully.

Alerts

- Fixed an issue when saving batch names that used spaces between delimiters, which caused an invalid error to occur. (ticket #97028)

Validate Source

- The Add Column Names feature is now removed from the Source tab. (ticket #96066)
    - Instead, use the query to edit/limit columns or use Update Scope.
- Fixed an issue where disabling source check on a cloned dataset resulted in an error. You can now disable source validation on cloned datasets. (ticket #97795)

Dupes

- The Advanced Filter is now hidden from the Dupes tab. (ticket #96065)

Shapes

- Fixed an issue when editing a dataset that reverted the Shape Detection setting (Off, Auto, or Manual) applied when it was created. (ticket #95471, 95473)

Schema

- Fixed an issue with schema detection on files where schema detection was performed on all columns when a subset of columns was selected. (ticket #92476)
    - Use the`headercheckoff` flag when it is necessary to see only when columns are added or dropped.
- Fixed an issue where schema changes were not correctly identified and updated. (ticket #96013)

Behavior

- Fixed an issue with behavior lookback(`-bhlb`) that caused Row Count changes to be misrepresented. (ticket #94840)

Connections

- Azure Blob connections in standalone environments require the following jars to be added to the `$SPARK_HOME/jars` folder:
  - hadoop-azure-3.2.0.jar
  - wildfly-openssl-1.1.3.Final.jar

API

- Fixed an issue with the DB import process to ensure JobSchedule records import without error. (ticket #98405)

Known Limitations

DQ Job

- Job termination is not supported for jobs in Unknown status.

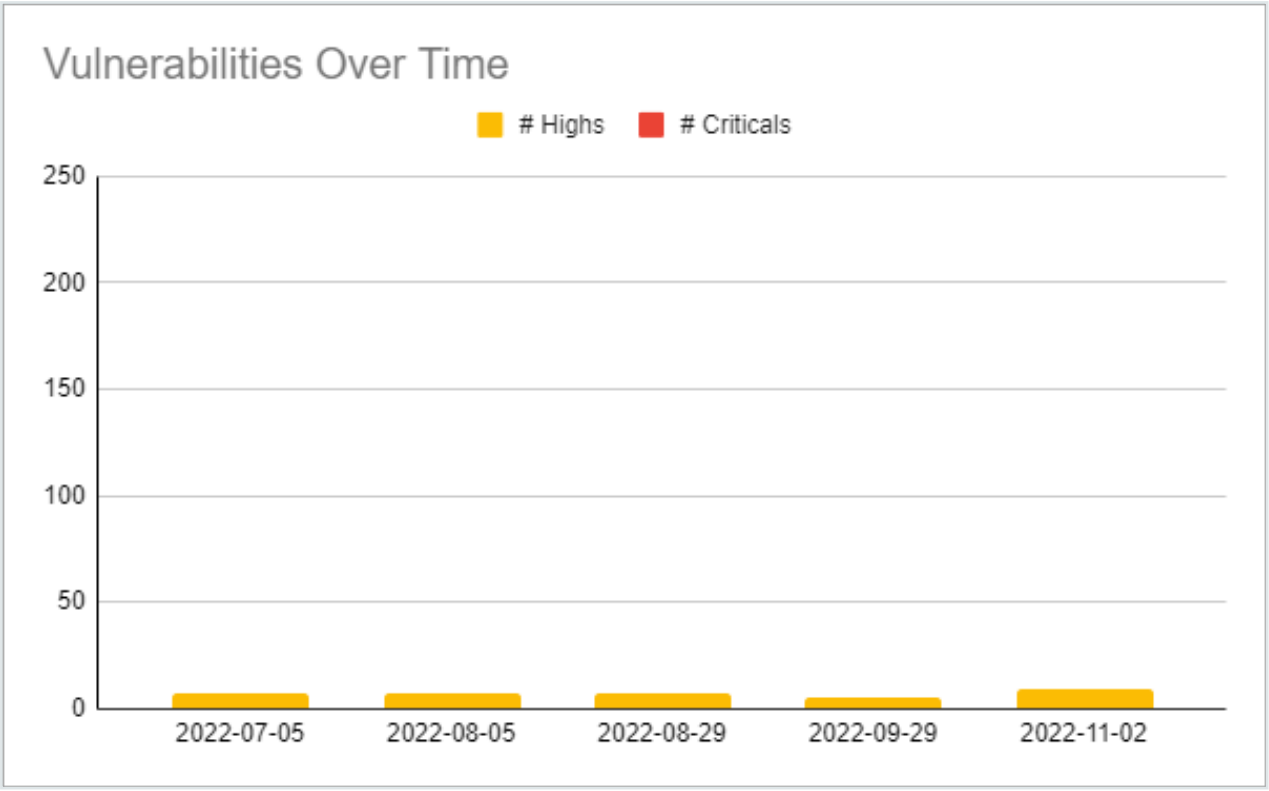
Validate Source

- Cloning and saving, enabling, or disabling the source tab is associated with the original dataset name and fails on the screen when an update is made, but does not affect the actual job run.

Connections

- When adding driver properties using the +Add Property option for Snowflake connections, semicolons are incorrectly appended to key values. Instead, use comma format to separate key values.

DQ Security Metrics



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-07-05	2022.07	0	0	0	7	13	JFrog Artifact
DQ	2022-08-05	2022.08	0	0	0	7	14	JFrog Artifact
DQ	2022-08-29	2022.09	0	0	0	7	12	JFrog Artifact
DQ	2022-09-29	2022.10	0	0	0	5	13	JFrog Artifact
DQ	2022-11-02	2022.11	0	0	0	9	6	JFrog Artifact

Release 2022.10

New Features

Warning For the Collibra Data Quality 2022.10 release, all Docker images run on JDK11. Standalone packages contain JDK8 and JDK11 options. If you are an existing customer who requires JDK11, please upgrade your runtime before upgrading to 2022.10. Most Hadoop environment versions (EMR/HDP/CDH) still run on JDK8, so customers using these environments can upgrade with the JDK8 packages. If you

prefer to upgrade to JDK11, you must follow the documentation of your respective Hadoop environment to upgrade to JDK11 before deploying the 2022.10 release.

The MS SQL driver that comes with JDK11 standalone packages does not currently work in the JDK11 environment. MSSQL requires a separate JAR for JDK11. Please contact your Customer Success Manager for the compatible driver.

Dremio is not currently supported for JDK11 standalone packages. If you plan to run JDK11, add `-Dcdjd.io.netty.tryReflectionSetAccessible=true` to owlmanage.sh as a JVM option for your Web and Spark instances. Please contact your Customer Success Manager for assistance.

As of October 18, 2022, all images for the 2022.10 release have a Critical CVE (CVE-2022-42889). If you picked up the 2022.10 release before October 18, 2022, there should be no issue with your scans. If issues persist, please contact your Customer Success Manager for a new build.

Rules

- You can now define a rule to detect the number of days a job runs without data by using `$daysWithoutData`.
- You can now define a rule to detect the number of days a job runs with 0 rows by using `$runsWithoutData`.
- You can now define a rule to detect the number of days since a job last ran by using `$daysSinceLastRun`.

Profile

- You can now use a string length feature by toggling the Profile String Length checkbox when you create a data set.
    - When Profile String Length is checked, the min/max length of a string column is saved to table dataset_field

Validate Source

- You can now write rules against a loaded source data frame when `-postclearcache` is configured in the agent.

> **Note** The DQ UI will be converted to the React MUI framework with the 2022.11 release. Prior to the 2022.11 release, you can turn the React flag on, but note that some features may be temporarily limited.

Enhancements

DQ Job

- Start Time and Update Time are now based on the server time zone of the DQ Web App.

Scheduler

- The Job Schedule page now has pagination.

Scorecards

- From Pulse View, you can now view missing runs, runs with 0 rows, and runs with failed scores.

Admin/Catalog

- Connection details are now masked when non-admin users attempt to view or modify database connection details from the Catalog page. Only users with role_admin or role_connection_manager have the ability to view connection details on this page. (ticket #94430)

API

- The /v2/getRunIdDetailsByDataset endpoint now provides the following:
  - The RunIDs for a given data set.
  - All completed DQ Jobs for a given data set.

Snowflake Pushdown (in preview)

- You can now detect shapes that do not conform to a data field. Pushdown jobs scan all columns for shapes by default.
- You can now view Histogram and Data Preview details for the Profile activity.

Connections

- The Snowflake JDBC driver is now updated to 3.13.14.

Fixes

Rules

- Fixed an issue with the Rule Validator that resulted in missing table errors. The Validator now correctly detects columns. (ticket #93430)

DQ Job

- Fixed an issue that caused queries with joins to fail on the load activity when Full Profile Pushdown was enabled. Pushdown profiling now supports SQL joins. (ticket #92409)
- Fixed an issue that caused jobs to fail at the load activity when using the CTE query. Please note that CTE support is currently limited to Postgres connections. (ticket #88287, 89150)
- Fixed an issue that caused inconsistencies between the time zones represented in the Start Time and Update Time columns.

Agent

- Fixed the loadBalancerSourceRanges for web and spark_history services in EKS environments. (ticket #95398)
    - The helm property `global.ingress.*` has been removed to separate the config for web and spark_history. Please update the property as follows:__`global.web.ingress.*``global.spark_history.ingress.*`
- Added support to specify the inbound CIDRs for the Ingress using the property `.global.web.service.loadBalancerSourceRanges`. (ticket #95398)

    - Though Ingress is supported as part of Helm charts, we recommend attaching your own Ingress to the deployment if you need further customization.
    - This requires a new Helm chart.
- Fixed an issue that caused Livy file estimates to fail for GCS on K8s deployments.
- Fixed an issue that caused jobs to fail for GCS on K8s deployments.

Validate Source

- The Add Column Names feature is scheduled for removal with the upcoming 2022.11 release. (ticket #96066)
    - This was a previous functionality before being able to limit the query directly (`srcq`) and Update Scope was added.
    - Use the query to edit/limit columns and also use Update Scope.
- Fixed an issue that caused the incorrect message to display for [VALUE_THRESHOLD] when validate source was specified for a matched case. (ticket #94435)

Dupes

- The Advanced Filter is scheduled for removal from the Dupes page with the upcoming 2022.11 release. (ticket #96065)

Explorer

- Fixed an issue that caused BigQuery connections to incorrectly update the library (`-lib`) path when a subset of columns was selected. (ticket #96768)

Scheduler

- Fixed an issue that prevented the scheduler from running certain scheduled jobs in multi-tenancy setups. Email server information is now captured from the correct tenant. (ticket #92898)

Known Limitations

Rules

- When a data set has 0 rows returned, stat rules applied to the data set are not executed. While a full fix is planned for a future release, this limitation is only partially fixed as of 2022.10.

DQ Job

- CTE query support is currently limited to Postgres connections. DB2 and MSSQL are currently unsupported.

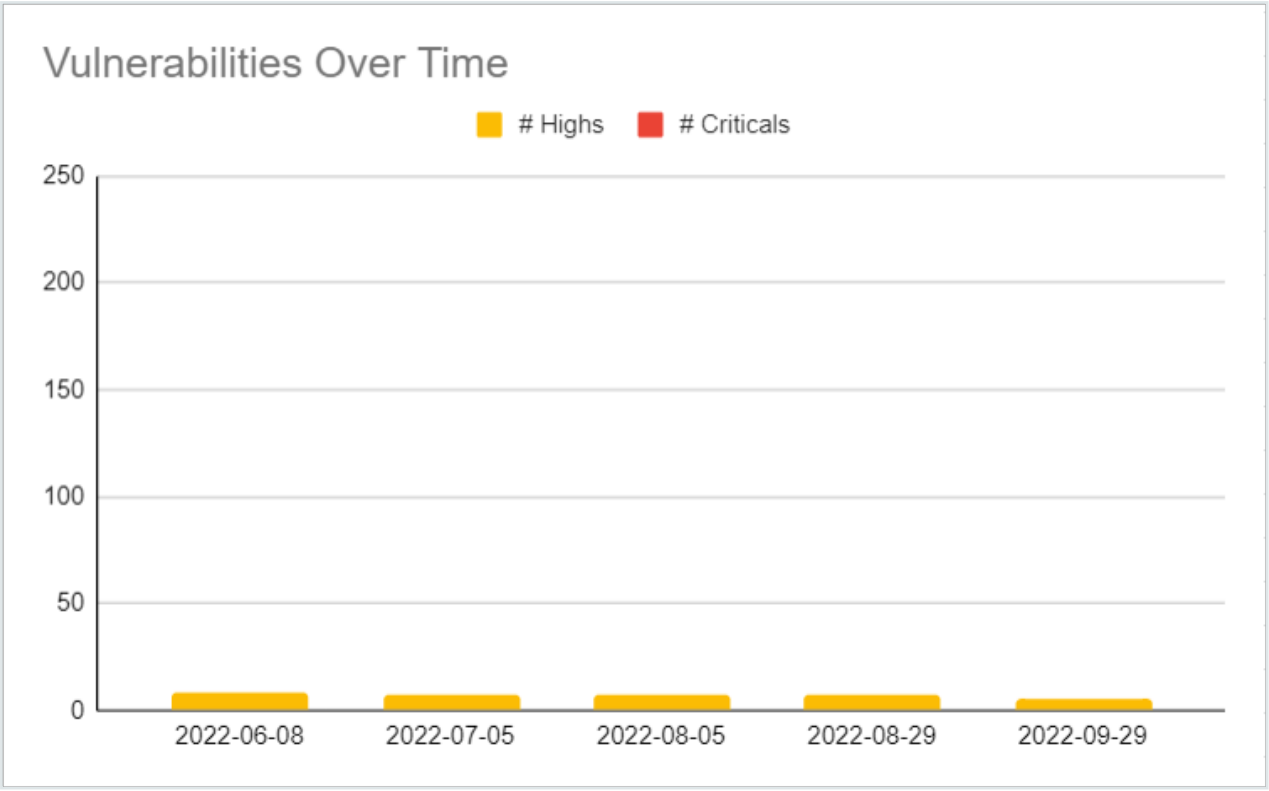
Catalog

- When using the new bulk actions feature, updates to your job are not immediately visible in the UI. Once you apply a rule, run a DQ Job against that data set. From the Rules tab, a row with the newly applied rule is visible.

Snowflake Pushdown (in preview)

- Freeform (SQLF) rules cannot use a data set name but instead must use `@dataset` because Snowflake does not explicitly understand data set names.
- When using the SQL Query workflow, selecting a subset of columns in your SQL query must be enclosed in double quotes to prevent the job from running infinitely and without failing.
- Min/Max precision and scale are only calculated for `double` data types. All other data types are currently out of scope.

DQ Security Metrics



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-06-08	2022.06	0	0	0	8	13	JFrog Artifact
DQ	2022-07-05	2022.07	0	0	0	7	13	JFrog Artifact
DQ	2022-08-05	2022.08	0	0	0	7	14	JFrog Artifact
DQ	2022-08-29	2022.09	0	0	0	7	12	JFrog Artifact
DQ	2022-09-29	2022.10	0	0	0	5	13	JFrog Artifact

Release 2022.09

Enhancements

Rules

- The Conditions column on the Rules tab now displays SQLG and SQLF rule definitions on hover.

DQ Job

- The Jobs chart now shows a dotted gray line to represent jobs in Submitted status.
- The Jobs chart now supports an hourly view option.
- When you run a Pushdown Job that has a data set that returns 0 rows, an unclear message displays.

Schema

- From the Config tab in Explorer, a Check Header checkbox under DQ Job is now available for when column names contain special characters. The Check Header checkbox is checked by default.
    - When checked, schema findings do not display when detected.
    - When unchecked, schema findings display when detected.

Behavior

- Mean values are now rounded on the Findings page.

Explorer

- SOH delimiters for files are now supported.
- The Only checkbox on all Build Layer tabs is now removed.
- The Profile activity is now always enabled and no longer has an on/off switch.

Alerts

- Only one email per alert is now sent when alerts are set up for a scheduled job.
- You can now check the logs to see when an alert does not send in order to resend the email.

Scheduler

- The findings page now displays a green indicator next to the Schedule icon when you schedule a job to run automatically. If Scheduler is inactive, a red indicator displays.

API

- The v2/gethoot API now properly returns rule dimension information for data sets. (ticket #89973)

Connections

- The Databricks connection template has changed, due to an upgrade of the driver. Any existing connection that uses the old driver must be updated. Refer to the new template. (ticket #19950)
- The drivers for Athena, BigQuery, MongoDB, GCS, Hive/Impala were also upgraded but no connection change is required.

Spark

- The 2025.06 release uses Spark 3.2.2.

> **Note** We recommend using Spark 3.x for standalone installs/upgrades.

Fixes

Explorer

- Fixed an issue that prevented the Job Estimator from properly displaying row estimates when the run date was modified during a new job run. (ticket #90860)
- Fixed an issue that prevented DQ jobs created using NFS connection types from displaying under the Remote File Connections dropdown. (ticket #92479)
- Fixed an issue that caused the file type parser to throw an error message when the default comma delimiter was not detected. The parser now detects a file's delimiter and updates the delimiter type in the UI automatically. (ticket #89489, 92480)

Files

- The error message for Failed Merging Schema now has extra logging to clarify the cause of failed schema merges for both Livy sessions and non-Livy paths. (ticket

#92694)

Security

- Fixed an issue with the v2/getcatalogtableshasrulesfromcxn API that triggered a 403 status code when Dataset Security was enabled. (ticket #93298, 94258)

Agent

- Fixed an issue that caused the Agent Check to no longer attempt check-ins to the metastore on K8s deployments, which resulted in red (unhealthy) status. (ticket #92055, 92963)
- Fixed an issue that prevented concurrent users from properly running Livy sessions. (ticket #92963, 90432)

Known Limitations

Rules

- The Rule Builder page becomes unusable if the user creates, validates, saves a new rule and then re-edits.
    - The workaround for this limitation is to do a full page refresh.
- When a user attempts to validate a rule that contains a stat, an exception error is returned.

Security

- The Assignments Queue feature is only available for local users. Support for externally connected users, such as SAML and AD connector, is not currently available.

Alerts

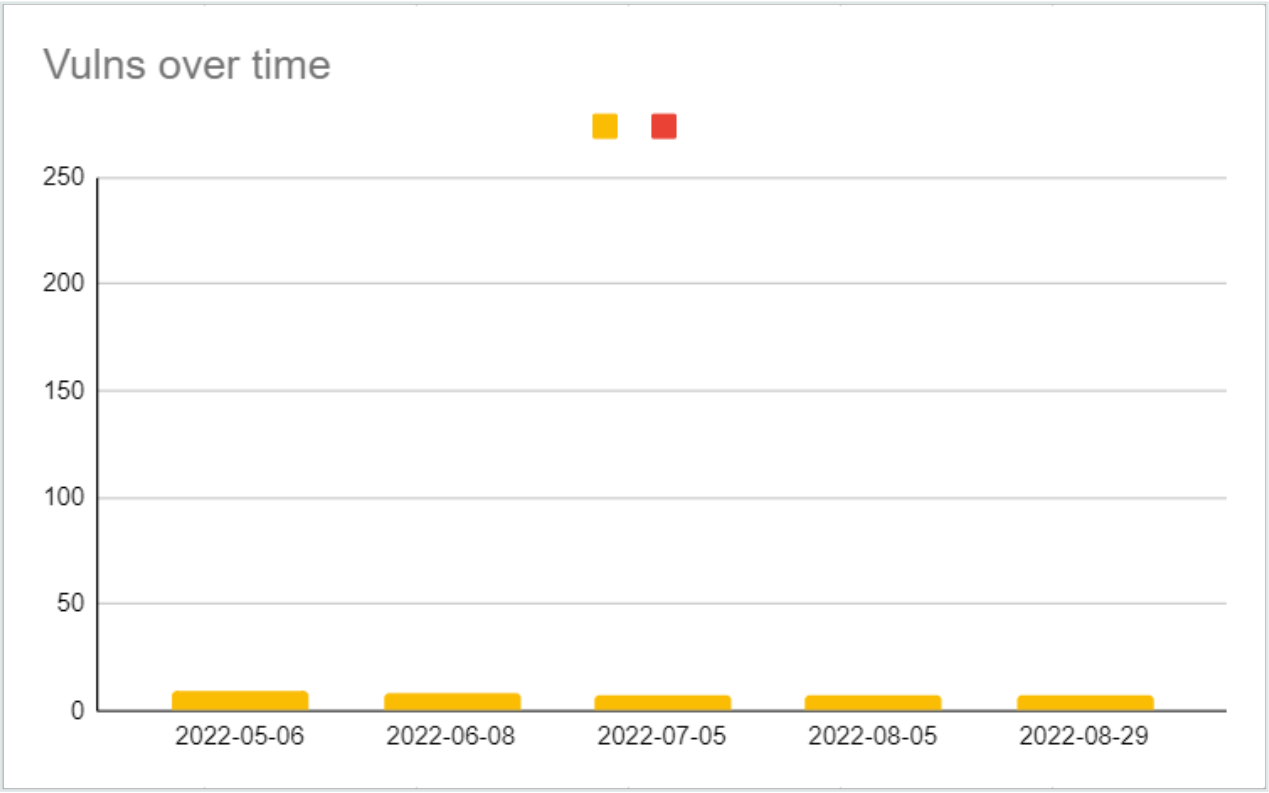
- When alert recipient email addresses are separated by semicolons `;`, alerts emails are not sent to the intended recipients.

- A workaround for this limitation is to separate alert recipient email addresses with commas , instead of semicolons.

Snowflake Pushdown

- When a Job is run, which has a data set that returns 0 rows, an unclear message displays.
- When a native rule is created that contains an embedded stat, its calculated value will not display on the Job results page.
- Data Set security is not supported.
- Disabling autometrics will not take effect, therefore, all autometrics are executed.
- Creating a DQ job using only "SQL Query" workflow doesn't allow you to set the rundate value.

DQ Security Metrics



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resol	# Highs	# Mediums	Artifact(s)
DQ	2022-05-06	2022.05	0	0	2	9	19	JFrog Artifact
DQ	2022-06-08	2022.06	0	0	0	8	13	JFrog Artifact
DQ	2022-07-05	2022.07	0	0	0	7	13	JFrog Artifact
DQ	2022-08-05	2022.08	0	0	0	7	14	JFrog Artifact
DQ	2022-08-29	2022.09	0	0	0	7	12	JFrog Artifact

Release 2022.08

New Features

Rules

- You can now write SQLG-type Dataset-level stat rules on mean.

Enhancements

Connections

- You can now authenticate Oracle JDBC connections with Kerberos TGT, Keytab, and Password. (tickets #75267, 76030)
- You can now authenticate SQL Server JDBC connections with Kerberos Keytab in addition to basic authentication.

Rules

- Rule Summary enhancements:
  - You can now select different time periods for analysis.
  - You can now view charts from three different pages, including Rule Detail Summary, Rule Breaks, and Rule Dimension Summary.

Security

- Vulnerabilities identified by Jfrog
    - Vulns 0, criticals 0, high severity 7
    - The majority of the current mediums are due to merging the dq-streaming module into core.
    - For a visual readout, see the DQ Security Metrics section below.

Agent

- You can now optionally configure individual time zones of DQ Job, Web, and Agent. You should only use this configuration when your instance and containers run in different system time zones. (tickets #87024, 87155)

Behavior

- The Behavior tab now has a new column, Delta Percent Change (Δ % Change).
- You can now hover over new tooltips in the following columns:
    - Baseline
    - % Change
    - Δ % Change
    - Zscore
    - Score

Outliers

- Outlier checks are now optimized to skip in certain circumstances. Outlier checks are only skipped when the history load of a specified date column is empty.
- You can now update and modify record flags from the command line with `-rc`, `-rcKeys`, `-rcDateCol`, and `-rcTbin`.

API

- The v2/gethoot API now properly returns rule dimension information for data sets.
- The v3/jobs/run API now has improvements to the 400 Bad Request error messages in specific circumstances.

Reports

- The PDF option is now removed from the Dataset Findingspage. To print dynamic column tables, use CSV or Excel options instead. (ticket #89739)

DQ Connector

- The version of Collibra Integration Library is now updated to 2.4.12.

Fixes

Connections

- The new GCS jars are required to use GCS spark-history-server. (ticket #90623)

DQ Job

- Fixed an issue that caused jobs using .TXT files to incorrectly render custom column names. (ticket #81808)
    - Files with .TXT extensions are now treated as delimited files. Files with .TXT extensions that are not delimited files should use their respective file type from the file type dropdown.
- Fixed an issue with deployments on K8s where jobs failed when the volume name exceeded 63 characters. (ticket #85372)

Agent

- Fixed an issue that caused the v2/updateagent API to fail when numCores was empty. (tickets #89737, 92404, 92680)
    - The numCores field is no longer a required field.

Validate Source

- Fixed an issue that caused validate source jobs to fail when the pkey was mapped to different column names. (ticket #88778)

Rules

- When using Freeform SQL rules with wild-card operators, rules again correctly pass validation. (ticket #89644)
- Fixed an issue with regex rules that use the characters `)`, `,` , and `;` in the rlike, which caused DQ to append spaces to those characters and prevented the regex from operating correctly. (tickets #89417, 92958)
- Fixed an issue that caused rules with column values containing parentheses `(` `)` to break due to the addition of padding before and after closing parentheses. (ticket #85176)
- Fixed an issue that caused rules with special characters such as @ to display incorrectly on the Rules page, Conditions tab, and when exported to Excel.
- Fixed an issue that prevented data sets with attached rules and roles from being renamed. (tickets #85731, 92059, 94315)

Profile

- Fixed an issue where certain results in TopN Values and Data Preview displayed in scientific notation. Scientific notation is now removed from the display. (tickets #82163, 89738)

Explorer

- Fixed an issue that allowed CLOB data types to be visible in the Drag Columns to Target map in the Source tab. (ticket #86902)

API

- The REST API endpoint v2/updateRoleDatasets again correctly saves roles to data sets.

Known Limitations

Rules

- The Findings page displays results from computational stat rules on mean as a single-quote string. For example, '573523.87' > 6763

- Column-level sorting for the Rule Summary feature is not currently available.

Admin

- When adding a Sensitive Label or a Data Category, the Edit and Update functions do not display the selected record. To properly display the record, you must first refresh the page before editing or updating.

Session Activity

- While the application UI is being redesigned, it is possible that when the application times out on the legacy side of the application, you might not be able to see it on the new React MUI side. This can happen when you have the DQ application open on multiple tabs.
    - We are not currently tracking session timeout from the legacy UI to React.

Features in preview

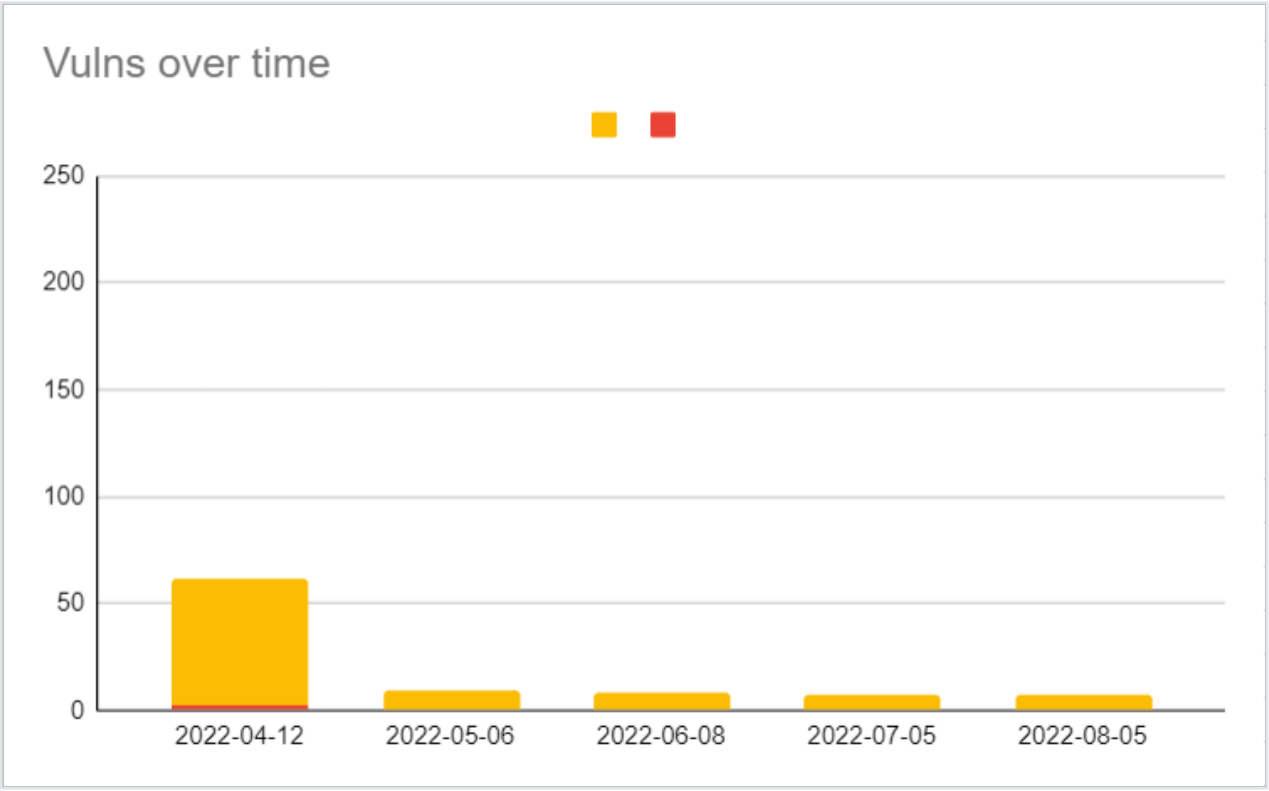
DQ Job

- Collibra is proud to launch a brand new feature, Snowflake Pushdown. Snowflake Pushdown allows for even faster processing and removes the need to set up a separate Spark compute platform to run Collibra Data Quality. Snowflake Pushdown is a private preview feature only available by request. Since this is a preview feature, some limitations are expected as we continue to improve its functionality. Contact your CSM to learn more about this feature.

DQ Security Metrics

> **Warning** There is a critical CVE `CVE-2016-1000027` that shows up in the image scan due to Spring version. This is a false positive and should be added to the exception list of the customer scan tools. We don't use `HttpInvokerServiceExporter` anywhere in the application and are not impacted by it.
>
> There is no fix version available for it from Spring. More details are available at Sonatype

ulnerability CVE-2016-1000027 in Spring-web project · Issue #24434 · spring-projects/spring-framework



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resol	# Highs	# Mediums	Artifact(s)
DQ	2022-04-12	2022.04	2	2	0	60	136	JFrog Artifact
DQ	2022-05-06	2022.05	0	0	2	9	19	JFrog Artifact
DQ	2022-06-08	2022.06	0	0	0	8	13	JFrog Artifact
DQ	2022-07-05	2022.07	0	0	0	7	13	JFrog Artifact
DQ	2022-08-05	2022.08	0	0	0	7	14	JFrog Artifact

Release 2022.07

Note Standalone packages for the 2022.07 release have a version naming convention of `-RC`. This will revert back to the standard naming convention with the 2022.08 release, and has no impact on the safety or stability of standalone packages.

Fixes / Enhancements

- DQ Job
  - Fixed an issue that prevented data from appearing in the Source tab when Source Observation RunID was clicked from the Assignments page.
  - Fixed an issue that caused Annotations with special characters to be truncated in the Labels tab.
  - Fixed an issue that caused the Column (name) column of the Rules tab to display incorrectly when Run Discovery was used.
  - Fixed an issue where the Retrain button on the Record tab was disabled.
  - You can again invalidate observations with single quotes ' from the Shapes tab.
  - The Hints tab now displays any available data.
  - You can no longer change agents from the Scheduler modal.
- Rules
  - SQLF is now supported for Generic rules.
  - When running a custom rule through Rule Discovery, the column names Repo and Column again display correctly.
- Alerts
  - You can now send emails using unauthenticated SMTP servers.
- Security
  - Vulnerabilities identified by Jfrog
    - Vulns 0, criticals 0, high severity 7
    - For a visual readout, see the DQ Security Metrics section below.
  - Fixed an issue that allowed jobs to be run from the command line regardless of connection permissions.
    - When Connection Security is enabled, lock the SQL Editor to prevent unauthorized access to other connections. (#87916)
  - Fixed an issue that allowed View Only users to access some profile results and export the data to a CSV file.
    - Added an authorization check for data set access to the profile export feature, which allows only users with data set access to export the profile. (#87720)
  - Backslashes \ are no longer supported characters for AD usernames without disabling XSS for the /v2/updateadsecurityconfiguration API. (#88499)
  - Fixed an issue that prevented navigation back to the log in page when tenant access was denied. (#89024)

- Profile
    - From the Labels tab, backslashes are now stripped from annotations when they are used for separation within strings.
- Admin
    - From Audit Trail, when administrators modify roles mapped to data sets or data sets mapped to roles, changes are now documented automatically, and display original and updated values.
    - The Agent Group (H/A) and its associated endpoints are now deprecated.
    - From Usage, you can now access a table and tiles reflective of your monthly usage metrics.
    - Salesforce account ID can now be configured for use with Pendo logs.
    - *Tech Preview* [TP] ServiceNow integration
        - You can now assign incidents (validate action) to ServiceNow groups and users with the following fields included in the same request: caller_id, description, short_description, cmdb_ci.
- Explorer
    - Fixed an issue with date range on Oracle connections, which caused end date to change to start date when Transform was selected.
    - The Job Estimate modal again displays the correct number of rows for Sybase connections.
    - Fixed an issue with Source to Target where double quotes " were removed from the source file in database to file targets.
- Scorecards
    - Enhanced the layout of the Assignment Queues page.
- API
    - v2/getallscheduledjobs is now available as an enhancement of the original, v2getscheduledjobs.
        - A UI integration is planned for a future release.
- Schedule
    - Added an Active column to the scheduler export.
        - The RunJob column was removed. (#88799)
- Reporting
    - Fixed an issue that created misalignment of column headers in PDF exports. (#89739)

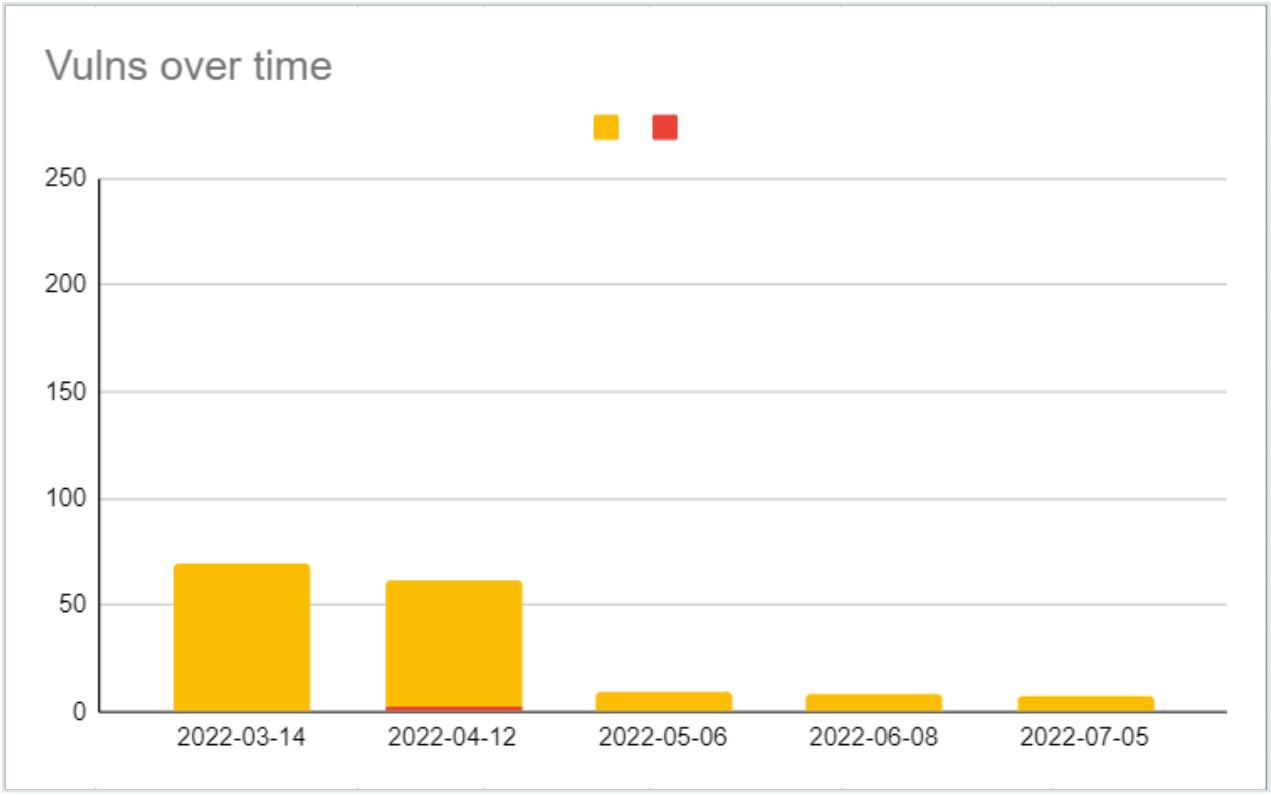
Known Limitations

- Rules
  - To use the new SQLF feature for Generic rules, you must manually update the Generic rule type from SQLG to SQLF.
    - A UI feature for this is planned for a future release.
  - Stat rules such as $rowCount do not work for secondary data sets or previous runId of the same data set via @t1 syntax.
    - To work around this limitation, run a subquery to select count(*) from the secondary data set or the previous runId.
- Explorer
  - Drill-ins and jobs on Sybase connections run successfully, but connections to Sybase with encrypted passwords are currently unsupported.
- Files
  - When using CSV files, you cannot use a comma `,` in the name.
- Admin
  - *Tech Preview* [TP] ServiceNow integration
    - Special characters `!@#$%^&*()` in the description are not supported and will not persist to the ServiceNow assignment queue at this time.
    - Empty or invalid ServiceNow group name does not return an error in CDQ.
      - As a result, the ServiceNow assignment is generated with the default admin account as the owner if left empty or invalid.
      - You must have a valid ServiceNow group name or its related sys_id.
    - The new REACT UI is not yet supported for the ServiceNow Group integration.

DQ Security Metrics

Warning There is a critical CVE `CVE-2016-1000027` that shows up in the image scan due to Spring version. This is a false positive and should be added to the exception list of the customer scan tools. We don't use `HttpInvokerServiceExporter` anywhere in the application and are not impacted by it.

There is no fix version available for it from Spring. More details are available at Sonatype vulnerability CVE-2016-1000027 in Spring-web project · Issue #24434 · spring-projects/spring-framework



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolv	# Highs	# Mediums	Artifact(s)
DQ	2022-03-14	2022.03	0	0	6	70	213	JFrog Artifact
DQ	2022-04-12	2022.04	2	2	0	60	136	JFrog Artifact
DQ	2022-05-06	2022.05	0	0	2	9	19	JFrog Artifact
DQ	2022-06-08	2022.06	0	0	0	8	13	JFrog Artifact
DQ	2022-07-05	2022.07	0	0	0	7	13	JFrog Artifact

Release 2022.06

Fixes / Enhancements

- DQ Job
  - Fixed an issue with the Learning Phase in the Behavior feature. (ticket #82907)
    - Once CDQ has the minimum number of completed successful scans, the learning status now changes to PASSING or BREAKING based on the results.
- Outliers
  - Fixed an issue where file lookback did not identify expected outliers. (#87967)

- Alerts
  - When configuring email alerts, SMTP Username and SMTP password fields are still required fields. (#86033)
    - Validation relaxation is planned for the 2022.07 release.
- Rules
  - Fixed an issue which caused rule breaks to report the opposite of what was defined when a Generic Rule utilizing regex/rlike was created. (#86977)
  - Fixed an issue where Data Classes with Date column types selected did not detect timestamps. (#83000)
  - Fixed an issue where Data Classes using the operators <, > or = caused the inverse rule created from this process to throw exceptions. (#83000)
  - When switching a data class from a regex to expression and then editing again, the regex checkbox is now correctly checked.
- Agent
  - The Explorer page and Scheduler modal now display the same agents. (#86175)
- Security
  - Vulnerabilities identified by Jfrog
    - Vulns 0, criticals 0, high severity 8
    - For a visual readout, see the DQ Security Metrics section below.
  - General advisory:
    - There is a critical CVE `CVE-2016-1000027` that shows up in the image scan due to Spring version. This is a false positive and should be added to the exception list of the customer scan tools. We don't use `HttpInvokerServiceExporter` anywhere in the application and are not impacted by it.
      - There is no fix version available for it from Spring. More details are available at Sonatype vulnerability CVE-2016-1000027 in Spring-web project · Issue #24434 · spring-projects/spring-framework
  - Major vulnerabilities related to Spring, ESAPI, and Swagger have been addressed.
  - No cross DB reference is allowed in explorer while accessing SQL database connections.
  - Sensitive UI fields such as username no longer allow autocomplete.
  - If configured, the ENV variable `XSS_CANONICALIZE_INPUT_ENABLED` should be removed from configmap or owl-env.sh.
  - When dataset security is turned on, you can now add role based authorization for editing existing datasets. (#87720)

    - You can now override the following mail settings from the App Config page within the Configuration section of the Admin Console:
        - "mail.transport.protocol" -- default = smtp
        - "mail.smtp.auth" -- default = true: If true, attempt to authenticate the user using the AUTH command
        - "mail.smtp.auth.login.disable" -- default = false: If true, prevents use of the AUTH LOGIN command
        - "mail.smtp.starttls.enable" -- default = true: If true, enables the use of the STARTTLS command (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands.
        - "mail.smtp.ssl.enable" -- default = false: If set to true, use SSL to connect and use the SSL port by default. Defaults to false for the "smtp" protocol and true for the "smtps" protocol.
        - "mail.smtp.ehlo" -- default = true
        - "mail.debug" -- default = true
        - "mail.smtp.ssl.trust" -- default = : If set, and a socket factory hasn't been specified, enables use of a MailSSLSocketFactory. If set to "*", all hosts are trusted. If set to a whitespace separated list of hosts, those hosts are trusted. Otherwise, trust depends on the certificate the server presents. (#76775, 88089)
- Profile
    - Mean value is now rounded appropriately within the Profile page.
        - For example: The value 2.4334334343345 is now rounded to 2.434.
- Connections
    - From the Athena driver, you can now use `MetadataRetrievalMethod=Query` for database queries from the Connection URL. (#86139)
    - Fixed an issue where error messages on failed connections did not display informational text. (#85527)
    - Fixed an issue where NFS file connections under Remote File connections caused jobs to fail. (#88156)
        - Added File protocol for Spark load for NFS file system.
        - Added nfs:// prefix wile adding a NFS connection.
            - This will prepend the URI with the file:// protocol when an NFS file connection is loaded via Spark.
- Catalog
    - The Graph option is no longer available in Quick links.

- Admin
  - The Pendo integration is now active by default.
    - No sensitive information is collected; only high-level usage stats are collected.
    - All new customers starting with 2022.06 onward will receive a new license.
    - If you install a standalone environment, modify the <install-dir>/config/owl-env.sh file by adding your license name
      `export DQ_INTEGRATION_PENDO_ACCOUNTID=<your-license-name>`
    - This new integration will not block or impair the functionality of the app in any way.
    - For more information on Collibra's subprocessors, please review Collibra's Subprocessors page.
  - The Agent Group (H/A) and its associated endpoints are now deprecated. (#83086)
  - Fixed an issue where the "Add Data Category" button was missing without required permissions. (#86625)
  - When a session expires on an Admin page, you are now redirected to the login page.
  - The Admin Limits page now displays informational text indicating that only limits of Tenant - Admin type are displayed on the page.
  - Fixed an issue when editing an existing data category which caused the 'Add new' modal to open instead of the 'Edit' modal. (#89617)
  - From Configuration Settings, DB Limits is now called Data Retention Policy.
- Explorer
  - You can now view calculated views for SAP HANA when creating a DQ Job on the Explorer page. (#83147, 84328)
  - Fixed an issue which caused the Date range condition to incorrectly display results when using an Oracle connection. (#85802)
  - Fixed an issue which threw an error message when Transform was checked with Date Range condition when using a Postgres connection. (#85802)
  - Fixed an issue where an equals sign = used in a `-transform` expression from Run CMD caused jobs to fail. (#71547)
  - Fixed an issue where schema and table names containing underscores _ were not accepted.
  - Fixed an issue that allowed jobs to run with a row limit of less than 1.

    - Fixed an issue where incorrect files loaded for preview from BLOB containers with Livy enabled.
    - CLOB data types are unsupported. (#86902)
    - Improved performance and logic when drilling into a database and schema from the Explorer page.
- API
    - You can now access API quick links page from the Admin Console React page.
    - When using Swagger, UI text now indicates when a field is case sensitive.
- Reporting
    - *Tech Preview* [TP] Rule Summary page enhancements
        - You can now filter rule breaks by most frequent violations, most severe violations, and least violations.
        - You can now view interactive pie charts with rules and dimension summaries.
- UI
    - The styling of the expandable legacy navigation pane and the react menu are now updated.
- Legal
    - Added a disclaimer to the DQ login page with a link to the Collibra Evaluation Agreement.

Known Limitations

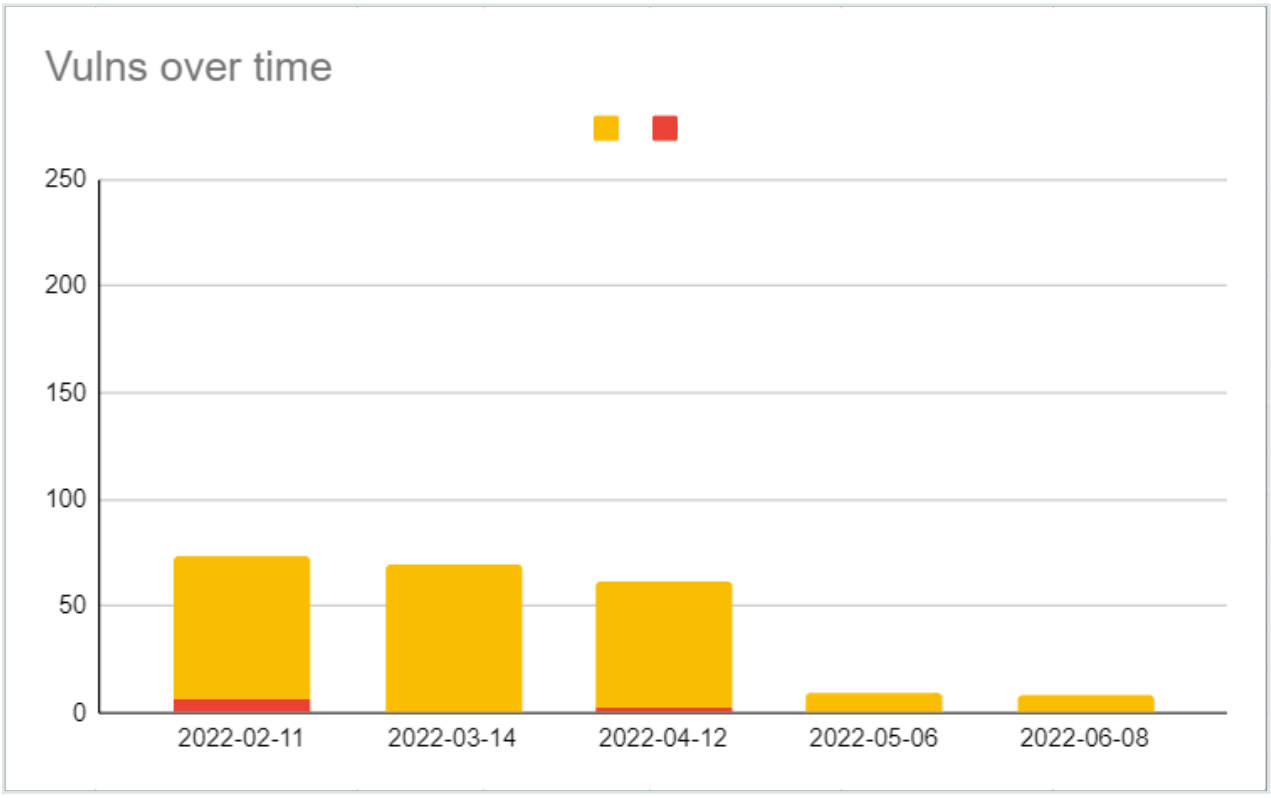
- Validate Source
    - When comparing JDBC (target) to remote files such as S3 (source), there is a known Spark bug for "Recursive view detected".
        - This validate source combination is not possible in 2022.06 using Spark 3.2.
    - When using Bigquery as the source, the -libsrc needs to be manually modified to include the core (Spark Bigquery connector) directory.
        - For example, /home/centos/owl/drivers/bigquery**/core**
- Profile
    - Spark does not currently support varchar data types. All varchar data types are converted to String. Other unsupported data types may also be converted incorrectly.

- Security
    - Permissions on the Export task have not yet been addressed when dataset security is turned on and you add a role based authorization for editing existing datasets. (#87720)

DQ Security Metrics

Warning There is a critical CVE `CVE-2016-1000027` that shows up in the image scan due to Spring version. This is a false positive and should be added to the exception list of the customer scan tools. We don't use `HttpInvokerServiceExporter` anywhere in the application and are not impacted by it.

There is no fix version available for it from Spring. More details are available at Sonatype vulnerability CVE-2016-1000027 in Spring-web project · Issue #24434 · spring-projects/spring-framework



Product SKU	Date	Release	# Criticals	New Criticals	Criticals Resolved	# Highs	# Mediums	Artifact(s)
DQ	2022-02-11	2022.02	***6***	5	5	68	80	JFrog Artifact
DQ	2022-03-14	2022.03	0	0	6	70	213	JFrog Artifact
DQ	2022-04-12	2022.04	2	2	0	60	136	JFrog Artifact
DQ	2022-05-06	2022.05	0	0	2	9	19	JFrog Artifact
DQ	2022-06-08	2022.06	0	0	0	8	13	JFrog Artifact

Release 2022.05

Fixes / Enhancements

- DQ Job
  - You can no longer update the dataset name (`-ds`) from the command line.
    - A helpful error message now appears if changes are made to `-ds`.
  - Stop Job action is no longer enabled for K8s.
  - Fixed an issue for Dremio jobs where jobs hang when editing or cloning an existing dataset.
- Outliers
  - Added "username" to outlier boundary table to track who creates the boundary.
    - The Outlier boundary again saves correctly after the addition of a username.
  - Fixed an issue that caused jobs to fail when Day from By dropdown was selected.
- Rules
  - Rule Preview drill-in capabilities are now improved:
    - You can now configure Preview Limits based on the individual rule.
      - Freeform and Simple rules are currently supported for the Preview Limit feature.
    - You can now set any positive number as the Rules Preview Limit.
      - When you update a Preview Limit value, you must re-run to apply the updated limit value.
    - On the DQ Job page, the details of an individual rule now displays a paginated sub-table of all the break records.
    - When a rule is labeled as BREAKING for rule types other than Freeform and SQL, UI text now displays, "Data preview records are only available for Freeform and Simple rules."
  - You can now hover over stat rules to see their conditions.
  - Data Concepts is renamed Data Categories.

    - Semantics is renamed Data Classes.
    - When a Data Class is assigned to a dataset via Profile controls, a rule is now created.
- Security
    - Vulnerabilities identified by Jfrog
        - Vulns 0, criticals 0, high severity 9
        - For a visual readout, see the DQ Security Metrics section below.
    - The OS vulnerabilities from the images of Collibra DQ 2022.04 have been resolved by using the base image of RHEL8 to build the images for Collibra DQ 2022.05. The following OS utilities will not be available in the 2022.05 release images:
        - Unified, OpenSSL crypto/stack
        - Full YUM stack
        - OS tools, including tar, gzip, and vi
    - AD users can again use auth/signin REST API.
    - The Highcharts CVSS2: 9.3/CVSS3: 9.8 vulnerability is resolved.
    - The LOGJAM (CVE-2015-400) SSL/TLS vulnerability is resolved.
    - The SpringShell (CVE-2022-22965) vulnerability is resolved.
    - TLS < 1.2 is no longer supported.
    - When Azure AD SSO sends a groups.link assertion, the application now tries to resolve the groups via the link.
        - You can now activate this setting by using the property, SAML_GROUP_LINK_PROP.
- Profile
    - You can now edit or delete semantics by clicking anywhere in the semantics cell of the Profile column table.
    - You can now save annotations with special characters.
        - Special characters that are not currently supported include percent sign `%`, backslash `\`, and caret `^`.
    - Fixed an issue where columns of broken rules were not highlighted.
- Connections
    - You can now view a list of all packaged and optionally packaged drivers on our new Builds page.
    - The Databricks JDBC driver is now available.
    - You can now add Databricks datasets using the Databricks Simba driver.

- Catalog
  - Fixed an issue where the deletion of a dataset caused orphaned links to datasets in other areas of Collibra DQ.
- Admin
  - *Tech Preview* [TP] You can now use the ServiceNow integration through a proxy server from the Assignment Queues screen.
  - You can now access the new Usage page to view monthly historical usage statistics.
  - AD users with Admin privileges can now add Business Units.
  - AD users with Admin privileges can now manage local users.
  - The Agent Groups (H/A) feature is marked for deprecation and will be removed from the app in the 2022.06 release.
- Explorer
  - You can again edit schema and table name from the Catalog page.
  - You can now navigate to a specific behavior tab directly from the Assignments page.
  - Fixed an issue when viewing Schemas in View Data wizard.
- Scorecard
  - Single-space ``, underscore _, and period . are now supported characters when saving Scorecard name.
- API
  - Improved API calls for the UserManagement Save function.
- Reporting
  - *Tech Preview* [TP] Rule Summary page enhancements
    - You can now filter rule breaks by a specified date range and view charts for Most Used Rule Types, Dataset with Most Rule, and Top Rules Run.

Known Limitations

> **Warning** Delta Files
>
> A bug was introduced as a result of removing CVEs in 2022.05. If you use Delta files - delta it is not advised to upgrade until an update is available.

- Explorer
    - Except for underscore _, special characters are not currently supported in schema or table names.
- Admin
    - *Tech Preview* [TP] ServiceNow integration
        - Only the local Docker container proxy has been tested and verified.
        - The Test Connection button's validating credentials capabilities is currently limited if the ServiceNow URL is valid.
        - The Validate All Rules function currently results in a failure.
        - You cannot edit an active ServiceNow assignment.
            - Invalidate/Validate or Resolve actions result in a failure.
        - You can assign a ServiceNow ticket with an embedded URL when escaped with double quotes.
            - No assignment is sent without this process.
- Multi-Tenant
    - Tenant names should be lower case. Use lower case characters, when creating a tenant from the multi tenant admin page. The current limitation is around the schema that is generated
- Reporting
    - *Tech Preview* Rule Summary page enhancements
        - Sorting any column returns an error.
        - User must use date picker as manual date entry is not honored.
        - The start and end date are out of order when navigating to the page.
        - The last page on the paginated list does not change when date criteria is updated.

Release 2022.04

Install

> **Note** For standalone installations, within the setup.sh script find/replace the variable for **spark_package**.
> Change spark-3.0.1-bin-hadoop3.2.tgz to spark-3.1.2-bin-hadoop3.2.tgz

```
spark_package=${SPARK_PACKAGE:-"spark-3.0.1-bin-hadoop3.2.tgz"}
# replace with
spark_package=${SPARK_PACKAGE:-"spark-3.1.2-bin-hadoop3.2.tgz"}
```

Fixes / Enhancements

- DQ Job
  - Entering negative values for the downscore is no longer supported and will now produce an error message.
  - You can now invalidate schema with special characters.
  - Spark table names of historical dataset loaded and other spark tables are now available on Jobs Log table.
  - Long type values larger than `Integer.Max` no longer breaks the Profile.
  - View Findings now displays user's full name, if applicable, in Validate Modal. Assignment queue page also displays the full name of user, if applicable.
- Alerts
  - You can once again use the Cancel action button on the Alerts page.
  - You can now set up alerts to reach multiple email recipients.
  - If email_server table is not yet configured, a helpful message will now display in the Description column in the job log directing you to register an email Server under Admin - Alerts. The job will still run successfully.
- Rules
  - You can now modify Rules definitions from the primary DQ Job dashboard without loading the Rules page.
  - Mean value check once again triggers correctly for Integer and Long columns.
    - This fix triggers the mean value check for Integer and Long columns and shows an infinity percentage change in behavior for a period, depending on -bhlb. After this period, it should disappear.
  - For Native SQL rules, jobs now behave the same whether or not a semicolon ";" is included in the SQL query.
  - You can now use a hyphen "-" in a dataset name.
    - Acceptable special characters now include a hyphen "-", period ".", and underscore "_".

    - Added a tooltip that displays which condition is being checked in a DQ Job when using a Stat rule when you hover your cursor over a condition in the Condition column.
    - Improved the exception message for when there are no values for a specific column while using a Stat rule.
    - The WebUI passing boundaries range has been updated to ().
    - For Freeform rules, IS Null and IS NOT NULL no longer return invalid results in the Validation tab.
    - Added a pop-up success message for when the correct syntax rule passes for Freeform rules with secondary datasets after the Validate button is clicked.
- Security
    - Vulnerabilities identified by Jfrog
        - Vulns 2, criticals 2, high vulnerabilities
        - For a visual readout, see the DQ Security Metrics section below.
    - Authorization restriction is now enforced for the following endpoints:
        - /v2/deletefiledir
        - /v2/getRunIdsByDataset
        - /v2/putDatasetWeight
        - /v2/checkListofFilesPath
        - /v2/getlistagents
        - /v2/checkDriver
        - /v2/getconnectionssensitive
        - /v2/getemailgroups
        - /v2/getemailserver
        - /v2/addemailgroup
        - /v2/validateEmailAddress
        - /v2/getlistoffiles
        - /v2/getlistoffilespath
        - /v2/getlistoffiles
        - /v2/getDriverDir
        - /v2/getlistrolesbydataset
        - /v2/getlistrolesbydistnctdatasets
        - /v2/getlistrolesbyfunctiontypename
        - /v2/getlistusersbyauthority
        - /v2/getlocalDBRoles
        - /v2/getsecuritysettingsbytype

        - /v2/getowlcheckinventory
        - /v2/getconnectionspwdmgrsensitive
        - /v2/getsecuritysettingsbycoltype
        - /v2/getdbuserlist
        - /v2/getdbuserdetailsbyuser
        - /v2/getexternaladgroupstointernalroles
        - /v2/getlistdatasets
        - /v2/getlistdatasetsbyrole
        - /v2/getaudittrailitems
        - /v2/get-all-audit
        - /v2/get-datasets-audit-trail-items
        - /v2/get-all-dataset-audit
        - /v2/getactivityaudit
        - /v2/getallactivityaudit
        - /v2/getlocaldbrolesbyuser
        - /v2/getdatasetaclsecurity
        - /v2/getexternaladgrouplist
        - /v2/getexternaladuserlist
        - /v2//external-service-configuration
    - Local user accounts now have an account lockout feature implemented with the following restrictions:
        - A user's account will be locked if a password is entered incorrectly more than 10 times. You can configure this by specifying an integer value for the Admin Console ▸ Application Configuration setting **USER_MAX_INVALID_ATTEMPTS**.
        - The locked account can only be unlocked by Admin user in user management screen.
        - If an Admin is locked, another Admin can unlock their account.
        - If all the Admins are locked, enable the account via DB (update users table "accountNonLocked" column to "1").
        - User cannot use forgot password to reset password while the account is locked.
    - CORS restriction is now enforced for SAML and multi-tenancy.
        - This breaks SAML unless the IDP is configured as a trusted origin in DQ, so the following property must be added to environment variables in order for DQ and SAML to work: CORS_ALLOWED_ORIGINS=\({IDP-BASE-URL},\)

        - Replace $ with the value of the actual IDP URL (For example: https://ping.auth.com)
        - Replace $ with the value of the actual DQ Base URL (For example: https://dq-env.com)
    - SAML login no longer automatically triggers on the login page during an existing session when accessing DQ base URL. For SAML login, you should instead use /saml/login.
        - API requests (v2/v3) return proper JSON response in case of failures.
        - auth/signin API is updated to provide JWT token for MT & local users.
- Profile
    - Mean value once again displays in the Volume column.
    - When connecting to MSSQL server on Windows from a Linux DQ environment, the connection no longer fails.
        - We recommend (not required) a TLS connection for MSSQL connections from a DQ Linux environment with a properly signed certificate setup on MSSQL server to connect only via TLS.
    - You can now edit annotations in the Labels tab.
- S3
    - Added an enhancement for -addlib flag.
- Connections
    - Added new Jconn4 driver for encrypted connections.
    - Tech Preview - You can now save a local (NFS) file directory as a connection type.
    - See our newest Supported Connections page for a definitive guide to driver support.
    - BigQuery is now certified for production, but removed from packaged install for K8s docker.
- Explorer
    - When toggling between fullfile and Union LookBack options, `-fullfile` and `-fllb` flags can no longer be generated together in the DQ Job command line.
    - Data Preview for Temp files loading in Explorer now correctly shows the order of columns of the original Temp file.
    - You can now drill in and search files within the connection.
    - You can now browse multiple local (NFS) file connections.
- Scorecard
    - You can now create scorecards with special characters "^[A-Za-z0-9]+$" in their names.

- Dupes
  - Added linkID column for exact match in both UI and REST API. linkID can now be either included or excluded from Dupes for exact match.
  - linkID is now shown at the aggregate level for Exact Match.
    - We recommend using this feature from a primary key perspective for its first iteration.
    - The aggregate function used is min().
      - For example: if you have 6 occurrences, you will get 1 example linkID, the min.
- API
  - Updated the /v2/getlistdataschemapreviewdbtablebycols API call method from GET to POST to support the long query (-q) or very large columns table.
  - Added a new SAML load balancer so the syestem picks the appropriate schema and SAML server URL for Swagger.

Known Limitations

- Profile
  - Special characters are not currently supported in annotations in the Label tab.
- Scorecard
  - Space " ", underscore "_", and period "." are not yet supported for scorecard edit.

(missing or bad snippet)

Release 2022.02

> **Note** For new Standalone Collibra DQ installations, please double check 'Number of Core(s)' field when setting up 'Edit Agent'.

Added UUIDs for Jobs may take additional time on initial startup after upgrade.

Enhancements

- DQ Job
  - Added UUIDs for jobs for better tracking between web and core
  - Improved DQ Job page load performance by optimizing calls

    - Fixed issue DQ jobs would fail when -rd is in "yyyy-mm-dd HH" format
- Outliers
    - *Tech Preview* [TP] Outlier Calibration
        - Feature flag can be set within owl-env.sh or configMap with `export outlier_calibration_enabled=true` (Default is off)
        - Ability to suppress Outlier observations for a user-determined length of time that would have otherwise surfaced as anomalies
        - Once feature is enabled, accessible within Outliers tab on DQ Job page
- Alerts
    - Ability to navigate to dataset specific Alerts from DQ Job page
    - Ability to test SMTP alert configurations when adding an email relay
    - Fixed issue where 'Reply Email' field did not properly accept user input value
        - Please note there are no (Collibra imposed) domain restrictions on Reply Email field
- Security
    - Stricter password policy is enforced on all user/tenant management screens/APIs.
        - The restriction is as follows: Minimum length of 8 characters
        - Maximum length of 20 characters.
        - At least one upper-case letter.
        - At least one numeric character.
        - At least one special character (supported are !,%,&,@,#,$,^,*,?,_,~)
        - User ID and password cannot be the same.
        - Password cannot contain user ID.
    - Change Password functionality on user profile requires a current password of the user.
    - Mitigated 64 critical, 15 high, and 12 medium vulnerabilities identified by JFrog
    - Upgrade Log4J to 2.17.1
        - Please follow Release 2022.02
    - Added connection security checks to users to prevent running jobs and query the tables that are not authorized per connection. This is applicable when `DB Connection Security` is enabled in the Admin Console under `General`.
    - Implemented stricter session management
    - Implemented CORS restriction to mitigate potential CSRF vulnerability
        - Enforced strict CORS policy by not allowing any domain. In order to allow other domains and tweak this behavior, we have exposed the following

properties as environment variables in owl-env:
        - CORS_ALLOWED_ORIGINS=http://facebook.com,http://google.com
        - CORS_ALLOWED_METHODS-S=GET,POST,OPTIONS,DELETE,PUT,PATCH
        - CORS_ALLOWED_HEADERS=X-Requested-With,Origin,Content-Type,Accept,Authorization
        - CORS_EXPOSE_HEADERS=
        - CORS_ALLOW_CREDENTIALS=false
        - CORS_MAX_AGE=0
- *Tech Preview* [TP] Collibra Native DQ Connector
    - Fixed issue where tenant specified on DQ Connector configuration (issuer of the jwt token field within DGC Edge Management page) was not properly accepted; only rules that existed with 'public' schema were brought over; now the DQ Connector will accept the proper values
- Agent
    - Upon potential deletion of an agent, added server side validation to indicate number of scheduled jobs so that users can understand if jobs fail going forward
- Rules
    - Enhanced stability on Parallel Rule execution to ensure all rules load by reverting back to fixed thread counts
    - Display exceptions upon rule execution failure to improve rule management experience
    - Improvements to user experience in Rule Library tab (within Rules page) including filters and column alignment
    - Quick Rule dropdown within the Rules page will save with default severity of 1 point and a threshold of 1 percent
    - Enhanced validation for rules generated in Profile tab
    - Fixed issue where removing semantic tag may not have removed corresponding auto-generated rule
    - Rule name character limit of 100
    - Rule Builder page now returns error messages where the dataset contained 0 records
- Catalog
    - Renaming Dataset from Catalog page keeps associated rules
        - Clone only creates the dataset shell (with DQ job run configs, no additional rules, etc.) will be copied

    - Bulk actions support for Data Concepts
    - Fixed issue where child of business unit could be assigned as parent
    - Fixed issue where clearing individual filters were not functioning
- Validate Source
    - *Tech Preview* [TP] New collapsible section for Query in Source tab; enables users to use custom srcq, similar to query on section on Home tab so that users do not need to edit -srcq in cmd line editor on Run tab
    - Introducing new observation types via `-valscrshowmissingkey` flag
        - Key not in source
        - Key not in target
    - Source Name should be fetched as part of getcatalogandconnsrcnamebydataset API call for a given dataset
    - Fixed issue which prevented Hive from working as Target
- Export / Import
    - Fixed issue that import could not accommodate more than one table insert
    - Fixed bug where certain values were inadvertently inserted into RegEx rules upon Export
    - New endpoints added for Export and Import API
- Connection
    - Fixed Out Of Memory issue with Dremio
        - Explicitly added limit clause in the preview query within Update Scope
        - Dremio driver requires double quotes in Schema, Table, and Column names e.g. "SchemaName"."TableName"
    - Fixed Oracle TIMESTAMPLTZ conversion error
- Explorer
    - Fixed issue where 'Analyze Table' option did not populate for Hive
    - Fixed the static date values showing up in Managed Template and Run Check while running the job via v2/runtemplate API call from swagger UI
- Files
    - File names with spaces are now handled with double quotes t
    - Implemented Supported File Type Check at time of uploading the Temp Files via Explorer
        - Default supported file types are “csv,json,parquet,avro,delta".
        - In order to add/update the supported file types and ensure validation, a new environment variable needs to be added in owl-env.sh as below: `export ALLOWED_UPLOAD_FILE_TYPES="csv,json,parquet,avro,delta"`

        - Tip: For remote files with delimiter, please use the csv dropdown options for files with .txt extension
    - *Tech Preview* [TP] Users have ability to assign an agent when using temp file and local file Explorer paths without manually appending -master to agent or job (previous known limitation)
    - LIMIT values are now properly accepted on the Scope & Range query panel
- Dupes
    - Fixed issue where column selections were not retained from the original DQ Job with Dupes ON for future runs

Known Limitations

- Rules
    - Cannot currently create rule with API /v3/rules; will be fixed in future release
        - Please use /v2/createrule API
- Profile
    - Stat Rules
        - Tool tips will only generate when Max Precision and Max Scale are greater than 0
- DQ Job
    - /v2/runtemplate API still creates 'zombie' job
        - Please use /v3/jobs/run
- LinkID
    - LinkID column selection is case sensitive; breaks may not appear if case does not match
- Outliers
    - Outlier Calibrate
        - Outliers cannot retrain on-demand; to suppress existing Outliers, must rerun the DQ Job for those date(s)
        - In-app labels do not exist for Outliers which have been subject to past, current, or future calibration; references only exist within the `outlier_boundary` table in the metastore

[Informational Only] New Tables Introduced To Metastore In 2022.02

- `outlier_boundary`

[Informational Only] Changes To Metastore Made In 2022.02

```
ALTER TABLE validate_source_metadata ADD COLUMN IF NOT EXISTS
validate_values_show_missing_keys boolean DEFAULT false
ALTER TABLE opt_source ADD COLUMN IF NOT EXISTS validate_val-
ues_show_missing_keys boolean DEFAULT false

ALTER TABLE opt_source ADD COLUMN IF NOT EXISTS filter_cols
character varying[]

ALTER TABLE user_profile ADD COLUMN IF NOT EXISTS external_
user_id VARCHAR

ALTER TABLE owlcheck_q ADD COLUMN IF NOT EXISTS agent_job_uuid
UUID
ALTER TABLE job_log ADD COLUMN IF NOT EXISTS job_uuid UUID
ALTER TABLE platform_logs ADD COLUMN IF NOT EXISTS job_uuid
UUID
ALTER TABLE platform_logs DROP CONSTRAINT IF EXISTS platform_
logs_job_uuid_ux
ALTER TABLE platform_logs ADD CONSTRAINT platform_logs_job_
uuid_ux UNIQUE (job_uuid)
ALTER TABLE opt_owl ADD COLUMN IF NOT EXISTS job_uuid UUID
```

Release 2022.01

Enhancements

- DQ Job
  - Fixed issue where backrun "-br" flag was inadvertently added on future runs (error contained in 2021.12) if the initial DQ Job setup Explorer selected backrun
  - Improved validation to not allow for slashes in dataset name
- Validate Source
  - Fixed potential DQ Job failure with Source turned on for some legacy installations when upgrading from older versions to 2021.11 and newer
- Explorer
  - `DB_VIEWS_ON` can be added with TRUE or FALSE values by adding new App Config (Add Custom within Admin -> Configuration)
  - -Addlib flag now working across JDBC connections
  - Update Scope now supports rdEnd

- Rules
  - When creating rules, run-time limit for each rule (in minutes) can be set on the Rule page UI and on the V3 API (by setting `runTimeLimit` property). The default is 30 minutes if not explicitly set. This 30 minute limit sets the overall timeout limit for all rules in a particular job. For example, if there are 10 rules with 9 rules with 30 min limit and 1 rule as 90 min limit, then the DQ Job will wait up to 90 min for all 10 rules to finish. This is because all rules must finish before the Rule stage in DQ Job to finish and move to the next stage. We do not support async stages where one long running rule is running while the job itself moves on to the next stage.
  - Added ability to specify score of 0 to a rule
  - Improvement to Stat Rules to fail without exception when result is not within range
- Profile
  - Fixed ability to remove a business unit from a dataset
  - Fixed issue where data concepts were not correctly displaying on a dataset's Profile page
  - Fixed sensitive labels not being assigned from Discovery
  - Treat certain doubles, floats, decimal types as Decimal format that preserves length and prevents Java from truncating to E11 format
  - Removed commas when displaying date columns
- Security
  - SAML Login fix for IDPs that use POST binding as default
- S3
  - Enhanced support where "." in column headers were causing large jobs to not complete
    - Underscores now replace periods and large jobs should no longer hang
- Connections
  - Updated default Snowflake template connection properties
    - Corrected 'db' parameter placeholder on connection string versus former 'databaseName'
  - Added Connectivity to BigQuery troubleshooting information

Known Limitations

- Local files using NO_AGENT require a valid $SPARK_HOME on the machine where the web server is running.

- Supported data types
  - CLOB datatypes are unsupported
- Explorer
  - -Addlib not yet supported for Remote Files e.g. S3

[Informational Only] Changes To Metastore Made In 2022.01

```
ALTER TABLE owl_rule ADD COLUMN IF NOT EXISTS run_time_limit
DOUBLE PRECISION NOT NULL DEFAULT 30.0;
ALTER TABLE owl_rule ADD COLUMN IF NOT EXISTS scoring_scheme
INT4 NOT NULL DEFAULT 0;

ALTER TABLE job_log ALTER COLUMN stage TYPE character varying;
-- stage set to varchar because RULE logs rule_nm into stage
ALTER TABLE job_log ALTER COLUMN log_desc TYPE character vary-
ing;
ALTER TABLE job_log ALTER COLUMN log_hint TYPE character
varying;
```

Release 2021.12

Note to Standalone Collibra DQ Customer Upgrades: We have upgraded to Log4J 2.17, please refer to Standalone Upgrade for additional steps

Enhancements

- Rules
  - Semantic and data concept management: Run Discovery feature
    - Run Discovery feature can be accessed from Catalog by selecting 'Data Concept' option from Actions or clicking the 'Run Discovery' button on the Rules tab of the DQ Job page. This will run a DQ Scan to detect for the semantics assigned to the selected data concept
    - Algorithm now selects best match if column matches 2 or more data classes based on % match and name distance

    - *Tech Preview* [TP] Configurable rule break preview limit
        - Global default is 6 max rows per rule
        - Any change from 6 must be specified with previewLimit (API /v2/createrule) or in the Preview Limit field (UI)
        - Maximum of 50 from UI
    - Introducing additional Stat Rules including minPrecision, maxPrecision, minScale, maxScale
- Behavior
    - Min and max value checks are now triggered for all numeric columns when selected, even if column contains zeroes in lookback period
    - AR column view graph now shows theMean value for current day (runId). No re-run of DQ Job is necessary. The displayed Mean makes it clear that the % change is the % change from the mean, not runId - 1 day.
    - Findings in behaviors that were directly correlated to a row count shift as the root cause have been optimized, such that a major deviation in row count will no longer down-score related fields in the dataset to reduce noise
- Catalog
    - Catalog now features intelligent ranking based on Recency, Most Scanned, User
- Outliers
    - Outliers (advanced) allows for gaps in dates when establishing lookback period, which is established by history with row count > x (specified by user)
    - Fixed issue where outlier data preview graphics were not displayed
    - Fixed issue where outlier results did not honor the initial scope where clause, in particular for Remote Files (S3)
- Connections
    - BigQuery: Enhanced support for cataloging host name
- Pulse View
    - Pulse view can filter Connections and Users
    - Pulse view can serve as proxy verification on whether scheduled jobs were successfully completed
- Profile
    - Viewable precision and scale statistics for double, float, and decimal data types
- Shapes
    - Fixed issue where data shape preview not available when same shape is detected on the same row for different columns

- Files
  - *Tech Preview* [TP] Users have ability to assign an agent when using temp file and local file Explorer paths
    - Known limitation: -master must be freeform appended to the agent or to each job
  - Support for multicharacter delimiters
  - Improved delimiter support to distinguish string commas versus actual CSV commas to align data to respective columns
- Agent
  - Fixed issue where certain completed jobs could not be re-run on the DQ Jobs page. In other words, NO_AGENT was the only available option in the Agent dropdown. Now, users can select valid agents in the dropdown and this will persist for future scheduled jobs
- Schedule
  - Implemented validation to enforce user to choose days when picking schedule to avoid Java error messages
- Explorer
  - Fixed issue where '&' was not properly supported when adding additional parameters
- API
  - JSESSIONID session time is configurable
  - Bearer token and JSESSIONID authentication paths are properly forked
- Pattern
  - Patterns activity now shows Count (number of times the current dataset has the Pattern breaks). This Count is interpreted the same way as Outlier activity Count

Release 2021.11

Enhancements

- Rules
  - *Tech Preview* [TP] Semantics and data concepts management
    - The application now supports dynamic semantics checks. This allows you to create custom semantics that can be checked for when running a DQ check on a data set. Previously the application checked against a predefined set of

semantics. You also have access to controls to organize and apply these semantics checks. The following is a list of changes:

- There is a new data concepts management page. You can access it from Catalog or Admin Console. You can assign multiple semantics to a data concept.
- When running a DQ check, you can select a data concept. The semantics assigned to this data concept will be checked against each column of dataset.
- You have a list of predefined semantics that are not editable. You also have the ability to create/edit/delete custom semantics.
- Repo on rules page has been added to Rules Library where semantics will be viewable.

- Resource Limits
  - You can edit the Performance Settings to supply limits to executors, cores, memory and cells so that a user can be warned if submitting a job that requires a lot of resources and admins can control maximum resources submitted.

Enhancements

- Explorer
  - *Tech Preview* [TP] Dynamic query reload allows you to view JOIN query columns in other activities.
    - User can update and reload the schema table with the custom query in the scope section by clicking the [Update Scope] button. It will enable using the new columns from the custom query in all activities (Profile, Outlier, Dupes, Patterns, Source)
    - Since the first tab is for compositing the query, updating fields will change the user's custom query. Therefore, all areas are locked except the "query" field in the first tab to keep the query unchanged after updating the scope table
  - Support for some special characters in table name.
  - Fixed the ability to add additional libs that were previously not being properly saved on subsequent runs. Under DQ Job tag, please utilize -fllb boolean (union lookback) and libsrc input box for lib directory path (will materialize as -addlib).

- Connections
  - *Tech Preview* [TP] BigQuery Views and Joins
  - Please add the following to the BigQuery connection property

```
viewsEnabled=true
```

- API
  - You can perform multiple imports without conflicts.
  - You can have an incremental import such as updating matching records / insert new / leave existing. There is no requirement to delete tables first before running import.
- Profile
  - Fixed backrun timebin to work with weeks and quarters instead of days.
- Outliers
  - Split historical load to avoid historical query rounding up.
  - *Tech Preview* [TP] Outliers (advanced).
- Source
  - Fixed an issue where settings were not sticky for subsequent runs.
- Security
  - SAML Enhancements
    - New SAML Authentication are available when the Load Balancer is set for SSL Termination.
    - You can now set theConfiguring multi-tenant support through SAML RelayState to route SSO to the proper tenant.

Patches

- 2021.11.1 Explorer
  - Allow ampersand in metastore host name for additional parameters
  - In below example, support for ampersand needed for required SSL flags

```
metastore01.us-east1-b.c.customer-dq-prod.in-
ternal:5432/dev?sslmode=required&currentSchema=public
```

Known Limitations

- Rules
  - Semantics and data concepts:
    - Not supported in pushdown mode
    - Exporting RegEx semantics not currently supported
  - While it is possible to create joins and cross-dataset rules using Freeform SQL, it is best practice to create a view and handle the join prior to running the DQ Job.
- Behavior
  - Schema is not eligible for invalidate
- Files
  - Local files using UPLOAD_PATH, UPLOAD_FILE_PATH, and temp files are only eligible to be deployed using the default NO_AGENT option. These are only intended for quick tests and not intended for production-scale use. Best practice is to use a remote file system connection (S3, Google storage or ADLS).
  - Delimiter support for special characters is limited. Supported file delimiters are comma, pipe, tab, semicolon, double quote and single quote. Custom delimiters will work for many characters, but not all combinations.
  - Temp files and NO_AGENT should have -master local[*] or -master spark://:7077 defined in freeform append of the agent options
- DQ Job
  - When submitting jobs via API from a different machine with a different timezone, timezone discrepancies are not accounted for automatically. Best practice is to align each component to use UTC.
  - Jobs submitted via API with a run date that include HH:MM in the -rd (run date) will submit to the job queue and leave a remnant 'STAGED' job
- Connections
  - Postgres limits max connections per spark job. The default is 100. Please refer to Postgres official documentation how to increase max_connection and shared_buffers.
    - https://www.postgresql.org/docs/9.6/runtime-config-connection.html
  - BigQuery
    - Updating scope to include joins in BigQuery can only be materialized when tables are part of the same dataset collection

        - Should you receive an error for pre-existing BigQuery jobs, please add -dssafeoff to the cmd line or select 'Allow Overwrite' to enable this from Edit mode in the Explorer
- Alerts
    - After an upgrade to 2021.11, you may need to set the environment variable `ALERT_SCHEDULE_ENABLED=true` in **owl-env.sh** and restart **owl-web** to enable email alerts to work again.

Release 2021.10

Enhancements

- DQ Job
    - Refactored DQ Job Score to Gauge Chart
- Explorer
    - Fixed issue where permissions are checked on datasets that do not yet exist
- Connections
    - Sybase 'Test / Preview' now available
    - Updated web model of saving additional connection properties
    - Fixed scenario where editing connection yields null instead of empty for multiple values
- Rules
    - Placeholder new searchable Rule Summary Page for Rule statistics / insights
- Alerts
    - Updated Alert Mailer to TLS 1.2 to resolve Third Party Error exception
    - Fixed issue where alerts are deleted even when clicking cancel button
- Behavior
    - Fixed issue where user must refresh to have invalidated item removed from UI
- Search
    - Fixed search on Audit Datasets and Dataset Management page
- Scorecards
    - Date ranges are now customizable
- Validate Source
    - Added feature that provides 'trim' option on String columns when running source-target validation, extra spaces in the cell are trimmed on both ends (left and right)

- Dupes
  - Resolved issue with white spaces in column headers blocking duplicate detection
- Security
  - Added configuration for setting the SAML_ENTITY_BASEURL, which sets the Consumer service url for the SP Metadata
- Shapes
  - Fixed issue where custom values override even after toggling Shapes back to auto or off
- Console
  - Fixed uncaught TypeError on login screen
  - Fixed GET timeout error on registration page
- Export/Import API
  - Users will be able to run the export/import API calls to conduct multiple promotions on the repo, schedule, and rule tables.

Patches

- 2021.10.1 Import / Export API without constraint conflicts
  - Import must match exactly to the format of our export in order to parse out columns and values to perform an update when existing records are already there

```
owl_rule
owl_check_repo
job_schedule
rule_repo
```

Known Limitations

- File sizes
  - Individual files greater than 5gb will experience performance degradation in Explorer for Standalone installs. Best practice is to save in smaller chunks and use bypass schema in the Explorer if needed.
  - Individual files greater than 25gb will experience performance degradation in Core for Standalone installs.
- Files
  - Explorer / browser will generally have difficulty supporting > 250 columns in files

- Profiling
  - Pushdown profiling on Bigquery, Redshift, Athena and Presto is available for specific datatypes.
  - Backrun option and flag will persist beyond the first run (-br). Please remove this flag if you do not want to backrun again.
- Explorer
  - QUARTER and WEEK are not supported time bins in this release.
  - On non-csv files, Explorer will not automatically infer file types. Users must change file type to the required value and click Step 2 "Load File". Nothing will change in Step 1 "File Information". A future enhancement will be added to automatically check filetypes by reading the first file
  - Dataset names should not contain special characters
- Rules
  - Out of the box semantic rules cannot be edited (STATECHECK, GENDERCHECK, etc). Users can still apply their own global rules which can be customized.
  - LinkId does not support alias columns that are not part of the -LinkId definition
- Connections
  - Connection names should not contain spaces
- Validate Source
  - Complex Validate Source queries can only be edited from the CMD line or JSON directly before hitting Run.
- Security
  - Active Directory in Azure SQL can connect via LDAP (basic auth) or Kerberos.
- S3 / GS / ADLS
  - Remote storage connections should be defined using the root bucket only.
- Estimate Job is only available for files when Livy is being used.
- Stop Job on jobs page is limited and does not work for all installation types.
- Bigquery connector does not work with views

About the latest Collibra DQ user interface

The latest Data Quality & Observability Classic user interface (UI) has been updated to provide a more modern, streamlined experience. The following table highlights what's changed between the classic and updated UI and what's exclusively available in the new user experience.

Component	Description	Available in Classic
Metadata Bar	The Metadata Bar is a dataset anchor that simplifies the navigation to some of your most frequently used pages, such as Dataset Overview, Profile, Rules, and Findings. It also provides quick insight into your dataset, such as the number of active rules, the data source from which the dataset was created, and whether or not your job is scheduled to run automatically. When an integration is set up, the Metadata Bar also allows you to easily enable or disable dataset metadata integrations into Collibra Platform. You can access the Metadata Bar on any dataset-level page, including: • Findings • Profile • Dataset Rules • Alert Builder • Dataset Overview	No
Dataset Overview	Dataset Overview lets you query your dataset to discover key data points and insights, which you can convert to data quality rules entirely within the Dataset Overview modal. With the power to write SQL to query your dataset, you can accelerate the process of data discovery and reveal important insights in real time. Dataset Overview also allows you to leverage Collibra AI to automatically write and troubleshoot SQL for faster rule writing and advanced exploration of your dataset.	No
Explorer	The new workflow simplifies the process of creating a DQ Job to run against a dataset. With just a few clicks, you can create a basic profile job in a matter of seconds instead of minutes. For a more advanced scan, the step-by-step guide walks you through the process, eliminating many of the more tedious elements of the classic Explorer. Additionally, we improved the performance when loading schemas with many tables to ensure that the Explorer page doesn't lock in an unresponsive state while they load.	Yes

Component	Description	Available in Classic
Findings	The new Findings page will feel similar to the classic page, but with a few important changes: • The daily score, pulse view, row count, and pass/fail charts are now broken out into individual tabs for an improved display. • The dataset metadata that previously resided next to the chart views is now anchored to the top of the page within the Metadata Bar for quicker analysis of your dataset's key data points. • The findings tabs of the various data quality dimensions now have enhanced readability and more clearly displayed actions.	Yes
Profile	While many of the same column- and dataset-level insights are unchanged from the classic UI, the presentation of information is now modernized for a crisper experience.	Yes
Rules	Dataset Rules lists all previously saved rules for a given dataset and provides an overview of their details, such as the definitions of SQL conditions, rule types, and whether or not rules pass validation checks. From here, you can access the Rule Workbench to create or edit a rule. The Rule Workbench replaces the classic Rule Builder, fusing an elegant SQL command line interface with the preview and advanced setting capabilities you expect from a modern SQL builder. Like the Dataset Overview, you can also use Collibra AI-generated SQL to write and troubleshoot rules on the Rule Workbench. See the SQL Assistant for Data Quality topic to learn more. We've also split Data Class and Template rules into their own pages to emphasize that they are independent of jobs and datasets and improve their overall organization.	No

Component	Description	Available in Classic
Alerts	The new Alert Builder provides an at-a-glance view of all alerts for a given dataset and simplifies adding new alerts or editing existing ones. New alert options include: • The ability to configure Condition alerts to generate an alert when a user-defined condition is satisfied during a dataset run. • The ability to configure Job Status alerts to generate an alert when a job run completes. • The ability to configure Rule Status alerts to generate an alert when a rule is breaking, passing, and/or has an exception. • The ability to set custom alert email signatures. • The incremental delivery of alerts from the automatically enabled alert queue. • General improvements to the body of alert emails.	Yes
Dataset Manager	Dataset Manager provides a list of all datasets in your Collibra DQ environment, as well as a variety of management options, such as assigning datasets to data categories and the ability to filter datasets by a variety of criteria. Additional enhancements include: • New filtering options. • New options for the management of datasets in bulk. • A link to view a dataset in Collibra Platform when an integration exists.	Yes
Column Manager	Column Manager is a detailed breakdown of all the columns in the datasets in your Collibra DQ environment that shows key data points like data type, various ratios, and the Pass/Fail status of a given column. You can also bulk apply rules, data classes, and sensitive labels to selected columns.	No

Component	Description	Available in Classic
Report Dashboards	The Reports section now has two new dashboards available in the updated UI: • Column Dimension is an overview to view the data quality dimensions of the columns of all or specific datasets. By filtering by business unit, dataset, column, and monthly periods, you can make more informed decisions and increase the value of your data. Column Dimension also provides insight into the total current dimension scores, dimension scores across time, and DQ scores (along with other metadata) for each column. • Dataset Dimension is an overview to view data quality dimensions with different filters to make more informed decisions and increase the value of your data. You can filter by specific datasets or all datasets in your Collibra DQ environment, by business unit, and by monthly periods.	No
Connections	The updated Connections page in the Admin Console does away with the connection tiles of the classic page in favor of a highly searchable and sortable paginated table format. The new page also features two tabs for Connections and the Drivers stored in your Collibra DQ environment. Additionally, when you add or edit a connection, the connection template is now organized in three tabs for Connection Details, Driver Properties, and Connection Variables. With these sections now clearly delineated, the process of creating or updating a connection is now much cleaner.	Yes
Admin Console	The Admin Console now lists each admin activity for better organization and simpler navigation than the tiles of the classic UI.	Yes

Known issues

For a list of all known issues in Data Quality & Observability Classic, go to the Support Portal.

Builds

This page provides information about Docker images and other information related to Collibra DQ build packages. Builds follow a naming convention to indicate which Optional Drivers drivers are available out of the box.

> **Note** The optional drivers only impact container versions of Collibra DQ and do not impact Standalone installation packages.

Available Containers

> **Note** The default build is considered the secure build without optional drivers or critical vulnerabilities.

2025

Data Quality & Observability Classic

- 2025.05-ABDGCSHILM-4252
- 2025.05-4251

Spark

- 3.5.3-2025.05-ABDGCSHILM-4252
- 3.5.3-2025.05-4251

Data Quality & Observability Classic

- 2025.04-ABDGCSHILM-4254
- 2025.04-4253

Spark

- 3.5.3-2025.04-ABDGCSHILM-4254
- 3.5.3-2025.04-4253

Data Quality & Observability Classic

- 2025.03-ABDGCSHILM-4240
- 2025.03-4239

Spark

- 3.5.3-2025.03-ABDGCSHILM-4240
- 3.5.3-2025.03-4239

Data Quality & Observability Classic

- 2025.02-ABDGCSHILM-4299
- 2025.02-4298

Spark

- 3.5.3-2025.02-ABDGCSHILM-4299
- 3.5.3-2025.02-4298

Data Quality & Observability Classic

- 2025.01-ABDGCSHILM-4246
- 2025.01-4248

Spark

- 3.4.1-2025.01-ABDGCSHILM-4246
- 3.4.1-2025.01-4248

2024

Data Quality & Observability Classic

- 2024.11.1-ABDGCSHILM-4213
- 2024.11.1-4214

Spark

- 3.4.1-2024.11.1-ABDGCSHILM-4213
- 3.4.1-2024.11.1-4214

Data Quality & Observability Classic

- 2024.10-ABDGCSHILM-4268
- 2024.10-4270

Spark

- 3.4.1-2024.10-ABDGCSHILM-4268
- 3.4.1-2024.10-4270

Data Quality & Observability Classic

- 2024.09-ABDGCSHILM-4243
- 2024.09-4245

Spark

- 3.4.1-2024.09-ABDGCSHILM-4243
- 3.4.1-2024.09-4245

Data Quality & Observability Classic

- 2024.08-ABDGCSHILM-4237
- 2024.08-4239

Spark

- 3.4.1-2024.08-ABDGCSHILM-4237
- 3.4.1-2024.08-4239

Data Quality & Observability Classic

- 2024.07-ABDGCSHILM-4254
- 2024.07-4256

Spark

- 3.4.1-2024.07-ABDGCSHILM-4254
- 3.4.1-2024.07-4256

2024.06.1

Data Quality & Observability Classic

- 2024.06.1-ABDGCSHILM-4208
- 2024.06.1-4209

Spark

- 3.4.1-2024.06-ABDGCSHILM-4208
- 3.4.1-2024.06-4209

2024.06

Data Quality & Observability Classic

- 2024.06-ABDGCSHILM-4249
- 2024.06-4250

Spark

- 3.4.1-2024.06-ABDGCSHILM-4249
- 3.4.1-2024.06-4250

2024.05.2

Data Quality & Observability Classic

- 2024.05.2-ABDGCSHILM-4206
- 2024.05.2-ABDGCSHILMH-4208
- 2024.05.2-4207

Spark

- 3.4.1-2024.05.2-ABDGCSHILM-4206
- 3.4.1-2024.05.2-ABDGCSHILMH-4208
- 3.4.1-2024.05.2-4207

2024.05

Data Quality & Observability Classic

- 2024.05-ABDGCSHILM-4252
- 2024.05-ABDGCSHILMH-4255
- 2024.05-4254

Spark

- 3.4.1-2024.05-ABDGCSHILM-4252
- 3.4.1-2024.05-ABDGCSHILMH-4255
- 3.4.1-2024.05-4254

Data Quality & Observability Classic

- 2024.04-ABDGCSHILM-249
- 2024.04-ABDGCSHILMH-248
- 2024.04-250

Spark

- 3.4.1-2024.04-ABDGCSHILM-249
- 3.4.1-2024.04-ABDGCSHILMH-248
- 3.4.1-2024.04-250

Data Quality & Observability Classic

- 2024.03-ABDGCSHILM-240
- 2024.03-ABDGCSHILMH-242
- 2024.03-239

Spark

- 3.4.1-2024.03-ABDGCSHILM-240
- 3.4.1-2024.03-ABDGCSHILMH-242
- 3.4.1-2024.03-239

Data Quality & Observability Classic

- 2024.02-ABDGCSHILM-211
- 2024.02-ABDGCSHILMH-209
- 2024.02-210

Spark

- 3.4.1-2024.02-ABDGCSHILM-211
- 3.4.1-2024.02-ABDGCSHILMH-209
- 3.4.1-2024.02-210

Data Quality & Observability Classic

- 2024.01-ABDGCSHILM-227
- 2024.01-ABDGCSHILMH-228
- 2024.01-229

Spark

- 3.4.1-2024.01-ABDGCSHILM-227
- 3.4.1-2024.01-ABDGCSHILMH-228
- 3.4.1-2024.01-229

2023

> **Note** As of Data Quality & Observability Classic 2023.11, the default Spark version for cloud native deployments is 3.4.1.

Data Quality & Observability Classic

- 2023.11-ABDGCSHILM-220
- 2023.11-ABDGCSHILMH-223
- 2023.11-221

Spark

- 3.4.1-2023.11-ABDGCSHILM-220
- 3.4.1-2023.11-ABDGCSHILMH-223
- 3.4.1-2023.11-221

Data Quality & Observability Classic

- 2023.10-ABDGCSHILM-310
- 2023.10-ABDGCSHILMH-312
- 2023.10-311

Spark

- 3.2.2-2023.10-ABDGCSHILM-310
- 3.2.2-2023.10-ABDGCSHILMH-312
- 3.2.2-2023.10-311

> **Note** As of Collibra DQ 2023.09, owl-web, owl-agent, owl-livy, and spark are now dq-web, dq-agent, dq-livy, and dq-spark.

Data Quality & Observability Classic

- 2023.09-ABDGCSHILM-3539
- 2023.09-ABDGCSHILMH-3530
- 2023.09-3529

Spark

- 3.2.2-2023.09-ABDGCSHILM-3539
- 3.2.2-2023.09-ABDGCSHILMH-3530
- 3.2.2-2023.09-3529

Data Quality & Observability Classic

- 2023.08-ABDGCSHILM-3292
- 2023.08-3298

Spark

- 3.2.2-2023.08-ABDGCSHILM-3292
- 3.2.2-2023.08-3298

2023.07.4

Data Quality & Observability Classic

- 2023.07.4-ABDGCSHILM-206
- 2023.07.4-208

Spark

- 3.2.2-2023.07.4-ABDGCSHILM-206
- 3.2.2-2023.07.4-208

2023.07.2

Data Quality & Observability Classic

- 2023.07.2-ABDGCSHILM-3180
- 2023.07.2-3181
- 2023.07.2-ABDGCSHILMH-3182

Spark

- 3.2.2-2023.07.2-ABDGCSHILM-3180
- 3.2.2-2023.07.2-3181

- 3.2.2-2023.07.2-ABDGCSHILMH-3182

2023.07

Data Quality & Observability Classic

- 2023.07-ABDGCSHILM-3156
- 2023.07-3162

Spark

- 3.2.2-2023.07-ABDGCSHILM-3156
- 3.2.2-2023.07-3162

Data Quality & Observability Classic

- 2023.06-ABDGCSHILM-2957
- 2023.06-2958

Spark

- 3.2.2-2023.06-ABDGCSHILM-2957
- 3.2.2-2023.06-2958

2023.05.1

Data Quality & Observability Classic

- 2023.05.1-ABDGCSHILM-258
- 2023.05.1-259

Spark

- 3.2.2-2023.05.1-ABDGCSHILM-258
- 3.2.2-2023.05.1-259

2023.05

Data Quality & Observability Classic

- 2023.05-ABDGCSHILM-254
- 2023.05-253

Spark

- 3.2.2-2023.05-ABDGCSHILM-254
- 3.2.2-2023.05-253

2023.04.2
Data Quality & Observability Classic

- 2023.04.2-ABDGCSHILM-235

Spark

- 3.2.2-2023.04.2-ABDGCSHILM-235

2023.04.1
Data Quality & Observability Classic

- 2023.04.1-LM-215
- 2023.04.1-ABDGCSILM-230
- 2023.04.1-ABDGCSHILM-214
- 2023.04.1-234

Spark

- 3.2.2-2023.04.1-LM-215
- 3.2.2-2023.04.1-ABDGCSILM-230
- 3.2.2-2023.04.1-ABDGCSHILM-214
- 3.2.2-2023.04.1-234

2023.04
Data Quality & Observability Classic

- 2023.04-LM-2480
- 2023.04-ABDGCSILM-2479
- 2023.04-ABDGCSHILM-2478
- 2023.04-2489

Spark

- 3.2.2-2023.04-LM-2480
- 3.2.2-2023.04-ABDGCSILM-2479
- 3.2.2-2023.04-ABDGCSHILM-2478
- 3.2.2-2023.04-2489

Data Quality & Observability Classic

- 2023.03-LM-2341
- 2023.03-ABDGCSILM-2340
- 2023.03-ABDGCSHILM-2339
- 2023.03-2342

Spark

- 3.2.2-2023.03-LM-2341
- 3.2.2-2023.03-ABDGCSILM-2340
- 3.2.2-2023.03-ABDGCSHILM-2339
- 3.2.2-2023.03-2342

Data Quality & Observability Classic

- 2023.02-LM-2188
- 2023.02-ABDGCSILM-2187
- 2023.02-ABDGCSHILM-2186
- 2023.02-2189

Spark

- 3.2.2-2023.02-LM-2188
- 3.2.2-2023.02-ABDGCSILM-2187
- 3.2.2-2023.02-ABDGCSHILM-2186
- 3.2.2-2023.02-2189

Data Quality & Observability Classic

- 2023.01-LM-2067
- 2023.01-ABDGCSILM-2066
- 2023.01-ABDGCSHILM-2065
- 2023.01-2068

Spark

- 3.2.2-2023.01-LM-2067
- 3.2.2-2023.01-ABDGCSILM-2066
- 3.2.2-2023.01-ABDGCSHILM-2065
- 3.2.2-2023.01-2068

Config Map Example

These are the configs to change the versions. For a complete list of versions for Collibra DQ and Spark, refer to the Available Containers section above.

```
--set global.version.dq=2024.01-1234 --set glob-
al.version.spark=3.4.1-2024.01-1234
```

Pull Examples

> **Note** As of Data Quality & Observability Classic 2023.09, owl-web, owl-agent, owl-livy, and spark are now dq-web, dq-agent, dq-livy, and dq-spark.

docker pull https://gcr.io/owl-hadoop-cdh/dq-web:2024.01-ABDGCSHILM-1234

docker pull https://gcr.io/owl-hadoop-cdh/dq-agent:2024.01-ABDGCSHILM-1234

docker pull https://gcr.io/owl-hadoop-cdh/dq-livy:3.4.1-2024.01-ABDGCSHILM-1234

docker pull https://gcr.io/owl-hadoop-cdh/dq-spark:3.4.1-2024.01-ABDGCSHILM-1234

Description

Example: 2024.01-ABDGCSHILM-1234

2024.01: Release number (YYYY.MM).

ABDGCSHILM: A package with optional drivers Athena, BigQuery, Databricks, Google Cloud Storage, Hive, Impala, Livy, and MongoDB.

1234 - A unique number appended to each build.

Default Drivers

Always Available Out of the Box

- SQL Server
- Oracle
- Snowflake
- Redshift
- S3
- ADLS
- PostgreSQL
- MySQL
- Teradata
- Sybase
- Db2
- Dremio

Optional Drivers

Please select drivers

- ABDGCSHILM (Drivers Initial in alphabetical order):
  - A : Athena
  - B : BigQuery
  - D = Databricks
  - GCS : Google Cloud Storage
  - H : Hive

- I : Impala
- L : Livy
- M : MongoDB
- H: Hudi

Installation

System Requirements

Before you install Data Quality & Observability Classic, you need all of the following information to ensure an easy, successful installation process. This section only focuses on the requirements of Collibra DQ and does not take into account the connections to the data sources to ingest data.

Supported Web Browsers

Browser	Version
Google Chrome (recommended)	70.0.3538.102 or newer
Mozilla Firefox	52.8.0 or newer
Safari	12.0.1 or newer

Encryption

Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction

PostgreSQL version

- Collibra DQ comes prepackaged with version 11.4 of PostgreSQL.
- Collibra DQ supports PostgreSQL version 9.6.5 and above if installing the DQ Metastore in an external metastore (recommended).
- Collibra DQ has been tested and validated up to PostgreSQL 15.

> **Note** Collibra recommends installing the DQ Metastore in an external PostgreSQL metastore.

Installation packages/files (BOM)

You can download the DQ installation package from the Collibra Product Resource Center.

> **Note** For additional download options, contact your Account Team.

- demoscripts.tar.gz
- log4j*
- owlcheck
- owl-core-2.1.0-jar-with-dependencies.jar
- owl-webapp-2.1.0.jar
- owl-agent-2.1.0.jar
- setup.sh
- owl-postgres.tar.gz
- notebooks.tar.gz
- owlmanage.sh

Installation-specific requirements

> **Important** You must upgrade to Java 17 and Spark 3.5.3 to install and use Data Quality & Observability Classic 2025.02. For more information about version compatibility, click "Java and Spark compatibility matrix" below.

Java and Spark compatibility matrix[1]

1

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

User permissions

Log in with a user account that has privileges to:

- Create directories
- Launch scripts
- Start java, Spark, and Collibra DQ processes

`SUDO` is required if you are including the PostgreSQL metastore in the installation. `SUDO` is not required if you are using an external PostgreSQL metastore (recommended).

In addition, configure `ULIMIT` settings to 4096 or higher. DQ services typically consume approximately 428 threads, and each DQ job consumes an additional 400 threads. Setting `ULIMIT` to 4096 allows for approximately nine concurrent DQ jobs on a Standalone install.

System requirements

Supported operating systems

- Red Hat Enterprise Linux 8.x
- Red Hat Enterprise Linux 9.x

Hardware requirements

Small Tier - 16 Core, 128G RAM (r5.4xlarge / E16s v3)

Component	RAM	Cores
Web	2g	2
Postgres	2g	2
Spark	100g	10
Overhead	10g	2

Medium Tier - 32 Core, 256G RAM (r5.8xlarge / E32s v3)

Component	RAM	Cores
Web	2g	2
Postgres	2g	2
Spark	250g	26
Overhead	10g	2

Large Tier - 64 Core, 512G RAM (r5.16xlarge / E64s v3)

Component	RAM	Cores
Web	4g	3
Postgres	4g	3
Spark	486g	54
Overhead	18g	4

> **Important** Collibra DQ requires a limit of 2TBs for large tier jobs. For DQ jobs that exceed 2TBs, you must filter down columns or rows.

Estimates

Sizing should allow headroom and based on peak concurrency and peak volume requirements. If concurrency is not a requirement, you just need to size for peak volume (largest tables). Best practice to efficiently scan is to scope the job by selecting critical columns. See JDBC vs Local Data for more information.

Keep in mind that Spark allocates overhead memory on top of the requested memory. For example, an executor pod configured with 6GB of memory may actually request 6.8GB of memory. For more information, go to Apache Spark Memory Management.

Bytes per Cell	Rows	Columns	Gigabytes	Gigabytes for Spark (3x)
16	1,000,000.00	25	0.4	1.2
16	10,000,000.00	25	4	12
16	100,000,000.00	25	40	120
16	1,000,000.00	50	0.8	2.4
16	10,000,000.00	50	8	24
16	100,000,000.00	50	80	240
16	1,000,000.00	100	1.6	4.8
16	10,000,000.00	100	16	48
16	1,000,000,000.00	100	1600	4800
16	100,000,000.00	100	160	480
16	1,000,000.00	200	3.2	9.6
16	10,000,000.00	200	32	96
16	100,000,000.00	200	320	960
16	1,000,000,000.00	200	3200	9600

Cluster

If your program requires more horsepower or (Spark) workers than the example tiers above which is fairly common in Fortune 500 companies than you should consider the horizontal and ephemeral scale of a cluster. Common examples include Amazon EMR and

Cloudera CDP. Collibra DQ is built to scale up horizontally and can scale to hundreds of nodes.

Network requirements

Default Ports used by Collibra DQ

Port	Description
443	Connectivity to data sources (databases, file storage, etc.)
5432	DQ Metastore (PostgreSQL). Enables DQ Web, DQ Agent, and Spark services to connect to the DQ Metastore.
9000	DQ Web
9101	Exposes the Health Check API to check that the DQ Agent is running and stable.

Other

- If your current Spark version is 3.2.2 or older, Collibra strongly recommends upgrading to Spark 3.4.1 to address various critical vulnerabilities present in Spark core library, including Log4J. To determine which Spark version you are using, sign into your Collibra DQ instance and click the ⓘ in the upper-right corner of any page. The **Spark Version** lists your current Spark version.
    - If you are not already using Spark 3.4.1, follow the steps outlined in Upgrading Spark versions.

User permissions

Prerequisites

- Kubernetes cluster -- EKS, GKE, AKS, Openshift, Rancher
- Helm(v3)
- kubectl
- Cloud command line SDK, such as gcloud CLI, AWS CLI or similar
- External PostgreSQL DB version 11.9 and above, storage size 100GB, cores 4 to 8 memory to 4 to 8 GB
- Private container registry -- to store images
- LoadBalancer -- IngressController -- Ingress
- Egress networking access
- Helm Chart
- Images, image access key
- Minimum pod requirement -- 2 cores, 2GB RAM
- If you bring in your own Spark executor pod launch template, ensure that the service account used to launch Spark executor pods has the permission to do so. Refer to the executor launch template for more information.

System requirements

Supported Kubernetes versions

Data Quality & Observability Classic supports Kubernetes versions 1.29 through 1.31.

> **Note** As of February 2025, we recommend upgrading to Kubernetes version 1.29 or newer, as version 1.28 has reached its end of life.

Application system requirements

Component	Processor	Memory	Storage
Collibra DQ Web	1 core	2 GB	10 MB PVC
DQ Agent	1 core	1 GB	100 MB PVC
DQ Metastore	1 core	2 GB	10 GB PVC
Spark*	2 cores	2 GB	-

Keep in mind that Spark allocates overhead memory on top of the requested memory. For example, an executor pod configured with 6GB of memory may actually request 6.8GB of memory. For more information, go to Apache Spark Memory Management.

> **Note** * This is the minimum quantity of resources required to run an a Spark job in Kubernetes. This amount of resources would only provide the ability to scan a few megabytes of data with no more than a single job running at a given time. Proper sizing of the compute space must take into account the largest dataset that may be scanned, as well as the desired concurrency.

Network service considerations

DQ Web is the only required component that needs to be directly accessed from outside of Kubernetes. History Server is the only other component that can be accessed directly by users, however, it is optional.

If the target Kubernetes platform supports a LoadBalancer service type, you can configure the Helm Chart to directly deploy the externally accessible endpoint.

> **Note** For testing purposes, you can also configure the Helm chart to deploy a NodePort service type.

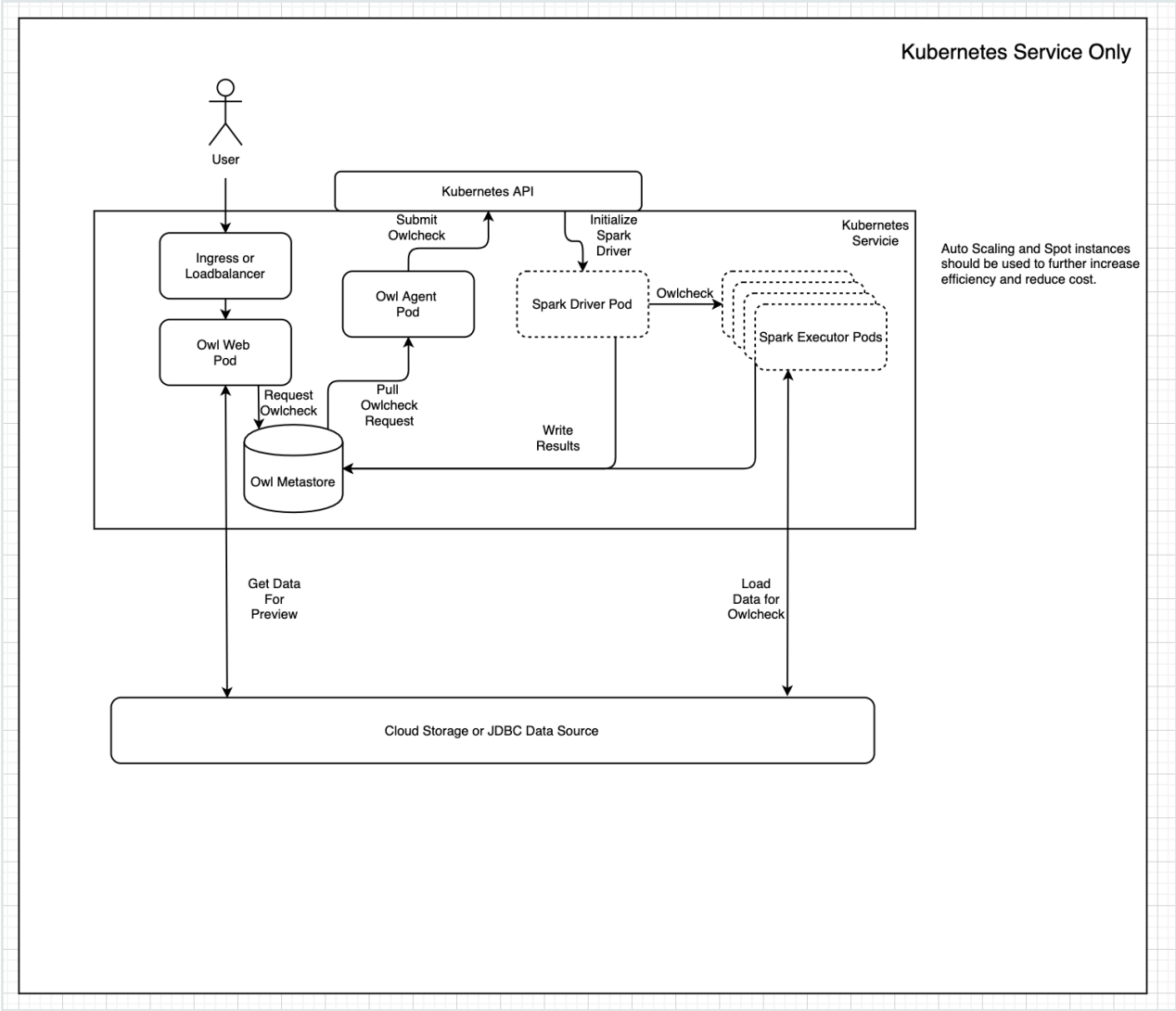
For the Ingress service type, deploy Collibra DQ without an externally accessible service and then attach the Ingress service separately. This applies when you use a third-party Ingress controller such as NGINX, Contour, etc.

> **Note** The Helm Chart is able to deploy an Ingress on GKE and EKS platforms, however, there is a wide variety of possible Ingress configurations that have not been tested.

Installing Data Quality & Observability Classic on Self-hosted Kubernetes

Data Quality & Observability Classic wholeheartedly embraces the principles of cloud native technologies in its design and deployment. Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds.

The diagram below depicts Data Quality & Observability Classic's cloud native deployment architecture:



In this form factor, you can deploy Collibra DQ in any public or private cloud while maintaining a consistent experience, performance, and management runbook.

Collibra DQ microservices

To achieve cloud native architecture, Collibra DQ is decomposed into several components, each of which is deployed as a microservice in a container.

Component	Microservice	Description
DQ Web	dq-web	The main point of entry and interaction between Collibra DQ and end users or integrated applications. DQ Web provides both a rich, interactive user experience and a robust set of APIs for automated integration.
DQ Agent	dq-agent	You can think of the Agent as the "foreman" of Collibra DQ. When a user or application requests a data quality check through DQ Web, DQ Agent will marshal compute resources to perform the work. DQ Agent does not actually do any of the data quality work. Instead, it translates the request submitted by DQ Web into a technical descriptor of the work that needs to be done and then launches the requested DQ job.
DQ Metastore	PostgreSQL	Where Collibra DQ stores all the metadata, statistics, and results of DQ jobs. It is also then main point of communication between DQ Web and DQ Agent. The metastore also contains the results of DQ jobs performed by transient containers (workers) in the compute space. Collibra recommends installing the DQ Metastore in an external PostgreSQL metastore.
Apache Spark	Apache Spark	The distributed compute framework that powers the Collibra DQ data quality engine. Spark enables DQ jobs to rise to the task of data quality on Terabyte scale datasets. Spark containers are completely ephemeral and only live for as long as necessary to complete a given DQ job.
Apache Livy	Apache Livy	The Session Manager that enables Collibra DQ to browse HDFS, S3, GCS, or Azure Data Lake (ADL). Interacts with Object Stores (similar to JDBC sources Explorer) to perform tasks such as estimating jobs and getting days with data.

Containerization

The binaries and instruction sets described in each of the Collibra DQ microservices are encompassed within Docker container images. Each of the images is versioned and maintained in a secured cloud container registry repository. To initiate a Collibra DQ cloud

native deployment, you must first obtain credentials to either pull the containers directly or download them to a private container registry.

> **Warning** Support for Collibra DQ cloud native deployment is limited to deployments using the containers provided from the Collibra container registry.
>
> Reach out to your customer contact for access to pull the Collibra containers.

Kubernetes

Kubernetes is a distributed container scheduler and has become synonymous with cloud native architecture. While Docker containers provide the logic and runtime at the application layer, most applications still require network, storage, and orchestration between multiple hosts in order to function. Kubernetes provides all of these facilities while abstracting away all of the complexity of the various technologies that power the public or private cloud hosting the application.

Collibra DQ Helm chart

While Kubernetes currently provides the clearest path to gaining the benefits of a cloud native architecture, it is also one of the more complex technologies in existence. This has less to do with Kubernetes itself and more with the complexity of the constituent technologies it is trying to abstract. Technologies like attached distributed storage and software defined networks are entire areas of specialization that require extensive expertise to navigate. Well implemented Kubernetes platforms hide all of this complexity and make it possible for anyone to leverage these powerful concepts. However, a robust application like Collibra DQ requires many descriptors (K8s manifests) to deploy its various components and all of the required supporting resources like network and storage.

This is where Helm comes in. Helm is a client side utility (since v3) that automatically generates all the descriptors needed to deploy a cloud native application. Helm receives instructions in the form of a **Helm chart** that includes templated and parameterized versions of Kubernetes manifests. Along with the Helm chart, you can also pass arguments like names of artifacts, connection details, enable and disable commands, and so on. Helm resolves the user defined parameters within the manifests and submits them to Kubernetes for deployment. This

enables you to deploy the application without necessarily having a detailed understanding of the networking, storage or compute that underpins the application.

> **Note** For details on configuring Helm charts for Collibra DQ, go to Deploy on Self-hosted Kubernetes.

What's next?

- Before You Install on self-hosted Kubernetes
- Install on self-hosted Kubernetes

Before You Install on Self-hosted Kubernetes

Prerequisites

- Review the self-hosted Kubernetes architecture described in Installing Data Quality & Observability Classic on Self-hosted Kubernetes.
- Verify that your system meets the System Requirements.

Obtain credentials

Kubernetes stores credentials in the form of secrets. Secrets are base64 encoded files that you can mount into application containers and that application components can reference at runtime. You use *pull secrets* to access secured container registries to obtain application containers.

> **Note** Deploying containers directly from the Collibra image repository is not recommended. You should only access the Collibra image registry for the initial download and validation of Docker images. After this, you should upload and store images to *your private registry* to provide you control over when the images are updated and eliminate any operational dependencies on Collibra's repository.

SSL certificates

To enable SSL for secure access to DQ Web, a keystore that contains a signed certificate, keychain, and private key is required. This keystore must be available in the target namespace before you deploy Collibra DQ.

> **Note** By default, Collibra DQ looks for a secret called `dq-ssl-secret` to find the keystore.

> **Note** Although it is possible to deploy with SSL disabled, is not recommended.

Cloud storage credentials

If you enable History Server, a distributed filesystem is required. Currently, Collibra DQ supports S3 and GCS for Spark history log storage.

> **Note** Azure Blob and HDFS on the near term roadmap.

Target storage system	Credentials requirements
S3	An IAM Role with access to the target bucket needs to be attached to the Kubernetes nodes of the namespace where Collibra DQ is being deployed.
GCS	You must create a secret from the JSON key file of a service account with access to the log bucket. The secret must be available in the namespace before you deploy Collibra DQ. By default, Collibra DQ looks for a secret called `spark-gcs-secret`, if GCS is enabled for Spark history logs. You can change this via a helm chart argument.

Container pull secret

Data Quality & Observability Classic containers are stored in a secured repository in Google Container Registry. For Collibra DQ to successfully pull the containers when deployed, a pull secret with access to the container registry must be available in the target namespace.

> **Note** By default, Collibra DQ looks for a pull secret named `dq-pull-secret`. You can change this via a helm chart argument.

Spark service account

To enable DQ Agent and the Spark driver to create and destroy compute containers, you must have a service account with a role that allows get/list/create/delete operations on pods/services/secrets/configMaps in the target namespace. By default, Collibra DQ attempts to create the required service account and the required RoleBinding within the scope of the namespace to the default Edit role available at the Kubernetes cluster. Edit is a role that is generally available in a Kubernetes cluster. If the Edit role is not available, you can manually create it.

If you manually create a role, be sure to include the patch verb on the system resources. Use the following example as a guide:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: dq-role-example
  namespace: dq-namespace-example
rules:
- apiGroups: [""]
  resources: [pods, services]
  verbs: [get, list, watch, create, update, patch, delete,
deletecollection]
- apiGroups: ["apps"]
  resources: [deployments]
  verbs: [get, list, watch, create, update, patch, delete]
- apiGroups: [""]
  resources: [secrets]
  verbs: [list, create, delete, deletecollection, patch, update,
get]
- apiGroups: [""]
  resources: [pods/attach, pods/exec, pods/portforward, pod-
s/proxy, services/proxy]
  verbs: [list, create, delete, deletecollection, patch, update,
get, watch]
- apiGroups: [""]
  resources: [configmaps]
  verbs: [create, delete, deletecollection, get, list, patch,
update, watch]
- apiGroups: [""]
  resources: [pods/log]
  verbs: [get, list, watch]
```

Configure access to the target platform

To deploy anything to a Kubernetes cluster, the first step is to install the required client utilities and configure access:

- **kubectl**: The main method of communication with a Kubernetes cluster. All configuration or introspection tasks will be preformed using kubectl.
- **helm v3**: Used to deploy the Collibra DQ Helm chart without hand coding manifests.

After you install the utilities, the next step is to configure a kube-context that points to and authenticates to the target platform. On cloud platforms like GKE and EKS, this process is completely automated through their respective CLI utilities.

```
aws eks --region <region-code> update-kubeconfig --name
<cluster_name>
```

```
gcloud container clusters get-credentials <cluster-name>
```

In private clouds, this process will vary from organization to organization, however, the platform infrastructure team should be able to provide the target kube-context entry.

Prepare secrets

Once access to the target platform is confirmed, you can prepare the namespace.

```
kubectl create namespace <namespace>
```

> **Note** The namespace may already be allocated by the platform team.

Create a container pull secret

> **Note** For details on how to install Collibra DQ on Kubernetes with docker containers, see Install on Self-hosted Kubernetes .

JSON key file credential

```
kubectl create secret docker-registry dq-pull-secret \
--docker-server=<cdq-registry-server> \
--docker-username=_json_key \
--docker-email=<service-account-email> \
--docker-password="$(cat /path/to/key.json)" \
--namespace <namespace>
```

Short lived access token

```
kubectl create secret docker-registry dq-pull-secret \
--docker-server=<cdq-registry-server> \
--docker-username=oauth3accesstoken \
--docker-email=<service-account-email> \
--docker-password="<access-token-text>" \
--namespace <namespace>
```

Warning GCP Oauth tokens are usually only good for 1 hour. This type of credential is excellent if the goal is to pull containers into a private registry. It can be used as the pull secret to access containers directly, however, the secret would have to be recreated with a fresh token before restarting any of the Collibra DQ components.

Create a GCS credential secret

```
kubectl create secret generic spark-gcs-secret \
--from-file /path/to/keystore.jks \
--namespace <namespace>
```

Warning The file name that you use in the `--from-file` argument should be *spark-gcs-secret*. If the file name is anything else, you must include an additional argument specifying the gcs secret name in the Helm command.

What's next?

Install on self-hosted Kubernetes

Install on Self-hosted Kubernetes

After performing the steps in Before You Install on Self-hosted Kubernetes, you are ready to install Collibra DQ on self-hosted Kubernetes.

Sign in to the Kubernetes cluster

1. Sign in to the Kubernetes cluster from a Linux compatible terminal.
2. Create a namespace in the cluster using the following command:

```
kubectl create namespace <owldq>
```

Pull images from the Collibra registry

Collibra DQ images are located in the Google Artifact Registry (http://gcr.io/). To access the images, you need a key file (`repo-key.json`). Contact Collibra Support to obtain a key file for your organization. Collibra strongly recommends that you designate a key file recipient to obtain this file and keep it secure.

> **Note** For more information on docker images related to DQ build packages, go to Builds.

1. Run the following command:

```
docker login -u _json_key -p "$(cat repo-key.json)"
https://gcr.io
```

> **Note** Contact Collibra Support to obtain a key file (`repo-key.json`) for your organization.

2. To pull the images, run the following docker pull commands:

```
docker pull gcr.io/owl-hadoop-cdh/dq-agent:<version and
build tag provided by Collibra>
docker pull gcr.io/owl-hadoop-cdh/dq-web:<version and build
tag provided by Collibra>
docker pull gcr.io/owl-hadoop-cdh/dq-spark:<version and
build tag provided by Collibra>
docker pull gcr.io/owl-hadoop-cdh/dq-livy:<version and build
tag provided by Collibra>
```

Push images into your private registry

1. Sign in to your private container registry.
2. Tag and push the images from Collibra to your private registry, by using the following commands:

```
docker tag gcr.io/owl-hadoop-cdh/dq-web:2023.11
<registryURL>/dq-web:2023.11
docker push <registryURL>/dq-web:2023.11
```

Example:

```
docker tag [OPTIONS] IMAGE [:TAG][REGISTRYHOST/][USERNAME/]NAME
[:TAG]
docker tag push NAME[:TAG]
```

Create an SSL keystore secret

To create an SSL keystore secret, run the following command:

```
kubectl create secret generic dq-ssl-secret \
--from-file /path/to/keystore.jks \
--namespace <namespace>
```

Warning The file name that you use in the `--from-file` argument should be *keystore.jks*. If the file name is anything else, you must include an additional argument specifying the keystore file name in the Helm command.

Create a pull secret

> **Note** Deploying containers directly from the Collibra image repository is not recommended. You should only access the Collibra image registry for the initial download and validation of Docker images. After this, you should upload and store images to *your private registry* to provide you control over when the images are updated and eliminate any operational dependencies on Collibra's repository.

To create a pull secret, run the following command:

```
kubectl create secret docker-registry dq-pull-secret \
--docker-server=<cdq-registry-server> \
--docker-username=_json_key \
--docker-email=<service-account-email> \
--docker-password="$(cat /path/to/key.json)" \
--namespace <namespace>
```

> **Note** If your private registry is used for images and if they are accessible from within the Kubernetes cluster, this secret need not be created. If credentials are required to access your private registry, create this secret by modifying the docker-server URL and docker-password.

What's next?

Deploy on self-hosted Kubernetes

Deploy on Self-hosted Kubernetes

After you have performed the steps in Install on self-hosted Kubernetes, you can begin the deployment of Collibra DQ.

Download Helm Chart Files

After you have your Collibra DQ license, you will receive an email from Collibra that includes the Helm Charts as zip files. Unzip the files on a Linux-compatible deployment location.

> **Note** For background information on how Collibra DQ uses Helm charts, go to Installing Data Quality & Observability Classic on Self-hosted Kubernetes.

The root directory, `dq`, contains the following directories and files:

- `drwxrwxr-x -- templates`. Directory that contains a set of YAML files, including:
  - k8s-enpoint.secret.yaml
  - _helpers.tpl
  - hadoop-conf.yaml
  - serviceaccount.yaml
  - krb5-conf.yaml
  - rbac.yaml
  - dq-secret.yaml
  - keytab-secret.yaml
- `drwxrwxr-x -- charts`. Directory that contains details for the core containers, including.
  - metastore
  - owl-livy
  - owl-agent
  - owl-web
  - spark-history-server
- `-rw-rw-r-- Chart.yaml`. File that contains the DQ Helm chart meta-information.
- `-rw-rw-r-- values.yaml`. File that contains the DQ configuration values.

You may pass parameters using one of two methods:

- Using the `values.yaml` file (or)
- Using the Helm set commands.

> **Note** The Helm set commands take precedence over the `values.yaml` file.

Minimal Install Settings

This example provided in this section installs the DQ Web, DQ Agent, and DQ Metastore. Collibra DQ is inaccessible until you manually add an Ingress or another type of externally accessible service.

Warning All of the following examples will pull containers directly from the Collibra DQ secured container registry. In most cases, InfoSec policies require that containers are sourced from a private container repository controlled by the local Cloud Ops team. Make sure to to add `--set global.image.repo=</url/of/private-repo>` so that you use only approved containers.

Note The DQ Metastore container must start first as the other containers use it to write data. On your initial deployment, the other containers might start before the metastore and fail.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=ClusterIP \
--set global.image.repo=<pathTolmageRepo> \
<deployment-name> \
/path to the helm chart root folder>
```

Value	Description
`<namespace>`	Enter the namespace that you created for this deployment.
`<cdq-version>`	Enter the version from the web image suffix. For example, **2023.11** from the image, **dq-web:2023.11**.
`<cdq-spark-version>`	Enter the Spark version from the Spark image suffix. For example, **3.4.1-2023.11** from the image, **spark:3.4.1-2023.11**.
`<cdq-license-key>`	Enter the license key provided to you by Collibra.
`<your-license-name>`	Enter the license name provided to you by Collibra.
`${email}`	Enter the default admin user email associated with the admin account.

Value	Description
`${password}`	Enter the default admin user password for the admin account. The password must adhere to the following password policy: • A minimum of 8 characters. • A maximum of 72 characters. • At least one upper-case character. • At least one numeric character. • At least one supported special character (`!@#%$^&*?_~`). • Cannot contain the user ID (admin). **Note** If a password that does not meet the password policy is entered, the install process proceeds as though the password is accepted, but the admin user becomes locked out. If this occurs, rerun the Helm command with a password that meets the password policy and restart the web pod.
`<pathTolmageRepo>`	This is your private registry key, where the Collibra images are available. When this is not provided, you will pull the images from the Collibra image registry, for which you should create a pull secret with the repo key provided by Collibra. See Install on Self-hosted Kubernetes for more details.
`<deployment-name>`	Any name of your choice for this deployment.

Note If you optionally pass credentials for the Postgres Metastore, ensure that you do not use the `$` symbol in the `global.metastore.pass` variable, as it is an unsupported special character for Postgres Metastore passwords.

The number of possible customizations is quite extensive and provides a great deal of flexibility for a wide variety of platforms. However, when deploying on a known platform (including EKS, GKE, and AKS), the number of required inputs is limited. In common cases, run a single CLI command including basic parameters like disable history server, configure the storage bucket for logs, specify the image repository, and so on.

Including an Externally Accessible Service

The following examples install the Collibra DQ with an externally accessible service.

Minimal Install with an Externally Accessible Service

The following example performs the minimal install and adds a preconfigured NodePort or LoadBalancer service to provide access to the Web.

> **Warning** A LoadBalancer service type requires that the Kubernetes platform is integrated with a Software Defined Network solution. This will generally be true for the Kubernetes services offered by major cloud vendors. Private cloud platforms more commonly use Ingress controllers. Check with the infrastructure team before attempting to use LoadBalancer service type.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=<NodePort || LoadBalancer> \
<deployment-name> \
/path/to/chart/dq
```

Install with SSL Enabled

The following example performs the install with an externally accessible service, but with SSL enabled.

> **Note** Ensure you have already deployed a keystore containing a key to the target namespace with a secret name that matches the `global.web.tls.key.secretName` argument (*dq-ssl-secret* by default). Also, ensure that the secret's key name matches the `global.web.tls.key.store.name` argument (*dqkeystore.jks* by default).

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=<NodePort || LoadBalancer> \
--set global.web.tls.enabled=true \
--set global.web.tls.key.secretName=dq-ssl-secret \
--set global.web.tls.key.alias=<key-alias> \
--set global.web.tls.key.type=<JKS || PKCS12> \
--set global.web.tls.key.pass=<keystore-pass> \
--set global.web.tls.key.store.name=keystore.jks \
<deployment-name> \
/path/to/chart/dq
```

Install on OpenShift 4.x with SSL Enabled

The following example performs the install with an externally accessible service on an OpenShift project with `RunAsAny` enforced, with SSL enabled.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=<NodePort || LoadBalancer> \
--set global.web.tls.enabled=true \
--set global.web.tls.key.secretName=dq-ssl-secret \
--set global.web.tls.key.alias=<key-alias> \
--set global.web.tls.key.type=<JKS || PKCS12> \
--set global.web.tls.key.pass=<keystore-pass> \
--set global.web.tls.key.store.name=keystore.jks \
--set global.security.securityContextConstraint.runAsAny=true \
<deployment-name> \
/path/to/chart/dq
```

Install with History Server for GCS Log Storage

The following example performs the install with an externally accessible service and Spark History Server enabled. In this example, the target log storage system is GCS.

> **Note** For Collibra DQ to be able to write Spark logs to GCS, create a secret from the JSON key file of a service account that has access to the log bucket. For more detailed information, see System Requirements.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=<NodePort || LoadBalancer> \
--set global.spark_history.enabled=true \
--set global.spark_history.logDirectory=gs://logs/spark-history/
\
--set global.spark_history.service.type=<NodePort || LoadBal-
ancer> \
--set global.cloudStorage.gcs.enableGCS=true \
<deployment-name> \
/path/to/chart/dq
```

Install with History Server for S3 Log Storage

The following example performs the install with an externally accessible service and Spark History Server enabled. In this example, the target log storage system is S3.

> **Note** For Collibra DQ to be able to write Spark logs to S3, make sure that an Instance Profile IAM Role with access to the log bucket is attached to all nodes serving the target namespace. For more detailed information, see System Requirements.

> **Important** We currently only support one set of S3 credentials to be configured in the Spark Server at a time within a single DQ Job. If you are using a Spark History Server

bucket and an S3 connection, the S3 credentials must also have access to the S3 Spark History Server bucket.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=<NodePort || LoadBalancer> \
--set global.spark_history.enabled=true \
--set global.spark_history.logDirectory=s3a://logs/spark-
history/ \
--set global.spark_history.service.type=<NodePort || LoadBal-
ancer> \
--set global.cloudStorage.s3.enableS3=true \
<deployment-name> \
/path/to/chart/dq
```

Install with External DQ Metastore

The following example performs the install with an externally accessible service and an external metastore, for example AWS RDS, Google Cloud SQL, or just PostgresSQL on its own instance.

Warning Collibra DQ currently supports PostgreSQL 9.6 and newer.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=<NodePort || LoadBalancer> \
--set global.metastore.enabled=false
--set global.configMap.data.metastore_url-
l=jdbc:postgresql://<host>:<port>/<database>
--set global.configMap.data.metastore_user=<user> \
--set global.configMap.data.metastore_pass=<password> \
<deployment-name> \
/path/to/chart/dq
```

Warning The `$` symbol is not a supported special character in your Postgres Metastore password.

Install with History Server for Windows Azure Storage Blob (WASB) Log Storage

The following example performs the install with an externally accessible service and Spark History Server enabled. In this example, the target log storage system is Windows Azure Storage Blob (WASB). However, wherever you see `azureblobstorage`, you can insert your own storage solution (for example, ADLS, S3, or GCS).

Note To access Azure Blob Storage, you must have the correct permissions to the Storage Account Key. For more information, go to Connecting to Azure Blob Storage.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.configMap.data.license_name=<your-license-name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=<NodePort || LoadBalancer> \
--set global.spark_history.enabled=true \
--set global.spark_history.logDirectory=wasbs://spark-history-
logs@azureblobstorage.blob.core.windows.net/ \
--set global.spark_history.service.type=<NodePort || LoadBal-
ancer> \
--set global.cloudStorage.wasbs.enableWASBS=true \
--set global.cloudStorage.wasbs.storageContainerName=spark-his-
tory-logs \
--set glob-
al.cloudStorage.wasbs.storageAccountName=azureblobstorage \
--set glob-
al.cloudSt-
orage.wasbs.storageAccountKey=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
<deployment-name> \
/path/to/chart/dq
```

Troubleshooting

Troubleshooting self-hosted Kubernetes Install

What's next?

- Securely pass sensitive values to the Data Quality & Observability Classic Helm Chart
- Configure custom JDBC data source drivers
- Configure Spark scratch disk space to handle large datasets
- Add CA certificates to the java keystore

Securely pass sensitive values to the Data Quality & Observability Classic Helm Chart

This section shows you how to securely pass sensitive data to the Collibra DQ Helm Chart through Kubernetes or Microsoft Azure Key Vault (AKV) Secrets. Passing sensitive data, such

as PostgreSQL Metastore credentials and the license key, through Kubernetes or AKV Secrets helps mitigate potential security risks to your organization by preventing the exposure of such values in the Helm Chart.

Passing Kubernetes Secrets to the values.yaml file in the Collibra DQ Helm Chart

> **Note** You can use the same Secret to pass additional sensitive data to your Data Quality & Observability Classic deployment.

1. Create a Kubernetes Secret containing the sensitive data, as shown in the example below.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: kubernetes-secret-name
type: Opaque
data:
  username: Cg==
  password: Cg==
```

2. Pass the Secret to values.yaml file in your Collibra DQ Helm Chart, as shown in the example below.

```
global:
  configMap:
    data:
      metastore_secret_name: "kubernetes-secret-name"
      metastore_secret_key_map:
        username: SPRING_DATASOURCE_USERNAME
        password: SPRING_DATASOURCE_PASSWORD
        # Add more key maps to pass additional sensitive
data to the DQ web and agent pods.
```

3. If you are updating an existing deployment of Collibra DQ, restart the DQ web and agent pods to complete the updates.

Passing Azure Key Vault secrets to the values.yaml file in the Collibra DQ Helm Chart

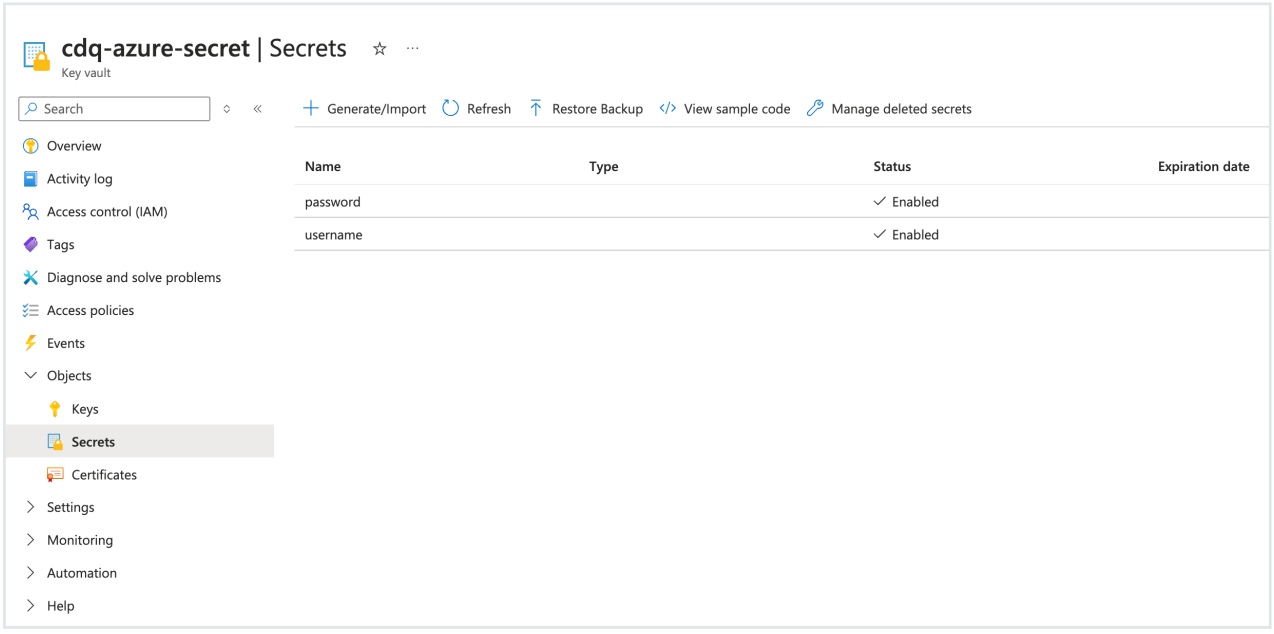
> **Note** You can use the same secrets to pass additional sensitive data to your Data Quality & Observability Classic deployment.

Prerequisites

- You have an Azure Key Vault secret containing sensitive secret objects.
- Your Azure Secret Storage CSI driver is installed on the Azure Kubernetes Service.
- You have grant access to the Azure Key Vault through Managed Identity.
- The base UBI 9 Micro image is available in the same container repository as where the Collibra DQ container images are stored.

  > **Note** You can pull this image from the same GCR credentials used for Collibra DQ containers.

1. Create an Azure Key Vault secret containing the sensitive data, as shown in the screenshot below.



2. Pass the following information to values.yaml file in your Collibra DQ Helm Chart, as shown in the example below.

a. Azure Cloud details, including:
    i. Tenant ID
    ii. Azure-managed identity ID given the grant access on the AKV
    iii. Resource Group
    iv. Subscription ID
b. Secrets from the AKV

```
global:
  configMap:
    data:
      metastore_secret_name: "kubernetes-secret-name"
      metastore_secret_key_map:
        username: SPRING_DATASOURCE_USERNAME
        password: SPRING_DATASOURCE_PASSWORD
        # Optionally add more key maps to pass additional
sensitive data to the DQ web and agent pods.
  vault:
    enabled: true
    provider: "akv" # akv is the Azure Key Vault
    vault_name: "cdq-azure-secret" # Azure Vault storage
name to create a K8s object of SecretProviderClass. Make
sure to set global.configMap.data.metastore_secret_name to
secretName value for the sync.
    vault_object_map: # Map Azure Vault object names with
the K8s Secret object key names, Azure Vault object name as
the key, and the K8s Secret object key as the value.
      username: username # The value should have a
corresponding key entry at global.configMap.data.metastore_
secret_key_map
      password: password # The value should have a
corresponding key entry at global.configMap.data.metastore_
secret_key_map
    tenant_id: ""
    identity_id: "" # Azure managed identity id
    resource_group: ""
    subscription_id: ""
    image:
      name: ubi9-micro
      version: latest
      pullPolicy: IfNotPresent
```

3. If you are updating an existing deployment of Collibra DQ, restart the DQ web and agent pods to complete the updates.

Configure Spark scratch disk space to handle large datasets

When you run large datasets, your Spark cluster's resources need to be sized appropriately to handle the computation of DQ rules. This section shows you how to configure on-demand Spark scratch disk space to handle huge Spark jobs.

Prerequisites

Before you configure Spark scratch disk space, ensure you have:

- A Collibra DQ deployment on a cloud native platform, such as EKS, GKS, and so on.
- A Kubernetes service account for Collibra DQ that meets the permission requirements to provision dynamic Persistent Volume Claims (PVCs) in ReadWriteOnce (RWO).
- Execute permissions on the **/tmp** directory or the directory used for temporary storage, to ensure Spark jobs launch successfully.

> **Note** This capability is only available for Collibra DQ versions 2023.02 or newer.

Steps

1. Open a terminal session.
2. Set the following values to deploy or update the Collibra DQ app with the provided Helm charts:

```
--set global.configMap.data.spark_scratch_type-
e="persistentVolumeClaim" --set global.configMap.data.spark_
scratch_storage_class="standard" --set glob-
al.configMap.data.spark_scratch_storage_size="20Gi" --set
global.configMap.data.spark_scratch_local_path-
h="/tmp/scratch"
```

> **Note** The values in the previous example are settings for GKE-based deployments.

   a. If your Collibra DQ app is managed by an alternative Kubernetes package manager, ensure you update the owl-agent-configmap with the following variables:

```
KUBERNETES_SCRATCH_TYPE: persistentVolumeClaim
KUBERNETES_SCRATCH_STORAGE_CLASS: standard
KUBERNETES_SCRATCH_STORAGE_SIZE: 20Gi
KUBERNETES_SCRATCH_LOCAL_PATH: /tmp/scratch
```

> **Note** Pick the value of Storage Class, Storage Size and mouthPath specific to your requirements.

b. Restart the DQ agent pod to complete the updates.

3. Sign in to Collibra DQ.
4. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
5. Click **Agent Configuration**.
   » The Agent Configuration page opens.
6. Click the **Actions**, then click **Edit**.
   » The Edit Agent modal opens.
7. In the **Free form (Appended)** text box option, enter the following string value:

```
-conf spark.kuber-
netes.driver.podTemplateFile=local:///opt/owl/config/k8s-
driver-
tem-
plate.ym-
l,s-
park.kuber-
netes.executor.podTemplateFile=local:///opt/owl/config/k8s-
executor-
tem-
plate.ym-
l,spark.kubernetes.driver.reusePersistentVolumeClaim=false
```

> **Tip** If you add any additional configuration properties based on your specific requirements, separate them with commas.

8. Click **Submit**.
9. Run a sample DQ job to process DQ rules against a dataset of your choice. The corresponding Spark job dynamically provisions PVCs for the driver and each executor to

handle large datasets by juggling between Memory and Disk at the mouth path of the PVC.

What's next?

For more information, go to the official Spark documentation on:

- Spark on Kubernetes using Pod Templates.
- Spark on Kubernetes using Persistent Volume Claims.

Configure custom JDBC data source drivers

This section shows you how to set up persistent external volumes on existing cloud native deployments of Collibra DQ to support the use of custom JDBC data source drivers.

Steps

1. Create a Persistent Volume (PV) and a Persistent Volume Claim (PVC) in the same Kubernetes cluster namespace where your cloud native deployment of Collibra DQ is running with `ReadWriteMany` access mode, as shown in the following example:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-pv
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  nfs:
    server: nfs-server.nfs.svc.cluster.local
    path: "/"

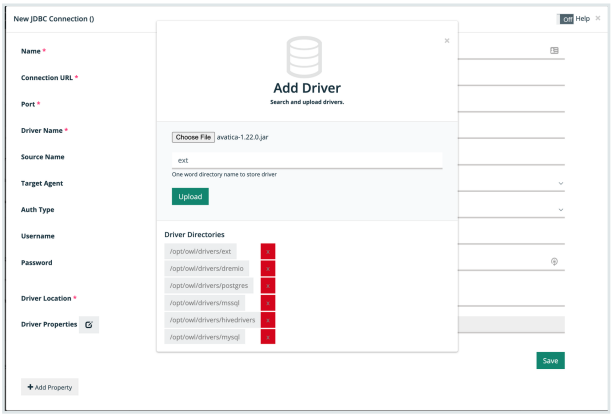
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: ""
  resources:
    requests:
      storage: 10Gi
```

> **Tip** You can choose any storage class that supports ReadWriteMany access mode only.

2. Patch the existing DQ Web and Agent StatefulSets with an additional Volume Mount, as shown in the following example:

```
spec:
  volumes:
    - name: owldq-ext-jdbc-jars
      persistentVolumeClaim:
        claimName: nfs-pvc
...
...
  containers:
    - volumeMounts:
        - name: owldq-ext-jdbc-jars
          mountPath: /opt/owl/drivers/ext
```

3. From the Connections page, click **Add** on the **Generic JDBC Connection** tile.
   » The **New JDBC Conection** modal appears.
4. Enter the required information.
5. Click the folder icon in the **Driver Location** option.
   » The **Add Driver** modal appears.
6. Click **Choose File** and select the JAR file to upload.
7. Enter a one-word directory name to store the driver, for example, *ext*.
8. Select `/opt/owl/drivers/ext` under **Driver Directories** to upload your driver to that location, as shown in the image below.



9. Click **Upload**.
   » A success message appears when your driver is successfully added.

> **Note**
> Uploads from DQ Web only work if the PVC's storageclass supports the fsGroup security context of Kubernetes. If the fsGroup security context is not supported, create a temporary dummy pod running as a root user with the PVC mounted, and copy the required JDBC jars to the mount location with 544 permissions.

10. Go to the **Remote Agent** section of the Admin Console and click the pencil icon to edit your agent.
    » The **Edit Agent** modal appears.
11. Add the following properties to the **Free Form (Appended)** option, as shown in the following example:

```
-conf
spark.kubernetes.driver.podTemplateFile=local:///opt/owl/
config/k8s-driver-
template.yml,spark.kubernetes.executor.podTemplateFile=lo
cal:///opt/owl/config/k8s-executor-
template.yml,spark.kubernetes.executor.volumes.persistent
VolumeClaim.nfs-pv.options.claimName=nfs-
pvc,spark.kubernetes.executor.volumes.persistentVolumeCla
im.nfs-
pv.options.sizeLimit=10Gi,spark.kubernetes.executor.volum
es.persistentVolumeClaim.nfs-
pv.mount.path=/opt/owl/drivers/ext,spark.kubernetes.execu
tor.volumes.persistentVolumeClaim.nfs-
pv.mount.readOnly=false,spark.kubernetes.driver.volumes.p
ersistentVolumeClaim.nfs-pv.options.claimName=nfs-
pvc,spark.kubernetes.driver.volumes.persistentVolumeClaim
.nfs-
pv.options.sizeLimit=10Gi,spark.kubernetes.driver.volumes
.persistentVolumeClaim.nfs-
pv.mount.path=/opt/owl/drivers/ext,spark.kubernetes.drive
r.volumes.persistentVolumeClaim.nfs-
pv.mount.readOnly=false
```

> **Note**
> The pod templates induce security context to the Spark driver and executor pods, which allows you to mount the PVC dynamically. You can also download the template file and host it in your own public space.

12. Click **Save**.
13. Go to the **Connections** page to start using your new custom JDBC data source driver.

Steps

1. Install the EFS Container Storage Interface (CSI) Driver on your cluster.
2. Create a Persistent Volume (PV) and a Persistent Volume Claim (PVC) in the same Kubernetes cluster namespace where your cloud native deployment of Collibra DQ is running with `ReadWriteMany` access mode, as shown in the following example:

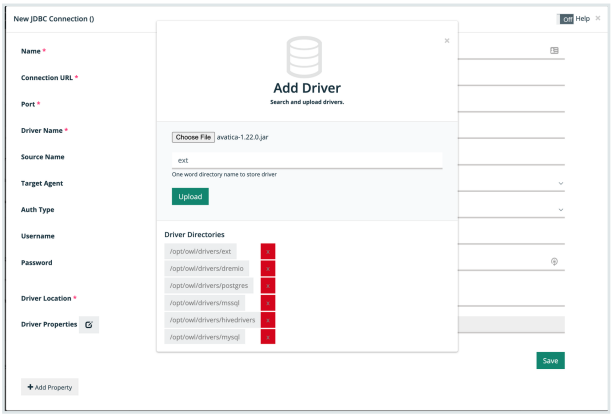
```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: efs-sc
provisioner: efs.csi.aws.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
---
apiVersion: v1
kind: PersistentVolume
metadata:
    name: efs-pv
spec:
    capacity:
        storage: 5Gi
    volumeMode: Filesystem
    accessModes:
        - ReadWriteMany
    storageClassName: efs-sc
    persistentVolumeReclaimPolicy: Retain
    csi:
        driver: efs.csi.aws.com
        volumeHandle: fs-e8a95a42
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
    name: efs-claim
spec:
    accessModes:
        - ReadWriteMany
    storageClassName: efs-sc
    resources:
        requests:
            storage: 5Gi
```

Tip You can choose any storage class that supports ReadWriteMany access mode only.

3. Patch the existing DQ Web and Agent StatefulSets with an additional Volume Mount, as shown in the following example:

```
spec:
  volumes:
    - name: owldq-ext-jdbc-jars
      persistentVolumeClaim:
        claimName: efs-claim
...
...
  containers:
    - volumeMounts:
        - name: owldq-ext-jdbc-jars
          mountPath: /opt/owl/drivers/ext
```

4. From the Connections page, click **Add** on the **Generic JDBC Connection** tile.
   » The **New JDBC Conection** modal appears.
5. Enter the required information.
6. Click the folder icon in the **Driver Location** option.
   » The **Add Driver** modal appears.
7. Click **Choose File** and select the JAR file to upload.
8. Enter a one-word directory name to store the driver, for example, *ext*.
9. Select `/opt/owl/drivers/ext` under **Driver Directories** to upload your driver to that location, as shown in the image below.



10. Click **Upload**.
    » A success message appears when your driver is successfully added.

> **Note**
> Uploads from DQ Web only work if the PVC's storageclass supports the fsGroup security context of Kubernetes. If the fsGroup security context is not supported, create a temporary dummy pod running as a root user with the PVC mounted, and copy the required JDBC jars to the mount location with 544 permissions.

11. Go to the **Remote Agent** section of the Admin Console and click the pencil icon to edit your agent.
    » The **Edit Agent** modal appears.
12. Add the following properties to the **Free Form (Appended)** option, as shown in the following example:

```
-conf
spark.kubernetes.driver.podTemplateFile=local:///opt/owl/
config/k8s-driver-
template.yml,spark.kubernetes.executor.podTemplateFile=lo
cal:///opt/owl/config/k8s-executor-
template.yml,spark.kubernetes.executor.volumes.persistent
VolumeClaim.efs-pv.options.claimName=efs-
claim,spark.kubernetes.executor.volumes.persistentVolumeC
laim.efs-
pv.options.sizeLimit=10Gi,spark.kubernetes.executor.volum
es.persistentVolumeClaim.efs-
pv.mount.path=/opt/owl/drivers/ext,spark.kubernetes.execu
tor.volumes.persistentVolumeClaim.efs-
pv.mount.readOnly=false,spark.kubernetes.driver.volumes.p
ersistentVolumeClaim.efs-pv.options.claimName=efs-
claim,spark.kubernetes.driver.volumes.persistentVolumeCla
im.efs-
pv.options.sizeLimit=10Gi,spark.kubernetes.driver.volumes
.persistentVolumeClaim.efs-
pv.mount.path=/opt/owl/drivers/ext,spark.kubernetes.drive
r.volumes.persistentVolumeClaim.efs-
pv.mount.readOnly=false
```

> **Note**
> The pod templates induce security context to the Spark driver and executor pods, which allows you to mount the PVC dynamically. You can also download the template file and host it in your own public space.

13. Click **Save**.
14. Go to the **Connections** page to start using your new custom JDBC data source driver.

Add CA certificates to the java keystore

This topic describes how to add CA certificates to the java keystore in a Kubernetes deployment. The following approach mounts the directory containing the SSL certificates from the Agent pod as a shared volume. The SSL certificates are then accessible to the Spark driver/executor pods, allowing them to establish a secure SSL/TLS connection with the email server and/or data source.

Steps

1. Mount the directory containing the SSL certificates in the Agent pod to a shared volume. This can be done in the deployment or pod configuration file for the Agent pod in Kubernetes. For example, you can add a volume and volumeMounts section to the Agent pod's configuration file like this:

```
volumes:
  - name: certs
    hostPath:
      path: /etc/pki/ca-trust/extracted/java/cacerts
```

2. Modify the Spark driver/executor pod's configuration to mount the shared volume containing the SSL certificates from the Agent pod. This can be done in the Spark configuration or pod configuration file. For example, you can add a volume and volumeMounts section to the Spark driver/executor pod's configuration file like this:

```
volumes:
  - name: certs
    hostPath:
      path: /path/to/agent/certs
```

3. Update the Java options for the Spark driver/executor pods to specify the path to the SSL certificates in the shared volume. You can do this by adding the `-Djavax.net.ssl.trustStore` and `-Djavax.net.ssl.trustStorePassword` options to the Spark driver/executor pod's configuration file, like this:

```
javaOptions:
  - "-Djavax.net.ssl.trustStore-
e=/path/to/agent/certs/cacerts"
  - "-Djavax.net.ssl.trustStorePassword=your_keystore_pass-
word"
```

4. Replace `/path/to/agent/certs` with the actual path to the shared volume where the SSL certificates are mounted in the Spark driver/executor pods, and `your_keystore_password` with the actual password for the keystore that contains the SSL certificates.
5. Restart the Spark driver/executor pods to apply the changes.

Troubleshooting Self-hosted Kubernetes Install

This section provides the most common commands to run when troubleshooting a DQ environment that is deployed on self-hosted Kubernetes.

Additional documentation

```
### Provide documentation on syntax and flags in the terminal

kubectl help

### To see how to use Kubernetes resources

kubectl api-resources -o wide
```

Viewing Kubernetes resources

```
### Get Pods, their names & details in all Namespaces

kubectl get pods -A -o wide

### Get all Namespaces in a cluster

kubectl get namespaces

### Get Services in all Namespaces

kubectl get services -A -o wide

### List all deployments in all namespaces:

kubectl get deployments -A -o wide
```

Logs and events

```
### List Events sorted by timestamp in all namespaces

kubectl get events -A --sort-by=.metadata.creationTimestamp

### Get logs from a specific pod:

kubectl logs [my-pod-name]
```

Resource allocation

```
### If the Kubernetes Metrics Server is installed,
### the top command allows you to see the resource consumption
for nodes or podscode

kubectl top node
kubectl top pod

### If the Kubernetes Metrics Server is NOT installed, use

kubectl describe nodes | grep Allocated -A 10
```

Configuration

```
### Get current-context

kubectl config current-context

### See all configs for the entire cluster

kubectl config view
```

Authorization issues

```
### Check to see if I can read pod logs for current user & con-
text

kubectl auth can-i get pods --subresource=log

### Check to see if I can do everything in my current namespace
("*" means all)

kubectl auth can-i '*' '*'
```

Jobs stuck in the Staged activity

If DQ Jobs are stuck in the Staged activity on the Jobs page, ensure you have the Helm Chart corresponding to the latest Data Quality & Observability Classic version and update the following properties:

```
--set global.configMap.data.metastore_max_wait_time=10000
--set global.configMap.data.metastore_max_active=2000
--set global.configMap.data.metastore_max_idle=100
--set global.configMap.data.metastore_initial_size=150
```

Restart the Web and Agent pods after you set the recommended properties.

Jobs fail with OOMKilled error

If DQ jobs fail with the error `OOMKilled`, you may need to allocate additional overhead. Follow these steps:

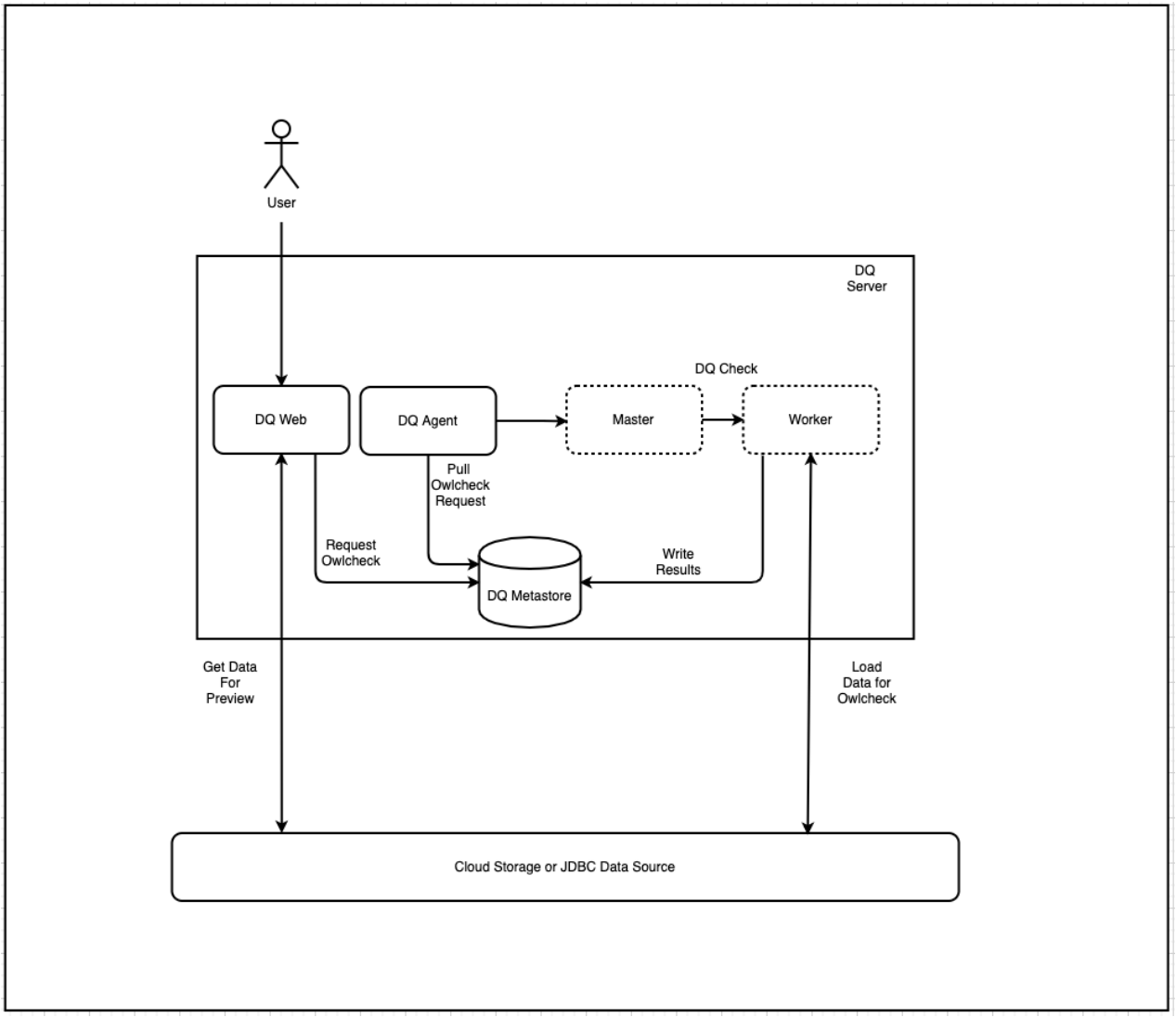
- Increase the executor memory by modifying the option `-executormemory`
- In the `-conf` list of properties, add the property `spark.kubernetes.memoryOverheadFactor=0.4`

Installing Data Quality & Observability Classic on Spark Standalone Self-hosted

Data Quality & Observability Classic can be installed and operated on a single standalone host, which is useful when large scale and high concurrency checks are not required. In this mode, DQ leverages a Spark Standalone pseudo cluster in which the master and workers run and use resources from the same server.

The DQ Standalone application consists of the following components:

- DQ Web
- DQ Agent
- DQ Metastore (PostgreSQL database)
- Spark (pullup only)



Data Quality & Observability Classic provides the option to include the PostgreSQL metastore in the installation or use an external PostgreSQL metastore (recommended).

What's next?

Before You Install Standalone

Before You Install Standalone

Prerequisites

- Review the Standalone architecture described in Installing Data Quality & Observability Classic on Spark Standalone Self-hosted.
- Verify that your system meets the System Requirements.

Download the Collibra DQ install package

You can download the DQ installation package from the Collibra Product Resource Center.

> **Note** For additional download options, contact your Account Team.

Save the package in a directory on the CDQ VM, to be extracted later for the install.

Extract the file:

```
tar -xvf dq-full-package.tar.gz
```

(Optional) Clean up:

```
rm dq-full-package.tar.gz
```

Set the environment variables

Set the following environment variables:

- OWL_BASE: The directory to install DQ files
- OWL_METASTORE_USER: The default username for the PostgreSQL server
- OWL_METASTORE_PASS: The default password for the PostgreSQL server. It must adhere to the following password policy:

    - Minimum length of 8 characters
    - Maximum length of 72 characters
    - At least one upper-case letter
    - At least one numeric character
    - At least one special character (supported are !,%,&,@,#,$,^,,?,_,~)*
    - Password cannot contain the PostgreSQL username
- SPARK_PACKAGE
- DQ_ADMIN_USER_PASSWORD: The password for the admin account on the DQ Web application. It must adhere to the following password policy:
    - Minimum length of 8 characters
    - Maximum length of 72 characters
    - At least one upper-case letter
    - At least one numeric character
    - At least one special character (supported are !,%,&,@,#,$,^,,?,_,~)*
    - Password cannot contain the username (admin)

For example, execute the following commands to set the environment variables:

```
export OWL_BASE=$(pwd)
export OWL_METASTORE_USER=postgres
export OWL_METASTORE_PASS=H55Mt5EbXh1a%\$aiX6
export SPARK_PACKAGE=spark-3.5.3-bin-hadoop3.tgz
export DQ_ADMIN_USER_PASSWORD=<password>
```

What's next?

Install Standalone

Install Standalone

After performing the steps in Before You Install Standalone, you are ready to install Collibra DQ Standalone.

> **Note** To perform a FIPS-compliant installation of Data Quality & Observability Classic, go to Configure FIPS-compliant algorithms in Data Quality & Observability Classic.

Run the setup script

After extracting the DQ package and setting the environment variables (described in Before you install), you are ready to run the `setup.sh` script. This script installs Collibra DQ and the required components, including Spark and PostgreSQL. You may install the components separately or at the same time.

When executing the `setup.sh` script, use the `-options` parameter to specify which components are installed. The options are:

- `spark` (required)
- `owlweb` (required)
- `owlagent` (required)
- `postgres` (optional)

Note that Java is a required component, but it is not included in the installation package.

You have the option to include the PostgreSQL metastore in the installation or use an external PostgreSQL metastore (recommended). Both options are described next.

Install Collibra DQ, Spark, and PostgreSQL metastore

To include PostgreSQL metastore in the installation process, include postgres in the `-options` parameter. For example, the following command executes `setup.sh` and installs PostgreSQL locally:

```
./setup.sh \
-owlbase=$OWL_BASE \
-user=$OWL_METASTORE_USER \
-pgpassword=$OWL_METASTORE_PASS \
-options=postgres,spark,owlweb,owlagent
```

When the setup script completes, you are prompted to finish the PostgreSQL installation:

```
Postgres DB needs to be intialized. Default location = <OWL_
BASE>/postgres/data
to change path please enter a FULL valid path for Postgres and
hit <enter>
DB Path [ <OWL_BASE>/owl/postgres/data ] =
```

Use the default location unless the PostgreSQL needs to be hosted at a specific location. If you change the location, verify that the directory is writable.

> **Note** To see the directory structure created by `setup.sh`, go to Data Quality & Observability Classic Directory Structure.

If you would like to install DQ Web on a machine with DQ Agent already installed, update the `owl-env.sh` file based on your specific configuration, then run the following command: `./owlmanage.sh start=owlweb`.

> **Note** For additional configuration options in `setup.sh`, `owl-env.sh`, and `owl.properties`, go to Additional Standalone Configuration Options.

Install Collibra DQ and Spark with an external PostgreSQL metastore (recommended)

If you would like to use an external PostgreSQL metastore, such as a GCP Cloud SQL service, omit `postgres` from the `-options` parameter and include `-pgserver` in the command. For example:

```
./setup.sh \
-owlbase=$OWL_BASE \
-user=$OWL_METASTORE_USER \
-pgpassword=$OWL_METASTORE_PASS \
-options=spark,owlweb,owlagent \
-pgserver="localhost:5432/postgres"
```

In this example, the installation process does not install PostreSQL and instead uses the existing PostgresDB server located in `localhost:5432` with the `postgres` database, which is the default DQ metadata storage.

> **Note** To see the directory structure created by `setup.sh`, go to Data Quality & Observability Classic Directory Structure.

> **Note** For additional configuration options in `setup.sh`, `owl-env.sh`, and `owl.properties`, go to Additional Standalone Configuration Options.

What's next?

Complete the Initial Setup

Complete the Initial Setup

After installing Collibra DQ Standalone, you are ready to complete the initial setup.

Set the license key and name

After executing the `setup.sh` script, set the license key and name. They are provided by Collibra in your license provision email. Use the following commands:

```
export DQ_INSTALL_DIR=/opt
echo "export LICENSE_KEY=<key>" >> $DQ_INSTALL_DIR/owl/-
config/owl-env.sh
echo "export LICENSE_NAME=<name>" >> $DQ_INSTALL_DIR/owl/-
config/owl-env.sh
```

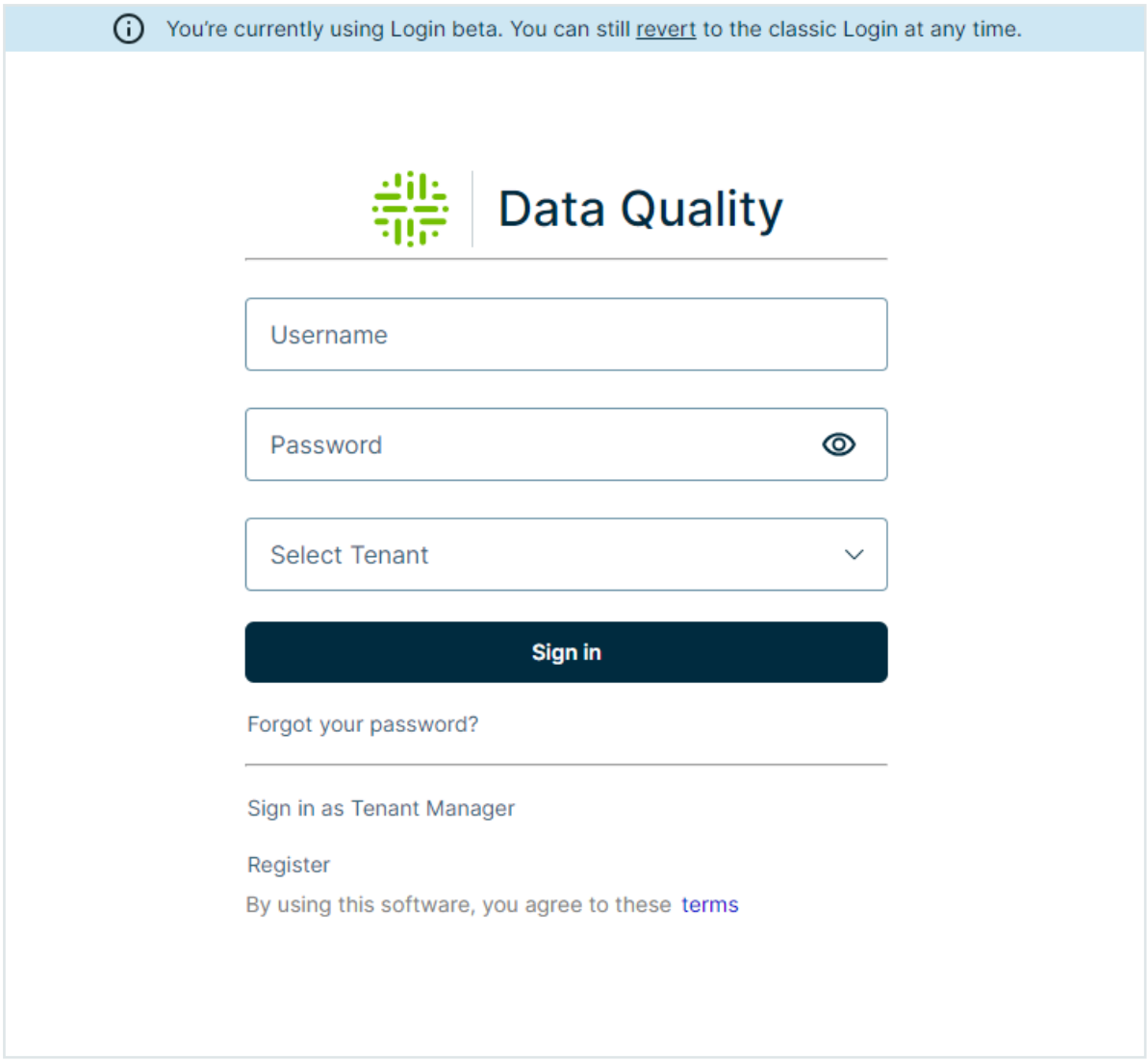
Then restart the DQ Web service:

```
./owlmanage.sh stop=owlweb
./owlmanage.sh start=owlweb
```

Log in to Collibra DQ

1. Verify that the DQ Web application is running by opening the following URL in your browser:
   `<YOUR_SERVER_IP>:9000`

   The Collibra DQ login page displays.



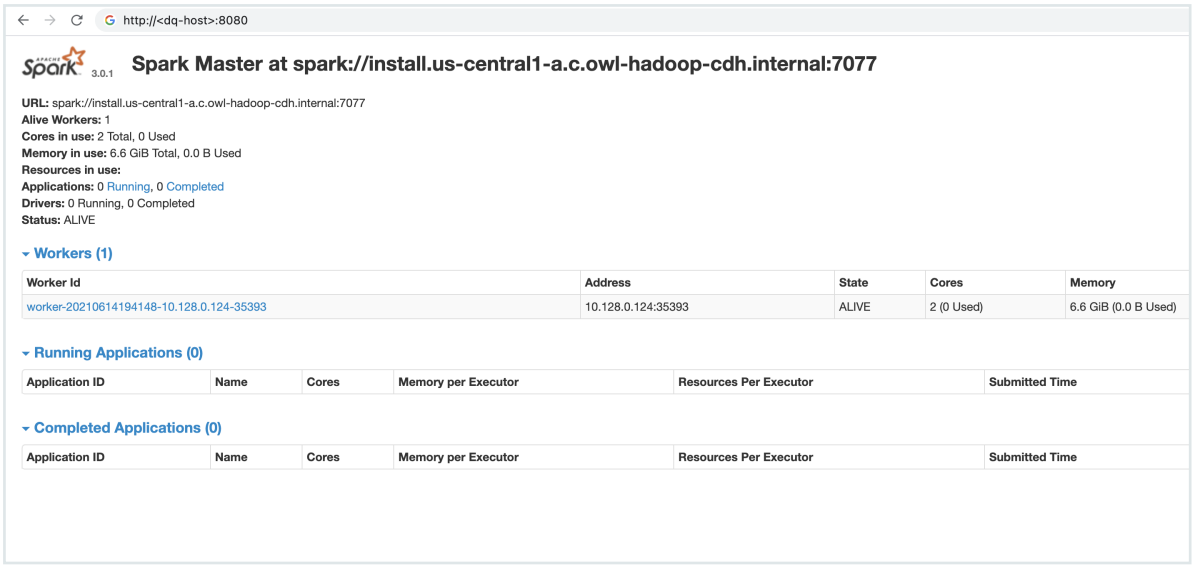
2. Log in to the admin user account. The username is **admin** and the password is specified in the environment variable DQ_ADMIN_USER_PASSWORD.

Verify that the Spark cluster has started

1. Open the following URL in your browser:
   `<YOUR_SERVER_IP>:8080`

   > **Note** This port may not be open depending on your environment.

   The Spark Master page displays.



2. Take note of the Spark Master URL. It is required during DQ Agent configuration.

Update DQ Agent properties

In this step, you'll start the DQ Agent, which creates a file called `agent.properties`. You'll then stop the agent, edit the `agent.properties` file, and then restart DQ Agent.

1. Start the DQ Agent. Enter the following commands:

```
cd $OWL_BASE/owl/bin
./owlmanage.sh start=owlagent
```

   This process creates the file `agent.properties` in the `$OWL_BASE/owl/config` directory.

   > **Note** To see the directory structure created by `setup.sh`, go to Data Quality & Observability Classic Directory Structure.

2. Stop the DQ Agent. Enter the following command:

```
./owlmanage.sh stop=owlagent
```

3. Add the following lines to the agent.properties file:

```
sparksubmitmode=native
sparkhome=</your/spark/home/folder>
```

4. Start the agent again. Enter the following command:

```
./owlmanage.sh start=owlagent
```

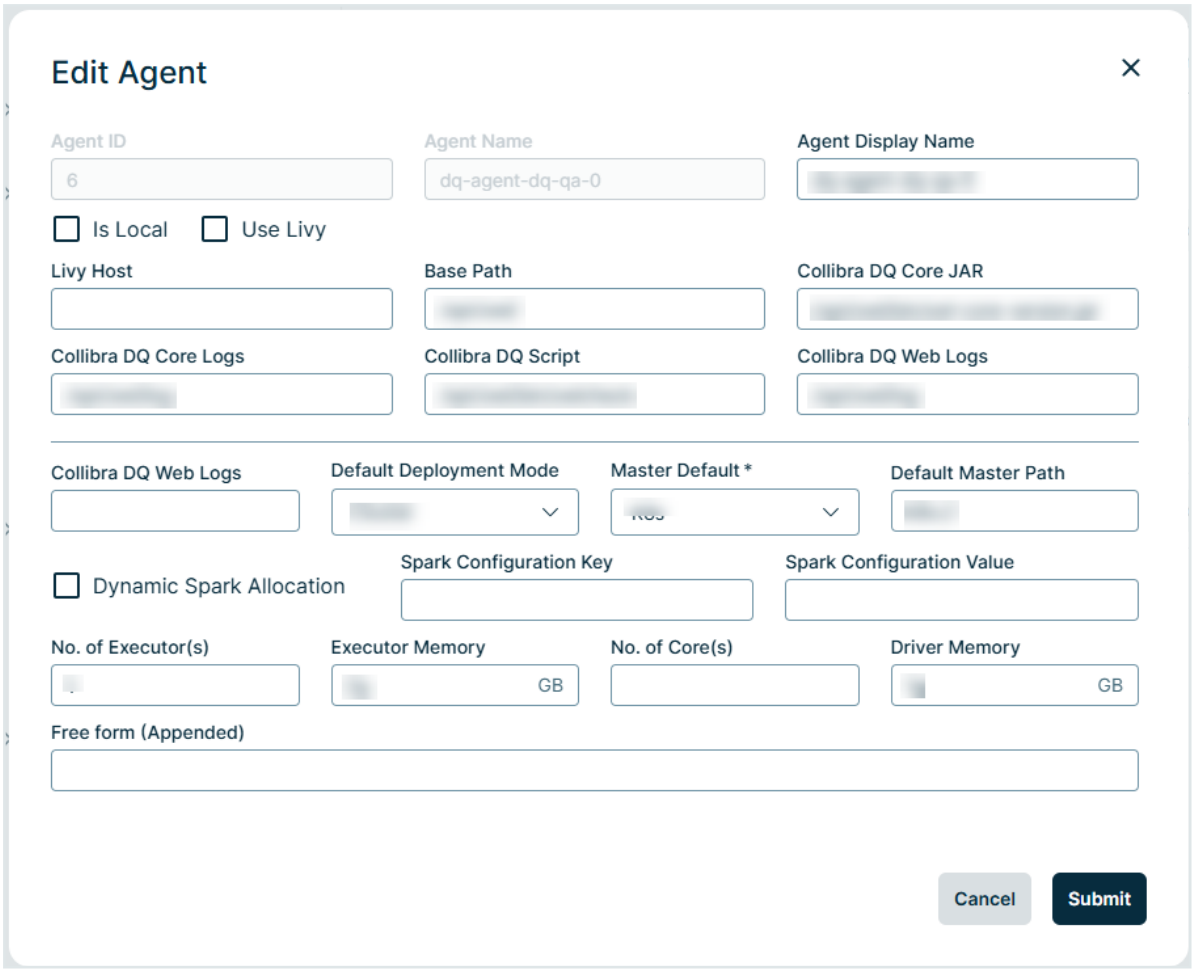
5. The `agent.properties` file identifies the number of agents installed on your server. Since this is the first non-default agent installed, the expected number of agents is 2. Verify that `agent.properties` contains the correct details.

```
cd $OWL_BASE/owl/config
cat agent.properties
```

   The output should include `agentid=2`.

Edit the DQ Agent configuration

1. In the DQ Web, open Admin Console and click **Agent Configuration**.
2. In the row that displays **2** under Agent ID, click **Actions** and then click **Edit**. The Edit Agent dialog displays.



Notice that the paths in the following fields use `/opt/owl/` as the base path, which do not match the base path to your instance of Collibra DQ:

- **Base Path**
- **Collibra DQ Core JAR**
- **Collibra DQ Core Logs**
- **Collibra DQ Script**
- **Collibra DQ Web Logs**

3. Update the paths in these fields to include the proper base path.

> **Important** The `owlmanage.sh start=owlagent` script does not respect the `OWL_BASE` environment variable. You must edit the Agent Configuration to follow the `OWL_BASE`.

4. In the **Default Master Path** field, enter the Spark Master URL that you copied from the Spark Master page.
5. Update the values in the following fields based on the requirements of your instance:
   - **Number of Executors(s)**
   - **Executor Memory (GB)**
   - **No. of Core(s)**
   - **Driver Memory (GB)**
6. In the **Free form (Appended)** field, enter the following command:

   ```
   -conf spark.cores.max=8
   ```

7. In the **Default Deployment Mode** field, select **Client**.

For more information on configuring the DQ Agent, go to Set up a DQ agent.

Map a connection to the DQ Agent

Follow the steps provided in Map data source connections to a DQ agent.

Run the first DQ job

1. In the DQ Web, open the Explorer page.
2. Select a connection, schema, and table.
3. Click **Create DQ Job**.

For more information on DQ jobs, go to Create a DQ job.

What's next?

Troubleshooting Standalone Install

Troubleshooting Standalone Install

This section provides a variety of tips for troubleshooting the Collibra DQ standalone installation process.

> **Note** To see the directory structure created by `setup.sh`, go to Data Quality & Observability Classic Directory Structure.

> **Note** For additional configuration options in `setup.sh`, `owl-env.sh`, and `owl.properties`, go to Additional Standalone Configuration Options.

Start and stop components

The `owlmanage.sh` script enables you to stop and start services or some components of services. It is executed from the `/owl/bin` directory.

To start different components:

```
./owlmanage.sh start=postgres
./owlmanage.sh start=owlagent
./owlmanage.sh start=owlweb
```

To stop different components:

```
./owlmanage.sh stop=postgres
./owlmanage.sh stop=owlagent
./owlmanage.sh stop=owlweb
```

Increase memory usage

To increase memory usage of Java processes in DQ Web, add the following environment variable to `owl-env.sh` and restart the DQ Web service:

```
export EXTRA_JVM_OPTIONS="-Xms2g -Xmx2g"
```

To increase memory usage of Java processes in DQ Agent, update `owlmanage.sh`. In the `start_owlagent()` function (at line number 47), update the `/java` command to include the options `-Xms2g -Xmx2g`. Then, restart the DQ Agent service.

Verify that the working directory has permissions

For example, if I SSH into the machine with user `owldq` and use my default home directory location `/home/owldq/`

```
### Ensure appropriate permissions
### drwxr-xr-x

chmod -R 755 /home/owldq
```

Reinstall PostgreSQL

```
### Postgres data directly initialization failed
### Postgres permission denied errors
### sed: can't read /home/owldq/owl/-
postgres/data/postgresql.conf: Permission denied

sudo rm -rf /home/owldq/owl/postgres
chmod -R 755 /home/owldq

### Reinstall just postgres
./setup.sh -owlbase=$OWL_BASE -user=$OWL_METASTORE_USER -pgpass-
word=$OWL_METASTORE_PASS -options=postgres
```

Change PostgreSQL password from SSH

```
### If you need to update your postgres password, you can lever-
age SSH into the VM
### Connect to your hosted instance of Postgres

sudo -i -u postgres
psql -U postgres
\password
#Enter new password: ### Enter Strong Password
#Enter it again: ### Re-enter Strong Password
\q
exit
```

Warning The `$` symbol is not a supported special character in your PostgreSQL Metastore password.

Add permissions for ssh keys when starting Spark

```
### Spark standalone permission denied after using ./start-
all.sh

ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Tip If the recommendation above is unsuccessful, use the following commands instead of `./start-all.sh`:

```
./start-master.sh
./start-worker.sh spark://$(hostname -f):7077
```

Change permissions if log files are not writable

```
### Changing permissiongs on individual log files

sudo chmod 777 /home/owldq/owl/pids/owl-agent.pid
sudo chmod 777 /home/owldq/owl/pids/owl-web.pid
```

Get the hostname of the instance

```
### Getting the hostname of the instance

hostname -f
```

Increase thread pool

If the thread pool is exhausted, for example, if the error logs include PoolExhaustedException, you may need to increase the thread pool and adjust timeout settings.

Update the following DQ Web settings in owl-env.sh:

```
export SPRING_DATASOURCE_TOMCAT_MAXIDLE = 10
export SPRING_DATASOURCE_TOMCAT_MAXACTIVE = 20
export SPRING_DATASOURCE_TOMCAT_MAXWAIT = 10000
```

Update the following DQ Agent settings in owl-env.sh:

```
export SPRING_DATASOURCE_POOL_MAX_WAIT=500
export SPRING_DATASOURCE_POOL_MAX_SIZE=30
export SPRING_DATASOURCE_POOL_INITIAL_SIZE=5
```

For DQ jobs with yarn or cluster modes, update the following settings in owl.properties:

```
spring.datasource.tomcat.initial-size=5
spring.datasource.tomcat.max-active=30
spring.datasource.tomcat.max-wait=1000
```

Be sure to restart DQ Web and DQ Agent after making changes to the settings.

Jobs stuck in the Staged activity

If DQ Jobs are stuck in the Staged activity on the Jobs page, update the following DQ Web settings in owl-env.sh:

```
export SPRING_DATASOURCE_POOL_MAX_WAIT=2500
export SPRING_DATASOURCE_POOL_MAX_SIZE=1000
export SPRING_DATASOURCE_POOL_INITIAL_SIZE=150
export SPRING_DATASOURCE_TOMCAT_MAXIDLE=100
export SPRING_DATASOURCE_TOMCAT_MAXACTIVE=2000
export SPRING_DATASOURCE_TOMCAT_MAXWAIT=10000
```

Depending on whether your agent is set to Client or Cluster default deployment mode, you may also need to update the following settings in owl.properties:

```
spring.datasource.tomcat.initial-size=5
spring.datasource.tomcat.max-active=30
spring.datasource.tomcat.max-wait=1000
```

Be sure to restart DQ Web and DQ Agent after making changes to the settings.

Active database queries

```
select * from pg_stat_activity where state='active'
```

Too many open files error

```
### "Too many open files error message"
### check and modify that limits.conf file
### Do this on the machine where the agent is running for Spark
standalone version

ulimit -Ha
cat /etc/security/limits.conf

### Edit the limits.conf file
sudo vi /etc/security/limits.conf

### Increase the limit for example
### Add these 3 lines
fs.file-max=500000
*               soft    nofile          58192
*               hard    nofile          100000
### do not comment out the 3 lines (no '#'s in the 3 lines
above)
```

Path not allowed error

If the following error occurs:

```
java.nio.file.InvalidPathException: Path not allowed, please contact
DQ admin:
```

Specify the allowed local paths. Add the following entry to owl-env.sh:

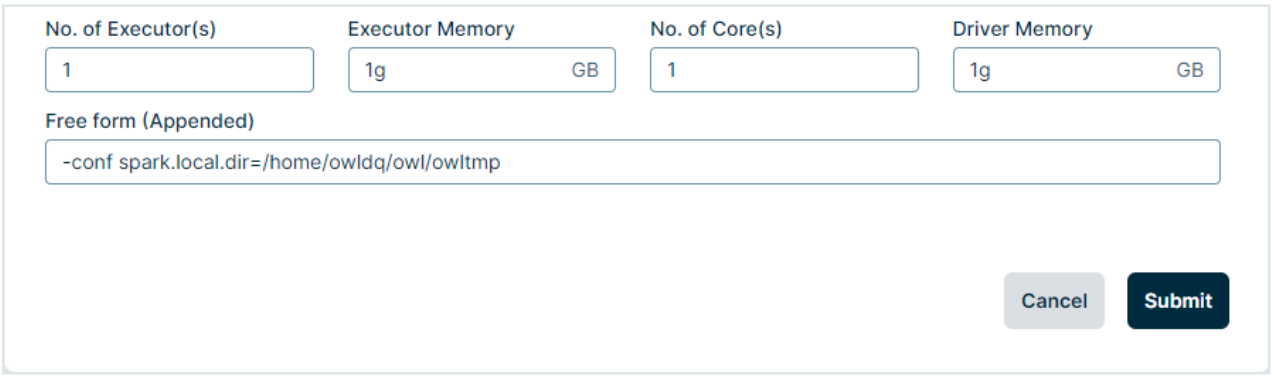
```
export ALLOWED_LOCAL_PATHS='*'
```

You may use the wildcard (**'*'**) to enable all local paths, or specify one or more specific paths separated by a comma.

Redirect Spark scratch

```
### Redirect Spark scratch to another location
SPARK_LOCAL_DIRS=/mnt/disks/sdb/tmp
```

Alternatively, you can add the following to the Free form (Appended) field on the Agent Configuration page to change Spark storage: `-conf spark.local.dir=/home/owldq/owl/owltmp`



Automate cleanup of Spark work folders

You can add the following line of code to **owl/spark/conf/spark-env.sh**, which must be copied from the **spark-env.sh.template** file, or to the bottom of **owl/spark/bin/load-spark-env.sh**.

```
### Set Spark to delete older files
export SPARK_WORKER_OPTS="${SPARK_WORKER_OPTS} -Dspark.-
worker.cleanup.enabled=true -Dspark.worker.cleanup.interval=1800
-Dspark.worker.cleanup.appDataTtl=3600"
```

Check disk space in the Spark work folder

```
### Check worker nodes disk space
sudo du -ah | sort -hr | head -5
```

Delete files in the Spark work folder

```
### Delete any files in the Spark work directory
sudo find /home/owldq/owl/spark/work/* -mtime +1 -type f -delete
```

Troubleshooting Kerberos

For debug logging, add the following to the owl-env.sh file:

```
# For Kerberos debug logging
export EXTRA_JVM_OPTIONS="-Dsun.security.krb5.debug=true"
```

Reboot the Collibra DQ web service with the following:

```
./owlmanage.sh stop=owlweb
./owlmanage.sh start=owlweb
```

> **Note** You can use the option for several purposes such as SSL debugging, setting HTTP/HTTPS proxy settings, setting additional keystore properties, and so on.

Add Spark home environment variables to profile

```
### Adding ENV variables to bash profile

### Variable 'owldq' below should be updated wherever installed
e.g. centos

vi ~/.bash_profile
export SPARK_HOME=/home/owldq/owl/spark
export PATH=$SPARK_HOME/bin:$PATH

### Add to owl-env.sh for standalone install

vi /home/owldq/owl/config/owl-env.sh
export SPARK_HOME=/home/owldq/owl/spark
export PATH=$SPARK_HOME/bin:$PATH
```

Spark launch scripts

For information on working with Spark launch scripts, go to Spark Launch Scripts.

Check that processes are running

```
### Checking PIDS for different components

ps -aef|grep postgres
ps -aef|grep owl-web
ps -aef|grep owl-agent
ps -aef|grep spark
```

Additional Standalone Configuration Options

This section describes additional configuration options in `setup.sh`, `owl-env.sh`, and `owl.properties`.

> **Note** To see the directory structure created by `setup.sh`, go to Data Quality & Observability Classic Directory Structure.

Setting additional options in setup.sh

The following table describes the arguments supported by `setup.sh`:

Argument	Description
`-non-interactive`	Skips asking to accept Java license agreement.
`-skipSpark`	Skips the extraction of Spark components.
`-stop`	Do not automatically start all components (Owl-Web, Zeppelin, Postgres).
`-port=`	Set DQ Web application to use the defined port.
`-user=`	Optional parameter to set the user to run Collibra DQ. The default is the current user.
`-owlbase=`	Sets the base path to where you want Collibra DQ installed.
`-owlpackage=`	Optional parameter to set the Collibra DQ package directory. The default is the current working directory.
`-help`	Display this help and exit.
`-options=`	The different Collibra DQ components to install in a comma-separated list format. For example, -options=owlagent,owlweb,postgres,spark
`-pgpassword=`	The password used to set for the PostgreSQL metastore. For unattended installs.
`-pgserver=`	The name of the PostgreSQL server. For example, -pgserver=owl-postgres-host.example.com:5432/owldb. For unattended installs.

Example setup.sh commands

A common example of the `setup.sh` command is:

```
./setup.sh -port=9000 -user=ec2-user -owlbase=/home/ec2-user -owlpackage=/home/ec2-user/packages
```

- The tar ball extracted to this folder on my EC2 Instance: ****/home/ec2-user-/packages/
- Collibra DQ is running as the ****ec2-user
- The DQ Web application runs on port 9000
- The base location for the setup.sh script to create the will be: /home/ec2-user/

This example installs only the DQ Agent:

```
./setup.sh -user=ec2-user -owlbase=/home/ec2-user -
owlpackage=/home/ec2-user/package -options=owlagent
```

- The package extracted to this folder on my EC2 Instance: ****/home/ec2-user-/packages/
- Owl-agent is running as the ****ec2-user
- The base location for the setup.sh script to create the Collibra DQ folder and place all packages under Collibra DQ is: /home/ec2-user/

Configuring environment settings with owl-env.sh

The owl-env.sh script holds the main variables that are reused across components. It is executed from the /owl/config directory.

owl-env.sh scripts	Description
export SPARK_CONF_DIR="/home/collibra/owl/cdh-spark-conf"	The directory on your machine where the Spark conf directory resides.
export INSTALL_PATH="/home/collibra/owl"	The installation directory of Collibra DQ.
export JAVA_HOME="/home/collibra/jdk1.8.0_131"	Java Home for Collibra DQ to leverage.
export LOG_PATH="/home/collibra/owl/log"	The log path.
export BASE_PATH="/home/collibra"	The base location under which the Collibra DQ directory resides.

owl-env.sh scripts	Description
`export SPARK_MAJOR_VERSION=2`	Spark Major version. Collibra DQ only supports 2+ version of Spark.
`export OWL_LIBS="/home/collibra/owl/libs"`	Lib Directory to inject in spark-submit jobs.
`export USE_LIBS=0`	Use the lib directory or not. 0 is the default. A value of 0 means the lib directory is not used. A value of 1 means the lib directory is used.
`export SPARKSUBMITEXE="spark-submit"`	Spark submit executable command. Collibra DQ using spark-submit as an example.
`export ext_pass_manage=0`	If using a password management system. You can enable for password to be pulled from it. A value of 0 disables an external password management system. A value of 1 enables an external password management system.
`export ext_pass_scriptt="/opt/owl/bin/getpassword.sh"`	Leverage password script to execute a get password script from the vault.
`TIMEOUT=900 #15 minutes in seconds`	Owl-Web user time out limits.

owl-env.sh scripts	Description
`export LOCAL_REGISTRATION_ENABLED=false`	Disables the Registration link on the login page, thereby preventing the creation of new user accounts.
`PORT=9003 #owl-web port NUMBER`	The default port to use for owl-web.
`export SPRING_LOG_LEVEL=ERROR`	The logging level to be displayed in the owl-web.log
`export SPRING_DATASOURCE_DRIVER_CLASS_NAME-E=org.postgresql.Driver`	The driver class name for postgres metastore (used by web).
`export SPRING_DATASOURCE_URL-L=jdbc:postgresql://localhost:5432/postgres`	JDBC connection string to Collibra DQPostgreSQL metastore.
`export SPRING_DATASOURCE_USERNAME=collibra`	Collibra DQPostgreSQL username.
`export SPRING_DATASOURCE_PASSWORD-D=3+017wfY1l1vmsvGYAyUcw5zGL`	Collibra DQPostgreSQL password.
`export AUTOCLEAN=TRUE/FALSE`	TRUE/FALSE Enable/Disable automatically delete old datasets.
`export DATASETS_PER_ROW=200000`	Delete datasets after this threshold is hit (must be greater than the default to change).
`export ROW_COUNT_THRESHOLD=300000`	Delete rows after this threshold is hit (must be greater than the default to change).
`export SERVER_HTTP_ENABLED=true`	Enabling HTTP to owl web

owl-env.sh scripts	Description
`export OWL_ENC=OFF #JAVA for java encryption`	Enable Encryption (NOTE need to add to owl.-properties also). Has to be in form owl.enc=OFF within owl.properties file to disable, and in this form owl.enc=JAVA to enable. the owl.properties file is located in the owl install path /config folder (/opt/owl/config).
`PGDBPATH=/home/collibra/owl/owl-postgres/bin/data`	Path for PostgreSQL DB
`export RUNS_THRESHOLD=5000`	Delete runs after this threshold is hit (must be greater than the default to change).
`export HTTP_SECONDARY_PORT=9001`	Secondary HTTP port to use which is useful when SSL is enabled.
`export SERVER_PORT=9000`	Same as PORT.
`export SERVER_HTTPS_ENABLED=true`	Enabling of SSL.
`export SERVER_SSL_KEY_TYPE=PKCS12`	Certificate trust store.
`export SERVER_SSL_KEY_PASS-S=t2lMFWEHsQha3QaWnNaR8ALaFPH15Mh9`	Certificate key password.
`export SERVER_SSL_KEY_ALIAS=owl`	Certificate key alias.
`export SERVER_REQUIRE_SSL=true`	Override HTTP on and force HTTPS regardless of HTTP settings.
`export MULTITENANTMODE=FALSE`	Flipping to TRUE will enable multi tenant support.

owl-env.sh scripts	Description
`export multiTenantSchemaHub=owlhub`	Schema name used for multi tenancy.
`export OWL_SPARKLOG_ENABLE=false`	Enabling deeper spark logs when toggled to true.
`export LDAP_GROUP_RESULT_DN_ATTRIBUTE`	The attribute to the full path of the group object, for example, **CN=OwlAppAdmin**, **OU=OwlGroups**,**OU=Groups**,**DC=owl**, **DC=com**. Default is **distinguishedname**.
`export LDAP_GROUP_RESULT_NAME_ATTRIBUTE`	The attribute to the simple name of the group, for example, **OwlAppAdmin**. Default is **CN**.
`export LDAP_GROUP_RESULT_CONTAINER_BASE`	Property used in the scenario where the LDAP_GROUP_RESULT_DN_ATTRIBUTE does not return a value. In this case, the LDAP_GROUP_RESULT_NAME_ATTRIBUTE prepends to this value, which creates a fully qualified LDAP path. For example, **OU=OwlGroups**, **OU=Groups**,**DC=owl**,**DC=com**. Default is <**null**>.

owl-env.sh scripts	Description
`export ALLOWED_LOCAL_PATHS='*'`	Enables Collibra DQ to use the specified local paths. Wildcard (**'*'**) is allowed. Separate multiple paths using a comma.
`export SERVER_FORWARD_HEADERS_STRATEGY-Y=FRAMEWORK`	Resolves an issue related to how Spring Boot handles forwarded headers that can cause an error to occur during login.

Configuring properties with owl.properties

The `owl.properties` file enables you to set the following properties. It is located in the `/owl/config` directory:

Example	Description
`key=XXXXXX`	The license key.
`spring.data-source.url-l=jdbc:postgresql://localhost:5432/postgres`	The connection string to the Collibra DQ metastore (used by owl-core).
`spring.datasource.password=xxxxxx`	The password to the Collibra DQ metastore (used by owl-core).
`spring.datasource.username=xxxxxx`	The username to the Collibra DQ metastore (used by owl-core).
`spring.datasource.driver-class-name-e=com.owl.org.postgresql.Driver`	Shaded PostgreSQL driver class name.

Example	Description
`spring.agent.datasource.url`	jdb- c:post- gresql://$host:$- port/owltrunk
`spring.agent.datasource.username`	{user}
`spring.agent.datasource.passwords`	{password}
`spring.agent.datasource.driver-class-name`	org.postgresql.Driver

Spark Launch Scripts

To launch a Spark standalone cluster with the launch scripts, you should create a file called `conf/workers` in your Spark directory, which must contain the hostnames of all the machines where you intend to start Spark workers, one per line. If conf/workers does not exist, the launch scripts defaults to a single machine (localhost), which is useful for testing. Note, the master machine accesses each of the worker machines via ssh. By default, ssh is run in parallel and requires password-less (using a private key) access to be setup. If you do not have a password-less setup, you can set the environment variable SPARK_SSH_FOREGROUND and serially provide a password for each worker.

Once you've set up this file, you can launch or stop your cluster with the following shell scripts, based on Hadoop's deploy scripts, and available in `SPARK_HOME/sbin`:

Shell Script	Description
`sbin/start-master.sh`	Starts a master instance on the machine the script is executed on.
`sbin/start-workers.sh`	Starts a worker instance on each machine specified in the `conf/workers` file.
`sbin/start-worker.sh`	Starts a worker instance on the machine the script is executed on.

Shell Script	Description
`sbin/start-all.sh`	Starts both a master and a number of workers as described above.
`sbin/stop-master.sh`	Stops the master that was started via the `sbin/start-master.sh` script.
`sbin/stop-worker.sh`	Stops all worker instances on the machine the script is executed on.
`sbin/stop-workers.sh`	Stops all worker instances on the machines specified in the `conf/workers` file.
`sbin/stop-all.sh`	Stops both the master and the workers as described above.

> **Note** These scripts must be executed on the machine you want to run the Spark master on, not your local machine.

Starting and stopping Spark standalone

To start Spark standalone:

```
cd /home/owldq/owl/spark/sbin
./start-all.sh
```

To stop Spark standalone:

```
cd /home/owldq/owl/spark/sbin
./stop-all.sh
```

Starting Spark standalone with separate workers

To start Spark Standalone with separate workers:

```
SPARK_WORKER_OPTS=" -Dspark.worker.cleanup.enabled=true -Dspark.-
worker.cleanup.interval=1799 -Dspark.-
worker.cleanup.appDataTtl=3600"

### 1 start master
/home/owldq/owl/spark/sbin/start-master.sh

### 2 start workers
SPARK_WORKER_INSTANCES=3;/home/owldq/owl/spark/sbin/start-
slave.sh spark://$(hostname):7077 -c 5 -m 20g
```

Configure FIPS-compliant algorithms in Data Quality & Observability Classic

Federal Information Processing Standards (FIPS) are the standards by which many federal and private sector organizations secure their computer systems. Support for FIPS-compliant algorithms in Data Quality & Observability Classic can be configured upon installation through the environment variables in the table below.

The OS environment variable `export DQ_APP_FIPS_ENABLED=true` is set in the **setup.sh**, **owlmanage.sh**, and **owl-env.sh** files, while all other optional environment variables are set in the **owl-env.sh**. While all environment variables are optional, if you do not set them, they will default to the out-of-the-box Bouncy Castle configurations.

Environment variable	Description
`export DQ_APP_FIPS_ENABLED=true`	Optional. Enables or disables FIPS-compliant algorithms. This must be enabled in the **setup.sh** and **owlmanage.sh** file upon the installation of Data Quality & Observability Classic. Default is **false**.
`export DQ_APP_AES_ENCRYPTION_KEY=<key-string>`	Optional. Only required when using your own Java security provider, algorithm, and encryption key. Enter the key for AES encryption from your preferred provider. Defaults to the Collibra DQ AES key.

Environment variable	Description
`export DQ_FIPS_PROVIDER_NAME=<java-provider-class>`	Optional. Only required when using your own Java security provider, algorithm, and encryption key. Provide your own provider implementation. Defaults to Bouncy Castle, which is supported in Collibra DQ and considered one of the secure and compliant FIPS providers.
`export DQ_FIPS_ALGORITHM=<algorithm>`	Optional. Only required when using your own Java security provider, algorithm, and encryption key. Provide an algorithm supported by your FIPS provider. Defaults to password-based AES 128 encryption algorithm complaint with FIPS.

Steps

1. Before installing Data Quality & Observability Classic, ensure that the entry **org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider** is set as the security provider in **$JAVA_HOME/conf/security/java.security**.

   > **Example**
   > `security.provider.13=org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider`

2. Optionally, set the OS environment variable `export DQ_APP_FIPS_ENABLED=true` in the **setup.sh** and **owlmanage.sh** files.
3. Follow the standard Standalone Installation steps.
   Show screenshot of FIPS-enabled installation

```
checking for Java
/bin/java
found java executable in PATH
Version is greater than 1.7
Setting JAVA_HOME to /usr/lib/jvm/java-11-openjdk-11.0.25.0.9-3.el9.x86_64
checking for FIPS mode
FIPS mode is enabled. Proceeding with FIPS compliance checks.
FIPS provider (org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider) found in /usr/lib/jvm/java-11-openjdk-11.0.25.0.9
-3.el9.x86_64/conf/security/java.security.
Do you want to proceed with enabling FIPS mode using this provider? (y/n): y
Enter DQ_APP_AES_ENCRYPTION_KEY (press enter to use default):
Using default AES encryption key.
*****
Postgres DB needs to be intialized. Default location = /opt/owl/postgres/data
to change path please enter a FULL valid path for Postgres and hit <enter>
```

4. Sign in to Data Quality & Observability Classic.

5. On the Collibra DQ home page, hover your pointer over ⚙ **Settings** and select **Admin Console**.
   » The Admin Console opens.
6. Click **Agent Configuration**.
   » The Agent Configuration page opens.
7. In the Actions column, click to edit your agent.
   » The **Edit Agent** modal appears.
8. In the Free Form (Appended) field, enter the following configuration and ensure that the Bouncy Castle FIPS JAR file, such as **bc-fips-x.jar**, is included in the directory.

```
-addlib $OWL_HOME/bin/external/
```

9. Restart the DQ web application and DQ agent services.

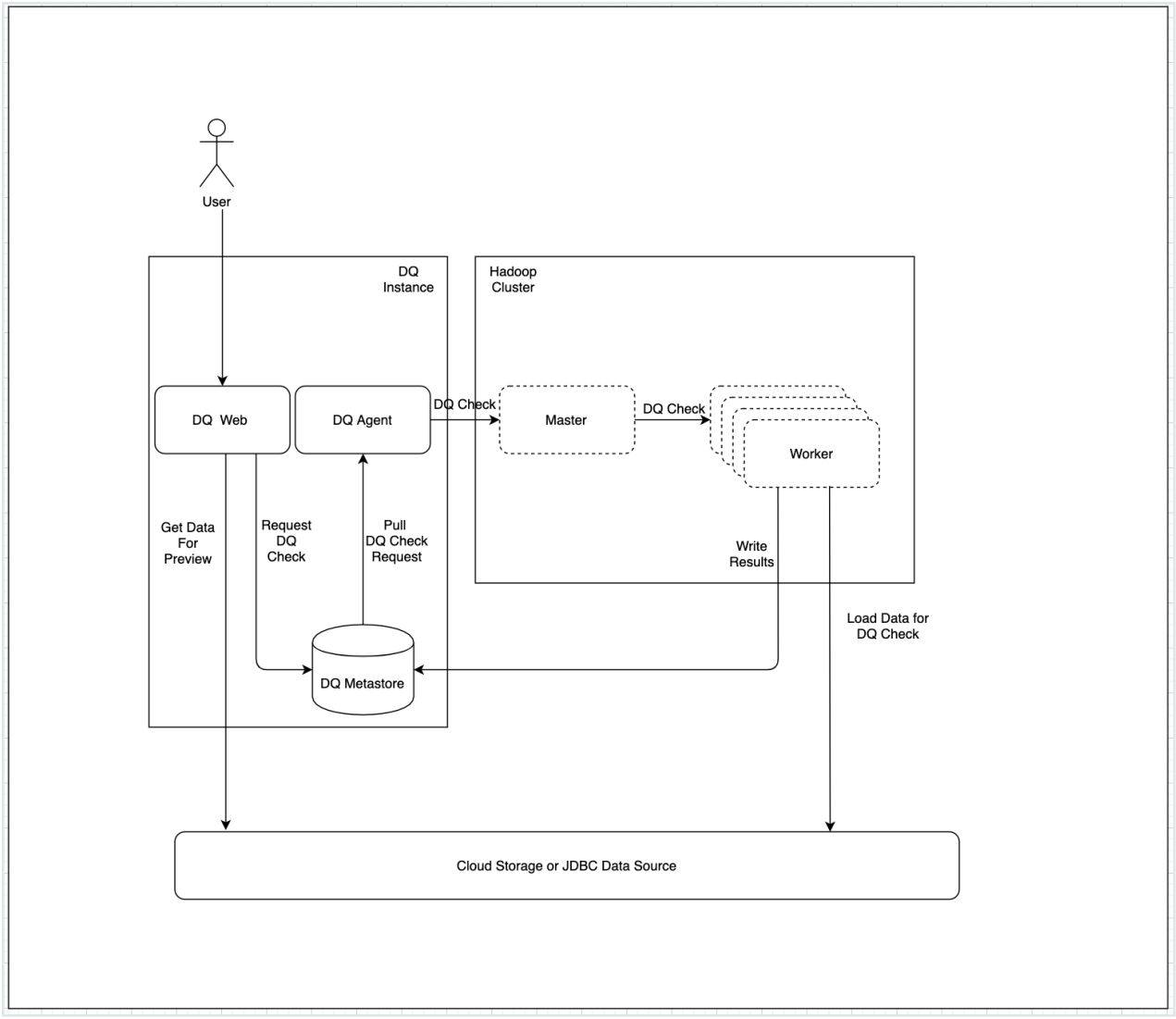
Example of environment variables when using the default Bouncy Castle FIPS provider and Collibra DQ values

```
export DQ_APP_FIPS_ENABLED=true
export DBPASSWORD="String to Encrypt"
# If successful, the following command will return encrypted
string
java -cp dq-core.jar:bc-fips-1.0.2.5.jar com.owl.-
core.security.SecurityManager -encryptString $DBPASSWORD
```

Configuring Apache Hadoop to Execute DQ Jobs

For large-scale processing and concurrency, a single vertically scaled Spark server may not be sufficient. To address large-scale processing, Collibra Data Quality & Observability can push compute jobs to an external Hadoop cluster. This section describes how to configure the DQ Agent to push DQ jobs to Hadoop.

The following diagram shows the DQ architecture with a Hadoop cluster:



Configure Hadoop

Overview of the configuration process

Hadoop configuration can be incredibly complex. However, with DQ, the goal is simply to leverage Hadoop to allocate compute resources to execute Spark jobs. As a result, the only client-side configurations required are:

- Security protocol definition
- Yarn Resource Manager endpoints

- Storage service (HDFS or Cloud storage).

Once you've defined the Hadoop client configuration, point the DQ Agent at the folder that contains the client configuration files. The DQ Agent is then able to use the Hadoop client configuration to submit jobs to the specified Hadoop cluster.

Additional considerations

DQ jobs running on Hadoop are Spark jobs. DQ will use the storage platform defined in the `fs.defaultFS` setting to distribute all of the required Spark libraries and specified dependency packages like drivers files. This allows DQ to use a version of Spark that is different than the one provided by the cluster. If it is a requirement to use the Spark version provided by the target Hadoop cluster, obtain and use a copy of the yarn-site.xml and core-site.xml from the cluster.

In some cases, the required Hadoop client configuration requires the DQ Agent to run on a Hadoop Edge node within the cluster. This can happen for a number of reasons. Namely, native dependency packages are required, network isolation from the subnet that hosts the DQ server, complex security configuration, and so on. In these circumstances, you can deploy the DQ Agent on a cluster Edge Node that contains the required configurations and packages. In this setup, the DQ Agent uses the existing Hadoop configuration and packages to run DQ checks on the Hadoop cluster.

Before you begin

Before you can configure Hadoop, create the config folder. Execute the following commands:

```
cd $OWL_HOME
mkdir -p config/hadoop
echo "export HADOOP_CONF_DIR=$OWL_HOME/config/hadoop" >> con-
fig/owl-env.sh
bin/owlmanage.sh restart=owlagent
```

Minimum configuration settings

The following is a minimal configuration with Kerberos and TLS disabled. It is typically only applicable in Cloud Hadoop scenarios (EMR/Dataproc/HDI). Cloud Hadoop clusters are

ephemeral and do not store any data, because the data is stored and secured in Cloud Storage.

```
export RESOURCE_MANAGER=<yarn-resource-manager-host>
export NAME_NODE=<namenode>

echo "
<configuration>
  <property>
    <name>hadoop.security.authentication</name>
    <value>simple</value>
  </property>
  <property>
    <name>hadoop.rpc.protection</name>
    <value>authentication</value>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://$NAME_NODE:8020</value>
  </property>
</configuration>
" >> $OWL_HOME/config/hadoop/core-site.xml

echo "
<configuration>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>$RESOURCE_MANAGER:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>$RESOURCE_MANAGER:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>$RESOURCE_MANAGER:8088</value>
  </property>
</configuration>
" >> $OWL_HOME/config/hadoop/yarn-site.xml
```

Configure cloud storage

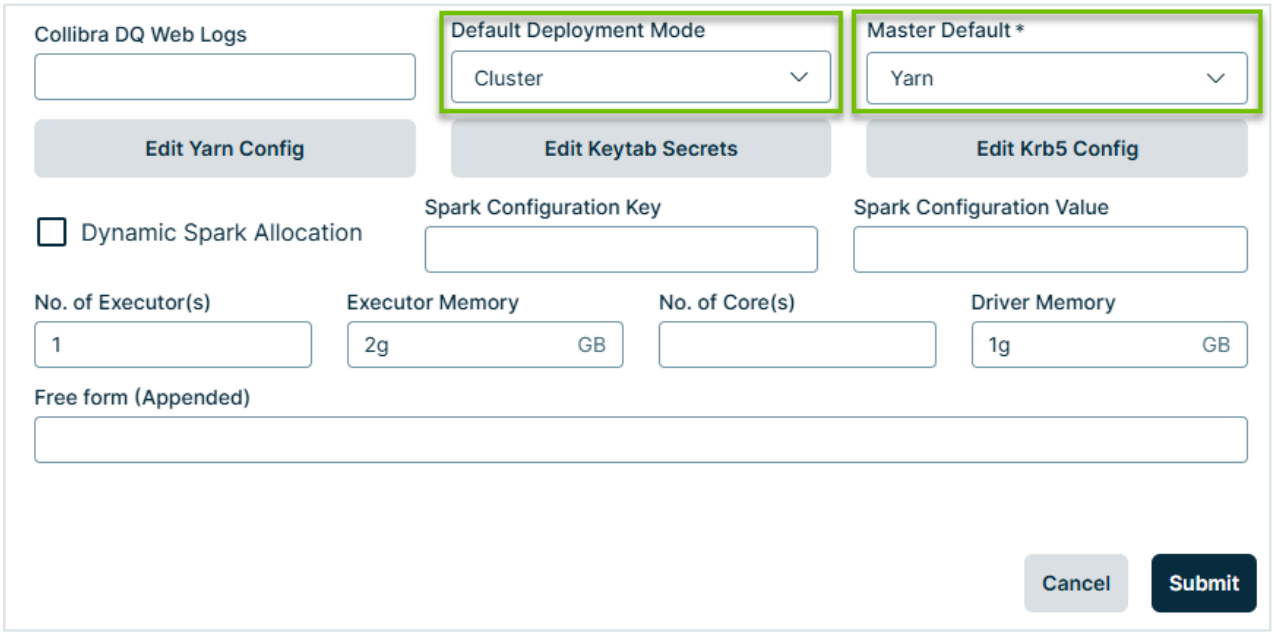
When deploying a Cloud Service Hadoop cluster from any of the major Cloud platforms, it is possible to use Cloud Storage rather than HDFS for dependency package staging and distribution.

1. Create a new storage bucket.
2. Provide both the Hadoop cluster and the instance running the DQ Agent access to the bucket. In most cases, use a Role that is attached to the infrastructure. For example, AWS Instance Role with bucket access policies.
3. In core-site.xml, set `fs.defaultFS` to the bucket path instead of HDFS.

Configure the DQ Agent for Hadoop

After creating the Hadoop client configuration, use the Agent Configuration console to configure the DQ Agent to use Hadoop.

1. Sign in to your web instance of Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
3. Click **Agent Configuration**.The Agent Configuration page opens.
4. In the list of DQ Agents, find the agent that you want to modify. Under Actions, click the **Actions** button and click **Edit**. The Edit Agent modal displays.
5. Set the **Master Default** field to **Yarn**.
6. Set the **Default Deployment Mode** to **Cluster**.



7. Click **Submit**.

Secure configuration settings using Kerberos

Use Kerberos to secure Hadoop clusters that are deployed on-premises, because they are typically multi-tenant and not ephemeral. Additionally, enable TLS for all HTTP endpoints. HDFS may also be configured for a more secure communication with additional RPC encryption.

```
export RESOURCE_MANAGER=<yarn-resoruce-manager-host>
export NAME_NODE=<namenode>
export KERBEROS_DOMAIN=<kerberos-domain-on-cluster>
export HDFS_RPC_PROTECTION=<authentication || privacy || integ-
rity>

echo "
<configuration>
  <property>
    <name>hadoop.security.authentication</name>
    <value>kerberos</value>
  </property>
  <property>
    <name>hadoop.rpc.protection</name>
    <value>$HDFS_RPC_PROTECTION</value>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://$NAME_NODE:8020</value>
  </property>
</configuration>
" >> $OWL_HOME/config/hadoop/core-site.xml

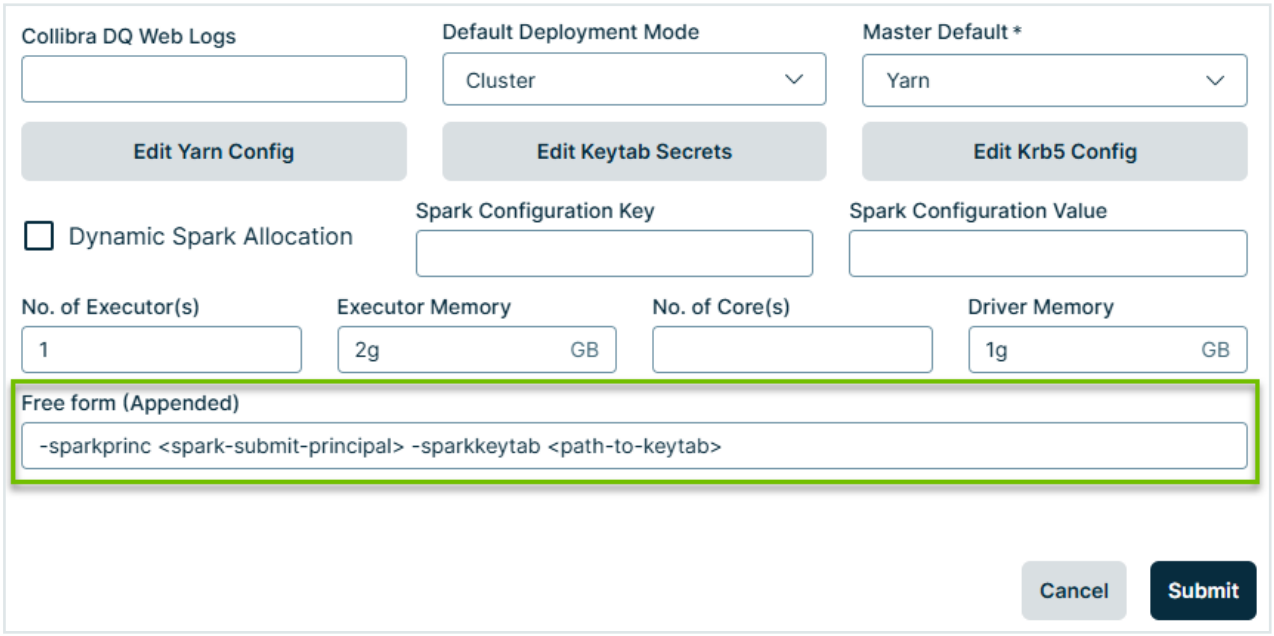
echo "
<configuration>
  <property>
    <name>hadoop.security.authentication</name>
    <value>HDFS/_HOST@$KERBEROS_DOMAIN</value>
  </property>
</configuration>
" >> $OWL_HOME/config/hadoop/hdfs-site.xml

echo "
<configuration>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>$RESOURCE_MANAGER:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>$RESOURCE_MANAGER:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.https.address</name>
    <value>$RESOURCE_MANAGER:8090</value>
  </property>
</configuration>
" >> $OWL_HOME/config/hadoop/yarn-site.xml
```

Configure DQ Agent for Kerberos

When the target Hadoop cluster is secured by Kerberos, DQ checks (Spark jobs) require a Kerberos credential. This typically means that the DQ Agent will need to be configured to include a Kerberos keytab with each DQ check.

1. Sign in to your web instance of Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
3. Click **Agent Configuration**.The Agent Configuration page opens.
4. In the list of DQ Agents, find the agent that you want to modify. Under Actions, click the **Actions** button and click **Edit**. The Edit Agent modal displays.
5. In the **Free form (Appended)** field, enter the following command:
   `-sparkprinc <spark-submit-principal> -sparkkeytab <path-to-keytab>`



6. Click **Submit**.

What's next?

Configure AWS EMR

Configure AWS EMR

Each cloud provider has unique steps and configuration options for running Apache Hadoop clusters. This section describes how to configure Amazon Web Services (AWS) Elastic MapReduce (EMR).

For additional information on Hadoop integration, go to Configuring Apache Hadoop to Execute DQ Jobs.

Data Quality & Observability Classic on EMR Architecture

Collibra DQ is able to use EMR as the compute space for DQ jobs. While it is possible to simply operate a long running EMR cluster, EMR's intended operating model is ephemeral infrastructure. Using EMR as an ephemeral compute space is the most cost effective approach both in terms of operational effort and infrastructure costs. Collibra DQ makes it possible to seamlessly leverage EMR in this operating model.

When there is not an EMR cluster available, Collibra DQ users are still able to browse datasets and DQ results in DQ Web. However, if a user attempts to deploy a DQ Job, they will see a red light icon next to the target agent. If the user still wants to request a DQ Job, it will simply wait in queue until the target agent comes back online the next time an EMR cluster is available.

Prepare for deployment

> **Note** Before allowing Collibra DQ to use EMR as the compute space, make sure that DQ Web and the DQ Metastore are already deployed. For more installation details, see Standalone Install.

1. Create a "bootstrap bucket" location in S3 where Collibra DQ binaries and bootstrap script (`install-agent-emr.sh`) will be staged. The EMR cluster instances will need to include an attached Role that has access to this location in order to bootstrap the cluster. This location should not contain any data or any kind of sensitive materials and thus should not require any special permissions. It just needs to be accessible by EMR clusters for bootstrap purposes.

2. Create or modify an instance Profile Role that will be attached to EMR clusters so that it enables read access to the bootstrap bucket location. This Role is separate from the EMR service role that EMR will use to deploy the infrastructure.
3. Stage the bootstrap script and the Collibra DQ binary package in the bootstrap location created above.
4. Make sure that the VPC where the Collibra DQ Metastore is deployed is accessible from the VPC where EMR clusters will be deployed.
5. Make sure that Security Groups applied to the Collibra DQ Metastore are configured to allow access from EMR master and worker Security Groups.
6. Decide whether to use EMR 5.x or EMR 6.x. This is important because EMR 6 introduces Spark 3 and Scala 2.12. If EMR 6 is chosen, make sure Collibra DQ binaries were compiled for Spark 3 and Scala 2.12.
7. (OPTIONAL) Create and store a private key to access EMR instances.

Deploy an EMR cluster

There are several ways to deploy EMR, but for dev-ops purposes, the best path is usually to use the AWS CLI utility. The example below will deploy an EMR cluster bootstrapped with Collibra DQ binaries and a functioning agent to deploy DQ Jobs.

> **Note** When defining the Bootstrap Location argument, do not include "s3://". For example: If Bootstrap Location is s3://bucket/prefix then BOOTSTRAP_LOCATION="bucket/prefix".

> **Warning**
> Before you deploy EMR, edit `install-agent-emr.sh` and add the following export variables to it for the setup script to run correctly:
>
> ```
> export DQ_ADMIN_USER_PASSWORD
> export DQ_ADMIN_USER_EMAIL
> ```

> **Important**
> As of Collibra DQ version 2023.09, we changed our package names from `owl-*` to `dq-*`. Therefore, you need to replace the following lines in your EMR file referencing **owl** with **dq**:
> `aws s3 cp s3://${OWL_PACKAGE_LOCATION}/dq-$OWL_VERSION-package-`

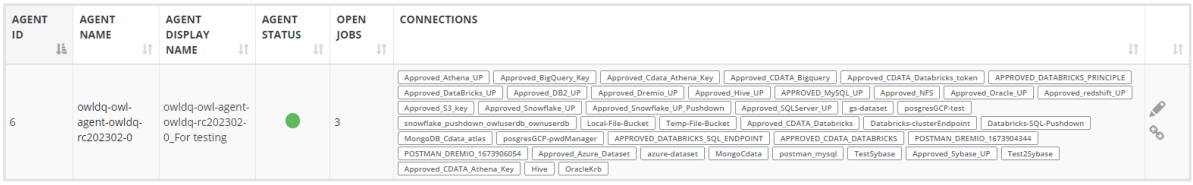
```
full.tar.gz ./
tar -xvf dq-$OWL_VERSION-package-full.tar.gz
```

```
aws emr create-cluster \
                --auto-scaling-role EMR_AutoScaling_DefaultRole
\
                --applications Name=Hadoop Name=Spark Name=Hive
Name=Tez \
                --name owl-emr \
                --release-label emr-6.2.0 \
                --region ${EMR_REGION} \
                --ebs-root-volume-size 10 \
                --scale-down-behavior TERMINATE_AT_TASK_
COMPLETION \
                --enable-debugging \
                --bootstrap-actions \
                "[{\"Path\":\"s3://${BOOTSTRAP_LOCATION}/in-
stall-agent-emr.sh\", \
                \"Args\":[ \
                \"${OWL_VERSION}\", \
                \"${OWL_AGENT_ID}\", \
                \"${METASTORE_HOST}:${METASTORE_
PORT}/${METASTORE_DB}?currentSchema=owlhub\", \
                \"${METASTORE_USER}\", \
                \"${METASTORE_PASSWORD}\", \
                \"${BOOTSTRAP_LOCATION}\", \
                \"${LICENSE_KEY}\", \
                \"native\"], \"Name\":\"install-owl-agent\"}]" \
                --ec2-attributes "{ \
                \"KeyName\":\"${EMR_INSTANCE_PRIVATE_KEY_
NAME}\", \
                \"InstanceProfile\":\"${BOOTSTRAP_ACCESS_
ROLE}\", \
                \"SubnetId\":\"${EMR_SUBNET}\", \
                \"EmrManagedSlaveSecurityGroup\":\"${EMR_WORKER_
SECURITY_GROUP}\", \
                \"EmrManagedMasterSecurityGroup\":\"${EMR_
WORKER_SECURITY_GROUP}\" \
                }" \
                --service-role ${EMR_SERVICE_ROLE} \
                --log-uri s3n://${EMR_LOG_LOCATION} \
                --instance-groups "[ \
                {\"In-
stanceCoun-
t\":1,\"InstanceGroupType\":\"MASTER\",\"InstanceType\":\"${EMR_
MASTER_INSTANCE_TYPE}\",\"Name\":\"Master - 1\"}, \
                {\"In-
stanceCoun-
t\":3,\"InstanceGroupType\":\"CORE\",\"InstanceType\":\"${EMR_
CORE_INSTANCE_TYPE}\",\"Name\":\"Core - 2\"} \
            ]"
```

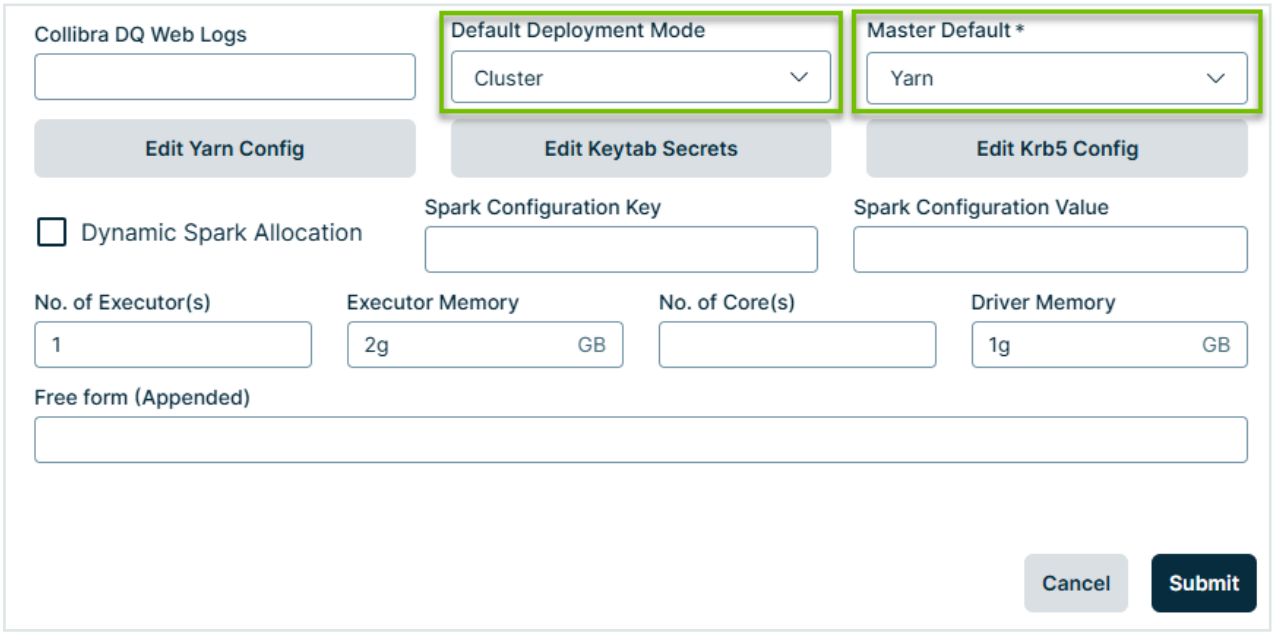
Configure DQ Agent

Once the EMR cluster and the DQ Agent is deployed, it needs to be configured in DQ Web.

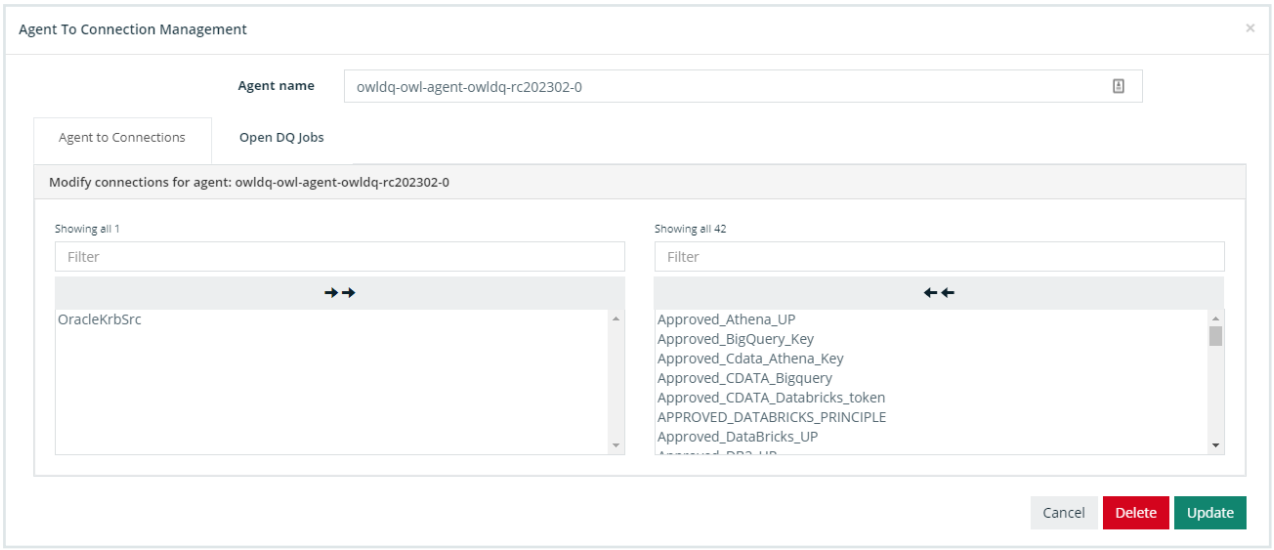
1. Sign in to your web instance of Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
3. Click the **Remote Agent** tile.
   » The **Agent Management** page opens.
   » In the following example, the newly created agent has a green indicator to show that it is healthy and active.



4. Click the pencil icon to the right of the Connections column.
   » The **Edit Agent** modal appears.
5. Set the **Master Default** field to **Yarn**.
6. Set the **Default Deployment Mode** to **Cluster**.



7. Click **Submit**.
8. On the Agent Management page, click the chain link icon next to the Connections column to map the DQ Agent to Connections.
   » The **Edit Agent** modal appears.



> **Note** Any data sources that are not listed in the right hand pane are not visible to the agent. Refer to the Agent section for more details.

9. Click **Update**.

Run DQ Jobs

You can now use EMR to run DQ Jobs on your data.

1. In the DQ Web, open the Explorer page.
2. Select a connection, schema, and table.
3. Click **Create DQ Job**.

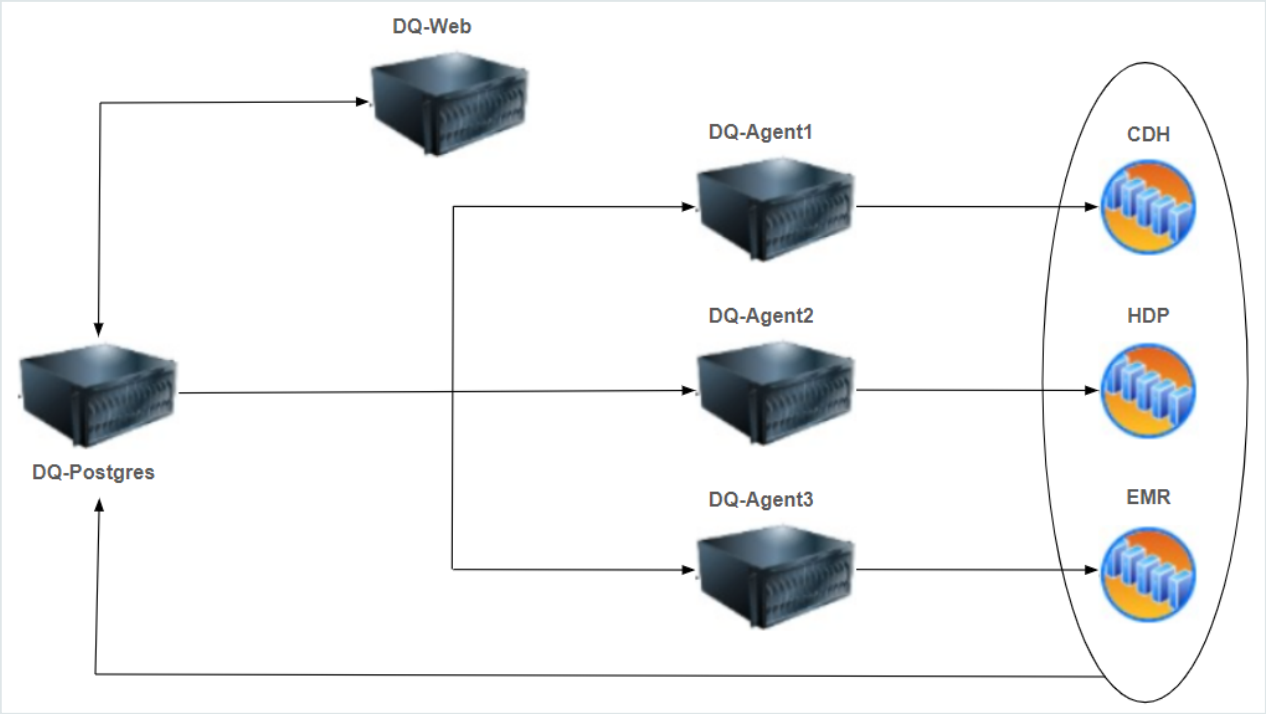
For more information on DQ Jobs, go to Explorer.

Agent

When an agent picks up a DQ Job, it launches the job either locally on the agent node itself or on a cluster as a Spark job (if the agent is set up as an edge node of the cluster). Depending on where the job launches, the results of the DQ Job will write back to the Metastore. The results then display in the DQ Web App, are exposed as REST API, and become available for direct SQL query against the Metastore.

The diagram below provides a high-level overview of how agents work within Collibra DQ. Job execution is driven by DQ Jobs that are written to an `agent_q` table inside the Metastore (**DQ-**

Postgres) via the Web App or REST API endpoint. Each active and available agent queries the DQ-Postgres table every 5 seconds to execute DQ Jobs for which the agent is responsible. For example, the EMR agent **DQ-Agent3** only executes DQ Jobs scheduled to run on EMR.



What's next?

- Set up a DQ agent
- Map data source connections to a DQ agent
- Manage agents (for admins)

Set up a DQ agent

This topic shows you how to configure a DQ agent.

Setting up a DQ Agent with `setup.sh` as part of the DQ package

Use the `setup.sh` script located in `/opt/owl/` (or other **Base Path** that your installation used). See the example in the code block below for setting up a DQ Agent with a Postgres server running `localhost` on port `5432` with database postgres and Postgres username/password combo `postgres/password`.

```
# PATH TO DIR THAT CONTAINS THE INSTALL DIR
                export BASE_PATH=/opt

                # PATH TO AGENT INSTALL DIR
                export INSTALL_PATH=/opt/owl

                # DQ Metadata Postgres Storage settings
                export METASTORE_HOST=localhost
                export METASTORE_PORT=5432
                export METASTORE_DB=postgres
                export METASTORE_USER=postgres
                export METASTORE_PASSWORD=password

                cd $INSTALL_PATH

                # Install DQ Agent only
                ./setup.sh \
                -owlbase=$BASE_PATH \
                -options=owlagent \
                -pguser=$METASTORE_USER \
                -pgpassword=$METASTORE_PASSWORD \
            -pgserver=${METASTORE_HOST}:${METASTORE_
PORT}/${METASTORE_DB}
```

The setup script automatically generates the `/opt/owl/config/owl.properties` file and encrypts the provided password.

Setting up a DQ agent manually

1. Open a terminal session and go to the directory with the installer.
2. Run the following command to encrypt your DQ Metastore password before it is stored in the `/opt/owl/config/owl.properties` file:

```
# PATH TO AGENT INSTALL DIR
                        export INSTALL_PATH=/opt/owl

                        cd $INSTALL_PATH

                        #Encrypt DQ Metadata Postgres Stor-
age password
            ./owlmanage.sh encrypt=password
```

> **Note** `owlmanage.sh` generates an encrypted string for the plain text password input. You can use the encrypted string in the `/opt/owl/config/owl.properties` configuration file to avoid exposing the DQ Metadata Postgres Storage password.

3. Run the following command to open the `/opt/owl/config/owl.properties` configuration file:

```
vi $INSTALL_PATH/config/owl.properties
```

4. Add the following properties to the configuration file:

```
spring.datasource.url=jdbc:postgresql://{DB_HOST}:{DB_PORT}/
{METASTORE_DB}
                        spring.datasource.username=
{METASTORE_USER}
                        spring.datasource.password=
{METASTORE_PASSWORD}
                        spring.datasource.driver-class-name-
e=com.owl.org.postgresql.Driver

                        spring.agent.datasource.url=jdbc:pos-
tgresql://{DB_HOST}:{DB_PORT}/{METASTORE_DB}
                        spring.agent.datasource.username=
{METASTORE_USER}
                        spring.agent.datasource.password=
{METASTORE_PASSWORD}
            spring.agent.datasource.driver-class-name-
e=org.postgresql.Driver
```

5. Restart the DQ Web App.

Setting up the DQ Agent from the Admin Console

1. On the Collibra DQ home page, hover your pointer over ⚙ **Settings** and select **Admin Console**.
   » The Admin Console opens.
2. Click **Agent Configuration**.
   » The Agent Configuration page opens.
3. In the Actions column, click to edit your agent.

» The **Edit Agent** modal appears.

4. Enter the required information.

Field	Description
Agent Id	The numerical identifier of your agent. For example, *6*. This field auto-generates and cannot be edited.
Agent Name	The unique name of your agent. This field auto-generates and cannot be edited.
Agent Display Name	The descriptive name of your agent that displays anywhere agent information is present in the DQ Web App. You can customize the Agent Display Name to make it easier to identify your agent. **Tip** There are no character restrictions for the Agent Display Name field, but it is best practice to use only alphanumeric characters, hyphens, and underscores.
Is Local	Select this option for Hadoop deployments only.
Use Livy	Not applicable.
Livy Host	Not applicable.
Base Path	The installation folder path for DQ. All other paths in the DQ Agent are relative to this installation path. This is the location that is set as `OWL_BASE` in Full Standalone Setup and other installation setups followed by `owl/` folder. For example, if the setup command is `export OWL_BASE=/home/centos` then the **Base Path** in the Agent configuration should be set to `/home/centos/owl/`. Default: `/opt/owl/`.
Collibra DQ Core JAR	The file path to the DQ Core jar file. Default `<Base Path>/owl/bin/`

Field	Description
Collibra DQ Core Logs	The folder path where DQ Core logs are stored. Logs from DQ Jobs are stored in this folder. Default: `<Base Path>/owl/log`
Collibra DQ Script	The file path to DQ execution script `owlcheck.sh`. This script is used to run DQ Job via command line without using the agent. Using `owlcheck.sh` for running DQ Jobs is superseded by DQ Agent execution model. Default: `<Base Path>/owl/bin/owlcheck`
Collibra DQ Web Logs	The folder path where DQ Web logs are stored. Logs from the DQ Web App are stored in this folder. Default: `<Base Path>/owl/log`
Default Queue	Only used for Yarn.
Deploy Deployment Mode	The Spark deployment mode can be either **Client** or **Cluster**. While we recommend **Cluster**, there are best practices to follow: ◦ If you only have one Spark Worker node, it is best practice to select **Client**. ◦ If you have more than one Spark Worker node, it is best practice to select **Cluster**.
Default Master	The Spark Master URL copied from the Spark cluster verification screen. For example, `spark://...`
Dynamic Spark Allocation	Not applicable.
Spark Configuration Key	Not applicable.
Spark Configuration Value	Not applicable.

Field	Description
Number of Executor(s)	The default number of executors allocated per DQ Job when using this Agent to run DQ Scans. The default is 1.
Executor Memory	The default RAM per executors allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1 gigabyte.
Number of Core (s)	The default number of cores per executors allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1.
Driver Memory	The default driver RAM allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1 gigabyte.
Free Form (Appended)	Other `spark-submit` parameters to append to each DQ Job when using this Agent to run DQ Scans.

Field	Description
Agent Id	The numerical identifier of your agent. For example, *6*. This field auto-generates and cannot be edited.
Agent Name	The unique name of your agent. This field auto-generates and cannot be edited.
Agent Display Name	The descriptive name of your agent that displays anywhere agent information is present in the DQ Web App. You can customize the Agent Display Name to make it easier to identify your agent.

> **Tip** There are no character restrictions for the Agent Display Name field, but it is best practice to use only alphanumeric characters, hyphens, and underscores.

Field	Description
Is Local	You can select this option to form the driver location path, which is normally applicable only when you run your agent in the master or edge node.
Use Livy	Not applicable.
Livy Host	Not applicable.
Base Path	The installation folder path for DQ. All other paths in the DQ Agent are relative to this installation path. This is the location that is set as `OWL_BASE` in Full Standalone Setup and other installation setups followed by `owl/` folder. For example, if the setup command is `export OWL_BASE=/home/centos` then the **Base Path** in the Agent configuration should be set to `/home/centos/owl/`. Default: `/opt/owl/`
Collibra DQ Core JAR	The file path to the DQ Core jar file. Default: `<Base Path>/owl/bin/`
Collibra DQ Core Logs	The folder path where DQ Core logs are stored. Logs from DQ Jobs are stored in this folder. Default: `<Base Path>/owl/log`
Collibra DQ Script	The file path to the DQ execution script `owlcheck.sh`. This script is used to run DQ Jobs via the command line without using an agent. Using `owlcheck.sh` for running DQ Jobs is superseded by the DQ Agent execution model. Default: `<Base Path>/owl/bin/owlcheck`
Collibra DQ Web Logs	The folder path where DQ Web logs are stored. Logs from the DQ Web App are stored in this folder. Default: `<Base Path>/owl/log`.

Field	Description
Default Queue	The default resource queue to submit jobs.
Default Deployment Mode	The Spark deployment mode for Yarn is **Cluster**.

Field	Description
Default Master	Set to `Yarn`. Click **Edit Yarn Config** to ensure you have the necessary Hadoop xml files. Edit the file templates as necessary:

XML File	Description
core-site.xml	Contains information about where authentication protocol, HDFS_RPC_PROTECTION, and the NAME_NODE run in the Hadoop cluster. Show example template

```
<configuration>
  <property>
    <name>hadoop.security.authentica
tion</name>
    <value>kerberos</value>
  </property>
  <property>
    <name>hadoop.rpc.protection</nam
e>
    <value>$HDFS_RPC_
PROTECTION</value>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://$NAME_
NODE:8020</value>
  </property>
</configuration>
```

Field	Description

XML File	Description

hdfs-site.xml	Contains the configuration settings for authentication protocol, the NAME_NODE, and DATA_NODE.
	Show example template
	`<configuration>`
	`  <property>`
	`  <name>hadoop.security.authentication</name>`
	`  <value>HDFS/_HOST@$KERBEROS_DOMAIN</value>`
	`  </property>`
	`</configuration>`

yarn-site.xml	Contains the Yarn resource manager settings.
	Show example template
	`<configuration>`
	`  <property>`
	`  <name>yarn.resourcemanager.scheduler.address</name>`
	`  <value>$RESOURCE_MANAGER:8030</value>`
	`  </property>`
	`  <property>`
	`  <name>yarn.resourcemanager.address</name>`
	`  <value>$RESOURCE_MANAGER:8032</value>`
	`  </property>`
	`  <property>`
	`  <name>yarn.resourcemanager.webapp.https.address</name>`
	`  <value>$RESOURCE_MANAGER:8090</value>`
	`  </property>`
	`</configuration>`

Field	Description
Dynamic Spark Allocation	Not applicable.
Spark Configuration Key	Not applicable.
Spark Configuration Value	Not applicable.
Number of Executor(s)	The default number of executors allocated per DQ Job when using this Agent to run DQ Scans. The default is 1.
Executor Memory	The default RAM per executors allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1 gigabyte.
Number of Core(s)	The default number of cores per executors allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1.
Driver Memory	The default driver RAM allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1 gigabyte.
Free Form (Appended)	Other `spark-submit` parameters to append to each DQ Job when using this Agent to run DQ Scans.

> **Note** Ensure that your service account has the permission to launch Spark executor pods. Refer to the executor launch template and permissions.

Field	Description
Agent Id	The numerical identifier of your agent. For example, *6*. This field auto-generates and cannot be edited.
Agent Name	The unique name of your agent. This field auto-generates and cannot be edited.
Agent Display Name	The descriptive name of your agent that displays anywhere agent information is present in the DQ Web App. You can customize the Agent Display Name to make it easier to identify your agent. **Tip** There are no character restrictions for the Agent Display Name field, but it is best practice to use only alphanumeric characters, hyphens, and underscores.
Is Local	Select this option for Hadoop deployments only.
Use Livy	Not applicable.
Livy Host	Not applicable.
Base Path	The installation folder path for DQ. All other paths in the DQ Agent are relative to this installation path. This is the location that is set as `OWL_BASE` in Full Standalone Setup and other installation setups followed by `owl/` folder. For example, if the setup command is `export OWL_BASE=/home/centos` then the **Base Path** in the Agent configuration should be set to `/home/centos/owl/`. Default: `/opt/owl/`
Collibra DQ Core JAR	The file path to the DQ Core jar file. Default: `<Base Path>/owl/bin/`

Field	Description
Collibra DQ Core Logs	The folder path where DQ Core logs are stored. Logs from DQ Jobs are stored in this folder. Default: `<Base Path>/owl/log`
Collibra DQ Script	The file path to DQ execution script `owlcheck.sh`. This script is used to run DQ Job via command line without using agent. Using `owlcheck.sh` for running DQ Jobs is superseded by DQ Agent execution model. Default: `<Base Path>/owl/bin/owlcheck`.
Collibra DQ Web Logs	The folder path where DQ Web logs are stored. Logs from the DQ Web App are stored in this folder. Default: `<Base Path>/owl/log`
Default Queue	Only used for Yarn.
Default Deployment Mode	The Spark deployment mode for Kubernetes is **Cluster**.
Default Master	The Kubernetes Master URL copied from the Kubernetes cluster verification screen. Set this value to `k8s://` instead of a specific URL. When you leave this value set to `k8s://`, this helps Collibra DQ auto-discover the High Availability endpoint of the Kubernetes control plane at runtime. **Warning** Only set this to a specific URL, such as `k8s://{hostname}:443`, if you are an advanced user or if your specific use case requires it.
Dynamic Spark Allocation	Not applicable.

Field	Description
Spark Configuration Key	Not applicable.
Spark Configuration Value	Not applicable.
Number of Executor(s)	The default number of executors allocated per DQ Job when using this Agent to run DQ Scans. The default is 1.
Executor Memory	The default RAM per executors allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1 gigabyte.
Number of Core (s)	The default number of cores per executors allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1.
Driver Memory	The default driver RAM allocated per DQ Job when using this Agent to run DQ Scans. Go to Hardware Sizing for more information. The default is 1 gigabyte.
Free Form (Appended)	Other `spark-submit` parameters to append to each DQ Job when using this Agent to run DQ Scans.

> **Note** If you bring in your own Spark executor pod launch template, ensure that the service account used to launch Spark executor pods has the permission to do so. Refer to the executor launch template and for more information.

5. Click **Save**.

What's next?

Map data source connections to a DQ agent

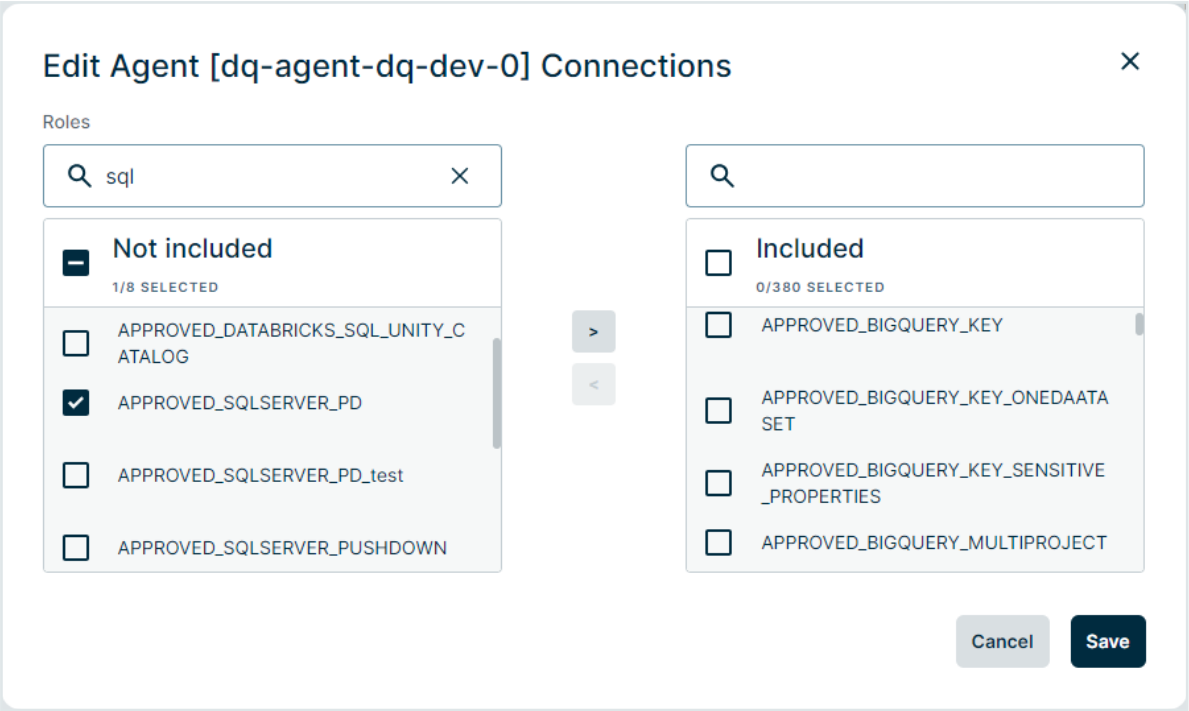
Map data source connections to a DQ agent

When you add new data source connections, the DQ agent requires permission to run DQ Jobs against data within them. This topic shows you how to map connections to an agent, to grant permission to the connection.

Steps

1. On the Collibra DQ home page, hover your pointer over ⚙ **Settings** and select **Admin Console**.
   » The Admin Console opens.
2. Click **Agent Configuration**.
   » The Agent Configuration page opens.
3. In the last column of the Agents table, to the right, click the chain link icon to link your agent to data source connections.
   » The **Agent to Connection Management** wizard appears.

   > **Note** The left panel contains a list of available connections that are not yet linked to the DQ agent and do not yet have permission to run DQ Jobs. The right panel contains a list of connections that are linked to the DQ agent and have permission to run DQ Jobs.



4. Click a connection in the left panel to link connections one at a time or click the double arrow icon to link all available connections at the same time.
5. Click **Update**.

> **Tip** You can unlink connections with the same methods listed above, but click the connections listed in the right panel instead of the left. Successfully unlinked connections appear in the left panel.

Upgrade Spark

> **Important** You must upgrade to Java 17 and Spark 3.5.3 to install and use Data Quality & Observability Classic 2025.02. For more information about version compatibility, click "Java and Spark compatibility matrix" below.

Java and Spark compatibility matrix[1]

1

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

This section details how to upgrade on-premises Apache Spark versions.

> **Important** We recommend upgrading to Spark 3.5.3, to address various critical vulnerabilities present in the Spark core library, including Log4j.

Steps

> **Note** The below steps are intended to reflect examples of what works in a simple Standalone environment. You may need to modify them to accommodate your specific deployment.

1. Run the following commands to stop Spark and Collibra DQ services.
    a. Set the **OWL_HOME** variable without a trailing slash after the value.

```
export OWL_HOME=<the owl folder where CDQ is installed>
```

> **Example** `export OWL_HOME=/home/ec2-user/owl`

    b. Stop Spark Master.

```
cd $OWL_HOME/spark/sbin
./stop-master.sh
```

    c. Stop Spark Worker.

```
cd $OWL_HOME/spark/sbin
./stop-worker.sh
```

    d. Stop DQ Web.

```
cd $OWL_HOME/bin
 ./owlmanage.sh stop=owlweb
```

    e. Stop DQ Agent.

```
cd $OWL_HOME/bin
./owlmanage.sh stop=owlagent
```

f. Verify that all processes are stopped.

```
ps -ef | grep owl

## No DQ processes should return as running. ##
```

> **Tip** If any DQ processes are still running, you can use the command `kill -9 <pid from the ps command in Step 1f>` to stop them manually.

2. Create a package installation folder one level up from **OWL_HOME** and set the environment variables accordingly.
   a. Move up one level from **OWL_HOME**.

```
cd $OWL_HOME/..
```

   b. Export the **INSTALLER_DIRECTORY** variable.

```
export INSTALLER_DIRECTORY=$OWL_HOME/dq_install
```

   c. Create the **INSTALLER_DIRECTORY** folder.

```
mkdir $INSTALLER_DIRECTORY
```

3. Create a backup folder.
   a. Open the **INSTALLER_DIRECTORY** folder.

```
cd $INSTALLER_DIRECTORY
```

   b. Create a backup folder.

```
mkdir backup
```

4. Download the Collibra DQ installer packaged with the necessary Spark version from the Data Quality & Observability Classic Downloads page in the Product Resource Center.
5. Run the following command to extract the installer.

```
tar -xvf <downloaded .tar.gz file>
```

6. Create backups of the following directories and files by moving or copying them to the backup folder.
    - spark
    - spark-extras (if it exists)
    - drivers
    - dq-core-*-dependencies.jar
    - dq-webapp*.jar
    - dq-agent*.jar
    - owlcheck
    - owlmanage.sh
7. Remove the folders and files using the following or similar steps.
    a. Open the **OWL_HOME/bin** folder.

```
cd $OWL_HOME/bin
```

    b. Move the dq-core-*-dependencies.jar, dq-webapp*.jar, dq-agent*.jar, owlcheck, and owlmanage.sh files to the **INSTALLER_DIRECTORY/backup/** folder.

```
mv dq*.jar owl* $INSTALLER_DIRECTORY/backup/
```

    c. Move up one folder level.

```
cd ..
```

    d. Move drivers, spark, and spark-extras to the **INSTALLER_DIRECTORY/backup/** folder.

```
mv drivers spark spark-extras $INSTALLER_
DIRECTORY/backup/
```

8. Confirm that the files and folders are in the backup folder.

```
ls -ltr $INSTALLER_DIRECTORY/backup/
```

9. Run the following command to go to the **packages/install-packages** folder.

```
cd $INSTALLER_DIRECTORY/packages/install-packages
```

> **Example** You can use the command `tar -xvf spark-3.5.3-bin-hadoop3.tgz` to extract Spark 3.5.3.

10. Extract the Spark installation files.

```
tar -xvf spark-3.5.3-bin-hadoop3.tgz
```

11. Extract the **spark-extras.tar.gz** file.

```
tar -xvf spark-extras.tar.gz
```

12. Rename and move the **spark-3.5.3-bin-hadoop3** file from the installer extract folder to the **OWL_HOME/spark** folder.
    a. Rename the **spark-3.5.3-bin-hadoop3** file **spark**.

```
mv spark-3.5.3-bin-hadoop3 spark
```

    b. Move **spark** to the home directory.

```
mv spark $OWL_HOME
```

13. Run the following command to copy the **spark-extras** folder into **spark/jars**.

```
cp spark-extras/* $OWL_HOME/spark/jars
```

14. Add the following line of code to **owl/spark/conf/spark-env.sh** (which can be copied from the spark-env.sh.template file provided), to automate the cleanup of Spark work folders and avoid filling up disk space.

```
export SPARK_WORKER_OPTS="${SPARK_WORKER_OPTS} -Dspark.-
worker.cleanup.enabled=true -Dspark.-
worker.cleanup.interval=1800 -
Dspark.worker.cleanup.appDataTtl=3600"
```

15. Copy in the Data Quality & Observability Classic application files.
    a. Open the **INSTALLER_DIRECTORY**.

    ```
    cd $INSTALLER_DIRECTORY
    ```

    b. Copy in the dq-core-*-dependencies.jar, dq-webapp*.jar, and dq-agent*.jar files.

    ```
    cp dq-*.jar $OWL_HOME/bin
    ```

    c. Copy in the owlcheck and owlmanage.sh files.

    ```
    cp owl* $OWL_HOME/bin
    ```

    d. Make the owlcheck and owlmanage.sh files executable.

    ```
    chmod +x $OWL_HOME/bin/owl*
    ```

16. Extract and copy in the updated drivers.
    a. Create a **drivers** folder.

    ```
    mkdir drivers
    ```

    b. Open the **drivers** folder.

    ```
    cd drivers
    ```

    c. Extract **drivers.tar.gz** into the **drivers** folder.

    ```
    tar -xvf ../drivers.tar.gz
    ```

d. Move up one folder level.

```
cd ..
```

e. Move the **drivers/** folder to the application home directory.

```
mv drivers/ $OWL_HOME
```

17. Run the following commands to start Spark and Collibra DQ services.

a. Start Spark Master.

```
cd $OWL_HOME/spark/sbin
./start-master.sh
```

b. Start Spark Worker.

```
cd $OWL_HOME/spark/sbin
./start-worker.sh spark://$(hostname -f):7077
```

c. Start DQ Web.

```
cd $OWL_HOME/bin
./owlmanage.sh start=owlweb
```

d. Tail the log file until you see a "Bootstrap process complete" message.

```
tail -f $OWL_HOME/log/owl-web.log
```

Example

```
[main] INFO  com.owl.web.OwlWebAppBootstrap - Bootstrap process complete
```

e. Start DQ Agent.

```
cd $OWL_HOME/bin
./owlmanage.sh start=owlagent
```

18. Visit the Spark page on port 8080 to confirm that the Spark version is the one you installed and the worker is alive.
19. Create a test DQ job without any optional DQ layers or rules to verify that the Spark driver, executors, and containers are able to launch successfully.

Upgrade Data Quality & Observability Classic

This section describes how to upgrade from an older Data Quality & Observability Classic version to a new one.

Upgrade requirements

Before you start the upgrade, you need all of the following information to ensure an easy, successful installation process. This section only focuses on the requirements of Collibra DQ and does not take into account the connections to the data sources to ingest data.

> **Important** You must upgrade to Java 17 and Spark 3.5.3 to install and use Data Quality & Observability Classic 2025.02. For more information about version compatibility, click "Java and Spark compatibility matrix" below.

Java and Spark compatibility matrix[1]

1

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

User permissions

Log in with a user account that has privileges to:

- Create directories
- Launch scripts
- Start java, Spark, and Collibra DQ processes

`SUDO` is required if you are including the PostgreSQL metastore in the installation. `SUDO` is not required if you are using an external PostgreSQL metastore (recommended).

In addition, configure `ULIMIT` settings to 4096 or higher. DQ services typically consume approximately 428 threads, and each DQ job consumes an additional 400 threads. Setting `ULIMIT` to 4096 allows for approximately nine concurrent DQ jobs on a Standalone install.

System requirements

Supported operating systems

- Red Hat Enterprise Linux 8.x
- Red Hat Enterprise Linux 9.x

Hardware requirements

Small Tier - 16 Core, 128G RAM (r5.4xlarge / E16s v3)

Component	RAM	Cores
Web	2g	2
Postgres	2g	2
Spark	100g	10
Overhead	10g	2

Medium Tier - 32 Core, 256G RAM (r5.8xlarge / E32s v3)

Component	RAM	Cores
Web	2g	2
Postgres	2g	2
Spark	250g	26
Overhead	10g	2

Large Tier - 64 Core, 512G RAM (r5.16xlarge / E64s v3)

Component	RAM	Cores
Web	4g	3
Postgres	4g	3
Spark	486g	54
Overhead	18g	4

> **Important** Collibra DQ requires a limit of 2TBs for large tier jobs. For DQ jobs that exceed 2TBs, you must filter down columns or rows.

Estimates

Sizing should allow headroom and based on peak concurrency and peak volume requirements. If concurrency is not a requirement, you just need to size for peak volume (largest tables). Best practice to efficiently scan is to scope the job by selecting critical columns. See JDBC vs Local Data for more information.

Keep in mind that Spark allocates overhead memory on top of the requested memory. For example, an executor pod configured with 6GB of memory may actually request 6.8GB of memory. For more information, go to Apache Spark Memory Management.

Bytes per Cell	Rows	Columns	Gigabytes	Gigabytes for Spark (3x)
16	1,000,000.00	25	0.4	1.2
16	10,000,000.00	25	4	12
16	100,000,000.00	25	40	120
16	1,000,000.00	50	0.8	2.4
16	10,000,000.00	50	8	24
16	100,000,000.00	50	80	240
16	1,000,000.00	100	1.6	4.8
16	10,000,000.00	100	16	48
16	1,000,000,000.00	100	1600	4800
16	100,000,000.00	100	160	480
16	1,000,000.00	200	3.2	9.6
16	10,000,000.00	200	32	96
16	100,000,000.00	200	320	960
16	1,000,000,000.00	200	3200	9600

Cluster

If your program requires more horsepower or (Spark) workers than the example tiers above which is fairly common in Fortune 500 companies than you should consider the horizontal and ephemeral scale of a cluster. Common examples include Amazon EMR and

Cloudera CDP. Collibra DQ is built to scale up horizontally and can scale to hundreds of nodes.

Network requirements

Default Ports used by Collibra DQ

Port	Description
443	Connectivity to data sources (databases, file storage, etc.)
5432	DQ Metastore (PostgreSQL). Enables DQ Web, DQ Agent, and Spark services to connect to the DQ Metastore.
9000	DQ Web
9101	Exposes the Health Check API to check that the DQ Agent is running and stable.

Other

- If your current Spark version is 3.2.2 or older, Collibra strongly recommends upgrading to Spark 3.4.1 to address various critical vulnerabilities present in Spark core library, including Log4J. To determine which Spark version you are using, sign into your Collibra DQ instance and click the ⓘ in the upper-right corner of any page. The **Spark Version** lists your current Spark version.
    - If you are not already using Spark 3.4.1, follow the steps outlined in Upgrading Spark versions.

User permissions

Prerequisites

- Kubernetes cluster -- EKS, GKE, AKS, Openshift, Rancher
- Helm(v3)
- kubectl
- Cloud command line SDK, such as gcloud CLI, AWS CLI or similar
- External PostgreSQL DB version 11.9 and above, storage size 100GB, cores 4 to 8 memory to 4 to 8 GB
- Private container registry -- to store images
- LoadBalancer -- IngressController -- Ingress
- Egress networking access
- Helm Chart
- Images, image access key
- Minimum pod requirement -- 2 cores, 2GB RAM
- If you bring in your own Spark executor pod launch template, ensure that the service account used to launch Spark executor pods has the permission to do so. Refer to the executor launch template for more information.

System requirements

Supported Kubernetes versions

Data Quality & Observability Classic supports Kubernetes versions 1.29 through 1.31.

> **Note** As of February 2025, we recommend upgrading to Kubernetes version 1.29 or newer, as version 1.28 has reached its end of life.

Application system requirements

Component	Processor	Memory	Storage
Collibra DQ Web	1 core	2 GB	10 MB PVC
DQ Agent	1 core	1 GB	100 MB PVC
DQ Metastore	1 core	2 GB	10 GB PVC
Spark*	2 cores	2 GB	-

Keep in mind that Spark allocates overhead memory on top of the requested memory. For example, an executor pod configured with 6GB of memory may actually request 6.8GB of memory. For more information, go to Apache Spark Memory Management.

> **Note** * This is the minimum quantity of resources required to run an a Spark job in Kubernetes. This amount of resources would only provide the ability to scan a few megabytes of data with no more than a single job running at a given time. Proper sizing of the compute space must take into account the largest dataset that may be scanned, as well as the desired concurrency.

Network service considerations

DQ Web is the only required component that needs to be directly accessed from outside of Kubernetes. History Server is the only other component that can be accessed directly by users, however, it is optional.

If the target Kubernetes platform supports a LoadBalancer service type, you can configure the Helm Chart to directly deploy the externally accessible endpoint.

> **Note** For testing purposes, you can also configure the Helm chart to deploy a NodePort service type.

For the Ingress service type, deploy Collibra DQ without an externally accessible service and then attach the Ingress service separately. This applies when you use a third-party Ingress controller such as NGINX, Contour, etc.

> **Note** The Helm Chart is able to deploy an Ingress on GKE and EKS platforms, however, there is a wide variety of possible Ingress configurations that have not been tested.

What's next?

- Prepare your environment for an upgrade

Prepare your environment for an upgrade

> **Important** You must upgrade to Java 17 and Spark 3.5.3 to install and use Data Quality & Observability Classic 2025.02. For more information about version compatibility, click "Java and Spark compatibility matrix" below.

Java and Spark compatibility matrix[1]

1

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

General preparation

- Review all release notes after the version from which you are upgrading for any changes to critical areas, including:
  - Helm Charts
  - Security updates
  - Property configurations
  - Driver updates
  - API updates
- You have met the requirements specific to your deployment type.
- Create a backup of your DQ Metastore.

Gather the artifacts

- Review the Builds page for any optional drivers you may need (for Cloud Native deployments only).
- Obtain the Helm Charts (for Cloud Native deployments only).

What's next?

- Create and restore a backup of your DQ Metastore
- Upgrade Collibra Data Quality & Observability

Create and restore a backup

This section details how to create a backup of the Metastore prior to an upgrade.

Internal vs. external Metastore

When you create a backup of the Metastore, it's important to consider whether your deployment of Collibra DQ uses the internal Metastore located on the same server as your Collibra DQ application, or an external metastore separate from the Collibra DQ server. While most Cloud Native and Standalone deployments of Collibra DQ use an external Metastore, such as GCP Cloud SQL or AWS RDS, some Standalone deployments use an internal

Metastore. Regardless of your Metastore setup, it's imperative you have a backup copy of Collibra DQ that you can restore if needed.

Backup and restore

Create a backup

To create a snapshot of an internal Metastore, copy the contents within the `owl/postgres` folder created during the installation.

Restore a backup

To restore a snapshot of an internal Metastore, replace the contents of the `owl/postgres` folder created during installation with the `owl/postgres` folder you copied when you created the backup.

Create a backup

External Metastores are the most commonly used Metastores for both Cloud Native and Standalone deployments of Collibra DQ. Because the procedure for backing up external Metastores varies from one environment setup to the next, there are no formal steps for this process. However, we strongly recommend referring to the official PostgreSQL documentation for information about backing up PostgreSQL Metastores.

If your organization has modified the Collibra DQ Helm Charts for a Cloud Native deployment, including the creation of custom Kubernetes objects either inside or outside of the Helm Charts provided to you by Collibra, you need to create a backup of your Metastore before upgrading to a new version of Collibra DQ. The purpose of this is to ensure that the definitions of the custom Kubernetes objects are preserved and that you can recreate them, as necessary.

> **Important** Any custom changes made to Kubernetes objects, whether inside or outside Collibra-provided Helm Charts, are the responsibility of your organization.

If your organization uses the Helm Command and Helm Chart provided by Collibra to complete the end-to-end upgrade and deployment of new versions of Collibra DQ and have not created custom Kubernetes objects, we still recommend creating a backup of the Metastore. While our Helm Charts will automatically create all of the objects you need to successfully upgrade, it is best practice to create a snapshot of your Metastore to provide additional fault-tolerance.

Restore a backup

As external Metastores can be set up in a variety of ways, such as on a VM, Cloud SQL, or RDS, guidance on how to restore a backup naturally varies. We recommend reviewing the official PostgreSQL documentation for a better understanding of how you and your organization might approach restoring a backup of an external Metastore.

What's next?

- Upgrade Collibra Data Quality & Observability

Upgrade Collibra Data Quality & Observability

> **Note** Before proceeding with any upgrades, back up your Collibra DQ metastore.

> **Warning** Rolling back Collibra DQ to an older version is not supported. Please contact Collibra Support with any questions.

Important You must upgrade to Java 17 and Spark 3.5.3 to install and use Data Quality & Observability Classic 2025.02. For more information about version compatibility, click "Java and Spark compatibility matrix" below.

Java and Spark compatibility matrix[1]

1

Data Quality & Observability Classic version	Java 8	Java 11	Java 17	Spark versions	Additional notes
2025.01 and earlier	✓ Yes	✓ Yes	✗ No	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only)	
2025.02	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.03	✗ No	✗ No	✓ Yes	3.5.3 only	
2025.04	✓ Yes	✓ Yes	✓ Yes	• 2.3.0 (Java 8 only) • 2.4.5 (Java 8 only) • 3.0.1 (Java 8 and 11) • 3.1.2 (Java 8 and 11) • 3.2.2 (Java 8 and 11) • 3.4.1 (Java 11 only) • 3.5.3 (Java 17 only)	**Important** The **Java 8 and 11** build profiles only contain the 2025.02 release and critical bug fixes addressed in 2025.03 and 2025.04. They do not contain any feature enhancements from the 2025.03 or 2025.04 releases. Only the **Java 17** build profile contains feature enhancements and bug fixes listed in the 2025.04 release notes.

Download the Collibra DQ Upgrade Package

> **Note** All Collibra DQ customers upgrading or patching receive the full package, not the base package, and should follow the same upgrade steps described in this section.

You can download the DQ installation package from the Collibra Product Resource Center.

> **Note** For additional download options, contact your Account Team.

Save the package in a directory on the CDQ VM.

Extract the file:

```
tar -xvf dq-full-package.tar.gz
```

(Optional) Clean up:

```
rm dq-full-package.tar.gz
```

Upgrade steps

> **Important** Verify the Data Quality & Observability Classic version from which you are upgrading before proceeding with the upgrade. To do this, sign into your Collibra DQ instance and click the ⓘ in the upper-right corner of any page. The **App Version** lists your current Collibra DQ version.

1. Copy the contents of the provided package, for example, dq-<new-version>-<SPARK353>-package-full.tar.gz to the system being upgraded (extract contents).

   > **Tip** Untar the contents into a uniquely named folder, for example *2025-02-dq-upgrade*.

2. Stop the Collibra DQ Web process.

```
cd /owlhome/owl/bin
./owlmanage.sh stop=owlweb
```

3. Stop the Collibra DQ Agent process.

```
cd /owlhome/owl/bin
./owlmanage.sh stop=owlagent
```

4. Move the old jars from the **owl/bin** folder.

```
mv dq-webapp-<oldversion>-<spark322>.jar /tmp
mv dq-agent-<oldversion>-<spark322>.jar /tmp
mv dq-core-<oldversion>-<spark322>.jar /tmp
```

5. Copy the new jars into the **owl/bin** folder from the extracted package.

```
mv dq-webapp-<newversion>-<spark353>.jar
/home/owldq/owl/bin
mv dq-agent-<newversion>-<spark353>.jar
/home/owldq/owl/bin
mv dq-core-<newversion>-<spark353>.jar
/home/owldq/owl/bin
```

6. Copy the latest `owlcheck` and `owlmanage.sh` to **/opt/owl/bin** directory.

```
mv owlcheck /home/owldq/owl/bin
mv owlmanage.sh /home/owldq/owl/bin
```

7. Start the Collibra DQ Web application.

```
./owlmanage.sh start=owlweb
```

8. Start the Collibra DQ Agent.

```
./owlmanage.sh start=owlagent
```

9. Validate the number of active services.

```
ps -ef | grep owl
```

10.

> **Important** If you are installing a Data Quality & Observability Classic version 2025.02 or newer, you only need to follow steps 10a and 10b once. For instance, if you performed these steps while upgrading to 2025.02, you do not need to follow them again when you upgrade to 2025.03. These steps are not required for Data Quality & Observability Classic 2025.01 or older.

a. Add the following environment variable to the owl-env.sh file.

> **Note** If you already have this variable set, you need to update it to

include the following values.

```
export EXTRA_JVM_OPTIONS="--add-opens
java.base/java.util=ALL-UNNAMED --add-opens
java.base/java.net=ALL-UNNAMED --add-opens
java.base/sun.nio.ch=ALL-UNNAMED --add-opens
java.base/java.nio=ALL-UNNAMED --add-opens
java.base/sun.util.calendar=ALL-UNNAMED"
```

b. Restart the Collibra DQ Web and Collibra DQ Agent services.

```
./owlmanage.sh stop=owlweb
./owlmanage.sh stop=owlagent
./owlmanage.sh start=owlweb
./owlmanage.sh start=owlagent
```

Important If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `true` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For additional steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.

1. Copy the contents of the provided package, for example, dq-<new-version>-<SPARK341>-package-full.tar.gz to the system being upgraded (extract contents).

Tip Untar the contents into a uniquely named folder, for example *2024-11-dq-upgrade*.

2. Stop the Collibra DQ Web process.

```
cd /owlhome/owl/bin
./owlmanage.sh stop=owlweb
```

3. Stop the Collibra DQ Agent process.

```
cd /owlhome/owl/bin
./owlmanage.sh stop=owlagent
```

4. Move the old jars from the **owl/bin** folder.

```
mv dq-webapp-<oldversion>-<spark322>.jar /tmp
mv dq-agent-<oldversion>-<spark322>.jar /tmp
mv dq-core-<oldversion>-<spark322>.jar /tmp
```

5. Copy the new jars into the **owl/bin** folder from the extracted package.

```
mv owl-webapp-<newversion>-<spark341>.jar
/home/owldq/owl/bin
mv owl-agent-<newversion>-<spark341>.jar
/home/owldq/owl/bin
mv owl-core-<newversion>-<spark341>.jar
/home/owldq/owl/bin
```

6. Copy the latest `owlcheck` and `owlmanage.sh` to **/opt/owl/bin** directory.

```
mv owlcheck /home/owldq/owl/bin
mv owlmanage.sh /home/owldq/owl/bin
```

7. Start the Collibra DQ Web application.

```
./owlmanage.sh start=owlweb
```

8. Start the Collibra DQ Agent.

```
./owlmanage.sh start=owlagent
```

9. Validate the number of active services.

```
ps -ef | grep owl
```

> **Important** If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `true` in the owl-env.sh script, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For additional steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.

Upgrading data source drivers

When new data source drivers are available, they are listed in the Release Notes or recommended to you directly by Collibra. Determine which drivers need to be updated and follow these steps:

1. Review the release notes to see which drivers need to be updated.
2. From the previously extracted tarball provided to you by Collibra, locate the **drivers.tar.gz** file and extract the contents into a new directory called *drivers*.
3. Rename your **/owl/drivers** folder "**/owl/old_drivers/**" to create a backup of the old drivers.
4. Replace the drivers from the **/owl/old_drivers/** folder with the new drivers extracted from the tarball **/owl/drivers/**.

   > **Example** If you replace an old Databricks driver with a new one, the file path might look like **/owl/drivers/databricks**.

Deploy the new Collibra DQ Helm Chart on your Kubernetes cluster to upgrade to a new version of Collibra DQ.

1. Download the Docker **key.json** from the Object Store and run the following command.

```
docker login -u _json_key -p "$(cat client-repo-
key.json)" https://gcr.io
```

2. Using the Docker Pull Commands you obtained from the Collibra Downloads page, run one command per container to pull the Collibra DQ containers. The Docker pulls should follow a similar format to the ones in the following example:

```
docker pull gcr.io/owl-hadoop-cdh/dq-web:2025.02-
ABDGCSHILM-123
docker pull gcr.io/owl-hadoop-cdh/dq-agent:2025.02-
ABDGCSHILM-123
docker pull gcr.io/owl-hadoop-cdh/dq-livy:3.5.3-2025.02-
ABDGCSHILM-123
docker pull gcr.io/owl-hadoop-cdh/dq-spark:3.5.3-2025.02-
ABDGCSHILM-123
```

3. Tag the pulled containers and upload them to your internal Artifactory for deployment.
4. To deploy the new Collibra DQ Helm Chart on your Kubernetes cluster and upgrade to a new version of Collibra DQ, run the following Helm commands, replacing the placeholder values with the actual values described in the table below.

> **Important**
> The following example command may vary depending on the options your organization supplied during the initial installation.
>
> When upgrading, it's crucial to consider Helm properties provided through extra options or defaulting in Helm Charts to avoid losing any existing configurations.
>
> Be sure to replace the entire Helm Chart package with each release.

```
helm upgrade --install --namespace <namespace> \
--set global.version.dq=<cdq_version> \
--set global.version.spark=<cdq_spark_version> \
--set global.configMap.data.license_key=<cdq_license_key>
\
--set global.configMap.data.license_name=<your_license_
name> \
--set global.web.admin.email=${email} \
--set global.web.admin.password=${password} \
--set global.web.service.type=ClusterIP \
--set global.image.repo=<pathTolmageRepo> \
<deployment_name> \
/path to the helm chart root folder>
```

Value	Description
<namespace>	The namespace that you created and into which you deployed Collibra DQ.
<cdq_version>	The version from the web image suffix. For example, **2023.11** from the image, **dq-web:2023.11**.
<cdq_spark_version>	The Spark version from the Spark image suffix. For example, **3.4.1-2023.11** from the image, **spark:3.4.1-2023.11**.
<cdq_license_key>	The license key provided to you by Collibra.
<your_license_name>	The license name provided to you by Collibra.
${email}	The default admin user email associated with the admin account.

Value	Description
${password}	The default admin user password for the admin account. The password must adhere to the following password policy: • A minimum of 8 characters. • A maximum of 72 characters. • At least one upper-case character. • At least one numeric character. • At least one supported special character (`!@#%$^&*?_~`). • Cannot contain the user ID (admin). **Note** If a password that does not meet the password policy is entered, the install process proceeds as though the password is accepted, but the admin user becomes locked out. If this occurs, rerun the Helm command with a password that meets the password policy and restart the web pod.
<pathToImageRepo>	This is your private registry key, where the Collibra images are available. When this is not provided, you will pull the images from the Collibra image registry, for which you should create a pull secret with the repo key provided by Collibra. See the Cloud native install documentation for more details about how to create a pull secret.
<deployment_name>	Any name of your choice for this deployment.

5. Check that all required Kubernetes resources are available and running.
   a. List all of the resources of the namespace in your Kubernetes cluster into which Collibra DQ is deployed, including pods, Ingress/LoadBalancer, and statefulsets.

```
kubectl get all -n <namespace>
```

6. Create a test DQ job without any optional DQ layers or rules to verify that the Spark driver, executors, and containers are able to launch successfully.

> **Important** If you use file-based SAML authentication with the `SAML_METADATA_USE_URL` variable set to `true` in the owl-web ConfigMap, update the Meta-Data URL option on the SAML Security Settings page with your metadata file. Use the **file:/opt/owl/config/idp-metadata.xml** format, ensuring the file name begins with the prefix **file:**. For additional steps on how to configure this, go to the "Enable the SAML SSO sign in option" section in SAML Authentication.

What's next?

- Create a basic job to test that your upgrade was successful.
- Troubleshoot your upgrade, as needed.

Troubleshooting the upgrade

No such file or directory error

If you receive a `Cannot run program "/opt/owl/bin/owlcheck": error=2, No such file or directory` error, follow these steps to verify that `sparksubmitmode=native`.

```
# 1 Stop the agent:
cd $OWL_BASE/owl/bin
./owlmanage.sh stop=owlagent

# 2 Check that sparksubmitmode=native in the agent.properties
file:
sparksubmitmode=native
sparkhome=</your/spark/home/folder>

# 3 Restart the agent:
./owlmanage.sh start=owlagent

# 5 Verify that agent.properties contains the correct
details.
cd $OWL_BASE/owl/config
cat $OWL_BASE/owl/config/agent.properties
```

Unable to log in after upgrade

If you attempt to log in and receive the following error message:

```
Error during login. Please refresh the page and try again
```

The issue may be related to how Spring Boot handles forwarded headers. To resolve this issue, add the following line to `owl-env.sh`:

```
export SERVER_FORWARD_HEADERS_STRATEGY=FRAMEWORK
```

Java error after upgrade

If you receive the following error message after upgrading:

```
java.lang.ClassNotFoundException:
org.bouncycastle.asn1.ASN1Encodable
```

The issue may be related to a missing dependency. To resolve this issue, follow these steps:

```
cd $OWL_HOME/bin/

mkdir external

cd external
wget https://repo1.maven.org/maven2/org/bouncycastle/bcprov-
jdk18on/1.78.1/bcprov-jdk18on-1.78.1.jar

-----------------------------
# Edit the owlmanage.sh script.
# Enter below line after line 13
bcprov="$(ls $INSTALL_PATH/bin/external/bcprov-*.jar
2>/dev/null || true)"

# Update the start_owlweb() function command with below
(Replace the existing one)
$JAVA_HOME/bin/java -Xms2g -Xmx2g -
Dcdjd.io.netty.tryReflectionSetAccessible=true -
Djava.system.class.loader=com.owl.web.config.DQClassLoader -
Djdk.tls.ephemeralDHKeySize=2048 -DowlAppender=owlRollingFile
-DowlLogFile=owl-web -
Dlog4j.configurationFile=file://$INSTALL_
PATH/config/log4j2.xml $EXTRA_JVM_OPTIONS -cp $owlweb:$bcprov
org.springframework.boot.loader.JarLauncher --
server.session.timeout=$TIMEOUT --
logging.level.org.springframework=$SPRING_LOG_LEVEL --
owlbase.path=$base_path --server.port=$PORT >> $INSTALL_
PATH/log/owl-web.log 2>&1 & echo $! >$INSTALL_PATH/pids/owl-
web.pid

-----------------------------
# Stop and Start the DQ web service
```

This section shows the most common commands to run when troubleshooting a Collibra DQ environment that is deployed on Kubernetes.

```
### Provide documentation on syntax and flags in the terminal
kubectl help
### To see how to use Kubernetes resources
kubectl api-resources -o wide
```

Viewing Kubernetes Resources

```
### Get Pods, their names & details in all Namespaces
kubectl get pods -A -o wide
### Get all Namespaces in a cluster
kubectl get namespaces
### Get Services in all Namespaces
kubectl get services -A -o wide
### List all deployments in all namespaces:
kubectl get deployments -A -o wide
```

Logs & Events

```
### List Events sorted by timestamp in all namespaces
kubectl get events -A --sort-by=.metadata.creationTimestamp
### Get logs from a specific pod:
kubectl logs [my-pod-name]
```

Resource Allocation

```
### If the Kubernetes Metrics Server is installed,
### the top command allows you to see the resource
consumption for nodes or podscode

kubectl top node
kubectl top pod

### If the Kubernetes Metrics Server is NOT installed, use

kubectl describe nodes | grep Allocated -A 10
```

Configuration

```
### Get current-context

kubectl config current-context

### See all configs for the entire cluster

kubectl config view
```

Authorization Issues

```
### Check to see if I can read pod logs for current user &
context

kubectl auth can-i get pods --subresource=log

### Check to see if I can do everything in my current
namespace ("*" means all)

kubectl auth can-i '*' '*'
```

Unable to log in after upgrade

If you attempt to log in and receive the following error message:

```
Error during login. Please refresh the page and try again
```

The issue may be related to how Spring Boot handles forwarded headers. To resolve this issue, add the following helm command:

```
--set global.web.security.forwardHeaders.strategy=FRAMEWORK
```

Connections

Adding Connections

Adding JDBC and remote file connections

This topic shows you how to create a new data source connection.

Prerequisites

To add a connection, you need:

- A Collibra DQ account with ROLE_ADMIN or ROLE_CONNECTION_MANAGER permissions.
- Access to the account credentials of your connection, for example, PostgreSQL.

Adding a JDBC or remote file connection

This section shows you how to connect to a JDBC or remote file data source from the Connection Management page. Select an option from the tabs below.

1. Sign into Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
3. Click **Connections**.
   » The **Connection Management** page opens.

4. Click **Add Connection** on the upper right side of the page.
   » The **Add New Connection** dialog opens to the JDBC Connections tab with a list of supported JDBC data sources.
5. Click a data source.
   » The **Add Connection** template appears.
6. Enter the required information.

Required	Option	Type	Value
Connection Details			
✓ Yes	Name	String	The unique name used for your connection.
✓ Yes	Connection URL	String	The connection string value of your connection. The Connection URL uses the following format: `jdbc:<datasource>://$host:<port>/<AdditionalProperties>` Replace the Connection URL to point to the database server to run DQ Jobs against.
✓ Yes	Port	Integer	The port for the connection to the data source host.

Required	Option	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
Connection Credentials			
✓ Yes	Authentication Type	Option	The method to authenticate your connection. **Note** The configuration requirements differ depending on the authentication type you select. Additional options on the connection template, such as Username and Password, will appear depending on the authentication type you select.
✕ No	Pushdown	Option	Enables Pushdown processing. This option is required for Pushdown Job submission. **Note** This option is currently only available for the connections listed on Pushdown processing.

Required	Option	Type	Value
✕ No	Source Name	String	The alias name of your connection. While this is optional, you can use this to make it easier to identify your connection in API payloads. **Tip** There are no character restrictions for the Source Name field, but it is best practice to use only alphanumeric characters, hyphens, and underscores.
✕ No	Target Agents	String	The Agent used to submit your DQ Job.

Required	Property	Type	Value
✓ Yes	Driver Location	File path	The location where your driver file is saved. Click **Upload** to specify the Directory Name and select a Driver File.
✓ Yes	Driver Name	String	The driver class name. For example, a BigQuery JDBC driver may use the following convention: `com.simba.googlebigquery.jdbc42.Driver`

Required	Property	Type	Value
✕ No	Driver Properties	String	The configurable driver properties of your connection. The supported format is `[{"name":"prop1","value":"val1"},{"name":"prop2","value":"val2"}]`

Required	Property	Type	Value
✓ Yes (when using)	Variable Name	String	The variable to display in place of the sensitive information. This can be any variable, such as `Password` In the Connection URL, replace the sensitive information with this value. In the following example, `dquser` and `dqpassword` are the property names that replace the sensitive variable values: `jdbc:postgresql://localhost:5432/postgres?user=${dquser}&password=${dqpassword}`
✓ Yes (when using)	Value	String	The sensitive value to hide in the Connection URL. For example, if the password value of the connection credential is "password123", enter `password123` in the property value field. **Tip** Commas and equals signs are supported characters. For example, **val1=,val2** is a valid format.

7. Click **Submit**.

1. Sign into Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.

3. Click **Connections**.
   » The **Connection Management** page opens.
4. Click **Add Connection** on the upper right side of the page.
   » The **Add New Connection** dialog opens to the JDBC Connections tab
5. Click the **Remote File Connections** tab.
   » The **Add New Connection** dialog opens to the JDBC Connections tab with a list of supported remote file data sources.
6. Click a data source.
   » The **Add Connection** template appears.
7. Enter the required information.

Required	Connection Detail	Type	Value
✓ Yes	Name	String	The unique name used for your connection.
✓ Yes	Connection URL	String	The connection string value of your connection. **Example** `s3://dq-datasets-123`

Required	Connection Detail	Type	Value
✓ Yes	Authentication Type	Option	The method to authenticate your connection. **Note** The configuration requirements differ depending on the authentication type you select. Additional options on the connection template, such as Keytab, will appear depending on the authentication type you select.
✕ No	Target Agents	String	The Agent used to submit your DQ Job.
✕ No	Archive Breaking Records	Option	Select this option to automatically export CSV files containing the breaking records of DQ Job to your remote file data source.

Required	Connection Detail	Type	Value
✕ No	Archive Location	String	The path along which break records will be archived when the Archive Break Records and HTTPS options are enabled. Specify a schema output location in your data source for break records to send. For example, **/write/FolderName** **Note** This option is only available when the HTTPS option is enabled.

Required	Property	Type	Value
✕ No	Driver Properties	String	The configurable driver properties of your connection. The supported format is `[{"name":"prop1","value":"val1"},{"name":"prop2","value":"val2"}]`

Required	Property	Type	Value
✓ Yes (when using)	Variable Name	String	The variable to display in place of the sensitive information. This can be any variable, such as `Password` In the Connection URL, replace the sensitive information with this value. In the following example, `dquser` and `dqpassword` are the property names that replace the sensitive variable values: `jdbc:postgresql://localhost:5432/postgres?user=${dquser}&password=${dqpassword}`
✓ Yes (when using)	Value	String	The sensitive value to hide in the Connection URL. For example, if the password value of the connection credential is "password123", enter `password123` in the property value field. **Tip** Commas and equals signs are supported characters. For example, **val1=,val2** is a valid format.

8. Click **Submit**.

What's next?

Map data source connections to a DQ agent

Known Limitations

- When you update the name of a connection, a separate connection with the updated name is created in addition to the old connection. Because Collibra DQ treats this as a new connection, the Jobs and details from the old connection do not persist to the new one.
- When you edit an existing connection and its details, another connection is created and any sensitive properties from the old connection do not persist to the new connection.

- A possible workaround is to recreate the sensitive properties from the old connection in the new connection.

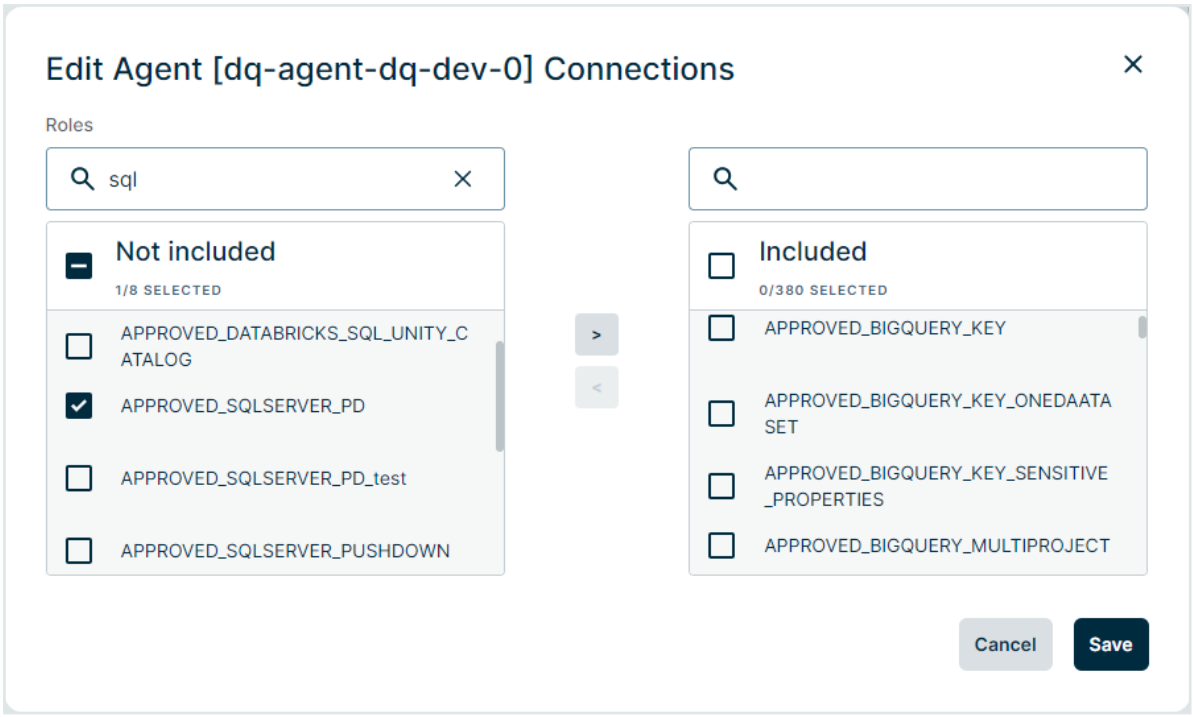
Map data source connections to a DQ agent

When you add new data source connections, the DQ agent requires permission to run DQ Jobs against data within them. This topic shows you how to map connections to an agent, to grant permission to the connection.

Steps

1. On the Collibra DQ home page, hover your pointer over **Settings** and select **Admin Console**.
   » The Admin Console opens.
2. Click **Agent Configuration**.
   » The Agent Configuration page opens.
3. In the last column of the Agents table, to the right, click the chain link icon to link your agent to data source connections.
   » The **Agent to Connection Management** wizard appears.

> **Note** The left panel contains a list of available connections that are not yet linked to the DQ agent and do not yet have permission to run DQ Jobs. The right panel contains a list of connections that are linked to the DQ agent and have permission to run DQ Jobs.



4. Click a connection in the left panel to link connections one at a time or click the double arrow icon to link all available connections at the same time.
5. Click **Update**.

> **Tip** You can unlink connections with the same methods listed above, but click the connections listed in the right panel instead of the left. Successfully unlinked connections appear in the left panel.

Supported Connections

This section shows the supported data source connections in Data Quality & Observability Classic. A supported data source is a data source that is shipped with the images or standalone bundles, and thus, eligible for support from the Collibra DQ team.

Supported JDBC data sources

This section includes the connection details of JDBC data sources supported by Collibra DQ.

> **Important** Data Quality & Observability Classic is designed to adhere to the best practices for naming tables, schemas, and columns in relation to the data sources we support. When setting up a data source connection, we highly recommend consulting the best practices of that specific data source to avoid issues with non-standard naming conventions.

Data Source	Packaged	Certified	Supports Push-down
Amazon Athena	✓ Yes	✓ Yes	✓ Yes
Amazon Redshift	✓ Yes	✓ Yes	✓ Yes
Apache Impala	✓ Yes	✓ Yes	✗ No
CockroachDB	✓ Yes	✓ Yes	✗ No
Databricks	✓ Yes	✓ Yes	✓ Yes
Db2	✓ Yes	✓ Yes	✗ No
Dremio	✗ No	✗ No	✗ No
Google BigQuery	✓ Yes	✓ Yes	✓ Yes
Hive	✗ No	✓ Yes	✗ No

Data Source	Packaged	Certified	Supports Push-down
MongoDB	✓ Yes	✓ Yes	✗ No
MySQL	✓ Yes	✓ Yes	✗ No
Oracle	✓ Yes	✓ Yes	✗ No
PostgreSQL	✓ Yes	✓ Yes	✗ No
SAP HANA	✓ Yes	✓ Yes	✓ Yes
Snowflake	✓ Yes	✓ Yes	✓ Yes
SQL Server	✓ Yes	✓ Yes	✓ Yes
Sybase	✓ Yes	✓ Yes	✗ No
Teradata	✓ Yes	✓ Yes	✗ No
Trino	✗ No	✓ Yes	✓ Yes

Limitations

Source queries that contain `WITH UR` are only supported by Db2 connections.

Connecting to Amazon Athena

This section contains details for Amazon Athena connections.

Select an option from the dropdown menu to display information for a particular driver class.



General information

Field	Description
Data source	Amazon Athena
Supported versions	2.0.35.1000 21.0.8214.0
Connection string	`jdbc:awsathena://jdbc:cdata:amazonathena`
Packaged?	✕ No ✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✓ Yes
Yarn agent	✓ Yes ✕ No
Parallel JDBC	✕ No
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Athena data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Glue catalog and S3 buckets.
- Write access on your S3 output location.
- ROLE_ADMIN assigned to your user in Collibra DQ.

```
{
    "Version": "YYYY-MM-DD",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "athena:StartQueryExecution",
                "s3:ListBucketMultipartUploads",
                "athena:GetQueryResultsStream",
                "glue:GetTables",
                "glue:GetPartitions",
                "athena:GetQueryResults",
                "glue:BatchGetPartition",
                "s3:ListBucket",
                "glue:GetDatabases",
                "athena:ListQueryExecutions",
                "s3:ListMultipartUploadParts",
                "glue:GetTable",
                "glue:GetDatabase",
                "athena:GetWorkGroup",
                "s3:PutObject",
                "s3:GetObject",
                "glue:GetPartition",
                "glue:GetCatalogImportStatus",
                "athena:StopQueryExecution",
                "athena:GetQueryExecution",
                "s3:GetBucketLocation",
                "athena:BatchGetQueryExecution",
                "athena:DeletePreparedStatement",
                "athena:CreatePreparedStatement"
            ],
            "Resource": [
                "arn:aws:athena:*:<AWSAccountID>:workgroup/prima-
ry",
                "arn:aws:s3:::<S3 bucket name>/*",
                "arn:aws:s3:::<S3 bucket name>",
                "arn:aws:glue:*:<AWSAccountID>:catalog",
                "arn:aws:glue:*:<AWSAccountID>:database/<databas-
e name>",
                "arn:aws:glue:*:<AWSAccountID>:table/<database
name>/*"
            ]
        }
    ]
}
```

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string path of your Athena connection. Use the appropriate JDBC connection URLs in your business tool configuration according to your private DNS configuration for your endpoint. • Use the following connection string if you turned off the private DNS: jdbc:awsathena://vpce-.athena.us-east-1.vpce.amazonaws.com:443 • Use the following connection string if you turned on the private DNS: jdbc:awsathena://athena.us-east-1.amazonaws.com:443 **Note** Athena's streaming API uses port 444 to stream the query results. When you use a JDBC/ODBC driver, Athena uses this port to stream the query results to the JDBC/ODBC driver installed on the client host. Therefore, unblock this port when you use a JDBC/ODBC driver to connect to Athena. If this port is blocked, your business intelligence tool might time out or fail to show query results when you run a query. Ensure that port 444 isn't blocked. If you use an AWS PrivateLink endpoint to connect to Athena, ensure that the security group attached to the AWS PrivateLink endpoint is open to inbound traffic on port 444. Athena uses port 444 to stream query results. If port 444 is blocked, then the results aren't streamed back to your client host. In such situations, you might receive an error message similar to "[Simba][AthenaJDBC](100123) An error has occurred. Exception during column initialization". This can also cause the business intelligence tool to stop responding and not display the query results. Also ensure that the security group attached to your VPC endpoint allows traffic from the host where you installed the JDBC/ODBC driver. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.

Required	Connection Property	Type	Value

			Example
			`jdbc:awsathena://AwsRegion=${region};S3OutputLocation=s3://${output_location};MetadataRetrievalMethod=Query`
			`jdbc:cdata:amazonathena:AwsRegion=northernvirginia;S3StagingDirectory=s3://<bucket name>;database=default;`

			Important `database=default;` must be included in the connection URL, as shown in the example above.

✓ Yes	Driver Name	String	The driver class name of your Athena connection.
			`com.simba.athena.jdbc.Driver`
			`cdata.jdbc.amazonathena.AmazonAthenaDriver`

✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `0`
✗ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page.
			Note When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✗ No	Source Name	String	N/A

Required	Connection Property	Type	Value
✗ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✗ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Your host can connect to Athena with either an Athena public service endpoint or an Athena private endpoint. For more information on setting the endpoint, see Command line options and Boto3 documentation.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.

Setting up a single role-based Athena connection using AWS EC2 roles for Instance Profile authentication

Important
This authentication option is only available for Athena CDATA driver versions 24.0.8994

and newer.

1. Set the Connection URL to reference the required parameters.

```
jdbc:cdata:amazonathena:AuthScheme=AwsEC2Roles;MetadataDiscoveryMethod=Athena;AWSRoleARN=arn:aws:iam::${AccountID}:role/${RoleName};Database=${ConnectionName};DataSource=${DBName};AWSRegion=${Region};S3StagingDirectory=${S3DirectoryPath};Workgroup=${WorkgroupName}
```

Tip
You can optionally include the log parameters by adding `Logfile=/tmp/example-log.log;Verbosity=5;` to the Connection URL to see any errors that occur with the CDATA driver.

2. Click **Submit**.

Setting up multiple role-based Athena connections using AWS EC2 roles for Instance Profile authentication

Important
This authentication option is only available for Athena CDATA driver versions 24.0.8994 and newer.

1. Create a cache location folder in the **/tmp** directory where Collibra DQ is running for each role-based Athena connection you wish to use to allow CDATA to add the Instance Profile credential file to them. Multiple role-based Athena connections cannot share the same folder in the /tmp directory.

Example
If you have 2 different role-based Athena connections, the folder for the first connection could be called **/tmp/connA/** and the second could be called **/tmp/connB/**.

2. Set the Connection URL to reference the required parameters.

```
jdbc:cdata:amazonathena:AuthScheme=AwsEC2Roles;MetadataDiscoveryMethod=Athena;AWSRoleARN=arn:aws:iam::${AccountID}:role/${RoleName};Database=${ConnectionName};DataSource=${DBName};AWSRegion=${Region};S3StagingDirectory=${S3DirectoryPath};Workgroup=${WorkgroupName};Location=${/tmp/connection/};CredentialsLocation=${/tmp/connection/CredentialsFile.txt}
```

> **Tip**
> You can optionally include the log parameters by adding `Logfile=/tmp/example-log.log;Verbosity=5;` to the Connection URL to see any errors that occur with the CDATA driver.

3. Click **Submit**.

Required	Field	Description
✓ Yes	Username	The username of your Athena account.
✓ Yes	Password	The password of your Athena account.
✓ Yes	Script	The file path containing the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/athena_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameter to authenticate your Athena connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your Athena connection.

Required	Field	Description
✕ No	Param $3	Optional. An additional parameter to authenticate your Athena connection.

Connecting to Amazon Redshift

This section contains details for Amazon Redshift connections.

General information

Field	Description
Data source	Amazon Redshift
Supported version	2.1.0.9
Connection string	`jdbc:redshift://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✓ Yes
Yarn agent	✓ Yes

Field	Description
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Redshift data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Redshift database tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your Redshift connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:redshift://${host}:5439/${database}`
✓ Yes	Driver Name	String	The driver class name of your Redshift connection. `com.amazon.redshift.jdbc42.Driver`

Required	Connection Property	Type	Value
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `5439`
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. > **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. > **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.

Required	Field	Description
✓ Yes	Username	The username of your Redshift account.
✓ Yes	Password	The password of your Redshift account.
✓ Yes	Script	The file path containing the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/redshift_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameter to authenticate your Redshift connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your Redshift connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your Redshift connection.

Locating the JDBC URL

Before you can connect to your Redshift cluster from a SQL client tool, you need to know the JDBC URL of your cluster. The JDBC URL has the following format: `jdbc:redshift://endpoint:port/database`.Note

A JDBC URL specified with the former format of `jdbc:postgresql://endpoint:port/database` still works.

The fields of the format shown preceding have the following values.

Field	Value
`jdbc`	The protocol for the connection.
`redshift`	The sub-protocol that specifies to use the Redshift driver to connect to the database.
`endpoint`	The endpoint of the Redshift cluster.
`port`	The port number that you specified when you launched the cluster. If you have a fire-wall, make sure that this port is open for you to use.
`database`	The database that you created for your cluster.

The following is an example JDBC URL:
`jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev`

For information about how to get your JDBC connection, see Finding your cluster connection string.

If the client computer fails to connect to the database, you can troubleshoot possible issues. For more information, see Troubleshooting connection issues in Amazon Redshift.\

https://docs.aws.amazon.com/redshift/latest/mgmt/configure-jdbc-connection.html

Limitations

When using Pushdown for Redshift, the following column data types are not supported:

- BINARY VARYING
- GEOGRAPHY
- GEOMETRY
- HLLSKETCH
- SQL_VARCHAR
- SUPER
- VARBINARY
- VARBYTE

Connecting to Apache Impala

This section contains details for Impala connections.

Select an option from the dropdown menu to display information for a particular driver class.

General information

Field	Description
Data source	Apache Impala
Supported versions	2.6.27.103221.0.8137.0
Connection string	`jdbc:impala://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes ✕ No
Analyze data	✓ Yes ✕ No
Schedule	✓ Yes ✕ No
Processing capabilities	
Pushdown	✕ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	

Field	Description
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Impala data into Data Quality & Observability Classic, you need the following permissions.

- The Kerberos user has read permissions on Impala tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✕ No	Hive Direct Eligible	Option	Uses the Hive server engine for distributed speed and scale.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string path of your Impala connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:impala://${host}:21050/default;AuthMech=1;KrbHostFQDN=${host};KrbServiceName=impala;SSL=1;AllowSelfSignedCerts=1`

✓ Yes	Driver Name	String	The driver class name of your connection.
			`com.cloudera.impala.jdbc.Driver`
			`cdata.jdbc.apacheimpala.ApacheImpalaDriver`

✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `21050`
✗ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✗ No	Source Name	String	N/A

Required	Connection Property	Type	Value
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true** Optionally add the following driver property to remove the enforcement of unique column names: `hive.resultset.use.unique.column.names=false`

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Configuring Access Token Manager authentication

Prerequisites

You have added a script file that contains or can retrieve an access token to a folder accessible to Data Quality & Observability Classic.

Steps

1. Update the Connection URL to the following format: `jdbc:impala://${host}:${port}/${catalog}?SSL=true&source=jdbc:impala&accessToken=${accessToken}`
2. Select **Access Token Manager** from the Authentication Type dropdown menu.
3. Enter the User ID of the IdP account in the Username input field.
4. Enter the file path containing the access token script file in the Script input field. For example, /opt/owl/config/get_impala.sh
5. Click Submit.

Required	Field	Description
✓ Yes	Principal	The Kerberos entity to authenticate and grant access to your connection.
✓ Yes	Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. **Example** /tmp/keytab/impala_user.keytab
✓ Yes	Password	The secret credential associated with your Kerberos principal.
✕ No	Script	The file path that contains the script file used to interact with and authenticate a Kerberos user. **Example** /tmp/keytab/impala_pwd_mgr.sh

Required	Field	Description
✕ No	Param $1	Optional. Additional Kerberos parameter.
✕ No	Param $2	Optional. Additional Kerberos parameter.
✕ No	Param $3	Optional. Additional Kerberos parameter.
✓ Yes	TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.
✓ Yes	Username	The User ID of the IdP account.
✓ Yes	Script	The file path containing the access token script file in the Script input field. **Example** /opt/owl/config/get_impala.sh
✕ No	Param $1	Optional. Additional parameter.
✕ No	Param $2	Optional. Additional parameter.
✕ No	Param $3	Optional. Additional parameter.

Connecting to Cassandra

This section contains details for Cassandra connections.

General information

Field	Description
Data source	Cassandra
Supported versions	23.0.8839.0 (CDATA)
Connection string	`jdbc:cassandra:`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Cassandra data into Data Quality & Observability Classic, you need the following permissions.

- Read access on Cassandra tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your Cassandra connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:cassandra:Port=${port};Server=${host};`
✓ Yes	Driver Name	String	The driver class name of your Cassandra connection. `cdata.jdbc.cassandra.CassandraDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `9042`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Cassandra account.
✓ Yes	Password	The password of your Cassandra account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/cassandra_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameters to authenticate your Cassandra connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your Cassandra connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your Cassandra connection.

Connecting to CockroachDB

This section contains details for CockroachDB connections.

General information

Field	Description
Data source	CockroachDB
Supported versions	42.5.1
Connection string	`postgresql://`

Field	Description
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your CockroachDB data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your CockroachDB tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your CockroachDB connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `postgresql://${user}:${password}@free-tier.gcp-us-central1.cockroachlabs.cloud:26257/defaultdb?sslmode=verify-full&sslrootcert=$HOME/.postgresql/root.crt&options=--cluster%3Dnoisy-koala-3822`
✓ Yes	Driver Name	String	The driver class name of your connection. `org.postgresql.Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `26257`
✗ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.

Required	Connection Property	Type	Value
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your CockroachDB account.
✓ Yes	Principal	The Kerberos entity to authenticate and grant access to your connection.
✓ Yes	Password	The password of your CockroachDB account.

Required	Field	Description
✓ Yes	Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. **Example** /tmp/keytab/impala_user.keytab
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/cockroachdb_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameters to authenticate your CockroachDB connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your CockroachDB connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your CockroachDB connection.
✓ Yes	TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

Connecting to Databricks

This section contains details for Databricks connections.

Select an option from the dropdown menu to display information for a particular driver class.

General information

Field	Description
Data source	Databricks
Supported versions	2.6.3621.0.8137.0
Connection string	`jdbc:databricks://jdb-c:cdata:Databricks:`
Packaged?	✕ No ✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✕ No
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✕ No ✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Databricks data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Unity Catalog.
- Access to the cluster endpoint that you use.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Use the dropdown menu to select a driver class.



Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

Required	Connection Property	Type	Value

✓ Yes	Connection URL	String	The connection string path of your Databricks connection.
			
			Example
			`jdbc:databricks://${host}:${port}/default;transportMode=http;ssl=1;HttpPath=${clusterHttpPath};AuthMech=3;UID=${username};PWD=${token}`
			`jdbc:cdata:Databricks:Server=https://${server}.cloud.databricks.com;httpPath=${path};User=token;Token=${token};`
			
			Use the following format when connecting to **Databricks SQL Warehouse**:
			
			jdbc:databricks://[Host]:[Port]/[Schema];[Property1]=[Value]; [Property2]=[Value];...
			
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			
			Example
			`jdbc:databricks://<example-account>.cloud.databricks.com:443/default;transportMode=http;ssl=1;AuthMech=3;httpPath=/sql/1.0/warehouses/xxx;UID=token;PWD=<your-token-here>`
			
			Use the following format when connecting to **Databricks Unity Catalog**:
			
			jdbc:databricks://[Host]:[Port]/[Schema];[Property1]=[Value]; [Property2]=[Value];...ConnCatalog=catalog_name
			
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.

Required	Connection Property	Type	Value

> **Example**
> `jdbc:databricks://${host}:${port}/${Schema};transportMode=http;ssl=1;AuthMech=3;HttpPath=${clusterHttpPath};UID=token;PWD=${access_token};ConnCatalog=${catalog}`

Specify `${access_token}` on the Properties tab where **Property Name** is set to `access_token` and **Value** is the password. This is so Collibra DQ treats the password as a sensitive property and does not display its actual value in the connection URL.

When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.

> **Note**
> If you do not specify ConnCatalog=${catalog} in the JDBC connection URL, a default catalog is not set. Instead, you must include the catalog name in the query, as shown in the following example.
> `select * from catalog.schema.table`
>
> When you specify a catalog in the connection URL, you do not need to explicitly reference it when you query schema and data in the catalog, as shown in the following example.
> `select * from schema.table`

Required	Connection Property	Type	Value

✓ Yes	Driver Name	String	The driver class name of your Databricks connection.
			`com.databricks.client.jdbc.Driver`
			`cdata.jdbc.databricks.DatabricksDriver`

Required	Connection Property	Type	Value
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `0`
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. > **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. > **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.

Required	Connection Property	Type	Value
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**
✕ No	Variables	String	When you specify `${access_token}` in the Connection URL, set the following properties on the **Variables** tab. **Variable Name** is set to `access_token`. **Value** is the password.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Databricks service account. Set Username to `token`.

Required	Field	Description
✓ Yes	Password	The password of your Databricks service account. Enter the token value that you entered in the Connection URL. **Note** To successfully establish a Databricks connection, verify that your token is valid, because they generally expire in 90 days.
✓ Yes	Script	The file path containing the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/databricks_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameter to authenticate your Databricks connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your Databricks connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your Databricks connection.
✓ Yes	Tenant ID	The tenant ID of your Microsoft account.
✓ Yes	Client ID	The client ID of your Microsoft account.
✓ Yes	Client Secret	The client secret of your Microsoft account.

JDBC Driver Jar

if necessary, you can download the Databricks JDBC zip file from one of the following links.

- For the latest Databricks driver version, refer to the official Databricks JDBC Driver page.
- For archived Databricks driver versions, refer to the official Databricks JDBC Driver archive.

> **Note** Databricks JDBC driver version 2.6.27 is packaged as part of both standalone and Kubernetes download packages.

Notebook (Supported)

- Pyspark SDK
- Scala SDK
- Collibra DQ + Databricks

Databricks no longer supports Runtime 6.5 or 10.3. Therefore, Collibra DQ Profile 2.45 is not runnable on Databricks.

https://docs.databricks.com/release-notes/runtime/10.3ml.html

The following table shows the latest supported versions of Collibra DQ Profiles and their matching Databricks Runtimes.

CDQ-DataBricks NoteBook status			
Databricks Cluster	7.3	9.1LTS	10.4
CDQ Profile	SPARK301	SPARK312	SPARK 320
Supported	YES	YES	YES

Spark Submit (Not Supported)

- Spark Master URL
- Databricks Jobs API
  - Rest
  - UI
  - DQ-Databricks Submit

> **Note** While these are not officially supported, there is a reference to architecture and implementation pattern for how to do a Databricks Job submission.

Limitations

- When Archive Break Records is enabled for Azure Databricks Pushdown connections authenticated over EntraID, the data preview does not display column names correctly and shows 0 columns in the metadata bar. Therefore, Archive Break Records is not supported for Azure Databricks Pushdown connections that use EntraID authentication.
- UTF-8 character encoding for Databricks JDBC connections functions as expected in standalone deployments of Data Quality & Observability Classic versions older than 2024.06. As a workaround in versions older than 2024.06, follow the instructions in the Collibra Support Portal.

Connecting to Db2

This section contains details for IBM Db2 connections.

General information

Field	Description
Data source	IBM Db2
Supported versions	4.27.25
Connection string	`jdbc:db2://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes

Field	Description
Processing capabilities	
Pushdown	✕ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Db2 data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Db2 tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string path of your Db2 connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:db2://${host}:50000/owldb2`
✓ Yes	Driver Name	String	The driver class name of your Db2 connection. `com.ibm.db2.jcc.DB2Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `50000`
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.

Required	Connection Property	Type	Value
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Db2 account.
✓ Yes	Password	The password of your Db2 account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/db2_pwd_mgr.sh

Required	Field	Description
✕ No	Param $1	Optional. An additional parameters to authenticate your Db2 connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your Db2 connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your Db2 connection.

Limitations

- Because the DATE_TRUNC does not exist in Db2 for z/OS operating systems, the date visualization graph under the Time Slice component on the Select rows step in Explorer will not load.
- Source queries that contain `WITH UR` are only supported by Db2 connections.

Connecting to Denodo

This section contains details for Denodo connections.

General information

Field	Description
Data source	Denodo
Supported versions	8.0-update-20240926
Connection string	`jdbc:vdb://`
Packaged?	✓ Yes
Certified?	✓ Yes

Field	Description
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Denodo data into Data Quality & Observability Classic, you need the following permissions.

- Read access on Denodo tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your Denodo connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:vdb://${host}:${port}database?ssl=true&sslTrustStoreLocation=${keystore_location&sslTrustStorePassword=${password}`
✓ Yes	Driver Name	String	The driver class name of your Denodo connection. `com.denodo.vdp.jdbc.Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `9999`
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.

Required	Connection Property	Type	Value
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true** When you use the OAuth2 authentication method, add the following semicolon-separated **;** properties, adjusting the parameterized section (`${ }`) according to your IdP details: • `dq.idp.url=${ }` • `dq.idp.scopes=${ }` • `dq.idp.grantType=${ }` **Example** `dq.idp.url=https://dev-12345678.okta.com/oauth2/abcde12fgh3ijKLmN4o5/v1/token;dq.idp.scopes=developers;dq.idp.grantType=client_credentials`

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Important
When you use OAuth2 authentication, you need to add the following IdP parameters to the Driver Properties input field on the Properties tab:

```
dq.idp.url=${ }
dq.idp.scopes=${ }
dq.idp.grantType=${ }
```

See the Driver Properties section above for more information.

Required	Field	Description
✓ Yes	Username	The username of your Denodo account.
✓ Yes	Password	The password of your Denodo account.
✓ Yes	Username	The username credential required for the IdP service to authenticate a user.
✓ Yes	Password	The password credential required for the IdP service to authenticate a user.

Required	Field	Description
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/denodo_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameters to authenticate your Denodo connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your Denodo connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your Denodo connection.

Known limitations

- Denodo connections do not support queries with a LIMIT filter, because the Denodo's proprietary query language (VQL) does not support the LIMIT operator.

Connecting to Dremio

This section contains details for Dremio connections.

General information

Field	Description
Data source	Dremio
Supported versions	24.3.0

Field	Description
Connection string	`jdbc:dremio:direct=`
Packaged?	✕ No
Certified?	✕ No
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✕ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

> **Note** The Dremio connection is compatible with JDK11 if you add the following to owlmanage.sh as a JVM option for the web and Spark instance:
> `-Dcdjd.io.netty.tryReflectionSetAccessible=true`

Minimum user permissions

In order to bring your Dremio data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Dremio tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your Dremio connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:dremio:direct=${host};schema=Samples`
✓ Yes	Driver Name	String	The driver class name of your Dremio connection. `com.dremio.jdbc.Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `31010`
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.

Required	Connection Property	Type	Value
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

This data source supports the following authentication type:

Username/Password

Required	Field	Description
✓ Yes	Username	The username of your Dremio account.
✓ Yes	Password	The password of your Dremio account.

Known limitations

- When you run a DQ job with a SQL query that attempts to perform a WHERE on a date column, the job fails. For example, `select * from "@admin"."myspace" WHERE registration_dttm = '2016-02-03T07:55:29'`. There is currently no known workaround for this limitation.

- When you run a DQ job with a SQL query that does not contain an explicit WHERE clause, the job fails with an exception message that the command line does not contain a `-rd`. A workaround for this limitation is to add WHERE 1=1 to the query.
- When you run a DQ job with a query that contains both a WHERE and LIMIT clause, the SQL preview returns all records and ignores both filters.

Connecting to Google BigQuery

This section contains details for Google BigQuery connections.

Select an option from the dropdown menu to display information for a particular driver class.



General information

Field	Description
Data source	Google BigQuery
Supported versions	1.5.2.100521.0.8160.0
Connection string	`jdbc:bigquery://jdbc:googlebigquery:`
Packaged?	✕ No ✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes ✕ No
Processing capabilities	
Pushdown	✓ Yes

Field	Description
Spark agent	✓ Yes
Yarn agent	✓ Yes ✕ No
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

To bring your BigQuery data into Data Quality & Observability Classic, you need the following permissions:

- Read access on your BigQuery dataset tables.
- BigQuery job editor access.
- Write access on the materialized view datasets.
- `ROLE_ADMIN` assigned to your user in Collibra DQ.

Networking and firewalls

Account for the following URLs from networking and firewall perspectives:

- logging.googleapis.com
- oauth2.googleapis.com
- googleapis.com
- bigquerystorage.googleapis.com
- bigquery.googleapis.com

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

Required	Connection Property	Type	Value

✓ Yes	Connection URL	String	The connection string path of your BigQuery connection.
			
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			
			Example
			`jdbc:bigquery://https://www.googleapis.com/bigquery/v2:443;OAuthType=0;ProjectId=${projectId};OAuthServiceAcctEmail=${service-account};OAuthPvtKeyPath=${/path/to/key};TimeOut=3600`
			`jdbc:googlebigquery:AuthScheme="OAuthJWT";OAuthSettingsLocation=memory://;ProjectId=${projectId};InitiateOAuth="GETANDREFRESH";OAuthJWTCert="/tmp/keytab/bigquery.json";OAuthJWTCertType="GOOGLEJSON";OAuthJWTIssuer="iam.service.account";`

Required	Connection Property	Type	Value

✓ Yes	Driver Name	String	The driver class name of your BigQuery connection.
			`com.simba.googlebigquery.jdbc42.Driver`
			`cdata.jdbc.googlebigquery.GoogleBigQueryDriver`

✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `443`

Required	Connection Property	Type	Value
No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
No	Source Name	String	N/A

Required	Connection Property	Type	Value
No	Target Agent	Option	The Agent that submits your Spark job for processing.
Yes	Auth Type	Option	The method to authenticate your connection.

> **Note** The configuration requirements differ depending on the Auth Type you select. See Authentication for more details on available authentication types.

Required	Connection Property	Type	Value
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.

Required	Field	Description
✓ Yes	Username	The username of your BigQuery service account. Set Username to any value.

Required	Field	Description
✓ Yes	Password	The password of your BigQuery service account. Enter the base64 encoded string of your OAuth key. base64 encoded commands To encode your credential JSON file as a base64 encoded string, use one of the following commands: `base64 your_json.json -w 0` or `cat your_json.json | base64 -w 0` **Tip** Your JSON file is typically available to download in GCP when you create a service account in your IAM profile.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/bigquery_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameters to authenticate your BigQuery connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your BigQuery connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your BigQuery connection.

Required	Field	Description

✓ Yes	JSON Credential File	The JSON file containing service account access credentials from your local file storage. Click anywhere in the Service Account Credential field, then upload the JSON credential file containing your Google IAM service account key.
		
		Plain text JSON example
		The JSON file contains the following service account information:
		
		`{`
		`  "type": "service_account",`
		`  "project_id": "owl-hadoop-cdh",`
		`  "private_key_id": "<your-private-key-id-string>",`
		`  "private_key": "-----BEGIN PRIVATE KEY-----\<encoded-string>\n-----END PRIVATE KEY-----\n",`
		`  "client_email": "<your-service-account>@developer.gserviceaccount.com",`
		`  "client_id": "your-service-account-id-string",`
		`  "auth_uri": "https://accounts.google.com/o/oauth2/auth",`
		`  "token_uri": "https://oauth2.googleapis.com/token",`
		`  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",`
		`  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/<your-service-account>@developer.gserviceaccount.com"`
		`}`
		
		Tip Your JSON file is typically available to download in GCP when you create a service account in your IAM profile.
		
		Note The cdata.jdbc.googlebigquery.GoogleBigQueryDriver driver does not support JSON Credential File authentication.

Require-d	Field	Description
✓ Yes	Workload Identity	Refer to Workload Identity for information on how to set it up. Because the Connection URL contains the Project ID associated with the GCP cluster that was set up with the access grant, you only need to select Workload Identity from the Authentication Type dropdown menu, specify an agent, and fill out the other required fields in the connection template once you configure Workload Identity.

Views

When creating views from different datasets (collections), you can add the `viewsEnabled=true` parameter to the connection property when defining the connection.

> **Note**
> For read/write access to BigQuery, you can use the Spark BigQuery connector. To use this connector, ensure that you have the following configurations:
>
> - Set `viewsEnabled=true`.
> - Set `materializationDataset` to a dataset where the GCP user has table creation permission.
> - Optionally set `materializationProject`.

> **Important**
> The connector is implemented by running the query on BigQuery and the results are saved into a temporary table. Spark will then read the results, which could result in additional costs on your BigQuery account.

Miscellaneous

Connection property	Description
`AllowLargeResults`	When set to `true`, this option allows results larger than the maximum response size when reading large tables.

Connection property	Description
`QueryPassthrough`	When set to `true`, this option passes the query to the Google BigQuery server as is, which can help overcome some SQL syntax errors for valid SQL queries.

Command line example for basic spark-submit job

```
-lib "/opt/owl/drivers/bigquery/bigquery/core/" \
-h <IP_ADDRESS>:5432/postgres \
-master spark://<SPARK_MASTER>:7077 \
-ds samples.loan_customer \
-deploymode client \
-q "select * from samples.loan_customer" \
-rd "2021-08-02" \
-driver "com.simba.googlebigquery.jdbc42.Driver" \
-cxn BigQuery
```

Troubleshooting a connection

1. Password for the BigQuery Connector form in Collibra DQ must be a base64 encoded string created from the JSON file and input as password. For example: `base64 your_json.json -w 0` or `cat your_json.json | base64 -w 0`
2. Check that this JAR exists and is on the path of the Collibra DQ Web UI server. For example, <INSTALL_PATH>/owl/drivers/bigquery/core). Look at your driver directory location which contains this BigQuery JAR: spark-bigquery_2.12-0.35.1.jar
3. Ensure these JARs present in <INSTALL_PATH>/owl/drivers/bigquery/:****animal-sniffer-annotations-1.23.jar,annotations-4.1.1.4.jar,api-common-2.20.0.jar,auto-value-1.10.4.jar,auto-value-annotations-1.10.4.jar,avro-1.11.3.-jar,cdata.jdbc.googlebigquery.jar,commons-codec-1.16.0.jar,commons-compress-1.26.0.-jar,commons-logging-1.2.jar,conscrypt-openjdk-uber-2.5.2.jar,core,error_prone_annotations-2.22.0.jar,failureaccess-1.0.1.jar,gax-2.37.0.jar,gax-grpc-2.37.0.jar,google-api-client-2.2.0.jar,google-api-services-bigquery-v2-rev20231008-2.0.0.jar,google-auth-library-credentials-1.20.0.jar,google-auth-library-oauth2-http-1.20.0.jar,google-cloud-bigquerystorage-2.46.0.jar,google-cloud-core-2.27.0.jar,google-http-client-1.43.3.-jar,google-http-client-apache-v2-1.42.3.jar,google-http-client-gson-1.43.3.jar,google-oauth-client-1.34.1.jar,GoogleBigQueryJDBC42.jar,grpc-alts-1.59.0.jar,grpc-api-

1.59.0.jar,grpc-auth-1.59.0.jar,grpc-context-1.59.0.jar,grpc-core-1.59.0.jar,grpc-google-cloud-bigquerystorage-v1-2.46.0.jar,grpc-google-cloud-bigquerystorage-v1beta1-0.170.0.jar,grpc-google-cloud-bigquerystorage-v1beta2-0.170.0.jar,grpc-googleapis-1.59.0.jar,grpc-grpclb-1.59.0.jar,grpc-inprocess-1.59.0.jar,grpc-netty-shaded-1.59.0.-jar,grpc-protobuf-1.59.0.jar,grpc-protobuf-lite-1.59.0.jar,grpc-stub-1.59.0.jar,grpc-util-1.59.0.jar,gson-2.10.1.jar,guava-32.1.2-jre.jar,httpclient-4.5.14.jar,httpcore-4.4.16.jar,j2objc-annotations-2.8.jar,jackson-annotations-2.14.2.jar,jackson-core-2.14.2.jar,jackson-data-bind-2.14.2.jar,javax.annotation-api-1.3.2.jar,joda-time-2.10.10.jar,json-20231013.jar,jsr305-3.0.2.jar,listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar,opencensus-api-0.31.1.jar,opencensus-contrib-http-util-0.31.1.jar,perfmark-api-0.26.0.jar,proto-google-cloud-bigquerystorage-v1-2.46.0.jar,proto-google-cloud-bigquerystorage-v1beta1-0.170.0.jar,proto-google-cloud-bigquerystorage-v1beta2-0.170.0.jar,proto-google-common-protos-2.28.0.jar,proto-google-iam-v1-1.23.0.-jar,protobuf-java-3.24.4.jar,protobuf-java-util-3.24.4.jar,slf4j-api-1.7.36.jar,threetenbp-1.6.8.jar

4. You may get a CLASSPATH conflict regarding the JAR files.
5. Ensure the BigQuery connector Scala version matches your Spark Scala version.

Limitations

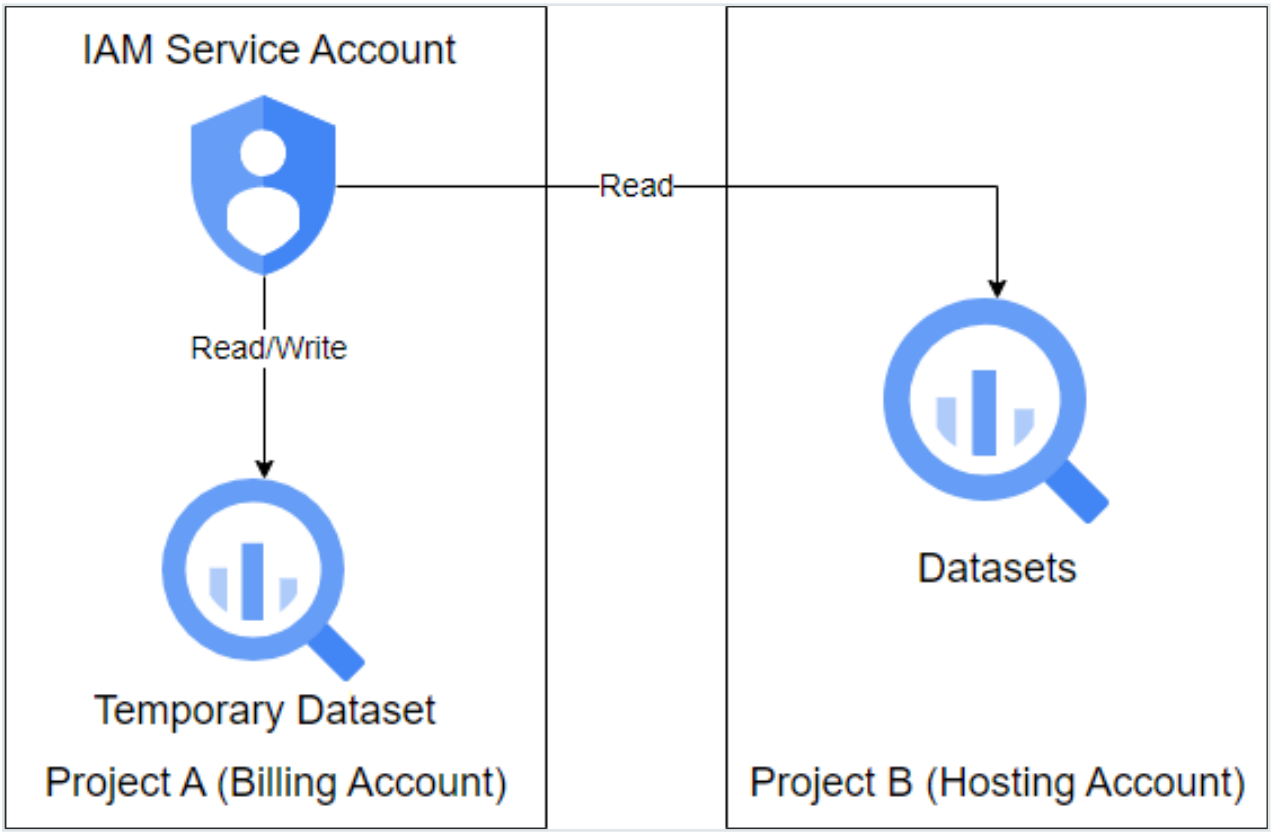
- A limitation exists where Spark is unable to process columns of DateTime data types from BigQuery, which causes it to incorrectly parse DateTime columns as string columns.
    - A possible workaround is to cast the DateTime column to timestamp so it processes as a timestamp column instead of a string.
- A limitation exists where Spark does not support decimal values with fixed precision greater than 38. For more information, see the official Spark documentation.
    - A possible workaround is to cast the data type as a string.
- The GEOGRAPHY column data type is not supported on Pushdown for BigQuery.
- When referencing quoted identifiers with backticks ` in the query connection, add –`corefetchmode` to the DQ Job command line to prevent the removal of the backticks ` at runtime.
- Using subqueires in a dataset query is currently unsupported. Instead, we recommend changing the query to a view.

Setting up cross-project access for BigQuery datasets

This page shows you how to make cross-account BigQuery datasets accessible to the GCP IAM service account in your project, and how to configure a BigQuery connection in Collibra DQ to run DQ checks against cross-account BigQuery datasets.

> **Note**
> In Pullup mode, BigQuery connections can only support one additional project.
> In Pushdown mode, BigQuery connections can support multiple additional projects. However, we do not recommend exceeding 3 additional projects.

The following diagram shows the architecture of cross-account access for BigQuery datasets. In this topic, Project A refers to the billing account in the panel on the left of this diagram, and Project B refers to the hosting account in the panel on the right.



Before you begin

- You have created a BigQuery connection.

Prerequisites

- You have a GCP IAM service account owned by Project A and its corresponding access JSON file.
- You have read access on BigQuery datasets hosted in Project B for the GCP service account owned by Project A.
    - You have at least roles/bigquery.dataViewer permissions on the BigQuery datasets.
- You have read and write access on the temporary dataset hosted by Project A for the GCP IAM service account.
    - You have at least roles/bigquery.jobUser and roles/bigquery.readSessionUser at the project level, and roles/bigquery.dataOwner permissions on the temporary dataset.
- The datasets of Project B and the temporary dataset of Project A are in the same zone.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Connections**.
   » The Connection Management page opens.
3. Locate your BigQuery connection, then click .
   » The Edit Connection modal appears.
4. Append the following properties to the **Connection URL**:

```
AdditionalProjects=project-b;QueryProperties=dataset_pro-
ject_id=project-b
```

> **Note** Set the value of `project-b` according to your specific requirements.

5. Click the **Properties** tab.
6. Add the following **Driver Properties** to enable materialized view for optimized performance of the connector:

```
materializationProject=project-a;materializationDataset=temporary_dataset_name
```

> **Note** Set the values of `project-a` and the `temporary_dataset_name` according to your specific requirements.

7. Click **Submit**.
8. Click **Explorer** and verify that the records in Explorer are the same as in your BigQuery connection.
9. When you create a new DQ Job, update the `-q` in the SQL Query editor with the following template query to identify the project:

```
-q "select * from project-b.dataset_name.table_name"
```

> **Note** Set the values of `project-b`, `dataset_name`, and `table_name` according to your specific requirements.

> **Warning** If you do not update the query with these values, the default project is used and Collibra DQ may throw an error that it cannot find your cross-project dataset in the expected location.

Configuring Workload Identity for Google BigQuery

Workload Identity

Workload Identity is the recommended method for authenticating BigQuery connections GKE-based deployments of Collibra DQ because it provides improved security and manageability of your service account credentials JSON file. To use Workload Identity to authenticate, ensure

that you fulfill the requirements and select **Workload Identity** from the Authentication Type dropdown menu when you set up your Google BigQuery connection.

Workload Identity permissions

To use Workload Identity to authenticate your connection between Collibra DQ and Google BigQuery, ensure that you satisfy the following requirements:

- Enable the GKE API.
- Enable the IAM Service Account Credentials API.
- You have the IAM role `roles/container.admin`.
- You have the IAM role `roles/iam.serviceAccountAdmin`.
- You have Workload Identity enabled and meet all additional service account requirements.
- You have `ROLE_ADMIN` assigned to your user in Collibra DQ.

> **Note** These roles and permissions are only required when you use the Workload Identity option to authenticate your connection.

Prerequisites

- Assign your service account to your Collibra DQ EKS Cluster or VM.
- Add your service account to the GCP project where Google BigQuery resources reside.
- Grant service account role(s) for your desired Google BigQuery access.

> **Important** Workload Identity is a supported authentication type for Collibra DQ deployments on GCP with either GKE or Google Compute Engine (GCE). If you use GKE, ensure you have Workload Identity Federation enabled against the GKE cluster and have BigQuery access enabled. If you use GCE, ensure the attached Service Account has BigQuery access enabled.

1. Authenticate to API

a. Use the following POST call to authenticate, replacing the placeholder variables in the `${}` with your actual values.

```
curl --location 'https://${dq-server-url.ex-
ample.com}/v3/auth/signin' \
--header 'Content-Type: application/json' \
--header 'Accept: */*' \
--data '{
  "username": "${ExampleUsername}",
  "password": "${ExamplePassword123}",
  "iss": "${ExampleTenantName}"
}'
```

b. Copy the token from the response.

```
"username": "ExampleUser",
"token": "${token}"
```

2. Create a connection

a. Copy the Agent UUID and ID number from the Collibra DQ UI, replacing the placeholder variables in the `${}` with your actual values.

> **Note** If you are using mutli-tenancy, the token for the multi-tenant admin user is required.

```
curl --location 'https://${dq-server-url.ex-
ample.com}/v2/getagents' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer ${token}'
```

b. Use the following POST call to create a connection, replacing the placeholder variables in the `${}` with your actual values.

```
curl --location
'https://${dq-server-url.ex-
ample.com}/v2/addconnection?Alias=${BigQuery_WI_Con-
nec-
tion}&Host-
=jd-
bc%3Abigquery%3A%2F%2Fht-
tps%3A%2F%2Fwww.-
googleapis.com%2Fbigquery%2Fv2%3A443%3BProjectId%${project_
id}%3BTimeOut%3D3600&Port-
=443&drive-
r=com.simba-
.google-
bigquery.jd-
bc42.Driver-
&user-
name-
=x&pas-
sword-
=x&drive-
rloca-
tion-
=%2-
Fop-
t%2Fowl%2Fdriver-
s%2Fbigquery&driver-
props-
=&isHiv-
e=0&use-
pwd-
mgr-
=0&iske-
rb=0&key-
ytab=&principal=&isglobal=1&conntype=jdbc&authtype=workload-
iden-
tity&isPush-
down=0&dbBrandName=BIGQUERY&agentId=${ID}&agentUUID=${agent-
UUID}&archiveBreaks=false' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer ${token}' \
--form
'agentIds="[{\"id\":${ID},\"uuid\":\"${agent-UUID}\"}]"'
```

3. Verify the connection

a. Use the following GET call to verify the connection was successfully created, replacing the placeholder variables in the `${}` with your actual values.

```
curl --location
'https://${dq-server-url.ex-
ample.com/v2/getconnectionsByDbBrand?dbBrandName=BIGQUERY' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer ${token}'
```

4. Use Workload Identity to authenticate your connection

a. From the Connections page in the Admin Console, add a BigQuery connection and, under Authentication Type, click the dropdown menu.
b. Select Workload Identity from the list.
c. Finish setting up your BigQuery connection and click **Submit** to save it.

Connecting to Hive

This section contains details for Hive connections.

General information

Field	Description
Data source	Hive
Supported versions	2.6.25
Connection string	`jdbc:hive2://`
Packaged?	✕ No
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes

Field	Description
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Hive data into Data Quality & Observability Classic, you need the following permissions.

- The Kerberos user has read permissions on Hive tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✗ No	Is Hive	Option	Uses the Hive server engine for distributed speed and scale. This option sets `hiveNative` options for Spark Jobs.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string path of your Hive connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:hive2://${host}:10000/default;AuthMech=1;KrbHostFQDN=_HOST;KrbServiceName=hive;SSL=1;AllowSelfSignedCerts=1`
✓ Yes	Driver Name	String	The driver class name of your connection. `com.cloudera.hive.jdbc.HS2Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `10000`
✗ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent that submits your Spark job for processing.

Required	Connection Property	Type	Value
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.

✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**
			
			Optionally add the following driver property to remove the enforcement of unique column names:
			
			`hive.resultset.use.unique.column.names=false`
			
			Optionally add the following driver properties to set the Spark conf spark.hadoop.hive.metastore.uris and allow Spark to read through a warehouse catalog, such as Hive:
			
			`dq.storage.cxn=${connection-name},dq.metastore.host=${hostname},dq.metastore.port=9083`

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Principal	The Kerberos entity to authenticate and grant access to your connection.

Required	Field	Description
✓ Yes	Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. **Example** /tmp/keytab/hive_user.keytab
✓ Yes	Password	The secret credential associated with your Kerberos principal.
✓ Yes	Script	The file path that contains the script file used to interact with and authenticate a Kerberos user. **Example** /tmp/keytab/hive_pwd_mgr.sh
✗ No	Param $1	Optional. Additional Kerberos parameter.
✗ No	Param $2	Optional. Additional Kerberos parameter.
✗ No	Param $3	Optional. Additional Kerberos parameter.
✓ Yes	TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

Known limitations

- When a column starts with a number, special character, or space, you may need to wrap the column name in square brackets `[ ]`, to prevent column names from being stripped.

Connecting to CDH 5.16 Hive SSL/TLS/Kerberos Setup

The Cloudera Hive JDBC drivers used
https://www.cloudera.com/downloads/connectors/hive/jdbc/2-5-16.html

JDBC Connection String used

jdbc:hive2://cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal:10000/default;AuthMech=1;KrbHostFQDN=cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal;KrbRealm=CW.COM;KrbServiceName=hive;SSL=1;SSLKeyStore=/opt/cloudera/security/pki/cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal-server.jks;AllowSelfSignedCerts=1;SSLKeyStorePwd=password;principal=hive/cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal@CW.COM

What Does each option mean

Base connection string = jdbc:hive2://cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal:10000/default

Authentication identifier = AuthMech=1 (which states Kerberos)

Kerberos Hive Server FQDN = KrbHostFQDN=cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal

Kerberos Realm used = KrbRealm=CW.COM *(Not necessarily needed)*

Kerberos Service name = KrbServiceName=hive

Enabling SSL = SSL=1

The SSL KeyStore to be used to = SSLKeyStore=/opt/cloudera/security/pki/cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal-server.jks *(Could use SSL TrustStore also)*

Allow for Self Signed certifications to be OK = AllowSelfSignedCerts=1 *(our environment used self signed certs)*

Password to the KeyStore = SSLKeyStorePwd=password *(Not necessarily needed)*

Kerberos Principal to use to Authenticate with = principal=hive/cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal@CW.COM

Within the Collibra DQ Web UI you have to specify the following

Name = hivessl

Connection URL = jdbc:hive2://cdh-instance1.us-east1-a.c.owl-hadoop-cdh.internal:10000/default;AuthMech=1;KrbHostFQDN=cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal;KrbRealm=CW.COM;KrbServiceName=hive;SSL=1;SSLKeyStore=/opt/cloudera/security/pki/cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal-server.jks;AllowSelfSignedCerts=1;SSLKeyStorePwd=password;principal=hive/cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal@CW.COM

Port = 10000

Driver Name = `com.cloudera.hive.jdbc.HS2Driver`

Username = userspark@CW.COM

Password = password

Connecting to MongoDB

This section contains details for MongoDB connections.
Select an option from the dropdown menu to display information for a particular driver class.



General information

Field	Description
Data source	MongoDB
Supported version	3.12.1122.0.8326.0

Field	Description
Connection string	`jdbc:mongodb://`
Packaged?	✕ No ✓ Yes
Certified?	✕ No ✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes ✕ No
Processing capabilities	
Pushdown	✕ No
Spark agent	✓ Yes
Yarn agent	✓ Yes ✕ No
Parallel JDBC	✕ No ✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your MongoDB data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your MongoDB tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

✓ Yes	Connection URL	String	The connection string path of your MongoDB connection.
			For successful job execution, you must first configure a Spark Scratch Directory to allocate space for a schema cache. After you configure the Spark Scratch Directory, add to the connection string the temporary writable location `schema=/tmp/scratch/${mongo_example_file_name.xml}` for the schema cache to write, as shown in the example below. Replace `${mongo_example_file_name.xml}` with the file name corresponding to the database name.
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			Example
			`jdbc:mongodb://${host}.mongodb.net?AuthDatabase=admin&AuthScheme=SCRAM-SHA-1&UseSSL=true&schema=/tmp/scratch/${mongo_example_file_name.xml}`
			`jdbc:mongodb://${host}.mongodb.net?AuthDatabase=admin&AuthScheme=SCRAM-SHA-1&UseSSL=true&database=${dbName}`

✓ Yes	Driver Name	String	The driver class name used for your connection.
			`mongodb.jdbc.MongoDriver`
			`cdata.jdbc.mongodb.MongoDBDriver`

✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `443`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent used to submit your DQ Job.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your MongoDB account.
✓ Yes	Password	The password of your MongoDB account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/mongodb_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameters to authenticate your MongoDB connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your MongoDB connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your MongoDB connection.
✓ Yes	Principal	The Kerberos entity to authenticate and grant access to your connection.
✓ Yes	Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. **Example** /tmp/keytab/hive_user.keytab

Required	Field	Description
✓ Yes	Password	The secret credential associated with your Kerberos principal.
✓ Yes	Script	The file path that contains the script file used to interact with and authenticate a Kerberos user. **Example** /tmp/keytab/mongodb_pwd_mgr.sh
✗ No	Param $1	Optional. Additional Kerberos parameter.
✗ No	Param $2	Optional. Additional Kerberos parameter.
✗ No	Param $3	Optional. Additional Kerberos parameter.
✓ Yes	TGT	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

Command line example for basic spark-submit job

```
-lib "/opt/owl/drivers/mongodb/"
-h localhost:5432/postgres
-master local[*]
-ds tpch.lineitem_7
-br 10 -deploymode client
-q "select * from tpch.lineitem where l_shipdate between '${rd}
00:00:00.000+0000'
and '${rdEnd} 00:00:00.000+0000' "
-bhlb 10 -rd "1998-12-01"
-driver "mongodb.jdbc.MongoDriver"
-loglevel INFO -cxn MongoDB -rdEnd "1998-12-02"
```

> **Note** For more details about the various Create and Alter SQL statements and table-level actions, see the official MongoDB documentation.

Connecting to MySQL

This section contains details for MySQL connections.

General information

Field	Description
Data source	MySQL
Supported versions	8.0.30
Connection string	`jdbc:mysql://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	

Field	Description
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your MySQL data into Data Quality & Observability Classic, you need the following permissions.

- Read access on MySQL tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your MySQL connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:mysql://${host}:3306/${dbName}?useSSL=false`
✓ Yes	Driver Name	String	The driver class name of your MySQL connection. `com.mysql.cj.jdbc.Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `3306`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your MySQL account.
✓ Yes	Password	The password of your MySQL account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/mysql_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameters to authenticate your MySQL connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your MySQL connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your MySQL connection.

Connecting to Oracle

This section contains details for Oracle connections.

General information

Field	Description
Data source	Oracle
Supported versions	12.2.0.1.0

Field	Description
Connection string	`jdbc:oracle:thin`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes

> **Note** Pushdown for Oracle is currently available as a public preview. Since this is a preview feature, some capabilities may be limited. For more information on our preview program, go to Previews at Collibra.

Field	Description
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Oracle data into Data Quality & Observability Classic, you need the following permissions.

- Read access on Oracle database tables.
- The Kerberos user has read permissions on Oracle tables (if using Kerberos authentication)
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

✓ Yes	Connection URL	String	The connection string path of your Oracle connection.
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			Example Using SID
			`jdbc:oracle:thin:@${host}:SID`
			Example Using database service name
			`jdbc:oracle:thin:@//${host}:${port}/service_name`

Required	Connection Property	Type	Value
✓ Yes	Driver Name	String	The driver class name of your connection. `oracle.jdbc.OracleDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `1521`
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Connection properties recognized by Oracle JDBC drivers

Name	Short Name	Type	Description
user	n/a	String	The user name for logging into the database
password	n/a	String	The password for logging into the database
database	server	String	The connect string for the database
internal_logon	n/a	String	A role, such as `sysdba` or `sysoper`, that allows you to log on as `sys`
defaultRowPrefetch	prefetch	String (containing integer value)	The default number of rows to prefetch from the server (default value is "10")
remarksReporting	remarks	String (containing boolean value)	"true" if getTables() and getColumns() should report TABLE_REMARKS; equivalent to using setRemarksReporting() (default value is "false")
defaultBatchValue	batchvalue	String (containing integer value)	The default batch value that triggers an execution request (default value is "10")
processEscapes	n/a	String (containing boolean value)	"false" to disable escape processing for statements (Statement or PreparedStatement) created from this connection. Set this to "false" if you want to avoid many calls to `Statement.setEscapeProcessing(false);`. This is espcially usefull for PreparedStatement where a call to `setEscapeProcessing (false)` would have no effect. The default is "true".

Name	Short Name	Type	Description
defaultNChar	n/a	String (containing boolean value)	"false" is the default. If set to "true", the default behavior for handling character datatypes is changed so that NCHAR/NVARCHAR2 become the default. This means that setFormOfUse() won't be needed anymore when using NCHAR/NVARCHAR2. This can also be set as a java property:`java -Doracle.jdbc.defaultNChar=true myApplication`
useFetchSizeWithLongColumn	n/a	String (containing boolean value)	"false" is the default. **Important** This is a thin-only property and should not be used with any other drivers. If set to "true", the performance when retrieving data in a 'SELECT' will be improved but the default behavior for handling LONG columns will be changed to fetch multiple rows (prefetch size). It means that enough memory will be allocated to read this data. So if you want to use this property, make sure that the LONG columns you are retrieving are not too big or you may run out of memory. This property can also be set as a java property:`java -Doracle.jdbc.useFetchSizeWithLongColumn=true myApplication`
SetFloatAndDoubleUseBinary	n/a	String (containing boolean value)	"false" is the default. If set to "true", causes the java.sql.PreparedStatment setFloat and setDouble API's to use internal binary format as for BINARY_FLOAT and BINARY_DOUBLE parameters. See oracle.jdbc.OraclePreparedStatement setBinaryFloat and setBinaryDouble

https://docs.oracle.com/cd/E11882_01/appdev.112/e13995/oracle/jdbc/OracleDriver.html

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Oracle account.
✓ Yes	Password	The password of your Oracle account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/oracle_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameters to authenticate your Oracle connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your Oracle connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your Oracle connection.
✓ Yes	Principal	The Kerberos entity to authenticate and grant access to your connection. **Example** test_user@example.collibra.com

Required	Field	Description
✓ Yes	Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. **Example** /tmp/keytab/oracle_user.keytab **Important** You must update your keytab secrets file with the following values. **Key**: cdq_user.keytab **Label**: base64 encoded string
✓ Yes	Password	The secret credential associated with your Kerberos principal.
✓ Yes	Script	The file path that contains the script file used to interact with and authenticate a Kerberos user. **Example** /tmp/keytab/oracle_pwd_mgr.sh
✗ No	Param $1	Optional. Additional Kerberos parameter.
✗ No	Param $2	Optional. Additional Kerberos parameter.
✗ No	Param $3	Optional. Additional Kerberos parameter.
✓ Yes	TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

Limitations

- There is an Apache Spark limitation where Pullup jobs on Oracle datasets that use Timestamp with Time Zone column types fail in the LOAD activity. For more information about the limitation and a workaround, see the support article.
- Because Oracle normally converts all characters in table and schema names to upper-case, Oracle also treats lower case characters as special characters. A possible work-around is to do the following:
    a. Go to Explorer and click Create DQ Job on an Oracle table.
    b. In the Query box, wrap the schema and table name with double quotes ". For example, `select * from "schema"."tablename"`.
    c. Go to the Run page and click the Run CMD tab.
    d. Click the lock icon to unlock the command line.
    e. Wrap backslashes \ around the double quoted schema and table names, and add a whitespace before the ending double quote of the -q section. For example, `-q "select * from \"schema\".\"table\" "`
- The SDO_GEOMETRY column data type is not supported on Oracle connections.
- When you add a timeslice to a column in an Oracle dataset, you must use the "TO_DATE" function to ensure the successful execution of your DQ Job. When a column includes a timeslice without the "TO_DATE" function, the DQ Job will fail with the error, "ORA-00907: missing right parenthesis."
    - When you include `to_date` within the command line query, your DQ Job will run successfully with a non-zero time. In the following example of a command line query with a `to_date`, `TRADE_DATE` is the timestamp column:
        - `-owluser user -numexecutors 1 -lib "/opt/owl/drivers/oracle/" -executormemory 1g -bhminon -h dq:5432/dq?currentSchema=validation -executorcores 2 -ds ORACLE_DQUSER.NYSE_2 -master k8s:// -drivermemory 1g -deploymode cluster -q "select * from DQUSER.NYSE where TRADE_DATE = to_date('${rd}', 'yyyy-mm-dd hh24:mi:ss')" -rd "2018-01-16 05:00:00" -bhlb 10 -driver "oracle.jdbc.OracleDriver" -bhmaxon -loglevel INFO -cxn ORACLE`

Connecting to PostgreSQL

This section contains details for PostgreSQL connections.

General information

Field	Description
Data source	PostgreSQL
Supported versions	42.5.1
Connection string	`jdbc:postgresql://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your PostgreSQL data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your PostgreSQL database tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your PostgreSQL connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:postgresql://${host}/template1`
✓ Yes	Driver Name	String	The driver class name of your PostgreSQL connection. `org.postgresql.Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `5432`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. > **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. > **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your PostgreSQL account.
✓ Yes	Password	The password of your PostgreSQL account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/postgresql_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameter to authenticate your PostgreSQL connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your PostgreSQL connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your PostgreSQL connection.

Connecting to SAP HANA

This section contains details for SAP HANA connections.

General information

Field	Description
Data source	SAP HANA
Supported versions	2.9.12
Connection string	`jdbc:sap://`

Field	Description
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes While the SAP HANA driver is compatible with JDK 8, it is not included in the standard package. However, the JDK 8 compatible SAP HANA driver is available for external download from the Maven Repository . After you download the JDK 8-compatible jar, replace the jar in the `/opt/owl/drivers/sap` directory.
JDK 11	✓ Yes

Minimum user permissions

In order to bring your SAP HANA data into Data Quality & Observability Classic, you need the following permissions.

- Read access to your SAP HANA database tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.
- When using Archive Break Records, you need write access to the schema output location in order for Data Quality & Observability Classic to create tables in your database.
    - While SAPHANADB is the default schema, depending on the SAP product and your installation method, your default schema may be one of the following instead:
        - SAP<SID>
        - SAPSR3
        - SAPABAP<#>

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your SAP HANA connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:sap://${host}:${port}/?databaseName=${dbName}`
✓ Yes	Driver Name	String	The driver class name of your SAP HANA connection. `com.sap.db.jdbc.Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `30015`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your SAP HANA account.
✓ Yes	Password	The password of your SAP HANA account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/sap_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameter to authenticate your SAP HANA connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your SAP HANA connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your SAP HANA connection.

Supported reserved words in SQL queries

The following SAP HANA reserved words are supported in SQL queries:

- `union`
- `all`
- `case`
- `when`
- `then`
- `else`
- `end`

- in
- current_date
- current_timestamp
- current_time
- weekday
- add_days
- add_months
- add_years
- days_between
- to_timestamp
- quarter
- year
- month
- to_char

> **Tip** If the SAP HANA reserved word you are using is not listed above, submit an ideation request through the Ideation Portal.

Known limitations

- While SAP HANA does not support parentheses **()** in table names, it does support the **.:$/-;@#%^&*?!{}~+=** special characters.

Connecting to Snowflake

> **Important** Click the Pullup or Pushdown tab to display the Snowflake connection information for either processing method.

This section contains details for Snowflake Pullup connections.

General information

Field	Description
Data source	Snowflake
Supported versions	3.20.0
Connection string	`jdbc:snowflake://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
	Important To see information for Snowflake Pushdown, click the Pushdown tab above.
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes

Field	Description
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Snowflake data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Snowflake database tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

Required	Connection Property	Type	Value

✓ Yes	Connection URL	String	The connection string path of your Snowflake connection.
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			Example
			`jdbc:snowflake://${host}.snowflakecomputing.com/?db=${dbName}`
			Important If your connection URL contains special characters and your Kubernetes deployment of Collibra DQ is a version before 2023.08, then you need to set the following Helm value to ensure you do not receive an invalid connection exception:
			`--set 'global.web.security.xss.byPassUrls=/mirror/api\,/v2/addconnection'`

✓ Yes	Driver Name	String	The driver class name of your Snowflake connection. `com.snowflake.client.jdbc.SnowflakeDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `0`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Pushdown	Option	Switches Pushdown processing on or off. **Important** This option should not be selected for Pullup connections.

Required	Connection Property	Type	Value
✗ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true** Optionally add the following connection property to limit the databases and tables displayed in Explorer to only the tables to which a user has access. `CLIENT_METADATA_REQUEST_USE_CONNECTION_CTX=TRUE`

For more information about optional JDBC connection parameters, see the official Snowflake documentation.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Snowflake service account.
✓ Yes	Password	The password of your Snowflake service account.

Required	Field	Description
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/snowflake_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameter to authenticate your Snowflake connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your Snowflake connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your Snowflake connection.

Tip
To use JWT authentication, enter any value as the Username and your token value as the Password.

Important
If you use additional encryption algorithms for JWT authentication, you must set one of the following parameters during your deployment of Data Quality & Observability Classic, depending on your deployment type:

Helm-based deployments

Set the following parameter in the Helm Chart: `--set global.web.extraJvmOptions="-Dnet.snowflake.jdbc.enableBouncyCastle=true"`

Standalone deployments

> Set the following environment variable in the owl-env.sh: `-export EXTRA_JVM_OPTIONS="-Dnet.snowflake.jdbc.enableBouncyCastle=true"`

Configuring key pair authentication via the Connection URL

The table below shows the properties used to configure key pair authentication with a Connection URL in the New JDBC Connection modal.

Property	Description
Connection URL	jdbc:snowflake://<URL>?db=owluserdb&warehouse=owluserdb&authenticator=snowflake_jwt&private_key_file=/<your-file-path>/keytab/snowflake_key.p8&private_key_file_pwd=<your-password> The file name should include a .p8 extension, as shown in the example above. **Note** Connection URL parameters must be appended by ampersand (&).
Username	The username for which the private key is created. For example, DQ_User.
Password	The password field must be left empty.

Configuring key pair authentication via the Driver Properties

The table below shows the properties used to configure key pair authentication with the Driver Properties in the New JDBC Connection modal.

Property	Description
Username	The username for which the private key is created. For example, DQ_User.

Property	Description
Password	The password field must be left empty.
Driver Properties	authenticator=snowflake_jwt,private_key_file=/<your-file-path>/snowflakeKeys/rsa_key2.p8,private_key_file_pwd=<your-password> The file name should include a .p8 extension, as shown in the example above.

> **Note** Driver properties must be comma separated, as shown in the example above.

Configuring key pair authentication via Helm

To be able to mount extra volumes to the DQ Agent pod, configure Snowflake key pair authentication at the DQ Agent pod.

1. Create a kubernetes secret to hold the snowflake private key. For example:

```
apiVersion: v1
kind: Secret
metadata:
name: dq-snowflake-private-key
data:
snowflake-private-key.p8: IA==
```

   In the example above, content in `snowflake-private-key.p8` is base64.

2. Update the values in the Helm script to attach the kubernetes secrets to DQ Web and DQ Agent pods. For example:

```
web:
  extraStorage:
    volumes:
      - name: dq-snowflake-private-key
        secret:
          secretName: dq-snowflake-private-key
    volumeMounts:
      - name: dq-snowflake-private-key
        mountPath: /tmp/dq-snowflake-private-key
agent:
  extraStorage:
    volumes:
      - name: dq-snowflake-private-key
        secret:
          secretName: dq-snowflake-private-key
    volumeMounts:
      - name: dq-snowflake-private-key
        mountPath: /tmp/dq-snowflake-private-key
```

> **Note** The mount path should be same for both DQ Web and DQ Agent

3. Create a DQ connection which uses the Snowflake private key located at:

```
/tmp/dq-snowflake-private-key/snowflake-private-key.p8
```

Additional connection parameters

Parameter	Description
`<account_identifier>`	Specifies the account identifier for your Snowflake account. For more details, see Account Identifiers. For examples of the account identifier used in a JDBC connection string, see Examples.
`<connection_params>`	Specifies a series of one or more parameters, in the form of `<param>=<value>`, with each parameter separated by the ampersand character (`&`), and no spaces anywhere in the connection string.

For more information on individual connection parameters, see the official Snowflake documentation about JDBC Driver Connection Parameter Reference.

Other parameters

Any session parameter can be included in the connection string. For example:

Parameter	Description
`CLIENT_SESSION_KEEP_ALIVE=<Boolean>`	Specifies whether to keep the current session active after a period of inactivity, or to force the user to login again. When the boolean value is `true`, Snowflake keeps the session active indefinitely, even if there is no activity from the user. When the boolean value is `false`, the user must log in again after four hours of inactivity. The default boolean value is `false`

For descriptions of all the session parameters, see the official Snowflake documentation about Parameters.

Optional parameter examples

The following is an example of the connection string that uses an account identifier that specifies the account `myaccount` in the organization `myorganization`.

```
jdbc:snowflake://myorganization-
myaccount.snowflakecomputing.com/?user=peter&warehou
se=mywh&db=mydb&schema=public
```

The following is an example of a connection string that uses the account locator `xy12345` as the account identifier:

```
jdbc:snowflake://xy12345.snowflakecomputing.com/?use
r=peter&warehouse=mywh&db=mydb&schema=public
```

> **Note** This example uses an account in the AWS US West (Oregon) region. If the account is in a different region or if the account uses a different cloud provider, you need to specify additional segments after the account locator.

PrivateLink

Please let us know if you are using PrivateLink for Snowflake. Setup can vary depending on the endpoint that is created. In most cases, use the private endpoint as a normal JDBC connection.

For additional information, see the Snowflake community article about troubleshooting AWS PrivateLink.

Advanced PrivateLink and proxy

The following is an example of a JDBC string connection we used to set up a Snowflake connection :

- <ACCOUNT_NAME> is the full link to the Snowflake instance with the PrivateLink.
- DQ installed on-prem in a private IaaS and DQ is behind a proxy.
- If the Snowflake instance is using a PrivateLink, whitelist the private link URL to bypass the proxy.
- In addition to connectivity to the Snowflake instance, the JDBC driver tries to access Snowflake Blob storage by connecting directly to some S3 buckets managed by Snowflake.
    - Those need to be whitelisted as well.

Command line example for basic spark-submit job

```
-h <IP_ADDRESS>:5432/postgres \
-drivermemory 4g \
-master spark://<SPARK_MASTER>:7077 \
-ds PUBLIC.TRANSLATION \
-deploymode client \
-q "select * from PUBLIC.TRANSLATION" \
-rd "2021-07-24" \
-driver "net.snowflake.client.jdbc.SnowflakeDriver" \
-cxn snowflake
```

This section contains details for Snowflake Pushdown connections.

General information

Field	Description
Data source	Snowflake
Supported versions	3.19.0
Connection string	`jdbc:snowflake://`

Field	Description
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✗ No
Yarn agent	✗ No
Parallel JDBC	✗ No
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Prerequisites

For first-time configurations, we recommend that you also successfully run the Pushdown setup script. If you do not run the setup script, ensure that you meet all the criteria that the script attempts to accomplish.

Minimum user permissions

In order to bring your Snowflake data into Data Quality & Observability Classic, you need the following permissions.

- USAGE and MONITOR privileges on all Snowflake database tables, external tables, views, schemas, and streams.
- OPERATE, USAGE, and MONITOR privileges on the Collibra DQ virtual data warehouse.
- When using Archive Break Records, ensure that you have the following:
    - USAGE privilege on the database and/or schema.
    - INSERT privilege on tables.
    - CREATE, READ, UPDATE, and DELETE privileges to prune records.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

Required	Connection Property	Type	Value

✓ Yes	Connection URL	String	The connection string path of your Snowflake connection.
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			Example
			`jdbc:snowflake://${host}.snowflakecomputing.com/?db=${dbName}`
			Important If your connection URL contains special characters and your Kubernetes deployment of Collibra DQ is a version before 2023.08, then you need to set the following Helm value to ensure you do not receive an invalid connection exception:
			`--set 'global.web.security.xss.byPassUrls=/mirror/api\,/v2/addconnection'`

✓ Yes	Driver Name	String	The driver class name of your Snowflake connection. `com.snowflake.client.jdbc.SnowflakeDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `0`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing. You do not need to select an option because Pushdown Jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✓ Yes	Pushdown	Option	Switches Pushdown processing on or off. **Important** This option must be selected for Pushdown processing to work correctly.

Required	Connection Property	Type	Value
✕ No	Archive Break Records	Option	Allows you to write break records directly to a Snowflake database or schema. **Important** Ensure that you have the necessary privileges to write break records to Snowflake.
✓ Yes	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true** Add the following driver property: `CLIENT_METADATA_REQUEST_USE_CONNECTION_CTX=TRUE`

For more information about optional JDBC connection parameters, see the official Snowflake documentation.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Snowflake service account.
✓ Yes	Password	The password of your Snowflake service account.

Required	Field	Description
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/snowflake_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameter to authenticate your Snowflake connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your Snowflake connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your Snowflake connection.

Tip To use JWT authentication, enter any value as the Username and your token value as the Password.

Important
If you use additional encryption algorithms for JWT authentication, you must set one of the following parameters during your deployment of Data Quality & Observability Classic, depending on your deployment type:

Helm-based deployments

Set the following parameter in the Helm Chart: `--set global.web.extraJvmOptions="-Dnet.snowflake.jdbc.enableBouncyCastle=true"`

Standalone deployments

Set the following environment variable in the owl-env.sh: `-export EXTRA_JVM_OPTIONS="-Dnet.snowflake.jdbc.enableBouncyCastle=true"`

Configuring key pair authentication via the Connection URL

The table below shows the properties used to configure key pair authentication with a Connection URL in the New JDBC Connection modal.

Property	Description
Connection URL	jdbc:snowflake://<URL>?db=owluserdb&warehouse=owluserdb&authenticator=snowflake_jwt&private_key_file=/<your-file-path>/keytab/snowflake_key.p8&private_key_file_pwd=<your-password> The file name should include a .p8 extension, as shown in the example above. **Note** Connection URL parameters must be appended by ampersand (`&`).
Username	The username for which the private key is created. For example, DQ_User.
Password	The password field must be left empty.

Configuring key pair authentication via the Driver Properties

The table below shows the properties used to configure key pair authentication with the Driver Properties in the New JDBC Connection modal.

Property	Description
Username	The username for which the private key is created. For example, DQ_User.

Property	Description
Password	The password field must be left empty.
Driver Properties	authenticator=snowflake_jwt,private_key_file=/<your-file-path>/snowflakeKeys/rsa_key2.p8,private_key_file_pwd=<your-password> The file name should include a .p8 extension, as shown in the example above.

> **Note** Driver properties must be comma separated, as shown in the example above.

Configuring key pair authentication via Helm

To be able to mount extra volumes to the DQ Agent pod, configure Snowflake key pair authentication at the DQ Agent pod.

1. Create a kubernetes secret to hold the snowflake private key. For example:

```
apiVersion: v1
                    kind: Secret
                    metadata:
                    name: dq-snowflake-private-key
                    data:
                snowflake-private-key.p8: IA==
```

   In the example above, content in `snowflake-private-key.p8` is base64.

2. Update the values in the Helm script to attach the kubernetes secrets to DQ Web and DQ Agent pods. For example:

```
web:
                    extraStorage:
                    volumes:
                    - name: dq-snowflake-private-key
                    secret:
                    secretName: dq-snowflake-private-key
                    volumeMounts:
                    - name: dq-snowflake-private-key
                    mountPath: /tmp/dq-snowflake-private-
key
                    agent:
                    extraStorage:
                    volumes:
                    - name: dq-snowflake-private-key
                    secret:
                    secretName: dq-snowflake-private-key
                    volumeMounts:
                    - name: dq-snowflake-private-key
                mountPath: /tmp/dq-snowflake-private-key
```

> **Note** The mount path should be same for both DQ Web and DQ Agent

3. Create a DQ connection which uses the Snowflake private key located at:

```
/tmp/dq-snowflake-private-key/snowflake-private-key.p8
```

Limitations

- Snowflake Pushdown connections support all regex special characters in column names except for square brackets **[ ]**.

Connecting to SQL Server

This section contains details for SQL Server connections.

General information

Field	Description
Data source	SQL Server
Supported versions	DQ currently ships with Microsoft JDBC Driver 12.6.4 for SQL Server. Please refer to the support matrix for Microsoft SQL Server version compatibility.
Connection string	`jdbc:sqlserver://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	

Field	Description
JDK 8	✕ No While the Java 8 version is not officially supported, you can replace the SQL Server driver in the /opt/owl/drivers/mssql folder with the Java 8 version of the supported driver. You can find Java 8 versions of the supported SQL Server on the Maven Repository.
JDK 11	✓ Yes

Minimum user permissions

In order to bring your SQL Server data into Data Quality & Observability Classic, you need the following permissions.

- Read access on SQL Server database tables.
- The Kerberos user has read permissions on SQL Server tables (if using Kerberos authentication).
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string path of your SQL Server connection. For more information about how to construct your connection URL, see Building the connection URL in the official SQL Server documentation. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:sqlserver://${host}:1433;databaseName=${dbName};encrypt=true;trustServerCertificate=true`
✓ Yes	Driver Name	String	The driver class name of your connection. `com.microsoft.sqlserver.jdbc.SQLServerDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `1433`
✗ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✗ No	Source Name	String	N/A

Required	Connection Property	Type	Value
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Important When you use Active Directory Password, you need to add **authentication=ActiveDirectoryPassword** to your connection URL and fill out the Username and Password fields described in the table below. For example, the following connection URL properly appends **authentication=ActiveDirectoryPassword** to the end of the URL:

```
jdbc:sqlserver://example.demo.azuresynapse.net:1433;database=exampleDB;encrypt=true;authentication=ActiveDirectoryPassword
```

Important When you use Active Directory Service Principal, you need to add **authentication=ActiveDirectoryServicePrincipal** to your connection URL and fill out the Principal and Principal Secret fields described in the table below. For example, the

following connection URL properly appends **authentication=ActiveDirectoryServicePrincipal** to the end of the URL:
`jdbc:sqlserver://example.demo.azuresynapse.net:1433;database=exampleDB;encrypt=true;`
`authentication=ActiveDirectoryServicePrincipal`

Important When you use Active Directory MSI, you need to add **authentication=ActiveDirectoryManagedIdentity** to your connection URL and fill out the MSI Client ID field described in the table below. For example, the following connection URL properly appends **authentication=ActiveDirectoryManagedIdentity** to the end of the URL:
`jdbc:sqlserver://example.demo.azuresynapse.net:1433;database=exampleDB;encrypt=true;`
`authentication=ActiveDirectoryManagedIdentity`

Additionally, your Synapse database must be on an Azure VM with Managed Identity enabled, as the Synapse database cannot be accessed outside of the Azure VM.

Important When you use NTLM, ensure you add the following parameters to the Connection end of the connection URL:
encrypt=true;trustServerCertificate=true;integratedSecurity=true;authenticationScheme=NTLM;domainName=EVHC

In a full connection URL this might look something like the following:
`jdbc:sqlserver://example.demo.azuresynapse.net:1433;database=exampleDB;`
`encrypt=true;trustServerCertificate=true;integratedSecurity=true;authenticationScheme=NTLM;domainName=EVHC`

Required	Field	Description
✓ Yes	Username	The username of your SQL Server account.
✓ Yes	Password	The password of your SQL Server account.

Required	Field	Description
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/sqlserver_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameter to authenticate your SQL Server connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your SQL Server connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your SQL Server connection.
✓ Yes	Username	The Azure Active Directory (Azure AD) User Principal Name.
✓ Yes	Password	The Azure Active Directory (Azure AD) password.
✓ Yes	Principal	The Microsoft Entra service principal ID.
✓ Yes	Principal Secret	The Microsoft Entra service principal secret.
✓ Yes	Principal	The Kerberos entity to authenticate and grant access to your connection. **Example** test_user@example.collibra.com

Required	Field	Description
✓ Yes	Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. **Example** /tmp/keytab/sqlserver_user.keytab
✓ Yes	Password	The secret credential associated with your Kerberos principal.
✓ Yes	Script	The file path that contains the script file used to interact with and authenticate a Kerberos user. **Example** /tmp/keytab/sqlserver_pwd_mgr.sh
✗ No	Param $1	Optional. Additional Kerberos parameter.
✗ No	Param $2	Optional. Additional Kerberos parameter.
✗ No	Param $3	Optional. Additional Kerberos parameter.
✓ Yes	TGT	The ticket-granting ticket cache that stores the TGT to authenticate your connection.
✓ Yes	MSI Client ID	The Microsoft Entra MSI Client ID

Troubleshooting

The Microsoft JDBC Driver for SQL Server requires that TCP/IP be installed and running to communicate with your SQL Server database. You can use the SQL Server Configuration

Manager to verify which network library protocols are installed.

A database connection attempt might fail for many reasons. These can include the following:

- TCP/IP is not enabled for SQL Server, or the server or port number specified is incorrect. Verify that SQL Server is listening with TCP/IP on the specified server and port. This might be reported with an exception similar to: "The login has failed. The TCP/IP connection to the host has failed." This indicates one of the following:
  - SQL Server is installed but TCP/IP has not been installed as a network protocol for SQL Server by using the SQL Server Network Utility for SQL Server 2000 (8.x), or the SQL Server Configuration Manager for SQL Server 2005 (9.x) and later.
  - TCP/IP is installed as a SQL Server protocol, but it is not listening on the port specified in the JDBC connection URL. The default port is 1433, but SQL Server can be configured at product installation to listen on any port. Make sure that SQL Server is listening on port 1433. Or, if the port has been changed, make sure that the port specified in the JDBC connection URL matches the changed port. For more information about JDBC connection URLs, see Building the connection URL.
  - The address of the computer that is specified in the JDBC connection URL does not refer to a server where SQL Server is installed and started.
  - The networking operation of TCP/IP between the client and server running SQL Server is not operable. You can check TCP/IP connectivity to SQL Server by using telnet. For example, at the command prompt, type `telnet 192.168.0.0 1433` where 192.168.0.0 is the address of the computer that is running SQL Server and 1433 is the port it is listening on. If you receive a message that states "Telnet cannot connect," TCP/IP is not listening on that port for SQL Server connections. Use the SQL Server Network Utility for SQL Server 2000 (8.x), or the SQL Server Configuration Manager for SQL Server 2005 (9.x) and later to make sure that SQL Server is configured to use TCP/IP on port 1433.
  - The port that is used by the server has not been opened in the firewall. This includes the port that is used by the server or optionally, the port associated with a named instance of the server.
- The specified database name is incorrect. Make sure that you are logging on to an existing SQL Server database.
- The user name or password is incorrect. Make sure that you have the correct values.
- When you use SQL Server Authentication, the JDBC driver requires that SQL Server is installed with SQL Server Authentication, which is not the default. Make sure that this option is included when you install or configure your instance of SQL Server.

See also

Diagnosing problems with the JDBC driver

Connecting to SQL Server with the JDBC driver

FAQ

SQL Server represents RDS, Azure SQL, and traditional SQL Server installations

Connecting to Sybase

This section contains details for Sybase connections.

General information

Field	Description
Data source	Sybase
Supported versions	7.2.0
Connection string	`jdbc:sybase:`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No

Field	Description
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✗ No
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Sybase data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Sybase database tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your Sybase connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:sybase:Tds:${host}:5000/template1?ENCRYPT_PASSWORD=false`

Required	Connection Property	Type	Value
✓ Yes	Driver Name	String	The driver class name of your Sybase connection. `com.sybase.jdbc4.jdbc.SybDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `5000`
✗ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✗ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Sybase service account.
✓ Yes	Password	The password of your Sybase service account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/sybase_pwd_mgr.sh
✗ No	Param $1	Optional. An additional parameter to authenticate your Sybase connection.
✗ No	Param $2	Optional. An additional parameter to authenticate your Sybase connection.
✗ No	Param $3	Optional. An additional parameter to authenticate your Sybase connection.

Connecting to Teradata

This section contains details for Teradata connections.

General information

Field	Description
Data source	Teradata
Supported versions	20.0.0.20
Connection string	`jdbc:teradata://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Pushdown	✗ No
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Teradata data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Teradata database tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✓ Yes	Connection URL	String	The connection string path of your Teradata connection. When referring to the example below, replace the `${value}` sections of the connection URL with your actual value. **Example** `jdbc:teradata://${host}/${database}`
✓ Yes	Driver Name	String	The driver class name of your Teradata connection. `com.teradata.jdbc.TeraDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `0`

Required	Connection Property	Type	Value
✕ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✓ Yes	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true** Optionally add the following driver property: `RBAC=USR_RBAC_CCB_PDQOWL_INTL`

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Required	Field	Description
✓ Yes	Username	The username of your Teradata account.
✓ Yes	Password	The password of your Teradata account.
✓ Yes	Script	The file path that contains the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/teradata_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameter to authenticate your Teradata connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your Teradata connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your Teradata connection.

Connecting to Trino

This section contains details for Trino connections.

General information

Field	Description
Data source	Trino
Supported versions	1.0.50

Field	Description
Connection string	`jdbc:trino://`
Packaged?	✓ Yes
Certified?	✕ No
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✓ Yes
Spark agent	✓ Yes
Yarn agent	✓ Yes
Parallel JDBC	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Trino data into Data Quality & Observability Classic, you need the following permissions.

- Read access on your Trino database tables.
- ROLE_ADMIN assigned to your user in Collibra DQ.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.

✓ Yes	Connection URL	String	The connection string path of your Trino connection.
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			Example
			`jdbc:trino://${host}:443/${catalogName}?SSL=true&source=jdbc:presto;ConnCatalog=${catalogName}`
			Important Your connection URL must contain the following substring: `&source=jdbc:presto`
			Important If your connection URL contains special characters and your Kubernetes deployment of Collibra DQ is a version before 2023.08, then you need to set the following Helm value to ensure you do not receive an invalid connection exception:
			`--set 'global.web.security.xss.byPassUrls=/mirror/api\,/v2/addconnection'`

✓ Yes	Driver Name	String	The driver class name of your Trino connection. `io.trino.jdbc.TrinoDriver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `443`

Required	Connection Property	Type	Value
✗ No	Limit Schemas	Option	Allows you to manage usage and restrict visibility to only the necessary schemas in the Explorer tree. See Limiting schemas to learn how to limit schemas from the Connection Management page. **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent that submits your Spark job for processing.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✓ Yes	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.

Configuring Access Token Manager authentication

Prerequisites

You have added a script file that contains or can retrieve an access token to a folder accessible to Data Quality & Observability Classic.

Steps

1. Update the Connection URL to the following format: `jdbc:trino://${host}:${port}/${catalog}?SSL=true&source=jdbc:presto&accessToken=${accessToken}`
2. Select **Access Token Manager** from the Authentication Type dropdown menu.
3. Enter the User ID of the IdP account in the Username input field.
4. Enter the file path containing the access token script file in the Script input field. For example, /opt/owl/config/get_trino.sh
5. Click Submit.

Required	Field	Description
✓ Yes	Username	The username of your Trino account. The User ID of the IdP account.
✓ Yes	Password	The password of your Trino account.
✓ Yes	Script	The file path containing the access token script file in the Script input field. **Example** /opt/owl/config/get_trino.sh

Required	Field	Description
✕ No	Param $1	Optional. Additional parameter.
✕ No	Param $2	Optional. Additional parameter.
✕ No	Param $3	Optional. Additional parameter.
✓ Yes	Script	The file path containing the script file that the password manager uses to interact with and authenticate a user account. **Example** /tmp/keytab/trino_pwd_mgr.sh
✕ No	Param $1	Optional. An additional parameter to authenticate your Trino connection.
✕ No	Param $2	Optional. An additional parameter to authenticate your Trino connection.
✕ No	Param $3	Optional. An additional parameter to authenticate your Trino connection.

Limitations

- Trino connection URLs must contain the following substring: `&source=jdbc:presto`
  - This requirement will be addressed in a future release.
- When using Pushdown for Trino, the following column data types are not supported:
  - HYPERLOGLOG
  - P4HYPERLOGLOG
  - QDIGEST
  - SETDIGEST
  - TDIGEST

Supported remote files

This section includes the connection details of remote files supported by Collibra DQ.

Supported remote file data sources

Data Source	Packaged	Certified	Archive Break Records
Amazon S3	✓ Yes	✓ Yes	✓ Yes
Azure Blob Storage	✓ Yes	✓ Yes	✓ Yes
Azure Data Lake Storage (Gen2)	✓ Yes	✓ Yes	✓ Yes
Google Cloud Storage	✓ Yes	✓ Yes	✗ No
Hadoop Distributed Files System (HDFS)	✓ Yes	✓ Yes	✗ No
Network File Storage (NFS)	✓ Yes	✓ Yes	✗ No

Supported file types

Because file formats differ in structure, you may need to prepare your data before establishing a connection.

Type	File structure	Notes
CSV (.csv)	Structured	The default delimiter is comma (for example, CSV).
PARQUET (.parquet)	Structured	N/A
AVRO (.avro)	Structured	N/A
JSON (.json)	Semi-structured	N/A

Type	File structure	Notes
ORC (.orc)	Semi-structured	N/A
XML (.xml)	Semi-structured	N/A
DELTA (.delta)	Semi-structured	N/A

Type	File structure	Notes
HUDI (.hudi)	Structured	You can only create DQ jobs for Hudi files from the directory-level, not from the individual file-level. When you create a DQ job from the Remote File Connections section of Explorer, expand a directory, then click **Create DQ Job**.

Type	File structure	Notes

Supported delimiters

The following table is a list of supported delimiters available in the Delimiter dropdown menu.

Type	Format	Description
Comma	CSV	`,` is used to separate values in the file. This is the default delimiter for files.
Tab	TSV	`tab` is used to separate values in the file.
Semicolon	CSV	`;` is used to separate values in the file.
Double Quote	CSV	`"` is used to separate values in the file.
Single Quote	CSV	`'` is used to separate values in the file.
Pipe	TXT	`|` is used to separate values in the text file.
SOH	TXT	A Unicode character 'START OF HEADING' (U+0001) is an invisible control character.
Custom	N/A	Add a custom delimiter. Support for custom delimiters may vary.

Known limitations

- Spark supports only one account configuration for remote connections. When archive break records are enabled, you need read and write access to the connection where break records are sent. Additionally, you must use the same remote account for the `-archivecxn` and `-cxn` command line options. For example, if you run a job on data from an ADLS connection, it must use the same ADLS connection to archive break records.
- DQ jobs that run on remote file connections with headers that contain white spaces fail with a requirement failed exception message. A possible workaround is to edit the DQ Job command line in the Run CMD tab and place single quotes `''` around the column name in `-q` and double quotes `""` around the contents of the `-header` flag.
- Filtergram on the Data Preview tab is not available for any remote file connection. Currently, there is not a workaround for this limitation.
- Array and nested array datatypes in JSON files are not supported.
- While Collibra DQ supports most UTF-8 encoded characters in column headers of file-based connections, some Chinese characters are not currently supported. Jobs that run with this type of unsupported characters fail with a mismatched input exception message.
- When you use Validate Source, the Update Source Scope button is not available for remote files. Update Source Scope is only visible for JDBC connections.

Connecting to Amazon S3

This section contains details for Amazon S3 connections.

General information

Amazon Simple Storage Service (Amazon S3) is an object storage service that Data Quality & Observability Classic treats as a Remote File Connection for accessing data. You can also use S3 buckets to export and store breaking records from Collibra DQ rules externally. This can be helpful if you need to control your storage environment behind a firewall.

Field	Description
Data source	Amazon S3

Field	Description
Supported versions	N/A
Connection string	`s3://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Analyze data	✓ Yes
Archive breaking records	✓ Yes
Estimate job	✓ Yes
Pushdown	✗ No
Processing capabilities	
Spark agent	✓ Yes
Yarn agent	✓ Yes

Minimum user permissions

In order for Collibra DQ to access your S3 bucket, you need the following permissions.

- ROLE_ADMIN in Collibra DQ.
- Read access on the S3 bucket where your data is stored.
- Read and write access on the S3 bucket where breaking records are archived from Collibra DQ. This is only necessary if you use the archive breaking records feature.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✕ No	HTTPS	Option	Converts the Connection URL field from an `s3://` to `https://` URI path.
✕ No	Path Style Access	Option	Allows access to object stores along HTTPS paths. This option is only available when HTTPS is enabled.
✓ Yes *When HTTPS is selected, Bucket Name is required.	Bucket Name	string	The exact name of the S3 bucket along the URI path you are attempting to access. **Example** `example_bucket`

Required	Connection Property	Type	Value

✓ Yes	Connection URL	String	The connection string path of your S3 connection. The path must start with s3:// and point to the root bucket, not a sub-folder.
			
			Example
			`s3://<bucket-name>`
			
			You can optionally add a key after the bucket name.
			
			Example
			`s3://<bucket-name>/key`

✓ Yes	Region	String	The AWS region in which the S3 bucket resides. The default region is US_EAST_1
✕ No	Target Agent	Option	The Agent used to submit your DQ Job.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.

Required	Connection Property	Type	Value
✓ Yes	Save Credentials	Option	Select this option after you enter your connection details. We recommend selecting this option to allow credentials to be shared with users across the Collibra DQ platform.
✕ No	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**
✕ No	Archive Breaking Records	Option	Select this option to automatically export CSV files containing the breaking records of DQ Job to your S3 bucket.
✕ No	Archive Location	String	The path along which break records will be archived when the Archive Break Records and HTTPS options are enabled. Specify a schema output location in your data source for break records to send. For example, **/write/FolderName** **Note** This option is only available when the HTTPS option is enabled.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



In the Driver Properties input field on the Properties tab, add the following property, adjusting the parameterized section (`${ }`) according to your IdP details: `dq.idp.url=${IdP-URL}`

> **Note** If your IdP uses scopes, you may need to add `dq.idp.scopes=${IdP-scope}` after the IdP URL.

Field	Description
Key	AWS security credentials that use an access key ID and secret access key combination to access S3 buckets.
Key	The access key ID for your Amazon S3 storage account.
Secret	The secret access key for your Amazon S3 storage account.
Instance Profile	The instance profile used to grant access to the EC2 instance to access your S3 bucket. Optionally select Assume Role and add an accessRole when you use Instance Profile. **Important** DQ Cloud does not support Instance Profile.
Username	The username credential required for the IdP service to authenticate a user.
Password	The password credential required for the IdP service to authenticate a user.

Field	Description
Assume Role	This is optional for Instance Profile and Key authentication. Select this option to generate a set of temporary security credentials that you can use to access AWS resources. To assume an S3 IAM Role on EKS-based deployments of Collibra DQ, select Assume Role, then optionally enter the information in the Role to Assume field.
Role to Assume	The IAM role associated with your EC2 instance. This is optional when assuming an S3 IAM Role on EKS-based deployments of Collibra DQ.

Password Manager

Field	Description
Username	The username used to sign into your password manager service.
Password	**Note** Password is not required when you use Password Manager.

Field	Description
Script	The script that contains the SAML role. The password manager program contains the information needed to automate password-related tasks when you attempt to sign in to Collibra DQ.
App ID	The Application Identifier used to help the IdP service authorize Collibra DQ when you attempt to sign in.
Safe	The storage vault where your password and other authentication information are stored.
Pwd Mgr Name	The name of your safe.
accessRole	The IAM role associated with your EC2 instance.

Connect to NetApp or Amazon S3 endpoint in URI format

To connect to NetApp or Amazon S3 endpoints in URI format, select the **HTTPS** option, specify a **Bucket Name**, and then add one of the following properties to the **Properties** tab, depending on the URI endpoint connection type to which you are connecting:

Connection	Property
Amazon S3 endpoint URI	**s3-endpoint=s3**
NetApp	**s3-endpoint=netapp**

Connect to MinIO with path style access

> **Note** The ability to connect to MinIO is only available in Collibra DQ versions 2024.03 or newer.

To connect to MinIO using path style access, select the **HTTPS** and **Path Style Access** options, specify a **Bucket Name**, and then add the following property to the **Properties** tab:

Connection	Property
MinIO	**s3-endpoint=minio**

> **Important** When connecting to MinIO, you must select the Path Style Access option.

Limitations

- Instance Profile is not supported on DQ Cloud deployments.
- Filtergram on the Data Preview tab is not available for S3 connections. Currently, there is not a workaround for this limitation.
- When leveraging the Spark 3.2.2 standalone installation that comes with the AWS marketplace installation, there is a limitation due to a jar mismatch for the AWS S3 Archive feature. The issue is that the existing hadoop-aws.3.2.1 .jar file is incompatible with the feature.
    - A workaround is to update the hadoop-aws-3.2.1.jar to hadoop-aws-3.3.1.jar in the spark/jars directory. The necessary .jar file can be obtained from the following link: Apache Downloads.

      > **Note** If you encounter any difficulties locating the necessary .jar file on the Apache Downloads page, contact your CS or SE for assistance.

Connecting to Azure Blob Storage

This section contains an overview of Azure Blob Storage.

General information

Field	Description
Data source	Azure Blob Storage
Supported versions	N/A
Connection string	`wasbs://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Analyze data	✓ Yes
Archive breaking records	✓ Yes
Estimate job	✓ Yes
Pushdown	✗ No
Processing capabilities	
Spark agent	✓ Yes
Yarn agent	✓ Yes

Minimum user permissions

In order for Collibra DQ to access your Azure Blob Storage containers, you need the following user permissions:

- ROLE_ADMIN in Collibra DQ.
- Read access on your Blob Storage containers to create a basic Azure Blob connection.
- Read and write access to the Blob Storage containers where breaking records are archived from Collibra DQ. This is only necessary if you use the archive breaking records feature.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name used for your connection.
✓ Yes	Connection URL	String	The connection string value of your ADLS connection. `wasbs://<container>@<account name>.blob.-core.windows.net/<path>/file name>` **Tip** See the Connection URL elements section for more information on the syntax.
✕ No	Target Agent	Text	The Agent used to submit your DQ Job.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Archive Breaking Records Location	Option	When you select this option and have followed the steps, you can export CSV files to the Azure Blob location.
✕ No	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Connection URL elements

Field	Description
File scheme	The `wasbs` protocol is used as the scheme identifier.
Container	The parent location that holds the files and folders. This is the same as file system in the Azure Data Lake Storage service.
Account name	The name given to your storage account during creation.
Path	A forward slash delimited `/` representation of the directory structure.
File name	The name of the individual file. This parameter is optional when you address a directory.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Field	Description
Storage Account	The Azure Storage account name.
Key	The authentication key for the storage account.
Storage Account	The Azure Storage account name.
Shared Access Signature	The string value of the Azure Blob shared access signature.

Configure private endpoints for key-based authentication

1. Fill out the required fields on the Details tab.
2. Select **Key** from the Authentication type dropdown menu, then fill out the **Storage Account** and **Key** fields.

3. In the Driver Properties field on the Properties tab, add **cloud.endpoint=<endpoint>** to enable the Azure Blob private endpoint for key-based authentication.

   > **Note** The **<endpoint>** value can be any Microsoft cloud endpoint. Depending on the location, Azure uses ***.windows.net** and ***.chinacloudapi.cn**, and Microsoft Entra ID uses **https://login.microsoftonline.com** and **https://login.chinacloudapi.cn**. For more information, go to the official Microsoft documentation.

   > **Example** To use the **usgovcloudapi.net** endpoint, add **cloud.endpoint=usgovcloudapi.net**.

4. Click **Submit**.

Known limitations

- There is a limitation with Azure Blob connections where a missing folder error occurs when you try to load files from Explorer.
  - A possible one-time workaround is to create the /opt/owl/previewcache folder before you load files from your Azure Blob connection.

Connecting to Azure Data Lake Storage

This section contains an overview of Azure Data Lake Storage Gen2 (ADLS).

General information

Field	Description
Data source	Azure Data Lake Storage Gen2 (ADLS)
Supported versions	N/A
Connection string	`abfss://`
Packaged?	✓ Yes

Field	Description
Certified?	✓ Yes
Supported features	
Analyze data	✓ Yes
Archive breaking records	✓ Yes
Estimate job	✓ Yes
Pushdown	✗ No
Processing capabilities	
Spark agent	✓ Yes
Yarn agent	✓ Yes

Minimum user permissions

In order for Collibra DQ to access your Azure Storage containers, you need the following user permissions:

- ROLE_ADMIN in Collibra DQ.
- Read access on your Azure Storage containers to create a basic ADLS connection.
- Read and write access to the Azure Storage containers where breaking records are archived from Collibra DQ. This is only necessary if you use the archive breaking records feature.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name used for your connection.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string value of your ADLS connection. `abfss://<file system>@<account name>.dfs.core.windows.net/<path>/file name>` **Tip** See the Connection URL elements section for more information on the syntax.
✕ No	Target Agent	Text	The Agent used to submit your DQ Job.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✕ No	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Connection URL elements

Field	Description
File scheme	The `abfss` protocol is used as the scheme identifier.
File system	The parent location that holds the files and folders. This is the same as containers in the Azure Storage Blob service.
Account name	The name given to your storage account during creation.
Path	A forward slash delimited `/` representation of the directory structure.

Field	Description
File name	The name of the individual file. This parameter is optional when you address a directory.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Field	Description
Storage Account	The Azure Storage account name.
Key	The authentication key for the storage account.
TenantId	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
ClientId	The client Id assigned when you register your application with an OAuth authorization server.
Secret	The client secret assigned when you register your application with an OAuth authorization server.

Configure private endpoints for key-based authentication

1. Fill out the required fields on the Details tab.
2. Select **Key** from the Authentication type dropdown menu, then fill out the **Storage Account** and **Key** fields.
3. In the Driver Properties field on the Properties tab, add **cloud.endpoint=<endpoint>** to enable the Azure Data Lake Storage (Gen2) private endpoint for key-based authentication.

> **Note** The **<endpoint>** value can be any Microsoft cloud endpoint. Depending on the location, Azure uses ***.windows.net** and ***.chinacloudapi.cn**, and Microsoft Entra ID uses **https://login.microsoftonline.com** and **https://login.chinacloudapi.cn**. For more information, go to the official Microsoft documentation.

> **Example** To use the **usgovcloudapi.net** endpoint, add **cloud.endpoint=usgovcloudapi.net**.

4. Click **Submit**.

Configure private endpoints for service principal-based authentication

1. Fill out the required fields on the Details tab.
2. Select **Service Principal** from the Authentication type dropdown menu, then fill out the **TenantId**, **ClientId**, and **Secret** fields.
3. In the Driver Properties field on the Properties tab, add **cloud.endpoint=usgovcloudapi.net;authority.host=gov** to enable the Azure Data Lake Storage (Gen2) private endpoint for key-based authentication.
4. Click **Submit**.

Limitations

Creating a job from an ADLS folder is not supported.

Connecting to Hadoop Distributed File System (HDFS)

This section contains an overview of Hadoop Distributed File System (HDFS).

General information

Field	Description
Data source	Hadoop Distributed File System (HDFS)

Field	Description
Supported versions	N/A
Connection string	`hdfs://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Analyze data	✓ Yes
Archive breaking records	✗ No
Estimate job	✓ Yes
Pushdown	✗ No
Processing capabilities	
Spark agent	✓ Yes
Yarn agent	✓ Yes

Minimum user permissions

In order for Collibra DQ to access your HDFS bucket, you need the following permissions.

- Read access to the path in your HDFS connection.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Do not use spaces in your connection name and only use valid characters.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string path of your HDFS connection. The path must start with `hdfs://` and point to the root bucket, not a sub-folder. **Example** `hdfs://<name node>:8020/`
✓ Yes	Target Agent	Option	The Agent used to submit your DQ Jobs.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✓ Yes	Save Credentials	Option	Select this option after you enter your connection details.
✗ No	Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true** To ensure that your remote procedures are secure within Collibra DQ, we recommend defining the following driver property: `hadoop.rpc.protection=privacy`

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Field	Description
Principal	The service principal used to let Collibra DQ access your connection.
Key	The keytab used to authorize your connection.
TGT	The Ticket Granting Ticket used to authorize your connection.

Connecting to Google Cloud Storage (GCS)

This section contains an overview of Google Cloud Storage (GCS).

General information

Field	Description
Data source	Google Cloud Storage (GCS)
Supported versions	N/A
Connection string	`gcs://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Analyze data	✓ Yes
Archive breaking records	✓ Yes
Estimate job	✓ Yes
Pushdown	✗ No
Processing capabilities	
Spark agent	✓ Yes
Yarn agent	✓ Yes

Minimum user permissions

In order for Collibra DQ to access your Cloud Storage bucket, you need the following permissions.

- Viewer permissions on your Cloud Storage bucket.
- When using the Archive Break Records feature, you need Editor and Viewer permissions on the Cloud Storage bucket location where break records will send.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Do not use spaces in your connection name and only use valid characters.

✓ Yes	Connection URL	String	The connection string path of your GCS connection. The path must start with `gcs://` and point to the root bucket, not a sub-folder.
			
			Example
			`gcs://<bucket-name>`
			
			You can optionally add a key after the bucket name.
			
			Example
			`gcs://<bucket-name>/key`

✓ Yes	Target Agent	Option	The Agent used to submit your DQ Jobs.

Required	Connection Property	Type	Value
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✓ Yes	Save Credentials	Option	Select this option after you enter your connection details.
✕ No	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Field	Description
GCS	Select this option to upload a JSON file locally that contains your GCS service account access credentials.

Field	Description
Authorization (JSON)	The JSON file that contains service account access credentials. Upload a JSON file locally that contains the following information:

```
{
  "type": "service_account",
  "project_id": "owl-hadoop-cdh",
  "private_key_id": "encodedstring",
  "private_key": "-----BEGIN PRIVATE KEY-----\en-
codedstring\n-----END PRIVATE KEY-----\n",
  "client_email": "<your-service-accoun-
t>@developer.gserviceaccount.com",
  "client_id": "your-service-account-id",
  "auth_uri": "https://ac-
counts.google.com/o/oauth2/auth",
  "token_uri": "https://oau-
th2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.-
googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.-
googleapis.com/robot/v1/metadata/x509/<your-ser-
vice-account>@developer.gserviceaccount.com"
}
```

> **Tip** This JSON file is typically available to download in GCP when you create a service account in your IAM profile.

Connecting to NetApp

This section contains details for NetApp connections.

General information

NetApp is a data storage platform that Data Quality & Observability Classic recognizes as a Remote File Connection, allowing you to access data stored in Amazon S3 buckets.

Field	Description
Data source	NetApp

Field	Description
Supported versions	N/A
Connection string	`https://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Analyze data	✓ Yes
Archive breaking records	✓ Yes
Estimate job	✓ Yes
Pushdown	✗ No
Processing capabilities	
Spark agent	✓ Yes
Yarn agent	✓ Yes

Minimum user permissions

In order for Collibra DQ to access your Amazon S3 bucket, you need the following permissions.

- ROLE_ADMIN in Collibra DQ.
- Read access on the Amazon S3 bucket where your data is stored.
- Read and write access on the Amazon S3 bucket where breaking records are archived from Collibra DQ. This is only necessary if you use the archive breaking records feature.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name of your connection. Ensure that there are no spaces in your connection name.
✗ No	Path Style Access	Option	Allows access to object stores along HTTPS paths.
✓ Yes	Connection URL	String	The connection string path of your NetApp connection. **Example** `https://<domain>.netapp.com`
✓ Yes	Region	Option	The AWS region in which the Amazon S3 bucket resides. The default region is US_EAST_1
✓ Yes	Bucket Name	String	The exact name of the Amazon S3 bucket along the URI path you are attempting to access. **Example** example_bucket
✓ Yes	Authentication Type	Option	The method to authenticate your connection. **Note** The configuration requirements are different depending on the Auth Type you select. See Authentication for more details on available authentication types.
✓ Yes	Save Credentials	Option	Select this option after you enter your connection details. We recommend selecting this option to allow credentials to be shared with users across the Collibra DQ platform.
✗ No	Target Agent	Option	The Agent used to submit your DQ Job.

Required	Connection Property	Type	Value
✕ No	Archive Breaking Records	Option	The unique name of your connection. Ensure that there are no spaces in your connection name.
✕ No	Archive Location	Option	The path to store archived breaking records when you select the Archive Breaking Records option. **Example** /folder/path/
✓ Yes	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true** To connect to a NetApp endpoint in URI format, select the **HTTPS** option, specify a **Bucket Name**, and then add one of the following property to the **Properties** tab: **s3-endpoint=netapp**
✕ No	Archive Breaking Records	Option	Select this option to automatically export CSV files containing the breaking records of DQ Job to your S3 bucket.
✕ No	Archive Location	String	The path along which break records will be archived when the Archive Break Records and HTTPS options are enabled. Specify a schema output location in your data source for break records to send. For example, **/write/FolderName** **Note** This option is only available when the HTTPS option is enabled.

Authentication

Select an authentication type from the dropdown menu. The options available in the dropdown menu are the currently supported authentication types for this data source.



Field	Description
Key	AWS security credentials that use an access key ID and secret access key combination to access S3 buckets.
Key	The access key ID for your Amazon S3 storage account.
Secret	The secret access key for your Amazon S3 storage account.
Instance Profile	The instance profile used to grant access to the EC2 instance to access your S3 bucket. Optionally select Assume Role and add an accessRole when you use Instance Profile.
Username	The username credential required for the IdP service to authenticate a user.
Password	The password credential required for the IdP service to authenticate a user.
Assume Role	This is optional for Instance Profile and Key authentication. Select this option to generate a set of temporary security credentials that you can use to access AWS resources. To assume an S3 IAM Role on EKS-based deployments of Collibra DQ, select Assume Role, then optionally enter the information in the Role to Assume field.
Role to Assume	The IAM role associated with your EC2 instance. This is optional when assuming an S3 IAM Role on EKS-based deployments of Collibra DQ.

Connect to NetApp in URI format

To connect to a NetApp endpoint in URI format, select the **HTTPS** option, specify a **Bucket Name**, and then add one of the following property to the **Properties** tab:

Connection	Property
NetApp	**s3-endpoint=netapp**

Connecting to Network File Storage (NFS)

This section contains an overview of Network File Storage (NFS).

General information

Field	Description
Data source	Network File Storage (NFS)
Supported versions	N/A
Connection string	`nfs:///`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Analyze data	✓ Yes
Archive breaking records	✕ No
Estimate job	✓ Yes
	Note Estimate job is only available for NFS connections on Standalone deployments of Collibra DQ.
Pushdown	✕ No
Processing capabilities	

Field	Description
Spark agent	✓ Yes **Note** • Spark agent is available for NFS connections on Standalone deployments of Collibra DQ. Additional configurations are required for Kubernetes and Hadoop deployments. • A Spark configuration must be added to the command line to mount the NFS directory path and run the NFS remote connections in Kubernetes. Refer to the Review page for more information on appending using the command line. The following Spark configurations are related to mounting NFS volume in Kubernetes: ◦ spark.kubernetes.driver.volumes.persistentVolumeClaim.nfs-pv-dq-dev.mount.path=/opt/owl/nfs-storage ◦ spark.kubernetes.driver.volumes.persistentVolumeClaim.nfs-pv-dq-dev.options.claimName=nfs-pvc-dq-dev ◦ spark.kubernetes.executor.volumes.persistentVolumeClaim.nfs-pv-dq-dev.mount.path=/opt/owl/nfs-storage ◦ spark.kubernetes.executor.volumes.persistentVolumeClaim.nfs-pv-dq-dev.options.claimName=nfs-pvc-dq-dev
Yarn agent	✓ Yes **Note** Yarn agent is available for NFS connections on Standalone deployments of Collibra DQ. Additional configurations are required for Kubernetes and Hadoop deployments.

Minimum user permissions

In order for Collibra DQ to access your local file system, you need the following permissions.

- Ensure that you can mount the NFS in your Collibra DQ service locally.
- Ensure that the Linux user has read permissions on the path that runs Collibra DQ services.

Recommended and required connection properties

Required	Connection Property	Type	Value
✓ Yes	Name	Text	The unique name used for your connection.
✓ Yes	Connection URL	String	The connection string value of your NFS connection. `nfs:///your/directory/path/`
✕ No	Target Agent	Text	The Agent used to submit your DQ Job.
✓ Yes	Auth Type	Option	The method to authenticate your connection. **Note** Auth Type is always `NFS`.
✕ No	Driver Properties	String	The configurable driver properties for your connection. Multiple properties must be comma delimited. For example, **abc=123,test=true**

Authentication

Auth Type must be set to **NFS**.

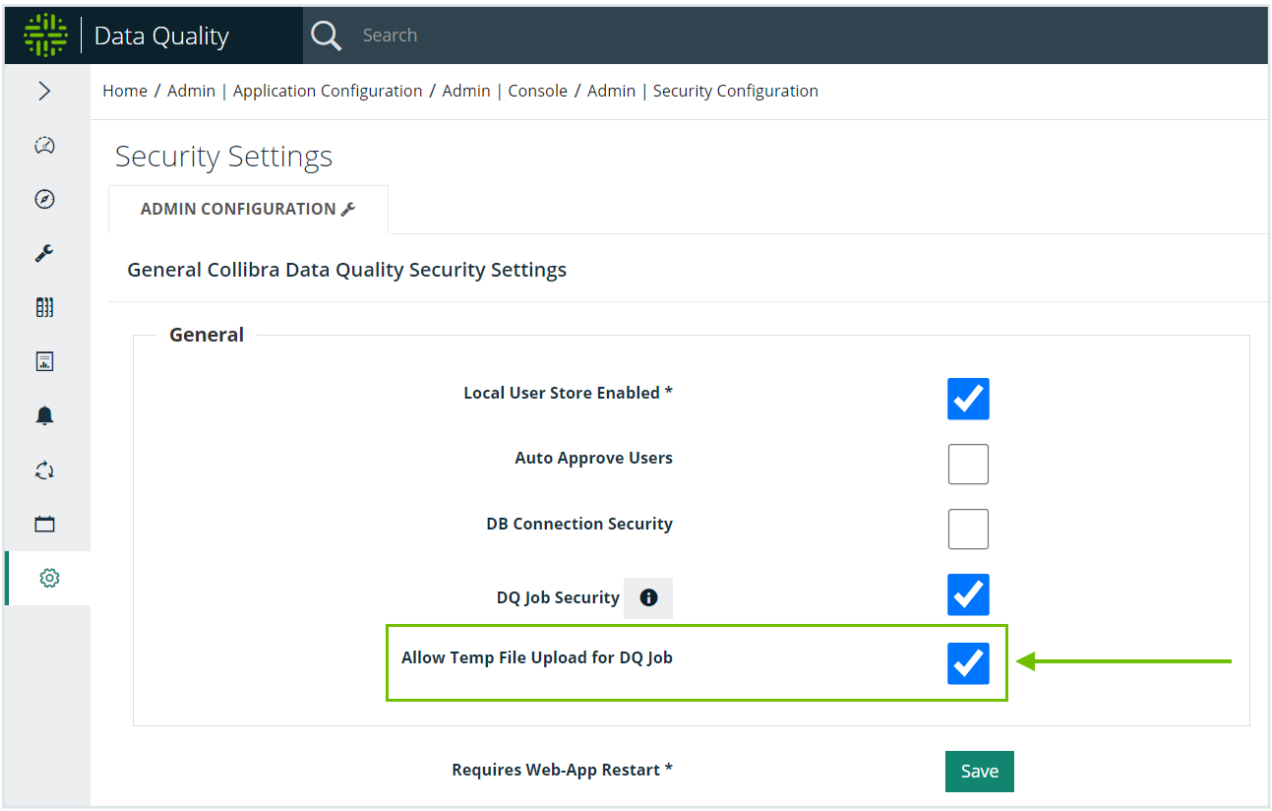
Connecting to Temp Files

Temp files allow you to upload and run a DQ job on files directly from your local machine. This is not a recommended connection for more advanced users but can be useful when you first get started in Collibra Data Quality.

Prerequisites

To connect to a temp file, you need:

- A file in a supported file format saved on your local drive.
- To verify that the Allow Temp File Upload for DQ Job checkbox under Admin Console > Security Configuration is checked.



Steps

1. In the main menu, click the **Explorer** button.
   » The Explorer expands.
2. Click **Temp Files**.
   » Temp Files expands.
3. Click **Add Temp File**.
   » Upload Temp File alert opens.
4. Click **Choose File**.
5. Select a file from your local drive.
6. Click **Open**.
   » Your file loads into the Temp Files folder.
7. In the Temp Files folder in the application, select your file.
8. Click **Create DQ Job**.

9. Verify your file information and enter the required information.
10. Click **Load File**.
    » The application automatically reads your file and opens Scope & Range.
11. Select your DQ layers.
    » You can also leave the defaults set.
12. Click **Save & Run**.
13. Enter a unique name and any additional information.
    » Temp files differ from other remote file connections insofar as they do not require an agent to run successfully. This is a legacy component.
14. Click the **Run CMD** tab.
    » This is a bypass step for temp files.
15. Click **Run**.
    » Your job is sent to the Jobs queue.

Warning All temp files are only temporarily stored in the application. At 11:59 PM EST, all temp files uploaded on a given day are automatically removed.

Known Limitations

Temp files do not run successfully on Kubernetes deployments of Collibra DQ. This is a legacy known limitation wherein container environments do not support temp files.

Unsupported data types

Data Quality & Observability Classic supports all standard data types in columns, but there are some limitations around certain data source connections and specific data types. The table below details the data types that Data Quality & Observability Classic does **not** support.

Unsupported data type	Unsupported on
ARRAY	All connections
BINARY	All connections
BINARY VARYING	Redshift Pushdown
BLOB	All connections

Unsupported data type	Unsupported on
CLOB	All connections
GEOGRAPHY	• BigQuery Pushdown • Redshift Pushdown
GEOMETRY	Redshift Pushdown
HASHMAP	All connections
HLLSKETCH	Redshift Pushdown
HYPERLOGLOG	Trino Pushdown
JSON	All connections
LIST	All connections
MAP	All connections
MONEY	PostgreSQL
NCLOB	All connections
OBJECT	All connections
P4HYPERLOGLOG	Trino Pushdown
QDIGEST	Trino Pushdown
SDO_GEOMETRY	Oracle
SETDIGEST	Trino Pushdown
SQL_VARCHAR	Redshift Pushdown
STRUCT	All connections
SUPER	Redshift Pushdown
TDIGEST	Trino Pushdown
UNKNOWN	All connections

Unsupported data type	Unsupported on
VARBINARY	Redshift Pushdown
VARBYTE	Redshift Pushdown
VARIANT	All connections

Authentication

Kerberos authentication

Kerberos is a network authentication protocol that uses secret-key cryptography to provide secure authentication over unsecured networks.

You can use Kerberos authentication for connecting to a data source, for example Hive, on Cloud Native (Kubernetes) and Hadoop deployments of Collibra DQ.

> **Note** Kerberos authentication is only supported for Kubernetes and Hadoop deployments of Collibra DQ, but not for Standalone deployments.

Authentication types

We support Kerberos password manager, keytab, password, and ticket granting ticket (TGT) for many of our supported data sources. For more information on which data sources support Kerberos authentication, see the supported data sources page.

> **Note** Ensure that you configure the krb5.conf file before you attempt to authenticate with Kerberos.

Create a krb5.conf file

Before you can authenticate connections with Kerberos, you need to create a storage space for your Kerberos configuration information. This section shows you how to create a krb5.conf

file to store your Kerberos information.

Steps

1. Open a terminal session.
2. Run the following command:

```
sudo vi /etc/krb5.conf
```

3. Ensure that you have the correct information for the following, as shown also in the example screenshot below:
    a. default_realm
    b. default_ccache_name
    c. [realms]
    d. kdc
    e. [domain_realm]

```
[logging]
 default=FILE:/var/log/krb5kdc.log

[libdefaults]
 default_realm= CW.COM
 default_ccache_name = FILE:/<root-folder>/krb5/krb5.ccache
 dns_lookup_realm = false
 dns_lookup_kdc = true
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true

[realms]
 CW.COM = {
 kdc = <admin-ip-address>
 admin_server = <admin-ip-address>
}

[domain_realm]
 .cw.com = CW.COM
 cw.com = CW.COM
~
~
~
```

4. Run the following command to add your hostname:

```
sudo vi /etc/hosts
```

5. Enter a hostname, for example `localhost`.

> **Note** Ensure that the host where the krb5.conf is configured can reach the KDC server.

Create Kerberos authentication files

In order to authenticate with Kerberos, you need to create the files with which Collibra DQ will access to authenticate your datasource connections.

Prerequisites

You have:

- A krb5.conf file to store your Kerberos configuration information.
- Admin user permissions.

Steps

1. Open a terminal session.
2. Run the following command with your own username:

```
kinit <your-username>
```

3. Enter the password of your keytab file.

> **Note** If any encryption-related messages appear, you can safely ignore them.

4. Run the following command with your file name:

```
kinit -c /tmp/<your-file-name>.keytab
```

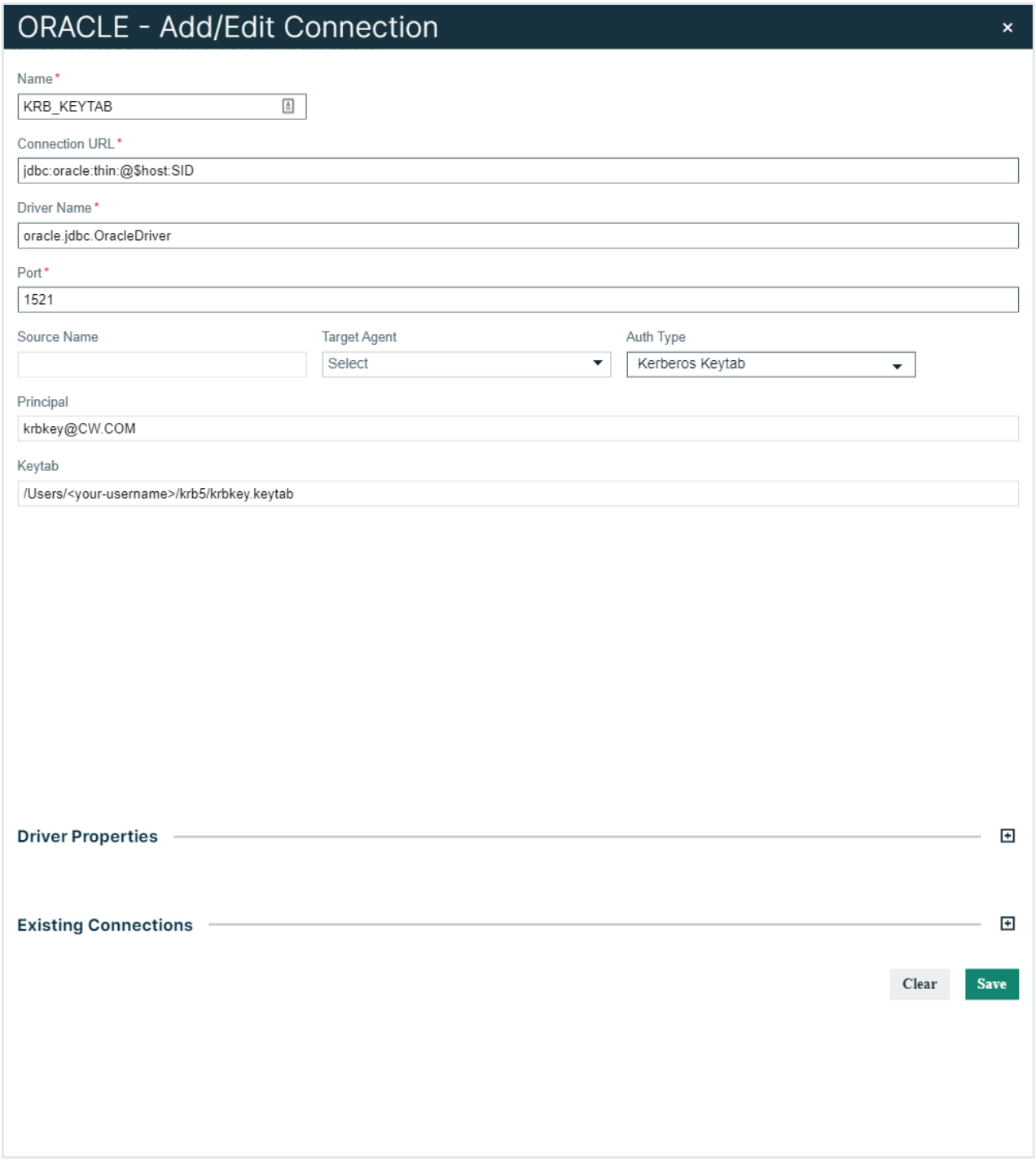
> **Note** You can define the location of your file, for example `/tmp/`.

5. Run the following command with your username to view the details of your keytab file:

```
klist <your-username>
```

View example screenshot...[1]

1



1. Open a terminal session.
2. Run the following command with your username:

```
kinit <your-username>
```

3. Enter the password of your TGT file.
4. Run the following command with your file name:

```
kinit -c /tmp/<your-file-name>.tgt
```

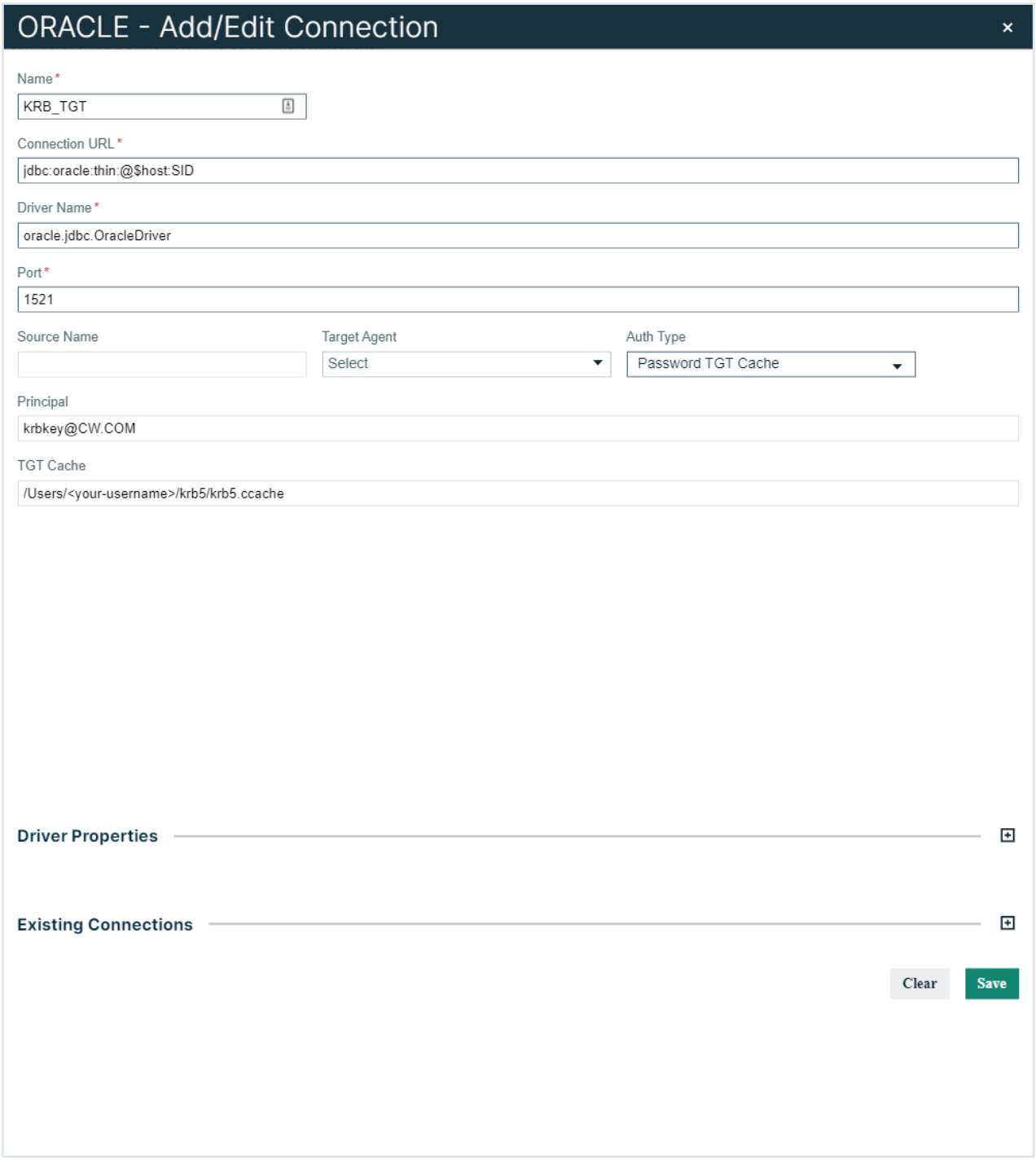
> **Note** You can define the location of your file, for example `/tmp/`.

5. Run the following command with your username to view the details of your TGT file:

```
klist <your-username>
```

View example screenshot...[1]

1



Configure AWS passwordless authentication using Amazon RDS PostgreSQL as the Metastore

This page shows you how to configure passwordless authentication for the Metastore in Amazon RDS.

AWS passwordless authentication is an optional feature that allows Data Quality & Observability Classic to connect to a database without specifying a password.

Typically, database passwords are set in the application configuration, but by leveraging the AWS passwordless option, Data Quality & Observability Classic retrieves a temporary password from AWS instead. This temporary password is most commonly referred to as a token, which is a lengthy credential containing a password and non-password information, such as how long the token is valid. When configured for the purpose of passwordless authentication, the AWS RDS PostgreSQL database accepts this token instead of a traditional password.

Data Quality & Observability Classic notes how long the token is valid and stores it in its memory. Whenever a database request is made, Data Quality & Observability Classic verifies whether the token is still valid, and if it isn't, a new token is requested via an HTTP call from Data Quality & Observability Classic to AWS.

> **Note** AWS passwordless authentication is currently only supported for EC2 Instance Profile-based authentication with an Amazon RDS Metastore for Data Quality & Observability Classic standalone and cluster-based deployments. IAM pod role-based authentication support will be available in a future release.

Prerequisites

- The Amazon RDS instance must have the IAM authentication method enabled.
- A user must be created in the Amazon RDS database and given the role **rds_iam**.
- A matching user must be created in AWS with permission to connect to Amazon RDS.

- The ec-2 instance to which Data Quality & Observability Classic is deployed must have permission to connect to Amazon RDS.

> **Tip** For more information, see the official AWS documentation on Using IAM Authentication with PostgreSQL and Creating and using an IAM policy for IAM database access .

- The Amazon RDS instance must have the IAM authentication method enabled.
- A user must be created in the Amazon RDS database and given the role **rds_iam**.
- A matching user must be created in AWS with permission to connect to Amazon RDS.
- The EKS instance to which Data Quality & Observability Classic is deployed must have permission to connect to Amazon RDS.

> **Tip** For more information, see the official AWS documentation on Using IAM Authentication with PostgreSQL and Creating and using an IAM policy for IAM database access .

Setting up AWS passwordless authentication

1. Open the **owl-env.sh** file.
2. Export the Spring profile "AWS_ENV" environment variable with the following command:
    - `export SPRING_PROFILES_ACTIVE=AWS_ENV`
3. Export the Spring data source URL environment variables with the following commands, replacing the parameterized `${RDS-url}` sections with your Amazon RDS JDBC URL:
    - `export SPRING_DATASOURCE_URL=${RDS-url}`
    - `export SPRING_AGENT_DATASOURCE_URL=${RDS-url}`

4. Export the Spring data source username environment variables with the following commands, replacing the parameterized `${RDS-username}` sections with the Amazon RDS username configured with the profile **rds_iam**:
   - `export SPRING_DATASOURCE_USERNAME=${RDS-username}`
   - `export SPRING_AGENT_DATASOURCE_USERNAME=${RDS-username}`
5. Export the Spring data source password environment variables with the following commands, replacing the parameterized `${value}` sections with a non-empty value:
   - `export SPRING_DATASOURCE_USERNAME=${value}`
   - `export SPRING_AGENT_DATASOURCE_USERNAME=${value}`

   > **Note** The password properties are not used for the connection, but Data Quality & Observability Classic requires them to be present.

6. Save and close the **owl-env.sh** file.
7. Open the **owl.properties** file.
8. Append the `authenticationPluginClassName=com.owl.common.datasource.aws.AwsPostgresAuthenticationPlugin` argument to the **spring.datasource.url** property in the configuration file, similar to the valid Spring data source URL shown in the following example:

   Example

   ```
   spring.datasource.url=jdbc:postgresql://my-rds-instance.xyz.us-east-1.rds.amazonaws.com:5432/mydatabasename?authenticationPluginClassName=com.owl.common.datasource.aws.AwsPostgresAuthenticationPlugin
   ```

9. Add the following properties to the configuration file:

```
# Replace {RDS_username} with the Amazon RDS username
configured with the profile rds_iam
spring.datasource.username={RDS_username}

# Replace {value} with a non-empty value
spring.datasource.password={value}
```

10. Save and close the **owl.properties** file.
11. Restart the Data Quality & Observability Classic Web App.

1. Deploy Data Quality & Observability Classic with the following Helm Chart variables, replacing the parameterized `${value}` sections with your AWS RDS Metastore details:

```
--set global.configMap.data.active_profile=AWS_ENV
--set global.configMap.data.metastore_url="${value}"
--set global.configMap.data.metastore_user="${value}"
--set global.configMap.data.metastore_pass="${value}"
```

2. Validate that the **dq-agent** and **dq-web** pods restarted with the appropriate configuration values after the deployment completes.

Troubleshooting

Problem	Possible solution
I cannot connect to the database. I see a PAM Authentication error in my stack-trace.	Generally, this indicates an issue with the AWS configuration. Confirm that you can connect to the database using AWS authentication via the AWS cli or a different method outside of Data Quality & Observability Classic. You can use AWS tools to generate a token and connect. Follow the instructions in the official AWS documentation to verify database connectivity . **Note** Depending on your database configuration you may need to perform these steps from within an EC2 instance that has appropriate access to your RDS IAM-enabled database.
I have verified that I can connect to my database via IAM authen-tication, but I still see authen-tication errors in my stack trace.	Closely analyze your IAM policy from the resource running Data Quality & Observability Classic to the RDS resource. See the official AWS documentation to using an IAM policy for IAM database access .
After one hour, I notice that the **Connections** sec-tion of Data Qual-ity & Observability Classic is missing connections. When I have developer tools open in my web browser, I see the failed connections in the network tab.	This is a known issue that affects certain configurations. It will be fixed in a future release. As of today, the only workaround is to restart your application.

Configure Azure passwordless authentication using Azure PostgreSQL as the Metastore

Azure's passwordless authentication balances security and user convenience to provide a robust solution for enhancing the security and usability of PostgreSQL database access. Passwordless authentication allows your organization to minimize the risks associated with traditional password management and offer users a more seamless experience. This page shows you how to configure passwordless authentication for the Metastore in Azure and outlines the various passwordless authentication methods that we support.

Prerequisites

- Configure your Azure database for PostgreSQL to enable passwordless authentication.
- Export the environment variable `SPRING_PROFILES_ACTIVE=AZURE_ENV` in the **owl-env.sh** file.

- Configure your Azure database for PostgreSQL to enable passwordless authentication.
- Pass the following Helm Chart variables:

```
--set global.configMap.data.active_profile={AZURE_ENV} \
--set
global.serviceAccount.annotations."azure\.workload\.ident
ity/client-id"=<client-id> \
```

Setting up Azure passwordless authentication

There are 3 methods you can use to authenticate Azure databases.

- Environmental credentials
- Managed identity credentials

- Azure CLI credentials

Environmental credentials

This method checks for credentials provided through environment variables. Export the following environment variables in the owl-env.sh file to enable this authentication method.

Variable	Description
AZURE_ TENANT_ID	The Microsoft Entra tenant (directory) ID.
AZURE_ CLIENT_ID	The client (application) ID of an App Registration in the tenant.
AZURE_ CLIENT_ SECRET	A client secret that was generated for the App Registration.

Managed identity credentials

Additional setup is not required to enable managed identity. For more information, see the official Azure documentation to configure managed identities to enable authentication for PostgreSQL Metastores .

Azure CLI credentials

> **Note** While this authentication method is supported, we recommend the environmental credentials or managed identity credentials options instead.

This method requires the Azure CLI to be enabled on the Azure virtual machine and the principal authenticated with the PostgreSQL database. When this is set up, Data Quality & Observability Classic executes the `az account get-access-token --output json` to retrieve the token to authenticate with PostgreSQL.

Workload identity credentials

1. Enable the Azure cluster for workload identity.

```
az aks update --resource-group "${RESOURCE_GROUP}" --name
"${CLUSTER_NAME}" --enable-oidc-issuer --enable-workload-
identity
```

2. Enable federation for the Spark service account used by Data Quality & Observability Classic.

```
az identity federated-credential create --name
${FEDERATED_IDENTITY_CREDENTIAL_NAME} --identity-name
"${USER_ASSIGNED_IDENTITY_NAME}" --resource-group
"${RESOURCE_GROUP}" --issuer "${AKS_OIDC_ISSUER}" --
subject system:serviceaccount:"${SERVICE_ACCOUNT_
NAMESPACE}":"${SERVICE_ACCOUNT_NAME}" --audience
api://AzureADTokenExchange
```

3. Deploy Data Quality & Observability Classic with the following Helm variables where `managed-identity-id` is the name of the managed identity and `client-id` is the client ID of the managed identity.

```
--set global.configMap.data.metastore_user=<managed-
identity-id> \
--set global.configMap.data.metastore_pass=blank \
--set global.configMap.data.active_profile={AZURE_ENV} \
--set
global.serviceAccount.annotations."azure\.workload\.ident
ity/client-id"=<client-id> \
```

4. Mount the Spark driver and executor pod templates with metadata enabled.

```
apiVersion: v1
Kind: Pod
metadata:
  labels:
    azure.workload.identity/use: "true"
spec:
  securityContext:
    runAsUser: 1000
    runAsGroup: 1000
    fsGroup: 1000
```

5. Add the following to the Freeform (Appended) field on the Agent Configuration template to use the custom executor and driver pod templates.

```
-conf
spark.kubernetes.driver.podTemplateFile=local:///opt/owl/
config/k8s-driver-
template.yml,spark.kubernetes.executor.podTemplateFile=lo
cal:///opt/owl/config/k8s-executor-template.yml
```

For more information about configuring your Azure cluster with workload identity, see the official Azure documentation .

Connect to a Google Cloud SQL Instance

This section shows you how to connect Collibra DQ to a Google Cloud (GCP) SQL instance with SSL enabled.

Steps

1. Create a new client certificate from CloudSQL.
2. Download the `client-cert.pem`, `client-key.pem`, `server-ca.pem` files from the Google Cloud Console to a deployment directory such as `/opt/owl/certs/`.
3. In that directory, run the following command:

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in client-
key.pem -out client-key.pk8  -nocrypt
```

> **Note** This command should not require a password.

4. In `owl.properties` and `owl-env.sh`, change your JDBC URL to the following:

```
jdb-
c:post-
gresql://$HOST:5432/postgres?sslcert=/opt/owl/certs/client-
cert.pem&sslkey=/opt/owl/certs/client-
key.pk8&sslrootcert=/opt/owl/certs/server-ca.pem
```

Configure Spark for a password manager

To enable DQ Jobs to access password manager scripts, follow these steps:

1. Create a kubernetes secret to hold the password manager script. For example:

```
apiVersion: v1
kind: Secret
metadata:
  name: dq-password-manager-script
data:
  file-name.sh: IA==
```

   In the example above, content in file-name.sh is base64.

2. Create a kubernetes secret to hold the spark pod templates. For example:

```
apiVersion: v1
kind: Secret
metadata:
  name: dq-spark-pod-templates
data:
  k8s-driver-template.yml: IA==
  k8s-executor-template.yml: IA==
```

   In the example above, the keys are base64.

An example of the Spark pod template is below:

```
bash-5.1$ cat k8s-driver-template.yml
apiVersion: v1
Kind: Pod
spec:
  securityContext:
    runAsUser: 1000
    runAsGroup: 1000
    fsGroup: 1000
  volumes:
   - name: dq-password-manager-script
     secret:
       secretName: dq-password-manager-script
       defaultMode: 493
  containers:
   - volumeMounts:
      - name: dq-password-manager-script
        mountPath: /tmp/dq-password-manager-script
```

3. Update the values in the Helm script to attach the password manager secret to the DQ Web pod and attach the Spark pod templates to the DQ Agent pod. For example:

```
web:
  extraStorage:
    volumes:
      - name: dq-password-manager-script
        secret:
          secretName: dq-password-manager-script
          defaultMode: 493
    volumeMounts:
      - name: dq-password-manager-script
        mountPath: /tmp/dq-password-manager-script

agent:
  extraStorage:
    volumes:
      - name: dq-spark-pod-templates
        secret:
          secretName: dq-spark-pod-templates
          defaultMode: 493
    volumeMounts:
      - name: dq-spark-pod-templates
        mountPath: /opt/owl/dq-spark-pod-templates
```

4. Create a DQ connection which uses the password manager script located at:

```
/tmp/dq-password-manager-script/file-name.sh
```

5. Update the Run command of the DQ Job with the following `conf-` command.

```
-conf spark.kuber-
netes.executor.podTemplateFile=local:///opt/owl/dq-spark-
pod-templates/k8s-executor-tem-
plate.yml,spark.executorEnv.ALLOWED_LOCAL_PATHS=/tmp/dq-pass-
word-manager-
script,s-
park.kubernetes.driver.podTemplateFile=local:///opt/owl/dq-
spark-pod-templates/k8s-driver-
template.yml,spark.kubernetes.driverEnv.ALLOWED_LOCAL_PATHS-
S=/tmp/dq-password-manager-script
```

> **Tip** To set this configuration globally, open the Agent Configuration> Edit Agent page and add this command to the **Free form (Appended)** field. For more information, go to Set up a DQ agent.

Core Concepts

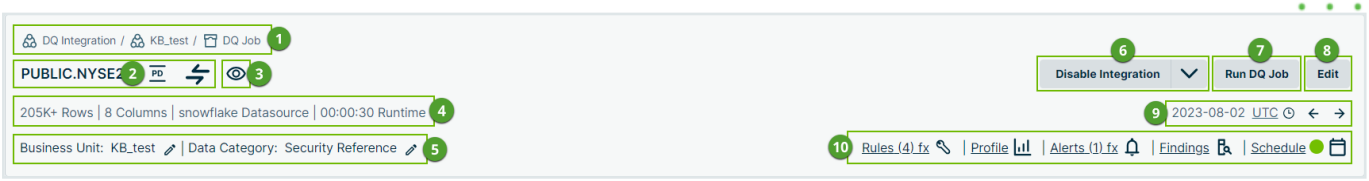
Navigation

Exploring the Metadata Bar

The Metadata Bar is a dataset anchor that empowers you with the ability to access crucial data quality components related to your dataset, such as its Findings, Profile, and associated rules and alerts. With the ability to access the dataset components you use most, you can quickly unlock the core value of Collibra DQ most powerful features.

The Metadata Bar is available at the top of the following dataset-level pages:

- Findings
- Profile
- Dataset Rules
- Alert Builder



No.	Component	Description
1	Integration breadcrumbs	When an integration with Collibra Platform is enabled, the breadcrumbs show the community hierarchy to which the integration is mapped. In order, the breadcrumbs show the Collibra Platform Community, Sub-Community, and Domain in Community / Sub-Community / Domain format. You can click any of the breadcrumbs to open the corresponding page in Collibra Platform.
2	Dataset details	Displays the name of your current dataset and, when applicable, the PD icon denotes that it is a Pushdown dataset. You can also click ⇄ to open the Select Dataset modal where you can switch to a different dataset by either searching for one in the search field or selecting one from the App Recent Runs list.
3	Dataset Overview	Opens the Dataset Overview modal where you can explore important details about your dataset, such as the type of datasource, and number of columns and rows. You can also use the query compiler to create SQL rules based on the discoveries you make. **Note** Either ROLE_ADMIN or ROLE_VIEW_DATA is required to use Dataset Overview.
4	Dataset details	Displays the number of rows and columns in your dataset, the name of the datasource, and the runtime of the most recent job run.
5	Business Unit and Data Category	Business Unit lets you assign your dataset to a business unit. **Tip** Best practice is to assign a business unit to a dataset when integrating Collibra DQ metadata in Collibra Platform Data Category lets you assign your dataset to a data category.
6	Integration options	Enable, disable, or reset an integration within Collibra Platform. **Note** ROLE_ADMIN and ROLE_DATASET_MANAGER are required to use this.
7	Run DQ Job	Submits your job for processing. When a job runs, you can find it on the Jobs page.

No.	Component	Description
8	Edit	Opens Explorer so you can edit your dataset. **Note** When you edit a dataset from the Metadata Bar, the dataset definition is updated in the Metastore and shown in all relevant locations in the application, such as on the Job tab.
9	Run Date	Displays the run date stamp and timezone. You can also click the timezone to update it and click the arrows to switch between the previous or next day's run.
10	Dataset-level pages	Click the links to open the following pages associated with your dataset: • Rules opens Dataset Rules where you can view or configure rules associated with your dataset. • Profile opens the Profile page of your dataset. • Alerts opens the Alert Builder to view or configure alerts associated with your dataset. • Findings opens the Findings page associated with the selected Run Date of your dataset. • Schedule opens the Edit Schedule modal where you can configure scheduled runs of your dataset.

Dataset Overview

Dataset Overview allows you to execute SQL queries, instantly view results, and conduct initial exploration of new datasets. Additionally, you can directly create and add rules to your jobs. With the power to write SQL to query your dataset, you can accelerate the data discovery process and extrapolate important insights in real time.

From the Metadata Bar on any dataset-level page, click the  to open Dataset Overview.

The dataset-level pages where the Metadata Bar is available include:

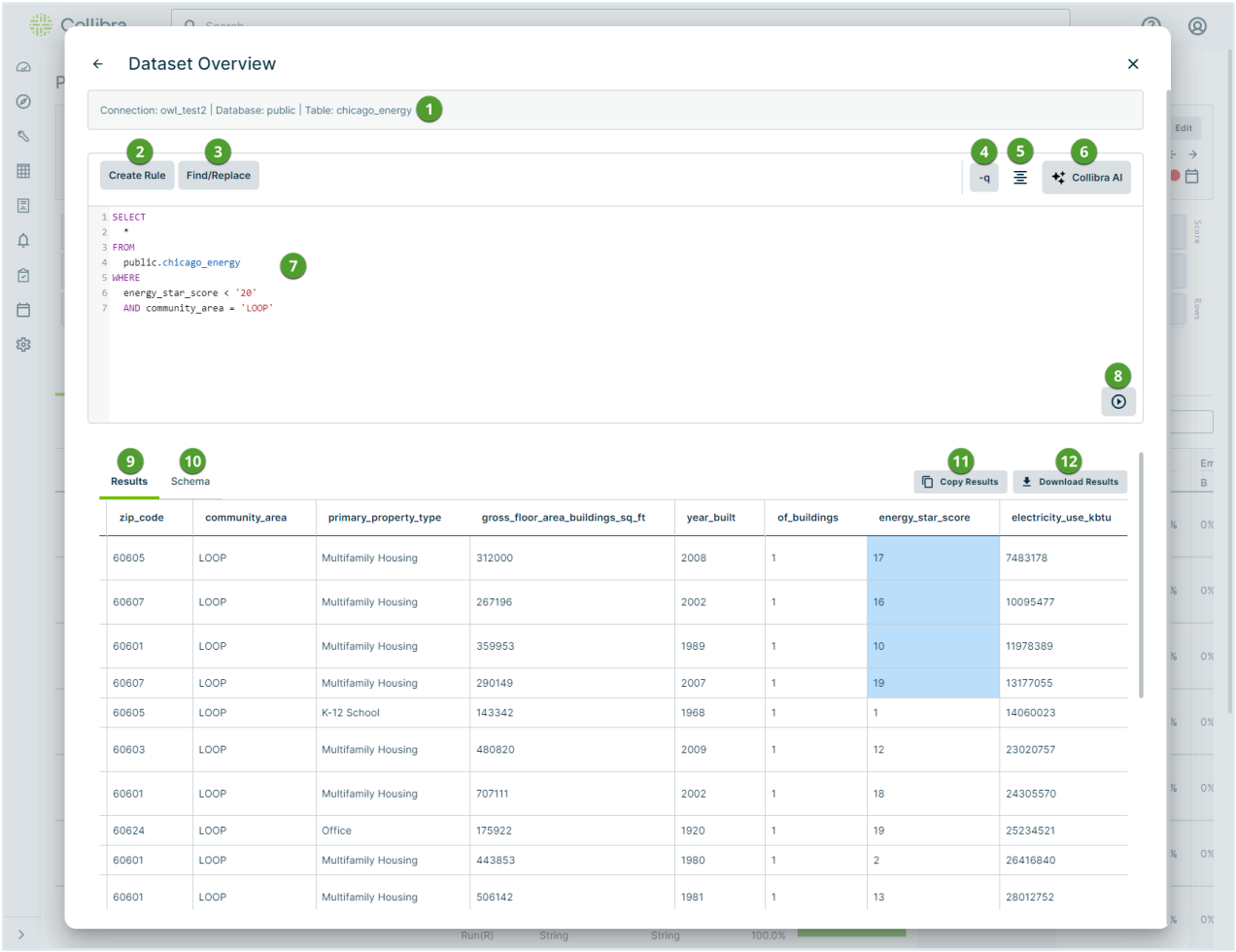
- Explorer
- Profile
- Findings
- Alert Builder
- Dataset Rules

Prerequisites

You have:

- ROLE_ADMIN, ROLE_VIEW_DATA, or ROLE_CONNECTION_MANAGER assigned to your user account.
- Access to the connection upon which the dataset was created.

About Dataset Overview



No.	Component	Description
1	Dataset information	Displays the name of the data source connection, database, and table from which your data originates.

No.	Component	Description
2	Create Rule	Lets you create a rule based on the query criteria in the editor below it. When you click Create Rule, you can give your rule a memorable name and save it to your dataset. You can find your new rule on the Dataset Rules of your dataset. **Note** The SQL type of rules created in Pullup mode depend on the rule type. For instance, if your rule type is Freeform SQL, the SQL type is Spark SQL, whereas a Native SQL rule saves as the SQL specific to your datasource. Likewise, Pushdown datasets always use SQL native to your datasource.
3	Find/Replace	Finds and replaces any string value in the SQL editor. For example, if your query in the SQL editor is `select * from public.chicago_energy WHERE energy_star_score < '20' AND community_area = 'LOOP'`, but you want to update your query by replacing `'community_area'` with `'community_name'`, click **Find/Replace** and in the Find and Replace with input fields, enter *community_area* and *community_name*, respectively. Click **Replace** after you update the input fields to update the SQL editor.
4	-q	Loads the contents of the dataset source query into the SQL editor. For example, if the most recent run of your dataset used the query `SELECT * FROM example.dataset limit 100`, when you click **-q**, that exact query will load into the SQL editor, replacing the default select * statement that loads when you open Dataset Overview. This can help you narrow the scope of the data returned, allowing you to more efficiently access the core of the data and insights from your most recent DQ job runs.
5	Pretty print	Formats your SQL query in a more readable layout.
6	Collibra AI	Opens SQL Assistant for Data Quality (powered by Collibra AI) to allow you to automate SQL rule writing and troubleshooting to help you accelerate the discovery, curation, and visualization of your data. **Note** ROLE_GENAI_USER and/or ROLE_ADMIN are required to use SQL Assistant for Data Quality.

No.	Component	Description
7	SQL editor	The SQL editor lets you explore your dataset based on the data points and criteria specific to your particular use case. When you update your SQL and then run the query, the overview table displays the results that meet the conditions of your query.
8	Run	Runs the SQL in the editor and updates the overview table. **Important** If you do not have connection access and ROLE_ADMIN, ROLE_VIEW_DATA, or ROLE_CONNECTION_MANAGER assigned to your user account, you cannot run queries in Dataset Overview.
9	Results	Contains an overview of the columns and cell values in your dataset in table format. You can click individual cells or click and drag to select a range of cells to copy to the clipboard. **Tip** Click  near the bottom right of the SQL editor to populate the overview table. The default query `select * from [table]` populates an overview of the entire table.
10	Schema	Contains a complete list of the columns in your dataset and their data types.
11	Copy Results	Copies the results of the query as shown in the overview table to the clipboard in comma delimited format. Choose to copy the first 5, 10, 20, or all rows in the overview table and then paste in either the SQL editor or elsewhere, such as a notepad. When you copy rows, the column headers also copy in comma delimited format.
12	Download Results	Downloads a CSV file containing the results of queries you run in the SQL editor.

Using the SQL editor

The SQL editor helps you quickly navigate and extract the information you need from your dataset.

1. There are two options to get started:
    a. Enter your SQL query in the SQL editor.
    OR
    b. Click Collibra AI and complete the steps in the modal.
2. Click  near the bottom right of the SQL editor to populate the overview table with the results of your query.
3. Repeat as necessary or optionally create a rule around your query.

Creating a rule

After you run a query with the SQL editor, you can save it as a rule. When you discover important or otherwise relevant data points from the results of your query, you can save the SQL query as a dataset rule.

1. Follow the steps outlined in Using the SQL editor.
2. Click **Create Rule**.
3. Give your rule a name in the **Name** input field.
4. Click **Save**.
5. You can access your newly created rule on the Dataset Rules page of your dataset.

Known limitations

When you name a rule that matches an existing rule name, the previous rule is replaced by the new one.

SQL Assistant for Data Quality

SQL Assistant for Data Quality automates SQL rule writing and troubleshooting to help you accelerate the discovery, curation, and visualization of your data.

About SQL Assistant for Data Quality

SQL Assistant for Data Quality allows you to automate SQL rule writing and troubleshooting to help you accelerate the discovery, curation, and visualization of your data. By leveraging the

SQL query generation capabilities of SQL Assistant for Data Quality, both advanced and beginner SQL users can quickly discover key data points and insights, and then convert them into rules. Furthermore, the troubleshooting capabilities of SQL Assistant for Data Quality allow you to elegantly correct syntactical mistakes, resolving the once tedious SQL syntax reviews of the past.

You can use SQL Assistant for Data Quality to assist with SQL rule writing and troubleshooting on the following pages:

- Dataset Overview
- Rule Workbench

> **Tip** This feature is powered by Collibra AI. For information on how we leverage AI in our products, including how we handle data used for training and input, please visit the Collibra Trust Site.

Prerequisites

There are two different sets of prerequisites, depending on whether you use the platform path with a Collibra Platform integration or the preview path with a Collibra-provided encrypted service account key.

To use SQL Assistant for Data Quality **with** a Collibra Platform integration, you need to:

- Have an active integration with Collibra Platform configured on the Admin Console > Integrations screen.

  > **Note** Only the Credentials step of the Integration Setup is required. You do not need to map connections, tenants, dimensions, or layers to use SQL Assistant for Data Quality.

- Set AI_PLATFORM_PATH to `TRUE` (default) on the Admin Console > Application Configuration Settings screen.
- Set AI_TENANT to `TRUE` on the Admin Console > Application Configuration Settings screen.

- Update the `gai_proxy_endpoint` in the owl-env.sh or Helm Chart, depending on your installation type.
  Steps to update the proxy endpoint

  > **Important**
  > You may need to safelist the proxy endpoint if the security settings of your installation do not allow outbound internet requests.
  >
  > While this is unlikely if you have an active integration with Collibra Platform set up, you may also need to safelist your Collibra Platform URL to allow outbound requests to it.

  Steps for Spark Standalone and Hadoop installations

  1. Export the environment variable `export gai_proxy_endpoint=https://kong-prod-gcp-ue1-1.collibra.com` in the **owl-env.sh** file.

     > **Note** The default value for the gai_proxy_endpoint is https://kong-prod-gcp-ue1-1.collibra.com. To change this value to a different cloud region, contact Collibra Support.

  2. Save and close the owl-env.sh file.
  3. Restart the DQ web application and DQ agent services.

  Steps for Kubernetes (cloud native) installations

  1. Set the following value in the Helm Chart provided by Collibra.

     ```
     --set gai_proxy_endpoint: "https://kong-prod-gcp-ue1-1.collibra.com"
     ```

     > **Note** The default value for the gai_proxy_endpoint is https://kong-prod-gcp-ue1-1.collibra.com. To change this value to a different cloud region, contact Collibra Support.

2. If you are updating an existing deployment of Data Quality & Observability Classic, restart the DQ web and agent pods to complete the update.

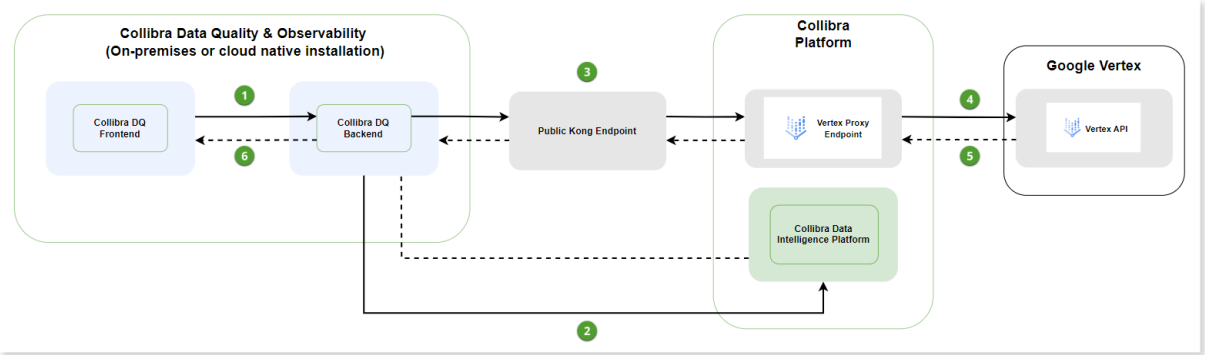
> **Important** On April 9, 2025, Google will deprecate the Vertex text-bison AI model, which the preview path option uses. To continue using SQL Assistant for Data Quality, you must switch to the platform path, which requires an integration with Collibra Platform.

When using SQL Assistant for Data Quality **without** a Collibra Platform integration, you have:

- AI_PLATFORM_PATH set to `FALSE` on the Admin Console > Application Configuration Settings screen.
- AI_TENANT set to `TRUE` on the Admin Console > Application Configuration Settings screen.
- An encrypted service account key provided by Collibra.

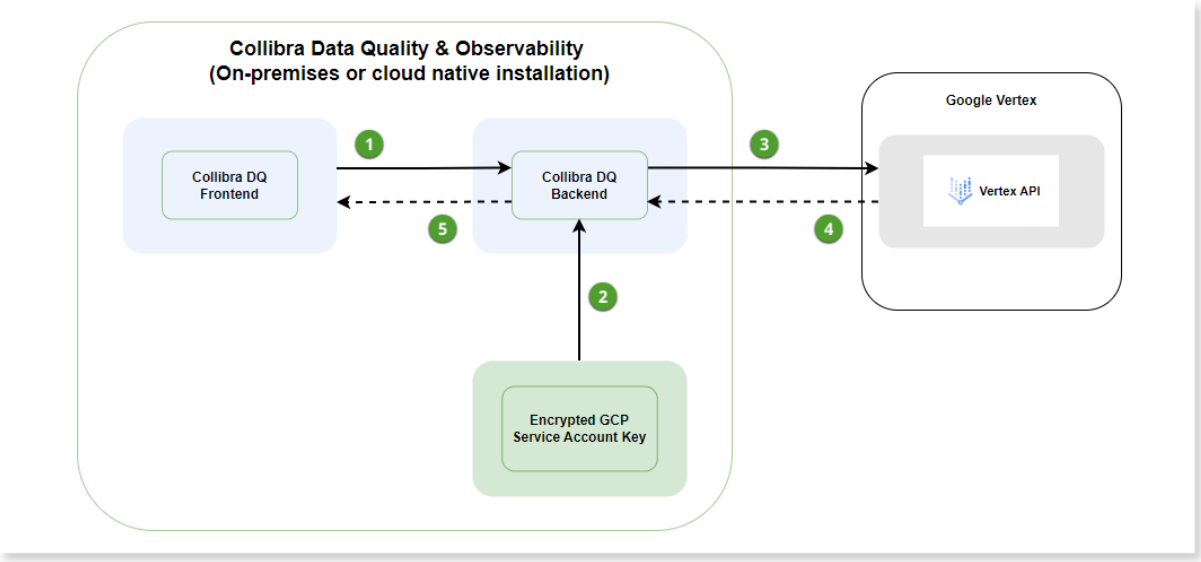
How does SQL Assistant for Data Quality work?

The architecture of SQL Assistant for Data Quality differs depending on whether you use the platform path with a Collibra Platform integration or the preview path with a Collibra-provided encrypted service account key.



1. A user initiates a Collibra AI prompt request from Dataset Overview or Rule Workbench in the Collibra DQ UI.
2. Collibra DQ registers the user and retrieves an OAuth key from their Collibra Platform instance.
3. The OAuth token is used to authenticate the request with the Public Kong Endpoint. Once authenticated, the request is redirected to the Vertex Proxy Endpoint in the Collibra Platform.
4. The Collibra Platform makes a request to the Google Vertex API.
5. Google Vertex AI returns a response to Collibra DQ, where it is logged in the Generative AI Audit Trail.
6. The response routes through Collibra DQ, where its contents display in the UI.

Important On April 9, 2025, Google will deprecate the Vertex text-bison AI model, which the preview path option uses. To continue using SQL Assistant for Data Quality, you must switch to the platform path, which requires an integration with Collibra Platform.

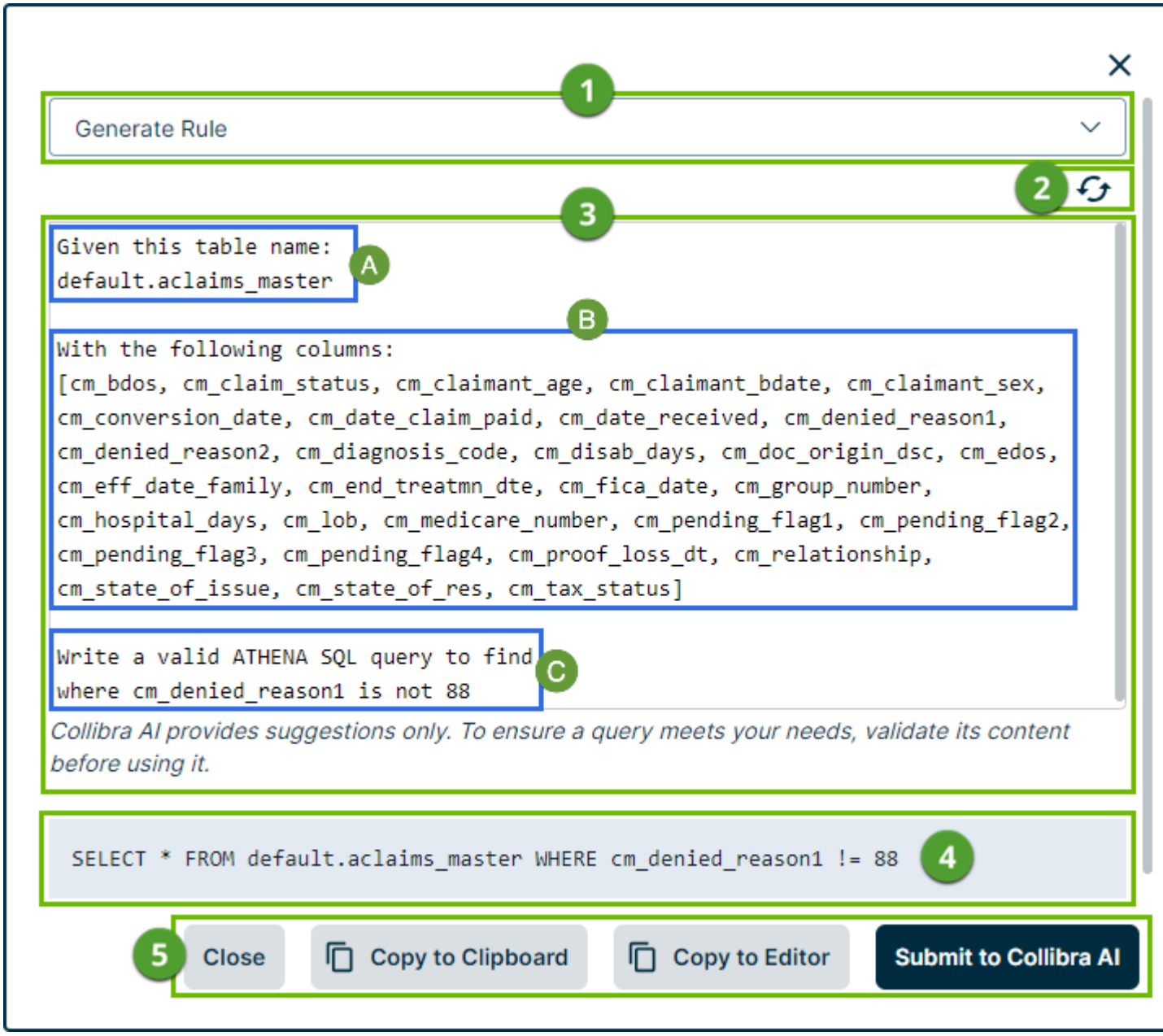


1. A user initiates a Collibra AI prompt request from Dataset Overview or Rule Workbench in the Collibra DQ UI.
2. The request routes through Collibra DQ where it decrypts the service account key.

> **Note** The encrypted service account key is specific to each customer.

3. Collibra DQ sends the request to Google Vertex AI using the decrypted key.
4. Google Vertex AI returns a response to Collibra DQ, where it is logged in the Generative AI Audit Trail.
5. The response routes through Collibra DQ, where its contents display in the UI.

Overview of the SQL Assistant for Data Quality components



No.	Component	Description
1	Collibra AI prompts	Click the dropdown menu and select a prompt. • **Generate Rule** creates SQL suggestions based on your plain-text input in the request prompt. • **Troubleshoot Rule** fixes and rewrites queries when there are issues with your SQL. • **Categorical** provides a suggested SQL check to uncover any categorical outliers in the column you specify. • **Dupe** provides a suggested SQL check to uncover any duplicate values in the column you specify. • **Record** provides a suggested SQL check to uncover any values that appear on a previous day but not the next. • **Pattern** provides a suggested SQL check to uncover infrequent combinations that appear less than 5 percent of the time in the columns you specify. • **Frequency Distribution** provides a SQL check to uncover the frequency distribution of all values within a column.
2	Reset prompt	Click ↻ to reset your request prompt.

<table>
<tr><th>No.</th><th>Component</th><th>Description</th></tr>
<tr><td>3</td><td>Request prompt</td><td>An input field where you can view dataset metadata and enter a prompt for SQL Assistant for Data Quality in plain-text format. There are three main sections in the request prompt:

<table><tr><th>Section</th><th>Description</th></tr><tr><td>A</td><td>The name of the table. This is typically one of the details before the <code>WHERE</code> clause in a SQL query.</td></tr><tr><td>B</td><td>The columns in the table that are available to query.</td></tr><tr><td>C</td><td>The input field where you can enter a prompt for SQL Assistant for Data Quality to generate a SQL suggestion.

Click the empty space at the bottom of the input field to place your cursor in the correct request prompt location.

Tip When many columns exist in your dataset, you may need to scroll to the bottom of the request prompt to enter your desired query criteria.</td></tr></table></td></tr>
<tr><td>4</td><td>Results</td><td>The results that SQL Assistant for Data Quality returns based on the contents of your prompt.</td></tr>
<tr><td>5</td><td>Action buttons</td><td>• Click Submit to Collibra AI to generate a SQL rule suggestion.
• Click Copy to Editor to automatically apply the suggested query to the SQL editor.
• Click Copy to Clipboard to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
• Click Close to exit the SQL assistant without copying the suggested query.</td></tr>
</table>

Create a SQL query with SQL Assistant for Data Quality

SQL Assistant for Data Quality uses Collibra AI to automatically write and troubleshoot SQL queries. When working on a Pullup dataset, the type of SQL that SQL Assistant for Data

Quality generates depends on the rule type. For instance, if your rule type is Freeform SQL, the SQL generated by SQL Assistant for Data Quality is Spark SQL, whereas a Native SQL rule returns the SQL specific to the datasource. Likewise, Pushdown datasets always return SQL native to the job source.

The Prompts are organized in two sections: **Basic** and **Advanced**.

Basic options for SQL query suggestions provided by SQL Assistant for Data Quality include:

Option	Description
Generate Rule	Collibra AI suggests a Spark SQL query to check where in your table, column, or view your input criteria are met.
Troubleshoot Rule	When a SQL syntax error occurs, you can use SQL Assistant for Data Quality to automatically troubleshoot and fix the error.

Advanced options for SQL query suggestions provided by SQL Assistant for Data Quality include:

Option	Description
Categorical	Collibra AI suggests a SQL query to detect categorical outliers.
Dupe	Collibra AI suggests a SQL query to detect duplicate values.
Record	Collibra AI suggests a SQL query to check for values that appear on a previous day but not for the next day.
Pattern	Collibra AI suggests a SQL query to check for infrequent combinations that appear less than 5 percent of the time in the columns you specify.
Frequency Distribution	Collibra AI suggests a SQL query to check for the frequency distribution of all values within a column.

Requirements and permissions

The architecture of SQL Assistant for Data Quality differs depending on whether you use the platform path with a Collibra Platform integration or the preview path with a Collibra-provided encrypted service account key.

To use SQL Assistant for Data Quality **with** a Collibra Platform integration, you need to:

- Have an active integration with Collibra Platform configured on the Admin Console > Integrations screen.

  > **Note** Only the Credentials step of the Integration Setup is required. You do not need to map connections, tenants, dimensions, or layers to use SQL Assistant for Data Quality.

- Set AI_PLATFORM_PATH to `TRUE` (default) on the Admin Console > Application Configuration Settings screen.
- Set AI_TENANT to `TRUE` on the Admin Console > Application Configuration Settings screen.
- Update the `gai_proxy_endpoint` in the owl-env.sh or Helm Chart, depending on your installation type.
  Steps to update the proxy endpoint

  > **Important**
  > You may need to safelist the proxy endpoint if the security settings of your installation do not allow outbound internet requests.
  >
  > While this is unlikely if you have an active integration with Collibra Platform set up, you may also need to safelist your Collibra Platform URL to allow outbound requests to it.

  ### Steps for Spark Standalone and Hadoop installations

  1. Export the environment variable `export gai_proxy_endpoint=https://kong-prod-gcp-ue1-1.collibra.com` in the **owl-env.sh** file.

     > **Note** The default value for the gai_proxy_endpoint is https://kong-prod-gcp-ue1-1.collibra.com. To change this value to a different cloud region, contact Collibra Support.

  2. Save and close the owl-env.sh file.
  3. Restart the DQ web application and DQ agent services.

Steps for Kubernetes (cloud native) installations

1. Set the following value in the Helm Chart provided by Collibra.

```
--set gai_proxy_endpoint: "https://kong-prod-gcp-ue1-1.collibra.com"
```

> **Note** The default value for the gai_proxy_endpoint is https://kong-prod-gcp-ue1-1.collibra.com. To change this value to a different cloud region, contact Collibra Support.

2. If you are updating an existing deployment of Data Quality & Observability Classic, restart the DQ web and agent pods to complete the update.

> **Important** On April 9, 2025, Google will deprecate the Vertex text-bison AI model, which the preview path option uses. To continue using SQL Assistant for Data Quality, you must switch to the platform path, which requires an integration with Collibra Platform.

When using SQL Assistant for Data Quality **without** a Collibra Platform integration, you have:

- AI_PLATFORM_PATH set to `FALSE` on the Admin Console > Application Configuration Settings screen.
- AI_TENANT set to `TRUE` on the Admin Console > Application Configuration Settings screen.
- An encrypted service account key provided by Collibra.

Steps

Basic options

1. From the top of the Explorer page or from the Metadata Bar at the top of any dataset-level page, such as Findings, Profile, Dataset Rules, and Alert Builder, click the  to open Dataset Overview:
   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
3. From the Prompt dropdown menu, select Generate Rule.

   > **Note** Generate Rule is the default option.

4. In the query input field, enter a plain language prompt.

   > **Tip** When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

   Examples
   - "high is more than 50"
   - "find all of the unique values in Column1 and show how many records exist for each"
   - "state is not Michigan, California, or Illinois"
   - "compare the 2 dates 2018-01-13 to 2018-01-14 - write 2 separate queries for each date, using trade date - write as a common table expression for tables a and b - join using symbol column - use a join to identify which values are in 2018-01-14 that are not in 2018-01-13 - the goal is to find values that are missing from the previous day"

5. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.
6. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.

b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL rule suggestion.

1. From the top of the Explorer page or from the Metadata Bar at the top of any dataset-level page, such as Findings, Profile, Dataset Rules, and Alert Builder, click the to open Dataset Overview:
   » The Dataset Overview modal appears.
2. In the SQL editor, enter an invalid SQL query that returns a syntax error.

   Example `SELECT * fomr public.nyse2 were onpe > 4.75`

3. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
4. From the Prompt dropdown menu, select **Troubleshoot Rule**.
   » The SQL Assistant for Data Quality tool shows the prompt to rewrite and fix your SQL query, an overview of the exception message, the SQL query that caused the error, the table and column names in your query, and helpful troubleshooting tips.

   Note You can optionally click the above and to the right of the overview text to manually edit the troubleshooting prompt.

5. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality troubleshoots your SQL query and displays the suggested query when it completes.
6. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
   b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.

c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL rule suggestion.

Advanced options

1. From the top of the Explorer page or from the Metadata Bar at the top of any dataset-level page, such as Findings, Profile, Dataset Rules, and Alert Builder, click the  to open Dataset Overview:
   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
3. From the Prompt dropdown menu, select **Categorical**.
   » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
4. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

> **Example** In this example, the default placeholder column "sale_state" will be replaced with the user-defined "cm_claimant_age" in the categorical outlier SQL suggestion.
>
> ```
> Replace the column sale_state with the columns named:
> cm_claimant_age
> ```

> **Tip** When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

5. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

   Example

   ```
   SELECT cm_claimant_age, COUNT(*) AS count, (COUNT(*) / (SELECT COUNT(*) FROM defau
   lt.aclaims_master)) * 100 AS percentage FROM default.aclaims_master GROUP BY cm_cl
   aimant_age HAVING (COUNT(*) / (SELECT COUNT(*) FROM default.aclaims_master)) * 100
   < 5 ORDER BY percentage DESC
   ```

6. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
   b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
   c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. From the top of the Explorer page or from the Metadata Bar at the top of any dataset-level page, such as Findings, Profile, Dataset Rules, and Alert Builder, click the  to open Dataset Overview:
   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
3. From the Prompt dropdown menu, select **Dupe**.
   » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
4. In the query input field, enter the name of the column in your dataset to include in the SQL query.

Example In this example, the default placeholder column "email" will be replaced with the user-defined "cm_denied_reason1" in the duplicate value SQL suggestion.

```
Replace the column email with the column named:
cm_denied_reason1
```

Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column name.

5. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

   Example

   ```
   SELECT cm_denied_reason1, COUNT(*) AS count, (COUNT(*) / (SELECT COUNT(*) FROM def
   ault.aclaims_master)) * 100 AS percentage FROM default.aclaims_master GROUP BY cm_
   denied_reason1 HAVING count > 1 ORDER BY count DESC
   ```

6. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
   b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
   c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. From the top of the Explorer page or from the Metadata Bar at the top of any dataset-level page, such as Findings, Profile, Dataset Rules, and Alert Builder, click the  to open Dataset Overview:
   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
3. From the Prompt dropdown menu, select **Record**.
   » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
4. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

   Example In this example, the default placeholder columns "vendor_type" and "transaction_date" will be replaced with the user-defined "cm_claim_status" and "cm_bdos" in the SQL suggestion for record changes.

   ```
   Replace the column vendor_type and the date column transaction_date with the
   columns named
   cm_claim_status and cm_bdos
   ```

   Important You must specify one column with numeric data and one date column.

   Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

5. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

Example

```
SELECT cm_claim_status from default.aclaims_master where cm_bdos = DATE'2022-01-0
7' EXCEPT SELECT cm_claim_status from default.aclaims_master where cm_bdos = DAT
E'2022-01-07' - INTERVAL '1' DAY
```

6. Choose one of the following options:
    a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
    b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
    c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
    d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. From the top of the Explorer page or from the Metadata Bar at the top of any dataset-level page, such as Findings, Profile, Dataset Rules, and Alert Builder, click the  to open Dataset Overview:
    » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
    » The SQL Assistant for Data Quality tool appears.
3. From the Prompt dropdown menu, select **Pattern**.
    » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
4. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

Example In this example, the default placeholder columns "sale_state" and "state_tax" will be replaced with the user-defined "cm_diagnosis_code" and "cm_denied_reason1" in the SQL suggestion for pattern detection.

```
Replace the sale_state and
state_tax columns with these 2 columns:
cm_diagnosis_code and cm_denied_reason1
```

Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

5. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

Example

```
SELECT cm_diagnosis_code, cm_denied_reason1, COUNT(*) AS COUNT, ROUND( 100.0 * COU
NT(*) / ( SELECT COUNT(*) FROM default.aclaims_master ), 2 ) AS percentage FROM de
fault.aclaims_master GROUP BY cm_diagnosis_code, cm_denied_reason1 HAVING percenta
ge < 5
```

6. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
   b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
   c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. From the top of the Explorer page or from the Metadata Bar at the top of any dataset-level page, such as Findings, Profile, Dataset Rules, and Alert Builder, click the  to open Dataset Overview:
   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
3. From the Prompt dropdown menu, select **Frequency Distribution**.
   » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
4. In the query input field, enter the name of the column in your dataset to include in the SQL query.

   Example In this example, a SQL query for frequency distribution will be suggested for the column "cm_denied_reason1".

   ```
   Find the frequency distribution for the column named:
   cm_denied_reason1
   ```

   Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column name.

5. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

   Example

   ```
   SELECT cm_denied_reason1, COUNT(*) FROM default.aclaims_master GROUP BY cm_denied_
   reason1
   ```

6. Choose one of the following options:
    a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
    b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
    c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
    d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

Explorer

Explorer is an essential component that lets you onboard data from a connection, create the build layers of your DQ job, run it, and then view the results. The pages in this section detail the various steps and pages of the Explorer workflow.

Processing methods

This section details Pullup and Pushdown, the two different processing methods you can choose to submit a DQ job.

Pullup processing

Pullup is a compute method where all of the processing executes inside the Apache Spark compute engine. Source data is stored inside a database, where Spark reads it out, and the parameters you set in the Explorer workflow are partitioned and sorted. The profile results of the job are then recorded in the DQ Metastore.

Depending on the size of your dataset and the number of DQ checks performed, this process can greatly slow run times because Spark has its own compute resources, such as memory and CPUs. Pullup has limited support for profiling because it cannot run without Spark. In some

cases, you might need to partition your job into smaller segments for Spark to process in parallel and offset some of the compute load. A workaround to some of these Spark processing limitations is to use Pushdown mode instead.

Before the introduction of Pushdown, Pullup was the standard mode for creating and running DQ jobs. Unlike Pushdown, Pullup data sources and datasets are not identified throughout the app with an icon. Therefore, any dataset without the Pushdown icon PD next to it is considered a Pullup dataset.

Benefits of Pullup

- Highly configurable at the Spark-level.
- All DQ Layers are available.
- All connections listed on the Supported Connections page are available for use.

Prerequisites for using Pullup

Before running Pullup jobs, a Collibra DQ user with Admin permissions must:

- Configure a DQ agent.
- Create a data source connection.

Pushdown processing

Pushdown is an alternative computing method for running a DQ job, where all of the job's processing is submitted to a SQL data warehouse. DQ jobs with Pushdown processing generate SQL queries that offload the compute directly to the data source, reducing the amount of data transfer and Spark compute.

With Pushdown, you can also scale your computing needs based on the specific requirements of your DQ job. This is because the architecture of Pushdown data sources feature auto-scaling, which allows you to automatically scale up, or *burst*, to 64 or 128 nodes when you require more significant processing needs. Pushdown also automatically scales down when your DQ job does not require robust processing. With auto-scaling, the processing of your data is enhanced, improving runtime performance and removing the egress costs of reading large amounts of data.

Throughout the app, you can identify Pushdown jobs by the PD icon.

Benefits of Pushdown

By running a Pushdown job, you can:

- Reduce latency.
- Eliminate dependencies on Spark compute to run Collibra DQ, and increase processing speeds.
- Eliminate egress latency when running DQ jobs against large datasets.
- Auto-scale to data warehouses, such as Snowflake and Databricks.

Prerequisites for using Pushdown

Before running Pushdown jobs, a Collibra DQ user with Admin permissions must enable Pushdown for your data source. Pushdown can be enabled from the data source connection template in Admin Console ▸ Connections.

> **Tip** For first-time Snowflake Pushdown configurations, we recommend that you also successfully run the Pushdown setup script. If you do not run the setup script, ensure that you meet all the criteria that the script attempts to accomplish.

Available data sources and layers

Pushdown is currently available for the following data sources and layers:

		Basic Layers				Advanced Layers					
Data Source	Life-cycle	Rul-es	Beha-vior (Adap-tive Rules)	Sche-ma	Archi-ve Break Reco-rds	Out-liers	Dup-es	Sha-pes	Pat-terns	Reco-rds	Map-ping (Sour-ce to Tar-get)
Snow-flake	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✓ Yes	✓ Yes

		Basic Layers					Advanced Layers				
Data Source	Life-cycle	Rul-es	Beha-vior (Adap-tive Rules)	Sche-ma	Archi-ve Break Reco-rds	Out-liers	Dup-es	Sha-pes	Pat-terns	Reco-rds	Map-ping (Sour-ce to Tar-get)
Dat-abricks	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✗ No
BigQu-ery	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✗ No
Athena	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No
Red-shift	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No
Trino	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✓ Yes
SQL Server	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No
SAP HANA	Gen-erally avail-able	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✗ No

		Basic Layers					Advanced Layers				
Data Source	Life-cycle	Rul-es	Beha-vior (Adap-tive Rules)	Sche-ma	Archi-ve Break Reco-rds	Out-liers	Dup-es	Sha-pes	Pat-terns	Reco-rds	Map-ping (Sour-ce to Tar-get)
Oracle	in pre-view	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No

Running rules on Pushdown jobs

All rules that execute on Pushdown jobs use Database SQL dialects native to the specific data warehouse to which they submit your data for processing. Because Pushdown submits queries directly to the data source, the processing time of SQL rules reduces considerably.

The following table provides examples of how rules run on Pushdown jobs.

> **Note** The Translated Query Example column in the following table demonstrates the translation of the rule shown in the Collibra DQ Query Example column when you submit it to the data warehouse for processing.

Feature	Query Type	Definition	From	Collibra DQ Query Example	Translated Query Example
Native SQL	Physical Query (PQ)	Runs the entire query in native database dialect.	nyse.dataset	`select * from nyse.dataset where fname like 'John'`	`select * from nyse.dataset where fname like 'John'`

Feature	Query Type	Definition	From	Collibra DQ Query Example	Translated Query Example
Freeform SQL	Logical Query (LQ)	Runs the query including the original scope.	@nyse-dq-dataset	`select * from @nyse-dq-dataset where fname = 'John'`	`select * from (select * from nyse-dq-dataset where d_date-='2023-01-01') where fname = 'John'`
Freeform SQL	Template Logical Query (TLQ)	Runs the query including the original scope without specifying the dataset name.	@dataset	`select * from @data-set where fname = 'John'`	`select * from (select * where d_ date='2023-01-01') where fname = 'John'`

Known limitations

- We strongly encourage avoiding the use of special characters in table and column names for Pushdown datasets.
- Currently, you cannot run a job from the command line.
- Mixed data type rules are not supported in Pushdown.
- Binary data types are not supported in Pushdown, because there is no way to quantify the quality of unstructured binary data.
- Cross-connection rules are not supported in Pushdown.
- We do not currently support an Okta integration. Support for this integration is planned for a future version.
- There is a limitation where Snowflake Pushdown jobs with queries containing REGEXP_REPLACE fail because Snowflake does not currently support arguments with collation specifications. There are two known workarounds:

- Make a clone of the table that is synced in Snowflake and has no collation set.
- Use Pullup mode instead of Pushdown to avoid this Snowflake limitation.

Setting up a Pushdown connection

Pushdown processing is currently available for the data sources listed within the tabbed section below. Click a tab to see the Pushdown setup information for an available data source.

This section shows you how to set up Pushdown for a Databricks connection.

Requirements

In Collibra DQ

You have:

- Admin permissions to set up the Databricks connection template and enable Pushdown.
- Databricks JDBC driver version 2.6.32+ for Standalone deployments of Collibra DQ.

> **Note** You do not need to install the Databricks JDBC driver for Cloud native deployments of Collibra DQ because it is already included in the installation package.

In Databricks

You have:

- A personal access token to establish a connection between Collibra DQ and Databricks.
- A Databricks workspace admin SQL warehouse that is set up in your Databricks workspace.

> **Note** Your SQL warehouse can use either the Hive Metastore or the Delta Lake Unity Catalog for metadata management. Note that when you run Pushdown jobs on tables within the Hive Metastore without adding Unity Catalog to the connection string, Explorer displays all tables available in your Databricks workspace. However, only the tables that are viewable from Databricks under the Hive Metastore can be accessed to run Pushdown jobs on them. We recommend adding a Unity Catalog to your connection URL to run Pushdown jobs on all tables within your Databricks workspace.

- Read access to the tables on which you plan to run DQ scans.
    - A Databricks workspace admin can enable the Can Read permission in the Permission Settings section of your Databricks workspace.

Configuring Pushdown for Databricks

The following steps show you how to set up the Databricks connection template with Pushdown enabled.

Setting up the Databricks connection template

The following steps show you how to set up the Databricks connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click the Databricks tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your Databricks connection.

Required	Property	Type	Description
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:databricks://<account>.cloud.databricks.com/?db=DATABRICKS_SAMPLE_DATA;ConnCatalog=<catalog-name>` **Important** `ConnCatalog=<catalog-name>` is required if you use Unity Catalog for metadata management. However, it is not required if you use the Hive Metastore.
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is 0.
✓ Yes	Driver Name	String	The driver class name used for your connection, for example, `com.databricks.client.jdbc.Driver`
✕ No	Source Name	String	Not applicable.
✕ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown Jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for Databricks connections, go to the official Databricks documentation.

Required	Property	Type	Description
✓ Yes	Pushdown	Option	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your Databricks connection. **Important** You must select this option for Pushdown to function.
✕ No	Archive Break Records	Option	Allows you to write break records directly to a Databricks database or schema.
✓ Yes	Archive Break Records Location	String	The break records archive location within Databricks. If you do not specify a location, then Collibra DQ uses PUBLIC by default.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver.
✕ No	Driver Properties	String	Optional. The configurable driver properties for your connection.

3. Click **Save**.

What's next?

You can now create a Pushdown job on your Databricks connection.

This section shows you how to set up Pushdown for a Snowflake connection.

- Running Snowflake setup scripts
- Setting up the Snowflake Connection template

Running Snowflake setup scripts

> **Note** The following scripts are examples only. Be sure to modify them for your environment.

Step	Details
1	Open a Snowflake worksheet.

2	Update the following session variables:
	`set dq_username='SERVICE_ACCOUNT_USER';`
	`set dq_password='SERVICE_ACCOUNT_PASSWORD';`
	`set dq_warehouse_name='COLLIBRA_DQ_WH';`
	`set dq_warehouse_size='XSMALL';`
	`set user_database='TARGET_DB';`
	`set dq_role_name='COLLIBRA_DQ_ROLE';`
	Note `dq_password` must be uppercase.

3	Run as an admin user. `USE ROLE ACCOUNTADMIN;` **Note** If the usage of ACCOUNTADMIN is restricted in your organization, you may use USERADMIN, SECURITYADMIN, and SYSADMIN instead.

Step	Details

4	Create a user and role for Collibra DQ.
	
	`CREATE ROLE IF NOT EXISTS identifier($dq_role_name);`
	`CREATE USER IF NOT EXISTS identifier($dq_username) PASSWORD=$dq_password DEFAULT_ROLE=$dq_role_name;`
	`GRANT ROLE identifier($dq_role_name) TO USER identifier($dq_username);`
	
	Note Customers may manage users and roles according to their own policies and procedures.

5	Create a virtual warehouse to run Collibra DQ. The virtual warehouse is referenced in the connection URL to direct DQ traffic to it. `CREATE WAREHOUSE IF NOT EXISTS identifier($dq_warehouse_name) WAREHOUSE_SIZE=$dq_warehouse_size INITIALLY_SUSPENDED=TRUE AUTO_SUSPEND = 5 AUTO_RESUME = TRUE;` **Note** Customers may manage warehouses according to their own policies and procedures.
6	Assign privileges to the Collibra DQ warehouse. `GRANT OPERATE, USAGE, MONITOR ON WAREHOUSE identifier($dq_warehouse_name) TO ROLE identifier($dq_role_name);` **Note** Customers may manage warehouses according to their own policies and procedures.

Step	Details

7	Assign metadata access to your Collibra DQ role.
	
	`GRANT USAGE,MONITOR on DATABASE identifier ($user_database) to identifier($dq_role_name);`
	`GRANT USAGE,MONITOR ON ALL SCHEMAS IN DATABASE identifier($user_database) to identifier($dq_role_name);`
	
	Note Customers may manage databases and schemas according to their own policies and procedures.

Step	Details
8	Update the session variable `user_database` and grant read access to objects in the user database. Run this portion for each target database within which you want to run DQ checks.

```
USE DATABASE identifier($user_database);
GRANT SELECT ON ALL TABLES IN DATABASE
identifier($user_database) TO ROLE identifier
($dq_role_name);
GRANT SELECT ON ALL VIEWS IN DATABASE
identifier($user_database) TO ROLE identifier
($dq_role_name);
GRANT SELECT ON ALL EXTERNAL TABLES IN
DATABASE identifier($user_database) TO ROLE
identifier($dq_role_name);
GRANT SELECT ON ALL STREAMS IN DATABASE
identifier($user_database) TO ROLE identifier
($dq_role_name);

GRANT SELECT ON FUTURE TABLES IN DATABASE
identifier($user_database) TO ROLE identifier
($dq_role_name);
GRANT SELECT ON FUTURE VIEWS IN DATABASE
identifier($user_database) TO ROLE identifier
($dq_role_name);
GRANT SELECT ON FUTURE EXTERNAL TABLES IN
DATABASE identifier($user_database) TO ROLE
identifier($dq_role_name);
GRANT SELECT ON FUTURE STREAMS IN DATABASE
identifier($user_database) TO ROLE identifier
($dq_role_name);
```

Note For information on configuring archiving, go to Archiving break records from Pushdown jobs.

Setting up the Snowflake Connection template

The following steps show you how to set up the Snowflake connection template with Pushdown enabled.

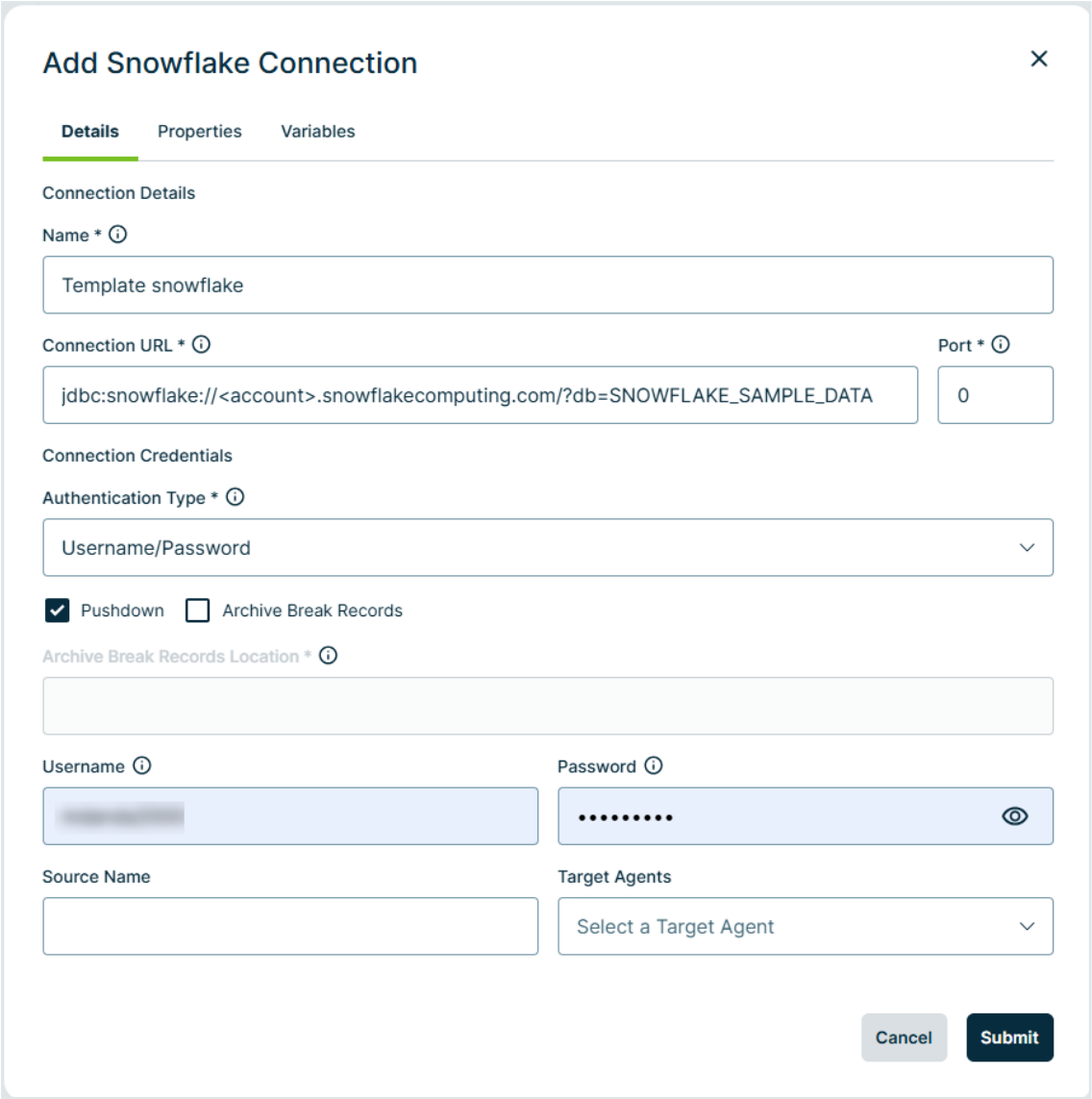
1. On the main menu, click ⚙, and then click **Connections**.
2. Click the Snowflake tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	Text	The unique name of your Snowflake connection.
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:snowflake://<account>.snowflakecomputing.com/?db=SNOWFLAKE_SAMPLE_DATA&warehouse=COLLIBRA_DQ_WH`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource.
✓ Yes	Driver Name	Text	The driver class name used for your connection, for example, `com.snowflake.client.jdbc.SnowflakeDriver`
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent used to submit your DQ Job. You do not need to select an option because Pushdown Jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. See the Snowflake documentation for more information about authentication methods for Snowflake connections.

Required	Property	Type	Description
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your Snowflake connection. **Warning** You must select this option for Pushdown to function.
✗ No	Archive Break Records	Option	Allows you to write break records directly to a Snowflake database or schema. **Important** Ensure that you have the necessary privileges to write break records to Snowflake.
✓ Yes	Driver Location	File path	A file path which designates the name and location of the driver.
✓ Yes	Driver Properties	String	The configurable driver properties for your connection. Add the following driver property: `CLIENT_METADATA_REQUEST_USE_CONNECTION_CTX=TRUE`

Show example

The following image shows an example of a Snowflake connection with Pushdown selected.



3. Click **Save**.

Known limitations

- There is a limitation where Snowflake Pushdown jobs with queries containing REGEXP_REPLACE fail because Snowflake does not currently support arguments with collation specifications. There are two known workarounds:

- Make a clone of the table that is synced in Snowflake and has no collation set.
- Use Pullup mode instead of Pushdown to avoid this Snowflake limitation.

What's next?

You can now create a Pushdown job on your Snowflake connection.

To run a Snowflake Pushdown job, you need to opt in when setting up your Snowflake connection. Select the Pushdown option in the Connection template to turn on Pushdown capabilities.

This section shows you how to set up Pushdown for a BigQuery connection.

Requirements

In Collibra DQ

- You have admin permissions (`ROLE_ADMIN`) to set up the BigQuery connection template and enable Pushdown.
- You have BigQuery JDBC driver version 1.3.2.1003 for Standalone deployments of Collibra DQ.

> **Note** You do not need to install the BigQuery JDBC driver for Cloud native deployments of Collibra DQ because it is already included in the installation package.

In BigQuery

- You have `Viewer` (`roles/viewer`) permissions.
- You have `Reader` access to BigQuery dataset tables.
- You have `Writer` access to materialized view datasets.

Configuring Pushdown for BigQuery

The following steps show you how to set up the BigQuery connection template with Pushdown enabled.

Setting up the BigQuery connection template

The following steps show you how to set up the BigQuery connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click the BigQuery tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your BigQuery connection.
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:bigquery://https://www.googleapis.com/bigquery/v2:443;OAuthType=0;ProjectId=owl-hadoop-cdh;OAuthServiceAcctEmail=<service-account-email>;TimeOut=3600`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is 443.
✓ Yes	Driver Name	String	The driver class name used for your connection, for example, `com.simba.googlebigquery.jdbc42.Driver` **Note** The `cdata.jdbc.googlebigquery.GoogleBigQueryDriver` does not support Pushdown.

Required	Property	Type	Description
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for BigQuery connections, go to the official BigQuery documentation.
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your BigQuery connection. **Important** You must select this option for Pushdown to function.
✗ No	Archive Break Records	Option	Allows you to write break records directly to a BigQuery database or schema.

Required	Property	Type	Description
✓ Yes	Archive Break Records Location	String	The break records archive location within BigQuery. If you do not specify a location, then Collibra DQ uses `PUBLIC` by default. **Note** Archive Break Records Location is only available and required when the Archive Break Records option is selected.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver. For example, /opt/owl/drivers/bigquery
✕ No	Driver Properties	String	Optional. The configurable driver properties of your connection.

3. Click **Save**.

Limitations

Rule validation for BigQuery Pushdown connections is not supported. However, you can manually execute SQL queries from the BigQuery console on GCP as a possible workaround.

What's next?

You can now create a Pushdown job on your BigQuery connection.

This section shows you how to set up Pushdown for a Amazon Athena connection.

Requirements

In Collibra DQ

- You have admin permissions (`ROLE_ADMIN`) to set up the Athena connection template and enable Pushdown.
- You have Athena JDBC driver version 2.0.35.1000 for Standalone deployments of Collibra DQ.

> **Note** You do not need to install the Athena JDBC driver for Cloud native deployments of Collibra DQ because it is already included in the installation package.

In Athena

- You have `Read` access to your Glue Catalog and S3 query result bucket location.
- You have `Write` access to your S3 query result bucket location.
- You have `Read` access to the Athena workgroup.

Configuring Pushdown for Athena

The following steps show you how to set up the Athena connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click the Athena tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your Athena connection.

Required	Property	Type	Description
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:awsathena://AwsRegion=${region};S3OutputLocation=s3://{output_location};MetadataRetrievalMethod=Query`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is 444.
✓ Yes	Driver Name	String	The driver class name of your connection, for example, `com.simba.athena.jdbc.Driver` **Note** The `cdata.jdbc.amazonathena` driver class does not support Pushdown.
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for Athena connections, go to the official Athena documentation.

Required	Property	Type	Description
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your Athena connection. **Important** You must select this option for Pushdown to function.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver. For example, /opt/owl/drivers/athena/
✕ No	Driver Properties	String	Optional. The configurable driver properties of your connection.

3. Click **Save**.

What's next?

You can now create a Pushdown job on your Athena connection.

Known limitations

- Only the AdaptiveRules layer is available from the Add Layers step.

This section shows you how to set up Pushdown for a Redshift connection.

Requirements

In Collibra DQ

- You have admin permissions (`ROLE_ADMIN`) to set up the Redshift connection template and enable Pushdown.
- You have Redshift JDBC driver version 2.1.0.9 for Standalone deployments of Collibra DQ.

> **Note** You do not need to install the Redshift JDBC driver for Cloud native deployments of Collibra DQ because it is already included in the installation package.

In Redshift

- You have `READ` access on your Redshift database tables.
- You have `USAGE` access on your Redshift schema.

Configuring Pushdown for Redshift

The following steps show you how to set up the Redshift connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click the Redshift tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your Redshift connection.
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:redshift://<host>:5439/dev`

Required	Property	Type	Description
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is 5439.
✓ Yes	Driver Name	String	The driver class name used for your connection, for example, `com.amazon.redshift.jdbc42.Driver`
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for Redshift connections, go to the official Redshift documentation.
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your Redshift connection. **Important** You must select this option for Pushdown to function.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver. For example, /opt/owl/drivers/redshift/

Required	Property	Type	Description
✕ No	Driver Properties	String	Optional. The configurable driver properties of your connection.

3. Click **Save**.

What's next?

You can now create a Pushdown job on your Redshift connection.

Known limitations

- Only the AdaptiveRules layer is available from the Add Layers step.

This section shows you how to set up Pushdown for a Trino connection.

Requirements

In Collibra DQ

- You have admin permissions (`ROLE_ADMIN`) to set up the Trino connection template and enable Pushdown.
- You have Trino JDBC driver version 405.0 for Standalone deployments of Collibra DQ.

> **Note** You do not need to install the Trino JDBC driver for Cloud native deployments of Collibra DQ because it is already included in the installation package.

In Trino

- You have Read access on your Trino database tables.

Configuring Pushdown for Trino

The following steps show you how to set up the Trino connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click the Trino tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your Trino connection.
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:trino://${host}:443/<catalog>/<schema>&source=jdbc:presto`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is 8080.
✓ Yes	Driver Name	String	The driver class name used for your connection, for example, `io.trino.jdbc.TrinoDriver`
✕ No	Source Name	String	N/A
✕ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown jobs do not require an Agent to submit them to the data warehouse.

Required	Property	Type	Description
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for Trino connections, go to the official Trino documentation.
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your Trino connection. **Important** You must select this option for Pushdown to function.
✕ No	Archive Break Records	Option	Allows you to write break records directly to a Trino database or schema.
✓ Yes	Archive Break Records Location	String	The break records archive location within Trino. If you do not specify a location, then Collibra DQ uses `PUBLIC` by default. **Note** Archive Break Records Location is only available and required when the Archive Break Records option is selected.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver. For example, /opt/owl/drivers/trino
✕ No	Driver Properties	String	Optional. The configurable driver properties of your connection.

3. Click **Save**.

Turning on Pushdown for Trino

To ensure that Pushdown for Trino is appropriately configured, verify that PUSHDOWN_JOB_ENABLED is set to `TRUE` on the Application Config page of the Admin Console.

1. On the main menu, click ⚙, and then click **Admin Console**.
2. Click **Application Config** under the Configuration Settings section.

   > **Tip** If PUSHDOWN_JOB_ENABLED is already set to `TRUE`, you can skip this section.

3. If PUSHDOWN_JOB_ENABLED is set to `FALSE`, click the ✎ to the right of the Value column.
4. In the Edit Configuration modal, enter `TRUE` into the Value input field.
5. Click **Save**.

What's next?

You can now create a Pushdown job on your Trino connection.

This section shows you how to set up Pushdown for a SQL Server connection.

Requirements

In Collibra DQ

- You have admin permissions (`ROLE_ADMIN`) to set up the SQL Server connection template and enable Pushdown.
- You have the SQL Server JDBC driver for Standalone deployments of Collibra DQ.

> **Note** You do not need to install the SQL Server JDBC driver for Cloud native deployments of Collibra DQ because it is already included in the installation package.

In SQL Server

- You have Read access on your SQL Server database tables.

Configuring Pushdown for SQL Server

The following steps show you how to set up the SQL Server connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click the SQL Server tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your SQL Server connection.
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:sqlserver://${host}:1433/databaseName=${dbName};encrypt=true;trustServerCertificate=true;`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is 1433.

Required	Property	Type	Description
✓ Yes	Driver Name	String	The driver class name used for your connection, for example, `com.microsoft.sqlserver.jdbc.SQLServerDriver`
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for SQL Server connections, go to the official SQL Server documentation.
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your SQL Server connection. **Important** You must select this option for Pushdown to function.

Required	Property	Type	Description
✕ No	Archive Break Records	Option	Allows you to write break records directly to a SQL Server database or schema. **Note** Only rule break records are eligible for source archival.
✓ Yes	Archive Break Records Location	String	The break records archive location within SQL Server. If you do not specify a location, then Collibra DQ uses `SQLS` by default. **Note** Archive Break Records Location is only available and required when the Archive Break Records option is selected.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver. For example, /opt/owl/drivers/mssql/
✕ No	Driver Properties	String	Optional. The configurable driver properties of your connection.

3. Click **Save**.

What's next?

You can now create a Pushdown job on your SQL Server connection.

Known limitations

- As of the Collibra DQ 2024.02 release, only AdaptiveRules are available from the Add Layers step.
- Only native Freeform SQL (PQ) is supported for SQLF rules, meaning that queries in the format of `@dataset` (TLQ) or `@job-name` (LQ) are not supported. Only queries that use native (PQ) are supported.
- The use of the 'TOP(limit)' clause for the source is not supported when Link ID is enabled for SQLF rules. This is because the 'TOP N' specification should be placed at the beginning of the query. When Link ID is enabled for rules, it is added to the beginning of the query, causing the Insert query to fail.

This section shows you how to set up Pushdown for a SAP HANA connection.

Requirements

In Collibra DQ

- You have admin permissions (`ROLE_ADMIN`) to set up the SAP HANA connection template and enable Pushdown.
- You have the SAP HANA JDBC driver for Standalone deployments of Collibra DQ.

> **Note** You do not need to install the SAP HANA JDBC driver for Cloud native deployments of Collibra DQ because it is already included in the installation package.

In SAP HANA

- You have Read access on the SAP HANA database tables on which you run data quality.

Configuring Pushdown for SAP HANA

The following steps show you how to set up the SAP HANA connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click **Add Connection**, then click SAP HANA.
3. Enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your SAP HANA connection.
✓ Yes	Connection URL	String	The base connection string, for example, `jdbc:sap://${host}:${port}/`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is 39015.
✓ Yes	Driver Name	String	The driver class name used for your connection, for example, `com.sap.db.jdbc.Driver`
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown jobs do not require an Agent to submit them to the data warehouse.
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for SAP HANA connections, go to the official SAP HANAdocumentation.

Required	Property	Type	Description
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your SAP HANA connection. **Important** You must select this option for Pushdown to function.
✗ No	Archive Break Records	Option	Allows you to write break records directly to a SAP HANA database or schema.
✓ Yes	Archive Break Records Location	String	The break records archive location within SAP HANA. If you do not specify a location, then Collibra DQ uses `SAPHANADB` by default. Your default output schema may be one of the following: • SAP<SID> • SAPSR3 • SAPBAP<#> • SAPHANADB **Note** Archive Break Records Location is only available and required when the Archive Break Records option is selected.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver. For example, /opt/owl/drivers/sap/
✗ No	Driver Properties	String	Optional. The configurable driver properties of your connection.

4. Click **Save**.

The following table contains an overview of the data quality features currently available for Pushdown for SAP HANA.

What's next?

You can now create a Pushdown job on your SAP HANA connection.

This section shows you how to set up Pushdown for a Oracle connection.

Prerequisites

- You have admin permissions (`ROLE_ADMIN`) to set up the Oracle connection template and enable Pushdown.
- You have read access on Oracle database tables.
- The Kerberos user has read permissions on Oracle tables (if using Kerberos authentication).

Configuring Pushdown for Oracle

The following steps show you how to set up the Oracle connection template with Pushdown enabled.

Setting up the Oracle connection template

The following steps show you how to set up the Oracle connection template with Pushdown enabled.

1. On the main menu, click ⚙, and then click **Connections**.
2. Click the Oracle tile, then enter the required information.

Required	Property	Type	Description
✓ Yes	Name	String	The unique name of your Oracle connection.

✓ Yes	Connection URL	String	The connection string path of your Oracle connection.
			When referring to the example below, replace the `${value}` sections of the connection URL with your actual value.
			Example Using SID
			`jdbc:oracle:thin:@${host}:SID`
			Example Using database service name
			`jdbc:oracle:thin:@//${host}:${port}/service_name`

✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `1521`
✓ Yes	Driver Name	String	The driver class name of your connection. `oracle.jdbc.OracleDriver`
✗ No	Source Name	String	N/A
✗ No	Target Agent	Option	The Agent used to submit your DQ Job. **Note** You do not need to select an option because Pushdown jobs do not require an Agent to submit them to the data warehouse.

Required	Property	Type	Description
✓ Yes	Auth Type	Option	The method to authenticate your connection. For more information on authentication methods for Oracle connections, go to the official Oracle documentation.
✓ Yes	Pushdown	Checkbox	Switches Pushdown processing on or off. Select the checkbox option to turn on Pushdown for your Oracle connection. **Important** You must select this option for Pushdown to function.
✗ No	Archive Break Records	Option	Allows you to write break records directly to a Oracle database or schema.
✓ Yes	Archive Break Records Location	String	The break records archive location within Oracle. If you do not specify a location, then Collibra DQ uses `PUBLIC` by default. **Note** Archive Break Records Location is only available and required when the Archive Break Records option is selected.
✓ Yes	Driver Location	File path	A file path that designates the name and location of the driver. For example, /opt/owl/drivers/oracle
✗ No	Driver Properties	String	Optional. The configurable driver properties of your connection.

3. Click **Save**.

What's next?

You can now create a Pushdown job on your Oracle connection.

Create a DQ job

This page provides an overview of the steps involved in the creation of a DQ job. Select from one of the options below depending on whether you are using Pullup or Pushdown mode.

Create a Pullup job

Step	Action	Description
1	Select a data source	Your data source is where your table, file, or view upon which your DQ job runs exists.
2	Select a scanning method	Determine whether to scan part of your table, file, or view, a sample of it, or its entirety.

Step	Action	Description
3	Select your columns	Select the columns to include in your DQ job. Optionally assign Key columns and Link IDs. Link IDs are required for archiving break records. **Note** This step is optional for Full Scans.
4	Select your rows	Apply filters and limits to your rows for enhanced scanning. **Note** This step is optional for Full Scans.
5	Add transforms	Modify or create new columns to meet specific formatting requirements at the Spark level.

Step	Action	Description
6	Create a mapping	Validate and link source and target column relationships between two data sources.
7	Add DQ layers	Customize your DQ job to gain more insights into the health of your data.
8	Review the job sizing	Review and modify details about your agent resources.
9	Review your DQ job	Review your query before you submit your DQ job.
10	Run your DQ job	Submit your DQ job for processing.

Create a Pushdown job

Step	Action	Description
1	Select a data source	Your data source is where your table, file, or view upon which your DQ job runs exists.
2	Select a scanning method	Determine whether to scan part of your table, file, or view, a sample of it, or its entirety.
3	Select your columns	Select the columns to include in your DQ job. Optionally assign Key columns and Link IDs. Link IDs are required for archiving break records.

> **Note**
> This step is optional for Full Scans.

Step	Action	Description
4	Select your rows	Apply filters and limits to your rows for enhanced scanning. **Note** This step is optional for Full Scans.
5	Add DQ layers	Customize your DQ job to gain more insights into the health of your data.
6	Review your DQ job	Review your query before you submit your DQ job.
7	Run your DQ job	Submit your DQ job for processing.

Connect to a data source

Connect to a data source to onboard its data and run a DQ job against it. When you expand a datasource in the **Explorer** page, a list of available databases and their associated tables display.

Creating a DQ job

1. Click the **Explorer** in the sidebar menu.
   » The **Manage Connections** treeview expands to show available data sources.

> **Note** When PD displays to the right of a data source, it indicates that the data source is a Pushdown connection. When the icon is not present, the data source is considered a Pullup connection.

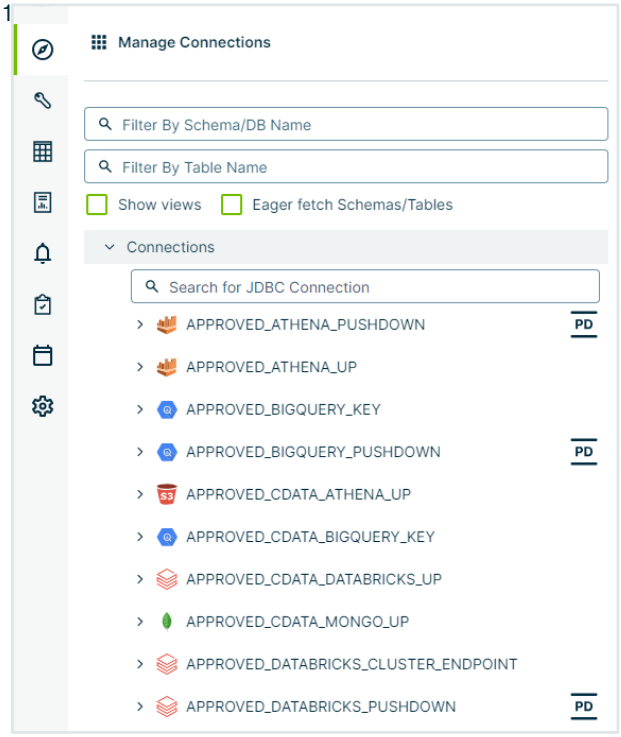
2. Select a data source.
   » The data source expands to show available databases.
3. Click a database.
   » The database expands to show available tables.
4. Click **Create DQ Job**.
   » The Job Creator page opens.

Example screenshot...[1]

Eager fetch schemas and tables

When you select the **Eager fetch Schemas/Tables** option from the Manage Connections treeview, all tables of all public schemas in your data source load together, all at once. While this option has more processing requirements and is thus more expensive, you can use it to

1



retrieve all of the children of a JDBC connection that would otherwise persist for a limited time only.

Coverage Stats

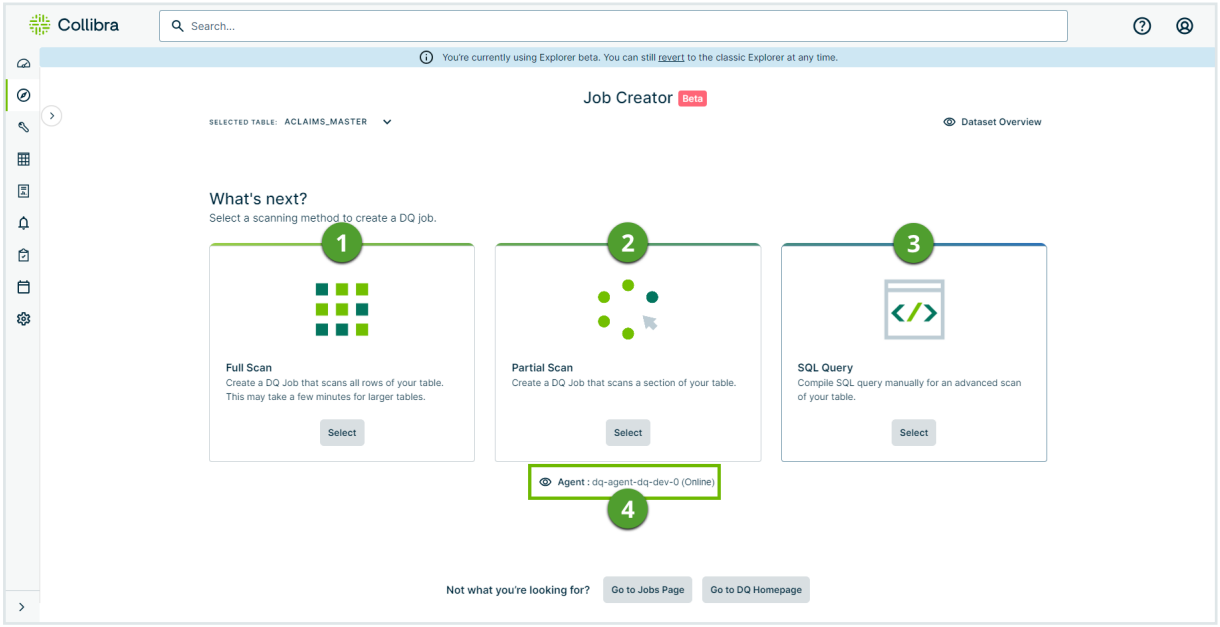
The Coverage Stats page opens when you click **Generate Report** from the Manage Connections treeview. The page includes a donut chart showing the percentage of all schemas in the database that are running data quality on them and interactive bar charts showing which data sets are running data quality jobs on a monthly basis.

What's next?

Select a scanning method to get started.

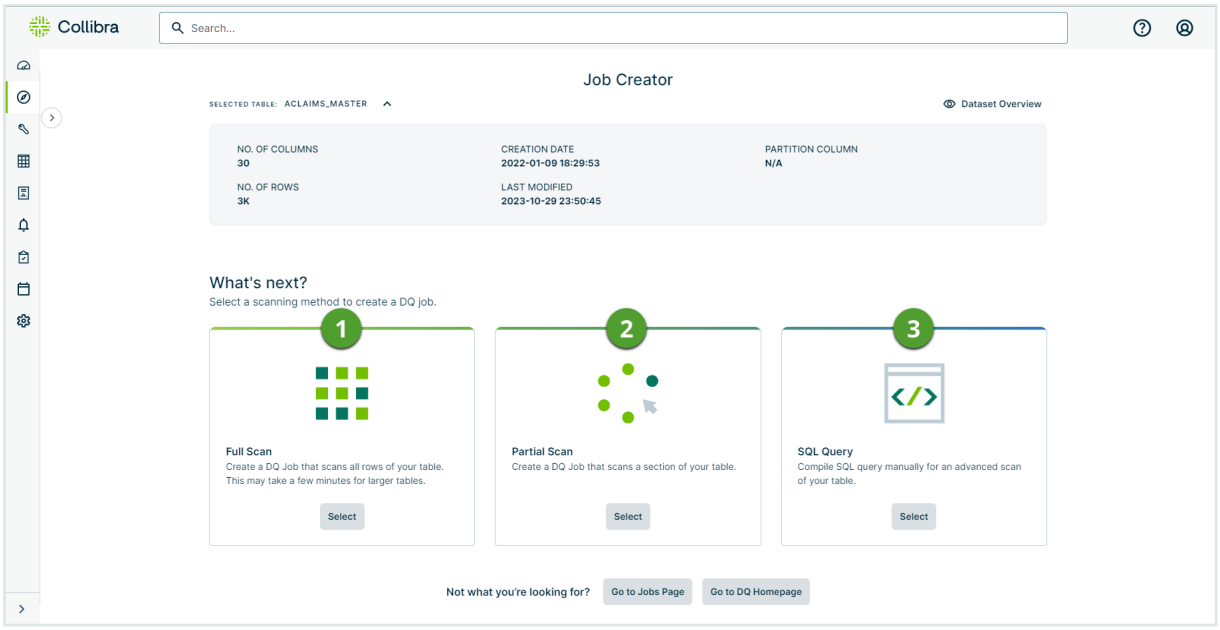
Select a scanning method

After you connect to a data source and begin to create a new DQ job, you need to select the scanning method to use on your table, file, or view. Select from one of the options below depending on whether you are using Pullup or Pushdown mode.



The following table shows the available scanning methods. Each option has a number that corresponds with the number in the image above.

Number	Option	Description
1	Full Scan	A full scan is a scan of your entire table. When running a full scan, you do not need to select columns or apply filters to rows because all columns are selected by default. Various DQ Layers, such as AdaptiveRules, are also applied by default. A full scan is commonly used to obtain a high-level overview of your data. If you select Full Scan, the next step is to Add Layers.
2	Partial Scan	A partial scan only scans the sections of your table that you specify Select columns, apply row filters, and add DQ layers to fine tune your DQ scan and obtain a targeted understanding of your data. If you select Partial Scan, the next step is to Select Columns.
3	SQL Query	The SQL Query option lets advanced users write and compile a query manually for an advanced scan of your table. If you select SQL Query, the next step is to Select Rows in the SQL View. **Warning** While Data Quality & Observability Classic supports the manual compilation of SQL, the misuse of raw SQL presents an elevated security risk (SQLi). Therefore, only advanced SQL users should use this scanning option.
4	Agent	Shows the agent and its status as either "Online" or "Offline". You can click the agent name to open the Agent status dialogue where you can view more details about it.



The following table shows the available scanning methods. Each option has a number that corresponds with the number in the image above.

Number	Option	Description
1	Full Scan	A full scan is a scan of your entire table. When running a full scan, you do not need to select columns or apply filters to rows because all columns are selected by default. Various DQ Layers, such as AdaptiveRules, are also applied by default. A full scan is commonly used to obtain a high-level overview of your data. If you select Full Scan, the next step is to Add Layers.
2	Partial Scan	A partial scan only scans the sections of your table that you specify Select columns, apply row filters, and add DQ layers to fine tune your DQ scan and obtain a targeted understanding of your data. If you select Partial Scan, the next step is to Select Columns.

Number	Option	Description
3	SQL Query	The SQL Query option lets advanced users write and compile a query manually for an advanced scan of your table. If you select SQL Query, the next step is to Select Rows in the SQL View. **Warning** While Data Quality & Observability Classic supports the manual compilation of SQL, the misuse of raw SQL presents an elevated security risk (SQLi). Therefore, only advanced SQL users should use this scanning option.

Select columns

Select the columns to include in your DQ job.

1. Select from the available options.

Column	Description
☐	The column(s) to include in your data quality scan. You can select the checkbox option in the column header to select or deselect all columns, or select each column individually. At least one column selection is required.
Column	The name of the column in your table or view. Click the **Assign** chip in a Key column to uniquely identify the rows of the column. > **Note** You can assign only one column as the Key column and it cannot be the column you select for outlier analysis.
Type	The data type of a column. This column is auto-filled and cannot be edited or selected. > **Note** Collibra DQ supports all standard data types. Collibra DQ also supports custom data types, but this may vary depending on the data source. For more information, see unsupported data types.
Key	The key column whose rows have unique identifiers. This column provides context by grouping each column by the key column. When detected, breaking records are grouped to their assigned column when assigned. Optional. Optionally click the **Assign** chip to uniquely identify the rows of a column.
Link ID	Links the findings of a DQ job back to the source record, or key, for remediation outside of Collibra DQ. The link ID should be unique and is most commonly the primary key. Link IDs are required for archiving break records. For more information, go to Working with Link IDs.

2. Click **Next** to select your rows.

Select rows

This page shows you the details about Time Slice filter, Row Filter, and Limit, and how to configure them. Select from one of the options below depending on whether you are using

Pullup or Pushdown mode.

Time Slice

A Time Slice filter is a dynamic filter that scans a range of dates or times and lets you perform incremental DQ runs on your time series data.

Time Slice uses the current timestamp as your run date to isolate the latest time slice data from your table and runs a DQ job on that portion for that day. For example, a run from the previous day checks for yesterday's date and groups it as a time slice, and, as long as a column that includes today's date exists in the table, a run from the current day will create a time slice for today's date and include it in the run. This grouping of time slices is necessary if you use Replay when you run your DQ Job because it replaces the run date with the run date from the time slice.

You can use it to perform incremental DQ runs on your time series data.

Click **Add** to create a Time Slice filter. The following table shows the available options for the Time Slice filter:

Option	Description
Column of Reference	The column that the Time Slice filter analyzes and includes in the query.
Date Format	The default format is YYYY-MM-DD. When you select a date column as your Column of Reference, the format is the same as the column you select from your table, file, or view.
Group by	Lets you change the grouping of the x-axis on the Data Distribution chart by Day, Week, or Year. Click **Visualize** for this option to appear. **Note** **Group by** is only available when the Column of Reference is a date column.

Option	Description
Data Distribution (chart)	A chart of the distribution of available data for the Column of Reference. Click any of the bars in the chart to apply your selection to the Start Date and automatically update the Query Preview. **Note** Data Distribution is only available after you select a Column of Reference option.
Operator	The symbol that specifies the action upon which the SQL query performs one or more expressions. Available operators include: • **=**: Equal to • **>**: Greater than • **>=**: Greater than or equal to • **<**: Less than • **<=**: Less than or equal to • **!=**: Not equal to • **24H Day**: 24 hour day, or one day • **Date Range**: Lets you select a **Start date** and an **End date** from the date picker.
Start Date	A run date that allows the application to scan sections of your table, add a new date to it, and then run again. By selecting Start Date, a dynamic result is returned at runtime. Select a datetime from the date picker.
Query Preview	A preview of the automatically compiled SQL query.

Row Filter

A Row Filter is a static filter that scans a section of rows. You can use this to isolate your DQ Jobs to filter based on certain groups of data. For example, you can use the Row Filter to filter data based on a particular region, such as North America.

Click **Add** to create a Row Filter. The following table shows the available options for the Row Filter:

Option	Description
Column of Reference	The column that the Time Slice filter analyzes and includes in the query.
Date Format	The default format is YYYY-MM-DD. When you select a date column as your Column of Reference, the format is the same as the column you select from your table, file, or view.
Group by	Lets you change the grouping of the x-axis on the Data Distribution chart by Day, Week, or Year. Click **Visualize** for this option to appear.
Data Distribution (chart)	A chart of the distribution of available data for the Column of Reference. Click any of the bars in the chart to apply your selection to the Start Date and automatically update the Query Preview. > **Note** Data Distribution is only available after you select a Column of Reference option.
Operator	The symbol that specifies the action upon which the SQL query performs one or more expressions. Available operators include: • **=**: Equal to • **>**: Greater than • **>=**: Greater than or equal to • **<**: Less than • **<=**: Less than or equal to • **!=**: Not equal to • **24H Day**: 24 hour day, or one day • **Date Range**: Lets you select a **Start date** and an **End date** from the date picker.

Option	Description
Start Date	A run date that allows the application to scan sections of your table, add a new date to it, and then run again. By selecting Start Date, a dynamic result is returned at runtime. Select a datetime from the date picker.
Query Preview	A preview of the automatically compiled SQL query.

> **Note** You can apply only one Row Filter from the UI. To add more filters, switch to the manual SQL query mode and modify the query.

Limit

Scans a random sample of rows based on the maximum number of rows you set. Enter a value greater than 0 or click the arrows in the **Max. rows** field to apply a limit.

Escape Character

The **Escape Character** dropdown menu allows you to optionally select the special character with which to wrap column names. This can be especially helpful if your column name contains a special character that you need to escape in order for Collibra DQ to properly identify it. If you do not specify an escape character, then the column name is used as it appears in your table, file, or view. Available escape characters include:

- Double Quotes (`" "`)
- Single Quote (`' '`)
- Backtick (`` ` ` ``)
- Brackets (`[ ]`)
- Custom
    - When you select Custom, enter your custom escape character into the input field to the right of the Escape Character dropdown menu.

SQL View

SQL View allows advanced SQL users to write and compile SQL queries manually. SQL View enforces the same syntax restrictions as the Native SQL rule type. For more information, go to Native.

To open SQL View, click the three dots menu in the upper right corner of the Filter Type window. Click the three dots again to switch back to Standard View.

> **Note** If you use the SQL View option and include a JOIN in your dataset query when there are two columns with the same name, you must alias one of the columns to allow them to be identified as distinct in the Spark dataframe and allow your DQ Job to run successfully.

Time Slice

A Time Slice filter is a dynamic filter that scans a range of dates or times and lets you perform incremental DQ runs on your time series data.

Time Slice uses the current timestamp as your run date to isolate the latest time slice data from your table and runs a DQ job on that portion for that day. For example, a run from the previous day checks for yesterday's date and groups it as a time slice, and, as long as a column that includes today's date exists in the table, a run from the current day will create a time slice for today's date and include it in the run. This grouping of time slices is necessary if you use Replay when you run your DQ Job because it replaces the run date with the run date from the time slice.

You can use it to perform incremental DQ runs on your time series data.

Click **Add** to create a Time Slice filter. The following table shows the available options for the Time Slice filter:

Option	Description
Column of Reference	The column that the Time Slice filter analyzes and includes in the query.
Date Format	The default format is YYYY-MM-DD. When you select a date column as your Column of Reference, the format is the same as the column you select from your table, file, or view.
Group by	Lets you change the grouping of the x-axis on the Data Distribution chart by Day, Week, or Year. Click **Visualize** for this option to appear. **Note** **Group by** is only available when the Column of Reference is a date column.
Data Distribution (chart)	A chart of the distribution of available data for the Column of Reference. Click any of the bars in the chart to apply your selection to the Start Date and automatically update the Query Preview. **Note** Data Distribution is only available after you select a Column of Reference option.
Operator	The symbol that specifies the action upon which the SQL query performs one or more expressions. Available operators include: • **=**: Equal to • **>**: Greater than • **>=**: Greater than or equal to • **<**: Less than • **<=**: Less than or equal to • **!=**: Not equal to • **24H Day**: 24 hour day, or one day • **Date Range**: Lets you select a **Start date** and an **End date** from the date picker.

Option	Description
Start Date	A run date that allows the application to scan sections of your table, add a new date to it, and then run again. By selecting Start Date, a dynamic result is returned at runtime. Select a datetime from the date picker.
Query Preview	A preview of the automatically compiled SQL query.

Row Filter

A Row Filter is a static filter that scans a section of rows. You can use this to isolate your DQ Jobs to filter based on certain groups of data. For example, you can use the Row Filter to filter data based on a particular region, such as North America.

Click **Add** to create a Row Filter. The following table shows the available options for the Row Filter:

Option	Description
Column of Reference	The column that the Time Slice filter analyzes and includes in the query.
Date Format	The default format is YYYY-MM-DD. When you select a date column as your Column of Reference, the format is the same as the column you select from your table, file, or view.
Group by	Lets you change the grouping of the x-axis on the Data Distribution chart by Day, Week, or Year. Click **Visualize** for this option to appear.
Data Distribution (chart)	A chart of the distribution of available data for the Column of Reference. Click any of the bars in the chart to apply your selection to the Start Date and automatically update the Query Preview. **Note** Data Distribution is only available after you select a Column of Reference option.

Option	Description
Operator	The symbol that specifies the action upon which the SQL query performs one or more expressions. Available operators include: • **=**: Equal to • **>**: Greater than • **>=**: Greater than or equal to • **<**: Less than • **<=**: Less than or equal to • **!=**: Not equal to • **24H Day**: 24 hour day, or one day • **Date Range**: Lets you select a **Start date** and an **End date** from the date picker.
Start Date	A run date that allows the application to scan sections of your table, add a new date to it, and then run again. By selecting Start Date, a dynamic result is returned at runtime. Select a datetime from the date picker.
Query Preview	A preview of the automatically compiled SQL query.

> **Note** You can apply only one Row Filter from the UI. To add more filters, switch to the manual SQL query mode and modify the query.

Limit

Scans a random sample of rows based on the maximum number of rows you set. Enter a value greater than 0 or click the arrows in the **Max. rows** field to apply a limit.

Escape Character

The **Escape Character** dropdown menu allows you to optionally select the special character with which to wrap column names. This can be especially helpful if your column name contains a special character that you need to escape in order for Collibra DQ to

properly identify it. If you do not specify an escape character, then the column name is used as it appears in your table, file, or view. Available escape characters include:

- Double Quotes (`" "`)
- Single Quote (`' '`)
- Backtick (`` ` ` ``)
- Brackets (`[ ]`)
- Custom
    - When you select Custom, enter your custom escape character into the input field to the right of the Escape Character dropdown menu.

SQL View

SQL View allows advanced SQL users to write and compile SQL queries manually. SQL View enforces the same syntax restrictions as the Native SQL rule type. For more information, go to Native.

To open SQL View, click the three dots menu in the upper right corner of the Filter Type window. Click the three dots again to switch back to Standard View.

> **Important** When you include a binary date type column in your query for a Pushdown dataset, ensure that you use an alias to prevent job failure. For example, `to_varchar("EXAMPLE_COLUMN") as EXAMPLE_ALIAS`

> **Important** Transform is only available in Pullup mode. If you are using Pushdown mode, proceed to the next available step.

Transform

Transform allows you to override existing columns or create new ones based on the data in existing ones by performing the following operations:

- Convert a column to a date, string, integer, double, or long data type.
- Concatenate 2 or 3 string columns.
- Replace empty spaces in specified columns with NULL.

- Perform window functions on groups of rows.
- Add custom transform expressions.

> **Note** Transform is disabled by default.

Steps

1. On the stepper, click **Transform**.
   » The Transform step opens.
2. Select from the following options, as needed.

Option	Description
Add Collibra DQ Date Column	Adds a synthetic run date column. > **Tip** This can be helpful when your file does not contain dates that are easy to handle.
Flatten	Refers to the level of nesting in your file.
Multi-line	Refers to the content of the file being more than one line.
Zero Fill Null	Replaces NULL values with 0.
Replace all field values of [value] with NULL	Replaces all values of the column you specify with NULL.

3. To the right of the Transform screen, click **+ Add**.
   » The Add a Transform modal appears.
4. Select one of the following options and follow the steps accordingly.

    1. Click the **Override a Column** option.
    2. Click the **Operation** dropdown menu and select an option.

Operation	Description
to Date	Converts a string column to a date type column.

Operation	Description
to String	Converts a column to a string type column.
to Int	Converts a column to an integer type column.
to Double	Converts a column to a double type column.
to Long	Converts a column to a long type column.
Concat(X, Y)	Concatenates two string type columns, where "X" represents the first column and "Y" the second. Select the columns from the Column dropdown menus to the right of the Operation dropdown menu.
Concat(X, Y, Z)	Concatenates three string type columns where "X" represents the first column, "Y" the second, and "Z" the third. Select the columns from the Column dropdown menus to the right of the Operation dropdown menu.
Replace	Replaces empty spaces in a specified column with NULL.
Window Functions	Performs window functions on groups of rows. **Note** This option requires additional selections outlined in the following steps. 1. Select an option from the **Function** dropdown menu. 2. Select a column from the **Column** dropdown menu. 3. Optionally select the **Group by** checkbox, then select the column (s) to group the window function. 4. Optionally select the **Order by** checkbox, then select the column (s) in the order you want them to appear.
Custom	Adds custom transform expressions. Manually update the Expression input field after selecting a column from the Column dropdown.

3. Click the **Column** dropdown and select an option.

> **Tip** If you already selected the columns after choosing Window Functions or one of the Concat options, you can skip this step.

4. Review or manually edit the **Expression** input field, as needed.

1. Click the **Create a new Column** option.
2. Enter a column name for your new column in the input field to the right of the Create a new Column option.
3. Click the **Operation** dropdown menu and select an option.

Operation	Description
to Date	Converts a string column to a date type column.
to String	Converts a column to a string type column.
to Int	Converts a column to an integer type column.
to Double	Converts a column to a double type column.
to Long	Converts a column to a long type column.
Concat(X, Y)	Concatenates two string type columns, where "X" represents the first column and "Y" the second. Select the columns from the Column dropdown menus to the right of the Operation dropdown menu.
Concat(X, Y, Z)	Concatenates three string type columns where "X" represents the first column, "Y" the second, and "Z" the third. Select the columns from the Column dropdown menus to the right of the Operation dropdown menu.
Replace	Replaces empty spaces in a specified column with NULL.

Operation	Description
Window Functions	Performs window functions on groups of rows. **Note** This option requires additional selections outlined in the following steps. 1. Select an option from the **Function** dropdown menu. 2. Select a column from the **Column** dropdown menu. 3. Optionally select the **Group by** checkbox, then select the column (s) to group the window function. 4. Optionally select the **Order by** checkbox, then select the column (s) in the order you want them to appear.
Custom	Adds custom transform expressions. Manually update the Expression input field after selecting a column from the Column dropdown.

4. Click the **Column** dropdown and select an option.

   Tip If you already selected the columns after choosing Window Functions or one of the Concat options, you can skip this step.

5. Review or manually edit the **Expression** input field, as needed.

5. Click **Save**.
6. Optionally repeat steps 3-5 to add additional transforms.
7. Continue to the next step on the stepper or click the right arrow on the right side of the page.

Important Mapping is only available in Pullup mode and for Snowflake connections in Pushdown mode between datasets from the same Snowflake Pushdown connection. If you are using other connections in Pushdown mode, proceed to the next available step.

Mapping

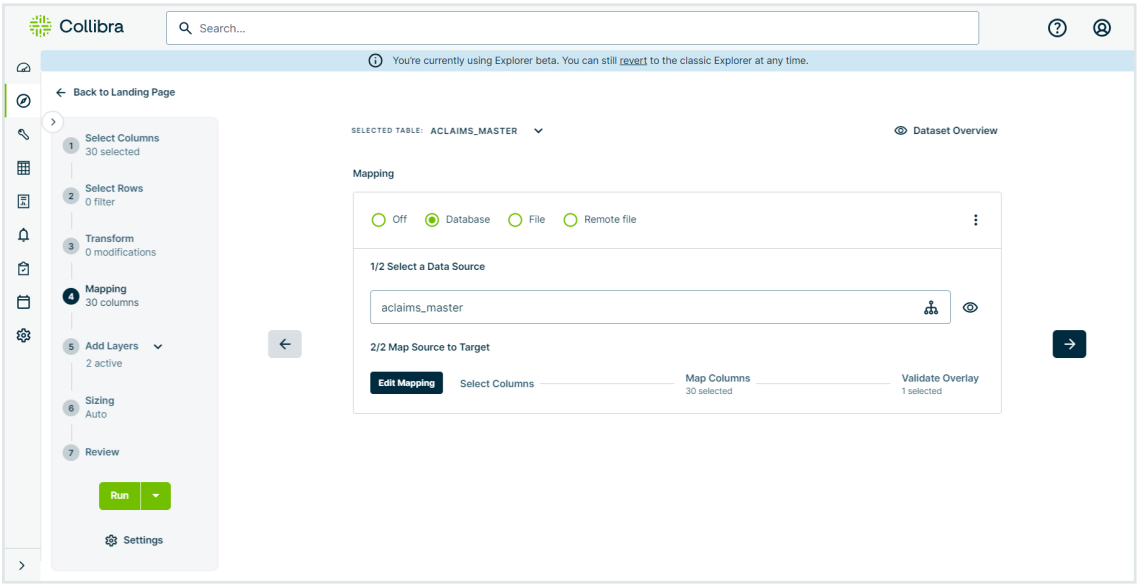
Mapping allows you to discover inconsistencies between the columns from one database, file, or remote file and those of another. Mapping checks for inconsistencies such as row count,

schema, and value differences in data copied from data sources to target tables. By mapping data from the source system to a different target system, you can protect against the two systems from getting out of sync from one another.

> **Note** Mapping is disabled by default.

Steps

1. On the stepper, click **Mapping**.
   » The Mapping step opens.
2. Select one of the following options and follow the steps accordingly.



   a. Select the **Database** option.
   b. Under the Select a Data Source section, click the **Select a resource** dropdown menu.
      » A list of available database connections appears.
   c. Select the **source database** from the list of connections to which your target columns will map for source validation.

> **Tip** You can click  to open Data Preview for a snapshot of the data you selected.

> **Note**
> An outer join is used to find discrepancies between source and target datasets and to ensure all records from both datasets are included in the resulting table, even if they don't have matching values in the other dataset. Keys are used to determine which rows should be joined together.
>
> - Target Column Missing in Source: If a selected column exists in the target dataset but not in the source, the resulting outer join table will display "null" for that column in the source dataset.
> - Source Column Missing in Target: If a selected column exists in the source dataset but not in the target, the resulting outer join table will display "null" for that column in the target dataset.
> - Mismatch Identification and Key Extraction: The extracted keys will be "null" for any source record that lacks a matching target record in the selected columns. The displayed keys are only from the target dataset.

a. Select the **File** option.
b. Under the Select File Location section, select a file protocol option from the Protocol dropdown menu, then enter the folder in the Folder input field where the file is stored. Alternatively, click the **Use Temporary file** dropdown menu to use a temp file.
c. Under the Select a Data Source section, click the **Select a resource** dropdown menu.
   » A list of available file-based connections appears.
d. Select the **source file** from the list of file-based connections to which your target columns will map for source validation.

> **Tip** You can click to open the Data Preview where you can see a preview of the data you selected.

e. Click the **Delimiter** dropdown menu and select a delimiter option reflective of how columns are separated in your file.
f. Optionally select the **Add Header** option and add comma-separated column names in the Column Names input field below.
g. Optionally select **Bypass Schema Evaluation** if Spark is unavailable.

> **Note** Bypass Schema Evaluation is a way to avoid loading a SparkSession for CSV and TXT files. You should only select this option when Spark is unavailable or you are unable to load Spark.

a. Select the **Remote file** option.
b. Under the Select a Data Source section, click the **Select a resource** dropdown menu.
   » A list of available database connections appears.
c. Select the **source remote file** from the list of file-based connections to which your target columns will map for source validation.

> **Tip** You can click to open the Data Preview where you can see a preview of the data you selected.

d. Click the **Delimiter** dropdown menu and select a delimiter option reflective of how columns are separated in your file.
e. Collibra DQ automatically detects the extension type, but you can manually select one by clicking the **Extension** dropdown and selecting an option.

> **Note** Depending on the extension type, there may be additional options that appear when you make your selection. See the table below for reference.

Option	Description	.csv	.txt	.orc	parquet	avro	json	delta	xml	hudi
Skip Lines	Add a numerical value to instruct Collibra DQ on the number of lines (rows) to skip. These are the lines in your file after the non-data information ends and the data to include in your job begins.	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No
Add Header	Denotes that the file contains a header row.	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No

Option	Description	.csv	.txt	.orc	parquet	avro	json	delta	xml	hudi
Column Names	A comma-separated list of the header column names. For example, Date,Name,Title	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No
Bypass Schema Evolution	A method to avoid loading a SparkSession for CSV and TXT files. You should only select this option when Spark is unavailable or you are unable to load Spark.	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No	✗ No
Flatten	Refers to the level of nesting in your file.	✗ No	✗ No	✗ No	✗ No	✗ No	✓ Yes	✗ No	✓ Yes	✗ No

Option	Description	.csv	.txt	.orc	parquet	avro	json	delta	xml	hudi
Multiline	Refers to the content of the file being more than one line.	✕ No	✕ No	✕ No	✕ No	✕ No	✓ Yes	✕ No	✓ Yes	✕ No

f. Click **Load Schema**.

3. Click **Start Mapping**.
   » The Column Selector modal appears. The Column Selector shows the available columns in your target dataset and includes details about the data type present in each column.

   > **Note** The Column Selector step allows advanced SQL users to optionally click the ⋮ in the upper right corner to switch to SQL View for manual SQL compilation.

4. Optionally click **Enable Timeslice** and select a **Timeslice Column** from the dropdown menu.
   » This adds a `${rd}` to your query, where the datetime value is substituted at runtime.
5. Select the columns to include in the mapping using the checkbox options next to each column name.

   > **Note** All columns are selected by default.

   > **Tip** If your source columns contain white spaces, they may not map properly to the target columns. As a workaround, you can remove the white spaces from the command line, then copy and paste the command line into a new DQ Job.

6. Click **To Mapper** in the lower right corner.
   » The Source to Target Mapper appears. The Source to Target Mapper allows you to map source columns on the left side of the screen to their targets on the right.

> **Note** Collibra DQ automaps columns it detects as matches between the source and target.

7. In the **Target Column**, click the dropdown menu within a given row and choose the columns you wish to map to the source column within the corresponding row. If the automapping that Collibra DQ suggested is correct, you can proceed to the next step without making modifications.
8. Click **Validate Overlay**.
   » The Overlay Preview appears. The Overlay Preview shows you the source to target mapping based on your selections in the previous steps. A helpful legend is available in the upper right corner of the screen.
9. Optionally select the **Key** checkbox at the top of any of the columns to assign them as key columns.
10. Click **Validate**.
    » The main Mapping screen displays the number of mapped columns in the stepper.
11. Continue to the next step on the stepper or click the right arrow on the right side of the page.

> **Tip** You can use rules with secondary datasets to improve performance while validating and writing a high number of failure records/break records to your choice of blob storage.

Mapping Configuration

You can click ⋮ in the upper right corner of the Mapping main screen to apply different configuration options. The following table shows the available options.

Option	Description
Pushdown Count	Runs `COUNT (*) FROM <database>` to limit the rows included in the mapping query to only those that meet the Pushdown Count criteria. This can increase the efficiency of your DQ job by preventing data that does not meet the criteria of Pushdown Count from being included in the Spark compute tier.

Option	Description
Check case	Checks for case sensitivity when comparing source and target schemas by creating a new mapping to instruct Collibra DQ about the case-sensitive source to target mapping.
Order	Checks for schema order consistency when comparing source and target schemas.
Check type	Checks for schema type consistency when comparing source and target schemas.
Validate values	Unavailable when Pushdown Count is enabled. Checks for cell value when comparing source and target cells. If no rows are present in either the source or target data source, the cell value comparison activity is skipped when the job runs. **Note** This requires a large amount of computational resources.
Validate for matches	Unavailable when Pushdown Count is enabled. Checks for matching instead of mismatching cell values when comparing source and target cells.
Trim	Unavailable when Pushdown Count is enabled. Removes leading and trailing spaces from strings to improve data cleanliness when comparing source and target cells.
Missing Keys	Unavailable when Pushdown Count is enabled. Shows keys values that are found in one data source but missing in the other.
Ignore Precision	Ignores precise matches between the total number of digits in numeric data type columns when comparing source and target schemas.
Strict Source Downscoring	Bases the downscore value on the number of cell mismatches across all columns. For example, if there are 20 cell mismatches, then 20 points are deducted from the overall data quality score.

Note Source queries that contain `WITH UR` are only supported by Db2 connections.

Add layers

This page shows the available DQ layers of a DQ job and how to configure them. Select from one of the options below.

About Adaptive Rules

Adaptive Rules are data quality checks that use machine learning to automatically observe and adapt to changes in numeric representations of data over time and downscore any values outside defined boundaries. When they occur, observations from Adaptive Rules are recorded under the Behavior tab of the Findings page. If you do not want Adaptive Rules findings to appear on the Findings page, deselect all Adaptive Rules options in this step.

Configuring Adaptive Rules

The following table shows details about Collibra DQ's Adaptive Rules, their sub-types and how to configure them.

> **Tip** You can prevent Collibra DQ from including specific behavioral checks by deselecting each AdaptiveRule option you wish to exclude. This does not turn off dataset profiling.

Adaptive Rules	Description	Default On
Availability	Observe changes to the row count and loading time in your table.	
Row Count	Monitor the row count change in your table. Select this option to include row count change observations in your DQ check.	✓ Yes
Loading Time	Monitor loading time changes. Select this option to include loading time change observations in your DQ check.	✕ No
Distribution	Observe the number of unique values in a table.	

Adaptive Rules	Description	Default On
Uniqueness	Monitor a column's cardinality within the range of previous DQ jobs. Select this option to include uniqueness observations in your DQ check.	✓ Yes
Completeness	Observe columns in your table containing null values or empty fields.	
Null Values	Monitor columns for null values. Select this option to include null value observations in your DQ check.	✓ Yes
Empty Values	Monitor columns for empty data. Select this option to include empty value observations in your DQ check.	✓ Yes
Conformity	Observe columns with values that fall outside of the normal range.	
Min	Monitor columns with min values outside the normal range. Select this option to include min value observations in your DQ check.	✗ No
Mean	Monitor columns with mean values outside the normal range. Select this option to include mean value observations in your DQ check.	✗ No
Max	Monitor columns with max values outside the normal range. Select this option to include max value observations in your DQ check.	✗ No
Schema	Schema checks are on by default and cannot be configured.	

Adaptive Rules	Description	Default On
Data Type Check (Integer, String, Date)	Monitors columns that shift from one data type to another.	✓ Yes
Schema Change (Add, Alter, Delete)	Monitors schema evolution, such as columns that are added or dropped from a dataset.	✓ Yes

About Shapes

Shapes are rare or inconsistent data formats in string columns that Collibra DQ can detect automatically or you can configure manually.

Configuring Shapes

Automatic shape detection

The Shapes layer is set to **Auto** by default to automatically discover possible shape issues. Auto shape detection depends on the row count of your dataset to determine the percentage an identified shape adheres to the shape format that Collibra DQ identifies as normal. Anything less than that percentage triggers a break. The following table provides a breakdown of occurrence percentage as determined by row count.

Row Count	Percentage
< 100	2.0
< 1,000	1.0
< 10,000	0.1
< 100,000	0.05

Row Count	Percentage
< 1,000,000	0.01
< 10,000,000	0.001
< 100,000,000	0.0001
Other	0.00001

Manually configuring shape detection

> **Note** With the exception of Data Shape Granular, these options are only configurable when the toggle is set to Manual.

1. Select the **Manual** option.
2. Select the option next to the column name to include that column in the Shapes layer.
   » You can select as many columns as there are available.
3. Optionally click ⋮, then click ⚙ **Configuration** for additional Shapes configuration options.
   » The **Shapes Configuration** modal appears.
   a. Optionally configure **Manual Options** and **Data Shape Granular**. The following table shows the available configuration options.

Option	Description	Default Value
Manual Options		

Option	Description	Default Value
Occurrences	Only shows shapes below the given percentage threshold. Drag the slider or enter a value in the input field to set the threshold to a value between 0.001 and 5.	0.001
Format per Column	Only shows columns with less than the given number of formats. For example, when there are 30 formats and the Format per Column value is set to 5, then only 5 columns display.	20
Character Length	Only shows Shape issues with fewer than the given number of characters. For example, if the value is set to 9, then all values less than 9 will show.	12

Option	Description	Default Value
Data Shape Granular		
Data Shape Granular	Checks the length of alphanumeric fields and checks for numbers and letters independently.	Auto **Note** The Auto setting leaves Data Shape Granular turned off by default. To apply a data shape granular check, select On.

b. Click **Save**.
   » Your configurations save and the **Shapes Configuration** modal closes.
4. Click **Save**.

Using Outliers

Outliers are values that differ significantly from the rest of the data and may indicate bad or incorrect data. Numerical outliers are detected with the IQR and box plot methods.

Adding Outliers

1. From the **Outliers** tab in the Add Layers workflow, click + **Add Outlier**.
   » The **Outlier** editor displays.

2. Select the required options.

Column	Description
☐	The column(s) used to detect outliers. You can select multiple columns for outlier detection. Required. Select one or more checkboxes next to the name of the columns to include them in Outlier analysis.
Column	The name of the column in your table or view. Click the **Assign** chip in a Key column to uniquely identify the rows of the column. > **Note** You can assign only one column as the Key column and it cannot be the column you select for outlier analysis.
Type	The type of data in a given column. This column is autofilled and cannot be edited or selected. The following is a list of the data types:

Data type	Description
Date	A date.
DOUBLE	A normal-size floating point number.
VARCHAR	A VARIABLE length string that can contain letters, numbers, and special characters.
text	A variable width character string.
numeric	Fixed precision and scale numbers.

Column	Description
Key	The key column whose rows have unique identifiers. This column provides context by grouping each column by the key column. When assigned, outliers are grouped to their assigned column when detected. Optional. Optionally click the **Assign** chip to uniquely identify the rows of a column. > **Note** You can assign only one column as the Key column and it cannot be the column you select for outlier analysis.
Date	The date or time dimension column(s) that are used to define time bins and processing intervals. > **Tip** When **Assigned**, you can configure the **Lookback** in the Outliers Configuration dialog. > **Note** You can assign only one column as the Date column.

3. Click **Save**.

> **Tip** Repeat this process to add more Outliers.

Outliers Configuration

Optionally, click to configure **Quartiles**, **Lookback**, and determine whether or not to detect for **Categorical** outliers. The following table shows the configuration options and their descriptions:

Option	Description
Quartile	

Option	Description
Quartile 1	The boundaries beyond which low (Q1) outliers are detected using the IQR formula. Use the slider or enter a value in the text box between 0.01 and 0.45.
Quartile 3	The boundaries beyond which high (Q3) outliers are detected using the IQR formula. Use the slider or enter a value in the text box between 0.55 and 0.99.
Lookback	The lookback period of your outlier scan. Visit the Lookback documentation to learn more. **Note** The options in this section are only available when Date is Assigned in the Add Outliers step.
Lookback Period	The period used to look back through the dataset using the date column. Use the slider or enter a value in the text box between 0 and 30.
Interval	The unit of measurement of the lookback period, such as DAY. Select an option from the dropdown menu.
By	Select an option from the dropdown menu.
Categorical	Select this option to detect for categorical outliers on specific string type columns.

Advanced Options

Optionally click ✎ near Advanced Options to adjust outlier detection sensitivity. The following table shows the available options.

Option	Description
Column	The name of the column in your table or view. You cannot edit this option.
Sensitivity	The outlier detection sensitivity. Click and drag the slider to adjust outlier detection sensitivity from Low to High for a given column.

Option	Description
Unit	The unit of outlier detection sensitivity. Select an option from the dropdown menu.

Warning Patterns are not yet available in Pushdown mode.

About Patterns

Patterns detect similarities among string values across columns and downscore any observations.

Configuring Patterns

1. From the **Patterns** tab in the Add Layers workflow, click + **Add Patterns**.
   » The **Patterns** editor displays.
2. Enter the required information.

Option	Description
☐	The column(s) used to detect patterns. You can select multiple columns for pattern detection. Required. Select at least one and no more than 10 checkboxes next to the name of the columns to include them in the pattern analysis.
Column	The name of the column in your table or view.
Type	The type of data in a given column. This column is auto-filled and cannot be edited or selected. Collibra DQ supports all standard data types. Collibra DQ also supports some custom data types depending on the data source, for example, Snowflake.

Option	Description
Key	The key column whose rows have unique identifiers. This column provides context by grouping each column by the key column. When assigned, patterns are grouped to their assigned column when detected. Optional. Optionally click the **Assign** chip to uniquely identify the rows of a column. > **Note** You can assign only one column as the Key column and it cannot be the column you select for pattern analysis.
Date	The date or time dimension column(s) that are used to define time bins and processing intervals. > **Tip** When **Assigned**, you can configure the **Lookback** in the Patterns Configuration dialog. > **Note** You can assign only one column as the Date column.

3. Click **Save**.
4. Repeat these steps for as many pattern detection configurations as you want to add.

Configuring lookback

Optionally, click to configure **Lookback**. When you do not configure lookback, the default lookback period is 5 days of data. The following table details the lookback configuration options:

Option	Description
Lookback	The lookback period of your patterns scan. Visit the Lookback documentation to learn more. **Note** The options in this section are only available when Date is Assigned in the Add Patterns step.
Lookback Period	The period used to look back through the dataset using the date column. Use the slider or enter a value in the text box between 0 and 30.
Interval	The unit of measurement of the lookback period, such as DAY. Select an option from the dropdown menu.

About Dupes

Dupes are values that match other existing values in columns and can be set to detect exact and fuzzy matches. Both exact and fuzzy support case-insensitive matching.

Exact matches

Exact matches detect entries that are potentially equivalent despite formatting issues. For example, Collibra DQ flags the column values John Doe, JOHN DOE, and john doe as exact match dupes of each other when Exact Match is on and Case Sensitive is off.

> **Note** If Case Sensitive is on, then the values in the previous example are not equivalent.

Fuzzy matches

Fuzzy matches detect entries that are potentially equivalent despite misspellings, common spelling variations, and more. For example, Collibra DQ flags the column values John Doe, Johnny Doe, and Johhnn Doe as fuzzy match dupes of each other. The fuzzy match algorithm complexity is O^2 and compares every two fields in the dataset.

Configuring Dupes

1. From the **Dupes** tab in the Add Layers workflow, click + **Add Dupes**.
   » The **Dupes** editor displays.
2. Enter the required information.

Option	Description
☐	Required. The column(s) used to detect dupes. You can select multiple columns for dupes detection. Select at least one and no more than 10 checkboxes next to the name of the columns to include them in the Dupes analysis.
Column	The name of the column in your table or view.
Type	The type of data in a given column. This column is auto-filled and cannot be edited or selected. Collibra DQ supports all standard data types. Collibra DQ also supports some custom data types depending on the data source, for example, Snowflake.
Case Sensitive	Detects exact duplicates that are dependent on the casing format of values. When switched on, Collibra DQ includes case sensitivity in the dupes detection and flags findings as exact matches. When switched off, Collibra DQ does not include case sensitivity in the dupes detection and flags findings as exact matches.

Option	Description
Exact Match	When switched on, Collibra DQ detects entries that may be potential matches despite formatting issues. **Note** Exact match findings are case insensitive by default. Ensure that **Case Sensitive** is switched off for Exact Match to work properly.
Dupes Score	The percentage match Collibra DQ will detect as fuzzy matches. Drag the scoring slider to define the match percentage limits.

3. Click **Save**.

Important Sizing is only available in Pullup mode. If you are using Pushdown mode, proceed to the next available step.

Sizing

As the sixth step in the Explorer workflow, Sizing provides an editable overview and estimation of the available resources in your Spark cluster required to efficiently execute your job. These resources include:

- Number of executors
- Number of executor cores
- Amount of executor memory
- Amount of driver memory
- Number of partitions

Together, these inputs allow you to scale resources to properly accommodate for the size of your job. From this step, you can also view details about your agent, the total number of cores and memory, and estimate how long your job will take to run.

Additionally, if you have the required sizing limit settings configured correctly in the Admin Console ▸ Admin Limits, Collibra DQ automatically calculates and updates the sizing recommendations for optimal performance. Allocating enough resources to your job is a

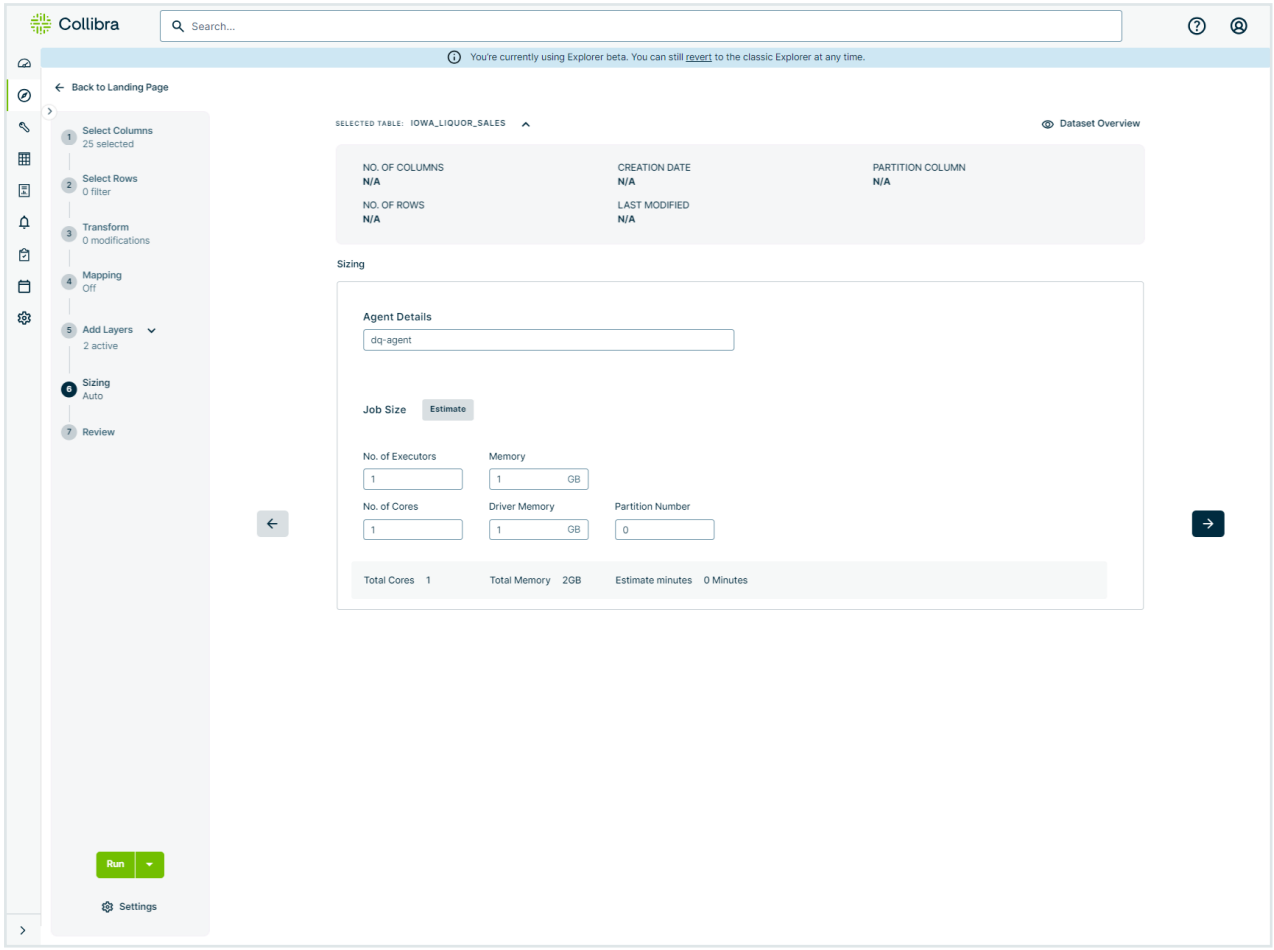
necessary step toward efficiently running jobs of all sizes, and the automatic estimator helps to simplify the calculations behind the resource estimation process.

Agent Details

The Agent Details field shows the name of the agent that will submit your job for processing when you run it. When you click the agent name, the **Agent status** dialog appears. From the Agent status dialog, you can:

- View the agent name.
- Viewthe base folder path where the agent is stored.
- View whether the agent is online or offline.
- Control which agent is selected when multiple agents are available.
    - To change the agent, click your preferred option in the Selected column.
- Upon click, an  icon displays the Agent overview of the various agent configuration details.

Job Size



The following table shows the Spark resource configurations on the Sizing step before job size estimation, corresponding with the image above.

Spark Resource	Description	Default Value
No. of Executors	The total number of Spark workers available.	1
Memory	The amount of memory allocated per Spark executor for Spark processing.	1 GB
No. of Cores	The number of cores per Spark executor.	1
Driver Memory	Total memory allocated per Spark driver.	1 GB

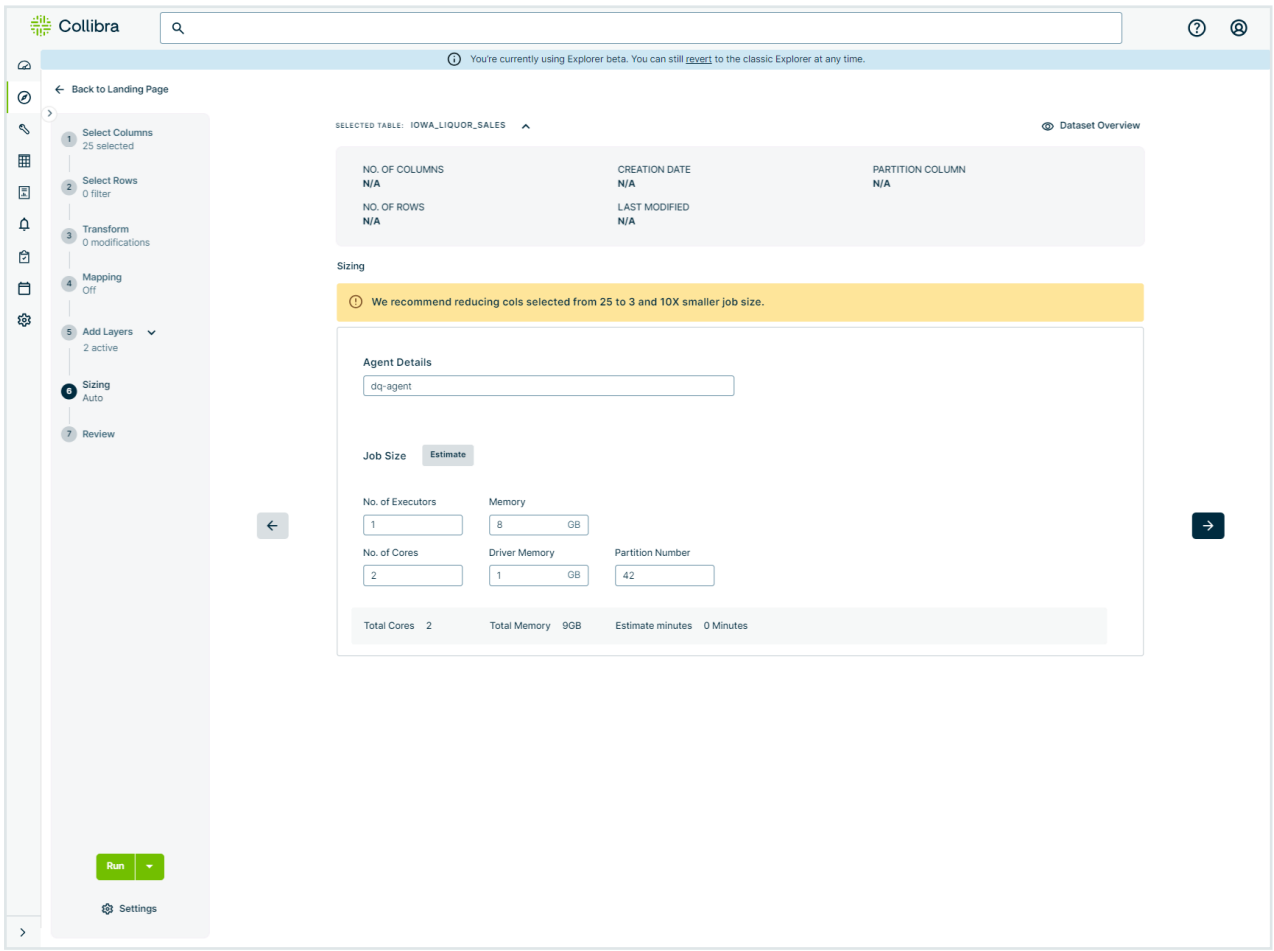
Spark Resource	Description	Default Value
Partition Number	The number of chunks of the dataset to split evenly across the Spark executors.	0

What's next?

Go to Steps to learn how to properly configure Admin Limit settings to automatically estimate your job size.

Using the job size estimator

This page shows you how to configure your environment to allow Collibra DQ to automatically estimate the sizing requirements of your job, as shown in the following screenshot of a job with updated Spark resources after job size estimation.



When you enable the automatic calculation of the sizing requirements of a job, the following parameters must be set from the Admin Console ▸ Admin Limits.

> **Note** While this is a one-time configuration, if you have a multi-tenant environment, this must be configured per tenant.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the sidebar menu
2. Click **Admin Console**.
3. Select **Configuration Settings** in the left navigation panel.
4. Select **Admin Limits**.
5. Review the configurations for the required parameters and make updates as needed.

Setting	Description	Recommended Value
partitionautocal	Enables the automatic partition calculation on the Review step in Explorer. For the automatic partition calculation to be available, this value must be set to `true`.	`true`
maxpartitions	The maximum number of partitions. You can change this depending on the performance needs of your environment.	200

Setting	Description	Recommended Value
totalworkers	The total numbers of workers available. In Spark Standalone this is the maximum number of Spark workers in your Spark cluster. In Kubernetes, this is the maximum number of VMs. In Hadoop, Dataproc, or EMR, this is the maximum number of task nodes. **Important** While the default for totalworkers is 100, we strongly advise you work with your team's internal DQ admins to determine the value required for your environment. If you are unsure of the value to set, we recommend setting this to 1.	100
maxworkercores	The maximum number of CPU cores per worker. This value is proportionate to the limit you set for totalworkers based on your environment.	2
maxworkermemory	The Agent Worker Maximum Memory (WMM) is the maximum amount of memory per worker in gigabytes. This value is proportionate to the total number of workers in your environment. **Example** An environment configuration with 8 worker cores and 4 workers would have 32 gigabytes of memory.	12
minworkermemory	The Agent Worker Minimum Memory (WMI) is the minimum amount of memory per worker in gigabytes. This value is proportionate to the maximum number of workers in your environment.	2

Setting	Description	Recommended Value
pergbpartitions	The minimum number of partitions for every 1 gigabyte of worker memory. This value will never exceed the value you set for maxpartitions. **Example** If your total memory is 10 GB and you have 6 partitions for every 1 GB of worker memory, then you would have 60 partitions in total.	6

Tip We highly recommend reviewing the Admin Limit values of **totalworkers**, **maxworkercores**, and **maxworkermemory** with a DQ admin on your internal team to determine the requirements specific to your environment. The recommended values above are based on benchmarks we've observed.

6. From Explorer, create a Pullup job.
7. In the Sizing step of the Explorer workflow, click **Estimate**.
   » The Job Size configurations automatically update to reflect the sizing requirements of your dataset.

Spark Resource	Description	Default Value
No. of Executors	The total number of Spark workers available.	1
Memory	The amount of memory allocated per Spark executor for Spark processing.	1 GB
No. of Cores	The number of cores per Spark executor.	1
Driver Memory	Total memory allocated per Spark driver.	1 GB
Partition Number	The number of chunks of the dataset to split evenly across the Spark executors.	0

> **Tip** While you can edit the application properties, we consider it best practice to use the default sizing recommendations.

8. Click **Run** or continue updating your Pullup job.

Known limitations

- There is a limitation with the estimator where it has limited ability to estimate the required resources of jobs with rules referencing secondary datasets or that have lookback enabled for outliers and patterns.
- Keep in mind that Spark allocates overhead memory on top of the requested memory. For example, an executor pod configured with 6GB of memory may actually request 6.8GB of memory. For more information, go to Apache Spark Memory Management.

Job size estimator configuration

This page illustrates for advanced client-side DQ admin users how to manually calculate Spark configurations to estimate job size.

Spark Configuration	Command Line Option	Example

Memory	`-executormemory`	To determine executor memory, you first need to determine your dataset size. To find the size of your dataset, calculate the following using inputs from your environment:
		`(number of rows * number of columns * 10) / 1024³`
		Next, determine the upper bound of maxworkermemory parameter on the Admin Limits page in the Admin Console:
		`maxworkermemory / 2`
		Lastly, set up a recursive function by dividing the dataset size by 2 until it reaches the new upper bound limit (maxworkermemory / 2).
		If the final value exceeds the upper bound, set Memory to the upper bound value. If the value is less than the upper bound, set Memory to the minworkermemory value.

No. of Cores	`-executorcores`	`(maxworkercores / max-workermemory) * executormemory = executorcores (rounded up)`

Spark Configuration	Command Line Option	Example
Partition Number	`-partitionnum`	`dataset size / 128MB = partitionnum` **Note** 128MB refers to the Spark SQL default data frame partition size. **Tip** If the value exceeds 200 based on the results of the calculation, set the Partition Number to 200.
No. of Executors	`-numexecutors`	`partitionnum / (pergbpartitions*executormemory) = numexecutors` **Warning** The number of executors should be considered a maximum limit and should never exceed 80% of the agent capacity.

Scaling your DQ Job

Storage Format	Num Rows	Num Columns	Bytes Disk	Num Executors	Executor Memory	Total RAM	Transfer Time	Process Time
Local File	1M	50	1G	1	3G	3G	0 mins	2 mins
HDFS File	10M	50	5G	3	8G	24G	0 mins	4 mins
Hive Table	10M	50	5G	3	8G	24G	0 mins	4 mins

Storage Format	Num Rows	Num Columns	Bytes Disk	Num Executors	Executor Memory	Total RAM	Transfer Time	Process Time
JDBC Table	50M	50	25G	8	10G	80G	3 mins	8 mins
JDBC Table	10M	100	10G	3	12G	36G	3 mins	6 mins
JDBC Table	250M	9	10G	5	7G	35G	14 mins	15 mins
JDBC Table	250M	145	70G	17	12G	204G	28 mins	30 mins

Using a 10/1 ratio of RAM to Executors is often a good rule of thumb, another and more simple option is to turn on dynamic.allocation and allow the resources to be provided as needed on demand.

Keep in mind that Spark allocates overhead memory on top of the requested memory. For example, an executor pod configured with 6GB of memory may actually request 6.8GB of memory. For more information, go to Apache Spark Memory Management.

> **Note** Collibra DQ requires that you have a maximum of 250 columns or less.

Limit Columns

In most cases, there is a large number of columns that go unused by the business or columns that don't require checking. One of the most efficient things you can do is limit the cols using the below cmds. As a best practice Collibra DQ strongly recommends using less than 80 columns per dataset.

```
-q "select colA, colB, colC, datCol, colD from table"
// vs
-q "select * from * from table"
```

How to limit columns when using a file

```
-fq "select colA, colB, colC from dataset"
// file query using keyword dataset
```

JDBC vs Local Data

Co-Located data (local data)

It is always a good performance practice to co-locate data and processing. That doesn't mean that your tech organization chooses to do this in it's architecture and design which is why Collibra DQ accounts for both. If the data is located on the cluster that is doing the processing use options like -hive for non JDBC and native file access. Skip tuning for JDBC as moving data to the cluster first will routinely reduce 50 percent of the performance bottleneck.

JDBC

Set fetchsize 1M rows -connectionprops fetchsize=1000 5M rows -connectionprops fetchsize=5000 10M rows -connectionprops fetchsize=10000

Set DriverMemory add more memory to the driver node as it will be responsible for the initial landing of data.

```
--driver-memory 7g
```

Add Parallel JDBC

4	12926	postgres	9509	16384	owluser	OwlCore	10.142.0.7	[null]	39726	2019-05-12 19:19:48.338379+00	2019-05-12 19:19:48.594062+00	2019-05-12 19:19:48.596377+00	2019-05-12 19:19:48.59638+00	false	active	SELEC...
5	12926	postgres	9511	16384	owluser	OwlCore	10.142.0.6	[null]	40370	2019-05-12 19:19:55.142277+00	2019-05-12 19:19:55.161628+00	2019-05-12 19:19:55.170877+00	2019-05-12 19:19:55.170881+00	false	active	SELEC...
6	12926	postgres	9524	16384	owluser	OwlCore	10.142.0.6	[null]	40386	2019-05-12 19:20:01.389868+00	2019-05-12 19:20:01.398637+00	2019-05-12 19:20:01.404266+00	2019-05-12 19:20:01.404269+00	false	active	SELEC...
7	12926	postgres	9508	16384	owluser	OwlCore	10.142.0.7	[null]	39724	2019-05-12 19:19:48.104646+00	2019-05-12 19:19:48.465015+00	2019-05-12 19:19:48.466819+00	2019-05-12 19:19:48.466823+00	false	active	SELEC...
8	12926	postgres	9510	16384	owluser	OwlCore	10.142.0.7	[null]	39736	2019-05-12 19:19:49.697508+00	2019-05-12 19:19:49.857844+00	2019-05-12 19:19:49.860883+00	2019-05-12 19:19:49.860886+00	false	active	SELEC...
9	12926	postgres	9505	16384	owluser	OwlCore	10.142.0.7	[null]	39678	2019-05-12 19:19:42.343537+00	2019-05-12 19:19:42.471948+00	2019-05-12 19:20:04.358557+00	2019-05-12 19:20:04.394756+00	false	idle in ...	SELEC...
10	12926	postgres	9512	16384	owluser	OwlCore	10.142.0.6	[null]	40374	2019-05-12 19:19:56.489011+00	2019-05-12 19:19:56.498728+00	2019-05-12 19:19:56.508854+00	2019-05-12 19:19:56.508857+00	false	active	SELEC...

Limit Features, Turn Flags Off

```
-corroff    //only losing visuals, 5% speed gain
-histoff    //only losing visuals, 4% speed gain
-hootonly   //speeds up 1% based on less logging
-readonly   //remove owl webapp read writes, 1% gain
-datashapeoff //removes Shape Detection 3% speed gain
```

Real World Scenario

Nine million rows with 46 columns on a daily basis for just 1 dataset. The data lives in Greenplum and we want to process it on a cluster platform where Collibra DQ runs. The first run results in a 12 minute runtime. While acceptable it's not ideal, here is what you should do:

1. Add Parallel JDBC for faster network.
2. Limit columns to the 18 that are of use in the downstream processing.
3. Turn off unneeded features.
4. Find out of the job is memory bound or CPU bound.

By setting the below configs this same job ran in six minutes.

```
# parallel functions
-columnname run_date -numpartitions 4 \
-lowerbound "2019-02-23 00:00:00" \
-upperbound "2019-02-24 00:00:00"
# driver optimization
-connectionprops fetchsize=6000
# analyst functions
-corroff \
-histoff
# hardware
-executormemory 4g
-numexecutors 3
```

The Full DQ Check

```
./owlcheck  \
-u u -p pass  \
-c jdbc:postgresql://$host/postgres \                    # jdbc
url
-ds aumdt  -rd 2019-05-05  \
-q "select * from aum_dt"  \
-driver org.postgresql.Driver \                          # driver
-lib /home/owl/drivers/postgres  \                       # driver
jar
-connectionprops fetchsize=6000 \                        # driver
performance setting
-master yarn -deploymode client \
-executormemory 2G -numexecutors 2 -drivermemory 3g \    # hard-
ware sizing
-h cdh-edge.us-east1-b.c.owl-hadoop-cdh.internal:2181 \ # Col-
libra DQ metastore
-corroff -histoff -statsoff \                            # Col-
libra DQ features
-loglevel INFO \                                         # log
level
-columnname updt_ts -numpartitions 12 \                  # par-
allel jdbc
-lowerbound 1557623033193 -upperbound 1557623051585
```

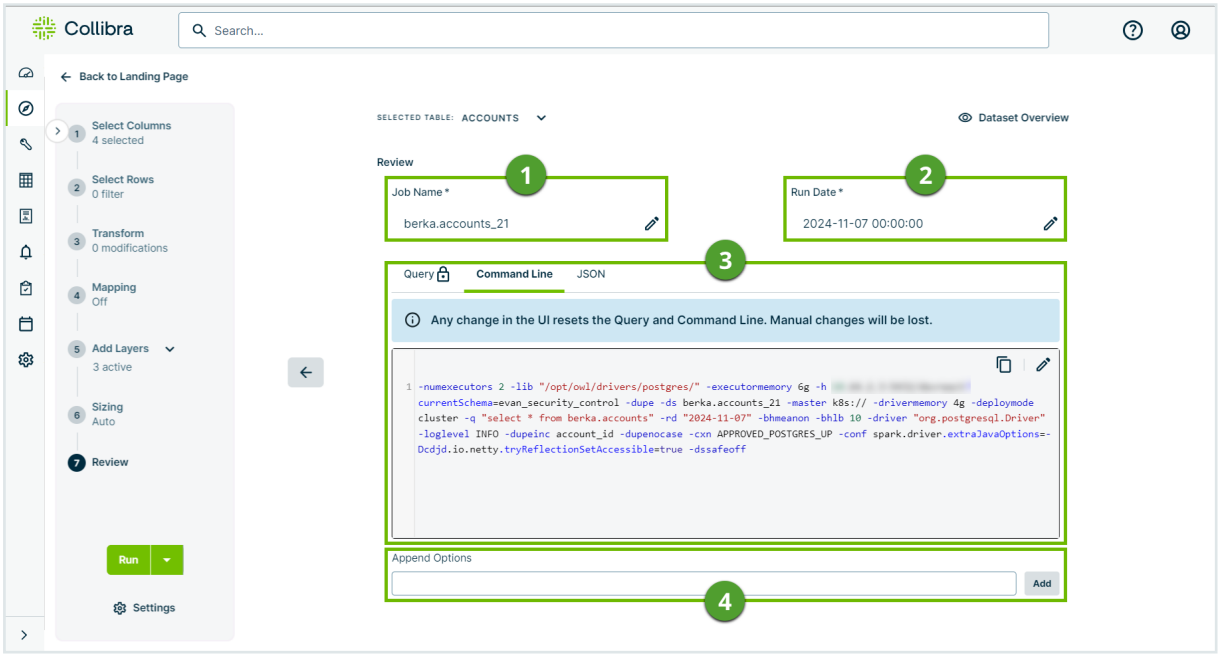
```
{
  "dataset": "aumdt",
  "runId": "2019-05-05",
  "score": 100,
  "behaviorScore": 0,
  "rows": 9000000,
  "passFail": 0,
  "peak": 0,
  "avgRows": 0,
  "cols": 46,
  "runTime": "00:05:23",
  }
```

Review

The Review step lets you review details of your DQ job, including Job Name, Run Date, and the SQL query. You can also edit the run command from the Command Line option when using

Pullup mode. Select from one of the options below depending on whether you are using Pullup or Pushdown mode.

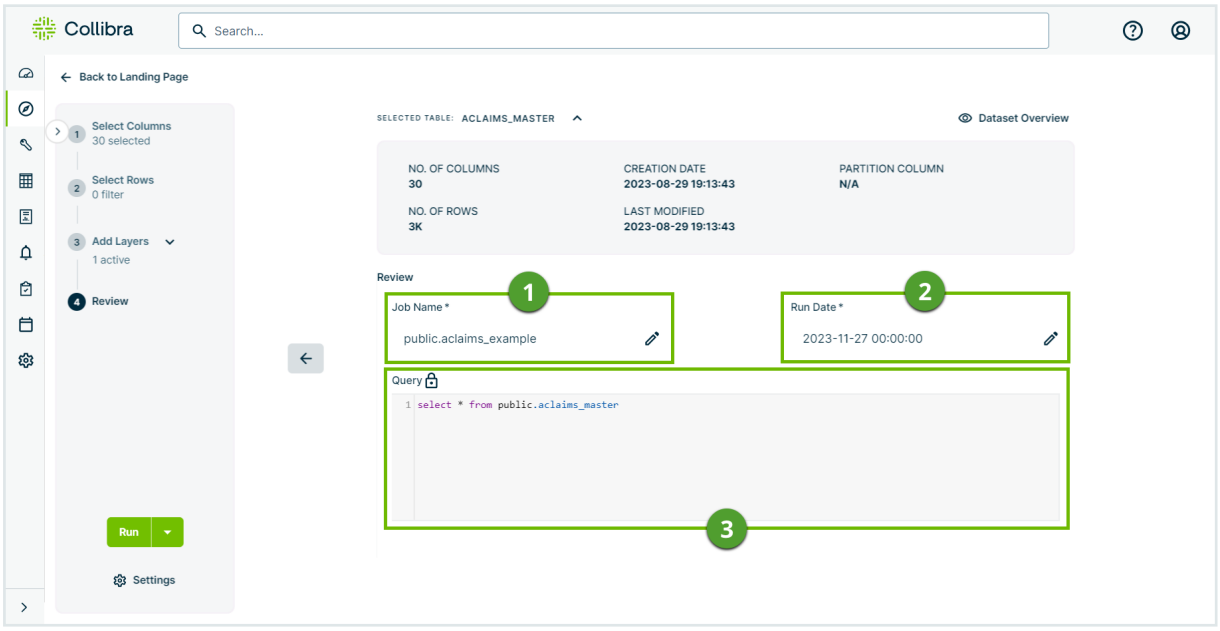
> **Note** To rename a dataset, display the dataset in Dataset Manager and use the **Rename Dataset** Action. You can also create a clone of the dataset from the Findings page by using the **Clone Dataset** option on the Job tab.



No.	Option	Description
1	Job Name	The name of your DQ Job. Job names must be unique per data source connection. For example, two jobs created on a PostgreSQL connection cannot both be named **example.job**. To avoid an error when you attempt to run the job in this scenario, you can name the second job **example.job_2** to distinguish between the two. Click the ✎ to update the name. Click the ✔ to save your changes. > **Note** You can't rename a dataset when editing an existing job. You can only rename it when creating a new dataset in the Dataset Manager.
2	Run Date	The datetime that is substituted in your timeslice filter. Click the ✎ to update the name. Click the 🗔 and select a date, then click and drag the dial to select a time. Click the ✔ to save your changes.
3	Query, Command Line, and JSON tabs	**Query** is the SQL query that automatically compiles according to your selections in the Select Columns, Select Rows, and Add Layers activities. > **Note** > • This is a read-only option. To make changes to the SQL query, return to a previous step and update your selections accordingly. > • The Query Compile feature is currently not supported for remote files and temp files. **Command Line** is the Spark SQL run command of your DQ Job, where the various inputs from the settings you selected in the previous steps are reflected in the query. Spaces in column names are not supported when editing the command line. **JSON** is the JSON payload that reflects the settings you specified in the previous steps. You can manually edit this and copy the output to your notebook.

No.	Option	Description
4	Append Options	Allows you to apply any additional parameters to the command line of your DQ Job. This option helps to avoid unintended consequences from manually editing the command line by passing options to it directly and validating that a given command is not already included.

> **Note** Append Options is only available on the Command Line tab.



No.	Option	Description
1	Job Name	The name of your DQ job. Job names must be unique per data source connection. For example, two jobs created on a Snowflake connection cannot both be named **example.job**. To avoid an error when you attempt to run the job in this scenario, you can name the second job **example.job_2** to distinguish between the two. Click the ✎ to update the name. Click the ✔ to save your changes. > **Note** You can't rename a dataset when editing an existing job. You can only rename it when creating a new dataset in the Dataset Manager.
2	Run Date	The datetime that is substituted in your timeslice filter. Click the ✎ to update the name. Click the 🗔 and select a date, then click and drag the dial to select a time. Click the ✔ to save your changes.
3	Query	The SQL query that automatically compiles according to your selections in the Select Columns, Select Rows, and Add Layers activities. > **Note** This is a read-only option. To make changes to the SQL query, return to a previous step and update your selections accordingly.

Run your DQ job

After you review your DQ job, you can run it. There are two options for running a DQ job:

- Run with your specified parameters.
- Run with Replay and selected back run parameters.

Running with Replay

Replay creates back runs of your DQ job for the number of days to establish a historical baseline profile of the data. When you use a Time Slice filter, Replay inserts the run date from

the time slice grouping for the number of days, months, or years you set as the Back Run value.

For example, if the Start date of the Time Slice filter is 2023-03-31 and you use Month as your Back Run Time Bin option, and set 10 as your Back Run value, then there will be 11 different runs with 11 different run dates. The first run is the entire month of March 2023 and has a run date of 2023-03-31, while the eleventh run is the entire month of May 2022 and has a run date of 2022-05-31. Every run between these months follows the same pattern where the run date is the last day of the month and the time bin is the first through last day of the month.

> **Tip** To create a back run for the last day each month, ensure your run date is the 31st of a month with 31 days in it and set the Back Run Time Bin option to "Month." With this configuration, any month without 31 days will run on its last day. For example, if September (a month with only 30 days) is included in the scope of the back run, a back run will occur on September 30.

Configuring Replay

1. Click the down arrow button next to **Run**, then click **Replay**.
   » The **Replay** modal appears.
2. Select from the available options.

Option	Description
Back Run Time Bin	The grouping of the replay. The available options are Day, Month, and Year.
Back Run	The number of days, months, or years that will replay according to the value you specify. Click and drag the slider or enter a value between 1-50 in the input field next to the slider.
Preview	Previews the Time Bin per replay run.

3. Click **Run**.

> **Note** This is an alternative run option to the default run without Replay.

What happens when a job runs?

When a job runs, Collibra DQ assigns it a Run Id and submits it to the Jobs page. When you run a job in Pullup mode, a SparkSession is created for Spark to submit the job for processing or the job's metadata is submitted directly to the data warehouse for processing. Conversely, a job running in Pushdown mode is submitted directly to its source database for processing without the need for intervention from a Spark agent.

DQ job settings

You can use the Settings modal to fine-tune your DQ job.

Using the Settings modal

Click ⚙ **Settings** below the **Run** button to open the Settings modal.

Option	Description
Profile	
Profile String Length	Ensures string-type data fits within the predefined schemas of its target data sources.
Data Analysis	
Relationship Analysis	Relationship analysis lets you discover relationships in your data and measures the strength of those relationships with a correlation matrix. Relationship Analysis is set to **Auto** by default.

Option	Description
Histogram Analysis	Segments data from your DQ Job with histograms. Histogram Analysis is set to **Auto** by default.
AdaptiveRules	
Data Lookback	The total number of prior job runs for your learning model to analyze. The lookback period determines the baseline on the Profile page. The default value is 10.
Learning Phase	The minimum number of job runs required before adaptive rule findings are generated. When the minimum number of job runs hasn't been met, the adaptive rule shows a "learning" state on the Findings page. The default value is 4.
Parallel JDBC	

Option	Description
Override Partition Column	Splits your selected column evenly for parallel JDBC Spark load execution. Select the checkbox option, then select a column from the dropdown menu. The default option when Override Partition Column is selected is OWLAUTOJDBC.

Option	Description
No. of Partitions	A partition is an even split of the total number of rows in your record. For large DQ jobs, increasing the number of partitions can improve performance and increase processing efficiency. Drag the slider or enter a value between 2 and 20 in the input field. **Example** If the row count of your table is 10 million, set the number of partitions to 10 to divide the record evenly into 10 partitioned blocks of 1 million rows. The job then executes the 10 blocks concurrently in parallel.

Data Quality Job

Option	Description
Metastore Host	The PostgreSQL metastore connection URL that determines which Metastore to use to register and record the results of your job.
Logging	Indicates the log level. Select an option from the dropdown menu. The default is Info.
Additional Lib	A directory path to include any additional drivers or jars in the classpath.

Option	Description
Union LookBack Min. Row	Indicates where to create the historical context of a scan with union lookback configured based on the number of rows in the preceding scans. If you are using union lookback, enter a value based on the number of recorded rows from previous scans. **Example** To exclude scans that recorded less than 10 rows from the historical load context, enter a value of 10.

Option	Description
Archive Breaking Records	The external storage container to which rule break records export in CSV format. Select the checkbox option, then select an archive location from the dropdown menu. **Important** When archive breaking records is turned on, rule break records no longer write to the PostgreSQL metastore. **Note** For more information, see the Archive Breaking Records section.
Check header	Excludes schema findings from the results of a DQ job. This is for when your schema contains special characters in its column names.

Option	Description
Core Fetch Mode	Overrides the `-q` in the command line by adding `-corefetchmode` to the command line, which allows the core to fetch the query from the load options table.

Option	Description
Profile	
Profiling	Creates a baseline sketch of your table or file over time. Profiling is on by default.
Advanced Profile	Determines whether a string field contains various string numerics, calculates TopN, BottomN, and TopN Shapes, and detects the scale and precision of double fields.
Data Analysis	

Option	Description
Relationship Analysis	Lets you discover relationships in your data and measures the strength of those relationships with a correlation matrix. Relationship Analysis is set to **Auto** by default.
Histogram Analysis	Segments data from your DQ Job with histograms. Histogram Analysis is set to **Auto** by default.
AdaptiveRules	
Data Lookback	The number of past DQ Job runs for your learning model to analyze. The lookback period determines the baseline on the Profile page. The default value is 10.
Learning Phase	The minimum number of job runs required before adaptive rule findings are generated. When the minimum number of job runs hasn't been met, the adaptive rule shows a "learning" state on the Findings page. The default value is 4.

Archive Break Records

Option	Description
Data Preview from Source	Prevents data preview records from storing on the PostgreSQL Metastore. When you select this option: • All data preview records are removed from the PostgreSQL Metastore and everything else remains in your data source. • When you view data preview records in the web application, the state of the records reflects how they currently appear in your data source. This option strengthens security by completely removing sensitive data from the PostgreSQL Metastore.
Archive Dupes Break Records	Allows the storage of dupe break records to the source system instead of the PostgreSQL Metastore.
Archive Outliers Break Records	Allows the storage of outlier break records to the source system instead of the PostgreSQL Metastore.

Option	Description
Archive Rules Break Records	Allows the storage of rule break records to the source system instead of the PostgreSQL Metastore.
Archive Shapes Break Records	Allows the storage of shapes break records to the source system instead of the PostgreSQL Metastore.
Source Output Schema	An alternative destination schema to create tables for break records storage instead of the schema provided in the connection. This can be either the database.schema or the schema and requires write access to the source output schema location.
Logging	
SQL Logging	Switches logging for all SQL queries on and off. SQL logging for all jobs is off by default.
Pushdown	

Option	Description
No. of Connections	The maximum number of connections to the data source to run your DQ Job. Using multiple connections lets your DQ Job execute queries in parallel. The default value is 10. This default value is determined by the **maxconcurrentjobs** limit on the Admin Limits page. If you use the default of 10, 10 DQ Jobs opens 100 connections (10 DQ Job Jobs multiplied by 10 connections) to the data source. Because the maximum number of parallel connections varies depending on the data source, you should account for the specific constraints of your data source when setting the number of connections. When setting the number of available connections, it's important to consider the number of open connections for your data source and the number of DQ Jobs you intend to run in parallel.

Option	Description
	Tip When running multiple DQ Jobs in parallel and you have a large data warehouse, you can increase the number of available connections for the DQ Job to which you wish to grant processing priority in the warehouse. Other scenarios where you may consider increasing the number of connections include: • The Job log shows that a particular activity is taking an unusually long amount of time to process. • A DQ Job fails with an exception message referring to the failure of a particular DQ layer, such as outliers. • Memory consumption is peaking in the data source. • You are not getting the maximum throughput of your data warehouse. • Your data

Option	Description
No. of Threads	The maximum number of threads a DQ layer can run in parallel. Use this parameter to divide the number of open connections between DQ layers. For instance, when a DQ Job has 10 connections and 2 DQ layers to process, you can set the number of threads to 5 to distribute the Job load evenly. The default value is 2. **Tip** You can consider adjusting the number of threads based on the total DQ layers in your DQ Job. For example, if your DQ Job includes checks for duplicates, outliers, rules, and shapes, increasing the number of available threads may improve the run time.

Run Date

Option	Description
Date Format	The run date format (`${rd}` or `${rdEnd}`) substituted on the command line at runtime. The option you select should match the date or datetime format of the timeslice column that you specified in the Time Slice option of the Select rows step.

Parallel JDBC

Parallel JDBC is a Spark activity whereby Spark uses a specified column in your dataset to evenly partitions data workloads across worker nodes in a Spark Cluster. With Parallel JDBC enabled, the default OWLAUTOJDBC setting instructs Spark to create two partitions. However, you can optionally override the default partition column to set between 2 and 20 partitions on a specific column. When you run a job with Parallel JDBC, all tasks in your workload execute concurrently and in parallel across each partition.

If the size of your dataset (total rows multiplied by total columns) exceeds `maxcellcountforparalleljdbc`, we recommend enabling Parallel JDBC to evenly split the workload for optimal performance. It's important to consider the volume of data when you set the number of partitions. For instance, larger datasets may benefit from more partitions in order to efficiently parallelize the data reading process.

The following data sources support Parallel JDBC for Pullup jobs.

- Amazon Athena
- Amazon Redshift
- Db2
- Dremio
- Hive
- Impala

- MySQL
- Oracle
- PostgreSQL
- Presto
- Qubole
- Snowflake
- SQL Server
- Teradata

Prerequisites

Before you use Parallel JDBC, an Admin (`ROLE_ADMIN`) user needs to set `maxcellcountforparalleljdbc` in **Admin Console** ▸ **Admin Limits**. We recommend setting this to any value likely to exceed your organization's maximum dataset size.

Using Parallel JDBC

Steps

1. From Explorer connect to a data source, then select a scanning method.

   > **Tip** Once you select a scanning method, you can follow steps 3 through 5 at any time during the Explorer process before you run your job.

2. Click ⚙ **Settings** below the **Run** button to open the Settings modal.
   » The **Settings** modal appears.
3. Click the switch next to **Parallel JDBC** to enable Parallel JDBC.
   » Override Partition Column and No. of Partitions appear.
4. Optionally select the Override Partition Column option, then adjust the following settings.

Setting	Description
Override Partition Column	Splits the column you select into even segments for Parallel JDBC Spark load execution. Select the checkbox option, then select a column from the dropdown menu. **Note** The default option when Override Partition Column is selected is OWLAUTOJDBC. When identifying a column to set as the partition column, it's important to choose one with a good distribution of values to avoid data skew. After you choose a partition column, you should also set the `lowerbound` and `upperbound` command line options according to the range of values within the column.
No. of Partitions	A partition is an even split of the total number of rows in your record. For large DQ jobs, increasing the number of partitions can improve performance and increase processing efficiency. Additionally, if your data is unevenly distributed you may need to increase the number of partitions accordingly in order to avoid data skew. Drag the slider or enter a value between 2 and 20 in the input field. The default value is 10. **Example** If the row count of your table is 10 million, set the number of partitions to 10 to divide the record evenly into 10 partitioned blocks of 1 million rows. The job then executes the 10 blocks concurrently in parallel.

5. Click **Save**.

Note While your settings update when you click Save, the command line might not automatically reflect your settings until you navigate to another step in the Explorer workflow from and then return to the Review step.

Command line options

Option	Description
`-numpartitions`	Overrides the number of partitions of the Spark loading configuration and sets the number of partitions for Parallel JDBC. The No. of Partitions setting in the Collibra DQ application controls this flag, and can only be used when `columnname` is set to a specific column, not to the default `OWLAUTOJDBC`. If you don't explicitly set this option, the Collibra DQ core internal logic automatically calculates the number of partitions.
`-columnname`	Specifies which column to partition for Parallel JDBC loading. Set this column to one with a good distribution of values to avoid data skew.
`-upperbound`	Overrides the upper bound of the Spark loading configuration. This value should be greater than the largest value in the partition column (`columnname`).
`-lowerbound`	Overrides the lower bound of the Spark loading configuration. This value should be less than the smallest value in the partition column (`columnname`).

Known limitations

- Parallel JDBC is not available for jobs created in Pushdown mode.
- Parallel JDBC does not work when there are no rows in your dataset.
- Parallel JDBC does not support dataset aliases in SQL queries.

Archiving break records from Pullup jobs

The ability to archive break records allows you to automatically export CSV files containing the break records of DQ jobs on JDBC connections to external storage containers, such as Amazon S3 buckets. By offloading break records to cloud storage, you have more control over how you manage the results of your DQ jobs, and you can store data in the supported remote connection of your choice. Additionally, this relieves potential capacity issues of the PostgreSQL Metastore where, when jobs with link IDs back to their source return break

records, the large amount of data included in the records can overload the PostgreSQL Metastore rule_breaks table and lead to database crashes.

You can send break records to the following remote file storage providers:

- Amazon S3
- ADLS
- Azure Blob
- Google Cloud Storage (GCS)

Important When archive breaking records is turned on, rule break records no longer write to the PostgreSQL Metastore.

Note When you leverage the Spark 3.2.2 standalone installation that comes with the AWS marketplace installation, be aware that the existing hadoop-aws-3.2.1.jar is incompatible with the AWS S3 Archive feature. To update the hadoop-aws-3.2.1.jar to hadoop-aws-3.3.1.jar in the spark/jars directory, obtain the file from the following link: Apache Downloads.

Contact your CS or SE for assistance if you encounter any difficulties locating the jar file.

Break record details

Pullup jobs can currently export the break records of rules to your cloud storage service in a CSV file so you can manage it outside of Data Quality & Observability Classic.

This table shows an example of the rule record metadata included in the export file.

Dataset	Run Id	Rule Name	Link Id
ORACLE_DQUSER.NYSE_001	Tue Jan 16 00:00:00 GMT 2018	SQLG_VOLUME_001	ES~|2079300.0000
ORACLE_DQUSER.NYSE_001	Tue Jan 16 00:00:00 GMT 2018	SQLG_VOLUME_001	ESE~|252600.0000
ORACLE_DQUSER.NYSE_001	Tue Jan 16 00:00:00 GMT 2018	SQLG_VOLUME_001	ESL~|410200.0000

> **Note** When there are multiple link IDs in a break record export file, they will be parsed into multiple columns.

Prerequisites

The following table shows the available external storage options and the requirements for each.

Storage option	Prerequisites
Amazon S3	• An Amazon S3 connection. • Read and write access on your Amazon S3 bucket. Minimum required bucket permissions...[1]
ADLS	• An ADLS connection. • Read and write access on your ADLS bucket.

1

```
{
    "Version": "YYYY-MM-DD",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:ListStorageLensConfigurations",
                "s3:ListAccessPointsForObjectLambda",
                "s3:GetAccessPoint",
                "s3:PutAccountPublicAccessBlock",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAllMyBuckets",
                "s3:ListAccessPoints",
                "s3:PutAccessPointPublicAccessBlock",
                "s3:ListJobs",
                "s3:PutStorageLensConfiguration",
                "s3:ListMultiRegionAccessPoints",
                "s3:CreateJob"
            ],
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::YOURS3BUCKETNAME",
                "arn:aws:s3:::YOURS3BUCKETNAME/*"
            ]
        }
    ]
}
```

Storage option	Prerequisites
Azure Blob	• An Azure Blob connection. • Read and write access on your Azure Blob bucket.
Google Cloud Storage (GCS)	• A GCS connection. • Editor and Viewer access on your Cloud Storage bucket.

Steps

1. From Explorer, connect to a data source.
2. Select a scanning method.
3. In the Select Columns step, assign a **Link ID** to a column.
4. In the lower left corner, click **Settings**.
5. Under the Data Quality Job section, select the **Archive Breaking Records** option.
6. From the Archive Breaking Records dropdown menu, select the external storage option to which break records will send.
7. Click **Save**.
8. Set up and run your DQ job.
9. When a record breaks, its metadata exports automatically to your external storage service.

> **Important** Ensure that a column is assigned as the Link ID for external break record archival to work properly.

Archiving break records from Pushdown jobs

Data Quality & Observability Classic Pushdown uses SQL queries native to the data source in order to connect directly to the source system and submit the data quality workload. This gives Collibra DQ the unique opportunity to store break records that are typically written to the PostgreSQL Metastore in a database or schema specifically allocated for Collibra DQ. By archiving break records alongside the database or schema, you can:

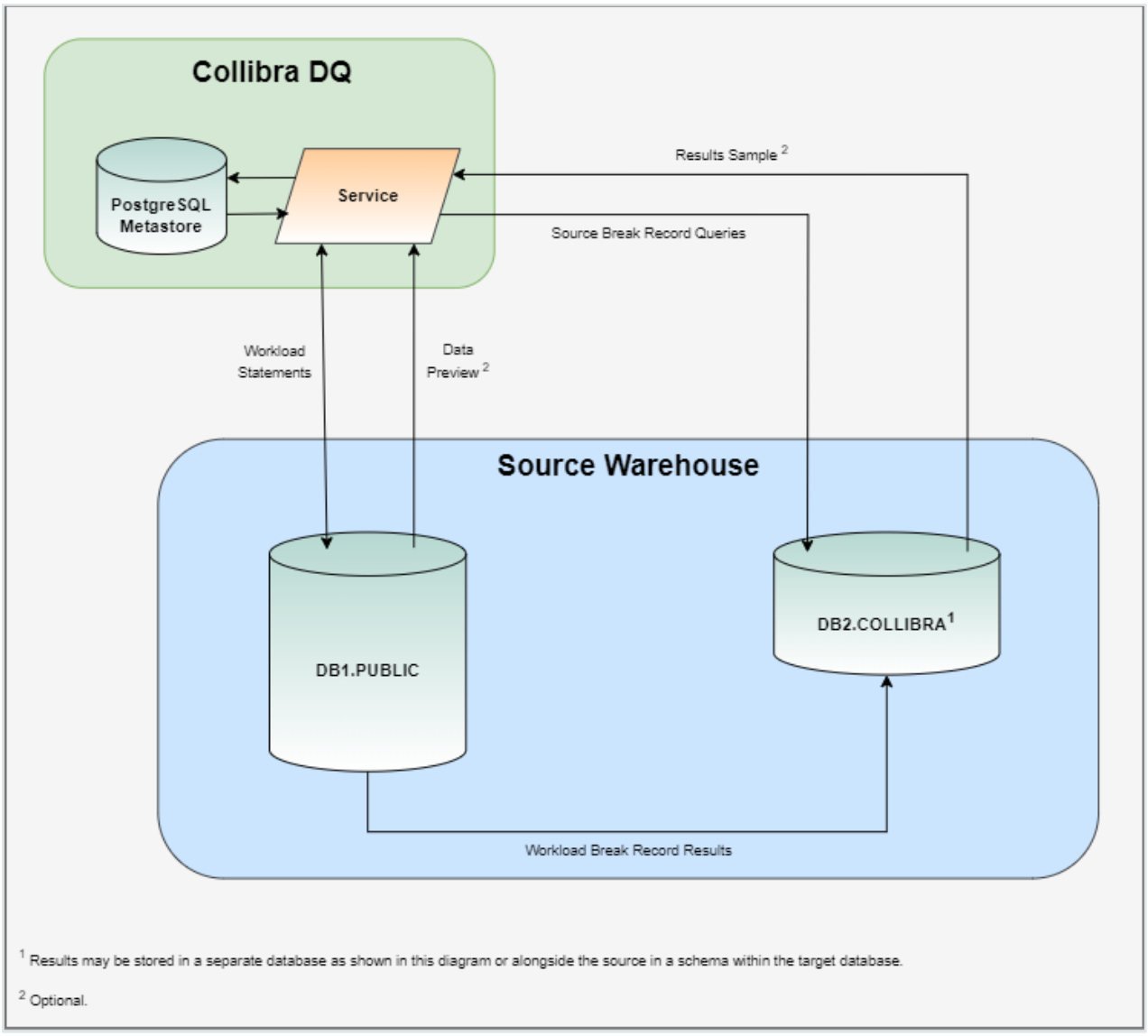
- Eliminate network traffic costs for better performance.
- Reduce storage limit restrictions.
- Improve security by generating break records that never leave the source system.

- Enhance your reporting abilities by removing the need to migrate break records from the PostgreSQL Metastore to your data source.
- Join datasets against your break records.

> **Note** When Archive Break Records is enabled, a limited number of break records are still stored in the PostgreSQL Metastore to power the DQ web application. To prevent break record storage in the Metastore, open the settings modal on the Explorer page of your dataset and turn on Data Preview from Source. This prevents records from leaving your data source and removes any others from the Metastore.

The Archive Break Records component uses a workload statement in the form of a data quality query to generate break records that get pushed into the source system for processing. Instead of returning a result set, the workload statement redirects its output to a table in your data source schema or database that is specifically allocated for Collibra DQ. The records written to the data source table contain source record linkage details that can be used to identify the source system record. These are called Link IDs and are required to write break records to the data source. Additionally, these records contain metadata generated by Collibra DQ that you may find useful when reviewing your break records.

When a workload is complete, a query is generated and stored in the data source, which allows you to easily access your break records. A results sample of break records may be optionally retrieved and used to generate a limited data preview of break records in the DQ Metastore.



Setting up break record archival

To automatically write break records to a database or schema in your data source, you first need to configure your connection's output location and enable Archive Break Records from the Settings modal on the Explorer page.

Compatible Pushdown connections

The following Pushdown connections support the archival of break records:

- Snowflake
- Databricks
- BigQuery
- Athena
- Redshift
- Trino
- SQL Server
- SAP HANA

Prerequisites

- You have a Pushdown connection.
- The service account user has write access to the database or schema to archive break records.

> **Important**
> Data sources may have different combinations of roles and permissions to allow the creation of break record tables and the ability to write records to them. Therefore, we recommend you review the connection requirements specific to your data source when you use Archive Break Records.

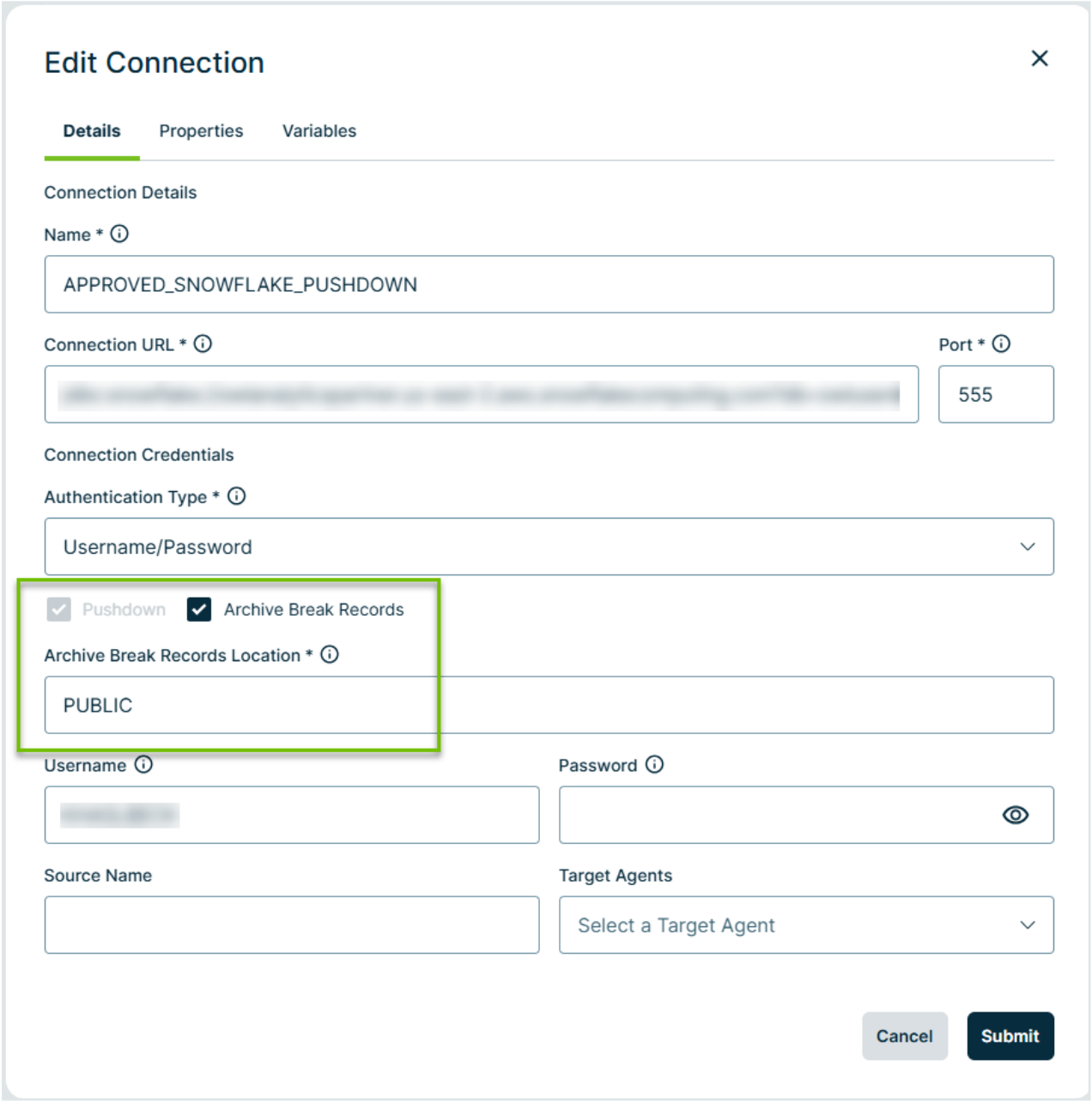
Specify the archive location for the connection

1. Open Connection Management and edit your connection.
2. In the Edit Connection dialog box, select **Archive Break Records**.
3. Specify the schema to archive break records in the **Archive Break Records Location** field. Keep in mind the following tips for selecting the schema:
    - If you enter an incorrect schema, Collibra DQ automatically uses the default output location, PUBLIC, instead.
    - Because it is optional to specify a database, if you only specify the schema, the database defaults to your current database.
    - You can enter database.schema when your schema resides outside of the current database. For example, db1.PUBLIC, db2.marketing, and db3.sales.
    - After you select the archive location, Collibra DQ creates the following tables in the database to store the break records:

- COLLIBRA_DQ_BREAKS
- COLLIBRA_DQ_DUPLICATES
- COLLIBRA_DQ_OUTLIERS
- COLLIBRA_DQ_RULES
- COLLIBRA_DQ_SHAPES

Example screenshot...[1]

1



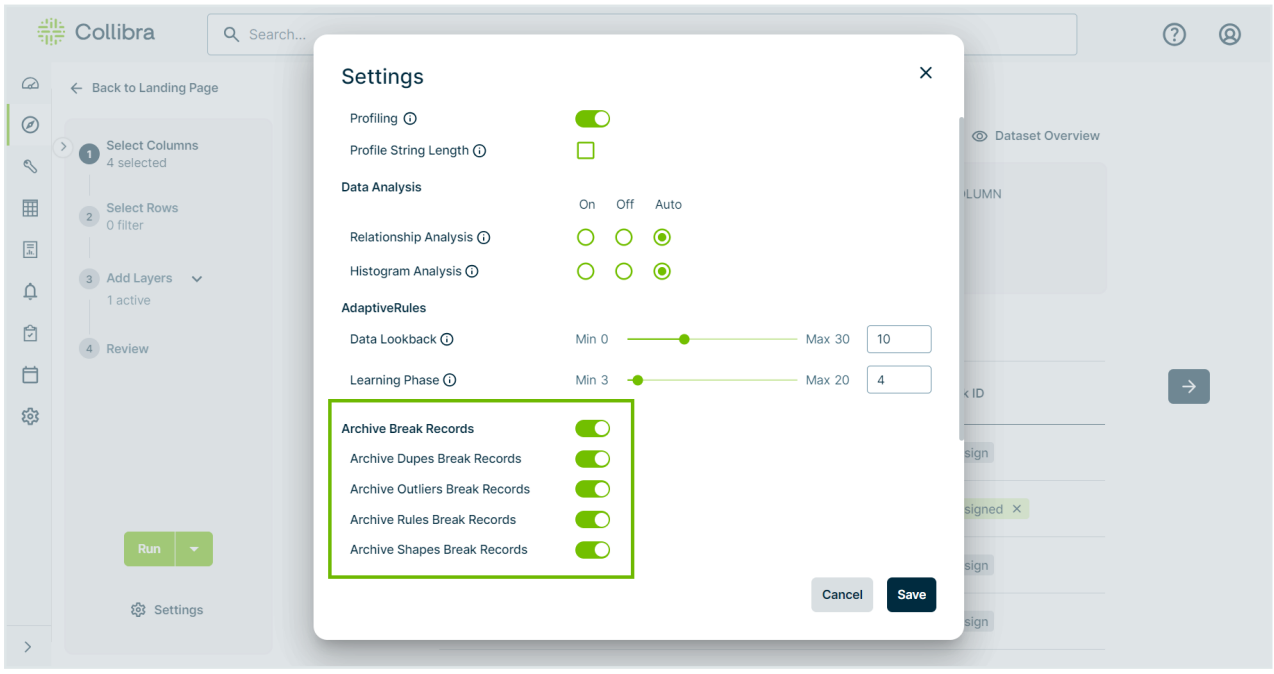
Create the job

1. Create a DQ job as described in Create a DQ Job.
2. In the Select Columns step, assign a **Link ID** to a column.

   > **Important** To get break records, it's important to specify a Link ID. The column you assign as Link ID should not contain NULL values and its values should be unique, most commonly the primary key. Composite primary key is also supported.

3. In the lower left corner, click **Settings**.
4. Select the **Archive Break Records** option.
   » By default, turning on Archive Break Records enables the export of break records for all available finding types.
   Example screenshot...[1]
5. Click **Save**.
6. Set up and run your DQ job.
7. When a record breaks, its metadata exports automatically to the data source.

1



Viewing break record details

You can archive break records from the following finding types of Pushdown jobs:

Finding type	Database table
Rules	COLLIBRA_DQ_RULES
Outliers	COLLIBRA_DQ_OUTLIERS
Shapes	COLLIBRA_DQ_SHAPES
Duplicates	COLLIBRA_DQ_DUPLICATES

The records for each of these finding types contain a unique set of metadata associated with the particular break type. When Archive Break Records is turned on, Collibra DQ sends this metadata to the break record table in the data source.

Rules

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
job_uuid	uuid	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset.
rule_name	varchar	NOT NULL	The name of the rule.
exception	varchar	NULL	The exception message of a rule that is neither Passing nor Breaking.

Column name	Type	Default	Description
link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.

Outliers

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
job_uuid	uuid	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset.
id	int	NOT NULL	The job ID associated with the break record.
key_column	varchar	NULL	When a key column is assigned, this is the name of that column.
key_value	varchar	NULL	When a key column is assigned, this is the value in that column.

Column name	Type	Default	Description
value_column	varchar	NOT NULL	The name of the column that contains the outlier.
value	varchar	NOT NULL	The value of the outlier finding.
date_value	date	NULL	When a date column is assigned, this is the date column value.
type	varchar	NOT NULL	Indicates whether the outlier finding is categorical or numerical.
median	varchar	NULL	The median of the value of a numerical outlier.
lb	double	NULL	The lower bound of the values of a numerical outlier.
ub	double	NULL	The upper bound of the values of a numerical outlier.
confidence	int	NOT NULL	The confidence score of the outlier finding.
is_outlier	boolean	NULL	Shows whether a finding is an outlier.
is_historical	boolean	NULL	Shows whether a finding is a historical outlier.
is_topn	boolean	NULL	Shows whether a finding is a topN value.
is_botn	boolean	NULL	Shows whether a finding is a bottomN value.

Column name	Type	Default	Description
source_date	varchar	NULL	The date when an outlier occurred.
frequency	long	NULL	The number of occurrences of an outlier.
percentile	double	NULL	The threshold that helps identify extreme outlier values.
link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.

Shapes

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
job_uuid	uuid	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset.

Column name	Type	Default	Description
column_name	varchar	NOT NULL	The name of the column with shape findings.
shape_type	varchar	NOT NULL	The data type of the shape finding. For example, *string*.
shape_value	varchar	NOT NULL	The column value of the shape finding.
shape_count	int	NOT NULL	The number of columns that conform to the shape that Collibra DQ discovered.
shape_rate	double	NOT NULL	The percentage a shape conforms to the format that Collibra DQ identifies as normal.
shape_length	int	NOT NULL	The format of the shape. For example, "xxx & xxxx" could represent the street "3rd & Main"
link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.

Duplicates

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
updts	timestamp	default	The timestamp of when the job ran.
job_uuid	varchar	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset that contains a possible duplicate.
run_id	timestamp	NOT NULL	The timestamp of when the job ran.
key_id	integer	NOT NULL	A unique identifier for each duplicate record that is unique even for similar duplicates found in other runs. Each time a job run, new key values are created for each duplicate record.
dupe1	varchar	NOT NULL	The name of the column that Collibra DQ identifies as a duplicate value of dupe2.

Column name	Type	Default	Description
dupe1_link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.
dupe2	varchar	NULL	The name of the column that Collibra DQ identifies as a duplicate value of dupe1.
dupe2_link_id	varchar	NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.
score	integer	NOT NULL	The duplicate score of the column where Collibra DQ detects a potential duplicate.

Unlike in Pullup mode, when Collibra DQ detects a duplicate value in a Pushdown job with Link IDs assigned, all columns included in the source query of the row where the break is detected display on the Findings page. In Pullup mode, only Link ID columns display on the Findings page when a duplicate value is detected, because the data_preview Metastore table only stores Link ID columns.

Limitations

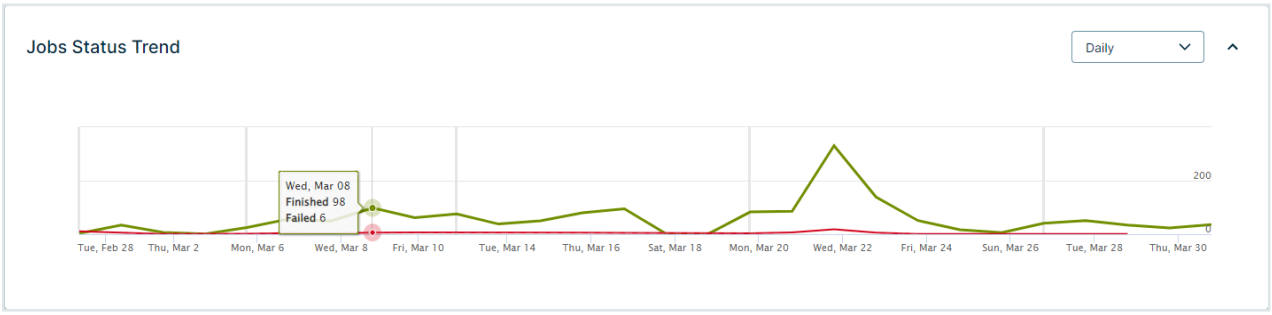
- Rule break record archival for complex rules, such as those with secondary datasets, is not supported. Rules that reference secondary datasets cannot share column names, because Collibra DQ cannot programmatically identify the table from which these columns come.
    - A workaround is to avoid tables with these shared column names or alias on the columns.

Managing the Jobs page

The Jobs page provides details about all of the DQ Jobs with attempts to run. When a DQ Job runs, this is typically the first place you should check to track its status.

Jobs Status Trend

The Job Status Trend is a graph that shows you details about the health of your DQ Job submission statuses over time. You can select the filter in the upper-right side of the graph to control the time period that displays.



Managing the Jobs table

The Jobs table is a list of all of the DQ Jobs with attempts to run. The Jobs table shows run details about the Job and lets you click a dataset to view its findings.

You can search for individual Jobs, filter the Jobs table by Job Status, and apply a Limit to control the number of past DQ Job runs that the list includes.

The following table shows the columns of the Jobs table.

Column	Description
Job Id	The automatically generated number that identifies a DQ Job as unique. You cannot edit the Job Id.
Agent Id	The identification number of your agent. **Note** Because Pushdown Jobs do not require an agent to process the Job, the Agent Id is always 0.
Dataset	The name of your dataset.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Status</td><td>The completion status of your DQ Job. Use the Job Status filter to filter by statuses as they appear in this column.

These statuses include:

<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td>Running</td><td>DQ Jobs that are currently processing the various runtime activities.</td></tr>
<tr><td>Finished</td><td>DQ Jobs that have completed all activities and whose results are available.</td></tr>
<tr><td>Failed</td><td>DQ Jobs that have not completed due to a critical event in one of the runtime activities.</td></tr>
<tr><td>Submitted</td><td>DQ Jobs that are in the Staged (to run) activity.

While this typically means that the DQ Job will run once its turn in the Jobs queue arrives, it may also indicate a problem with the agent or another exception if it takes an abnormally long time to progress beyond Staged. If this occurs, check the Exception column and Job logs for any unusual events, such as an exception or a fatal error.

Common unusual events include:

• Insufficient resources allocated to the compute pool preventing the agent from processing the Spark job.
• An incorrect or incomplete DQ Job query.
• One of the data models is using too many workers to run a part of the DQ Job. This is especially common for complex rules, including @t1 rules.

If the issue is Job-related, we recommend editing the DQ Job and reducing its scope. For example, you can limit the number of rows included in the DQ Job.

If the issue is related to infrastructural processing elements, we recommend allocating more memory or threads to the agent to allow the DQ Job to run more efficiently.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td>Unknown</td><td>DQ Jobs whose statuses are not explicitly defined by Data Quality & Observability Classic. This can include DQ Jobs that timed out because a given runtime activity took too long to process, failed to load a table, failed to connect to an offline or misconfigured driver pod. or DQ Jobs whose agents have crashed.

Common unusual events include:

• Insufficient resources allocated to the compute pool preventing the agent from processing the Spark job.
• An incorrect or incomplete DQ Job query.
• One of the data models is using too many workers to run a part of the DQ Job. This is especially common for complex rules, including @t1 rules.

If the issue is Job-related, we recommend editing the DQ Job and reducing its scope. For example, you can limit the number of rows included in the DQ Job.

If the issue is related to infrastructural processing elements, we recommend allocating more memory or threads to the agent to allow the DQ Job to run more efficiently.</td></tr>
</table>
</td></tr>
<tr><td>Activity</td><td>The task that the system performs while it processes the DQ Job.</td></tr>
<tr><td>Start Time</td><td>The timestamp when the DQ Job starts to run. The format is YYYY-MM-DD HH:MM:SS.</td></tr>
<tr><td>Update Ts</td><td>The timestamp of the last processing activity. The format is YYYY-MM-DD HH:MM:SS.</td></tr>
<tr><td>Job Exception</td><td>When an error or exception occurs, a helpful description of the event displays in this column.</td></tr>
<tr><td>Username</td><td>The username of the person in your organization who ran the DQ Job.</td></tr>
<tr><td>Actions</td><td>The action you can take to troubleshoot or terminate your DQ Job.</td></tr>
</table>

> **Tip** Click ↻ next to the Limit filter to refresh the Jobs table. Sometimes this is helpful to monitor the activity changes and track the status of a DQ Job.

Exporting details from the Jobs page

You can click ⇪ to export the details of the Jobs table to the following formats:

- CSV
- Excel
- PDF
- Print

Logging

When you encounter an issue in Data Quality & Observability Classic, the log file and logs within the UI can provide information about events and help you better diagnose any problems. It is a best practice to check the log files first

Whenever you encounter something wrong within the web application, it is good practice to check the log file first.

When you report a problem to Collibra Support, it is recommended to attach the log file(s) to your support ticket.

Depending on the log and log stage, Collibra DQ logs can show you:

- Errors that have occurred on the server or agent side.
- Basic information about tasks that are performed.
- Detailed information about every action that has been performed.
- The date and time an event occurred.
- The thread that is executing an action.
- The log level of the message.

You can find the log files for Collibra DQ under **<INSTALL_PATH>/log/**.

View DQ Job log files

This section describes all of the log files available in the log directory.

> **Note** You can find all Collibra DQ logs under **<INSTALL_PATH>/log/**.

Log	Description
dq-web.log	This log file contains Web UI issues and error messages.
dq-agent.log	This file contains information about errors that occur when the Job is with the Agent. This is when Jobs are submitted to the Spark agent for processing.
dq-core.log	This file contains information about issues that occur after the Job returns to Collibra DQ after Spark submission. This log can include errors when processing drivers or any other information within the DQ Core file.

Log	Description
3rd Party Logs	These logs contain information about resource management or any 3rd party errors that pertain to DQ. These logs can include Yarn resource logs, Spark history logs, and any resource or historical logs from external data sources.
Job Logs	These logs contain the Stage 2 and Stage 3 logs and can also be accessed from the UI on Jobs page.

> **Note**
> Stage 2 and 3 logs are only applicable when Jobs submit to a cluster. If your Jobs are not submitted to a cluster, you can find your logs locally.

Collibra DQ Log Stages

This section outlines the three-stage lifecycle of a DQ Job and shows you where to find the logs for each stage.

Stage 1 logs

In Stage 1, the Agent picks up the Job from the Metastore and translates it into a valid Spark Submit request. This includes credential acquisition and injection for Cloud and Kerberos. If a job never makes it out of STAGING, the first thing to do is to check the Agent logs (**<INSTALL_HOME>/log/agent.log** or on K8s `kubectl logs <agent-pod-name> -n <namespace>`.

Stage 2 logs

Stage 2 logs contain information about particular Jobs that fail due to issues with the Agent.

In this stage, the Agent uses a Spark Submit request to pass the DQ Job to Spark. With the Job under Spark's guidance outside of Collibra DQ, the Job is not yet currently running. Spark Submit creates its own JVM to manage each submission as part of the Spark Job to the cluster or runtime. When Jobs fail at this stage, error messages such as "Failed with reason Null" appear on the Jobs page.

You can find these logs under **<INSTALL_HOME>/log/<name-of-job>.log** or from the Actions drop-down list on the Jobs page.

Stage 3 logs

Stage 3 logs contain information about particular Jobs that fail due to issues within DQ Core.

In this stage, Spark Submit instantiates the Job in the target Spark Runtime (Hadoop/K8s/Spark-Master). With the DQ Core code active, the Job is back inside Collibra DQ. Typically, when Jobs make it to this stage, they are no longer in STAGING status and any error messages appear on the Jobs page.

You can find these logs directly from the Spark Runtime, for example, Hadoop Resource Manager, K8s API via Kubectl, and so on, or from the Actions drop-down list on the Jobs page.

View Job logs in the DQ user interface

You can view the logs for individual DQ Jobs from the Jobs page. This section shows you how to access the logs from the DQ UI.

In Stage 2, Collibra DQ submits the DQ Job via Spark Submit to Spark for processing. Jobs that fail in this stage are generally due to issues with the Agent.

You can access the Stage 2 logs from the DQ UI with the following steps:

1. Go to the **Jobs** page.
2. Click **Job Actions** on the same line as your Job ID.
   » The Job Actions drop-down list expands.
3. Select **Stage 2 Logs** from the options.
   » The Stage 2 Logs open in a new window.

> **Note** Stage 2 logs are only applicable when Jobs submit to a cluster.

Unlike the Stage 2 logs, the DQ Core code is active and the Job is no longer under Spark's guidance, but back inside Collibra DQ. Because the Job runs in DQ Core at this point, the Stage 3 log is necessary to troubleshoot problems that happen within DQ Core.

You can access the Stage 3 logs from the DQ UI with the following steps:

1. Go to the **Jobs** page.
2. Click **Job Actions** on the same line as your Job ID.
   » The Job Actions drop-down list expands.
3. Select **Stage 3 Logs** from the options.
   » The Stage 3 Logs open in a new window.

> **Note** Stage 3 logs are only applicable when Jobs submit to a cluster.

> **Tip**
> Stage 1 logs are not accessible from the DQ UI. If a DQ Job is stuck in **Staging** status, check the Agent logs from the agent log file (**<INSTALL_HOME>/log/agent.log**) or K8s kubectl logs (`<agent-pod-name> -n <namespace>`)

Retrieve Stage 2 and 3 Logs Using a Managed Hadoop Spark Cluster

When you are using a managed solution, such as AWS EMR, GCP Dataproc, etc., for Collibra DQ, you need to make certain files available to the Agent.

Where your Agent is installed, review the `owl-env.sh` file and find the value for the variable HADOOP_CONF_DIR and navigate to that path.

If the path does not exist or if that directory does not contain the `core-site.xml`, `hdfs-site.xml`, and `yarn-site.xml` files, it's possible that you need to do one of the following:

- Update the HADOOP_CONF_DIR directory within the owl-env.sh, enabling the process to point to the folder where these files exist.
- Move these XML files to the directory that already contains the files.

Typically for a typical Hadoop installation, this is under/etc/:
https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html#Configuring_Environment_of_Hadoop_Daemons

For the `yarn-site.xml`, the following configurations must be present:

```
yarn.resourcemanager.scheduler.address
yarn.resourcemanager.address
yarn.resourcemanager.webapp.address
yarn.resourcemanager.webapp.https.address
yarn.resourcemanager.principal
```

For the `core-site.xml`, the following minimum configurations must be present:

```
hadoop.security.authentication
hadoop.rpc.protection
fs.defaultFS
```

For the `hdfs-site.xml`, the following minimum configurations must be present:

```
dfs.client.use.datanode.hostname
dfs.namenode.kerberos.principal
```

> **Note** The `hdfs-site.xml` is typically only needed for self-hosted/standalone Hadoop environments, which are not hosted in the Cloud. For more information, go to Configure Hadoop.

View Audit Web Logs

The Collibra DQ web app tracks and stores the most recent key function web logs in a table, which is accessible via the **Show Web Audit Logs** button in the UI.

> **Note** Currently, only the Explorer, Connections, and Rules web logs are tracked and displayed from the Show Web Audit Logs button.

Explorer Web Audit Logs

For the Explorer, Collibra DQ logs the queries that the Explorer runs for you as you proceed through the steps in the web app, as well as a limited scope of error messages. Specifically, the Explorer logs:

- When you do a preview query.
- Exceptions getting the database table preview.
- When you run a query for a DQ Job estimate.
- When you query to analyze data on the Explorer.

Connections Web Audit Logs

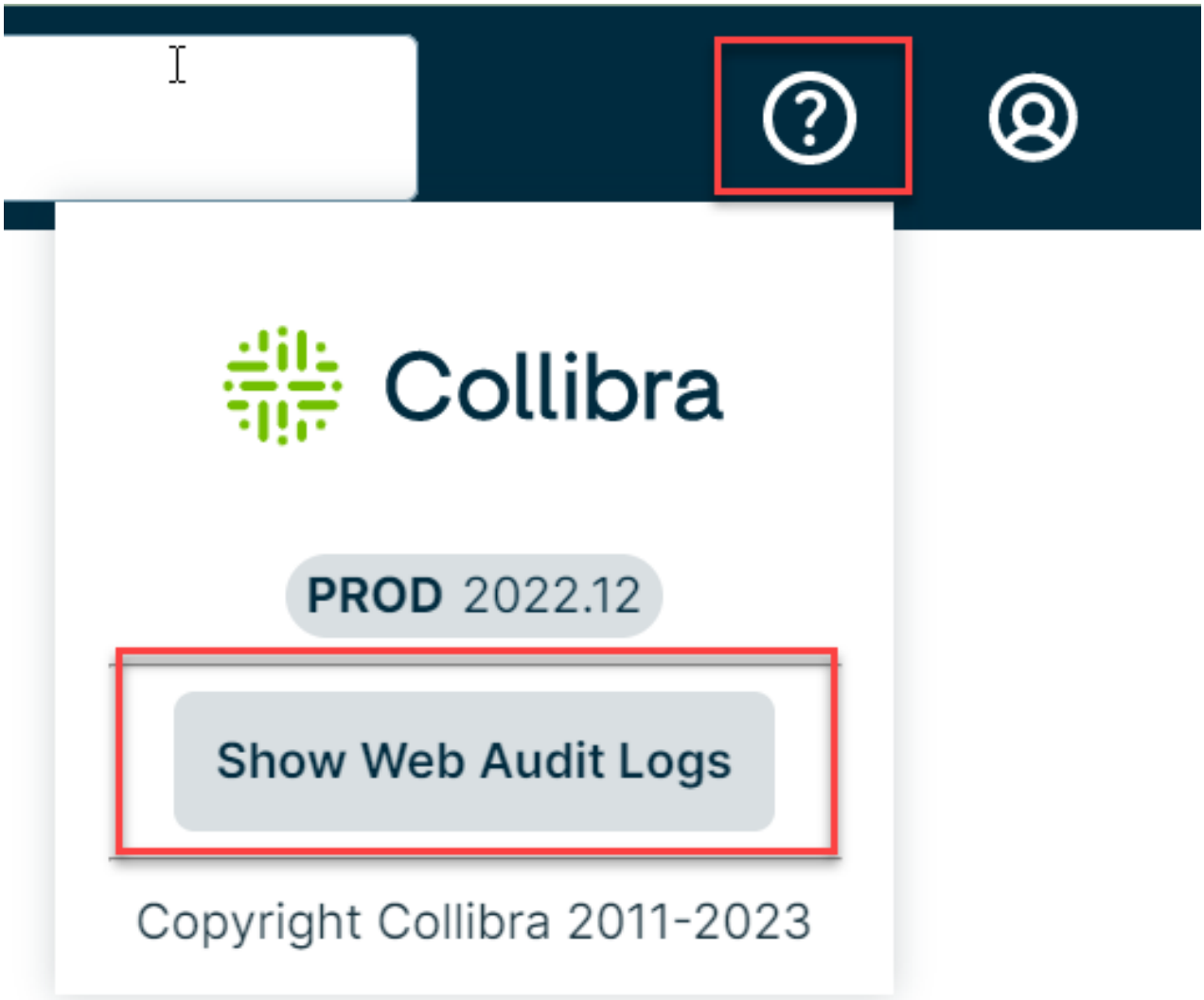
For Connections, Collibra DQ logs errors and success messages when attempting a connection.

Rules Web Audit Logs

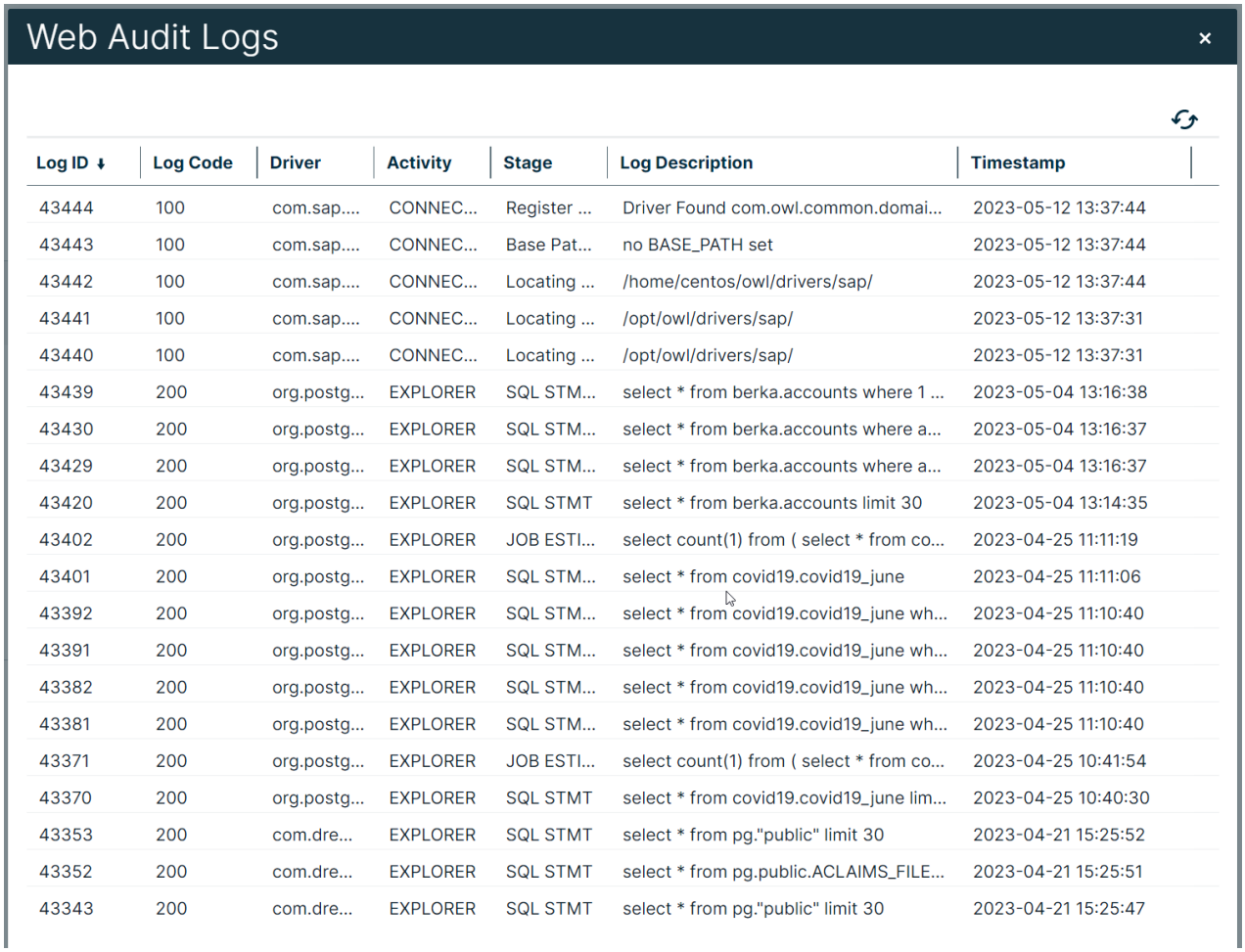
For Rules, the web app only logs the SQL produced to get the DQ Job estimate.

Steps

1. Sign into Collibra DQ.
2. Click the Help icon in the top-right of the Home page.
3. Click the **Show Web Audit Logs** button, as shown in the following screenshot.



A table displays the web audit trail logs of the key functions performed.



Log ID	Log Code	Driver	Activity	Stage	Log Description	Timestamp
43444	100	com.sap....	CONNEC...	Register ...	Driver Found com.owl.common.domai...	2023-05-12 13:37:44
43443	100	com.sap....	CONNEC...	Base Pat...	no BASE_PATH set	2023-05-12 13:37:44
43442	100	com.sap....	CONNEC...	Locating ...	/home/centos/owl/drivers/sap/	2023-05-12 13:37:44
43441	100	com.sap....	CONNEC...	Locating ...	/opt/owl/drivers/sap/	2023-05-12 13:37:31
43440	100	com.sap....	CONNEC...	Locating ...	/opt/owl/drivers/sap/	2023-05-12 13:37:31
43439	200	org.postg...	EXPLORER	SQL STM...	select * from berka.accounts where 1 ...	2023-05-04 13:16:38
43430	200	org.postg...	EXPLORER	SQL STM...	select * from berka.accounts where a...	2023-05-04 13:16:37
43429	200	org.postg...	EXPLORER	SQL STM...	select * from berka.accounts where a...	2023-05-04 13:16:37
43420	200	org.postg...	EXPLORER	SQL STMT	select * from berka.accounts limit 30	2023-05-04 13:14:35
43402	200	org.postg...	EXPLORER	JOB ESTI...	select count(1) from (select * from co...	2023-04-25 11:11:19
43401	200	org.postg...	EXPLORER	SQL STM...	select * from covid19.covid19_june	2023-04-25 11:11:06
43392	200	org.postg...	EXPLORER	SQL STM...	select * from covid19.covid19_june wh...	2023-04-25 11:10:40
43391	200	org.postg...	EXPLORER	SQL STM...	select * from covid19.covid19_june wh...	2023-04-25 11:10:40
43382	200	org.postg...	EXPLORER	SQL STM...	select * from covid19.covid19_june wh...	2023-04-25 11:10:40
43381	200	org.postg...	EXPLORER	SQL STM...	select * from covid19.covid19_june wh...	2023-04-25 11:10:40
43371	200	org.postg...	EXPLORER	JOB ESTI...	select count(1) from (select * from co...	2023-04-25 10:41:54
43370	200	org.postg...	EXPLORER	SQL STMT	select * from covid19.covid19_june lim...	2023-04-25 10:40:30
43353	200	com.dre...	EXPLORER	SQL STMT	select * from pg."public" limit 30	2023-04-21 15:25:52
43352	200	com.dre...	EXPLORER	SQL STMT	select * from pg.public.ACLAIMS_FILE...	2023-04-21 15:25:51
43343	200	com.dre...	EXPLORER	SQL STMT	select * from pg."public" limit 30	2023-04-21 15:25:47

The columns in the table display the following elements.

Column	Description
Log ID	Identification number of the log.
Log Code	Numeric code for the log.
Driver	Name of the driver.
Activity	Where the activity occurred. Options include **Explorer**, **Connections**, or **Rule**.
Stage	Stage of the activity for the log, for example, during a DQ Job estimate query.
Log Description	Column that describes the issue(s) in the log.
Timestamp	Time that the web audit log was created.

> **Tip** You can see the most recent web audit logs by clicking the  refresh icon, at the top-right of the logs table.

Convert log outputs to JSON formatting

If you want to ingest your log outputs into a third-party log consumer, such as Elasticsearch, you can convert DQ logs to JSON formatting. This page shows you how to convert DQ logs to JSON formatting in Standalone deployments of Data Quality & Observability Classic.

> **Note** This option is currently unavailable for Cloud Native deployments of Data Quality & Observability Classic.

Steps

1. Open the **dq-web** folder and then open the **log4j2.xml** file.
2. In the **<Appenders>** section, look for the following line of code:
   ```
   <PatternLayout pattern="%d{yy/MM/dd HH:mm:ss.SSS} %X{sessionId} %X{sessionHost} %X{userName} [%t] %-5level %logger{36} - %m{nolookups}%n"/>
   ```
3. Comment out the `<PatternLayout` section with the following `<JsonLayout` to convert the output from Pattern to JSON:
   ```
   <!--<PatternLayout pattern="%d{yy/MM/dd HH:mm:ss.SSS} %X{sessionId} %X{sessionHost} %X{userName} [%t] %-5level %logger{36} - %m{nolookups}%n"/>-->
   <JsonLayout objectMessageAsJsonObject="true" complete="false" eventEol="true" compact="true" properties="true" includeTimeMillis = "true"></JsonLayout>
   ```

   > **Note** You can remove the `<!-- ... -->` from the commented out line of code to revert from the JsonLayout to the PatternLayout format.

4. Set the `compact=` attribute to one of the following options:

a. `compact="true"` for compact JSON payloads.
b. `compact="false"` for decompressed JSON payloads.

5. Optionally review the other attributes in the JSON layout.

Attribute	Data Type	Description
complete	boolean	When set to *true*, the appender includes the JSON header and footer. Records are comma-delimited. The default value is *false*.
includeTimeMillis	boolean	When set to *true*, the appender includes the timeMilis attribute in the JSON payload instead of the instant. timeMillis contains the number of milliseconds since January 1, 1970 at midnight UTC.
objectMessageAsJsonObject	boolean	When set to *true*, ObjectMessage is serialized as JSON object to the message field of the output log. The default values is *false*.
compact	boolean	When set to *true*, the appender does not use end-of-lines or indentation. The default value is *false*.

Attribute	Data Type	Description
eventEol	boolean	When set to *true*, the appender appends an end-of-line after each record. The default value is *false*. To see one record per line, set both `eventEol="true"` and `compact="true"`.
properties	boolean	When set to *true*, the appender includes the thread context map in the generated JSON. The default value is *false*.
includeStacktrace	boolean	Optional. When set to *true*, the appender includes full stacktrace of any logged Throwable. The default value is *true*.

Tip You can optionally keep both the JSON and Pattern layouts by writing logs in a different file.

Findings

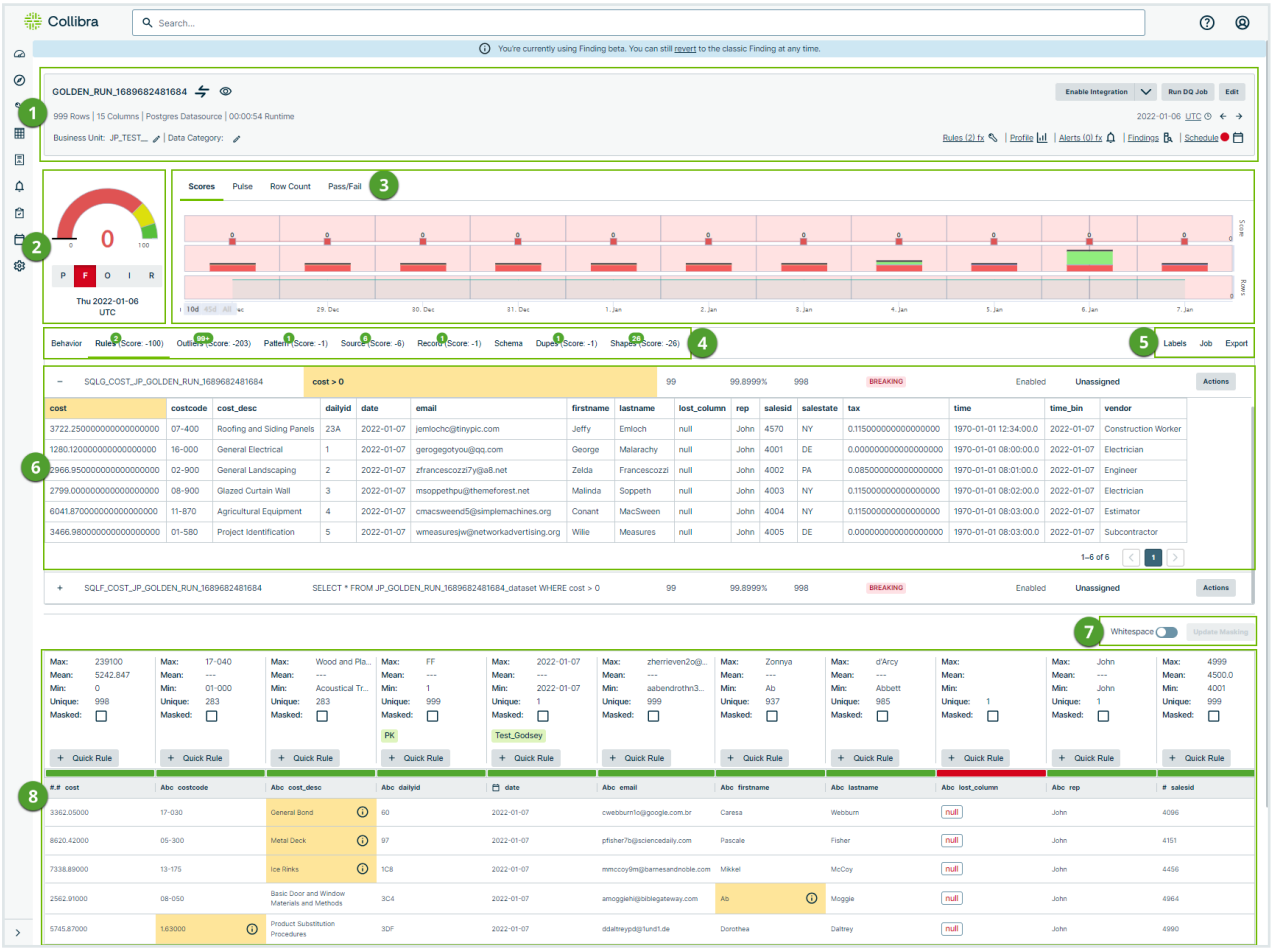
The Findings page is a dashboard that displays the results and health of a DQ job run. Findings lets you explore the details of a job run and gives you the ability to drill down into the various data quality dimensions to better understand your dataset.

Collibra DQ profiles the data and builds a model for each dataset it scans. This allows Collibra DQ to learn what *normal* means within the context of each dataset. As the data changes, the definition of normal also changes. Instead of requiring you to adjust rule settings, Collibra DQ continues to adjust its model. This approach enables Collibra DQ to provide automated,

enterprise-grade data quality coverage that removes the need to write dozens or even hundreds of rules per dataset.

Findings overview

When Collibra DQ observes a change or anomaly, it records the finding for each dimension and includes a numeric indicator next to the tab where Collibra DQ detects the issue.



The following table provides an overview of the key elements of the Findings page with numbers that correspond to the image above.

No.	Component	Description
1	Metadata bar	A dataset anchor that empowers you with the ability to access crucial data quality components related to your dataset, such as its Findings, Profile, and associated rules and alerts.

No.	Component	Description
2	Data quality score	The Data Quality score is an aggregated value between 0-100 that represents a summary of the integrity of your data. A score of 100 means that Collibra DQ does not detect data quality issues in your dataset. Conversely, a score of 0 indicates that Collibra DQ has observed enough potential data quality issues that it has down-scored the findings to the lowest possible value. **Note** The findings from each data quality dimension where Collibra DQ detects potential data quality issues contribute to the overall data quality score. Below the score meter are five clickable options.

Option	Description
P	Pass (P) indicates a job run that initially failed but is determined to be passing after further investigation of the findings. This option trains the Collibra DQ learning model to pass future runs that are similar.
F	Fail (F) indicates a job run that initially passed but was determined to be failing after further investigation of the findings. This option trains the Collibra DQ learning model to fail future runs that are similar.
O	Off-peak (O) indicates a job run on a weekend or holiday that is valid but does not align with peak time runs. This option trains Collibra DQ to label similar future runs as off peak.
I	Ignore (I) indicates a valid run that fits the run cycle profile but should not be used for training the model. This option ignores the findings of a given run.
R	Remove (R) indicates a run that triggered by mistake and should be removed.

No.	Component	Description

Option	Description
Timestamp	The runId of the current DQ Job. You can click this option to set the runId to a specific timezone. When there is only one DQ Job run on a given runId date, the timestamp is always 00:00. However, when there are multiple DQ Job runs on a given runId date, the timestamp reflects the time each DQ Job ran, for example, 01:00 or 10:45. **Note** The timestamp displays in 24-hour time format. For example, 01:00 is 1 AM, 13:00 is 1 PM, and so on.

No.	Component	Description
3	Score chart view selector	The **Scores**, **Pulse**, **Row Count**, and **Pass/Fail** tabs each provide different graphical representations of your data quality scores, run status, row count evolution, and pass or fail status over time. **Note** The times along the x-axis use random values that do not reflect or relate to the run ID of a given DQ Job run.

No.	Component	Description
4	Data quality dimension findings tabs	When Collibra DQ detects potential data quality issues for a given dimension, a numeric indicator shows how many issues were discovered and the data quality score impact aggregated from the findings. Click the tabs to reveal more details about the findings.
	Behavior	Automatically detected. Behavior is an assortment of behavioral observations based on Collibra DQ's AdaptiveRules, which automatically observe and adapt to changes in numeric representations of data over time, down-scoring any values outside the defined boundaries.
	Rules	Opt-in for detection. Rules are user-defined SQL conditions that help you identify and understand anomalies in your data, and ensure that your data is fit for use to meet your business requirements.
	Outliers	Opt-in for detection. Outliers are values that differ significantly from the rest of the data and may indicate bad or incorrect data.
	Patterns	Opt-in for detection. Patterns detect similarities among string values across columns and down-score any observations.
	Source	Opt-in for detection. Detects row count, schema, and cell value inconsistencies between the source file or table and the target file or table and down-scores any observations.
	Record	Opt-in for detection. Records detect rows of data that drop out of a data set and down-score any observations.
	Schema	Automatically detected. Schema detects changes to columns and data types and down-score any observations.
	Dupes	Opt-in for detection. Duplicates (dupes) are values that match other existing values in columns and can be set to detect exact and fuzzy matches.
	Shapes	Automatically detected. Shapes detect rare or inconsistent data formats in string columns and down-score any observations.
5	Labels, Job, and Export tabs	These three tabs let you do the following: • Labels - Apply labels and annotations. • Job - View and edit the run command line of the current and previous job. • Export - Export individual or all data quality findings as .xlsx files.

No.	Component	Description
6	Finding drill-in	Click data quality dimension tabs with findings to discover more insights about potential data quality issues.
7	Whitespace and masking	When you click the **Whitespace** toggle above and to the right of the Data Preview, gray rectangles highlight empty spaces between string data in the Data Preview. When you select the Masked checkbox option at the top of a Data Preview column and click **Update Masking** above and to the right of the Data Preview, the Findings page refreshes and masks column entries with "XXXXX" instead of displaying potentially sensitive data.
8	Data Preview	A preview of the columns in your dataset where you can view cell values, column data types, passing or failing columns, and where Shapes are detected. You can also add column-level Quick Rules, view sensitive labels, parent key columns labels, and column statistics, such as max, mean, in, and number of uniques. **Note** The data preview limit is 10 records and is non-configurable.

What's next?

Drill down into your data quality findings to learn more about what Collibra DQ discovered and how the observations contribute to the overall data quality score of your dataset.

Data quality monitors

When Collibra DQ observes a change or anomaly, it records the finding for each data quality monitor and includes a numeric indicator next to the tab where it detects the issue. You can drill down into the various data quality monitor tabs on the Findings page to better understand the quality and reliability of your dataset.

This section provides details about the following data quality monitors:

- Behaviors
- Rules
- Outliers
- Patterns

- Source
- Record
- Schema
- Dupes
- Shapes

Behaviors

Collibra DQ learns from column-level profiling to create Adaptive Rules, which contribute to the overall Behavior score. Adaptive Rules are rules that automatically observe and adapt to changes in numeric representations of data over time and downscore any values outside defined boundaries.

Behavioral anomalies most commonly appear when you use behavior lookback or Replay, which allows Collibra DQ to learn the behavior of a dataset over a period of time.

The following table describes the information available on the Behaviors tab of the Findings page.

Column	Description
Blind Spot	The column where a behavioral anomaly is detected.
Type	The type of Adaptive Rules that the behavioral model detects from the profiling activity on a given column. The behavioral model consists of the following factors: • NULL values • Empty values • Cardinality • Datatype shifting • Row counts • Load time • Minimum value • Maximum value • Mean value See the table in the View AR section for more information on these factors.

Column	Description
Baseline	The mean value of past lookback runs determined by the data lookback value on the settings modal in Explorer. For example, the mean number of NULLs in column *x* for the past 10 runs. **Note** Some runs are excluded from the baseline based on annotations. For example, adaptive rules marked as "breaking" are excluded from the baseline or "normal" values.
Today	The value of the behavioral observation on the day that Collibra DQ detects it.
% Change	The percent change from the value of one row to another. $x = \frac{V_2 - V_1}{\lvert V_1 \rvert} * 100$
Δ % Change	The delta percent change from the value of one row to another. Delta percent change does not apply to absolute baselines, such as min, max, and mean. $x = (V_2 - V_1)$ **Note** Delta percent change is only available for volume-weighted metrics, such as null, empty, and shift.
Zscore	The number of standard deviations away from the expected baseline value.
Description	The description of the type of DQ check performed on a given column.
Score	The value subtracted from your overall DQ score. The distance from the expected ranges set by the variance and boundaries of the baseline value. Expected ranges are also visible in the AR panel with graphs available in the Details panel for each line item.
Status	Allows you to validate or resolve an observation and, when applicable, assign it to a user for further analysis.
Profile	The user account that is assigned to this behavioral finding. When the Status is Assigned, a user profile displays in this column. **Note** When a behavioral finding is unassigned, the profile column is empty.

Column	Description
Details	Click to open the Change Detection modal. **Note** If you change an Adaptive Rule from Manual to Adaptive, you need to retrain your dataset by clicking Retrain on the Behavior tab of the Findings page.

Pass All

Pass All gives you the option to pass all observations on a given day at once. When you pass all observations, your DQ score updates when you refresh the page.

View AR

View AR opens the Adaptive Rules of a DQ Job.

Collibra DQ profiles the data and builds a model for each dataset it scans. This allows Collibra DQ to learn what *normal* means within the context of each dataset. As the data changes, the definition of normal also changes. Instead of requiring you to adjust rule settings, Collibra DQ continues to adjust its model. This approach enables Collibra DQ to provide automated, enterprise-grade data quality coverage that removes the need to write dozens or even hundreds of rules per dataset.

The data in the graph may differ from the data shown on the Adaptive Rules page. This is intentional to provide interactive functionality that is calculated with each run, rather than storing historical ranges.

The following table shows the possible Adaptive Rules:

Adaptive Rules	Description	Scoring calculation when the value is greater than the upper bound	Scoring calculation when the value is less than the upper bound
Row Count	Row count changes in your table. **Note** To establish a baseline reference point, we inspect the row counts from the *x* most recent **successful** runs. Here, *x* represents the value you set as the data lookback period when you create the DQ Job. A "successful" run refers to a job that either doesn't have a row count data quality issue or an issue that has been approved (passed) by a user.	(value - upper bound)	(lower bound - value)
Loading Time	Loading time changes.	((value - upper bound) / upper bound) * 50	((lower bound - value) / lower bound) * 50
Uniqueness	Cardinality changes to a column within the range of previous DQ Jobs.	(value - upper bound)	(lower bound - value)
Null Values	Null values detected in a column.	(value - upper bound)	(lower bound - value)
Empty Values	Empty data detected in a column.	(value - upper bound)	(lower bound - value)
Min	Columns with min values outside the normal range.	((value - upper bound) / upper bound) * 50	((lower bound - value) / lower bound) * 50
Mean	Columns with mean values outside the normal range.	((value - upper bound) / upper bound) * 50	((lower bound - value) / lower bound) * 50

Adaptive Rules	Description	Scoring calculation when the value is greater than the upper bound	Scoring calculation when the value is less than the upper bound
Max	Columns with max values outside the normal range.	((value - upper bound) / upper bound) * 50	((lower bound - value) / lower bound) * 50
Data Type Check (Integer, String, Date)	Columns that shift from one data type to another.	(value - upper bound)	(lower bound - value)

The Change Detection modal appears when you click the button. See the Working with the Change Detection modal page for more information.

Rules

Collibra DQ takes a strong stance that data should first be profiled, auto-discovered and learned before applying basic rules. This methodology commonly removes thousands of rules that will never need to be written and evolve naturally overtime. However there are still many cases to add a simple rule, complex rule or domain specific rule.

The score on the Rules tab is the rule percentage (% column) multiplied by the number of points (Points column), rounded to a whole number. If a rule with a Percent Scoring Type does not meet or exceed its defined percentage, the score on the Rules tab remains 0.

The following table describes the information available on the Rules tab of the Findings page.

Column	Description
Rule Name	The name of the rule.

Column	Description
Condition	The rule condition that is defined in the rule. An alert generates when the condition is met. In Pullup mode, the Condition column is derived from the rule's SQL column. In Pushdown mode: • If the rule is PASSING, the Condition column is derived from the rule's SQL column. • If the rule is BREAKING: ◦ If the Rule Type is SQLG, the Condition column is derived from the rule's SQL column. ◦ If the Rule Type is a Check type (EMPTYCHECK, NULLCHECK, DATECHECK, INTCHECK, DOUBLECHECK, STRINGCHECK), the Condition column is derived from the rule's SQL column. ◦ If the Rule Type is a Data Class (CUSTOM), the Condition column displays the Regular Expression. ◦ If the Rule Type is SQLF, the Condition column displays the SQL query.
Points	The number of points to deduct from the quality score for a given rule break.
Perc	The ratio of the total number of breaking records over the total number of rows.
Breaking Records	The number of rows with records that did not pass the conditions of the rule. If the status of the rule is Exception or Passing, then the value should be 0.
Passing Records	The number of rows with records that passed the conditions of the rule.
Status	The status of the rule. The following table shows the possible statuses:

Status	Description
PASSING	Passing indicates that the dataset passes the rule check and no points are deducted from the data quality score.
EXCEPTION	Exception indicates that there is an error with the rule check. Expand the row to view the error message.
BREAKING	Breaking indicates that the rule has detected an anomaly in a column and points are deducted from the data quality score. Expand the row to view the data preview of the break record.

Dimension	The DQ Dimension that is identified in the rule finding, for example, Completeness.

Column	Description
State	Shows if a rule is enabled or disabled.
Status	Allows you to validate or resolve an observation and, when applicable, assign it to a user for further analysis.
Profile	The user account that is assigned to this rule finding. When the Status is Assigned, a user profile displays in this column. **Note** When a rule finding is unassigned, the profile column is empty.

Action	Opens a menu with the following options:
	• **Archived Breaks** lets you preview the rule break export file and download either a CSV or JSON file, depending on your processing mode, giving you more control over how you use and share break records.
	  • In Pullup mode, you can download a CSV file containing details of the break records.
	  • In Pushdown mode, you can download either a CSV or JSON file containing details of the break records.
	  Note The ability to copy rule breaks is limited to secure Cloud Native deployments of Collibra DQ. You cannot copy rule breaks in Standalone deployments.
	• **Preview Breaks** shows all breaking records for a given rule, within the defined Preview Limit. This option requires the ROLE_DATASET_ACCESS role.

Rule Discovery

Rule Discovery detects the data classes assigned to a selected data category. The Rule Discovery algorithm automatically selects the best match if a column matches two or more data classes. Data class match criteria are determined by percent match and name distance.

Click **Rule Discovery**, then select an option from the Data Category dropdown menu. Click **Run Discovery** to assign your selection as a data category and run the discovery job.

Break records in the PostgreSQL Metastore

When a rule returns breaking records, the following query inserts unique records into the PostgreSQL Metastore **rule_breaks** table based on dataset, run_id, rule_nm, and link_id:

```
INSERT INTO rule_breaks (dataset, run_id, rule_nm, link_id)
VALUES (:dataset, :runId, :ruleNm, :linkId)
ON CONFLICT (dataset, run_id, rule_nm, link_id)
DO UPDATE SET
    dataset = :dataset,
    run_id  = :runId,
    rule_nm = :ruleNm,
    link_id = :linkId
WHERE
    rule_breaks.dataset = :dataset
    AND rule_breaks.run_id = :runId
    AND rule_breaks.rule_nm = :ruleNm
    AND rule_breaks.link_id = :linkId;
```

> **Note** Because only unique records are inserted into the rule_breaks table, the number of records in the rule_breaks table might not match the number of breaking records displayed on the Findings page.

Pulse View Preview

The Pulse View gives you a data preview box plot for simple rules that are breaking. You can click any available box to drill into the data of that day.

> **Note** Pulse View Preview is not available for passing rules or runs without data.

Exporting rule break records

There are three options to export the details of your rule break records as .xlsx files:

- **Export** generates an Excel file with the details from the drill-in.
- **Export LinkIds** generates an Excel file with the name of the dataset, Run Id, Rule Name, and Link Id, when available.

- **Export with Details** generates an Excel file with the details from the drill-in and the data preview, when available.

> **Note** The ability to copy rule breaks is limited to secure Cloud Native deployments of Collibra DQ. You cannot copy rule breaks in Standalone deployments.

Limitations

Total records are calculated by adding the total number of breaking and passing records. Therefore, when referencing a secondary dataset in a native rule, the extra rows from the secondary dataset are included in the query results on the Findings page, skewing the total rows calculation and the percentage of breaking records.

Outliers

The Outliers monitor detects values that differ significantly from the rest of the data and may indicate bad or incorrect data. Numerical outliers are detected with the Interquartile Range (IQR) and box plot methods.

Multiple simultaneous outlier configurations can be defined for a single dataset. However, some settings are shared between all configurations. For example, columns analyzed, key columns, and date columns are selected independently on each configuration, while categorical analysis, quartile ranges, and date look back settings are shared on all configurations for the dataset.

> **Note** Enabling categorical outlier detection will not interfere with numerical outlier detection.

The following table describes the information available on the Outliers tab of the Findings page.

Column	Description
Key	The column assigned as a Key column. Any outliers that Collibra DQ are grouped by this column if and when a Key is assigned.
Column	The column where a potential outlier is detected.

Column	Description
Value	The value of the column of the detected outlier.
Count	The number of potential outliers in the column.
Predicted	The type of value that Collibra DQ predicts for a given run, for example, categorical. This prediction is based on the observed values of previous runs.
Conf	The confidence score, ranging from 0 to 100, indicates how far the current value is from the lower or upper bound. Lower scores such as 0 or 1, indicate a higher likelihood of the value being an outlier. Conversely, higher scores, such as 97, suggest a lower likelihood of the value being an outlier.
Status	Lets you label and train a finding. a finding. The available dropdown menu options are Validate, Invalidate, and Resolve. **Validate** instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation. **Invalidate** instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options: • **Save** lets you mark a finding as invalidated. • **Save & Retrain** lets you invalidate a finding and any previously saved invalidated findings, if any. > **Tip** When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed. **Resolve** Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.
Profile	The user account that is assigned to this outlier finding. When the Status is Assigned, a user profile displays in this column. > **Note** When an outlier finding is unassigned, the profile column is empty.

Column	Description
Link ID	Links back to the detected record for remediation. **Note** Link ID is not available for categorical outliers.
Action	Opens the **Calibrate outlier** modal.

Note When you assign a Date and Key column in an Outlier configuration, Collibra DQ may also discover Record finding.

Outlier percentage changes

A bar chart and percentage change display when you click the + icon in the Outlier findings table, as shown in the example below.



The outlier percentage change is calculated between the value in the latest history data point and the current value. It isn't calculated between the value and predicted value. The calculation uses a buffer (the "weight" factor) to adjust the lower and upper boundaries for outliers, making the outlier detection more flexible and adaptable to the specific dataset.

Specifically, outliers are identified using the following process:

1. Define what is considered "typical" in the dataset. This is done using the IQR, which is a statistical measure that represents the range within which the middle 50% of the values

in the dataset fall.

2. Calculate the IQR by determining the 25th percentile (Q1) and the 75th percentile (Q3) of the data. These percentiles represent the values below which 25% and 75% of the data fall, respectively. The IQR is the difference between Q3 and Q1.
3. Define what is considered an "outlier" in the dataset. An outlier is defined as any value that falls below (Q1 - 1.5IQR) or above (Q3 + 1.5IQR) the standard range.
4. Introduce a weight factor to make the range more flexible and adaptable to the specific characteristics of the data. This weight factor adjusts the standard range. For example, if the weight is 1.5, then the range is narrowed; if the weight is 0.5, then the range is broadened. For any other weight value, the range is adjusted proportionally.
5. The result is an adjusted range for "typical" values. Any value that falls outside this adjusted range is considered an outlier.

Invalidate All

Invalidate All instructs Collibra DQ to ignore all outlier findings and allow the values to pass.

Exporting outlier records

There are two options above the drill-in table to export the details of your outlier records as .xlsx files:

- **Export All (Basic Only)** generates an Excel file with the details from the drill-in.
- **Export All with Details** generates an Excel file with the details from the drill-in and the data preview, when available.

Patterns

The Patterns monitor detects similarities between string values across columns and down-scores any uncommon observations.

The following table describes the information available on the Patterns tab of the Findings page.

Column	Description
	A data preview of the actual value and the predicted value. Click to expand the observation. The first row in the data preview is the actual value and the second row is the predicted value.
Type	The type of pattern detected, for example, Suggestive.
Observation	An overview of the pattern observation. The format of this column is [actual value in columns where a pattern is detected -> predicted value in columns where a pattern is detected].
Count	The number of patterns detected in the column.
Status	Lets you label and train a finding. The available dropdown menu options are Validate, Invalidate, and Resolve. **Validate** instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation. **Invalidate** instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options: • **Save** lets you mark a finding as invalidated. • **Save & Retrain** lets you invalidate a finding and any previously saved invalidated findings, if any. **Tip** When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed. **Resolve** Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.
Profile	The user account that is assigned to this pattern finding. When the Status is Assigned, a user profile displays in this column. **Note** When a pattern finding is unassigned, the profile column is empty.

Adjusting the downscore value

You may find that the default downscore value of 1 does not meet your requirements. Click the input field to the upper-left of the findings table and enter any whole number next to **DownScore Value** to adjust the downscore value of the pattern observations that Collibra DQ detects.

Exporting rule break records

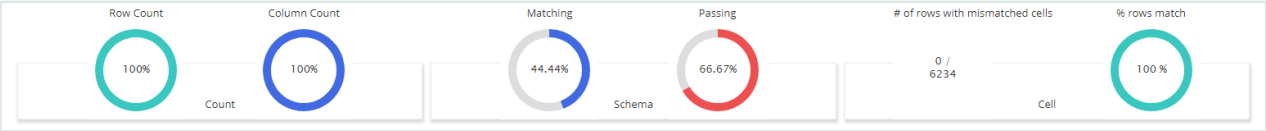
There are two options to export the details of your rule break records as .xlsx files:

- **Export** generates an Excel file with the details from the drill-in.
- **Export with Details** generates an Excel file with the details from the drill-in and the data preview, when available.

Source

The Source monitor is configured in the Mapping step in Explorer, and is used to detect row count, schema, and cell value inconsistencies between the source file or table and the target file or table and down-scores any observations.

The Source tab provides doughnut charts to show an overview of any source issues and whether they occur at the Count-, Schema-, or Cell-level.



Count

Column	Description
Type	The type of count issue, for example, Dataset Column Count and Dataset Row Count.
Target Count	The number of columns or rows that Collibra DQ counts in the target dataset.

Column	Description
Source Count	The number of columns or rows that Collibra DQ counts in the source dataset.
Change (%)	The percentage count difference between the target and source datasets.
Description	The description of the count issue.
Action	Lets you label and train a finding. The available dropdown menu options are Validate, Invalidate, and Resolve. **Validate** instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation. **Invalidate** instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options: • **Save** lets you mark a finding as invalidated. • **Save & Retrain** lets you invalidate a finding and any previously saved invalidated findings, if any. **Tip** When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed. **Resolve** Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.
Profile	The user account that is assigned to this source finding. When the Status is Assigned, a user profile displays in this column. **Note** When a source finding is unassigned, the profile column is empty.

Column Schema

Column	Description
Target Column	The column of your target table, file, or view where a an anomaly is detected. This value is mapped to the Source Column. **Note** If a selected column exists in the target dataset but not in the source, the resulting outer join table will display "null" for that column in the source dataset.
(T) Col Order	The column order of the target dataset.
(T) Type	The data type of the target dataset.
Source Column	The column of your source table, file, or view where a an anomaly is detected. This value is mapped to the Target Column. **Note** If a selected column exists in the source dataset but not in the target, the resulting outer join table will display "null" for that column in the target dataset.
(S) Col Order	The column order of the source dataset.
(S) Type	The data type of the source dataset.
Matches	Describes whether the target column schema matches the source column schema. If the column schemas between the target and source match, then a **Y** displays. If the column schemas between the target and source do not match, then a **N** displays.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Passing</td><td>There are two possible options:
<table><tr><th>Option</th><th>Description</th></tr><tr><td>✓</td><td>The checkmark indicates that the source or target passes the column schema check. This is only visible when the Show failing only option is not selected.</td></tr><tr><td>✖</td><td>The X indicates that the source or target does not pass the column schema check.</td></tr></table></td></tr>
<tr><td>Description</td><td>The description of the column schema issue.</td></tr>
<tr><td>Action</td><td>Lets you label and train a finding. The available dropdown menu options are Validate, Invalidate, and Resolve.

Validate instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation.

Invalidate instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options:

• Save lets you mark a finding as invalidated.
• Save & Retrain lets you invalidate a finding and any previously saved invalidated findings, if any.

Tip When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed.

Resolve Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.</td></tr>
<tr><td>Status</td><td>The status of your data quality item, for example, Observation.</td></tr>
<tr><td>Profile</td><td>The user account that is assigned to this source finding. When the Status is Assigned, a user profile displays in this column.

Note When a source finding is unassigned, the profile column is empty.</td></tr>
</table>

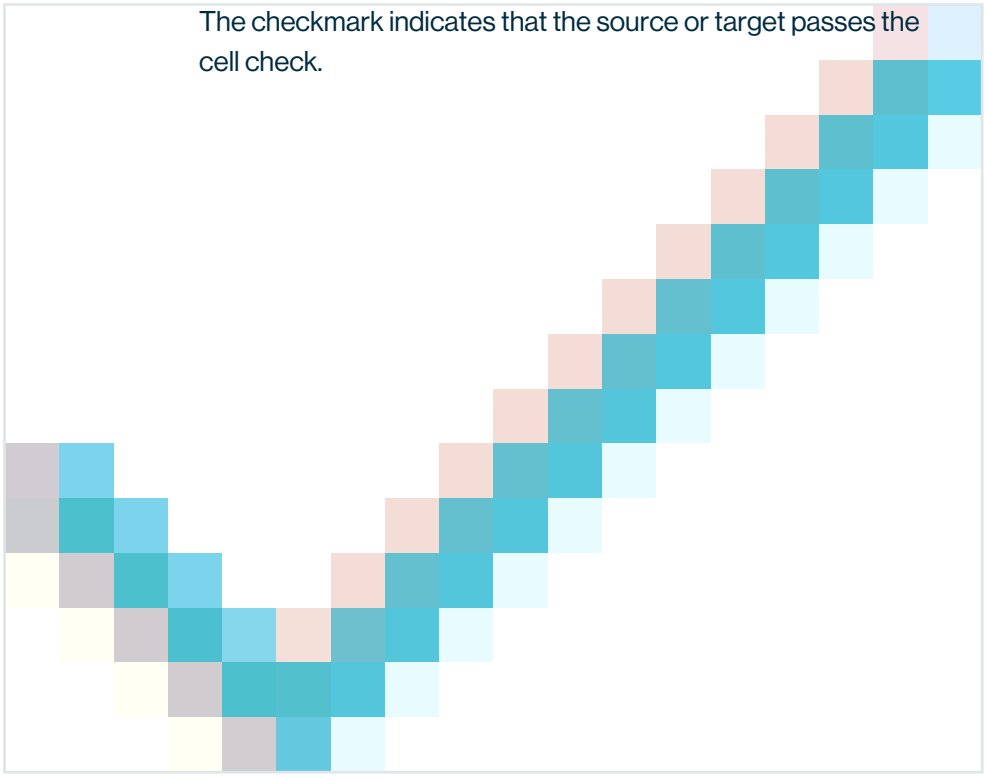
Cell

# of rows compared are the total number of rows from the target and source datasets that Collibra DQ includes in its cell check.

Column shifts are the number of columns that Collibra DQ flags as potential cell issues.

Issues/count are the number of issues in a particular column that Collibra DQ flags as a potential issue.

Column	Description
System (Type)	The database or file of the source and target data.
Key	The column assigned as a Key column. Any cell issues that Collibra DQ detects are grouped by this column if and when a Key is assigned. **Note** An outer join is used to find discrepancies between source and target datasets. The extracted keys will be "null" for any source record that lacks a matching target record in the selected columns. The displayed keys are only from the target dataset.
Column	The column where an anomaly is detected.
Value	The value of the column where the anomaly is detected. **Note** You can select an option from the Value column dropdown menu to downtrain any finding based on column and value.

Column	Description
Passing	There are two possible options: **Option** — **Description**  — The checkmark indicates that the source or target passes the cell check.

Column	Description

Option	Description
	The X indicates that the source or target does not pass the cell check.

Count	The number of rows in either the source or target dataset.
Description	The description of the column schema issue.

Column	Description
Action	Lets you label and train a finding. The available dropdown menu options are Validate, Invalidate, and Resolve. **Validate** instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation. **Invalidate** instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options: • **Save** lets you mark a finding as invalidated. • **Save & Retrain** lets you invalidate a finding and any previously saved invalidated findings, if any. **Tip** When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed. **Resolve** Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.
Status	The status of your data quality item, for example, Observation.
Profile	The user account that is assigned to this source finding. When the Status is Assigned, a user profile displays in this column. **Note** When a source finding is unassigned, the profile column is empty.

Record

The Record monitor detects rows of data that drop out of or are added to a dataset and down-score any observations. To see record findings, select at least one numeric data type column in the Outliers layer, a Key column, and a Date column. If you do not select at least one numeric column, a Key column, and a Date column, Collibra DQ skips the record check activity. Alternatively, you can exclude Outlier findings and detect only Records by selecting a Key column and a Date column but excluding the numeric data type column from your Outlier configuration.

The following table describes the information available for the Record tab on the Findings page.

Column	Description
Observation	The type of record anomaly detected.
Status	Allows you to validate or resolve an observation and, when applicable, assign it to a user for further analysis.
Profile	The user account that is assigned to this record finding. When the Status is Assigned, a user profile displays in this column. **Note** When a record finding is unassigned, the profile column is empty.

Schema

The Schema monitor detects changes to columns and data types and down-scores any observations.

> **Note** Collibra DQ requires a minimum of 4 previous runs to establish a baseline calculation of a dataset's table properties before it can detect any schema changes.

Column	Description
Observation	The type of schema anomaly detected.

Column	Description
Action	Lets you label and train a finding. The available dropdown menu options are Validate, Invalidate, and Resolve. **Validate** instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation. **Invalidate** instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options: • **Save** lets you mark a finding as invalidated. • **Save & Retrain** lets you invalidate a finding and any previously saved invalidated findings, if any. **Tip** When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed. **Resolve** Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.
Status	The status of your data quality item, for example, Observation.
Profile	The user account that is assigned to this schema finding. When the Status is Assigned, a user profile displays in this column. **Note** When a schema finding is unassigned, the profile column is empty.

Exporting schema records

Click **Export** above the drill-in table to generate an Excel file with the details from the drill-in.

Dupes

The Dupes monitor detects column values that match other existing column values.

The following table shows the available columns in the Dupes tab.

Column	Description
Type	The type of finding, for example, **DUPE**.
Score	The percentage that two or more duplicate values match. A score of 100 indicates that the duplicate values are exact matches of each other, whereas a score of 85 indicates that the duplicate values are fuzzy matches.
[Column]	The column where the duplicate value appears. Case insensitive exact match duplicates for Pullup datasets display in all lower case. While the casing of exact match duplicates may vary, this is the expected behavior. > **Note** The name of this column is dynamic depending on how it appears in your table, file, or view. For example, when the name of the column in your data source that contains the duplicate values is called *last name*, then the column in the Findings table will be **last name**.
Occurs	The number of duplicate values in a column.
Profile	The user account that is assigned to this dupes finding. When the Status is Assigned, a user profile displays in this column. > **Note** When a dupes finding is unassigned, the profile column is empty.

Column	Description
Status	Lets you label and train a finding. The available dropdown menu options are Validate, Invalidate, and Resolve. **Validate** instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation. **Invalidate** instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options: • **Save** lets you mark a finding as invalidated. • **Save & Retrain** lets you invalidate a finding and any previously saved invalidated findings, if any. > **Tip** When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed. **Resolve** Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.
Action	In Pushdown mode, you can download a CSV or JSON file containing details of the break records. > **Note** This column does not display for DQ Jobs created in Pullup mode.

Exporting dupes records

Click **Export** above the drill-in table to generate an Excel file with the details from the drill-in.

Shapes

The Shapes monitor detects rare or inconsistent data formats in string columns. Collibra DQ uses a sampling of the metadata to detect potential data issues, which can then be drilled into by creating queries from the data model finding.

> **Note** The sampling of metadata may change and the number of shapes returned may differ.

The following table shows the information available on the Shapes tab of the Findings page.

Column	Description
Column	The column where Collibra DQ detects a shape.
Schema	The datatype schema of the column where Collibra DQ detects a shape.
Shape	The format of the shape.
Count	The number of times a particular shape format appears in a column.
Row Count	The number of rows in the table, file, or view where Collibra DQ detects shape issues.
Percent	The percentage a shape conforms to the format that Collibra DQ identifies as normal.
Shapes/Col	The number of shape issues in a particular column.
Status	Lets you label and train a finding. The available dropdown menu options are Validate, Invalidate, and Resolve. **Validate** instructs Collibra DQ to either assign a finding to a specific user for review, which then appears in the View the Assignment Queue or acknowledge without an assignee that the finding is a valid observation. **Invalidate** instructs Collibra DQ to ignore a finding and allow the value to pass. There are two invalidation options: • **Save** lets you mark a finding as invalidated. • **Save & Retrain** lets you invalidate a finding and any previously saved invalidated findings, if any. > **Tip** When you have many findings to invalidate, it may be best to use the Save option to invalidate them at the same time, once all findings are reviewed. **Resolve** Instructs Collibra DQ to mark the finding as an observation and prevents it from appearing in future runs. Resolving a finding does not immediately affect data quality scores.

Column	Description
Profile	The user account that is assigned to this shapes finding. When the Status is Assigned, a user profile displays in this column. **Note** When a shapes finding is unassigned, the profile column is empty.
Action	In Pushdown mode, you can download either a CSV or JSON file containing details of the break records. **Note** This column does not display for DQ Jobs created in Pullup mode.

Exporting shape records

There are two options above the drill-in table to export the details of your outlier records as .xlsx files:

- **Export** generates an Excel file with the details from the drill-in.
- **Export with Details** generates an Excel file with the details from the drill-in and the data preview, when available.

Configuring manual options

You can configure additional shape options on the Findings page. Click the ⚙ above the upper-right corner of the Shapes Findings table.

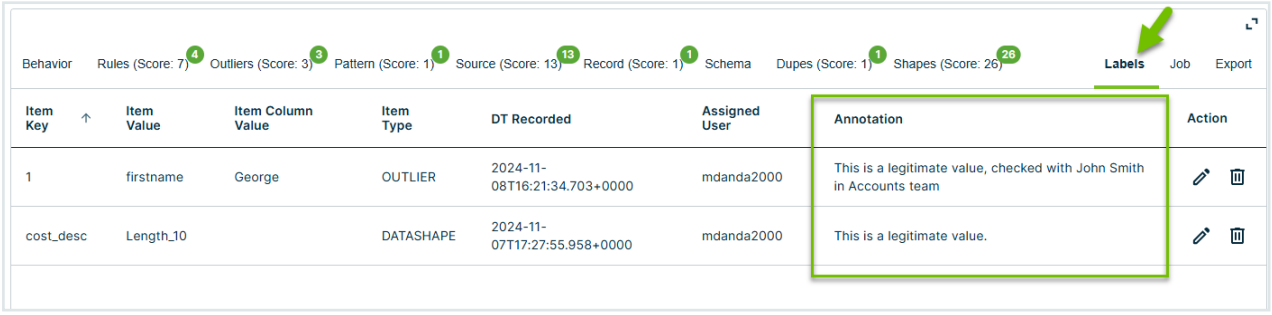
The following table shows the available options.

Option	Description	Default Value
Occurrence % < [X]	Only shows shapes below the given percentage threshold.	0.001
Format per Col < [X]	Only shows columns with less than the given number of formats. For example, when there are 30 formats and the Format per Column value is set to 5, then only 5 columns display.	20

Option	Description	Default Value
Character Length < [X]	Only shows Shape issues with fewer than the given number of characters. For example, if the value is set to 9, then all values less than 9 will show.	12

Labels tab

The Labels tab on the Findings page provides a list of the findings that have been invalidated, including the annotations assigned to each invalidated finding. For more information, go to Invalidating Findings.



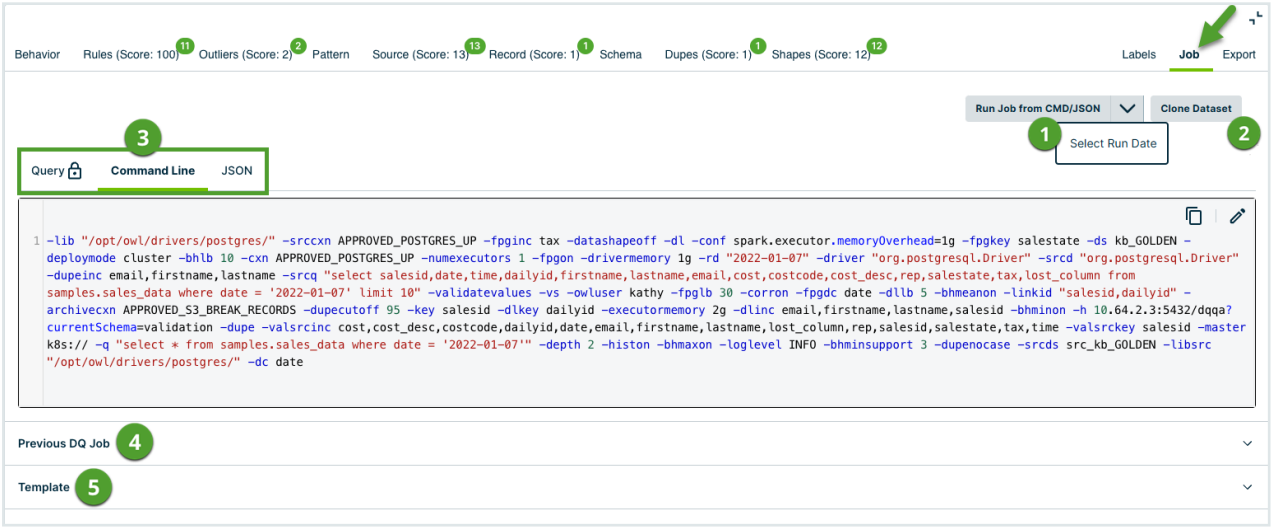
The Labels tab provides the following information for each invalidated finding:

Parameter	Description
Item Key	The column assigned as a key column of the finding.
Item Value	The value of the column of the finding.
Item Column Value	The value of the column of the finding.
Item Type	The dimension of the finding.
DT Recorded	The date and time the finding was created.
Assigned User	The user that invalided the finding.
Annotation	The annotation provided by the user when the finding was invalided.

Parameter	Description
Action	Provides options to edit or delete the annotation. Deleting the annotation returns the status of the finding to Unassigned.

Job tab

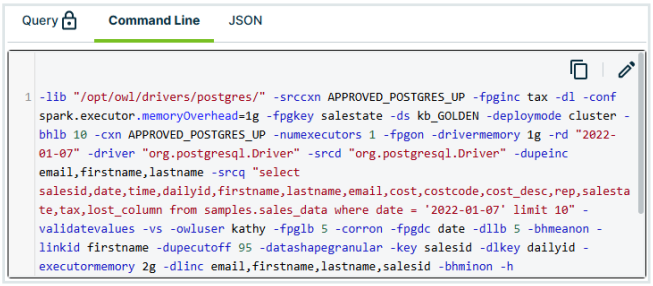
The Job tab on the Findings page provides details about the job run, including the SQL query and command line parameters. It also enables you to edit the run command line, run the job again, and create a clone of the dataset.

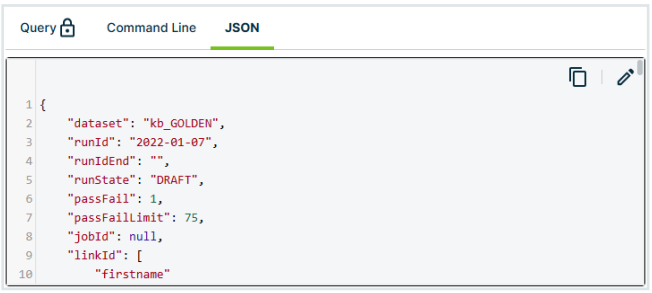


The following table provides and overview of the key elements of the Job tab with numbers that correspond to the image above.

No.	Component	Description
1	**Run Job from CMD/JSON** **Note** This option is called **Run Job from JSON** for Pushdown jobs since the command line option is not available in Pushdown.	Runs the job using the parameters specified in the Command Line or JSON tab. When you update parameters in the command line or JSON, you can click **Run Job from CMD/JSON** to re-run your jobs with your changes. Alternatively, click , then **Select Run Date**, to select a new run date (the `-rd` parameter in the command line and `runId` in the JSON). After selecting a new run date, click , then **Submit**. The job re-runs with the new run date.

No.	Component	Description
2	**Clone Dataset**	Creates a new copy of the dataset. Note that alerts and rules are not cloned. **Note** You can rename a dataset from the Dataset Manager by using the **Rename** action.

No.	Component	Description
3	**Query**, **Command Line**, and **JSON** tabs	Provides details of the job run. Select one of the following tabs: The Query tab displays the source query that creates the job. It is not editable. For example:  The Command Line tab displays the run command with the associated parameters for the current job. Use the buttons on the right to **Copy to clipboard** or **Edit** the command. For example:  For more information on the parameters, go to Job Command Line Parameters. The JSON tab displays the run command for the current job in JSON format. Use the buttons on the right to **Copy to clipboard** or **Edit** the command. For example:

No.	Component	Description
		

4	**Previous DQ Job**	A read-only representation of the command line and parameters used the last time this job was run.
5	**Template**	A template of the command line used to run the previous job. If the template uses variables, the variables are listed in place of the actual values.

Job Command Line Parameters

This topic provides information on using command line parameters to run jobs. It includes information specific to managing Spark resources, as well as a reference guide describing the commands available in the job run command.

> **Note** You can display the run command used to run the current job on the Job tab of the Findings page.

Managing Spark resources from the command line

Scale linearly with your data

Scale linearly with your data by adding executors and/or memory. For example:

```
-f "file:///Users/home/salary_data.csv" \
-d "," \
-rd "2018-01-08" \
-ds "salary_data"
-numexecutors 2 \
-executormemory 2g
```

Yarn Master

If Collibra DQ is run on an edge node on a popular hadoop distribution such as HDP, CDH, EMR it will automatically register the jobs with Yarn Resource Manager.

Spark Master

Collibra DQ also runs using spark master by using the `-master` command and passing in `spark:url`.

Spark Standalone

Collibra DQ typically runs in standalone, but it will not by default distribute the processing beyond the hardware it was activated on.

Options	Description
deploymode	The Spark deploymode option. For example, `cluster`.
drivermemory	The driver memory of your local Spark instance in gigabytes.
executorcores	Spark executor cores.
executormemory	The total Spark executor memory in gigabytes, for example, `3G`.
master	Overrides `local[*]`
sparkprinc	Kerberos principal name, example: owl@OWL.COM.

Use Spark-Submit directly bypassing DQCheck

```
spark-submit \
--driver-class-path /opt/owl/drivers/postgres42/postgresql-
42.2.4.jar \
--driver-library-path /opt/owl/drivers/postgres42/postgresql-
42.2.4.jar \
--driver-memory 3g --num-executors 2 --executor-memory 1g \
--master spark://Kirks-MBP.home:7077 \
--class com.owl.core.cli.OwlCheck /opt/owl/bin/owl-core-trunk-
jar-with-dependencies.jar \
-u user -p pass -c jdbc:postgresql://xyz.chzid9w0hpyi.us-east-
1.rds.amazonaws.com/postgres \
-ds accounts -rd 2019-05-05 -dssafeoff -q "select * from
accounts"
-driver org.postgresql.Driver -lib /opt/owl/drivers/postgres42/
```

Parallel JDBC Spark-Submit

```
spark-submit \
--driver-class-path /opt/owl/drivers/postgres42/postgresql-
42.2.4.jar \
--driver-library-path /opt/owl/drivers/postgres42/postgresql-
42.2.4.jar \
--conf spark.driver.extraJavaOptions=-Dlo-
g4j.configuration=file:///opt/owl/config/log4j-TRACE.properties
\
--conf spark.executor.extraJavaOptions=-Dlo-
g4j.configuration=file:///opt/owl/config/log4j-TRACE.properties
\
--files /opt/owl/config/log4j-TRACE.properties \
--driver-memory 2g --num-executors 2 --executor-memory 1g --mas-
ter spark://Kirks-MBP.home:7077  \
--class com.owl.core.cli.OwlCheck /opt/owl/bin/owl-core-trunk-
jar-with-dependencies.jar \
-u us -p pass -c jdbc:postgresql://xyz.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com/postgres \
-ds aumdt -rd 2019-05-05 -dssafeoff -q "select * from aum_dt" \
-driver org.postgresql.Driver -lib
/opt/owl/drivers/postgres42/  \
-connectionprops fetchsize=6000 -master spark://Kirks-
MBP.home:7077 \
-corroff -histoff -statsoff \
-columnname updt_ts -numpartitions 4 -lowerbound 1557597987353 -
upperbound 1557597999947
```

Command Line Reference

The following table describes the parameters available for a job run command.

Parameter	Description
adddc	Add Date Column as Run Id that is used. To see an example, go to Add Date Column.
addlib	Additional library directory to be added to the classpath for the DQ Job (spark-submit)
agentjobid	Internal use only
agg	Grouping function for flexibility
aggq	select * from dataset where `time_bin = '2018-12-10 10'`, example: for aggregate override.
alertemail	Automatically add an alert with score greater than 75, to the email value supplied
alias	Dataset name alias, example: userTable or users or user_file
archivecxn	Connection name to archive break records
avro	avro file data flag
avroschema	avro schema file
bd	Column count if you want to group by a particular set of values for behavioral statistics
bdcol	Behavioral function if you want to aggregate for behavioral statistics
bdfunc	Behavioral function if you want to aggregate for behavioral statistics
bdgrp	Behavioral group to dynamically collect stats if you want to group by a particular set of values for behavioral statistics
behaviorscoreoff	Turn off behavior scoring

Parameter	Description
bhemptyoff	Behavior empty check detection off
bhlb	The behavior lookback period where the entered value represents the number of days. For example, a value of 12 looks back 12 days of data. For more information, go to Lookback.
bhmaxoff	Behavior max value detection off
bhmaxon	Behavior max value detection on
bhmeanoff	Behavior mean value detection off
bhmeanon	Behavior mean value detection on
bhminoff	Behavior min value detection off
bhminon	Behavior min value detection on
bhminsupport	Behavior min support, set to 4 by default, min number of days to learn from, learning phase
bhnulloff	Behavior null check detection off
bhrowoff	Behavior row count detection off
bhsensitivity	Behavior sensitivity: `NARROW`, `NULL`, `WIDE`
bhtimeoff	Behavior load time detection off
bhtimeon	Behavior load time detection on
bhuniqueoff	Behavior unique detection off
bhuniqueon	Behavior unique detection on
br	Number of back runs to fill training history, should be an integer value. To see an example, go to Advanced and DQ Job Back Run.
brbin	Time bin for back runs, example: `-brbin DAY`
bt	Back-tick character (`) to escape SQL queries when returning to database

Parameter	Description
by	Compare by `DAY`, `HOUR`, or `MIN`.
c	jdbc://hive:3306 (connection URL)
cacheoff	Turn caching off. Caching is on by default. It can be turned off if the dataset is too large or cache optimization is not desired.
cardoff	Turn off profiling section of owlcheck
categoricallimit	Limit for categorical outliers stored
categoricallimitui	Limit for categorical outliers displayed
categoricalscore	Score for each categorical outlier
catoff	Disables categorical outlier detection.
caton	Turn on categorical outliers
catOutAlgo	Specify ML algorithm for categorical outliers. Default: "" (no ML)
catOUtAlgoParams	Optional params for catOutAlgo to override Owl-suggested params. Default: "". E.g. "k=5,initSteps=5"
catOutBottomN	Max number of categorical outliers in a column
catOutConfType	Method to use to calculate likelihood of category level
catOutMaxCategoryN	Maximum number of categories within key that will trigger homogenous past categorical outlier case
catOutMaxConf	Confidence upper bound to qualify as an outlier
catOutMaxFreqPercentile	Frequency percentile upper bound to qualify as an outlier
catOutMinFreq	Minimum frequency needed to be considered an outlier. Raise to make less sensitive
catOutMinVariance	Minimum frequency count variance (within key) required to be considered an outlier. set to negative to be more sensitive
catOutParallelOff	Turn off parallel column-wise processing of categorical outliers

Parameter	Description
catOutTopN	Number of top frequently appearing level in a column to include in preview
columnname	Column name to split on for spark JDBC
concat	No arguments. Concatenate option for categorical oultiers columns
conf	The Spark configuration option. For example, `spark.kubernetes.memoryOverheadFactor=0.4,spark.kubernetes.executor.podTemplateFile=local:///opt/owl/config/k8s-executor-template.yml,spark.kubernetes.driver.podTemplateFile=local:///opt/owl/config/k8s-driver-template.yml`
connectionprops	`key=value,hive.resultset.use.unique.column.names=false` To see an example, go to DQ Job JDBC.
connectionpropssrc	`key=value,hive.resultset.use.unique.column.names=false`
corefetchmode	Let core go fetch query from meta store in stead using the one passed in command line
corroff	Dataset correlation flag force off
corron	Dataset correlation flag force on
cxn	The name of the saved database or file connection from which your dataset originates.

Parameter	Description
d	Delimiter, ','
dataconceptid	Identifier of the group of semantic rules by datatype
datashapeexc	Exclude a column from data shapes discovery
data-shapegranular	Check length for alphanumeric fields, and independent check for numbers and letters
datashapeinc	Include a column that has been excluded from data shapes discovery
datashapelimit	Limit for datashapes stored
datashapelimitui	Limit for datashapes displayed
data-shapemaxcolsize	Maximum length of a string column before it is disqualified from shapes detection
data-shapemaxpercol	Maximum number of shapes per column before column is ignored during shapes processing
datashapeoff	Turn DataShape Activity Off
datashapescore	Score for each datashape
datashapesense	Maximum occurrence rate (%) to be considered a shape
dateoff	Turn date detection off. In some cases date detection is a costly operation with little value
dblb	DB lookback to check owl check history for previous histories
dc	The date column for outlier detection.
delta	Delta file data flag
deploymode	The Spark deploymode option. For example, cluster. For more information, see Deploy Mode.
depth	The depth of duplicate discovery between 1-3, increasing runtime non-linearly. The default value is 1.
df	Date Format, example: `yyyy-MM-dd`

Parameter	Description
diff	Percentage difference between two days to do a reference for keys missing
divisor	Divisor for unix timestamp. `s` for seconds or `ms` for milliseconds. Default is `ms`.
dl	Deep learning. This enables the outliers activity.
dlcombine	When numerical outlier appears more than once, combine them as single outlier
dlcombineoff	When numerical outlier appears more than once, do not combine them as single outlier
dlexc	Deep learning col exclusion, example: open,close,high,volume
dlinc	The column limit for deep learning. This can be a comma delimited list of columns to include in your job. For example, if you want to include columns called account_id, date, and frequency, the correct syntax would be `account_id,date,frequency`.
dlkey	The natural key for deep learning. This is the column in your dataset that you set as the key column.
dllb	The deep learning lookback period where the entered value represents the number of days included in the outlier activity lookback.
dlminhist	An automatically generated flag based on the outlier lookback setting -dllb that defines the minimum number of days before DQ flags data as potential outliers. -dlminhistensures that the number of days in the algorithm is relative to the total scope of the lookback period. Default dllb - 2
dlmulti	Pass multiple dlkey=dlinc key value pairs. Split by pipe for multiple
dn	Driver name `org.apache.jdbc.PhoenixDriver`
dpoff	Do not store data preview records
dprev	Data preview turned off, same as onReadOnly
dq	Double-quote character (") to escape SQL queries when returning to database
driver	The driver class name of a custom driver. To see an example, go to DQ Job Hive.
drivermemory	The driver memory of your local Spark instance in gigabytes.

Parameter	Description
ds	The name of the dataset.
dssafeoff	Best practice naming convention flag, provides a globally unique and meaningful natural key to all datasets
dupe	Enables the dupe activity.
dupeapprox	Approximate groupings default value =1 [0-3]
dupecutoff	The duplicate score lower boundary for non-exact matching percentage. For example, if you set the dupecutoff value to 40, then the lowest percentage of a potential duplicate match would be 40%. This can be used in conjunction with -dupepermatchupperlimit to specify a range of matches. **Note** If Exact Match is enabled, this value cannot be set.
dupeexc	Duplicate record detection, column exclusion list
dupeinc	The column limit for duplicate record detection. This can be a comma delimited list of columns to include in your job. For example, if you want to include columns called account_id, date, and frequency, the correct syntax would be `account_id,date,frequency`.
dupelb	Duplicate lower bounds on percent match, default [85]
dupelimit	Limit for dupe rows stored
dupelimitui	Limit for dupe rows displayed
dupenocase	Duplicate record case sensitivity off
dupeonly	Only run duplicate section
dupeper-matchupperlimit	The duplicate score upper boundary for non-exact matching percentage, set to 100 by default.
dupescore	Score for each duplicate record
dupesperdupe	Max dupes to calculate per duplicate match
dupetruecase	Enables case sensitivity.

Parameter	Description
dupeub	Duplicate upper bounds on percent match, default [100]
ec	Add custom escape character to escape SQL queries when returning to database
encoding	Load file charset encoding other than UTF-8
erlq	Explicit k,v string of rule_name and rule sql for secondary datasets
executorcores	Spark executor cores
executormemory	The total Spark executor memory in gigabytes, for example, 3G.
f	File path for load, /dir/filename.csv
files	Pass additional spark files for distribution on cluster
filter	Only use rows containing this value. For more information, see Filter & Filter Not.
filtergram	filtergram
filternot	Exclude rows containing this value. For more information, see Filter & Filter Not.
flatten	Option to flatten json and explode arrays
fllb	File Lookback to check owl check history for previous files. For more information, go to Lookback.
fllbminrow	Minimum number of rows (inclusive) that owl check history needs to be considered for File Lookback. Default 0 (which includes all owlchecks). For more information, go to Lookback.
fpgbucketlimit	Limit bucket size for Pattern algorithm, example: `–fpgbucketlimit 20000`
fpgconfidence	Minimum occurrence rate at which an association rule has to be found to be true
fpgdc	The column in your dataset that you set as the date column.
fpgdupeoff	Pattern mining do not remove dupe cols, helps performance impacts quality
fpgexc	Pattern mining is expensive use this input to limit the observed cols

Parameter	Description
fpginc	The column limit for pattern mining. This can be a comma delimited list of columns to include in your job. For example, if you want to include columns called account_id, date, and frequency, the correct syntax would be account_id,date,frequency Because pattern mining is expensive, limiting the number of columns in your query can be an effective way to control costs.
fpgkey	The natural key for pattern mining. This is the column in your dataset that you set as the key column.
fpglb	The lookback period where the entered value represents the number of days included in the pattern activity lookback. For more information, go to Lookback.
fpglimit	Limit for frequent pattern mining results stored
fpglimitui	Limit for frequent pattern mining results displayed
fpgmatchoff	Turn off match for only patterns that appear in today dataset scope
fpgmulti	Pass multiple fpgkey=fpginc key value pairs. Split by pipe for multiple. To see an example, go to Multiple Pattern Relationships.
fpgon	Enables the pattern (mining) activity.
fpgq	Select * from file (sql)
fpgscore	Score for pattern mining records
fpgsupport	Minimum occurrence rate for an itemset to be identified as frequent
fpgtbin	Time bin for pattern mining, example: `-fpgtbin DAY`
fq	Select * from file (sql). To see an example, go to DQ Job JSON.
fullfile	Use entire file for lookbacks instead of just filequery. Includes the historical context of a single file in the outlier and/or patterns scans. For more information, go to Lookback.
h	The hostname where CDQ is installed. This option is for running DQ jobs remotely.
header	Comma delimited list of headers: fname,lname,price
headercheckoff	Check headers for invalid chars. For more information, see Header Check.

Parameter	Description
help	Print this message
histlimit	Limit for histograms stored
histlimitui	Limit for histograms displayed
histoff	Dataset histogram flag force off
histon	Dataset histogram flag force on
hive	Turn on native Hive connection. For more information, go to DQ Job Hive and DQ Job Validate Source.
hivehwc	Use hive warehouse connector to access data in HDP3.x Hive Warehouse
hootonly	Only display hoot at stdout
hootprettyoff	Hoot json pretty print flag off
host	Owl metadata store host
hudi	hudi file data flag
in	Validate distinct column values against another dataset
inferschemaoff	Turn off inferschema when loading files
iot	Automatically store a numeric column without specifying a tsk, tsv
jars	Spark - Comma-separated list of jars to include on the driver and executor classpaths.
jdbckeytab	Path and location to jdbc principal keytab file
jdbcprinc	Kerberos principal name specifically to connect to Kerberized JDBC, example: owl@OWL.COM
jdbctgt	Path and location to jdbc principal tgt file
jobschemaname	Mainly needed for Big Query, but can be used for any database to set the schema name explicitly versus parsing it out of the sql query later
json	Json data flag

Parameter	Description
kafka	Indicates that the target data source is Kafka
kafkabroker	Kafka port, example: 9092
kafkagroup	Kafka consumer group, example: machine-group
kafkakeyserde	Optional `--kafka_key_deserializer org_apache_kafka_common_serialization_StringSerializer`
kafkaport	Kafka host, example: localhost
kafkasasl	Enable kafka SASL (Kerberos) authentication. If this option is set, also set `kafkasaslservice` flag
kafkasaslservice	The name of the SASL service fr authenticate
kafkassl	Enable kafka SSL 1-way and/or 2-way ssl. If this option is set, also set `ssltruststore/sslkeystore` flags
kafkatopic	Topic, topic name, example: test_stream
kafkavalserde	Optional `--kafka_value_deserializer org_apache_kafka_common_serialization_StringSerializer`
kerbpwd	Kerberos password to aquire TGT
key	Primary key or unique key, typically business key, example : sym,exch (compound use comma)
keyDelim	Delimiter for primary key or unique key when concatenating the values to single string, example: `sym,exch -> sym~~exch`
lib	The library class directory, for example, "/opt/owl/drivers/postrgres/".
libsrc	Library class directory for val src cxn
lickey	Passes lickey from owlcheck to owl-core
linkid	linkid is for client datasets to pass their primary_key or record_id field so that when Owl serves back the results they are linked to the original dataset

Parameter	Description
logfile	Allow user to add their own custom logging
loglevel	The logging level. This can be either `INFO` or `DEBUG`.
lookbacklimit	Limit for lookback intervals
lower	Median Q2 multiplier to impact lower boundary
lowerbound	Number or Timestamp
maps	Contains maps in json that requires extra handling
master	Overrides local[*]
matches	(Deprecated) Show matches in validate source instead of mismatches
maxcolumnlimit	Limit for max columns
minintervals	Minimum streaming intervals for profiling
mixedjson	Contains non-json and json flag
mu	Measurement unit for outliers
multiline	Multiline json flag
notin	Validate distinct column values against another dataset
nulls	`-nulls` 'null' treats 'null' as NULL. For more information, see Nulls in Datasets.
numericlimit	Limit for numeric outliers stored
numericlimitui	Limit for numeric outliers displayed
numericscore	Score for numeric outliers
numexecutors	The number of Spark executors.
numpartitions	Number of partitions or splits for parallel queries, default 3. To see an example, go to DQ Job JDBC.
obslimit	Limit for observations stored

Parameter	Description
obslimitui	Limit for observations displayed
obsscore	Score for observation record
opt	key=value, [escape='', quote='', timestampFormat='yyyy-MM-dd']
optpath	/file/path/to/dsOption.properties [escape=value]
orc	orc file data flag
otlrq	Select * from file (sql)
outlierlimit	Limit for outliers stored
outlierlimitui	Limit for outliers displayed
outlieronly	Only run outlier section
outlierscore	Score for mismatching source to target records
owluser	The username of the CDQ user running the job.
p	Password
packages	Spark - Comma-separated list of maven coordinates of jars to include on the driver and executor classpaths. Will search the local maven repo, then maven central and any additional remote repositories given by --repositories. The format for the coordinates should be groupId:artifactId:version
parallel	Turn on parallel task execution vs sequential (default). Performance advantage but uses more hardware resources
parquet	Parquet file data flag
partitionnum	Number of partitions calculated by estimator/overide by user
passfail	Set the failing score, example: 75
passfaillimit	Limit for passing or failing runs
patternonly	Only run pattern mining section
pgpassword	Password for Owl's postgres metastore

Parameter	Description
pguser	Username for Owl's postgres metastore
pipeline	List of activities to analyze
plan	Turn on execution plan. Describes the executions plan
port	Owl metadata store port
postclearcache	Delay clear cache process to the end of owlcheck
precisionoff	Turn Profile Precision Off, do not calculate the length of doubles
profile2	Run inline version of column stats
profileonly	Only run profile and shape section
profilepushdown	Compute profile in the target database
pro-fileStringLength	Profile min/max length for String type columns on
profoff	Turn off profiling section of owlcheck
pwdmgr	lookup a password manager password via script and obtain the password for the JDBC connection
q	The SQL query of your job. For example, `select * from [table].`
q1	The lower quartile boundary impact (IQR) value between 0-0.45. If this is not specified, the lower quartile is 0.15 by default.
q3	The upper quartile boundary impact (IQR) value between 0.55-1. If this is not specified, the upper quartile is 0.85 by default.
qhist	Select * from table (sql)
queue	YARN queue name
rc	Record detection
rcBy	Record compare by function
rcDateCol	Record detection date column

Parameter	Description
rcKeys	Record detection keys
rcTbin	Record detction time bin
rd	The run date of your job in either yyyy-MM-dd or yyyy-MM-dd HH:mm format. For more information, go to Date Time Variable Options.
rdAdj	Dynamically calculates and substitute the run date variable (rd) at the run time of file-based and database connection DQ Jobs to include historical data in your job schedule lookback. The run date adjustment variable uses the following format, where each element defined in the table below should be replaced with an actual value: `-rdAdj [XX]:[+/-][NNN]`

Element	Description
[XX]	The date time variable. Accepts yyyy, MM, dd, hh, mm, and ss.
[+/-]	Determines whether the [XX] element will be added or subtracted from the run date.
[NNN]	The number of units to add or subtract.

> **Example**
> `-rdAdj dd:-1 -df "yyyy-MM-dd" -f "example-dataset/nyse=${yyyy}-${MM}-${dd}/part*.csv -rd "2018-01-20"` substitutes 2018-01-19, which is one day prior to the run date of 2018-01-20, as specified by the -rdAdj variable.

> **Note**
> If you are using a file without a date format (-df), yyyy-MM-dd is used by default. If a date format is not set, it needs to be adjusted in the command line and parameterized in the file name with `${}` for each date time variable. For example, `-f "example-dataset/nyse=${yyyy}-${MM}-${dd}/part*.csv`. The parameterized variables must match the format used in the -df variable.

Parameter	Description
rdEnd	End date for query ranges `t_date >= ${rd}` and `t_date < ${rdEnd}`, must be in format 'yyyy-MM-dd' or for incremental use Hours or Minutes yyyy-MM-dd HH:mm
readonly	Do not connect to meta store good for testing or trials
record	Validate distinct column values against runs
recordoff	Check for records that were added or dropped from dataset
repartitionoff	Do not repartition
rlc	Rule secondary src jdbc://hive:3306 (connection URL)
rld	Rule secondary src driver path
rlds	srcDataset (silo.account)
rlp	Rule secondary src password
rlq	Rule secondary src SQL
rlu	Rule secondary src username
rootds	Context based predictions you can assign a root dataset, example: `user -> userLoan, user -< userCredit. rootds = user`
rulename	Only for rules validation testing to run single rule
ruleserial	Run rules in serial mode
rulesoff	Rules section flag off
rulesonly	Only run rules section off
schem-aregistrypass	Password to login to schema registry where stream schema can be found
schemaregistryurl	url of schema registry where stream schema can be found
schem-aregistryuser	Username to login to schema registry where stream schema can be found

Parameter	Description
schemascore	Score for schema changes
scorecardsize	Limit for size of scorecard displayed
sdriver	Classname for custom secondary driver entered by user for complex rule
selectall	Select * override cols
semanticoff	Semantic forced off
semanticon	Semantic forced on
skipfirstrow	Indicates that the first row contains header values
skiplines	Skip first N lines of a csv file before loading
sourceonly	Only run validate source section
sp	Sample percentage [0.0 - 1.0], default value 1.0 = 100%
sparkkeytab	Path and location to keytab file
sparkprinc	Kerberos principal name, example: owl@OWL.COM
sq	Single quote (') character to escape SQL queries when returning to database
srcauto	Auto generates validate source params from owl check history. Only needs –`srcds` and –`valsrcfq` or –`q`
srcavro	avro file data flag for source
srcavroschema	Validate source avro schema file
srcc	jdbc://hive:3306 (connection URL)
srccxn	Instead providing the user, pass, connectionurl for a connection, provide the saved connection name for validate source
srcd	src driver oracle.driver.JDBC
srcdel	Source delimiter ,
srcdelta	Delta file data flag for source

Parameter	Description
srcds	srcDataset (silo.account)
srcencoding	Load source file charset encoding other than UTF-8
srcfile	Validate source file
srcflatten	Option to flatten json and explode arrays for source
srcfullfile	Use entire file for lookbacks instead of just filequery for source
srcheader	Validate source header for a file
srchive	-srchive for validate source on Hive using HCat non JDBC
srcinferschemaoff	Turn off inferschema when loading files or source
srcjson	json data flag for source
srcjsonmaps	Contains maps in json that requires extra handling for source
srcmixedjson	Validate source contains non-json and json flag
srcmultiline	Multiline json flag for source
srcorc	orc file data flag for source
srcp	src password
srcparquet	Parquet file data flag for source
srcpwdmgr	Lookup a password manager password via script and obtain the password for the JDBC connection
srcq	src SQL
srcskiplines	Skip first N lines of a source csv file before loading ()
srcu	src username
srcxml	Xml data flag for source
srcxmlrowtag	Xml Row Tag for source

Parameter	Description
sslciphers	Comma separated list of valid ciphers for the target secure socket connection
ssldisablehostverify	Disable SSL hostname verification when deciding whether to trust the host's certificate
sslkeypass	ssl key password (Only required when ssl key stored in keystore has a password)
sslkeystore	Location of the ssl keystore
sslkeystorepass	ssl keystore password
sslkeystoretype	Type of the ssl keystore (Default: JKS)
ssltruststore	Location of the ssl truststore
ssltruststorepass	ssl truststore password
ssltruststoretype	Type of the ssl truststore (Default: JKS)
statsoff	Column stats flag off, on by default
stock	Optimized for stock data, price history
stream	Indicates that the target data source is a stream of data
streamformat	Format, example: `csv,avro,json,xml`
streaminterval	Interval, in second format, example: 10
streammaxlull	The maximum time in seconds that a stream should not be empty
streamprops	`key-y=value,hive.resultset.use.unique.column.names=false`
streamschema	`col:integer,col1:double,col2:string,col3:long`
streamtype	Type of stream. Possible values: `Kafka`
stringmode	All data types forced to strings for type safe processing
t1	Select * from @dataset.column (sql)

Parameter	Description
t1q	Select * from @dataset.column (sql)
tbin	MIN -> minute [14:27], HOUR -> hour military [13], DAY -> [05], SEC -> Second [14:27:35]. To see an example, go to Advanced and DQ Job Back Run.
tbq	Select * from dataset where time_bin = '2018-12-10 10'. Ex: for time bin outliers override.
tc	Time Column for cases when date time are separate
timestamp	Converts timestamp column to date format. Uses -dc date column flag as column to convert. Must be accompnaied with -transform flag to transform string to DateType/TimestampType
todq	Select * from dataset where time_bin = '2018-12-10 10', example: for today override.
transform	Transform expressions. can be on or delimited by |. Example: colname=cast(colname as string),colname2=colname2(cast as date). For more information, go to Transform.
ts	Flag this dataset as a Time-Series dataset
u	Username
ubon	Use boundaries flag off
upper	Median Q2 multiplier to impact higher boundary
upperbound	Number or Timestamp
usespark	usespark flag, forces spark, intended for datasets > 30 mil rows
usesql	usesql implies to use the -q select * from table where etc as a subselect of the partitioning
usetemplate	Does not require cmd line params uses saved properties, can override by adding them. To see an example, go to DQ Job Chron.
validates-chemaorderon	Validate source column name order

Parameter	Description
validatevalues	Validate source matches on cell values and show mismatches
validatevaluesfilter	Spark sql where clause to limit rows to validate values, example: "id = 123"
validate-valuesignoreempty	Validate value ignores empty string as an issue
validate-valuesignorenull	Validate values ignores null as an issue
validate-valuesignoreprecision	Validate value ignore precision for decimal values
validate-valuesshowall	Validate values shows findings for all columns instead of one per row
validate-valuesshow-missingkeys	Provide options to show missing keys on both target and source for validating source with key case
validate-valuesthreshold	Validate value threshold ratio. Default .9 (=90%)
validate-values-threshold-strictdownscore	Validate values turn on strict downscore for threshold category
validatevaluestrimon	Provide options to trim extra space for source target join and cell to cell comparison

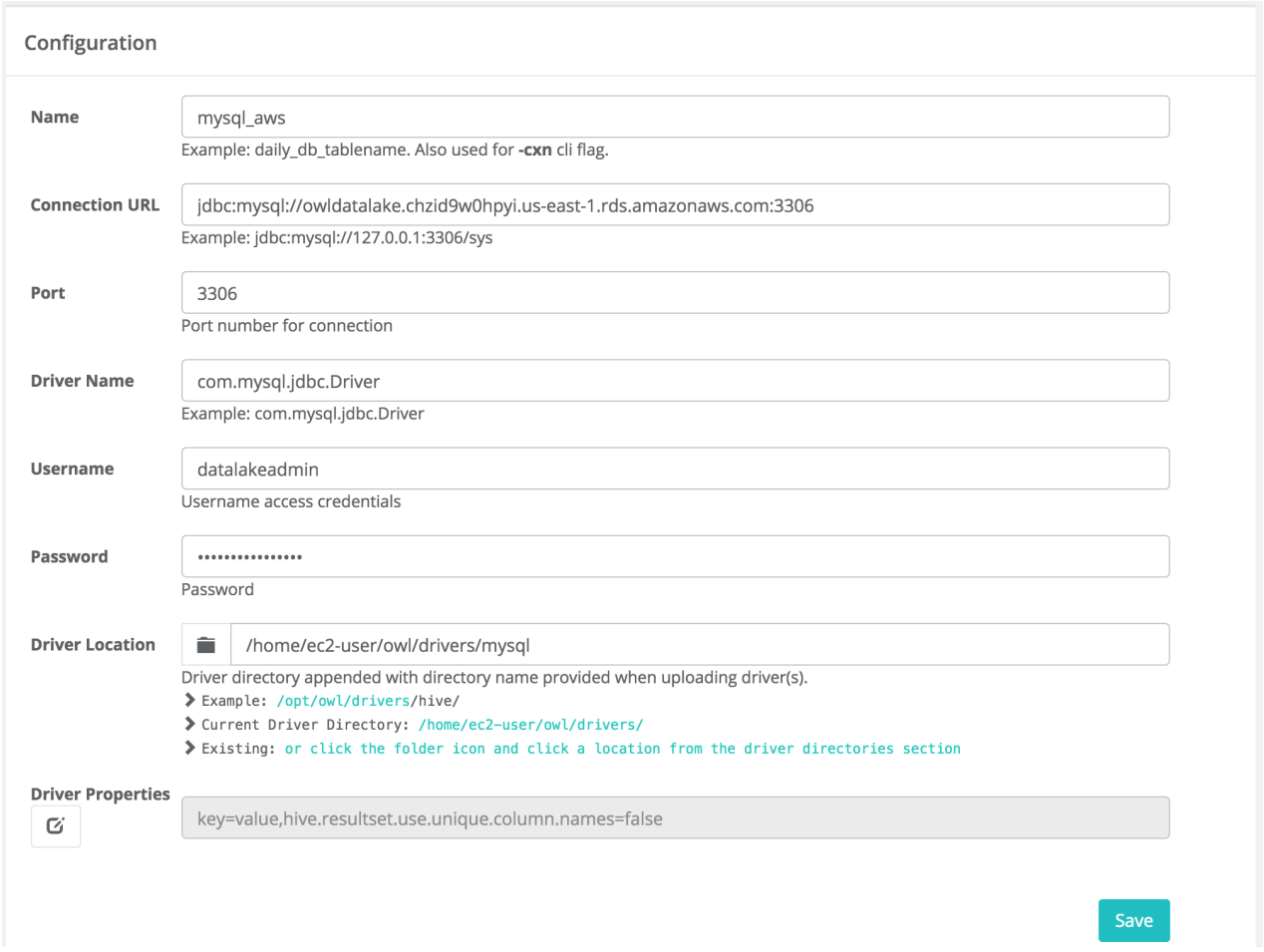
Parameter	Description
valsrccaseon	Validate source column name case sensitivity off
valsrcexc	Validate source column exclusion list for target dataset
valsrcexcsrc	Validate source column exclusion list for source dataset
valsrcfq	Validate source file query
valsrcinc	Validate source column inclusion list for target dataset
valsrcincsrc	Validate source column inclusion list for source dataset
valsrcjoinonly	Skip validate source row count, schema comparison and validate values, pair use with postcacheclear
valsrckey	Validate source column key list for target dataset
valsrclimit	Limit for validate source stored
valsrclimitui	Limit for validate source displayed
valsrcmap	Validate source file column mapping (sourceCol-l=targetCol,sourceCol2=targetCol2)
valsrcpdc	Push row count to source database
valsrctypeoff	Validate source don't check for schema type
version	0.1
vs	Turn on validate source
where	Allows you to place a common where clause and still accept partitioning
xml	Xml data flag
xmlRowTag	Xml Row Tag
zfn	Zero fill null, NULL values will be 0.0. For more information, see Nulls in Datasets.
zkhost	zk host
zkpath	zk path

Parameter	Description
zkport	Zookeeper port

DQ Job JDBC

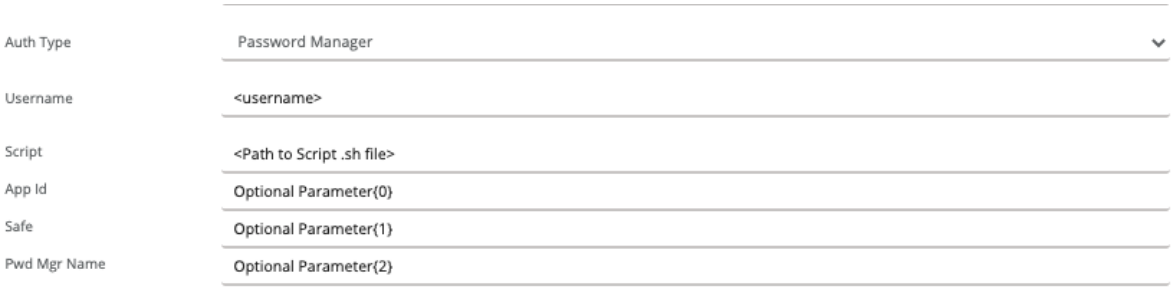
Connect to any database via JDBC.

```
-q "select * from lake.stock_eod where date = '2017-01-20' " \
-u username -p password \
-c "jdbc:mysql://instance.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306" \
-rd "2017-01-20" \
-dc "date" \
-ds "stocks" \
-driver com.mysql.jdbc.Driver \
-lib "/home/ec2-user/owl/drivers/mysql/"
```



Password Manager

You can configure Collibra DQ to call a script file to retrieve a password from a password manager vault or other storage container. The customer is responsible for generating a script to pull just the password and DQ uses that value dynamically when the connection is needed for the UI or when kicking off an DQCheck.



In the connection dialog, select Password Manager from the Auth Type dropdown, and supply a user name. The script is the path to the .sh script on the machine where the web application is running, and the user account that runs DQ-web should be allowed to execute the script. You can either use the optional parameters or pass any parameters your script needs directly inline on the Script value.

Fetch Size

It is important to consider the drivers fetch size when loading greater than 1 Million rows across the network. DQ allows you to set this driver property in the WebApp but this is only for web interaction therefore "fetchsize" will not help here. DQ also allows fetchsize in the DQCheck by passing in a connection property.

CMD line

```
-connectionprops "fetchsize=3000"
```

Notebook

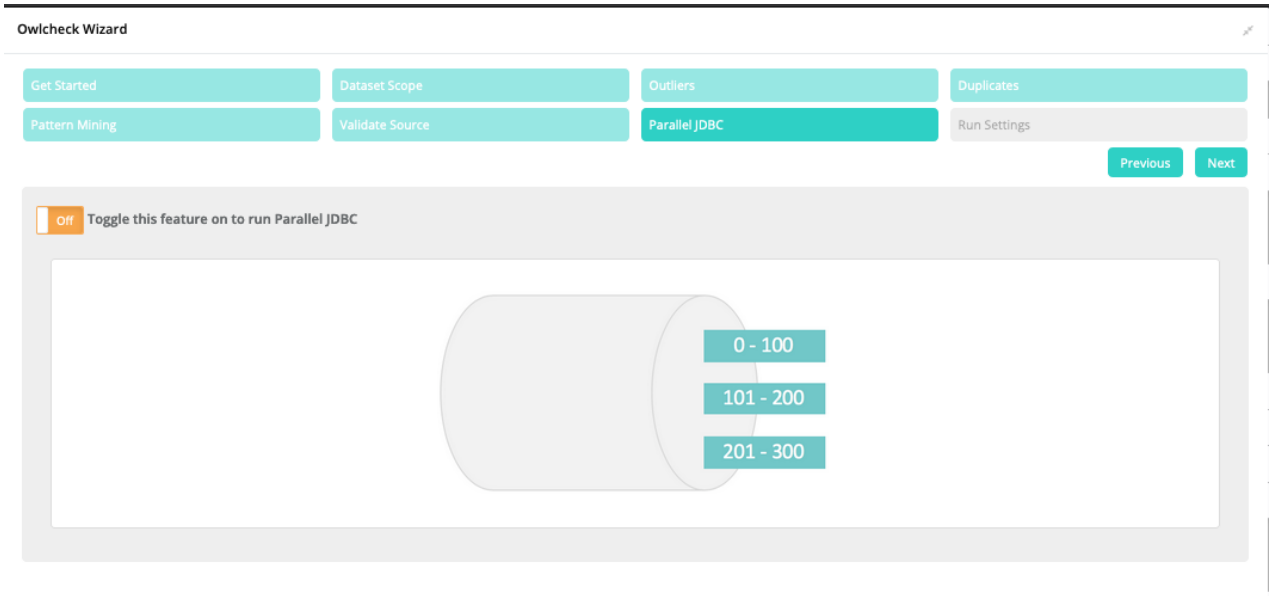
```
props.connectionProps.put("fetchsize", "3000")
```

Parallel JDBC

For greater performance or moving large datasets across a network DQ supports parallel JDBC, which can be enabled by passing `numpartitions` to DQCheck. This can be a 2-5X improvement in many cases.

```
-lib "/opt/owl/drivers/mysql8/"
-cxn mysql
-q "select * from lake.nyse where trade_date = '${rd}' "
-rd 2018-01-01
-ds nyse
-columnname volume
-numpartitions 4
-lowerbound "0"
-upperbound "5000000000"
-usesql
```

DQ also supports auto parallelization, which will configure the `numPartitions` parameter for you based on the size of your data. This is enabled in the UI when you create a dataset using the DQCheck wizard.



DQ Job Hive

Run a data quality check on a Hive table. Use the -hive flag for a native connection via the HCat, this does not require a JDBC connection and is optimized for distributed speed and scale.

Hive Native, no JDBC Connection

Open source platforms like HDP, EMR and CDH use well known standards and because of this DQ takes advantage of things like HCat and it removes the need for JDBC connection details as well as offers optimum data read speeds. DQ recommends and supports this with the -hive flag.

```
./owlcheck -ds hive_table -rd 2019-03-13 \
-q "select * from hive_table" -hive
```

Example output. A hoot is a valid JSON response

```
{
  "dataset": "hive_table",
  "runId": "2019-02-03",
  "score": 100,
  "behaviorScore": 0,
  "rows": 477261,
  "prettyPrint": true
}
```

Hive JDBC

1. You need to use the hive JDBC driver, commonly org.apache.hive.HiveDriver.
2. You need to locate your driver JDBC Jar with the version that came with your EMR, HDP or CDH
    a. This jar is commonly found on an edge node under /opt/hdp/libs/hive/hive-jdbc.jar etc...

```
./owlcheck -rd 2019-06-07 -ds hive_table \
-u <user> -p <pass> -q "select * from table" \
-c "jdbc:hive2://<HOST>:10000/default" \
-driver org.apache.hive.HiveDriver \
-lib /opt/owl/drivers/hive/ \
-master yarn -deploymode client
```

HDP Driver - org.apache.hive.HiveDriver

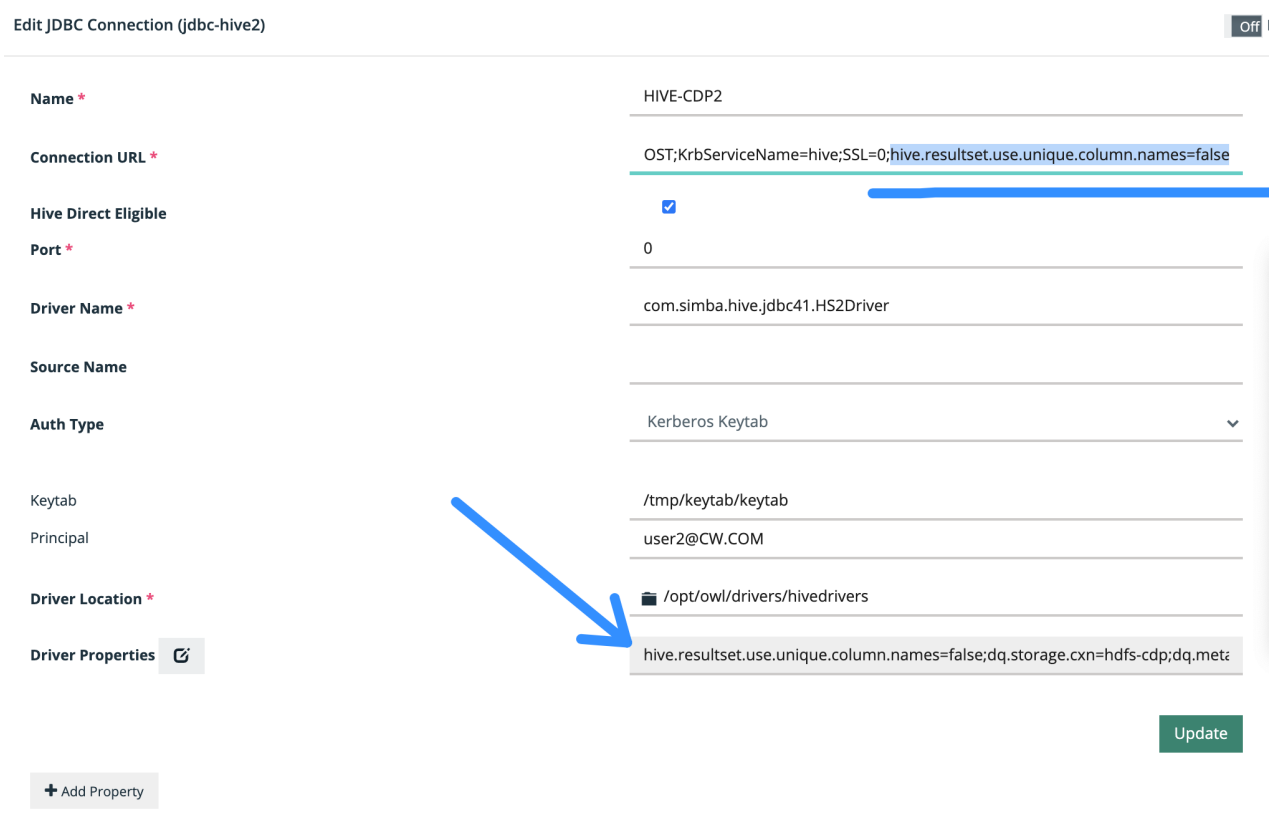
CDH Driver - com.cloudera.hive.jdbc41.Datasource

For CDH all the drivers are packaged under, HiveJDBC41_cdhversion.zip.

Troubleshooting

A common JDBC connection is hive.resultset.use.unique.column.names=false.

This can be added directly to the JDBC connection url string or to the driver properties section.



Test your hive connection via beeline to make sure it is correct before going further.

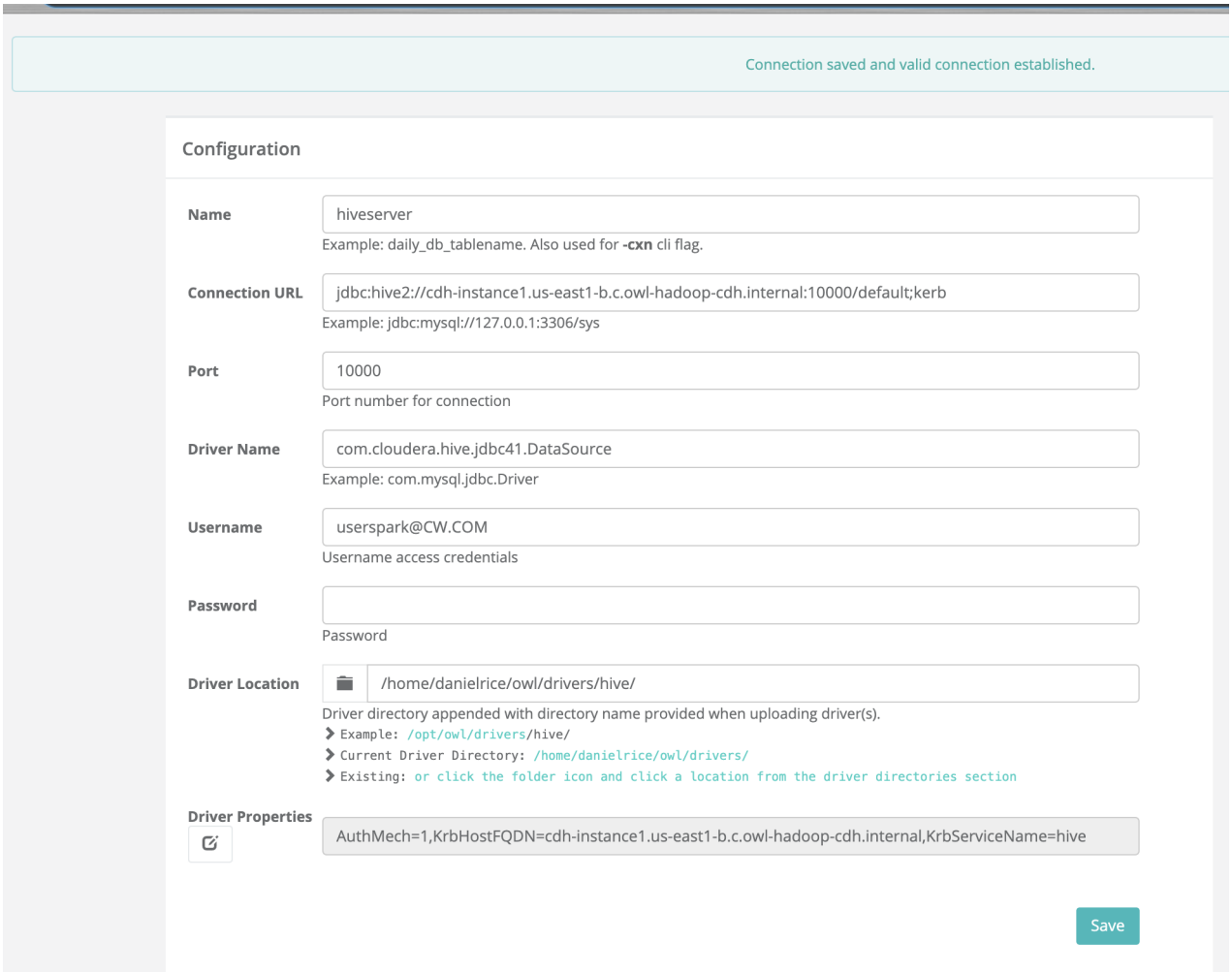
```
beeline -u 'jdb-
c:hive2://<HOST>:10000/default;principal=hive/cdh-instance1.us-
east1-b.c.company-hadoop-cdh.internal@CW.COM;useSSL=true' -d
org.apache.hive.jdbc.HiveDriver
```

Kerberos Example

```
jdbc:hive2://<HOST>:10000/default;principal=hive/cdh-
instance1.us-east1-b.c.company-hadoop-cdh.in-
ternal@CW.COM;useSSL=true
```

Connecting DQ WebApp to Hive JDBC

Notice the driver properties for kerberos and principals.



In very rare cases where you can't get the jar files to connect properly one workaround is to add this to the DQ-web startup script.

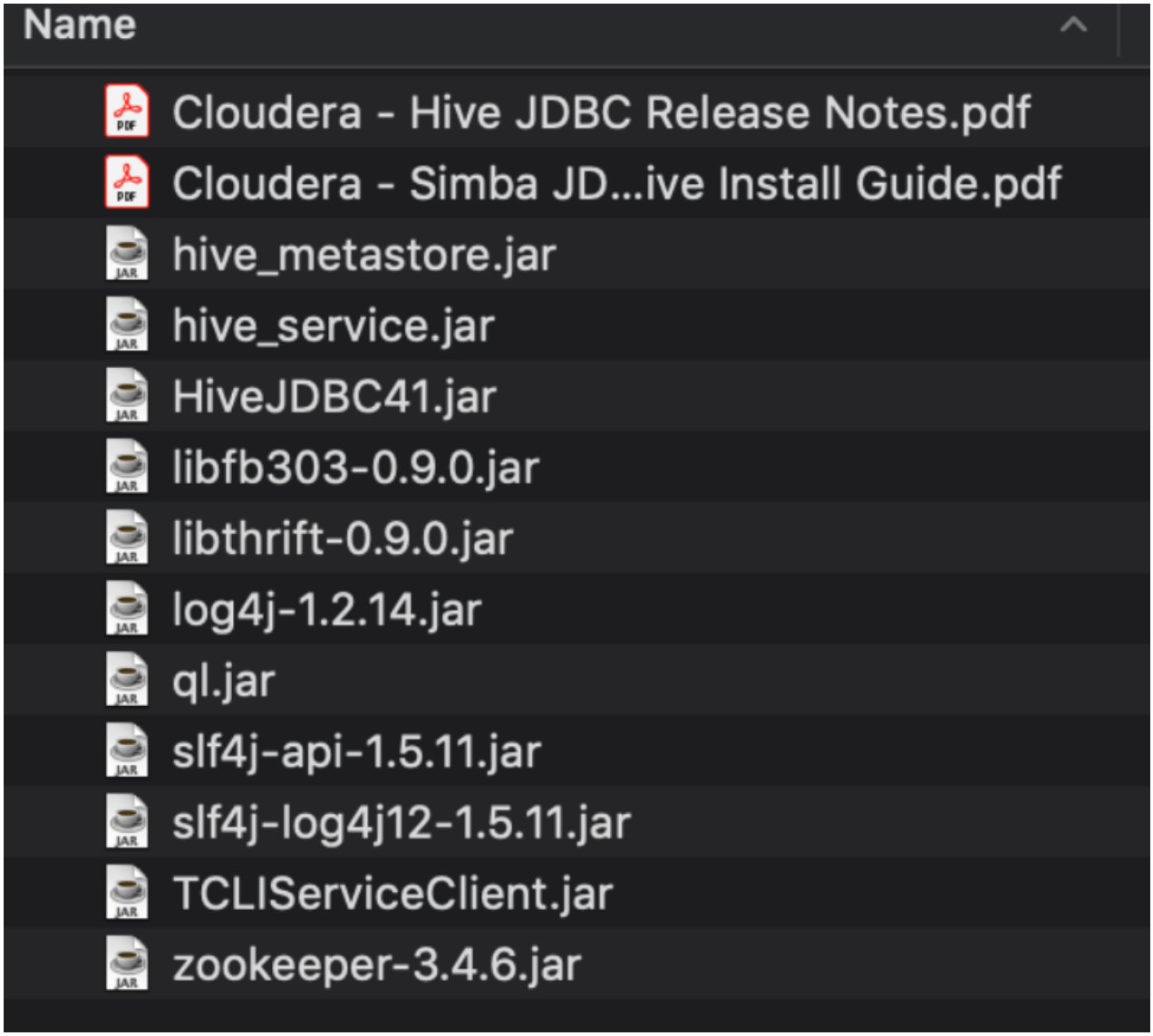
```
$JAVA_HOME/bin/java -Dload-
er.path=lib,/home/danielrice/owl/drivers/hive/ \
-DowlAppender=owlRollingFile \
-DowlLogFile=owl-web -Dlog4j.configurationFile=file://$INSTALL_
PATH/config/log4j2.xml \
$HBASE_KERBEROS -jar $owlweb $ZKHOST_KER \
--logging.level.org.springframework=INFO $TIMEOUT \
--server.session.timeout=$TIMEOUT \
--server.port=9001 > $LOG_PATH/owl-web-app.out 2>&1 & echo $!
>$INSTALL_PATH/pids/owl-web.pid
```

Class Not Found apache or client or log4j etc...

Any class not found error means that you do not have the "standalone-jar" or you do not have all the jars needed for the driver.

Hive JDBC Jars

It is common for Hive to need a lot of .jar files to complete the driver setup.



Java jar cmds

Sometimes it is helpful to look inside the jar and make sure it has all the needed files.

```
jar -tvf hive-jdbc.jar
```

DQ Job Files

For example, a large file `transaction_2021-01-01.csv` might contain the following transaction data with two transaction per day spanning all of January.

transaction_id	account_id	date	amount
1	1	2021-01-01	100
2	2	2021-01-01	120
3	1	2021-01-02	90
4	2	2021-01-02	115
...		...	...
61	1	2021-01-31	100
62	2	2021-01-31	999

and this file might be located on the directory `~/customer/transaction-2021-01-01/`.

```
~/customer
   ├── transaction-2021-01-01
   │   └── transaction_2021-01-01.csv
   ├── transaction-2021-02-01
   │   └── transaction_2021-02-01.csv
   ... # folders for 2021-03-01 to 2021-11-01 ommitted
   ├── transaction-2021-12-01
   │   └── transaction_2021-12-01.csv
```

Other folders with similar pattern may exist in your directory, such as `~/customer/transaction-2021-02-1`. Note that February data is located in a separate directory with a similar pattern for all the months of 2021. This dataset could similarly have 2 account IDs and 1 transaction per account per day (= 28 x 2 = 56 rows of data). For this example, let's assume this is the case for all the files.

To run an DQCheck on this single file containing multiple dates, you have the following choices.

DQChecks with file

1. Run an DQCheck on all the rows in a single file.

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset"
    ... # other relevant options
```

Here we assume that run date (`-rd`) is "2021-01-01" because it is currently January 1, 2021. The above command would lead to an DQCheck on 62 rows of data spanning all of January 2021 from a single file located at `~/customer/transaction-2021-01-01/transaction_20210101.csv`. If you were to schedule a job to run this job monthly and next job ran on February 1st, 2021, then same DQ checks will be performed on the **same set of 62 rows** with same score as your DQCheck run from January 1, 2021. For example, the follow-up scheduled job running on February 1st, 2021 would be:

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-02-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset"
    ... # other relevant options
```

This type of DQCheck on a single file is suitable if you are verifying a static file `~/customer/transaction-2021-01-01/transaction_20210101.csv` that does not change over time and expect the score to be the same every run. Hence, it is suggested to name the dataset that reflect this, such as `DQCheck_transaction_jan21` to reflect the idea that this dataset is checking the Data Quality of transaction table containing January 2021 data. Similar DQCheck for February 2021 data would then be a separate and independent dataset named `DQCheck_transaction_feb21` This type of DQCheck can also be used if `~/customer/transaction-2021-01-01/transaction_20210101.csv` is changing (the rows are changing values or new rows are being added) and want to detect data quality changes. Transaction file doesn't fit with this scenario, but the idea is that the above command

specifies Data Quality DQChecks on the entirety of the file. The run date is a date that you choose to assign for that DQCheck. It is conventional to have one-to-one mapping between run date and the date corresponding to the date that DQ checks are being performed. Run date does not have to match with the data underlying the file.

2. Run an DQCheck on subset of rows from a single file

The single file contains daily data for January of 2021. To run Data Quality checks on January 1st, January 2nd, … , and January 31st, you need to run 31 DQChecks, each with subset of rows from the file. Note the `where` clause in `-fq` matching with the run date `-rd`

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-01'"
    ... # other relevant options

./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-02"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-02'"
    ... # other relevant options

... # DQ checks for -rd 2021-01-03 to 2021-01-30 ommitted

./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-31"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-31'"
    ... # other relevant options
```

By using the same dataset name `-ds`, all 31 DQCheck will appear under one dataset `DQCheck_transaction_jan21` on the Findings page.

A convenient way to parameterize this run date is to use `${rd}` in the query.

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '${rd}'"
    ... # other relevant options
```

A daily scheduled job starting on January 1st, 2021 to January 31, 2021 will automatically replace the `${rd}` with "2021-01-01", "2021-01-02", ... , and "2021-01-31" for the respective run date.

3. Run an DQCheck on subset of rows from a single file with day lookback

For certain core components like **Outlier**, a set of rows corresponding to historical training data can be used to establish a baseline. For example, the row with `transaction_id` 62 has amount of 999. This looks like an outlier that we want to catch. This value of 999 seems to be an outlier because past transaction amounts for `account_id`2 are in the 100s range. We can use historical data from January 15th to January 30th and use that info to see if January 31st data contains any outliers. In this scenario, our single file `~/customer/transaction-2021-01-01/transaction_2021-01-01.csv` contains such historical data because that file contains all the data for all of January. How do we use the same file for both current data (January 31st) and historical (January 15th to January 30th) data? You do not have to split the files into two. You can simply do exactly what you would do for DQCheck on "2021-01-31" with a `-fullfile` flag. The `-fullfile` flag tells the DQCheck that "the file in `-f` contains the historical data. Construct a query and subset those rows for me".

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-31"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-31'"
    -fullfile
    # outlier options
    -dc "date"
    -dl
    -tbin "DAY" # look back time bin is day
    -dllb 15 # look back up to 15 days
    ... # other relevant options
```

DQChecks with *multiple files*

4. Run an DQCheck on a single file with lookback using series of file

Recall our folder structure:

```
~/customer
   ├── transaction-2021-01-01
   │   └── transaction_2021-01-01.csv
   ├── transaction-2021-02-01
   │   └── transaction_2021-02-01.csv
  ... # ommitted for space
   ├── transaction-2021-12-01
   │   └── transaction_2021-12-01.csv
```

If we want to run an DQCheck for December of 2021 and use July of 2021 to November of 2021 as our historical training dataset, how can we load *multiple files*? Just like how `-fullfile` provides a convenient way to create historical training dataset on a single file, `-fllb` (file lookback) provides a convenient way to load *series of files* with patterns while still pointing to the target file (December file) in `-f`

```
./owlcheck
    -ds DQCheck_transactions_dec21
    -rd "2021-12-01"
    -f "~/customer/transaction-2021-12-01/transaction_2021-12-
01.csv"
    -fq "select * from dataset"
    -fllb
    # outlier options
    -dc "date"
    -dl
    -tbin "MONTH"
    -dllb 5 # look back up to 5 months
    ... # other relevant options
```

One caveat to this `-fllb` method __ is that the DQCheck history must be "primed" first so that the DQ knows the file path of the past series of files. In fact, `-fllb` does not use the file path provided in `-f` and loads different files from different folders. *It relies on the DQCheck history* under the same `-ds` name. `-fllb` means lookback up to N number of past consecutive DQChecks. For each of those past DQCheck, look up the file path `-f` used in the past and follow those paths. The number N is determined by the maximum number of lookbacks from

Outlier (`-dllb`) and Patterns (`-fpglb`). In the DQCheck above, because `-dllb 5` is provided along with `-fllb`, it means "Look up 5 past DQChecks and load those files as historical dataset". In summary, in order to run an DQCheck on "2021-12-01" and have that DQCheck for that date "look up" the files in `~/customer/transaction-2021-07-01/transaction_2021-07-01.csv,~/customer/transaction-2021-08-01/transaction_2021-08-01.csv,~/customer/transaction-2021-09-01/transaction_2021-09-01.csv,~/customer/transaction-2021-10-01/transaction_2021-10-01.csv,` and `~/customer/transaction-2021-11-01/transaction_2021-11-01.csv`, you need to have ran DQChecks for "2021-07-01", "2021-08-01", ... , and "2021-11-01" under the same dataset name. Therefore, it would be more logical, best-practice is to name the dataset `-ds DQCheck_transaction_2021` and run series of monthly owlchecks up to "2021-12-01" (but the name of the dataset is up to you)/

```
# Prime past DQ checks so that "2021-12-01" knows the file path
of past months
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-07-01"
    -f "~/customer/transaction-2021-07-01/transaction_2021-07-
01.csv"
    -fq "select * from dataset"

./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-08-01"
    -f "~/customer/transaction-2021-08-01/transaction_2021-08-
01.csv"
    -fq "select * from dataset"


./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-09-01"
    -f "~/customer/transaction-2021-08-01/transaction_2021-09-
01.csv"
    -fq "select * from dataset"

./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-10-01"
    -f "~/customer/transaction-2021-08-01/transaction_2021-10-
01.csv"
    -fq "select * from dataset"

./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-11-01"
    -f "~/customer/transaction-2021-11-01/transaction_2021-11-
01.csv"
    -fq "select * from dataset"

# Priming 5 past DQ checks complete. Now run the 2021-12-01
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-12-01"
    -f "~/customer/transaction-2021-12-01/transaction_202-11-
201.csv"
    -fq "select * from dataset"
    -fllb
    # outlier options
    -dc "date"
    -dl
```

```
    -tbin "MONTH"
    -dllb 5 # look back up to 5 months
    ... # other relevant options
```

In this scenario, since the folder paths have a pattern, we can use `-br` for priming in one command instead of writing five DQCheck commands. The flag `-br` runs DQChecks consecutively from the past and increments by monthly if `-tbin "MONTH"` (by default `-tbin DAY`so the default behavior is to increment daily). The different folder paths on each past consecutive run dates are replaced with `${rd}`. The below command is identical to the above:

```
# Prime past DQ checks so that "2021-12-01" knows the file path
of past months
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-12-01"
    -f "~/customer/transaction-${rd}/transaction_${rd}.csv"
    -fq "select * from dataset"
    -br 5 # run 5 runs to in the past
    -tbin "MONTH" # <-- required since we want a MONTHLY back-
run. Default "DAY"

# Priming complete. Now run the 2021-12-01
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-12-01"
    -f "~/customer/transaction-2021-12-01/transaction_2021-12-
01.csv"
    -fq "select * from dataset"
    -fllb
    # outlier options
    -dc "date"
    -dl
    -tbin "MONTH"
    -dllb 5 # look back up to 5 months
    ... # other relevant options
```

This pattern is designed so that a single DQCheck command can be scheduled and `${rd}` be used to replace the folder & file path. Your `~/customer` folder could contain transactions for all the years, spanning all the way back to 1992 and into the future like so:

```
~/customer

   ├── transaction-1992-01-01
   │   └── transaction_1992-01-01.csv
   ├── transaction-1992-01-01
   │   └── transaction_1992-01-01.csv
   ... # ommitted for space
   ├── transaction-2021-01-01
   │   └── transaction_2021-01-01.csv
   ├── transaction-2021-02-01
   │   └── transaction_2021-02-01.csv
   ... # ommitted for space
   ├── transaction-2021-12-01
   │   └── transaction_2021-12-01.csv
   ... # hasn't happened yet!
```

In this scenario, a monthly scheduled job would get rid of the need to "prime" the DQCheck history, since your past scheduled jobs would have already ran the past DQChecks.

DQ Job JSON

Files

Run against a file using -json. Additionally, options are available for -flatten and -multiline. This is helpful for nested and various formats.

```
-ds json_file_example \
-f s3a://bucket_name/file.json \
-h instance.us-east4-c.c.owl-node.internal:5432/postgres \
-master spark://instance.us-east4-c.c.owl-node.internal:7077 \
-json \
-flatten \
-multiline
```

> **Note** Automatic flattening will infer schema and explode all structs, arrays, and map types.

Using Spark SQL

```
-ds public.json_sample \
-lib "/opt/owl/drivers/postgres/" \
-h instance.us-east4-c.c.owl-node.internal:5432/postgres \
-master spark://instance.us-east4-c.c.owl-node.internal:7077
-q "select * from public.jason"
-rd "2021-01-17"
-driver "org.postgresql.Driver"
-cxn postgres-gcp
-fq "select  \
get_json_object(col_3, '$.data._customer_name') AS  `data_cus-
tomer_name` , \
get_json_object(col_3, '$.data._active_customer') AS  `data_act-
ive_customer` , \
from dataset "
```

> **Note** Pass in the path to Collibra DQ's -fq parameter. This is great for mixed data types within a database. For example, if you store JSON data as a string or a blob among other data.

```
// Flatten
val colArr = new JsonReader().flattenSchema(df.schema)
colArr.foreach(x => println(x))
```

> **Note** This Collibra DQ utility traverses the entire schema and print the proper get JSON object spark sql strings. You can use this instead of typing each query statement into the command line -fq parameter as seen above.

Using DQ Libraries

```
import com.owl.common.options._
import com.owl.core.util.OwlUtils
import com.owl.core.activity2.JsonReader

val connProps = Map (
  "driver"   -> "org.postgresql.Driver",
  "user"     -> "user",
  "password" -> "password",
  "url"      -> "jdbc:postgresql://10.173.0.14:5432/postgres",
  "dbtable"  -> "public.data"
)

// Spark
var rdd = spark.read.format("jdbc").options
(connProps).load.select($"col_name").map(x=>x.toString()).rdd
var df = spark.read.json(rdd)

// Flatten
val colArr = new JsonReader().flattenSchema(df.schema)
val flatJson = df.select(colArr: _*)
flatJson.cache.count

// Opts
val dataset = "json_example"
val runId = s"2021-01-14"
val opt = new OwlOptions()
opt.dataset = dataset
opt.runId = runId
opt.datasetSafeOff = true

// DQ check
OwlUtils.resetDataSource("instance.us-east4-c.c.owl-node.in-
ternal","5432/postgres","user","pass", spark)
val owl = OwlUtils.OwlContext(flatJson, opt)
owl.register(opt)
owl.owlCheck
```

> **Note**
> JsonReader()
> This uses DQ's JsonReader to do the heavy lifting.

Known Limitations

- Array and nested array datatypes in JSON files are not supported.

DQ Job S3

S3 permissions need to be setup appropriately.

> **Note** S3 connections should be defined using the root bucket. Nested S3 connections are not supported.

Example Minimum Permissions

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketMultipartUploads",
                "s3:ListBucket",
                "s3:ListMultipartUploadParts",
                "s3:GetObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:athena:*:<AWSAccountID>:workgroup/prima-
ry",
                "arn:aws:s3:::<S3 bucket name>/*",
                "arn:aws:s3:::<S3 bucket name>",
                "arn:aws:glue:*:<AWSAccountID>:catalog",
                "arn:aws:glue:*:<AWSAccountID>:database/<databas-
e name>",
                "arn:aws:glue:*:<AWSAccountID>:table/<database
name>/*"
            ]
        }
    ]
}
```

(Needs appropriate driver) http://central.maven.org/maven2/org/apache/hadoop/hadoop-aws/ Hadoop AWS Driver hadoop-aws-2.7.3.2.6.5.0-292.jar

```
-f "s3a://s3-location/testfile.csv" \
-d "," \
-rd "2018-01-08" \
-ds "salary_data_s3" \
-deploymode client \
-lib /home/ec2-user/owl/drivers/aws/
```

Databricks Utils Or Spark Conf

```
val AccessKey = "xxx"
val SecretKey = "xxxyyyzzz"
//val EncodedSecretKey = SecretKey.replace("/", "%2F")
val AwsBucketName = "s3-location"
val MountName = "kirk"

dbutils.fs.unmount(s"/mnt/$MountName")

dbutils.fs.mount(s"s3a://${AccessKey}:${SecretKey}@${AwsBuck-
etName}", s"/mnt/$MountName")
//display(dbutils.fs.ls(s"/mnt/$MountName"))

//sse-s3 example
dbutils.fs.mount(s"s3a://$AccessKey:$SecretKey@$AwsBucketName",
s"/mnt/$MountName", "sse-s3")
```

Databricks Notebooks using S3 buckets

```
val AccessKey = "ABCDED"
val SecretKey = "aaasdfwerwerasdfB"
val EncodedSecretKey = SecretKey.replace("/", "%2F")
val AwsBucketName = "s3-location"
val MountName = "abc"

// bug if you don't unmount first
dbutils.fs.unmount(s"/mnt/$MountName")

// mount the s3 bucket
dbutils.fs.mount
(s"s3a://${AccessKey}:${EncodedSecretKey}@${AwsBucketName}",
s"/mnt/$MountName")
display(dbutils.fs.ls(s"/mnt/$MountName"))

// read the dataframe
val df = spark.read.text(s"/mnt/$MountName/atm_customer/atm_cus-
tomer_2019_01_28.csv")
```

Advanced

A DQ Check is a bash script that is essentially the launch point for any DQ Job to scan a data set. A data set can be a flat file, such as textfile, json file, parquet file, etc, or a table from any number of databases, such as Oracle, Postgres, Mysql, Greenplum, DB2, SQLServer, Teradata, etc.

Example Run a data quality check on any file by setting the file path.

```
./owlcheck -ds stock_trades -rd 2019-02-23 -f /path/to/file.csv
-d ,
```

Example output below. A hoot is a valid JSON response

```
{
  "dataset": "stock_trades",
  "runId": "2019-02-03",
  "score": 100,
  "behaviorScore": 0,
  "rows": 477261,
  "passFail": 1,
  "peak": 1,
  "dayOfWeek": "Sun",
  "avgRows": 0,
  "cols": 5,
  "activeRules": 0,
  "activeAlerts": 0,
  "runTime": "00:00:23",
  "dqItems": {},
  "datashapes": [],
  "validateSrc": [],
  "alerts": [],
  "prettyPrint": true
}
```

Monthly Data

Sometimes you may want to run monthly profiles with aggregated data. In this case, the scheduling tool can supply the $ as a variable such as $runDate and the end date as $endDate. 1 line examples for bash or shell below.

```
echo "Hello World Collibra DQ"

runDate=$(date +"%Y-%m-%d")
endDate=$(date -d "$runDate +1 month" +%Y-%m-%d)

echo $runDate
echo $endDate

./owlcheck \
-q "select * from table where date >= '$runDate' and date <
'$endDate' " \
-ds example \
-rd $runDate \
-tbin MONTH
```

Monthly BackRun (Using Collibra Data Quality's built-in Monthly)

Collibra Data Quality has 2 convenient features here:

1. The use of built-in $ and $ removes the need for any shell scripting.
2. Using -br, DQ will replay 20 months of data using this template automatically.

```
./owlcheck \
-q "select * from table where date >= '${rd}' and date <
'${rdEnd}' " \
-ds example
-rd 2019-01-01
-rdEnd 2019-02-01
-tbin MONTH
-br 20
```

Daily Data

One of the most common examples is data loading or running once a day. A job control framework can pass in this value or you can pull it from shell.

```
echo "Hello World Collibra DQ"

runDate=$(date +"%Y-%m-%d")
echo $runDate

./owlcheck \
-q "select * from table where date = '$runDate' " \
-ds example \
-rd $runDate \
-tbin DAY
```

Daily Data (Using Collibra Data Quality's built-in Daily)

```
./owlcheck \
-q "select * from table where date = '${rd}' " \
-ds example \
-rd 2019-03-14
```

Daily Data with Timestamp instead of Date

```
./owlcheck \
-q "select * from table where TS >= '${rd} 00:00:00' and TS <=
'${rd} 23:59:59' " \
-ds example \
-rd 2019-03-14
```

OR Timestamp using $

```
./owlcheck \
-q "select * from table where TS >= '${rd} 00:00:00' and TS <
'${rdEnd} 00:00:00' " \
-ds example \
-rd 2019-03-14 \
-rdEnd 2019-03-15 \
-tbin DAY
```

Hourly Data

```
./owlcheck \
-q "select * from table where TS >= '${rd}' and TS < '${rdEnd}'
" \
-ds example \
-rd    "2019-03-14 09:00:00" \
-rdEnd "2019-03-14 10:00:00" \
-tbin HOUR
```

DQ Check Template with Service Hook

The best practice is to make a generic job that would be repeatable for every DQ Check. Below is an example that first hits Collibra Data Quality using a REST call and then runs the response.

```
curl -X GET "http://$host/v2/-
getowlchecktemplate?dataset=lake.loan_customer" \
-H "accept: application/json"
```

The above REST call returns the below DQ Check. It is left up to the Job Control to replace the $ with the date from the Job Control system. You can use Collibra DQ's built-in scheduler to save these steps.

```
./owlcheck \
-lib "/home/danielrice/owl/drivers/mysql/" \
-cxn mysql \
-q "select * from lake.loan_customer where load_dt = '${rd}' " \
-key post_cd_num -ds lake.loan_customer \
-rd ${rd} \
-dc load_dt -dl -dlkey usr_name,post_cd_num -dllb 5 \
-tbin DAY -by DAY -dupe -dupeinc ip_address_home,usr_name -
dupecutoff 85 \
-fpgon -fpgkey usr_name,post_cd_num -fpgdc load_dt -fpglb 5 -
fpgtbin DAY \
-loglevel INFO \
-h $host:5432/owltrunk \
-owluser {user}
```

REST API End Point

The easiest option is to use the **runtemplate** end point API call to make requests to from cmdLine or JobControl System. This endpoint gets the DQ Check saved in Collibra instead of the client needing to know the DQ Check details.

POST https://$host/v2/runtemplate?dataset=lake.spotify

RunTemplate

Parameters

Path

dataset	string	name of dataset. -ds OR opt.dataset
rd	string	yyyy-MM-dd format can add time or timezone. if note passed in it will use the current day
rdEnd	string	yyyy-MM-dd format can add time or timezone. if not passed it will not be used

Responses

- 200

```
{
  "msg": "Success, DQ Check is Running as process 13996",
  "pid": "13996",
  "runid": "2017-01-01",
  "starttime": "Thu Oct 17 13:27:01 EDT 2019",
  "cmd": "cmd": "-ds lake.spotify -rd 2019-10-17   -q \"select *
from lake.spotify\" -cxn mysql -lib /opt/owl/drivers/mysql/ -
drivermemory 2G -histoff   -owluser {user}",
  "dataset": "lake.spotify"
}
```

Curl example for the above Rest Call

```
TOKEN=$(curl -s -X POST http://$host/auth/signin -H "Content-
Type:application/json" -d "{\"username\":\"$username\", \"pass-
word\":\"$password\"}" | jq -r '.token')

curl -i -H 'Accept: application/json' \
  -H "Authorization: Bearer ${TOKEN}" \
  http://$host/v2/runtemplate?dataset=lake.spotify
```

Bash Script

A generic and repeatable DQCheck script for job schedulers, that hooks into Collibra to get the template.

```
#1 authenticate
curl -sb -X POST -d username={user} -d password={password}
http://$OWL_HOST/login -c cookies.txt

#2 get template
owlcheck_args=$(curl -b cookies.txt -H "accept: applic-
ation/json" -X GET http://$OWL_HOST/v2/-
getowlcheckcmdlinebydataset\?dataset=insurance | sed 's/.*\[\
(.*\)\]/\1/' | sed -e "s/^\"//" -e "s/\"$//"  | sed 's/\\\"\
(.*\)\\\"/\x27\1\x27/')

#3 replace ${rd} with job_run_date
job_run_date="2019-03-14 10:00:00"
owlcheck_args=${owlcheck_args//'${rd}'/$job_run_date}

#4 run owlcheck
eval owlcheck $owlcheck_args
```

For more Information on Collibra Data Quality's Scheduler, visit the DQ Job Cron page.

DQ Job Back Run

How to Replay a Data Test

Many times you will want to see how a dataset plays out over time. This could be five days or five months. Using this slider the tool will automatically create training sets and profiles as well as run any rules or outliers you've put in place.

Quickly Replay 30 days of data, -br 30

Add -br to any DQCheck and replay in time order. Jan 1st, Jan 2nd, Jan 3rd...To do this we need to use the ${rd} variable that DQ provides as a run_date replacement for job control and templates. Also note that if you run from the cmdline you need to escape "$"s. So use \${rd}. If you are running from a Notebook or Java or Scala or the Rest API you do not need to escape the ${rd} variable.

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 -rd "2018-01-14" \
-lib "/opt/owl/drivers/db2/" \
-cxn db2 \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE =
'\${rd}'" \
-br 4
```

Replay 4 Months of data, -br 4 -tbin MONTH

In situations where your data rolls up into Months, you may want to re-run several months of data but not a day at a time. In this case we will use -br with -tbin.

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 \
-rd "2018-01-01" \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE =
'\${rd}'" \
-br 4 \
-tbin MONTH \
-lib "/opt/owl/drivers/db2/" \
-cxn db2
```

Monthly using a range for the entire Month

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 \
-rd "2018-01-01" \
-rdEnd "2018-02-01" \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE >= '${rd}'
and TRADE_DATE < '${rdEnd}'" \
-br 4 \
-tbin MONTH
-lib "/opt/owl/drivers/db2/" \
-cxn db2
```

Monthly on the last day of the month

To create a back run for the last day each month, ensure your run date is the 31st of a month with 31 days in it and the back run time bin is set to `MONTH`. If a month included in the back run does not have 31 days, the back run job will run on the next available day. For example, if September (a month with only 30 days) is included in the scope of the back run, a back run will occur on September 30.

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 \
-rd "2024-12-31" \
-q "select * from OWLDB2.NYSE_STOCKS\
-br 4 \
-brbin MONTH
-lib "/opt/owl/drivers/db2/" \
-cxn db2
```

DQ Job Cron

Template for Job Control

Cron / Autosys / Control M / Oozie

It is common for organization to need to run jobs on a schedule. Below are a few shell tricks to get a date from bash and use an DQCheck with template variables.

Kinit and get run_date from shell or job control variable, pass it into Collibra DQ using $run_date.

```
%sh
run_date=$(date +%Y-%m-%d)
run_date=$(date '+%Y-%m-%d %H:%M')
echo $run_date

#kinit
echo "password" | kinit  userabc@CW.COM

~/owl/bin/owlcheck -q "select * from lake.stock_eod where date =
'$run_date' " \
-u user -p pass \
-c "jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306" \
-rd "$run_date" \
-dc "date" \
-dl \
-dllb 7 \
-dlminhist 2 \
-tbin DAY \
-dlkey sym,exch \
-ds "lake.stock_nasdaq" \
-driver "com.mysql.jdbc.Driver" \
-lib "/home/ec2-user/owl/drivers/mysql/" \
-master yarn -deploymode client -numexecutors 1 -executormemory
1g \
-loglevel DEBUG
```

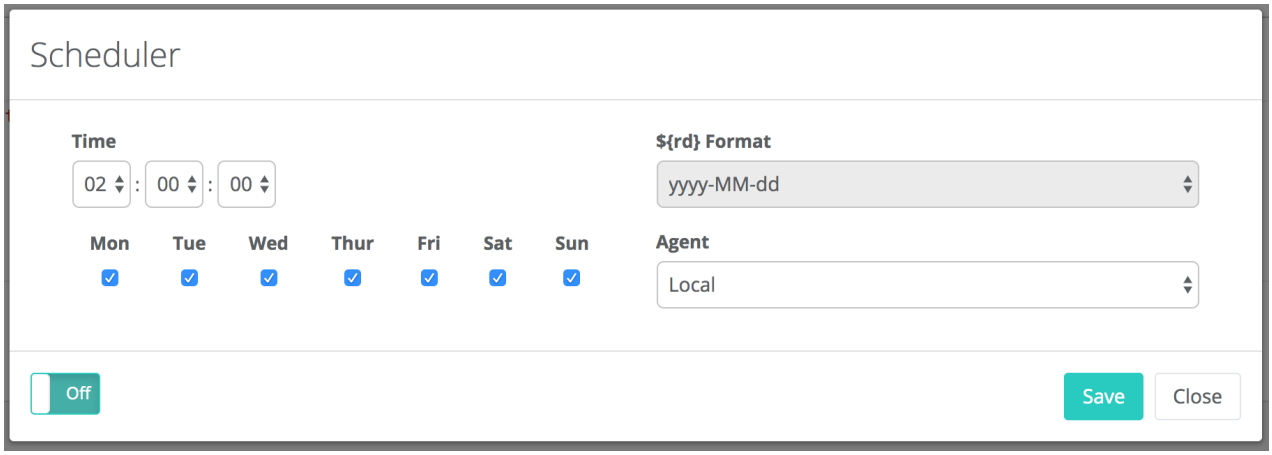
Template

You can also use -template to use DQ as a service hook and remove the need to pass in almost anything. In this case, DQ looks up the template automatically from either a previous run or if you've saved a template, and use these variables. Any variable at the cmdline will override and win/replace. This is a great way to remove connection and other information from being hard coded into the job control framework and allows edit ability from DQ Webapp.

```
%sh
~/owl/bin/owlcheck -usetemplate -ds lake.stock_nasdaq -rd $run_
date
```

Collibra DQ Scheduler - Built In

A quick option is to use DQ's built in scheduler. DQ automatically substitutes the runtime variables like $ into the job. This also gives you control to edit the DQCheck.

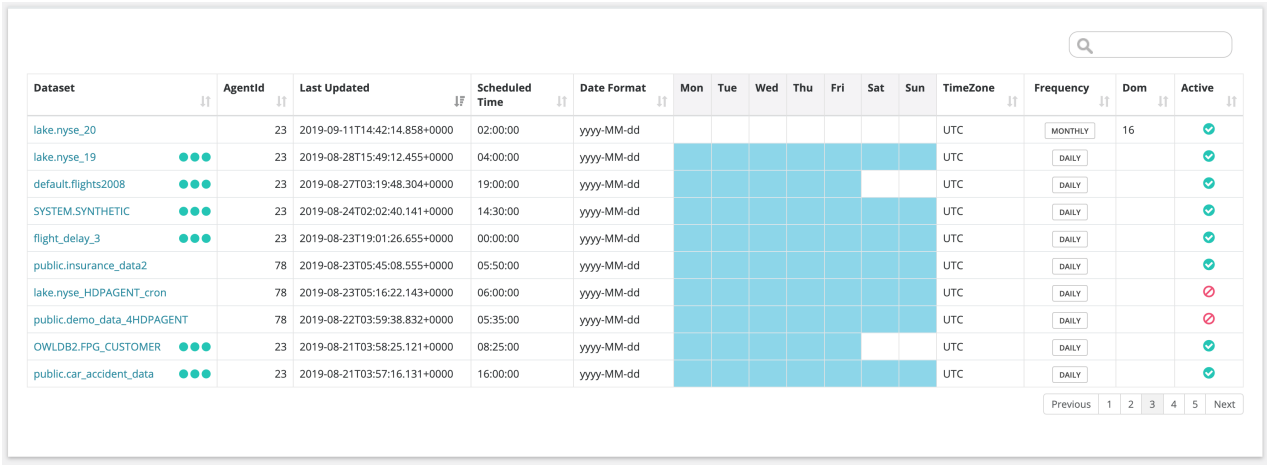


The schedule is based on the DQCheck Template. This way the runtime variables are replaced in each run. Notice the $ below.



All Scheduled Jobs in One Place

Under the jobs dashboard you can see an overview schedule with all running jobs and their status.



Dataset	AgentId	Last Updated	Scheduled Time	Date Format	Mon	Tue	Wed	Thu	Fri	Sat	Sun	TimeZone	Frequency	Dom	Active
lake.nyse_20	23	2019-09-11T14:42:14.858+0000	02:00:00	yyyy-MM-dd								UTC	MONTHLY	16	✓
lake.nyse_19	23	2019-08-28T15:49:12.455+0000	04:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
default.flights2008	23	2019-08-27T03:19:48.304+0000	19:00:00	yyyy-MM-dd	■	■	■	■	■			UTC	DAILY		✓
SYSTEM.SYNTHETIC	23	2019-08-24T02:02:40.141+0000	14:30:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
flight_delay_3	23	2019-08-23T19:01:26.655+0000	00:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
public.insurance_data2	78	2019-08-23T05:45:08.555+0000	05:50:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
lake.nyse_HDPAGENT_cron	78	2019-08-23T05:16:22.143+0000	06:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		⊘
public.demo_data_4HDPAGENT	78	2019-08-22T03:59:38.832+0000	05:35:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		⊘
OWLDB2.FPG_CUSTOMER	23	2019-08-21T03:58:25.121+0000	08:25:00	yyyy-MM-dd	■	■	■	■	■			UTC	DAILY		✓
public.car_accident_data	23	2019-08-21T03:57:16.131+0000	16:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓

Previous 1 2 3 4 5 Next

Working with Link IDs

Link ID is an out-of-the-box feature that lets you link the findings of a DQ Job back to the source record, or key, for remediation outside the application. Link IDs are required for archiving break records. Specify the link ID when creating a DQ job and selecting columns.

Importance of uniqueness

The link ID should be unique and should not contain any null values. It is most commonly the primary key. Depending on the dataset, you may need to select multiple columns to create a composite primary key. Otherwise, if the link ID is not unique, the DQ job may not be able to isolate the specific break records.

Link ID example using sales data

Suppose the dataset contains the following sales data:

NAME	TRDATE	SALES
John	2022-05-01	1000
John	2022-05-02	2000
John	2022-05-03	1000
John	2022-05-04	2000
John	2022-05-05	0
Steve	2022-05-01	300
Steve	2022-05-02	400
Steve	2022-05-03	200
Steve	2022-05-04	500
Steve	2022-05-05	10000

In this example, the rule specifies that (SALES=0) is a break record. Notice that this condition exists for "John" on "2022-05-05".

Suppose the link ID is set to only one column, NAME, which is not a unique value. In this case, the rule correctly identifies a break record, however, the DQ job cannot identify the row that constitutes the break record. It can only determine the NAME of the sales associate with the break record ("John").

If the link ID is set to both NAME and TRDATE, which together constitutes a unique key, the job can correctly identify the row that constitutes the break record.

Link ID example using NYSE data

In this example, a NYSE dataset with the columns CLOSE and SYMBOL selected as link ID columns. The following rule queries allow you to view rule break records:

- ```
  SELECT CLOSE, SYMBOL FROM ...     # both link ID columns are
  included in the SELECT statement
  ```
- ```
  SELECT * FROM ...     # SELECT * statement includes all
  columns, including link IDs
  ```

To view the rule break records, specify the link ID columns in your `SELECT` query. Alternatively, `SELECT * FROM` also allows you to view rule break records because it includes all dataset columns in your query.

Conversely, when CLOSE and SYMBOL are the only specified link ID columns in the same NYSE dataset, the following query will not display rule break records in the Rule Breaks modal on the Findings page:

- ```
  SELECT close FROM ...     # only one of the link ID columns
  included in the SELECT statement
  ```

Collibra DQ supports one or many primary key columns in your dataset for record linkage to your original table, file, or data frame. If your primary key column contains many columns, use a comma to delineate.
  ```

Link ID and the Metastore

No personal data is stored in the Metastore when using link ID (unless, of course, the column used as the link ID contains PII). The Metastore only stores:

- The rule applied to your DQ Job.
- The dataset used for your DQ Job.
- The column of reference.
- The link ID.

For further reading on sensitive information, see the Data Class and Sensitivity Labels documentation.

Set up link IDs

1. Create a DQ job as described in Create a DQ Job. Be sure to assign link IDs when selecting your columns.
2. Before running the job, open the settings to configure the desired break record archive behavior. Depending on the type of job, follow the steps provided based on the type of job:
    - Archiving break records from Pullup jobs
    - Archiving break records from Pushdown jobs

View break records

You can view rule break records using one of two methods:

- View break records from the Rules tab of the Findings page.
- Export a CSV using the Actions button on the Rules tab.

Rules with break records have associated link IDs that link back to the original dataset.

View break records from the Findings page

1. On the Findings page, open the Rules tab.
2. Click the **Actions** button and select **Rule Breaks**. The Rule Breaks dialog box opens.

> **Note** You can also use the Rule Discovery feature to apply sensitivity labels to data classes and trigger breaks for all the records that do not match your link ID. The Rule Discovery feature is described in more detail on the Rules tab of the Rule Exploring data quality findings page.

Export a CSV with link IDs

1. On the Findings page, open the Rules tab.
2. Click the **Actions** button and select **Rule Breaks**. The Rule Breaks dialog box opens.
3. Click **Download CSV**.

> **Note** You can also export a .xlsx file that lists the link IDs set up for your dataset from the Export tab on the Findings page.

Activity Usage

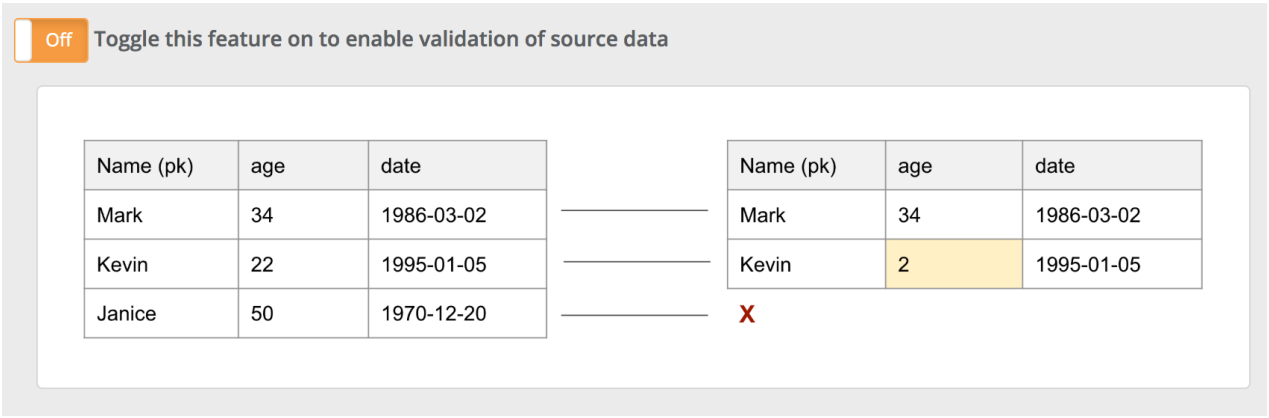
Activity	Supported	Description
SHAPE	Yes	One example of each shape issue will have a link back to the corrupt record for remediation.
OUTLIER	Yes	Each outlier will have a link back to the detected record for remediation. If you apply a limit you will only get the limited amount. Not available on categorical.
DUPE	No	Each duplicate or fuzzy match will have a link back to the record for remediation.

Activity	Supported	Description
SOURCE	Partial	Each source record that has a cell value that doesn't match to the target will have a link for remediation. SOURCE will not have links for row counts and schema as these are not record level findings.
RULE	Yes	Break records for Freeform and Simple rule types will be stored (any records that did meet the condition of the RULE will be provided with the linkID columns). These are stored as delimited strings in the rule_breaks table along with the dataset, run_id and rule name. Please note that, when using Freeform SQL, the linkID columns should be part of the select statement. LinkID columns should be unique identifiers. When incorporating secondary datasets of the same or different connection type into your rule, please ensure that your rule structure supports data preview visibility. In cases in which no preview records appear, consider refining the query to reference only the required columns (for example, the link ID from the primary dataset) instead of using broad `SELECT *` statements across both datasets.
BEHAVIOR	No	This class of data change is when a section of your data is drifting from its normal tendency there is no 1 record to link.
SCHEMA	No	This class of data change is at a schema/dataset level there are no records to link.
RECORD	Partial	In some cases when a record is added or removed it may be available for linking.
PATTERN	No	Patterns are not always a direct link.

DQ Job Validate Source

Reconciliation

Commonly data driven organizations have a need to ensure that two tables or a table and file match. This match might be a daily reconciliation or any snapshot in time. Collibra DQ calls this Source to Target or Left to Right matching. It covers row differences, schema differences and all cell values.



Impala/Hive -> DB2

Below is an example of comparing a table in DB2 to the same table in Impala.

```
./owlcheck \
-lib "/home/install/owl/drivers/db2" \
-cxn db2 \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE = '${rd}'
" \
-ds NYSE_STOCKS_VS \
-rd "2018-01-10" \
-vs \
-valsrckey SYMBOL \
-validatevalues \
-h $host/owltrunk \
-srcq "select * from nyse where TRADE_DATE = '${rd}' " \
-srccxn impala-jdbcuser \
-libsrc /home/isntall/owl/drivers/hivedrivers \
-jdbcprinc jdbcuser@CW.COM -jdbckeytab /tmp/jdbcuser.keytab \
-owluser admin \
-executorcores 4 -numexecutors 6 -executormemory 4g -driver-
memory 4g -master yarn -deploymode cluster \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM
```

DB2 -> Hive (Native)

Most databases only expose data through a JDBC connection but Hive offers a second path which does not require a JDBC connection. Hive has the ability to push down its processing to the local worker nodes and read directly from disk in the case when the processing is happening locally on a cluster. If your processing is not happening local to the cluster then you must use HiveJDBC. Take note of the -hive flag.

```
./owlcheck \
-hive \
-q "select * from nyse" \
-ds hiveNativeNyse \
-rd "2019-10-01" \
-vs \
-valsrckey exch,symbol,trade_date \
-validatevalues \
-srcq "select * from OWLDB2.NYSE_STOCKS" \
-srccxn db2 -libsrc /home/install/owl/drivers/db2 \
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
-sparkkeytab /home/install/owl/bin/user2.keytab -sparkprinc
user2@CW.COM
```

MySQL -> Oracle

This example compares the entire table instead of just a single day. Notice the 3 part valsrckey EXCH,SYMBOL,TRADE_DATE. Adding the date field ensures our key is unique and won't create a cartesian product. If the goal was to compare day over day with Oracle make sure to add TO_DATE('YYYY-MM-DD', '2019-10-01') to the where clause.

```
./owlcheck \
-lib /home/install/owl/drivers/mysql/ \
-cxn mysql \
-q "select * from lake.nyse" \
-ds lake.nyse \
-rd 2019-10-01 \
-vs \
-valsrckey EXCH,SYMBOL,TRADE_DATE \
-validatevalues \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM \
-srcq "select * from SYSTEM.NYSE" \
-srccxn oracle \
-libsrc /home/danielrice/owl/drivers/oracle/
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
```

File -> MySQL Table

Taking a file and loading it into a staging table or final table is a common part of every ETL process. However it is extremely common that the file values do not match or coherence into the table properly and these silent errors are usually not caught until a business user sees the data far long down stream.

```
./owlcheck \
-ds lake.nyse \
-rd 2019-10-01 \
-cxn "mysql" \
-q "select * from lake.nyse" \
-vs \
-valsrckey EXCH,SYMBOL,TRADE_DATE \
-validatevalues \
-srcfile "hdfs:///user/source/nyse.csv" \
-srcd "," \
-lib /home/install/owl/drivers/mysql/ \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM \
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
```

File -> File

DQ compares a File to a File. This is common in landing zones and staging areas where a file might be moved or changed and you need to know if anything changed or is incorrect.

```
./owlcheck \
-ds lake.nyse \
-rd 2019-10-01 \
-f "hdfs:///user/target/nyse.csv" \
-d "," \
-vs \
-valsrckey EXCH,SYMBOL,TRADE_DATE \
-validatevalues \
-srcfile "hdfs:///user/source/nyse.csv" \
-srcd "," \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM \
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
```

Add Date Column

Example

```
./owlcheck \
-ds "datataset_date_column" \
-rd "2019-07-01" \
-f "/Users/Downloads/csv2/2019010.csv" \
-adddc
```

> **Note** Add date column will use the run date supplied and add a date column named DQ_RUN_ID.

```
-adddc
```

This is used when you are using datasets that do not contain a date column or a malformed date string.

Known Limitations

This feature is only available for files, not for database tables.

Cloudera CLASSPATH

What is a CLASSPATH?

A CLASSPATH is essentially a list of jars that get injected into a JVM on the start of a job execution. Like many applications, Spark can have jars injected when a job is run. Cloudera has defined a list of predefined jars (rightfully called classpath.txt):

```
/etc/spark2/conf/classpath.txt
```

That will get injected whenever Spark is called. Here is an example list of jars as defined within a cluster we have stood up @ Collibra DQ:

```
[danielrice@cdh-edge ~]$ cat /etc/spark2/conf/classpath.txt
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/activation-1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/aopalliance-1.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/apacheds-
i18n-2.0.0-M15.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/apacheds-
kerberos-codec-2.0.0-M15.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/api-asn1-
api-1.0.0-M20.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/api-util-
1.0.0-M20.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/asm-3.2.-
jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/avro-
1.7.6-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/aws-java-
sdk-bundle-1.11.134.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/azure-
data-lake-store-sdk-2.2.9.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
beanutils-1.9.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
beanutils-core-1.8.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
codec-1.4.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
codec-1.9.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
configuration-1.6.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
daemon-1.0.13.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
digester-1.8.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
el-1.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
logging-1.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
math-2.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
math3-3.1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
net-3.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/core-
3.1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/curator-
client-2.7.1.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/curator-
framework-2.7.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/curator-
recipes-2.7.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/disruptor-3.3.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/findbugs-
annotations-1.3.9-1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/guava-
11.0.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/guava-
12.0.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/guice-
3.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
annotations-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
ant-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
archive-logs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
archives-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
auth-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
aws-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
azure-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
azure-datalake-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
common-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
datajoin-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
distcp-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
extras-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
gridmix-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
hdfs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
hdfs-nfs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-app-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-common-2.6.0-cdh5.16.1.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-core-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-hs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-hs-plugins-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-jobclient-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-nativetask-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-shuffle-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-examples-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
nfs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
openstack-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
rumen-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
sls-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
streaming-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-api-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-applications-distributedshell-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-applications-unmanaged-am-launcher-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-client-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-common-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-registry-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-applicationhistoryservice-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-common-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-nodemanager-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-resourcemanager-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-web-proxy-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hamcrest-
core-1.3.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
annotations-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-cli-
ent-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-com-
mon-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
examples-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
external-blockcache-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
hadoop-compat-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
hadoop2-compat-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-it-
1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-pre-
fix-tree-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-pro-
cedure-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-pro-
tocol-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
resource-bundle-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
rest-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
rsgroup-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
server-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
shell-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
thrift-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/high-
scale-lib-1.1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hsqldb-
1.8.0.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/htrace-
core-3.2.0-incubating.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/htrace-
core4-4.0.1-incubating.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/httpclient-4.2.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/httpcore-
4.2.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hue-plu-
gins-3.9.0-cdh5.16.1.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
annotations-2.2.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
core-2.2.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
core-asl-1.8.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
databind-2.2.3-cloudera.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
jaxrs-1.8.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
mapper-asl-1.8.10-cloudera.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
xc-1.8.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jamon-
runtime-2.4.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jasper-
compiler-5.5.23.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jasper-
runtime-5.5.23.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/java-xml-
builder-0.4.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/javax.inject-1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jaxb-api-
2.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jaxb-api-
2.2.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jaxb-
impl-2.2.3-1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jcodings-
1.0.8.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jets3t-
0.9.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jettison-
1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jettison-
1.3.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jline-
2.11.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/joni-
2.1.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jruby-
cloudera-1.0.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsch-
0.1.42.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsp-2.1-
6.1.14.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsp-api-
2.1-6.1.14.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsp-api-
2.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsr305-
3.0.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/leveldbjni-all-1.8.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/log4j-
1.2.16.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/log4j-
1.2.17.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/metrics-
core-2.2.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/metrics-
core-3.0.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/microsoft-windowsazure-storage-sdk-0.6.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/mockito-
all-1.8.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/netty-
3.10.5.Final.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/netty-
all-4.0.50.Final.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/okhttp-
2.4.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/okio-
1.4.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/paranamer-2.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/protobuf-
java-2.5.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/slf4j-
api-1.7.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/slf4j-
log4j12-1.7.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/snappy-
java-1.0.4.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/spark-
1.6.0-cdh5.16.1-yarn-shuffle.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/spymemcached-2.11.6.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/stax-api-
1.0-2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/xercesImpl-2.9.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/xml-apis-
1.3.04.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/xmlenc-
0.52.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/zookeeper-3.4.5-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/lib/hadoop/LICENSE.txt
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/lib/hadoop/NOTICE.txt
/usr/java/jdk1.8.0_131/lib/tools.jar
```

At each execution of any Spark job (including the use of spark-submit) this list of jars above will automatically get loaded.

What is a JAR?

A jar file is essential a compressed list of classes and methods. It is important to note that when jar files are built they will typically have an associated version number.

Someone can look at the contents of a jar file by executing:

```
jar -tvf phoenix-4.13.1-HBase-1.3-client.jar
```

Or you can wrap the above in a for loop that will look at the contents of every jar that might contain a method you are looking for.

```
for i in `ls -1 *.jar`;do jar -tvf $i | grep -i
htrace/trace;echo $i;done;
```

Common issues that can occur with CLASSPATH's

> **Note** **Caused by: java.lang.NoClassDefFoundError: org/apache/htrace/Trace** at org.apache.hadoop.hbase.zookeeper.RecoverableZooKeeper.exists (RecoverableZooKeeper.java:218) at org.apache.hadoop.hbase.zookeeper.ZKUtil.checkExists(ZKUtil.java:481) at org.apache.hadoop.hbase.zookeeper.ZKClusterId.readClusterIdZNode (ZKClusterId.java:65) at org.apache.hadoop.hbase.client.ZooKeeperRegistry.getClusterId (ZooKeeperRegistry.java:86) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.retrieveClusterId(ConnectionManager.java:850) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.<init>(ConnectionManager.java:635)

When a situation like this occurs it means that a method cannot be found in the classpath for the job that is trying to execute. This can indicate a couple things:

1. The job cannot find a jar file that contains the method flagged (in the example above the org/apache/htrace/Trace method).
2. Sometimes different versions of the same jar file gets loaded and the first jar loaded will always win. Older jars that get loaded first may not have a method defined in new jars.

At DQ, we have solved CLASSPATH / CLASSLOAD issues by automatically injecting jars defined in our ***owl/libs*** directory, and allowing users the ability to simply toggle loading them or not.

Column Matching

How much is your redundant data costing you?

Reclaim Gigabytes of Redundant Data

As data engineers, first we copy files into a landing zone, next we load the files into a staging area. After that we transform (ETL) the data into the final table. Soon that same data is copied to a lake for other groups to run analytics on. Eventually a group of analysts will need the data in another format and a data engineer will copy the data in a newly joined or transposed fashion. Sounds familiar?

The result is the same data or similar columns of the same data being copied many times. **The answer**: *Buy more hardware...* could be OR *run a Collibra DQ health report* and gain an understanding of how much data could be removed, reclaiming disk space and instantly seeing a return on investment after clicking the button.

Tabular breakdown of percentage of fingerprint matches

dataset_1	dataset_2	col_1	col_2	perMatch
silo.account	silo.user_account	networth	acc_branch	16
silo.account	silo.user_account	networth	networth	100
silo.account	silo.user_account	acc_name	acc_name	88
silo.account	silo.user_account	acc_name	acc_branch	0
silo.account	silo.user_account	acc_name	networth	0
silo.account	silo.user_account	acc_branch	acc_name	0
silo.account	silo.user_account	acc_branch	acc_branch	100
silo.account	silo.user_account	acc_branch	networth	14
silo.user_account	silo.user_account	acc_name	acc_branch	0
silo.user_account	silo.user_account	acc_name	networth	0
silo.user_account	silo.user_account	acc_branch	networth	14

Sometimes its not as simple as comparing two tables from the same database. Collibra DQ allows a technical user to setup multiple DB connections before executing health check.

```
import com.owl.common.Props
import com.owl.core.Owl

val c1 = new Connection()
c1.dataset = "silo.account"
c1.user = "user"
c1.password = "pass"
c1.query = "select id, networth, acc_name, acc_branch from
silo.account limit 200000"
c1.url = "jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306"

val c2 = new Connection()
c2.dataset = "silo.user_account"
c2.user = "user"
c2.password = "pass"
c2.query = "SELECT acc_name, acc_branch, networth FROM silo.ac-
count limit 200000"
c2.url = "jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306"

val props = new Props()
props.dataset = "colMatchTest1"
props.runId = "2017-02-04"
props.connectionList = List(c1,c2).asJava
props.colMatchBatchSize = 2
props.colMatchDurationMins = 3

val matchDF = new Owl(props).colMatchDF
matchDF.show

matchDF.createOrReplaceTempView("matches")
```

High level view of data overlap



Date Time Variable Options

Date Time Variables can enhance your query and file date templating and variable options. This allows easier scheduling and programmatic templating for common date variables.

> **Important** File names must use the `{yyyy}-{MM}-{dd}` format in file date templates.

Key	Function
${rd}	Replaces with -rd values in CMD. For example, 2023-01-20.
${rdEnd}	Replaces with -rdEnd values in CMD.
${yyyy}	Replaces with 4-digit year portion of -rd values in CMD. For example, 2023.
${yy}	Replaces with 2-digit year portion of -rd values in CMD. For example, 23.
${M}	Replaces with 1-digit month portion of -rd values in CMD.
${MM}	Replaces with 2-digit month portion of -rd values in CMD.
${MMM}	Replaces with 3 letter month name portion of -rd values in CMD For example, Jan, Jul, Dec.
${MMMM}	Replaces with long month name portion of -rd values in CMD For example, January, July, December.
${d}	Replaces with 1 digit day portion of -rd values in CMD.
${dd}	Replaces with 2 digit day portion of -rd values in CMD.
${HH}	Replaces with 2 digit hour portion of -rd values in CMD.
${KK}	Replaces with 1 digit hour portion of -rd values in CMD.
${mm}	Replaces with 2 digit minute portion of -rd values in CMD.
${ss}	Replaces with 2 digit second portion of -rd values in CMD.

Deploy Mode

Yarn and Cluster

Deploy Mode Client

```
--deploy-mode client
```

Deploy Mode Cluster

```
--deploy-mode cluster
```

Job Stuck in ACCEPTED State

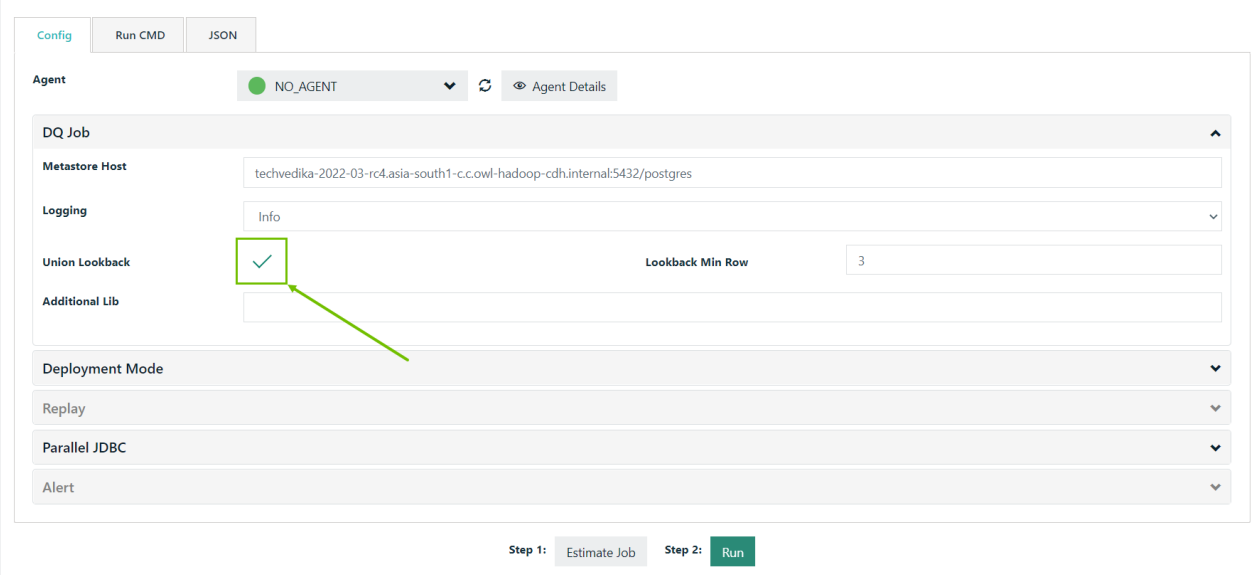
yarn.Client: Application report for application_1557720962505_0085 (state: ACCEPTED)
yarn.Client: Application report for application_1557720962505_0085 (state: ACCEPTED).

If running in cluster mode make sure you are passing in the below.

```
--deploy-mode cluster
--master yarn          # or spark master
-h 123.45.6.77:2181    # host to owl metastore
```

Lookback

Lookback lets you specify a historical period of past DQ scans to include in a DQ Job. For example, a lookback period of 5 includes DQ checks over the last 5 days of available data. By using the date column of the table, file, or view to scan the history of your dataset, lookback can help you identify potential outliers, patterns, and behavioral anomalies.



Important When a job that includes a lookback period runs, a temporary job appears in the Jobs queue. The temporary job will not resolve until all other lookback jobs are complete.

Lookback options

There are two lookback options to consider.

Option	Description
Union Lookback	Loads a specified number of historical runs of files on certain dates under the same dataset name and includes them in the DQ scans.
Full File Lookback	Includes the historical context of a single file in the outlier and/or patterns scans. To string together historical files that contain a single timeslice and include a larger historical window, use union lookback instead.

Union Lookback (-fllb)

Union Lookback, or File Lookback (-fllb) as it is also known, is used with deep learning and pattern matching. In the example below, it is used with deep learning.

File Lookback is used to check DQ Check history for previous files.

```
-fllb
```

This is often used with files and in conjunction with -adddc in cases where a date column is not in an ideal format or you do not have a date column on the given dataset.

Despite the name, this can be used with file or database storage formats.

> **Note** File look back (-fllb) should only be used when a SQL layer is not available. This is considered for advanced use cases, but may not be suitable for all file types and folder structures. Best practice is to expose a date signature somewhere in the file or directory naming convention.

Example

```
-ds "demo_lookback" \
-rd "2017-07-29" \
-lib "/opt/owl/drivers/mysql" \
-cxn "mysql" \
-q "select  * from lake.dateseries where DATE_COL = '2017-07-29'
" \
-dc DATE_COL \
-dl \
-dlkey sym \
-dllb 4 \
-fllb
```

> **Note** This look back will load your past 4 runs as your historical training set

Full File Lookback (-fullfile)

Like Union Lookback, Full File Lookback (-fullfile) is used with deep learning and pattern matching.

Fullfile Lookback uses the entire file for lookbacks instead of just filequery.

Understanding lookback command line flags

When you look at the DQ Job command line with lookback enabled, you may see several different lookback flags. The following table describes what each lookback flag means and to which DQ layers they apply.

Flag	Description	Layer
`-fllb`	Union lookback loads the DQ check history of files or database tables on certain dates under the same dataset name and includes them in the DQ Job.	Outliers and Patterns
`-fllbminrow`	The minimum number of rows in a dataset before it is included in a file lookback. This is automatically applied when Union Lookback is selected.	Outliers and Patterns
`-dllb`	Deep learning lookback loads DQ checks over a specified period of days and includes them in the scan for outliers.	Outliers
`-dlminhist`	An automatically generated flag based on the outlier lookback setting -dllb that defines the minimum number of days before DQ flags data as potential outliers. -dlminhistensures that the number of days in the algorithm is relative to the total scope of the lookback period.	Outliers
`-fullfile`	Includes the historical context of a single file in the outlier and/or patterns scans.	Outliers and Patterns
`-bhlb`	Behavior lookback loads a specified period of past DQ checks to include in a DQ Job. This controls the baseline profiling of a dataset.	Behavior
`-fpglb`	Pattern lookback loads DQ checks over a specified period of days and includes them in the scan for patterns.	Patterns

Filter & Filter Not

Filter & Filter Not is similar to a grep for limiting a dataframe to rows containing a substring.

> **Note** This feature should only be used when -q (query) and -fq (filequery) are not applicable. This feature is primarily used with raw files for limited filtering and not advanced conditional logic.

Example

```
./owlcheck \
-ds "dataset_name" \
-rd "2018-07-23 10" \
-d "," \
-f "/Users/Documents/file.csv" \
-filter "2018-07-23"
```

> **Note** -filter "2018-07-23"
> If file.csv contained multiple strings, but you only wanted rows containing "2018-07-23".

The inverse

> **Note** -filternot "2018-07-23"
> To exclude rows containing "2018-07-23".

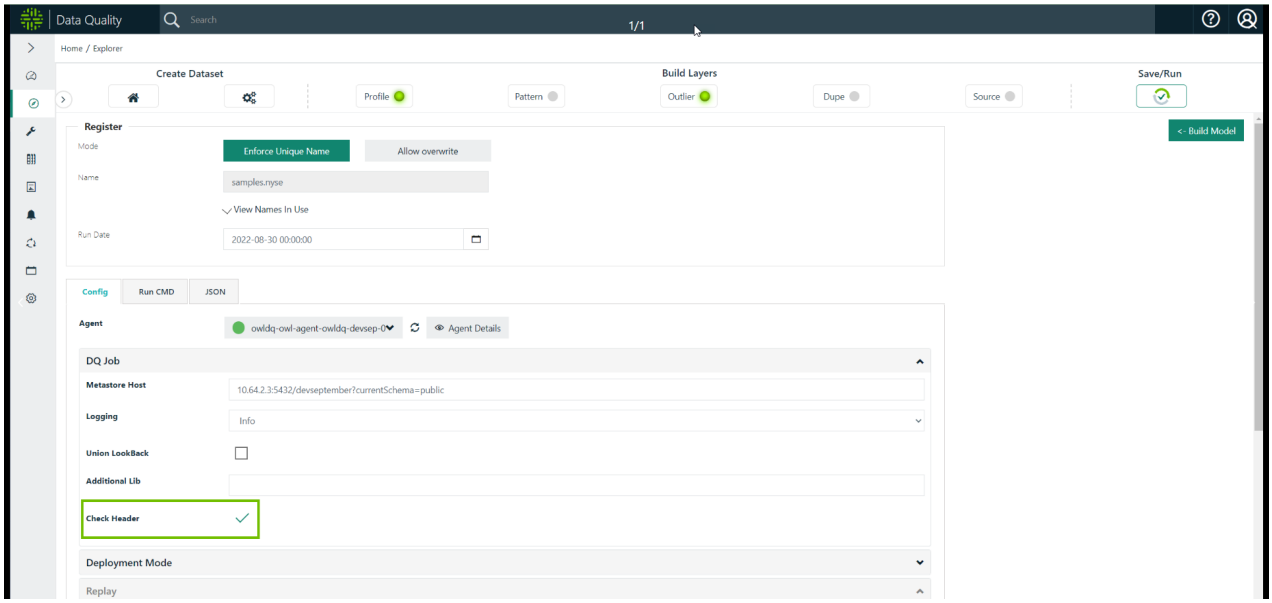
Header Check

Header Check lets you toggle column name detection on and off so column names containing special characters are not detected as schema changes. This is configurable with the **Check Header** checkbox or from the command line.

Configuring with the Check Header checkbox

The Check Header checkbox is checked by default. When it is checked, schema findings do not display when detected.

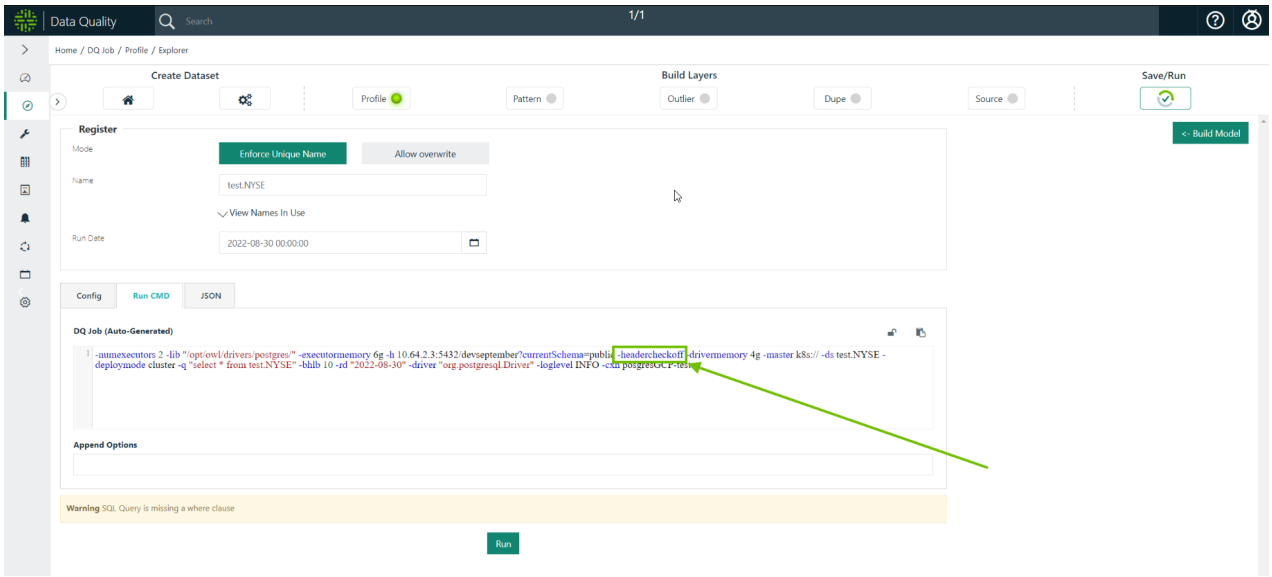
To disable the header check, uncheck the Check Header checkbox. This allows the schema findings to display when detected.



Configuring from the command line

Header checks are enabled by default and do not appear in the command line when enabled. To disable header checks from the command line, click the lock icon to unlock the command line and use the -headercheckoff variable, as shown in the screenshot below. When you are

done editing the command line, click the lock icon again to lock the command line; then, click **Run**.



Multiple Pattern Relationships

Run more than one relationship through pattern matching.

Example

```
./owlcheck \
-ds "fpg_multiple" \
-rd "2018-10-04" \

-cxn "postgres" \
-lib   "/opt/owl/drivers/postgres/" \
-q "select * from public.fpg_accounts where d_date = '2018-10-
04'" \

-fpgon \
-fpgdc "d_date" \
-fpglb "4" \
-fpgmulti "id,ssn_num=first_name,email|id,ssn_num=first_name,-
gender|id,ssn_num=last_name"
```

> **Note** Instead of -fpgkey and -fpgcol
> key1=cols1|keys2=cols2
> Enter multiple key=cols combinations separated by a pipe.

```
-fpgmulti "id,ssn_num=first_name,email|id,ssn_num=first_name,-
gender"
```

Nulls in Datasets

Example

```
./owlcheck \
-ds "datataset_date_column" \
-rd "2019-07-01" \
-f "/Users/Downloads/csv2/2019010.csv" \
-zfn \
-nulls "N.A."
```

Zero if null will replace null values with zero.

```
-zfn
```

To replace characters that represent a null to actual null values.

```
-nulls "N.A."
```

Spark-shell Sample

```
./bin/spark-shell --jars /opt/owl/bin/owl-core-trunk-jar-with-
dependencies.jar,/opt/owl/drivers/postgres/postgresql-42.2.5.jar --
deploy-mode client --master local[*]
```

Import lib's, if you get a dependency error, please import a second time.

```
import com.owl.core.util.{OwlUtils, Util}

import com.owl.common.domain2.OwlCheckQ

import com.owl.common.options._
```

Set up connection parameters to the database we want to scan if you don't already have a dataframe.

```
val`` ``url`` ``=
"jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/db?currentSchema=schema"
val`` ``connProps``=``Map("driver" -> "org.postgresql.Driver","user"
-> "user","password" -> "pwd","url" ->`` ``url,"dbtable" ->
"db.table")
```

Create a new OwlOptions object so we can assign properties.

```
val opt = new OwlOptions()
```

Set up variables for ease of re-use.

```
val dataset = "nyse_notebook_test_final"

val runId = "2017-12-18"

var date = runId

var query = s"""select * from <table> where <date_col> = '$date' """

val pgDatabase = "dev"val pgSchema = "public"
```

Set OwlOptions values to the metastore.

```
opt.dataset`` ``= datasetopt.runId`` ``= runIdopt.host`` ``=
"xxx.xxx.xxx.xxx"opt.pgUser`` ``= "xxxxx"opt.pgPassword`` ``=
"xxxxx"opt.port`` ``= s"5432/$pgDatabase?currentSchema=$pgSchema"
```

Create a connection, build the dataframe, register and run.

With inline processing you will already have a dataframe so you can skip down to setting the OwlContext.

```
val conn =`` ``connProps`` ``+ ("dbtable" -> s"($query)``
``$dataset")val df =`` ``spark.read.format("jdbc").options
(conn).load

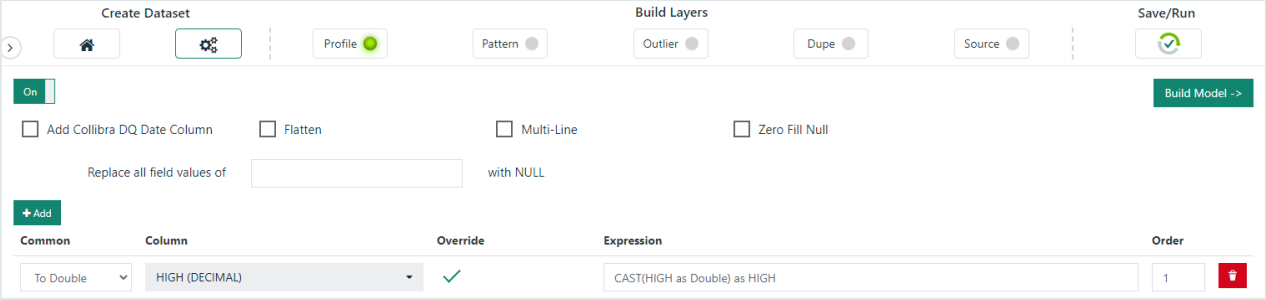
val owl = OwlUtils.OwlContext(df, opt)owl.register(opt)owl.owlCheck
```

Transform

Transform Date

When you create a dataset, you can transform columns such as Dates and Numbers to preferred formats. It is a common need to replace N.A. with nulls or empty white space.

> **Note** Because the Transform option defaults to a most common use case of `to_date`, you need to determine the proper transform function required for the data source in use.



Option	Description
Add Collibra DQ Date Column	Adds a synthetic run date column. **Tip** This can be helpful when your file does not contain dates that are easy to handle.
Flatten	Refers to the level of nesting in your file.
Multi-line	Refers to the content of the file being more than one line.
Zero Fill Null	Replaces NULL values with 0.
Replace all field values of [value] with NULL	Replaces all values of the column you specify with NULL.

Example

```
./owlcheck \
-ds  "dataset_transform" \
-rd  "2018-01-31" \
-f   "/Users/Documents/file.csv" \
-transform "purch_amt=cast(purch_amt as double)|return_amt=cast
(return_amt as double)"
```

> **Note** Submit an expression to transform a string to a particular type.
> In this example, transform the purch_amt column to a double.

```
-transform "purch_amt=cast(purch_amt as double)"
```

> **Note** Example of converting a string to a date.

```
-transform "RECEIVED_DATE=to_date(CAST(RECEIVED_DATE AS STRING),
'yyyyMMdd') as RECEIVED_DATE"
```

Converting an integer data type to a string

When you transform your target column is an integer data type, such as 20230414, you need to convert it to a string data type before you can convert it to a date.

1. From the Explorer page, select the **Transform** option.
2. In the **Expression** input field, enter the following expression:

   ```
   date_parse(cast(<your_column_name> as varchar), '%Y%m%d')
   ```

3. Click **Save**.

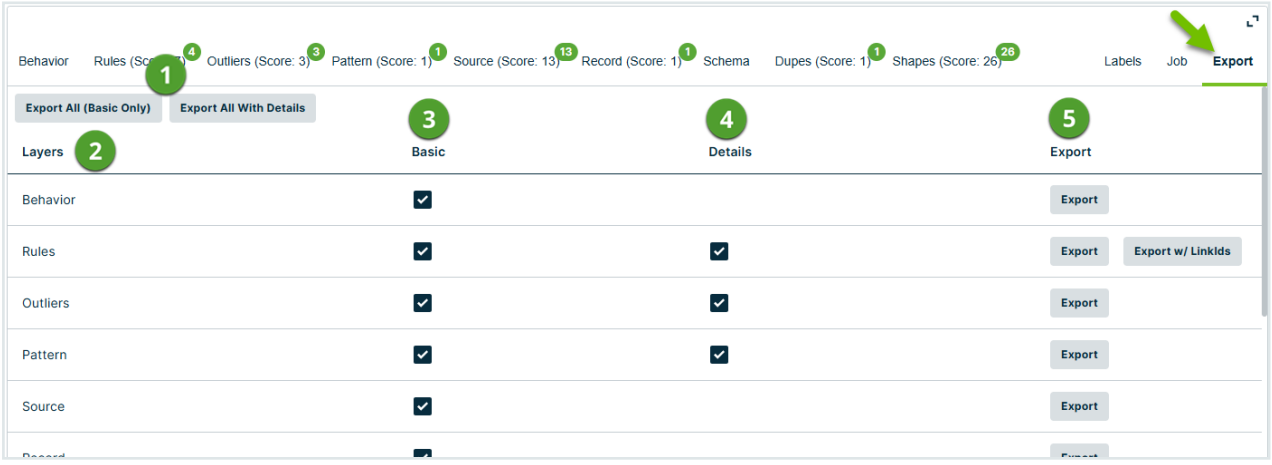
> **Note** If your query does not contain a ${rd} in the where clause, you need to wrap it with the DATE() function in the command line. For example, `select * from default.dataset where date_parse(CAST(example_date_column as varchar), '%Y%m%d') = DATE('${rd}')`

Known Limitations

- The UI does not run multi-line. A workaround is to supply these options manually in the command line.
- Explorer can flatten nested JSON array, but DQ Core can only flatten nested JSON with JSON object. Because of this limitation, you may encounter a scenario where the results in Explorer are flattened, but not in DQ Core.

Export tab

The Export tab on the Findings page enables you to export individual or all data quality findings as .xlsx files.



The following table provides and overview of the key elements of the Export tab with numbers that correspond to the image above.

No.	Component	Description
1	Export All buttons	**Export All (Basic Only)** - Export a .xlsx file that lists the basic information displayed in all the layers' tabs. **Export All With Details** - Export a .xlsx file that lists basic information as well as additional details (if applicable) of all layers. Details are only available for the Rules, Outliers, Pattern, and Shapes layers.
2	Layers	Each row represents a layer or dimension of the findings that can be exported.
3	Basic	Select **Basic** to export the same information displayed in the associated layer tab.
4	Details	Select **Details** to export the same information displayed in the associated layer tab plus additional details about the information included in the columns. This option is only available for the Rules, Outliers, Pattern, and Shapes layers.
5	Export	**Export** - For the given row, export a .xlsx file that lists the selected type of information, either Basic or Basic and Details. Details are only available for the Rules, Outliers, Pattern, and Shapes layers. **Export w/LinkIds** - For the given row, export a .xlsx file that lists the Link IDs set up on your dataset.

Working with Assignments

Assignments allow you to assign DQ Job findings (observations) to other users to review those findings and make determinations based on whether the findings are legitimate.

> **Important** The Assignments actions (Validate, Invalidate, Resolved) are disabled by default. To enable them, in Admin Limits, set **valsrcdisableaq** to **false**, and then re-run the job.

Sometimes a DQ Job run results in findings that include what seem to be anomalies in the data source. For example, when analyzing a job, you might notice a job finding with outliers that are

not what you were expecting. In this case, you could assign this job finding to a user to either **validate**, **invalidate**, or **resolve** the finding.

When an assigned user validates a job finding, it means that the results are expected and that we can confirm that it is, in fact, what the behavior model (outlier, dupe, shape, etc.) is reporting. The result is legitimate and, therefore, we want it to continue reporting as is.

If the user invalidates a finding, it means that the results are not expected and it is not what the behavior model is reporting it to be. In this case, the finding could be reporting an actual problem so that finding would need to be modified so that, when fixed, it will not be shown on the next job run.

> **Note** Collibra recommends assigning the job finding validation to a user who has access to the source data.

You can assign job findings to internal Data Quality & Observability Classic (CDQ) users or to external users with a third-party account, such as the ServiceNow tool. When you assign a user with a ServiceNow account, the assignments are sent as incidents, which can be tracked via the Service Now tool. You can select an individual or a group of users to manage the assignments for actions, such as escalations, de-escalations, and reassignments.

Action labeling options

The following action labels instruct Data Quality & Observability Classic on how to handle a finding:

Action	Description
Validate	Instructs Data Quality & Observability Classic to assign a finding to a specific user for review, which then appears in the Viewing the Assignment Queue. **Note:** Validating a finding does not improve your score. For more information, go to Validating Findings.

Action	Description
Invalidate	Instructs Data Quality & Observability Classic to ignore a finding and allow the value to pass. There are two invalidation options: **Save:** Marks a finding as invalidated but does not retrain the dataset. Use this option if you have a large number of findings that DQ has flagged, and you want to invalidate all of them at once, instead of clicking through one at a time. On your last finding, select Save and Retrain. All previously saved invalidated findings are removed and DQ retrains your dataset. **Save and Retrain:** Marks a finding as invalidated and retrains your dataset. This option restores any previously deducted points from a flagged finding and reflects it in your overall data quality score. If you do not have many findings to invalidate, you can click Save and Retrain individually. For more information, go to Invalidating Findings.
Resolve	Instructs Data Quality & Observability Classic to mark the finding as an observation. The finding is valid, but the data in the finding is not valid. The data issue needs to be resolved outside of DQ, and it should be evaluated in future runs. Resolving a finding does not immediately affect data quality scores. For more information, go to Resolving Findings.
Observation	The finding is an observation and cannot be assigned an action (Validate, Invalidate, Resolved). This occurs when both the **Matches** and **Passing** columns are true, or when the **vasrcdisableq** Admin Limit flag is set to **true**.

Available actions by feature

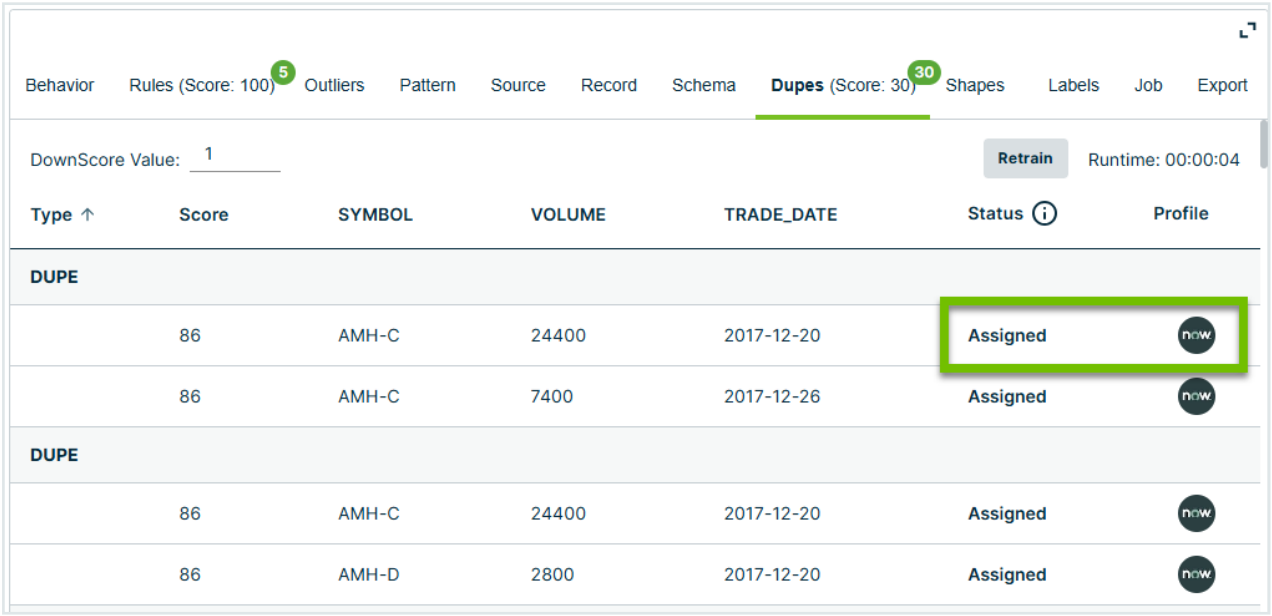
Feature	Available actions
Behaviors	Validate, Resolve
Rules	Validate, Resolve
Outliers	Validate, Invalidate, Resolve
Pattern	Validate, Invalidate, Resolve
Source	Validate, Invalidate, Resolve
Record	Validate, Resolve

Feature	Available actions
Dupes	Validate, Invalidate, Resolve

Warning Some findings are ineligible for all labeling options. For example, you can only apply Validate and Resolve labels to findings that result from Rules.

Distinguishing the type of user assigned

You can distinguish whether the user is a CDQ or ServiceNow assignment in the Profile column. Users are represented by an icon, rather than their names or usernames.



Modifying a previously labeled finding

To modify a previously labeled finding, you can always access them through the Labels tab. Here you can edit an annotation or delete a label entirely. If you delete a label, it returns to the findings page, unassigned. From there you can again choose to Validate, Invalidate, or Resolve it.

To closely analyze when a finding has received a label, who has applied it, and more, see also the Dataset Audit Trail.

Validating Findings

When you apply a validate label to a finding, you can assign it to another Data Quality & Observability Classic user to review. You can assign any job findings to a user who has previously logged into the Collibra DQ application (internal user) or one who has a matching ServiceNow account (external user). This marks the finding for future runs and sends it to the internal Assignments Queue.

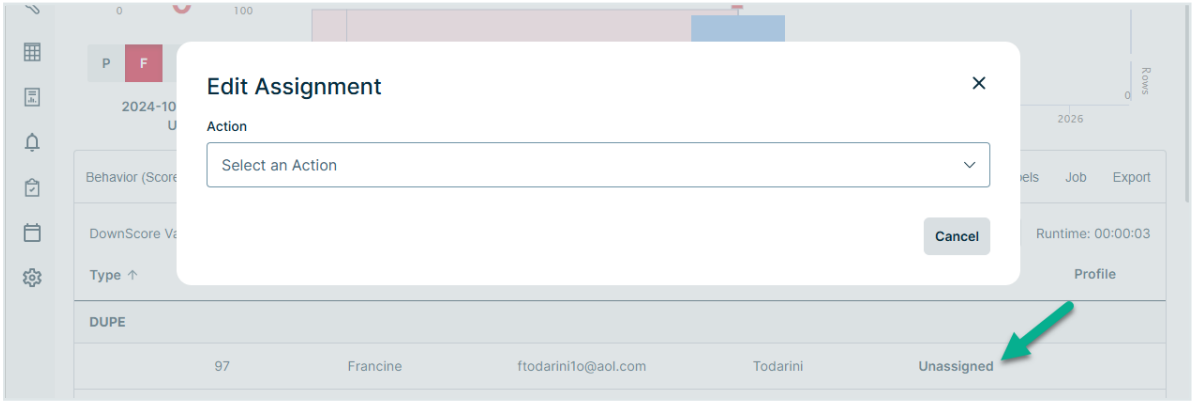
> **Important** The Assignments actions (Validate, Invalidate, Resolved) are disabled by default. To enable them, in Admin Limits, set **valsrcdisableaq** to **false**, and then re-run the job.

> **Note** The ServiceNow integration does not currently support SSO users.

> **Warning** To configure a queue, you must have ROLE_ADMIN or ROLE_CONNECTION_MANAGER.

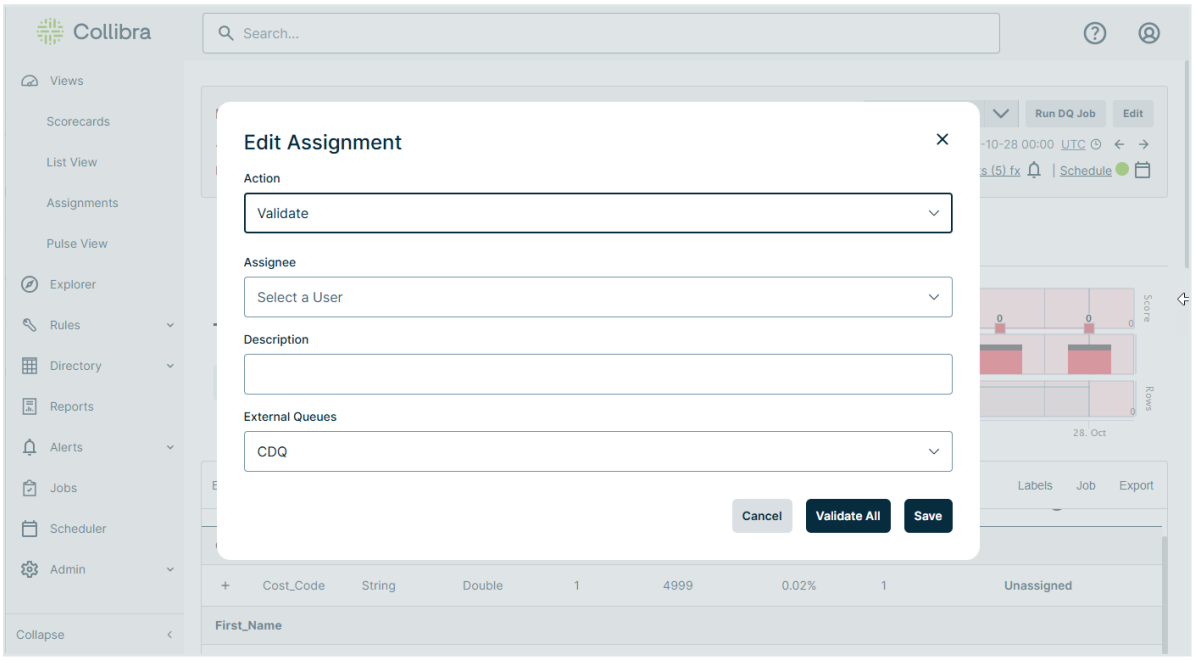
Steps

1. Select one of the findings of a previously run DQ job.
2. In the Status column, click **Unassigned**. The Edit Assignment dialog box appears.



> **Note** The options available to a feature (Validate, Invalidate, Resolve) vary depending on the feature. For more information on available actions by feature, see Working with DQ Assignments.

3. Click the **Action** dropdown and click **Validate**. The Edit Assignment dialog box updates to display additional options.



4. Enter the required information for the option you select.

Assign any job findings to a user who has previously logged into the Collibra DQ application (internal user).

Field	Description
Assignee	Collibra DQ user. Select the user from this dropdown.
Description	Enter a message for the user, such as "Please validate now".
External Queues	Select **CDQ** to assign the finding to a DQ user.

Assign any job findings to a user who has a matching ServiceNow account (external user).

Field	Description
Dataset	Dataset of the finding. This field is already populated.
Run Id	ID of the DQ job run. This field is already populated.
Assign To	Select one of the following options: **Individual** - Assign to one user **Group** - Assign to a group of users
Assignee	Collibra DQ user. Select the user from this dropdown. > **Note** If you do not select an assignee, the item is marked as valid but unassigned: **Acknowledged**.
Description	Enter a message for the user, such as "Please validate now", to provide details for the assignee. This field is optional.
External Queues	Select **ServiceNow** to assign this finding to an external user (with a ServiceNow account).

5. Select one of the following actions.

Action	Description
Validate All	Saves the validation settings for all the assignments listed on the current tab. > **Note** When you validate all findings without assigning them to a user, they are marked as "Acknowledged" on the Findings page and "Verified" on the Assignments Queue.

Action	Description
Save	Saves the validation settings for the selected assignment. **Note** When you validate a finding without assigning it to a user, it is marked as "Acknowledged" on the Findings page and "Verified" on the Assignments Queue.
Cancel	Closes the dialog box without saving.

Invalidating Findings

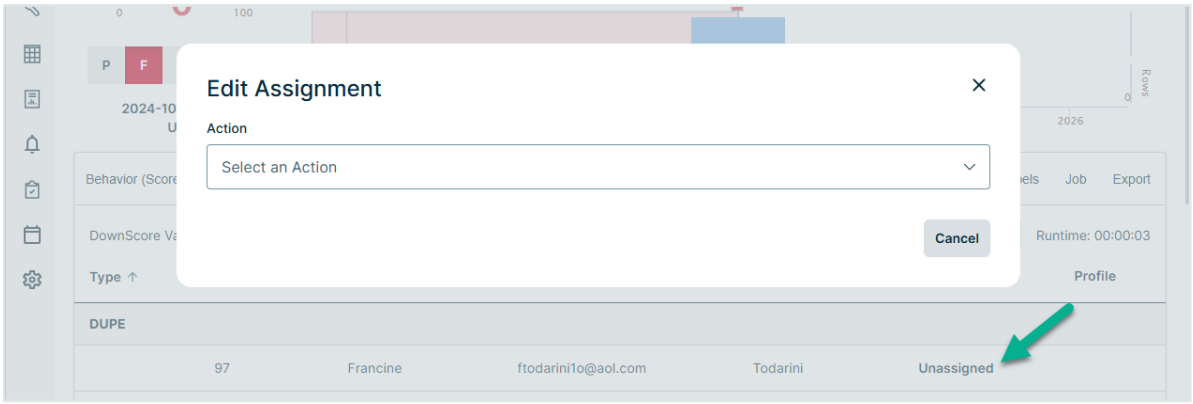
Sometimes the Findings page flags issues with your data that DQ discovers during a job run, but maybe you want DQ to ignore certain flagged issues. In those cases, assign the Invalidate label to indicate that the finding is not valid, and it should be ignored in this run and in future runs. After you add a descriptive annotation of your action, you can then select either **Save** or **Save and Retrain**. To display existing annotations, go to the Labels tab.

Important The Assignments actions (Validate, Invalidate, Resolved) are disabled by default. To enable them, in Admin Limits, set **valsrcdisableaq** to **false**, and then re-run the job.

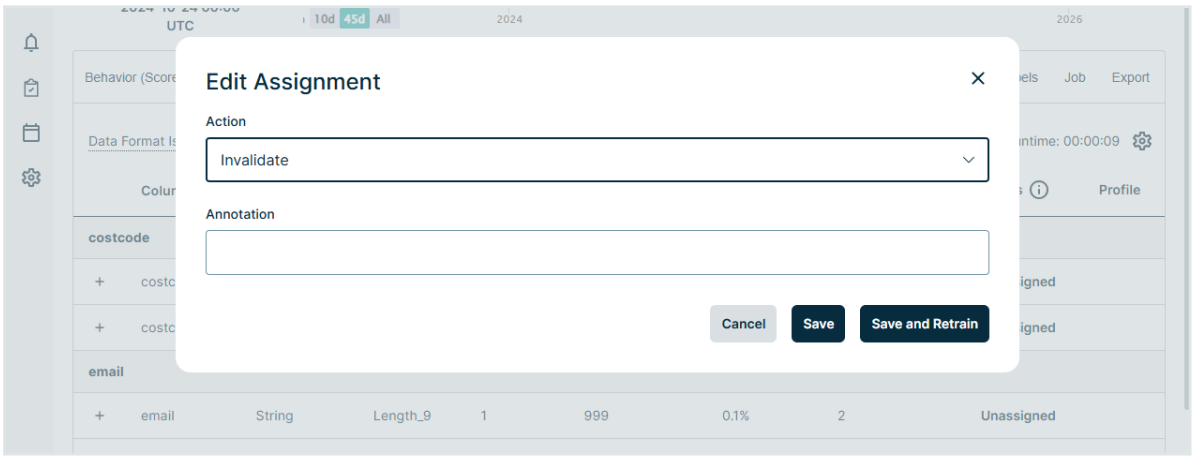
Note The options available to a feature (Validate, Invalidate, Resolve) vary depending on the feature. For more information on available actions by feature, see Working with DQ Assignments.

Steps

1. Select one of the findings of a previously run DQ job.
2. In the Status column, click **Unassigned**. The Edit Assignment dialog box appears.



3. Click the **Action** dropdown and click **Invalidate**. The Edit Assignment dialog box updates to display the Annotation box.



4. Enter a descriptive annotation in the Annotation field. To display existing annotations, go to the Labels tab.
5. Select one of the following actions.

Action	Description
Save	Marks a finding as invalidated but does not retrain the dataset. Use this option if you have a large number of findings that DQ has flagged, and you want to invalidate all of them at once, instead of clicking through one at a time. On your last finding, select Save and Retrain. All previously saved invalidated findings are removed and DQ retrains your dataset.

Action	Description
Save and Retrain	Marks a finding as invalidated and retrains your dataset. This option restores any previously deducted points from a flagged finding and reflects it in your overall data quality score. If you do not have many findings to invalidate, you can click Save and Retrain individually.
Cancel	Closes the Edit Assignment dialog box without saving.

Resolving Findings

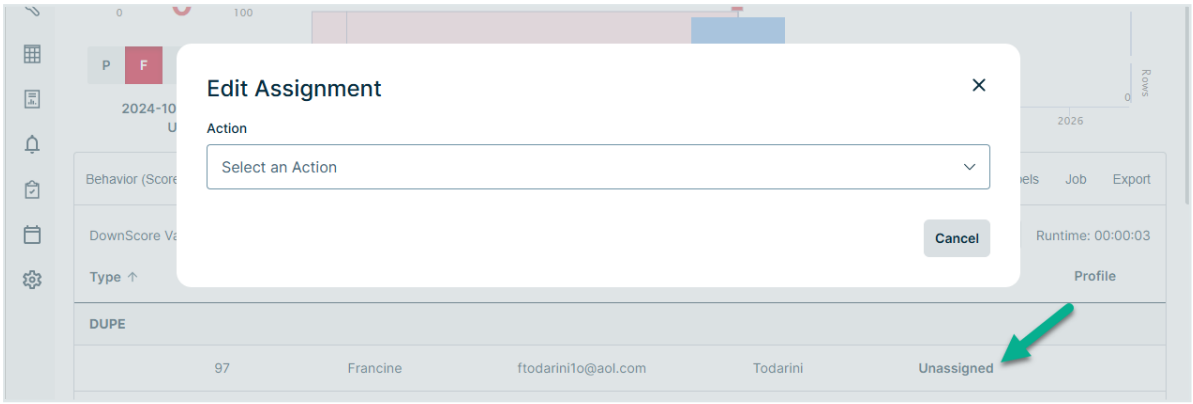
Resolve a finding to mark the finding as an observation. This occurs when the finding is valid, but the data in the finding is not valid. The data issue needs to be resolved outside of DQ, and it should be evaluated in future runs. Resolving a finding does not immediately affect your data quality score when applied.

> **Important** The Assignments actions (Validate, Invalidate, Resolved) are disabled by default. To enable them, in Admin Limits, set **valsrcdisableaq** to **false**, and then re-run the job.

> **Note** The options available to a feature (Validate, Invalidate, Resolve) vary depending on the feature. For more information on available actions by feature, see Working with DQ Assignments.

Steps

1. Select one of the findings of a previously run DQ job.
2. In the Status column, click **Unassigned**. The Edit Assignment dialog box appears.



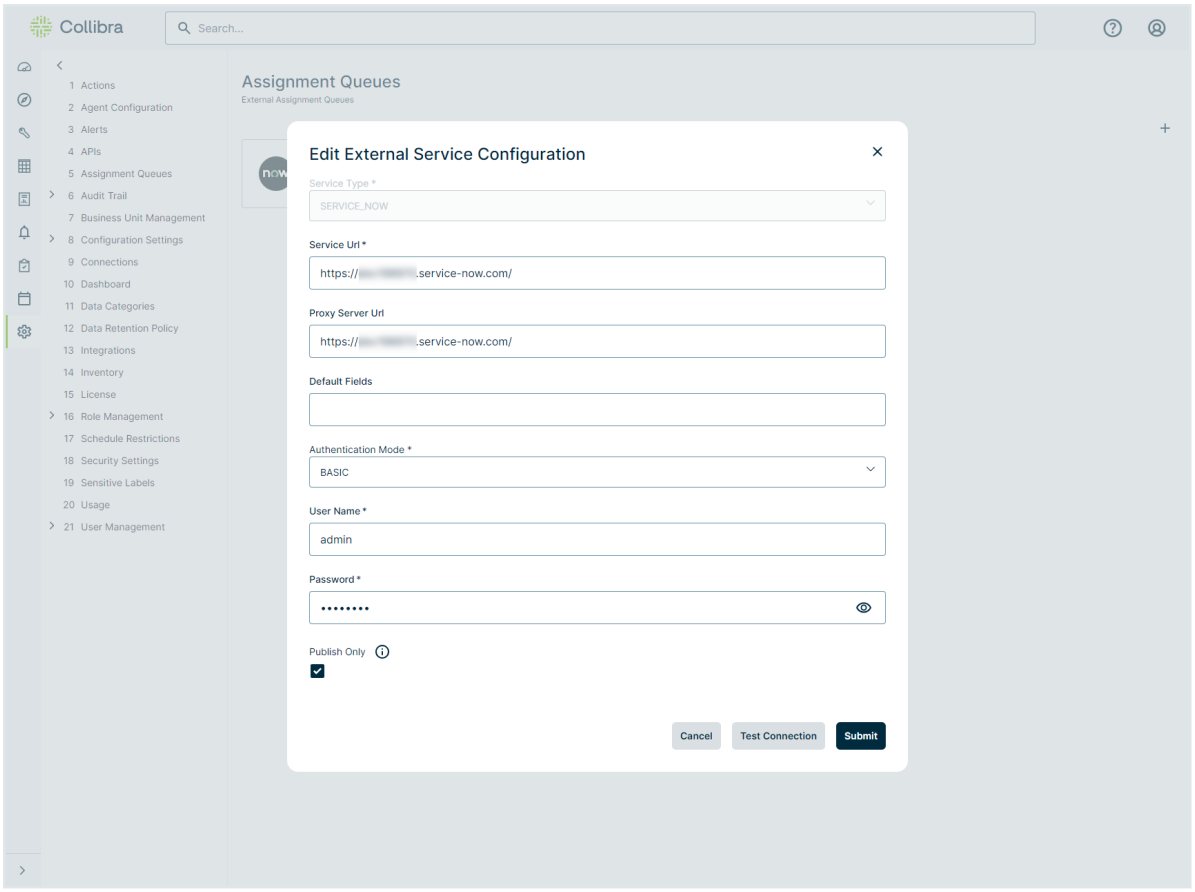
3. Click the **Action** dropdown and click **Resolve**.
4. Click **Save**.

Configure an External Assignment Connection

Add or edit an external service configuration connection in the Admin Console.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console** and then select **Assignment Queues** from the list.
3. Click to add a new external service configuration connection, or to edit an existing connection.



4. Complete the following fields in the Edit External Service Configuration window.

Field	Description
Service Type	Type of assignment queue service you are configuring. Default is ServiceNow.
Service URL	URL of the service you are configuring.
Proxy Server URL	URL of the proxy server. **Note** This is only applicable based on infrastructure configuration.

Field	Description
Default Fields	Semicolon separated key=value pairs on the Admin page, which override related field values on created assignment service records. **Note** There is a hierarchy whereby Additional Fields take precedence, if there is a matching competing key=value pair between both fields.
Authentication Mode	Type of authentication.
Username	Authentication credentials to the service.
Password	Authentication credentials to the service.
Publish Only	Creates a ServiceNow incident as an unlinked, separate object from the validated Collibra DQ finding. Because Collibra DQ findings and ServiceNow incidents are decoupled when Publish Only is selected, you can edit, validate, and resolve findings in Collibra DQ and incidents in ServiceNow. **Important** If Publish only is *not* selected when a ServiceNow assignment is configured, you must close the ServiceNow incident before resolving the finding in Collibra DQ.

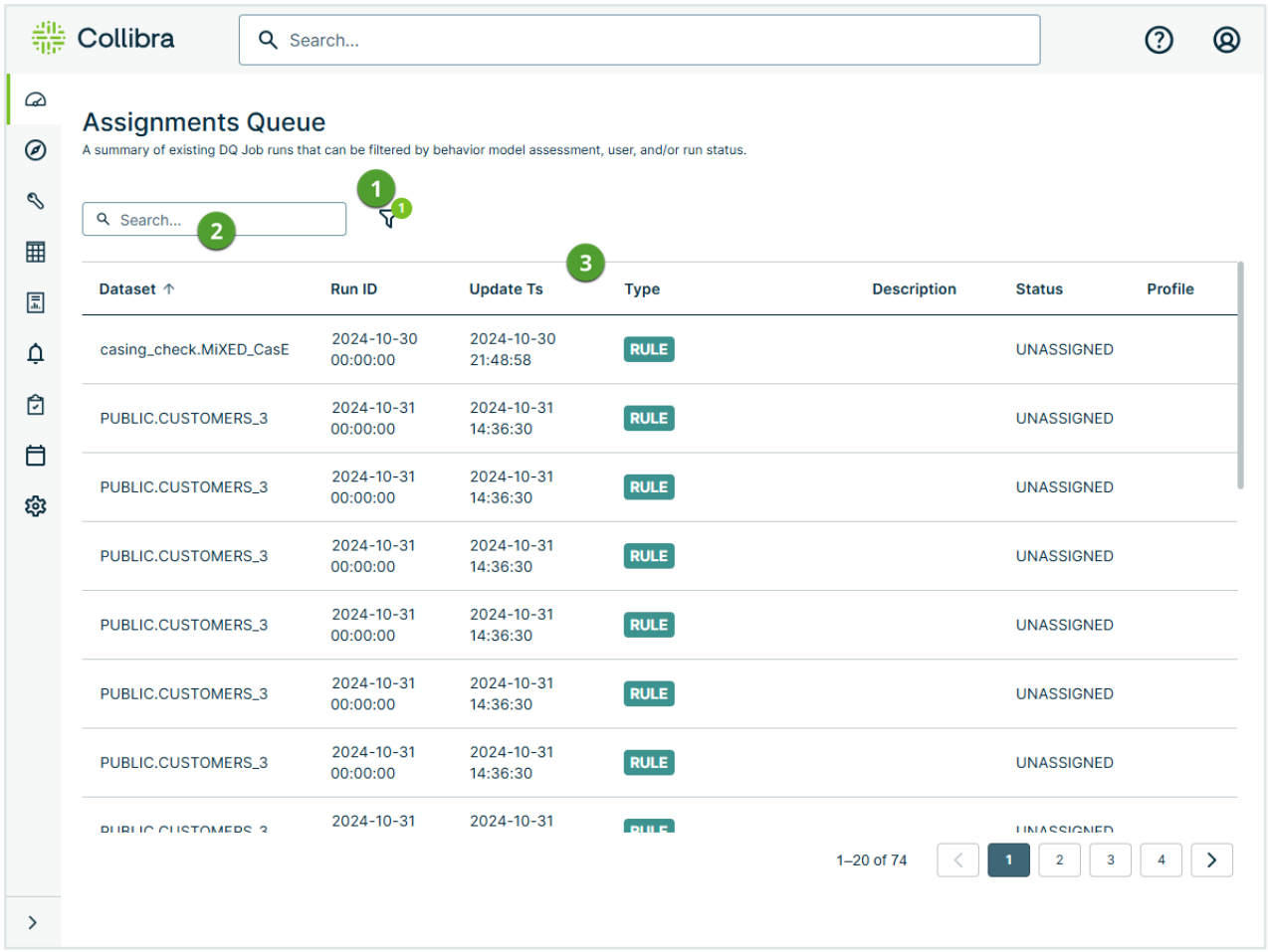
5. Click **Save**.

> **Note** You can select **Test Connection**, to ensure that the configuration for your connection is valid. You will receive a message indicating if the connection test was successful.

Viewing the Assignment Queue

The Assignments Queue is a summary of existing DQ Job runs that can be filtered by behavior model assessment (outliers, dupes, shapes, etc.), users, and run status.

View the Assignments Queue by clicking the icon or the **Workflow** option on the Data Quality & Observability Classic Home page.


Collibra
Search...
Assignments Queue
A summary of existing DQ Job runs that can be filtered by behavior model assessment, user, and/or run status.
Search...
1
2
3
Dataset
Run ID
Update Ts
Type
Description
Status
Profile
casing_check.MiXED_CasE
2024-10-30 00:00:00
2024-10-30 21:48:58
RULE
UNASSIGNED
PUBLIC.CUSTOMERS_3
2024-10-31 00:00:00
2024-10-31 14:36:30
RULE
UNASSIGNED
1–20 of 74
1
2
3
4

No.	Component	Description
1	Filter icon	Enables you to filter the Assignment Queue by selecting from the following criteria: **Type** - Filter by type of behavior model assessment by selecting one of the following options: • Outlier • Shape • Rule • Behavior • Dupe • Pattern • Record Changes • Schema Evolution • Mismatch Source **Status** - Filter by status by selecting one of the following options: • Unassigned - Assignments that have not been assigned to a user to validate. • Assigned - Assignments that have been assigned to a user to validate. • Verified - Assignments that have been verified. • Resolved - Assignments that have been resolved. **Date Range** - Filter by date range by selecting one of the following options: • All • 1 Day • 5 Days • 1 Month **User** - Filter by user by selecting a user from the dropdown.
2	Search field	Enables you to filter by entering your criteria in the Search field. .

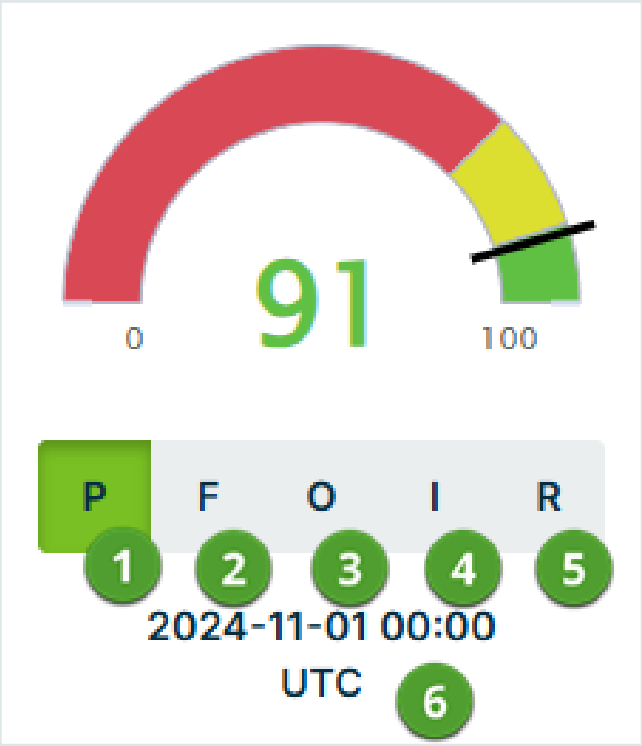
No.	Component	Description
3	Assignment Queue results	The Assignment Queue results include the following: **Dataset** - Link to a specific dataset. **RunID** - ID of the DQ job run. **Type** - Behavior model assessment of the Assignment. **Description** - Annotation or message added in the Validate Finding window. For more information, go to Manage Collibra DQ Assignments. **State** - State of the Assignment. **Profile** - User who was assigned the Job Finding.

Data Quality Score Meter

The data quality score is an aggregated value between 0-100 that represents a summary of the integrity of your data. A score of 100 means that Collibra DQ does not detect data quality issues in your dataset. Conversely, a score of 0 indicates that Collibra DQ has observed enough potential data quality issues that it has down-scored the findings to the lowest possible value.

> **Note** The findings from each data quality dimension where Collibra DQ detects potential data quality issues contribute to the overall data quality score.

Below the score meter are five clickable options plus a time stamp.



No.	Component	Description
1	Pass	Runs that failed but after investigation were valid. Train the model to pass future runs that are similar.
2	Fail	Runs that passed but after investigation were invalid. Train the model to fail future runs that are similar.
3	Off-Peak	A valid run that does not align with peak time runs, for example, over a weekend or a holiday. Label as Off Peak to train the model to recognize the pattern for future runs.
4	Ignore	A valid run that fits the run cycle profile but should not be used for training. You simply want to ignore.
5	Delete	Runs that triggered by mistake and should be removed.
6	Timestamp	The runId of the current DQ Job. When there is only one DQ Job run on a given runId date, the timestamp is always 00:00. However, when there are multiple DQ Job runs on a given runId date, the timestamp reflects the time each DQ Job ran, for example, 01:00 or 10:45.

Break records and Link IDs

Working with Link IDs

Link ID is an out-of-the-box feature that lets you link the findings of a DQ Job back to the source record, or key, for remediation outside the application. Link IDs are required for archiving break records. Specify the link ID when creating a DQ job and selecting columns.

Importance of uniqueness

The link ID should be unique and should not contain any null values. It is most commonly the primary key. Depending on the dataset, you may need to select multiple columns to create a composite primary key. Otherwise, if the link ID is not unique, the DQ job may not be able to isolate the specific break records.

Link ID example using sales data

Suppose the dataset contains the following sales data:

NAME	TRDATE	SALES
John	2022-05-01	1000
John	2022-05-02	2000
John	2022-05-03	1000
John	2022-05-04	2000
John	2022-05-05	0
Steve	2022-05-01	300
Steve	2022-05-02	400
Steve	2022-05-03	200

NAME	TRDATE	SALES
Steve	2022-05-04	500
Steve	2022-05-05	10000

In this example, the rule specifies that (SALES=0) is a break record. Notice that this condition exists for "John" on "2022-05-05".

Suppose the link ID is set to only one column, NAME, which is not a unique value. In this case, the rule correctly identifies a break record, however, the DQ job cannot identify the row that constitutes the break record. It can only determine the NAME of the sales associate with the break record ("John").

If the link ID is set to both NAME and TRDATE, which together constitutes a unique key, the job can correctly identify the row that constitutes the break record.

Link ID example using NYSE data

In this example, a NYSE dataset with the columns CLOSE and SYMBOL selected as link ID columns. The following rule queries allow you to view rule break records:

- ```
  SELECT CLOSE, SYMBOL FROM ...     # both link ID columns are
  included in the SELECT statement
  ```
- ```
  SELECT * FROM ...     # SELECT * statement includes all
  columns, including link IDs
  ```

To view the rule break records, specify the link ID columns in your `SELECT` query. Alternatively, `SELECT * FROM` also allows you to view rule break records because it includes all dataset columns in your query.

Conversely, when CLOSE and SYMBOL are the only specified link ID columns in the same NYSE dataset, the following query will not display rule break records in the Rule Breaks modal on the Findings page:

```
SELECT close FROM ...     # only one of the link ID columns
included in the SELECT statement
```

Collibra DQ supports one or many primary key columns in your dataset for record linkage to your original table, file, or data frame. If your primary key column contains many columns, use a comma to delineate.

Link ID and the Metastore

No personal data is stored in the Metastore when using link ID (unless, of course, the column used as the link ID contains PII). The Metastore only stores:

- The rule applied to your DQ Job.
- The dataset used for your DQ Job.
- The column of reference.
- The link ID.

For further reading on sensitive information, see the Data Class and Sensitivity Labels documentation.

Set up link IDs

1. Create a DQ job as described in Create a DQ Job. Be sure to assign link IDs when selecting your columns.
2. Before running the job, open the settings to configure the desired break record archive behavior. Depending on the type of job, follow the steps provided based on the type of job:
    - Archiving break records from Pullup jobs
    - Archiving break records from Pushdown jobs

View break records

You can view rule break records using one of two methods:

- View break records from the Rules tab of the Findings page.
- Export a CSV using the Actions button on the Rules tab.

Rules with break records have associated link IDs that link back to the original dataset.

View break records from the Findings page

1. On the Findings page, open the Rules tab.
2. Click the **Actions** button and select **Rule Breaks**. The Rule Breaks dialog box opens.

> **Note** You can also use the Rule Discovery feature to apply sensitivity labels to data classes and trigger breaks for all the records that do not match your link ID. The Rule Discovery feature is described in more detail on the Rules tab of the Rule Exploring data quality findings page.

Export a CSV with link IDs

1. On the Findings page, open the Rules tab.
2. Click the **Actions** button and select **Rule Breaks**. The Rule Breaks dialog box opens.
3. Click **Download CSV**.

> **Note** You can also export a .xlsx file that lists the link IDs set up for your dataset from the Export tab on the Findings page.

Activity Usage

Activity	Supported	Description
SHAPE	Yes	One example of each shape issue will have a link back to the corrupt record for remediation.
OUTLIER	Yes	Each outlier will have a link back to the detected record for remediation. If you apply a limit you will only get the limited amount. Not available on categorical.
DUPE	No	Each duplicate or fuzzy match will have a link back to the record for remediation.

Activity	Supported	Description
SOURCE	Partial	Each source record that has a cell value that doesn't match to the target will have a link for remediation. SOURCE will not have links for row counts and schema as these are not record level findings.
RULE	Yes	Break records for Freeform and Simple rule types will be stored (any records that did meet the condition of the RULE will be provided with the linkID columns). These are stored as delimited strings in the rule_breaks table along with the dataset, run_id and rule name. Please note that, when using Freeform SQL, the linkID columns should be part of the select statement. LinkID columns should be unique identifiers. When incorporating secondary datasets of the same or different connection type into your rule, please ensure that your rule structure supports data preview visibility. In cases in which no preview records appear, consider refining the query to reference only the required columns (for example, the link ID from the primary dataset) instead of using broad `SELECT *` statements across both datasets.
BEHAVIOR	No	This class of data change is when a section of your data is drifting from its normal tendency there is no 1 record to link.
SCHEMA	No	This class of data change is at a schema/dataset level there are no records to link.
RECORD	Partial	In some cases when a record is added or removed it may be available for linking.
PATTERN	No	Patterns are not always a direct link.

Archiving break records from Pullup jobs

The ability to archive break records allows you to automatically export CSV files containing the break records of DQ jobs on JDBC connections to external storage containers, such as Amazon S3 buckets. By offloading break records to cloud storage, you have more control over how you manage the results of your DQ jobs, and you can store data in the supported remote connection of your choice. Additionally, this relieves potential capacity issues of the PostgreSQL Metastore where, when jobs with link IDs back to their source return break records, the large amount of data included in the records can overload the PostgreSQL Metastore rule_breaks table and lead to database crashes.

You can send break records to the following remote file storage providers:

- Amazon S3
- ADLS
- Azure Blob
- Google Cloud Storage (GCS)

> **Important** When archive breaking records is turned on, rule break records no longer write to the PostgreSQL Metastore.

> **Note** When you leverage the Spark 3.2.2 standalone installation that comes with the AWS marketplace installation, be aware that the existing hadoop-aws-3.2.1.jar is incompatible with the AWS S3 Archive feature. To update the hadoop-aws-3.2.1.jar to hadoop-aws-3.3.1.jar in the spark/jars directory, obtain the file from the following link: Apache Downloads.
>
> Contact your CS or SE for assistance if you encounter any difficulties locating the jar file.

Break record details

Pullup jobs can currently export the break records of rules to your cloud storage service in a CSV file so you can manage it outside of Data Quality & Observability Classic.

This table shows an example of the rule record metadata included in the export file.

Dataset	Run Id	Rule Name	Link Id
ORACLE_DQUSER.NYSE_001	Tue Jan 16 00:00:00 GMT 2018	SQLG_VOLUME_001	ES~|2079300.0000
ORACLE_DQUSER.NYSE_001	Tue Jan 16 00:00:00 GMT 2018	SQLG_VOLUME_001	ESE~|252600.0000
ORACLE_DQUSER.NYSE_001	Tue Jan 16 00:00:00 GMT 2018	SQLG_VOLUME_001	ESL~|410200.0000

> **Note** When there are multiple link IDs in a break record export file, they will be parsed into multiple columns.

Prerequisites

The following table shows the available external storage options and the requirements for each.

Storage option	Prerequisites
Amazon S3	• An Amazon S3 connection. • Read and write access on your Amazon S3 bucket. Minimum required bucket permissions...[1]
ADLS	• An ADLS connection. • Read and write access on your ADLS bucket.

1

```
{
    "Version": "YYYY-MM-DD",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:ListStorageLensConfigurations",
                "s3:ListAccessPointsForObjectLambda",
                "s3:GetAccessPoint",
                "s3:PutAccountPublicAccessBlock",
                "s3:GetAccountPublicAccessBlock",
                "s3:ListAllMyBuckets",
                "s3:ListAccessPoints",
                "s3:PutAccessPointPublicAccessBlock",
                "s3:ListJobs",
                "s3:PutStorageLensConfiguration",
                "s3:ListMultiRegionAccessPoints",
                "s3:CreateJob"
            ],
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::YOURS3BUCKETNAME",
                "arn:aws:s3:::YOURS3BUCKETNAME/*"
            ]
        }
    ]
}
```

Storage option	Prerequisites
Azure Blob	• An Azure Blob connection. • Read and write access on your Azure Blob bucket.
Google Cloud Storage (GCS)	• A GCS connection. • Editor and Viewer access on your Cloud Storage bucket.

Steps

1. From Explorer, connect to a data source.
2. Select a scanning method.
3. In the Select Columns step, assign a **Link ID** to a column.
4. In the lower left corner, click **Settings**.
5. Under the Data Quality Job section, select the **Archive Breaking Records** option.
6. From the Archive Breaking Records dropdown menu, select the external storage option to which break records will send.
7. Click **Save**.
8. Set up and run your DQ job.
9. When a record breaks, its metadata exports automatically to your external storage service.

> **Important** Ensure that a column is assigned as the Link ID for external break record archival to work properly.

Archiving break records from Pushdown jobs

Data Quality & Observability Classic Pushdown uses SQL queries native to the data source in order to connect directly to the source system and submit the data quality workload. This gives Collibra DQ the unique opportunity to store break records that are typically written to the PostgreSQL Metastore in a database or schema specifically allocated for Collibra DQ. By archiving break records alongside the database or schema, you can:

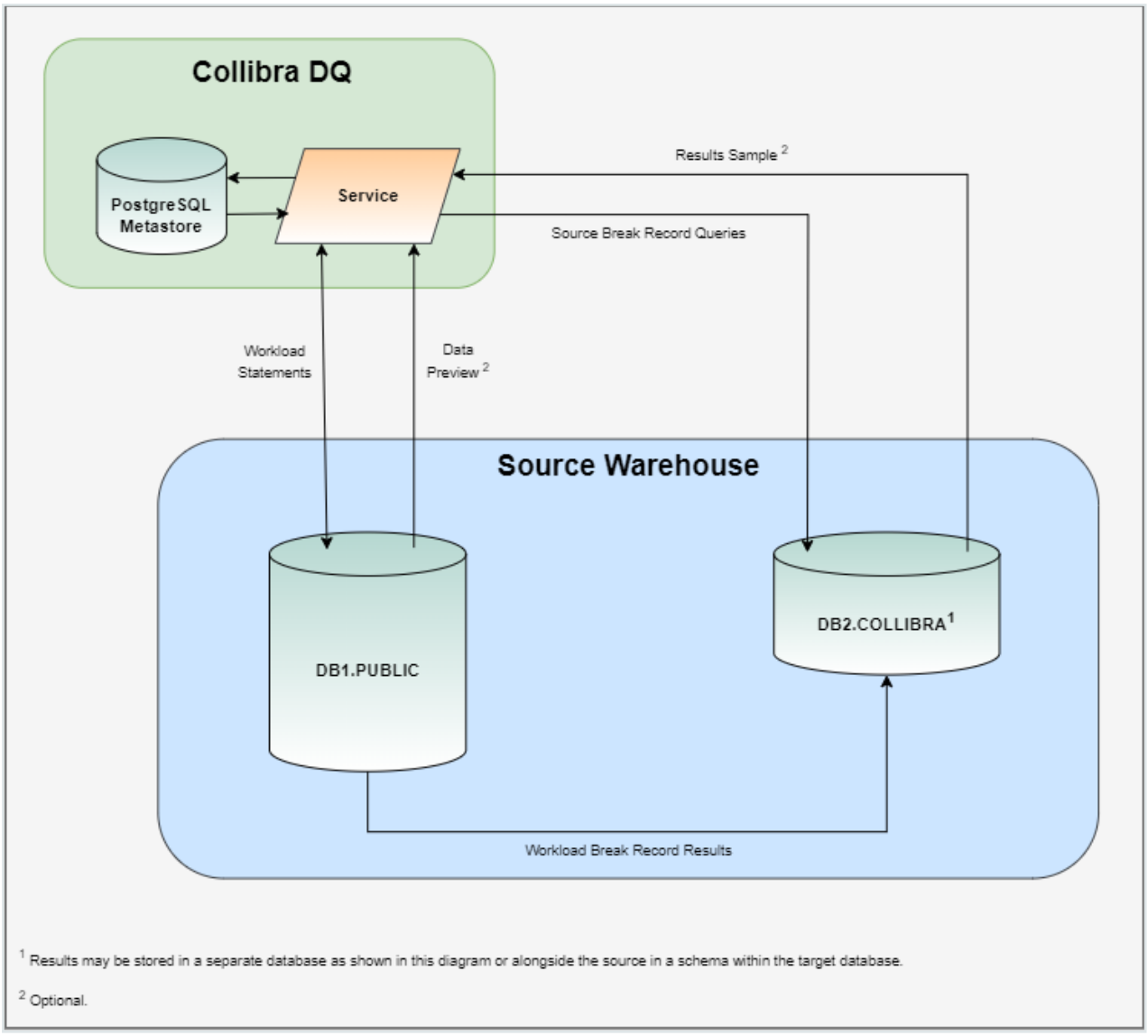
- Eliminate network traffic costs for better performance.
- Reduce storage limit restrictions.
- Improve security by generating break records that never leave the source system.

- Enhance your reporting abilities by removing the need to migrate break records from the PostgreSQL Metastore to your data source.
- Join datasets against your break records.

> **Note** When Archive Break Records is enabled, a limited number of break records are still stored in the PostgreSQL Metastore to power the DQ web application. To prevent break record storage in the Metastore, open the settings modal on the Explorer page of your dataset and turn on Data Preview from Source. This prevents records from leaving your data source and removes any others from the Metastore.

The Archive Break Records component uses a workload statement in the form of a data quality query to generate break records that get pushed into the source system for processing. Instead of returning a result set, the workload statement redirects its output to a table in your data source schema or database that is specifically allocated for Collibra DQ. The records written to the data source table contain source record linkage details that can be used to identify the source system record. These are called Link IDs and are required to write break records to the data source. Additionally, these records contain metadata generated by Collibra DQ that you may find useful when reviewing your break records.

When a workload is complete, a query is generated and stored in the data source, which allows you to easily access your break records. A results sample of break records may be optionally retrieved and used to generate a limited data preview of break records in the DQ Metastore.



Setting up break record archival

To automatically write break records to a database or schema in your data source, you first need to configure your connection's output location and enable Archive Break Records from the Settings modal on the Explorer page.

Compatible Pushdown connections

The following Pushdown connections support the archival of break records:

- Snowflake
- Databricks
- BigQuery
- Athena
- Redshift
- Trino
- SQL Server
- SAP HANA

Prerequisites

- You have a Pushdown connection.
- The service account user has write access to the database or schema to archive break records.

> **Important**
> Data sources may have different combinations of roles and permissions to allow the creation of break record tables and the ability to write records to them. Therefore, we recommend you review the connection requirements specific to your data source when you use Archive Break Records.

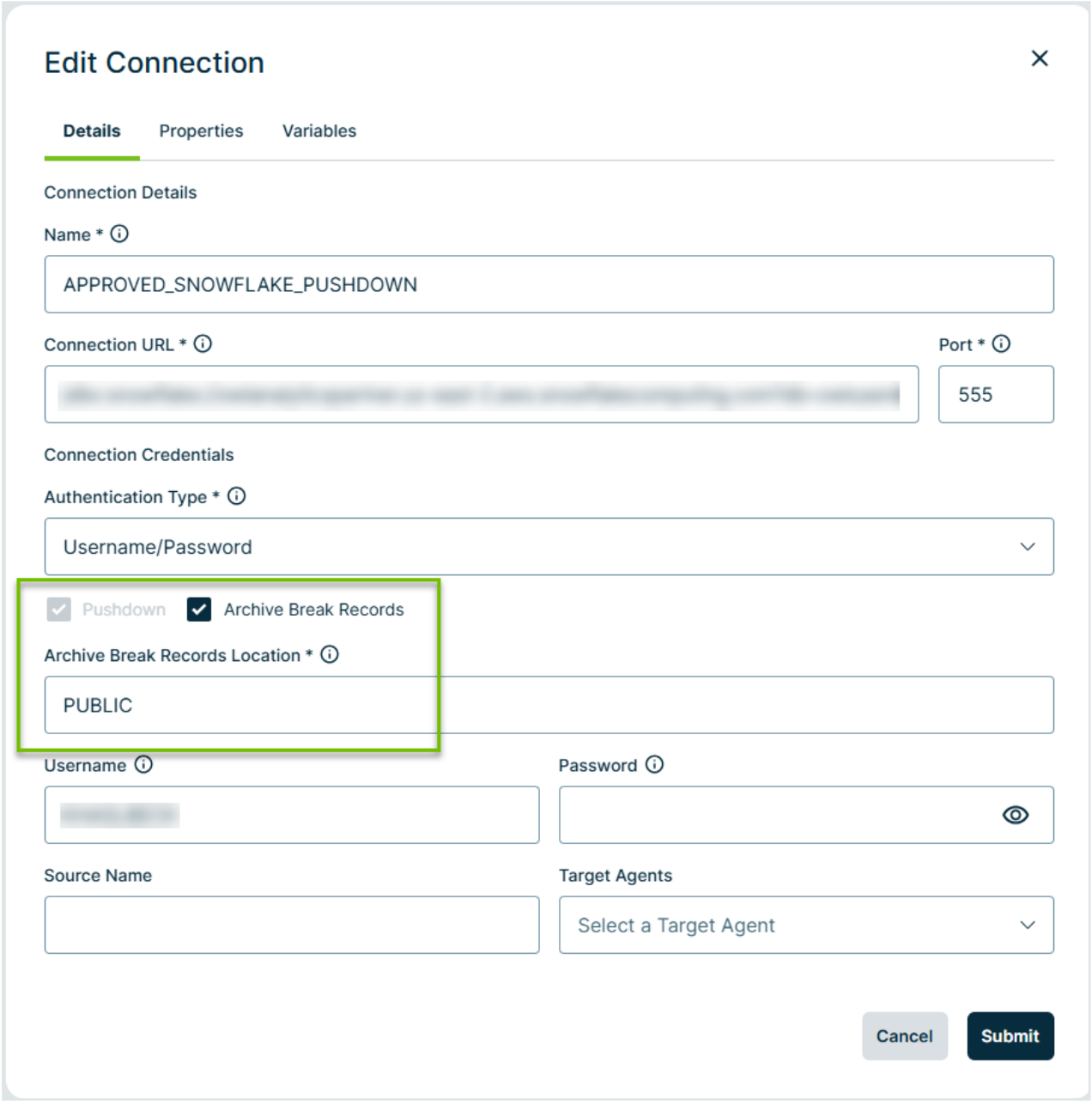
Specify the archive location for the connection

1. Open Connection Management and edit your connection.
2. In the Edit Connection dialog box, select **Archive Break Records**.
3. Specify the schema to archive break records in the **Archive Break Records Location** field. Keep in mind the following tips for selecting the schema:
    - If you enter an incorrect schema, Collibra DQ automatically uses the default output location, PUBLIC, instead.
    - Because it is optional to specify a database, if you only specify the schema, the database defaults to your current database.
    - You can enter database.schema when your schema resides outside of the current database. For example, db1.PUBLIC, db2.marketing, and db3.sales.
    - After you select the archive location, Collibra DQ creates the following tables in the database to store the break records:

- COLLIBRA_DQ_BREAKS
- COLLIBRA_DQ_DUPLICATES
- COLLIBRA_DQ_OUTLIERS
- COLLIBRA_DQ_RULES
- COLLIBRA_DQ_SHAPES

Example screenshot...[1]

1



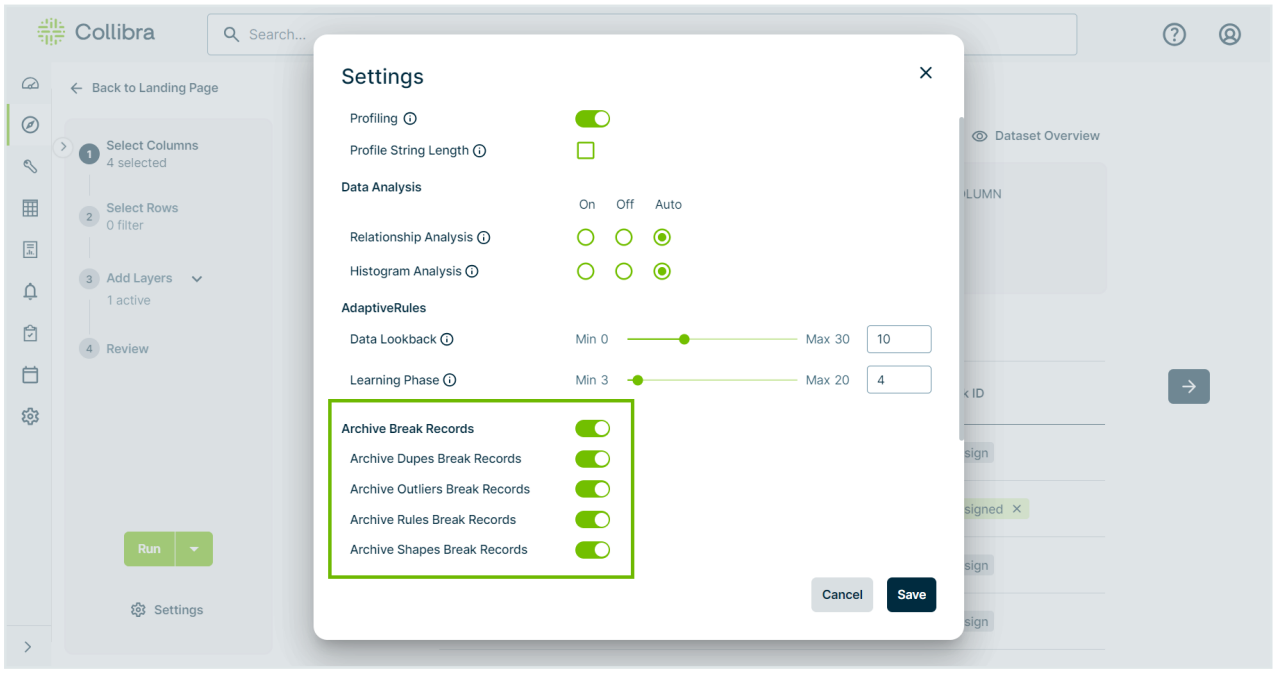
Create the job

1. Create a DQ job as described in Create a DQ Job.
2. In the Select Columns step, assign a **Link ID** to a column.

   > **Important** To get break records, it's important to specify a Link ID. The column you assign as Link ID should not contain NULL values and its values should be unique, most commonly the primary key. Composite primary key is also supported.

3. In the lower left corner, click **Settings**.
4. Select the **Archive Break Records** option.
   » By default, turning on Archive Break Records enables the export of break records for all available finding types.
   Example screenshot...[1]
5. Click **Save**.
6. Set up and run your DQ job.
7. When a record breaks, its metadata exports automatically to the data source.

1



Viewing break record details

You can archive break records from the following finding types of Pushdown jobs:

Finding type	Database table
Rules	COLLIBRA_DQ_RULES
Outliers	COLLIBRA_DQ_OUTLIERS
Shapes	COLLIBRA_DQ_SHAPES
Duplicates	COLLIBRA_DQ_DUPLICATES

The records for each of these finding types contain a unique set of metadata associated with the particular break type. When Archive Break Records is turned on, Collibra DQ sends this metadata to the break record table in the data source.

Rules

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
job_uuid	uuid	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset.
rule_name	varchar	NOT NULL	The name of the rule.
exception	varchar	NULL	The exception message of a rule that is neither Passing nor Breaking.

Column name	Type	Default	Description
link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.

Outliers

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
job_uuid	uuid	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset.
id	int	NOT NULL	The job ID associated with the break record.
key_column	varchar	NULL	When a key column is assigned, this is the name of that column.
key_value	varchar	NULL	When a key column is assigned, this is the value in that column.

Column name	Type	Default	Description
value_column	varchar	NOT NULL	The name of the column that contains the outlier.
value	varchar	NOT NULL	The value of the outlier finding.
date_value	date	NULL	When a date column is assigned, this is the date column value.
type	varchar	NOT NULL	Indicates whether the outlier finding is categorical or numerical.
median	varchar	NULL	The median of the value of a numerical outlier.
lb	double	NULL	The lower bound of the values of a numerical outlier.
ub	double	NULL	The upper bound of the values of a numerical outlier.
confidence	int	NOT NULL	The confidence score of the outlier finding.
is_outlier	boolean	NULL	Shows whether a finding is an outlier.
is_historical	boolean	NULL	Shows whether a finding is a historical outlier.
is_topn	boolean	NULL	Shows whether a finding is a topN value.
is_botn	boolean	NULL	Shows whether a finding is a bottomN value.

Column name	Type	Default	Description
source_date	varchar	NULL	The date when an outlier occurred.
frequency	long	NULL	The number of occurrences of an outlier.
percentile	double	NULL	The threshold that helps identify extreme outlier values.
link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.

Shapes

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
job_uuid	uuid	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset.

Column name	Type	Default	Description
column_name	varchar	NOT NULL	The name of the column with shape findings.
shape_type	varchar	NOT NULL	The data type of the shape finding. For example, *string*.
shape_value	varchar	NOT NULL	The column value of the shape finding.
shape_count	int	NOT NULL	The number of columns that conform to the shape that Collibra DQ discovered.
shape_rate	double	NOT NULL	The percentage a shape conforms to the format that Collibra DQ identifies as normal.
shape_length	int	NOT NULL	The format of the shape. For example, "xxx & xxxx" could represent the street "3rd & Main"
link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.

Duplicates

Column name	Type	Default	Description
key	long	auto increment	The automatically generated number that identifies your break record.
updts	timestamp	default	The timestamp of when the job ran.
job_uuid	varchar	NOT NULL	The automatically generated number that uniquely identifies your job. For example, 19745.
dataset	varchar	NOT NULL	The name of the dataset that contains a possible duplicate.
run_id	timestamp	NOT NULL	The timestamp of when the job ran.
key_id	integer	NOT NULL	A unique identifier for each duplicate record that is unique even for similar duplicates found in other runs. Each time a job run, new key values are created for each duplicate record.
dupe1	varchar	NOT NULL	The name of the column that Collibra DQ identifies as a duplicate value of dupe2.

Column name	Type	Default	Description
dupe1_link_id	varchar	NOT NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.
dupe2	varchar	NULL	The name of the column that Collibra DQ identifies as a duplicate value of dupe1.
dupe2_link_id	varchar	NULL	The link ID value. For example, if from an NYSE dataset you assign one column called "Symbol" and another column called "Trade Date" as Link IDs, then you might see a value in a format such as *DST~|2018-01-18*.
score	integer	NOT NULL	The duplicate score of the column where Collibra DQ detects a potential duplicate.

Unlike in Pullup mode, when Collibra DQ detects a duplicate value in a Pushdown job with Link IDs assigned, all columns included in the source query of the row where the break is detected display on the Findings page. In Pullup mode, only Link ID columns display on the Findings page when a duplicate value is detected, because the data_preview Metastore table only stores Link ID columns.

Limitations

- Rule break record archival for complex rules, such as those with secondary datasets, is not supported. Rules that reference secondary datasets cannot share column names, because Collibra DQ cannot programmatically identify the table from which these columns come.
    - A workaround is to avoid tables with these shared column names or alias on the columns.

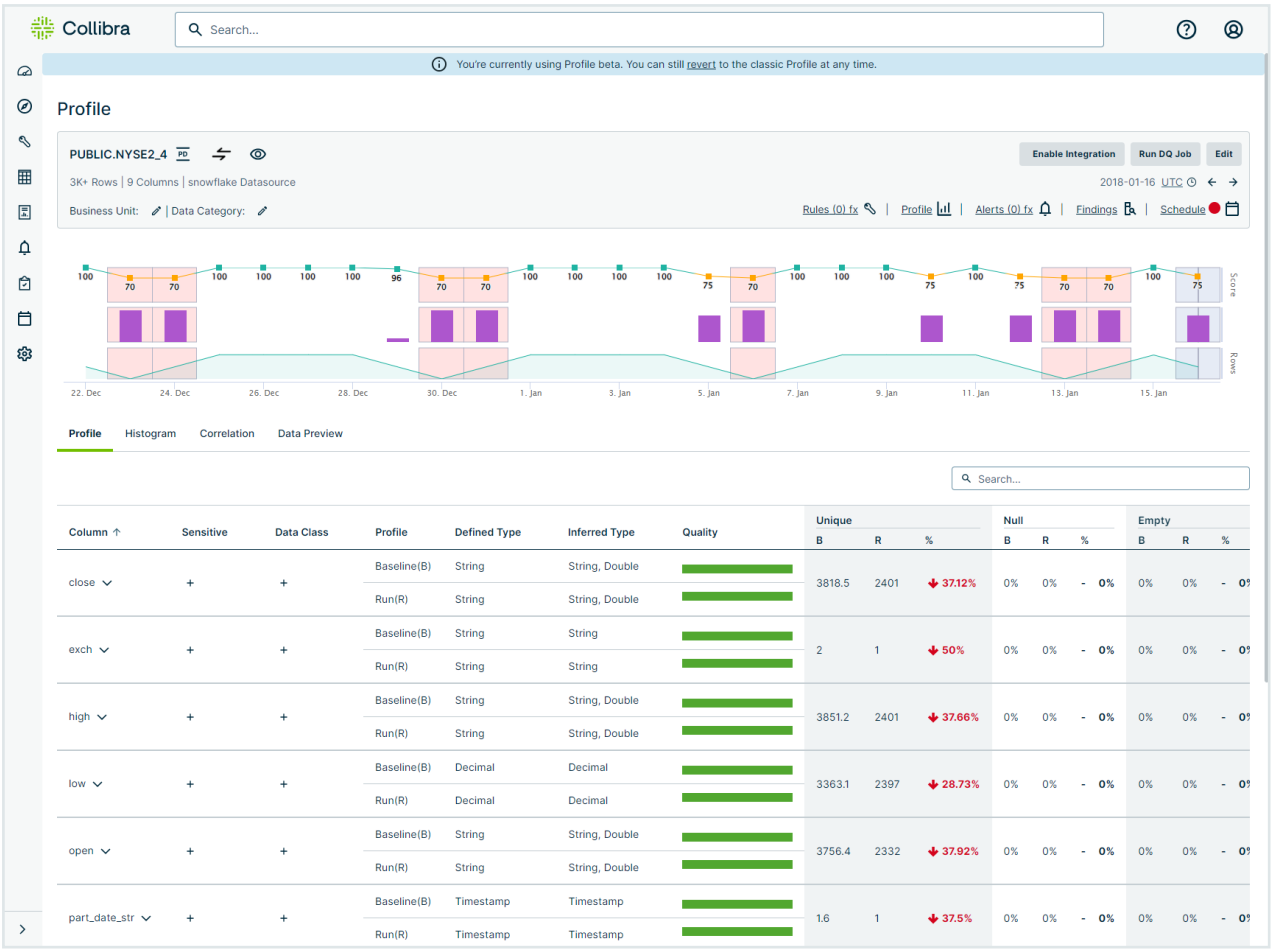
Profile

Profile provides a comprehensive analysis of the behavior of your dataset over time. When you run a DQ job, the results of the scan become available for analysis on the Findings and Profile pages. With rich options like the ability to drill down into column-level statistics, create quick rules, and evaluate charts and graphs that show common and emergent trends in your data, you can obtain a granular understanding of the contours of your dataset.

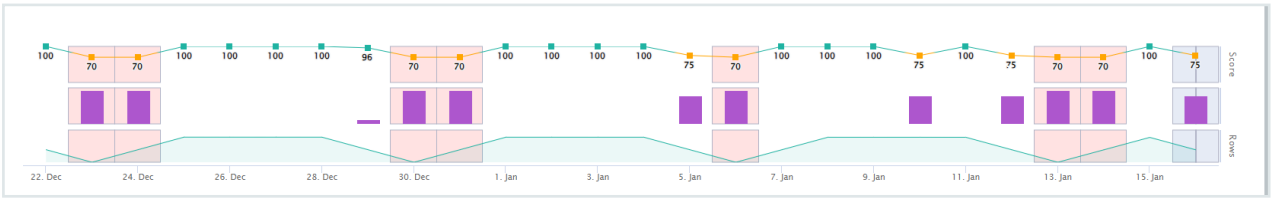
Collibra DQ automatically profiles datasets over time to allow you to drill down into its columns for detailed insights and automated data quality. The creation of a dataset profile is just the first step towards an amazing amount of auto-discovery. Visualize segments of the dataset and how the dataset changes over time.

When you create a job from Explorer, the options on the AdaptiveRules tab of the Add Layers step help to generate a complete profile. While many of the options are enabled by default, you may need to enable additional options, such as Min, Mean, and Max, to refine the profile of your dataset to fit your needs.

> **Tip** Many of the graphs and charts on the Profile page have additional insights when you hover your cursor over them.



Dataset profile graph



The dataset profile graph shows dataset run details, including score, observation type, and row count changes. This is similar to the graph at the top of a dataset's Findings page, where you can hover your cursor over the various graph elements to reveal additional details, such as the number of points deducted per observation type. Additionally, you can click the score entry on the line graph to open the dataset profile of that day's run.

Dataset profile table

The dataset profile table contains a variety of insights about column-level data quality metrics. These automatic insights bring together numerous different statistics based on the results of your DQ Job to reveal the baseline complexion of your dataset. The profile can discover attributes that help to delineate between the relative metrics around numeric versus non-numeric data.

> **Tip** You can click anywhere in a row on the Profile page to expand it and view additional column-level insights.

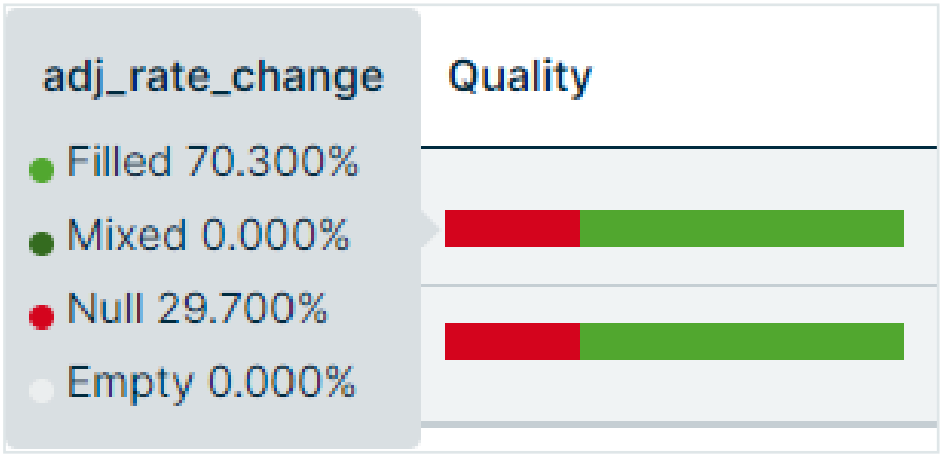
Column	Description
Column	The column of your dataset whose insights the row contains.
Sensitive	View, add, or edit the sensitive label of a column. When a column has a sensitive label, a green chip with the name of the sensitive label displays. If a column does not have a sensitive label, a displays in the column. Go to the Adding Sensitive and Data Class labels section to learn how to add sensitive labels. **Tip** Admin users can manage sensitive labels on the Sensitive Labels Management page of the Admin Console.
Data Class	View, add, or edit the data class of a column. When a column has a data class, a green chip with the name of the data class displays. If a column does not have a data class, a clickable + displays in the column. Go to the Adding Sensitive and Data Class labels section to learn how to add data classes. **Tip** You can manage data classes on the Data Classes page.

Column	Description
Profile	**Baseline (B)** is the historical profile of a dataset, which is the mean value of past lookback runs. For example, the mean number of NULLs in column *x* for the past 10 runs. The baseline is determined by the data lookback period, which you can configure in the settings when creating or editing a DQ Job. While the default lookback is 10 runs, you can set it to anywhere between 3 and 20 runs. The sliding lookback window establishes guardrails for the baseline as the DQ Job runs along a schedule. To reset the baseline, you need to create a new DQ Job. You cannot reset the baseline of an active DQ Job because it would reduce the value of the profile insights from previous days, as there is not a way to identify when the baseline was reset. > **Note** Some runs are excluded from the baseline based on annotations. For example, adaptive rules marked as "breaking" are excluded from the baseline or "normal" values. **Run (R)** is the profile of the current run of a dataset.
Defined Type	The data type of a column when the dataset was created.
Inferred Type	The data type(s) of a column that Collibra DQ learns are possible deviations from the defined type. If a column does not contain a potential deviation, then the data type matches the defined type.

Column	Description
Quality	Hover your cursor over the chart to see a percentage breakdown of the field values of the rows in a column. The represented field values are Filled, Mixed, Null, and Empty, which reflect the baseline profile data, not the current run.

Metric	Type	Description
Filled	[1] Integer	The percentage of data that is numeric (or non-numeric) in a numeric (or non-numeric) discovered column.
Mixed	[String] Integer	The percentage of data that is non-numeric (or numeric) in a numeric (or non-numeric) discovered column.
Null	[]	The percentage of data that has no value at all.
Empty	[""]	The percentage of data that has a string instance of zero length.

See an example



Column	Description
Unique	The unique distribution, also known as cardinality, in a column. This can be an important part of understanding if the values in a column continues to be unique, or if it contains the same values. For example, the expectation of an "ID" column might be that all values should be unique, whereas the expectation of a "date" column might be that all values should be the same. **B** is the number of unique values in the baseline profile of a column. **R** is the number of unique distributions in the profile of a column from the current run. **%** is the percentage the baseline and run values deviate from each other.
Null	The percentage of data that has no value at all. **B** is the percentage of null values in the baseline profile of a column. **R** is the percentage of null values in the profile of a column from the current run. **%** is the percentage the baseline and run values deviate from each other.
Empty	The percentage of data that has a string instance of zero length. **B** is the percentage of empty values in the baseline profile of a column. **R** is the percentage of empty values in the profile of a column from the current run. **%** is the percentage the baseline and run values deviate from each other.
Mixed	The percentage of data that Collibra DQ observes as non-numeric (or numeric) in a numeric (or non-numeric) column. **B** is the percentage of mixed values in the baseline profile of a column. **R** is the percentage of mixed values in the profile of a column from the current run. **%** is the percentage the baseline and run values deviate from each other.

Column insights

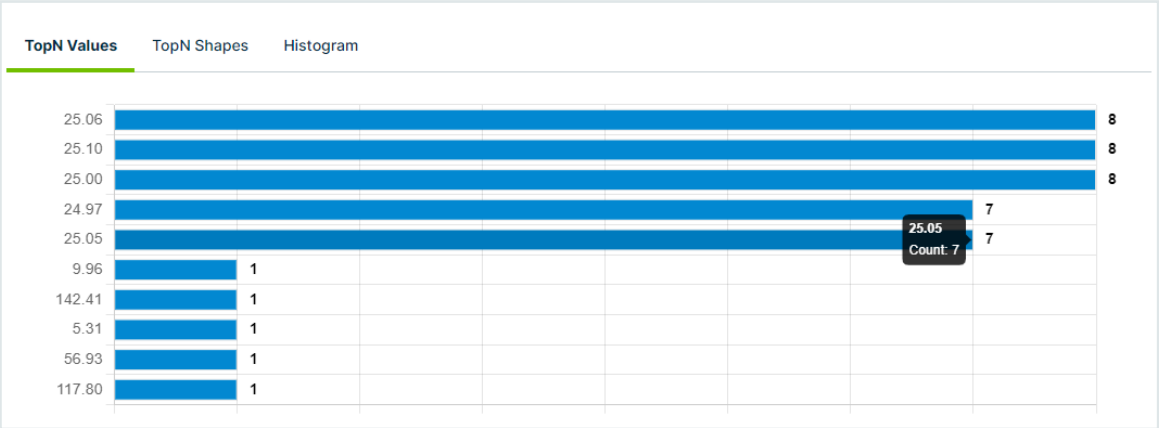
Column insights become visible when you drill down into a column. These insights for baseline and current runs include key statistical data points, data visualization tools, and overviews of Adaptive Rules and Rules. Click the expand icon to the right of the column name to drill down into the column and reveal insights.

Column-level statistics

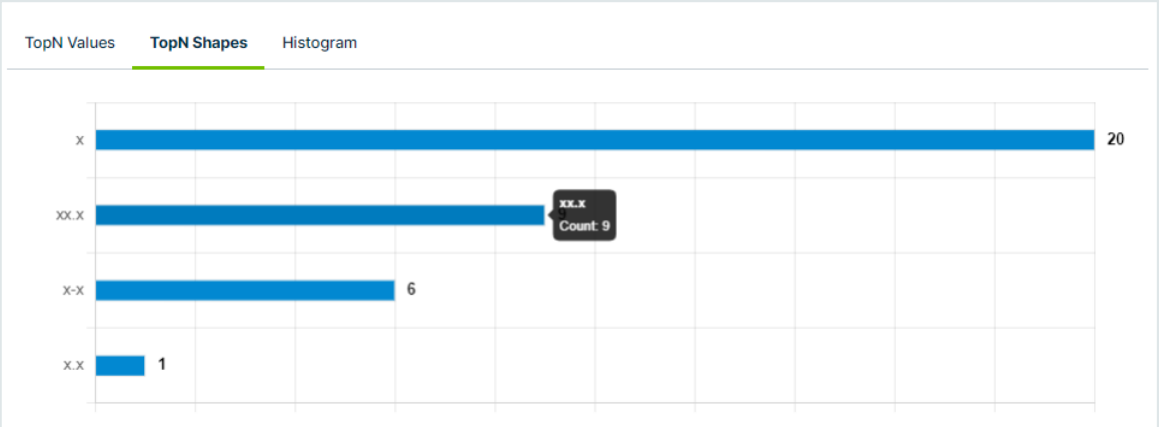
When data is available, a drill down into a column reveals four sections on the left:

- The **Stats** section shows high-level statistics for current run data, including:
  - Min value
  - Q1 (displays "N/A" when the data type is string or a date)
  - Median value (displays "N/A" when the data type is string or a date)
  - Mean value
  - Q3 (displays "N/A" when the data type is string or a date)
  - Max Value
  - Min string length (only displays when the data type is string)
  - Max string length (only displays when the data type is string)
  - Min precision (only displays when the data type is double, float, or decimal)
  - Max precision (only displays when the data type is double, float, or decimal)
  - Min Scale (only displays when the data type is double, float, or decimal)
  - Max Scale (only displays when the data type is double, float, or decimal)
- The **Distribution** section shows the number of unique values for current run data.
- The **Completeness** section shows the following donut charts for baseline run data:
  - Null percentage
  - Empty percentage
  - Mixed data type percentage
- The **Business Term** section shows the name of the business term and its description.
  - You can click the ✎ to the right to apply a business term to your column or edit an existing one.

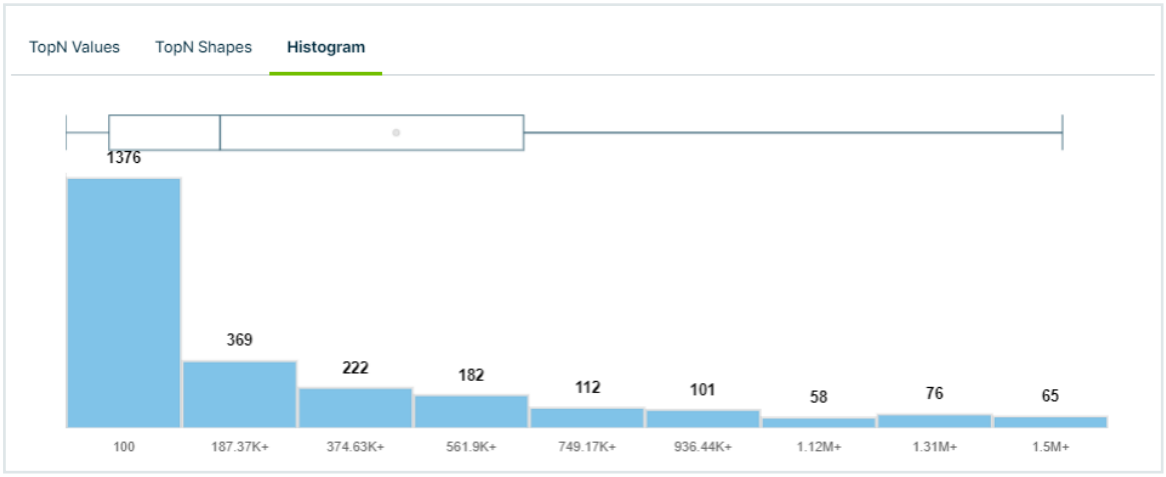
Column-level data visualization



The TopN Values tab shows the cell values that appeared the most and least frequently in the column of reference for the current run. This can help you understand the data distribution within a column, identify the top- and lowest-performing data, and spot trends or patterns. In the example above, the most frequently observed values are 25.06, 25.10, and 25.00, all of which were counted 8 times. The least frequent observed values are 9.96, 142.41, 5.31, 56.93, and 117.80. You can hover your cursor over a bar on the chart to see the exact value and its count.



The TopN Shapes tab shows the string data type cell values that appeared the most and least frequently in the column of reference for the current run. This can help you identify inconsistent data types or unexpected string values, recognize frequently recurring patterns, and understand the distribution of data in the column. In the example above, the most frequently observed shape value is "x," which was counted 20 times. The least frequently observed shape value is "x.x," which was counted only once. You can hover your cursor over a bar on the chart to see the exact shape value and its count.



The Histogram tab shows a graphical representation of the distribution of data within a column for the current run. This provides insights into important data points, including:

- Minimum value
- Lower and upper quartiles
- Median
- Mean
- Maximum value

Hover your cursor over the box and whisker plot to see the above data points, or hover over an individual bar to see the values it represents.

Column-level rules

Column	Description
Field	The column or data quality metric represented in the row.
Check	The type of AdaptiveRule that uses to check for common data quality issues.
Condition	The SQL expression related to the Check that Collibra DQ's machine learning model automatically applies to the column or metric based on what it identifies as normal.
Status	Passing indicates that the values in the column do not contradict the condition of the AdaptiveRule. Breaking indicates that the values in the column contradict the condition of the AdaptiveRule. Suppressed indicates that any findings from the AdaptiveRule are ignored.
	Shows the data point of a given column relative to its baseline. In the example to the left, the red line represents the data point of the value for the current run, overlaid atop the baseline and expected range of that run. When a column has a value well outside the expected range boundary, it triggers a "Breaking" status for that type of check. This range also appears on the Findings page for the breaking column.
Action	Click to open the Change Detection modal.

Shows when a user-generated rule is applied to a column.

Column	Description
Rule Name	The name of your column-level rule.

Column	Description
Value	The condition of your column-level rule.
Valid	Shows whether a rule passes validation. ✓ indicates that a rule is valid. ⚠ indicates that a rule is invalid.
Repo	The data class or template from which a rule is created. This only applies to custom rules, such as Data Type, Data Class, and Template.
Category	The data category that you optionally define on the Workbench.
Dimension	The DQ Dimension that you optionally assign to your rule on the Workbench.
Status	Shows whether your rule is active for future runs of the dataset. Click the switch to change the active status of your rule.



The Histogram tab provides a visual overview of all of the columns in your dataset where histogram data is available. From this tab, you can see the number of distinct values that a column contains, as well as the frequency when you hover your cursor over a histogram bar.

The histograms are calculated using the number of buckets. For example, **numeric type** buckets can be calculated based on the difference between Q1 and Q3 values. The number of buckets are then used in a WIDTH_BUCKET Snowflake function, which defines the number of bars and creates the histogram.

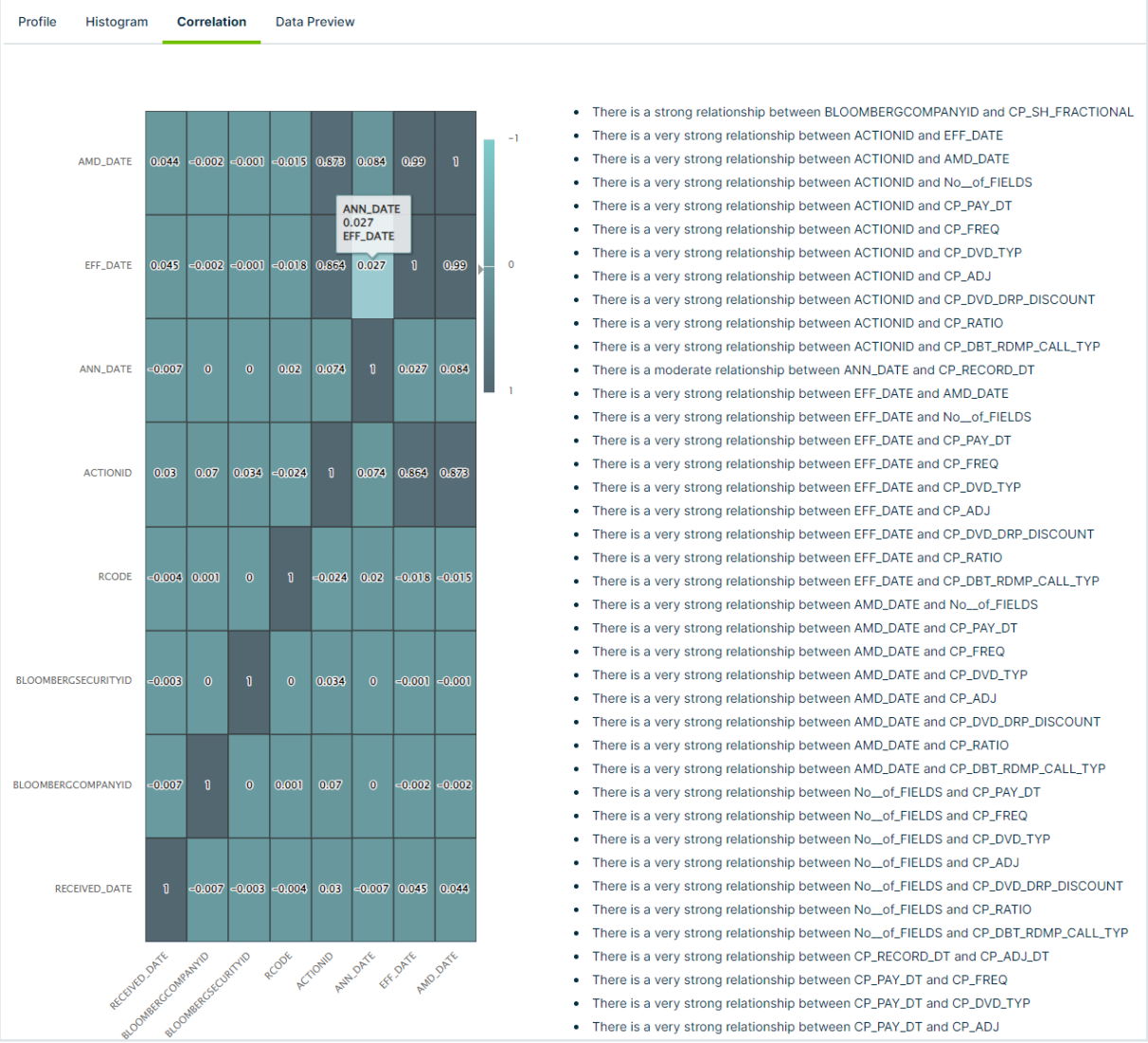
The number of bars vary. For example, **string types** can show 2 bars (1 for min count and 1 for max count) or only 1 bar if the min and max counts are identical.

For example, the following queries for `DQ_String` return the same results and would only show as one bar in the histogram:

- `SELECT 'DQ_STRING' AS column_name, "DQ_STRING" AS minimum_string, COUNT(1) AS minimum_count FROM (select * from PUBLIC.TEST_16744) GROUP BY column_name, minimum_string ORDER BY minimum_count ASC LIMIT 1`
- `SELECT 'DQ_STRING' AS column_name, "DQ_STRING" AS maximum_string, COUNT(1) AS maximum_count FROM (select * from PUBLIC.TEST_16744) GROUP BY column_name, maximum_string ORDER BY maximum_count DESC LIMIT 1`

> **Note**
> - The histograms are only generated for numeric and string types, not for date/timestamp types.
> - Values outside of the upper and lower bound ranges are not included in the histograms. Specifically, values that are less than the lower bound and greater than or equal to the upper bound.

Correlation



The correlation tab shows a graphical matrix of the correlation coefficient between two or more columns in your dataset. This can help you discover hidden column relationships in your dataset and measure their strength. Using just a few examples from the above screenshot, Collibra DQ observed the following relationships:

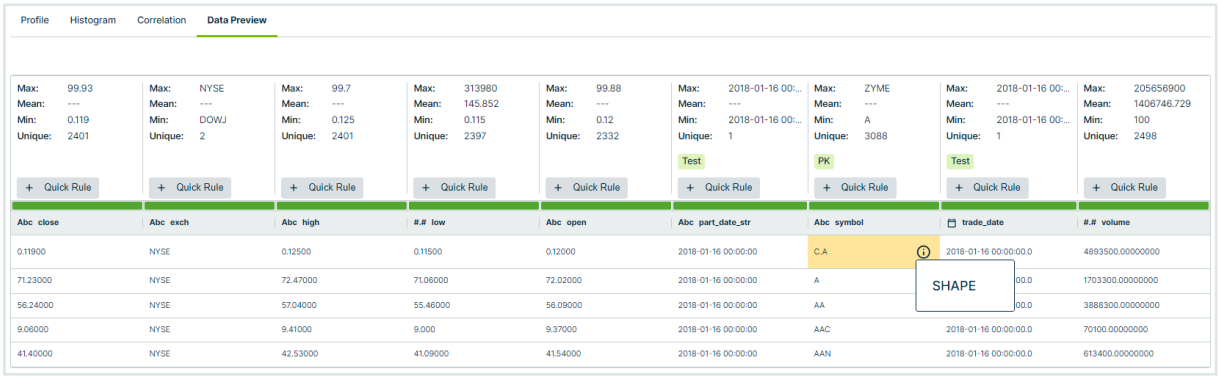
- There is a very strong relationship between the columns "ACTIONID" and "EFF_DATE"

- There is a strong relationship between the columns "BLOOMBERGCOMPANYID" and "CP_SH_FRACTIONAL"
- There is a moderate relationship between the columns "ANN_DATE" and "CP_RECORD_DT"

You can hover your cursor over any cell to see the correlation details between the two columns.

> **Note**
>
> - Larger datasets with many relationships between columns might not display all columns in your dataset on the matrix. However, relationships between non-visible columns will still display in the list on the right side of the matrix, when applicable.
> - The matrix displays a maximum of 8 columns, regardless of the number of columns returned from the API response.

Data Preview



The Data Preview tab shows a preview of the data in the columns in your dataset. This gives you insights into high-level statistics, such as:

- Maximum value in the column
- Mean value
- Minimum value

- Number of unique values
- Columns with labels or designations, such as Sensitive Labels and Parent Keys

You can also see columns where Collibra DQ discovered specific detections, such as the Shape observation in the "symbol" column in the example above. Additionally, you can click **+ Quick Rule** and select either a Data Type, Data Class, or Template rule from the dropdown menu to automatically apply it to your dataset. You can then find the quick rule on the Dataset Rules page of your dataset.

Working with the Change Detection modal

The Change Detection modal appears when you click the Actions button. Select either Adaptive, Manual, or Suppress from the options below to see details about each alert type.

> **Note** If you change an Adaptive Rule from Manual to Adaptive, you need to retrain your dataset by clicking Retrain on the Behavior tab of the Findings page.

Adaptive Level	Description
Strict	Rigidly enforce and alert of any data points that fall outside what Collibra DQ's behavioral model learns to be normal. Any values outside the strict boundaries will be flagged as potential data quality issues.
Neutral/System Detected	Automatically enforce and alert any data points that fall outside what Collibra DQ's behavioral model learns to be normal. Any values outside the neutral boundaries will be flagged as potential data quality issues.
Lenient	Relax the enforcement and alerting when data points fall outside of the range that Collibra DQ's behavioral model learns to be normal. Any values outside the lenient boundaries will be flagged as potential data quality issues. **Tip** This can help avoid excessive alerts when Collibra DQ is in the learning phase of a new dataset.

Adaptive Level	Description
Pass Value on [Date]	Select this option to accept the value from a given date into the rolling behavioral baseline.

Adaptive level calculations

The adaptive level calculations vary depending on the specific type of data quality check being performed. The table below shows the check type, the thresholds of the three adaptive levels in relation to the check type, and the areas each check type impacts.

Adaptive level	Threshold	Impacts
Cardinality (Standard Deviation)		
Strict	Cardinality standard deviation threshold (ABS): 2.1 Cardinality standard deviation min/max threshold (z-score for min/max drift): 3.1	• Uniques (Cardinality) • Mean (Standard Deviation)
Neutral/System Detected	Cardinality standard deviation threshold (ABS threshold): 2.5 Cardinality standard deviation min/max threshold (z-score for min/max drift): 3.4	
Lenient	Cardinality standard deviation threshold (ABS threshold): 3.9 Cardinality standard deviation min/max threshold (z-score for min/max drift): 6.1	
Cardinality (Percent Change)		

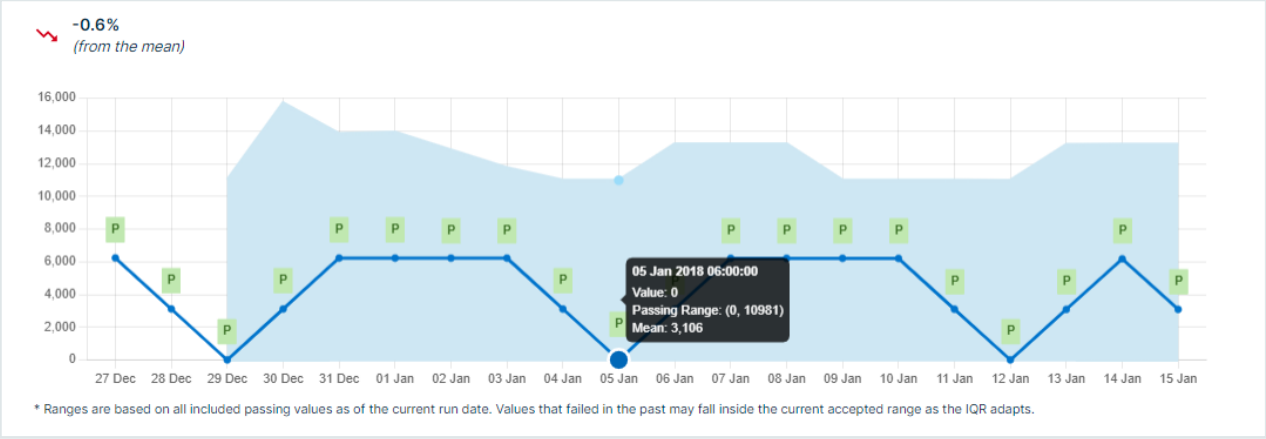
Adaptive level	Threshold	Impacts
Strict	Cardinality percent change threshold: 11%	• Uniques (Cardinality) • Type (Percent Mixed Types)
Neutral/System Detected	Cardinality percent change threshold: 14%	
Lenient	Cardinality percent change threshold: 45%	
Min/Max (Standard Deviation)		
Strict	Min/max standard deviation threshold (z-score for min/max drift): 3.1	• Row Count TopN/BottomN (Standard Deviation/MinMax) • Minimum/Maximum (Min/Max) • Mean (Standard Deviation)
Neutral/System Detected	Min/max standard deviation threshold (z-score for min/max drift): 3.4	
Lenient	Min/max standard deviation threshold (z-score for min/max drift): 6.1	
Min/Max (Percent Change)		
Strict	Min/max percent change threshold: 12%	• Row Count TopN/BottomN (Standard Deviation/MinMax) • Minimum/Maximum (Min/Max) • Type (Percent Mixed Types)
Neutral/System Detected	Min/max percent change threshold: 18%	
Lenient	Min/max percent change threshold: 65%	
Standard Deviation Threshold (Deviation from the baseline)		
Strict	Standard deviation threshold: 2.3	• Mean (Standard Deviation) • Percent Null/Empty (for Lenient only)
Neutral/System Detected	Standard deviation threshold: 2.8	
Lenient	Standard deviation threshold (null, empty): 3.5	
Percent Change Threshold		

Adaptive level	Threshold	Impacts
Strict	Percent change threshold: 3%	• Percent Null/Empty • Type (Percent Mixed Types)
Neutral/System Detected	Percent change threshold: 8%	
Lenient	Percent change threshold: 40%	

Option	Description
Lower Bound	Alerts when Collibra DQ discovers data in the selected column lower than the limit you set for the Adaptive Rule.
Upper Bound	Alerts when Collibra DQ discovers data in the selected column higher than the limit you set for the Adaptive Rule.
pt(s)	The number of points to deduct from the data quality score for a given finding. This relates to the percent change and Z-score when the Z-score ranges from 0.0-6.0.

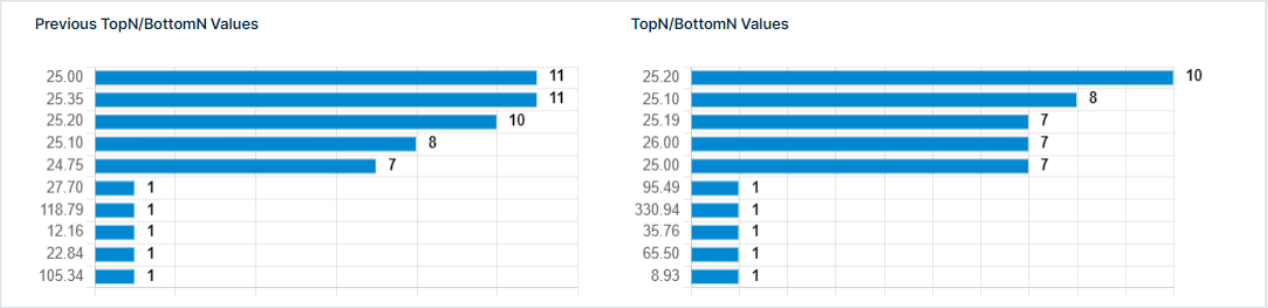
Turns off alerts for unique findings in the selected column.

Change Detection chart



The Change Detection line chart shows the range of all passing, failing, and inactive values between the current run date and 20 runs in the past. You can hover your cursor over any available day to view important statistics for that run.

Previous and current TopN/BottomN Values charts



The previous and current TopN/BottomN Values charts show the row values that appeared the most and least frequently and satisfy the condition of the Adaptive Rule in the column of reference. In the above example, the Adaptive Rule is a Unique Range check on the "high" column where the condition is `high.$uniqueCount NOT BETWEEN 739.9 AND 6963.0`. Therefore, the values that appear the most and least frequently and are either greater than 6963.0 or less than 739.9 are considered the TopN and BottomN values.

Indeed, in the above screenshot, the TopN row values of the previous run in the chart on the left were 25.00 and 25.35, which both appeared 11 times. The BottomN values were 27.70, 118.79, 12.16, 11.84, and 105.34, which all appeared only once for that run. On the chart to the right, the TopN value from the current run is 25.20 with 10 occurrences, while the BottomN values are 95.49, 330.94, 35.76, 65.50, and 8.93 with one occurrence apiece.

Adding Quick Rules

You can create quick rules from the Profile page based on common column-level checks, including unique count, valid values, and percentage distribution.

Steps

1. From the search bar at the top of the page, enter the name of a dataset, click it when it appears in the results, and then click **Profile** to the right of the search bar.
   » The Profile page of your dataset opens.

2. In the first column of the Profile table, click the name of a column to expand it.
   » The column expands.
3. In between the column-level statistics and rules sections, click **+ Add Rule**.
4. Select an option and enter the required information in the modal.

Unique rules checks the specified column to verify that it only contains unique values. This can help you identify columns containing duplicate values by measuring the number of unique values ($uniqueCount) against the number of rows ($rowCount). If the number of unique values is not the same as the number of rows in the column, this indicates that the column contains duplicates.

Setting	Description	Default Value
Dataset	The name of the dataset that this unique rule queries. **Note** This field is not editable.	N/A
Rule Name	The name of your unique rule.	The naming convention uses the following format: [column_name]_unique
Where	The condition of the quick rule. **Note** This field is not editable.	The condition uses the following format: `[column_name].$uniqueCount != $rowCount`
Points	The number of points deducted from the total DQ score when your rule breaks. Use the default or select a value greater than 0.	1

Setting	Description	Default Value
Percentage	Deducts points according to the point threshold you set against every percentage of records in violation of your rule. For example, If 25 of 100 records break your rule, and the values you set in both the Points and Percent input fields are 1, then a total of 25 points are deducted from your overall DQ score.	1
Preview Limit	The number of available rows to preview when you drill into a rule record on the Findings page.	1
Run Time Limit (minutes)	The number of minutes that a rule takes to process before it times out and proceeds to the next activity in the job.	1

Valid value rules check a variety of range- or match-based conditions within your column. For example, a valid value rule based on matching (valid) values on a stock exchange dataset with a column called "trade_date" where `trade_date NOT IN ('2018-01-16 00:00:00.00')` checks if any rows in the trade_date column do not match the date stamp of 2018-01-16 00:00:00.00. This can help filter columns containing many rows of data to only show the rows that do not meet the criteria of the valid value rule.

Another type of valid value rule is based on whether a value within a column falls outside a range of specified values. For example, using the same stock exchange dataset, a valid value rule that checks the range of values within a column called "high" where `high not between 0.00000 and 387831` checks if any rows in the high column are not within the range of 0.00000 and 387831. Like the match-based valid value rule, this also helps to filter out values that fall outside the constraints of the query.

Setting	Description	Default Value
Dataset	The name of the dataset that this valid value rule queries. **Note** This field is not editable.	N/A
Rule Name	The name of your valid value rule.	The naming convention uses the following format: [column_name]_valid_value
Where	The condition of your valid rule.	The condition uses the following formats: `[column_name] not between [value] and [value]` or `[column_name] NOT IN ('$rowValue')`
Points	The number of points deducted from the total DQ score when your rule breaks. Use the default or select a value greater than 0.	1
Percentage	Deducts points according to the point threshold you set against every percentage of records in violation of your rule. For example, If 25 of 100 records break your rule, and the values you set in both the Points and Percent input fields are 1, then a total of 25 points are deducted from your overall DQ score.	1
Preview Limit	The number of available rows to preview when you drill into a rule record on the Findings page.	1

Setting	Description	Default Value
Run Time Limit (minutes)	The number of minutes that a rule takes to process before it times out and proceeds to the next activity in the job.	1

Distribution rules check if the uniqueness percentage of a column falls outside of a specified range of possible values. Distribution rules can be set up as either a combined or individual rule.

In a combined distribution rule, you can select from a list of TopN and BottomN values to automatically add the value to a compound condition using the operator "OR" to include multiple checks in the condition. So, when the uniqueness percentage of any of the TopN or BottomN values you include in your condition does not meet the specified percentage range, the condition is satisfied.

In an individual distribution rule, you can select from the same list of TopN or BottomN column values as in the combined distribution rule, but instead of combining each check into one condition, each check is treated as its own rule.

<table>
<tr><th>Setting</th><th>Description</th><th>Default Value</th></tr>
<tr><td>TopN Values</td><td>The most frequently occurring values in a column. The following table details the columns in the TopN Values table.

<table><tr><th>Column</th><th>Description</th></tr><tr><td>Value</td><td>The value of a row in the selected column.</td></tr><tr><td>Count</td><td>The number of times the value occurs in the column.</td></tr><tr><td>%</td><td>The percentage of occurrences of the value within the selected column.</td></tr><tr><td>Change %</td><td>Set a distributed range percentage to check for values that are outside of the +/- of that value, applied to the % column.

Example If a value in the % column is 2.273 and you set the Change % to 10, then you can expect the condition in the Where column to be <code>.$uniquePercent not between 2.045 and 2.500</code> because 2.045 and 2.500 are respectively -10% and +10% of 2.273.</td></tr><tr><td>Add rule?</td><td>Select the checkbox option to include the rule in the Where condition. The value you set in the Change % field determines the <code>not between</code> distribution values when you add the rule to the Where condition.</td></tr></table></td><td>N/A</td></tr>
</table>

Setting	Description	Default Value
Dataset	The name of the dataset that this unique rule queries. **Note** This field is not editable.	N/A

Under the Show Rule section, select **Combined** to combine all of the TopN or BottomN values you selected in the **Add rule?** column into one rule, or select **Individual** to create individual rules per TopN or BottomN value.

Setting	Description	Default Value
Rule Name	The name of the quick rule.	The naming convention uses the following format: [column_name]_valid_value **Note** If you select Individual Rule and you include multiple TopN values, then multiple Rule Name columns will display.
Where	The condition of the quick rule.	The condition uses the following formats: `$columnName ['$rowValue'].$uniquePercent not between [value] and [value]` **Note** If you select Individual Rule and you include multiple TopN values, then multiple Where columns will display.

Setting	Description	Default Value
Points	The number of points deducted from the total DQ score when your rule breaks. Use the default or select a value greater than 0.	1
Percentage	Deducts points according to the point threshold you set against every percentage of records in violation of your rule. For example, If 25 of 100 records break your rule, and the values you set in both the Points and Percent input fields are 1, then a total of 25 points are deducted from your overall DQ score.	1

5. Click **Save**.

> **Note** Your quick rule will be checked against your dataset the next time your job runs.

Adding Data Class and Sensitivity Labels

You can add Data Class and Sensitivity labels at the column-level on the Profile page. Collibra DQ can automatically detect the following types of PII:

- EMAIL
- PHONE
- ZIP CODE
- STATE CD
- CREDIT CARD

- GENDER
- SSN
- IP ADDRESS
- EIN

Once Collibra DQ discovers a sensitive data type in a column, you can then add a Sensitivity Label to flag it as sensitive. You can also apply a Data Class label to a column to automatically create a Data Class rule against the column you select. You can then see your newly created Data Class rule on the Dataset Rules page.

Steps

1. From the search bar at the top of the page, enter the name of a dataset, click it when it appears in the results, and then click **Profile** to the right of the search bar.
   » The Profile page of your dataset opens.
2. In either the Sensitive or Data Class column of the Profile table, click the **+** icon.
   » The Add Data Class/Sensitivity modal appears.
3. Enter the required information.

Option	Description
Dataset	The name of your dataset.
Column	The column to which the sensitivity or data class label will apply.

Option	Description
Sensitivity	A dropdown menu of all available sensitivity labels in your Collibra DQ environment. **Tip** Admin users can manage sensitivity labels from the Sensitive Labels Management page in the Admin Console.
Data Class	A dropdown menu of all available data classes in your Collibra DQ environment. **Tip** You can manage data class labels from the Data Classes page under the Rules tab.

4. Click **Save**.

Note
If you remove a Data Class from a column on the Profile page, the Data Class rule created when you applied it is still considered an active rule. To remove the Data Class rule as well, go to the Dataset Rules page and delete it from the Actions dropdown menu. Furthermore, when you delete a Data Class rule from the Dataset Rules page, the Data Class label is still applied to the column on the Profile page.

DQ Rules

Data quality rules are custom or out-of-the-box SQL conditions that help you verify that your data adheres to your organization's business requirements. The ability to detect when data deviates from standards can be critical in meeting regulatory compliance rules, discovering unusual or inaccurate records, and maintaining the overall health of your organization's data. While Data Quality & Observability Classic automatically makes many different types of behavioral observations and creates AdaptiveRules based on what it learns, you may discover situations where you need to write specific compliance-related rules or import your own. This section shows you how you can create and manage rules to ensure that your data is fit for use to meet your business requirements.

> **Warning** While Data Quality & Observability Classic supports the manual compilation of SQL, the misuse of raw SQL presents an elevated security risk (SQLi).

Rule Builder

The Rule Builder consists of two primary components:

- Dataset Rules where you can view and manage all SQL rules on the dataset-level.
- The Rule Workbench where you can build and refine new and existing SQL rules on the dataset-level, apply DQ Dimensions, and configure scoring weights.

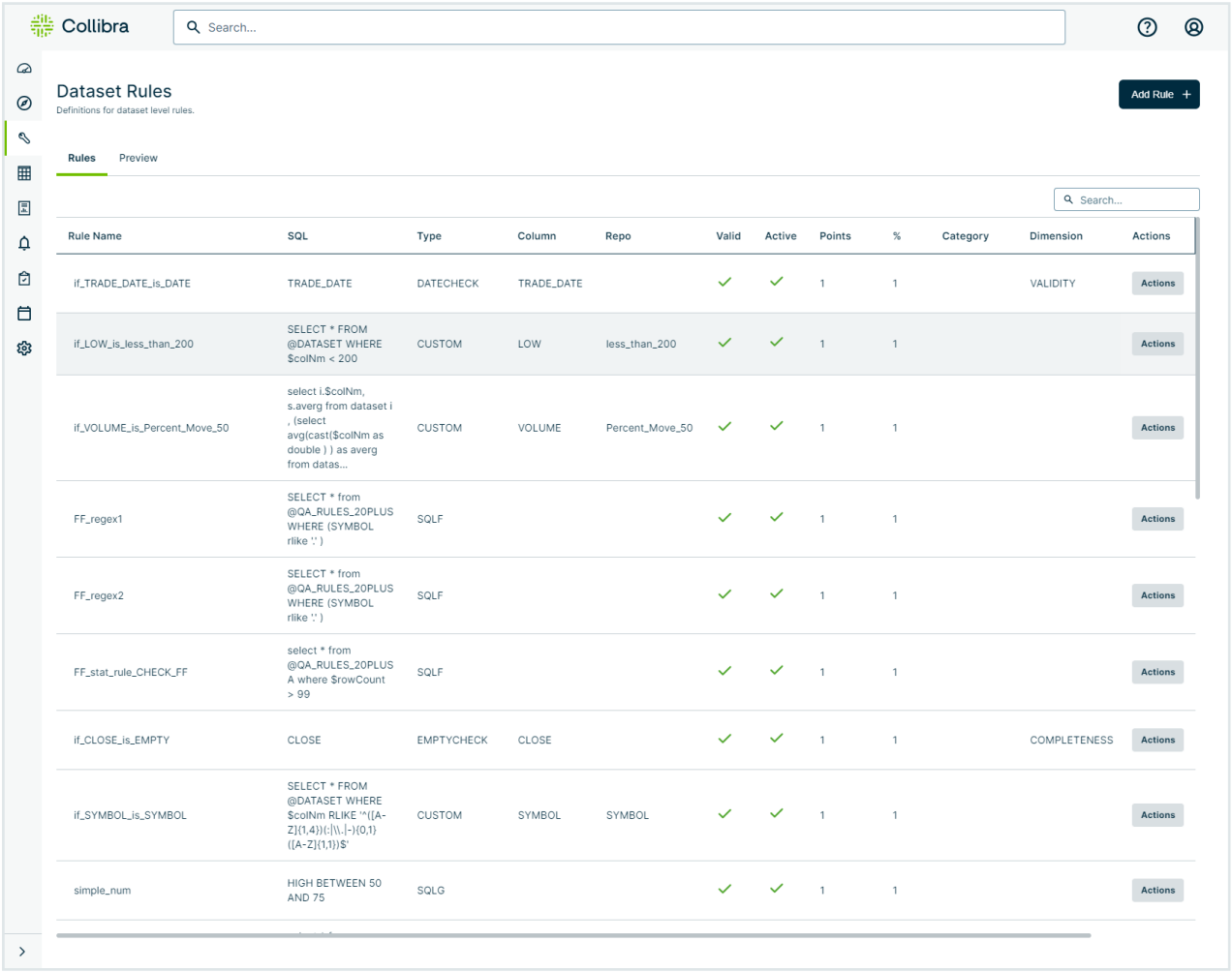
Prerequisites

You need the following roles to view, create, and edit rules:

- ROLE_ADMIN
  or
- ROLE_DATASET_RULES
  or
- ROLE_DATASET_MANAGER

Working with Dataset Rules

The Dataset Rules page lists all previously saved rules for a given dataset and provides an overview of their details, such as the definitions of SQL conditions, rule types, and whether or not rules pass validation checks. You can also take several quick actions from this page, such as accessing the Rule Workbench or deleting a rule from a dataset. The ability to manage rules from a single page can shorten the amount of time you spend assessing what is needed to meet your business requirements.



Opening Dataset Rules

The following steps show you how to open the list of applicable rules on your dataset.

1. Click the 🔧 in the sidebar menu, then click **Rule Builder**.
   » The Dataset Rules page opens.

2. On the Dataset Rules page, enter the name of your dataset in the **Search for a Dataset** searchbar, then select it.
   » The Dataset Rules page displays any existing rules on your dataset.

Viewing rule details

When one or more rules are available on the selected dataset, the table on the Dataset Rules page contains the following detailed information.

Column	Description
Rule Name	The name of the rule. Hover your cursor over the rule name and click ✎ to edit the rule.
Rule Query	The SQL condition of the rule.
Type	The type of rule.
Column	The primary column that the rule queries.
Repo	The data class or template from which the rule is created. This only applies to custom rules, such as Data Type, Data Class, and Template.
Valid	Shows whether the rule passes rule validation. ✓ shows that the rule passes validation. ⚠ shows that the rule does not pass validation. **Note** If you see ⚠, you can click it to view more details about why the rule does not pass validation.
Active	Shows whether the rule is active for future runs of the dataset. Click the icon to change the active status of the rule. ✓ shows that the rule is active. ⊗ shows that the rule is inactive.

Column	Description
Scoring Type	Shows whether the rule uses absolute- or percentage-based scoring. For more information about scoring types, go to Create a data quality rule and Understanding the rules score.
Tolerance	Specifies the tolerance threshold as a percentage. For more information, go to Understanding the rules score.
Points	The number of points that Collibra DQ deducts from the data quality score when data breaches the conditions of the rule. You can set this value on the Rule Details modal of the Rule Workbench. For more information, go to Understanding the rules score. If you do not customize this value, then Collibra DQ uses the default value of 1.
%	The ratio of the total number of breaking records over the total number of rows. For more information, go to Understanding the rules score. If you do not customize this value on the Rule Details modal, then Collibra DQ uses the default value of 1.
Category	The data category that you optionally define on the Workbench.
Dimension	The DQ Dimension that you optionally assign to the rule on the Workbench. **Note** Tagging rules with custom dimensions in the Metastore is not supported.

Column	Description
Timeout Limit	The number of minutes that any active rule can take to process before it times out and the tool automatically cancels the job. This limit is useful for keeping problematic rules from consuming too many resources. The maximum timeout limit is determined by the greatest Timeout Limit value of any active rule displayed on the table under the Rules tab. **Example** If there are 3 active rules associated with a dataset, where Rule 1 has a timeout limit of 20 minutes, Rule 2 has a limit of 30 minutes, and Rule 3 has a limit of 60 minutes, then the dataset has 60 minutes per rule to process. If Rule 3 is inactive, the maximum timeout limit for the dataset is 30 minutes, because Rule 2 has the next highest maximum limit of any active rule. **Tip** You can increase this value for an individual rule from Rule Details on the Rule Workbench.
Actions	Click **Actions** to edit, delete, rename, and view the history of the rule. **Note** • After a rule is renamed, its new name is automatically reflected across all relevant pages in Data Quality & Observability Classic. • Updated rule names cannot match the name of an existing rule and must contain only alphanumeric characters without spaces. **Warning** When you rename a rule, all associated rule output records and rule break records on the Findings page, as well as exports stored in the Metastore, are renamed. To retain the history with the original rule name, create a new rule and deactivate the original one.

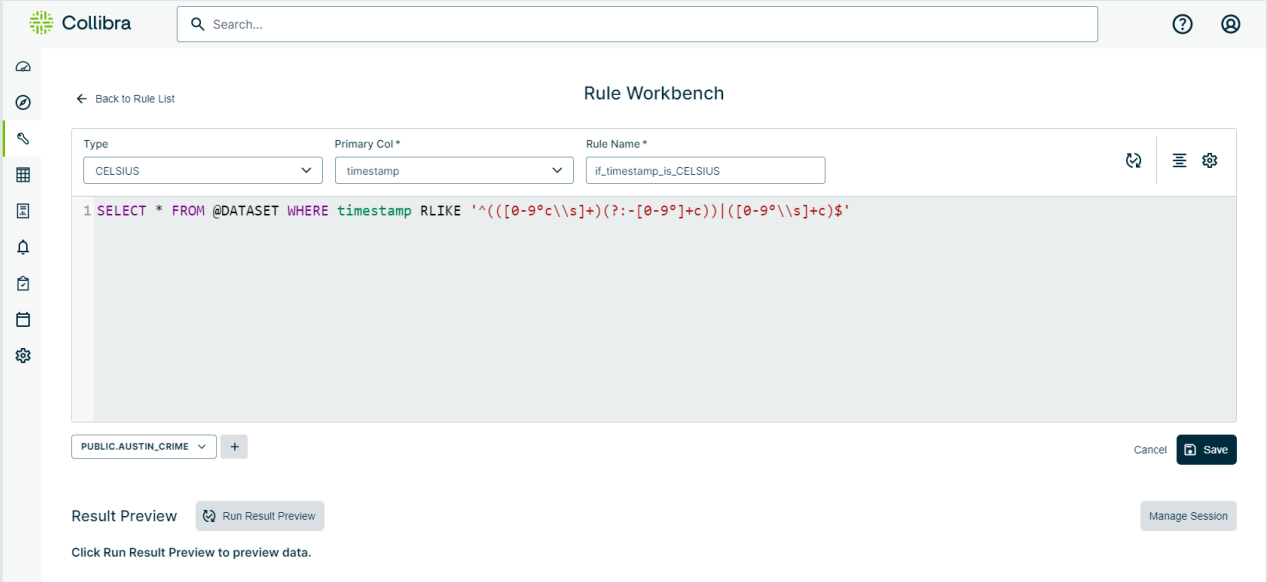
Adding new rules

Click **Add Rule** to create new rules to run against your dataset.

Working with the Rule Workbench

The Rule Workbench is a SQL IDE rule builder that lets you create, edit, and preview SQL conditions to help verify that your data adheres to your organization's business requirements. The Rule Workbench has built-in auto-complete functionality, allowing you to generate more complex SQL such as rules with multi-part conditions. You can also update the settings of your rule, such as applying custom scoring weights, tagging DQ Dimensions and Categories, and other options to reflect the specifics of your data models.

This section shows how you can design and build a SQL rule on the Rule Workbench.



Known limitations

Workbench Component	Limitation
Result Preview	Rule Result Preview is not yet available for Pushdown. This will be addressed in an upcoming release.
Result Preview	Native rules for Pullup jobs are currently not supported.
Result Preview	Stat rules, @runId, and secondary datasets are currently not supported for Pullup jobs.
Result Preview	Secondary datasets are currently not supported for Pushdown jobs.

What's next?

- Create a rule
- Filter a rule
- Use SQL Assistant for Data Quality to automatically generate or troubleshoot a rule

Create a data quality rule

This section shows how you can design and build a SQL rule on the Rule Workbench.

Prerequisites

You have:

- ROLE_ADMIN, ROLE_DATASET_RULES, or ROLE_VIEW_DATA
- An existing dataset against which you want to write a rule.

Steps

Select a navigation method from the following tabs.

> **Tip**
> If you plan to create a rule with SQL Assistant for Data Quality, ensure you meet the prerequisites detailed on Prerequisites and follow the steps on Requirements and permissions.

1. Click the in the sidebar menu, then click **Rule Builder**.
   » The Dataset Rules page opens.
2. There are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench page opens with the preferences of your existing rule.
   b. Click **Add Rule** in the upper right corner to create a new rule.
      » The Rule Workbench page opens.

3. Select a rule type from the Type dropdown menu.

Option	Description
User Input	User Input rules are custom SQL statements that can be either short or long form, depending on the complexity.
Simple Rule (SQLG)	Simple rules are SQL statements that use just the condition and primary column name. For example, `AMOUNT > 100` Simple Rules use Spark SQL syntax.
Freeform SQL (SQLF)	Freeform rules are complete SQL statements that are commonly used when more complex SQL rules are necessary. Freeform Rules can include multiple conditions, table joins, common table expressions (CTEs), and window statements, among other operations. Freeform Rules use Spark SQL syntax.
Native SQL	Native rules use SQL statements with functions or expressions that may be specific to the underlying connection or database (Database SQL). For example, a Native SQL rule written on a dataset whose data resides in Snowflake uses the SQL query syntax native to Snowflake. Remote files are not eligible for Native SQL rules. **Note** All Pushdown rules use the SQL dialect native to the specific database of your source data. Because all Pushdown rules technically use Native (Database) SQL, the Native SQL option is not available for Pushdown jobs on the Rule Workbench. To check whether a rule conforms to the SQL dialect of the source database, click **Validate** after you write your query.
Data Type	Data Type rules check for nulls, empty values, or specific data types.
Empty Check	Checks whether the target column has empty values or not.
Null Check	Checks whether the target column has NULL values or not.

Option	Description
Date Check	Checks whether the target column has **only** DATE values or not.
Int Check	Checks whether the target column has **only** integer values or not.
Double Check	Checks whether the target column has **only** DOUBLE values or not.
String Check	Checks whether the target column has **only** STRING values or not.
Mixed Type Check	Checks the dataType of the field. **Note** Mixed datatype rules are not supported in Pushdown.
Data Class	See the Data Class page for a list of out-of-the-box Data Class rules.
Template	See the Template page for a list of out-of-the-box Template rules.

4. Enter a name for your rule in the **Rule Name** input field.

   Important Rule names cannot contain spaces.

5. Select a primary column from the Primary Col dropdown menu.

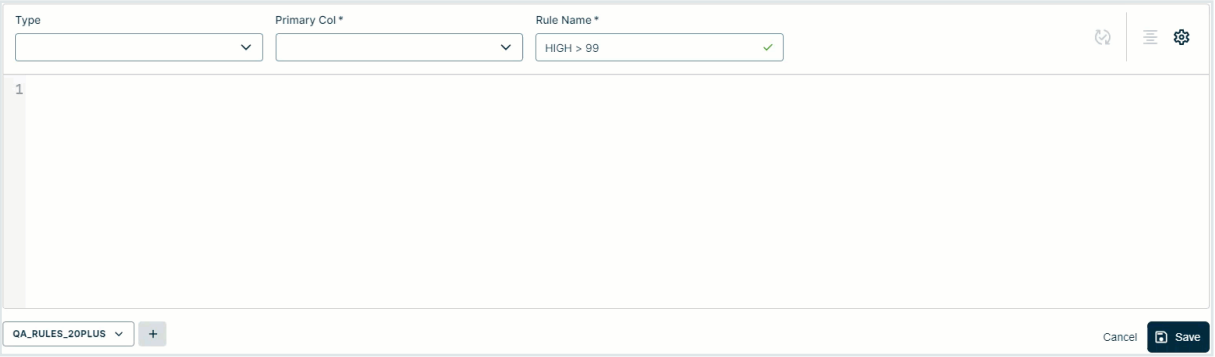
   Note A primary column is required only when the **MANDATORY_PRIMARY_RULE_COLUMN** application configuration setting is set to **TRUE** or if you use a Data Type, Data Class, or Template rule.

6. Optionally click + to add a secondary dataset.
   a. Enter the name of your secondary dataset in the search bar when the **Add Secondary Dataset** dialog appears, then click **Add Secondary Dataset**.

      Important If you edit the rule later, you will need to add the secondary dataset again. Secondary datasets do not persist when a rule is edited.

> Secondary datasets in the query itself are still referenced when the job is run.

7. Write your SQL query. When you select the Freeform rule type, as shown in the GIF below, the Workbench pre-populates the beginning of the query for you. For example, `SELECT * FROM @<dataset name>`. As you type, the Workbench provides a suggested options with common SQL operators and column names relative to your primary and secondary datasets.



> **Tip**
>
> - If your SQL query contains columns with special characters, you need to escape them with backticks ` ` to ensure the rule runs successfully. For example, `` `col_à` IS NOT NULL `` and `SELECT * FROM example_table where` `` `col_à` `` `is not null`. This is due to a Spark limitation which causes a syntax parsing error when the DQ Job runs.
> - For a complete list of Spark SQL operators and functions available for use on SQLG and SQLF rule types, see the official Spark documentation.

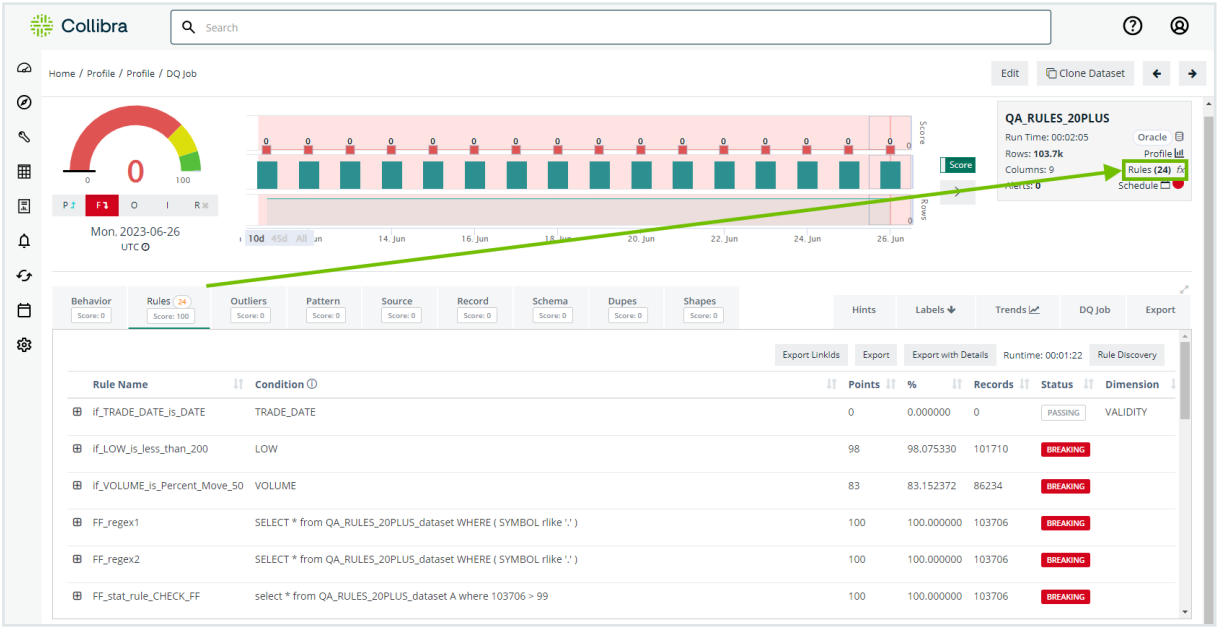
> **Note** If your SQL query is very lengthy, there is an increased likelihood that it may fail to validate or save. To mitigate this issue, we recommend breaking down such queries into small, more manageable rules. A possible way of doing this is by leveraging an `IN` clause.

8. Optionally add a filter query in the Filter input field.

9. Click **Validate** to validate your expression.
10. Optionally click **Beautify** to format your SQL query in a more readable layout.
11. Click **Save** to apply your rule to your dataset.

> **Tip**
> If you plan to create a rule with SQL Assistant for Data Quality, ensure you meet the prerequisites detailed on Prerequisites and follow the steps on Requirements and permissions.

1. On the upper right corner of the Findings page, click the Rules link in the metadata box.
   » The Dataset Rules page opens.



2. There are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench page opens with the preferences of your existing rule.
   b. Click **Add Rule** in the upper right corner to create a new rule.
      » The Rule Workbench page opens.
3. Select a rule type from the Type dropdown menu.

Option	Description
User Input	User Input rules are custom SQL statements that can be either short or long form, depending on the complexity.
Simple Rule (SQLG)	Simple rules are SQL statements that use just the condition and primary column name. For example, `AMOUNT > 100` Simple Rules use Spark SQL syntax.
Freeform SQL (SQLF)	Freeform rules are complete SQL statements that are commonly used when more complex SQL rules are necessary. Freeform Rules can include multiple conditions, table joins, common table expressions (CTEs), and window statements, among other operations. Freeform Rules use Spark SQL syntax.
Native SQL	Native rules use SQL statements with functions or expressions that may be specific to the underlying connection or database (Database SQL). For example, a Native SQL rule written on a dataset whose data resides in Snowflake uses the SQL query syntax native to Snowflake. Remote files are not eligible for Native SQL rules. **Note** All Pushdown rules use the SQL dialect native to the specific database of your source data. Because all Pushdown rules technically use Native (Database) SQL, the Native SQL option is not available for Pushdown jobs on the Rule Workbench. To check whether a rule conforms to the SQL dialect of the source database, click **Validate** after you write your query.
Data Type	Data Type rules check for nulls, empty values, or specific data types.
Empty Check	Checks whether the target column has empty values or not.
Null Check	Checks whether the target column has NULL values or not.
Date Check	Checks whether the target column has **only** DATE values or not.

Option	Description
Int Check	Checks whether the target column has **only** integer values or not.
Double Check	Checks whether the target column has **only** DOUBLE values or not.
String Check	Checks whether the target column has **only** STRING values or not.
Mixed Type Check	Checks the dataType of the field. **Note** Mixed datatype rules are not supported in Pushdown.
Data Class	See the Data Class page for a list of out-of-the-box Data Class rules.
Template	See the Template page for a list of out-of-the-box Template rules.

4. Enter a name for your rule in the **Rule Name** input field.

   Important Rule names cannot contain spaces.

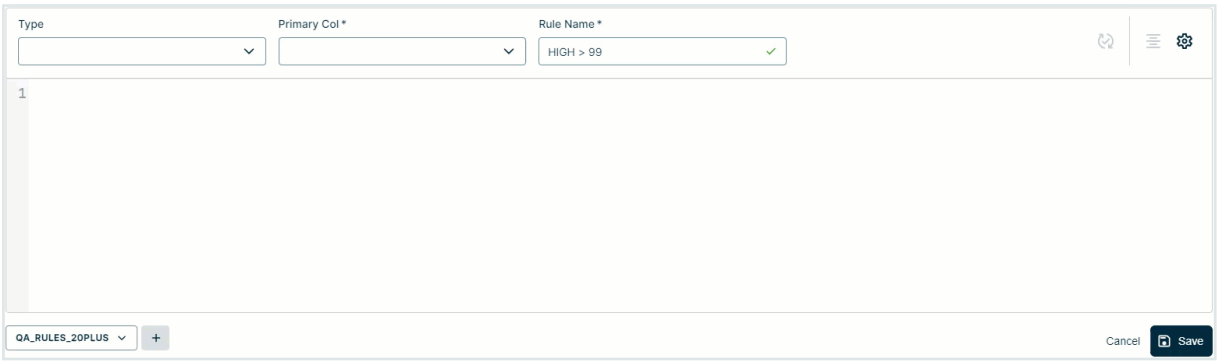
5. Select a primary column from the Primary Col dropdown menu.

   Note A primary column is required only when the **MANDATORY_PRIMARY_RULE_COLUMN** application configuration setting is set to **TRUE** or if you use a Data Type, Data Class, or Template rule.

6. Optionally click + to add a secondary dataset.
   a. Enter the name of your secondary dataset in the search bar when the **Add Secondary Dataset** dialog appears, then click **Add Secondary Dataset**.

      Important If you edit the rule later, you will need to add the secondary dataset again. Secondary datasets do not persist when a rule is edited. Secondary datasets in the query itself are still referenced when the job is run.

7. Write your SQL query. When you select the Freeform rule type, as shown in the GIF below, the Workbench pre-populates the beginning of the query for you. For example, `SELECT * FROM @<dataset name>`. As you type, the Workbench provides a suggested options with common SQL operators and column names relative to your primary and secondary datasets.



> **Tip**
> - If your SQL query contains columns with special characters, you need to escape them with backticks ` ` to ensure the rule runs successfully. For example, `` `col_à` IS NOT NULL `` and `SELECT * FROM example_table where` `` `col_à` `` `is not null`. This is due to a Spark limitation which causes a syntax parsing error when the DQ Job runs.
> - For a complete list of Spark SQL operators and functions available for use on SQLG and SQLF rule types, see the official Spark documentation.

> **Note** If your SQL query is very lengthy, there is an increased likelihood that it may fail to validate or save. To mitigate this issue, we recommend breaking down such queries into small, more manageable rules. A possible way of achieving this is by leveraging an `IN` clause.

8. Optionally add a filter query in the Filter input field.
9. Click **Validate** to validate your expression.
10. Optionally click **Beautify** to format your SQL query in a more readable layout.
11. Click **Save** to apply your rule to your dataset.

Configuring Rule Details

To configure rule details, click the in the upper right corner of the Workbench to open the Rule Details modal.

The following table shows the optional settings when you set up a rule in the Rule Workbench.

Setting	Description
DQ Dimension	Associate your rule with a DQ Dimension, as shown in the following table. This can be especially useful when you use the Collibra Platform integration and custom reporting.

Dimension	Description
Completeness	Completeness refers to the degree to which potential data contained in a dataset is reflected. Refers to the percentage of columns that have neither EMPTY nor NULL values.
Timeliness	Timeliness refers to the degree to which data is relevant and up-to-date at a certain point in time.
Uniqueness	Uniqueness refers to the degree to which data contains only one record of how an entity is identified. Refers to the cardinality of columns in your dataset.
Validity	Validity refers to the degree to which data conforms to its defining constraints or conditions, which can include data type, range, or format.
Consistency	Consistency refers to the degree to which data contains differing, contradicting, or conflicting entries.
Accuracy	Accuracy refers to the degree to which data correctly reflects its intended values.

> **Note** Tagging rules with custom dimensions in the Metastore is not supported.

Setting	Description
Category	You can create and apply custom categories to your rule to group similar rules together. For example, you can create categories for "Rules for missing data" or "Rules to identify format issues".
Description	Provide a detailed description of your rule.
Purpose	Provide details about Collibra Platform assets associated with your rule in Data Quality & Observability Classic. This allows you to see which assets should relate to your Data Quality Rule asset upon integration.
Preview Limit (Rows)	The number of available rows to preview when you drill into a rule record. The default value is 6 records and the maximum value is 500. **Note** Rule Break Preview is only supported for Freeform and Simple rules.
Run Time Limit (Minutes)	The maximum number of minutes any rule associated with a dataset can take to process before it times out and proceeds to the next activity in the job. If you increase this value to one greater than the previous maximum Run Time Limit of any rule associated with your dataset, it becomes the new maximum timeout limit.
Score	Provides three options, or presets, for calculating the rules score: Low, Medium, and High. Select an option to automatically populate the Points field and Percentage field (if applicable) with predefined values. For more information, go to Understanding the rules score.
Scoring Type	Determines the methodology used to calculate the rules score: Percent or Absolute. For more information, go to Understanding the rules score.
Points	A value between 0-100, representing the number of points deducted from the total quality score when a record violates your rule. For more information, go to Understanding the rules score.
Percent	When you use the Percent scoring type, this can be a value between 0-100. This represents the ratio of the total number of breaking records over the total number of rows. For more information, go to Understanding the rules score.

Setting	Description
Tolerance	Specifies the tolerance threshold as a percentage. For more information, go to Understanding the rules score.

Managing a Livy Session

You can preview the columns in your dataset to help you create SQL queries for rules on Pullup jobs. To do this, ensure that you allocate ample resources for Livy to run efficiently. Livy starts when you click **Run Result Preview** and has 1 core, 1 GB of memory, and 1 worker by default. For large datasets that require increased processing resources, you may need to modify your Livy session to improve the loading time of your Result Preview.

> **Note** When you use Result Preview, there is a 500-row limit on the preview, regardless of the number of rows are in the results of your query.

1. Click **Manage Session**.
   » The Manage Livy Session modal appears.
2. Enter the required information.

Setting	Description
Cores	The number of executor cores for Spark processing. The minimum value is 1.
Memory	The amount of memory allocated for Spark processing. Allocate at least 1 GB of memory per Livy pod.
Workers	The number of executors available in your Livy session. The minimum value is 1. When you increase the number of workers, an additional Livy session is created for each additional worker you specify. For example, if you increase your workers from 1 to 3, then 2 additional Livy sessions are created for a total of 3.

> **Tip** Adjust the Livy resources based on the number of rows included in your query. For columns with a large amount of rows, you may find that you need to increase the memory and workers.

3. Choose whether to acknowledge that all cached preview data will be lost when you update your Livy session.

   > **Important** To update your Livy session, you must select this option.

4. Click **Update**.

> **Note** Livy is not required to use the Result Preview feature for Pushdown.

Working with COUNT and GROUP BY

Be careful when using COUNT and GROUP BY in Freeform SQL and Native SQL rules, because they may create unexpected or unintended results. For best results, keep in mind the following points:

- COUNT. Returns the number of rows that meet the specified criteria. If used on its own, without additional operators and functions, COUNT will create a breaking rule by default.
- GROUP BY. Does not function when used on its own. GROUP BY must be used as part of a group or aggregate function (SUM, COUNT, etc.).

A rule passes when it does not find any rows that meet a specified criteria, and it breaks when it finds rows that need a customer's attention. To create rules with this behavior, include WHERE and/or HAVING in the query. For example, you can use HAVING with both COUNT with GROUP BY to create the following query:

```
SELECT column_A, COUNT(column_A)
FROM table_1
WHERE column_B = value_X
GROUP BY column_A
HAVING COUNT(column_A) > number_Y
```

Filter a data quality rule

The rule filter option on the Rule Workbench allows you to narrow the scope of your SQL query by adding a second query definition to your data quality rule limiting the total number of rows included in the rule score calculation.

Prerequisites

You have:

- ROLE_ADMIN, ROLE_DATASET_RULES, or ROLE_VIEW_DATA
- A user with ROLE_ADMIN has set the **RULE_FILTER** setting to **TRUE** on the Application Configuration Settings page.
- An existing dataset against which you want to write a rule.

Using a filter query

Rule scores are always based on the total number of rows in your dataset, which means irrelevant records are often included in your rule. When these extraneous records are not filtered out of your SQL query, you may encounter unintended skew to the results of your rule score. By applying a filter query, you can limit the total number of rows that are considered when Data Quality & Observability Classic calculates the rule score, improving the relevance of the rule results.

Using a filter can also eliminate the need to create additional duplicate datasets for each rule that requires a different filter. In addition to reducing the amount of time it takes to set up these synthetic filters, the usage of filter queries on the Rule Workbench reduce the cost and computational waste from duplicate profiling.

By defining the SQL for the subset of relevant records (filter) and the invalid conditions (rule) independently, you can increase the visibility on the ratio between them.

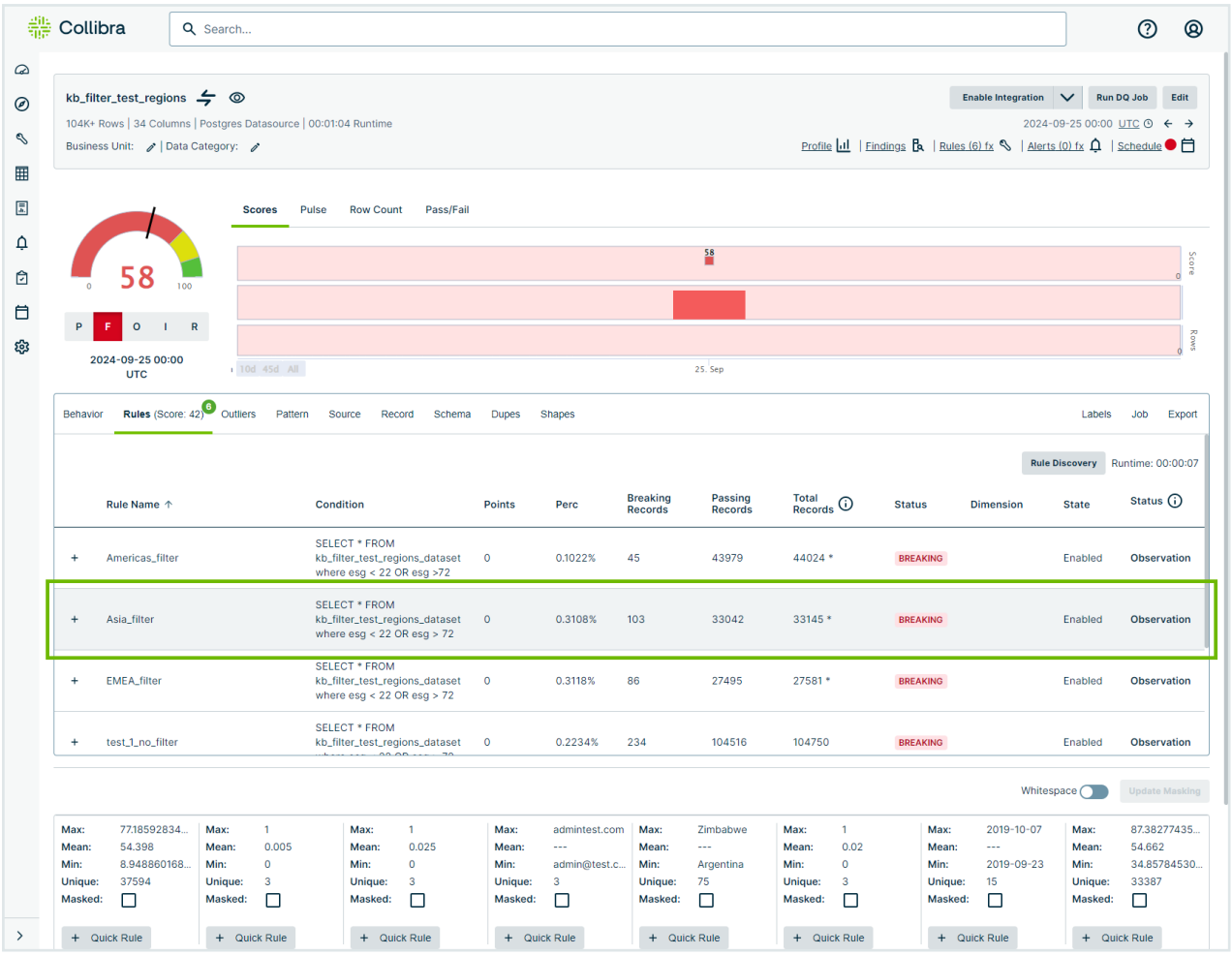
Data steward use case

Suppose you are a data steward whose organization maintains a table containing massive amounts of data about its operations in different regions of the world. Each region has a data steward responsible for monitoring the data therein, and your region is Asia.

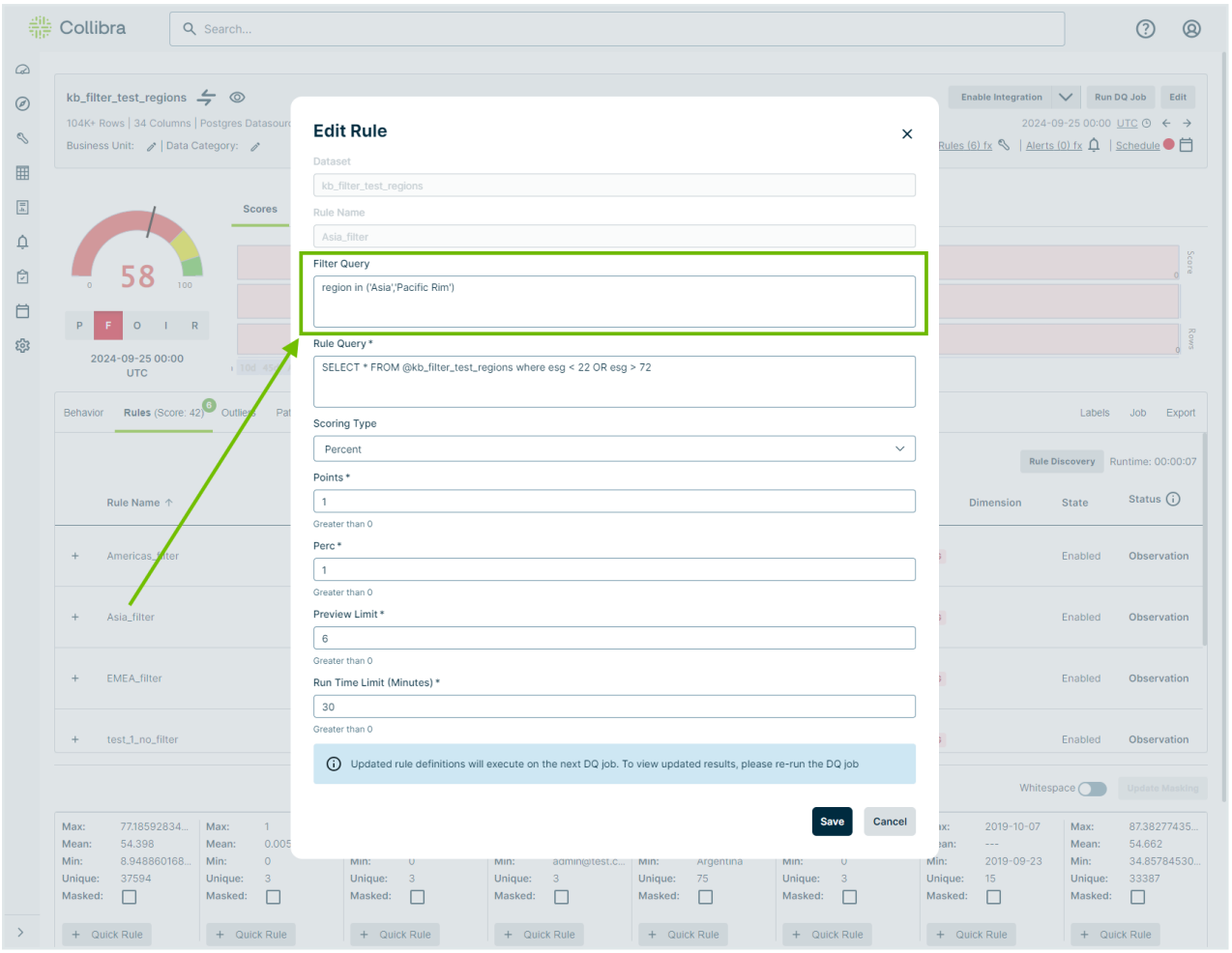
Previously, to isolate the most relevant data, a data steward would need to create a new dataset containing only the data of a given region. While this technically works, it is not an optimal approach to separating data, because it introduces unnecessary operational cost and can be more cumbersome to maintain in the long term.

Now, you can rely on a single dataset as the source of truth, where, instead of creating new datasets with rules set up for each region, you can create filter queries on top of the same rule query to only include results for the Americas, Asia, and so on.

In the below screenshot, several rules are set up to query the same dataset. The first 3 rules have filter queries to filter the exact data relevant to each data steward's region. An asterisk displays to the right of the value in the Total Records column to denote that the rule in that row has a filter query.



In the below example, only data for your region, Asia (which includes the Pacific Rim), is being queried with the addition of the filter query, `region in ('Asia','Pacific Rim')`.



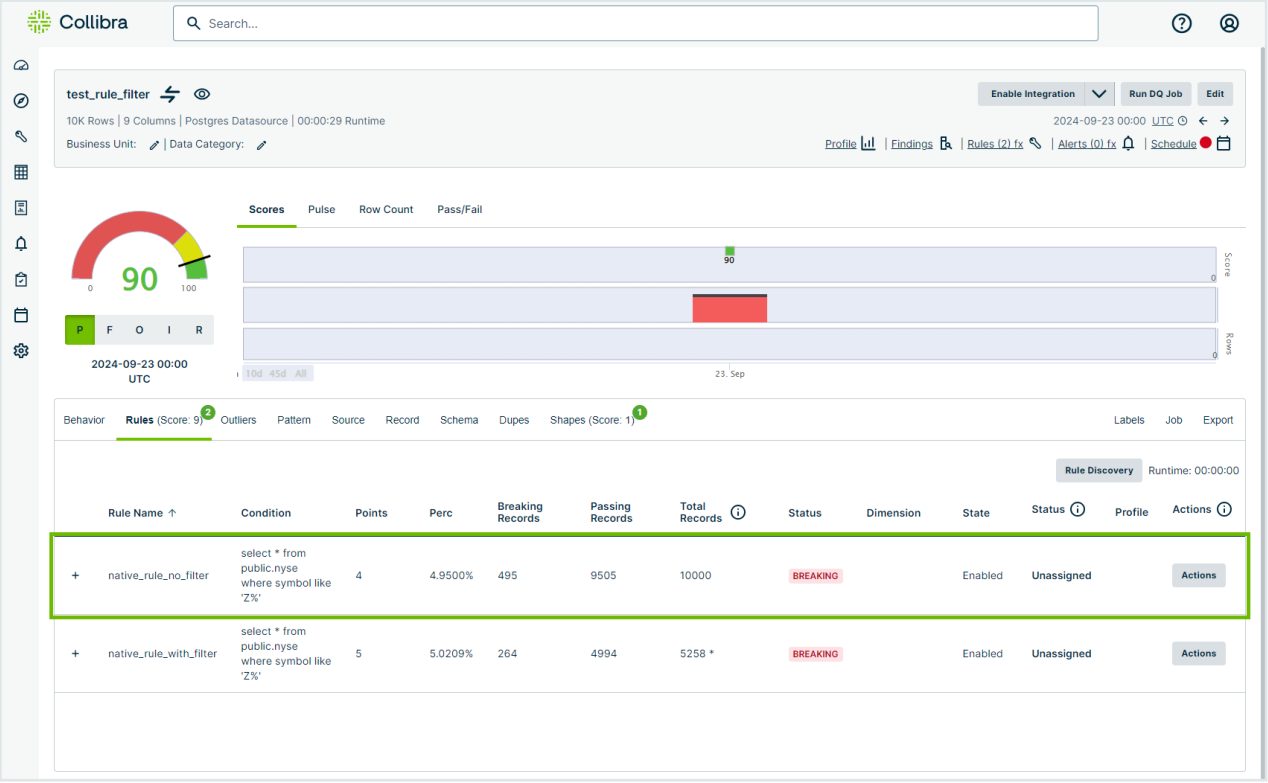
After the results of freeform or simple rules with filter queries separating each one into regions are available on the Findings page, you can view and export their break records when a column in the dataset is assigned as the link ID column. From the ability to pare down the results of your rules with a filter query to the exportation of their break records, you now have more complete end-to-end control over the management of your data region.

Filter query scoring impact

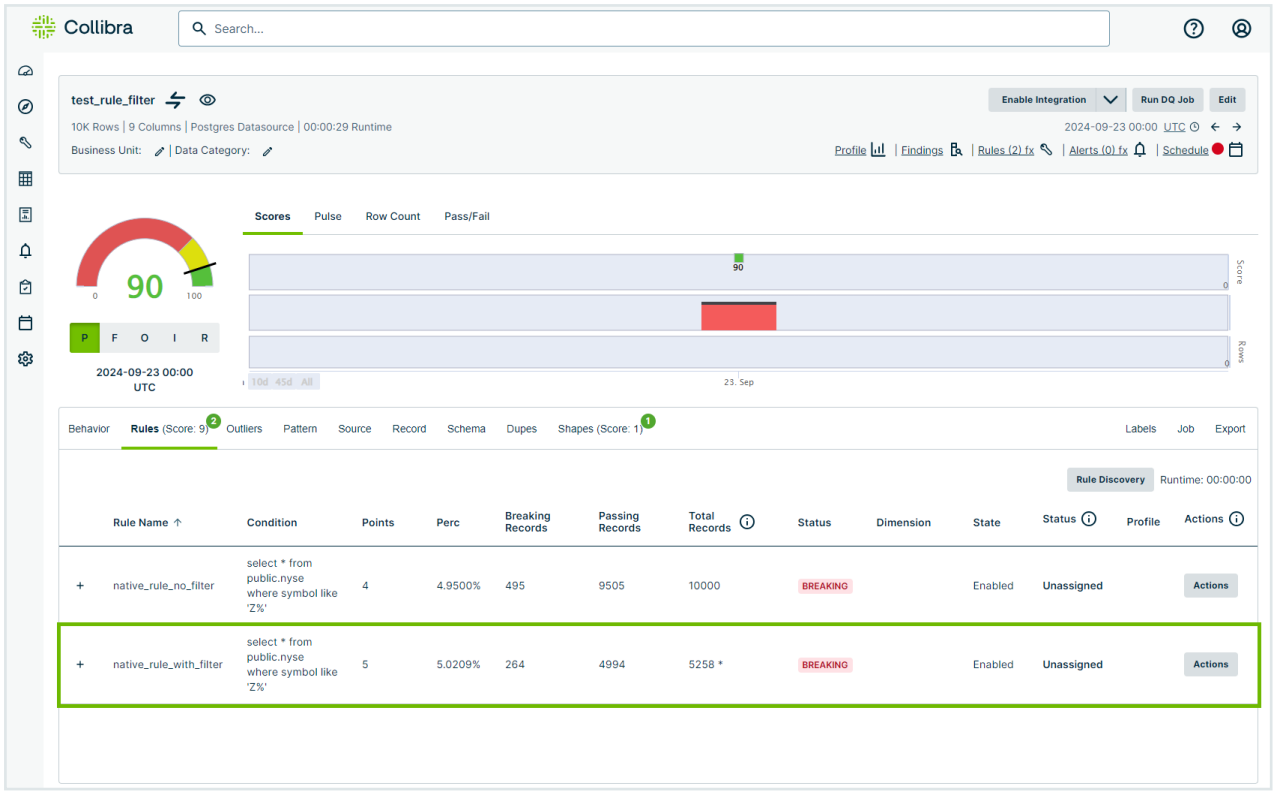
The inclusion or exclusion of a filter query impacts the rule score on the Findings page by changing the denominator to reflect the number of filtered or unfiltered rows.

In the following example based on a New York Stock Exchange dataset, the rule "native_rule_filter" uses the rule query `select * from public.nyse where symbol like 'Z%'` to examine all of its 10,000 rows for records in the "symbol" column beginning with the letter "Z."

Without a filter query this returns 495 breaking records when the DQ Job runs. 1 point is deducted for each percentage point, which is the total number of breaking records (495) over the total number of rows (10,000). In this case, because the rule queries the entire dataset, the percentage of rule breaks is 4.95% for a total of 4 points deducted from the overall data quality score.



The same New York Stock Exchange dataset uses an identical rule query in the rule "native_rule_with_filter" but applies the filter query `low > 25` on top of it to exclude all records that do not also contain values greater than 25 in the "low" column. This results in 5,258 total records and 264 breaking records for 5.02% breaking and 5 points deducted from the overall data quality score.


Collibra
Search...
test_rule_filter
10K Rows | 9 Columns | Postgres Datasource | 00:00:29 Runtime
Business Unit: | Data Category:
Enable Integration
Run DQ Job
Edit
2024-09-23 00:00 UTC
Profile | Findings | Rules (2) fx | Alerts (0) fx | Schedule
90
0
100
P F O I R
2024-09-23 00:00 UTC
Scores
Pulse
Row Count
Pass/Fail
Score
Rows
23. Sep
Behavior
Rules (Score: 9)
Outliers
Pattern
Source
Record
Schema
Dupes
Shapes (Score: 1)
Labels
Job
Export
Rule Discovery
Runtime: 00:00:00
Rule Name
Condition
Points
Perc
Breaking Records
Passing Records
Total Records
Status
Dimension
State
Status
Profile
Actions
native_rule_no_filter
select * from public.nyse where symbol like 'Z%'
4
4.9500%
495
9505
10000
BREAKING
Enabled
Unassigned
Actions
native_rule_with_filter
select * from public.nyse where symbol like 'Z%'
5
5.0209%
264
4994
5258 *
BREAKING
Enabled
Unassigned
Actions

Adding a rule filter

1. Click the Rules link on the metadata bar on a dataset-level page or open the Dataset Rules page of a dataset by clicking the in the sidebar menu, then clicking **Rule Builder**.
   » The Dataset Rules page opens.
2. Follow the steps to add a rule from the Rule Workbench.
   Show Rule Workbench steps[1]
3. In the **Filter (optional)** input field, enter a filter query in the format of a simple query, such as `symbol is not null` or `low > 25`.

   > **Tip**
   > - The filter query should only contain conditions and predicates found after the `where` clause in a SQL query. For example, the correct syntax of a filter query for results in the "low" column of a NYSE dataset might look like `low > 25`, never `select * from public.nyse where low > 25`.
   > - The most relevant results return when the rule query is based solely on data in the source query.

   Show screenshot[2]
4. Click **Save** to apply your rule to your dataset.
5. Rerun your DQ Job.
   » The results of your run are available on the Rules tab of the Findings page.

1
There are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench page opens with the preferences of your existing rule.
   b. Click **Add Rule** in the upper right corner to create a new rule.
      » The Rule Workbench page opens.

Select a rule type from the Type dropdown menu.
Enter a name for your rule in the **Rule Name** input field.

> **Important** Rule names cannot contain spaces.

Select a primary column from the Primary Col dropdown menu.

> **Note** Primary Column is only required when you use a Data Type, Data Class, or Template rule.

Optionally click + to add a secondary dataset.
   a. Enter the name of your secondary dataset in the search bar when the **Add Secondary Dataset** dialog appears, then click **Add Secondary Dataset**.

Enter your SQL query in the **Rule (required)** input field.

> **Note** Rules that contain a filter query are marked with an asterisk to the right of the Total Records value. For example, 5258*.

Limitations

- You cannot use @t1 rules in the filter query.
- SQL Assistant for Data Quality is not available for the filter query.
- When the rule refers to a dataset that is different from the origination dataset, the rule results reflect the rule filter query combination from the dataset referenced in the filter query, not the origination dataset.
- The filter query does not support LIMIT clauses.

Apply a tolerance threshold to a data quality rule

The rule tolerance option on the Rule Workbench allows you to define the percentage of rule breaks that must be found to affect rule scores and statuses, and for any condition alerts tied to the rule to send.

By setting a rule tolerance threshold, you can reduce the number of "breaking" rule results displayed on the Findings page. This helps minimize the number of alert notifications that are sent due to less significant breaking rules. As a result, you can focus on the most critical data quality problems while reducing irrelevant alerts, making it easier to identify actual data quality issues and keeping your inbox or notification feed less cluttered.

Prerequisites

You have:

- ROLE_ADMIN, ROLE_DATASET_RULES, or ROLE_VIEW_DATA
- An existing dataset against which you want to write a rule.

Adding a rule tolerance threshold

1. Click the Rules link on the metadata bar on a dataset-level page or open the Dataset Rules page of a dataset by clicking the in the sidebar menu, then clicking **Rule Builder**.
   » The Dataset Rules page opens.
2. Follow the steps to add a rule from the Rule Workbench.
   Show Rule Workbench steps[1]
3. Click in the upper right corner of the Rule Workbench.
   » The Rule Details dialog appears.
4. In the **Tolerance** input field, enter or select a value between 0 and 100.

   > **Note** The value 0 is the same as not adding a rule tolerance threshold at all. In other words, any percentage of rule breaks will be considered breaking.

   Show screenshot[2]
5. Click **Submit**.
6. Click **Save** to apply the rule to your dataset.
7. Rerun your DQ Job.
   » The results of your run are available on the Rules tab of the Findings page.

1

There are two ways to open the Rule Workbench.

a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
   » The Rule Workbench page opens with the preferences of your existing rule.
b. Click **Add Rule** in the upper right corner to create a new rule.
   » The Rule Workbench page opens.

Select a rule type from the Type dropdown menu.
Enter a name for your rule in the **Rule Name** input field.

> **Important** Rule names cannot contain spaces.

Select a primary column from the Primary Col dropdown menu.

> **Note** Primary Column is only required when you use a Data Type, Data Class, or Template rule.

Optionally click + to add a secondary dataset.

a. Enter the name of your secondary dataset in the search bar when the **Add Secondary Dataset** dialog appears, then click **Add Secondary Dataset**.

Enter your SQL query in the **Rule (required)** input field.

> **Note** Rules that contain a tolerance threshold are marked with an asterisk to the right of the Points value. For example, 0*.

Viewing rule tolerance results

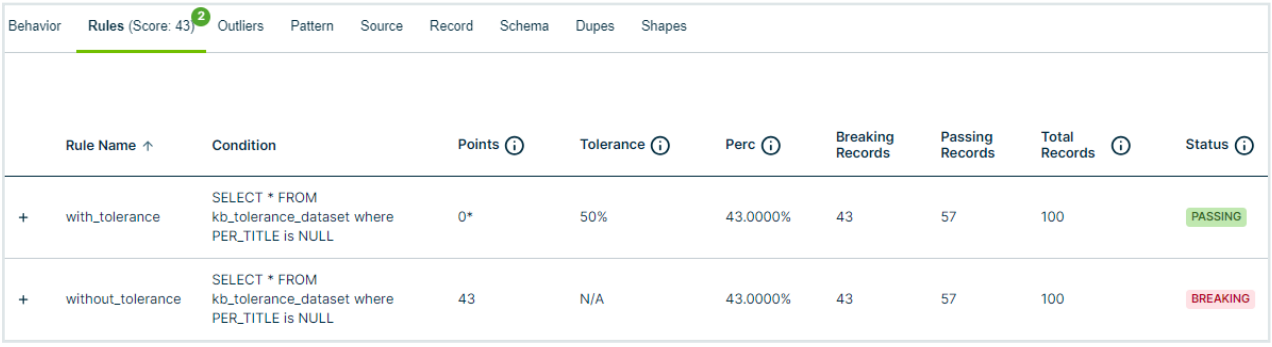
When you run a dataset with rules that include tolerance thresholds, only the rules with percentages of breaking records greater than their tolerance thresholds affect the data quality score on the Findings page. When a rule contains a percentage of breaking records under its tolerance threshold, an asterisk displays next to a value of 0 in the Points column of the Rules tab.

To illuminate the scoring and alerting impact of various rule thresholds, consider a scenario where there are two identical rules that query a total of 100 records and a tolerance threshold of 50 is set for both rules.

In the first rule, 75 records break the conditions of the rule, meaning 75% of the total records are breaking. In this case, even with a tolerance threshold set, 75 points are deducted from the overall data quality score, because 75% exceeds the tolerance threshold of 50%. This rule is considered "Breaking" and, if configured, an alert will be sent.

However, in the second rule, only 43% of all records queried break its conditions. Because 43% is below the tolerance threshold of 50%, the rule is considered "Passing," no points are deducted from the data quality score, and alerts associated with the rule are suppressed.

The screenshot and table below further illustrate a different scenario where two identical rules query a column with and without a rule tolerance threshold applied.



	Rule Name	Condition	Points	Tolerance	Perc	Breaking Records	Passing Records	Total Records	Status
+	with_tolerance	SELECT * FROM kb_tolerance_dataset where PER_TITLE is NULL	0*	50%	43.0000%	43	57	100	PASSING
+	without_tolerance	SELECT * FROM kb_tolerance_dataset where PER_TITLE is NULL	43	N/A	43.0000%	43	57	100	BREAKING

The following table provides an explanation of the values present in each relevant column in the above screenshot of the Rules tab on the Findings page.

	Points	Tolerance	Perc	Breaking, Passing, and Total Records	Status
With Tolerance	Because a percentage of 43 is under the tolerance threshold of 50, no points (0*) are being deducted from the data quality score. **Note** Rule scores that are impacted by tolerance are marked with an asterisk to the right of the Points value. For example, 0*.	A tolerance threshold of 50% means that any percentage of breaking records less than or equal to it will be ignored in the score calculation. In this case, 43% of the records are breaking, but they do not impact the scoring calculation because of the rule threshold of 50%.	The values in these columns remain unchanged with or without a tolerance value, because the actual percentages need to be available for informational purposes.		The status is noted as Passing because the percentage of breaking records (43%) is under the tolerance threshold of 50%.

	Points	Tolerance	Perc	Breaking, Passing, and Total Records	Status

Note In some rare cases when the tolerance threshold and percentage breaking appear to be identical on the Findings page, a status can be noted as Breaking.

However, because the Perc column only shows four decimal places, its precise value might not display in full, resulting in a percentage that, in fact, exceeds the tolerance threshold. You can hover your cursor over the value in the Perc column or reference the rule_output table in the Metastore to see its exact value.

	Points	Tolerance	Perc	Breaking, Passing, and Total Records	Status
Without Tolerance	Because 43 percent of records are breaking and a tolerance threshold is not set, 43 points are being deducted from the data quality score.	A tolerance threshold is not being imposed on the rule.			Since a rule tolerance threshold is not set, the rule is considered breaking because the percentage of breaking records is greater than 0.

Create a data quality rule with SQL Assistant for Data Quality

SQL Assistant for Data Quality uses Collibra AI to automatically write and troubleshoot SQL queries. When working on a Pullup dataset, the type of SQL that SQL Assistant for Data Quality generates depends on the rule type. For instance, if your rule type is Freeform SQL, the SQL generated by SQL Assistant for Data Quality is Spark SQL, whereas a Native SQL rule returns the SQL specific to the datasource. Likewise, Pushdown datasets always return SQL native to the job source.

The Prompts are organized in two sections: **Basic** and **Advanced**.

Basic options for SQL query suggestions provided by SQL Assistant for Data Quality include:

Option	Description
Generate Rule	Collibra AI suggests a Spark SQL query to check where in your table, column, or view your input criteria are met.
Troubleshoot Rule	When a SQL syntax error occurs, you can use SQL Assistant for Data Quality to automatically troubleshoot and fix the error.

Advanced options for SQL query suggestions provided by SQL Assistant for Data Quality include:

Option	Description
Categorical	Collibra AI suggests a SQL query to detect categorical outliers.
Dupe	Collibra AI suggests a SQL query to detect duplicate values.
Record	Collibra AI suggests a SQL query to check for values that appear on a previous day but not for the next day.
Pattern	Collibra AI suggests a SQL query to check for infrequent combinations that appear less than 5 percent of the time in the columns you specify.
Frequency Distribution	Collibra AI suggests a SQL query to check for the frequency distribution of all values within a column.

Requirements and permissions

The architecture of SQL Assistant for Data Quality differs depending on whether you use the platform path with a Collibra Platform integration or the preview path with a Collibra-provided encrypted service account key.

To use SQL Assistant for Data Quality **with** a Collibra Platform integration, you need to:

- Have an active integration with Collibra Platform configured on the Admin Console > Integrations screen.

  > **Note** Only the Credentials step of the Integration Setup is required. You do not need to map connections, tenants, dimensions, or layers to use SQL Assistant for Data Quality.

- Set AI_PLATFORM_PATH to `TRUE` (default) on the Admin Console > Application Configuration Settings screen.
- Set AI_TENANT to `TRUE` on the Admin Console > Application Configuration Settings screen.
- Update the `gai_proxy_endpoint` in the owl-env.sh or Helm Chart, depending on your installation type.
  Steps to update the proxy endpoint

> **Important**
> You may need to safelist the proxy endpoint if the security settings of your installation do not allow outbound internet requests.
>
> While this is unlikely if you have an active integration with Collibra Platform set up, you may also need to safelist your Collibra Platform URL to allow outbound requests to it.

Steps for Spark Standalone and Hadoop installations

1. Export the environment variable `export gai_proxy_endpoint=https://kong-prod-gcp-ue1-1.collibra.com` in the **owl-env.sh** file.

   > **Note** The default value for the gai_proxy_endpoint is https://kong-prod-gcp-ue1-1.collibra.com. To change this value to a different cloud region, contact Collibra Support.

2. Save and close the owl-env.sh file.
3. Restart the DQ web application and DQ agent services.

Steps for Kubernetes (cloud native) installations

1. Set the following value in the Helm Chart provided by Collibra.

   ```
   --set gai_proxy_endpoint: "https://kong-prod-gcp-ue1-1.collibra.com"
   ```

   > **Note** The default value for the gai_proxy_endpoint is https://kong-prod-gcp-ue1-1.collibra.com. To change this value to a different cloud region, contact Collibra Support.

2. If you are updating an existing deployment of Data Quality & Observability Classic, restart the DQ web and agent pods to complete the update.

> **Important** On April 9, 2025, Google will deprecate the Vertex text-bison AI model, which the preview path option uses. To continue using SQL Assistant for Data Quality, you must switch to the platform path, which requires an integration with Collibra Platform.

When using SQL Assistant for Data Quality **without** a Collibra Platform integration, you have:

- AI_PLATFORM_PATH set to `FALSE` on the Admin Console > Application Configuration Settings screen.
- AI_TENANT set to `TRUE` on the Admin Console > Application Configuration Settings screen.
- An encrypted service account key provided by Collibra.

Steps

Basic options

1. Open the Dataset Rules page of your dataset.
2. From the Dataset Rules page, there are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench opens with the preferences of your existing rule.
   b. Click **Add Rule** in the upper right corner to create a new rule.
      » The Rule Workbench opens.
3. In the upper right corner of the Rule Workbench SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
4. From the Prompt dropdown menu, select Generate Rule.

   > **Note** Generate Rule is the default option.

5. In the query input field, enter a plain language prompt.

> **Tip** When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

Examples

- "high is more than 50"
- "find all of the unique values in Column1 and show how many records exist for each"
- "state is not Michigan, California, or Illinois"
- "compare the 2 dates 2018-01-13 to 2018-01-14 - write 2 separate queries for each date, using trade date - write as a common table expression for tables a and b - join using symbol column - use a join to identify which values are in 2018-01-14 that are not in 2018-01-13 - the goal is to find values that are missing from the previous day"

6. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.
7. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
   b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
   c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL rule suggestion.

When a SQL syntax error occurs, you can use SQL Assistant for Data Quality to automatically troubleshoot and fix the error.

1. Open the Dataset Rules page of your dataset.
2. From the Dataset Rules page, there are two ways to open the Rule Workbench.
    a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
        » The Rule Workbench opens with the preferences of your existing rule.
    b. Click **Add Rule** in the upper right corner to create a new rule.
        » The Rule Workbench opens.
3. In the upper right corner of the Rule Workbench SQL editor, click **Collibra AI**.
    » The Collibra AI tool appears.
4. In the SQL editor, enter an invalid SQL query that returns a syntax error.

    > **Example** `SELECT * fomr public.nyse2 were onpe > 4.75`

5. In the upper right corner of the SQL editor, click **Collibra AI**.
    » The SQL Assistant for Data Quality tool appears.
6. From the Prompt dropdown menu, select **Troubleshoot Rule**.
    » The SQL Assistant for Data Quality tool shows the prompt to rewrite and fix your SQL query, an overview of the exception message, the SQL query that caused the error, the table and column names in your query, and helpful troubleshooting tips.

    > **Note** You can optionally click the ✎ above and to the right of the overview text to manually edit the troubleshooting prompt.

7. Click **Submit to Collibra AI**.
    » SQL Assistant for Data Quality troubleshoots your SQL query and displays the suggested query when it completes.
8. Choose one of the following options:
    a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
    b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
    c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
    d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL rule suggestion.

Advanced options

1. Open the Dataset Rules page of your dataset.
2. From the Dataset Rules page, there are two ways to open the Rule Workbench.
    a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
        » The Rule Workbench opens with the preferences of your existing rule.
    b. Click **Add Rule** in the upper right corner to create a new rule.
        » The Rule Workbench opens.
3. In the upper right corner of the Rule Workbench SQL editor, click **Collibra AI**.
    » The SQL Assistant for Data Quality tool appears.
4. From the Prompt dropdown menu, select **Categorical**.
    » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
5. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

    Example In this example, the default placeholder column "sale_state" will be replaced with the user-defined "cm_claimant_age" in the categorical outlier SQL suggestion.

    ```
    Replace the column sale_state with the columns named:
    cm_claimant_age
    ```

    Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

6. Click **Submit to Collibra AI**.
    » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

Example

```
SELECT cm_claimant_age, COUNT(*) AS count, (COUNT(*) / (SELECT COUNT(*) FROM defau
lt.aclaims_master)) * 100 AS percentage FROM default.aclaims_master GROUP BY cm_cl
aimant_age HAVING (COUNT(*) / (SELECT COUNT(*) FROM default.aclaims_master)) * 100
< 5 ORDER BY percentage DESC
```

7. Choose one of the following options:
    a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
    b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
    c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
    d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. Open the Dataset Rules page of your dataset.
2. From the Dataset Rules page, there are two ways to open the Rule Workbench.
    a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
        » The Rule Workbench opens with the preferences of your existing rule.
    b. Click **Add Rule** in the upper right corner to create a new rule.
        » The Rule Workbench opens.
3. In the upper right corner of the Rule Workbench SQL editor, click **Collibra AI**.
    » The SQL Assistant for Data Quality tool appears.
4. From the Prompt dropdown menu, select **Dupe**.
    » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
5. In the query input field, enter the name of the column in your dataset to include in the SQL query.

Example In this example, the default placeholder column "email" will be replaced with the user-defined "cm_denied_reason1" in the duplicate value SQL suggestion.

```
Replace the column email with the column named:
cm_denied_reason1
```

Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column name.

6. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

   Example

   ```
   SELECT cm_denied_reason1, COUNT(*) AS count, (COUNT(*) / (SELECT COUNT(*) FROM def
   ault.aclaims_master)) * 100 AS percentage FROM default.aclaims_master GROUP BY cm_
   denied_reason1 HAVING count > 1 ORDER BY count DESC
   ```

7. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
   b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
   c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. Open the Dataset Rules page of your dataset.
2. From the Dataset Rules page, there are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench opens with the preferences of your existing rule.

b. Click **Add Rule** in the upper right corner to create a new rule.
   » The Rule Workbench opens.

3. In the upper right corner of the Rule Workbench SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
4. From the Prompt dropdown menu, select **Record**.
   » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
5. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

> **Example** In this example, the default placeholder columns "vendor_type" and "transaction_date" will be replaced with the user-defined "cm_claim_status" and "cm_bdos" in the SQL suggestion for record changes.
>
> ```
> Replace the column vendor_type and the date column transaction_date with the
> columns named
> cm_claim_status and cm_bdos
> ```

> **Important** You must specify one column with numeric data and one date column.

> **Tip** When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

6. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

Example

```
SELECT cm_claim_status from default.aclaims_master where cm_bdos = DATE'2022-01-0
7' EXCEPT SELECT cm_claim_status from default.aclaims_master where cm_bdos = DAT
E'2022-01-07' - INTERVAL '1' DAY
```

7. Choose one of the following options:
    a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
    b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
    c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
    d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. Open the Dataset Rules page of your dataset.
2. From the Dataset Rules page, there are two ways to open the Rule Workbench.
    a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
        » The Rule Workbench opens with the preferences of your existing rule.
    b. Click **Add Rule** in the upper right corner to create a new rule.
        » The Rule Workbench opens.
3. In the upper right corner of the Rule Workbench SQL editor, click **Collibra AI**.
    » The SQL Assistant for Data Quality tool appears.
4. From the Prompt dropdown menu, select **Pattern**.
    » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
5. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

Example In this example, the default placeholder columns "sale_state" and "state_tax" will be replaced with the user-defined "cm_diagnosis_code" and "cm_denied_reason1" in the SQL suggestion for pattern detection.

```
Replace the sale_state and
state_tax columns with these 2 columns:
cm_diagnosis_code and cm_denied_reason1
```

Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column names.

6. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

Example

```
SELECT cm_diagnosis_code, cm_denied_reason1, COUNT(*) AS COUNT, ROUND( 100.0 * COU
NT(*) / ( SELECT COUNT(*) FROM default.aclaims_master ), 2 ) AS percentage FROM de
fault.aclaims_master GROUP BY cm_diagnosis_code, cm_denied_reason1 HAVING percenta
ge < 5
```

7. Choose one of the following options:
   a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
   b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
   c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

1. Open the Dataset Rules page of your dataset.
2. From the Dataset Rules page, there are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench opens with the preferences of your existing rule.
   b. Click **Add Rule** in the upper right corner to create a new rule.
      » The Rule Workbench opens.
3. In the upper right corner of the Rule Workbench SQL editor, click **Collibra AI**.
   » The SQL Assistant for Data Quality tool appears.
4. From the Prompt dropdown menu, select **Frequency Distribution**.
   » The prompt automatically uses the correct SQL type of your rule. For instance, a Freeform rule utilizes Spark SQL, while data source-specific SQL adapts based on the connection type of your dataset.
5. In the query input field, enter the name of the column in your dataset to include in the SQL query.

   Example In this example, a SQL query for frequency distribution will be suggested for the column "cm_denied_reason1".

   ```
   Find the frequency distribution for the column named:
   cm_denied_reason1
   ```

   Tip When there are many columns in your dataset, you may need to scroll down to the bottom of the input field to enter your column name.

6. Click **Submit to Collibra AI**.
   » SQL Assistant for Data Quality generates your SQL query and displays the suggested query when it completes.

   Example

   ```
   SELECT cm_denied_reason1, COUNT(*) FROM default.aclaims_master GROUP BY cm_denied_
   reason1
   ```

7. Choose one of the following options:
    a. Click **Copy to Editor** to automatically apply the suggested query to the SQL editor on the Dataset Overview modal.
    b. Click **Copy to Clipboard** to copy the suggested query to your computer's clipboard, allowing you to paste the query elsewhere.
    c. Click **Close** to exit the SQL Assistant for Data Quality tool without copying the suggested query.
    d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

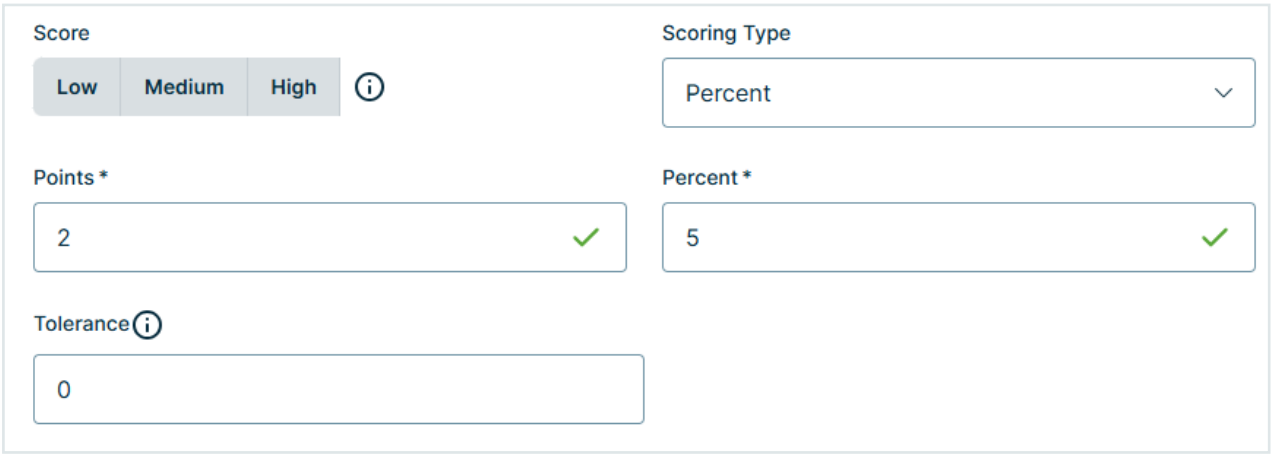
Understanding the rules score

The Rules tab on the Findings page displays a rules score, which helps indicate the severity of the rule breaks. You can configure the methodology used to calculate the rules score on the Rule Details dialog.

> **Note** The rules score is entirely separate from the data quality score. For more information on the data quality score, go to Data Quality Score Meter.

Rules settings

The Rule Details dialog provides the following options for calculating the rules score.



Score

The Score widget provides three options, or presets, for calculating the rules score. Select an option to automatically populate the Points field and Percentage field (if applicable) with predefined values. You may modify the values as needed.

Low	Downscores 1 point for every 1% of records that violate a rule. For example, if 25 of 100 records break a rule, 25 points are deducted from the rules score.
Medium	Downscores 5 points for every 1% of records that violate a rule. For example, if 2 of 100 records break a rule, then 10 points are deducted from the rules score.
High	Downscores 25 points for every 1% of records that violate a rule. For example, if 3 of 100 records break a rule, 75 points are deducted from the rules score.

Scoring type

The Scoring Type field determines the methodology used to calculate the rules score.

Percent	Deducts points based on the percentage of records that violate the rule. Use the **Points** and **Percent** fields to set the percentage threshold. For example, suppose you have 100 records, and the **Points** and **Percent** fields are both set to 1. If 25 of the records break your rule (which is 25%), then a total of 25 points are deducted from your rules score.
Absolute	Deducts a fixed number of points when one or more rules break, regardless of the percentage of records that break the rule. For example, if you set the value of the **Points** field to 10 and a rule breaks, a total of 10 points are deducted from your rules score. The Absolute option is useful when you want to know if any rule breaks occur in the findings. Note that if more than one rule breaks, the point deduction is the same as if only one rule breaks.

Points

Used with both the Percent and Absolute scoring types. Set Points to a value between 0 and 100, which is the number of points deducted from the rules score when a record violates a rule.

Percent

Used with the Percent scoring type. Set Percent to a value between 0 and 100, which is the ratio of the number of breaking records over the total number of records.

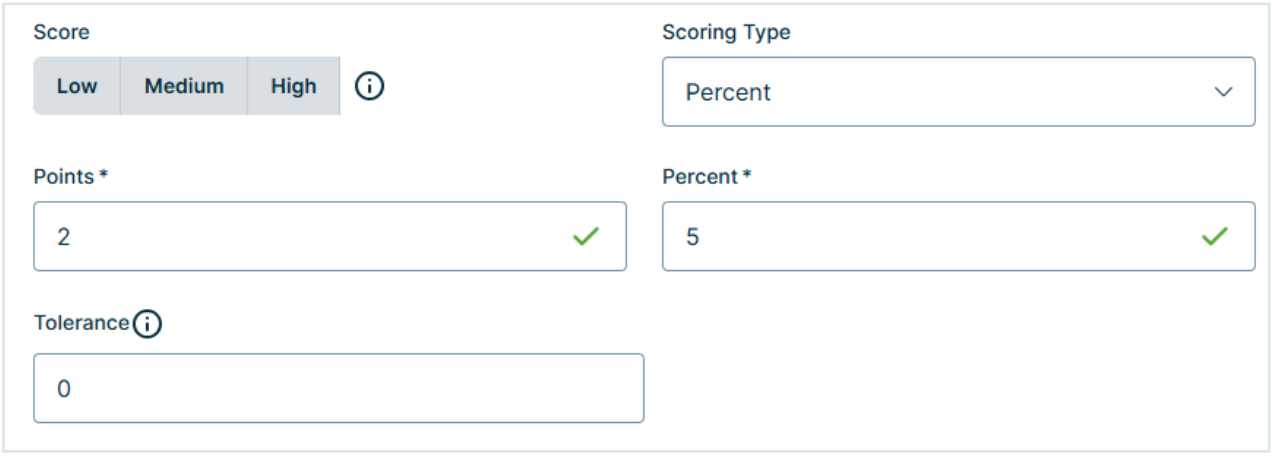
Tolerance

Specifies the tolerance threshold as a percentage. Enter a percentage that represents the acceptable amount of rule breaks. For example, if the Tolerance is set to 10%, and the percentage of rule breaks is less than or equal to 10%, such as 5%, the rule status is passing and the score is unaffected. If the percentage of rule breaks is greater than 10%, such as 15%, the rules score is calculated based on the settings provided.

Example of the Percent scoring method

In this example, you have 100 records and you would like to deduct 2 points for every 5% of records that break the rule. Set the following parameters:

- Scoring Type: Percent
- Points: 2
- Percent: 5



Example results are:

- If 3 records break the rule, 0 points are deducted from the rules score.
  - 3% of the records break (3/100)
  - 3% is less than or equal to the 5% threshold.
- If 15 records break the rule, 6 points are deducted from the rules score.
  - 15% of the records break (15/100)
  - 2 points are deducted per 5% of rule breaks (2 points * (15% / 5%) = 6).
- If 25 records break the rule, 10 points are deducted from the rules score.
  - 25% of the records break (25/100)
  - 2 points are deducted per 5% of rule breaks (2 points * (25% / 5%) = 10).



Behavior | Rules (Score: 16) | Outliers | Pattern | Source | Record | Schema | Dupes | Shapes | Labels | Job | Export

Rule Discovery Runtime: 00:00:03

	Rule Name	Rule Query	Points	Tolerance	Perc	Breaking Records	Passing Records	Total Records	Status	Dimension	State	Status	Profile	Actions
+	limit15	SELECT * FROM berka_accounts_17_dataset limit 15	6	N/A	15.0000%	15	85	100	BREAKING		Enabled	Unassigned		Actions
+	limit25	SELECT * FROM berka_accounts_17_dataset limit 25	10	N/A	25%	25	75	100	BREAKING		Enabled	Unassigned		Actions
+	limit3	SELECT * FROM berka_accounts_17_dataset limit 3	0	N/A	3.0000%	3	97	100	BREAKING		Enabled	Unassigned		Actions

Now let's change the value in the **Tolerance** field from 0 to 15%, indicating the acceptable amount of rule breaks. The results are:

- If 3 records break the rule, it is below the Tolerance of 15%. 0 points are deducted from the rules score. An asterisk (**0***) indicates that rule breaks occurred, but they did not affect the rule score.
- If 15 records break the rule, it is equal to the Tolerance of 15%. 6 points are deducted from the rules score.
- If 25 records break the rule, it is above the Tolerance of 15%. 10 points are deducted from the rules score.



Behavior | Rules (Score: 16) | Outliers | Pattern | Source | Record | Schema | Dupes | Shapes | Labels | Job | Export

Rule Discovery Runtime: 00:00:03

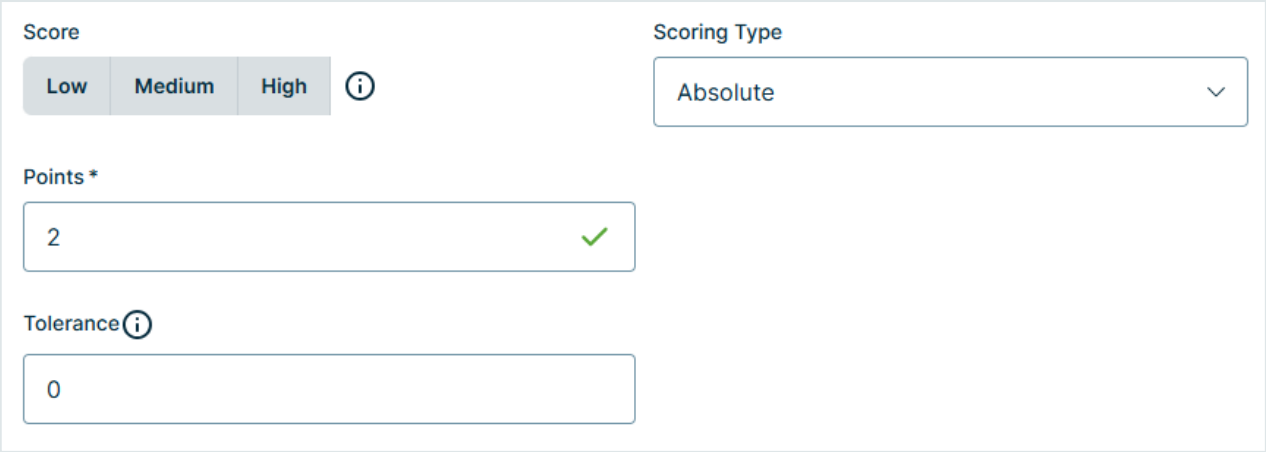
	Rule Name	Rule Query	Points	Tolerance	Perc	Breaking Records	Passing Records	Total Records	Status	Dimension	State	Status	Profile	Actions
+	limit15	SELECT * FROM berka_accounts_17_dataset limit 15	6	15%	15.0000%	15	85	100	BREAKING		Enabled	Unassigned		Actions
+	limit25	SELECT * FROM berka_accounts_17_dataset limit 25	10	15%	25%	25	75	100	BREAKING		Enabled	Unassigned		Actions
+	limit3	SELECT * FROM berka_accounts_17_dataset limit 3	0*	15%	3.0000%	3	97	100	PASSING		Enabled	Unassigned		Actions

> **Note** If the number of rule breaks is above the Percent value but below the Tolerance value, 0 points are deducted from the rules score. The asterisk (**0***) indicates that rule breaks occurred, but they did not affect the rule score.

Example of the Absolute scoring method

In this example, you have 100 records and you would like to deduct 2 points whenever a rule is broken, regardless of the number of records breaking the rule. Set the following parameters:

- Scoring Type: Absolute
- Points: 2



Example results are:

- If 3 records break the rule, 2 points are deducted from the rules score.
- If 15 records break the rule, 2 points are deducted from the rules score.
- If 25 records break the rule, 2 points are deducted from the rules score.



The Absolute method ensures that a consistent penalty is applied each time the rule is violated, regardless of the extent of the violation.

Now let's change the value in the **Tolerance** field from 0 to 15%, indicating the acceptable amount of rule breaks. The results are:

- If 3 records break the rule, it is below the Tolerance of 15%. 0 points are deducted from the rules score. An asterisk (**0***) indicates that rule breaks occurred, but they did not affect the rule score.
- If 15 records break the rule, it is equal to the Tolerance of 15%. 2 points are deducted from the rules score.
- If 25 records break the rule, it is above the Tolerance of 15%. 2 points are deducted from the rules score.



Behavior | Rules (Score: 4) | Outliers | Pattern | Source | Record | Schema | Dupes | Shapes | Labels | Job | Export

Rule Discovery Runtime: 00:00:02

	Rule Name	Rule Query	Points	Tolerance	Perc	Breaking Records	Passing Records	Total Records	Status	Dimension	State	Status	Profile	Actions
+	Limit15	SELECT * FROM temp_1743780826752_berka_accounts_17_dataset limit 15	2	15%	15.0000%	15	85	100	BREAKING		Enabled	Unassigned		Actions
+	Limit25	SELECT * FROM temp_1743780826752_berka_accounts_17_dataset limit 25	2	15%	25%	25	75	100	BREAKING		Enabled	Unassigned		Actions
+	Limit3	SELECT * FROM temp_1743780826752_berka_accounts_17_dataset limit 3	0*	15%	3.0000%	3	97	100	PASSING		Enabled	Unassigned		Actions

Rule Types

SQL-Based Rules

Depending on the complexity, you can choose from short-form or long-form rules.

Simple

Just the condition (short-form). For example, using the column email_address. This runs against the dataframe and uses Spark SQL syntax. Simple rules can be thought of as everything after the where clause.

See the Simple section for more details.

Freeform

Where 'Simple' rules just use the condition, 'Freeform' rules use the complete SQL statement. When more complex SQL is required, you can express more with Freeform including joins and window statements.

See the Freeform section for more details.

Native

Native rules use the SQL dialect of the underlying connection and database. Files are not eligible for native SQL rules. This is ideal if you want to use pushdown profiling and you want to use existing SQL logic. When coupled with pushdown profiling, you can achieve a very minimal infrastructure footprint.

See the Native section for more details.

Stat rules

Write rules against meta data and profiling stats. Complex counts and ratios can be referenced with simple syntax.

See the Dataset-level stat rules section for more details.

Data Type rules

See the Data Type Rules section for more details.

SQL Based Rules

Collibra Data Quality uses ANSI-compliant SQL and offers pushdown rules that leverage specific database syntax.

Simple

This is the condition after the "where" clause.

Depending on the complexity, users can choose from short form or long form rules.

Simple

Just the condition (short form). For example, using the column email_address. This runs against the dataframe and uses Spark SQL syntax. Simple rules can be thought of as everything after the where clause.

```
email_address is not null and email_address != ''
```

> **Note** All built-in spark functions are available to use. For more information about simple and freeform SQL rules, see the official Apache Spark documentation.

Freeform

Fully defined SQL for more detailed checks.

Where 'Simple' rules just use the condition, 'Freeform' rules use the complete SQL statement. When more complex SQL is required, you can express more with Freeform including joins and window statements.

```
select * from @DATASET_NAME where
email_address is not null and email_address != ''
```

> **Note** Use the built-in Spark functions for non-Pushdown connections. Native SQL will not work, unless you switch to Native SQL rules. See (https://spark.apache.org/docs/2.3.0/api/sql/) for Simple and Freeform SQL rules.

> **Important** Column name references on the Spark dataframe do not necessarily match the column names.

For more information, go to Freeform SQL.

Add Secondary Datasets

You can add one or more secondary datasets to a Data Quality Rule. Secondary datasets are useful when you want to compare one table against another table.

For optimal performance, if possible, define and scan a view instead of adding a secondary dataset.

If the dataset is via a Pull Up connection, you can add it from either the same connection type or from another connection type (also referred to as a cross-connection rule). If the dataset is via a Pushdown connection, you can only add secondary datasets from the same connection.

> **Important** Total records are calculated by adding the total number of breaking and passing records. Therefore, when referencing a secondary dataset in a native rule, the extra rows from the secondary dataset are included in the query results on the Findings page, skewing the total rows calculation and the percentage of breaking records.

A typical use case is to verify the validity of a column by comparing it to a similar column in another table. For example, a health insurance provider may use proprietary codes that are maintained in a main list. In this case, you would compare the code values in a daily activity table with the code values in the main list.

To compare the `id` column in one table to the `id` column in another table, the rule may appear as follows:

```
select * from @table1 where id
not in ( select id from @table2 )
```

To compare multiple columns from one table to another, the rule may appear as follows:

```
select id, app_id, email, guid_num from @table1
EXCEPT
select id, app_id, email, guid_num from @table2
```

There is a known limitation where a Cartesian join which produces a greater number of rows than the initial dataset shows a score greater than 100%. There is not a workaround other than the query for the DQ job be the query that produces that larger number of rows.

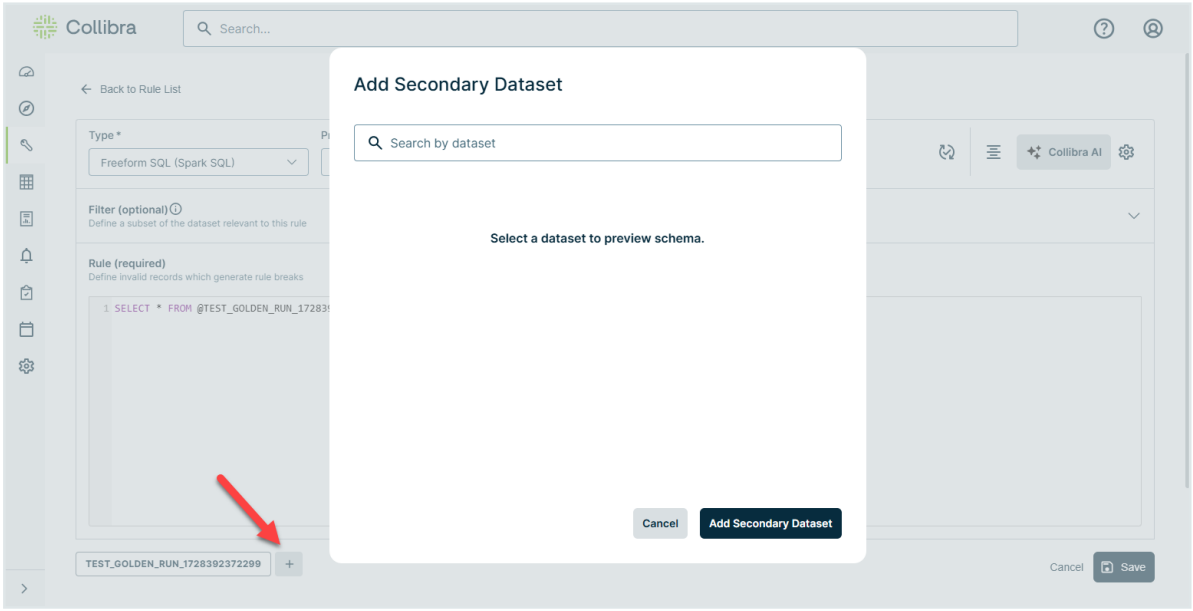
> **Note** @t1 rules where Cartesian joins are created are not supported. A @t1 rule is an out-of-the-box rule type that creates a dataset using the previous run's date as the run date to read data from a multi-day run period so that data from the current and previous days can be evaluated.

For additional information on writing SQL queries, including using Joins, go to Freeform SQL.

Adding a secondary dataset from the same connection

You can add one or more secondary datasets from the same connection type in both Pull Up and Pushdown connection types. This feature is for convenience purposes and not required for rule processing.

1. In the Rule Workbench, click the + icon and enter the name of the dataset in the Add Secondary Dataset dialog. For more information, go to Create a data quality rule.



2. Add your SQL rule.
3. Click **Save**.

> **Important** If you edit the rule later, you will need to add the secondary dataset again. Secondary datasets displayed do not persist when a rule is edited. Secondary datasets in the query itself are still referenced when the job is run.

Adding a secondary dataset from a different connection

Pullup connections support adding secondary datasets from a different connection. This feature is not currently available for Pushdown connections.

1. In the Rule Workbench, click the + icon and enter the name of the dataset in the Add Secondary Dataset dialog. For more information, go to Create a data quality rule.
2. Add your SQL rule.
3. Click **Save**.
4. Update the command line to load the drivers for the other connection type.
    a. Display Data Findings, and open the **Job** tab.
    b. Select the **Command Line** tab.
    c. Add the following parameter to the command line:
    ```
    -addlib "/opt/owl/drivers/bigquery/"
    ```

Referencing secondary datasets

When you create a rule with secondary datasets, keep in mind the following points in order to avoid missing data set exceptions:

- Ensure that the name of your first data set matches the name of the main data set on which your rule is created.
- The column names in your table must be exact matches with the syntax of your query.

> **Note** If the column name uses CAPS, then your query must also use CAPS.

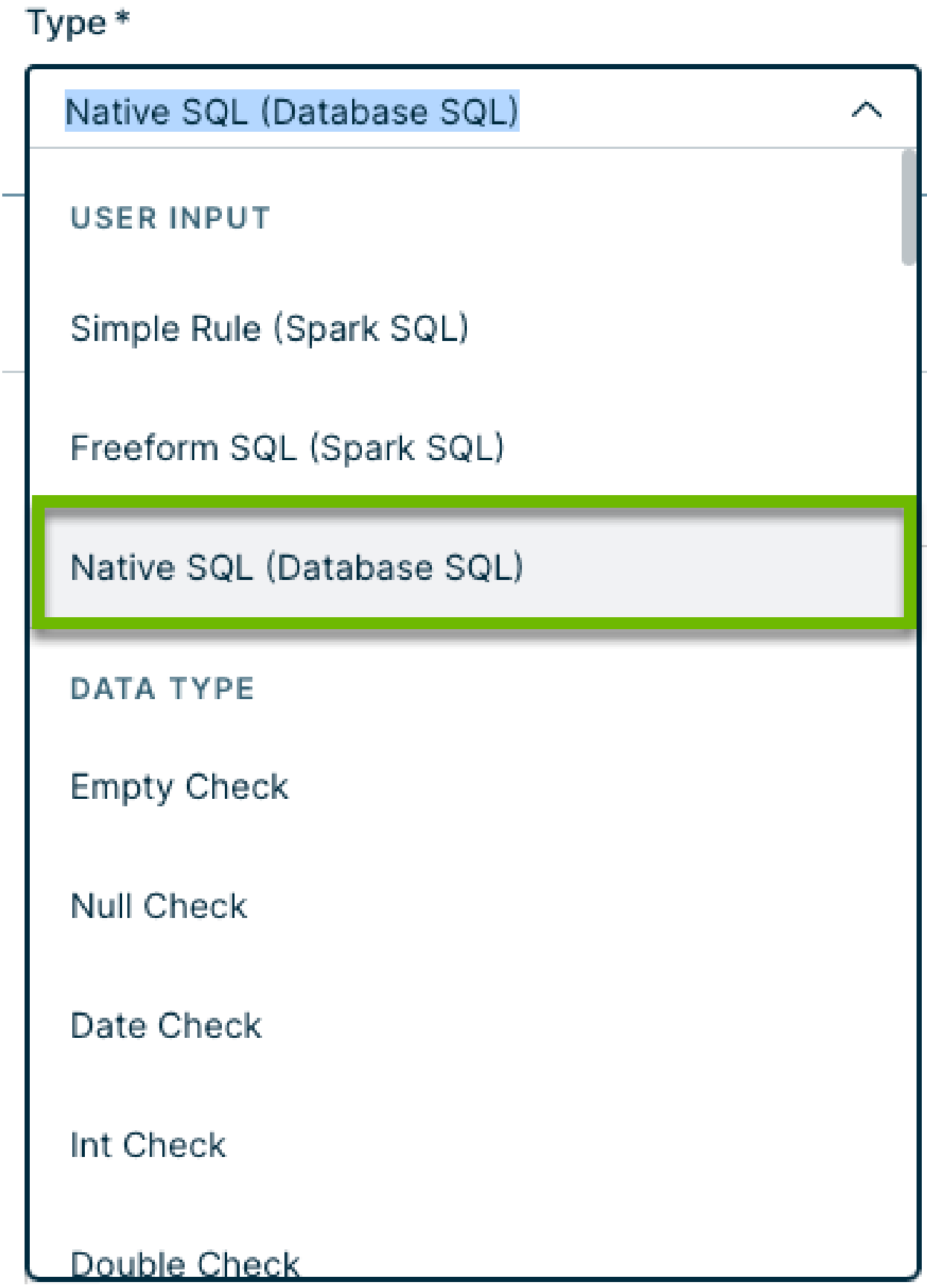
When you run a Job that references a secondary dataset, a temporary Job to load the secondary dataset appears on the Jobs page. This Job temporarily changes the dataset name while the Spark process runs. When you click the temporary Job after its status updates to Finished, the Findings page may show Rows: undefined in the Profile details. If this happens,

you can refresh the Jobs page until the Job reverts to its actual name, and then click the Job to view the results.

Native

With Native SQL expressions we provide capability to use your existing validation statements, even if they use database-specific (Postgres/DB2/Oracle/MSSQL/etc.) functions or expressions.

This is commonly used when using pushdown profiling and/or pre-existing SQL validations exist in a specific syntax. Native rules are often referred to as pushdown rules.


Type *
Native SQL (Database SQL)
USER INPUT
Simple Rule (Spark SQL)
Freeform SQL (Spark SQL)
Native SQL (Database SQL)
DATA TYPE
Empty Check
Null Check
Date Check
Int Check
Double Check

If you have rules already written in Oracle, Sybase, or DB2 syntax - Copy/Paste the rule directly into the Native SQL section.

When creating a Native SQL rule, keep the following in mind:

- The SQL query must be a valid expression that can be run as a subquery. To avoid pulling large amounts of data into memory, Collibra Data Quality will wrap your expression so it only fetches the number of rows returned. Rules should be written such that the query returns the anomalous rows.
- The SQL query must take less than 30 minutes to run. It is recommended to use partitioned columns for efficiency.
- When testing the SQL query from the app, it is helpful if it takes less than 30 seconds to run. You can add a limit to reduce query time, or test the query in your SQL Editor.
- CTE is not supported for Native SQL rules (but it is supported for Freeform SQL rules).

> **Note** Testing native rules can be done quickly by limiting the results or using an external SQL IDE as well.

> **Warning** This rule type is not eligible to store break records, just the score itself! Additionally the rule will apply the logic directly to the JDBC connection. Native rules will run on the entire population of the data, regardless of any scope set for the DQ job (-q flag defined in the scope section of Explorer). Any filter or predicate clause entered in the -q flag should be added to the logic of the Native rule as well.

Stat Rules

Stat rules are a powerful method for accessing profile statistics. These are typically sub-second operations that do not require scanning or iterating. There are several cases where SQL struggles to support rules, such as: isNull but not "null count" or nullRatio or nullPercent. Or having access to types without doing crazy cast() operations.

For example, here is a simple stat rule that breaks when the symbol column contains less than 800 rows of data that are not strings:

```
select * from @dataset where
fname.$type != 'String' AND $rowCount < 800
```

Dataset-level stat rules

Provides observability into your data at the dataset level, enabling you to detect issues with the entire dataset.

Dataset-level Stat Rule	Rule Example	Description of Rule Example
$totalTimeInSeconds	$totalTimeInSeconds > 25	Breaks when the DQ job runs longer than 25 seconds.
$totalTimeInMinutes	$totalTimeInMinutes > 5	Breaks when the DQ job runs longer than 5 minutes.
$totalTimeInHours	$totalTimeInHours > 1	Breaks when the DQ job runs longer than 1 hour.
$rowCount	$rowCount < 9000	Breaks when the row count less than 9,000.
$runDate	$runDate='2025-01-24'	Applies the rule to DQ jobs run on 2025-01-24.
$daysWithoutData	$daysWithoutData > 4	Breaks when a dataset has missing or no rows for 5 days.
$runsWithoutData	$runsWithoutData > 4	Breaks when a dataset runs but has missing or no rows for 5 days.
$daysSinceLastRun	$daysSinceLastRun > 4	Breaks when a dataset has not run for 5 days.

Column-level stat rules

Provides observability into your data at the column level.

Column-level Stat Rule	Rule Example	Description of Rule Example
.$type	symbol.$type != 'String'	Breaks when a row in the symbol column is not a string.
.$min	symbol.$min > 'apple'	Breaks when a row in the symbol column is greater than a given value. $min performs a lexicographical sort that works for strings and numbers.

Column-level Stat Rule	Rule Example	Description of Rule Example
.$minNum	high.$minNum > 13	Breaks when a row in the age column is greater than 13. The row is type-casted to a numeric for simple number checks.
.$mean	row_id.$mean > '4.500'	Breaks when the mean is greater than a given value.
.$max	symbol.$max > 'apple'	Breaks when a row in the symbol column is greater than a given value.
.$maxNum	high.$maxNum > 13	Breaks when the numeric value falls outside an acceptable range.
.$uniqueCount	id.$uniqueCount != $rowCount	Breaks when the uniqueCount of a field doesn't match the rowCount.
.$uniqueRatio	gender.$uniqueRatio between .4 and .6	Breaks when the ratio of uniqueCounts of a given field doesn't match the rowCount.
.$nullRatio	lname.$nullRatio not between .4 and .6	Breaks when the ratio of nulls no longer falls within acceptable range.
.$nullPercent	lname.$nullPercent not between 40 and 60	Breaks when the percent of nulls no longer falls within acceptable range
.$nullCount	symbol.$nullCount >= 1	Test for a single null.
.$emptyRatio	open.$emptyRatio > 0.2	Breaks when the ratio of empties no longer falls within acceptable range.
.$emptyPercent	open.$emptyPercent > 20	Breaks when the percent of empties no longer falls within an acceptable range.
.$emptyCount	open.$emptyCount > 100	Breaks when the emptyCounts of a field no longer fall within an acceptable range.
.$mixedTypeRatio	open.$mixedTypeRatio > 0.2	Breaks when the ratio of mixed data types no longer falls within an acceptable range. For example, Strings and Ints in the same field.

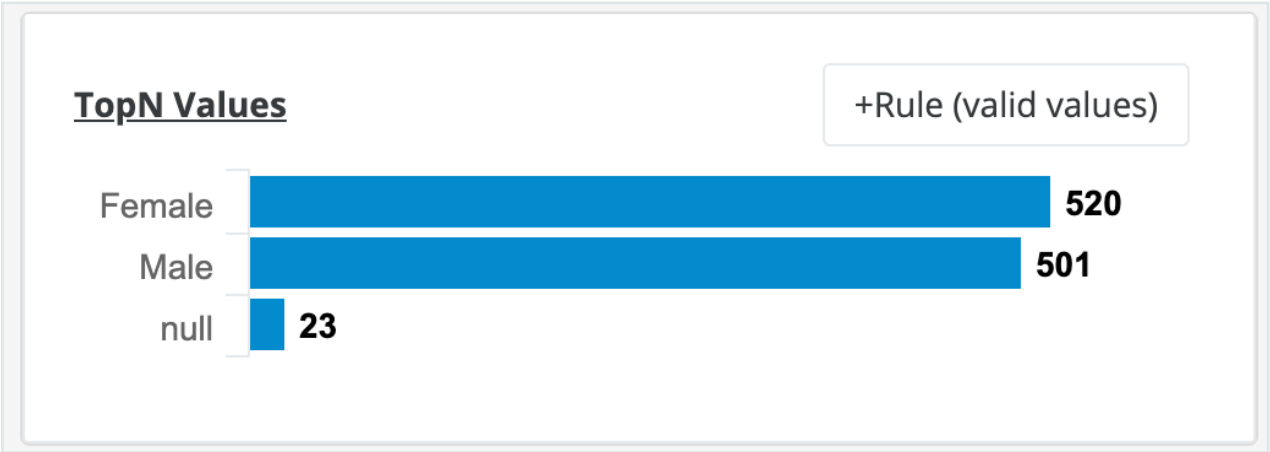
Column-level Stat Rule	Rule Example	Description of Rule Example
.$mixedTypePercent	open.$mixedTypeRatio > 20	Breaks when the percent of mixed data types no longer falls within an acceptable range. For example, Strings and Ints in the same field.
.$mixedTypeCount	open.$mixedTypeCount >= 1	Breaks when the mixed data typeCount no longer falls within an acceptable range. For example, Strings and Ints in the same field.

Distribution Rule

There is a common case in DQ where you want to know the distribution of a column's value. Consider gender. It can be expected that a column named gender consists of roughly 40-60% males and roughly 40-60% females if the data set is large and represents the population. This can be difficult to express in plain SQL, but it is very easy with the below syntax.

```
gender['Male'].$uniquePercent between 40 and 60
```

Column Value-level	Rule
.$uniqueCount	credit_rating['FAIR'].$uniqueCount > 7
.$uniquePercent	credit_rating['GOOD'].uniquePercent between 40 and 60



Known Limitations

- The $daysWithoutData stat rule has a limitation where rules that are not named "daysWithoutData" do not display their Pulse View charts when you drill into the rule on the Findings page. A possible workaround for this is to name your stat rule exactly as it appears here: daysWithoutData.
- You cannot combine stat rules or distribution rules with regex rules in the same rule. For example, `car_vin rlike '$[asdf][0-9]' and car_vin.$uniqueCount`

Data type rules

> **Note** See also: Data type rules.

Data type

Rule	Rule type	Description
Empty check	EMPTYCHECK	Checks whether the target column has empty values or not.
Null check	NULLCHECK	Checks whether the target column has *NULL* values or not.
Date check	DATECHECK	Checks whether the target column has **only** DATE values or not.
Integer check	INTCHECK	Checks whether the target column has **only** integer values or not.
Double check	DOUBLECHECK	Checks whether the target column has **only** DOUBLE values or not.
String check	STRINGCHECK	Checks whether the target column has **only** STRING values or not.

Rule	Rule type	Description
Mixed datatype check	DATATYPECHECK	Checks the dataType of the field. **Note** Mixed datatype rules are not supported in Pushdown.

Rule Summary

The Rule Summary page provides a list of rules and insightful donut chart views of rule details, rule breaks, and rule dimensions over a specific period of time. By understanding these details, you can better assess the amount of time and effort you spend writing and managing rules.

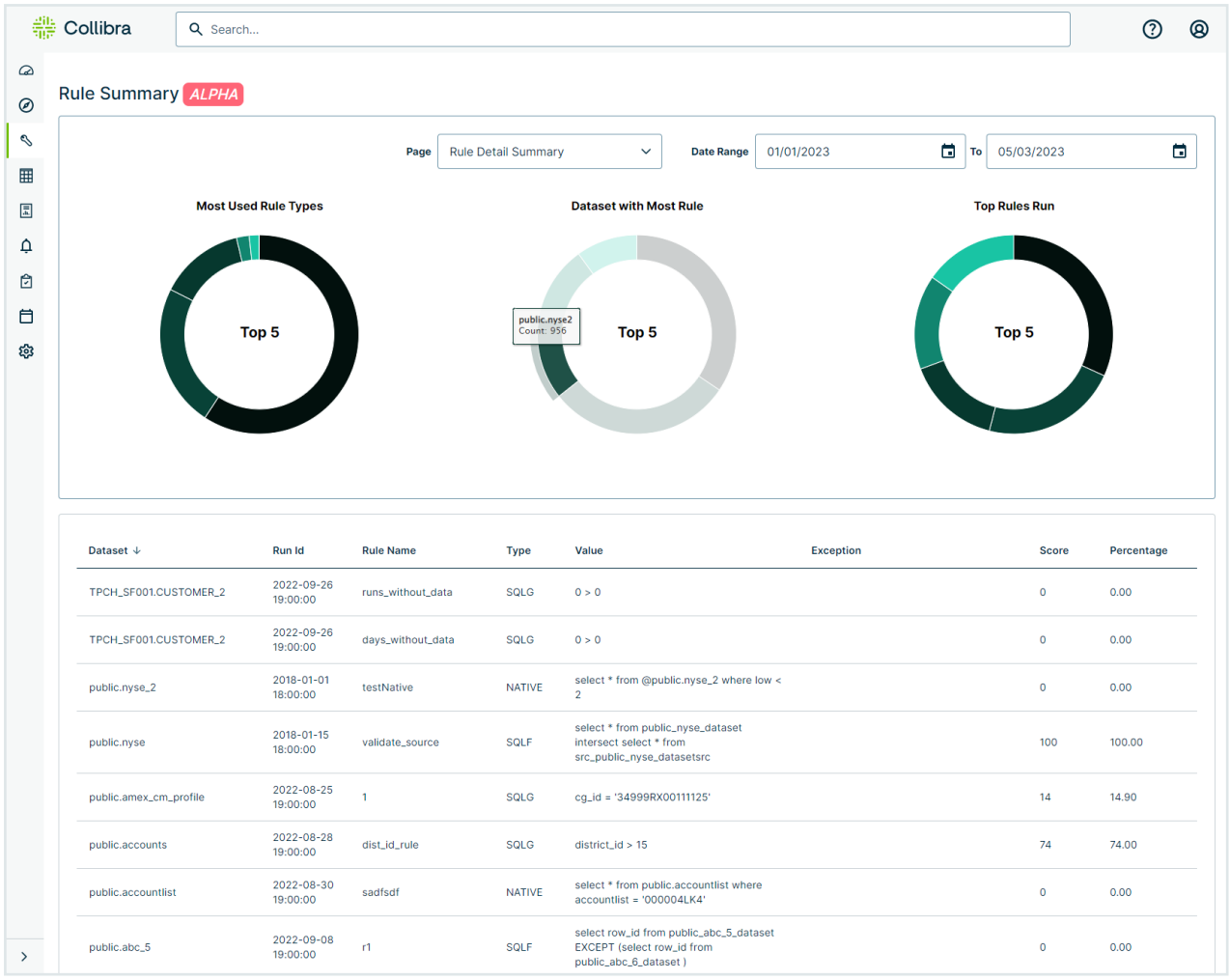
Use the **Page** dropdown menu to select a summary one of the following summaries, then select a **Date Range**:

- Rule Detail Summary
- Rule Breaks
- Rule Dimension Summary

Rule Detail Summary

The Rule Detail Summary shows summaries of Top 5 rule values in three charts:

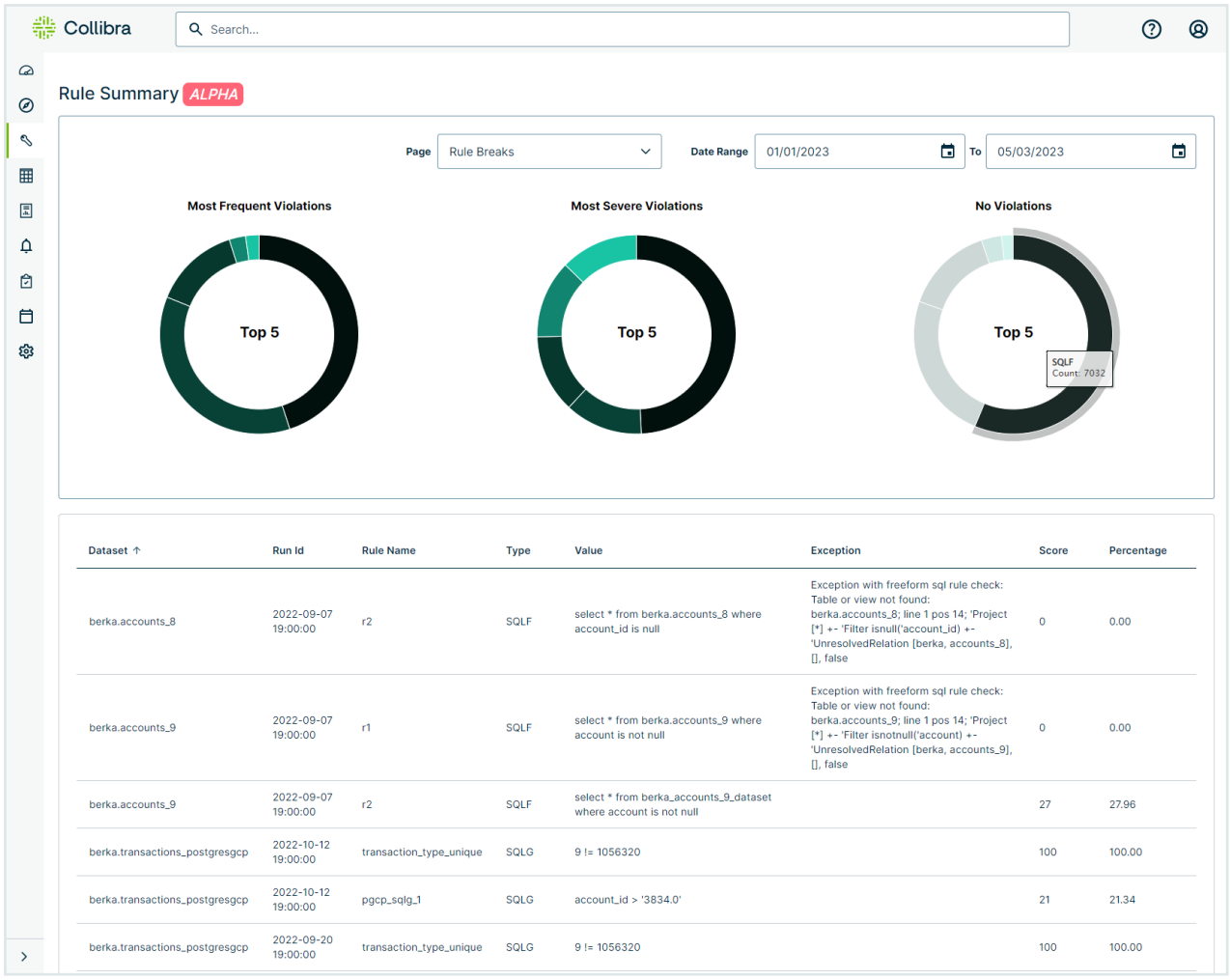
Chart	Description
Most Used Rule Types	Shows the most frequently used rule types with details about the number of times they are used and the percentage of all rules for which a particular rule type accounts. Hover your cursor over the donut chart to view the details.
Datasets with Most Rules Written	Shows the datasets with the most rules Hover your cursor over the donut chart to view the details.
Top Rules Run	Shows the name of the rule and its count with the most runs in your environment. Hover your cursor over the donut chart to view the details.



Rule Breaks

The Rule Breaks summary shows summaries of Top 5 rule break details in three charts:

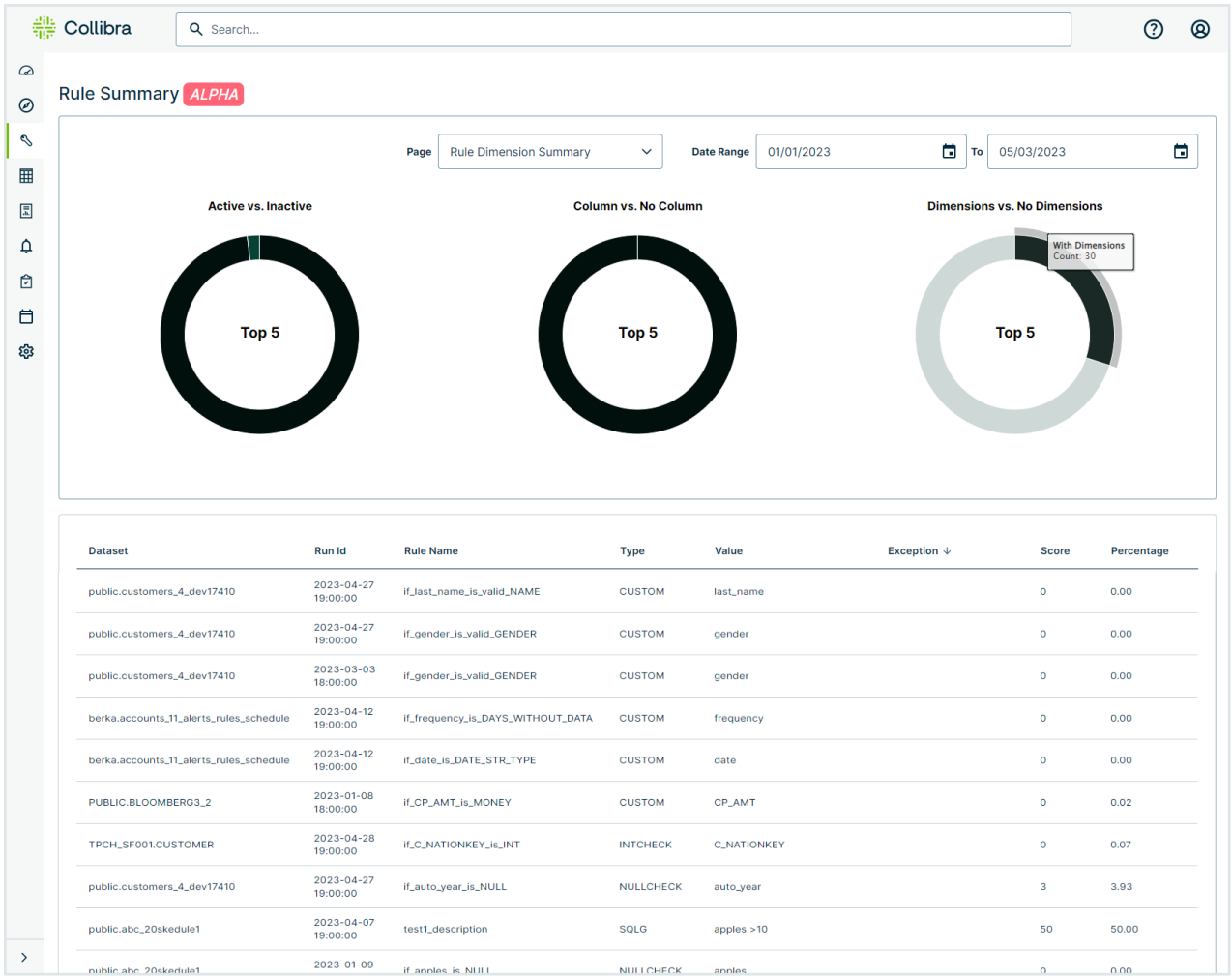
Chart	Description
Most Frequent Violation	Shows the rule types that most frequently violates its conditions. Hover your cursor over the donut chart to view the details.
Most Severe Violations	Shows the rule types with violations that result in the lowest scores in your environment. Hover your cursor over the donut chart to view the details.
No Violations	Shows the rule types with the fewest number of violating runs in your environment. For example, if SQLF rules run 7032 times without violation, and SQLF is the most frequently run rule in your environment, then it will appear as the largest section of the donut chart. Hover your cursor over the donut chart to view the details.



Rule Dimension Summary

The Rule Dimension Summary shows summaries of rule activity, rules with columns, and rules with dimensions in three charts:

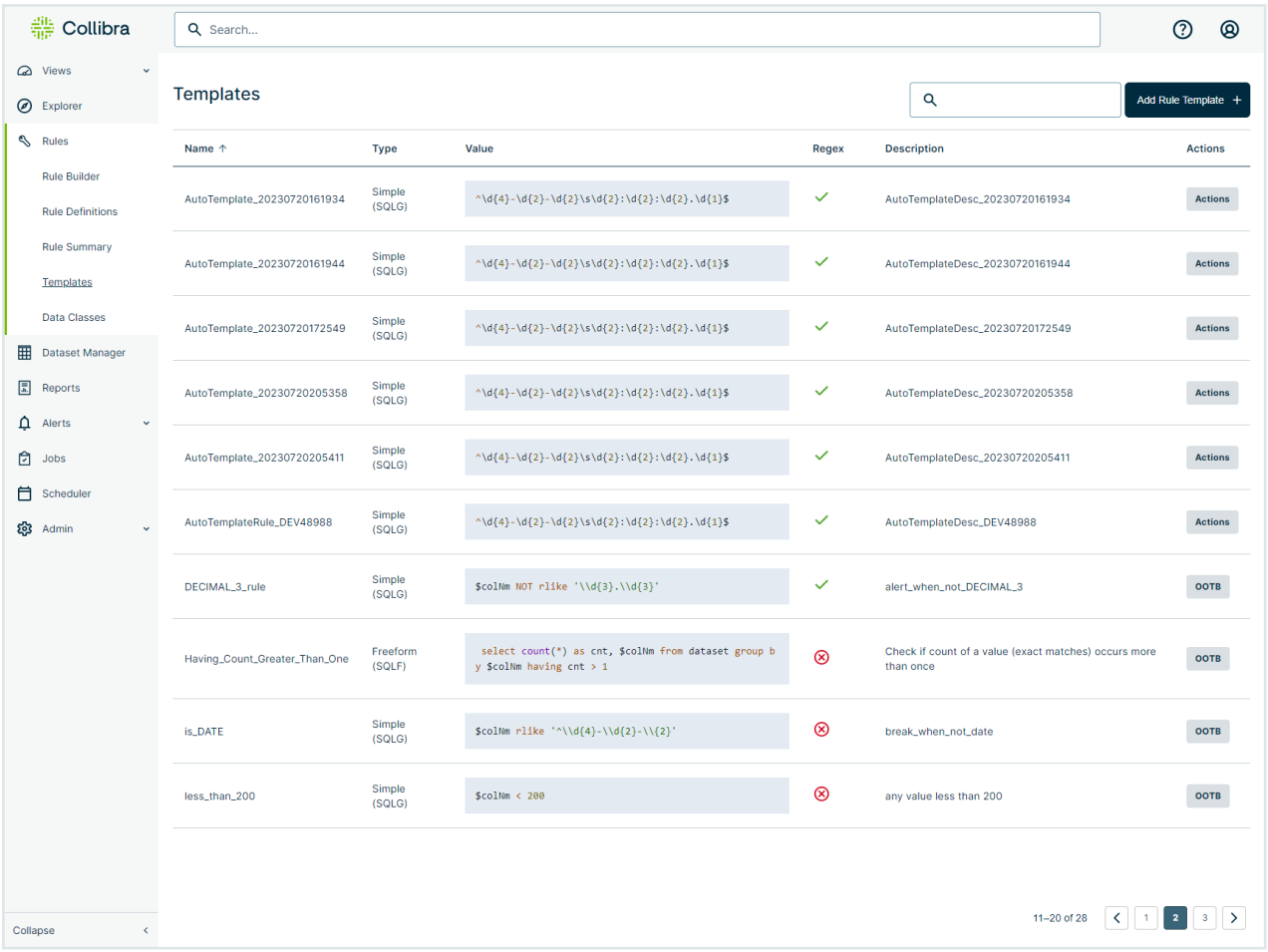
Chart	Description
Active vs. Inactive	Shows the number of active and inactive rules on all of your datasets over a period of time. Hover your cursor over the donut chart to view the details.
Column vs No Column	Shows the number of rules with columns. Hover your cursor over the donut chart to view the details.
Dimensions vs. No Dimensions	Shows the number of rules that are and are not associated with a DQ Dimension. Hover your cursor over the donut chart to view the details.



Templates

Templates are custom and out-of-the-box reusable rules that are ideal for rules that apply to more than one dataset. Because templates are dataset independent, you can write a rule once and then run it against any dataset in your Collibra DQ environment as many times as necessary. This can eliminate the need to write new rules for frequently used compliance checks from one dataset to the next, and can also reduce the amount of time you spend writing complex rules, such as ones that include regex and cast operations.

> **Important** As of version 2025.03, ROLE_ADMIN permission is required to add or edit templates.



The Templates page shows a list of all template rules in your Collibra DQ environment and lets you add new ones with the Add Rule Template wizard.

> **Note** You may input any character in the **Search** field, including special symbols, to perform regex-based search queries.

Adding a Template Rule

> **Important** As of version 2025.03, ROLE_ADMIN permission is required to add or edit templates.

1. On the sidebar menu, click **Rules**, then click **Templates**.
   » The Templates page opens.
2. In the upper right corner of the Templates page, click **Add Rule Template**.
   » The Add Rule Template form opens.

3. Enter the required information.

Option	Description	Required
Name	The name of your template rule. Enter a unique name without spaces to identify your Template rule easily.	✓ Yes
Type	The rule type of your template rule. Only Simple and Freeform are eligible rule types for templates. Select an option from the dropdown menu. > **Note** If you select Freeform (SQLF), you can use a more complex SQL expression that includes regex, but the validation function will be unavailable.	✓ Yes
Is Regex?	Select this option if the Value field contains a regular expression (regex). When you select Is Regex, the Type is SQLG and contains just the regex value, and neither `select * from` statement nor quotes are required for the rule to run. When Is Regex is not selected, the Type is SQLF and requires a `select * from` statement to run. > **Important** When you select Is Regex, the Type is always SQLG. The format of your regex must be just the expression itself without quotes or any other SQL.	✗ No
Value/Regex	The query or regex that your rule template applies to run against a dataset. Enter a query or expression for your rule template to run against a dataset. > **Note** Value displays when Is Regex is not selected, and Regex displays when Is Regex is selected.	✓ Yes

Option	Description	Required
Input/Validate Input	Because regex treats inputs as one continuous string, the input field informs the system to treat each line separately. Optionally enter a value or a comma delimited list of values, then click Validate Input to test them against your regex. **Note** Input only displays when Is Regex is selected.	✕ No
Description	The description that helps identify your template rule. It is best practice to base your description on what the Value field queries.	✕ No

4. Click **Save**.
5. Your template rule is available on the Templates page and on the Rule Workbench when you select it from the Templates section of the Type dropdown menu.

Using a Template Rule on the Rule Workbench

To use a data class rule to run against a dataset, follow these steps.

1. Click the in the sidebar menu, then click **Rule Builder**.
   » The Dataset Rules page opens.
2. There are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench page opens with the preferences of your existing rule.
   b. Click **Add Rule** in the upper right corner to create a new rule.
      » The Rule Workbench page opens.
3. Click the Type dropdown menu and scroll down to the Template section, then select your Template rule.
   » The Template rule populates in the query builder.
4. Enter a name for your rule in the **Rule Name** input field.

   Important Rule names cannot contain spaces.

5. Select a primary column from the Primary Col dropdown menu.

> **Note** Primary Column is required when you use Template rules.

6. Click **Validate** to validate your expression.
7. Optionally click **Beautify** to format your SQL query in a more readable layout.
8. Click **Save** to apply your rule to your dataset.

Out-of-the-box Templates

Collibra DQ has 12 out-of-the-box (OOTB) templates for common rule checks. OOTB templates cannot be edited or deleted.

Name	Value	Description
DECIMAL_3_rule	`^\d{4}-\d{2}-\d{2}\s\d{2}:\d{2}:\d{2}.\d{1}$`	The rule breaks when a column value does not contain a sequence of 3 digits before a decimal, and three digits after the decimal.
Having_Count_Greater_Than_One	`select count(*) as cnt, $colNm from dataset group by $colNm having cnt > 1`	Checks if the count of an exact match value occurs more than once.
is_DATE	`$colNm rlike '^\\d{4}-\\d{2}-\\{2}'`	The rule breaks when a column value is in YYYY-MM-DD format.

Name	Value	Description
less_than_200	`$colNm < 200`	Checks for any column values less than 200.
Not_In_Current_run	`select $colNm from @t1 where $colNm not in ( select distinct ($colNm) from dataset )`	Checks if column values are not present in the current run.
Not_In_Previous_run	`select $colNm from dataset where $colNm not in ( select distinct ($colNm) from @t1 )`	Checks if column values are not present in the previous run.

Name	Value	Description
Percent_Move_5	`select i.$colNm, s.averg from dataset i , (select avg (cast ($colNm as double ) ) as averg from dataset ) s where i.$colNm < (s.averg * 0.95 ) or i.$colNm > (s.averg * 1.05)`	Checks if there are column values present that are 5% greater or less than the average value in that column.

Name	Value	Description
Percent_Move_10	`select i.$colNm, s.averg from dataset i , (select avg (cast ($colNm as double ) ) as averg from dataset ) s where i.$colNm < (s.averg * 0.90 ) or i.$colNm > (s.averg * 1.10)`	Checks if there are column values present that are 10% greater or less than the average value in that column.

Name	Value	Description
Percent_Move_20	`select i.$colNm, s.averg from dataset i , (select avg (cast ($colNm as double ) ) as averg from dataset ) s where i.$colNm < (s.averg * 0.80 ) or i.$colNm > (s.averg * 1.20)`	Checks if there are column values present that are 20% greater or less than the average value in that column.

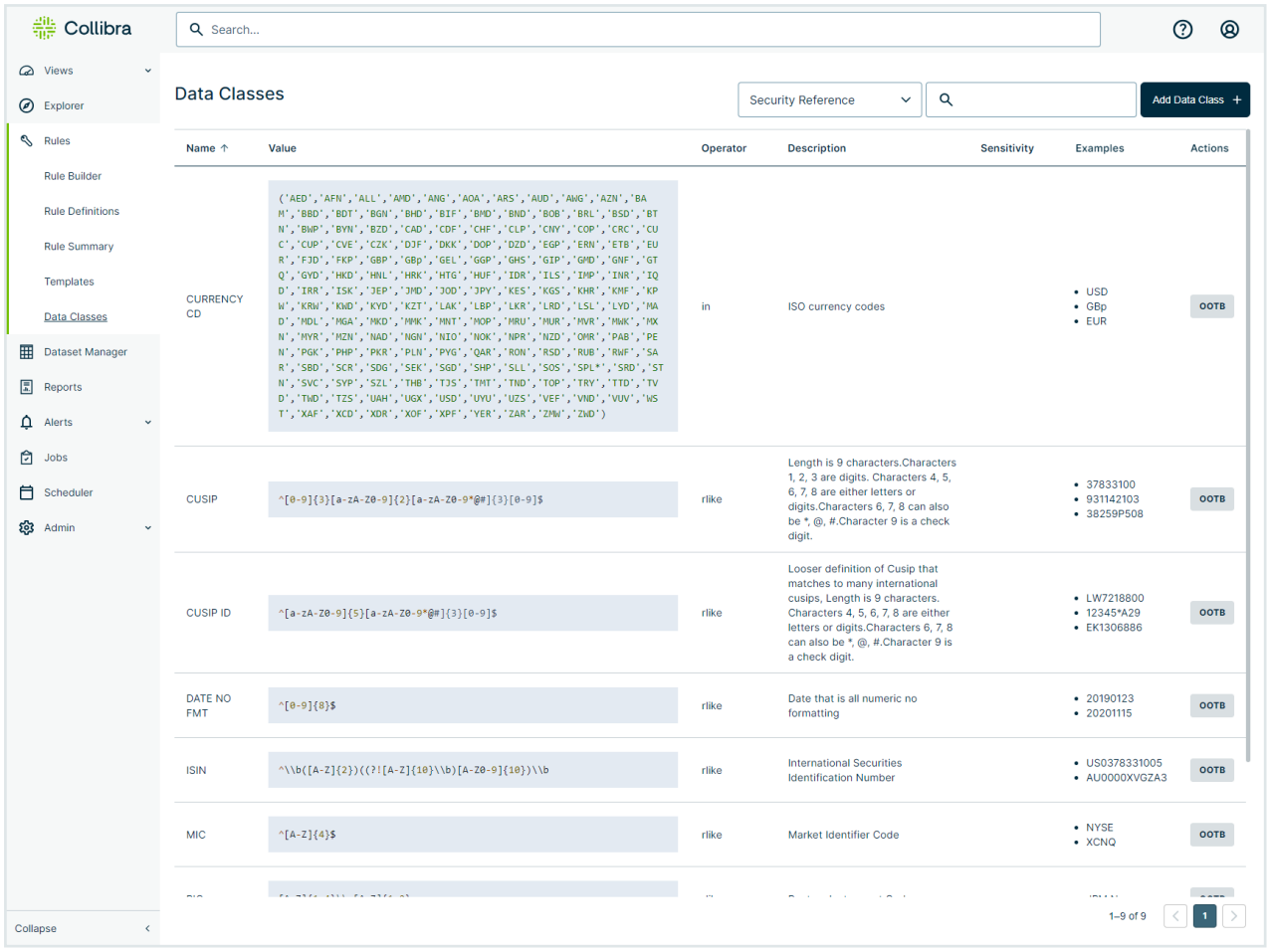
Name	Value	Description
Percent_Move_50	`select i.$colNm, s.averg from dataset i , (select avg (cast ($colNm as double ) ) as averg from dataset ) s where i.$colNm < (s.averg * 0.50 ) or i.$colNm > (s.averg * 1.50)`	Checks if there are column values present that are 50% greater or less than the average value in that column.
Two_Decimal_Places	`select d[1] as rs from ( select SPLIT ($colNm, '\\.') as d from dataset ) where length(d [1]) > 2`	Checks if the column decimal place precision exceeds a length of 2.

Name	Value	Description
USTAX_rule	`$colNm NOT rlike 'USTAX'`	The rule breaks when the specified column does not contain "USTAX".

Data Classes

Data Classes are common checks that help to identify and tag column-level semantic traits of a rule, such as credit score, SSN, and ZIP code. With many out-of-the-box Data Class rules to choose from, as well as the ability to create your own custom Data Classes, save time and effort by reusing common data class checks when writing rules. With just a matter of clicks, you can quickly insert saved Data Classes from the Rule Workbench to run against your dataset, apply them to columns on the Profile page of a dataset, or add them to Data Categories in the Admin Console.

You can use the Category filter to see the data classes that an Admin has grouped together into data categories. For example, the following image shows Data Classes grouped by the Security Reference data category.



Adding a new Data Class

1. On the sidebar menu, click **Rules**, then click **Data Classes**.
   » The Templates page opens.
2. In the upper right corner of the Templates page, click **Add Data Class**.
   » The Add Data Class form opens.
3. Enter the required information.

Option	Description	Required
Name	The name of your data class rule. Enter a unique name to easily identify your Data Class rule.	✓ Yes
Is Regex?	Select this option if the Value field contains a regular expression (regex).	✗ No

Option	Description	Required
Operator	The symbol or keyword that your query uses to perform the specified operations on your data. Select an option from the dropdown menu. > **Note** When Is Regex is selected, the operator is always rlike.	✓ Yes
Value/Regex	The query or regex that your Data Class applies to run against a dataset. Enter a query or expression for your Data Class to run against a dataset. > **Note** Value displays when Is Regex is not selected, and Regex displays when Is Regex is selected.	✓ Yes
Multi-line Input	Because regex treats inputs as one continuous string, multi-line input informs the system to treat each line separately. Optionally enter a value or a comma delimited list of values, then click Validate Input to test them against your regex. > **Note** Multi-line Input only displays when Is Regex is selected.	✗ No
Examples 1, 2, and 3	A bullet list of real-world values that your data class describes. Optionally enter a sample value that meets the criteria of your data class rule. For example, a credit score is a string of three numbers, such as 790.	✗ No
Sensitivity Label	Tags any findings from your rule as confidential data with a sensitivity badge on the Findings page. Admins can define Sensitivity labels on the Sensitive Labels page in the Admin Console.	✗ No

Option	Description	Required
Description	The description that helps identify your data class rule. It is best practice to base your description on what the Value or Regex fields query.	✕ No
Column Name	The name of the column being checked that a value must exactly equal to be classified as this data class. **Tip** You can leave this field blank to apply this data class to columns with different names or types.	✕ No
Column Type	The data type of the column being checked that a value must exactly equal to be classified as this data class. **Tip** You can leave this field blank to apply this data class to columns with different names or types.	✕ No
Column Match %	The minimum percentage that a column being checked must exactly equal for it to be classified as this data class. The default value is 75%.	✓ Yes
Allow Null	Controls whether to include (selected) or exclude (not selected) NULL values from the query. This option is not selected by default.	✕ No
Allow Empty	Controls whether to include or exclude empty values from the query. This option is not selected by default. Select the checkbox option to include empty values in the query, or deselect it to exclude them.	✕ No

4. Click **Save**.
5. Your data class rule is available to add to a Data Category in the Admin Console, apply to a column on the Profile page of a dataset, or to use to create a dataset rule on the Rule Workbench.

Using a Data Class rule

You can use Data Class rules to identify the desired patterns for data. Unlike custom and Template rules, Data Classes return results for all values that do not match their exact expressions.

> **Example**
> When you run the rule `Column A < 50` against a dataset with 5 rows where the values in Column A are 0, 1, 100, 200, and 300, custom and Template rules will return just 2 break records for the rows with the values of 0 and 1, whereas Data Class rules will return 3 break records for the rows with the values of 100, 200, and 300.

To use a data class rule to run against a dataset, follow these steps.

1. Click the in the sidebar menu, then click **Rule Builder**.
   » The Dataset Rules page opens.
2. There are two ways to open the Rule Workbench.
   a. From an existing rule on the Rules tab, click **Actions**, then click **Edit**.
      » The Rule Workbench page opens with the preferences of your existing rule.
   b. Click **Add Rule** in the upper right corner to create a new rule.
      » The Rule Workbench page opens.
3. Click the Type dropdown menu and scroll down to the Data Class section, then select your Data Class rule.
   » The Data Class rule populates in the query builder.
4. Enter a name for your rule in the **Rule Name** input field.

   > **Important** Rule names cannot contain spaces.

5. Select a primary column from the Primary Col dropdown menu.

   > **Note** Primary Column is required when you use Data Class rules.

6. Optionally click + to add a secondary dataset.
   a. Enter the name of your secondary dataset in the search bar when the **Add Secondary Dataset** dialog appears, then click **Add Secondary Dataset**.
7. Click **Validate** to validate your expression.
8. Optionally click **Beautify** to format your SQL query in a more readable layout.

9. Click **Save** to apply your rule to your dataset.

Out-of-the-box Data Classes

Data Quality & Observability Classic has 57 out-of-the-box (OOTB) Data Classes. OOTB Data Classes cannot be edited or deleted.

Name	Value	Description
AMEX	`^(?<amex>3[47][0-9]{13})$`	Validates American Express credit card numbers.
CELSIUS	`^(([0-9°c\\s]+)(?:-[0-9°]+c))|([0-9°\\s]+c)$`	Checks for single temperature values in Celsius, and multiple temperature values within a specified range.

Name	Value	Description
CREDIT CARD	`^((4\\d{3})|(5[1-5]\\d{2})|(6011))-?\\d{4}-?\\d{4}-?\\d{4}|3[4,7]\\d{13}$`	Checks if the value does not match the credit card format.
CREDIT SCORE	`300 and 850`	Checks a range of credit scores between a poor 300 and an excellent 850.

Name	Value	Description
CURRENCY CD	('AED','AFN','ALL','AMD','ANG','AOA','ARS','AUD','AWG','AZN','BAM', 'BBD','BDT','BGN','BHD','BIF','BMD','BND','BOB','BRL','BSD', 'BTN','BWP','BYN','BZD','CAD','CDF','CHF','CLP','CNY','COP', 'CRC','CUC','CUP','CVE','CZK','DJF','DKK','DOP','DZD','EGP', 'ERN','ETB','EUR','FJD','FKP','GBP','GBp','GEL','GGP','GHS', 'GIP','GMD','GNF','GTQ','GYD','HKD','HNL','HRK','HTG','HUF', 'IDR','ILS','IMP','INR','IQD','IRR','ISK','JEP','JMD','JOD', 'JPY','KES','KGS','KHR','KMF','KPW','KRW','KWD','KYD','KZT', 'LAK','LBP','LKR','LRD','LSL','LYD','MAD','MDL','MGA','MKD', 'MMK','MNT','MOP','MRU','MUR','MVR','MWK','MXN','MYR','MZN', 'NAD','NGN','NIO','NOK','NPR','NZD','OMR','PAB','PEN','PGK', 'PHP','PKR','PLN','PYG','QAR','RON','RSD','RUB','RWF','SAR', 'SBD','SCR','SDG','SEK','SGD','SHP','SLL','SOS','SPL*','SRD', 'STN','SVC','SYP','SZL','THB','TJS','TMT','TND','TOP','TRY', 'TTD','TVD','TWD','TZS','UAH','UGX','USD','UYU','UZS','VEF', 'VND','VUV','WST','XAF','XCD','XDR','XOF','XPF','YER','ZAR', 'ZMW','ZWD')	Checks ISO currency codes.

Name	Value	Description
CUSIP	`^[0-9]{3}[a-zA-Z0-9]{2}[a-zA-Z0-9*@#]{3}[0-9]$`	Checks CUSIP numbers. The length is 9 characters. Characters 1-3 are numbers. Characters 4-8 are letters or numbers. Characters 6-8 can also represent *, @, and #. Character 9 is a check digit.

Name	Value	Description
CUSIP ID	`^[a-zA-Z0-9]{5}[a-zA-Z0-9*@#]{3}[0-9]$`	A looser definition of CUSIP that matches to many international CUSIP numbers. The length is 9 characters. Characters 1-3 are numbers. Characters 4-8 are letters or numbers. Characters 6-8 can also represent *, @, and #. Character 9 is a check

Name	Value	Description
		digit.
DATE MM/DD-/YY	`^\\d{1,2}/\\d{2}/\\d{2}$`	Checks if the date is in MM/DD-/YY format.
DATE MM/DD-/YYYY	`^\\d{1,2}/\\d{2}/\\d{4}$`	Checks if the date is in MM/DD-/YYYY format.
DATE NO FMT	`^[0-9]{8}$`	Checks if the date is all numeric without formatting.
DATE YYYY-MM-DD	`^\\d{4}-\\d{2}-\\d{2}`	Checks if the date format is in YYYY-MM-DD.

Name	Value	Description
DEA NUMBER	`^([ABFGMPR][\\w9]\\d{7})([-]\\d{3}\\w{2})?$`	Checks the formatting of a Drug Enforcement Agency number.
DISCOVER	`^(?<discover>5[1-5][0-9]{14})$`	Validates Discover credit card numbers.
EIN	`^[1-9]\\d?-\\d{7}$`	Checks if a value does not match the correct Employer Identification Number format.

Name	Value	Description
EMAIL	`'%_@__%.__%'`	Checks if a value does not match an email format.
FAHRENHEIT	`^(([0-9°f\\s]+)(?:-[0-9°]+f))|([0-9°\\s]+f)$`	Checks for single temperature values in Fahrenheit, and multiple temperature values within a specified range.

Name	Value	Description
FDA NDC BILLING	`^(0\\d{4}-?\\d{4}-?\\d{2}|\\d{5}-?0\\d{3}-?\\d{2}|\\d{5}-?\\d{4}-?0\\d{1})$`	Checks if a value is not in Food and Drug Administration billing code format.
FDA NDC PACKAGE	`^(\\d{4}-\\d{4}-\\d{2}|\\d{5}-\\d{3}-\\d{2}|\\d{5}-\\d{4}-\\d{1})$`	Checks if a value is not in the Food and Drug Administration package code format.

Name	Value	Description
FDA NDC PRODUCT	`^(\\d{4}-\\d{4}|\\d{5}-\\d{3}|\\d{5}-\\d{4})$`	Checks if a value is not in the Food and Drug Administration product code format.
GENDER	`^(?:m|M|male|Male|MALE|f|F|female|Female|FEMALE)$`	Checks if a value does not match a gender format.
GUID	`^[{]?[0-9a-fA-F]{8}-([0-9a-fA-F]{4}-){3}[0-9a-fA-F]{12}[}]?$`	Checks if a value is not a GUID.
ICD10 DIAGNOSIS	`^([A-TV-Z][0-9][A-TV-Z0-9])([.]{0,1})[A-TV-Z0-9]{0,4}$`	Checks if a value is not in ICD10 diagnosis code format.

Name	Value	Description
ICD10 PROCEDURE	`^[0-9BCDFGHX][0-9A-HJ-NP-Z]{2}[0-9A-HJ-NP-Z]{0,4}$`	Checks if a value is not in ICD10 procedure code format.
ICD9 DIAGNOSIS	`^(([V]\\d{2}[.]?\\d{0,2})|([E][089]\\d{2}[.]?\\d?)|(\\d{3}[.]?\\d{0,2}))$`	Checks if a value is not in ICD9 diagnosis code format.
ICD9 PROCEDURE	`^(\\d{2}[.]?\\d{1,2})$`	Checks if a value is not in ICD9 procedure code format.

Name	Value	Description
INT PASSPORT	`^[A-Z0-9<]{9}[0-9]{1}[A-Z]{3}[0-9]{7}[A-Z]{1}[0-9]{7}[A-Z0-9<]{14}[0-9]{2}$`	Checks if a value does not match an international passport format.
IP ADDRESS	`^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$`	Checks if a value does not match an IP address format.
ISIN	`^\\b([A-Z]{2})((?![A-Z]{10}\\b)[A-Z0-9]{10})\\b`	Checks if a value does not match an International Securities Identification Number.

Name	Value	Description
LAT	`^(\\+|-)?(?:90(?:(?:\\.0{1,6})?)|(?:[0-9]|[1-8][0-9])(?:(?:\\.[0-9]{1,6})?))$`	Validates that latitude coordinates are between -89 and +89 degrees, or -89.000001 and +899.000001 degrees.
LNG	`^(\\+|-)?(?:180(?:(?:\\.0{1,6})?)|(?:[0-9]|[1-9][0-9]|1[0-7][0-9])(?:(?:\\.[0-9]{1,6})?))$`	Validates that longitude coordinates are between -179 and +179 degrees, or -179.000001 and +179.000001 degrees.

Name	Value	Description
MASTERCARD	`^(?<mastercard>6(?:011|5[0-9]{2})[0-9]{12})$`	Validates Mastercard credit card numbers.
MIC	`^[A-Z]{4}$`	Market Identifier Code, such as NYSE.
MONEY	`^\\p{Sc}[0-9]`	Money format including $ and €.
NAME	`^\\p{L}+[\\p{L}\\p{Z}\\p{P}]{0,}`	A first name, last name, or full name.
NAME TITLE	`^(Mr|Ms|Dr|Sr|Miss|Mr.|Ms.|Mrs|Mrs.|Dr.|Sr.|Rev|Honorable)\\.?$`	Checks if a value contains a title, such as Mr., Mrs. , and Dr.

Name	Value	Description
NHS NUMBER	`^(\\d{3}\\s?\\d{3}\\s?\\d{4})$`	Checks the formatting of a National Health Service number.
NPI NUMBER	`^(\\d{10})$`	Checks the formatting of a National Provider Identifier number.
OPCS4	`^(([A-Z]\\d{2})(\\.-|\\.?\\d)?)$`	Checks if a value is not in OPCS-4 format.
PASSWORD MEDIUM	`^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%^&+=])(?=\\S+$).{8,20}$`	Checks password values containing 1 number, 1 letter, and 1 uppercase letter.

Name	Value	Description
PERCENT	`^\\d+(?:\\.\\d+)?%$`	Checks that a string is a valid percentage.
PHONE	`^(\\+\\d{1,2}\\s)?\\(?\\d{3}\\)?[\\s.-]\\d{3}[\\s.-]\\d{4}$`	Checks if a value does not match a phone number format.
RIC	`[A-Z]{1,4}\\.[A-Z]{1,2}`	Validates that a Reuters Instrument Code contains 1-4 uppercase letters.1-2 uppercase letters. For example, ABCD.-EF.

Name	Value	Description
ROUTING NUM	`^((0[0-9])|(1[0-2])|(2[1-9])|(3[0-2])|(6[1-9])|(7[0-2])|80)([0-9]{7})$`	Validates that a string value matches a valid bank Routing Transit Number (RTN), such as 013456-7890.
SEDOL	`^(?<Sedol>[B-Db-dF-Hf-hJ-Nj-nP-Tp-tV-Xv-xYyZz\\d]{6}\\d)$`	Validates that the alpha-numeric value is a valid Stock Exchange Daily Official List (SEDOL) code, such as B12345-6.

Name	Value	Description
SHIRT SIZE	`('L','M','S','XS','XL','XXL','2XL','3XL','l','m','s','xs','xl','xxl','Small','Medium','Large','X-Large')`	Validates that a value contains a valid shirt size.
SSN	`^(?!(000|666|9))\\d{3}-(?!00)\\d{2}-(?!0000)\\d{4}$`	3 digits dash 2 digits dash 4 digits.
STATE CD	`('AL','AK','AZ','AR','CA','CO','CT','DE','FL','GA','GU','HI','ID','IL','IN','IA','KS','KY','LA','ME','MD','MA','MI','MN','MS','MO','MP','MT','NE','NV','NH','NJ','NM','NY','NC','ND','OH','OK','OR','PA','PR','RI','SC','SD','TN','TX','UT','VT','VA','VI','WA','WV','WI','WY','AS','DC')`	Checks if a value does not match a state code format.

Name	Value	Description
STATE NAME	`('Alabama','Arkansas','Arizona','Alaska','California','South Dakota','Colorado','Connecticut','Delaware','Florida','Georgia','Guam','Hawaii','Idaho','Illinois','Indiana','Kansas','Kentucky','Louisiana','Maryland','Massachusetts','Michigan', 'Missouri', 'Minnesota','Mississippi','Montana','Oregon','Nebraska','Northern Mariana Islands','Maine','Nevada','New Hampshire','New Jersey','New Mexico','New York','North Carolina','North Dakota','Ohio','Oklahoma','Pennsylvania','Puerto Rico','Iowa','Rhode Island','South Carolina','Tennessee','Texas','Utah','Vermont','Virgin Islands','Virginia','Washington','West Virginia','Wisconsin','Wyoming','American Samoa','District of Columbia')`	Checks if a value does not match a state name format.
STREET ADDRESS	`^\\d{1,3}.?\\d{0,3}\\s[a-zA-Z]{2,30}\\s[a-zA-Z]{2,15}`	Checks that a string value is in a valid US street address format, such as 123 Broadway.

Name	Value	Description
SUFFIX	`('Jr','Jr.', 'Sr', 'Sr.', 'II', 'III', 'IV')`	Validates that a value contains a valid suffix after surnames, such as Jr., Sr., and II.
SYMBOL	`^([A-Z]{1,4})(:|\\.|-){0,1}([A-Z]{1,1})$`	Validates that a stock symbol contains 1-4 uppercase letters, an optional colon :, period ., or hyphen -, and 1 uppercase letter. For example, ABCD:-E.

Name	Value	Description
TIME HHMM-SS	`^(\\d\\d:\\d\\d:\\d\\d)`	Time in a HHMM-SS format, such as 23:59:5-9 or 08:30:0-0.
UK POSTA-L CD	`^[A-Z]{1,2}[0-9R][0-9A-Z]? [0-9][ABD-HJLNP-UW-Z]{2}$`	Checks that a string value is in a valid UK postal code format.
URL	`(http|https|www.|ftp)(www.)?[a-zA-Z0-9@:%._ \\+~#?&//=]{2,256}\\.[a-z]{2,6}\\b([-a-zA-Z0-9@:%._ \\+~#?&//=]*)`	Checks that a string value is in a valid website URL format.

Name	Value	Description
VIN	`^(?<wmi>[A-HJ-NPR-Z\\d]{3})(?<vds>[A-HJ-NPR-Z\\d]{5})(?<check>[\\dX])(?<vis>(?<year>[A-HJ-NPR-Z\\d])(?<plant>[A-HJ-NPR-Z\\d])(?<seq>[A-HJ-NPR-Z\\d]{6}))$`	Checks if a value does not match a US Vehicle Identification Number format.
VISA	`^(?<visa>4[0-9]{12}(?:[0-9]{3})?)$`	Validates Visa credit card numbers.
ZIP CODE	`^[0-9]{5}(?:-[0-9]{4})?$`	Must have 5 digits between 0-9 and can have an additional 4 digits.

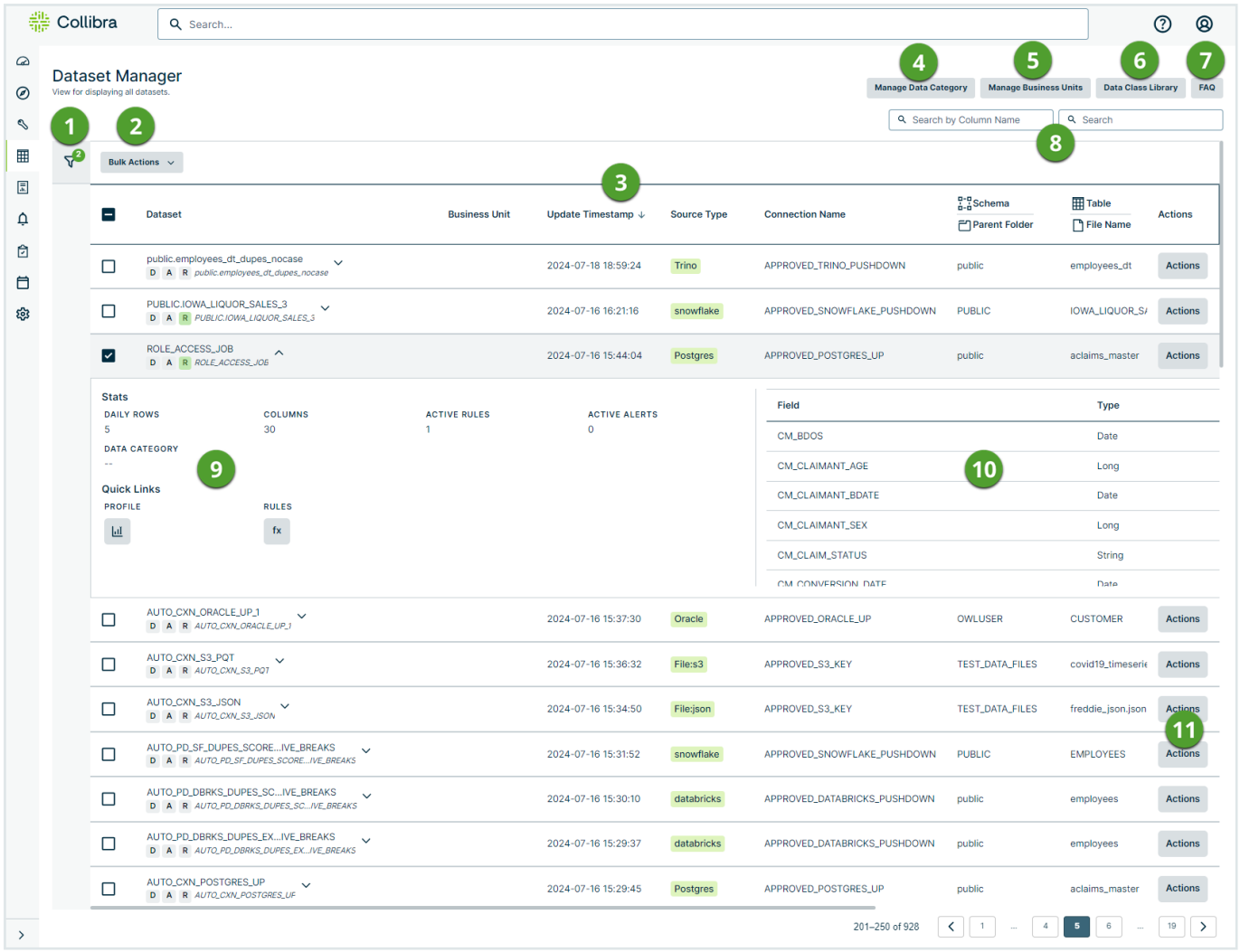
Dataset Directory

This section provides information on the Dataset Manager and Column Manager.

Dataset Manager

Dataset Manager is a repository of all the datasets in your Collibra DQ environment. On this page, you can drill down into individual datasets to review key data points and access their Profile and Dataset Rules pages. You can also filter them based on numerous criteria, apply individual or bulk actions, manage Data Categories and Business Units, and view the Data Class Library. Dataset Manager is a vital tool to help analysts organize and manage the datasets in their organization's Collibra DQ environment.

> **Important**
> When the dataset security setting **Require DATASET_ACTIONS role for dataset rule create/edit access** and **Dataset security** are enabled from the security settings page, users with ROLE_DATASET_ACTIONS are granted permission to edit, rename, publish, assign data categories and business units, enable integrations, and schedule DQ Jobs from the Dataset Manager. These permissions are also accessible by users with the ROLE_ADMIN and ROLE_DATASET_MANAGER roles.


Collibra
Search...
Dataset Manager
View for displaying all datasets.
Manage Data Category
Manage Business Units
Data Class Library
FAQ
Search by Column Name
Search
Bulk Actions
Dataset
Business Unit
Update Timestamp
Source Type
Connection Name
Schema
Parent Folder
Table
File Name
Actions
public.employees_dt_dupes_nocase
2024-07-18 18:59:24
Trino
APPROVED_TRINO_PUSHDOWN
public
employees_dt
PUBLIC.IOWA_LIQUOR_SALES_3
2024-07-16 16:21:16
snowflake
APPROVED_SNOWFLAKE_PUSHDOWN
PUBLIC
ROLE_ACCESS_JOB
2024-07-16 15:44:04
Postgres
APPROVED_POSTGRES_UP
public
aclaims_master
Stats
DAILY ROWS
5
COLUMNS
30
ACTIVE RULES
1
ACTIVE ALERTS
0
DATA CATEGORY
--
Quick Links
PROFILE
RULES
Field
Type
CM_BDOS
Date
CM_CLAIMANT_AGE
Long
CM_CLAIMANT_BDATE
Date
CM_CLAIMANT_SEX
Long
CM_CLAIM_STATUS
String
AUTO_CXN_ORACLE_UP_1
2024-07-16 15:37:30
Oracle
APPROVED_ORACLE_UP
OWLUSER
CUSTOMER
AUTO_CXN_S3_PQT
2024-07-16 15:36:32
File:s3
APPROVED_S3_KEY
TEST_DATA_FILES
AUTO_CXN_S3_JSON
2024-07-16 15:34:50
File:json
freddie_json.json
AUTO_PD_SF_DUPES_SCORE...IVE_BREAKS
2024-07-16 15:31:52
EMPLOYEES
AUTO_PD_DBRKS_DUPES_SC...IVE_BREAKS
2024-07-16 15:30:10
databricks
AUTO_PD_DBRKS_DUPES_EX...IVE_BREAKS
2024-07-16 15:29:37
AUTO_CXN_POSTGRES_UP
2024-07-16 15:29:45
201–250 of 928
1
4
5
6
19
1
2
3
4
5
6
7
8
9
10
11

No.	Component	Description
1	Filter	Apply one or many filters to refine the list of datasets based on: **Sensitivity** Filter by sensitivity labels associated with datasets in your Data Quality & Observability Classic environment. **Run Mode** Filter datasets based on whether they are in published or draft status. **Connection Type** Filter datasets by Pushdown or Pullup processing mode. **Attributes** Filter datasets that only contain rules without alerts or alerts without rules. **Source Type** Filter datasets by data source. **Data Class Type** Filter datasets by data class. **Business Units** Filter datasets by business unit. **Data Categories** Filter datasets by data category. **Row Count** Filter datasets by the number of rows they contain. **Column Count** Filter datasets by the number of columns they contain **Integrations** Filter datasets based on whether they are integrated into Collibra Platform. **# of DQ Scans** Filter datasets by the number of times they have been run as DQ Jobs.

No.	Component	Description
2	Bulk Actions	Select the checkbox options in the column to the left of the Dataset column, then click Bulk Actions and select an option from the dropdown menu. The available Bulk Actions options include: **Bulk Manage Host** Allows admin users to update the host URL of multiple datasets at once. Select the checkbox options next to the datasets you want to manage, then click Bulk Manage Host from the Bulk Actions drop-down menu. From the Bulk Manage Host dialog, enter the new host URL and click Save. **Bulk Manage Agent** Allows admin users to update the agent of multiple Pullup datasets at once. Select the checkbox options next to the datasets you want to manage, then click Bulk Manage Agent from the Bulk Actions drop-down menu. From the Bulk Manage Agent dialog, select the new agent from the dropdown menu and click Save. **Important** Updating the host of multiple datasets at once applies changes to both scheduled and unscheduled DQ Jobs. The connection of any dataset included in the bulk update is mapped to agent you select. **Bulk Manage Spark Settings** Allows admin users to update the Spark settings of multiple Pullup datasets at once. Select the checkbox options next to the datasets you want to manage, then click Bulk Manage Spark Settings from the Bulk Actions dropdown menu. From the Bulk Manage Spark Settings dialog, fill out the required fields and click Save. Show fields

Option	Description
Number of Executors	The default number of executors allocated per DQ Job. The default is 1.
Driver Memory	The default driver RAM allocated per DQ Job. The default is 1 gigabyte.
Executor Memory	The default RAM per executors allocated per DQ Job. The default is 1 gigabyte.

No.	Component	Description

Option	Description
Executor Cores	The default number of cores per executors allocated per DQ Job. The default is 1.
Conf	Additional `spark-submit` parameters appended to the DQ Job run command.
Master	The Spark Master URL copied from the Spark cluster verification screen. For example, `spark://...`
Deploy Mode	The Spark deployment mode can be either **Client** or **Cluster**. While we recommend **Cluster**, there are best practices to follow: • If you only have one Spark Worker node, it is best practice to select **Client**. • If you have more than one Spark Worker node, it is best practice to select **Cluster**.

Bulk Manage Business Units

Allows you to apply a business unit to multiple datasets at once. Select the checkbox options next to the datasets you want to manage, then select Bulk Manage Business Units from the Bulk Actions dropdown menu. From the dropdown menu, select your preferred business unit and click Save.

> **Note** To assign a business unit to a dataset, you need to have dataset access or ROLE_DATASET_ACTIONS.

Bulk Manage Data Categories

Allows you to apply a data category to multiple datasets at once. Select the checkbox options next to the datasets you want to manage, then select Bulk Manage Data Categories from the Bulk Actions dropdown menu. From the dropdown menu, select your preferred data category and click Save.

No.	Component	Description

> **Note** To assign a data category to a dataset, you need to have dataset access, ROLE_DATASET_TRAIN, or ROLE_DATASET_ACTIONS.

Bulk Delete

Allows you to delete multiple datasets at once. Select the checkbox options next to the datasets you want to delete, then select Bulk Delete from the Bulk Actions dropdown menu.

> **Note** To delete a dataset, you need to be its owner or have ROLE_ADMIN or ROLE_DATASET_MANAGER.

Bulk Enable Integrations

Allows you to update the host URL of multiple datasets at once. Select the checkbox options next to the datasets you want to manage, click Bulk Enable Integrations from the Bulk Actions dropdown menu, then click Enable integrations.

> **Note** To toggle a dataset integration, you need to have ROLE_ADMIN, ROLE_DATASET_MANAGER, ROLE_DATASET_ACTIONS.

No.	Component	Description
3	Dataset table	This table lists all datasets in your Collibra DQ environment and provides the high-level data points described in the following table.

Column	Description
☐	Click these options to select one or more datasets to allow you to apply bulk actions. You can also select ☐ in the column header to select all datasets.
Dataset	The name of your dataset. You can also see the following in this column: • When a dataset is in draft status, a gray D displays beneath the dataset name. • When a dataset is in published status, a green P displays beneath the dataset name. • When a dataset does not have alerts configured, a gray A displays beneath the dataset name. • When a dataset has alerts configured, a red A displays beneath the dataset name. • When a dataset does not have rules configured, a gray R displays beneath the dataset name. • When a dataset has rules configured, a green R displays beneath the dataset name. • Click to the right of the dataset name to drill down into the dataset.
Business Unit	The business unit with which the dataset is associated.
Update Timestamp	The timestamp of the last run of the dataset in YYYY-MM-DD hh-mm-ss format.
Source Type	The data source of the dataset. For example, SQL Server or BigQuery.
Connection Name	The unique name of the data source. For example, EXAMPLE_SQLSERVER_CXN.

No.	Component	Description

Column	Description
Schema/Parent Folder	The schema or parent folder from which the data-set originates.
Table/File Name	The table or file name from which the dataset was created.
Meta Tags	The meta tags with which the dataset is associated. Optionally click **Actions** ▸ **Edit**, then enter meta tags in the **Meta Tags** input fields. **Note** Meta tags have a 40 character limit.
Server/File Path	The server or file path associated with the origin of your dataset.
Actions	Click Actions to take a variety of actions on your dataset. Available actions include: • **Edit** your dataset. • **Edit Job** to edit your dataset. • **Clone Dataset** to create a new replica of your dataset. • **Rename** your dataset. • **Publish** your dataset or revert it **To Draft**. • Assign a **Data Category** to your dataset. • Assign a **Business Unit** to your dataset. • **Enable** or **Disable Integrations** of your dataset into Collibra Platform. • **Delete** your dataset. **Note** You can also create a clone of the dataset from the Findings page by using the **Clone Dataset** option on the Job tab.

No.	Component	Description
4	Manage Data Category	Opens the Data Categories page in the Admin Console where you can create and manage data categories. **Note** ROLE_ADMIN or ROLE_DATA_GOVERNANCE_MANAGER is required to access the Admin Console.
5	Manage Business Units	Opens the Business Units page in the Admin Console where you can create and manage business units. **Note** ROLE_ADMIN or ROLE_DATA_GOVERNANCE_MANAGER is required to access the Admin Console.
6	Data Class Library	Opens the Data Class Library modal where you can view and search the data classes in your Collibra DQ environment.
7	FAQ	Opens the Top Questions and Answers modal where a sample of common dataset management questions appear.
8	Search	There are two search options. The first search field on the left lets you search for datasets containing columns that match your search criteria. The second search field on the right lets you search for items that match your criteria based on dataset name, source type, schema/parent folder, or table/-file name.

No.	Component	Description
9	Dataset drill down	Click to the right of the dataset name to drill down into the dataset and reveal the following high-level data points.

Field	Description
Stats	
Daily Rows	The number of rows scanned in the last run.
Columns	The number of columns in the dataset.
Active Rules	The number of active rules included in the last run.
Active Alerts	The number of alerts configured to notify specified users when their conditions were met in the last run.
Data Category	The data category assigned to your dataset on the Rule Workbench or Metadata Bar.
Quick Links	
Profile	Click the  to open the Profile page of your dataset.
Rules	Click the  to open the Dataset Rules page of your dataset.
Description	The description of your dataset given to it on the Explorer ▸ Review page.

No.	Component	Description
10	Column overview	This section of the dataset drill down shows the following column-level details of all columns in your dataset.

Column	Description
Field	The name of the column.
Type	The data type of the column.
Data Class	The data class labels of a column, when applied.
Sensitive Label	The sensitive label to obscure sensitive information, when applied.

No.	Component	Description
11	Actions	Click Actions to take a variety of actions on your dataset. Available actions include: • **Edit** your dataset details. > **Note** To edit the alias, schema, table or file name, server or file path, or meta tags of a dataset, you need to have the proper role. Source Type is a read-only field. > • Users with ROLE_ADMIN can edit the Alias, Schema/Parent Folder, Table/File Name, and Server/File Path. > • Users with ROLE_DATA_GOVERNANCE_MANAGER or ROLE_DATASET_ACTIONS can edit Meta Tags. > **Important** When you edit a dataset from the Dataset Manager, the dataset definition is updated in the and shown in all relevant locations in the application, such as on the Job tab. • **Edit Job** to edit your dataset. • **Clone Dataset** to create a new replica of your dataset. > **Note** To clone a dataset, you need to be its owner or have ROLE_OWL_CHECK and be assigned to the dataset. • **Rename** your dataset. > **Note** To rename a dataset, you need to be its owner or have ROLE_ADMIN or ROLE_DATASET_MANAGER. • **Publish** your dataset or revert it **To Draft**. > **Note** To publish a dataset, you need to have dataset access, ROLE_DATASET_TRAIN, or ROLE_DATASET_ACTIONS. • Assign a **Data Category** to your dataset. > **Note** To assign a data category to a dataset, you need to have dataset access, ROLE_DATASET_TRAIN, or ROLE_DATASET_ACTIONS. • Assign a **Business Unit** to your dataset.

No.	Component	Description

> **Note** To assign a business unit to a dataset, you need to have dataset access or ROLE_DATASET_ACTIONS.

- **Enable** or **Disable Integrations** of your dataset into Collibra Platform.

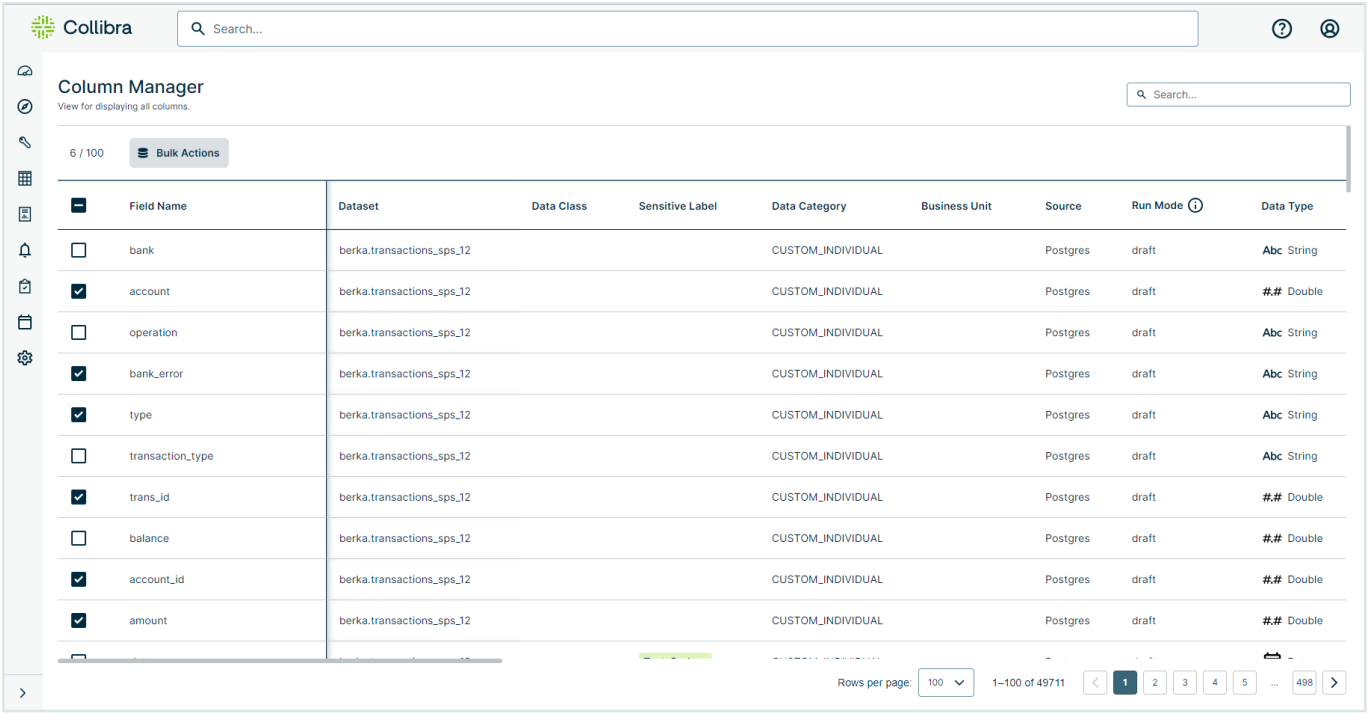
  > **Note** To toggle a dataset integration, you need to have ROLE_ADMIN, ROLE_DATASET_MANAGER, ROLE_DATASET_ACTIONS.

- **Delete** your dataset.

  > **Note** To delete a dataset, you need to be its owner or have ROLE_ADMIN or ROLE_DATASET_MANAGER.

Column Manager

Column Manager is a view for displaying all columns associated with datasets with prior runs in your environment, and provides data analysis of common data points in columns, such as mean, min, and max absolute values, whether a column is passing or failing, and more. Column manager also gives you the flexibility of applying actions to columns in bulk, removing the need to individually create data class rules against columns or tag them as containing sensitive information.



The following table shows the columns on the Column Manager table and their descriptions. From the Column Manager UI, use the horizontal scroll bar at the bottom of the page to reveal additional columns that may not be visible depending on your screen size.

Column	Description
☐	Click these options to select one or more datasets to allow you to apply bulk actions. You can also select ☐ in the column header to select all datasets.
Field Name	The name of the column in a dataset.
Dataset	The name of the dataset where the column is located. You can click the dataset name to open its Findings page.
Data Class	The Data Class applied to a column, when assigned.
Sensitive Label	The Sensitive Label applied to a column, when assigned.
Data Category	The Data Category applied to a column, when assigned.
Business Unit	The Business Unit assigned to a column, when applicable.
Source	The data source from which the data originates.

Column	Description
Run Mode	Indicates whether the dataset is published or is still in draft state.
Data Type	The data type of a column.
Null Ratio	The percentage of values that are null in a given column, where 1.00 is 100%.
Empty Ratio	The percentage of values that are empty in a given column, where 1.00 is 100%.
Unique Ratio	The percentage of values that are unique in a given column, where 1.00 is 100%.
Unique Count	The number of unique values in the baseline profile of a column.
Type Ratio	The percentage of values that Collibra DQ observes as non-numeric (or numeric) in a numeric (or non-numeric) column, where 1.00 is 100%.
Mean Abs	The average of the absolute values of all the data points in a column. This is calculated by summing up the absolute values of each data point, then dividing by the total number of data points in the column.
Min Abs	The data point in a column with the smallest absolute value.
Max Abs	The data point in a column with the largest absolute value.
Pass/Fail	The Passing, Breaking, or Learning status of AdaptiveRules from your most recent scan.
Min Abs Num	A numeric type data point in a column with the smallest absolute value.
Max Abs Num	A numeric type data point in a column with the largest absolute value.
Mean Abs Num	The average of the absolute values of all the numeric type data points in a column.
Min Precision	The minimum number of digits that a numeric data type value can be.
Max Precision	The maximum number of digits that a numeric data type value can be.
Min Scale	The minimum number of digits that the fractional part of a decimal number can be.
Max Scale	The maximum number of digits that the fractional part of a decimal number can be. For example, if a decimal number is 139.745 and the max scale is 2, then the fractional part will round to .75 to adhere to the max scale of 2 digits. The full value would then be 139.75.

Column	Description
Update Timestamp	The timestamp of the last run of the dataset in YYYY-MM-DD hh-mm-ss format.
Run Id	The timestamp of the dataset's initial job run.

Bulk Manage Rules

1. In the far left column of the Column Manager table, click the header-level checkbox to select all rows on the page or the checkbox of individual rows to only include those columns.
2. Click Bulk Actions near the upper left corner, then click **Bulk Manage Rules**.
3. Select one or more options from the Rules dropdown menu.
   » The Column rule update modal appears.

   > **Note** The Column rule update modal includes a preview of all the rows you selected in the first step.

4. Click **Apply**.
   » The rules will be checked against your dataset the next time your job runs and are available on the Dataset Rules page.

Bulk Manage Data Class

1. In the far left column of the Column Manager table, click the header-level checkbox to select all rows on the page or the checkbox of individual rows to only include those columns.
2. Click Bulk Actions near the upper left corner, then click **Bulk Manage Data Class**.
3. Select one or more options from the Rules dropdown menu.
   » The Column data class update modal appears.

   > **Note** The Column data class update modal includes a preview of all the rows you selected in the first step.

4. Click **Apply**.
   » The data class rules will be checked against your dataset the next time your job runs

and are available as a badge on Column Manager and to view or edit on the Dataset Rules page.

Bulk Manage Sensitive Label

1. In the far left column of the Column Manager table, click the header-level checkbox to select all rows on the page or the checkbox of individual rows to only include those columns.
2. Click Bulk Actions near the upper left corner, then click **Bulk Manage Sensitive Label**.
3. Select one or more options from the Rules dropdown menu.
   » The Column sensitive label update modal appears.

   > **Note** The Column sensitive label update modal includes a preview of all the rows you selected in the first step.

4. Click **Apply**.
   » The column with the sensitive label will be is flagged as containing sensitive information and a badge displays in that column's row on Column Manager.

Reports

Completeness Report

This section provides information on how to generate a Collibra DQ Completeness Report to determine what percentage of your data is complete.

What is Data Completeness?

Completeness answers the question of what percentage of your data is complete, or "filled in" (i.e., not `EMPTY` or `NULL`). Using this report, you can view the completeness of a column, a collection of tables, a business unit or data set (file or table), or almost any completeness query.

Completeness of your data is important because it shows whether data is inaccurate, invalid, the wrong type, or missing altogether, which can leave you without any data. Sometimes data values can be missing for valid reasons, which requires a better understanding of the context of whether the missing data is bad for business or acceptable.

You can generate the Completeness report from the Reports page.

Steps

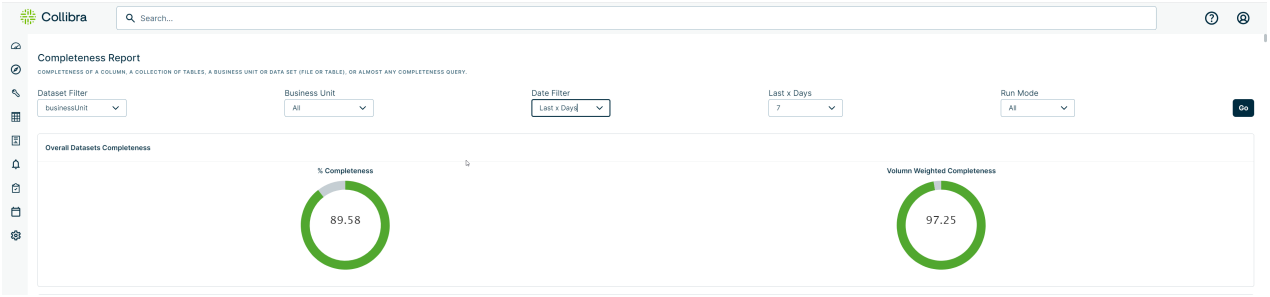
1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. Click the **Completeness Report** link.

3. From the Dataset Filter dropdown list, select one of the following options:
    - **Business Unit**
        i. From the Business Unit dropdown list, select a specific business unit or **All**, to select all datasets.
        ii. Proceed to step 4.
    - **Dataset**
        i. In the Dataset search field, enter a dataset for which you want to run the report. You can also enter a partial word to locate data sets in the system.
        ii. Proceed to step 4.
4. From the Date Filter dropdown list, select one of the following options:
    - **Last x Days**: the number of days back to include in the report.
        i. From the Last x Days dropdown list, select one of the following options:
            - **1 day**
            - **2 days**
            - **7 days**
            - **10 days**
            - **15 days**
        ii. Proceed to step 5.
    - **Custom Range**: the custom range you set for Run Date/Update Time.
        i. From the Date Column dropdown list, select one of the following options:
            - **Run Date**: the date/time the data represents.
            - **Update Time**: time the DQ job ran.
        ii. In the Date Range fields, select a date range by clicking in the from/to fields and selecting the dates using the interactive calendars.
        iii. Proceed to step 5.
5. From the Run Mode dropdown list, select one of the following options:
    - **All**: includes all jobs in the system.
    - **DRAFT**: includes only the jobs that are in draft mode.
    - **PUBLISHED**: includes the jobs that have been published.
6. Click **Go**.

> **Note** When looking at completeness over time, you should differentiate between the time the DQ job ran (update time) or the date/time the data represents (run date). For example, you could load stock data today but the data loaded was for last week.

All View

The **All** view represents the completeness of datasets throughout the entire Collibra DQ app. The **% Completeness** chart measures all the data, which shows around 90% complete in the following example. The **Volume Weighted Completeness** chart also measures the volume of the data, which shows around 97%.



Column View

The column view shows the completeness of specific datasets, which makes it easy to see the columns that are least complete and, therefore, of possible concern. The columns range from 0% to 100% complete.



FieldNm	% completeness	volume weighted completeness
Employee_LeaveDate	31.8	31.8
Employee_Office	100	100
Report_Date	100	100
Employee_SSN	100	100
Employee_Email	100	100
Employee_Division	100	100
Employee_AddressStreet	100	100

Volume Weighted Completeness and % Completeness example

To better understand how the Volume Weighted Completeness and % Completeness charts differ, consider the following example.

In the following table, there is a sample of customers who called the example company and created an issue with the support team.

ID	Customer Name	Phone Number	Created on (date)
1	Noah	123-456-7890	2024-01-01
2	Oliver	123-456-7891	2024-01-01
3	Stephanie		2024-01-02
4	Arturo		2024-01-02
5	Lisa		2024-01-02
6	Irene	123-456-7892	2024-01-02
7	Chidi	123-456-7893	2024-01-02
8	Miles	123-456-7894	2024-01-02

If a scheduled job runs with the query `select * from public.users where created_at = ${rd}`, the completeness data for the column "Phone Number" will be:

runId	Completeness %	Row Count (rc)
2024-01-01	100	2
2024-01-02	50	6

The calculation of the Completeness % of the "Phone Number" column based on the above data is an average value:

Completeness = (100 + 50) / 2 = 75 (%)

Whereas, the calculation of the Volume Weighted is a mean value:

Completeness weighted = (100 * 2 + 50 * 6) / (2 + 6) = 62.5 (%)

Behavioral Analytics for Completeness

For a different approach to completeness management, see the Collibra DQ Behavior feature. This approach uses the data itself to create baselines and profiles to understand which completeness issues matter and, therefore, require you to take some kind of action.

To generate statistical process around completeness in the events you are most concerned about, and alert you to a change in slope (a drastic change in completeness), see the DQ Job Back Run and Profile (automatic) features.

Dataset Findings

What is the Dataset Findings Report?

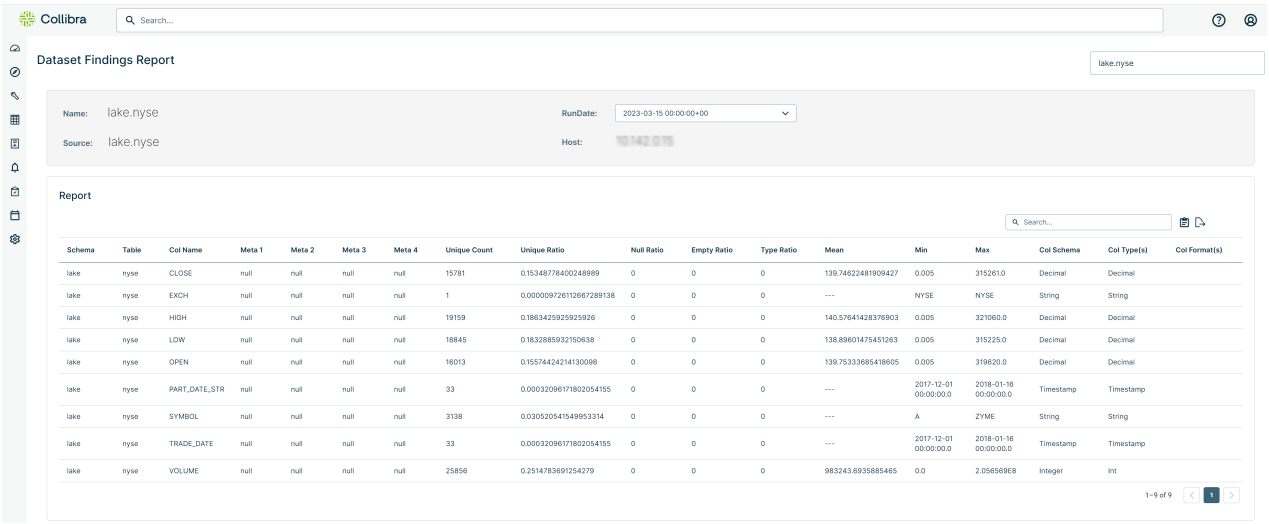
The Data Quality Dataset Findings Report allows you to search for a particular dataset and generate, view, copy, print, and export the report to an Excel or CSV file format.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. Select the **Dataset Findings** link.
3. In the search bar in the upper right corner, enter the name of your target dataset.
   A list of available datasets populates in the drop-down menu.
4. Select a **dataset**.
   The Dataset Findings Report page displays for your dataset.
   The report includes the name, source, Run Date, and host of your dataset, as well as columns that highlight specific data about it.
5. Select the RunDate from the dropdown list to display the data in the table for that particular RunDate.
6. Toggle the ↑ ↓ icons at the top of the columns to sort the data that displays in the columns in ascending or descending order.

7. Filter the report results by entering information in the search bar. For example, if you enter a number in the search field, any report result that includes the number displays.
8. To make a copy of the reports, click the icon in the top right of the columns. The rows are copied to the clipboard.
9. To export the reports to CSV, Excel, PDF, and to print the reports, click the icon and select the desired option.
10. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located bottom-right of the columns.

The following screenshot is an example of a Dataset Findings Report for the lake.nyse dataset.



Meta fields can be used to classify datasets. You can assign Meta tags by clicking **Edit** in the Actions dropdown for a dataset in the DQ Catalog. See Collibra DQ Catalog for more information.

The meta tags you added populate in the Meta Tags field.

Summary Reports

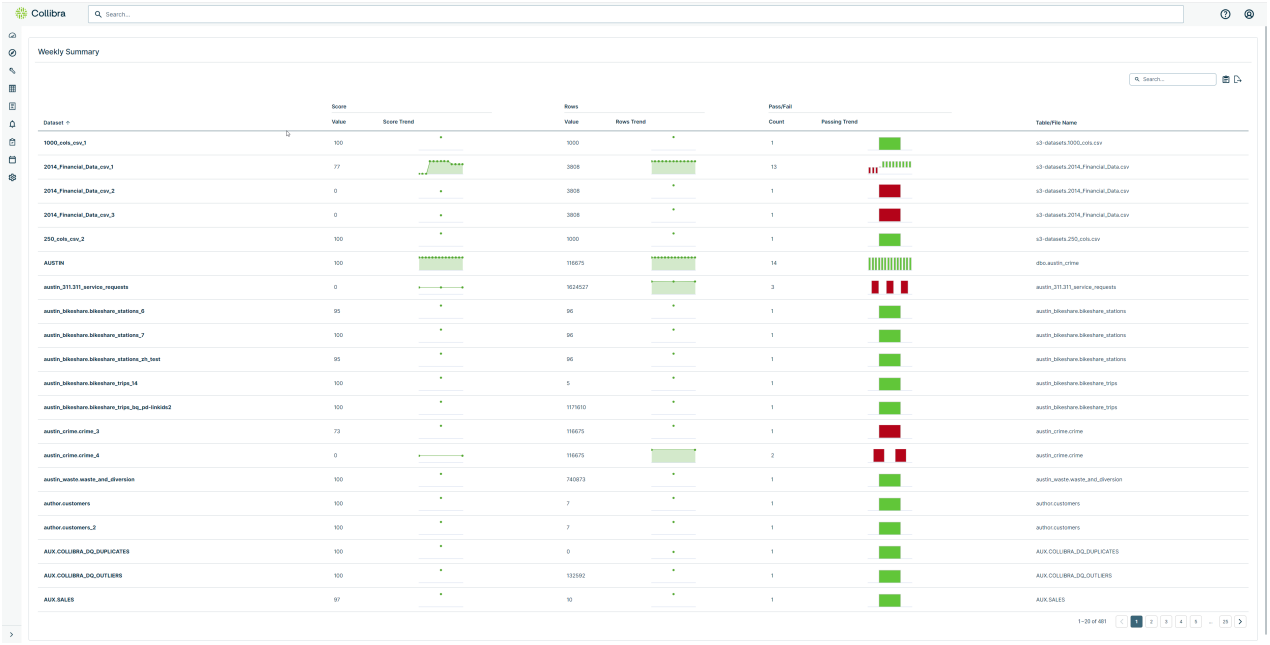
Collibra Data Quality Summary reports provide information about your datasets rolled up in high level summaries. These reports include the What is the Weekly Summary Report? and What is the Data Quality Checks Report?.

> **Note** These reports provide a look at the most simplistic form of the rule. However, this view does not associate itself with a run's score.

What is the Weekly Summary Report?

When you operate a large data lake or several large data environments, it's helpful to have a way to report across different dimensions at an executive summary level. You may want to know the health or coverage per line of business, department or tenet, or per database.

The Collibra Data Quality Weekly Summary Report automatically aggregates a simple series of the high level trends for each dataset, which allows you to see the DQ scores and trends, as well as row counts and passing runs in a weekly report. You can copy, export, and print this report to an Excel, CSV, or PDF file format.



Steps

To generate a Weekly Summary report, follow these steps.

1. Sign in to the Collibra DQ instance and click the Reports icon in the left navigation pane.
   » The Reports page opens.

2. Select the **Weekly Summary** link.
   The Weekly Summary report displays.

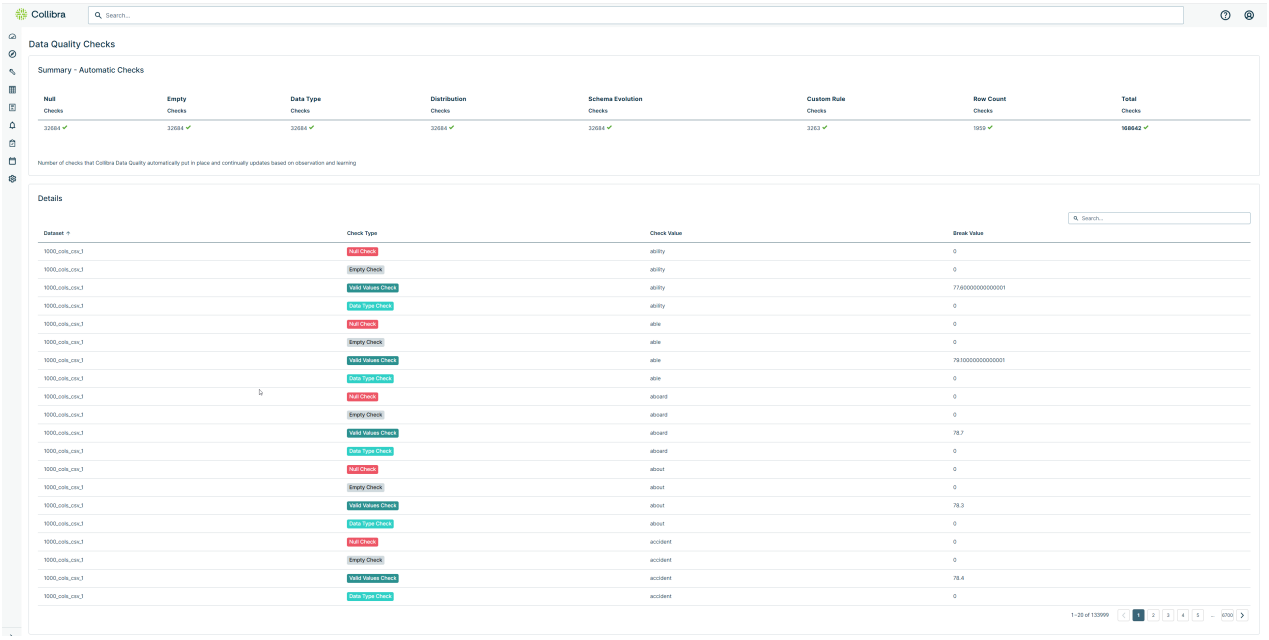
The following table describes the report columns.

Column	Description
Dataset	The datasets scanned over one week.
Score	The average data quality score of a dataset over one week.
Score Trend	A line graph representation of data quality scores over one week. Hover over the blue dots to see the score trend.
Rows	The total number of rows of a particular dataset scanned over one week.
Rows Trend	A line graph representation of row count over one week. Hover over the blue dots to see the rows trend.
Pass/Fail	Total number of DQ scans, whether they pass or fail.
Passing Trend	A histogram of DQ scans that passed (blue) or failed (red) over one week. Hover over the blue dots to see the passing trend.
Table/File Name	The name of the table or file in use.

3. Toggle the ↑ ↓ icons at the top of the column to sort the data that displays in the columns in ascending or descending order.
4. Filter the report results by entering information in the search bar. For example, if you enter a number in the search field, any report result that includes that number displays.
5. To make a copy of the reports, click the  icon in the top right of the columns. The rows are copied to the clipboard.
6. To export the reports to CSV, Excel, PDF, and to print the reports, click the  icon and select the desired option.
7. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located at the bottom of the columns.

What is the Data Quality Checks Report?

The Data Quality Checks Report provides a number of automatic checks of your datasets that are continually updated, based on observation and learning. This report includes the type of check performed and the check value and break value per dataset. The top portion of this report rolls up the highest results of the checks.



> **Note** The Schema Evolution, located in the automatic checks summary at the top of the report, is a schema finding that shows how the schema has evolved from the last run.

Steps

To see the Data Quality Checks report, follow these steps.

1. Sign in to the Collibra DQ instance and click the Reports icon in the left navigation pane.
   » The Reports page opens.
2. Select the **DQ Check Summary** link.
   The Data Quality Checks report displays.

This report includes a summary of the number of checks that Collibra DQ automatically ran at the top of the report and the details, which are described in the following table.

Column	Description
Dataset	The name of the dataset.
Check Type	The type of check that DQ ran.
Check Value	The value associated with this check.
Break Value	The number of points associated with a rule break, which are then subtracted from a data quality score.

3. Toggle the ↑ ↓ icons at the top of the column to sort the data that displays in the columns in ascending or descending order.
4. Filter the report results by entering information in the search bar. For example, if you enter a number in the search field, any report result that includes that number displays.
5. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located at the bottom right of the columns.

Missing Jobs Report

What is a Missing Jobs Report?

The Missing Jobs report provides a view of Collibra DQ jobs that were expected but didn't run on schedule.

You can generate a Missing Jobs report from the Reports page.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.

2. From the Reports page, select the **Missing Jobs Report** link.
3. To configure the type of chart configuration and data you want to see in the chart, click the ⋮ ellipsis icon.
4. Click  **Chart Configuration**.
   The Chart Configuration window displays.

   > **Note** As a best practice, keep the Chart Configuration as the default.

5. For x-axis, select one of the following options from the X-Axis Column dropdown list:
   - **count**
   - **updt_ts**
6. For y-axis, select one or more of the following options from the Y-Axis Columns dropdown list:
   - **count**
   - **updt_ts**
7. For the type of chart you want to display, select one of the following options from the Chart Type dropdown list:
   - **Line**
   - **Area**
   - **Column**
   - **Scatter**
8. Click **Save**.
9. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a Missing Jobs report.

```
SELECT jsa.updt_ts::date AS updt_ts,
        jsa.dataset,
        bus.name AS business_unit,
        js.freq,
        js.active,
        js.mon,
        js.tue,
        js.wed,
        js.thu,
        js.fri,
        js.sat,
        js.sun
    FROM job_schedule_attempt jsa
    JOIN job_schedule js ON js.dataset::text = jsa.dataset::text
    LEFT JOIN dataset_scan ds ON ds.dataset::text = jsa.data-
set::text AND ds.updt_ts::date = jsa.updt_ts::date
    LEFT JOIN business_unit_to_dataset bu ON jsa.dataset::text =
bu.dataset::text
    LEFT JOIN business_units bus ON bus.id = bu.id
    WHERE ds.updt_ts IS NULL AND jsa.updt_ts >= (now() - '20
days'::interval)
    ORDER BY (jsa.updt_ts::date) DESC
```

Hardware Usage Report

What is a Hardware Usage Report?

The Hardware Usage report provides a view of the Collibra DQ jobs that have consumed the most hardware.

You can generate a Hardware Usage report from the Reports page.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, click the **Hardware Usage Report** link.

3. For x-axis, select one of the following options from the dropdown:
    - num_executors
    - executor_memory
    - total_memory
    - total_cores
    - executor_cores
    - avg_row_count
    - dataset
4. For y-axis, select one or more of the following options:
    - num_executors
    - executor_memory
    - total_memory
    - total_cores
    - executor_cores
    - avg_row_count
    - dataset
5. For the type of chart you want to display, select one of the following options from the drop-down:
    - Line
    - Area
    - Column
    - Scatter
6. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a Hardware Usage report.

```
with most_current_dataset_scan as ( select dataset, avg(rc)::Int
as avg_row_count from dataset_scan group by dataset )

     select opt.dataset, num_executors, executor_cores,
executor_memory, (executor_cores * num_executors) total_cores,
            (NULLIF(regexp_replace(executor_memory, '\D', '',
'g'), '')::numeric * num_executors) as total_memory,
            ds.avg_row_count
     from opt_spark opt, most_current_dataset_scan ds
     where opt.dataset = ds.dataset
     order by total_cores desc limit 75
```

Oversized Job Report

What is an Oversized Job Report?

The Oversized Job report provides a view of the Collibra DQ jobs that have more hardware assigned than is available to run. For example, say you have a scenario where 20 jobs all request 100 servers to kick off at the same time, but 2,000 servers are not available to run. The Oversized Job report identifies those 20 jobs, which are oversized. Those jobs can now be resized to the optimal amount of executors, cores, ram, etc.

Steps

To generate an Oversized Job report, follow these steps.

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, click the **Oversized Job Report** link.
3. To configure the type of chart configuration and data you want to see in the chart, click the ⋮ ellipsis icon.
4. Click  **Chart Configuration**.
   The Chart Configuration window displays.

> **Note** As a best practice, keep the Chart Configuration as the default.

5. For x-axis, select one of the following options from the X-Axis Column dropdown list:
    - dataset
    - reason
    - recommended_total_memory
    - recommended_total_cores
    - recommended_num_executors
    - recommended_executor_memory
    - num_executors
    - executor_cores
    - executor_memory
    - total_cores
    - total_memory
    - avg_row_count
    - column_count
    - avg_total_time_in_minutes
    - cell_count
6. For y-axis, select one or more of the following options from the Y-Axis Columns dropdown list:
    - dataset
    - reason
    - recommended_total_memory
    - recommended_total_cores
    - recommended_num_executors
    - recommended_executor_memory
    - num_executors
    - executor_cores
    - executor_memory
    - total_cores
    - total_memory
    - avg_row_count
    - column_count
    - avg_total_time_in_minutes
    - cell_count

7. For the type of chart you want to display, select one of the following options from the Chart Type dropdown list:
    - **Line**
    - **Area**
    - **Column**
    - **Scatter**
8. Click **Save**.
9. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL queries for an Oversized Job report.

```
DROP VIEW IF EXISTS report_oversized_jobs;
CREATE OR REPLACE VIEW report_oversized_jobs AS
(
WITH most_current_dataset_scan AS (
    SELECT dataset_scan.dataset,
           avg(dataset_scan.rc)::integer AS avg_row_count
    FROM dataset_scan
    GROUP BY dataset_scan.dataset
),
     dataset_schema_col AS (
         SELECT dataset_schema.dataset,
                count(*) AS column_count
         FROM dataset_schema
         GROUP BY dataset_schema.dataset
     ),
     dataset_activity_time AS (
         SELECT dataset_activity.dataset,
                round(avg(dataset_activity.total_time_in_
minutes), 2) AS avg_total_time_in_minutes
         FROM dataset_activity
         GROUP BY dataset_activity.dataset
     ),
     highest_hardware_usage AS (
         SELECT opt.dataset,
                opt.num_executors,
                opt.executor_cores,
                opt.executor_memory,
                opt.executor_cores * opt.num_executors AS total_
cores,
                NULLIF(regexp_replace(opt.executor_memory::text,
'\D'::text, ''::text, 'g'::text), ''::text)::numeric * opt.num_
executors::numeric AS total_memory,
                ds.avg_row_count,
                sch.column_count,
                dat.avg_total_time_in_minutes,
                ds.avg_row_count * sch.column_count AS cell_
count
         FROM opt_spark opt,
              most_current_dataset_scan ds,
              dataset_schema_col sch,
              dataset_activity_time dat
         WHERE opt.dataset::text = ds.dataset::text AND sch.data-
set::text = ds.dataset::text AND dat.dataset::text = ds.data-
set::text AND opt.executor_memory IS NOT NULL AND opt.executor_
memory::text NOT LIKE ''::text
         ORDER BY (opt.executor_cores * opt.num_executors) DESC
     )
```

```
SELECT
    highest_hardware_usage.dataset AS dataset,
    CASE
        WHEN (highest_hardware_usage.cell_count::numeric /
highest_hardware_usage.total_memory) > 20000000::numeric THEN
'Too Much Memory'::text
        WHEN (highest_hardware_usage.cell_count / highest_hard-
ware_usage.total_cores) > 190000000 THEN 'Too Many Cores'::text
        ELSE 'Too Much Memory & Cores'::text
        END AS reason,
    GREATEST(highest_hardware_usage.cell_count / 20000000, 1::-
decimal) AS recommended_total_memory,
    GREATEST(highest_hardware_usage.cell_count / 119000000, 1::-
decimal) AS recommended_total_cores,
    GREATEST(highest_hardware_usage.cell_count / 119000000 / 2,
1::decimal) AS recommended_num_executors,
    GREATEST(highest_hardware_usage.cell_count / 20000000, 1::-
decimal) / GREATEST(highest_hardware_usage.cell_count /
119000000 / 2, 1::decimal) AS recommended_executor_memory,
    highest_hardware_usage.num_executors,
    highest_hardware_usage.executor_cores,
    highest_hardware_usage.executor_memory,
    highest_hardware_usage.total_cores,
    highest_hardware_usage.total_memory,
    highest_hardware_usage.avg_row_count,
    highest_hardware_usage.column_count,
    highest_hardware_usage.avg_total_time_in_minutes,
    highest_hardware_usage.cell_count
FROM highest_hardware_usage
WHERE (
            (highest_hardware_usage.cell_count::numeric /
highest_hardware_usage.total_memory) < 20000000::numeric
        OR (highest_hardware_usage.cell_count / highest_hard-
ware_usage.total_cores) < 119000000)
  AND highest_hardware_usage.executor_cores > 1
  AND highest_hardware_usage.num_executors > 1
    );
```

Observability Score Roll-Up Report

What is an Observability Score Roll-Up Report?

The Observability Score Roll-Up report provides aggregated scores of all adaptive rules (includes all data sets and all columns), as well as averages passing and breaking for all

columns over 30 days. The scores are aggregated by check type and dimension.

You can generate an Observability Score Roll-Up report from the Reports page.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, click the **Observability Score Roll-Up Report** link.
3. To configure the type of chart configuration and data you want to see in the chart, click the ⋮ ellipsis icon.
4. Click  **Chart Configuration**.
   The Chart Configuration window displays.

   > **Note** As a best practice, keep the Chart Configuration as the default.

5. For x-axis, select one of the following options from the X-Axis Column dropdown list:
   - **check_type**
   - **dimension**
   - **avg_observability_score**
6. For y-axis, select one or more of the following options from the Y-Axis Columns dropdown list:
   - **check_type**
   - **dimension**
   - **avg_observability_score**
7. For the type of chart you want to display, select one of the following options from the Chart Type dropdown list:
   - **Line**
   - **Area**
   - **Column**
   - **Scatter**

8. Click **Save**.
9. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for an Observability Score Roll-Up report.

```
with a as ( select dataset, col_nm from dataset_schema ),
     b as ( select distinct dataset, 'Row Count', 'ROW_COUNT',
'AVAILABILITY' from dataset_scan where updt_ts >= (NOW() -
INTERVAL '30 DAY') and updt_ts <= NOW()),
     c as ( select * FROM
               (VALUES
                    ('NULL', 'COMPLETENESS'),
                    ('EMPTY', 'COMPLETENESS'),
                    ('TYPE', 'SCHEMA'),
                    ('CARDINALITY', 'DISTRIBUTION'),
                    ('MIN', 'CONFORMITY'),
                    ('MAX', 'CONFORMITY'),
                    ('MEAN', 'CONFORMITY')) as t (type, dimension)
),
     d as ( select * from a,c  ),
     e as ( select * from d UNION (select * from b)),
     f as ( select dataset as be_dataset, be_score, field,
issue_type, run_id from behavior ),
     g as ( select * from owl_catalog ),
     h as ( select distinct dataset, run_id, rc from dataset_
scan where updt_ts >= (NOW() - INTERVAL '30 DAY') and updt_ts <
NOW()),
     i as ( select e.dataset, col_nm, type, dimension, run_id,
rc from e, h where e.dataset = h.dataset ),
     j as ( select distinct dataset, col_nm, col_semantic
from  dataset_schema ),
     k as ( select * from business_unit_to_dataset ),
     l as ( select * from business_units ),
     m as (
            select i.col_nm, type,
                   case when be_score is null then 100 else be_
score end as score, i.dimension,
                   i.run_id, rc, be_score, issue_type, i.data-
set, g.source_name, g.source, g.table_nm, g.run_mode, g.data_
concept_id, j.col_semantic,
                   l.name
            from i
                LEFT join f on f.be_dataset = i.dataset and
f.field = i.col_nm and f.issue_type = i.type and f.run_id =
i.run_id
                INNER JOIN g on i.dataset = g.dataset
                LEFT JOIN j on i.dataset = j.dataset and i.col_
nm = j.col_nm
                LEFT JOIN k on i.dataset = k.dataset
                LEFT JOIN l on l.id = k.id
            )
    select avg(score) as avg_observability_score, type as check_
type, dimension from m group by type, dimension order by
dimension
```

Rules Passing Fraction Roll-Up Report

What is a Rules Passing Fraction Roll-Up Report?

The Rules Passing Fraction Roll-Up report provides the passing and total rows scanned for user-defined rules over the past 30 days, which are aggregated by dimensions.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, click the **Rules Passing Fraction Roll-Up** link.
3. To configure the type of chart configuration and data you want to see in the chart, click the ellipsis icon.
4. Click  **Chart Configuration**.
   The Chart Configuration window displays.

   > **Note** As a best practice, keep the Chart Configuration as the default.

5. For x-axis, select one of the following options from the X-Axis Column dropdown list:
   - **avg_percent_rows_passing**
   - **dimension**
6. For y-axis, select one or more of the following options from the Y-Axis Columns dropdown list:
   - **dimension**
   - **avg_observability_score**
7. For the type of chart you want to display, select one of the following options from the Chart Type dropdown list:
   - **Line**
   - **Area**

    - **Column**
    - **Scatter**
8. Click **Save**.
9. Navigate the pages of your report by clicking the **Previous** and **Next** pagination buttons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a Rules Passing Fraction Roll-Up report.

```
WITH a AS (
    SELECT * FROM rule_output
    WHERE updt_ts BETWEEN (NOW() - INTERVAL '30 DAY') AND NOW()
),
b AS (
    SELECT * FROM dataset_scan WHERE rc > 1
),
c AS (
    SELECT * FROM owl_rule
),
e AS (
    SELECT * FROM dq_dimension
),
g AS (
    SELECT * FROM owl_catalog
),
h AS (
    SELECT * FROM business_unit_to_dataset
),
i AS (
    SELECT * FROM business_units
),
j AS (
    SELECT DISTINCT dataset, col_nm, col_semantic FROM dataset_
schema
),
f AS (
    SELECT
        a.dataset,
        a.rule_nm,
        CASE
            WHEN c.column_name = '' OR c.column_name IS NULL
THEN 'global'
            ELSE c.column_name
        END AS column_name,
        COALESCE(e.dim_name, 'UNSPECIFIED') AS dim_name,
        c.dim_id,
        ROUND(a.perc) AS perc,
        ROUND(a.perc * b.rc) AS breaking_rows,
        100 - ROUND((a.perc * b.rc) / CAST(b.rc AS DECIMAL)) AS
score,
        b.rc AS row_count,
        a.run_id,
        g.alias,
        g.catalog_rank,
        g.db_nm,
        g.run_mode,
        g.source_name,
```

```
        g.table_nm,
        i.name,
        j.col_nm,
        j.col_semantic
    FROM a
    LEFT JOIN b ON a.dataset = b.dataset AND a.run_id = b.run_id
    INNER JOIN c ON a.dataset = c.dataset AND a.rule_nm =
c.rule_nm
    LEFT JOIN e ON e.dim_id = c.dim_id
    INNER JOIN g ON g.dataset = a.dataset
    LEFT JOIN h ON h.dataset = g.dataset
    LEFT JOIN i ON i.id = h.id
    LEFT JOIN j ON a.dataset = j.dataset AND c.column_name =
j.col_nm
)
SELECT
    CASE
        WHEN AVG(score) < 0 THEN 0
        ELSE AVG(score)
    END AS avg_percent_rows_passing,
    dim_name AS dimension
FROM f
GROUP BY dim_name;
```

User Activity Report

What is a User Activity Report?

The User Activity report provides a view of the number of clicks in the Collibra DQ application by user. You can filter this report by various criteria from a Filter Type dropdown, as well as set a limit to how many users you want to view, based on the filter you choose. You can also configure the kind of chart you want to use to view the data.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, select the **User Activity** link.

3. To display results by filter type, select one of the following options from the **Filter Type** dropdown list:
    - **Last x Days** - filter by last number of days.
        i. Select one of the following options from the **Last x Days** dropdown list:
            - 1 day
            - 2 days
            - 7 days
            - 10 days
            - 15 days
            - 30 days
        ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

        > **Note** The chart and table update, based on your selections.

        iii. Proceed to step 4.
    - **Custom Range** - filter by configuring a custom range.
        i. Select a from and to date range in the Date Range fields.

        > **Tip** You can also select the  icon, which displays a calendar, in which you can select a specific date.

        ii. To limit the number of users that the chart displays, enter a number in the **Limit** field.

        > **Note** The chart and table update, based on your selections.

        iii. Proceed to step 4.

- **All** - filter by username and user clicks.
    i. To limit the number of users that the chart displays, enter a number in the **Limit** field.

    > **Note** The chart and table update, based on your selections.

    ii. Proceed to step 4.

4. Click **Go**.
5. To configure the type of chart configuration and data you want to see in the chart, click the ⋮ ellipsis icon.
6. Click **Chart Configuration**.
   The Chart Configuration window displays.

   > **Note** As a best practice, keep the Chart Configuration as the default.

7. For the x-axis column, select one of the following options from the **X-Axis Column** dropdown list:
    - **username**
    - **userClicks**
8. For the y-axis columns, select one or more of the following options from the **Y-Axis Columns** dropdown list:
    - **userClicks**
    - **username**
9. For the type of chart you want to display, select one of the following options from the **Chart Type** dropdown list:
    - **Line**
    - **Area**

- Column
- Scatter

10. Click **Save**.
    The chart displays based on the criteria you selected. The table below the chart displays the specific usernames and user clicks, based on your selections.

> **Note** To close the Chart Configuration window, click **Cancel** or the **X** in the upper-right corner of the window. You can hide or show the filter options by clicking the ^ or v icons, located to the right of the filter options.

> **Tip** To narrow your results in the data table, you can enter information in the Search field.

11. Navigate the pages of your report by clicking the and icons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a User Activity report.

```
select username, count(*) user_clicks from activity_audit
where audit_ts > '2022-03-05'
and username <> 'admin'
group by username
order by user_clicks desc limit 22
```

Row Count by Dataset Report

What is a Row Count by Dataset Report?

The Row Count by Dataset report provides a view of the row count in the Collibra DQ application by dataset. You can filter this report by various criteria from a Filter Type

dropdown, as well as set a limit to the number of datasets you want to view, based on the filter you choose. You can also configure the kind of chart you want to use to view the data.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, select the **Row Count by Dataset** link.
3. To display results by filter type, select one of the following options from the **Filter Type** dropdown list:
   - **Last x Days** - filter by last number of days.
     i. Select one of the following options from the **Last x Days** dropdown list:
        - **1 day**
        - **2 days**
        - **7 days**
        - **10 days**
        - **15 days**
        - **30 days**

     ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

     > **Note** The chart and table update, based on your selections.

     iii. Proceed to step 4.
   - **Custom Range** - filter by configuring a custom range.

i. Select a from and to date range in the Date Range fields.

> **Tip** You can also select the  icon, which displays a calendar, in which you can select a specific date.

ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

iii. Proceed to step 4.

- **All** - filter by datasets and row counts.

i. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

ii. Proceed to step 4.

4. Click **Go**.
5. To configure the type of chart configuration and data you want to see in the chart, click the ellipsis icon.
6. Click  **Chart Configuration**.
   The Chart Configuration window displays.

> **Note** As a best practice, keep the Chart Configuration as the default.

7. For the x-axis column, select one of the following options from the **X-Axis Column** dropdown list:
   - **rowCount**
   - **dataset**

8. For the y-axis columns, select one or more of the following options from the **Y-Axis Columns** dropdown list:
    - **rowCount**
    - **dataset**
9. For the type of chart you want to display, select one of the following options from the **Chart Type** dropdown list:
    - **Line**
    - **Area**
    - **Column**
    - **Scatter**
10. Click **Save**.
    The chart displays based on the criteria you selected. The table below the chart displays the specific datasets and row counts, based on your selections.

    > **Note** To close the Chart Configuration window, click **Cancel** or the **X** in the upper-right corner of the window. You can hide or show the filter options by clicking the ^ or v icons, located to the right of the filter options.

    > **Tip** To narrow your results in the data table, you can enter information in the Search field.

11. Navigate the pages of your report by clicking the < and > icons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a Row Count by Dataset report.

```
select dataset,rc as row_count from dataset_scan order by rc
desc
```

Alert Details Report

What is an Alert Details Report?

The Alert Details report provides a view of the details of all the alerts in the Collibra DQ application and the method by which they are sent. You can filter this report by various criteria from a Filter Type dropdown, as well as set a limit to the number of alert details you want to view, based on the filter you choose. You can also configure the kind of chart you want to use to view the data.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, select the **Alert Details** link.
3. To display results by filter type, select one of the following options from the **Filter Type** dropdown:
   - **Last x Days** - filter by last number of days.
     i. Select one of the following options from the **Last x Days** dropdown:
     - **1 day**
     - **2 days**
     - **7 days**
     - **10 days**
     - **15 days**

- 30 days

ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

iii. Proceed to step 4.

- **Custom Range** - filter by configuring a custom range.

i. Select a from and to date range in the Date Range fields.

> **Tip** You can also select the  icon, which displays a calendar, in which you can select a specific date.

ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

iii. Proceed to step 4.

- **All** - filter by all alerts and alert value formats.

i. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

ii. Proceed to step 4.

4. Click **Go**.
5. To configure the type of chart configuration and data you want to see in the chart, click the ⋮ ellipsis icon.

6. Click **Chart Configuration**.
   The Chart Configuration window displays.

   > **Note** As a best practice, keep the Chart Configuration as the default.

7. For the x-axis column, select one of the following options from the **X-Axis Column** dropdown:
   - **alerts**
   - **alertFormatValue**
8. For the y-axis columns, select one or more of the following options from the **Y-Axis Columns** dropdown:
   - **alerts**
   - **alertFormatValue**
9. For the type of chart you want to display, select one of the following options from the **Chart Type** dropdown:
   - **Line**
   - **Area**
   - **Column**
   - **Scatter**
10. Click **Save**.
    The chart displays based on the criteria you selected. The table below the chart displays the specific alert details, based on your selections.

    > **Note** To close the Chart Configuration window, click **Cancel** or the **X** in the upper-right corner of the window. You can hide or show the filter options by clicking the ^ or v icons, located to the right of the filter options.

> **Tip** To narrow your results in the data table, you can enter information in the Search field.

11. Navigate the pages of your report by clicking the and icons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for an Alert Details report.

```
select alert_format_value, count(*) alerts
from alert_cond
group by alert_format_value
order by alerts desc limit 10
```

Job Runs per User

What is a Job Runs per User Report?

The Job Runs per User report provides a view of the count of DQ Jobs run in the Collibra DQ application by user. You can filter this report by various criteria from a Filter Type dropdown, as well as set a limit to the number of usernames and job runs you want to view, based on the filter you choose. You can also configure the kind of chart you want to use to view the data.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, select the **Job Runs Per User** link.

3. To display results by filter type, select one of the following options from the **Filter Type** dropdown list:
    - **Last x Days** - filter by last number of days.
        i. Select one of the following options from the **Last x Days** dropdown:
            - 1 day
            - 2 days
            - 7 days
            - 10 days
            - 15 days
            - 30 days
        ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

        > **Note** The chart and table update, based on your selections.

        iii. Proceed to step 4.
    - **Custom Range** - filter by configuring a custom range.
        i. Select a from and to date range in the Date Range fields.

        > **Tip** You can also select the  icon, which displays a calendar, in which you can select a specific date.

        ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

        > **Note** The chart and table update, based on your selections.

        iii. Proceed to step 4.

- **All** - filter by all job runs and users.

  i. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

  > **Note** The chart and table update, based on your selections.

  ii. Proceed to step 4.

4. Click **Go**.
5. To configure the type of chart configuration and data you want to see in the chart, click the  ellipsis icon.
6. Click  **Chart Configuration**.
   The Chart Configuration window displays.

   > **Note** As a best practice, keep the Chart Configuration as the default.

7. For the x-axis column, select one of the following options from the **X-Axis Column** dropdown list:
   - **jobs**
   - **username**
8. For the y-axis columns, select one or more of the following options from the **Y-Axis Columns** dropdown list:
   - **jobs**
   - **username**
9. For the type of chart you want to display, select one of the following options from the **Chart Type** dropdown list:
   - **Line**
   - **Area**

- Column
- Scatter

10. Click **Save**.
    The chart displays based on the criteria you selected. The table below the chart displays the specific datasets and row counts, based on your selections.

    > **Note** To close the Chart Configuration window, click **Cancel** or the **X** in the upper-right corner of the window. You can hide or show the filter options by clicking the ^ or v icons, located to the right of the filter options.

    > **Tip** To narrow your results in the data table, you can enter information in the Search field.

11. Navigate the pages of your report by clicking the and icons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a Job Runs per User report.

```
select username, count(*) jobs
from owlcheck_q
where updt_ts > '2022-03-07'
and username <> 'admin'
group by username
order by jobs desc
```

Daily Job Runs per User

What is a Daily Job Runs per User Report?

The Daily Job Runs per User report provides a view of the count of DQ Jobs run in the Collibra DQ application by user, per day. You can filter this report by various criteria from a Filter Type dropdown list, as well as set a limit to the number of usernames, dates, and job runs you want to view, based on the filter you choose. You can also configure the kind of chart you want to use to view the data.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, select the **Daily Job Runs Per User** link.
3. To display results by filter type, select one of the following options from the **Filter Type** dropdown:
   - **Last x Days** - filter by last number of days.
     i. From the Last x Days dropdown list, select one of the following options:
        - **1 day**
        - **2 days**
        - **7 days**
        - **10 days**
        - **15 days**
        - **30 days**

     ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

iii. Proceed to step 4.

- **Custom Range** - filter by configuring a custom range.
  i. Select a from and to date range in the Date Range fields.

> **Tip** You can also select the  icon, which displays a calendar, in which you can select a specific date.

  ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

  iii. Proceed to step 4.
- **All** - filter by all daily job runs and users.
  i. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

  ii. Proceed to step 4.

4. Click **Go**.
5. To configure the type of chart configuration and data you want to see in the chart, click the  ellipsis icon.
6. Click  **Chart Configuration**.
   The Chart Configuration window displays.

> **Note** As a best practice, keep the Chart Configuration as the default.

7. From the X-Axis Column dropdown list, select one of the following options:
   - **jobs**
   - **updtTsStr**
   - **username**
8. From the Y-Axis Columns dropdown list, select one or more of the following options:
   - **jobs**
   - **updtTsStr**
   - **username**
9. From the Chart Type dropdown list, select one of the following options:
   - **Line**
   - **Area**
   - **Column**
   - **Scatter**
10. Click **Save**.
    The chart displays based on the criteria you selected. The table below the chart displays the specific datasets and row counts, based on your selections.

> **Note** To close the Chart Configuration window, click **Cancel** or the **X** in the upper-right corner of the window. You can hide or show the filter options by clicking the ^ or v icons, located to the right of the filter options.

> **Tip** To narrow your results in the data table, you can enter information in the Search field.

11. Navigate the pages of your report by clicking the and icons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a Job Runs per User report.

```
select updt_ts::date, username, count(*)
from owlcheck_q
where updt_ts > '2022-3-09'
and username <> 'admin'
group by updt_ts::date, username
order by updt_ts asc
```

Rule Updates per User

What is a Rule Updates per User Report?

The Rule Updates per User report provides a view of when a rule is inserted, updated, or deleted and applied to the dataset in the Collibra DQ application, by user for a specific time. You can filter this report by various criteria from a Filter Type dropdown, as well as set a limit to how many users and rules you want to view, based on the filter you choose. You can also configure the kind of chart you want to use to view the data.

Steps

1. Sign in to the Collibra DQ instance and click the  Reports icon in the left navigation pane.
   » The Reports page opens.
2. From the Reports page, select the **Rule Updates per User** link.
3. To display results by filter type, select one of the following options from the **Filter Type** dropdown list:

- **Last x Days** - filter by last number of days.
    i. Select one of the following options from the **Last x Days** dropdown list:
        - **1 day**
        - **2 days**
        - **7 days**
        - **10 days**
        - **15 days**
        - **30 days**
    ii. In the Limit input field, enter a value to limit the number of top end results that display in the chart and data table.

> **Note** The chart and table update, based on your selections.

    iii. Proceed to step 4.
- **Custom Range** - filter by configuring a custom range.
    i. Select a from and to date range in the Date Range fields.

> **Tip** You can also select the  icon, which displays a calendar, in which you can select a specific date.

    ii. To limit the number of users that the chart displays, enter a number in the **Limit** field.

> **Note** The chart and table update, based on your selections.

    iii. Proceed to step 4.
- **All** - filter by username and rules.

i. To limit the number of users that the chart displays, enter a number in the **Limit** field.

> **Note** The chart and table update, based on your selections.

ii. Proceed to step 4.

4. Click **Go**.
5. To configure the type of chart configuration and data you want to see in the chart, click the ⋮ ellipsis icon.
6. Click ⚙ **Chart Configuration**.
   The Chart Configuration window displays.

> **Note** As a best practice, keep the Chart Configuration as the default.

7. For the x-axis column, select one of the following options from the **X-Axis Column** dropdown list:
   - **rules**
   - **username**
8. For the y-axis columns, select one or more of the following options from the **Y-Axis Columns** dropdown list:
   - **rules**
   - **username**
9. For the type of chart you want to display, select one of the following options from the **Chart Type** dropdown list:
   - **Line**
   - **Area**
   - **Column**
   - **Scatter**

10. Click **Save**.
    The chart displays based on the criteria you selected. The table below the chart displays the specific usernames and user clicks, based on your selections.

    > **Note** To close the Chart Configuration window, click **Cancel** or the **X** in the upper-right corner of the window. You can hide or show the filter options by clicking the ^ or v icons, located to the right of the filter options.

    > **Tip** To narrow your results in the data table, you can enter information in the Search field.

11. Navigate the pages of your report by clicking the and icons, located bottom-right of the columns.

Sample SQL query

You can use the following sample SQL query for a User Activity report.

```
select user_nm, count(*) rules from rule_history
where updt_ts > '2022-03-05'
and user_nm <> 'admin'
group by user_nm order by rules desc
```

Report Dashboards

Column Dimension Dashboard

What is the Column Dimension Dashboard?

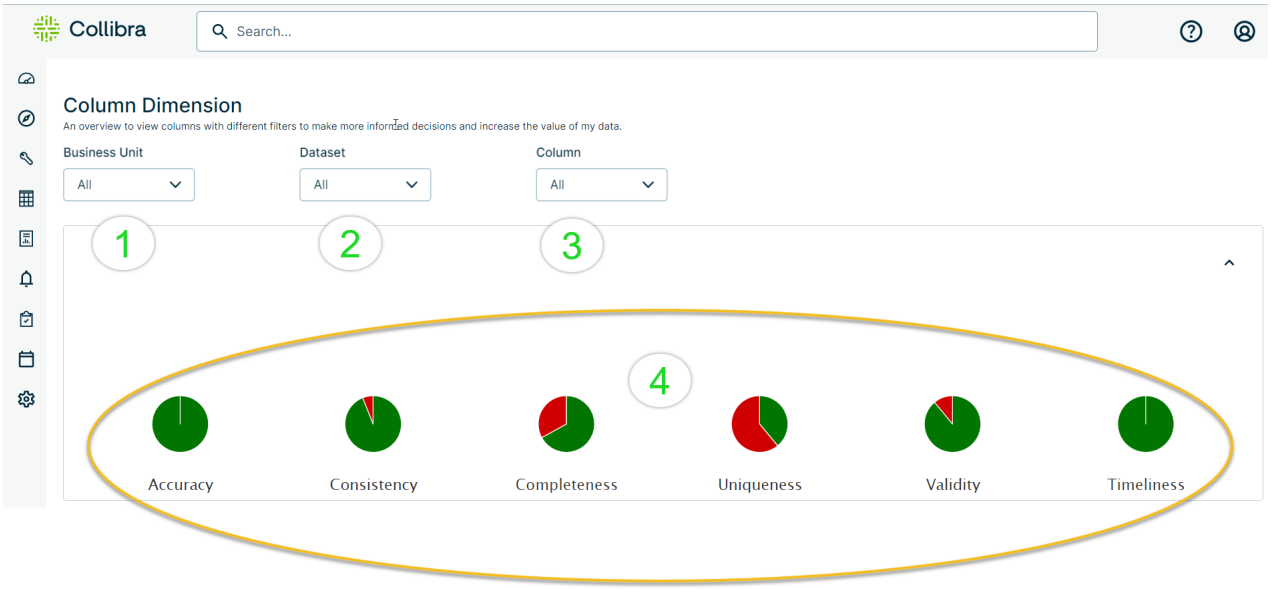
The Column Dimension Dashboard provides a way to view and filter Collibra DQ dimension scores by business units, datasets, and column. This dashboard is essentially an overview of all

business units, datasets, and column in the Collibra DQ application, which provides insights about the total current DQ Dimension scores, Dimension scores over time, and DQ scores (as well as other metadata) for each column.

Filter by Business Unit, Dataset, and Column

The following table describes the Business Unit, Dataset, and Column filters.

> **Note** To show/hide any of the charts and tables, click ⌄ or ⌃.



1	Dimension results by business unit.	Select a business unit from the **Business Unit** dropdown.
2	Dimension results by dataset.	Select the dataset from the **Dataset** dropdown.
3	Dimension results by column.	Select the column option from the **Column** dropdown. **Tip** Select **All** from a dropdown list, to display data for all columns, datasets, and business units in your report.

4	Dimension results pie charts.	Charts that display dimensions based on the criteria you select from the dropdown lists. These charts measure the accuracy, consistency, completeness, uniqueness, validity, and timeliness of your data.

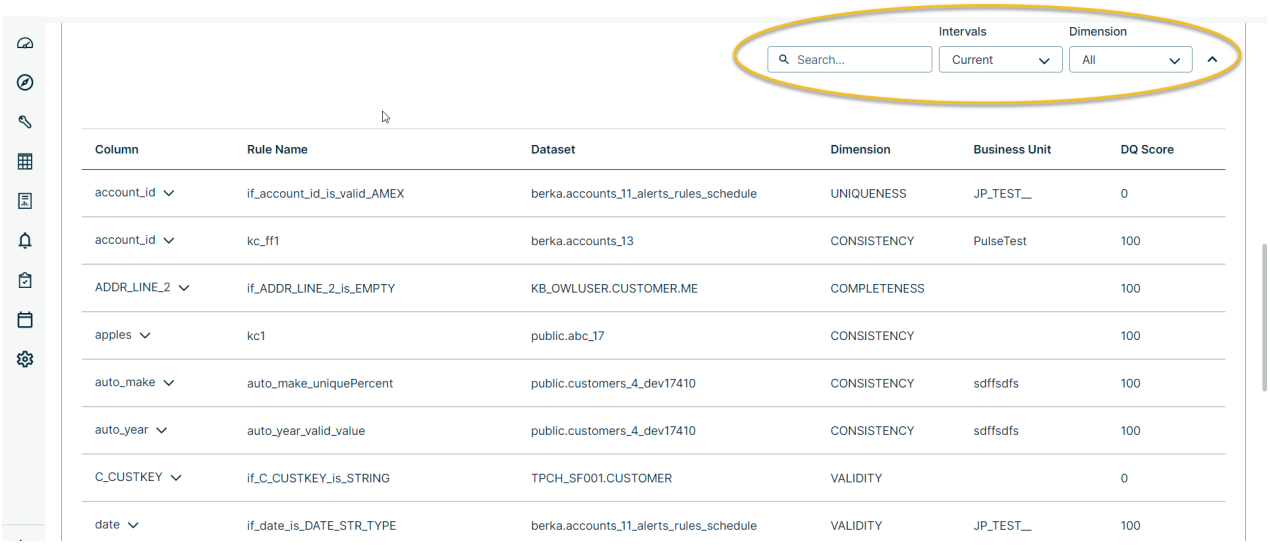
> **Tip** You can hover over a chart to view the percentages for that specific dimension.

Filter by Time Frames and Dimension

You can view and filter the table at the bottom of the page by intervals and dimension, to display detailed information and metadata. You filter by intervals, such as the current day, or the last 3, 6, or 12 months, by selecting them from the **Intervals** dropdown.

Filter the table by dimension from the **Dimension** dropdown.

> **Tip** You can also narrow your results in the table by entering criteria in the **Search...** field.



Dataset Dimension Report

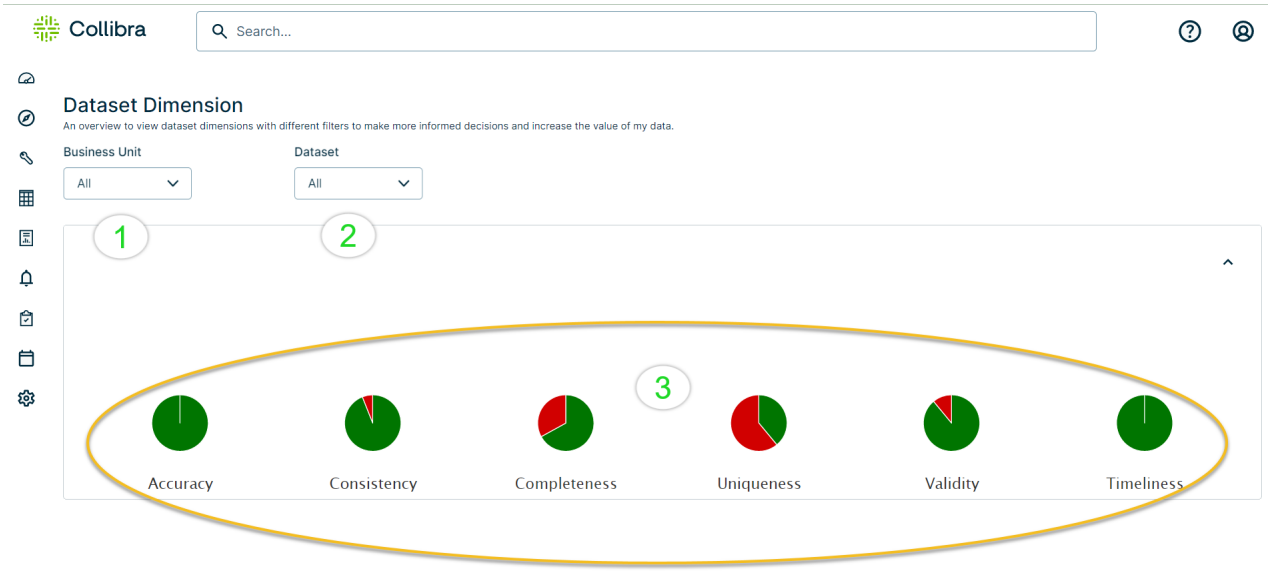
What is the Dataset Dimension Report?

The Dataset Dimension Report provides a way to view and filter Collibra DQ dimension scores by datasets and business units. This report is essentially an overview, which provides insights about averaged or aggregated scores across datasets and business units in the Collibra DQ application.

Filter by Dataset and Business Unit

The following table describes the Dataset and Business Unit filters.

> **Note** To show/hide any of the charts and tables, click ⌄ or ⌃.



1	Dimension results by business unit.	Select a business unit from the **Business Unit** dropdown.
2	Dimension results by dataset.	Select the dataset from the **Dataset** dropdown.

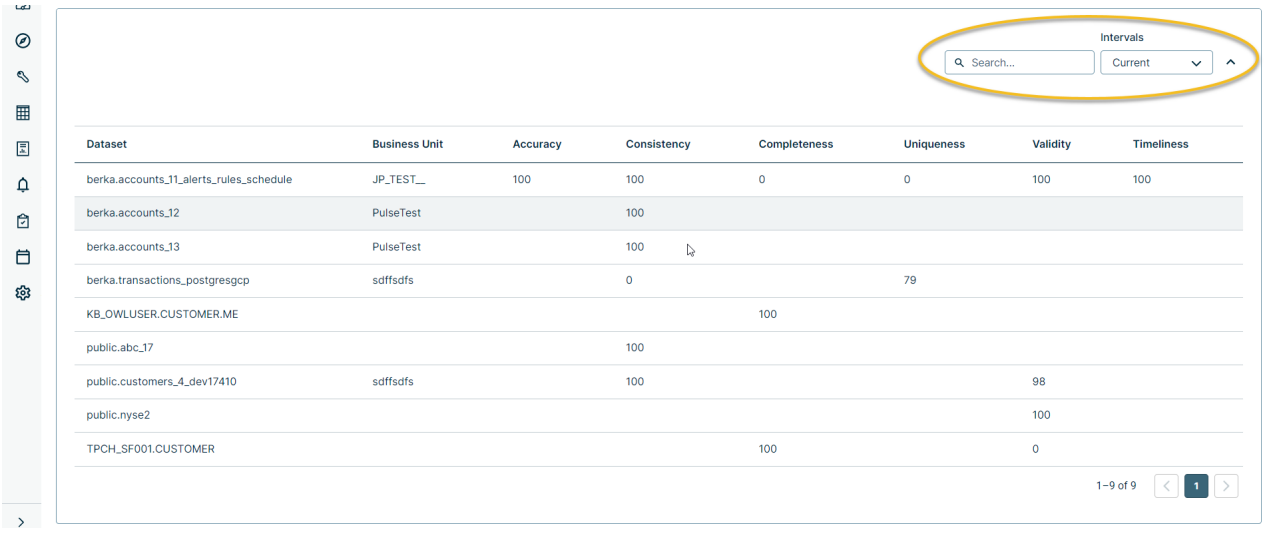
3	Dimension results pie charts.	Charts that display dimensions based on the criteria you select from the dropdown lists. These charts measure the accuracy, consistency, completeness, uniqueness, validity, and timeliness of your data.

> **Tip** You can hover over a chart to view the percentages for that specific dimension.

Filter by Time Frames

You can view and filter the table at the bottom of the page by intervals, to display detailed information and metadata. You filter by intervals, such as the current day, or the last 3, 6, or 12 months, by selecting them from the **Intervals** dropdown.

> **Tip** You can also narrow your results in the table by entering criteria in the Search... field.



Custom

Custom reports can be leveraged by connecting your favorite BI tool on the underlying reporting mart. Below are a few queries that can be used as inspiration for building your own reports. Please refer to the ERD diagram for a larger list of tables.

Long Running Jobs

```
select dataset,run_id,total_time from dataset_activity where total_
time is not null order by total_time desc
```

Jobs Submitted

```
select * from owlcheck_q
```

Jobs by User

```
select count(*) as owlchecks, username from owlcheck_q where updt_ts
< now() group by username order by owlchecks desc
```

Jobs by User, Dataset

```
select count(*), user_nm, dataset from dev.public.owl_check_history
group by user_nm, dataset order by count desc
```

Largest by Row Count

```
select dataset,rc as row_count from dataset_scan order by rc desc
```

Jobs by Month

```
with grp as ( select date_trunc('MONTH', run_id) as by_month from
dataset_scan where run_id < now() ) select count(*) as owlchecks,
by_month from grp group by by_month order by by_month desc
```

Rules by User

```
select count(*) as rules, user_nm from owl_rule group by user_nm
order by rules desc
```

By Spark(Cluster) Usage

```
select * from opt_spark order by num_executors desc
```

Jobs IDs from Agent

```
select remote_job_id from agent_q where remote_job_id is not null
```

Dataset Activity

```
select dataset,run_id,total_time from dataset_activity where total_
time is not null order by total_time desc
```

Jobs with Enriched Metrics

```
with activity as ( select dataset,run_id,total_time from dataset_
activity where total_time is not null order by total_time desc limit
100), scans as ( select * from dataset_scan where dataset in (select
dataset from activity) ), configs as ( select * from opt_spark where
dataset in (select dataset from activity)), schema as ( select count
(*) as col_cnt, dataset from dataset_schema where dataset in (select
dataset from activity) group by dataset ) SELECT A.dataset, A.run_
id, C.total_time, A.rc, D.col_cnt, B.driver_memory, B.num_
executors,B.executor_cores, B.executor_memory, B.master FROM scans A
INNER JOIN configs B ON A.dataset = B.dataset INNER JOIN activity C
ON A.dataset = C.dataset and A.run_id = C.run_id INNER JOIN schema D
on A.dataset = D.dataset ORDER BY C.total_time desc
```

Jobs. Load Times and Resources

```
with activity as ( select dataset,run_id,total_time from
public.dataset_activity where total_time is not null order by total_
time), scans as ( select * from public.dataset_scan where dataset in
(select dataset from activity) ), configs as ( select * from
public.opt_spark where dataset in (select dataset from activity)),
schema as ( select count(*) as col_cnt, dataset from public.dataset_
schema where dataset in (select dataset from activity) group by
dataset ) SELECT A.dataset, A.run_id, A.updt_ts, C.total_time, A.rc,
D.col_cnt, B.driver_memory, B.num_executors,B.executor_cores,
B.executor_memory, B.master FROM scans A INNER JOIN configs B ON
A.dataset = B.dataset INNER JOIN activity C ON A.dataset = C.dataset
and A.run_id = C.run_id INNER JOIN schema D on A.dataset = D.dataset
ORDER BY A.updt_ts desc limit 10
```

Dataset Scans and Scores By Schema

```
select * from public.dataset_scan where dataset like 'public.%';
```

Dataset Scans and Scores By Name

```
select * from public.dataset_scan where dataset ='public.atm_
customer';
```

Scans By Month By Schema - 'Public'

```
select dataset, DATE_TRUNC('MONTH', run_id) as run_id, count(*) as
Total_Scans from dataset_scan where dataset like 'public%' group by
dataset, run_id order by run_id asc
```

Rule Breaks Past 30 Days

```
select * from rule_output where run_id < NOW() - INTERVAL '30 DAY';
```

Scheduled Jobs Queue

```
select job_id,agent_id,dataset,run_id,status,activity,start_time
from public.owlcheck_q;
```

Column Counts from Dataset Schema

```
select dataset, count(*) from dataset_schema group by dataset;
```

Profiling Stats

```
select dataset, run_id, field_nm, (null_ratio * 100) as null_
percent, (empty_ratio * 100) as empty_percent, ROUND( CAST( ( 100 -
((null_ratio * 100) + (empty_ratio * 100)) ) as numeric), 3) as
completeness from public.dataset_field where updt_ts > '2020-06-01'
and dataset = 'ProcessOrder' and run_id > '2021-03-17 00:00:00+00'
order by completeness desc
```

Metadata / Schema / Datatypes

```
select * from public.dataset_schema;
```

Profile Stats

```
select * from public.dataset_field;
```

Locate Similar Columns

```
select distinct dataset, field_nm, max_abs from dataset_field where
max_abs = 'Wireless Telecommunications'
```

Same Column Names

```
select distinct dataset, field_nm from dataset_field where field_nm
= 'authenticated_user'
```

Similar Column Names

```
select distinct dataset,field_nm from dataset_field where field_nm
like '%id%'
```

Behavior Findings

```
select * from behavior where dataset='esg_data'
```

All Columns for Schema from Postgres Stats

```
SELECT table_name FROM information_schema.tables WHERE table_schema
= 'public' ORDER BY table_name
```

Alerts

This section provides information about how to configure alerts in the Collibra DQ application.

Score-based conditions ..... mccxc

Row count condition ..... mccxci

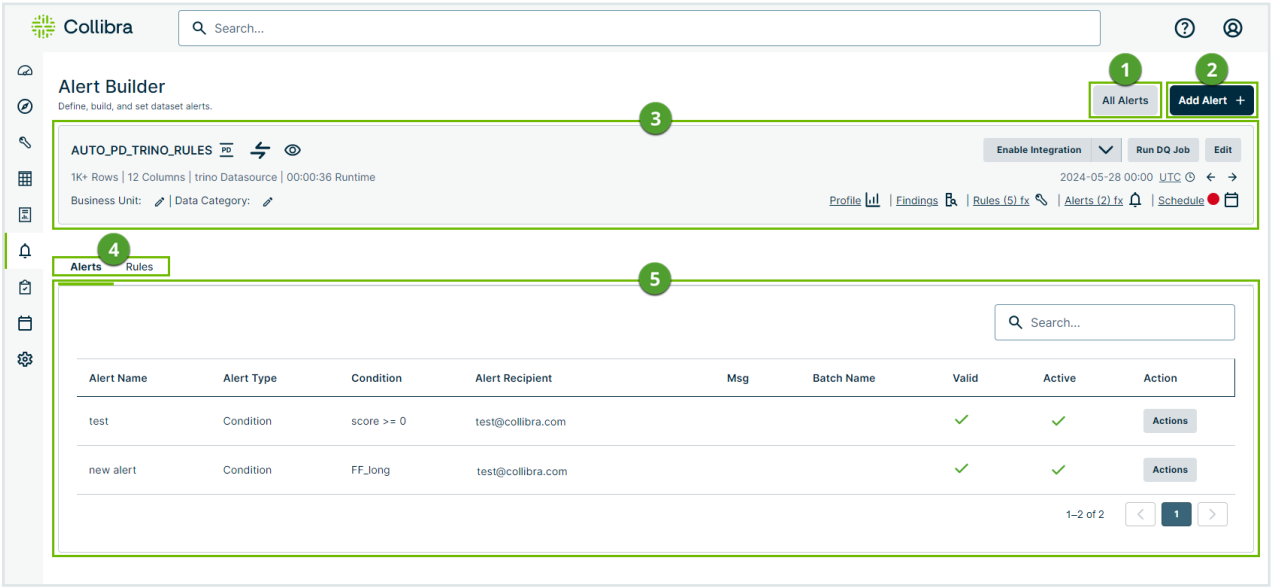
Run time conditions ..... mccxci

Stale data conditions ..... mccxci

Rule conditions ..... mccxcii

Working with the Alert Builder

Alert Builder lets you manage alerts for specific datasets. This section shows you how to define, build, and set dataset run and job status alerts for your dataset.



No.	Component	Description
1	All Alerts	Click **All Alerts** to view all of the dataset alerts configured in your Collibra DQ environment. The paginated All Alerts table includes alert details, including: Dataset The name of the dataset with an alert configured. Alert Name The user-defined name of the dataset alert. Condition The condition of the dataset alert that must be satisfied for the alert to trigger. Format The alert type, for example, EMAIL. Format Value The alert recipient, for example, an EMAIL format would have a Format Value of admin.istrator@test.com. You can also scroll to the far right of the table and optionally click  to delete a dataset alert.
2	Add Alert	Optionally click **Add Alert +** to create a new alert for when a job run of your dataset meets a certain condition or status.
3	Metadata bar	A dataset anchor that empowers you with the ability to access crucial data quality components related to your dataset, such as its Findings, Profile, and associated rules and alerts.
4	Alerts and Rules tabs	**Alerts** (default tab) Shows the alert overview table containing details about the alerts configured for your dataset. **Rules** Shows the rule overview table containing details about rules configured against your dataset. You can configure alerts to send to recipients when a dataset run satisfies the condition of the rule.
5	Overview table	Depending on whether the Alerts or Rules tab is selected, this is an overview table of the alerts or rules configured for your dataset. See the Managing Alerts and Alerts for Rules section below for more details.

Managing Alerts and Alerts for Rules

You can manage existing alerts and configure alerts for rules from the following pages.

> **Tip** To sort a column, click the ↑ ↓ icons next to the column header.

Column	Description
Alert Name	The name of the alert for the selected dataset.
Dataset	The name of the selected dataset.
Condition	The condition configured for the alert.
Alert Recipient	The webhook name or email address of the recipient of the alert email.
Msg	A custom message pertaining to the alert.
Batch Name	The name of the batch alert, if applicable. This is a group name for a set of email addresses of alert recipients. You can search for and select an existing batch name to quickly apply frequently used email recipient groups to an alert.
Valid	The validity of the alert. If the alert is valid, a check mark displays in this column.
Active	Shows whether the alert is active for future runs of the dataset. Click the icon to activate or deactivate the alert. ✓ shows that the alert is active. ⊗ shows that the alert is inactive.
Action	Click **Actions** and then select one of the following options from the dropdown menu: **Edit** Modify the values of the selected alert. **Rename** Update the name of an alert. > **Note** The updated name applies only to alerts generated in future job runs and does not affect historical data. **Delete** Delete the selected alert.

Tip To sort a column, click the ↑ ↓ icons next to the column header.

Column	Description
Rule Name	The name of the rule. **Note** Rule names are subject to Spark naming conventions, which generally only allows letters, numbers, and underscores (_)in names. If your rule name contains hyphens, you can enclose it with backticks (`) to ensure that the alert works correctly. However, rule names that contain unsupported characters may result in alert delivery failure. For more information about Spark naming conventions, see the official Spark documentation .
Dataset	The name of the selected dataset.
Value	The SQL condition of your rule.
Points	The number of points that Collibra DQ deducts from your data quality score when data breaches the conditions of your rule.
%	If you do not customize this value on the Rule Details modal, Collibra DQ uses the default value of 1.
Valid	Shows whether your rule passes rule validation. ✓ Shows that your rule passes validation. ⚠ Shows that your rule does not pass validation.
Type	The type of rule, for example, SQLF.
Action	Click **Create Alert +** to create a new alert for the dataset. After you configure an alert for a rule, you can access it from the **Alerts** tab.

Adding dataset alerts

This topic shows you how to configure dataset alerts. You can set alerts to send via email or to a webhook.

Steps

1. Sign in to Data Quality & Observability Classic.
2. On the left sidebar navigation, click  **Alerts**, then select **Alert Builder**.
   » The Alert Builder page opens.
3. Search for your dataset, or select one from either the **App Frequent Runs:** section below the search bar or from the **App Recent Runs** list.
   » The dataset you selected displays. You can now view, create, and edit alerts for this dataset.
4. In the upper right corner, click **Add Alert +**.
   » The Add Alert dialogue appears.
5. From the Add Alert dialogue, select an option and then click **Next**.
   » Depending on the option you select, the Condition Alert, Job Status Alert, or Rule Status Alert modal appears.
6. Enter the required information for the option you select.

   A condition alert generates an alert when the user-defined condition is satisfied during a dataset run.

Option	Description
Alert Name	The name of the alert for the selected dataset.
Dataset	The name of the dataset. This field cannot be edited.

Option	Description
Condition	The user-defined condition that triggers an alert to send when its criteria are met during a dataset run. You can specify conditions such as: • score < 75 • daysWithoutData > 1 • rc = 0
Add Rule Details	Enter the name of your rule in the Condition field to include the following details of your rule in the email: • Rule name • Rule condition • Total number of points to deduct from the quality score when breaking • Percentage of records that are breaking • Number of breaking records • Assignment Status **Note** Add Rule Details is unavailable by default. When you enter the name of a rule in the Condition field, this option becomes available.
Alert delivery method	
Email	Select this option to use email for alerting. **Note** You must have SMTP configured to use this option. For more information on how to set up SMTP, go to Alert Configuration.
Email Batch Trigger	Select the checkbox option to create a batch alert to multiple recipients. When this option is enabled, the Batch Name field appears and the Alert Recipient field locks automatically.

Option	Description
Batch Name	Enter a batch name to send the alert to multiple recipients. To search for an existing batch name, enter its characters into the Batch Name field to populate a dropdown list containing all available matches. **Note** This option is only available when the Batch option is enabled.
Alert Recipient	The email address where alerts will send. You can specify either a single recipient or multiple. **Note** When you specify multiple email addresses, ensure that you also fill out the Batch Name field. Multiple recipients are specified by comma (,) and/or semicolon (;) delimiters. The accepted formats for email are "name@email.com" OR "name1@email.com,name2@email.com" OR "name1@email.com;name2@email.com". **Tip** When the batch option is enabled, the Alert Recipient field locks automatically. You can click the 🔒 to the right of the Alert Recipient field to unlock it and edit the alert recipients.
Webhook	Select this option to use a webhook for alerting. **Note** You must have a webhook configured to use this option. For more information on how to set up webhooks, go to Alert Webhook Configuration.
Webhook	Select a webhook from the drop-down list where the JSON payload containing your alert details will be sent.
Custom Msg	A custom message relevant to the alert.

A job status alert generates an alert when a job run completes.

Option	Description
Alert Name	The name of the alert for the selected dataset.
Dataset	The name of the dataset. This field cannot be edited.
Job Alert Status	**Job Completion** Select this option to generate an alert when a job run completes for the selected dataset. **Job Failure** Select this option to generate an alert when a job run fails.
Alert delivery method	
Email	Select this option to use email for alerting. **Note** You must have SMTP configured to use this option. For more information on how to set up SMTP, go to Alert Configuration.
Email Batch Trigger	Select the checkbox option to create a batch alert to multiple recipients. When this option is enabled, the Batch Name field appears and the Alert Recipient field locks automatically.
Batch Name	Enter a batch name to send the alert to multiple recipients. To search for an existing batch name, enter its characters into the Batch Name field to populate a dropdown list containing all available matches. **Note** This option is only available when the Batch option is enabled.

Option	Description
Alert Recipient	The email address where alerts will send. You can specify either a single recipient or multiple. **Note** When you specify multiple email addresses, ensure that you also fill out the Batch Name field. Multiple recipients are specified by comma (,) and/or semicolon (;) delimiters. The accepted formats for email are "name@email.com" OR "name1@email.com,name2@email.com" OR "name1@email.com;name2@email.com". **Tip** When the batch option is enabled, the Alert Recipient field locks automatically. You can click the 🔒 to the right of the Alert Recipient field to unlock it and edit the alert recipients.
Webhook	Select this option to use a webhook for alerting. **Note** You must have a webhook configured to use this option. For more information on how to set up webhooks, go to Alert Webhook Configuration.
Webhook	Select a webhook from the drop-down list where the JSON payload containing your alert details will be sent.
Custom Msg	A custom message relevant to the alert.

A rule status alert generates an alert when a rule is breaking, passing, and/or has an exception.

Option	Description
Alert Name	The name of the alert for the selected dataset. **Important** If the name of your alert already exists, the new alert will replace the existing one.
Dataset	The name of the dataset. This field cannot be edited.
Rule Status Triggers	You can select one or more of the following options: **Breaking** Select this option to generate an alert when a rule is breaking. **Exception** Select this option to generate an alert when a rule is throwing an exception. **Passing** Select this option to generate an alert when a rule is passing.
Alert delivery method	
Email	Select this option to use email for alerting. **Note** You must have SMTP configured to use this option. For more information on how to set up SMTP, go to Alert Configuration.
Email Batch Trigger	Select the checkbox option to create a batch alert to multiple recipients. When this option is enabled, the Batch Name field appears and the Alert Recipient field locks automatically.
Batch Name	Enter a batch name to send the alert to multiple recipients. To search for an existing batch name, enter its characters into the Batch Name field to populate a dropdown list containing all available matches. **Note** This option is only available when the Batch option is enabled.

Option	Description
Alert Recipient	The email address where alerts will send. You can specify either a single recipient or multiple. **Note** When you specify multiple email addresses, ensure that you also fill out the Batch Name field. Multiple recipients are specified by comma (,) and/or semicolon (;) delimiters. The accepted formats for email are "name@email.com" OR "name1@email.com,name2@email.com" OR "name1@email.com;name2@email.com". **Tip** When the batch option is enabled, the Alert Recipient field locks automatically. You can click the 🔒 to the right of the Alert Recipient field to unlock it and edit the alert recipients.
Webhook	Select this option to use a webhook for alerting. **Note** You must have a webhook configured to use this option. For more information on how to set up webhooks, go to Alert Webhook Configuration.
Webhook	Select a webhook from the drop-down list where the JSON payload containing your alert details will be sent.
Custom Msg	A custom message relevant to the alert.

7. Click **Save**.

Using conditions for alerts

This topic provides an overview of the types of condition alerts you can use when configuring a condition alert.

Score-based conditions

You can use score-based conditions to identify when the overall data quality score or the score of particular monitors exceed a given threshold. You can use any number of `AND` and `OR` operators in the condition.

Type	Variable	Description	Condition example
Score	`score`	Send an alert when the data quality score meets the required conditions.	`score < 75`
Behavior score	`behaviorscore`	Send an alert when the behavior score meets the required conditions.	`behaviorscore < 20`
Outlier score	`outlierscore`	Send an alert when the outlier score meets the required conditions.	`outlierscore > 15`
Pattern score	`patternscore`	Send an alert when the pattern score meets the required conditions.	`patternscore = 0`
Source score	`sourcescore`	Send an alert when the source score meets the required conditions.	`sourcescore < 75`
Record score	`recordscore`	Send an alert when the record score meets the required conditions.	`recordscore < 50`
Schema score	`schemascore`	Send an alert when the schema score meets the required conditions.	`schemascore < 50`
Dupe score	`dupescore`	Send an alert when the dupe score meets the required conditions.	`dupescore > 10`
Shape score	`shapescore`	Send an alert when the shape score meets the required conditions.	`shapescore > 0`

Row count condition

You can use the row count condition to help you monitor changes to the number of rows in your dataset.

Type	Variable	Description	Condition example
Row count	`rc`	Send an alert when the row count meets the required conditions.	`rc = 0`

Run time conditions

You can use run time conditions to monitor how long it takes for a DQ Job to run.

Type	Variable	Description	Condition example
Run time in seconds	`totalTimeInSeconds`	Send an alert when the run time meets the required conditions.	`totalTimeInSeconds > 20`
Run time in hours	`totalTimeInHours`	Send an alert when the run time meets the required conditions.	`totalTimeInHours > 2`
Run time in minutes	`totalTimeInMinutes`	Send an alert when the run time meets the required conditions.	`totalTimeInMinutes > 20`

Stale data conditions

You can use stale data conditions to monitor for when DQ Jobs have missing or no rows, or have not run for a given number of days.

Type	Variable	Description	Condition example
Days without data	`daysWithoutData`	Send an alert when the number of days without data meets the required conditions.	`daysWithoutData >= 1`
Runs without data	`runsWithoutData`	Send an alert when the number of runs without data meets the required conditions.	`runsWithoutData > 2`
Days since last run	`daysSinceLastRun`	Send an alert when the number of days since a dataset last ran meets the required conditions.	`daysSinceLastRun > 1`

Rule conditions

You can configure rule condition alerts in the Rules tab of the Alert Builder to notify users when the percentage of breaking records or the score meets the specified conditions of the rule.

To use a rule condition as an alert, select **Condition Alert**, then enter the name of an existing rule in the **Condition** field. Optionally, select **Add Rule Details** to include the following details of your rule in the alert email:

- Rule name
- Rule condition
- Total number of points to deduct from the quality score when breaking
- Percentage of records that are breaking
- Number of breaking records
- Assignment Status

Examples of alert conditions for rule break percentages

Condition alerts for rule break percentages require a `<ruleName> <operator> <percentage>` syntax.

Example alert condition	Expected result
`RegionNullCheck > 0`	An alert sends when there are any breaking records.
`RegionNullCheck < 80`	An alert sends when the percentage of rule breaks is less than or equal to 80%.
`RegionNullCheck >= 10`	An alert sends when the percentage of rule breaks is greater than or equal to 10%.

Examples of alert conditions for rule scores

Condition alerts for rule scores use a `<ruleName>.score <operator> <score>` syntax.

Example alert condition	Expected result
`RegionNullCheck.score > 0`	An alert sends when one or more points are deducted from the rule score.
`RegionNullCheck.score < 85`	An alert sends when fewer than 85 points are deducted from the rule score.
`RegionNullCheck.score = 20`	An alert sends when 20 total points are deducted from the rule score.

Configuring alert emails

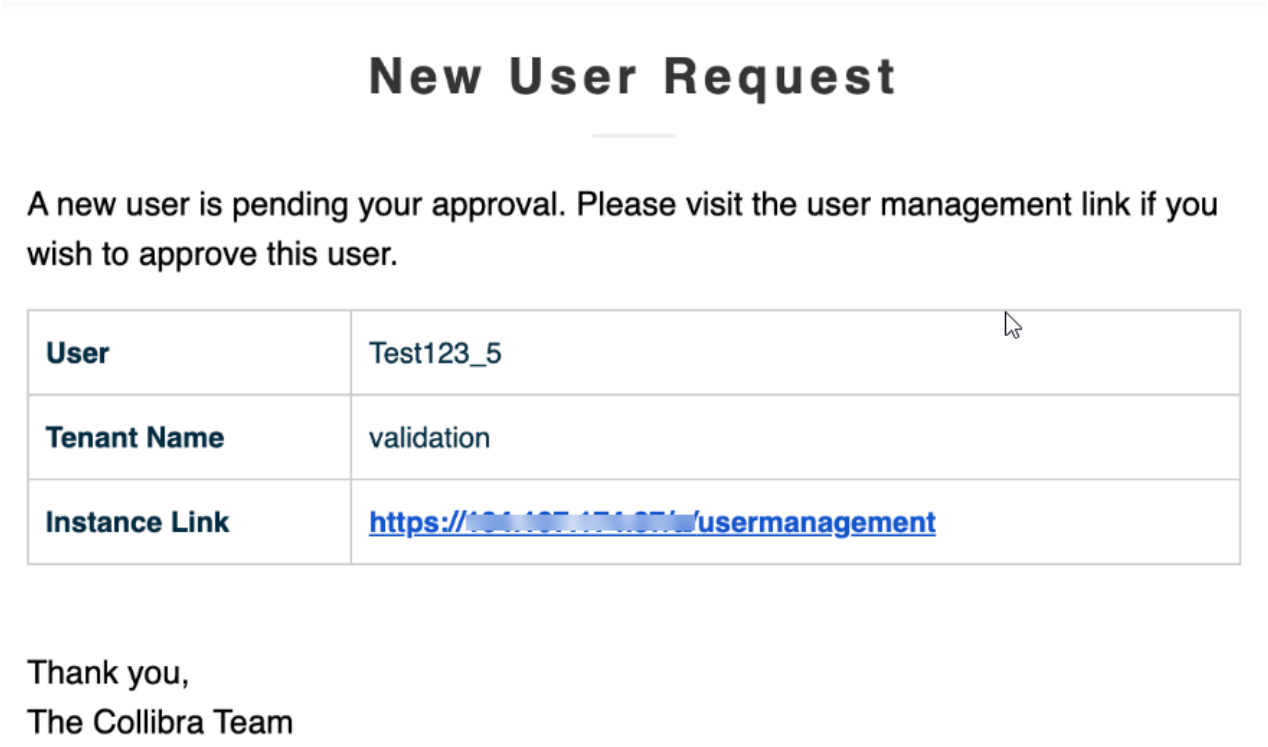
Email alerts let you send emails to specified recipients when an alert condition is met for a given dataset.

Setting Up an Email Server

You can set up the SMTP server using the DQ web application. For more information about setting up an SMTP email server, go to Set up SMTP.

New User Requests

When you receive a new user request, the request includes the username, tenant name, and link to the instance where the request came from, for quick access.



New User Request

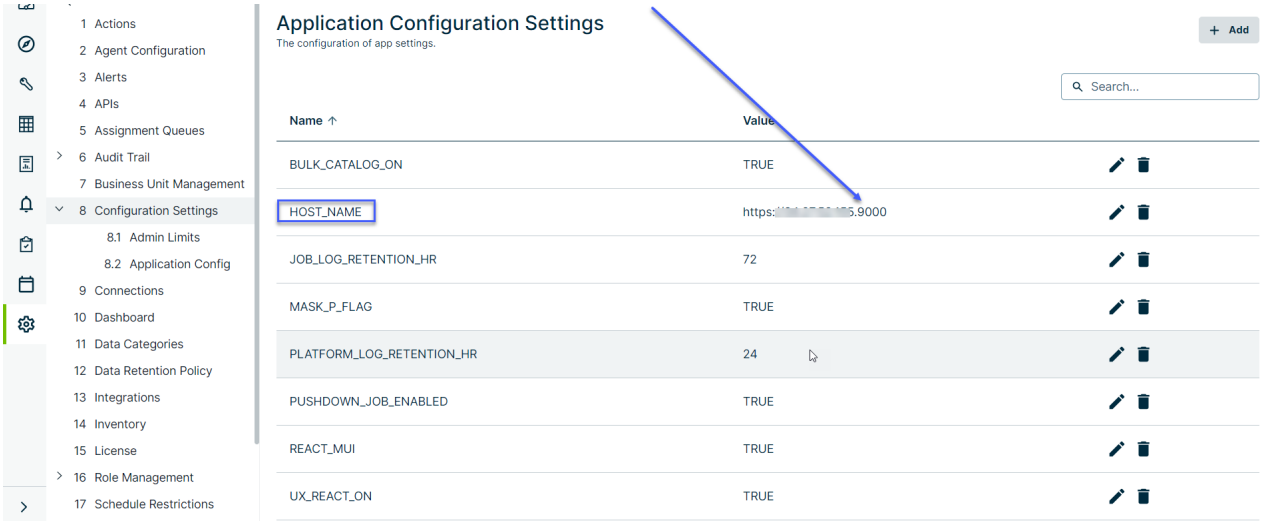
A new user is pending your approval. Please visit the user management link if you wish to approve this user.

User	Test123_5
Tenant Name	validation
Instance Link	https://[redacted]/usermanagement

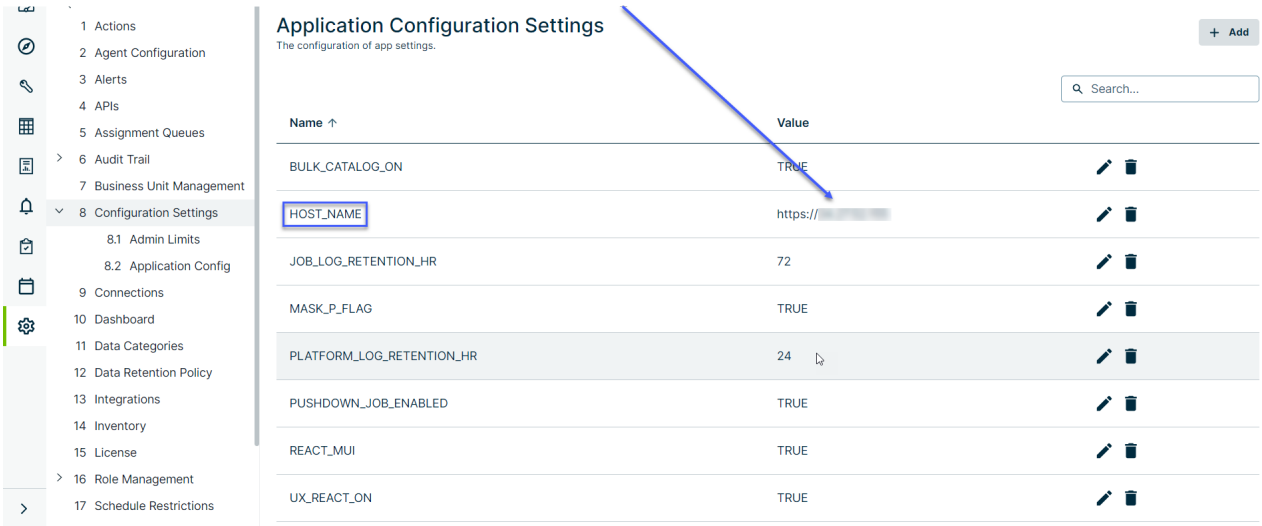
Thank you,
The Collibra Team

> **Note** If the instance comes from a standalone environment, the port number displays in the URL. If it is from another environment, such as Kubernetes, the port number is not included in the URL. See the following screenshots.

Hostname for Standalone Environment



Hostname for Other Environment



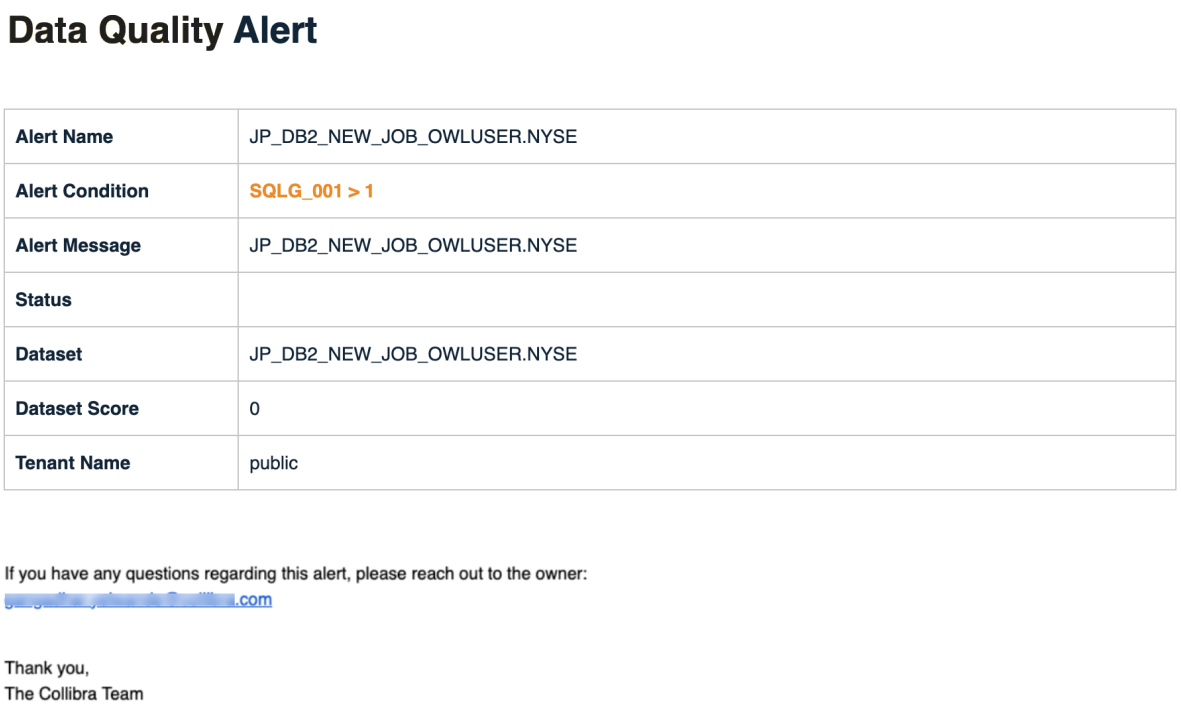
To use the batch name to create a consolidated list of alerts and distribution lists for a set of notifications per dataset, see Email Batch Alerts.

Setting DQ Alerts for Datasets

You can set DQ alerts for datasets so that you are notified based on certain conditions that are triggered on the datasets. Below are some examples of what a dataset email looks like in your inbox.

> **Note** Ensure that the SMTP server was set up properly and that your email client didn't mark the email as spam. For more information about setting up SMTP, go to Set up SMTP.

The following example shows a DQ alert email without links from the instance(s)/page(s).



Data Quality Alert

Alert Name	JP_DB2_NEW_JOB_OWLUSER.NYSE
Alert Condition	SQLG_001 > 1
Alert Message	JP_DB2_NEW_JOB_OWLUSER.NYSE
Status	
Dataset	JP_DB2_NEW_JOB_OWLUSER.NYSE
Dataset Score	0
Tenant Name	public

If you have any questions regarding this alert, please reach out to the owner:
[redacted].com

Thank you,
The Collibra Team

The following example shows a DQ alert email with links from the instance(s)/page(s).

> **Tip** Whenever you click the link for **Findings - Most Recent Run**, you are always directed to the most recent DQ Job run, no matter what date you click the link. The Rules and Profile pages have similar functionality.

Data Quality Alert

Alert Name	JP_DB2_NEW_JOB_OWLUSER.NYSE
Alert Condition	SQLG_001 > 1
Alert Message	JP_DB2_NEW_JOB_OWLUSER.NYSE
Status	
Dataset	JP_DB2_NEW_JOB_OWLUSER.NYSE
Dataset Score	0
Tenant Name	public
Findings - Run which Produced the Alert	http://35.188.59.77:9000/dq/hoot?dataset=JP_DB2_NEW_JOB_OWLUSER.NYSE&runId=2023-04-23 00:00:00.0
Findings - Most Recent Run	http://35.188.59.77:9000/dq/hoot?dataset=JP_DB2_NEW_JOB_OWLUSER.NYSE
Profile	http://35.188.59.77:9000/dq/profile?dataset=JP_DB2_NEW_JOB_OWLUSER.NYSE&runId=2023-04-23 00:00:00.0
Alerts	http://35.188.59.77:9000/alert?dataset=JP_DB2_NEW_JOB_OWLUSER.NYSE
Rules	http://35.188.59.77:9000/dq/rule?dataset=JP_DB2_NEW_JOB_OWLUSER.NYSE&runId=2023-04-23 00:00:00.0

If you have any questions regarding this alert, please reach out to the owner:
[redacted].com

Thank you,
The Collibra Team

DQ Alerts for Failed Jobs

Another scenario is when the DQ Job fails to run or has an exception and, therefore, never gets the chance to score the data or run the alert condition. This is a failed alert that's automatically sent to the email address based on the Admin/SMTP settings defined in the To Email (Default) fields in the Admin console.

Alert Name:	default.infosec_2 (FAILED) **(FAILED)**
Dataset:	default.infosec_2
Condition:	
Msg:	DQ Job did not complete

Setting up the Email Server Programmatically

If you are in a notebook or pipeline, you may prefer to use the Scala/Spark API to create the Email Server.

```
    val emailServer = OwlUtils.createEmailServer("smtp-
relay.sendinblue.com", 587)
    emailServer.setUsername("abc@collibra.com")
    emailServer.setPassword("abc")
    emailServer.setCurrentSet(1)
    Util.EmailServerDaoFactory.delete(emailServer)
    Util.EmailServerDaoFactory.insert(emailServer)
```

Setting Up SMTP Alerts Without a Username or Password

Some alert settings are configurable without requiring a username or password when you set up an email server.

Steps

1. Sign into Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
3. Click **Configuration Settings**, then click **Application Config**.
   » The **Application Configuration Settings** page opens.
4. Click **+ Add** in the upper right corner.

> **Note** All fields in the Add a Custom Configuration dialog box are required.

5. Enter a name and a value, where **Name** is the setting name and **Value** is its corresponding value, such as **TRUE** or **FALSE**. The following table shows the available settings and their descriptions.

Setting name	Default value	Description
mail.debug	**TRUE**	When the value is set to **TRUE**, detailed logging is enabled. When the value is set to **FALSE**, detailed logging is disabled.
mail.smtp.auth	**TRUE**	When the value is set to **TRUE**, the server attempts to authenticate the user using the `AUTH` command. When the value is set to **FALSE**, username and password authentication are turned off.
mail.smtp.ssl.enable	**FALSE**	When the value is set to **FALSE**, SSL is not used to connect to the SMTP server. When the value is set to **TRUE**, SSL is used to connect.
mail.smtp.starttls.enable	**TRUE**	When the value is set to **TRUE** and TLS is supported by the server, this enables the use of the `STARTTLS` command to switch the connection to a TLS-protected connection before issuing any login commands. If TLS is not supported by your mail server, the value of this setting must be set to **FALSE**.
mail.smtp.ssl.trust	*	If set, and a socket factory hasn't been specified, this setting enables the use of a MailSSLSocketFactory. If its value is set to *, all hosts are trusted. However, if the value is set to a whitespace-separated list of hosts, those hosts are trusted. For all other values, trust depends on the certificate the server presents. **Note** Collibra DQ does not recommend using the * default value for this property.

> **Note** These settings are preset to their default values. For example, mail.smtp.auth is preset to **TRUE**.

6. Click **Save**.
7. Repeat steps 4–6 for each setting you want to add.

Working with Alert Notifications

Alert Notifications provide a sortable list of alerts generated by conditions met during dataset runs. This helps you see that email alerts sent successfully and to whom the alert notification was sent. The list of alert notifications is presented in a table format containing the following details.

Column	Description
Dataset	The name of the dataset that triggered the alert notification.
Run Date	The date the job that triggered the alert notification ran.
Alert Name	The name of the alert.
Alert Condition	The condition that was met for the alert to send.
Email Address	The email address of the recipient of the alert email.
Message	A custom message relevant to the alert.
Update TS	The time stamp for when the alert sent.

> **Tip** You can sort the following columns by hovering your cursor over the column headers and clicking the ↑ ↓ icons next to each column name. Additionally, you can search for a dataset by entering its criteria in the Search field in the upper right corner.

Chapter 9

Scheduler

Where are timestamps and time zones in the UI? ..... mcccviii

Setting the time zone ..... mcccx

Admin Configuration ..... mcccxxiii

Prerequisites ..... mcccxxviii

Steps ..... mcccxxviii

What's next? ..... mcccxxix

Prerequisites ..... mcccxxxi

Steps ..... mcccxxxi

What's next? ..... mcccxxxii

Steps ..... mcccxlviii

Steps ..... mcccxlix

Steps ..... mccclii

Steps ..... mccclv

Steps ..... mccclxxxv

Prerequisites ..... mccclxxxviii

Steps ..... mccclxxxix

Steps ..... mcdiii

Steps ..... mcdiv

Prerequisites ..... mcdxvi

Configure Security Settings for Active Directory and LDAP ..... mcdxvi

Required script settings to configure LDAP ..... mcdxxiv

Script Settings for Search Configuration ..... mcdxxv

AD Group to Role Mapping Overview ..... **mcdxxvi**

What's next? ..... mcdxxvi

Mapping AD security groups to roles ..... **mcdxxvii**

Steps ..... mcdxxvii

Mapping security groups to roles when using SSO with SAML ..... **mcdxxx**

Configuring user search filter ..... mcdxxxiv

Steps ..... mcdxxxiv

Known limitations ..... mcdxxxvii

What's next? ..... mcdxxxviii

Steps ..... mcdxxxix

Known limitations ..... mcdxli

What's next? ..... mcdxli

Steps ..... mcdxlii

Set the SAML authentication properties ..... mcdxlv

Download service provider metadata for the IDP ..... mcdli

Steps ..... mcdli

Enable the SAML sign in option ..... mcdlii

Steps ..... mcdlii

Steps ..... mcdliii

Steps ..... mcdlv

Steps ..... mcdlxi

Configure the Azure AD B2C portal ..... mcdlxv

Configure the DQ Application ..... mcdlxvii

Restart the DQ Application ..... mcdlxxii

Test basic login functionality ..... mcdlxxii

Basic role mapping in DQ ..... mcdlxxiv

What's next? ..... mcdlxxiv

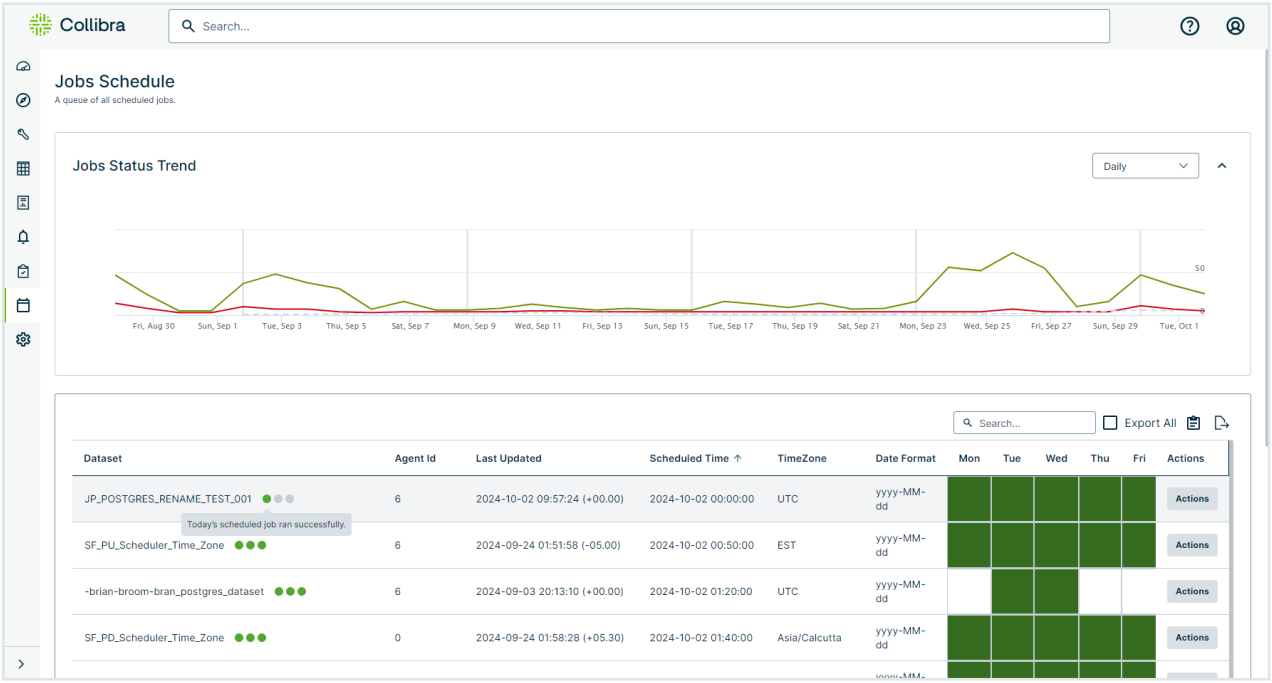
Populating user roles in Azure AD B2C via Microsoft Graph API ..... mcdlxxv

Test login with populated roles ..... mcdlxxix

Security Configuration ..... mcdxcviii

Jobs Schedule

Jobs Schedule shows the historical run status of DQ Jobs and allows you to view and manage their run schedules.



Column	Description
Dataset	The name of your dataset.
Agent Id	The identification number of your agent.

> **Note** Because Pushdown Jobs do not require an agent to process the Job, the Agent Id is always 0.

Column	Description
Last Updated	The datetime of the last processing activity. This timestamp displays in UTC time and includes an offset value of the difference between UTC and the time zone you have set for the dataset. For example, the difference between UTC and Asia/Calcutta (GMT) is 5 hours and 30 minutes, so the timestamp includes an offset value of (+05:30). The format is YYYY-MM-DD HH:MM:SS (+/- HH:MM).
Scheduled Time	The datetime a DQ Job is scheduled to run. This timestamp reflects the time zone you have set for the dataset. The format is YYYY-MM-DD HH:MM:SS.
TimeZone	The time zone in which the DQ Job is scheduled to run. The default time zone is UTC.
Date Format	The format of the run date in yyyy-MM-dd format.
Day	The days of the week a DQ Job is scheduled to run. When a given day displays a green cell, it is scheduled to run on that day. When it displays a white cell, it is not scheduled to run on that day. To edit the frequency, click Actions in the far right column, then Edit.
Frequency	The frequency with which a DQ Job is scheduled to run. To edit the frequency, click Actions in the far right column, then Edit.
Dom	The day of the month (DOM) a DQ Job is scheduled to run. This is only applicable when a DQ Job is configured to run on a monthly schedule.
Rerun	If a DQ Job is scheduled to run on the current day, this option reruns the last attempted DQ Job run.
Active	Shows whether a DQ Job is active for future scheduled runs. Click the icon to change the active status of the scheduled DQ Job. ✓ shows that the defined schedule is active and the DQ Job will run according to it. ⊗ shows that the defined schedule is inactive and the DQ Job will not run according to it.
Actions	Click **Actions** to edit or delete the run schedule of a DQ Job.

Scheduling a DQ Job

After you successfully run a DQ Job, you can schedule it to run automatically. If your monthly or quarterly DQ Jobs typically load after the month or quarter has ended, you can schedule it to run on the day the data lands, but set the offset to the proper run date required for charting or reporting.

Steps

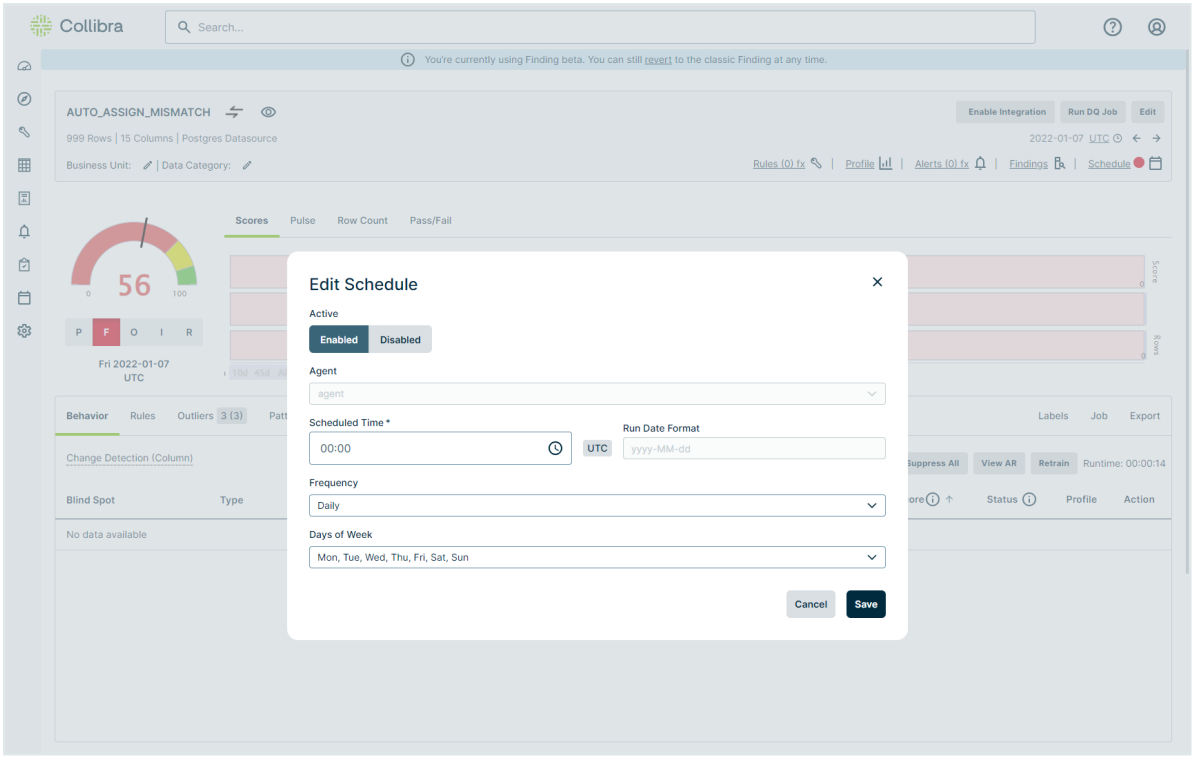
1. On the right side of the metadata bar of a dataset, click **Schedule**.
   » The Edit Schedule modal appears.
2. Enter or select from the options.

Option	Description
Active	Enables or disables the scheduled runs of a DQ Job. Click **Enabled** to enable DQ Jobs to run according to the schedule, or **Disabled** to disable them.
Agent	The agent used to submit your DQ Job for processing. This field is not editable.
Schedule Time	Enter a time in 24-hour clock format, or use the clock icon to click and drag a time. The scheduler checks for scheduled DQ Jobs every two minutes. Therefore, if you want to stagger your DQ Jobs, you need to space their scheduled times two minutes apart.
Run Date Format	The format of the run date in yyyy-MM-dd format. This field is not editable.
Frequency	A dropdown menu that contains options of how frequently your scheduled DQ Job should run.
Daily	Select this option from the Frequency dropdown menu to run your DQ Job every day at the scheduled time and the specified days.
Days of Week	A dropdown menu that contains the days of the week that your DQ Job should run. Select or deselect all that apply.

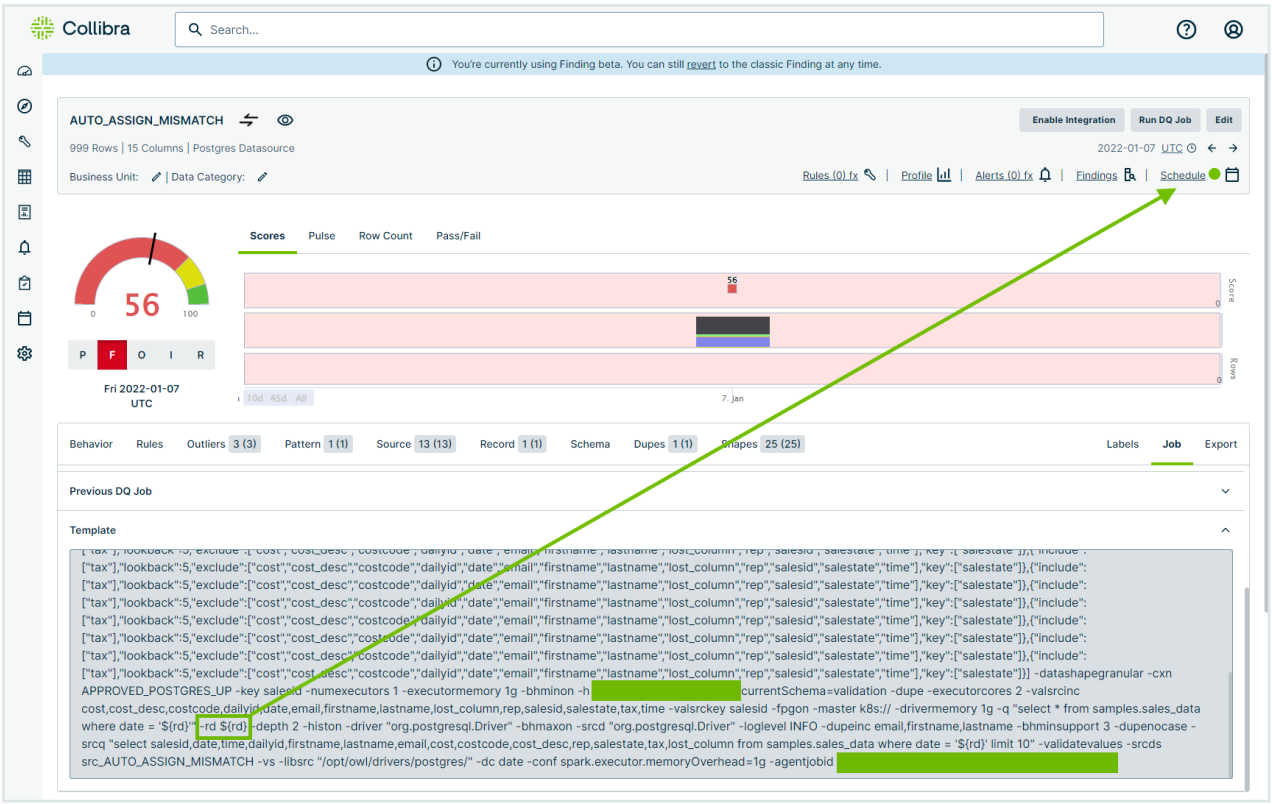
Option	Description
Monthly	Select this option from the Frequency dropdown menu to run your DQ Job every month at a specified time and date.
Days of Month	The day of the month when your scheduled DQ Job runs.
As of (run date)	The date to use as the run date. For example, if you select First of Prior Month and First of Month is set in the Days of Month option, then your run includes data from the previous month on the first day of the month.
Day of (scan date)	The day of the month whose data your scheduled DQ Job includes in the scan. **Note** This option is only available when you select **Day of** in the As of (run date) option.
Quarterly	Select this option from the Frequency dropdown menu to run your DQ Job every quarter at a specified time and date.
Days of Month	The day of the month when your scheduled job runs.
As of (run date)	The date to use as the run date. For example, if you select First of Prior Month and First of Month is set in the Days of Month option, then your run includes data from the previous month on the first day of the month.
Day of (scan date)	The day of the month whose data your scheduled DQ Job includes in the scan. **Note** This option is only available when you select **Day of** in the As of (run date) option.
Month of Quarter	The specific month of a given quarter whose data your scheduled DQ Job includes in the scan.
Start Month of Q1	The month to use as the starting place for the quarter.

3. Click **Save**.

See screenshot...



Tip You can use the -rd variable $ in your Template query on the Findings page to set dynamic dates or date ranges for your scheduled DQ Job, as shown in the following screenshot.



Updating the time zone

This topic shows you where in the application time zones and timestamps are and how to update time zones from the Findings page.

While scheduled DQ Jobs run according to the default UTC time zone, you may find you want to configure them to run according to the runId of a specific time zone.

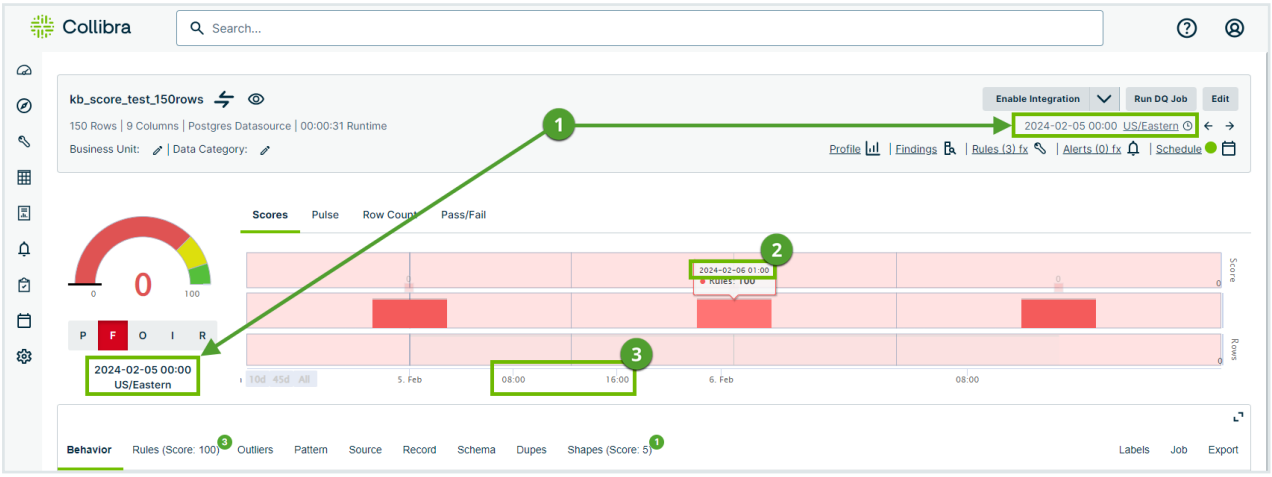
Where are timestamps and time zones in the UI?

RunId timestamps display in their respective time zones in 24-hour time format in the following locations:

- Metadata bar
- Findings
- Dataset Manager
- Alert Notifications

- Jobs
- Jobs Schedule

Depending on where you are in the application, timestamps may represent different things. In the below screenshot of the Findings page, three different timestamps are represented at the same time.



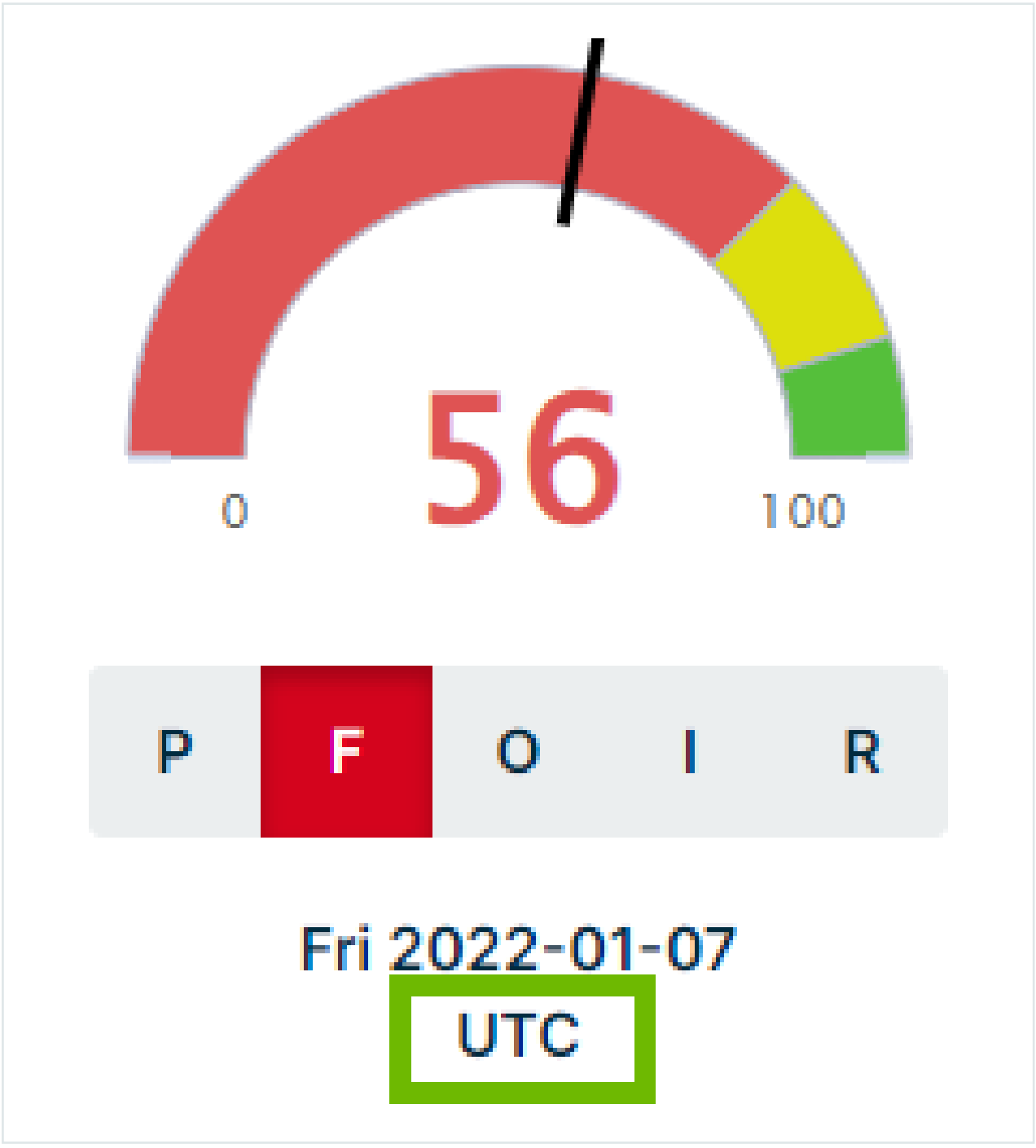
The following table explains the various timestamps in the screenshot above.

No.	Description
1	The first timestamp, on the right side of the metadata bar and below the score meter, indicates a runId date of 2024-02-05 00:00 (February 5, 2024 at midnight). When there is only one DQ Job run for a given runId date, such as the lone run on 2024-02-05, the timestamp always displays as 00:00 (midnight) after the date. **Note** Changes to the time zone do not affect the runId the date and time on the Findings page.
2	When there are multiple DQ Job runs on a given runId date, the timestamp reflects the time each DQ Job ran, for example, 01:00 or 10:45.
3	The times along the x-axis use random values that do not reflect or relate to the runId of a given DQ Job run. Instead, these are meant to provide visual context of when certain DQ Jobs ran.

Setting the time zone

1. Below the score meter on the Findings page of a dataset or on the right side of the metadata bar, click the time zone.
   See screenshot...[1]

1



> **Note** The default time zone is UTC, but if an alternative time zone was previously set, the one that displays may be different.

» The Update Time Zone modal appears.

2. Select an option from the dropdown menu.
3. Click **Save**.

> **Note** Scheduled DQ Jobs now use the runId of your updated time zone.

Schedule Restrictions

Admin users can manage schedule restrictions from the Admin Console ▸ Schedule Restrictions.

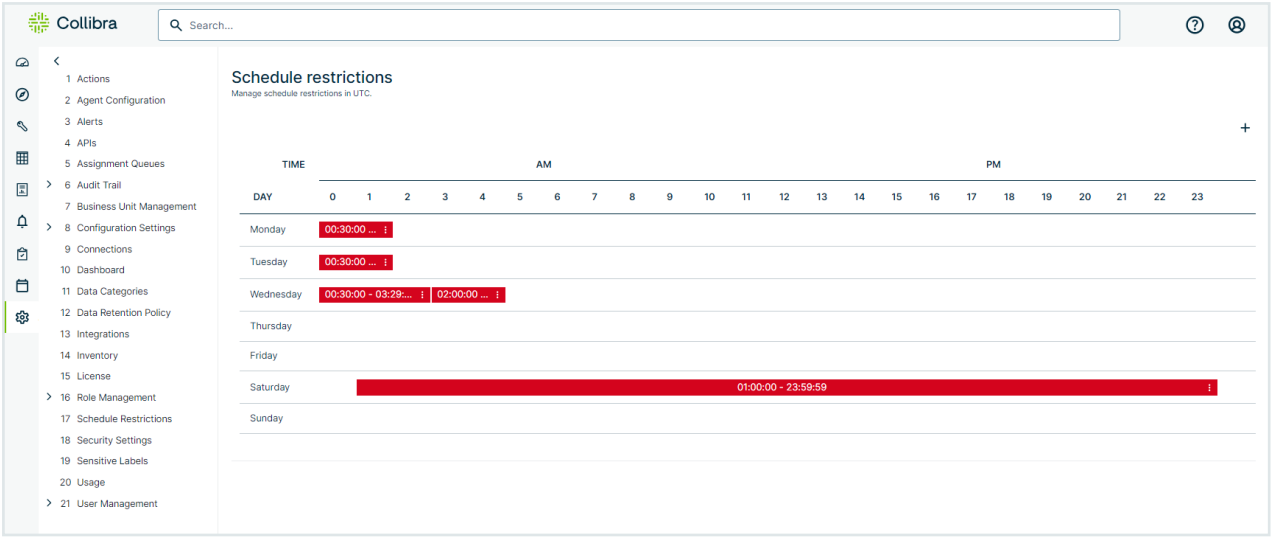
> **Note** All times on the Schedule Restrictions page reflect the UTC timezone.

In Standalone deployments of Collibra DQ, you can enable the Job Scheduler by setting the environment variable `SCHEDULE_ENABLED=TRUE` in the **owl-env.sh** file. This is enabled out-of-the-box, so if you need to turn the Job Scheduler off, you can set the value to `FALSE` instead.

In Cloud Native deployments of Collibra DQ, the **Web ConfigMap** contains the `SCHEDULE_ENABLED=true` variable.

Add a schedule restriction

When you don't want automated jobs to run during business hours or during a particular day/time, you can set off-peak times to prevent authorized users from selecting them when scheduling a job.



1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **User Management** and then click **Schedule Restriction**.
4. On the right side of the page, click .
5. From the **Add schedule restriction** dialog, use the dropdown menus to select a day of the week to restrict and a range of times.
6. Click **Submit**.
   » The restricted day and times appear on the Schedule Restrictions page. To edit a restricted time, click the and update the day or range of times.

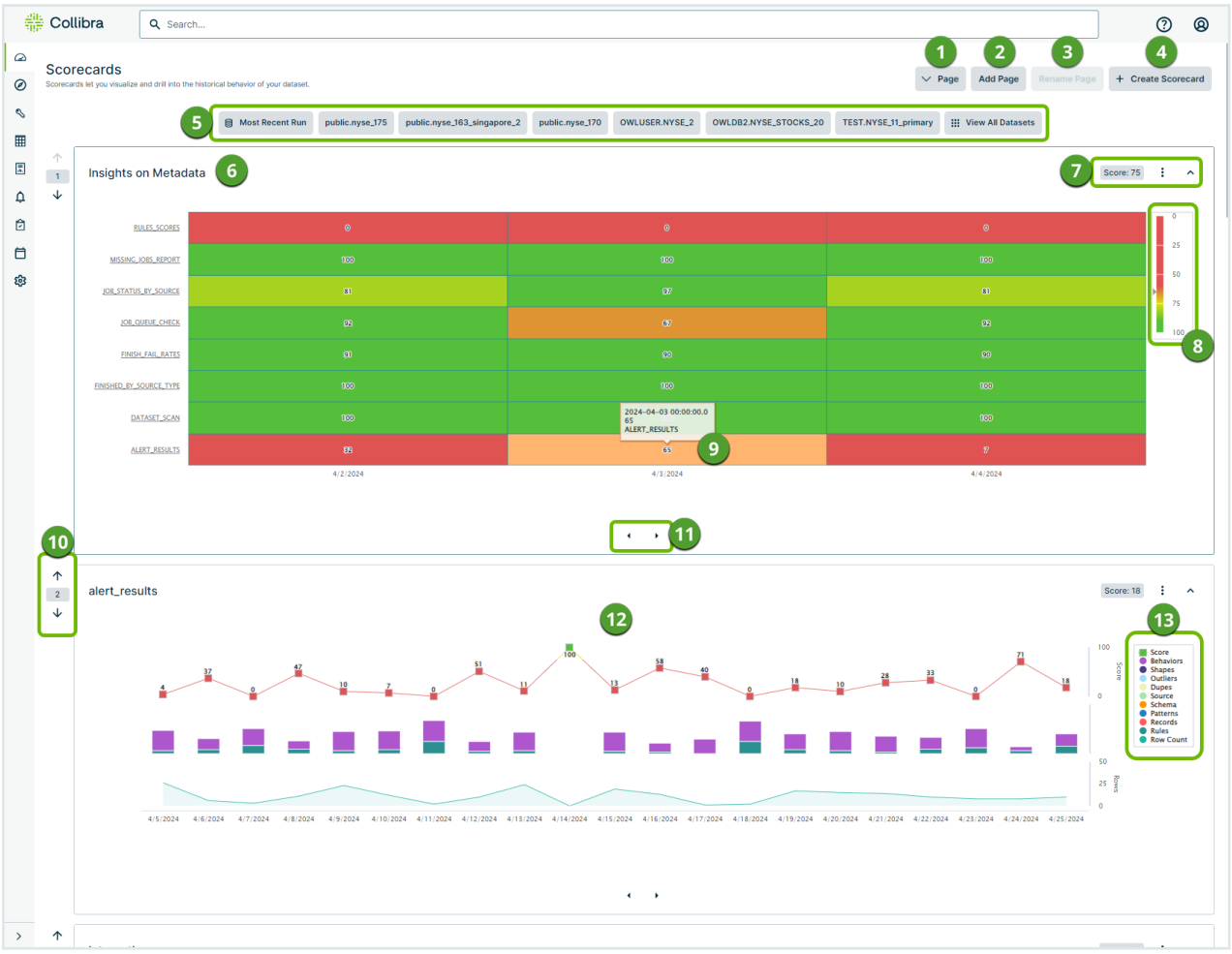
DQ Views

Scorecards

Scorecards provide a historical view of the performance of your datasets, which lets you visualize the health, consistency, and evolution of a dataset over a user-defined date range. By understanding the health of your data through scorecard analysis, you can more accurately schedule routine health checks and help protect downstream process consumers from pulling erroneous data into their models.

There are two ways to use scorecards:

- Create a scorecard for a single dataset for an overview of all of the layers discovered in particular runs.
- Create a scorecard for multiple datasets at once as a heatmap representing the health of particular runs and giving you an elegant way to compare their scores and quickly access the findings.



Number	Description
1	The page dropdown menu allows you to search for and select any available page of scorecards in your environment. A page can contain one or multiple scorecards. When on a specific page, you can click **+ Create Scorecard** to add new scorecards to it.
2	Allows you to create a new page. Click Add Page and enter a unique name in the text input field, then click Save. Your new page is available in the page dropdown menu.

Number	Description
3	Allows you to optionally rename a page containing scorecards. Select a page from the page dropdown menu to enable the Rename Page button, then enter a new page name. **Note** When renaming a page, your new page name must be unique, meaning it cannot be the same as another existing page. If you attempt to save a page with a name that already exists, an error message will appear.
4	Click **+ Create Scorecard** to create a new scorecard based on one or more datasets in your environment.
5	The dataset navigation buttons provide quick access to the most recent dataset run, frequently run datasets, and Dataset Manager.
6	An example of a scorecard based on a compilation of multiple datasets. This scorecard option shows a heatmap of the selected datasets and their scores on the runs from certain days. This option is an excellent way to compare the performance of different datasets over a common date range.
7	The upper right corner of all scorecards show three components: • A composite score of the dataset(s) you select. • A three dots menu from which you can edit or delete your scorecard. • A carat icon to expand or contract a scorecard.
8	When you hover your cursor over a score in the scorecard, this meter shows the score for the corresponding run.
9	Hover your cursor over a score to see the run date and dataset name on scorecards with multiple datasets, or the run date and score for scorecards with a single dataset. You can also click the score to open the Findings page of that run.
10	Click the arrows to display data for the available date ranges.
11	Click the up or down arrow to change the order in which the scorecard displays on the page.
12	An example of a scorecard based on a single dataset. This scorecard option shows an overview of the score and detected data quality layers from a certain day's run.

Number	Description
13	A legend that corresponds with the data quality layers depicted on the scorecard.

What's next?

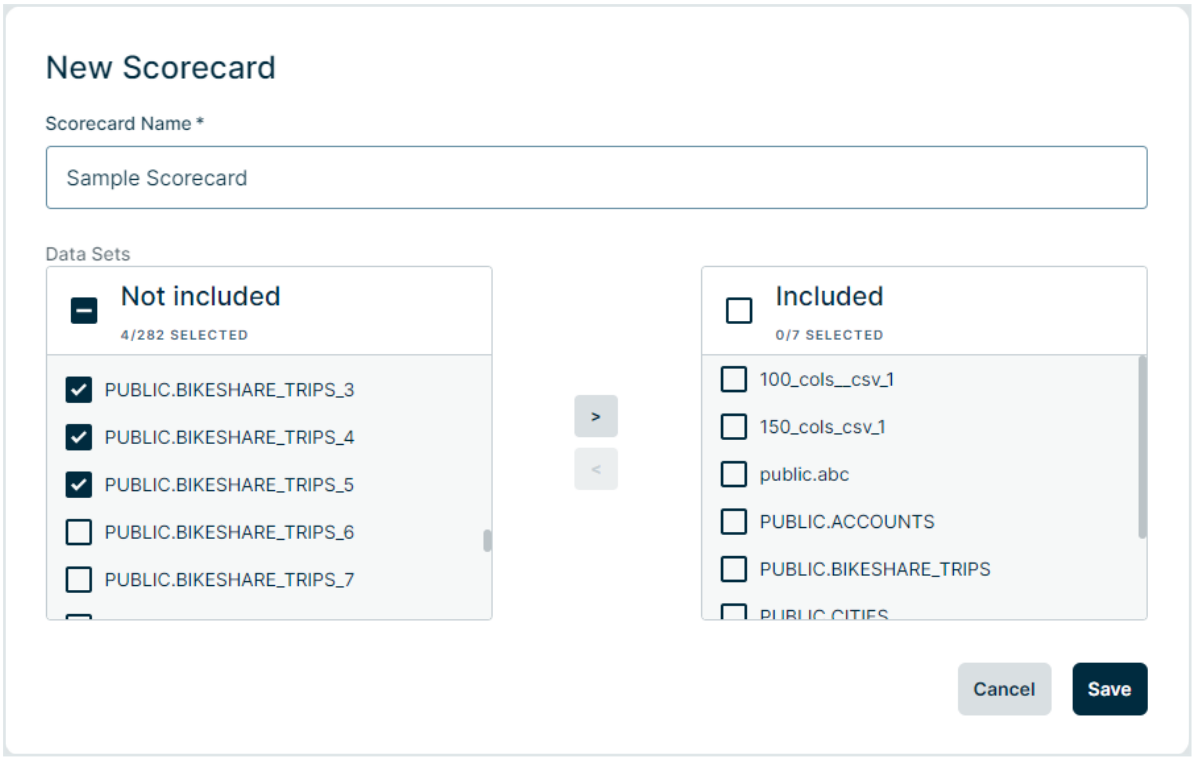
Learn how to create a scorecard to view blindspots and scoring trends in your data.

Create a scorecard

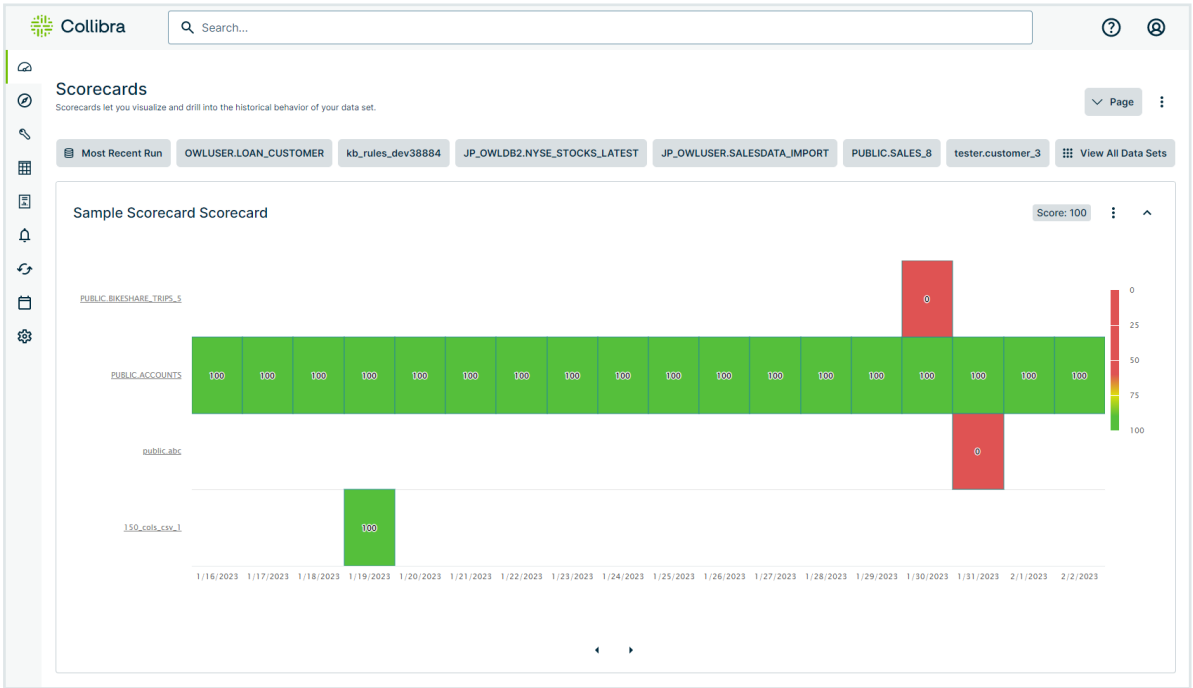
If you want to view the performance and health of specific datasets, you can create a scorecard.

Steps

1. On the left navigation panel, click , then select Scorecards.
   » The Scorecards page opens.
2. In the upper right corner, click **+ Create Scorecard**.
   » The New Scorecard modal appears.
3. In the required Scorecard Name input field, give your scorecard a unique name.
4. On the Not included side of the Datasets selector, select one or more datasets from the checkbox options.
5. Click > to move the selected dataset(s) to the Included side of the Datasets selector.
   » The selected datasets are now included in your scorecard.



6. Click **Save**.
   » Your scorecard is created.

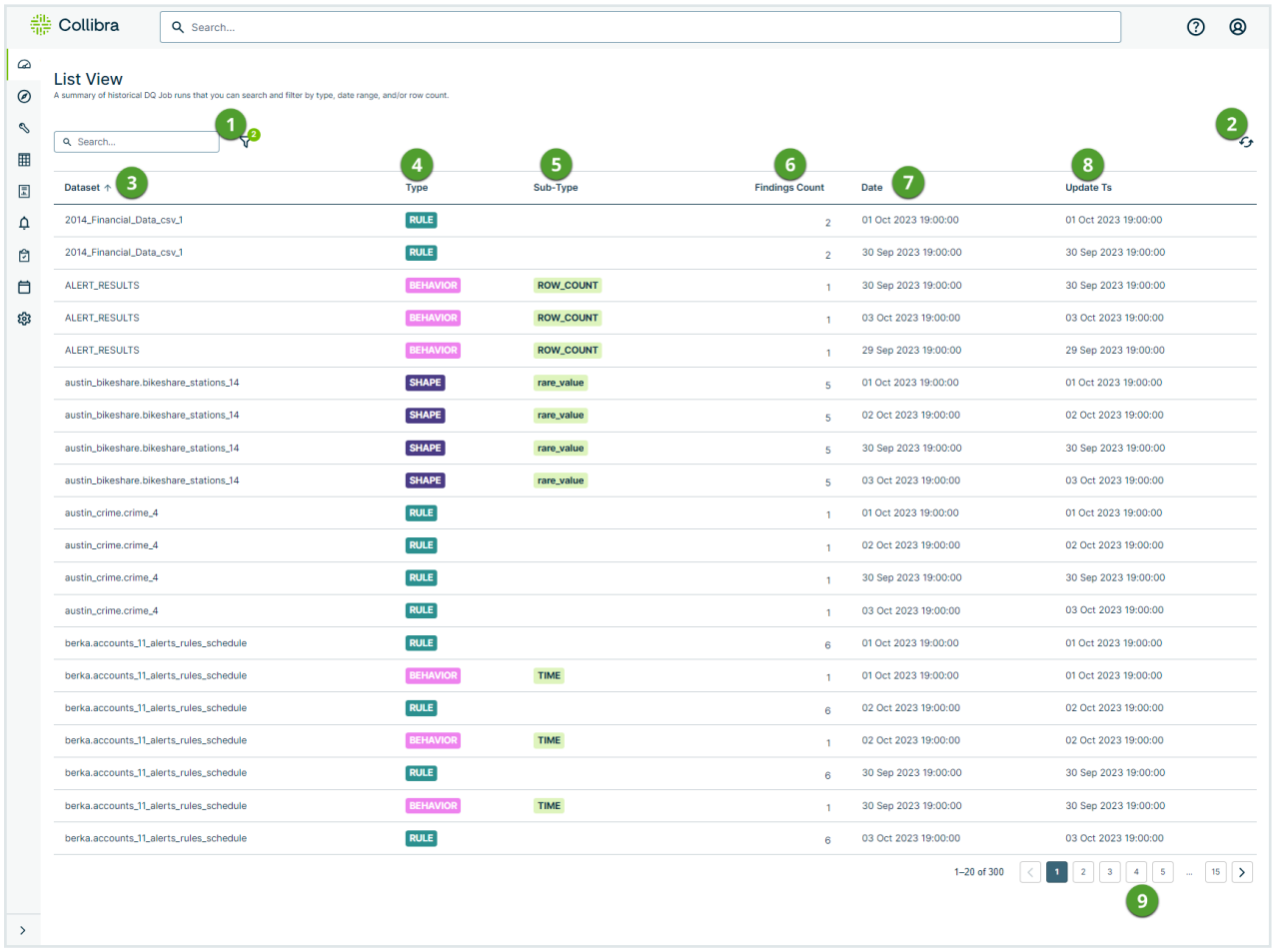


> **Tip** To remove a dataset from your scorecard, select it from the Included section and then click <.

List View

List View provides a list of findings across historical job runs that you can search and filter by type, date range, and/or row count. When you have a multitude of data quality issues in your environment, the sheer quantity can be overwhelming. List View can help you manage this.

Search or filter issues by data quality finding type, date range, and/or row count limit to identify the issues that matter most to you and your organization. When you reduce the issue count to only those that meet your search or filter criteria, you can easily identify the issues that you need to prioritize and move into a remediation queue.



Number	Element	Description
1	Search and filter	Use the search bar to find items that match your criteria. The search bar can locate items in the Dataset, Type, and Sub-Type columns. For example, to find items with the rare_value sub-type of shape findings, enter "rare_value" into the search bar. You can also use the to create filters based on type of data quality finding, date range, and/or record limit.
2	Refresh	Refreshes the list view table.
3	Dataset	The name of the dataset where a finding exists.
4	Type	The type of finding in the dataset. For example, Behavior, Rule, and Shape.
5	Sub-Type	The sub-type of the finding. For example, a Behavior finding might include Mean Value and Time sub-types.
6	Findings Count	The number of findings in a given run.
7	Date	The timestamp of when the dataset ran in YYYY-MM-DD HH:MM:SS format.
8	Update Ts	The timestamp of the last processing activity in YYYY-MM-DD HH:MM:SS format.
9	Page selector	Select a page number or use the arrows to populate more pages (when available).

Tip Click the name of the dataset or the timestamp in the Date column to open the Findings page.

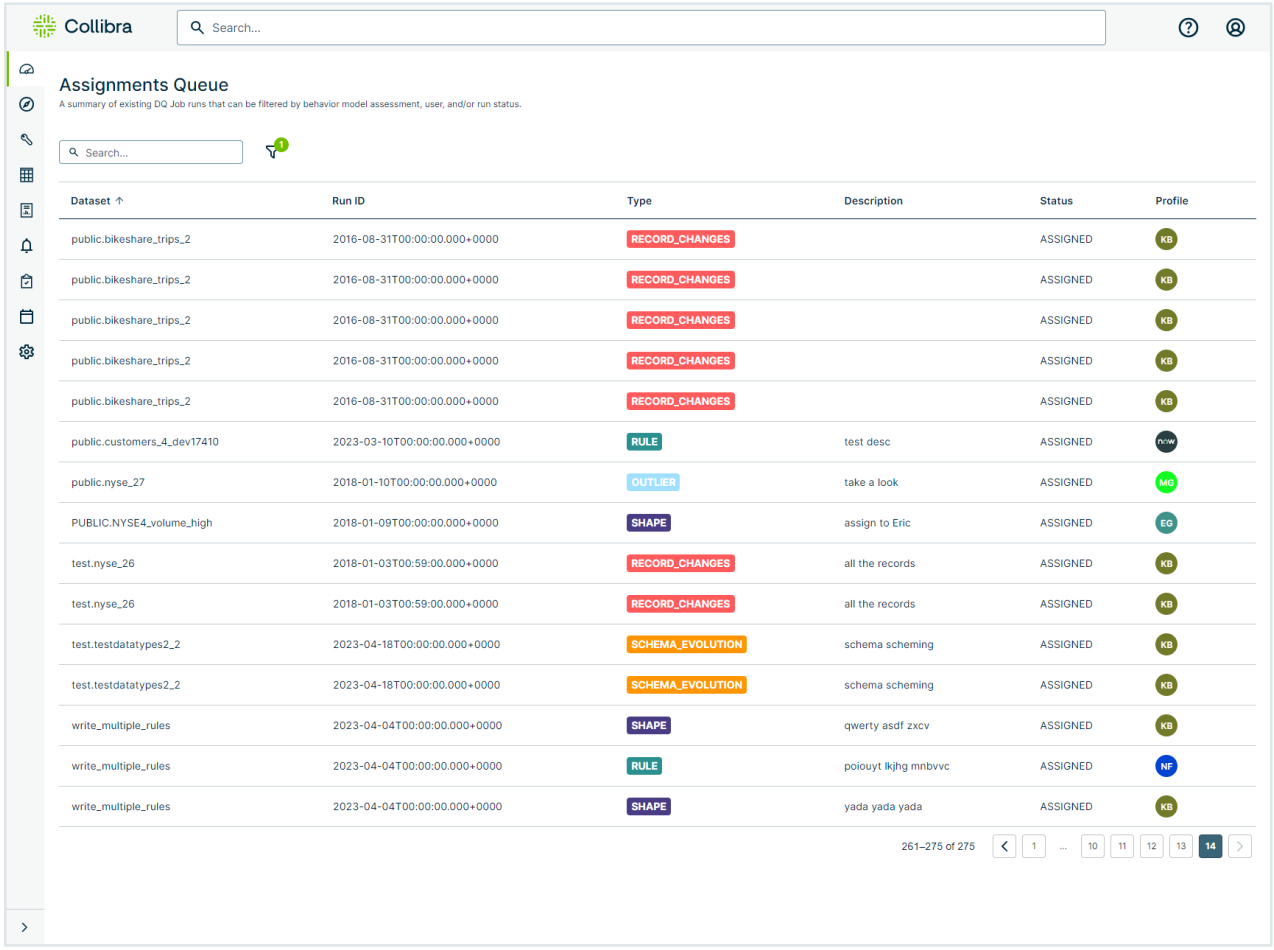
Assignments

The Assignments Queue is a summary of all job runs in your environment where Collibra DQ detected a finding. You can filter these findings by finding type, assignment status, date range, and the user to whom the finding is assigned. When Collibra DQ detects a potential quality

issue, it can be important to assign it to someone in your organization to review and determine whether the finding is legitimate.

When the results of a data quality check return a finding, such as an unexpected outlier, you can assign it to a member of your organization for validation. The assignee can then use the Assignments Queue to track all of their assigned findings. From here, the assignee can click the dataset or run ID of the finding assigned to them to open the Findings page where they can either validate, invalidate, or resolve the finding(s).

The Assignments Queue also allows you to filter all findings without an assignee, which can help to ensure that all datasets in your environment are actively managed. For instance, if a scheduled job ordinarily runs without any day-over-day issues but then detects an unusual schema evolution on the latest run, it may be beneficial to assign someone to validate it.

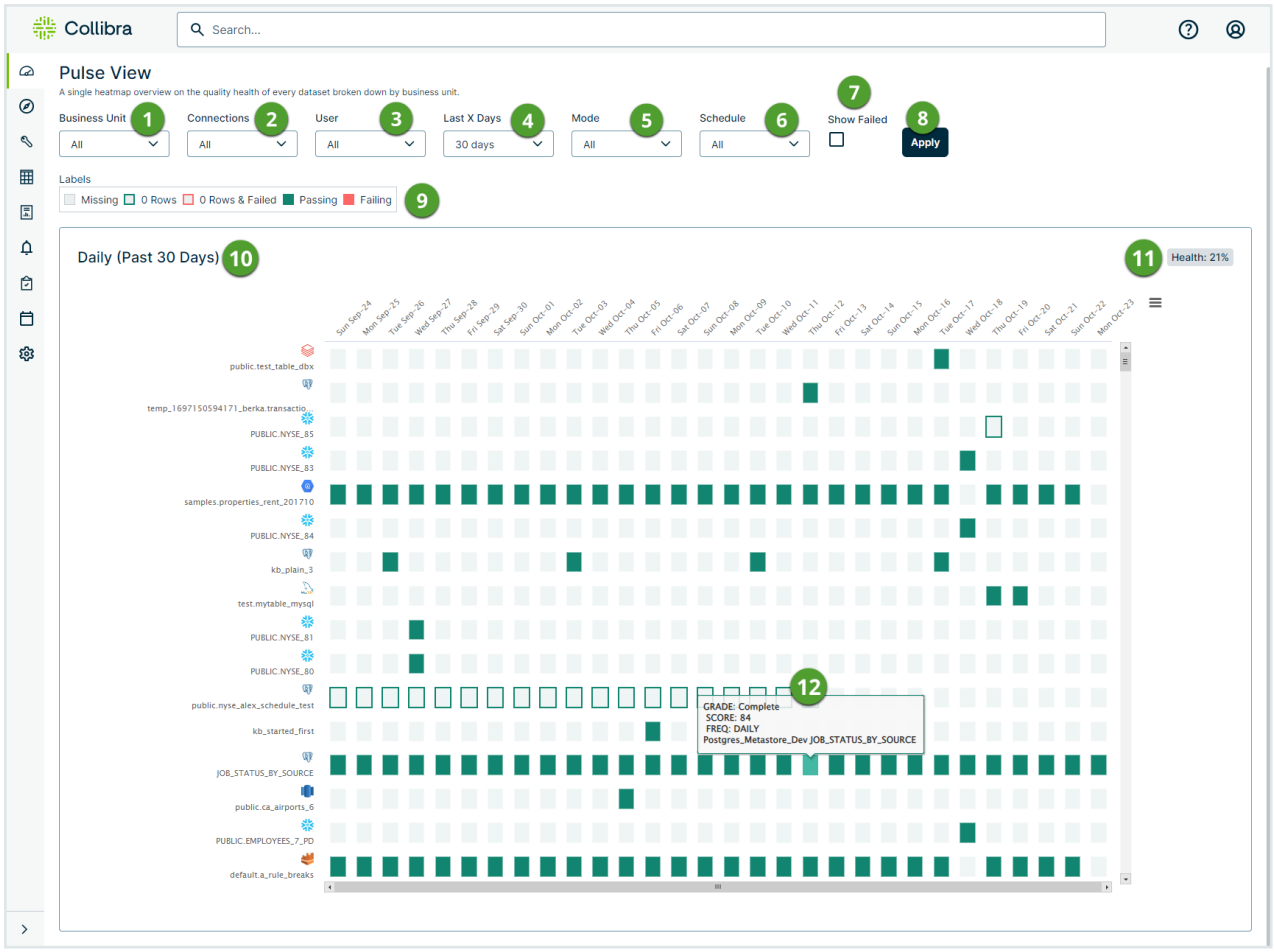


Column	Description
Dataset	The name of the dataset with a finding.
Run ID	The run date of the dataset with a finding.
Update Ts	The timestamp of the last processing activity in YYYY-MM-DD HH:MM:SS format.
Type	The type of finding, for example, Behavior, Rule, and so on.
Description	A description given to a finding during the validation process. If a description was not provided during the validation process, this column is empty.
Status	The assignment status of a finding. **Assigned** - The finding is assigned to the user listed in the Profile column. **Unassigned** - The finding is not assigned to any users or ServiceNow. **Verified** - The finding was validated by a user on the Findings page, but a user was not assigned to it. A "verified" status corresponds to an "acknowledged" status on the Findings page.
Profile	The user or ServiceNow integration assigned to the finding.

Pulse View

The Pulse View provides a snapshot of DQ jobs that were executed, whether manually or on a schedule. On the Pulse View dashboard, you can view heat maps of the health of your jobs, and filter by the following criteria:

- Business Unit
- Connections
- User
- Last X Days
- Mode
- Schedule
- Failed Runs



The following screenshot shows an example of the type of data you can view when you hover your cursor over a heat map label, for example, a grade, score, and scheduled frequency.



Number	Element	Description
1	Business Unit	Filter by business unit.
2	Connections	Filter by connection name.

Number	Element	Description
3	User	Filter by username.
4	Last X Days	Filter by the last 9, 14, 30, or 60 days.
5	Mode	Filter by draft, published, or all modes.
6	Schedule	Filter by jobs scheduled to run daily, monthly, or display all.
7	Show Failed	When you select this check box, only the runs that failed display. When this option is not selected, every job that ran displays, whether the score was passing or failing.
8	Apply	Applies your selected filters to the heat map.
9	Labels	
10	Pulse view heat map	A heat map of all the datasets in your environment and their run details.
11	Health	A composite score of the health of dataset runs in your environment.

Label	Label icon	Description
Missing		A run that is missing either failed at run time or was never created, started, or picked up.
0 Rows		A run that is passing or complete and has resulted in 0 rows of data.
0 Rows & Failing		A run that is failing or incomplete and has resulted in 0 rows of data.
Passing		A run that is passing or complete and has resulted in rows of data.
Failing		A run that is failing or incomplete and has resulted in rows of data.

Number	Element	Description
12	Run information	Hover your cursor over a specific run on the heat map to show its grade, score, scheduled frequency, and the name of the dataset.

Admin Configuration

This section includes details about working with the Admin Console and provides additional guidance on other admin settings and configurations.

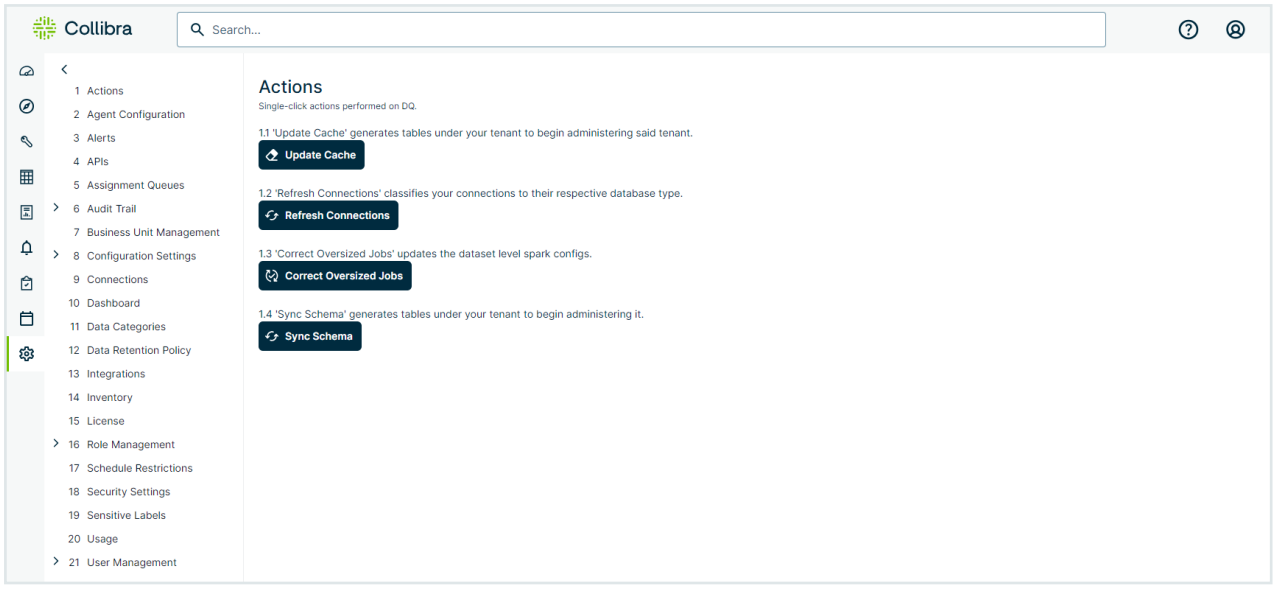
Admin Console

The Admin Console provides administrators the ability to control system-wide security settings, user and dataset roles, connect to data sources, and more.

Actions

Actions provides single-click options for administrators to apply globally across Data Quality & Observability Classic. Available actions include:

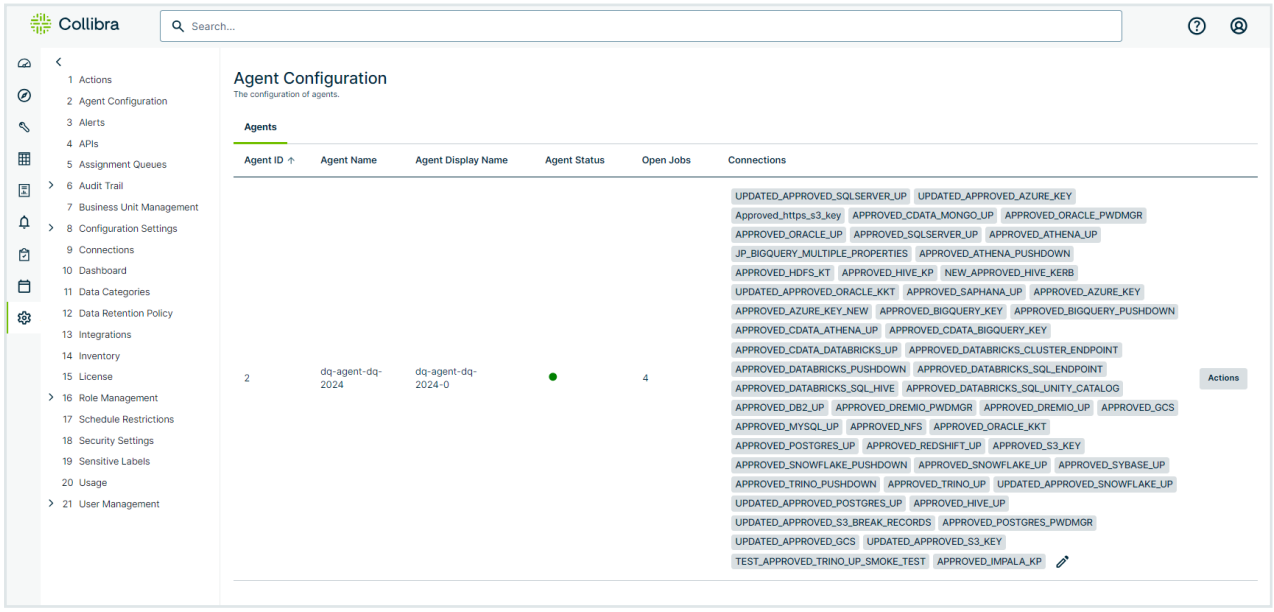
- Update Cache
- Refresh Connections
- Correct Oversized Jobs
- Sync Schema



Agent Configuration

Agent Configuration displays the setup and environment details of each DQ agent, both active and inactive, in Collibra DQ, allowing administrators to:

- Edit or delete agents.
- Map connections to an agent.
- View agent IDs, agent names, agent display names, agent status, open jobs, and connections mapped to an agent.



> **Tip** You can learn more about configuring agents in Set up a DQ agent and how to map connections to agents configure agents in Map data source connections to a DQ agent.

The following table provides descriptions of the columns present on the Agent Configuration page.

Column	Description
Agent ID	The numerical identifier of an agent. For example, *6*. This field auto-generates at the time of creation and cannot be edited.
Agent Name	The unique name of an agent. This field auto-generates at the time of creation and cannot be edited.
Agent Display Name	The descriptive name of an agent that displays anywhere agent information is present in the DQ Web App. You can customize the Agent Display Name to make it easier to identify it. **Tip** There are no character restrictions for the Agent Display Name field, but it is best practice to use only alphanumeric characters, hyphens, and underscores.
Agent Status	Shows whether an agent is online and ready to use, or offline and unavailable for use. ● indicates the agent is online. ● indicates the agent is offline.

Column	Description
Open Jobs	Shows the number of jobs that have been submitted or are running but have not yet completed. When there are open jobs, you can click the number to view the details on the Open DQ Jobs modal. These details include: • jobId • agentId • dataset • runId • urnIdStr • status • activity • activityStatus • cmdLine • description • jobException • username • updtTs • updtTsStr • pid • remoteJobId • batchId • owlcheckQJobId
Connections	The connections that are mapped to an agent. When a connection appears in this column, Pullup jobs created on data within it can use the agent job submission and processing. Click the to edit the connection mapping.

Alerts

Alerts are notifications that signal important data quality events and issues as they occur. You can configure alerts to send via email to specific users or to generic webhooks.

Alert Configuration

Simple Mail Transport Protocol, or SMTP, is an internet standard for email transmission. Collibra DQ allows you to configure a single SMTP server to send alerts to the attention of the

dataset owner in case a specific condition is met, such as:

- Data Quality Score is below a specific threshold.
- Row Count is below a specific threshold.
- If a rule is triggered.

Field	Description
SMTP Host	The name or the IP address of the SMTP server.
SMTP Port	The port used by the SMTP server.
SMTP Username	The username or the account that is configured on the SMTP server for use.
SMTP Password	The password of the SMTP server username or account.
To Email (Default)	The sender email address.
Reply Email	The reply-to email address.
Include Links	Links to the specific instance(s)/page(s) where the alert came from in the DQ alert email. If you select this check box, the links to the specific instance(s)/page(s) where the alert came from display in the Collibra DQ alert email. This allows you to quickly access the environment(s), from which the alert originated. **Note** When you enable the **Include Links** functionality, you must also paste in your host address (e.g., https://x.x.x.x) into the HOST_NAME value, found in the application configuration settings. **Important** Be aware that some email servers, such as Microsoft Exchange, require the **SMTP username** and **Reply email** fields to be the same.
Remove Data Quality & Observability Classic from email subjects	Removes "Data Quality & Observability Classic" from the subject of alert emails. When you select this option, the subject of the alert email is the name of the alert.
Alert Email Signature	Enter a custom email signature to include at the end of all alert emails.

What's next?

- Working with the Alert Builder
- Adding dataset alerts

Configure SMTP

This topic shows you how to configure an SMTP server from the Alerts page to allow Data Quality & Observability Classic to send alerts to recipients via email.

Prerequisites

- You have the **ROLE_ADMIN** role.
- You have an SMTP server available outside of Data Quality & Observability Classic.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Alerts** from the list, then **Alert Configuration**.
   » The **Alerts** page opens.
4. Enter the required information.

Field	Description
SMTP Host	The name or the IP address of the SMTP server.
SMTP Port	The port used by the SMTP server.
SMTP User-name	The username or the account that is configured on the SMTP server for use.
SMTP Password	The password of the SMTP server username or account.
To Email (Default)	
Reply Email	The reply-to email address.

Field	Description
Include Links	Links to the specific instance(s)/page(s) where the alert came from in the DQ alert email. If you select this check box, the links to the specific instance(s)/page(s) where the alert came from display in the Collibra DQ alert email. This allows you to quickly access the environment(s), from which the alert originated. **Note** When you enable the **Include Links** functionality, you must also paste in your host address (e.g., https://x.x.x.x) into the HOST_NAME value, found in the application configuration settings. **Important** Be aware that some email servers, such as Microsoft Exchange, require the **SMTP username** and **Reply email** fields to be the same.

5. Click **Save**.

What's next?

- Working with the Alert Builder
- Adding dataset alerts

Alert Webhook Configuration

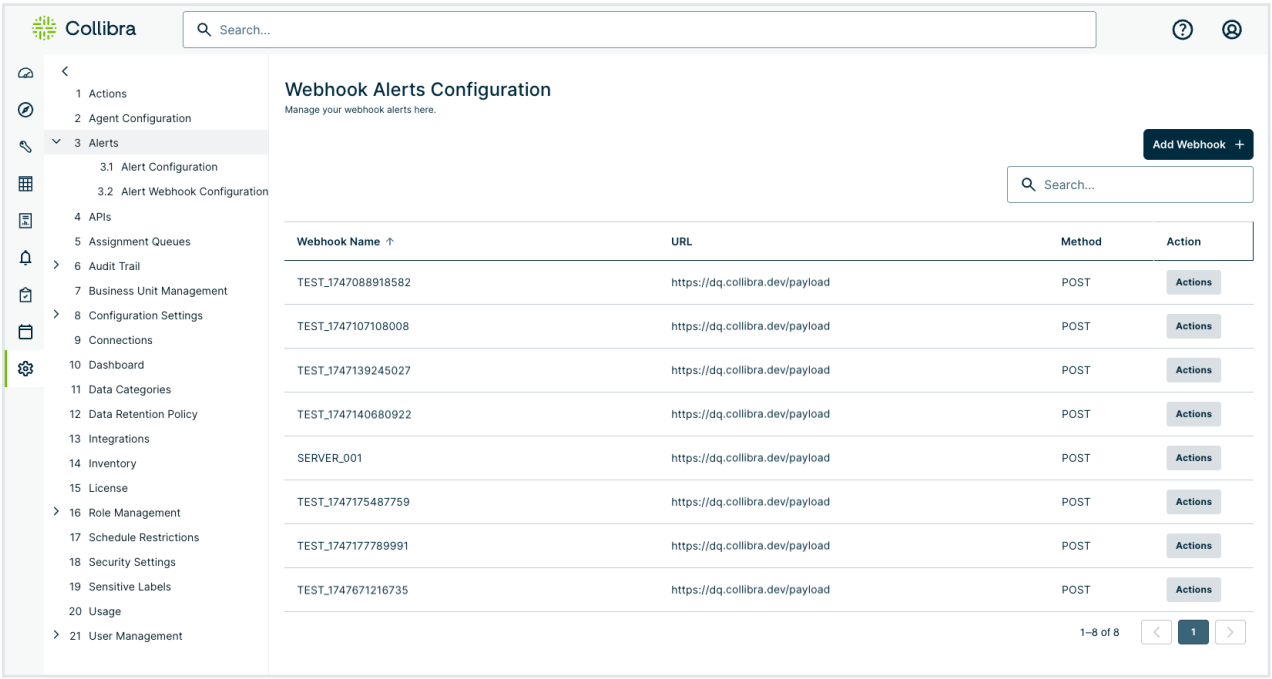
Webhooks allow Data Quality & Observability Classic to communicate data quality events to a web server in real time. When a data quality event, such as an alert condition being met, Data Quality & Observability Classic sends an HTTP POST or PUT request in the form of a JSON payload to the specified URL. The receiving URL, which can be an endpoint in an application such as Microsoft Teams or Slack, processes the alert based on how you have configured handle such requests outside of Data Quality & Observability Classic.

Webhooks offer the following benefits over SMTP:

- Instantaneous delivery.
- Improves downstream automation.

- JSON format provides dynamic scalability and easier integrations with other systems.

> **Note** You cannot use webhooks for global alerts. However, this will feature will be available in a future release. Currently, you can configure job-level alerts to capture runtime events.



Column	Description
Webhook Name	The name you assign to a webhook to help you identify and use it in Data Quality & Observability Classic.
URL	The location where JSON payloads containing alert information are sent.
Method	The API request method. The options are POST (default) and PUT, which correspond to the following endpoints: • **POST v2/webhook** - The endpoint used to create a webhook. • **PUT v2/webhook** - The endpoint used to edit an existing webhook.
Action	Click **Actions** to edit or delete a webhook.

What's next?

- Add a webhook
- Working with the Alert Builder
- Adding dataset alerts

Add webhooks

This topic shows you how to add a generic webhook from the Webhook Alerts Configuration page.

Prerequisites

- You have the **ROLE_ADMIN** role.
- You have an endpoint URL available outside of Data Quality & Observability Classic that can handle HTTP POST and PUT requests from Data Quality & Observability Classic.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Alerts** from the list, then **Alert Webhook Configuration**.
   » The **Webhook Alerts Configuration** page opens.
4. Click **Add Webhook** in the upper-right corner.
   » The Add Webhook dialog box appears.
5. Enter the required information.

Field	Description
Webhook Name	The name you assign to a webhook to help you identify and use it in Data Quality & Observability Classic. This name does not need to match the naming conventions of your systems outside Data Quality & Observability Classic. **Note** Webhook names cannot contain spaces.

Field	Description
URL	The location where JSON payloads containing alert information are sent.
Method	The API request method to the service associated with the URL. Select either POST (default) or PUT.
Authentication Type	Optional. The method that the service associated with the URL uses to authenticate each API request.

> **Note** Selecting an authentication type is optional. However, if you select **Username/Password**, the **Username** and **Password** fields are required. Currently, only the **Username/Password** authentication type is supported.

Username/Password	
Username	Username credential to the service associated with the URL.
Password	Password credential to the service associated with the URL.

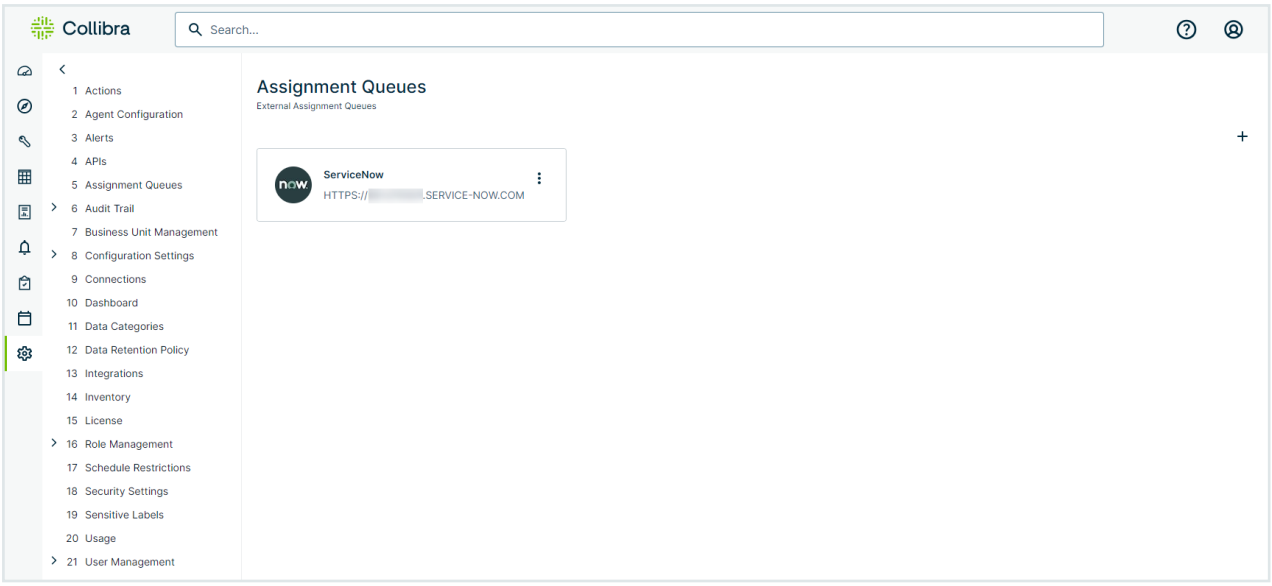
6. Click **Save**.
   » The webhook appears in the table on the Webhook Alerts Configuration page.

What's next?

- Working with the Alert Builder
- Adding dataset alerts

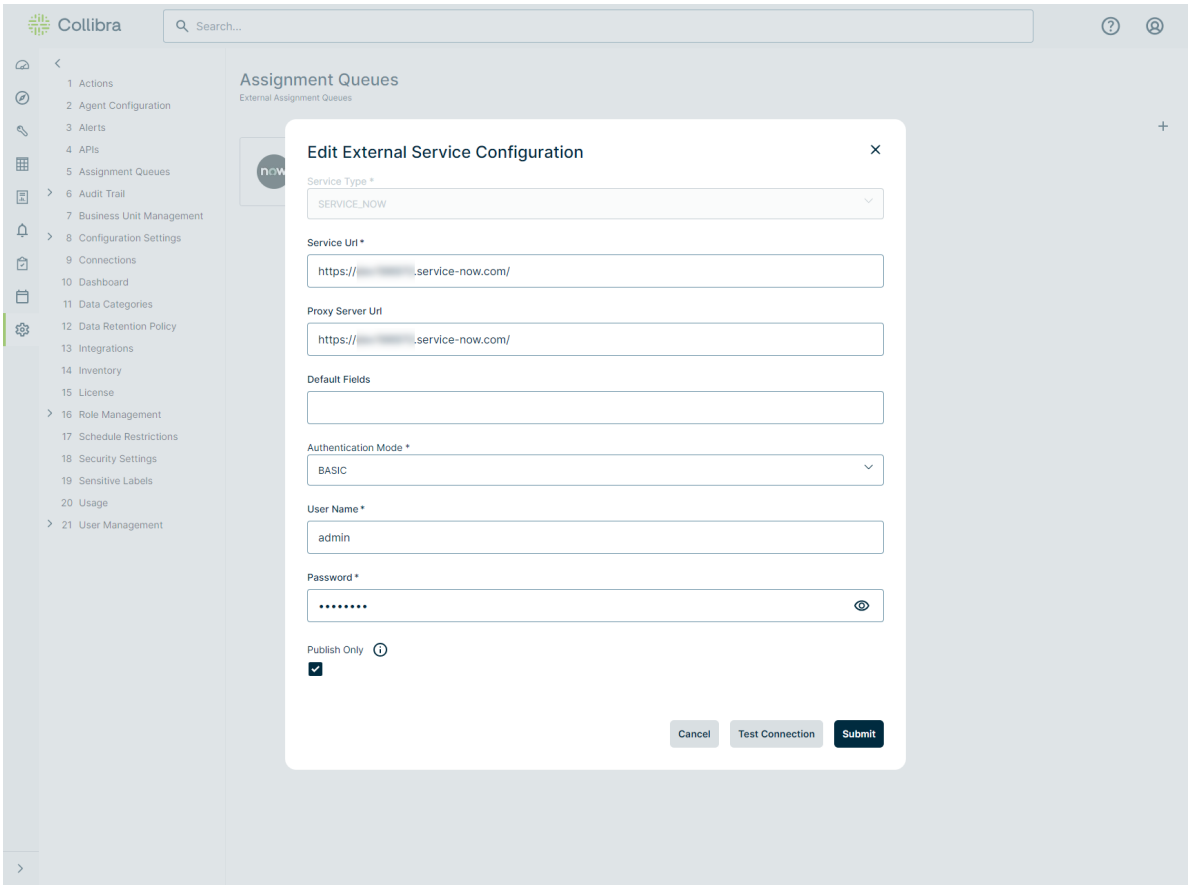
Assignment Queues

Assignment Queues allow admin users to add external services, such as ServiceNow, so users and user groups outside of Collibra can review and manage dataset findings.



Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console** and then select **Assignment Queues** from the list.
3. Click to add a new external service configuration connection, or ⋮ to edit an existing connection.



4. Complete the following fields in the Edit External Service Configuration window.

Field	Description
Service Type	Type of assignment queue service you are configuring. Default is ServiceNow.
Service URL	URL of the service you are configuring.
Proxy Server URL	URL of the proxy server. **Note** This is only applicable based on infrastructure configuration.

Field	Description
Default Fields	Semicolon separated key=value pairs on the Admin page, which override related field values on created assignment service records. **Note** There is a hierarchy whereby Additional Fields take precedence, if there is a matching competing key=value pair between both fields.
Authentication Mode	Type of authentication.
Username	Authentication credentials to the service.
Password	Authentication credentials to the service.
Publish Only	Creates a ServiceNow incident as an unlinked, separate object from the validated Collibra DQ finding. Because Collibra DQ findings and ServiceNow incidents are decoupled when Publish Only is selected, you can edit, validate, and resolve findings in Collibra DQ and incidents in ServiceNow. **Important** If Publish only is *not* selected when a ServiceNow assignment is configured, you must close the ServiceNow incident before resolving the finding in Collibra DQ.

5. Click **Save**.

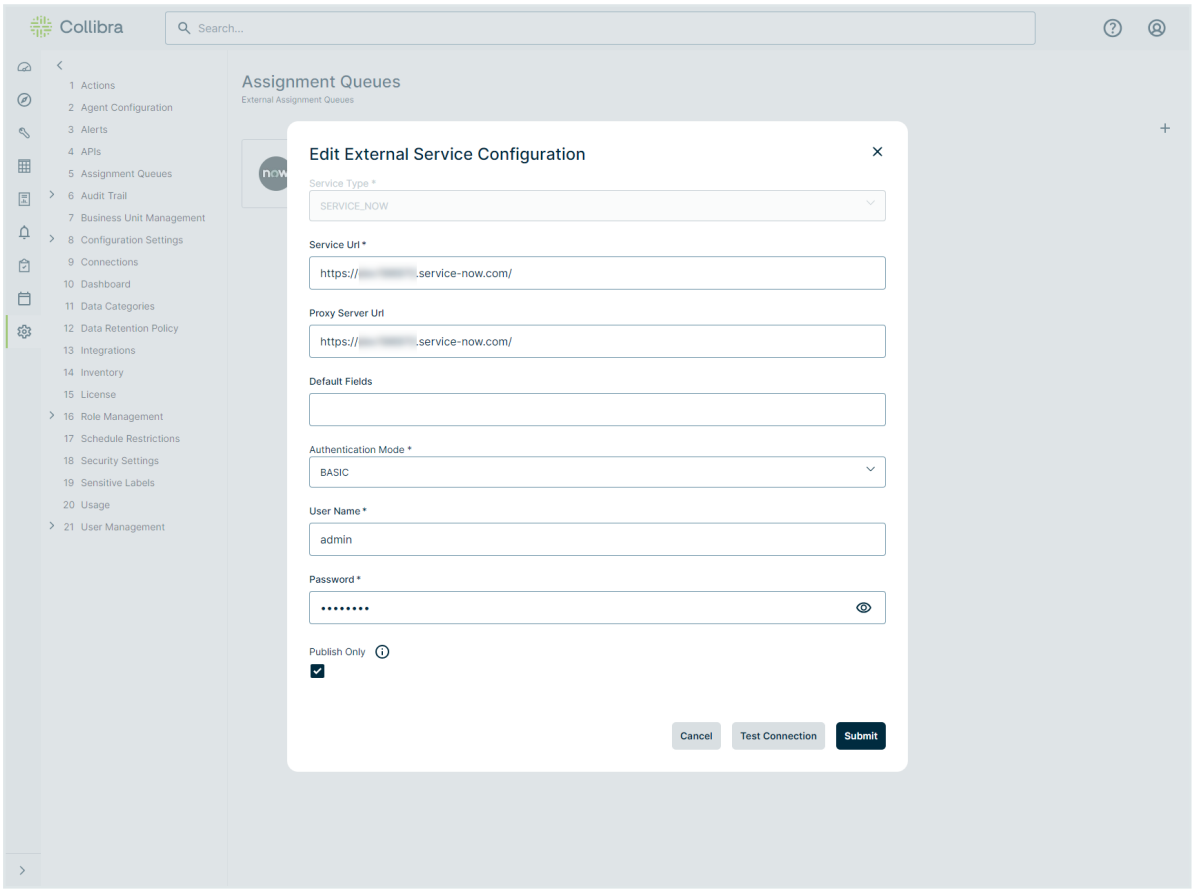
> **Note** You can select **Test Connection**, to ensure that the configuration for your connection is valid. You will receive a message indicating if the connection test was successful.
> You can also click **Close** to close the window without saving.

Configure an External Assignment Connection

Add or edit an external service configuration connection in the Admin Console.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console** and then select **Assignment Queues** from the list.
3. Click to add a new external service configuration connection, or to edit an existing connection.



4. Complete the following fields in the Edit External Service Configuration window.

Field	Description
Service Type	Type of assignment queue service you are configuring. Default is ServiceNow.
Service URL	URL of the service you are configuring.

Field	Description
Proxy Server URL	URL of the proxy server. **Note** This is only applicable based on infrastructure configuration.
Default Fields	Semicolon separated key=value pairs on the Admin page, which override related field values on created assignment service records. **Note** There is a hierarchy whereby Additional Fields take precedence, if there is a matching competing key=value pair between both fields.
Authentication Mode	Type of authentication.
Username	Authentication credentials to the service.
Password	Authentication credentials to the service.
Publish Only	Creates a ServiceNow incident as an unlinked, separate object from the validated Collibra DQ finding. Because Collibra DQ findings and ServiceNow incidents are decoupled when Publish Only is selected, you can edit, validate, and resolve findings in Collibra DQ and incidents in ServiceNow. **Important** If Publish only is *not* selected when a ServiceNow assignment is configured, you must close the ServiceNow incident before resolving the finding in Collibra DQ.

5. Click **Save**.

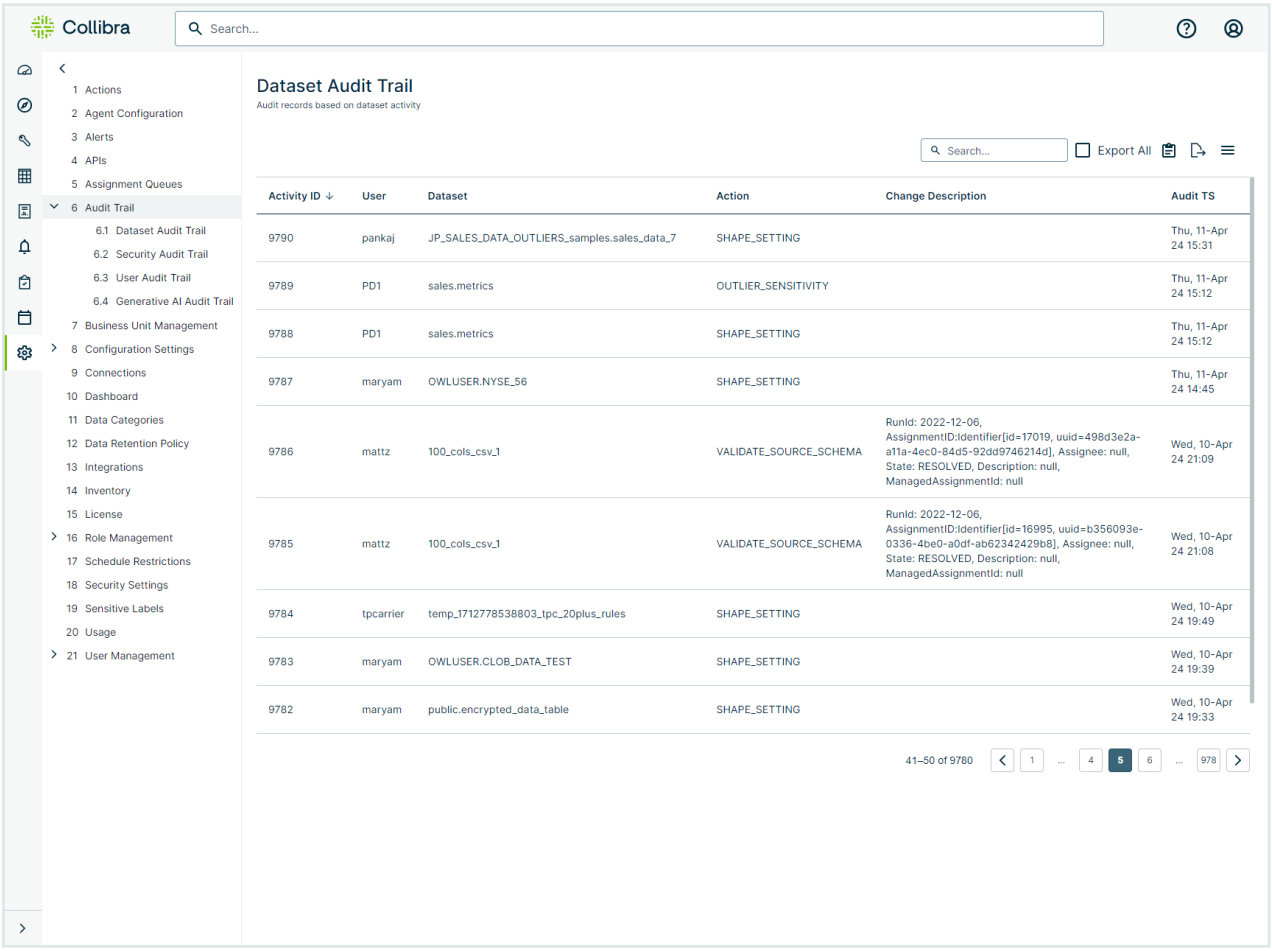
> **Note** You can select **Test Connection**, to ensure that the configuration for your connection is valid. You will receive a message indicating if the connection test was successful.

Audit Trail

Out-of-the-box auditing allows you to track usage and modifications across dataset-, security-, and user-levels.

Dataset Audit Trail

The Dataset Audit Trail displays audit records based on dataset activity. You can view dataset records in your Collibra DQ environment on this page, or copy or export them to view them outside the application.



The following table describes the columns included on the table of the Dataset Audit Trail page.

Column	Description
Activity ID	The internal ID assigned to each event.
User	The username of the user who submitted the request.

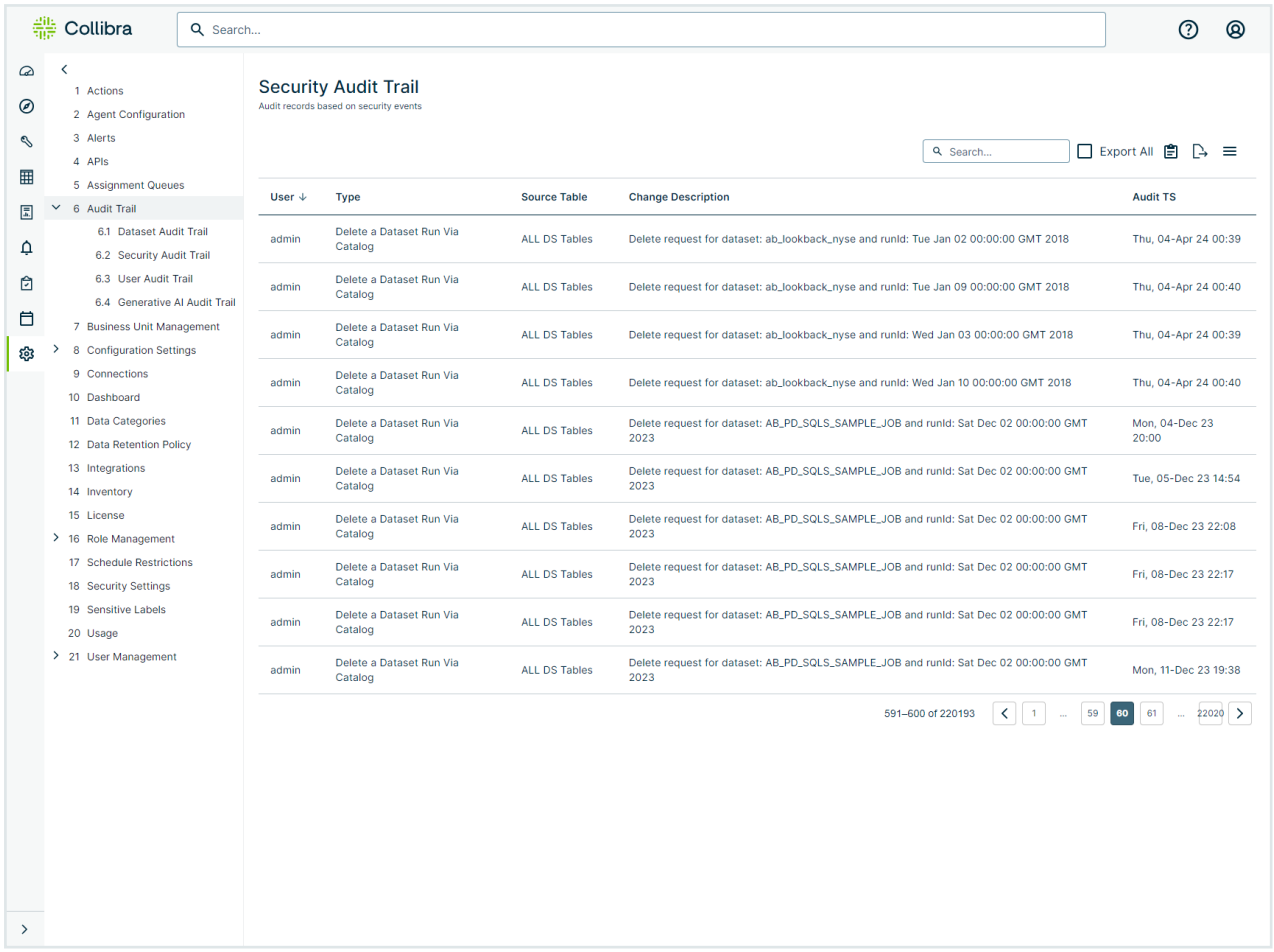
Column	Description
Dataset	The name of the dataset from which the request was submitted.
Action	The type of finding or setting that was changed in the request.
Change Description	The exact change that was requested. For example, an Adaptive Rule on the Behavior tab of the Findings page that was retrained might display a change description similar to the following: "Adaptive:: runId: 2022-01-07, Item: cost, MetricType: CARDINALITY, AdaptiveTier: WIDE, Retrain: true"
Audit TS	The date and time of the submission of the request.

Security Audit Trail

The Security Audit Trail displays audit records based on security events. You can view security records in your Collibra DQ environment on this page, or copy or export them to view them outside the application.

The following events are captured in the table on the Security Audit Trail page:

- User profile updates
- Role updates
- Reference table from metastore
- Dataset deletion requests
- Data source connection updates
- Job schedule attempts



When administrators modify roles mapped to datasets or datasets mapped to roles, changes are documented automatically in the Security Audit Trail. The information in the entry log includes:

- New and original datasets added or removed during the modification.
- New and original roles added or removed during the modification.
- A timestamp of when the modification occurs.
- Type of change.
- Username by which the modifications are made.

The following table describes the columns included on the table of the Security Audit Trail page.

Column	Description
User	The username of the user who submitted the request.
Type	The type of change that was requested. For example, INSERT, UPSERT, or API.

Column	Description
Source Table	The Metastore table where the change was requested.
Change Description	The exact change that was requested. For example, an agent that was added to an Athena connection might display a change description similar to the following: "Added agent connection: ATHENA_CONNECTION To Agent Id: Identifier[id=1, uuid=f123b1f3-a1bc-1c23-456d-7ef89012gh3i]"
Audit TS	The date and time of the submission of the request.

User Audit Trail

The User Audit Trail displays audit records based on user activity. You can view user activity records in your Collibra DQ environment on this page, or copy or export them to view them outside the application.

The table on this page shows details for the following data points:

- Logins: Successful / Failed
- Privileged User Access
- User Actions / Activities
- Account Actions
- Source IP
- Timestamps

The following table describes the columns included on the table of the User Audit Trail page.

Column	Description
Activity ID	The internal ID assigned to each event.
User	The username of the user who submitted the request.
Activity	The application page or action where a logging event occurred.
Source IP	The IP address of the source system.
Session ID	The internal session ID assigned to each user session.
Log String	The user activity that was captured as a logging event.

Column	Description
Audit TS	The date and time of the submission of the request.
Session End TS	The date and time when the user session ended.

Generative AI Audit Trail

The Generative AI Audit Trail provides audit records from Collibra AI requests generated within Collibra DQ.

When a request is sent to the AI engine, all events from the request and response are logged on this page. This allows Collibra DQ administrators to see who used Collibra AI to generate or troubleshoot SQL and when, the input prompt and suggested response query, and how long it took for Collibra AI to generate a suggested query.

You can also export the audit trail records to CSV, Excel, or PDF, print them, or copy them to the clipboard.

The following table describes the columns included on the table of the Generative AI Audit Trail page.

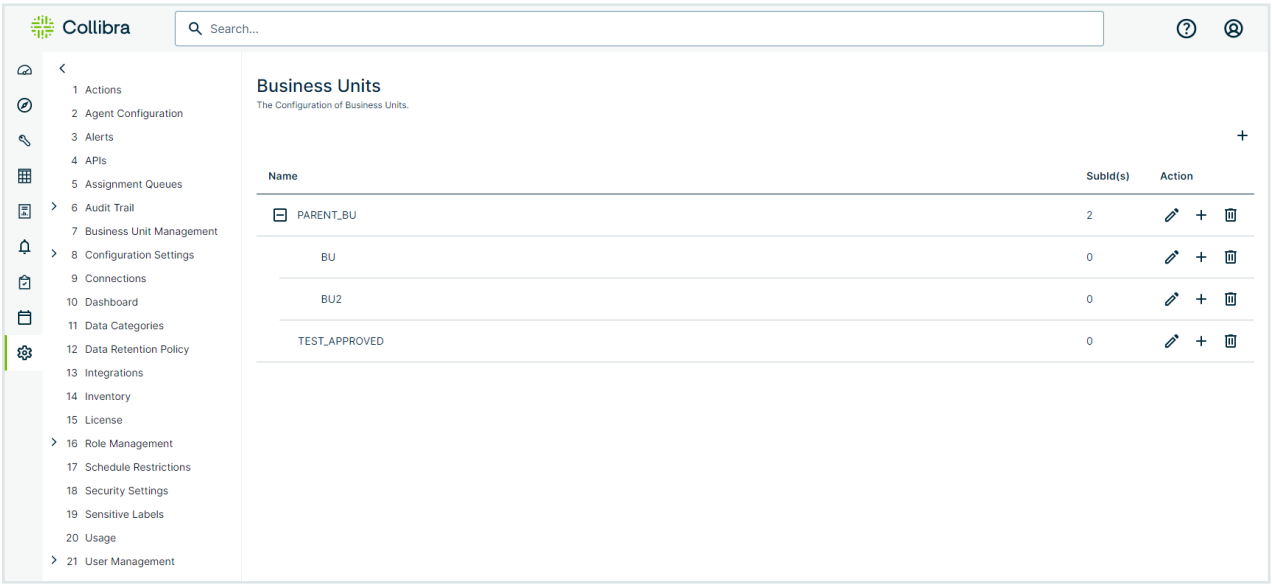
Column	Description
Activity ID	The internal ID assigned to each event.
User	The username of the user who submitted the request.
Dataset	The name of the dataset from which the request was submitted.
Generation Time	The amount of time that elapsed between the time of the request and the response.
Page	The page in Collibra DQ from which the request was sent. /alert signifies that the request originates from the Alert Builder page. /dq/explorer2 signifies that the request originates from the Explorer page. **/dq/finding** signifies that the request originates from the Findings page. **/dq/profile** signifies that the request originates from the Profile page. **/dq/rule** signifies that the request originates from the Rule Workbench.

Column	Description
Prompt	The exact content of the request sent to the Collibra AI engine.
Response	The exact response from the Collibra AI engine.
Error	The contents of the error message when the Collibra AI engine returns an error.
Audit TS	The date and time of the submission of the request.

Business Unit Management

Business Unit Management lets admin users add, edit, and delete business units in Collibra DQ. Creating business units can have several benefits, including:

- Clearer aggregation paths of datasets when integrated with Collibra Platform.
- Better grouping of datasets for assignment tasks.
- Improved filtering on reports and views.



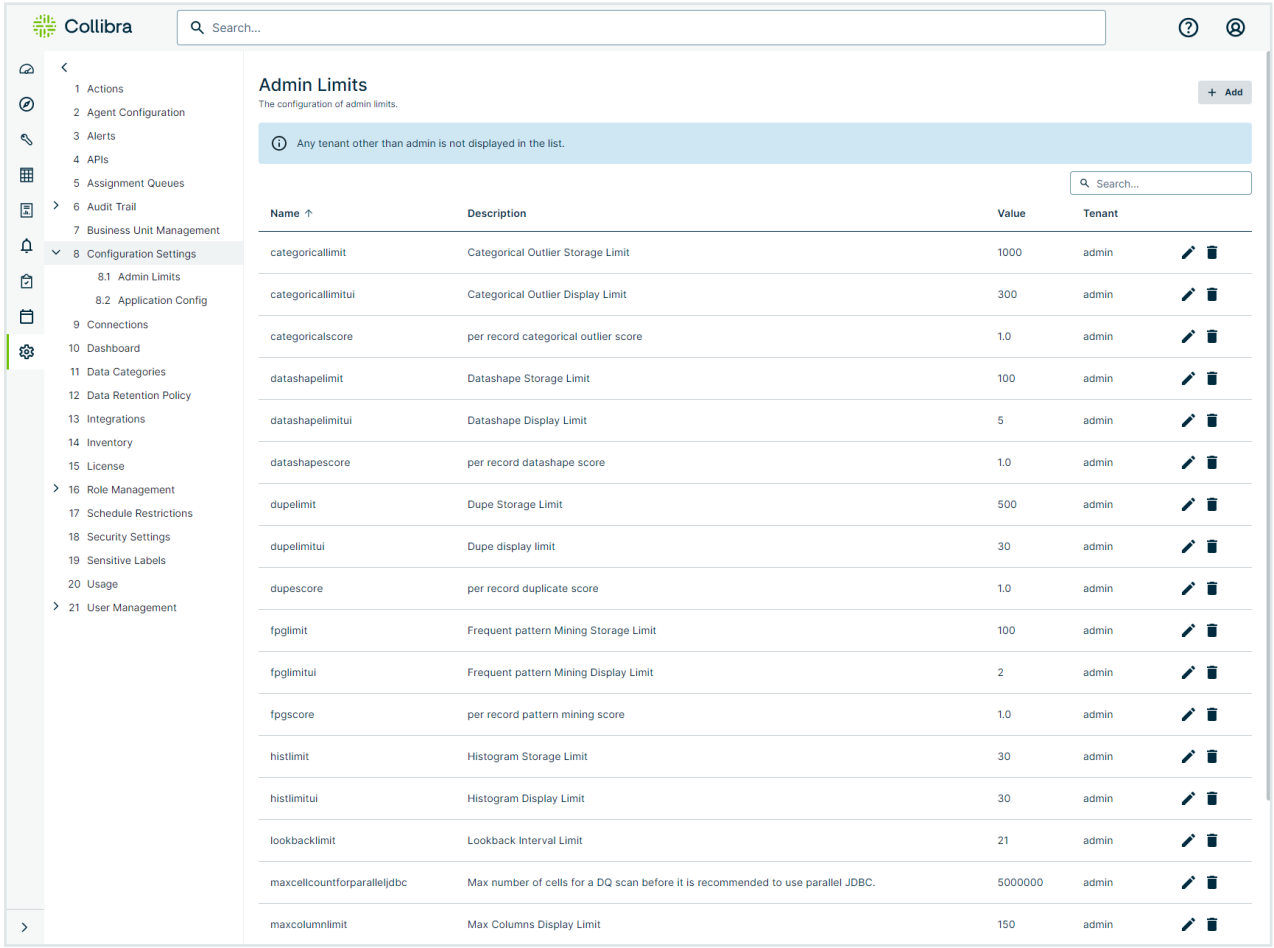
Column	Description
Name	The name of your business unit.
SubId(s)	Indicates the number of children a parent business unit has. Click the ⊞ to see the children under the parent business unit.

Column	Description
Action	• Click the to edit the business unit. • Click the to add a new business unit. • Click the to delete the business unit.

Configuration Settings

Admin Limits

Admin Limits define the maximum values of parameters passed during the execution of a DQ Job.



> **Note** You should generally modify the following Admin Limits settings to prevent too much data from being stored in the database. For example, if you set the `datashapelimit` to `50` and the `datashapelimitui` to `25`, when you run a check, only 50 shape issues get stored in the database and only the first 25 records display in the UI on the Findings page.

The following table shows the available settings on the Admin Limits page.

Setting	Description
categoricallimit	Limit of categorical outlier storage findings that get stored in the database.
categoricallimitui	Limit of categorical outlier findings that display in the UI.
categoricalscore	Default categorical outlier score, per record.
datashapelimit	Limit of datashape storage findings that get stored in the database.
datashapelimitui	Limit of datashape findings that display in the UI.
datashapescore	Default datashape score, per record.
dupelimit	Limit of dupe storage findings that get stored in the database.
dupelimitui	Limit of dupe findings that display in the UI.
dupescore	Default dupe score, per record.
fpglimit	Limit of frequent pattern mining storage findings that get stored in the database.
fpglimitui	Limit of frequent pattern mining findings that display in the UI.
fpgscore	Default frequent pattern mining score, per record.
histlimit	Limit of histogram storage findings that get stored in the database.
histlimitui	Limit of histogram findings that display in the UI.
lookbacklimit	Limit of lookback intervals findings that get stored in the database.

Setting	Description
maxcellcountforparalleljdbc	Maximum number of cells for a DQ scan before it is recommended to use parallel JDBC, which gets stored in the database.
maxcolumnlimit	Limit of maximum columns display findings that get stored in the database.
maxconcurrentjobs	Maximum number of Pushdown jobs that can run concurrently.
maxexecutorcores	Maximum number of executors cores to be set during estimation on a DQ scan, which gets stored in the database. Set the default value at the agent level.
maxexecutormemory	Maximum number of executor memory (GB) to be set during estimation on a DQ scan, which gets stored in the database. Set the default value at the agent level.
maxnumexecutors	Maximum number of executors to be set during estimation on a DQ scan, which gets stored in the database. Set the default value at the agent level.
maxpartitions	Maximum number of partitions to be set during estimation of a DQ scan.
maxworkermemory	Agent Worker Maximum Memory (WMM)
maxworkercores	Maximum cores per worker.
minworkermemory	Agent Worker Minimum Memory (WMI)
obslimit	Limit of observation storage findings that get stored in the database.
obslimitui	Limit of observation storage findings that display in the UI.
obsscore	Default observation score, per record.
outlierlimit	Limit of outlier findings that get stored in the database.
outlierlimitui	Limit of outlier findings that get stored in the database.
outlierscore	Default outlier score, per record.
partitionautocal	Calculate the number of Spark partitions automatically for optimal performance.

Setting	Description
passfaillimit	Limit of passing or failing threshold findings that get stored in the database.
pergbpartitions	Per 1 GB memory partitions
previewlimit	Set the maximum number of preview results shown in the Dataset Overview and Rule Workbench.
schemascore	Default schema score, per record.
scorecardsize	Limit of scorecard size display findings that get stored in the database.
sourcedupelimit	Source dupe limit for Pushdown jobs (0 = unlimited)
sourceoutlierlimit	Source outlier limit for Pushdown jobs (0 = unlimited)
sourceoutputenabled	Source output for Pushdown jobs
sourcerulelimit	Source rule limit for Pushdown jobs (0 = unlimited).
sourcesampleenabled	Source sample for Pushdown jobs.
sourceshapelimit	Source shape limit for Pushdown jobs (0 = unlimited).
totalworkers	Total number of workers available.
valsrcdisableaq	Disable the Assignment actions (Validate, Invalidate, Resolved) on the Validate Source data model on the Findings page.
valsrclimit	Limit of validate source storage findings that get stored in the database.
valsrclimitui	Limit of validate source findings that display in the UI.
valsrcscore	Default validate source score, per record.

> **Note** Settings with "score" in their name represent the default score for that setting's type of findings.

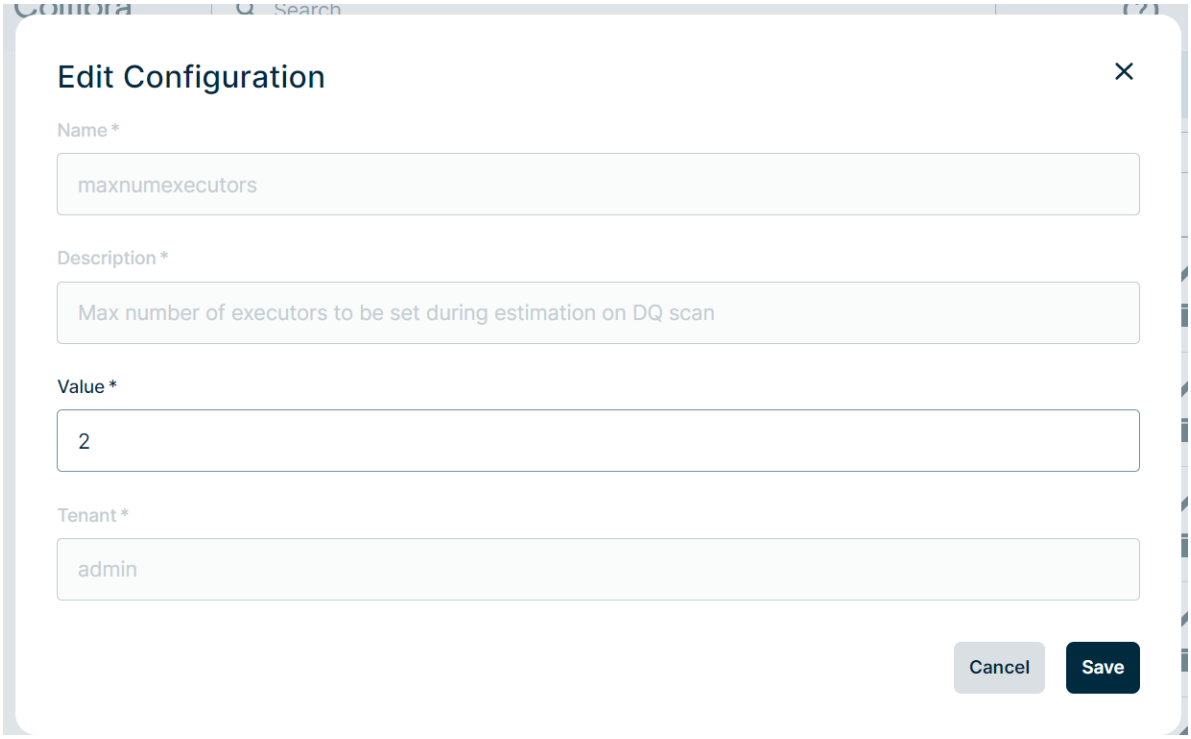
Configuring admin limit settings

This section shows you how to edit the default values of admin limit settings or delete the setting entirely.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Configuration Settings** and then click **Admin Limits**.
   » The Application Configuration Settings page opens.
4. To modify application settings, click the , next to the setting for which you want to change the value.
5. Click the icon next to the setting whose default value you want to change.

> **Important** If you click the icon, the application setting deletes immediately and no warning prompt displays.



> **Warning** When you increase the UI display limits, the performance of the web application may decrease.

6. Click **Save**.

Adding custom admin limits

This page shows you how to create custom admin limit settings for the system to enforce across Data Quality & Observability Classic.

Steps

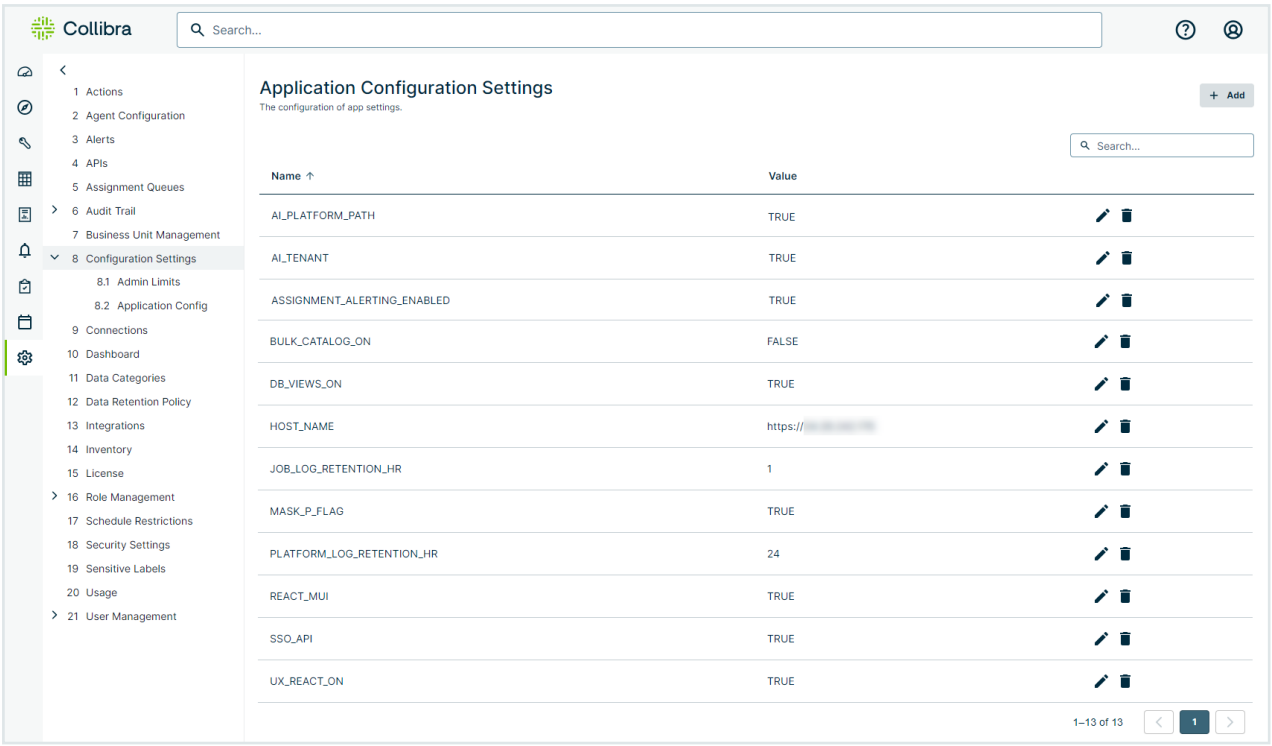
1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Configuration Settings** and then click **Admin Limits**.
   » The Admin Limits page opens.
4. Click **Add** near the top right corner of the page and fill out the required fields, as shown in the following table.

Input Field	Description
Name	Enter a name for the limit.
Description	Enter a meaningful description of the limit.
Value	Set a default value for the limit.
Tenant	Enter the name of the tenant to which the limit applies.

5. Click **Save**.
   » Your custom admin limit setting displays in the table on the Admin Limits page.

Application Configuration Settings

Application Configuration Settings allows admin users to toggle various platform controls on and off. Admins can also add custom application configuration settings from this page.



The following table shows the out-of-the-box application configuration settings.

Setting	Description	Value
AI_PLATFORM_PATH	When set to `TRUE`, SQL Assistant for Data Quality uses the Collibra Platform path and utilizes the Gemini AI model. This is the default and recommended setting. When set to `FALSE`, SQL Assistant for Data Quality uses the preview path with an encryption key provided by Collibra and utilizes the text-bison-002 AI model.	`TRUE/FALSE` Default: `TRUE`
AI_TENANT	Enables SQL Assistant for Data Quality automates SQL rule writing and troubleshooting to help you accelerate the discovery, curation, and visualization of your data. when set to `TRUE`.	`TRUE/FALSE` Default: `FALSE`
BULK_CATALOG_ON	When set to `TRUE`, allows/disallows using the catalog feature from the Wizard in the UI.	`TRUE/FALSE` Default: `TRUE`

Setting	Description	Value
HOST_NAME	Hostname of your URL. **Note** When you enable **Include Links** on the Alerts Configuration page of the Admin Console, you must also enter your host address in the HOST_NAME input field. For more information, go to Configuring alert emails.	Enter the host-name in the input field.
JOB_LOG_ RETENTION_HR	The number of hours that Collibra DQ retains DQ Job logs.	Enter the number of hours for Col-libra DQ to retain DQ Job logs.
MANDATORY_ PRIMARY_RULE_ COLUMN	When set to `TRUE`, users are required to select a primary column when creating a rule. This requirement is enforced when a user creates a new rule or saves an existing rule for the first time after the setting is enabled. Existing rules are not affected automatically.	
MASK_P_FLAG	When set to `TRUE`, sets/unsets `-p/-srcp` masking in the UI.	`TRUE/FALSE` Default: `TRUE`
PLATFORM_ LOG_ RETENTION_HR	The number of hours platform logging is retained.	Enter the number of hours for Collibra DQ to retain platform logs.
RULE_FILTER	When set to `TRUE`, you can use filter queries when creating a rule on the Rule Workbench.	
REACT_MUI	When set to `TRUE`, enables React MUI. **Note** This option is not applicable in Data Quality & Observability Classic versions 2024.07 and newer.	`TRUE/FALSE` Default: `TRUE`

Setting	Description	Value
UX_REACT_ON	When set to `TRUE`, turns React MUI on. When `FALSE`, the Collibra DQ web application displays the legacy UI. **Note** This option is not applicable in Data Quality & Observability Classic versions 2024.07 and newer.	`TRUE/FALSE` Default: `FALSE`

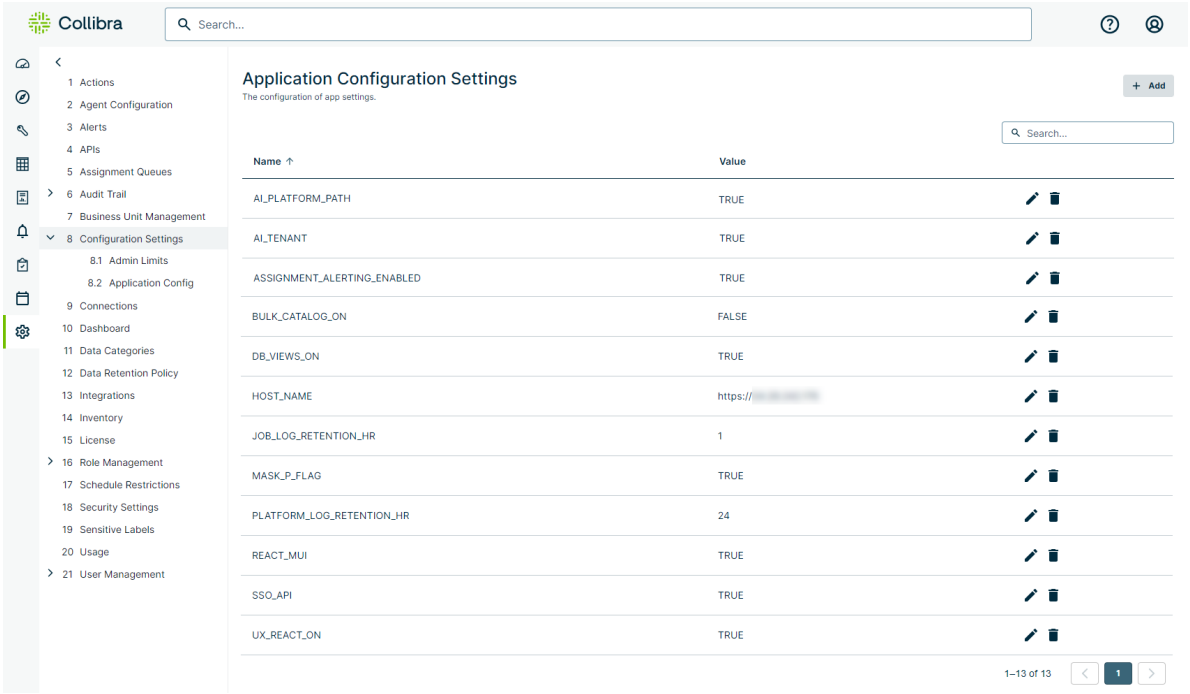
Configuring application settings

This page shows admins how to edit the values of application settings or delete them entirely.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Configuration Settings** and then click **Application Config**.
   » The Application Configuration Settings page opens.
4. Click the next to the setting whose value you want to change.

   Important If you click , the application setting deletes immediately without displaying a confirmation prompt.



5. The following table shows the Data Quality & Observability Classic application settings.

App Setting	Description	Value	Notes
AI_PLATFORM_PATH	When set to `TRUE`, SQL Assistant for Data Quality uses the Collibra Platform path and utilizes the Gemini AI model. This is the default and recommended setting. When set to `FALSE`, SQL Assistant for Data Quality uses the preview path with an encryption key provided by Collibra and utilizes the text-bison-002 AI model.	`TRUE/FALSE` Default: `TRUE`	

App Setting	Description	Value	Notes
AI_TENANT	Enables SQL Assistant for Data Quality automates SQL rule writing and troubleshooting to help you accelerate the discovery, curation, and visualization of your data. when set to `TRUE`.	`TRUE/FALSE` Default: `FALSE`	
BULK_CATALOG_ON	When set to `TRUE`, allows/disallows using the catalog feature from the Wizard in the UI.	`TRUE/FALSE` Default: `TRUE`	
DB_VIEWS_ON	When set to `TRUE`, Explorer includes views from databases as data to which you can connect and query.	`TRUE/FALSE` Default: `FALSE`	
HOST_NAME	Hostname of your URL.	Enter the hostname in the input field.	When you enable **Include Links** on the Alerts Configuration page of the Admin Console, you must also enter your host address in the HOST_NAME input field. For more information, go to Configuring alert emails.
JOB_LOG_RETENTION_HR	The number of hours that Collibra DQ retains DQ Job logs.	Enter the number of hours for Collibra DQ to retain DQ Job logs.	
MASK_P_FLAG	When set to `TRUE`, sets/unsets `-p/-srcp` masking in the UI.	`TRUE/FALSE` Default: `TRUE`	

App Setting	Description	Value	Notes
PLATFORM_LOG_RETENTION_HR	The number of hours platform logging is retained.	Enter the number of hours for Collibra DQ to retain platform logs.	
PUSHDOWN_FOR_DATABRICKS	When set to `TRUE`, you can run Databricks Pushdown Jobs.	`TRUE/FALSE` Default: `TRUE`	You must have a valid Databricks connection with the Pushdown option, selected in the Connection template.
REACT_MUI	When set to `TRUE`, this setting enables the latest UI for user-level (non-admin) pages in the Collibra DQ web application.	`TRUE/FALSE` Default: `TRUE`	
UX_REACT_ON	When set to `TRUE`, this enables the latest UI for admin-level pages in the Collibra DQ web application.	`TRUE/FALSE` Default: `FALSE`	

6. Click **Save**.

Adding custom application settings

This page shows you how to create custom application settings.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Configuration Settings** and then click **Application Config**.
   » The Application Configuration Settings page opens.
4. Click **Add** near the top right corner of the page and fill out the required fields, as shown in the following table.

Input Field	Description
Name	Create a name for the custom configuration.
Value	Enter a value for the custom configuration.

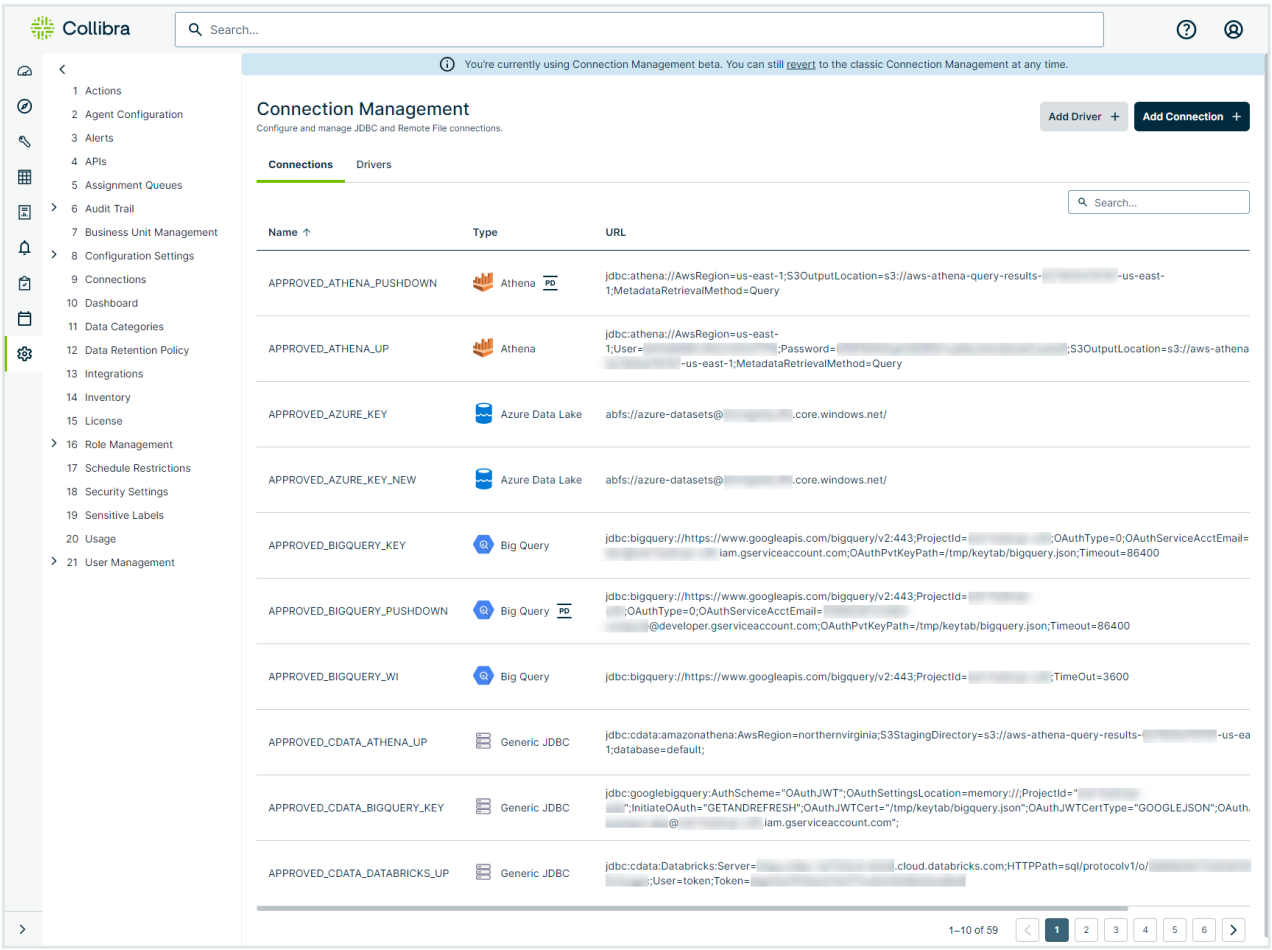
5. Click **Save**.
   » Your custom application setting displays in the table on the Application Configuration Settings page.

Connections

Connection Management lets administrators add, edit, and delete drivers and connections to data sources outside of Collibra DQ. In order for Collibra DQto run data quality checks, you need to first connect to source data. Collibra DQ is able to establish connections to many common data sources, including:

- JDBC
- Remote files, such as Amazon S3 and Google Cloud Storage
- Local network files (NFS)

> **Tip** For more information about supported data sources and how to connect to them, go to Connections.



Column	Description
Name	The name of your connection.
Type	The name of the JDBC or file-based data source.
URL	The connection URL or file path where the source data of your connection resides.
Action	Edit or delete connections, or limit the schemas of JDBC connections that are visible to users in Explorer.

> **Note** The Limit Schemas option is only available when it is enabled from the connection template. See Limiting schemas for more information.

Limiting schemas

Limiting schemas allows you to manage usage and restrict visibility to only the necessary schemas of JDBC connections in the Explorer tree. This helps to reduce the clutter from excess schemas, which allows you to access the data you need more quickly.

> **Note** When you include a restricted schema in the query of a DQ Job, the query scope may be overwritten when the job runs. While only the schemas you selected when you set up the connection are shown in the Explorer menu, users are not restricted from running SQL queries on any schema from the data source.

Prerequisites

You have:

- ROLE_ADMIN or ROLE_CONNECTION_MANAGER

Limiting schemas on new connections

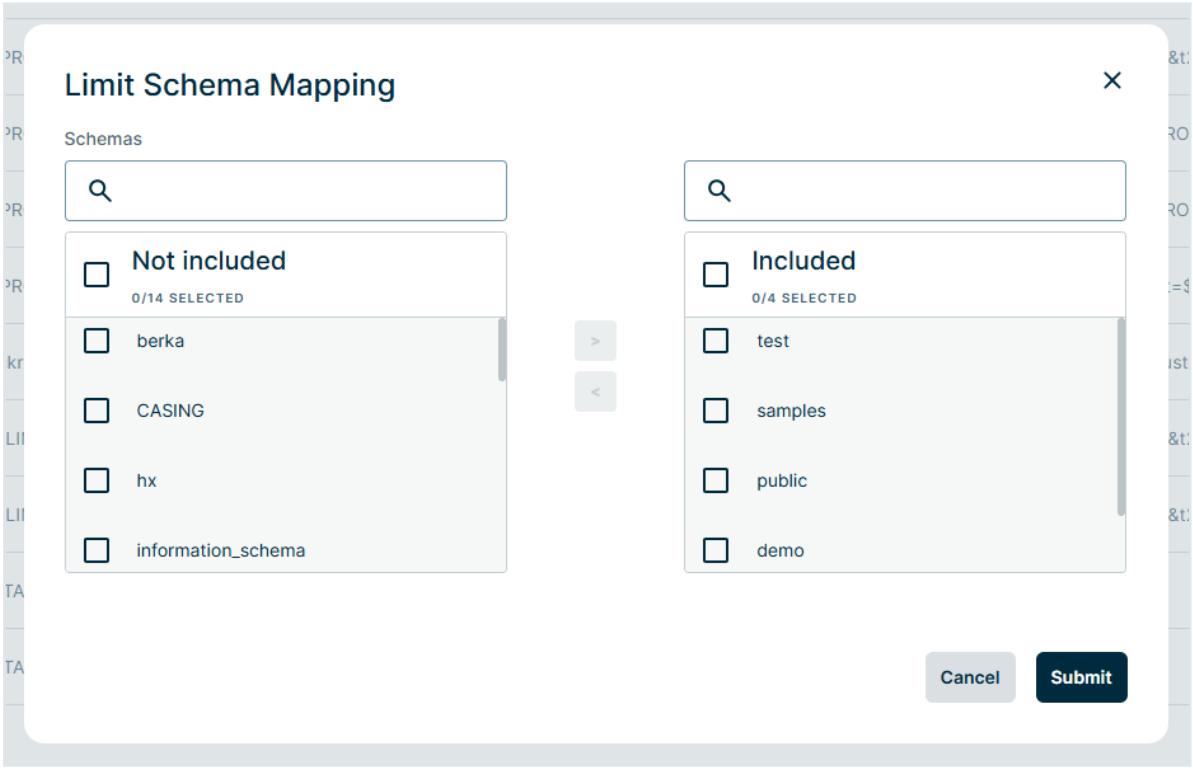
> **Note**
>
> - Limiting schemas on connections limits the available schemas in Explorer and the Connections step when setting up an integration with Collibra Platform.
> - If you plan to use the limit schemas option, avoid replacing or modifying schemas in your connection after running a job. Doing so may result in an expected scenario where the selectable columns in the Rule Workbench reference the new schema instead of the schemas of the original job.

1. On the main menu, click ⚙, then click **Connections**.
2. Click **Connections**.
   » The **Connection Management** page opens.
3. Click **Add Connection** on the upper right side of the page.
   » The **Add New Connection** dialog opens to the JDBC Connections tab with a list of supported JDBC data sources.
4. Click a data source.
   » The **Add Connection** template appears.
5. Enter the required information, and select the **Limit Schemas** option on the Details tab.

> **Important** The required information in the connection may vary depending on your connection type. However, you must enable the **Limit Schemas** option to limit schemas from your connection.

6. Click **Submit**.
7. From the Connections page, on the far right side of the row containing your newly created connection, click **Actions**, then click **Limit Schemas**.
   » The **Limit Schema Mapping** dialog appears.
8. Limit schemas by moving them from the **Included** box to the **Not included** box. To do this, select the schema(s) you wish to limit, then click ›. You can also use the search fields to filter schemas.

> **Note** You can remove included schemas by selecting them from the **Included** box, then clicking ‹ to move them to the **Not included** box.



9. Click **Submit**.
   » The schemas you added to the "Not included" box in the Limit Schema Mapping dialog are no longer visible in the Explorer tree.

Limiting schemas on existing connections

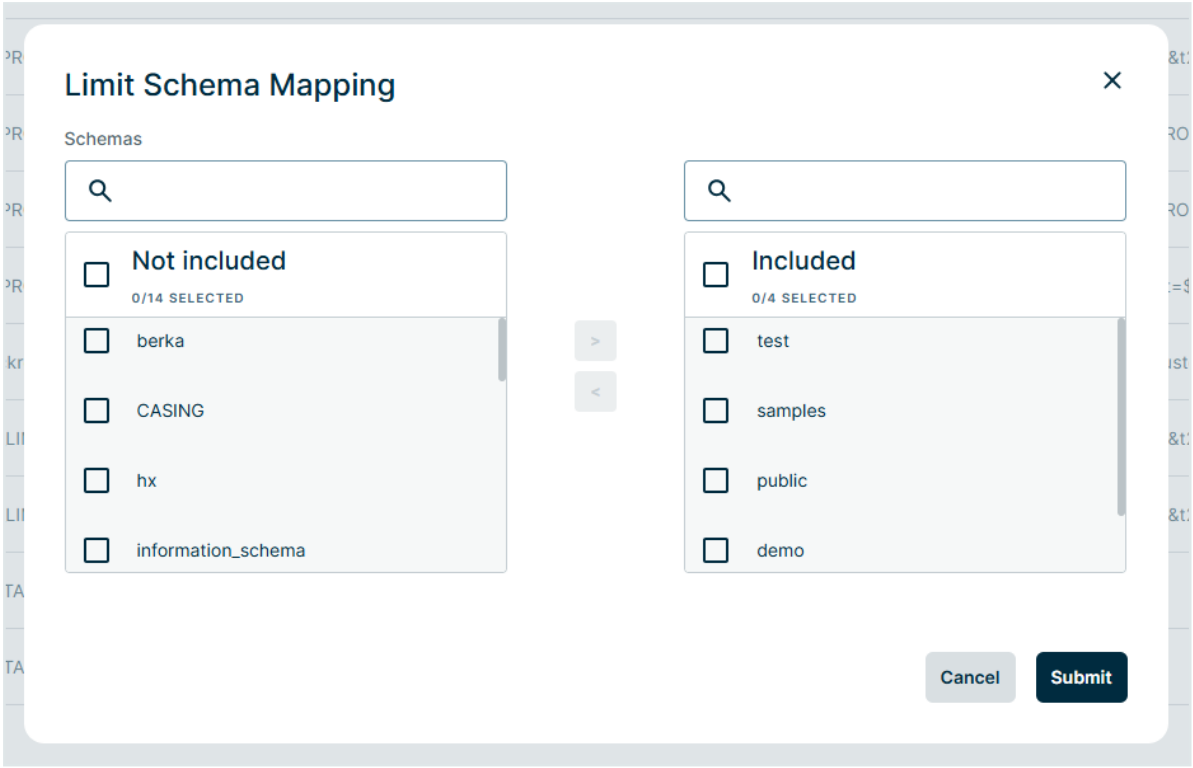
> **Note**
>
> - Limiting schemas on connections limits the available schemas in Explorer and the Connections step when setting up an integration with Collibra Platform.
> - If you plan to use the limit schemas option, avoid replacing or modifying schemas in your connection after running a job. Doing so may result in an expected scenario where the selectable columns in the Rule Workbench reference the new schema instead of the schemas of the original job.

1. On the main menu, click ⚙, then click **Connections**.
2. Click **Connections**.
   » The **Connection Management** page opens.
3. On the far right side of the row containing an existing JDBC connection, click **Actions**, then click **Edit**.
   » The **Edit Connection** template appears.
4. Select the **Limit Schemas** option.

   > **Important** You must enable the **Limit Schemas** option on the Details tab to limit schemas from your connection.

5. Click **Submit**.
6. On the far right side of the row containing an existing JDBC connection, click **Actions**, then click **Limit Schemas**.
   » The **Limit Schema Mapping** dialog appears.
7. Limit schemas by moving them from the **Included** box to the **Not included** box. To do this, select the schema(s) you wish to limit, then click ›. You can also use the search fields to filter schemas.

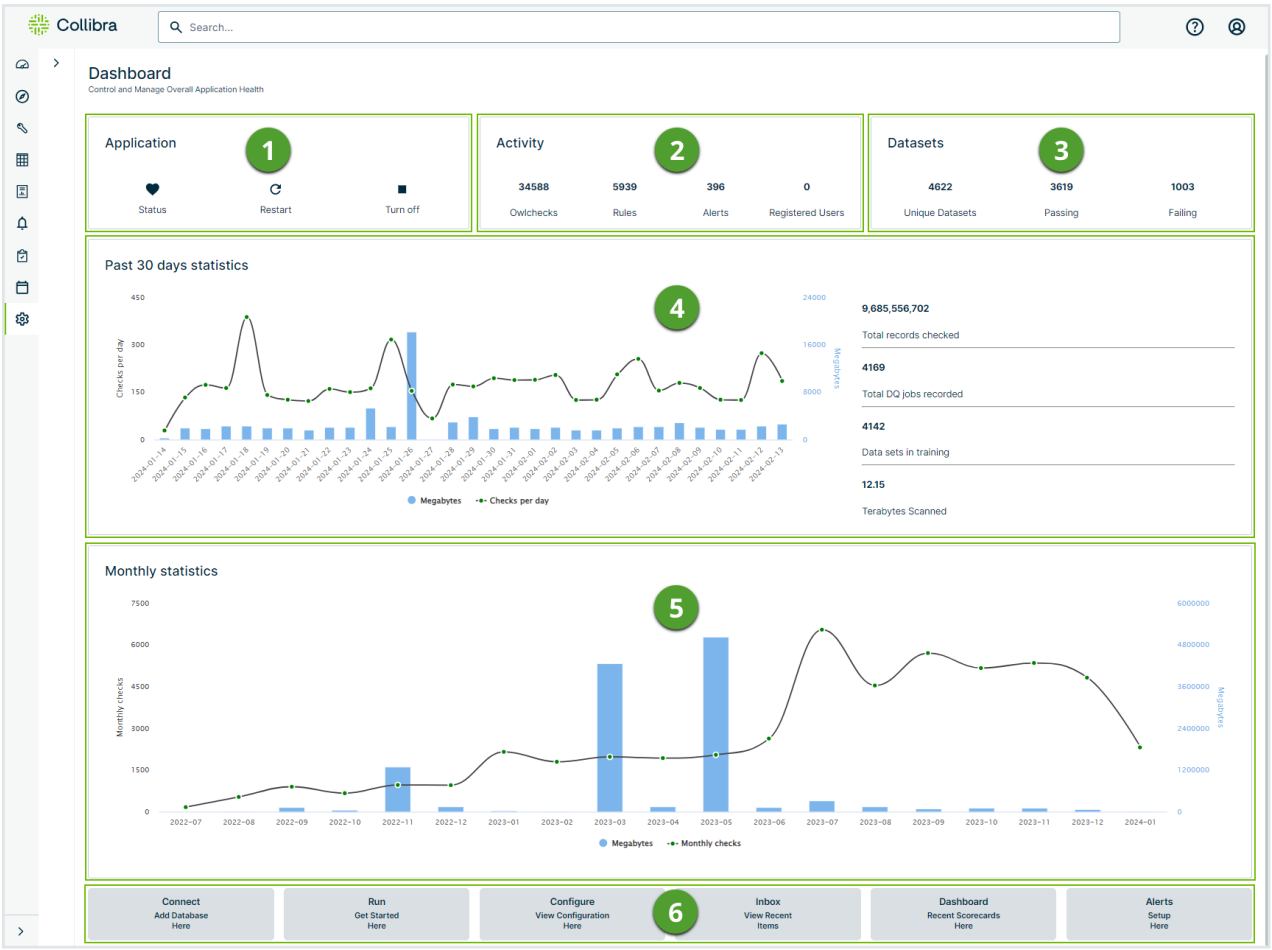
   > **Note** You can remove included schemas by selecting them from the **Included** box, then clicking ‹ to move them to the **Not included** box.



8. Click **Submit**.
   » The schemas you added to the "Not included" box in the Limit Schema Mapping dialog are no longer visible in the Explorer tree.

Dashboard

Dashboard, Inventory, Connections, and Configuration are shortcut links to different operations within Data Quality & Observability Classic.



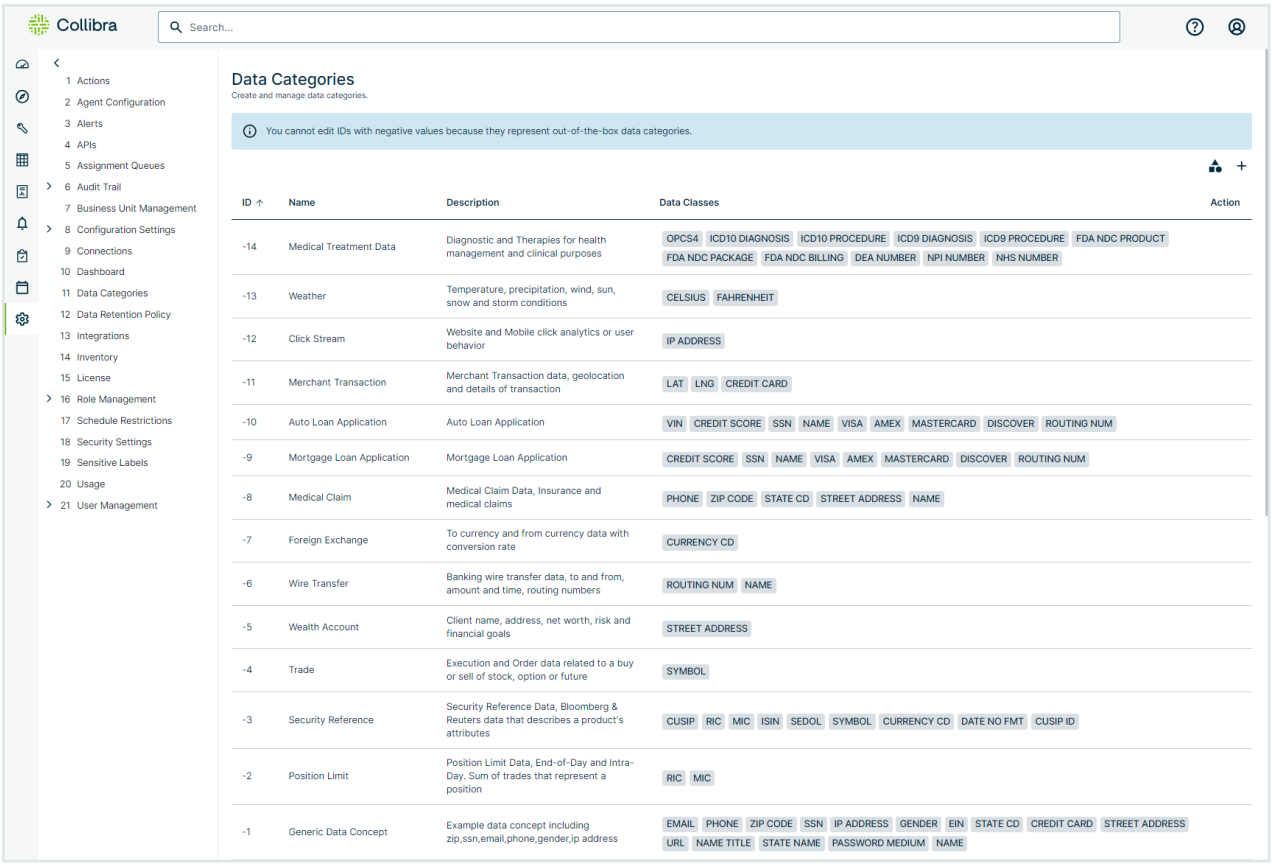
No.	Component	Details
1	Application controls	These buttons allow you to: • Check the status of your Data Quality & Observability Classic instance. • Restart your instance of Data Quality & Observability Classic. • Turn off your instance of Data Quality & Observability Classic.
2	User activity	Displays the number of DQ Jobs, rules, alerts, and registered users in Collibra DQ.
3	Dataset details	Shows the number of unique datasets, passing and failing datasets in Collibra DQ.

No.	Component	Details
4	Usage statistics over the past 30 days	Measures the number of DQ Jobs that ran per day and the number of megabytes consumed by those runs over the most recent 30 day period. Additionally, the data points to the right of the chart include: • Total number of records checked over 30 days. • Total DQ Jobs that ran over 30 days. • Total datasets in training over 30 days. • Total amount of terabytes scanned over 30 days. **Tip** You can hover your cursor over any of the days on the chart to view the precise details for that day.
5	Monthly usage statistics	Measures the number of DQ Jobs that ran per month and the number of megabytes consumed by those runs over the most recent 21 month period. **Tip** You can hover your cursor over any of the months on the chart to view the precise details for that month.
6	Shortcuts	Quickly navigate to the following pages in the application: • Connections • Explorer • Admin Limits • List View • Scorecards • Alerts

Data Categories

Data Categories displays out-of-the-box and custom data categories, which are logical groupings of data classes to meet your business needs. You can add new custom data categories and manage existing ones from this page.

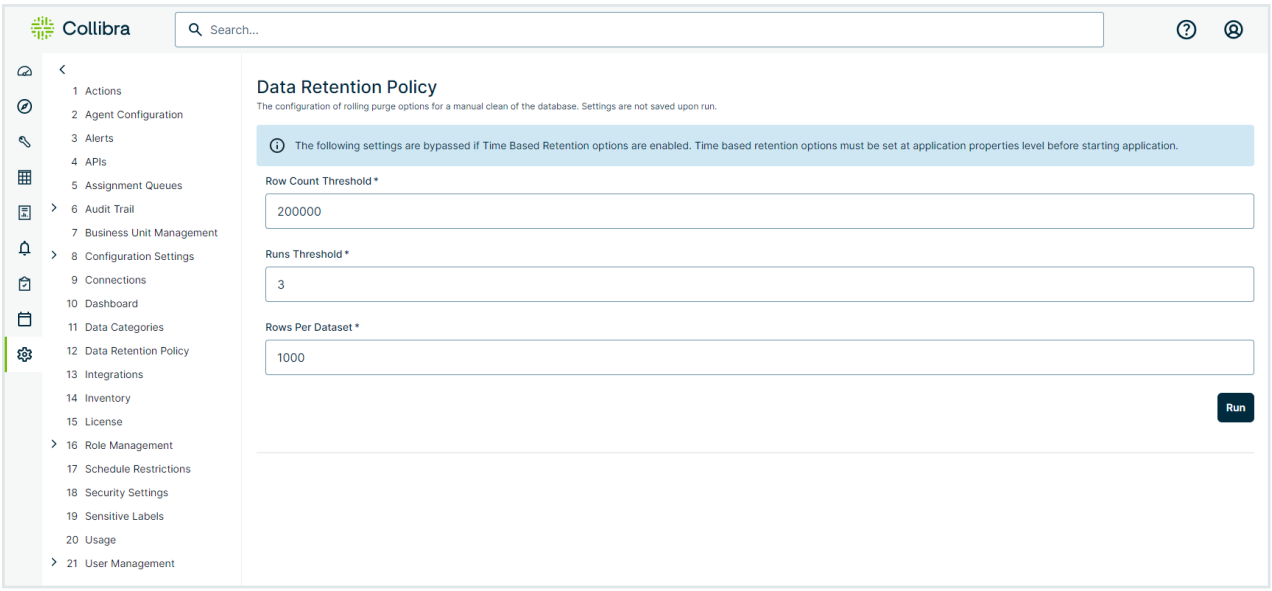
> **Tip** For more on working with data classes and categories, see Data Classes.



Column	Description
ID	The ID of a corresponding data category. Negative IDs, such as -14, cannot be edited because they represent out-of-the-box data categories. To display your custom data categories first, click the ID column header to sort the IDs in reverse order.
Name	The name of a data category.
Description	A meaningful description of a corresponding data category.
Data Classes	The data classes grouped under the data category.
Action	Edit or delete custom data categories.

Data Retention Policy

Data Retention Policy allows admins to purge data from Data Quality & Observability Classic based on time- or size-based policies in both single- and multi-tenant environments.



Option	Description
Row Count Threshold	The minimum number of rows in the data_preview table for the system to consider purging data from it.
Runs Threshold	The minimum number of DQ Job runs of a given dataset that exceeds the Rows Per Dataset and Row Count Threshold minimum requirements for the system to consider purging data from it.
Rows Per Dataset	The minimum number of rows of a dataset that exceeds the Runs Threshold and Row Count Threshold minimum requirements for the system to consider purging data from it.

How does it work?

1. Data Quality & Observability Classic checks if data needs to be purged by comparing the row count threshold to the total number of rows in the data_preview table of the Metastore.
2. It generates a list of purge candidates for each dataset based on the number of runs by run ID and total rows in the table.
3. It removes rows for the oldest run IDs in each table by looping through the candidate

list.
4. A new candidate list is created, and step 3 is repeated.

Option	Description
Retention by Fields	Options include: **By Run Date (run_id)** Considers the runId field of the dataset. **By Update Timestamp (updt_ts)** Considers the last time a dataset run was updated.
Retention Days	The number of days data is retained before being automatically purged.

How does it work?

1. Data Quality & Observability Classic checks the dataset_scan table of the Metastore to determine if data needs to be purged. It identifies days outside the threshold based on your selection of either **By Run Date (run_id)** or **By Update Timestamp (updt_ts)**.
2. It generates a list of purge candidates for rows in the dataset_scan table that exceed the threshold of the selected field.
3. The dataset name and run ID of all rows in the candidate list are joined with the dataset name and run ID in the specific table to identify the rows to be purged from that table.

> **Note** When Data Quality & Observability Classic attempts to create a purge candidate list, it may not identify any rows for removal in the specific tables.

What is purged?

When data is purged from Data Quality & Observability Classic, data from each dataset included in the purge is cleared on a rolling basis. When a dataset is cleaned as part of this process, the following tables are purged from the Metastore.

Table	Table description	Data
Shared by many activities		
data_preview	The drill-in records for rules, outliers, shapes, and so on.	**dataset** and **runId**
observation	The type of finding discovered during a DQ Job run.	**dataset** and **runId**
Profile		
dataset_scan	The findings scores and pass and fail information for a given DQ Job run.	**dataset** and **runId**
dataset_field	The profiling stats relative to any given column for a database table scan.	**dataset** and **runId**
dataset_field_value	The top and bottom N values of a dataset, including the unique count.	**dataset** and **runId**
datashape	The data shape format, associated linkID, and assignments.	**dataset** and **runId**
Dataset Histogram		
dataset_hist	The historical job run values, including the averages, medians, and quartile information for a given dataset.	**dataset** and **runId**
Dataset Correlation		
dataset_corr	The correlation data between columns.	**dataset** and **runId**
Behavior		
behavior	The profile data observed during a DQ Job run, including min, max, and stats relative to a given column.	**dataset** and **runId**
Rules		
rule_breaks	The rule breaks and link IDs for a given dataset and DQ Job run.	**dataset** and **runId**

Table	Table description	Data
rule_output	The results of a rule observed during a DQ Job run.	**dataset** and **runId**
Outliers		
outlier	The outlier column, type, value, confidence score, associated link ID, and assignments.	**dataset**, **runId**, and **true**
Validate Source		
validate_ source	The source dataset and its associated observation types and counts related to validate source correlation.	**dataset** and **runId**
dataset_ schema_ source	The schema differences between source and target datasets observed during a DQ Job run.	**dataset** and **runId**
Other		
alert_output	The contents of the alert email that send when the required conditions are met to trigger an alert during a DQ Job run.	**dataset** and **runId**
dataset_ activity	A rollup of all the aggregate stats for a given DQ Job run, including the time it takes to run per activity.	**dataset** and **runId**
hint	The DQ Job activity stages and the stage details that populate on the Job log. For example, LOAD - 15520 rows loaded to Historical dataframe.	**dataset** and **runId**
dq_inbox	All processed findings with rank values to help calculate the final impact to the DQ Job score.	**dataset** and **runId**

Additionally, the following dataset run data is included in the purge:

Element	Description	Data
Dataset	Any dataset containing data flagged as sensitive.	**String dataset**, **Date runId**, **dataset**, **runId**, and **"admin"**
PII	Personally identifiable information observed in datasets during DQ Job runs.	**String dataset**, **String colName**, **dataset**, and **colName**

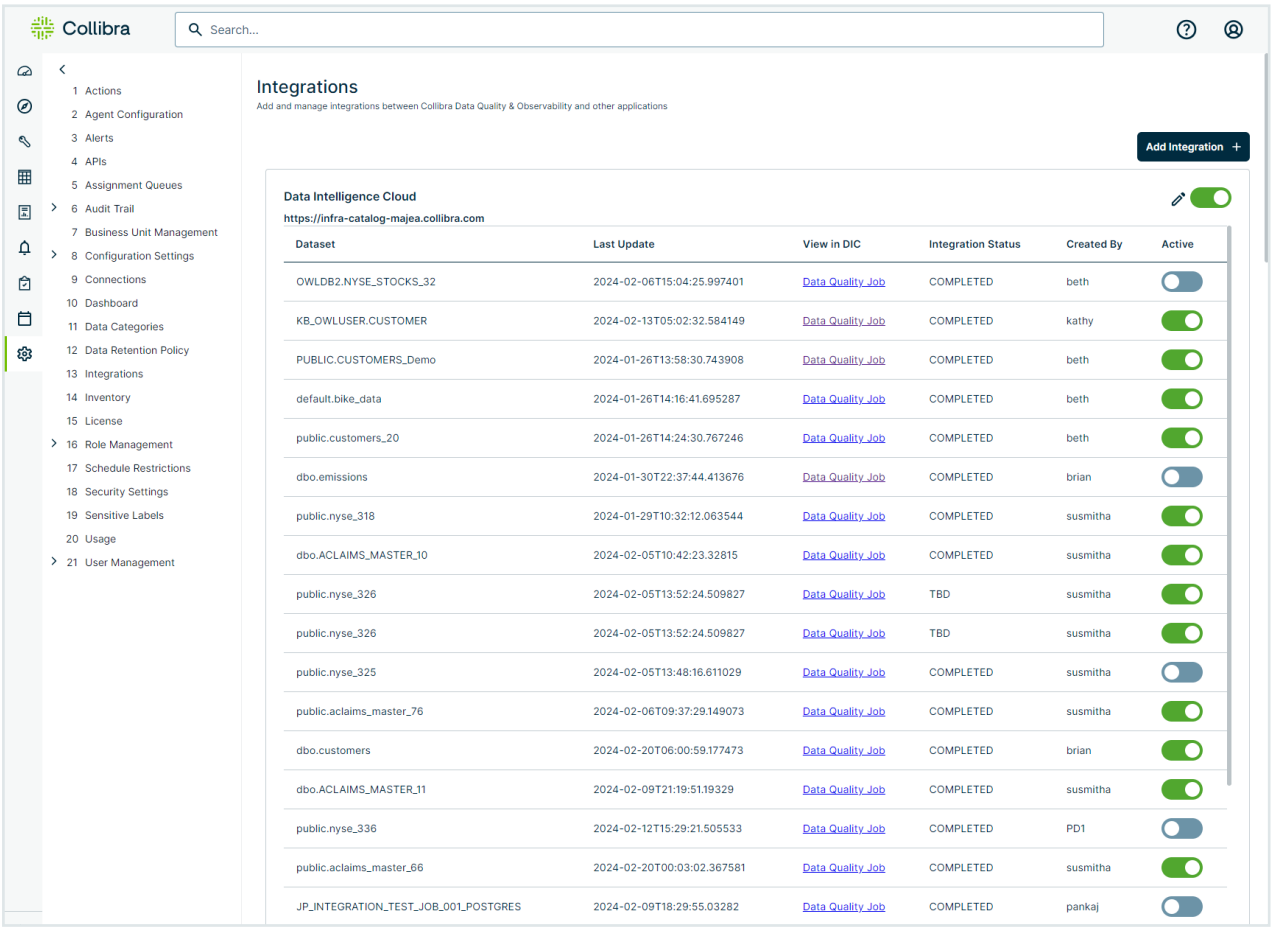
Element	Description	Data
MNPI	Material nonpublic information observed in data-sets during DQ Job runs.	**String dataset**, **String colName**, **dataset**, and **colName**

After a data purge, dataset and runId details are included in the Security Audit.

Integrations

Integrations allows admins to add, edit, and toggle integrations between Data Quality & Observability Classic and other applications, such as Collibra Platform. This page displays all active and inactive integrations that have been configured within Data Quality & Observability Classic.

Tip For more information about configuring integrations, go to Configuring a new integration.



Column	Description
Dataset	The name of the Collibra DQdataset whose metadata integrated into Collibra Platform.
Last Update	The timestamp of the most recent integration of a given dataset in date-time format. For example, 2024-02-24T11:12:35.461612.
Last Run Status	The last job run of your dataset.
Last Integration Status	The status of the last attempt to send metadata to Collibra Platform.
Integration Status	The current status of the integration of your dataset.

Configuring a new integration

This section shows you how to use the Integration configuration wizard to set up a new integration of Data Quality & Observability Classic metadata from profile findings, rules, and alerts into Collibra Platform.

> **Important** If you have multiple integrations set up, only one can be active at a given time.

> **Note** This is a one-time configuration per environment.

Prerequisites

For the Integration API to run successfully, ensure that you review and meet the following criteria.

- Your Data Quality & Observability Classic and Collibra Platform versions are 2025.02 or newer.
- You have an Admin role in Data Quality & Observability Classic to use the configuration wizard.
- You have an Admin or Dataset Manager role in Data Quality & Observability Classic to enable or disable individual datasets for integration.

- Your Collibra Platform profile must have Sysadmin and DataSteward global roles to manage all assets and objects of the integration.
- You have followed the steps on how to add Data Quality Rules to Assets in Collibra Platform.
- You have an aggregation path set for your Data Quality Rule in Collibra Platform ▸ Settings ▸ Operating model ▸ Data quality rules.

> **Important** Do not modify the cardinality of out-of-the-box operating model relations that impact the integration.

Steps

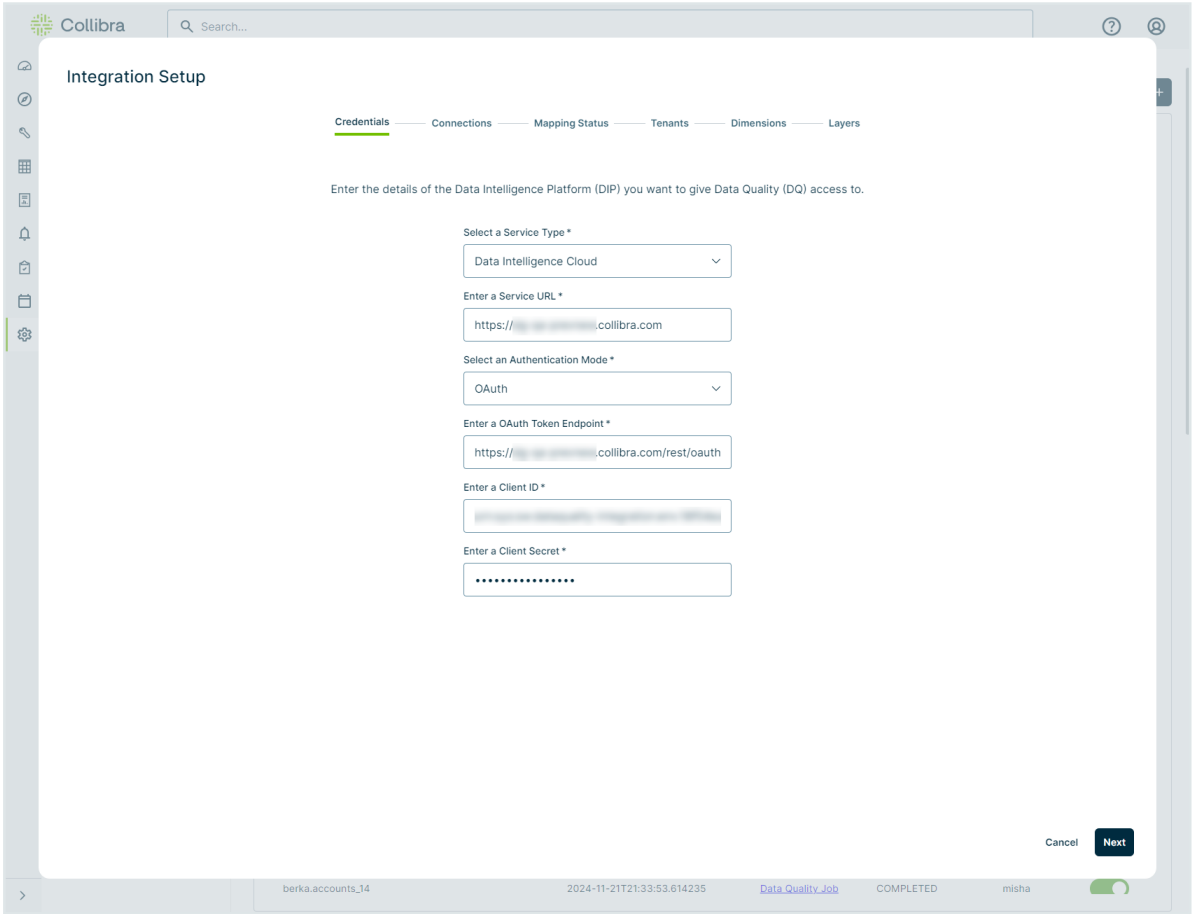
1. Hover your cursor over the ⚙ icon and click **Integrations**.
   » The Integrations page opens.
2. In the upper right corner, click **Add Integration**.
   » The Integration Setup wizard appears.
3. On the **Credentials** step, enter the required information.

Option	Description	Required
Service Type	The application with which to integrate Data Quality & Observability Classic metadata. This value is always **Data Intelligence Cloud**.	✓ Yes
Service URL	The URL of the Collibra Platform environment to send Collibra DQ metadata. For example, "https://dg-example.collibra.com".	✓ Yes
Authentication Mode	The method of authenticating the API call to Collibra Platform. You can select either Basic or OAuth.	✓ Yes

Option	Description	Required
OAuth Token Endpoint	The Collibra Platform URL where Data Quality & Observability Classic sends requests to obtain the OAuth access token. For example, "https://dg-example.collibra.com/rest/oauth/v2/token"	✓ Yes **Tip** Required when using **OAuth** as the authentication mode.
Service Account	The username of your Collibra Platform environment. For example, "integration.user" or "Admin".	✓ Yes **Tip** Required when using **Basic** as the authentication mode.
Password	The password of your Collibra Platform environment.	✓ Yes **Tip** Required when using **Basic** as the authentication mode.
Client ID	The public identifier provided by your OAuth service provider.	✓ Yes **Tip** Required when using **OAuth** as the authentication mode.

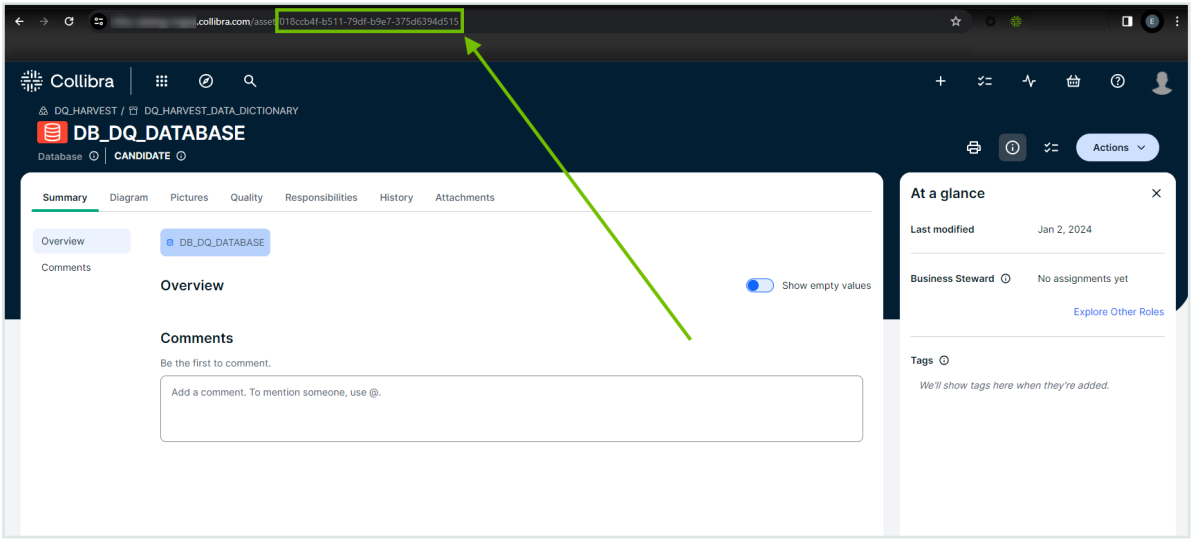
Option	Description	Required
Client Secret	The credential string from your OAuth service provider used to verify the identity of Data Quality & Observability Classic during the token request process.	✓ Yes

> **Tip** Required when using **OAuth** as the authentication mode.



4. Click **Next**.
   » The Connections step opens.
5. On the **Connections** step, map your Collibra DQ connections to the appropriate database, schema, table, and column assets in Collibra Platform.
   a. Optionally, select the **Map to primary column only** switch to see only the column identified as the primary column related to the rule in Collibra Platform.

b. From the **Select Connection** drop-down list, select a Collibra DQ database to integrate into Collibra Platform.
c. In yourCollibra Platform environment, open the asset page of the database you intend to map, then copy the URL from the UUID of the database asset from the URL.



d. From the Connections tab in Collibra DQ, paste the copied UUID from Collibra Platform, or enter the UUID manually.
e. Click **Run** to create an automatic mapping between the schemas within your selected Collibra DQ database and the schemas within Collibra Platform.

> **Important**
> Only schemas ingested in Collibra Platform return when you click Run.

f. When the connection loads, click **Show Full Mapping** to expand the entire database mapping, including schemas, tables, and columns.

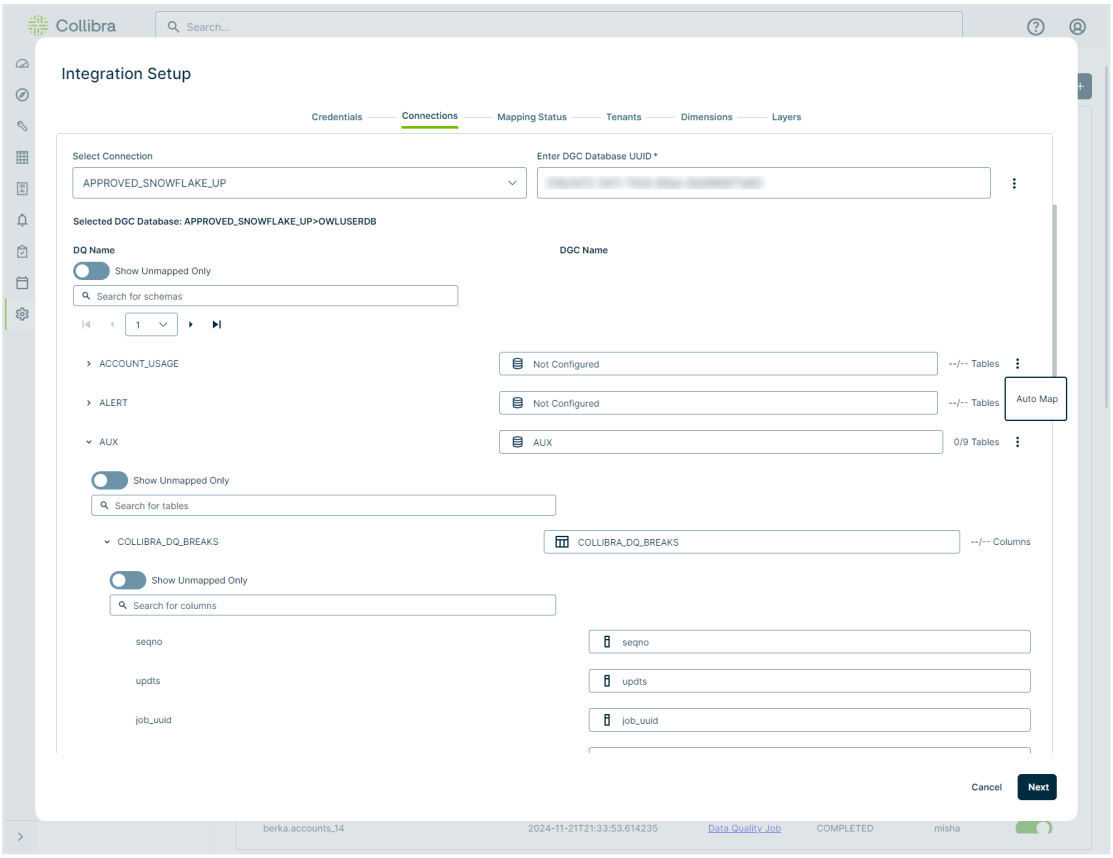
> **Important**
> It is critical that you expand all schemas, tables, and columns you intend to map. If you do not expand each item down to the column level, your table and column relations will not be present in your integration.

i. In the **DGC Name** column, optionally click the drop-down list and select a different schema ( ) to update the schema mapping.

> **Note** When a name match cannot be made, a "Not Configured" message displays.

> **Tip** Optionally click , then **Auto Map** to map tables in your DQ schema to those in DGC automatically. You can track the mapping status or cancel an in-progress mapping on the Mapping Status step.

ii. In the **DQ Name** column, click the schema name to expand the available tables.
   » The selected schema expands to show the available tables, and their corresponding tables ( ) appear across from them in the DGC Name column.
iii. In the **DGC Name** column, click the drop-down list and select a different table to update the table mapping, if necessary.
iv. In the **DQ Name** column, click the table name to expand the available columns.
   » The selected schema expands to show the available columns, and their corresponding columns ( ) appear across from them in the DGC Name column.
v. In the **DGC Name** column, click the drop-down list and select a different column to update the column mapping, if necessary.

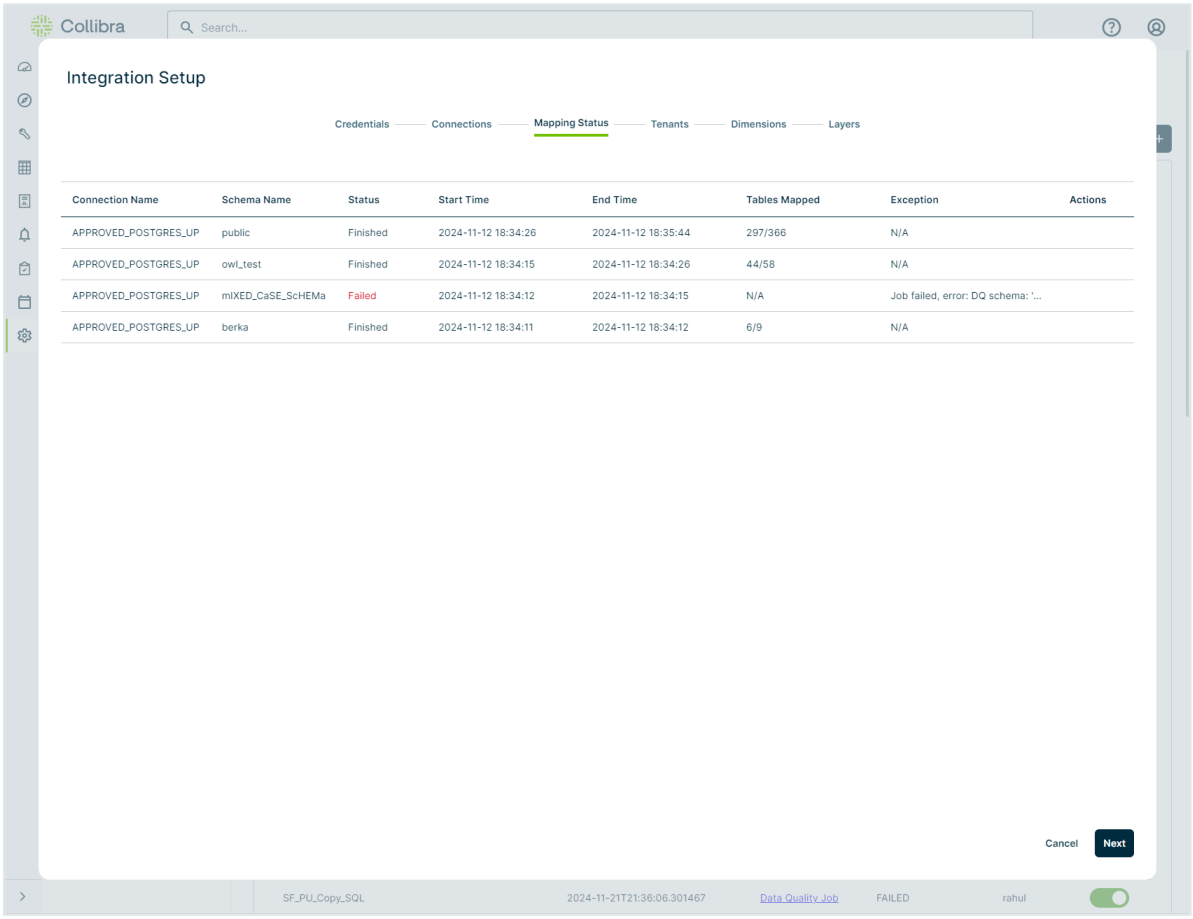


g. Optionally click **Add New** to map another connection and then repeat this process.

6. Click **Next**.
   » The Mapping Status step opens.
7. On the **Mapping Status** step, optionally review mapping details.
   Mapping Status allows you to view mapping details, such as the number of available tables in a schema are mapped and when a connection mapping fails.

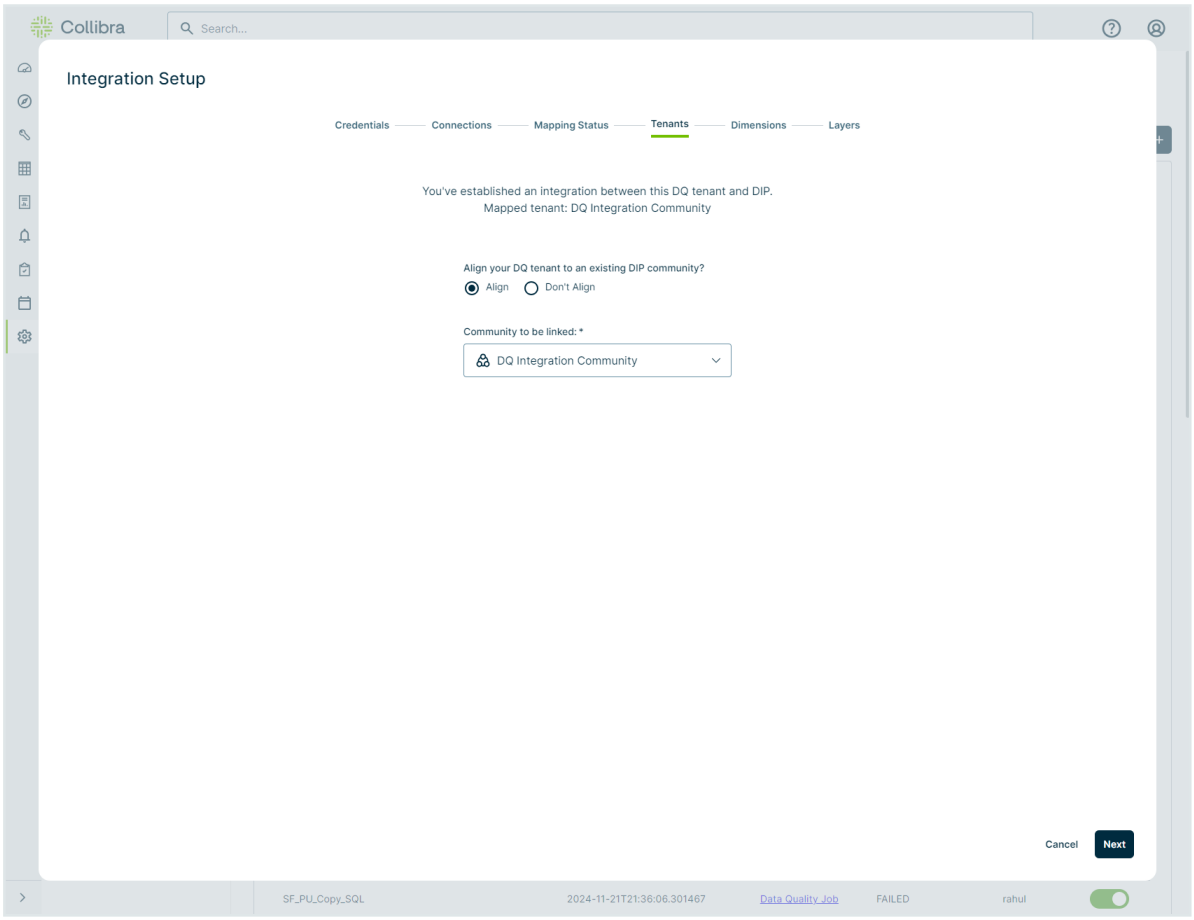
Column	Description
Connection Name	The name of the data source included in the mapping.
Schema Name	The name of the schema included in the mapping.
Status	The state of completion of the mapping.
Start Time	The time stamp, in yyyy-MM-dd hh:mm:ss format, of when the mapping began.

Column	Description
End Time	The time stamp, in yyyy-MM-dd hh:mm:ss format, of when the mapping stopped.
Tables Mapped	The number of tables that mapped out of the total tables available in the schema.
Exception	When an error during the mapping occurs, a message describes the issue. If the error message is truncated, you can hover your cursor over it to view the full message in a tooltip. When an exception did not occur, the value in this column is N/A.
Actions	When a mapping job is in progress, a option appears in this column to allow you to cancel the job if necessary.



8. Click **Next**.
   » The Tenants step opens.

9. On the **Tenants** step, enter the required information.
    a. Optionally, choose to align your Collibra DQ tenant with an existing Collibra Platform community.
        i. Align
            - Select a Collibra Platform community from the drop-down list with which to align your tenant.
        ii. Don't Align,
            - Enter the name of a new community with which to align your tenant.

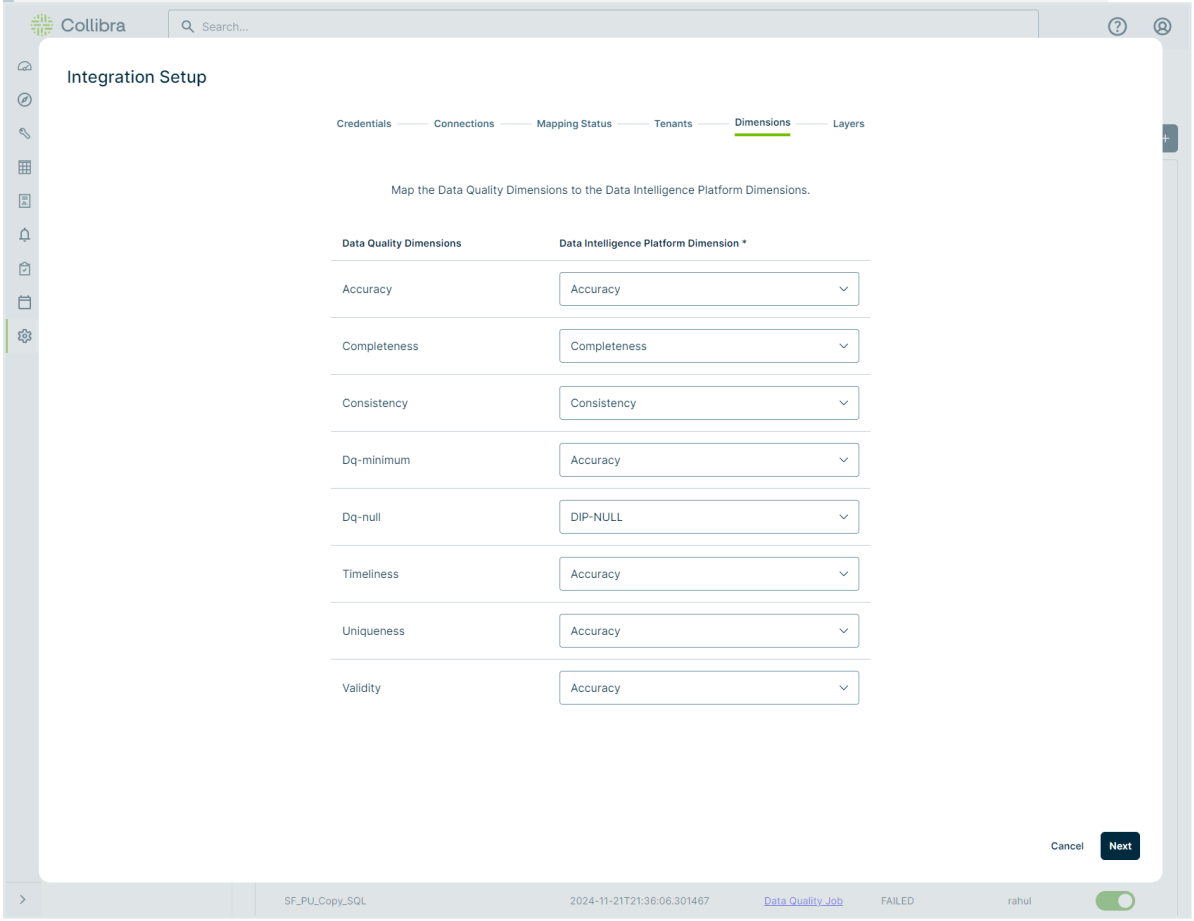


> **Note** Beneath each tenant-level community, a sub-community is created for each Business Unit that has datasets assigned to it in Collibra Platform. If a dataset is not assigned to a Business Unit, it is added to a community called *Unassigned DQ Job* by default.

10. Click **Next**.
    » The Dimensions step opens.
11. On the **Dimensions** step, enter the required information.

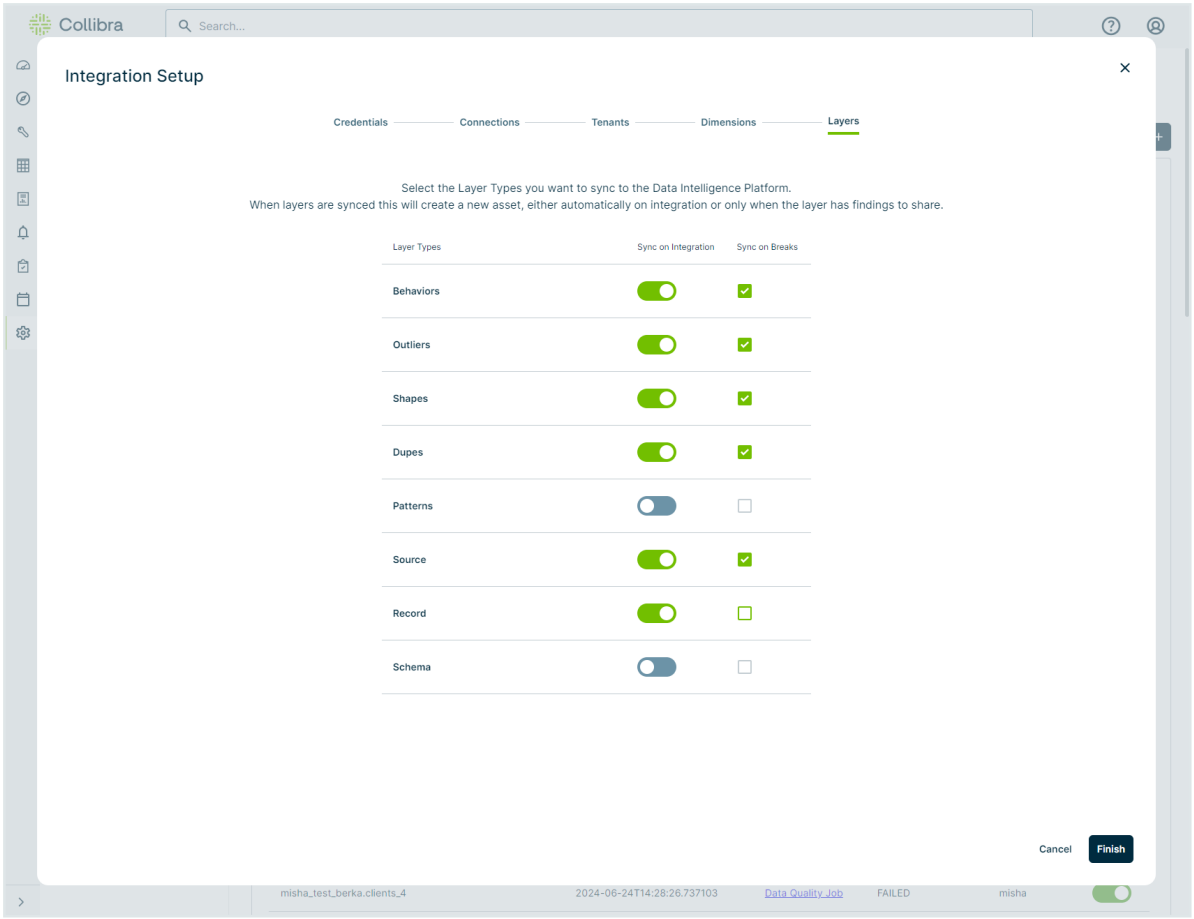
a. Select a dimension from the drop-down list of the Data Intelligence Platform Dimension column that corresponds with the Data Quality Dimension you want to identify as a match.

> **Note** All Data Quality Dimensions from your Collibra Platform environment are available to map to Data Quality Dimensions in Collibra DQ, including custom dimensions. However, Data Quality Dimensions from Collibra DQ are not customizable at this time.



12. Click **Next**.
    » The Layers step opens.
13. On the **Layers** step, select which layers to synchronize to Collibra Platform.
    a. Select a layer in the **Sync on Integration** column to automatically create a new asset in Collibra Platform when the integration runs.
    b. Select the **Sync on Breaks** option to create a new asset in Collibra Platform only

when the layer has a finding.



14. Click **Finish**.

Known limitations

The integration only processes the primary schema or table and their associated columns. This is because the columns of the Collibra Platform table used by the integration map to the Data Quality & Observability Classic rules that contain them. A possible workaround to map columns from other tables referenced in the load query is to use a view to join multiple tables, then use that view as the Data Quality & Observability Classic data source.

What's next?

Enable a dataset for integration and start ingesting metadata from Collibra DQ into Collibra Platform.

Modifying an existing integration

This section shows you how to use the Integration Configuration Wizard to modify an existing integration of Data Quality & Observability Classic with Collibra Platform.

Steps

1. Hover your cursor over the ⚙ icon and click **Integrations**.
   » The Integrations page opens.
2. On the right side of the status table, click the ✎ icon.
   » The Update Integration Credentials wizard appears.
3. Update the information on as many tabs as necessary.
4. Click **Save**.

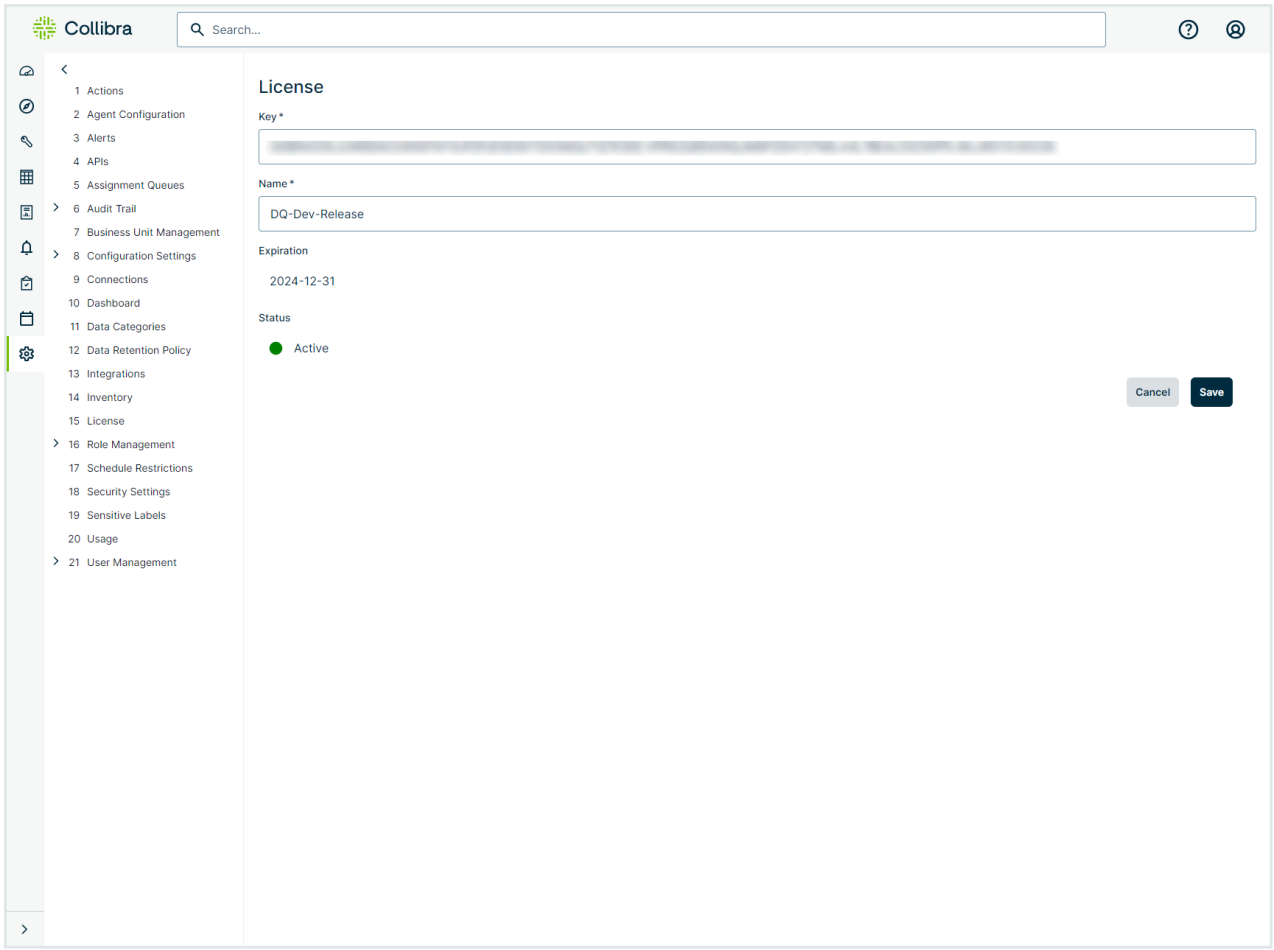
Inventory

Inventory lists all the DQ Jobs that have ever run in Data Quality & Observability Classic.

Column	Description
Run date	The date and timestamp of the DQ Job run.
Dataset	The dataset name.
Command line	Shows the command line of the DQ Job run.
File	When the data source is a remote file, the name of the file upon which the DQ Job is based displays in this column.
Query	The SQL query that ran against the dataset.
Connection	The data source connection upon which the DQ Job is based.

License

The License page is where you can view and manage your Data Quality & Observability Classic license information, including the license key, license name, expiration date, and whether your Collibra DQ license is currently active or inactive.



Managing Collibra DQ licenses

This section shows you how to manage your license information provisioned by Collibra.

> **Note** Please refer to your contract for the exact validity period of your license.

Prerequisites

You have:

- Admin (ROLE_ADMIN) privileges.
- A license previously provisioned by Collibra.

Steps

1. From the Admin panel, select **License** to open the License page. The table below shows the options available on this page.
2. Manage your license by configuring the options in the following table.

Option	Description	Can edit
Key	Displays your current license key. Starting 30 days before your license expires, a counter in a banner message on the Collibra DQ home page displays how many days remain before your license expires. You can update your license key in this field to avoid any interruptions in the functionality of your DQ Jobs. Click into the field and enter your text to edit your license key. > **Note** To obtain a new license key, reach out to your CSM for assistance.	✓ Yes
Name	Displays your current license name. Click into the field and enter your text to edit your license name.	✓ Yes
Expiration	Displays the expiration date of your license in a yyyy-mm-dd format.	✕ No
Status	Displays the current status of your license as either Active or Expired. A green indicator means that your license is active. A red indicator means that your license is inactive or expired.	✕ No

3. Click **Save** after making changes to the license options.

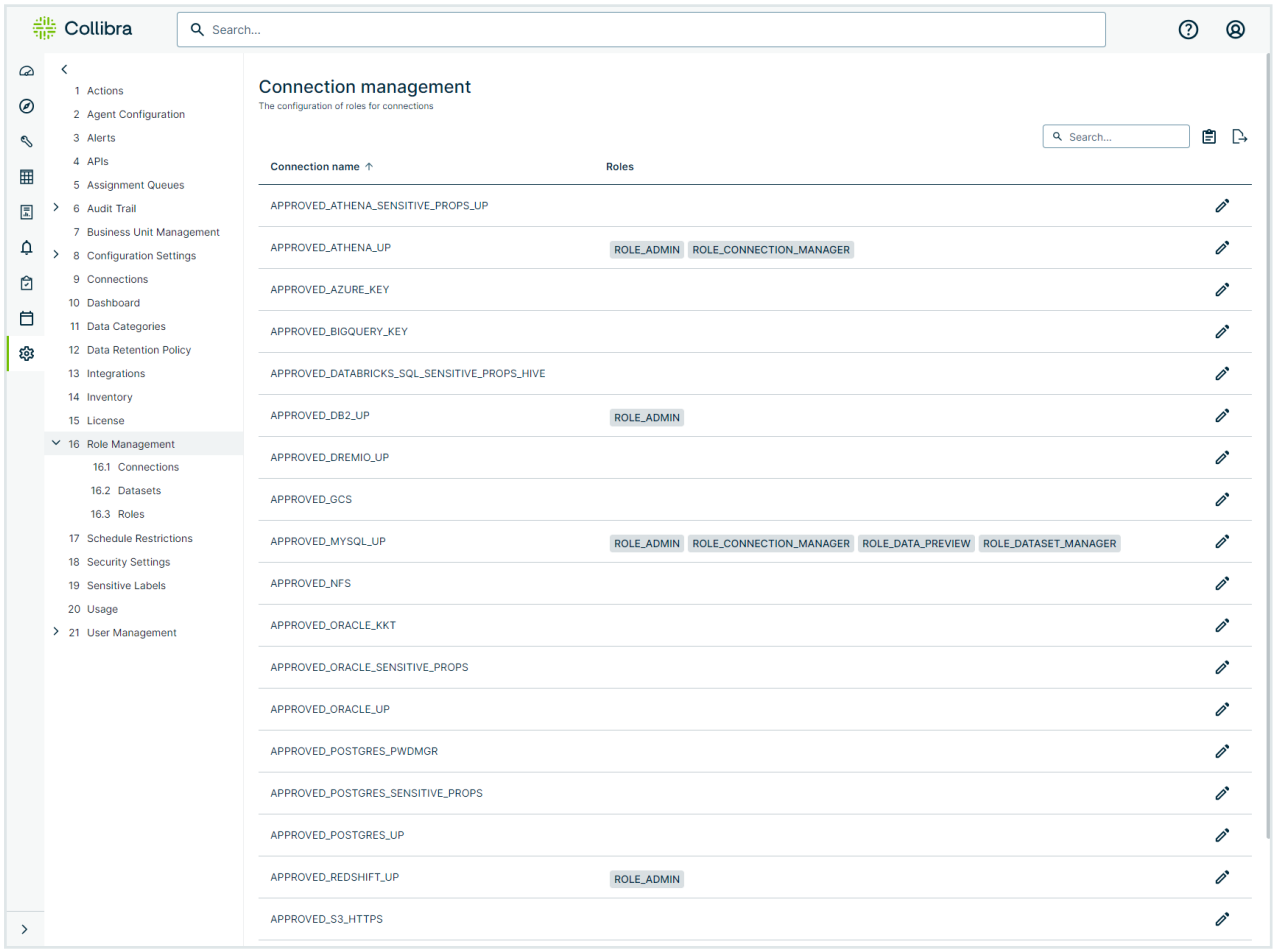
> **Warning** Any DQ Jobs that run on accounts with expired or invalid license keys will no longer run until a valid license key is saved.

Role Management

Connection Management

Connection management allows administrators to manage the roles mapped to specific connections so that only users with those roles can access those connections.

When one or more roles are mapped to a data source connection, only users with one or all of the roles can access data from that connection. Any users without one of or all of the roles mapped to the connection are restricted from accessing it. When roles are not mapped to a connection, all users can access it.



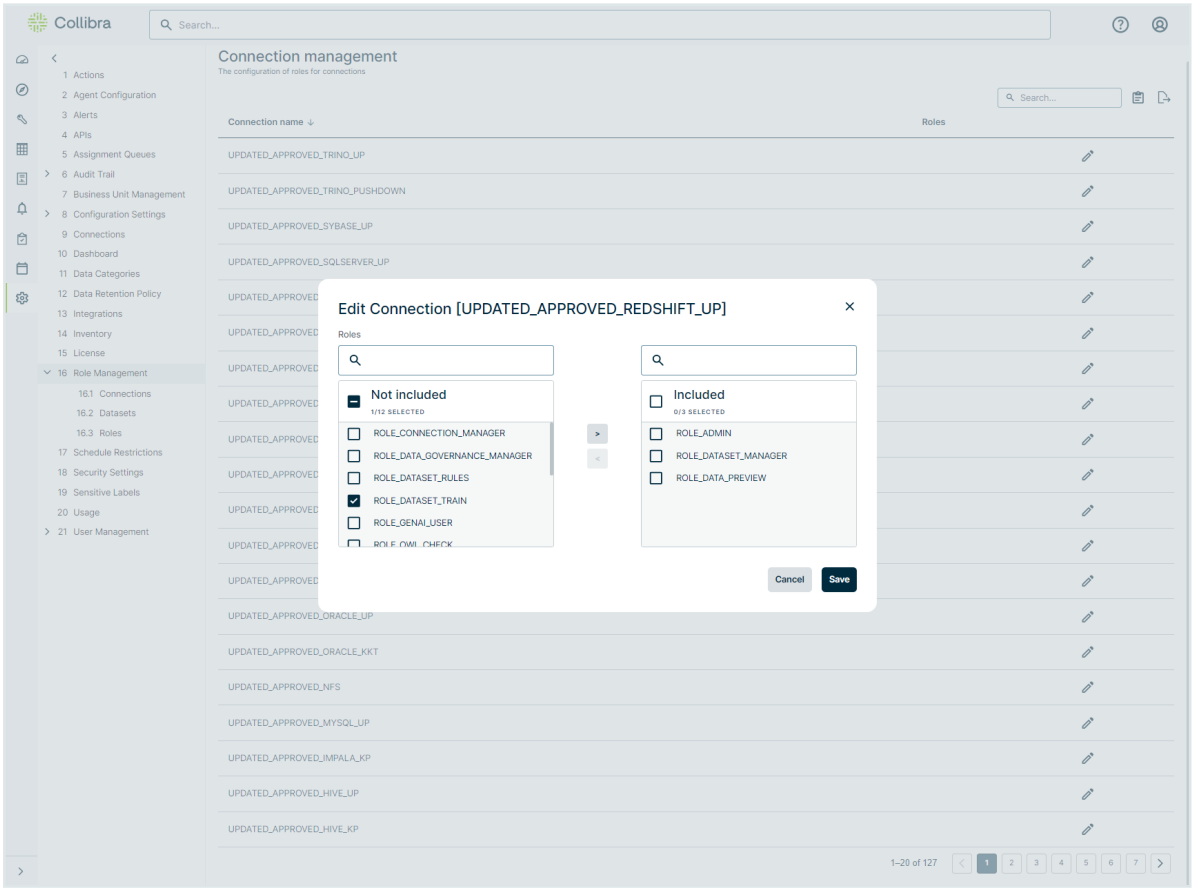
Adding roles to connections

Administrators can manage the roles assigned to connections on the Connection Management page.

Steps

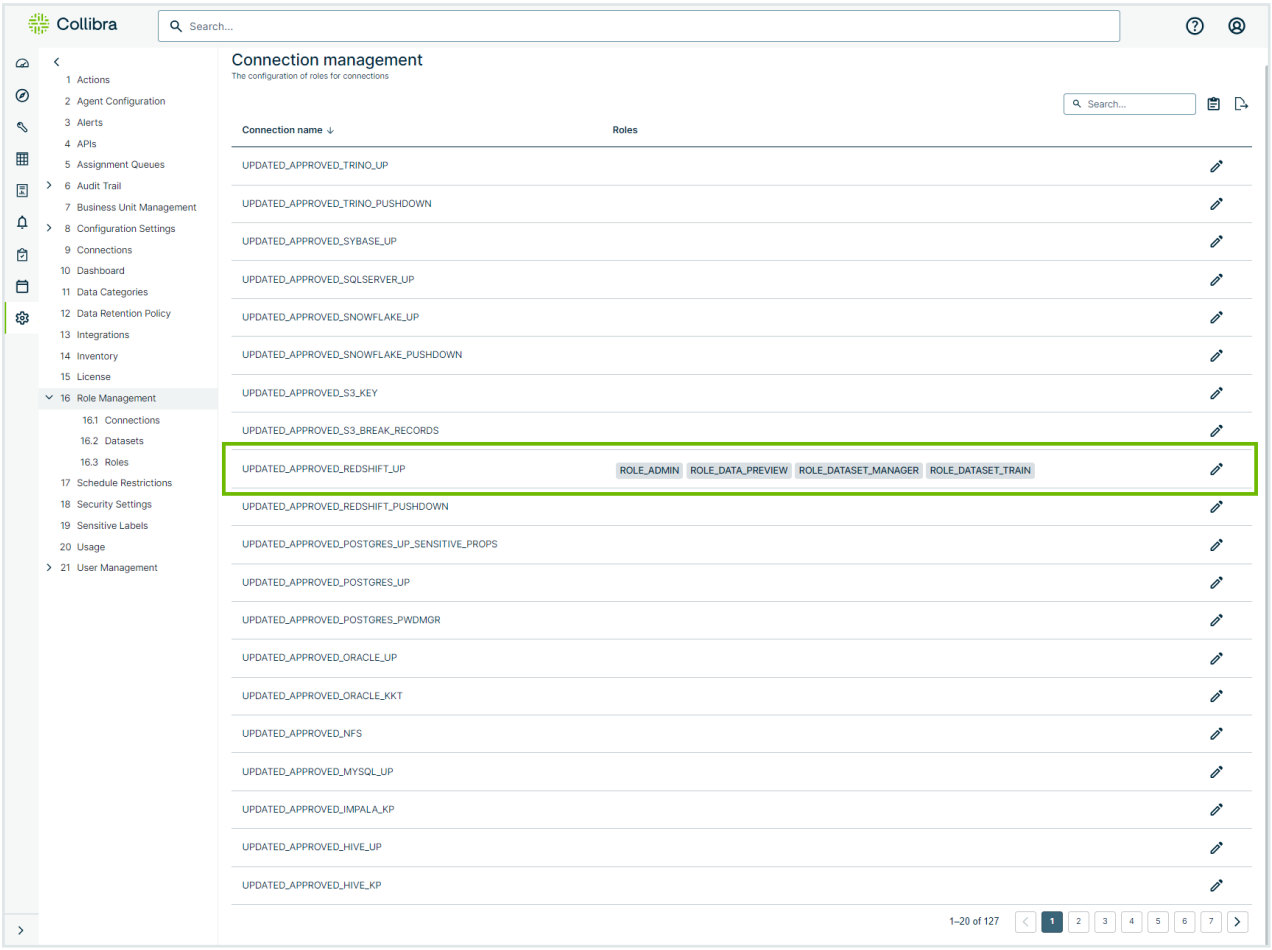
1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Role Management** and then click **Connections**.
4. In the far right column, click .
   » The **Edit Connection [your connection]** modal displays for that specific role.
5. Map roles to your connection by moving the roles from the **Not included** box to the **Included** box. To do this, select the role(s) you wish to map to your connection, then click . You can also filter the roles in the search fields.

> **Note** To remove a role mapping, you can click the left arrow to move the role(s) you select from the **Included** column back to the **Not included** column.



6. Click **Save**.

The role(s) you mapped to the connection display in the Roles column of the Connection Management table.



> **Note** After mapping roles to a connection, any user with those mapped roles can see the connection in Collibra DQ.

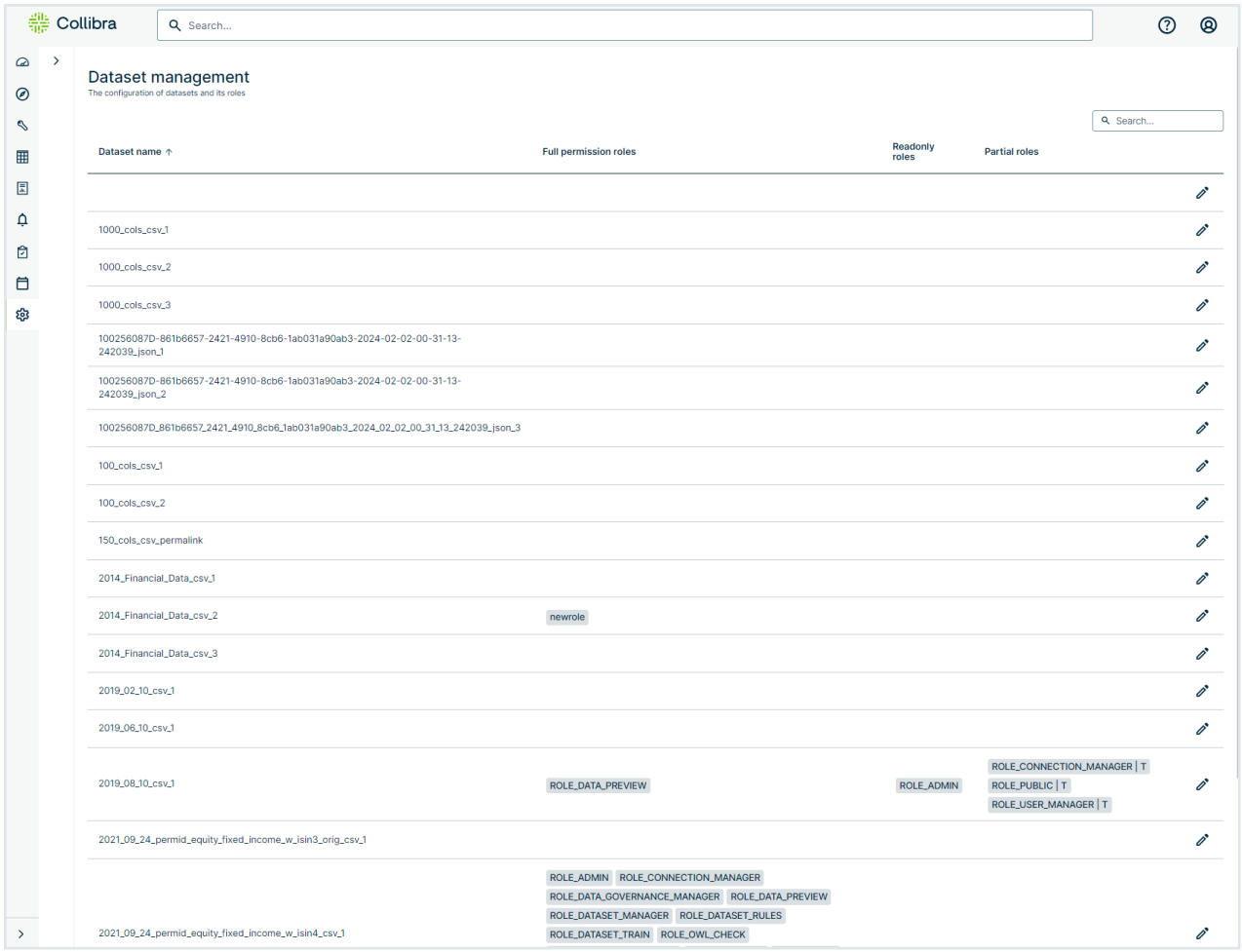
Dataset Management

Dataset management allows admins to configure the roles that can access specific datasets in Data Quality & Observability Classic. There are three different degrees of access permissions that can be mapped to datasets:

- **Full permission roles**: Roles that allow unrestricted access to the dataset to which they are assigned. These roles can:

    - Use Explorer to create, edit, and run DQ Jobs.
    - Access the Findings page to see results from DQ Job runs.
    - View source data in Data Preview.
    - Access the Rules page to write and manage rules.
    - Retrain datasets.
- **Read-only roles**: Roles that allow access to the Findings page to see results from DQ Job runs, but have restricted access to other areas of the application. These roles have:
    - Restricted access to Explorer to create, edit, and run DQ Jobs.
    - Restricted access to view source data in Data Preview.
    - Restricted access to the Rules page to write and manage rules.
    - Restricted access to retrain datasets.
- **Partial roles**: Roles that allow any combination of Data Preview access, the ability to retrain datasets, and the ability to access the Rules page to write and manage rules.

> **Note** When Dataset Security is enabled, dataset access is required to edit, map, or retrieve datasets or business units. For more information about Collibra DQ roles, see Role Management.



Assigning roles to datasets

You can manage the roles assigned to datasets on the Dataset Management page.

Prerequisites

You have enabled ACLs for dataset access on the Security Settings page in the Admin Console.

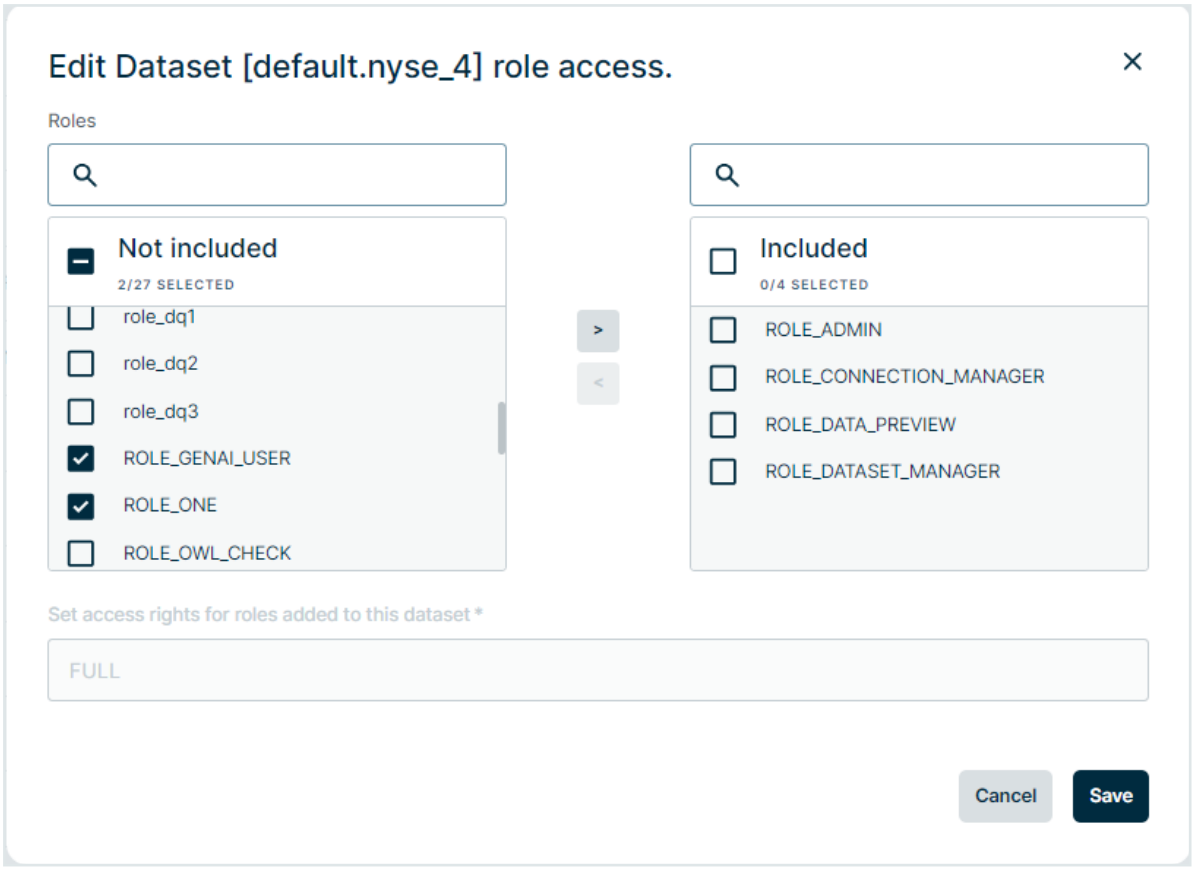
Note The access rights for roles dropdown menu is unavailable unless you enable ACLs for dataset access. We recommend selecting all four ACL options, including:

- Enable ACLs for dataset access (this is mandatory to access the access rights for roles dropdown menu)
- Require DATA_PREVIEW role to see source data
- Require DATASET_TRAIN role for dataset training access
- Require DATASET_RULES role for dataset rule create/edit access

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Role Management** and then click **Datasets**.
4. In the far right column in the row containing your dataset, click .
   » The **Edit Dataset [your dataset] role access** modal displays.
5. Map roles to your dataset by moving the roles from the **Not included** box to the **Included** box. To do this, select the role(s) you wish to map to your dataset, then click . You can also filter the roles in the search fields.

   Note You can remove mapped roles from the dataset by selecting the role from the **Included** box and then clicking .



6. Click the **Set access rights for roles added to this dataset** dropdown menu and select one of the following options:

> **Important** This step is only available when Dataset Security is enabled on the Security Settings page.

Permission	Description
FULL	Grants users with the roles mapped to this dataset unrestricted access to the dataset to which they are assigned. These roles can: • Use Explorer to create, edit, and run DQ Jobs. • Access the Findings page to see results from DQ Job runs. • View source data in Data Preview. • Access the Rules page to write and manage rules. • Retrain datasets.

Permission	Description
PARTIAL	Grants users with the roles mapped to this dataset any combination of Data Preview access, the ability to retrain datasets, and the ability to access the Rules page to write and manage rules. After you select this option, you can grant partial access to the roles mapped to the datasets by selecting one or more of the following options: • Enable Data Preview Access • Enable Training • Enable Rules Create/Edit
READ_ONLY	Grants users with the roles mapped to this dataset access to the Findings page to see results from DQ Job runs, but restricts their access to other areas of the application. These roles have: • Restricted access to Explorer to create, edit, and run DQ Jobs. • Restricted access to view source data in Data Preview. • Restricted access to the Rules page to write and manage rules. • Restricted access to retrain datasets.

7. Click **Save**.

> **Note** When Dataset Security is enabled, dataset access is required to edit, map, or retrieve datasets or business units.

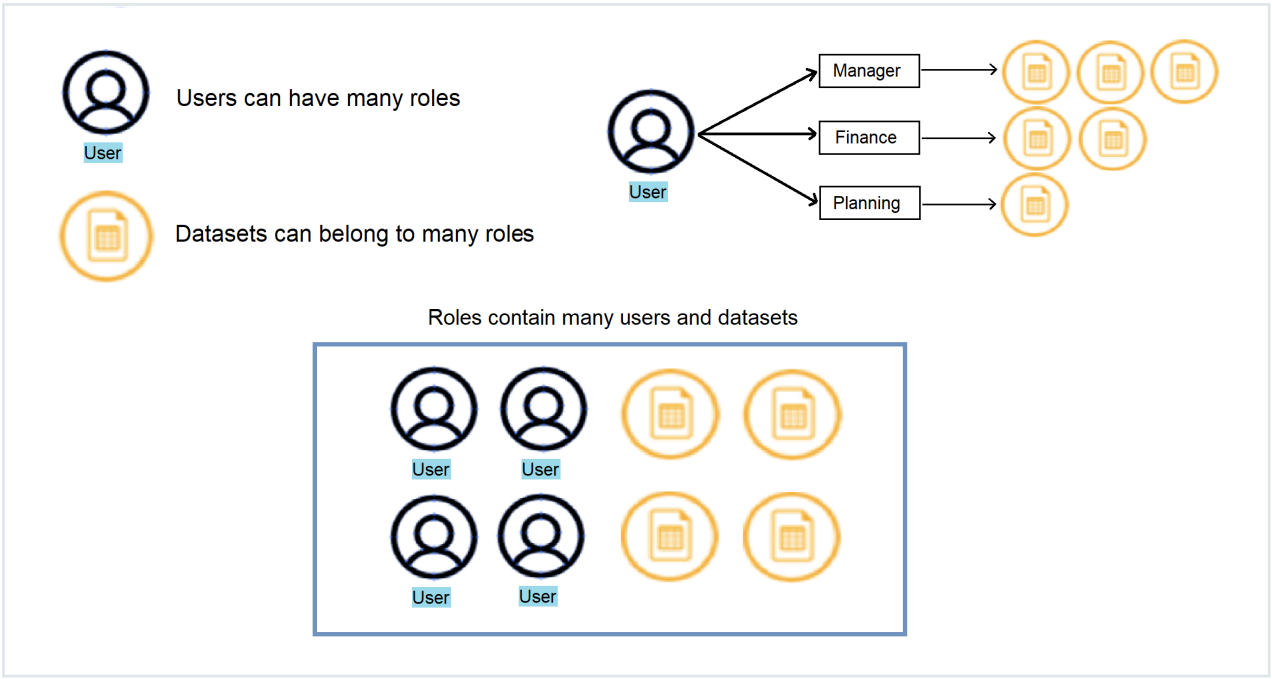
For a better understanding about Collibra DQ roles, go to Role Management.

Role Management

Role management allows users with ROLE_ADMIN, ROLE_USER_MANAGER, or ROLE_OWL_ROLE_MANAGER to manage the roles assigned to Data Quality & Observability Classic users.

Access to different parts of the application and its functions can be restricted based on user roles, allowing users to belong to one or more roles that align with their responsibilities within your organization. This type of application security is referred to as role-based access control (RBAC), which is the method of regulating user access to the Data Quality & Observability Classic application by the permissions granted through role assignments. RBAC is enhanced

by leveraging out of the box or custom roles, particularly when combined with security settings.



In this topic

- Internal and external users
- Out of the box roles
- Security settings and user tasks
- Custom user roles
- Assigning users to roles
- Assigning user roles to datasets

Internal and external users

Data Quality & Observability Classic supports internal and external users, who are authenticated through different mechanisms.

Internal users are those who follow the registration link on the sign in page, complete the registration form, and are approved and assigned roles by a user with ROLE_ADMIN, ROLE_USER_MANAGER, or ROLE_OWL_ROLE_MANAGER. The default Admin user account that is created automatically upon installation of Data Quality & Observability Classic is also

considered an internal user. Internal users are authenticated when they enter their Username and Password on the sign in page.

> **Important**
> When a user creates a new account and the security setting **Auto approve users** is disabled, another user with ROLE_ADMIN, ROLE_USER_MANAGER, or ROLE_OWL_ROLE_MANAGER must grant the new user account at least one role before the new user can sign in to Data Quality & Observability Classic This is unnecessary when Auto approve users is enabled, because new user accounts are unlocked and have ROLE_PUBLIC by default.

External users are users of the server where Data Quality & Observability Classic is installed. As an alternative to the basic internal user authentication method, admins can grant external users access to Data Quality & Observability Classic by mapping external groups from authentication mechanisms such as Active Directory (AD) to Data Quality & Observability Classic roles, which allows them to pass the authentication requirements and sign in.

> **Note**
> When an external user's associated userid object is purged from an external user store, such as AD, they can no longer sign in to Data Quality & Observability Classic, as the system cannot locate the userid object needed to authenticate their account.

Out of the box roles

Data Quality & Observability Classic comes with many out of the box roles designed to provide different users varied levels of access to components of the application.

Out of the box roles can be grouped into three categories:

- Admin roles: These roles have privileges to perform some system-control tasks.
- Technical steward roles: These roles have privileges to run DQ Jobs, manage rules and datasets, and view and analyze DQ Job findings.
- Business steward roles: These roles have privileges to view and analyze scorecards, but do not have dataset access when dataset security is enabled.

Admin roles

Role	Description
ROLE_ADMIN	Users with this role can manage connections, security and admin settings, and delegate roles. This is the only role that can delete datasets. **Tip** Users with this role can assign roles to other users.
ROLE_DATA_GOVERNANCE_MANAGER	Users with this role can create, edit, and delete business units, data categories, and meta tags.
ROLE_USER_MANAGER	Users with this role can create and edit users and add users to roles. **Tip** Users with this role can assign roles to other users.
ROLE_OWL_ROLE_MANAGER	Users with this role can create custom roles and assign and edit the mappings of roles to users, AD groups, and datasets. **Tip** Users with this role can assign roles to other users.
ROLE_CONNECTION_MANAGER	Users with this role can create, edit, and delete connections to data sources.
ROLE_DATASET_MANAGER	Users with this role can perform the following tasks: • Map roles to datasets • Mask dataset columns • Re-run or delete scheduled DQ Jobs • Assign new user accounts to roles • Assign or edit the roles assigned to datasets on the Dataset Management page • When **Dataset security** is enabled, users with **ROLE_DATASET_MANAGER** can configure the roles that have access to specific datasets

Role	Description
ROLE_ADMIN_VIEWER	Users with this role can access the following Admin Console pages: • Actions • All Audit Trail sub-pages • Dashboard **Note** Users with ROLE_ADMIN_VIEWER cannot access the pages to which the quick access buttons on the Dashboards page are linked. • Inventory • Schedule Restrictions **Note** Users with ROLE_ADMIN_VIEWER cannot add or delete schedule restrictions. • Usage **Tip** All other Admin Console pages are not accessible to users with this role.

Users with admin roles are typically responsible for a broad range of tasks related to system control, such as the management of users, connections, and datasets. When dataset security is disabled, users with ROLE_ADMIN have unrestricted access to any part of Data Quality & Observability Classic, including the Admin Console.

Other admin roles maintain access to major sections of the application, but do not necessarily have the global permissions that ROLE_ADMIN does. For example, a user with ROLE_CONNECTION_MANAGER can set up and test connections to data sources, but cannot create or edit datasets unless they also have ROLE_DATASET_MANAGER.

Common tasks performed by users with admin roles include but are not limited to:

- Creating and deleting connections to data sources.
- Creating and deleting users.
- Creating and deleting roles.
- Managing users.
- Assigning roles to users.

- Assigning datasets to roles.
- Creating and deleting business units and data categories.
- Enabling and disabling security options.
- Deleting datasets.

Business steward roles

Role	Description
ROLE_PUBLIC	Users with this role can access scorecards and reports. When the dataset security setting Dataset security is enabled, users with this role cannot access datasets. When used in conjunction with other security settings, this can be used as a "read-only" role, whereby users with ROLE_PUBLIC cannot create DQ Jobs, access or manage connections, and so on. When this role is assigned to a dataset on the Dataset Management page, the dataset is accessible to all users, irrespective of role.

ROLE_PUBLIC holds the lowest level of privilege of all the out of the box roles. By default, all user accounts have the privileges granted to ROLE_PUBLIC when they are created, even if they are not explicitly assigned to it.

Technical steward roles

> **Important** Many of the roles described in the table below are heavily dependent on whether corresponding security settings are enabled or disabled. As such, it is important that you are familiar with the implications of security settings before assigning these roles to users and datasets.

Role	Description
ROLE_OWL_CHECK	When the general security setting DQ Job security is enabled, users with this role can run DQ Jobs.
ROLE_DATA_PREVIEW	When the dataset security setting Require DATA_PREVIEW role to view source data is enabled on the Security Settings page, users with this role can view the following: • Sample data on dataset profile pages • Sample data on dataset findings pages

Role	Description
ROLE_DATASET_TRAIN	When the dataset security setting **Require DATASET_TRAIN role for dataset training access** is enabled, users with this role can train datasets. Users with this role can validate, invalidate, resolve, and retrain dataset findings.
ROLE_DATASET_RULES	When the dataset security setting **Require DATASET_RULES role for dataset rule create/edit access** is enabled, users with this role can manage data quality rules.
ROLE_DATASET_ACTIONS	When the dataset security setting **Require DATASET_ACTIONS role for dataset rule create/edit access** is enabled, users with this role can manage DQ Jobs from the Dataset Manager.
ROLE_VIEW_DATA	Controls which users can access the DQ SQL editor to run the SQL against the database. Users with this role can view the following: • Dataset Overview results • Sample data on dataset profile pages (if the general security setting **DB connection security** is also enabled) • Sample data on dataset findings pages (if the general security setting **DB connection security** is also enabled)
ROLE_GENAI_USER	Users with this role can use SQL Assistant for Data Quality for automated SQL rule writing and troubleshooting.
ROLE_USER	Do not use this role, as it was deprecated.

Users with technical steward roles are typically responsible for specific tasks, such as running DQ Jobs and reviewing their findings. Therefore, users with data steward roles often have one or more of these roles assigned to them for more granular access to these specialized tasks. For example, a user responsible for creating SQL rules to run against a dataset once it exists might have the rule writer role, ROLE_DATASET_RULES.

Common tasks performed by users with technical steward roles include all data analyst tasks, as well as the following:

- Creating, running, and terminating DQ Jobs.
- Managing dataset rules.
- Previewing data in DQ Job findings.
- Training datasets.
- Masking columns in datasets.
- Re-running and deleting scheduled DQ Jobs.

Security settings and user tasks

When the security setting **Dataset security** is enabled, users with roles that ordinarily have unrestricted access to datasets, including ROLE_ADMIN, must be assigned to the dataset they are attempting to access in order to access it.

When the Dataset Security setting **Default owner dataset access** is enabled, unless a dataset is assigned to other users, only the dataset owner can run DQ Jobs against it, view its findings, or manage its rules. Otherwise, even the dataset owner must be assigned to the dataset to access it.

Access to datasets

- When the Dataset Security setting **Default owner dataset access** is enabled, the creator of a dataset is automatically granted access and other users must be granted access via a role.
- When a dataset is assigned to a role, all users with this role can access it.
- When a dataset is assigned to ROLE_PUBLIC, all users can access it.

Data steward and data analyst tasks with security settings enabled

The table below shows how different security settings can restrict which roles can perform certain tasks when enabled.

Security setting	Description
Dataset security	When enabled, if a user does not have a full permission role granting them the ability to perform specific tasks, they cannot perform them. The following list shows specific tasks that dataset security can prevent users without the correct full permission role from performing: • Run DQ Jobs • Manage rules • View DQ Job findings • Preview data • Take action on DQ Job findings • Train datasets • Mask columns • View assignments • Assign business units and data categories **Note** In the list above, a single role encompassing all tasks does not exist. Instead, multiple roles must be mapped to relevant datasets as full permission roles to allow users to perform all listed tasks. Users with ROLE_ADMIN or ROLE_DATASET_MANAGER can map roles to datasets as full permission roles on the Dataset Management page.

<table>
<tr><th>Security setting</th><th>Description</th></tr>
<tr><td>DQ Job security</td><td>When DQ Job security is enabled, Dataset security is disabled, and a user does not have ROLE_ADMIN or ROLE_OWL_CHECK assigned to the dataset to which they are attempting to access, they cannot create, run or schedule DQ Jobs.

Similarly, when DQ Job security and Dataset security are enabled and a user has a role other than ROLE_ADMIN or ROLE_OWL_CHECK assigned to the dataset to which they are attempting to access, they cannot create, run or schedule DQ Jobs.

However, when DQ Job security and Dataset security are enabled and a user has ROLE_ADMIN or ROLE_OWL_CHECK assigned to their user account and the dataset to which they are attempting to access, they can create, run or schedule DQ Jobs.

Show table
<table>
<tr><th>DQ Job security</th><th>Dataset security</th><th>Role</th><th>DQ Job access level</th></tr>
<tr><td>Enabled</td><td>Disabled</td><td>No role assigned to user or dataset</td><td>• Cannot create
• Cannot run
• Cannot schedule</td></tr>
<tr><td>Enabled</td><td>Enabled</td><td>A role other than ROLE_ADMIN or ROLE_OWL_CHECK assigned to the dataset</td><td>• Cannot create
• Cannot run
• Cannot schedule</td></tr>
<tr><td>Enabled</td><td>Enabled</td><td>ROLE_ADMIN and/or ROLE_OWL_CHECK assigned to the user and dataset</td><td>• Can create
• Can run
• Can schedule</td></tr>
</table>
</td></tr>
<tr><td>Require DATA_PREVIEW role to see source data</td><td>When enabled, only users with ROLE_DATA_PREVIEW can preview source data on the Explorer, Findings, Profile, Rules, and Dataset Overview pages.</td></tr>
<tr><td>Require DATASET_TRAIN role for dataset training access</td><td>When enabled, only users with ROLE_DATASET_TRAIN can train datasets, which includes the ability to adjust the scoring model, modify thresholds of Adaptive Rules, and validate, invalidate, or resolve data quality findings.</td></tr>
</table>

Security setting	Description
Require DATASET_ RULES for data-set rule cre-ate/edit access	When enabled, only users with ROLE_DATASET_RULES can manage rules. **Note** ROLE_DATA_PREVIEW is also required to manage rules.
Require DATASET_ ACTIONS role for dataset man-agement actions	When enabled, only users with ROLE_DATASET_ACTIONS can edit, rename, pub-lish, assign data categories and business units, enable integrations, and schedule DQ Jobs from the Dataset Manager.
Default owner dataset access	The default owner of a dataset is the user who created it. When enabled, this option automatically grants access to the creator of a given dataset. When this option is disabled, datasets do not have a default owner. When **Dataset security** and **Default owner dataset access** are both enabled, users with a role mapped to a dataset **or** the dataset owner can view, add, edit, and remove the dataset to which they are assigned or own. **Note** Users without a role mapped to any existing dataset can still create new datasets, view their profiles, and view, add, and remove business units from them. Additionally, when **Dataset security** and **Default owner dataset access** are both enabled, users can view, edit, and remove business units on datasets to which they are assigned or own. **Note** Only users with ROLE_ADMIN or ROLE_DATA_GOVERNANCE_ MANAGER can create business units, which they can then add to datasets.

Important Some tasks, such as masking columns and re-running or deleting scheduled DQ Jobs, can only be performed by users with ROLE_ADMIN and ROLE_DATASET_ MANAGER, even with all of the security settings disabled.

Custom user roles

Custom roles implement **Connection security** and **Dataset security** and are intended to provide users with dataset and connection access, not operational controls.

If connection and/or dataset security are enabled, users with ROLE_ADMIN, ROLE_USER_MANAGER, or ROLE_OWL_ROLE_MANAGER should create custom roles to grant users access to specific datasets and/or connections.

When using Active Directory, you can also add custom roles as needed from the Mappings tab on the Active Directory Security Settings page.

For instructions on how to create custom roles, see Adding custom roles.

Assigning users to roles

When a new user account is created in Data Quality & Observability Classic, it does not have any roles assigned to it by default. However, all user accounts have the privileges granted to ROLE_PUBLIC when they are created, even though they are not explicitly assigned to it out of the box. Therefore, it is best practice for users with ROLE_ADMIN, ROLE_OWL_ROLE_MANAGER, or ROLE_DATASET_MANAGER to assign new user accounts to roles to allow them to access the parts of the application relevant to them and restrict access to other parts that are not.

For instructions on how to add users to roles, see Adding users to roles.

Assigning user roles to datasets

> **Important** This is only necessary when **Dataset security** is enabled.

Datasets can only be assigned to user roles, not individual users. This means that users gain access to datasets through their roles. When the dataset security setting **Dataset security** is enabled, users with ROLE_ADMIN or ROLE_DATASET_MANAGER can configure the roles that have access to specific datasets in Data Quality & Observability Classic from the Dataset Management page.

When **Dataset security** is enabled, roles can be mapped to datasets to provide three different degrees of dataset access permissions:

- **Full permission roles**: Roles that grant users unrestricted access to the dataset to which they are assigned.
- **Read-only roles**: Roles that grant users viewer access to the areas of the application that they encompass. Users with read-only roles can also create, delete, and edit alerts, templates, and scorecards.
- **Partial roles**: Roles that allow any combination of Data Preview access, the ability to retrain datasets, and the ability to access the Rules page to write and manage rules.

For instructions on how to add roles to datasets, see Assigning roles to datasets.

What's next?

- Assign users to roles
- Add custom roles
- Manage the roles assigned to datasets
- Manage the roles assigned to connections

Adding custom roles

Administrators can create custom roles on the Role Management page.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Role Management** and then click **Roles**.
4. Near the far right corner of the page, click .
   » The **Add Role** modal appears.
5. In the **Role Name** input field, give your role a unique name.

   > **Note** Role names can only contain alphanumerical characters and underscores.

6. Click **Submit**.

> **Tip** You can delete a role by clicking the in the far right column of the row containing the role you wish to delete.

Adding users to roles

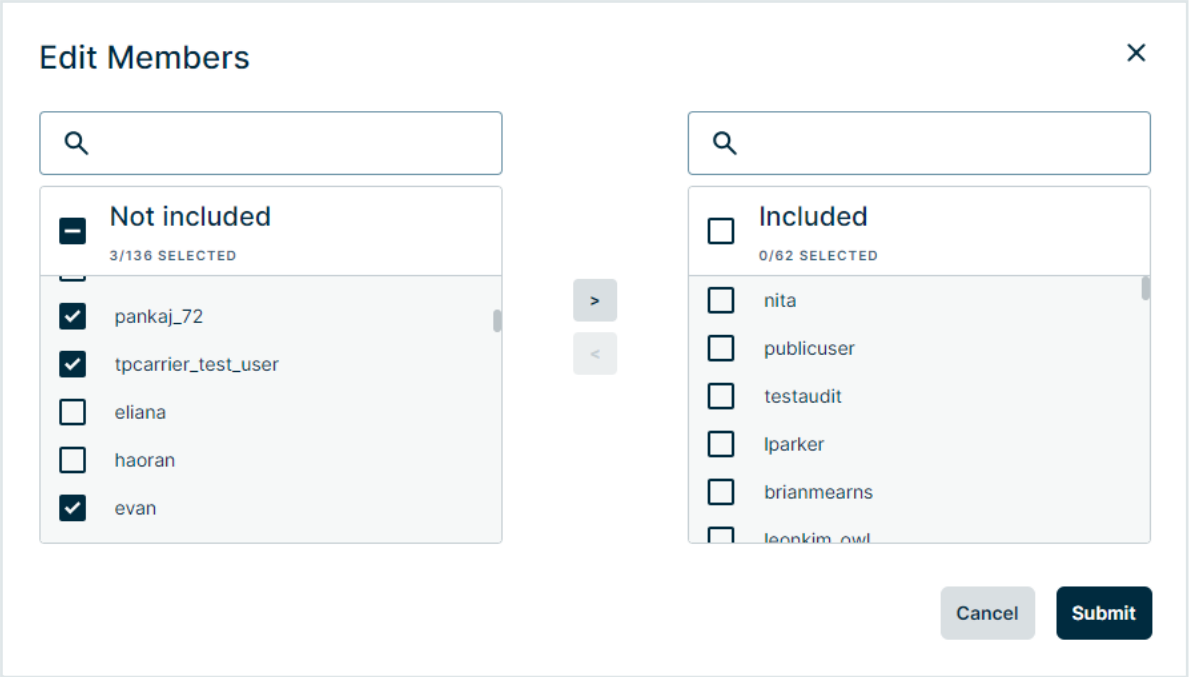
Administrators can manage the users assigned to roles on the Role Management page.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Role Management** and then click **Roles**.
4. In the far right column, click .
   » The **Edit Members** modal displays for that specific role.
5. Map users to your role by moving users from the **Not included** box to the **Included** box.

   To do this, select the user(s) you wish to map to your role, then click . You can also filter users in the search fields.

   > **Note** You can remove mapped users from the role by selecting them from the **Included** box and then clicking .



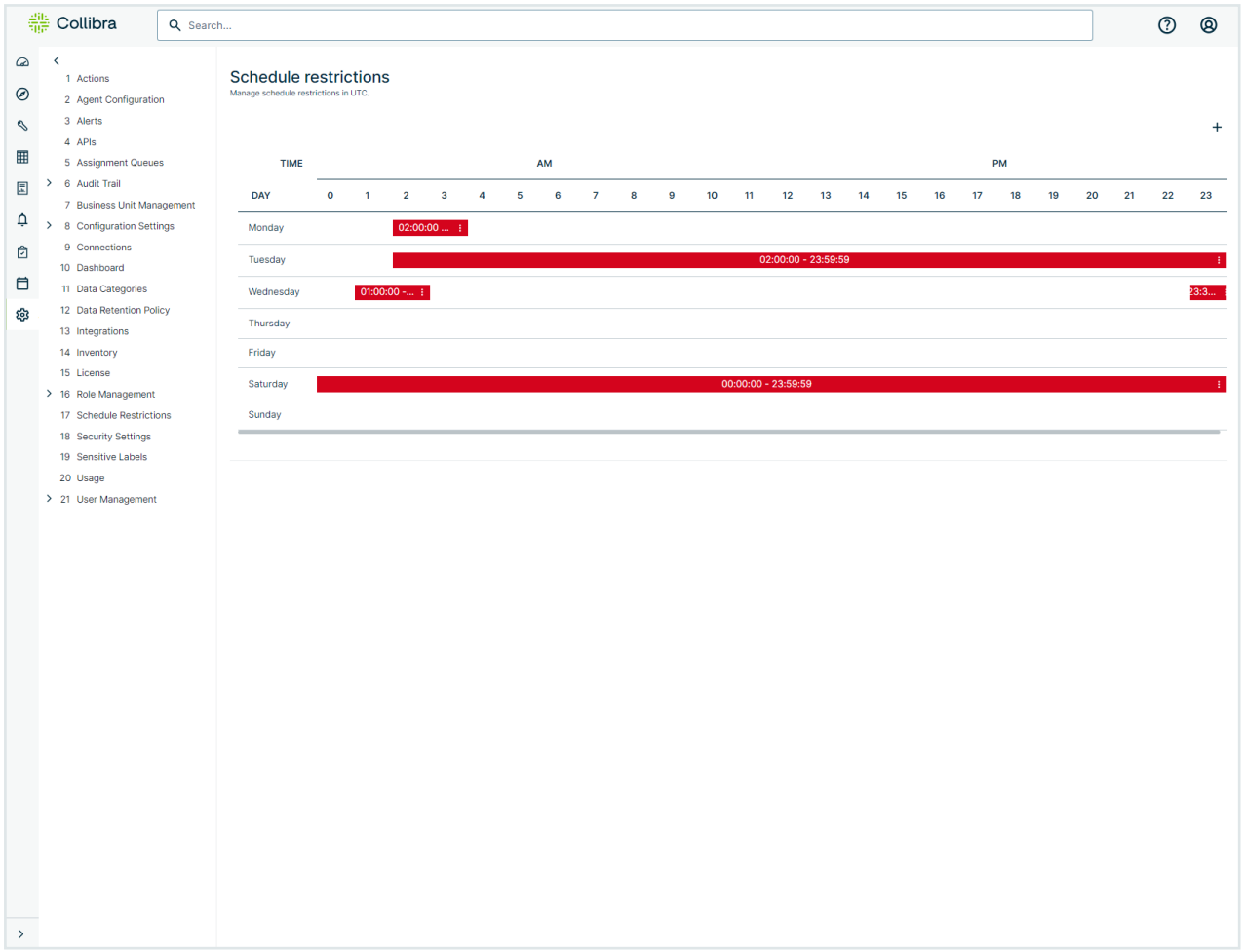
6. Click **Submit**.

Schedule Restrictions

Schedule Restrictions lets administrators prohibit jobs from running on certain days or during specific time frames. Limiting job scheduling to certain days can be especially helpful for ensuring the freshness of data and reducing operational costs when running DQ Jobs during off-peak times.

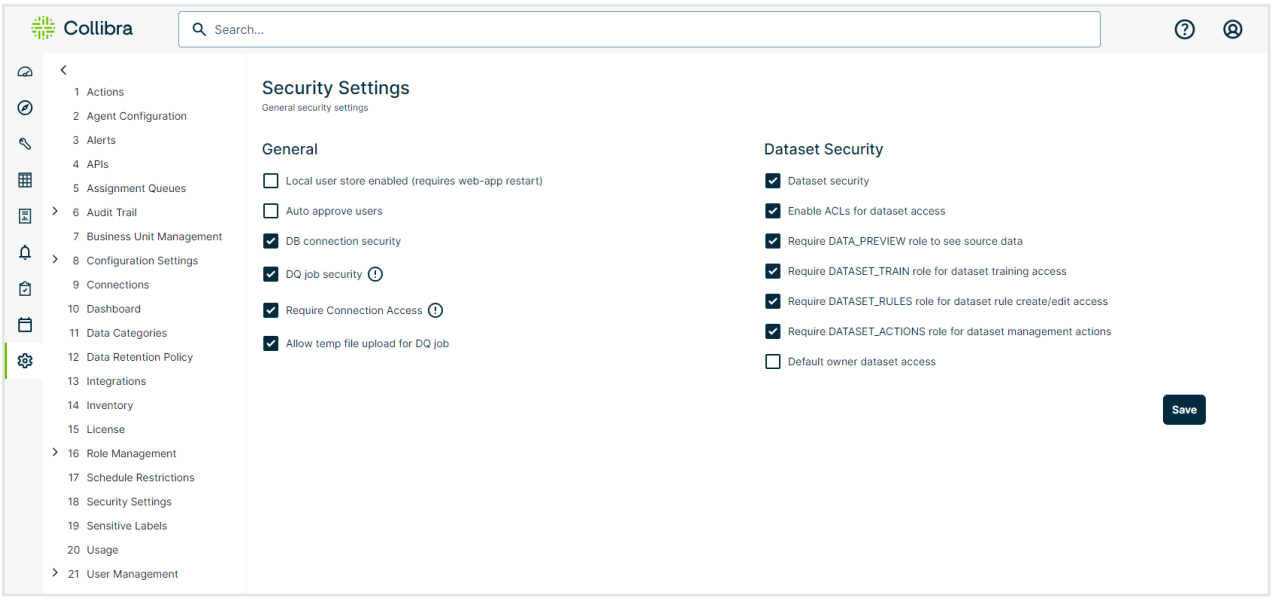
You can add a new schedule restriction by clicking the in the upper right corner of the page, then selecting options from the **Day of week**, **From**, and **To** dropdown menus. After clicking Submit, the range you specified between the To and From dropdown menus is reflected on the main Schedule Restrictions page in the day row you selected in the Day of week dropdown menu.

To edit or delete a schedule restriction, click the next to the range you intend to modify, then select either **Edit** or **Delete**.



Security Settings

Security Settings lets admin users configure general and dataset security settings in Data Quality & Observability Classic.



General

Setting	Description
Local user store enabled	Allows you to create and administer users stored in the internal Metastore. After enabling or disabling this setting, you must restart Data Quality & Observability Classic. For more information, go to Local User Store Authentication.
Auto approve users	Automatically approves local users upon sign-in. When enabled, the accounts of new users are enabled, unlocked, and have ROLE_PUBLIC automatically assigned to them. When disabled, the accounts of new users are disabled, locked, and do not have any roles assigned to them. Another user with ROLE_ADMIN, ROLE_USER_MANAGER, or ROLE_OWL_ROLE_MANAGER must grant the new user account at least one role before the new user can sign in to Data Quality & Observability Classic.
DB connection security	Allows admin users to select and map roles to connections. **Note** Any user with those mapped roles can see the connection in the Data Quality & Observability Classic application.

Setting	Description
DQ Job security	Makes ROLE_OWL_CHECK a requirement for users to run DQ Jobs. **Note** ROLE_ADMIN and ROLE_OWL_CHECK are the only roles that can run DQ Jobs when DQ Job security is enabled. Show table

DQ Job security	Dataset security	User role	DQ Job access level
Enabled	Disabled	No role	• Cannot create • Cannot run • Cannot schedule
Enabled	Enabled	A role other than ROLE_ADMIN or ROLE_OWL_CHECK assigned to the dataset	• Cannot create • Cannot run • Cannot schedule
Enabled	Enabled	ROLE_ADMIN and/or ROLE_OWL_CHECK	• Can create • Can run • Can schedule

Require Connection Access	When DQ Job security and Require Connection Access are enabled, ROLE_ADMIN or ROLE_OWL_CHECK must be assigned to both the user and connection for the user to run, schedule, and create DQ Jobs on the connection.
Allow temp file upload for DQ Job	Allows users to upload temporary files from their local machines to use as data sources in Explorer.

Setting	Description
Set token expiration duration	Sets the expiration duration of the authentication token for DQ client API calls. If an API call is made and the token has reached the time limit, the API response is 401. The default setting is one hour. To change the duration, modify the following command in `owl-env.sh`: `export JWT_TOKEN_EXPIRY_TIME=3600000` The time is specified in milliseconds (ms). In this example, 3600000 milliseconds=60 minutes.

Dataset Security

Setting	Description
Dataset security	When enabled, if a user does not have a full permission role granting them the ability to perform specific tasks, they cannot perform them. The following list shows specific tasks that dataset security can prevent users without the correct full permission role from performing: • Run DQ Jobs • Manage rules • View DQ Job findings • Preview data • Take action on DQ Job findings • Train datasets • Mask columns • View assignments • Assign business units and data categories **Note** In the list above, a single role encompassing all tasks does not exist. Instead, multiple roles must be mapped to relevant datasets as full permission roles to allow users to perform all listed tasks. Users with ROLE_ADMIN or ROLE_DATASET_MANAGER can map roles to datasets as full permission roles on the Dataset Management page.

Setting	Description
Enable ACLs for dataset access	When access control list (ACL) is enabled, administrators can configure the following options to limit usage/permissions on datasets: • DATA_PREVIEW • DATASET_TRAIN • DATASET_RULES • DATASET_ACTIONS
Require DATA_PREVIEW role to see source data	When enabled, only users with ROLE_DATA_PREVIEW can preview source data on the Explorer, Findings, Profile, Rules, and Dataset Overview pages.
Require DATASET_TRAIN role for dataset training access	When enabled, only users with ROLE_DATASET_TRAIN can train datasets, which includes the ability to adjust the scoring model, modify thresholds of Adaptive Rules, and validate, invalidate, or resolve data quality findings.
Require DATASET_RULES role for dataset rule create/edit access	When enabled, only users with ROLE_DATASET_RULES can manage rules. **Note** ROLE_DATA_PREVIEW is also required to manage rules.
Require DATASET_ACTIONS role for dataset management actions	When enabled, users with ROLE_DATASET_ACTIONS are granted permissions to edit, rename, publish, assign data categories and business units, enable integrations, and schedule DQ Jobs from the Dataset Manager. These permissions are also accessible by users with the ROLE_ADMIN and ROLE_DATASET_MANAGER roles.
Default owner dataset access	An access control that, when enabled, restricts the ability to access datasets to users who are not the creator or owner of them.

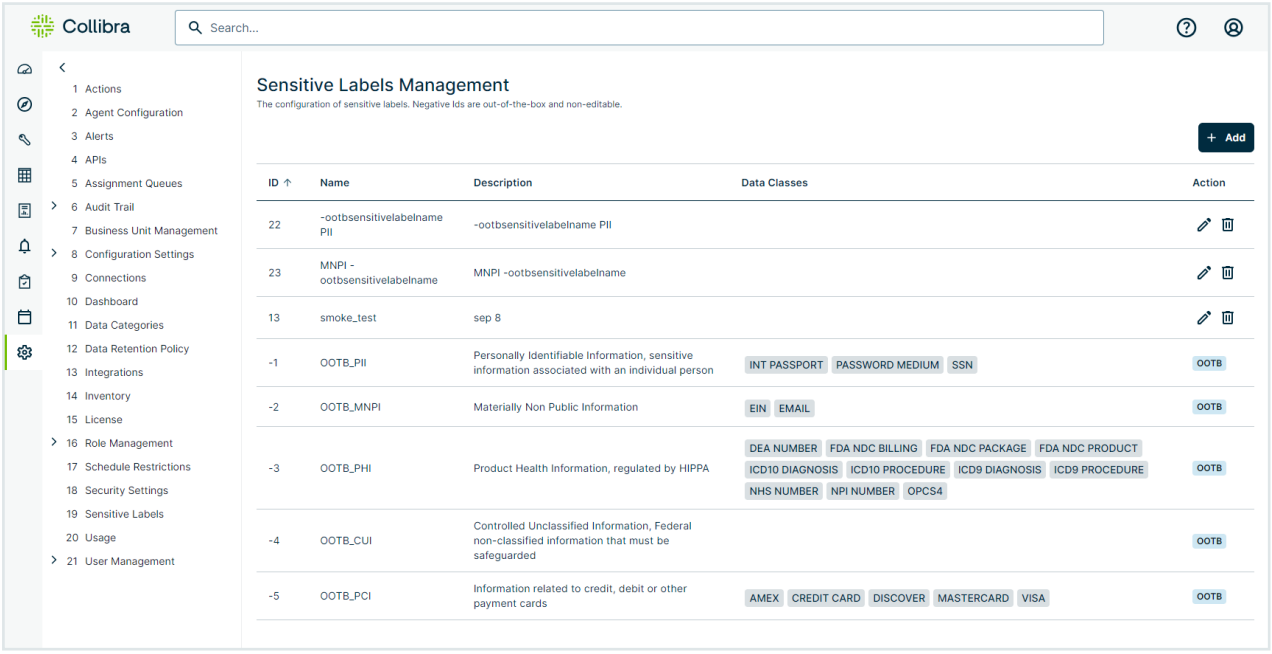
Sensitive Labels

Sensitive Labels Management lets administrators add, edit, and delete sensitive labels in Data Quality & Observability Classic.

While there are five out-of-the-box (OOTB) sensitive labels, you can create additional custom sensitive labels that meet your needs. Sensitive labels appear on dataset Profile (in the

Sensitive column) and Findings (in Data Preview), and can help identify data marked as sensitive.

When a DQ Job runs, all additional checks associated with the data classes included in the sensitive label are also performed. For example, the OOTB sensitive label, OOTB_MNPI, includes the EIN and EMAIL data classes. Therefore, when a DQ Job with the OOTB_MNPI sensitive label runs, it includes the RegEx `^[1-9]\\d?-\\d{7}$` (EIN) and SQL `'%_@__%._ _%'` (EMAIL) in the data quality check.



Column	Description
ID	An automatically generated system identifier. When you create a new sensitive label, its label ID matches the sequence in which it was created, irrespective of whether or not the sensitive label with the ID that precedes it still exists. For example, if you create a sensitive label, and the label ID assigned to it is 25, this is because it is the 25th sensitive label created in your environment, even if the 24th has been deleted. Negative (-) label IDs denote that the sensitive label is OOTB. These cannot be edited or deleted.

Column	Description
Name	The name of the sensitive label. Out-of-the-box sensitive labels begin with OOTB_. There are five OOTB sensitive labels, including: • OOTB_PII • OOTB_MNPI • OOTB_PHI • OOTB_CUI • OOTB_PCI
Description	A meaningful description of the sensitive label.
Data Classes	The data classes associated with a sensitive label. You can only assign a data class to a single sensitive label. **Warning** If you assign a data class to a custom sensitive label when that data class was previously assigned to an OOTB sensitive label, you cannot reassign the data class back to its original OOTB sensitive label.
Action	Allows you to edit an existing sensitive label. Only user-created sensitive labels can be edited. Allows you to delete an existing sensitive label. Only user-created sensitive labels can be deleted. OOTB Indicates that a sensitive label is out-of-the-box. OOTB sensitive labels cannot be editor or deleted.

Adding sensitive labels

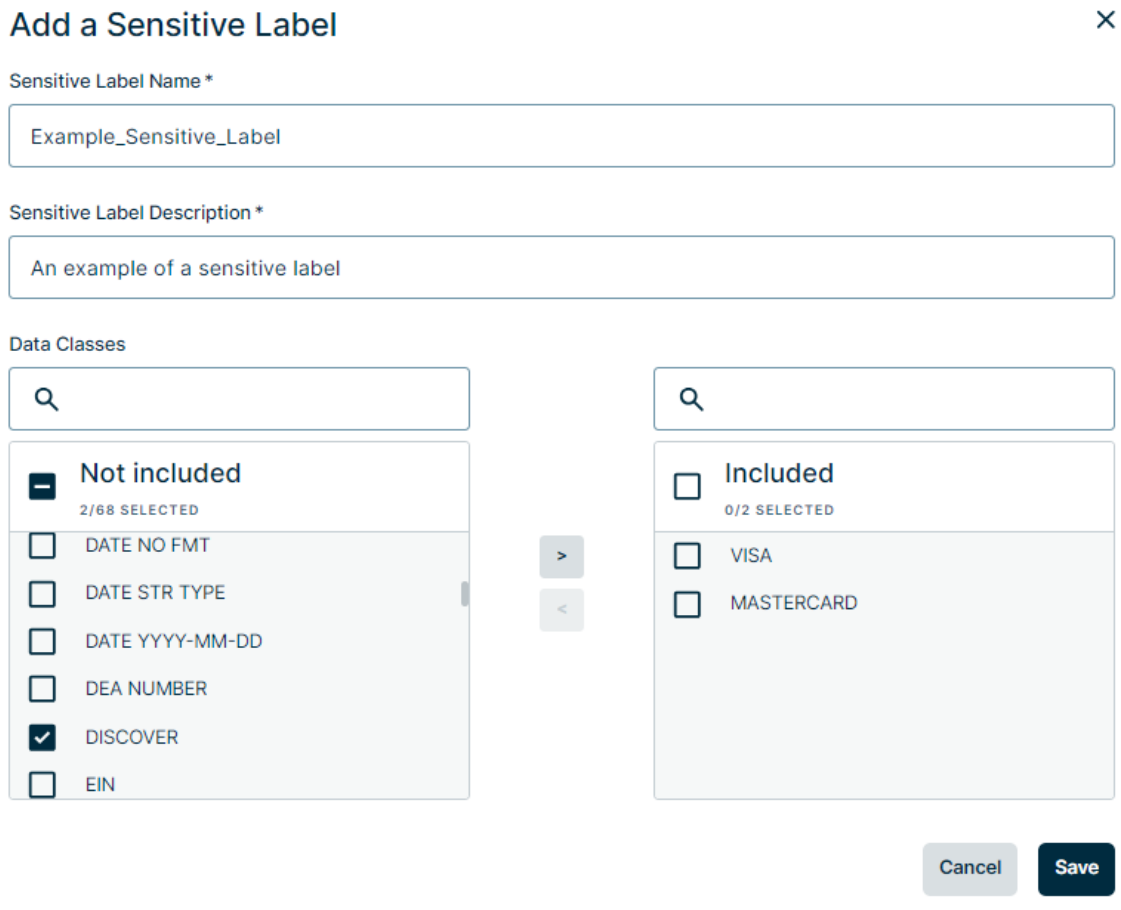
Delete this text and replace it with your own content.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Sensitive Labels**.
   » The Sensitive Label Management page opens.

4. On the upper-right side of the page, click **+ Add**.
   » The **Add a Sensitive Label** modal appears.
5. In the **Sensitive Label Name** input field, give the label an easily identifiable name.
6. In the **Sensitive Label Description** input field, give the label a meaningful description.
7. Select the Data Class(es) in the **Not included** box, then click to move your selection(s) to the **Included** box to include Data Classes in your sensitive label. You can also filter the roles in the search fields.

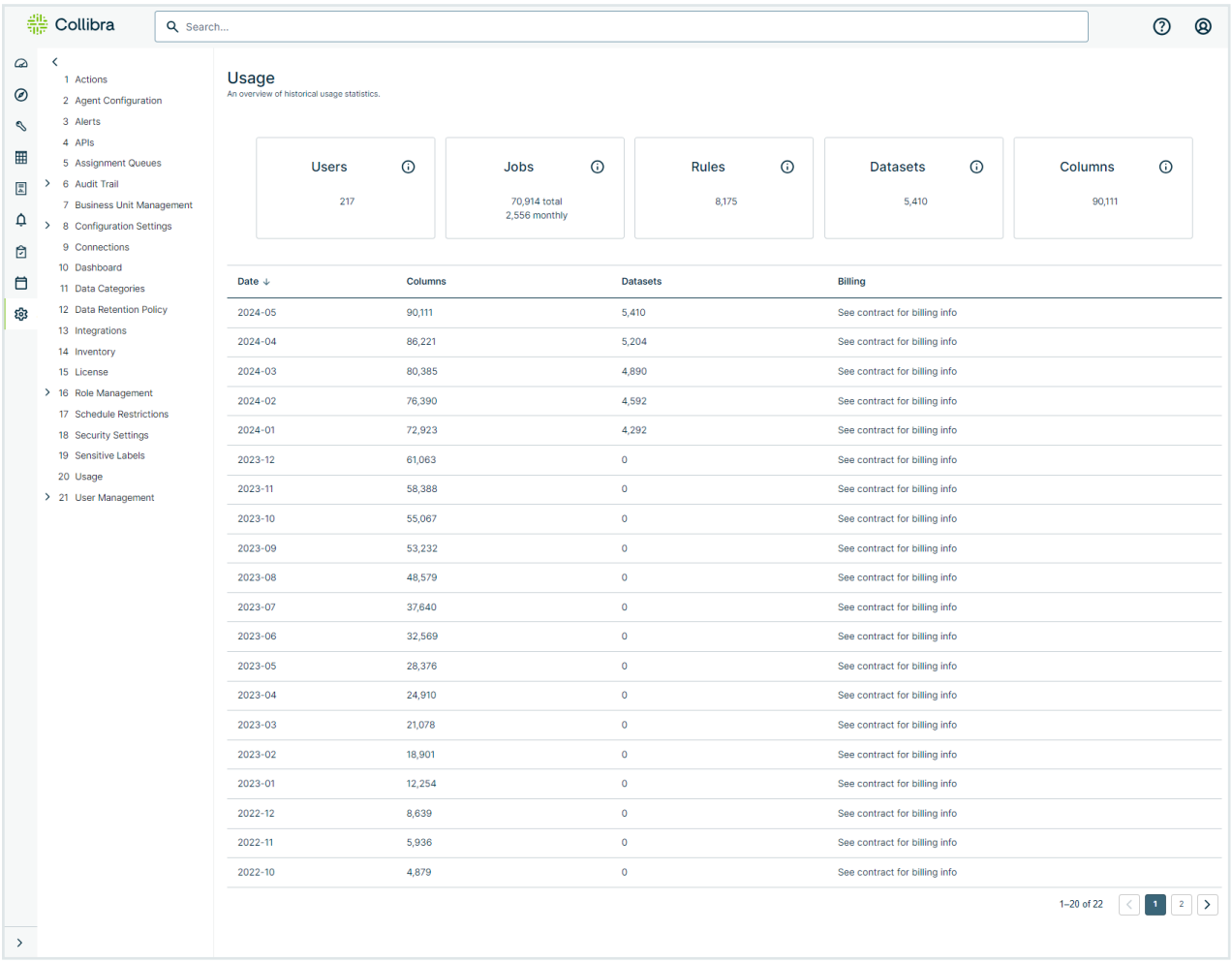
   > **Note** You can remove Data Classes from the sensitive label by selecting the Data Class from the **Included** box and then clicking  .



8. Click **Save**.

Usage

Usage is a reference page for administrators to view historical usage statistics in their organization's Data Quality & Observability Classic environment.



The tiles across the top of the page break down usage statistics by users, jobs, rules, datasets, and columns for the most recent month. The table below the tiles shows the number of datasets and columns scanned for quality in your organization's environment for the month reflected in the Date column.

> **Note** For more details about billing information, see your Data Quality & Observability Classic contract.

The following table shows the descriptions and calculations of the data points included in the tiles at the top of the page.

Data point	Description	Calculation
Users	The total number of users in your Data Quality & Observability Classic environment, including both active and inactive users.	`count (*) from user_profile`
Jobs	All DQ Jobs that ran in your Data Quality & Observability Classic environment over the most recent month.	`max (id) from owlcheck_q` `count (*) from owlcheck_q where jobs between 01 and 30`
Rules	The total number of rules applied to datasets in your Data Quality & Observability Classic environment.	`count (*) from owl_rule where is_active = 1`
Datasets	The total number of unique DQ Jobs on tables and files in your Data Quality & Observability Classic environment over the most recent month.	`count (distinct (datasets)) from dataset_schema`
Columns	The total number of unique columns under quality control in your Data Quality & Observability Classic environment over the most recent month.	`count (distinct (dataset, columns)) from dataset_schema`

User Management

AD Security

AD and LDAP Security Settings Overview

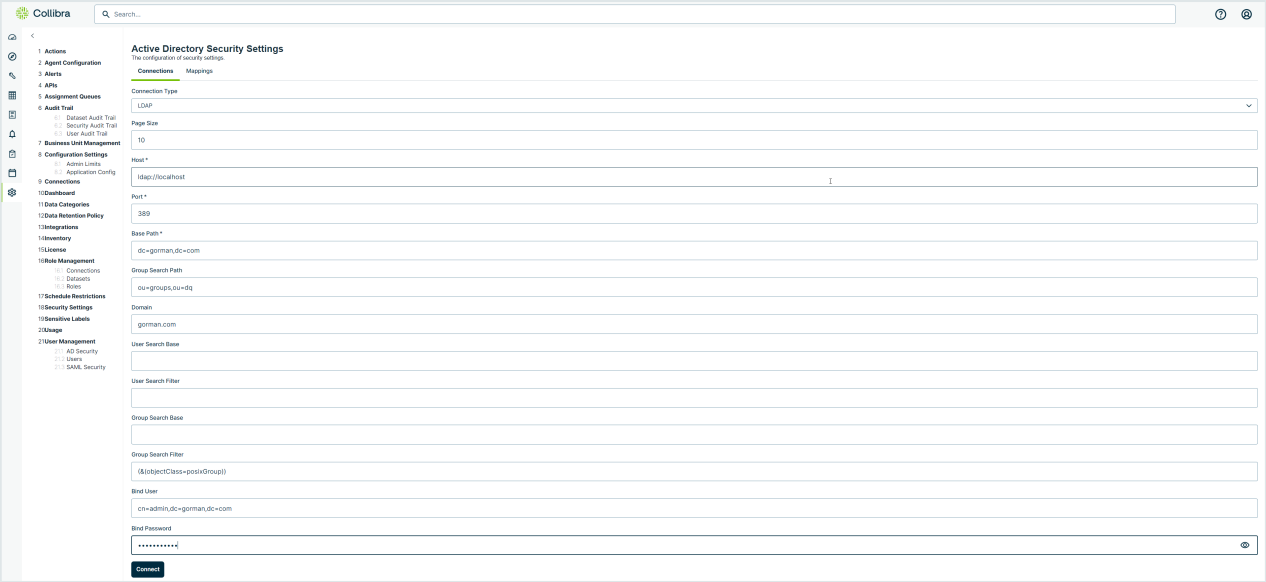
You can configure security settings from User Management in the left navigation panel.

Prerequisites

- You have admin permissions (ROLE_ADMIN) assigned to your User Profile.
- You have the ability to look at (or get information from) attributes on User and Group LDAP objects. This can be done through CLI, using ldapsearch or through an LDAP object browser.

Configure Security Settings for Active Directory and LDAP

This section describes how to configure security settings for the Active Directory and LDAP connection types.



> **Note** You must add specific script settings in owl-env.sh or ConfigMap when configuring LDAP and Active Directory. For more information, go to Configuring LDAP settings.
> For information about mapping an AD Group to a Collibra DQ role, go to AD Group to Role Mapping Overview.

For information about configuration settings for Active Directory and LDAP, select the applicable tab below.

Configure Active Directory Settings

Configure security settings for Active Directory by navigating to User Management, in the left navigation panel, and selecting **AD Security**.

The following table shows the available input fields for the Active Directory connection type.

> **Note** An asterisk (*) next to a setting represents a required input field.

Setting	Description
Connection Type	Select **Active Directory** from this dropdown.
Page Size	Set a value greater than 0 to control query page size.
Host*	Hostname or URL of your LDAP or LDAPS server, for example, ldap://12.345.678.90.
Port*	Port to connect to your LDAP or LDAPS server. > **Note** The default ports are **389** for LDAP and **636** for LDAPS.
Base Path*	Base path configured for LDAP. The value entered is the base domain information and the format is usually **DC=xxxx,DC=com**.

Setting	Description
Group Search Path	Organizational Units where the Groups are located. It is based on the LDAP setup. The value entered is the domain object path where the groups are located; for example, **OU=OwlGroups,OU=Groups** (this value is combined with the Base Path to look for groups). 

Tip You can use a wildcard in this field to limit the search results. For example, to see only the groups where the common name starts with "Collibra", enter **cn=Collibra*** in the Group Search Path field. You will now see only groups that start with the name "Collibra" (CollibraPublic, CollibraAdmin, etc.).

Setting	Description
	Note After Group Search Path is configured, it is recommended that you restart the Collibra DQ web application.
Domain	The domain name used to signify when non-local users log in. **Note** This field applies only to Active Directory configurations.
Bind User	DN of an admin user that is used for authentication; for example, *admin@collibra.com*.
Bind Password	Password of an admin user for the bind account.

Configure LDAP Settings

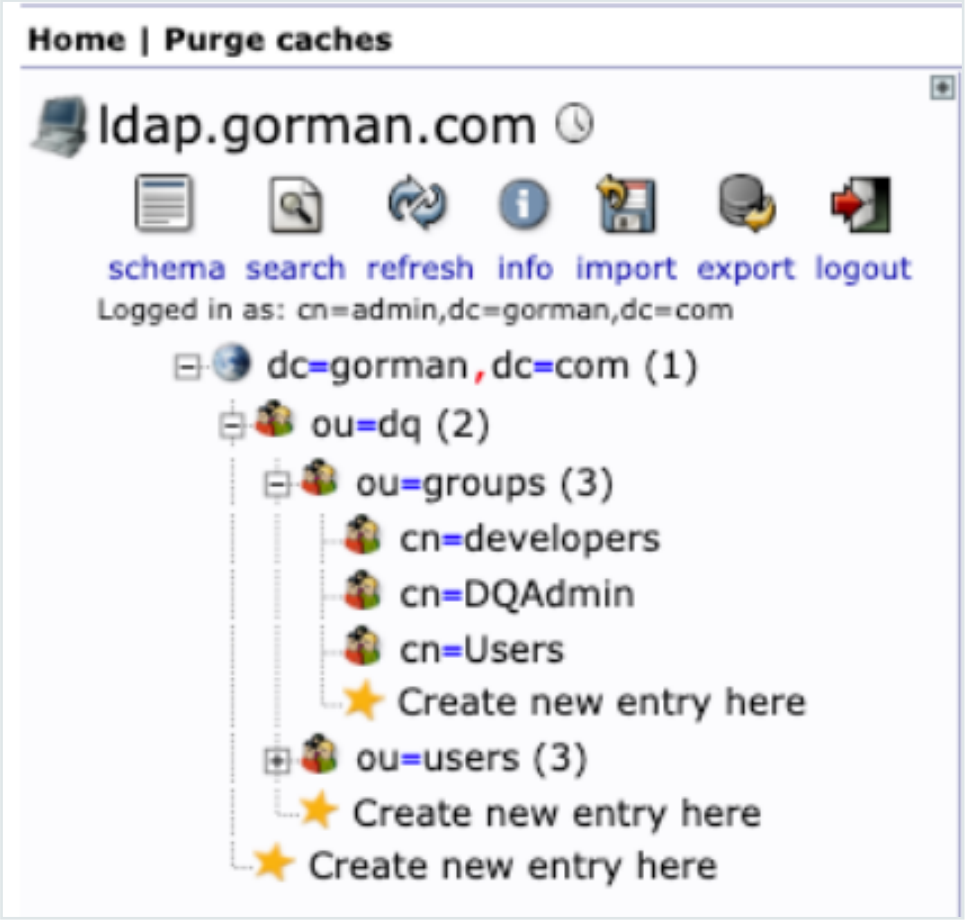
Configure LDAP security settings by navigating to User Management, in the left navigation panel, and selecting **AD Security**.

The following table shows the available input fields for the LDAP connection type.

Note An asterisk (*) next to a setting represents a required input field. A restart of the Collibra DQ web application may be needed for values to take effect.

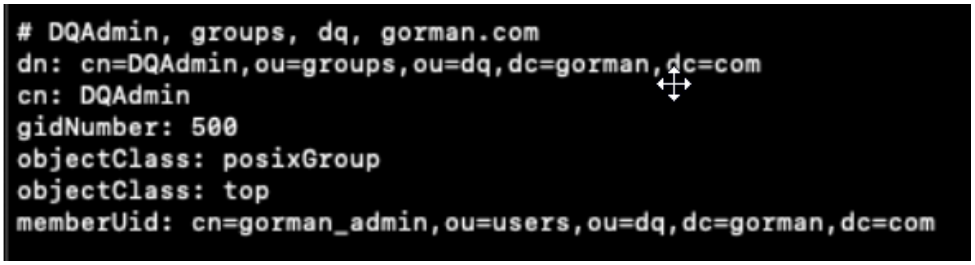
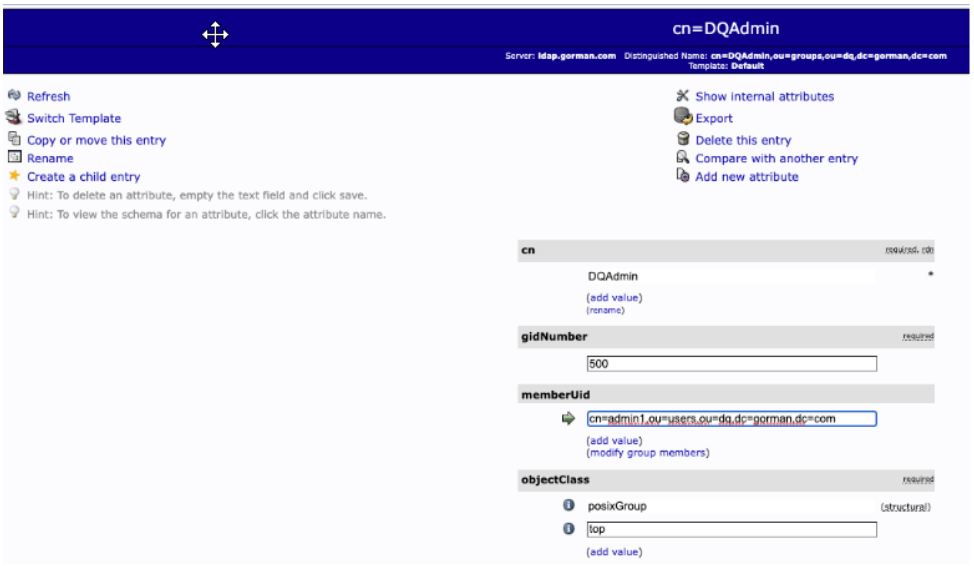
Setting	Description
Connection Type	Select **LDAP** from this dropdown.
Page Size	Set a value greater than 0 to control query page size.

Setting	Description
Host*	Hostname or URL of your LDAP or LDAPS server, for example, ldap://12.345.678.90.
Port*	Port to connect to your LDAP or LDAPS server. **Note** The default ports are **389** for LDAP and **636** for LDAPS.
Base Path*	Base path configured for LDAP. The value entered is the base domain information and the format is usually **DC=xxxx,DC=com**.

Setting	Description
Group Search Path	Organizational Units where the Groups are located. It is based on the LDAP setup. The value entered is the domain object path where the groups are located; for example, **OU=OwlGroups,OU=Groups** (this value is combined with the Base Path to look for groups). 

Tip You can use a wildcard in this field to limit the search results. For example, to see only the groups where the common name starts with "Collibra", enter **cn=Collibra*** in the Group Search Path field. You will now see only groups that start with the name "Collibra" (CollibraPublic, CollibraAdmin, etc.).

Setting	Description
	Note After Group Search Path is configured, it is recommended that you restart the Collibra DQ web application.
Domain	The domain name used to signify when non-local users log in. **Note** This field applies only to Active Directory configurations.
User Search Base	Base DN where the LDAP users for Collibra DQ are located; for example, **CN=Users**.
User Search Filter	When set, this LDAP filter is used to locate users at login. This filter is based on your LDAP configurations.
Group Search Base	The base DN where all the groups are located. **Note** This field applies only to LDAP configurations.

Setting	Description
Group Search Filter	LDAP filter used to narrow down group objects located under a base DN. Default: **(&(objectclass=group))** **Note** This field applies only to LDAP configurations.  
Bind User	DN of an admin user that is used for authentication; for example, *admin@collibra.com*.
Bind Password	Password of an admin user for the bind account.

Configuring LDAP settings

When configuring LDAP, you must add the settings on this page in the **owl-env.sh** (Standalone) or **Web ConfigMap** (Cloud Native).

This configuration occurs upon start-up of the Collibra DQ web application. See Standalone Install for a complete list of the **owl-env.sh** scripts.

> **Note** These settings are not required for Active Directory because the default properties specific to the AD library load upon the start-up of the Collibra DQ WebApp.

Required script settings to configure LDAP

The following table provides descriptions of the **owl-env.sh** (Standalone) or **Web ConfigMap** (Cloud Native) settings to add when you configure LDAP.

> **Note** You can override the default script settings, however, it is dependent on your LDAP provider and how LDAP is implemented.

owl-env.sh setting	Description
LDAP_GROUP_RESULT_DN_ATTRIBUTE	The fully qualified name of the group with the base path; for example, **CN=OwlAppAdmin,OU=OwlGroups,OU=Groups,DC=owl,DC=com**. Default: **distinguishedname**.
LDAP_GROUP_RESULT_NAME_ATTRIBUTE	The common name of the group. Default: **CN**.

owl-env.sh setting	Description
LDAP_GROUP_RESULT_CONTAINER_BASE	This value is used only when the LDAP Group Object has no attributes for the previous properties. In this case, this would get set to the base path and organizational units, where the group is located. Default: **None**. **Note** If the LDAP_GROUP_RESULT_DN_ATTRIBUTE is not found, the LDAP_GROUP_RESULT_NAME_ATTRIBUTE is combined with the LDAP_GROUP_RESULT_CONTAINER_BASE, to save the fully qualified group name.

Script Settings for Search Configuration

The following table provides descriptions of the **owl-env.sh** (Standalone) or **Web ConfigMap** (Cloud Native) settings to add when you configure the search filter.

owl-env.sh setting	Description
LDAP_ROLES_PROP_NAME	Attribute of the group object to find the name. This is usually the common name of the group and is used to map the LDAP group to the internal role. Default: **memberOf**.
LDAP_USER_SEARCH_ATTRIBUTE	Property used as part of a filter to look for user membership in a group. The Group Object in LDAP has an attribute where users are listed. This value must be set for this property. Default: **distinguishedname**.

AD Group to Role Mapping

AD Group to Role Mapping Overview

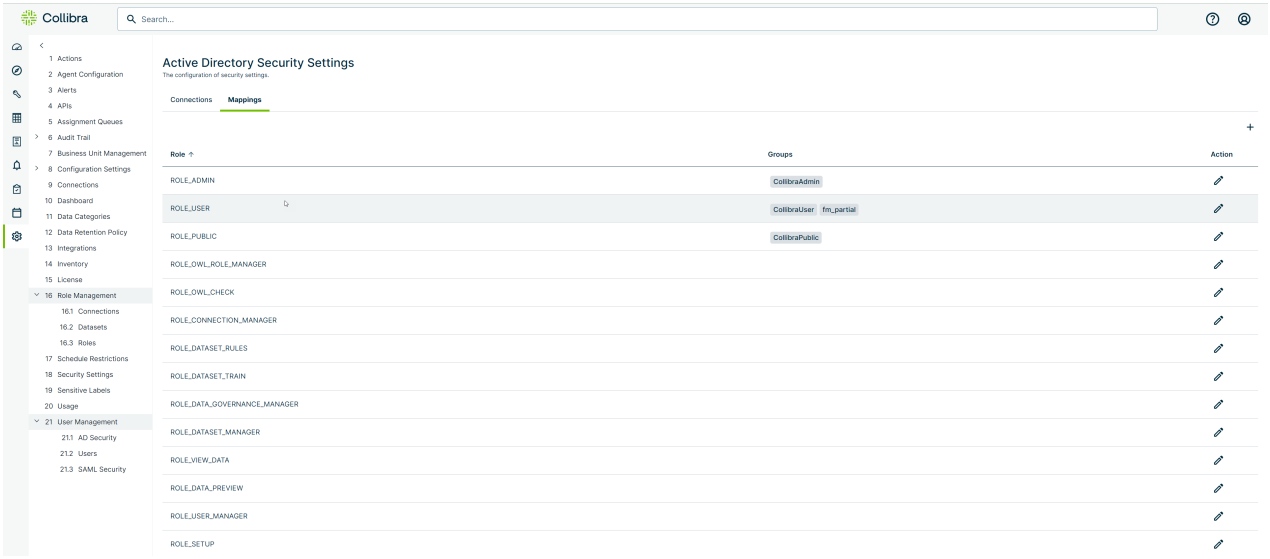
When you map an Active Directory (AD) security group to a Data Quality & Observability Classic role, you grant role-based access to all users from the selected AD Group to the selected Data Quality & Observability Classic role. You can find additional information on creating custom application roles on the Role-Based Access Control page.

Application properties set in the **owl-env.sh** file or **Web ConfigMap** can be configured to determine which LDAP properties correspond to LDAP query results. Go to Configurations in owl-env.sh or Web ConfigMap for more information about the LDAP properties.

When group mapping, the full path (unique) and display name are required.

Example

```
LDAP_GROUP_RESULT_DN_ATTRIBUTE=distinguishedname
LDAP_GROUP_RESULT_NAME_ATTRIBUTE=CN
```



What's next?

- Map AD groups to Data Quality & Observability Classic roles
- Map security groups to Data Quality & Observability Classic roles when using SSO with SAML

Mapping AD security groups to roles

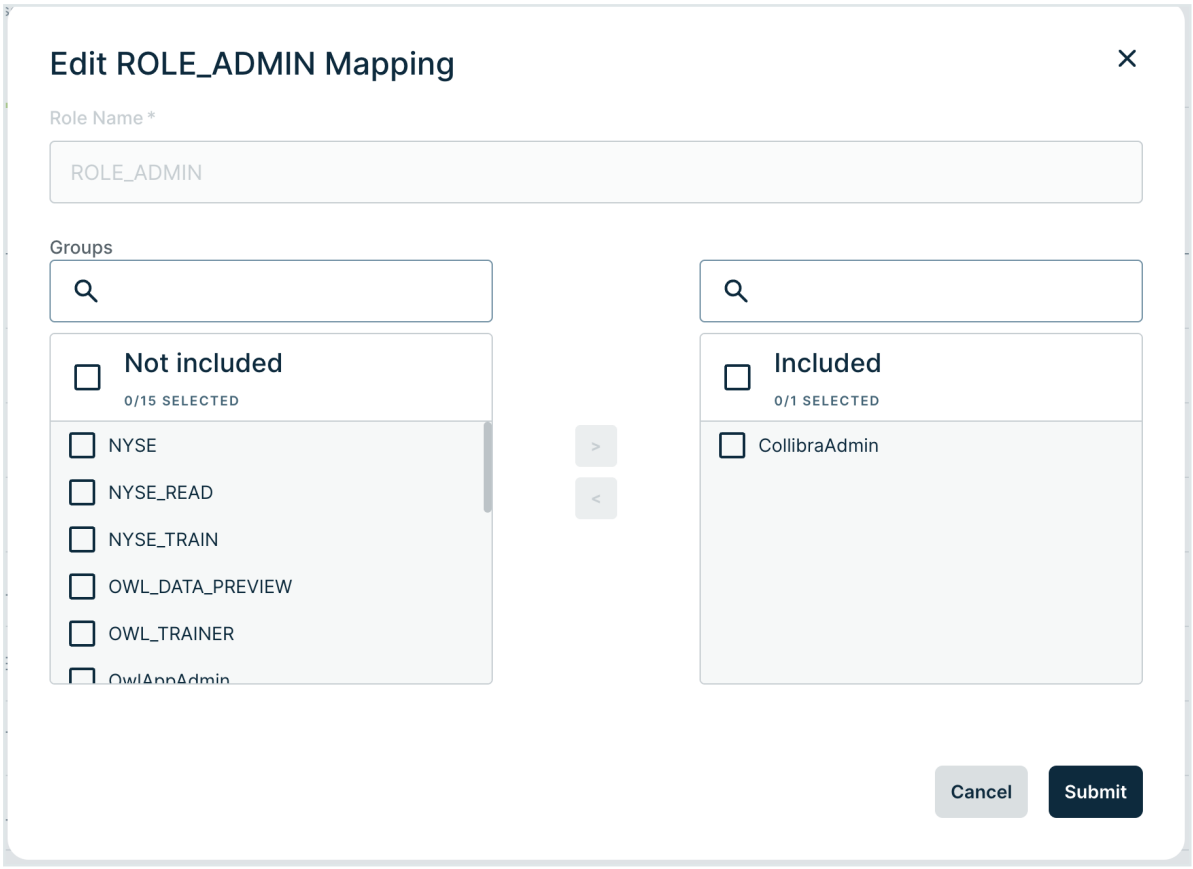
This section shows you how to map Active Directory (AD) security groups to Data Quality & Observability Classic user roles.

When Active Directory or LDAP is configured, the available groups are based on the settings under the Connections tab of the AD Security page.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **User Management** and then click **AD Security**.
   » The AD and LDAP Security Settings page opens.
4. Click the **Connections** tab and ensure that Active Directory is selected as the Connection Type.
5. Click the **Mappings** tab to display the table of roles and AD groups mapped to them.
6. To map a group to an existing role, find the row containing the role to which you want to map one or multiple AD groups, then click the ✎ in the **Action** column.
   » The edit role mapping dialog appears, where you can map AD groups to the selected role.
7. Map AD groups to the role by moving them from the **Not included** box to the **Included** box. To do this, select the AD group(s) you wish to map to the role, then click the right arrow. You can also filter the AD group in the search fields.

   > **Note** To remove a AD group to role mapping, click the left arrow to move the role (s) you select from the **Included** column back to the **Not included** column.

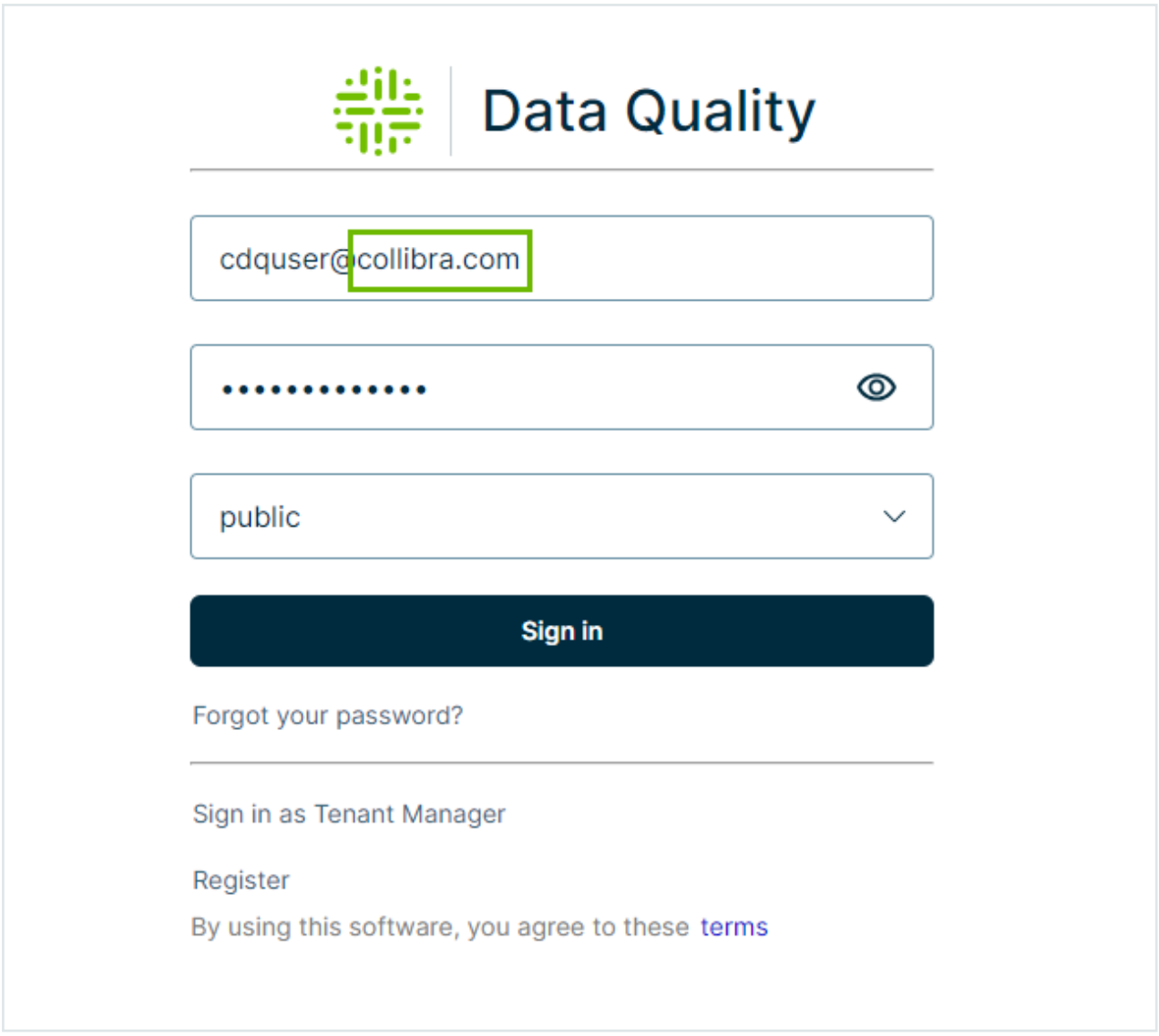


8. Click **Submit**.

After successfully mapping an AD group to a role, sign out of Collibra DQ and sign back in as a Domain user.

> **Note** You must restart the Collibra DQ web application by running `./owlmanage.sh restart_owlweb` for Standalone or restart the Web Pod for Cloud Native.

When signing into Data Quality & Observability Classic, append the domain to the end of the username.


Data Quality
cdquser@collibra.com
public
Sign in
Forgot your password?
Sign in as Tenant Manager
Register
By using this software, you agree to these terms

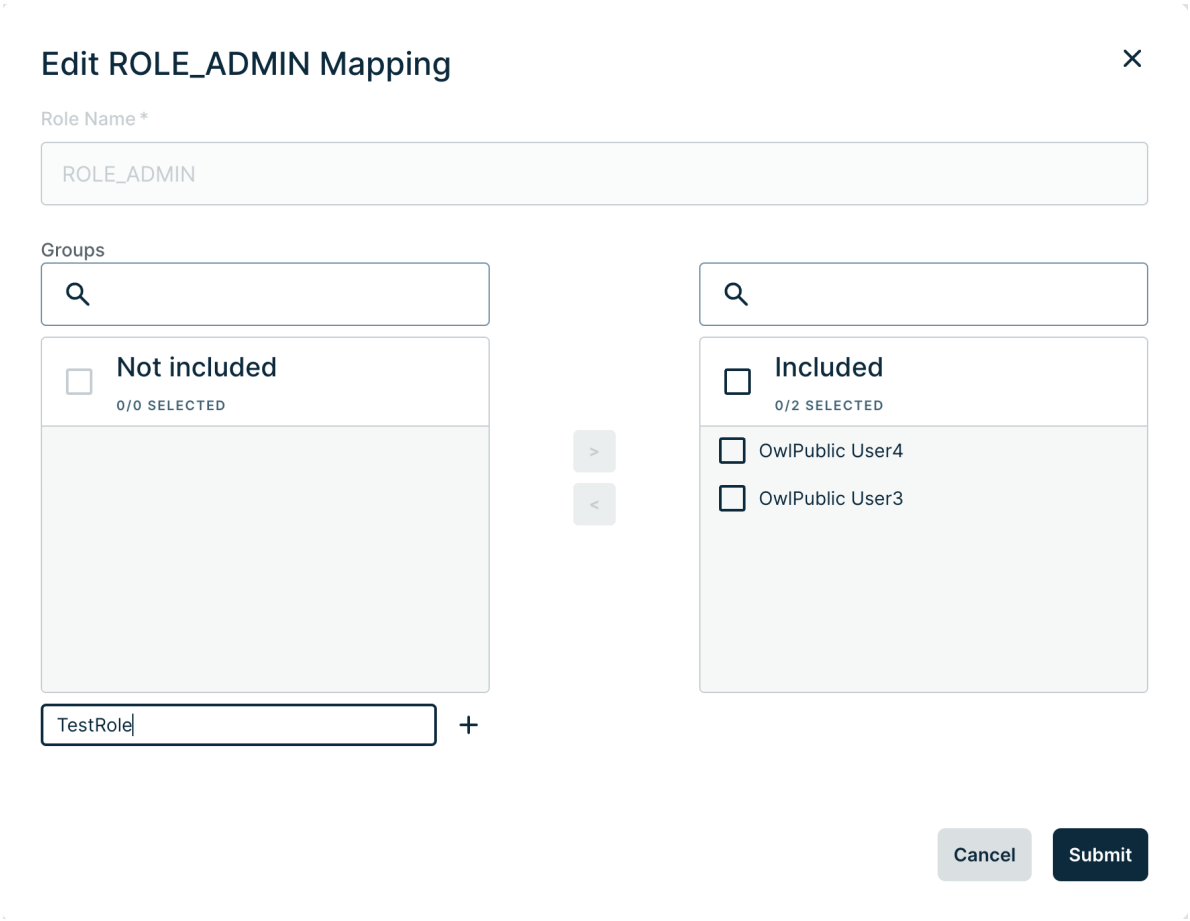
Mapping security groups to roles when using SSO with SAML

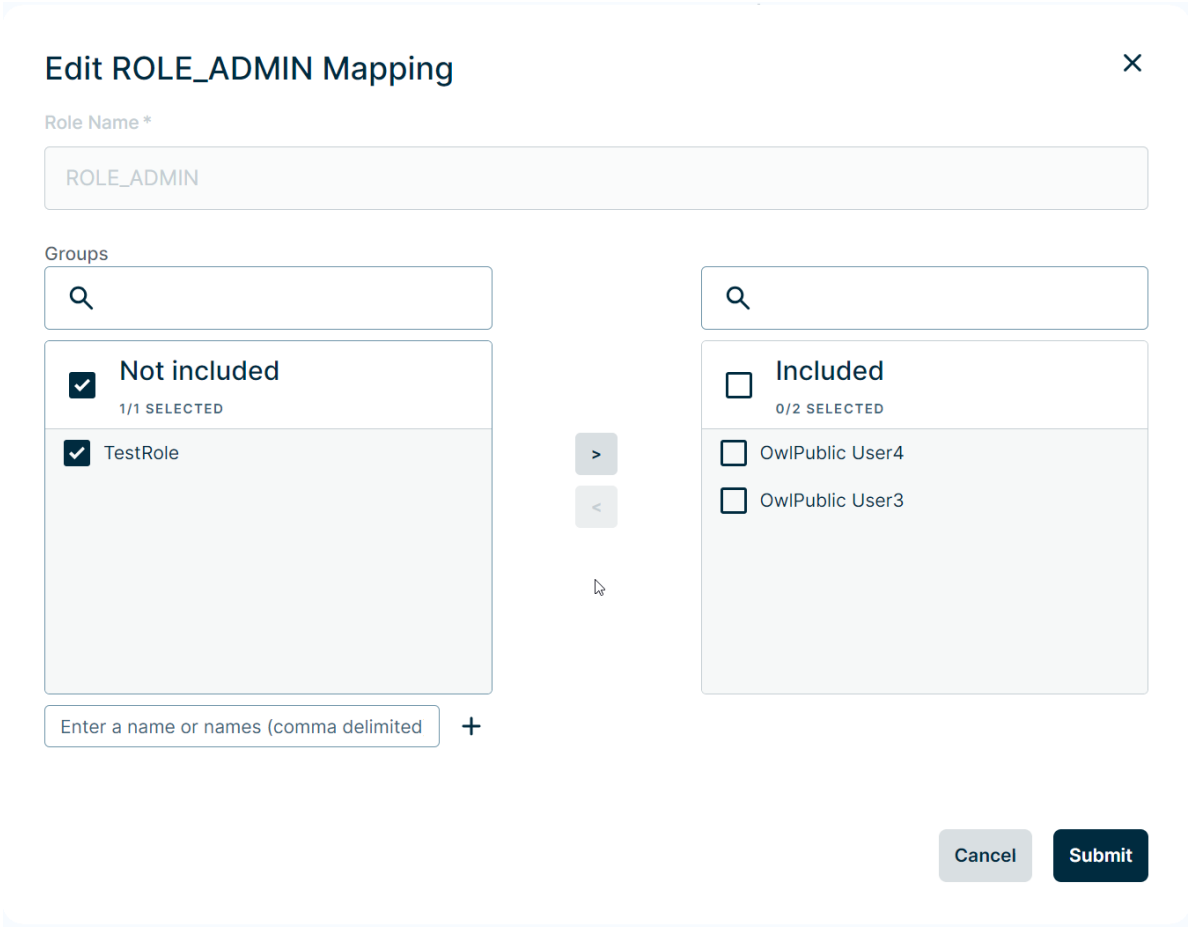
This section shows you how to map security groups to Data Quality & Observability Classic user roles when using SSO with SAML.

> **Note** To enable SAML, go to SAML Authentication.

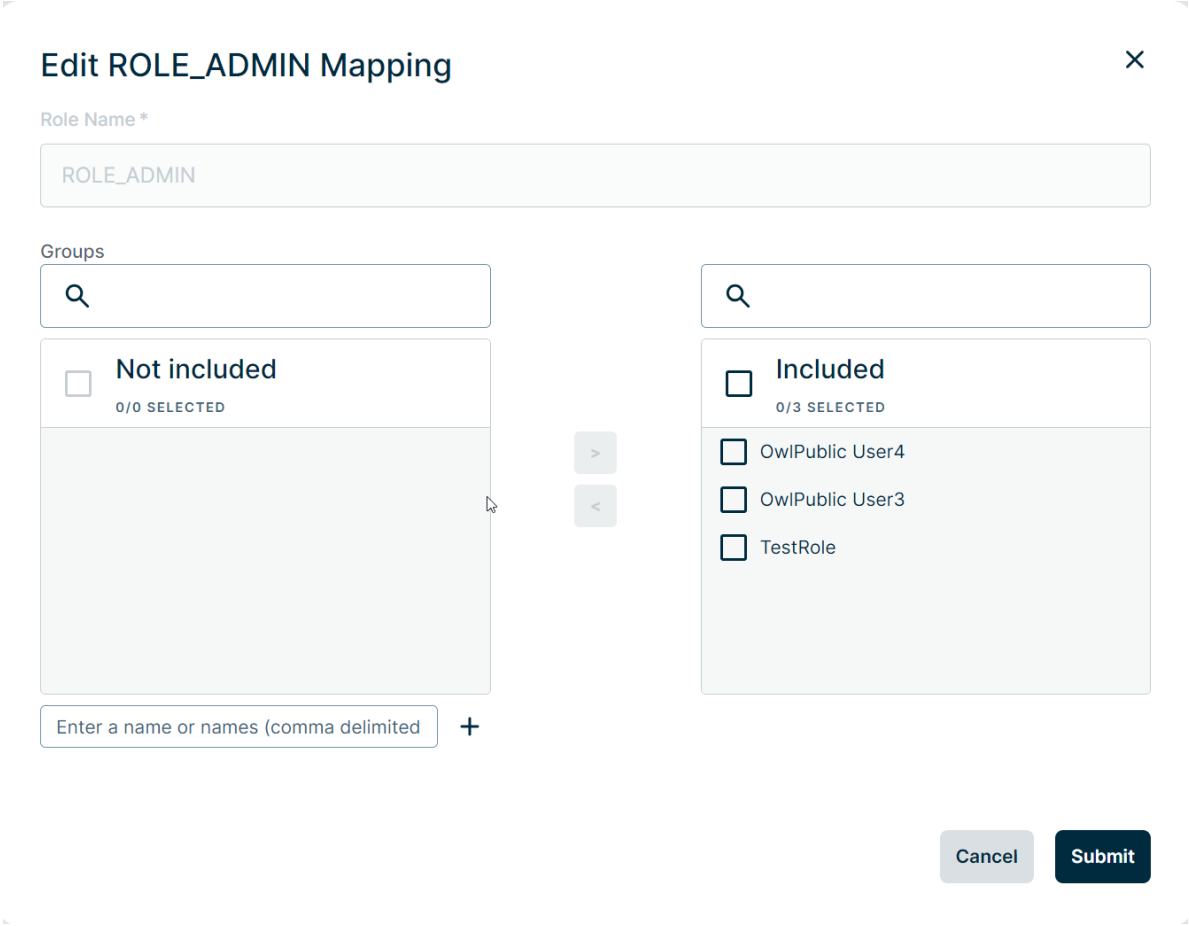
Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Select **Admin Console**.
3. From User Management in the left navigation panel, select **AD Security**.
4. Select the **Mappings** tab.
5. To map a group to an existing role, click the  icon for that role, in the **Action** column. The edit role mapping window displays for that specific role. In this window, you can map groups to Collibra DQ roles by moving the groups from the **Not included** box to the **Included** box. To do this, click the left and right arrows. You can also filter the groups and roles in the search fields.
6. Enter the name of a group in the input field below the **Not included** box and click the + icon.

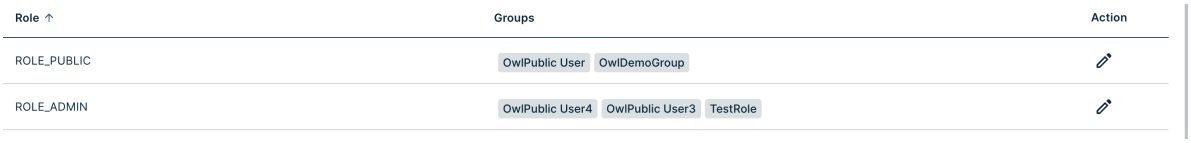

Edit ROLE_ADMIN Mapping
Role Name *
ROLE_ADMIN
Groups
Not included
0/0 SELECTED
Included
0/2 SELECTED
OwlPublic User4
OwlPublic User3
TestRole
Cancel
Submit


Edit ROLE_ADMIN Mapping
Role Name *
ROLE_ADMIN
Groups
Not included
1/1 SELECTED
TestRole
Included
0/2 SELECTED
OwlPublic User4
OwlPublic User3
>
<
Enter a name or names (comma delimited
Cancel
Submit

7. You can now click the right arrow to map this group to the role.



8. When you have finished adding your group(s), click **Submit**.
   Your new group(s) now appears in the Groups column for the selected role and users who are members of the group(s) should now have the permissions for the role.



Configuring user search for LDAP

This section describes how to set up user search for LDAP.

Configuring user search filter

To configure the user search filter for LDAP, you must configure the **User Search Filter** and **Group Search Base** properties on the Active Directory Security Settings page. Go to Configuring Security Settings (AD/LDAP) for details about the security settings for LDAP and Active Directory.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Select **Admin Console**.
3. From User Management in the left navigation panel, select **AD Security**.
4. Select the **Connections** tab.
5. From the Connection Type dropdown menu, select **LDAP**.
6. In the **User Search Filter** input field, enter the username that was used at login. This will query the LDAP provider.

   > **Note** The filter should include an objectClass and an additional attribute to search for. It should also include a search term, such as **uid={0}**. Go to Configuring Security Settings (AD/LDAP).

7. In the **Group Search Base** field, enter the base path plus the OU where the groups are located.

   > **Note** This value is used after the user is found to exist in the LDAP, after searching with the user search filter. Go to Configuring Security Settings (AD/LDAP).

8. Configure the following values in either owl-env.sh or ConfigMap:
   - **LDAP_ROLES_PROP_NAME**
   - **LDAP_USER_SEARCH_ATTRIBUTE**

   For the details about the owl-env.sh script settings for Search Configuration, go to Configurations in owl-env.sh or ConfigMap.

Users

User Management Overview

Administrators can manage local and Active Directory (AD) mapped users from the User Management page.

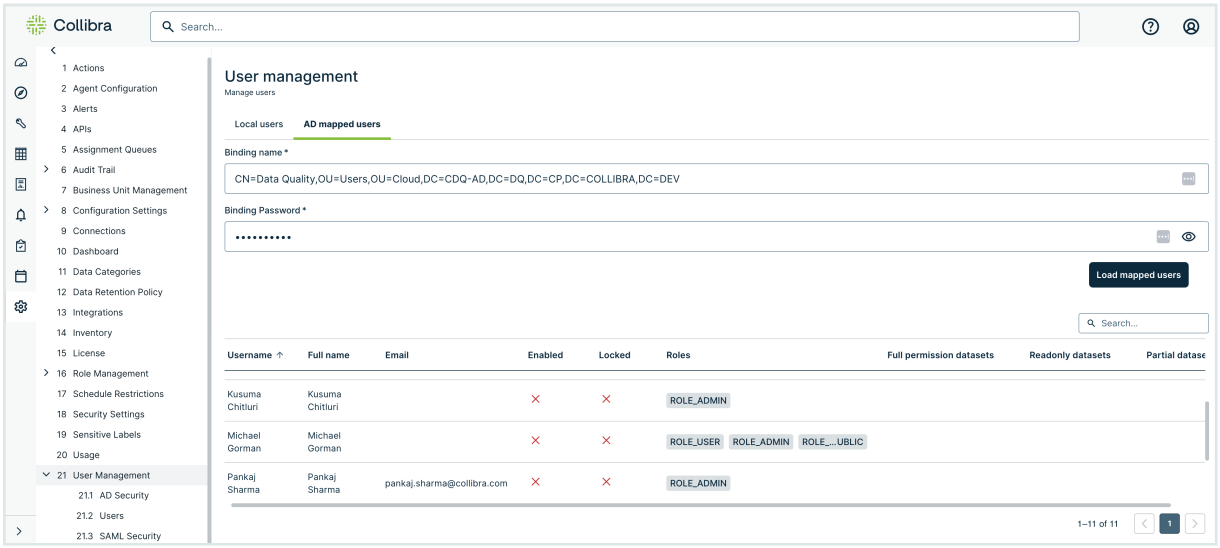
> **Note** Administrators cannot currently add local users from this page.

Column	Description
Username	The name of the user.
Full name	The first and last name of the user.
Email	The user's email address that they used to create their user account.
Enabled	Indicates whether a user account is enabled or not enabled. Click the in the far right column to change the account status. ✓ indicates that the user's account is enabled. ⊗ indicates that the user's account is not enabled.
Locked	Indicates whether a user account is locked or unlocked. An account may be locked after 10 consecutive unsuccessful login attempts. Click the in the far right column to change the account status. ✓ indicates that the user's account is locked. ⊗ indicates that the user's account is unlocked.
Roles	The roles assigned to the user. You can manage the roles assigned to users from the Role Management page.
Full permission datasets	The datasets where the user has full access permission. You can manage user access permissions to datasets on the Dataset Management page.
Readonly datasets	The datasets where the user has read-only access permission. You can manage user access permissions to datasets on the Dataset Management page.

Column	Description
Partial datasets	The datasets where the user has partial access permission. You can manage user access permissions to datasets on the Dataset Management page.
Actions	Click the to edit a user's account. Click the to delete a user from Data Quality & Observability Classic.

To load AD-mapped users, enter the binding name and binding password in the **Binding Name** and **Binding Password** input fields, then click **Load mapped users**.

> **Tip**
> **Binding Name**: The username associated with an account that can query AD or LDAP.
> **Binding Password**: The password of the user account listed in the binding name field that can query AD or LDAP.



Column	Description
Username	The name of the user.

Column	Description
Full name	The first and last name of the user.
Email	This is not used for AD-mapped users.
Enabled	This is not used for AD-mapped users.
Locked	This is not used for AD-mapped users.
Roles	The roles assigned to the user. You can manage the roles assigned to users from the Role Management page.
Full permission datasets	The datasets where the user has full access permission. You can manage user access permissions to datasets on the Dataset Management page.
Readonly datasets	The datasets where the user has read-only access permission. You can manage user access permissions to datasets on the Dataset Management page.
Partial datasets	The datasets where the user has partial access permission. You can manage user access permissions to datasets on the Dataset Management page.

> **Note** Unlike local users, AD-mapped users cannot be edited or deleted from this page.

Known limitations

- When using the latest UI, admin users cannot add local users from the User Management page.
- When editing a user from the Edit [username] modal, the name input fields cannot contain spaces. As a workaround, you can use an underscore instead.

> **Note** For additional user configuration options, go to Security Settings.

What's next?

- Steps
- Steps

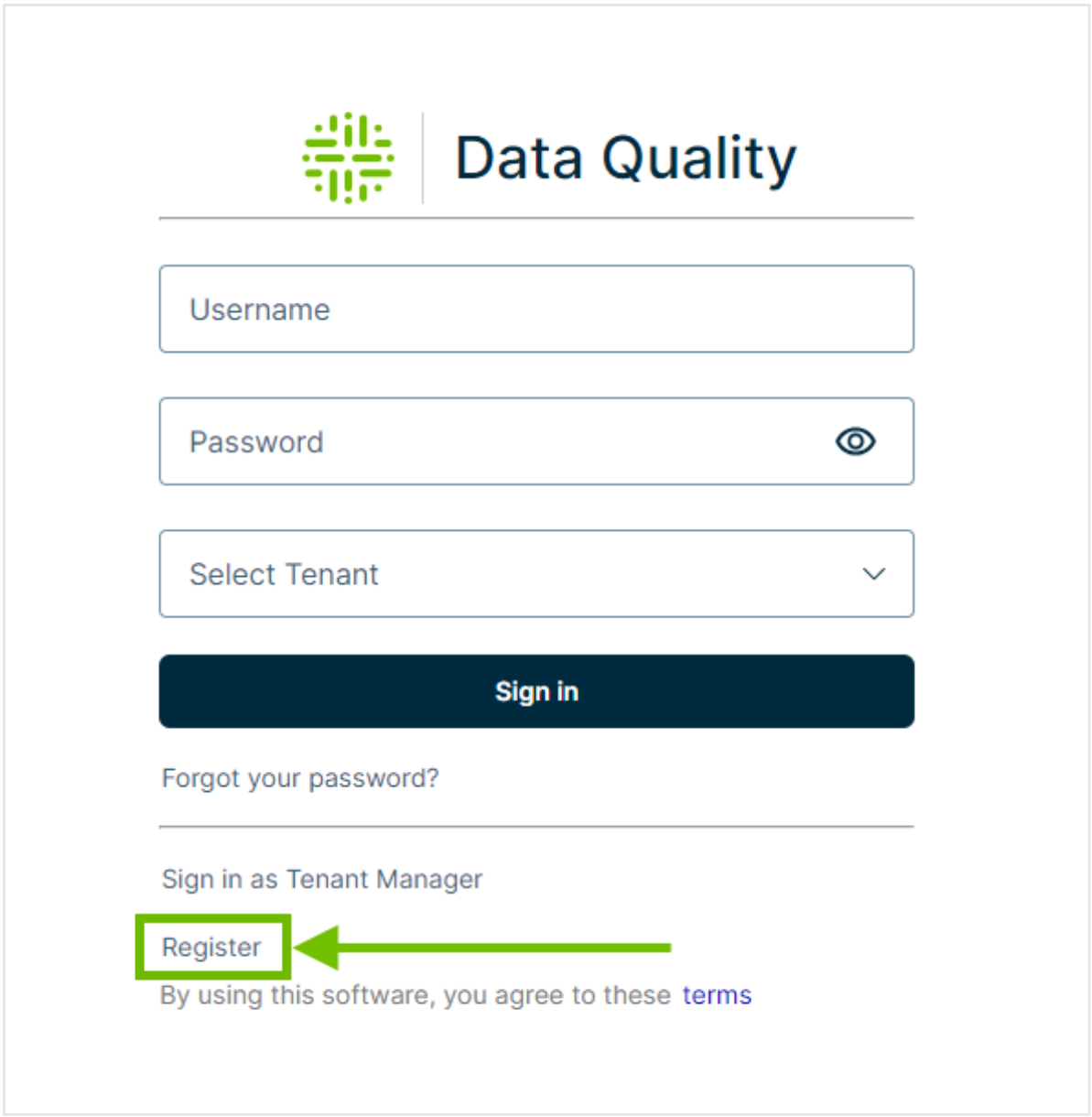
Creating new user accounts

Users can request access to the Data Quality & Observability Classic by creating a new user account on the sign in page.

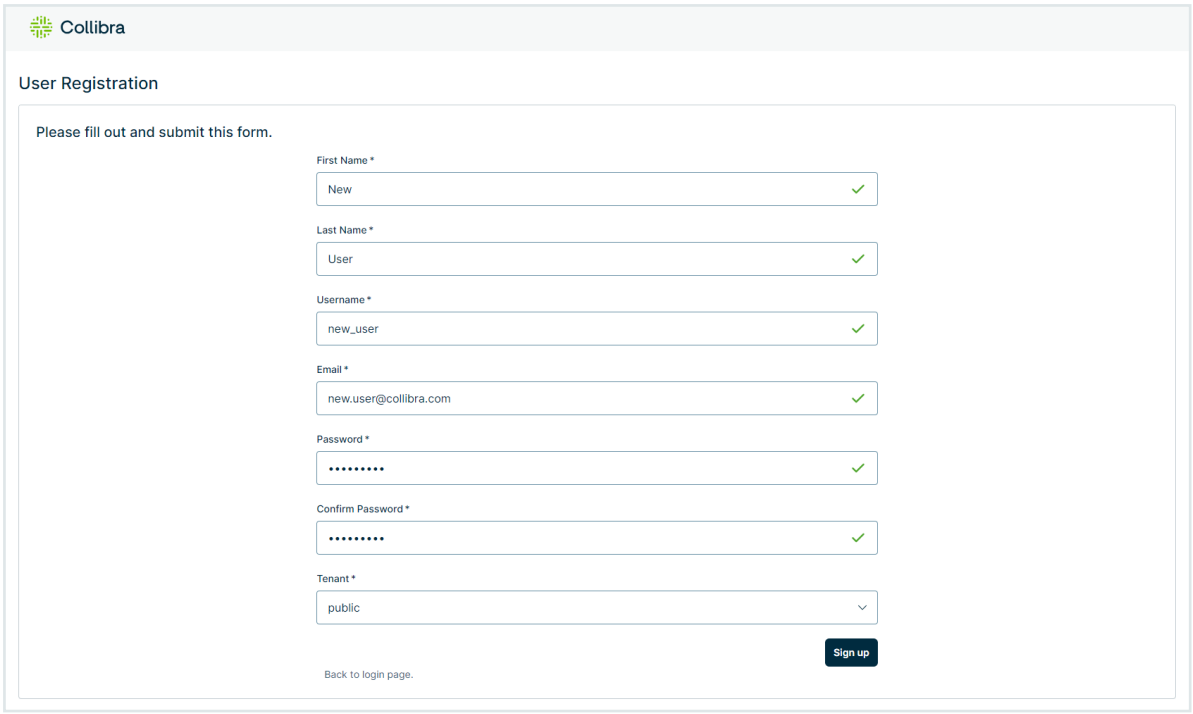
> **Note** For additional details on administering user accounts, go to User Management.

Steps

1. From the Data Quality & Observability Classic sign in page, click the "Register" link.



   » The User Registration form opens.
2. Enter information for all fields.



Field	Description
First Name	Your given name. Do not include spaces.
Last Name	Your surname. Do not include spaces.
Username	The username of your account, which will display throughout the application.

> **Important** Usernames can only contain alphanumerical characters and underscores. Do not include spaces.

Email	The email address associated with your user account.
Password	The password used to sign in to your account.

> **Important** Passwords must meet the following requirements:
> - A minimum of 8 characters
> - At least one upper case letter
> - At least one lower case letter
> - At least one number
> - At least one special character

Field	Description
Confirm Password	Re-enter the password you entered in the Password field.
Tenant	Select a tenant from dropdown menu.

3. Click **Sign up**.

> **Important** The new user account does not have access to the Data Quality & Observability Classic until an application administrator grants user access to one or multiple Data Quality & Observability Classic roles.

> **Note** If LDAP is enabled and a local user password or username is invalid and fails to authenticate, Data Quality & Observability Classic attempts to authenticate against the AD/LDAP server.

Known limitations

- When using the latest UI, admin users cannot add local users from the User Management page.
- When editing a user from the Edit [username] modal, the name input fields cannot contain spaces. As a workaround, you can use an underscore instead.

What's next?

Steps

Editing users and their roles

Administrators can manage local users and their roles from the User Management page.

> **Note** For additional details on administering user accounts, go to User Management Overview.

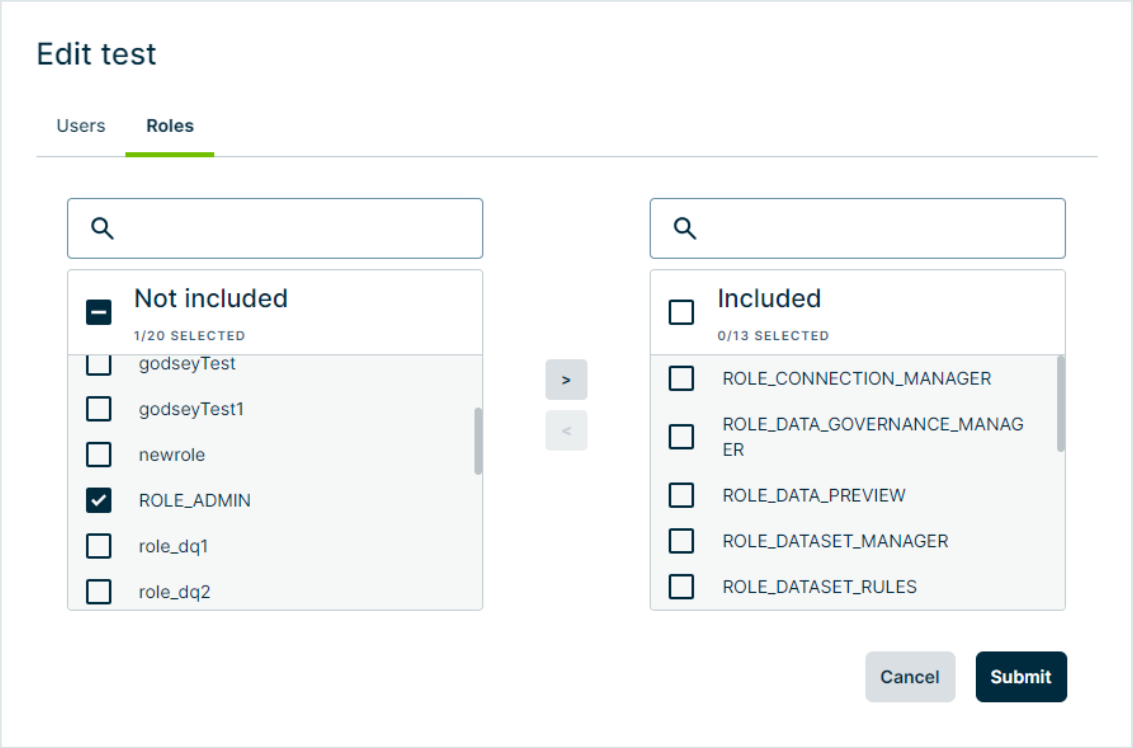
Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **User Management** and then click **Users**.
4. In the far right column, click .
   » The **Edit [user]** modal appears.
5. Select either the **Users** or **Roles** tab and fill out the information.

Field	Description
First Name	The user's first name.
Last Name	The user's last name.
Username	The username of the user.
Email	The email address associated with the user's account.
Enabled	Select this option to enable or disable a user's account.
Locked	Select this option to lock or unlock a user's account. An account may be locked automatically after 10 consecutive unsuccessful login attempts.

Map roles to a user by moving them from the **Not included** box to the **Included** box. To do this, select the role(s) you wish to map to the user, then click . You can also filter the roles in the search fields.

> **Note** To remove a role mapping, you can click the left arrow to move the role (s) you select from the **Included** column back to the **Not included** column.



6. Click **Submit**.

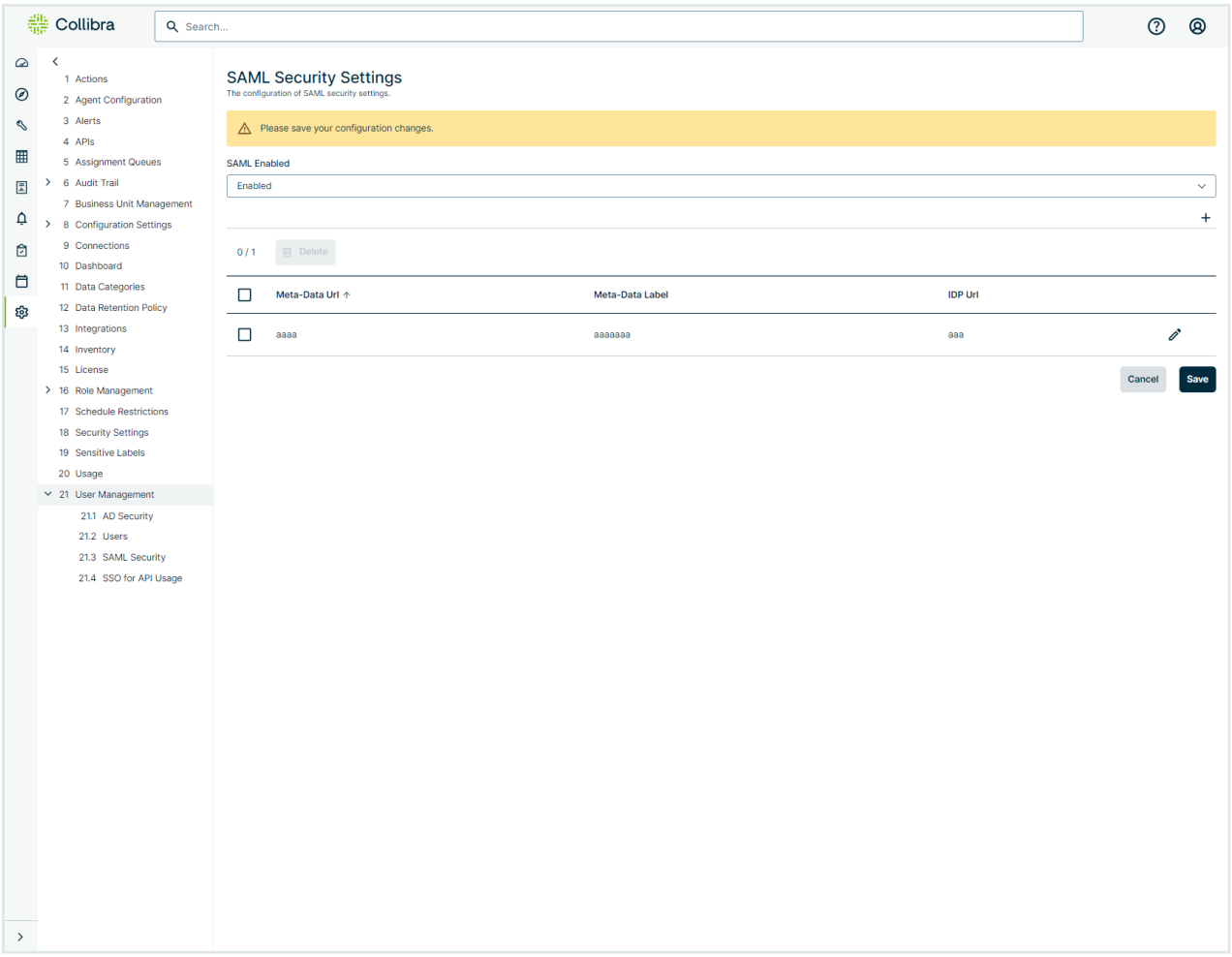
SAML Security

SAML Security Settings Overview

SAML Security Settings lets administrators manage Security Assertion Markup Language (SAML) configurations. Data Quality & Observability Classic uses SAML protocol in order to support single sign-on (SSO) for users to sign into the application with a single identity provider (IdP).

Data Quality & Observability Classic supports both Service Provider (SP) and IdP in single-tenant mode. In multi-tenant mode, only IdP is supported due to a limitation regarding the sharing of tenant information to and from IdP via SAML requests.

For more information about how to configure SAML, see SAML Authentication.



Field	Description
Meta-Data URL	The Identity Provider HTTP URL that returns the SAML metadata document.
Meta-Data Label	The unique label to identify the SAML metadata configuration.
IdP URL	The URL to be used for SSO initiated from the Collibra DQ login page.

> **Note** This is a required field.

Configuring SAML authentication

This page shows you how to integrate Data Quality & Observability Classic with an existing SAML solution and have your application act as a service provider. Once you set the

environment variables, you can access and configure SAML security settings as an administrator from the **SAML Security** page of the **Admin Console**.

- Set the SAML authentication properties
- Download service provider metadata for the IDP
- Enable the SAML sign in option

Set the SAML authentication properties

Before configuring SAML authentication, you must add the following required properties to your configuration.

Standalone installation

1. Add the properties as environment variables to your **owl-env.sh** file located in **<installation_directory>/owl/config/**.
2. Prefix all properties with the `export` statement.
3. Restart the web app.

Cloud native installation

1. Add the properties as environment variables to your **DQ-web ConfigMap**.
2. Recycle the pod.

Required properties

Property	Description
SAML_ENABLED	Whether Collibra DQ uses SAML. If set to `false`, users sign in with a username and password. If set to `true`, SAML handles the authentication request.
SAML_ENTITY_ID	The name of the application for the identity provider, for example *Collibra DQ*. It is an immutable unique identifier of the service provider for the identity provider (IDP).

Warning Please see CORS_ALLOWED_ORIGINS in the **Optional properties** section below if you have SAML configured in Collibra DQ, or if the app sits behind a load balancer.

Optional properties: general

You can further configure your SAML setup with the following optional properties.

Property	Description
CORS_ALLOWED_ORIGINS=${IDP-BASE-URL},${DQ-BASE-URL}	Add this property to environment variables. Replace ${IDP-BASE-URL} with the value of the actual IDP URL (For example: https://ping.auth.com) Replace ${DQ-BASE-URL} with the value of the actual DQ Base URL (For example: https://dq-env.com) **Important** The CORS restriction is enforced for SAML and multi-tenancy, which will break SAML unless the IDP is configured as a trusted origin in Collibra DQ. Therefore, you must add this property for DQ and SAML to work properly. **Note** When using Azure AD, you should use the **User Access URL**, which is located under **Properties** in the Azure SSO configuration.
SAML_ENTITY_BASEURL	The base URL that is provided in the service provider metadata. **Note** Set this property when you use DNS.
SAML_LB_EXISTS	Whether the application needs to configure a load balancer. You generally need this setting only when the **Load Balancer** is set for **SSL Termination**. The default value is `false`. If set to `true`, you must also provide a value for `SAML_LB_SERVER_NAME`.

Property	Description
SAML_METADATA_USE_URL	Whether Collibra DQ uses a URL or a file for the identity provider metadata. The default value is `true`. If set to `false`, the file must be accessible to the owl-web and the path provided in the **Meta-Data URL** field of the **Meta Data Configurations** section under **Admin Console** --> **SAML Setup** --> **Connection**.
SAML_ROLES_PROP_NAME	The attribute name that contains role ID/name in the incoming SAML assertion. The default value is `memberOf`.
SAML_GRANT_ALL_PUBLIC	Whether any user authenticated by the identity provider is allowed to login the Collibra DQ application. The default value is `true`.
SAML_USER_NAME_PROP	The name of the attribute in the SAML response that contains the username of the user who is authenticating.
SAML_TENANT_PROP_NAME	If using multi-tenant mode, the variable in which the identity provider stores the tenant name of the user authenticating in the SAML response. The app will attempt to use the `RelayState` parameter to identify the tenant and then fall back on this property.
SAML_TENANT_PROP_FROM_URL	If using multi-tenant mode, this property controls where the tenant is set to use for authentication. The default value is `false`. If set to `true`, the app attempts to use the `RelayState` property to identify the tenant. If it is set to `false`, the app attempts to find the tenant in a SAML response attribute, which is defined by the `SAML_TENANT_PROP_NAME` property (default=`tenant`).
SAML_KEYSTORE_FILE	The path to the keystore for SSL validation. The store should contain the keypair of the identity provider for SSL verification.
SAML_KEYSTORE_PASS	The password for the keystore provided in `SAML_KEYSTORE_FILE`.

Property	Description
SAML_KEYSTORE_ALIAS	The alias of the keypair (private and public) in the keystore used for SSL verification.
LOCAL_REGISTRATION_ENABLED	An optional property, which allows you to display or hide the registration link for local users on the Collibra DQ sign-in page. Because the registration link is displayed on the sign-in page by default, this property is set to true. To display or hide the registration link from the owl-env.sh file, select from the following properties:

Property	Description
export LOCAL_REGISTRATION_ENABLED=true	Displays the registration link on the sign-in page for local users. This is set to `true` by default.
export LOCAL_REGISTRATION_ENABLED	Hides the registration link on the sign-in page for local users.

To display or hide the registration link from a K8s config map, select from the following properties:

Property	Description
LOCAL_REGISTRATION_ENABLED="true"	Displays the registration link on the sign-in page for local users. This is set to `true` by default.
LOCAL_REGISTRATION_ENABLED="false"	Hides the registration link on the sign-in page for local users.

> **Note** This property must be set globally on deployment.

SAML_MAX_AUTH_AGE	The number of seconds that an IDP authentication is accepted by the application. If the IDP authentication occurred outside this time range, the application considers the value too old to trust and the authentication is not accepted. The default is `14400 seconds` (4 hours).

Property	Description
CSRF_TOKEN_ENABLED	Enables the use of Spring CSRF token implementation.
CSRF_TOKEN_IGNORE_URL_PATTERN	Requests URL patterns to be ignored for CSRF token validation.
CSRF_TOKEN_IGNORE_REFERER_PATTERN	Requests header referer patterns to be ignored for CSRF token validation.

Note While CORS is still an optional configuration, it is required if you have SAML configured in Collibra DQ, or if you have Collibra DQ behind a load balancer.

Note CORS is also enforced for multi-tenancy.

Optional Properties: Meta-Data

When `SAML_METADATA_USE_URL` is set to `true` (default), the following additional properties are available.

Property	Description
SAML_METADATA_TRUST_CHECK	Whether to enable Collibra DQ to do trust verification of the identity provider. The default value is `false`.
SAML_METADATA_REQUIRE_SIGNATURE	Whether Collibra DQ signs authentication requests to the identity provider. The default value is `false`.
SAML_INCLUDE_DISCOVERY_EXTENSION	Whether to enable Collibra DQ to indicate in the SAML metadata that it's able to consume responses from an IDP Discovery Service. The default value is `false`.

Optional Properties: Load Balancer

When `SAML_LB_EXISTS` is set to `true`, the following additional properties are available.

Property	Description
SAML_LB_INCLUDE_PORT_IN_REQUEST	Whether to include the port number in the request. The default value is `false`.
SAML_LB_PORT	The port number of the load balancer. The default value is `443`.
SAML_LB_SCHEME	The protocol of the load balancer. The default value is `https`.
SAML_LB_SERVER_NAME	The server or DNS name. Usually, the same as `SAML_ENTITY_BASEURL` without specifying the protocol, for example without *https://*. This property is required and has no default.
SAML_LB_CONTEXT_PATH	Any path that may be defined on the load balancer.

Example

```
#enable SAML & show the SAML SSO option on the login page
SAML_ENABLED=true

#set SSL communication properties for SAML
SAML_KEYSTORE_FILE=/keystore.p12
SAML_KEYSTORE_PASS=****
SAML_KEYSTORE_ALIAS=****

#in multi-tenant mode set the name of the IDP variable to hold
the tenat name
SAML_TENANT_PROP_NAME=tenant

#set the name of the IDP variable to hold the user roles in the
response
```

```
SAML_ROLES_PROP_NAME=memberOf

#allow login if authenticated to the IDP
SAML_GRANT_ALL_PUBLIC=true

#set the EntityId of the application to be supplied to the IDP
SAML_ENTITY_ID=OwlOneLogin

#optinally set a property that contains the username in the
response
SAML_USER_NAME_PROP=""

#optionally use a file for the IDP metadata vs a URL (default
is true)
SAML_METADATA_USE_URL=false

#optional security settings to
SAML_METADATA_TRUST_CHECK=false
SAML_METADATA_REQUIRE_SIGNATURE=false
SAML_INCLUDE_DISCOVERY_EXTENSION=false
```

Download service provider metadata for the IDP

Once you have enabled and configured SAML authentication, you can download the service provider metadata that is required by your identity provider.

Steps

1. Locate the URL of your Collibra DQ environment in the address bar of your browser window.
2. From the following URL, download the service provider metadata you will need to enable the SAML sign in option:
   `https://<your_dq_environment_url>/saml/metadata`
3. Follow the steps in Enable the SAML sign in option.

Enable the SAML sign in option

When you are ready with your IDP settings, add the final configuration settings in the **Admin Console**:

Steps

1. Sign in as an administrator and select the tenant you want to configure.
2. In the left navigation pane, click **Admin Console**.
3. Click **User Management** and then click **SAML Security**.
   » The SAML Security Settings page opens.
4. From the SAML Enabled dropdown menu, select **Enabled**.
5. Click the to the far right side of the page, under the SAML Enabled dropdown menu.
   » The Add Meta Data Configuration dialog appears.
6. Enter the required information.

Input field	Description
Meta-Data URL	The Identity Provider HTTP URL that returns the SAML metadata document.
Meta-Data Label	The unique label to identify the SAML metadata configuration.
IDP_URL*	The URL to be used for SSO initiated from the Collibra DQ login page.

> **Note** This is a required field.

7. Click **Add**.

> **Tip** Once you complete this setup, restart Data Quality & Observability Classic and sign in using the SAML SSO option.

You can now map SAML Groups to Collibra DQ roles. Go to AD Group to Role Mapping.

Configuring SAML SSO for Azure

When using SAML SSO with Azure as the provider, some additional configuration is required to read the groups and map roles.

When groups are pulled from Azure Active Directory SSO and more than five groups are assigned to a user, the group claims return as a link, rather than the groups list:

```
<Attribute Name="http://schemas.microsoft.com/claims/groups.link">
```

> **Note** Go to https://learn.microsoft.com/en-us/azure/active-directory/develop/reference-saml-tokens for more information about SAML security tokens

> **Note** For customers who have setup and configured the application to return groups that are only present in the SSO application, this issue is not applicable.

When Azure SSO sends a group.link assertion, the application tries to resolve the groups via the link. To activate the link, you must configure the `SAML_GROUPS_LINK_PROP` property.

Steps

1. Add the following property to your **owl-env.sh** file:

Property	Description
SAML_GROUPS_LINK_PROP	Configure this property equal to the group link claim name. Go to SAML Authentication for more information about configuring SAML authentication properties.

2. Obtain a client key from Azure and configure both `AZURE_CLIENT_SECRET` and `AZURE_CLIENT_ID` properties, using the following documentation:

   https://learn.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal#option-2-create-a-new-application-secret

   > **Note** You must also ensure that the application, not delegate, has the MS Graph API application permission **directory.read.all**.

   > **Note** In the IdP setup in Azure, verify that the Sign-on URL is not populated.

Configuring multi-tenant support through SAML RelayState

In multi-tenant environments, you can route SSO to the proper tenant with the SAML provided `RelayState` or `SAML_TENANT_PROP_FROM_URL` properties.

While in a multi-tenant environment, if you set the `SAML_TENANT_PROP_FROM_URL` property to `true`, the application attempts to use the `RelayState` parameter to identify the tenant. When you set `SAML_TENANT_PROP_FROM_URL` to `false`, Collibra DQ attempts to find the tenant in a SAML response attribute, which is defined by the `SAML_TENANT_PROP_NAME` property (default=`tenant`).

You can set the `RelayState` property on the **SAML Security** page of the **Admin Console**.

For more information, go to Set the SAML authentication properties.

SSO for API Usage

SSO for API Usage Overview

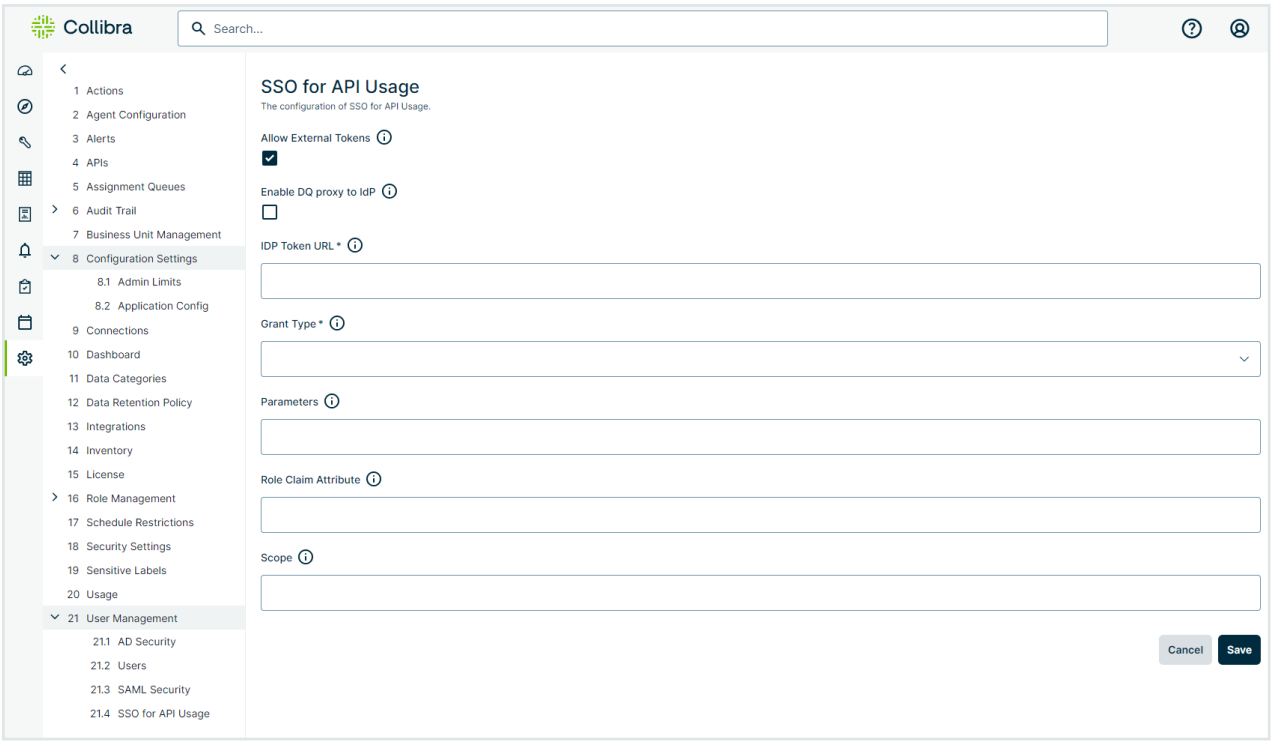
SSO for API Usage allows you to configure single sign-on (SSO) to access Data Quality & Observability Classic APIs using authorization bearer tokens. This section shows you how to set up SSO for API usage in your Data Quality & Observability Classic environment.

There are two methods to configure OAuth or OIDC access:

- Externally managed JWTs (recommended)
- Data Quality & Observability Classic as a proxy to the IdP

Configuring an externally managed JWT

This recommended method allows you to manage the generation and retrieval of access tokens. Once you have an access token, you can then call the API by including it as an authorization bearer token in the request header. Data Quality & Observability Classic processes and validates this token and maps the user roles to grant users access to the application.



Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Configuration Settings**, then click **Application Config**.
   » The Application Configuration Settings page opens.
4. Set the **SSO_API** setting to **TRUE**.
5. From the Admin Console treeview menu, click **User Management**, then **API for SSO Usage**.
   » The SSO for API Usage page opens.
6. Select the **Allow External Tokens** option to enable the acceptance of externally managed JSON Web Tokens (JWT).
7. Set the following properties in the owl-env.sh script:

Property	Description
`export SECURITY_JWT_ALLOW_EXTERNAL=true`	Enables this globally in Data Quality & Observability Classic, but still requires each tenant to enable this in the Allow External Tokens option from the SSO for API Usage from the User Management section of the Admin Console menu.
`export SECURITY_JWT_EXTERNAL_ROLE_CLAIM_ATTRIBUTE=<claim attribute>`	Maps to a property in the JWT that holds an array of ROLES that Data Quality & Observability Classic uses to map users to roles.
`export SECURITY_JWT_EXTERNAL_USERNAME_PROPERTY=<username>`	Maps to a property in the JWT that Data Quality & Observability Classic displays as the username.
`export SECURITY_JWT_TENANT_PROPERTY=<tenant>`	Required in multi-tenant environments. This property is included in the JWT claims to identify the Data Quality & Observability Classic tenant that users are logging into.
`export SECURITY_JWT_TOKEN_ISSUER=<iss>`	The issuer of the JWT that is included in the process to validate the token.

Property	Description
`export SECURITY_JWT_TOKEN_JWKPROVIDERURL=<cert endpoint exposed by the IdP>`	An endpoint of the IdP where the certificates can be retrieved. This property is used to validate the token. You can retrieve the cert endpoint by accessing the oidc/.well-known/openid-configuration endpoint to return the jwks_uri in the response. For example, when you access **https://{{subdomain}}.onelogin.com/oidc/2/.well-known/openid-configuration**, the response shows the jwks_uri as `"jwks_uri": "https://{{subdomain}}.onelogin.com/oidc/2/certs"`. In this case, the cert endpoint is **https://{{subdomain}}.onelogin.com/oidc/2/certs**.

8. Restart the Data Quality & Observability Classic web application.

Example of the JSON payload of a JWT

The following is an example of the decoded JSON payload of a JWT:

```
{
  "sub": "123456789",
  "preferred_username": "example.user",
  "name": "Example User",
  "updated_at": 1234567890,
  "given_name": "Example",
  "family_name": "User",
  "groups": [
    "CollibraAdmin"
  ],
   "tenant": "public",
   "post_logout_redirect_uri":
"http://localhost:9000/logout",
   "accountName": "example.user"
  },
  "aud": "012345a6-abc1-012a-0ab1-01a23b45cd6789012",
  "exp": 1234567890,
  "iat": 1234567890,
  "iss": "https://<subdomain>.onelogin.com/oidc/2"
}
```

In this example, the values used in the owl-env.sh correspond to the JSON payload properties as follows:

owl-env.sh property	Value
SECURITY_JWT_EXTERNAL_USERNAME_PROPERTY	example.user
SECURITY_JWT_TENANT_PROPERTY	tenant
SECURITY_JWT_TOKEN_ISSUER	https://<subdomain>.onelogin.com/oidc/2

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Configuration Settings**, then click **Application Config**.
   » The Application Configuration Settings page opens.
4. Set the **SSO_API** setting to **TRUE**.
5. From the Admin Console treeview menu, click **User Management**, then **API for SSO Usage**.
   » The SSO for API Usage page opens.
6. Select the **Allow External Tokens** option to enable the acceptance of externally managed JSON Web Tokens (JWT).
7. Set the following properties in the **values.yaml** file:

```
security:
  externalJwt:
    enabled: false
    externalRoleClaim: "groups"
    externalUserNamePropName: "accountName"
    externalTenantPropName: "tenant"
    externalTokenIssuer: ""
    externalTokenJwkProviderURL: ""
```

8. Restart the Data Quality & Observability Classic web pod.

Example of a JSON payload of a JWT

The following is an example of the JSON payload of a JWT:

```
{
  "sub": "123456789",
  "preferred_username": "example.user",
  "name": "Example User",
  "updated_at": 1234567890,
  "given_name": "Example",
  "family_name": "User",
  "groups": [
    "CollibraAdmin"
  ],
   "tenant": "public",
   "post_logout_redirect_uri":
"http://localhost:9000/logout",
   "accountName": "example.user"
  },
  "aud": "012345a6-abc1-012a-0ab1-01a23b45cd6789012",
  "exp": 1234567890,
  "iat": 1234567890,
  "iss": "https://<subdomain>.onelogin.com/oidc/2"
}
```

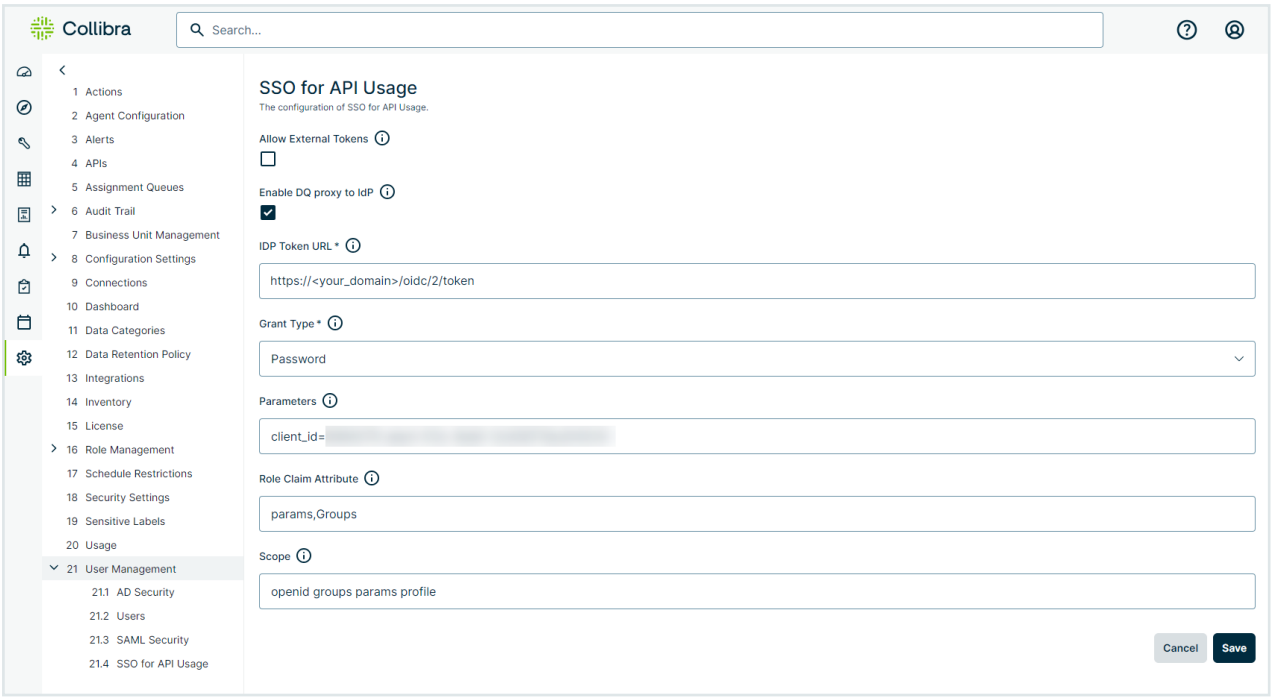
In this example, the values used in the Web ConfigMap correspond to the JSON payload properties as follows:

Web ConfigMap property	Value
SECURITY_JWT_EXTERNAL_USERNAME_PROPERTY	example.user
SECURITY_JWT_TENANT_PROPERTY	tenant
SECURITY_JWT_TOKEN_ISSUER	https://<subdomain>.onelogin.com/oidc/2

Using Collibra DQ as a proxy to the IdP

> **Important** While this method is supported, we recommend using the externally managed JWTs method instead.

This method calls the IdP to retrieve the access_token itself. Data Quality & Observability Classic has an /oath/signin endpoint, which accepts username, password, iss (tenantname) parameters. Data Quality & Observability Classic sends these inputs to the IdP to retrieve an access_token, which Data Quality & Observability Classic then uses to extract the information it needs to create its own token.



Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **Configuration Settings**, then click **Application Config**.
   » The Application Configuration Settings page opens.
4. Set the **SSO_API** setting to **TRUE**.
5. From the Admin Console treeview menu, click **User Management**, then **API for SSO Usage**.
   » The SSO for API Usage page opens.
6. Select the **Enable DQ Proxy to IdP** option.
7. Enter the required information.

Field	Description
Enable DQ proxy to IdP	Enables SSO for APIs.

Field	Description
IdP Token URL	The token endpoint URL of the identity provider (IdP). **Example** https://<your_domain>/oidc/2/token
Grant Type	The grant type used for authentication. **Note** The only option currently available is Password.
Parameters	Static query parameters that are appended to the call to the IdP. This is typically found in the URL of the IdP and is specific to the IdP implementation. **Example** client_id=012345a6-abc1-012a-0ab1-01a23b45cd6789012
Role Claim Attribute	Attributes in the claims array from the access_token that define where DQ roles and user groups should be mapped. **Example** params,Groups
Scope	Optional. IdP-specific properties that define the scope of what is returned in the access_token from the OAuth or OIDC request. **Example** openid groups params profile

8. Click **Save**.

Available global configurations

owl-env.sh property	Description	Default value
`export SECURITY_OAUTH_USECLIENTAUTHORIZATIONHEADER=<value>`	This is the useClientAuthorizationHeader IdP value. When set to `TRUE`, this configuration attempts to use clientID and clientSecret to build an authorization header for the IdP request.	`FALSE`
`export SECURITY_OAUTH_CLIENTID=<value>`	This is the clientID IdP value, which Data Quality & Observability Classic uses to build the authorization header when useClientAuthorizationHeader is enabled.	N/A
`export SECURITY_OAUTH_CLIENTSECRET=<value>`	This is the clientSecret IdP value, which Data Quality & Observability Classic uses to build the authorization header when useClientAuthorizationHeader is enabled.	N/A
`export SECURITY_OAUTH_FALLBACK_ROLES_IDTOKEN=<value>`	When set to `TRUE`, if the access_token fails, Data Quality & Observability Classic extracts the roles information from the id_token.	`FALSE`

Web ConfigMap property	Description	Default value
`SECURITY_OAUTH_USECLIENTAUTHORIZATIONHEADER: <value>`	This is the useClientAuthorizationHeader IdP value. When set to `TRUE`, this configuration attempts to use clientID and clientSecret to build an authorization header for the IdP request.	`FALSE`
`SECURITY_OAUTH_CLIENTID: <value>`	This is the clientID IdP value, which Data Quality & Observability Classic uses to build the authorization header when useClientAuthorizationHeader is enabled.	N/A
`SECURITY_OAUTH_CLIENTSECRET: <value>`	This is the clientSecret IdP value, which Data Quality & Observability Classic uses to build the authorization header when useClientAuthorizationHeader is enabled.	N/A
`SECURITY_OAUTH_FALLBACK_ROLES_IDTOKEN: <value>`	When set to `TRUE`, if the access_token fails, Data Quality & Observability Classic extracts the roles information from the id_token.	`FALSE`

OAuth2 Provider Configuration

This section describes how to configure Microsoft Azure Active Directory B2C (Azure AD B2C) as the primary OAuth2 authentication provider for Data Quality & Observability Classic. Once configured, users can log in to DQ using their Azure AD B2C credentials.

Configuring Azure AD B2C user authentication

This topic describes how to configure Microsoft Azure Active Directory B2C (Azure AD B2C) as the primary OAuth2 authentication provider for Data Quality & Observability Classic. Once configured, users can log in to Collibra DQ using their Azure AD B2C credentials.

> **Important** After successful login, users will only gain access to the Collibra DQ application if their Azure AD B2C roles are correctly mapped to DQ roles. This topic covers basic role mapping; advanced role population is covered in Configuring Azure AD B2C for user authentication via graph API.

Configure the Azure AD B2C portal

The following procedures are performed in the Azure AD B2C portal, in which you will create the application registration that represents your Collibra DQ application. You must have an active Azure subscription with permissions to manage Azure AD B2C tenants, applications, and user flows. For more detailed information, refer to the Microsoft documentation on registering applications.

If you have not yet created an Azure AD B2C tenant, do so now. For more information, see Microsoft's tutorial on creating an Azure AD B2C tenant.

Register the DQ application in Azure AD B2C

Begin by creating the application registration in Azure AD B2C that represents your Collibra DQ application.

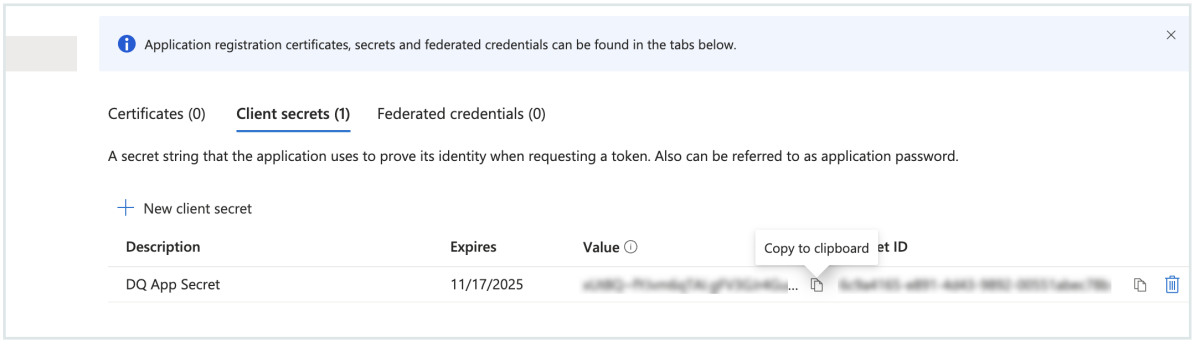
1. Navigate to **Azure AD B2C** in your Azure portal.
2. Under **Manage**, click **App registrations**.
3. Click **New registration**.
4. Provide a meaningful name for your application (for example, "DQ Application").
5. Under **Supported account types**, select the appropriate option for your B2C tenant.
6. Do not specify a **Redirect URI** at this stage. The DQ application will provide this value later.
7. Click **Register**.

8. Keep this application's overview page open in a separate tab or window. You will need its Application (client) ID.

Create a client secret for the DQ Application

The client secret is a confidential key used by your Collibra DQ application to authenticate with Azure AD B2C.

1. From the registered application's overview page in the Azure portal, go to **Certificates & secrets** in the left-hand navigation.
2. Click **New client secret**.
3. Provide a description (for example, "DQ App Secret") and set an expiration (for example, 24 months).
4. Click **Add**.
5. The client secret value displays. Copy this value immediately and store it securely. You will need it for the Collibra DQ application configuration. This value will not be visible again once you leave this page.



The client secret value is displayed only once. Copy this value immediately and store it securely.

Create a sign-up and sign-in user flow

This user flow defines the user experience for sign-up and sign-in in your Azure AD B2C tenant.

1. From the main Azure AD B2C screen in the Azure portal, under **Policies**, click **User flows**.
2. Click **New user flow**.
3. Select **Sign up and sign in** and choose the recommended version.
4. Click **Create**.

5. Give your user flow a name (for example, "B2C_1_YourOrganizationSignin").
6. Select the Identity providers you want to use (for example, "Email signup").
7. Under **User attributes and token claims**, you can keep the defaults for now. You will add a custom attribute for roles in the next section.
8. Click **Create**. Note the name of this user flow, as you will need it for the Collibra DQ application configuration.

Create a custom attribute for roles

To enable role mapping from Azure AD B2C to Collibra DQ, you need to create a custom user attribute. This attribute stores a comma-separated list of roles for each user in Azure AD B2C. For more information, refer to the Microsoft documentation on user flow custom attributes.

1. From the main Azure AD B2C screen in the Azure portal, under **Manage**, click **User attributes**.
2. Click **Add**.
3. For **Name**, enter a descriptive name (for example, "dqroles").
4. For **Data Type**, select **String**.
5. Enter a **Description** (for example, "Comma-separated list of DQ roles").
6. Click **Create**.

> **Note** When this attribute appears in the OAuth2 token, Azure AD B2C automatically prepends `extension_` to its name. So, if you named it dqroles, it will appear as `extension_dqroles` in the token. You will use this version of the name in the DQ configuration.

Configure the DQ Application

Next, configure your Collibra DQ application to integrate with the Azure AD B2C setup.

Enable the OAuth2 login feature

The OAuth2 login feature must be enabled in the Collibra DQ application in order for the OAuth2 Provider Configuration section to be active and editable.

Add the following value to the `owl-env.sh` file:

```
export OAUTH2_LOGIN_ENABLED=true
```

Add the following value to the Web ConfigMap:

```
OAUTH2_LOGIN_ENABLED: true
```

Configure the Azure AD B2C OAuth2 Provider in DQ

This section details how to add your Azure AD B2C tenant as an authentication provider within the Collibra DQ application's administrative interface.

1. Sign in to Collibra DQ as an administrator and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **User Management**, then click **OAuth2 Provider Configuration**.
   A setup page displays with fields to add a new OAuth2 Provider.
4. Fill in the following fields:

Field Name	Description	Details & Where to Find	Example Value
Identity Provider	Specifies the type of OAuth2 provider.	Select **Azure B2C** (currently the only supported provider).	**Azure B2C**

Field Name	Description	Details & Where to Find	Example Value
Registration ID	A unique, arbitrary name for this provider within your Collibra DQ environment. It is used internally by DQ to identify this specific configuration.	Choose a name that is meaningful to you (for example, **my-azureb2c**, **company-b2c**).	**my-azureb2c**
Enabled	Controls whether the **Azure AD B2C** login button displays on the Collibra DQ login screen.	Set to **False** initially. You can toggle this to **True** after completing all configurations.	**False**

Field Name	Description	Details & Where to Find	Example Value
Client ID	The unique identifier for your Collibra DQ application in Azure AD B2C.	Found in Azure AD B2C on the Overview page of your registered DQ application.	**c3250bf3-bcb3-81c5-9acd-d31998da7b29**
Client Secret	The confidential key generated for your Collibra DQ application in Azure AD B2C.	Found in Azure AD B2C, copy this value from **Certificates & secrets** in your registered DQ application. This value is only shown once in Azure AD B2C.	**someSuperSecretValue...**
OAuth2 Claim Field Name	The name of the custom attribute in your Azure AD B2C token that contains the user's roles.	This must be the name of your custom user attribute (for example, **dqroles**) prefixed with `extension_`. So, if you named it **dqroles**, enter `extension_dqroles`.	**extension_dqroles**
Azure B2C Tenant ID	Your Azure AD B2C tenant name.	Often found in the **Application ID URI** of your Azure AD B2C app registration (for example, `<https://yourorganization.onmicrosoft.com/...>`).	**yourorganization**

Field Name	Description	Details & Where to Find	Example Value
Azure B2C Signup/Signin Policy	The name of the Sign-up and Sign-in user flow you created in Azure AD B2C that defines the user authentication experience.	The Name of the user flow you created in Azure AD B2C.	**B2C_1_YourOrganizationSignin**
Azure B2C Profile Edit Policy	Reserved for future use.	Leave blank or default.	
Azure B2C Password Reset Policy	Reserved for future use.	Leave blank or default.	

5. Click **Save**. After saving, your newly configured provider will appear in a list. Note the **Redirect URI** displayed for your provider. This URI will typically be in the format `<https://yourdqinstall.yourdomain.com/login/oauth2/code/{registrationId}>`.
6. Copy the **Redirect URI**.

Update the Azure AD B2C Application Redirect URI

This is a critical step for successful authentication redirection. The Redirect URI tells Azure AD B2C where to send the user after they've successfully logged in.

1. Go back to your Azure portal and navigate to the App registration for your Collibra DQ application.
2. Under **Manage**, click **Authentication** in the left-hand navigation.
3. Under **Platform configurations**, click **Add a platform** and select **Web**.
4. In the **Redirect URIs** field, paste the Redirect URI you copied from the Collibra DQ application.
5. Click **Configure**.

> **Note** It can take up to 30 minutes for Azure to propagate this change.

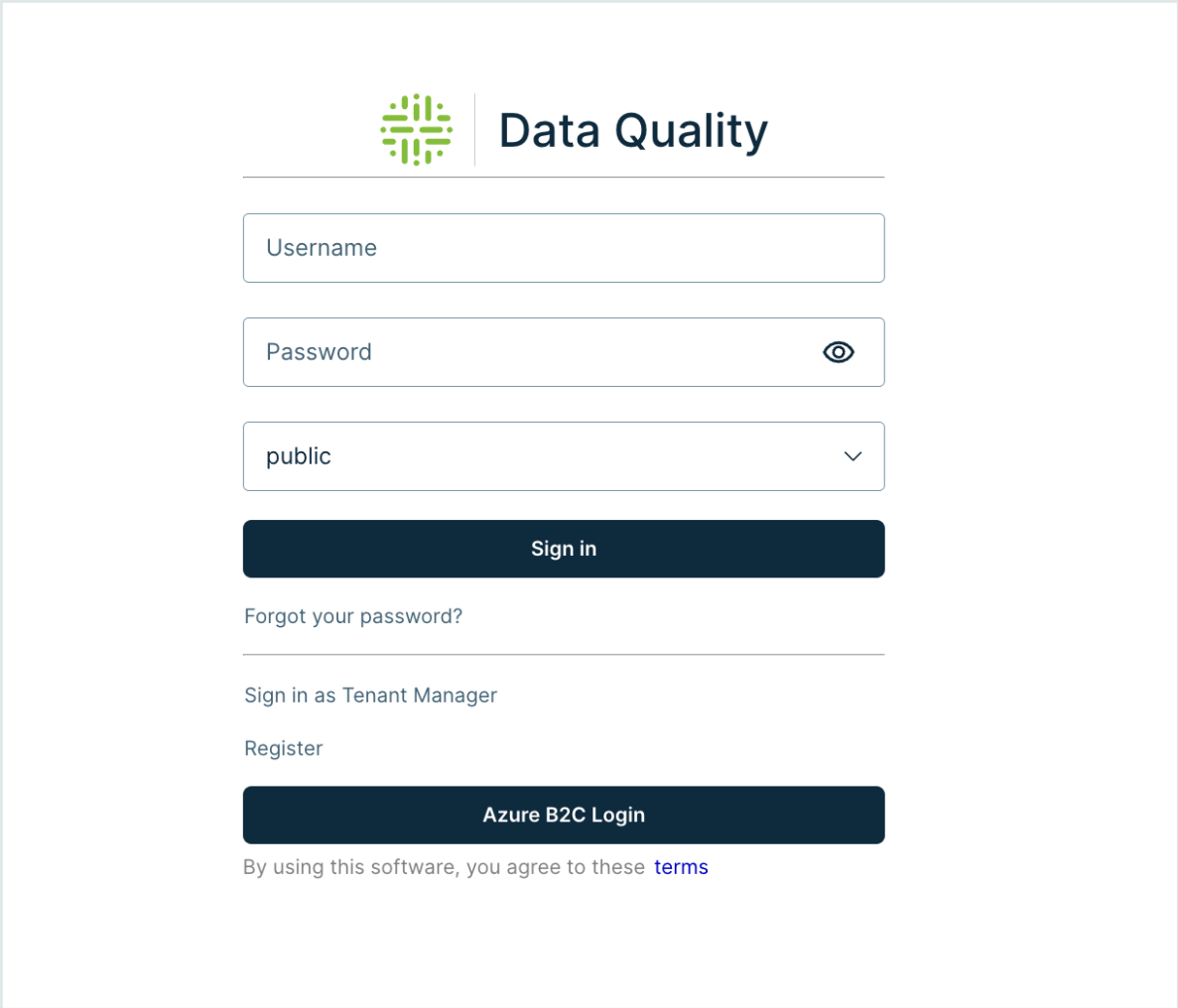
Restart the DQ Application

You must restart the Collibra DQ application when a new OAuth2 provider is added to ensure it is properly registered. If you are only modifying an existing provider (e.g., enabling/disabling), a restart is generally not necessary.

Test basic login functionality

> **Important** At this point in the configuration, the user does not yet have any roles assigned or mapped that grant them access. As a result, you may see an "Access Denied" message despite a successful authentication. This is expected behavior. In the following section, you will map roles to the user.

1. Open the Data Quality & Observability Classic login page.
   You should now see a new button, typically labeled **Azure B2C Login** or similar, below the standard login fields. If the button is not present, ensure that the `OAUTH2_LOGIN_ENABLED` property is set to true.



2. Click the **Azure B2C Login** button.
    - If the configuration is correct, you will be redirected to the Microsoft Azure AD B2C sign-in window.
    - If nothing happens or you encounter an immediate error, check your Collibra DQ application logs and browser developer console network logs. The most common issue at this stage is an incorrect or unpropagated Redirect URI in Azure AD B2C.
3. From the Microsoft login window, sign in with an existing Azure AD B2C user account or create a new one through the configured user flow.
    - Upon successful authentication in Azure AD B2C, you should be redirected to the Collibra DQ application. If you cannot successfully authenticate into Azure B2C, you may need to grant this user permission in your Azure AD B2C application.
    - Even after logging in through Microsoft, you will likely see a message from Collibra DQ indicating "Access Denied" or similar. This is expected behavior at this point. It

confirms that authentication through Azure AD B2C was successful, but the user does not yet have any roles assigned or mapped that grant them access to the Collibra DQ application's features.

4. To grant users access, you must populate user roles in Azure AD B2C and complete the role mapping in Collibra DQ.

Basic role mapping in DQ

Without role mapping, even successfully authenticated users will be denied any access to the Collibra DQ application. This basic setup allows you to map one or two roles to quickly test user access. For advanced role management, refer to Configuring Azure AD B2C for user authentication via graph API.

1. Sign in to Collibra DQ as an administrator (using a non-B2C login, if still available, or an account with existing roles).
2. Click in the left navigation pane.
3. Click **Admin Console**.
4. Click **User Management**, then click **OAuth2 Provider Configuration**.
5. For your Azure B2C provider, click to edit the role mappings.
   The **Edit Mapping for [Provider Name]** page displays.

6. Map the B2C roles to the Collibra DQ application roles:
   - For each role you want to map, find it in the **Not Included** section. If it's not there, you must add your expected B2C roles:
     - In the text box with the placeholder **Enter a name or names (comma delimited)**, type the exact name(s) of the roles you expect to assign to users in Azure AD B2C (for example, **dq-user**, **dq-admin**).
     - Click the next to the textbox. These will appear in the **Not Included** section.
   - Click the next to the **Application Role (Destination)** field for that specific row.
   - From the list of available DQ Application Roles, select the desired internal DQ role (for example, **ROLE_ADMIN**, **ROLE_CONNECTION_MANAGER**).
   - Click **Submit** to save the mapping for that specific role.
   - Repeat this process for all Azure AD B2C roles you intend to use.

What's next?

Configuring Azure AD B2C for user authentication via graph API

Configuring Azure AD B2C user authentication via Graph API

When Microsoft Azure Active Directory B2C (Azure AD B2C) is configured as the primary OAuth2 authentication provider, Data Quality & Observability Classic relies on roles passed in the OAuth2 token from Azure AD B2C to grant user permissions. Microsoft Azure AD B2C does not offer a built-in UI for assigning arbitrary roles to users in custom attributes. Instead, you can use the Microsoft Graph API to populate these attributes.

This topic describes how to populate user roles in Azure AD B2C via Microsoft Graph API. It also provides additional details on role mapping within the Collibra DQ application. This topic assumes you have configured Azure AD B2C for user authentication and your users can successfully authenticate via Azure AD B2C (even if they currently receive a "No Access" message within the Collibra DQ application).

Populating user roles in Azure AD B2C via Microsoft Graph API

The following procedures are performed in the Azure AD B2C portal. You will create an application in Azure AD B2C specifically for interacting with the Microsoft Graph API, and then use it to update user-specific custom attributes (for example, `extension_dqroles`) that will contain the necessary role information. You must have an active Azure subscription with permissions to register applications and grant API permissions in your Azure AD B2C tenant. For general information on Graph API operations, refer to the Microsoft documentation, Managing Azure AD B2C with Microsoft Graph.

> **Note** In this topic, you will manually set up a few users with roles to learn how the Collibra DQ application reads roles from a B2C Oauth2 token. In most enterprise environments, the property you create will need to be updated by some automated flow, script, or custom policy.

Create an API connector application for Graph API access

This application registration in Azure AD B2C will be used solely for authenticating to the Microsoft Graph API to manage user data.

1. Navigate to **Azure AD B2C** in your Azure portal.
2. Under **Manage**, click **App registrations**.
3. Click **New registration**.
4. Provide a meaningful name for your application (for example, "GraphAPI-Access").
5. Under **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multitenant)**. This allows it to access Graph API.
6. Click **Register**.
7. Make note of the Application (client) ID. You will need it later.
8. Select **API permissions** in the left-hand navigation.
9. Click **Add a permission**.
10. Select **Microsoft Graph**.
11. Select **Application permissions**.
12. Grant the following permissions:

    ```
    User.ReadWrite.All
    ```

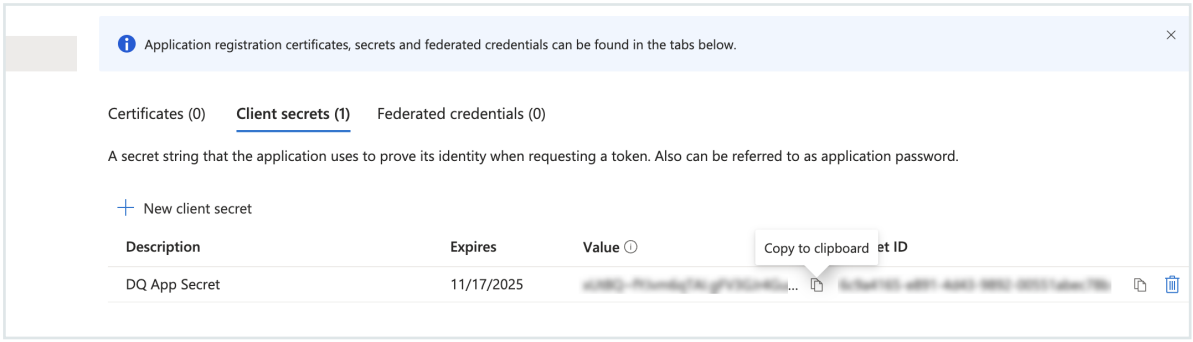
13. Click **Add permissions**.
14. Click **Grant admin consent for {your-tenant-name}** and confirm the prompt. This step grants the necessary permissions for the application to act on behalf of the tenant.

Create a client secret for Graph API access

The client secret is a confidential key used by the GraphAPI application to authenticate with Azure AD B2C.

1. In the GraphAPI-Access application's overview page, go to **Certificates & secrets** in the left-hand navigation.
2. Click **New client secret**.
3. Provide a description (for example, "GraphAPI App Secret") and set an expiration (for example, 24 months).
4. Click **Add**.

5. The client secret value displays. Copy this value immediately and store it securely. You will need it to obtain an access token for GraphAPI calls. This value will not be visible again once you leave this page.



The client secret value is displayed only once. Copy this value immediately and store it securely.

Obtain an access token for Graph API

You will use both the Application (client) ID and the client secret value of the GraphAPI-Access application to obtain an OAuth2 access token. This example uses Postman as the API client.

1. Open Postman.
2. Create a new POST request.
3. Set the URL to: `<https://login.microsoftonline.com/{your-azure-b2c-tenant-name}.onmicrosoft.com/oauth2/v2.0/token>`
4. In the request Body, select `x-www-form-urlencoded` and add the following key-value pairs:
   - `client_id: {The application (client) ID of your GraphAPI-Access application}`
   - `client_secret: {The client secret value that you copied for your GraphAPI-Access application}`
   - `grant_type: client_credentials`
   - `scope: <https://graph.microsoft.com/.default>`
5. Send the request.
6. The response will include an `access_token` (a long string). Copy this token. It is valid for a limited time (for example, 1 hour).

Update user custom attribute via Graph API

Next, use the obtained access token to update the custom attribute (for example, `extension_dqroles`) for a specific user in Azure AD B2C. This attribute will contain the comma-separated roles for that user.

1. Identify the user's Object ID. You need the unique id of the user you wish to update. Use one of the following methods:
    - In the Azure portal, go to **Azure AD B2C** > **Users**
    - Use a GET request via Graph API:
      ```
      GET
      <https://graph.microsoft.com/v1.0/users?$filter=userPrincipalName> eq 'user@yourdomain.com'
      ```
2. Create a new PATCH request in Postman.
3. Set the URL to:
   ```
   <https://graph.microsoft.com/v1.0/users/{id-of-user-to-update}>
   ```
4. In the request Headers, add:
    - `Authorization: Bearer {access_token from prior step}`
    - `Content-Type: application/json`
5. In the request Body, select **raw** and **JSON** and enter the following JSON:

   ```
   {
      "extension_{the id of your b2c-extensions-app}_{custom-role-attribute-name}": "{role1},{role2}"
   }
   ```

    - `{the id of your b2c-extensions-app}`: This is the Application (client) ID of a special app that Azure AD B2C automatically creates for extensions. It's usually named something like b2c-extensions-app (it's typically the one with the description, "Do not modify used by AAD B2C for storing user data."). You can find its ID in App registrations in your Azure AD B2C tenant.
    - `{custom-role-attribute-name}`: The raw name of your custom attribute without `extension_` (for example, `dqroles`).

- `"{role1},{role2}"`: The comma-separated list of roles you want to assign to this user (for example, `dq-admin,dq-user`). These role names should match the Provider Role (Source) values you configured in the Collibra DQ application's role mapping.

6. In the **Authorization** tab, choose **Bearer Token** and paste in the `access_token` (a long string) that you obtained above.
7. Send the request. A successful response will typically have the status code `204 No Content`.

(Optional) Verify user custom attribute via Graph API

To confirm the roles have been updated, you can retrieve the user's details via Graph API.

1. Create a new GET request in Postman.
2. Set the URL to:
   `<https://graph.microsoft.com/beta/users/{id-of-user-to-update}>`
3. In the request Headers, add:
   `Authorization: Bearer {access_token from prior step}`
4. Send the request. The response JSON will include all user attributes, including your `extension_{custom-role-attribute-name}` with the assigned roles.

Test login with populated roles

After populating a user's custom role attribute via Graph API, perform a final login to confirm role access in Collibra DQ.

1. Log out of Collibra DQ.
2. Open the Collibra DQ login page, and click the **Azure B2C login** button.
3. Sign in with the Azure AD B2C user account for which you just populated the custom role attribute.
4. Upon successful login, the user should have the correct roles and access permissions

based on the roles you assigned and mapped. You can verify the assigned roles in the user's profile within the Collibra DQ application.

More Admin Settings

Managing multi-tenancy

Multi-tenancy provides the ability to instantiate different organizations within one entity. For example, an organization could have two separate divisions called OrgTraders and OrgInsurance, respectively. The OrgTraders division is not allowed to see the DQ Check results of the OrgInsurance division, making it critical that these organizations remain separate in Collibra DQ. With multi-tenancy, you can segregate these organizations into two separate tenants within the overarching Collibra DQ web application to protect the activity of one organization from the other.

Prerequisites

- DNS entry for owl web server IP.
- DNS entry for multiTenantSchemaHub.

Prerequisites for URL-based tenancy: tenant.host

- DNS entry for each tenant.
- DNS entry for every tenant you want to create (the example below is called "auto-mation").

> **Note** Each of the records above point to the same IP address.

Enabling multi-tenancy

> **Note** If you are upgrading your version of Collibra DQ, go to Standalone Upgrade.

1. Make sure the web application has started up one time and that you have successfully logged into it with the default credentials.
2. Stop all the components using the `./owlmanage.sh stop` command.
3. Modify the owl-env.sh file to include the following parameters:
    - `export multiTenantSchemaHub=owlhub`: This is a new schema that will get created on owlweb start (note the name of the TenantSchemaHub can be changed to the desired name at setup time).
    - `export MULTITENANTMODE=TRUE`: This enables multi-tenancy mode.
    - `export URLBASEDMULTITENANTMODE=TRUE/FALSE`
        - `TRUE` (default): This value means that you are using the tenant name as a sub-domain (see prerequisites).
        - `FALSE`: This value means you will let Collibra DQ manage tenants via sessions/tokens.
4. If using agents as part of the DQ operation, modify the owl.properties file to include the following:
    - spring.agent.datasource.url=jdbc:postgresql://cdh-edge-dan.us-east4-c.c.owl-hadoop-cdh.internal:5432/postgres**?currentSchema=owlhub** (matching the name of the schema set on step 3-1 above).
    - jdbc:postgresql://cdh-edge-dan.us-east4-c.c.owl-hadoop-cdh.internal:5432/postgres**?currentSchema=owlhub** (matching the name of the schema set on step 3-1 above).
5. Once the settings have been configured for multi-tenancy, start up the owlweb host first, using `./owlmanage.sh start=owlweb` command.
6. Start up the agents, using the `./owlmanage.sh start=owlagent` command.

> **Note** To use multi-tenancy when `URLBASEDMULTITENANTMODE=TRUE`, make sure there are DNS entries to the tenant endpoints.

You can enable multi-tenancy in the web ConfigMap for Kubernetes. The high-level setup for Kubernetes is virtually the same as standalone deployments. Where you add or modify

properties in owl-env.sh for standalone deployments, you add the properties to the ConfigMap for Kubernetes deployments.

If you are using Helm charts, you can customize and modify configuration files to suit the needs of different environments. The Helm charts define all sets of properties.

The following is an example of the type of configurations you might see when running the Helm chart to deploy Kubernetes. These configurations are located in the values.yaml file.

```
flyway: false
server_port: 9005
server_port_2: 9001
multitenant_mode: false
multitenant_schema_hub: owlhub
url_based_multitenantmode: false
owl_base: /opt
owl_agent_id: 0
install_path: /opt/owl
spark_home: /opt/spark
```

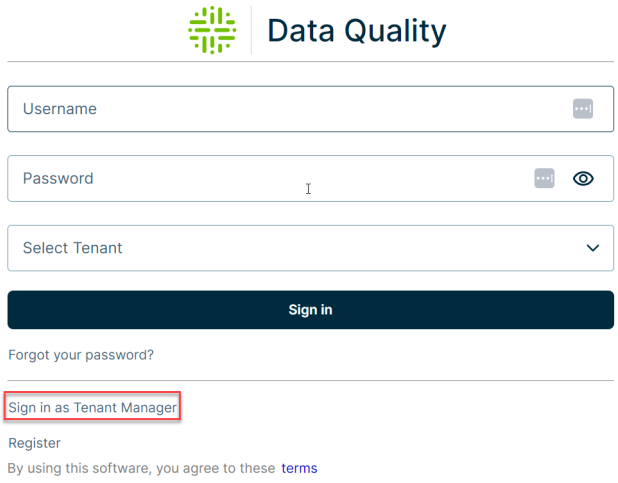
For more information about using Helm charts, go to **Collibra DQ Helm chart** in Installing Data Quality & Observability Classic on Self-hosted Kubernetes.

Adding, editing, and deleting a tenant

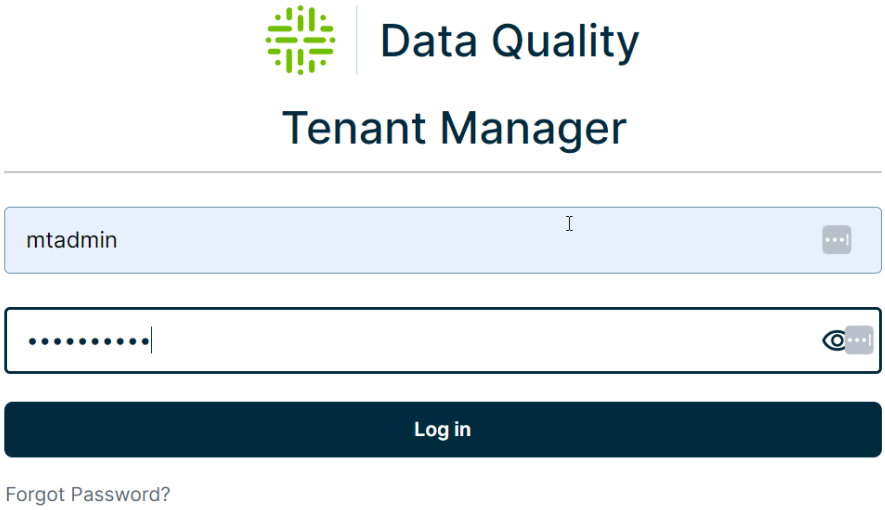
Regardless of the need for isolated environments, Collibra DQ supports dynamically created tenants, as part of the DQ web application. You can add, edit, or delete a tenant on the Tenant Management page.

Steps

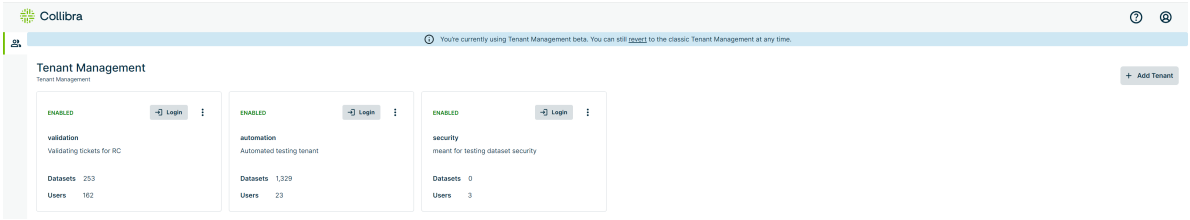
1. Sign into Collibra DQ.
2. Click **Sign in as Tenant Manager**, which appears below the **Forgot your password?** link.



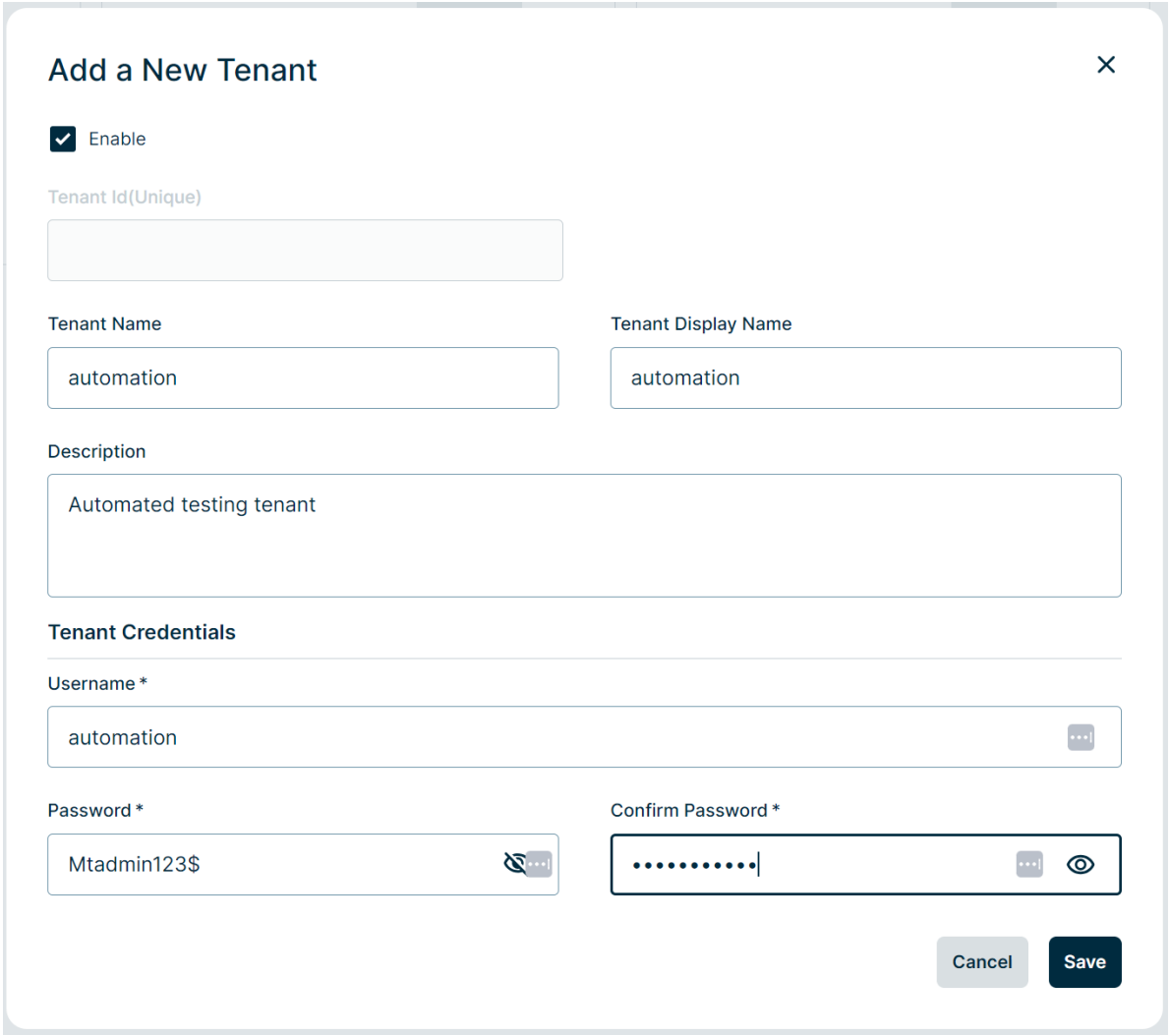
3. On the Tenant Manager login page, enter the following credentials:
   - Username: **mtadmin**
   - Password: **mtadmin123**



4. Click **Log in**.
   » The Tenant Management page opens, where you can add, edit, and delete your tenants.



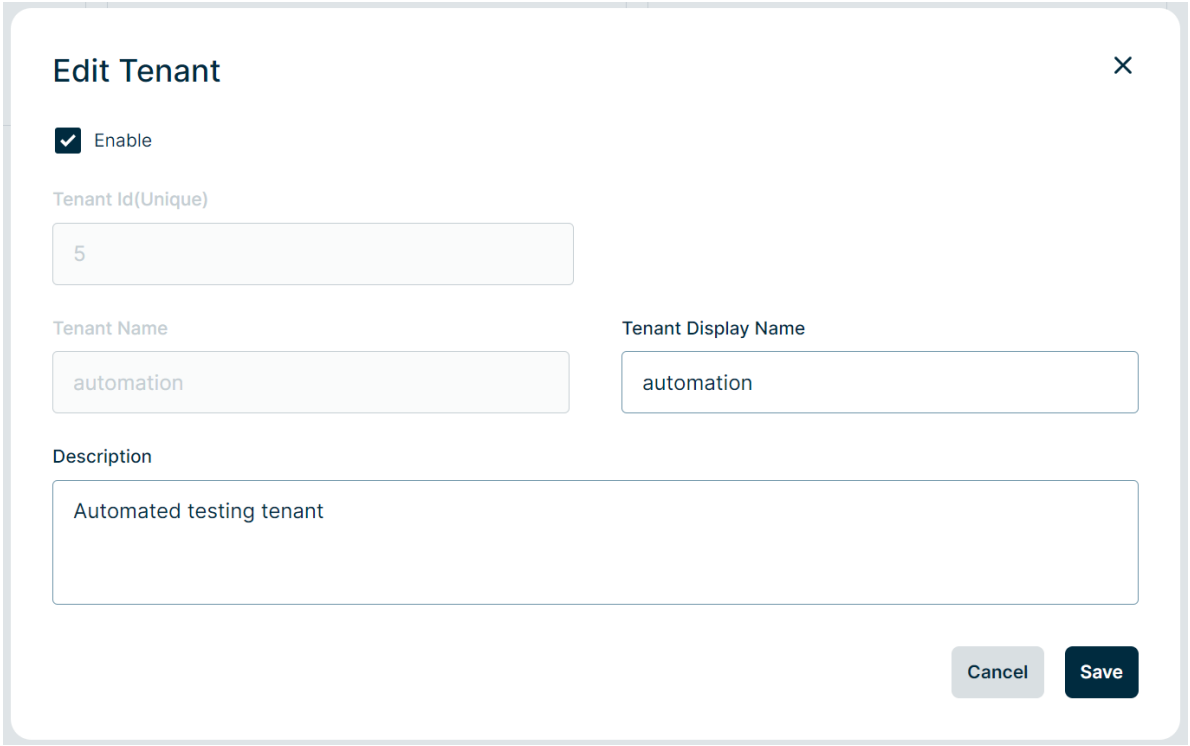
5. To add a new tenant, click the **+ Add Tenant** button in the top-right corner of the Tenant Management window.
   » The Add a New Tenant modal appears.



6. Enter the following information:
   - **Tenant Name**: Name for the tenant.
   - **Tenant Display Name**: Tenant name that displays in the Tenant Management page.
   - **Description**: Description for the tenant.

- **Username**: Username you use when you login to the tenant.
- **Password**: Password you use when you login to the tenant.

7. To enable the tenant, select the **Enable** box.
8. Click **Save**.
   » The new tenant displays on the Tenant Management page.
9. To edit a tenant, select the tenant from the Tenant Management page and click the icon next to the **Login** button.
10. Select **Edit** from the dropdown.
    » The Edit Tenant modal appears.



11. Make the desired changes and click **Save**.
12. To delete a tenant, click the icon and select **Delete**.

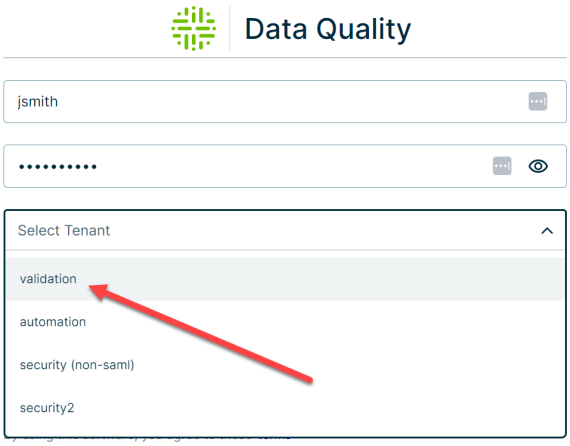
Signing into a tenant

You can select a tenant from a dropdown list on the main Collibra DQ login page.

> **Note** For more information about adding, editing, and deleting tenants, go to Adding, editing, and deleting a tenant.

Steps

1. Sign into Collibra DQ.
2. Enter your username and password credentials.
3. Select your tenant from the Select Tenant dropdown list.



> **Note** All enabled tenants will display in the Select Tenant drop down list. Access to tenants are handled by the administrator(s) within each tenant individually.

4. Click **Sign in**.

> **Note** If you haven't registered as a Collibra DQ user, click the **Register** link, which displays under the **Sign in as Tenant Manager** link and complete the registration form.

Job Management

Concurrency

Concurrency in Collibra DQ is when multiple DQ Jobs run at the same time. Collibra DQ applies a default batch size limit of 5 Jobs that can run at once. These batch size limits ensure that the DQ Agent can properly allocate resources so it does not overload in high concurrency situations. As soon as the initial 5 Jobs begin to complete, any remaining Jobs can then enter the Jobs queue and begin to process.

Concurrency sizing

Before you run DQ Jobs concurrently, it is good to understand sizing specifications for and when you might need to allocate more resources to run your Jobs concurrently.

We generally advise that you allocate around 200 megabytes of Executor Memory per Job. For example, a batch of 10 concurrent Jobs should allocate 2 gigabytes of RAM to run correctly. The following table shows general guidance for resource allocation for concurrent DQ Jobs.

Batch Size	Agent Pod Memory
5	1 GB
10	2 GB
20	4 GB
40	8 GB
80	16 GB
160	32 GB
320	64 GB

Keep in mind that Spark allocates overhead memory on top of the requested memory. For example, an executor pod configured with 6GB of memory may actually request 6.8GB of memory. For more information, go to Apache Spark Memory Management.

> **Note** The default batch size is 5. If you run more than 5 Jobs concurrently, ensure your agent has sufficient memory. See the batch size configuration page for more details.

Overriding the default batch size

Unless you set the batch size in the owl-env.sh script for Standalone deployments or the Helm chart for Cloud native deployments, the batch size defaults to 5. This means that the DQ Agent applies a limit of 5 Jobs that can run concurrently. This page shows you how to override the default batch size for both Standalone and Cloud native deployments.

1. Open a terminal session.
2. Open the owl-env.sh file with the following command:

```
vi /home/owldq/owl/config/owl-env.sh
```

3. Use a base64 encoder, such as the free encoder from base64encode.org, to encode the string `default?batchSize=5` (replacing `5` with your desired value) and enter the encoded value after `DEFAULT_BATCH_ID` in the following command:

```
# The string "default?batchSize=5" needs to be base64
encoded
export DEFAULT_BATCH_ID=default?batchSize=5
```

> **Example** Your base64 encoded value should resemble the following example:
>
> ```
> export DEFAULT_BATCH_ID=ZGVmYXVsdD9iYXRjaFNpemU9NQ==
> ```

4. Enter the following command:

```
export DEFAULT_BATCH_SIZE=<your-batch-size>
```

> **Tip** The value in `<your-batch-size>` should meet the concurrency sizing requirements of your use case.

5. End the terminal session.

1. Open a terminal session.
2. Enter the following command to pass your new default values to your agent batch ID and agent batch size:

```
--set global.configMap.data.agent_batch_
id="default?batchSize=<your-batch-size>"
--set global.configMap.data.agent.data="<your-batch-size>"
```

> **Tip** Enter a value in `<your-batch-size>` that meets the concurrency sizing requirements for your use case.

   a. If your Collibra DQ app is managed by an alternative Kubernetes package manager, ensure you update the **owl-agent-configmap** with the following variables:

```
AGENT_BATCH_ID: "your-base64-encoded-value"
AGENT_BATCH_SIZE: <your-batch-size>
```

> **Example** Your base64 encoded string should resemble the following example:
>
> ```
> export AGENT_BATCH_
> ID=QUdFbVlYWnNkRDRpWVpiZXZGTnBlbVU5Tkw9PQ==
> ```

> **Tip** Enter a value in `<your-batch-size>` that meets the concurrency sizing requirements for your use case.

   b. Restart the DQ agent pod to complete the updates.
3. Choose one of the following options to update your DQ agent resources:
   a. Update the DQ agent from the user interface
      i. Sign in to Collibra DQ.
      ii. Hover your cursor over the ⚙ icon and click **Admin Console**.
         » The Admin Console opens.

i. Click **Remote Agent**.
   » The **Agent Management** page opens.
ii. Click the pencil icon.
   » The **Edit Agent** modal opens.
iii. Update **Number of Executors**, **Executor Memory**, **Number of Cores**, and **Driver Memory** according to the concurrency sizing requirements for your use case.
iv. Click **Save**.

b. Update the DQ agent through Helm to set your new limits
   i. Open a terminal session.
   ii. In your Helm values.yaml file, update the DQ agent's `memory` and `cpu` quotas according to the concurrency sizing requirements for your use case:

```
agent:
   enabled: true
   replicaCount: 1
   nodeSelector:
     key: ""
     value: ""
   resources:
     requests:
       memory: "<your-new-memory-value>"
       cpu: "<your-new-cpu-value>"
     limits:
       memory: "<your-new-memory-value>"
       cpu: "<your-new-cpu-value>"
```

   iii. Perform the Helm upgrade on the existing DQ release.
   iv. End the terminal session.

Livy

This section contains information about Apache Livy.

Working with Livy

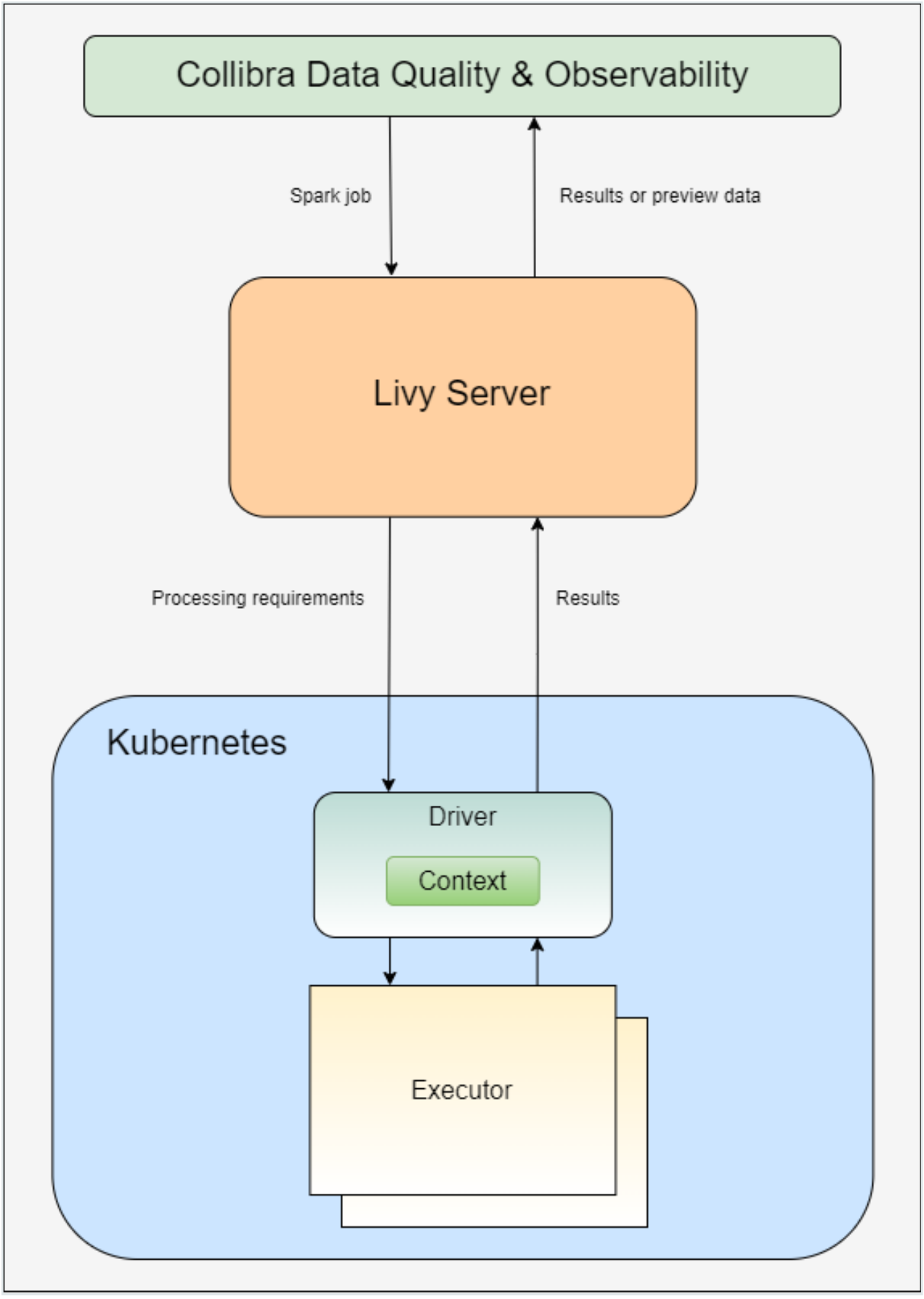
What is Livy?

Apache Livy is a RESTful web service that serves as a session manager for Spark jobs on Kubernetes and Hadoop deployments of Data Quality & Observability Classic. Livy creates temporary Spark pod sessions (Spark Contexts) to efficiently run jobs in the Spark cluster. Data Quality & Observability Classic uses Livy to help estimate the number of resources it needs to process and preview jobs that run against datasets created from remote file connections, such as S3 and ADLS. Without Livy, you cannot preview the data that you bring in from remote files.

Additionally, when you create rules on the Rule Workbench, Livy's ability to read and transform large datasets allows you to use the Result Preview component for a convenient sample of available columns in your dataset. With a preview of the columns in your dataset next to the rule builder, Livy enables you to more efficiently write SQL business rules without having to leave the page to reference your dataset.

> **Note** Livy is included as an optional component in most Collibra DQ release packages. Contact your Collibra Account Team to help you choose a release package with Livy.

Livy architecture within Collibra DQ


Collibra Data Quality & Observability
Spark job
Results or preview data
Livy Server
Processing requirements
Results
Kubernetes
Driver
Context
Executor

1. Data Quality & Observability Classic submits a Spark job to the Livy server.
2. Livy:
    - Examines the data to determine the optimal number of resources it needs to efficiently process the job.
    - Creates a Spark context to communicate with the Spark runtime and coordinate the execution of the job.

    > **Note** Kubernetes provides the compute space for the ephemeral Spark containers that comprise the Spark session needed to run, but there is not a Spark Cluster when running on Kubernetes.

3. Spark workers execute the job inside the cluster.
4. The results return to Data Quality & Observability Classic.

Supported deployment types

Livy is currently supported on Kubernetes deployments of Data Quality & Observability Classic.

> **Important** Livy is not currently supported on Standalone or Hadoop deployments.

Supported remote file data sources

The following table shows the remote file connections that support Livy.

Connection	Supported	Estimate Job
Amazon S3	✓ Yes	✓ Yes
HDFS	✓ Yes	✓ Yes
Google Cloud Storage	✓ Yes	✓ Yes
Azure Data Lake Storage	✓ Yes	✓ Yes
Azure Blob Storage	✓ Yes	✓ Yes
Network File Storage		

Connection	Supported	Estimate Job
CSV	✓ Yes	✓ Yes
ORC	✓ Yes	✓ Yes
Parquet	✓ Yes	✓ Yes
Avro	✓ Yes	✓ Yes
JSON	✓ Yes	✓ Yes
XML	✓ Yes	✓ Yes
Delta	✓ Yes	✓ Yes
Hudi	✓ Yes	✓ Yes

Setting up Livy

Livy is required to preview data and use Estimate Job on remote file connections. It is also required to use the Result Preview feature on the Rule Workbench. This section shows you how to configure Livy for a Kubernetes deployments Collibra DQ.

Steps

1. To create a Livy pod, perform a Helm upgrade or update the values.yaml file with the following command:

```
--set global.livy.enabled=true
```

> **Tip** `global.livy.enabled` is set to `false` by default. In order to use Livy, this must be set to `true`

2. When the Livy pod is running in Ready status, restart the web pod.

> **Note** The Livy pod will update with the web pod.

3. Sign in to Collibra DQ.
4. Click in the sidebar navigation menu, then click **Session Management** next to Remote File Connections.
   » The **Session Management** dialog opens.

> **Note** When Livy is unavailable, Session Management appears greyed-out

5. Enter the required information.

Option	Description
Cores	The number of executor cores for Spark processing. The default value is 1.
Memory	The amount of RAM allocated as storage during Spark processing. Allocate at least 1 GB of memory per Livy pod.
Workers	The number of Livy sessions available in a Spark cluster for Livy to distribute processing tasks. When you increase the number of workers, an additional Livy session is created for each additional worker you specify. For example, if you increase your workers from 1 to 3, then 2 additional Livy sessions are created for a total of 3. The minimum value is 1.

> **Tip** Set the memory and worker values based on your file size. For larger files, you may need to increase the memory and workers. We do not recommend increasing the number of cores to a value greater than 1.

6. Click **Update Session**.
   » A confirmation message appears and asks you to acknowledge that all cached preview data will be evicted.
7. Create a DQ Job on a Remote File Connection.

Troubleshooting Livy

- If Livy does not show RUNNING status in the session manager, you can validate that pods are available with the following command:

```
$ kubectl get pods
```

- If you find that there are errors or data does not load on the Load File step, this may indicate an issue with Livy. You can try to update the session. If there are ongoing issues, you can terminate your session from Session Management ▸ **Terminate Session**.

Pendo

> **Note** The Pendo integration is active by default.

Pendo is an analytics application embedded in Collibra DQ that helps us analyze, develop, and improve our product. No sensitive information is ever collected, we only leverage high-level usage statistics to improve our offerings. If no modifications are made to the default settings, Pendo will not block or impair the intended functionality of Collibra DQ in any way.

If you install to a Spark Standalone environment, modify the <install-dir>/config/owl-env.sh file by adding your license name.

```
export DQ_INTEGRATION_PENDO_ACCOUNTID=<your-license-name>
export DQ_INTEGRATION_PENDO_ENABLED=true
```

> **Note** For more information, please review Collibra's Subprocessors page.

If you install a Kubernetes environment, modify the <install-dir>/config/configMap file by adding your license name.

```
export DQ_INTEGRATION_PENDO_ACCOUNTID=<your-license-name>
export DQ_INTEGRATION_PENDO_ENABLED=true
```

> **Note** For more information, please review Collibra's Subprocessors page.

Security Configuration

Overview

Data Quality & Observability Classic offers multiple methods of user authentication, including a local user store and Active Directory or generic LDAP integration.

Security can be configured to meet your needs. Advanced options to segment groups and roles are available. Additionally, options for SAML and SSO are available.

You can control configurations at the Web (UI), PostgreSQL, and application layers depending on your security requirements.

Encryption

Encryption is available for data in transit and at rest. When enabled, encryption of data in Data Quality & Observability Classic is forward-looking.

We support the following encryption protocols for **data in transit**:

TLS

- TLS 1.2
- TLS 1.3

AES

- TLS_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

ChaCha20

- TLS_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256

We support the following encryption protocol for **data at rest**:

AES secret key

- PBEWithHMACSHA512AndAES_256 algorithm

Configuring Security Settings (AD/LDAP)

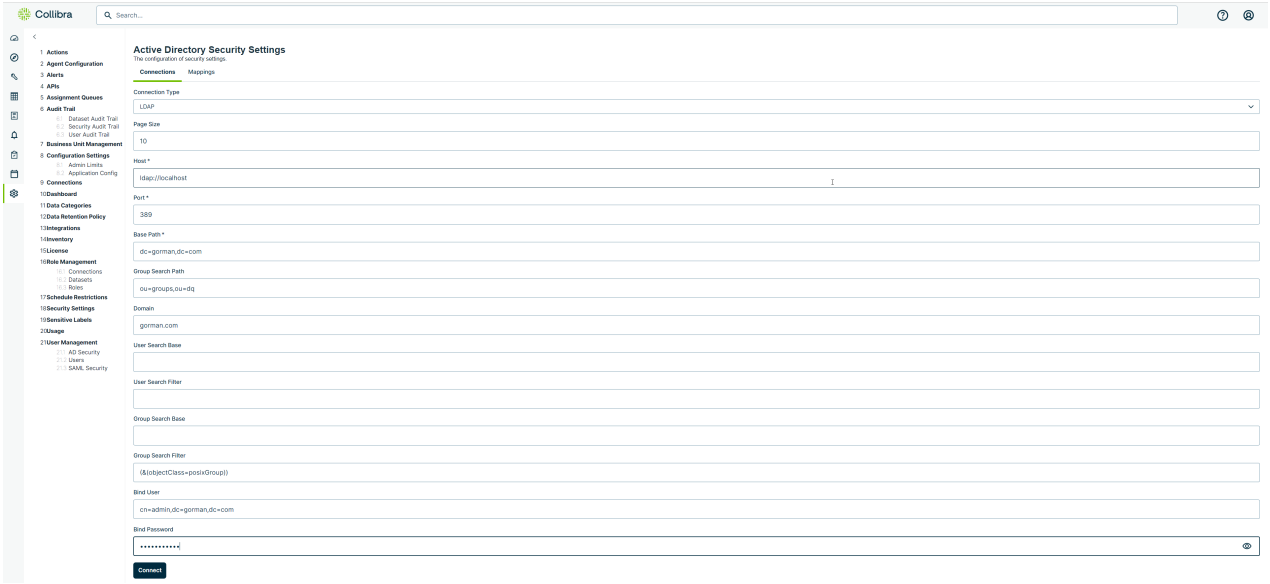
You can configure security settings from User Management in the left navigation panel.

Prerequisites

- You have Admin permissions (ROLE_ADMIN) assigned to your User Profile.
- You have the ability to look at (or get information from) attributes on User and Group LDAP objects. This can be done through CLI, using ldapsearch or through an LDAP object browser.

Configure Security Settings for Active Directory and LDAP

This section describes how to configure security settings for the Active Directory and LDAP connection types.



> **Note** You must add specific script settings in owl-env.sh or ConfigMap when configuring LDAP and Active Directory. For more information, go to Configurations in owl-env.sh or ConfigMap.
> For information about mapping an AD Group to a Collibra DQ role, go to AD Group to Role Mapping.

For information about configuration settings for Active Directory and LDAP, select the applicable tab below.

Configure Active Directory Settings

Configure security settings for Active Directory by navigating to User Management, in the left navigation panel, and selecting **AD Security**.

The following table shows the available input fields for the Active Directory connection type.

> **Note** An asterisk (*) next to a setting represents a required input field.

Setting	Description
Connection Type	Select **Active Directory** from this dropdown.
Page Size	Set a value greater than 0 to control query page size.
Host*	Hostname or URL of your LDAP or LDAPS server, for example, ldap://12.345.678.90.
Port*	Port to connect to your LDAP or LDAPS server. **Note** The default ports are **389** for LDAP and **636** for LDAPS.
Base Path*	Base path configured for LDAP. The value entered is the base domain information and the format is usually **DC=xxxx,DC=com**.

Setting	Description
Group Search Path	Organizational Units where the Groups are located. It is based on the LDAP setup. The value entered is the domain object path where the groups are located; for example, **OU=OwlGroups,OU=Groups** (this value is combined with the Base Path to look for groups).



Tip You can use a wildcard in this field to limit the search results. For example, to see only the groups where the common name starts with "Collibra", enter **cn=Collibra*** in the Group Search Path field. You will now see only groups that start with the name "Collibra" (CollibraPublic, CollibraAdmin, etc.).

Setting	Description
	Note After Group Search Path is configured, it is recommended that you restart the Collibra DQ web application.
Domain	The domain name used to signify when non-local users log in. **Note** This field applies only to Active Directory configurations.
Bind User	DN of an admin user that is used for authentication; for example, *admin@collibra.com*.
Bind Password	Password of an admin user for the bind account.

Configure LDAP Settings

Configure LDAP security settings by navigating to User Management, in the left navigation panel, and selecting **AD Security**.

The following table shows the available input fields for the LDAP connection type.

Note An asterisk (*) next to a setting represents a required input field. A restart of the Collibra DQ web application may be needed for values to take effect.

Setting	Description
Connection Type	Select **LDAP** from this dropdown.
Page Size	Set a value greater than 0 to control query page size.
Host*	Hostname or URL of your LDAP or LDAPS server, for example, ldap://12.345.678.90.

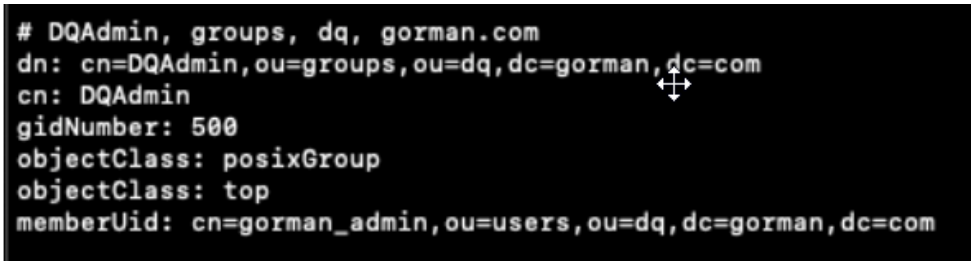
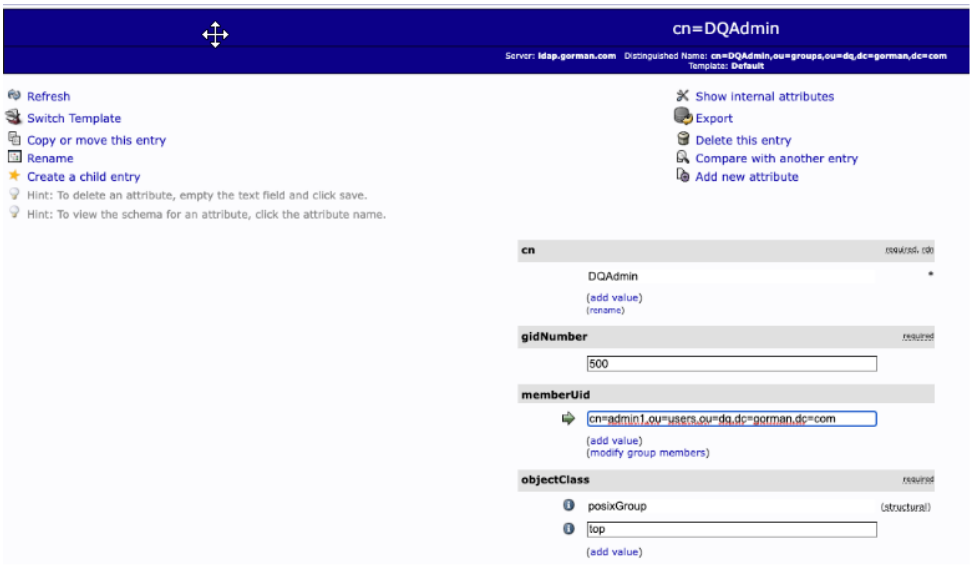
Setting	Description
Port*	Port to connect to your LDAP or LDAPS server. **Note** The default ports are **389** for LDAP and **636** for LDAPS.
Base Path*	Base path configured for LDAP. The value entered is the base domain information and the format is usually **DC=xxxx,DC=com**.

Setting	Description
Group Search Path	Organizational Units where the Groups are located. It is based on the LDAP setup. The value entered is the domain object path where the groups are located; for example, **OU=OwlGroups,OU=Groups** (this value is combined with the Base Path to look for groups).



Tip You can use a wildcard in this field to limit the search results. For example, to see only the groups where the common name starts with "Collibra", enter **cn=Collibra*** in the Group Search Path field. You will now see only groups that start with the name "Collibra" (CollibraPublic, CollibraAdmin, etc.).

Setting	Description
	Note After Group Search Path is configured, it is recommended that you restart the Collibra DQ web application.
Domain	The domain name used to signify when non-local users log in. **Note** This field applies only to Active Directory configurations.
User Search Base	Base DN where the LDAP users for Collibra DQ are located; for example, **CN=Users**.
User Search Filter	When set, this LDAP filter is used to locate users at login. This filter is based on your LDAP configurations.
Group Search Base	The base DN where all the groups are located. **Note** This field applies only to LDAP configurations.

Setting	Description
Group Search Filter	LDAP filter used to narrow down group objects located under a base DN. Default: **(&(objectclass=group))** **Note** This field applies only to LDAP configurations.  
Bind User	DN of an admin user that is used for authentication; for example, *admin@collibra.com*.
Bind Password	Password of an admin user for the bind account.

Configuring LDAP settings

When configuring LDAP, you must add the settings on this page in the **owl-env.sh** (Standalone) or **Web ConfigMap** (Cloud Native).

This configuration occurs upon start-up of the Collibra DQ web application. See Standalone Install for a complete list of the **owl-env.sh** scripts.

> **Note** These settings are not required for Active Directory because the default properties specific to the AD library load upon the start-up of the Collibra DQ WebApp.

Required script settings to configure LDAP

The following table provides descriptions of the **owl-env.sh** (Standalone) or **Web ConfigMap** (Cloud Native) settings to add when you configure LDAP.

> **Note** You can override the default script settings, however, it is dependent on your LDAP provider and how LDAP is implemented.

owl-env.sh setting	Description
LDAP_GROUP_RESULT_DN_ATTRIBUTE	The fully qualified name of the group with the base path; for example, **CN=OwlAppAdmin,OU=OwlGroups,OU=Groups,DC=owl,DC=com**. Default: **distinguishedname**.
LDAP_GROUP_RESULT_NAME_ATTRIBUTE	The common name of the group. Default: **CN**.

owl-env.sh setting	Description
LDAP_GROUP_RESULT_CONTAINER_BASE	This value is used only when the LDAP Group Object has no attributes for the previous properties. In this case, this would get set to the base path and organizational units, where the group is located. Default: **None**. **Note** If the LDAP_GROUP_RESULT_DN_ATTRIBUTE is not found, the LDAP_GROUP_RESULT_NAME_ATTRIBUTE is combined with the LDAP_GROUP_RESULT_CONTAINER_BASE, to save the fully qualified group name.

Script Settings for Search Configuration

The following table provides descriptions of the **owl-env.sh** (Standalone) or **Web ConfigMap** (Cloud Native) settings to add when you configure the search filter.

owl-env.sh setting	Description
LDAP_ROLES_PROP_NAME	Attribute of the group object to find the name. This is usually the common name of the group and is used to map the LDAP group to the internal role. Default: **memberOf**.
LDAP_USER_SEARCH_ATTRIBUTE	Property used as part of a filter to look for user membership in a group. The Group Object in LDAP has an attribute where users are listed. This value must be set for this property. Default: **distinguishedname**.

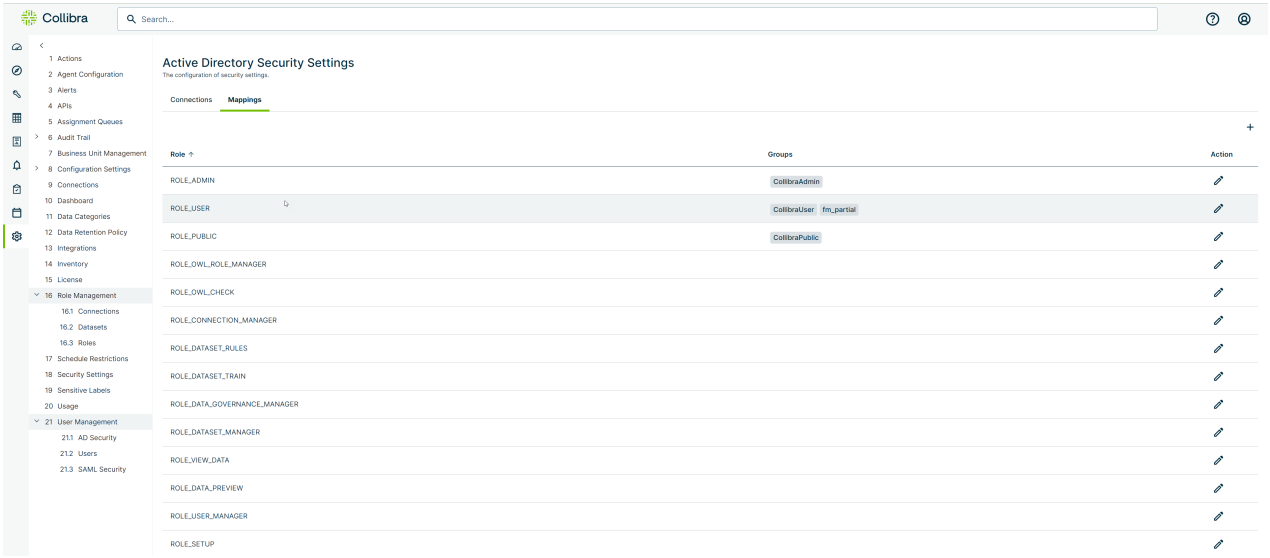
AD Group to Role Mapping

When you map an AD Group to a Collibra DQ role, you grant role-based access to all users from the selected AD Group to the selected Collibra DQ role outlined in the steps below. You can find additional information on creating custom application roles on the Role-Based Access Control page.

Application properties set in the **owl-env.sh** file or **Web ConfigMap** can be set to determine which LDAP properties correspond to LDAP query results. Go to Configurations in owl-env.sh or Web ConfigMap for more information about the LDAP properties.

When group mapping, you need the full path (unique) and the display name.

> **Example**
> `LDAP_GROUP_RESULT_DN_ATTRIBUTE=distinguishedname`
> `LDAP_GROUP_RESULT_NAME_ATTRIBUTE=CN`



Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Click **Admin Console**.
3. Click **User Management** and then click **AD Security**.
4. Ensure the connection type you want is selected in the **Connections** tab.

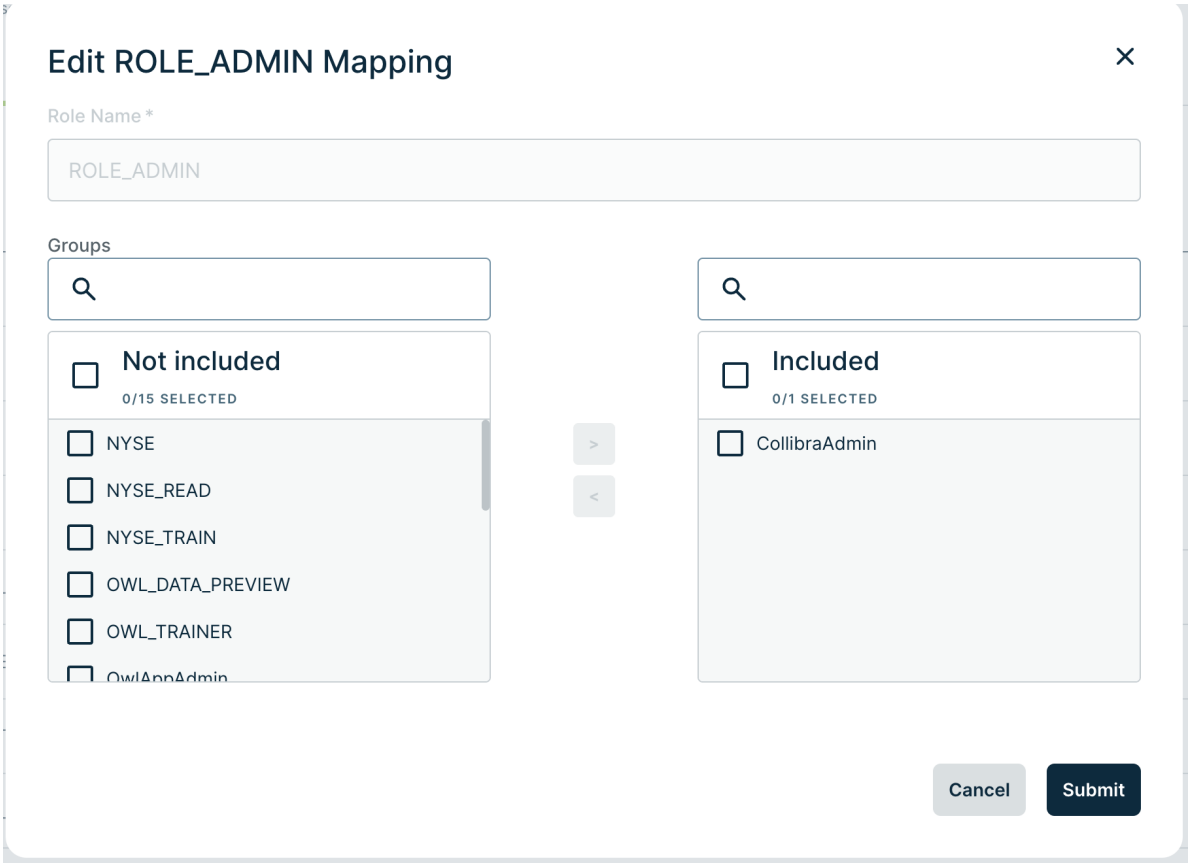
   > **Note** To enable SAML, go to SAML Authentication.

5. Select the **Mappings** tab.
6. To map a group to an existing role, click the ✎ icon for that role, in the **Action** column.

   The edit role mapping window displays for that specific role. In this window, you can map groups to Collibra DQ roles by moving the groups from the **Not included** box to the

Included box. To do this, click the  and icons. You can also filter the groups and roles in the search fields.

> **Note** If Active Directory or LDAP is configured, the groups populate in the list, based on settings from the Connections page.

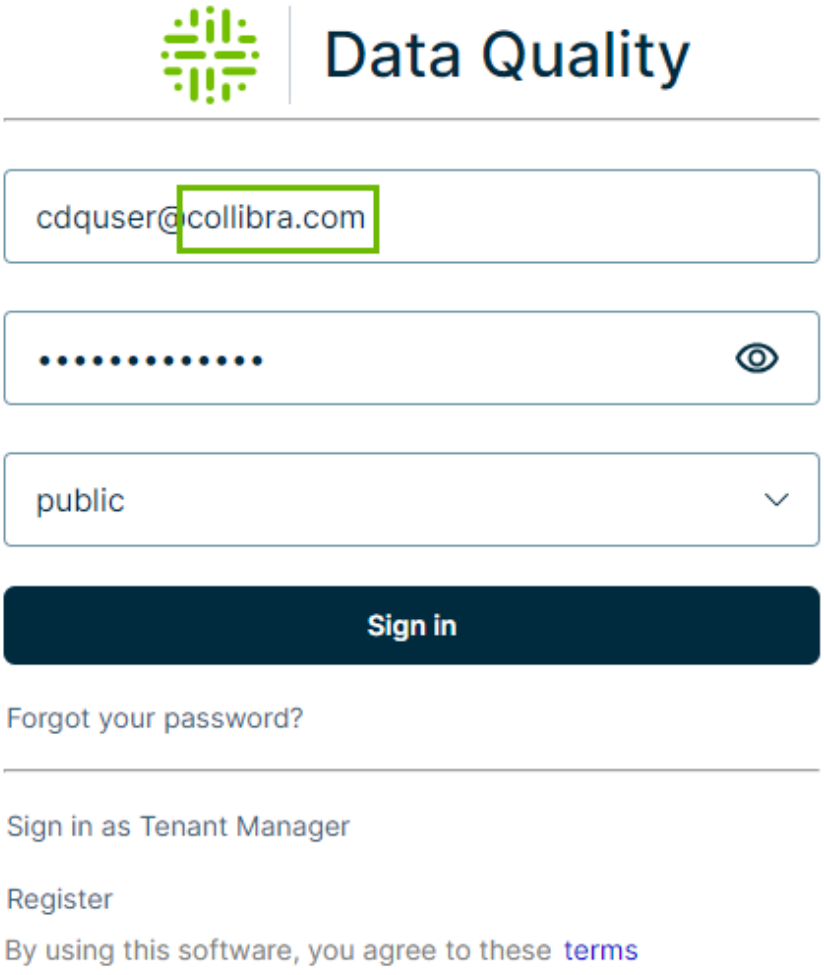


7. Click **Submit**.

Once you successfully map an AD Group to an AD Role, log out of Collibra DQ and log in again as a Domain user.

> **Note** You must restart the Collibra DQ web application by running `./owlmanage.sh restart_owlweb` when toggling AD Enabled.

When logging into the Collibra DQ web application, append the domain to the end of the username.



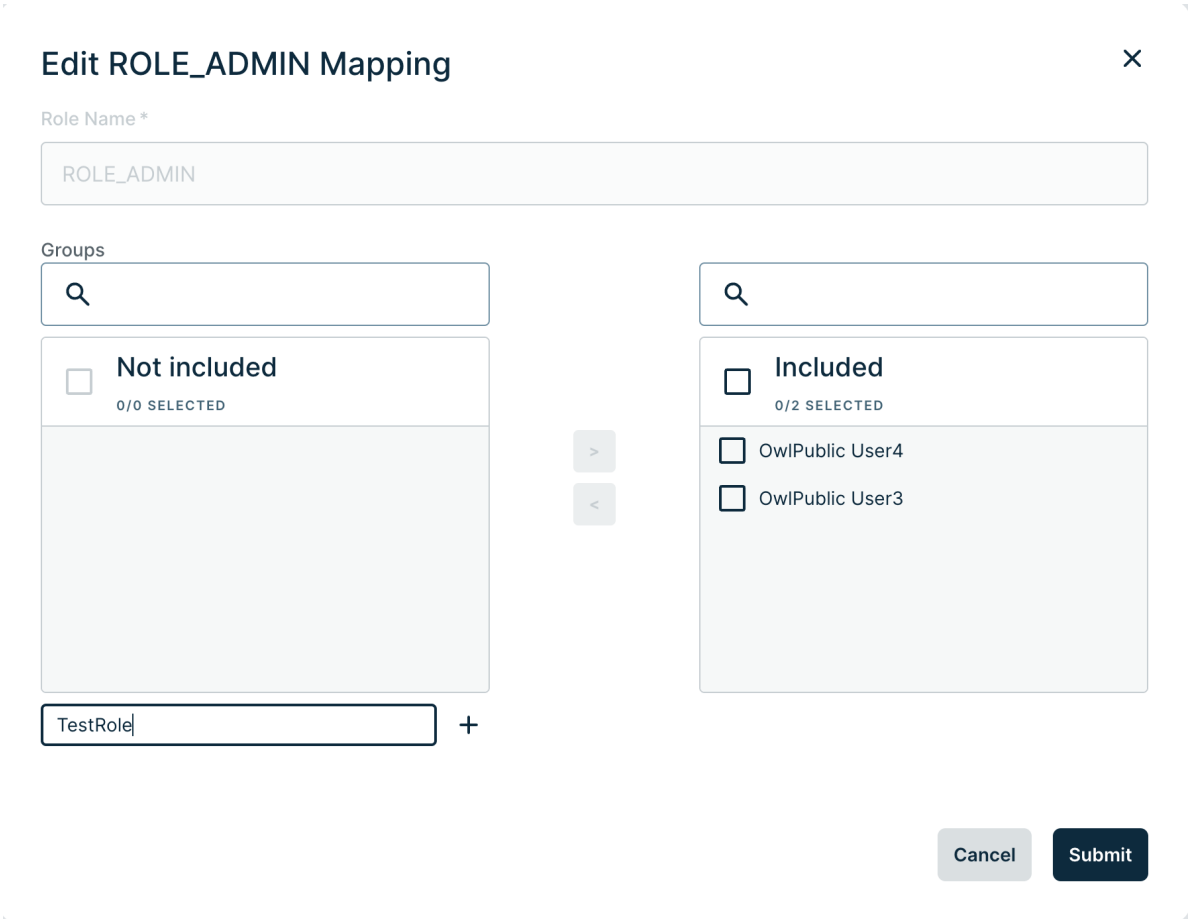
Group to Role Mapping Without AD

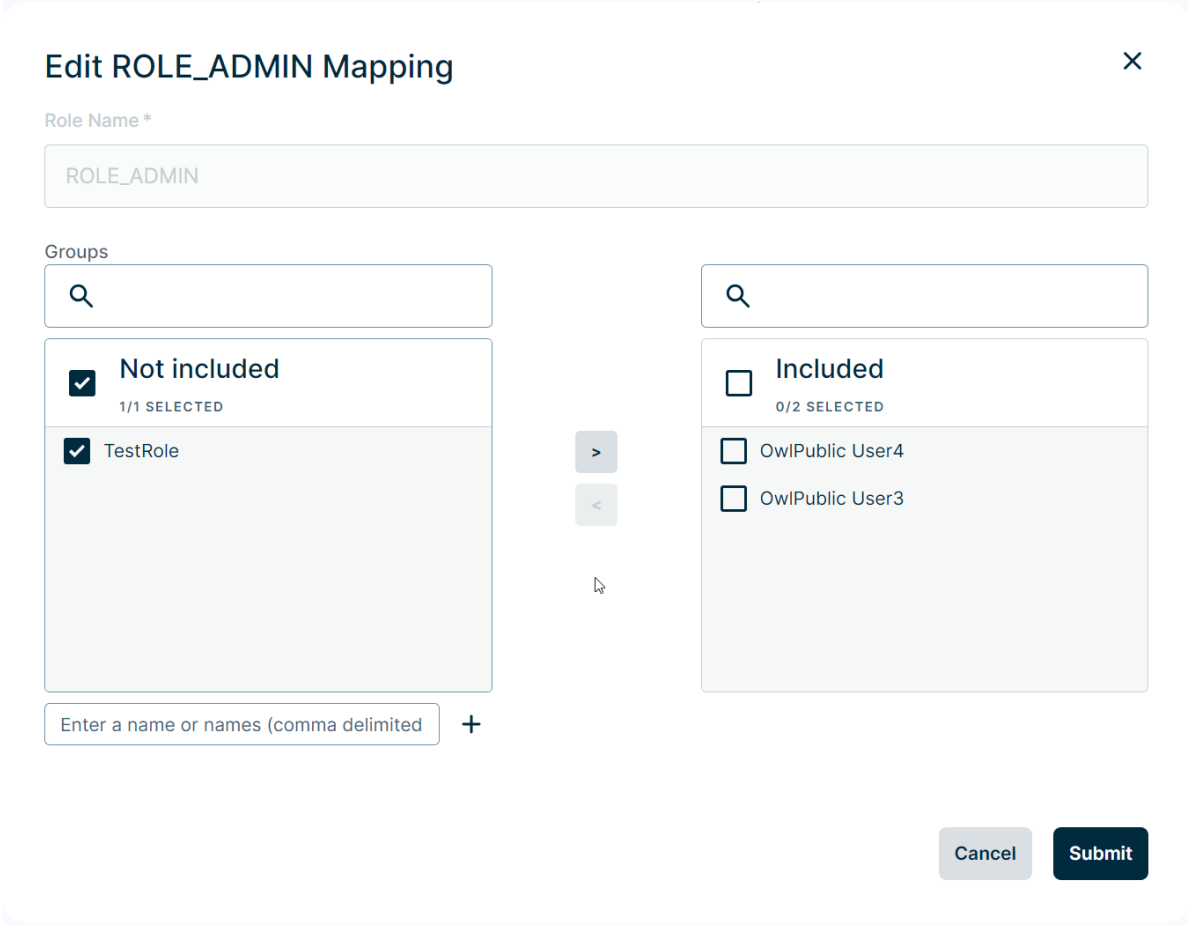
If you are configuring SSO with SAML, you can map your groups to Collibra DQ roles without Active Directory configuration.

> **Note** To enable SAML, go to SAML Authentication.

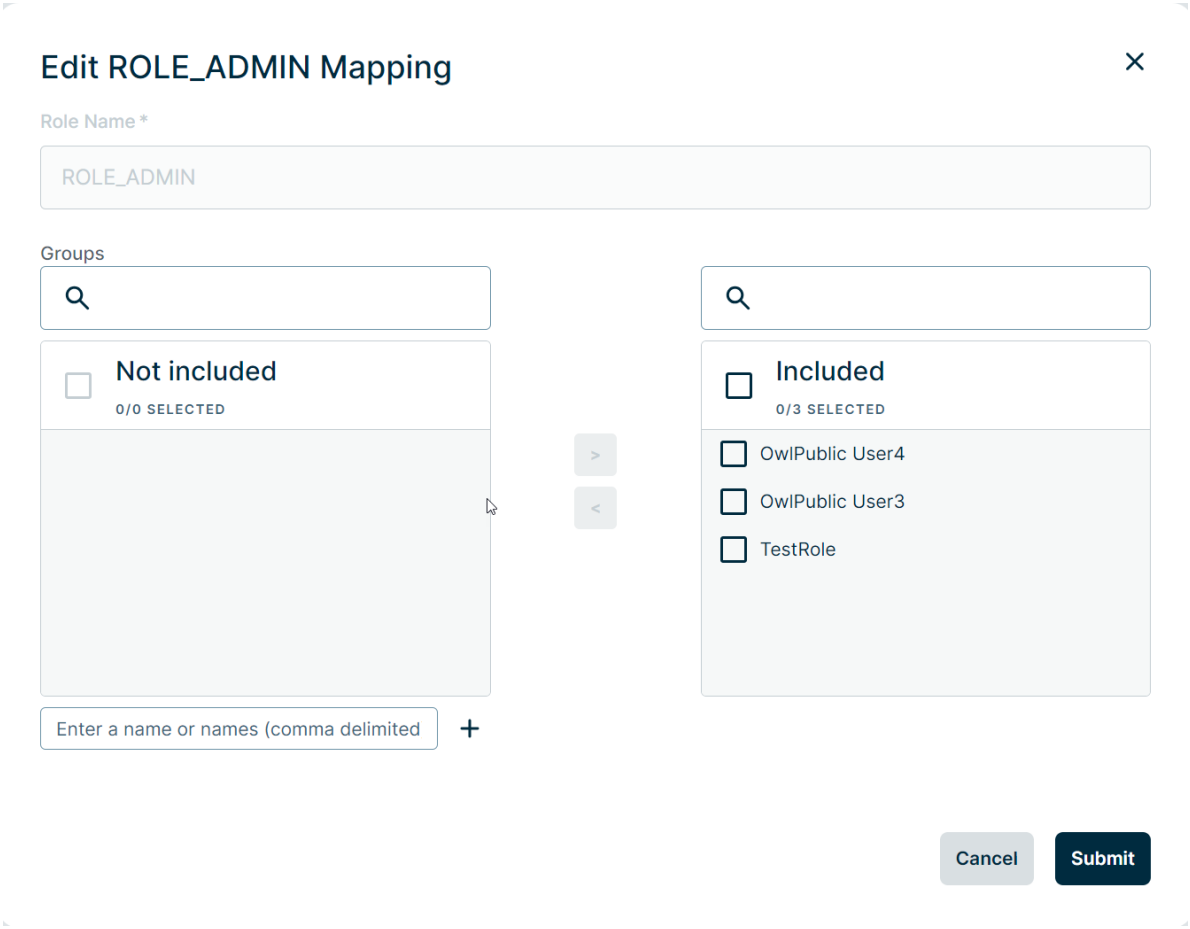
Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Select **Admin Console**.
3. From User Management in the left navigation panel, select **AD Security**.
4. Select the **Mappings** tab.
5. To map a group to an existing role, click the icon for that role, in the **Action** column. The edit role mapping window displays for that specific role. In this window, you can map groups to Collibra DQ roles by moving the groups from the **Not included** box to the **Included** box. To do this, click the and icons. You can also filter the groups and roles in the search fields.
6. Enter the name of a group in the input field below the **Not included** box and click the icon.




Edit ROLE_ADMIN Mapping
Role Name *
ROLE_ADMIN
Groups
Not included
1/1 SELECTED
TestRole
Included
0/2 SELECTED
OwlPublic User4
OwlPublic User3
>
<
Enter a name or names (comma delimited
Cancel
Submit

7. You can now click the icon to map this group to the role.



8. When you have finished adding your group(s), click **Submit**.
   Your new group(s) now appears in the Groups column for the selected role and users who are members of the group(s) should now have the permissions for the role.



Configuring user search for LDAP

This section describes how to set up user search for LDAP.

Configuring user search filter

To configure the user search filter for LDAP, you must configure the **User Search Filter** and **Group Search Base** properties on the Active Directory Security Settings page. Go to Configuring Security Settings (AD/LDAP) for details about the security settings for LDAP and Active Directory.

Steps

1. Sign in to Data Quality & Observability Classic and click in the left navigation pane.
2. Select **Admin Console**.
3. From User Management in the left navigation panel, select **AD Security**.
4. Select the **Connections** tab.
5. From the Connection Type dropdown menu, select **LDAP**.
6. In the **User Search Filter** input field, enter the username that was used at login. This will query the LDAP provider.

   > **Note** The filter should include an objectClass and an additional attribute to search for. It should also include a search term, such as **uid={0}**. Go to Configuring Security Settings (AD/LDAP).

7. In the **Group Search Base** field, enter the base path plus the OU where the groups are located.

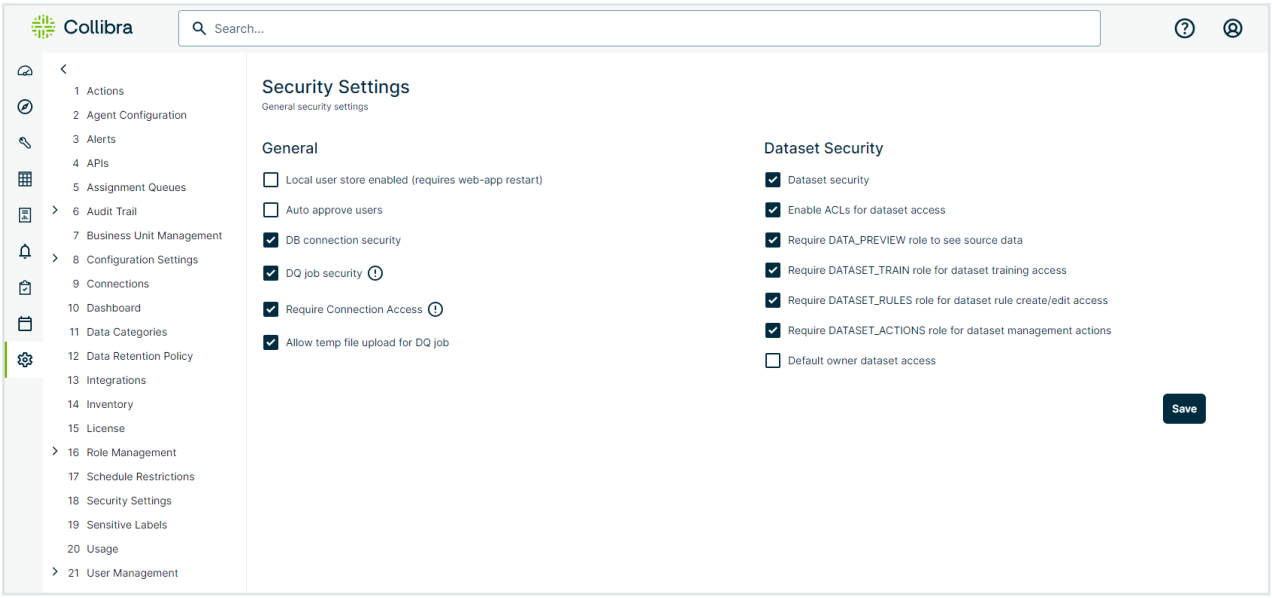
   > **Note** This value is used after the user is found to exist in the LDAP, after searching with the user search filter. Go to Configuring Security Settings (AD/LDAP).

8. Configure the following values in either owl-env.sh or ConfigMap:
   - **LDAP_ROLES_PROP_NAME**
   - **LDAP_USER_SEARCH_ATTRIBUTE**

   For the details about the owl-env.sh script settings for Search Configuration, go to Configurations in owl-env.sh or ConfigMap.

Security Settings

Security Settings lets admin users configure general and dataset security settings in Data Quality & Observability Classic.



General

Setting	Description
Local user store enabled	Allows you to create and administer users stored in the internal Metastore. After enabling or disabling this setting, you must restart Data Quality & Observability Classic. For more information, go to Local User Store Authentication.
Auto approve users	Automatically approves local users upon sign-in. When enabled, the accounts of new users are enabled, unlocked, and have ROLE_PUBLIC automatically assigned to them. When disabled, the accounts of new users are disabled, locked, and do not have any roles assigned to them. Another user with ROLE_ADMIN, ROLE_USER_MANAGER, or ROLE_OWL_ROLE_MANAGER must grant the new user account at least one role before the new user can sign in to Data Quality & Observability Classic.

<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>DB connection security</td><td>Allows admin users to select and map roles to connections.

Note Any user with those mapped roles can see the connection in the Data Quality & Observability Classic application.</td></tr>
<tr><td>DQ Job security</td><td>Makes ROLE_OWL_CHECK a requirement for users to run DQ Jobs.

Note ROLE_ADMIN and ROLE_OWL_CHECK are the only roles that can run DQ Jobs when DQ Job security is enabled.

Show table
<table><tr><th>DQ Job security</th><th>Dataset security</th><th>User role</th><th>DQ Job access level</th></tr><tr><td>Enabled</td><td>Disabled</td><td>No role</td><td>• Cannot create
• Cannot run
• Cannot schedule</td></tr><tr><td>Enabled</td><td>Enabled</td><td>A role other than ROLE_ADMIN or ROLE_OWL_CHECK assigned to the dataset</td><td>• Cannot create
• Cannot run
• Cannot schedule</td></tr><tr><td>Enabled</td><td>Enabled</td><td>ROLE_ADMIN and/or ROLE_OWL_CHECK</td><td>• Can create
• Can run
• Can schedule</td></tr></table></td></tr>
<tr><td>Require Connection Access</td><td>When DQ Job security and Require Connection Access are enabled, ROLE_ADMIN or ROLE_OWL_CHECK must be assigned to both the user and connection for the user to run, schedule, and create DQ Jobs on the connection.</td></tr>
<tr><td>Allow temp file upload for DQ Job</td><td>Allows users to upload temporary files from their local machines to use as data sources in Explorer.</td></tr>
</table>

Setting	Description
Set token expiration duration	Sets the expiration duration of the authentication token for DQ client API calls. If an API call is made and the token has reached the time limit, the API response is 401. The default setting is one hour. To change the duration, modify the following command in `owl-env.sh`: `export JWT_TOKEN_EXPIRY_TIME=3600000` The time is specified in milliseconds (ms). In this example, 3600000 milliseconds=60 minutes.

Dataset Security

Setting	Description
Dataset security	When enabled, if a user does not have a full permission role granting them the ability to perform specific tasks, they cannot perform them. The following list shows specific tasks that dataset security can prevent users without the correct full permission role from performing: • Run DQ Jobs • Manage rules • View DQ Job findings • Preview data • Take action on DQ Job findings • Train datasets • Mask columns • View assignments • Assign business units and data categories **Note** In the list above, a single role encompassing all tasks does not exist. Instead, multiple roles must be mapped to relevant datasets as full permission roles to allow users to perform all listed tasks. Users with ROLE_ADMIN or ROLE_DATASET_MANAGER can map roles to datasets as full permission roles on the Dataset Management page.

Setting	Description
Enable ACLs for dataset access	When access control list (ACL) is enabled, administrators can configure the following options to limit usage/permissions on datasets: • DATA_PREVIEW • DATASET_TRAIN • DATASET_RULES • DATASET_ACTIONS
Require DATA_PREVIEW role to see source data	When enabled, only users with ROLE_DATA_PREVIEW can preview source data on the Explorer, Findings, Profile, Rules, and Dataset Overview pages.
Require DATASET_TRAIN role for dataset training access	When enabled, only users with ROLE_DATASET_TRAIN can train datasets, which includes the ability to adjust the scoring model, modify thresholds of Adaptive Rules, and validate, invalidate, or resolve data quality findings.
Require DATASET_RULES role for dataset rule create/edit access	When enabled, only users with ROLE_DATASET_RULES can manage rules. **Note** ROLE_DATA_PREVIEW is also required to manage rules.
Require DATASET_ACTIONS role for dataset management actions	When enabled, users with ROLE_DATASET_ACTIONS are granted permissions to edit, rename, publish, assign data categories and business units, enable integrations, and schedule DQ Jobs from the Dataset Manager. These permissions are also accessible by users with the ROLE_ADMIN and ROLE_DATASET_MANAGER roles.
Default owner dataset access	An access control that, when enabled, restricts the ability to access datasets to users who are not the creator or owner of them.

Dataset Masking

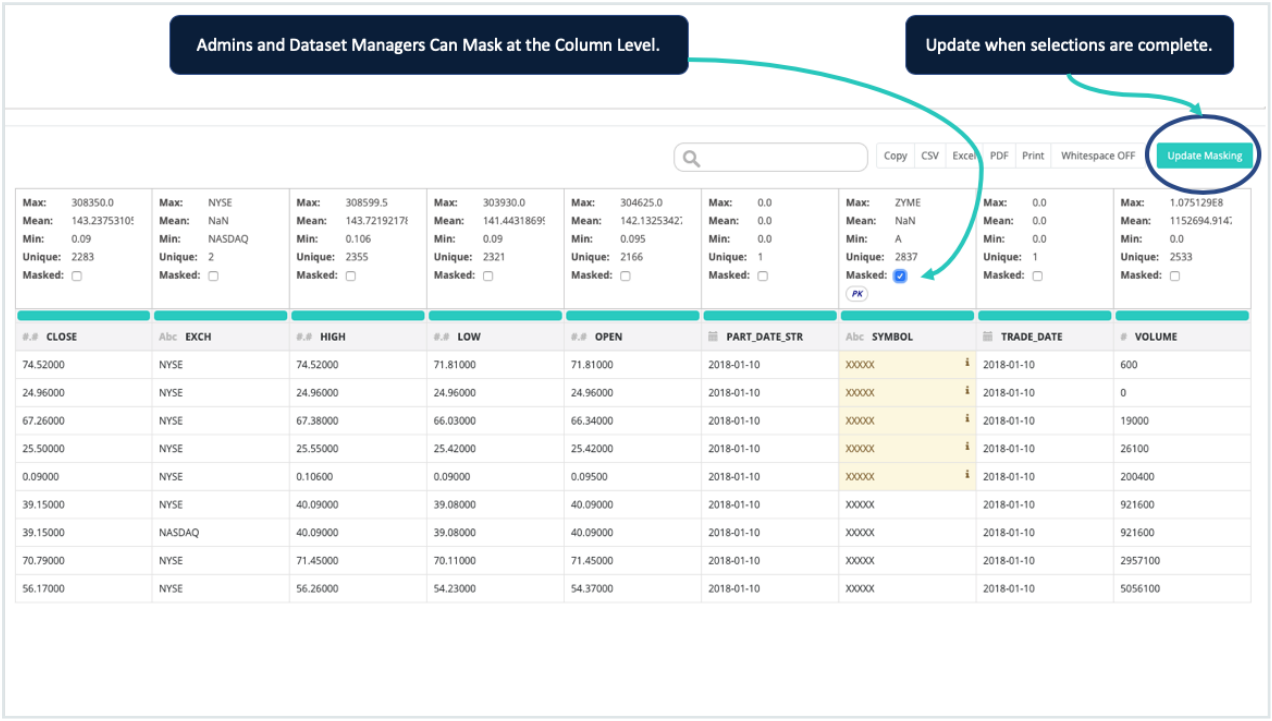
Add column level masking for sensitive data in a dataset.

Masking from the UI

After a DQ check runs, you can perform column level masking from the findings page if you have ROLE_ADMIN or ROLE_DATASET_MANAGER assigned to your user.

> **Note** Masking updates will take effect for all existing and future runs of the selected dataset.

> **Note** ROLE_ADMIN and ROLE_DATASET_MANAGER also have the ability to unmask.

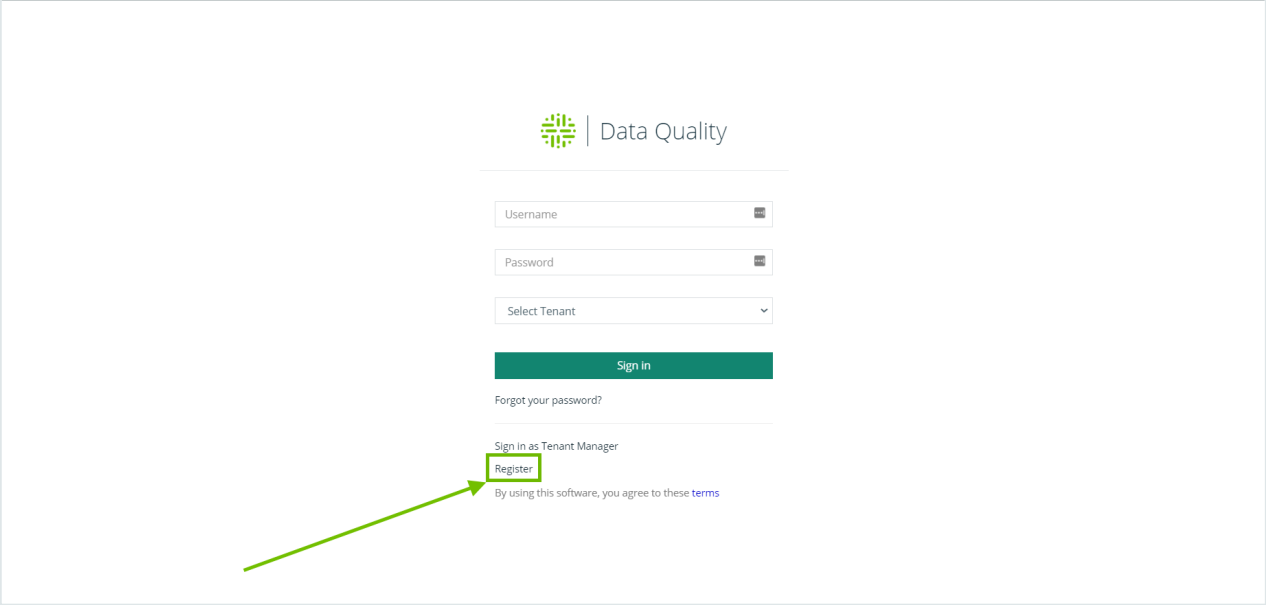


Local User Store Authentication

Local user store authentication is enabled by default, and the Data Quality & Observability Classic ships with a default user with admin privileges. If you should need to enable/disable this mechanism, you can find the setting by navigating to the Admin Console and clicking on the **Security** icon. The setting labeled **Local User Store Enabled**, when selected, allows your company to create and administer users stored in Collibra Data Quality's internal database user store. Enabling/disabling this feature requires a restart of the web application.

Adding Local Users

Users can request access to the application from the login page by clicking the **Register** link beneath the **Sign in** button.



Enter your details into the account registration form to create a new user record. At this stage, the new user account does not have access to the application until an application administrator grants it access to one or more Collibra DQ roles.

> **Note** If LDAP is enabled and a local user password or username is invalid and fails to authenticate, Collibra DQ attempts to authenticate against the AD/LDAP server.

Granting Local User Roles

Application administrators can manage user accounts and role access by navigating to the Admin Console and clicking the **Users** tile. On that page, locate the user you would like to modify and click their username in the table of users. This loads a pop-up window that allows you to modify the user account, including Enabled/Disabled, as well as the roles they belong to. Click the **Roles** tab to begin granting roles to the selected user. The available roles established are listed on the left. Click a role from the left-hand side to apply that role to the user. You can also add all roles by clicking the double arrow icon above the list. When you are satisfied with the role listings on the right-hand side for that user, click the **Update** button.

Known limitations

- When using the latest UI, admin users cannot add local users from the User Management page.
- When editing a user from the Edit [username] modal, the name input fields cannot contain spaces. As a workaround, you can use an underscore instead.

What's next?

- For details on administering user accounts from the Admin Console, go to User Management.
- For additional user configuration options, go to Security Settings.

SAML Authentication

You can integrate Data Quality & Observability Classic with an existing SAML solution and have your application act as a service provider. Once you set up the environment variables, you can access and configure SAML security settings as an administrator in the **SAML Setup** section of the **Admin Console**.

Set the SAML authentication properties

Before configuring SAML authentication, you must add the following required properties to your configuration.

Standalone installation

1. Add the properties as environment variables to your **owl-env.sh** file located in **<installation_directory>/owl/config/**.
2. Prefix all properties with the `export` statement.
3. Restart the DQ Web app.

Cloud native installation

1. Add the properties as environment variables to your **owl-web ConfigMap**.
2. Recycle the pod.

Required properties

Property	Description
SAML_ENABLED	Whether Collibra DQ uses SAML. If set to `false`, users sign in with a username and password. If set to `true`, SAML handles the authentication request.
SAML_ENTITY_ID	The name of the application for the identity provider, for example *Collibra DQ*. It is an immutable unique identifier of the service provider for the Identity Provider (IdP).
SAML_ENTITY_BASEURL	The base URL that is provided in the service provider metadata. Set this property when you use DNS.

Warning See CORS_ALLOWED_ORIGINS in the Optional properties section below if you have SAML configured in DQ, or if the app sits behind a load balancer.

Optional properties: general

You can further configure your SAML setup with the following optional properties.

Property	Description
CORS_ALLOWED_ORIGINS	Allows cross-origin requests between Collibra DQ and SAML. Replace `${IDP-BASE-URL}` with the value of the actual IdP URL. For example, `https://ping.auth.com/`. Replace `${DQ-BASE-URL}` with the value of the actual DQ Base URL. For example, `https://dq-env.com`.
SAML_LB_EXISTS	Whether the application needs to configure a load balancer. You generally need this setting only when the **Load Balancer** is set for **SSL Termination**. The default value is `false`. If set to `true`, you must also provide a value for **SAML_LB_SERVER_NAME**.
SAML_ROLES_PROP_NAME	The attribute in which the identity provider stores the role of the user authenticating in the SAML response. The default value is `memberOf`.
SAML_GRANT_ALL_PUBLIC	Whether any user authenticated by the identity provider is allowed to login the Collibra DQ application. The default value is `true`.
SAML_USER_NAME_PROP	The name of the attribute in the SAML response that contains the username of the user who is authenticating.
SAML_TENANT_PROP_NAME	If using multi-tenant mode, the variable in which the identity provider stores the tenant name of the user authenticating in the SAML response. The app will attempt to use the `RelayState` parameter to identify the tenant and then fall back on this property.
SAML_KEYSTORE_FILE	The path to the keystore for SSL validation. The store should contain the keypair of the identity provider for SSL verification.
SAML_KEYSTORE_PASS	The password for the keystore provided in `SAML_KEYSTORE_FILE`.

Property	Description
SAML_KEYSTORE_ALIAS	The alias of the keypair (private and public) in the keystore used for SSL verification.
LOCAL_REGISTRATION_ENABLED	An optional property which allows you to display or hide the registration link for local users on the Collibra DQ sign-in page. Because the registration link is displayed on the sign-in page by default, this property is set to true. To display or hide the registration link from the owl-env.sh file, select from the following properties: **Property** / **Description** export LOCAL_REGISTRATION_ENABLED=true — Displays the registration link on the sign-in page for local users. This is set to `true` by default. export LOCAL_REGISTRATION_ENABLED=false — Hides the registration link on the sign-in page for local users. To display or hide the registration link from a K8s ConfigMap, select the from the following properties: **Property** / **Description** LOCAL_REGISTRATION-ENABLED:"true" — Displays the registration link on the sign-in page for local users. This is set to `true` by default. LOCAL_REGISTRATION-ENABLED:"false" — Hides the registration link on the sign-in page for local users. **Note** This property must be set globally on deployment.

Note While CORS is still an optional configuration, it is required if you have SAML configured in Collibra DQ, or if you have Collibra DQ behind a load balancer.

CORS is also enforced for multi-tenancy.

Download service provider metadata for the IdP

Once you have enabled and configured SAML authentication, you can download the service provider metadata that is required by your identity provider from `https://<your_dq_environment_url>/saml/metadata/SSO`.

> **Important** If you use Data Quality & Observability Classic with a Java version older than 17, such as Java 11, download your service provider metadata from `https://<your_dq_environment_url>/saml/metadata` instead.

Enable the SAML SSO sign in option

When you are ready with your IdP settings, add the final configuration settings in the **Admin Console**:

1. Sign in to Data Quality & Observability Classic as an administrator with a username and password to the tenant you want to configure.
2. Click **Admin Console**.
3. Click **User Management**, then click **SAML Security**.
   » The SAML Security Settings page opens.
4. Click the **SAML Enabled** dropdown menu, then select **Enabled**.
5. Under the SAML Enabled dropdown menu, to the far right side of the page, click .
   » The Add Meta Data Configuration dialog box appears.
6. Enter the required information.

Option	Description
Meta-Data URL	The URL of the identity provider metadata XML file or the location of the downloaded XML file, depending on how you configured the SAML_METADATA_USE_URL property. **Important** The name of the XML file must begin with the prefix **file:** and typically uses **file:/opt/owl/config/idp-metadata.xml** format.
Meta-Data Label	The name for this specific configuration.
IDP URL	The URL of the Collibra DQ application that is provisioned by the identity provider.

7. Click **Add**.

After you complete this setup, you can restart your application and sign in using the SAML SSO option.

> **Note** SAML SSO authentication via the /v3/auth/signin API is not supported.

> **Note** SAML authentication is only supported for IdP-initiated requests.

Configuring multi-tenant support through SAML RelayState

In multi-tenant environments, you can route SSO to the proper tenant with the SAML provided `RelayState` or `SAML_TENANT_PROP_FROM_URL` properties.

While in a multi-tenant environment, if you set the `SAML_TENANT_PROP_FROM_URL` property to `true`, the application attempts to use the `RelayState` parameter to identify the tenant. When you set `SAML_TENANT_PROP_FROM_URL` to `false`, Collibra DQ attempts to find the tenant in a SAML response attribute, which is defined by the `SAML_TENANT_PROP_NAME` property (default=`tenant`).

You can set the `RelayState` property on the **SAML Security** page of the **Admin Console**.

For more information, go to Set the SAML authentication properties.

Configuring SAML SSO for Azure

When using SAML SSO with Azure as the provider, some additional configuration is required to read the groups and map roles.

When groups are pulled from Azure Active Directory SSO and more than five groups are assigned to a user, the group claims return as a link, rather than the groups list:

```
<Attribute Name="http://schemas.microsoft.com/claims/groups.link">
```

> **Note** Go to https://learn.microsoft.com/en-us/azure/active-directory/develop/reference-saml-tokens for more information about SAML security tokens

> **Note** For customers who have setup and configured the application to return groups that are only present in the SSO application, this issue is not applicable.

When Azure SSO sends a group.link assertion, the application tries to resolve the groups via the link. To activate the link, you must configure the `SAML_GROUPS_LINK_PROP` property.

Steps

1. Add the following property to your **owl-env.sh** file:

Property	Description
SAML_GROUPS_LINK_PROP	Configure this property equal to the group link claim name. Go to SAML Authentication for more information about configuring SAML authentication properties.

2. Obtain a client key from Azure and configure both `AZURE_CLIENT_SECRET` and `AZURE_CLIENT_ID` properties, using the following documentation:

   https://learn.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal#option-2-create-a-new-application-secret

   > **Note** You must also ensure that the application, not delegate, has the MS Graph API application permission **directory.read.all**.

   > **Note** In the IdP setup in Azure, verify that the Sign-on URL is not populated.

Migrating to a custom SAML keystore

To improve the security of Data Quality & Observability Classic, we removed the default keystore password from the installation packages in the Data Quality & Observability Classic 2024.07 release. This page shows you how to migrate to a custom SAML keystore.

Depending on whether your organization has SSL enabled, there are two options for configuring a custom SAML keystore.

1. In the owl-env.sh for Standalone or Web ConfigMap for Cloud Native deployments of Data Quality & Observability Classic, set the SAML properties for the keystore the same as the SSL keystore file. For example:

```
#enable SAML & show the SAML SSO option on the login page
SAML_ENABLED=true

#set SSL communication properties for SAML
SAML_KEYSTORE_FILE=file:/etc/ssl/dev-keystore.jks
SAML_KEYSTORE_PASS=****
SAML_KEYSTORE_ALIAS=****

#set the name of the IdP variable to hold the user roles
in the response
SAML_ROLES_PROP_NAME=memberOf

#set the EntityId of the application to be supplied to
the IdP
SAML_ENTITY_ID=OwlOneLogin

#optionally use a file for the IdP metadata vs a URL
(default is true)
SAML_METADATA_USE_URL=false
```

Variable	Description
`SAML_ENABLED`	Whether Data Quality & Observability Classic uses SAML. If set to `false`, users sign in with a username and password. If set to `true`, SAML handles the authentication request.
`SAML_KEYSTORE_FILE`	This path can be changed to any path as long as it contains the public and private key combination to be used by Data Quality & Observability Classic and necessary certificates required for SAML.
`SAML_KEYSTORE_PASS`	The password used while creating the public and private key for the keystore file.

Variable	Description
`SAML_KEYSTORE_ALIAS`	The alias used while creating the public and private key for the keystore file.
`SAML_ROLES_PROP_NAME`	The name of the IdP variable to hold the user roles in the response.
`SAML_ENTITY_ID`	The unique identifier for the SP entity that must match the SP entity ID used in the IdP circle of trust configuration.
`SAML_METADATA_USE_URL`	Whether Collibra DQ uses a URL or a file for the identity provider metadata. Set to `false` for file-based SAML and `true` for URL-based configurations.

2. Start the Data Quality & Observability Classic application.
3. (Required only if signing or encrypting SAML assertion) Download the Data Quality & Observability Classic SP metadata from the **https://<your_dq_environment_url>/saml/metadata** endpoint and share it with the IdP to configure the SP.

Important When using SAML in its current implementation, you must enable SSL to use a custom keystore. If you rely on a Load Balancer to manage SSL, you can still enable SSL by stopping SSL at the Load Balancer. You can also enable an HTTP port to allow a Load Balancer to connect to it, as shown in step 3 on the **Without SSL enabled** tab of the Migrating to a custom SAML keystore topic. In a future release, we plan to provide support for custom keystores without SSL enabled.

1. Create a keystore file, as demonstrated in Setting up SSL (HTTPS).
2. For Cloud Native deployments of Data Quality & Observability Classic:
    a. Create a K8s secret using the keystore file:

```
kubectl create secret generic dq-ssl-secret --from-
file <file-name.jks> --namespace <dev>
```

    b. Mount the JKS file through the following Helm Charts variable:

```
--set global.web.tls.enabled=true --set
global.web.security.saml.keyStoreFile=file:///etc/ss
l/dqkeystore.jks --set
global.web.security.saml.keyStoreAlias=<alias-name>
--set
global.web.security.saml.keyStorePass=<keystore-
pass>
```

3. Start the Data Quality & Observability Classic application.
4. Download the Data Quality & Observability Classic SP metadata from the **https://<your_dq_environment_url>/saml/metadata** endpoint and share it with the IdP to configure the SP.

Securing Passwords

Security is of the utmost importance for Collibra DQ and our customers. To avoid sending plain text passwords when you run DQ jobs from the command line, you can encrypt passwords instead. To encrypt your password, execute the following command:

```
owlmanage.sh encrypt=password
```

The output password should look similar to the following example:

```
Q+Ri1S+ljpG+fDefXLY4/vXtUosspAoL
```

You can use this password in any DQ job from the command line where you would normally use a plain text password.

The following is an example of a DQ job with an encrypted password instead of a plain text password:

```
./owlcheck -q "SELECT id, browser->'$.name' browser FROM events"
-c "jdbc:mysql://54.212.36.218:2212/test" -u "owl" -p
"Q+Ri1S+ljpG+fDefXLY4/vXtUosspAoL" -driver "com.mysql.cj.jd-
bc.Driver" -lib "/opt/owl/drivers/mysql8" -ds jsonremotemysql -
rd "2022-07-25"
```

If you run a DQ job from within the DQ Web UI, it automatically encrypts your password, eliminating the need to manually encrypt it. For added security, all passwords are masked in the logs and plain text passwords are never stored.

Securing passwords in a Kubernetes environment

To ensure that your password is secure in a Kubernetes environment, you must use Kubernetes Secrets. A Secret stores sensitive data, for example, a username or password, which are used by pods to access services.

You can create the Secret by using either of the following methods:

- Pass the raw data in the command.
- Store the credentials in files that you pass in the command.

The following commands create a Kubernetes Secret that stores the username `admin` and the password `S!B\*d$zDsb=`.

Create Secrets using raw data

Execute the following command:

```
kubectl create secret generic db-user-pass \
                --from-literal=username=admin \
            --from-literal=password='S!B\*d$zDsb='
```

> **Note** To prevent your shell from interpreting special characters, such as `$`, `\`, `*`, `=`, and `!` in your strings, you must use single quotes `''` to escape those characters.

Create Secrets using source files

You can create Kubernetes Secrets by using source files.

Steps

1. Store the sensitive data/credentials in files:

```
echo -n 'admin' > ./username.txt
            echo -n 'S!B\*d$zDsb=' > ./password.txt
```

In this command, the `-n` flag ensures that the generated files have no newline characters at the end of the text. When `kubectl` reads a file and then encodes the content into a base64 string, the extra newline character also gets encoded. You don't need to escape special characters in strings that you have included in a file.

2. Pass the file paths in the `kubectl` command:

```
kubectl create secret generic db-user-pass \
                --from-file=./username.txt \
            --from-file=./password.txt
```

The file name in this case is the default key name. You can also set the key name using the following command:

```
kubectl create secret generic db-user-pass \
                --from-file=username=./username.txt \
            --from-file=password=./password.txt
```

With either method, the output is similar to the following:

```
secret/db-user-pass created
```

Verify the Kubernetes Secret

Now that you have created the Secret, you should verify that it was created with the following command:

```
kubectl get secrets
```

The output is similar to the following:

```
NAME             TYPE             DATA             AGE
            db-user-pass     Opaque           2                51s
```

Now, view the details:

```
kubectl describe secret db-user-pass
```

The output is similar to the following:

```
Name:             db-user-pass
                Namespace:        default
                Labels:           <none>
                Annotations:      <none>

                Type:             Opaque

                Data
                ====
                password:    12 bytes
            username:    5 bytes
```

By default, the kubectl get and kubectl describe commands avoid showing the contents of a Secret. This protects the Secret from being exposed by mistake or from being stored in a terminal log.

Decode the Kubernetes Secret

You should now decode the Secret.

Steps

1. View the contents of the secret with the following command:

```
kubectl get secret db-user-pass -o jsonpath='{.data}'
```

The output is similar to the following:

```
{ "password": "UyFCXCpkJHpEc2I9", "username": "YWRtaW4=" }
```

2. Decode the `password` data with the following command:

```
echo 'UyFCXCpkJHpEc2I9' | base64 --decode
```

The output is similar to the following:

```
S!B\*d$zDsb=
```

Warning This particular example is strictly for documentation purposes only. This method could cause the command with the encoded data to be stored in your shell history, meaning that anyone with access to your computer could potentially locate the command and decode the secret. Therefore, you should combine the view and decode commands, as shown in the following command:

```
kubectl get secret db-user-pass -o jsonpath='{.data.password}'
| base64 --decode
```

Setting up SSL (HTTPS)

By default, Collibra DQ has plain HTTP enabled for testing. When you are ready to enable SSL for the web application, you can set the following environment variables in the **owl-env.sh** file for Standalone deployments or the **Web ConfigMap** for Cloud Native to enable HTTPS.

Note The URL you are using must include a fully qualified domain name.

The settings listed at the bottom of this page enable secure HTTPS and allow you to point to your certificate keystore and credentials. A restart of the web application is required.

> **Important** You must restart the web application from the bin directory: `./owlmanage.sh restart=owlweb` (Standalone) or Web Pod (Cloud Native).

Supported keystore formats

Data Quality & Observability Classic supports the following keystore formats:

- Java KeyStore (JKS): The default keystore format for Java, which stores private keys and certificates in a proprietary format.
- PKCS12: A more universal keystore format that is widely supported across different systems and applications, and is often used for interoperability with non-Java applications.

Create a Java KeyStore

Before starting the SSL setup, you need to generate a Java KeyStore. A Java KeyStore stores private keys and certificates.

You can use a `keytool` JDK utility to generate a keystore, which helps to manage private and public keys and associated certificates. With this utility, you can administer your own public or private key pairs and associated certificates. The keystore allows you to protect your private keys with a password.

All certificates within the keystore are associated with a unique alias that gets used as a pointer to perform keytool operations, such as import, export, delete, and/or change certificates and keys.

> **Note** Implementation is dependent upon your specific architecture.

The following example is a command that creates a keystore.

```
keytool -genkeypair -alias mykey -keyalg RSA -keysize 2048 -key-
store mykeystore.jks -dname "CN=Your Name, OU=Your Org Unit,
O=Your Org, L=Your City, ST=Your State, C=Your Country" -valid-
ity 365
```

The following table breaks down the various options in the above command.

Options	Description
`-alias`	The alias name for the key entry.
`-keyalg`	The algorithm to use for the key pair. In this example, the algorithm in use is RSA.
`-keysize`	The key size in bits, usually 2048.
`-keystore`	The file name of the keystore. In this example, the command generates a keystore called **mykestore.jks** with a single, password-protected key entry.
`-dname`	The distinguished name for the certificate.
`-validity`	The validity period for the certificate in days.

(Optional) Generate a Certificate Signing Request from the keystore

If you need to obtain a certificate signed by a Certificate Authority (CA), generate a Certificate Signing Request (CSR) using a command similar to the following example:

```
keytool -certreq -alias mykey -keystore mykeystore.jks -file
myrequest.csr
```

> **Note** When prompted, enter the alias name and password.

You can send myrequest.csr to a CA to get it signed.

Import a signed certificate

After a CA signs the certificate, you can import it into the keystore using a command like the following:

```
keytool -importcert -alias mykey -keystore mykeystore.jks -file
mycert.cer
```

For more information on importing certificates in cloud native deployments, go to Add CA certificates to the java keystore.

Verify the keystore

After you import the signed certificates, you can verify the contents of the keystore with the following command:

```
keytool -list -v -keystore mykeystore.jks
```

This command shows all the certificates and key pairs within the keystore.

Using a keystore or truststore

Keystores and truststores can be the same file, but they serve different purposes in Java's security infrastructure.

Purpose of a truststore

When an application (client or server) establishes a secure connection, it needs to verify the identity of the other party. This is done by checking the certificate of the other party against a list of trusted certificates stored in the truststore. If the CA-signed certificate is trusted and present in the truststore, the connection is considered secure.

Purpose of a keystore

When a server hosts a secure service, for example, HTTPS, it uses a keystore to house its private key and the associated public certificate. This certificate is presented to clients during the SSL/TLS handshake to prove the server's identity. In some cases, such as mutual auth, clients also need to authenticate themselves to the server. A client-side keystore can store the client's private key and certificate for this purpose.

Configure a keystore and truststore in Collibra DQ

In Data Quality & Observability Classic, you typically specify the keystore and truststore with the environment variable `EXTRA_JVM_OPTIONS`:

```
-Djavax.net.ssl.keyStore=path/to/keystore.jks -
Djavax.net.ssl.keyStorePassword=keystorepassword -
Djavax.net.ssl.trustStore=path/to/truststore.jks -
Djavax.net.ssl.trustStorePassword=truststorepassword
```

If you use the same file for both, set the following options:

```
-Djavax.net.ssl.keyStore=path/to/store.jks -
Djavax.net.ssl.keyStorePassword=password -
Djavax.net.ssl.trustStore=path/to/store.jks -
Djavax.net.ssl.trustStorePassword=password
```

Depending on your deployment type, the location where this configuration is set differs. In Cloud Native deployments, this is set in the ConfigMap. In Standalone deployments, this is set in the owl-env.sh file.

Generate a PCKS12 file using the public and private key from CA

You will receive a bundle from CA containing your public and private key, which you can then export by using a command such as the following.

```
export SERVER_SSL_KEY_STORE=<path to your keystore>
```

You can call Collibra DQ's built-in 256-bit encryption for the SERVER_SSL_KEY_PASS value from the bin directory: `./owlmanage.sh encrypt=<sensitive plain text string>`. Use the response value instead of the plain text value to secure your password.

> **Note** Encrypting the SSL key pass is only required for Standalone deployments. However, you can optionally set the `SERVER_SSL_KEY_PASS` property as a Kubernetes Secret for Cloud Native deployments.

```
export SERVER_SSL_KEY_PASS=<secure result from dq encryption
script>
```

Define the SSL properties

Add the following values to the `owl-env.sh` file.

```
export SERVER_HTTP_ENABLED=false
export SERVER_HTTPS_ENABLED=true
export SERVER_REQUIRE_SSL=true

####START KEYSTORE SETTINGS####
export SERVER_SSL_KEY_TYPE=PKCS12
#SET PATH TO KEYSTORE
export SERVER_SSL_KEY_STORE=KeystorePathHere
export SERVER_SSL_KEY_PASS=*******
export SERVER_SSL_KEY_ALIAS=keystoreAliasNameHere
```

> **Note** The most common SSL types are JKS and PKCS12.

Add the following values to the Web ConfigMap.

```
SERVER_HTTP_ENABLED: false
SERVER_HTTPS_ENABLED: true
SERVER_REQUIRE_SSL: true
SERVER_SSL_KEY_TYPE: PKCS12
SERVER_SSL_KEY_STORE: KeystorePathHere
SERVER_SSL_KEY_PASS: *******
SERVER_SSL_KEY_ALIAS: keystoreAliasNameHere
```

Setting up a custom keystore in Collibra DQ

This topic shows you how to set up the keystore and truststore file for the Data Quality & Observability Classic application to use across various deployment types.

1. Create a Java KeyStore, as demonstrated in Setting up SSL (HTTPS).
2. Copy the JKS or PKCS12 file to the location inside the VM where DQ Web is running. For example, **file:///etc/ssl/dev-keystore.jks**
3. Update the owl-env.sh with the following properties:
    a. **For SSL**

i.
```
SERVER_SSL_KEY_STORE=file:///etc/ssl/dev-
keystore.jks
```

ii.
```
SERVER_SSL_KEY_
ALIAS=alias                    #This is the
alias used when creating the keystore
```

iii.
```
SERVER_SSL_KEY_PASS=<keystore-
password>        #The password of the keystore
```

iv.
```
SERVER_SSL_KEY_
TYPE=JKS                       #This is either
JKS or PKCS12
```

    b. **For SAML**

i.
```
SAML_KEYSTORE_FILE=file:///etc/ssl/dev-
keystore.jks
```

ii.
```
SAML_KEYSTORE_
ALIAS=alias                    #This is the
alias used when creating the keystore
```

iii.
```
SAML_KEYSTORE_PASS=<keystore-
password>        #The password to the keystore
```

4. Restart the Data Quality & Observability Classic application.
   » Data Quality & Observability Classic now uses the keystore file for SSL and SAML.

> **Note** Depending on the IdP configuration you many need to re-import the SP metadata for SAML. To do this, download the Data Quality & Observability Classic SP metadata from the **https://<your_dq_environment_url>/saml/metadata** endpoint and share it with the IdP to configure the SP.

To use a custom keystore or truststore in a Cloud Native deployment requires you to mount the keystore file inside the web pod.

Enabling SSL

1. Create a K8s secret using the keystore file:

```
kubectl create secret generic dq-ssl-secret --from-file
<file-name.jks> --namespace <dev>
```

2. Enable SSL with the following Helm Charts variable:

```
--set global.web.tls.enabled=true --set
global.web.tls.key.alias=<alias-name> --set
global.web.tls.key.pass=<keystore-pass>
```

3. Start the Data Quality & Observability Classic application.

Using SAML with SSL enabled

1. Create a K8s secret using the keystore file:

```
kubectl create secret generic dq-ssl-secret --from-file
<file-name.jks> --namespace <dev>
```

2. Set the SAML keystore parameters through the following Helm Charts variable:

```
--set
global.web.security.saml.keyStoreFile=file:///etc/ssl/dqk
eystore.jks --set
global.web.security.saml.keyStoreAlias=<alias-name> --set
global.web.security.saml.keyStorePass=<keystore-pass>
```

3. Start the Data Quality & Observability Classic application.

Using SAML without SSL enabled

> **Important** When using SAML in its current implementation, you must enable SSL to use a custom keystore. If you rely on a Load Balancer to manage SSL, you can still enable SSL by stopping SSL at the Load Balancer. You can also enable an HTTP port to allow a Load Balancer to connect to it, as shown in step 3 below. In a future release, we plan to provide support for custom keystores without SSL enabled.

1. Create a K8s secret using the keystore file:

```
kubectl create secret generic dq-ssl-secret --from-file
<file-name.jks> --namespace <dev>
```

2. Set the SAML keystore parameters through the following Helm Charts variable:

```
--set
global.web.security.saml.keyStoreFile=file:///etc/ssl/dqk
eystore.jks --set
global.web.security.saml.keyStoreAlias=<alias-name> --set
global.web.security.saml.keyStorePass=<keystore-pass>
```

3. (Optional) If you want to include a non-SSL HTTP port, manually update the Web ConfigMap with the following property:

```
SERVER_HTTP_ENABLED=true
```

4. Start the Data Quality & Observability Classic application.

Add CA certificates to the java keystore

This topic describes how to add CA certificates to the java keystore in a Kubernetes deployment. The following approach mounts the directory containing the SSL certificates from the Agent pod as a shared volume. The SSL certificates are then accessible to the Spark driver/executor pods, allowing them to establish a secure SSL/TLS connection with the email server and/or data source.

Steps

1. Mount the directory containing the SSL certificates in the Agent pod to a shared volume. This can be done in the deployment or pod configuration file for the Agent pod in Kubernetes. For example, you can add a volume and volumeMounts section to the Agent pod's configuration file like this:

```
volumes:
  - name: certs
    hostPath:
      path: /etc/pki/ca-trust/extracted/java/cacerts
```

2. Modify the Spark driver/executor pod's configuration to mount the shared volume containing the SSL certificates from the Agent pod. This can be done in the Spark con-

figuration or pod configuration file. For example, you can add a volume and volumeMounts section to the Spark driver/executor pod's configuration file like this:

```
volumes:
  - name: certs
    hostPath:
      path: /path/to/agent/certs
```

3. Update the Java options for the Spark driver/executor pods to specify the path to the SSL certificates in the shared volume. You can do this by adding the `-Djavax.net.ssl.trustStore` and `-Djavax.net.ssl.trustStorePassword` options to the Spark driver/executor pod's configuration file, like this:

```
javaOptions:
  - "-Djavax.net.ssl.trustStore-
e=/path/to/agent/certs/cacerts"
  - "-Djavax.net.ssl.trustStorePassword=your_keystore_pass-
word"
```

4. Replace `/path/to/agent/certs` with the actual path to the shared volume where the SSL certificates are mounted in the Spark driver/executor pods, and `your_keystore_password` with the actual password for the keystore that contains the SSL certificates.
5. Restart the Spark driver/executor pods to apply the changes.

Integrations

Configuring integrations

This section provides information about how to integrate Data Quality & Observability Classic into Collibra Platform.

About Collibra DQ integrations

Collibra DQ Integrations allow DQ admin users to synchronize metadata from Data Quality & Observability Classic into Collibra Platform. You can use the Integration Configuration wizard for a guided mapping of objects between Collibra DQ and Collibra Platform in only a few clicks. When you integrate Collibra DQ metadata into Collibra Platform, relations between your DQ rules and the tables and columns they govern are automatically established.

By using the Integrations page, you can achieve the following benefits:

- Broader data sharing between platforms to include not only user defined rules but also adaptive rules, shapes, schema, source, record, pattern, outlier, and duplicate findings along with their scores.
- Automated synchronization between integrated platforms based on job execution and job scheduling.
- Simplified configuration and maintenance of the integration.

Prerequisites

- Your Data Quality & Observability Classic and Collibra Platform versions are 2025.02 or newer.

- You have an Admin role in Data Quality & Observability Classic to use the configuration wizard.
- You have an Admin or Dataset Manager role in Data Quality & Observability Classic to enable or disable individual datasets for integration.
- Your Collibra Platform profile must have Sysadmin and DataSteward global roles to manage all assets and objects of the integration.
- You have an aggregation path set for your Data Quality Rule in Collibra Platform ▸ Settings ▸ Operating model ▸ Data quality rules.

Version compatibility

The following table shows the compatibility of Collibra Platform and Data Quality & Observability Classic versions, including the available features when the minimum version requirements between the two platforms are met.

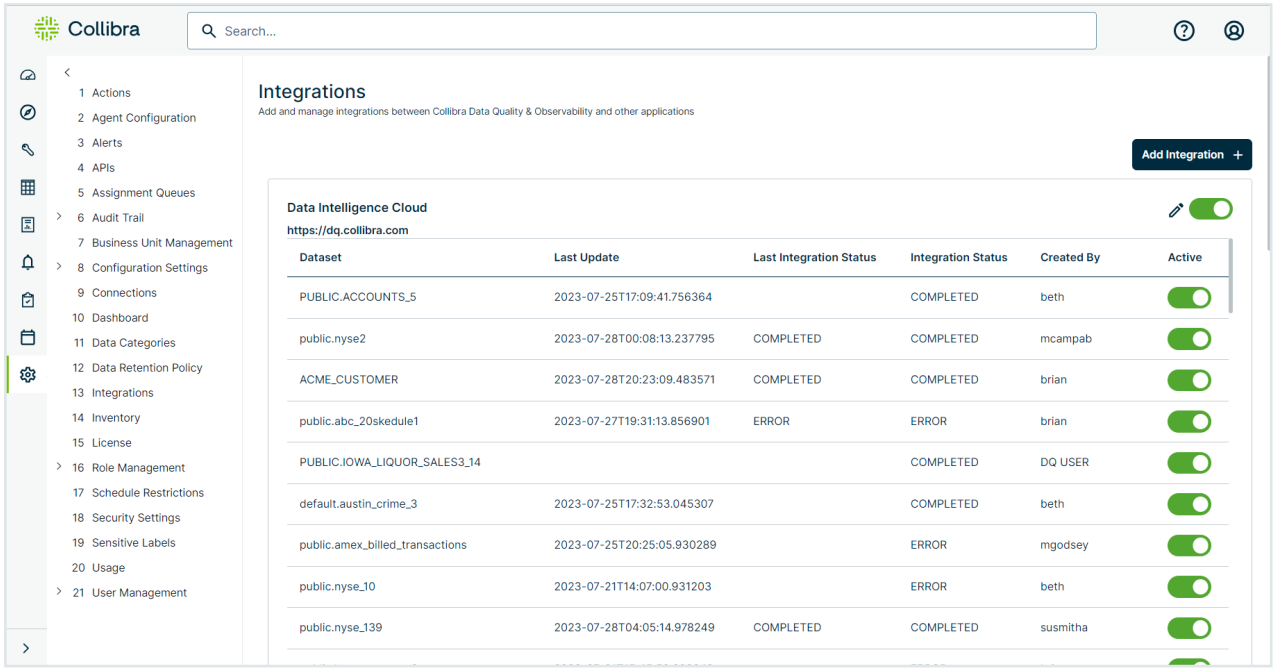
Available features and additional details	Collibra Platform minimum version	Data Quality & Observability Classic minimum version
Support for the definition of custom aggregation paths in the Collibra Platform operating model to allow you to view data quality scores for assets other than databases, schemas, tables, and columns.	2024.10	2024.07
Support for viewing data quality scores and running jobs from the Data Quality Jobs modal on asset pages.	2024.06	2024.05

> **Note** This table is not intended to provide an exhaustive list of available features. Rather, it is meant to highlight the required versions to be able to use some of the most important features. For bugfixes and other feature enhancements, see the release notes.

Integrations

From the Integrations page in the Admin Console, you can add and manage integrations between Data Quality & Observability Classic and Collibra Platform. You can conveniently

switch integrations between Collibra DQ and Collibra Platform instances on and off, and find an overview of the status of all datasets with an attempted or completed integration.



The following table describes each column of the dataset integration status table on the Integrations page.

Column	Description
Dataset	The name of the Collibra DQ dataset whose metadata integrated into Collibra Platform.
Last Update	The timestamp of the most recent integration of a given dataset in date-time format. For example, 2023-07-24T11:12:35.461612.
Last Run Status	The last job run of your dataset.

Column	Description
Last Integration Status	The status of the last attempt to send metadata to Collibra Platform.
Integration Status	The current status of the integration of your dataset.
Created By	The username of the user who integrated a dataset.
Active	Shows whether a data-set can actively integ-rate into Collibra Platform.

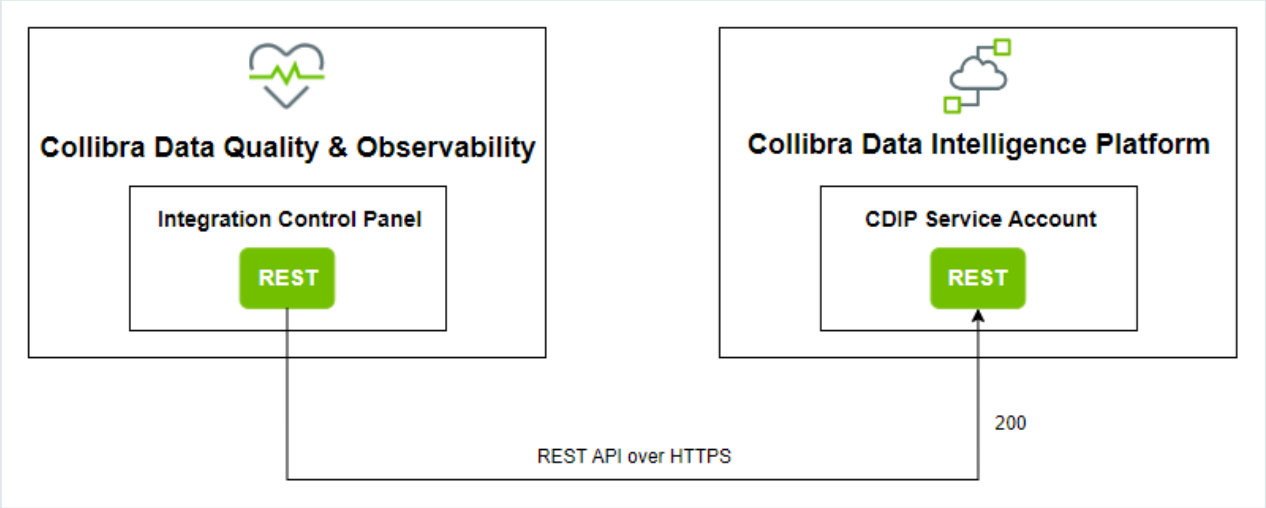
Known limitations

- Because data quality for Table assets in Collibra Platform is calculated based on the Data Quality Job asset, the data quality dimension cards are not visible on the Quality tab because the relation to DQ dimensions does not exist on DQ Job assets.
- Rows Passed/Rows Failed do not populate on the integration.
- Rules nested under the Data Quality Job are not visible on the Table asset.

Integration architecture

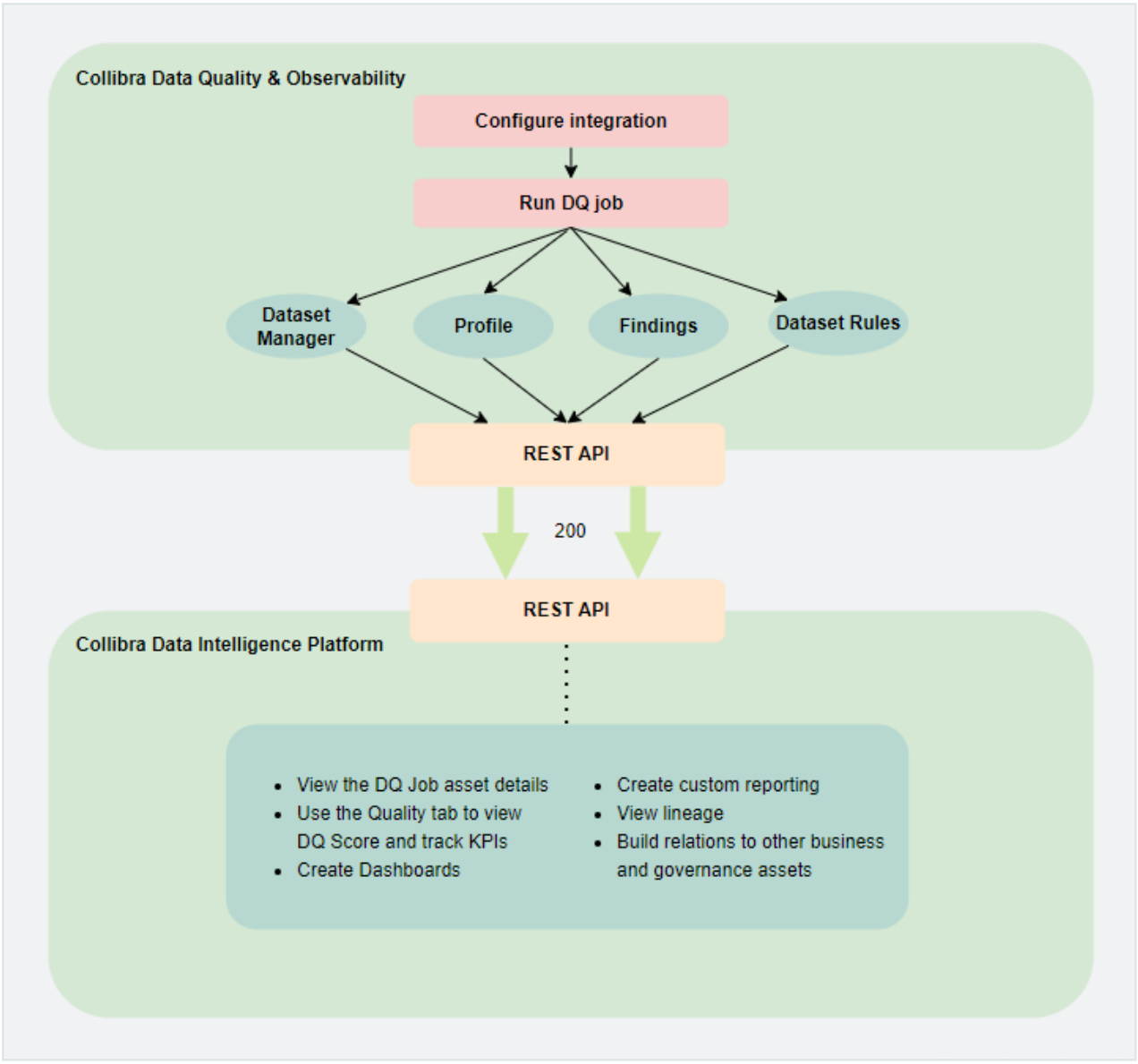
This section shows the architecture of an integration of Data Quality & Observability Classic with Collibra Platform.

Integration Control Panel

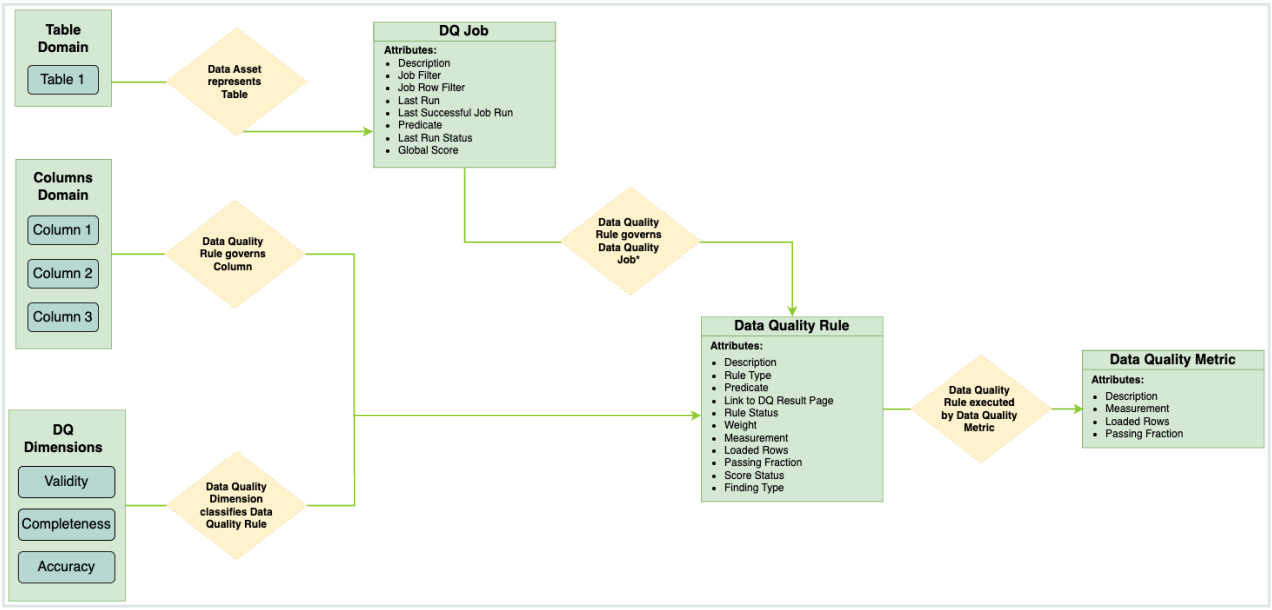


This diagram shows the flow of events upon the initial configuration of a new integration between Data Quality & Observability Classic and Collibra Platform. This uses the Integration API to communicate between Collibra DQ and Collibra Platform.

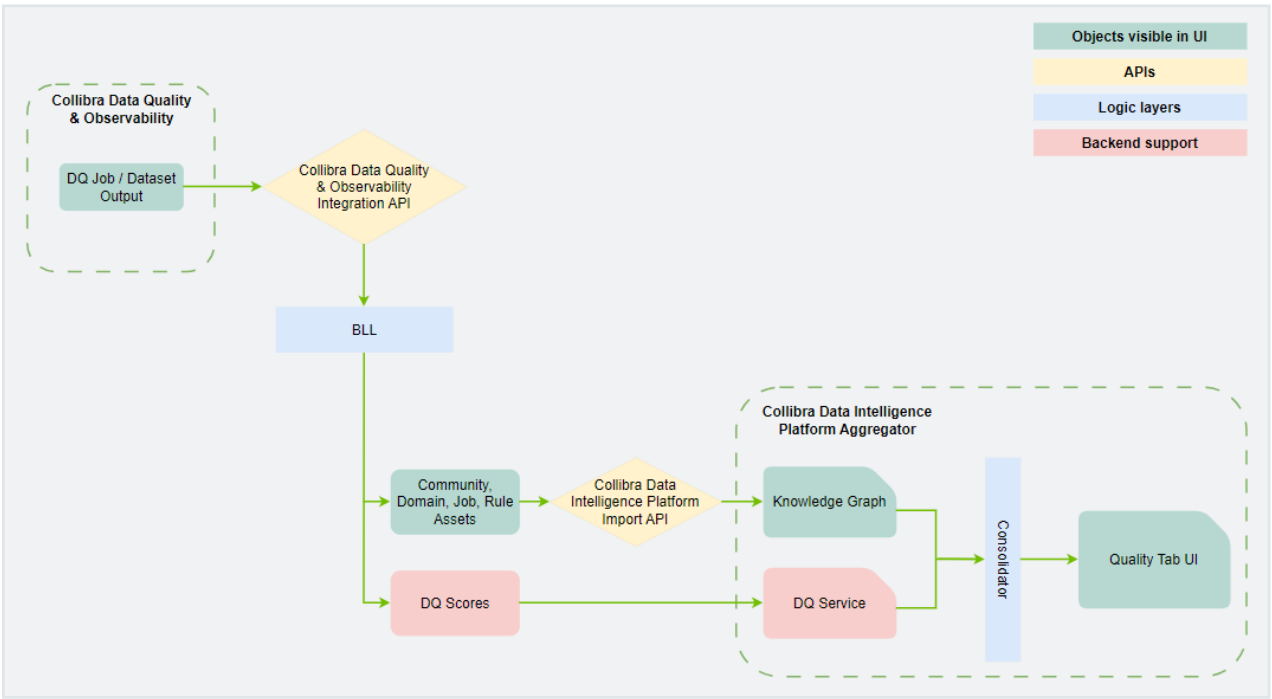
Integrating a dataset with Collibra Platform



Operating model



DQ service architecture



Configuring a new integration

This section shows you how to use the Integration configuration wizard to set up a new integration of Data Quality & Observability Classic metadata from profile findings, rules, and alerts into Collibra Platform.

> **Important** If you have multiple integrations set up, only one can be active at a given time.

> **Note** This is a one-time configuration per environment.

Prerequisites

For the Integration API to run successfully, ensure that you review and meet the following criteria.

- Your Data Quality & Observability Classic and Collibra Platform versions are 2025.02 or newer.
- You have an Admin role in Data Quality & Observability Classic to use the configuration wizard.
- You have an Admin or Dataset Manager role in Data Quality & Observability Classic to enable or disable individual datasets for integration.
- Your Collibra Platform profile must have Sysadmin and DataSteward global roles to manage all assets and objects of the integration.
- You have followed the steps on how to add Data Quality Rules to Assets in Collibra Platform.
- You have an aggregation path set for your Data Quality Rule in Collibra Platform ▸ Settings ▸ Operating model ▸ Data quality rules.

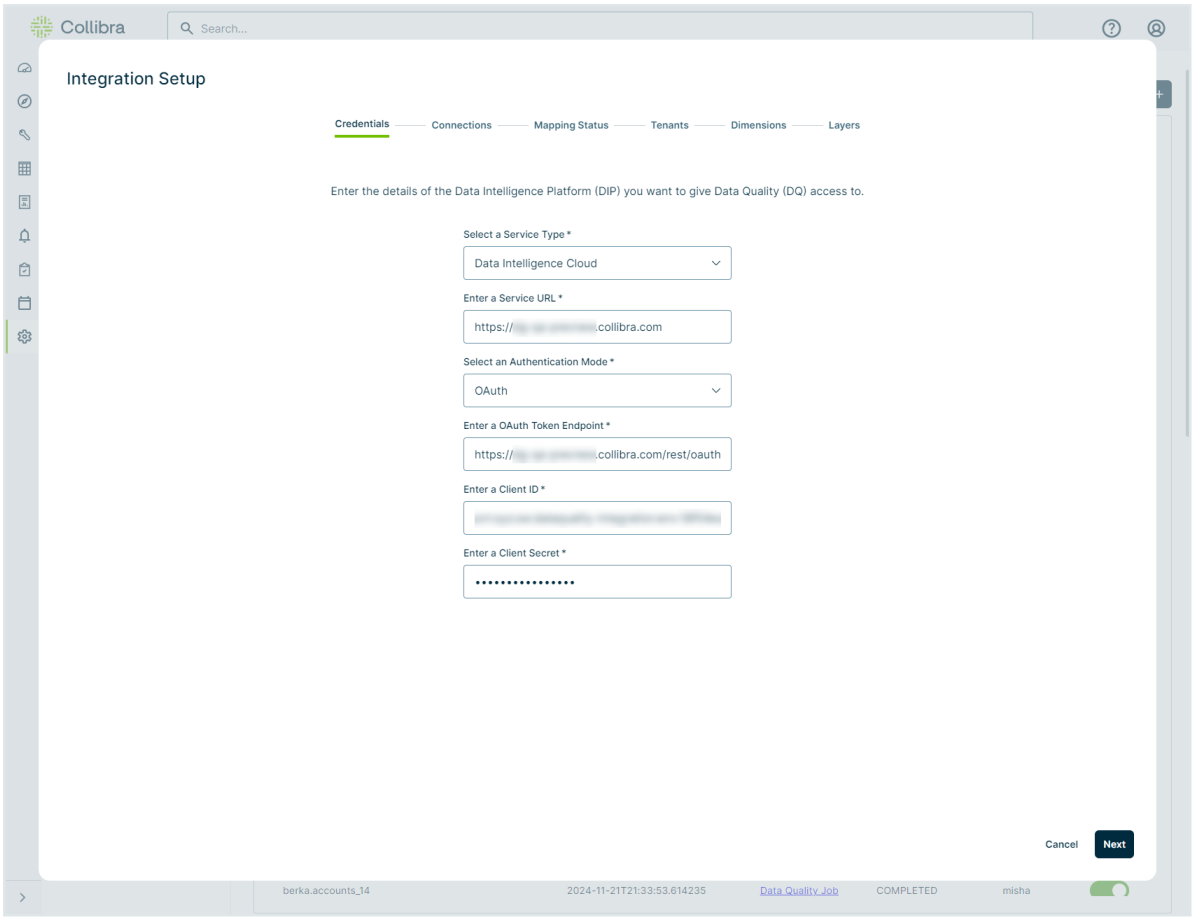
> **Important** Do not modify the cardinality of out-of-the-box operating model relations that impact the integration.

Steps

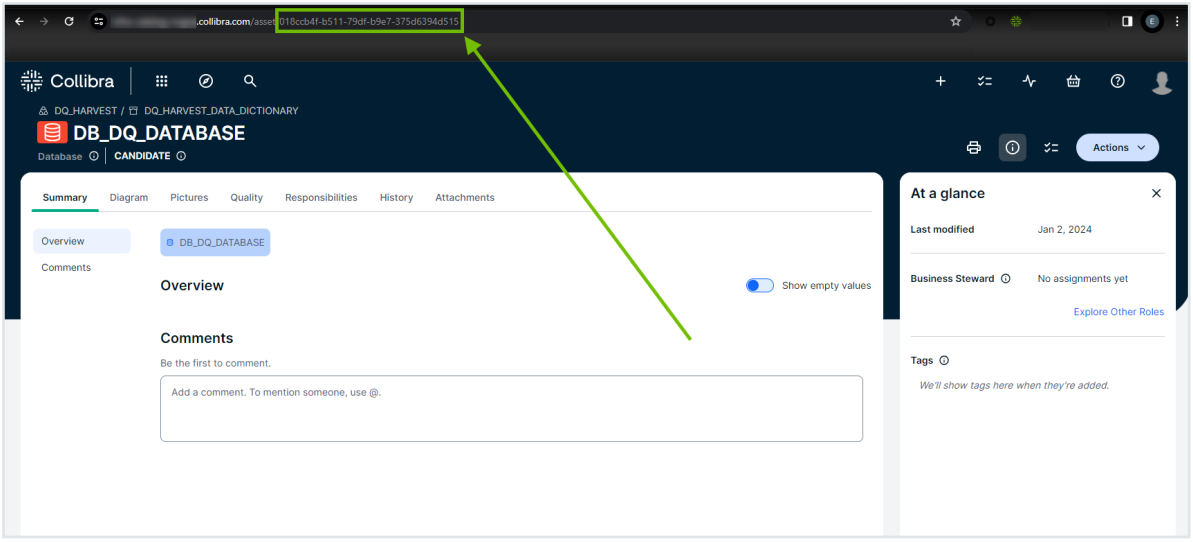
1. Hover your cursor over the ⚙ icon and click **Integrations**.
   » The Integrations page opens.
2. In the upper right corner, click **Add Integration**.
   » The Integration Setup wizard appears.
3. On the **Credentials** step, enter the required information.

Option	Description	Required
Service Type	The application with which to integrate Data Quality & Observability Classic metadata. This value is always **Data Intelligence Cloud**.	✓ Yes
Service URL	The URL of the Collibra Platform environment to send Collibra DQ metadata. For example, "https://dg-example.collibra.com".	✓ Yes
Authentication Mode	The method of authenticating the API call to Collibra Platform. You can select either Basic or OAuth.	✓ Yes
OAuth Token Endpoint	The Collibra Platform URL where Data Quality & Observability Classic sends requests to obtain the OAuth access token. For example, "https://dg-example.collibra.com/rest/oauth/v2/token"	✓ Yes **Tip** Required when using **OAuth** as the authentication mode.
Service Account	The username of your Collibra Platform environment. For example, "integration.user" or "Admin".	✓ Yes **Tip** Required when using **Basic** as the authentication mode.

Option	Description	Required
Password	The password of your Collibra Platform environment.	✓ Yes **Tip** Required when using **Basic** as the authentication mode.
Client ID	The public identifier provided by your OAuth service provider.	✓ Yes **Tip** Required when using **OAuth** as the authentication mode.
Client Secret	The credential string from your OAuth service provider used to verify the identity of Data Quality & Observability Classic during the token request process.	✓ Yes **Tip** Required when using **OAuth** as the authentication mode.



4. Click **Next**.
   » The Connections step opens.
5. On the **Connections** step, map your Collibra DQ connections to the appropriate database, schema, table, and column assets in Collibra Platform.
   a. Optionally, select the **Map to primary column only** switch to see only the column identified as the primary column related to the rule in Collibra Platform.
   b. From the **Select Connection** drop-down list, select a Collibra DQ database to integrate into Collibra Platform.
   c. In yourCollibra Platform environment, open the asset page of the database you intend to map, then copy the URL from the UUID of the database asset from the URL.



d. From the Connections tab in Collibra DQ, paste the copied UUID from Collibra Platform, or enter the UUID manually.
e. Click **Run** to create an automatic mapping between the schemas within your selected Collibra DQ database and the schemas within Collibra Platform.

> **Important**
> Only schemas ingested in Collibra Platform return when you click Run.

f. When the connection loads, click **Show Full Mapping** to expand the entire database mapping, including schemas, tables, and columns.

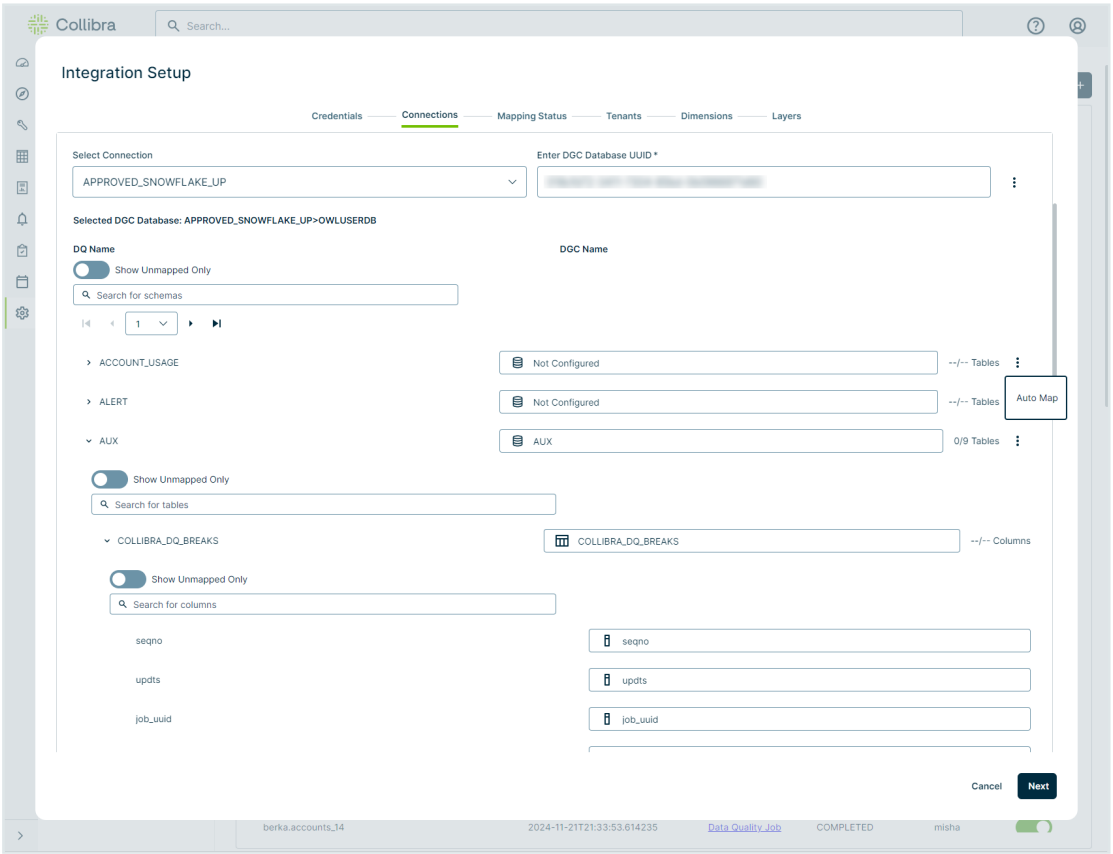
> **Important**
> It is critical that you expand all schemas, tables, and columns you intend to map. If you do not expand each item down to the column level, your table and column relations will not be present in your integration.

i. In the **DGC Name** column, optionally click the drop-down list and select a different schema ( ) to update the schema mapping.

> **Note** When a name match cannot be made, a "Not Configured" message displays.

> **Tip** Optionally click , then **Auto Map** to map tables in your DQ schema to those in DGC automatically. You can track the mapping status or cancel an in-progress mapping on the Mapping Status step.

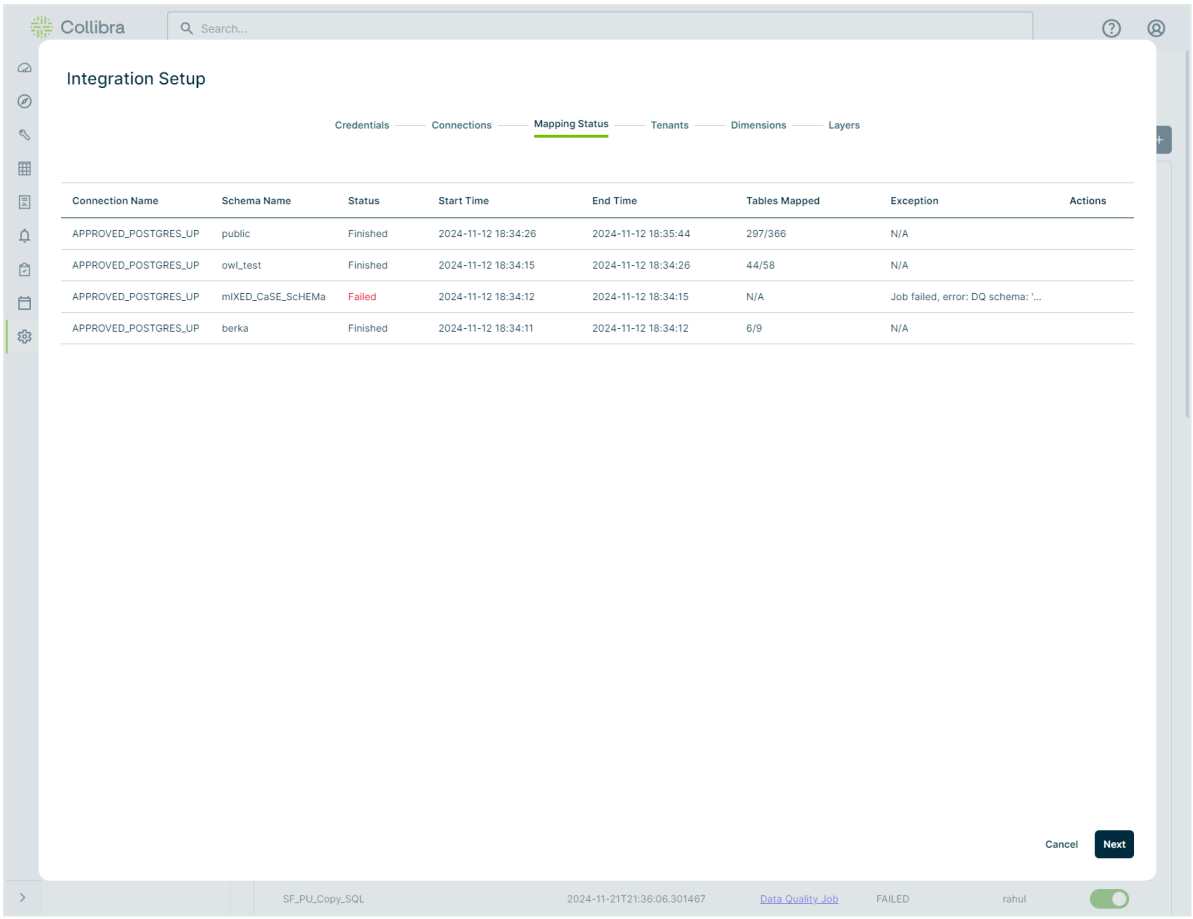
ii. In the **DQ Name** column, click the schema name to expand the available tables.
   » The selected schema expands to show the available tables, and their corresponding tables ( ) appear across from them in the DGC Name column.
iii. In the **DGC Name** column, click the drop-down list and select a different table to update the table mapping, if necessary.
iv. In the **DQ Name** column, click the table name to expand the available columns.
   » The selected schema expands to show the available columns, and their corresponding columns ( ) appear across from them in the DGC Name column.
v. In the **DGC Name** column, click the drop-down list and select a different column to update the column mapping, if necessary.



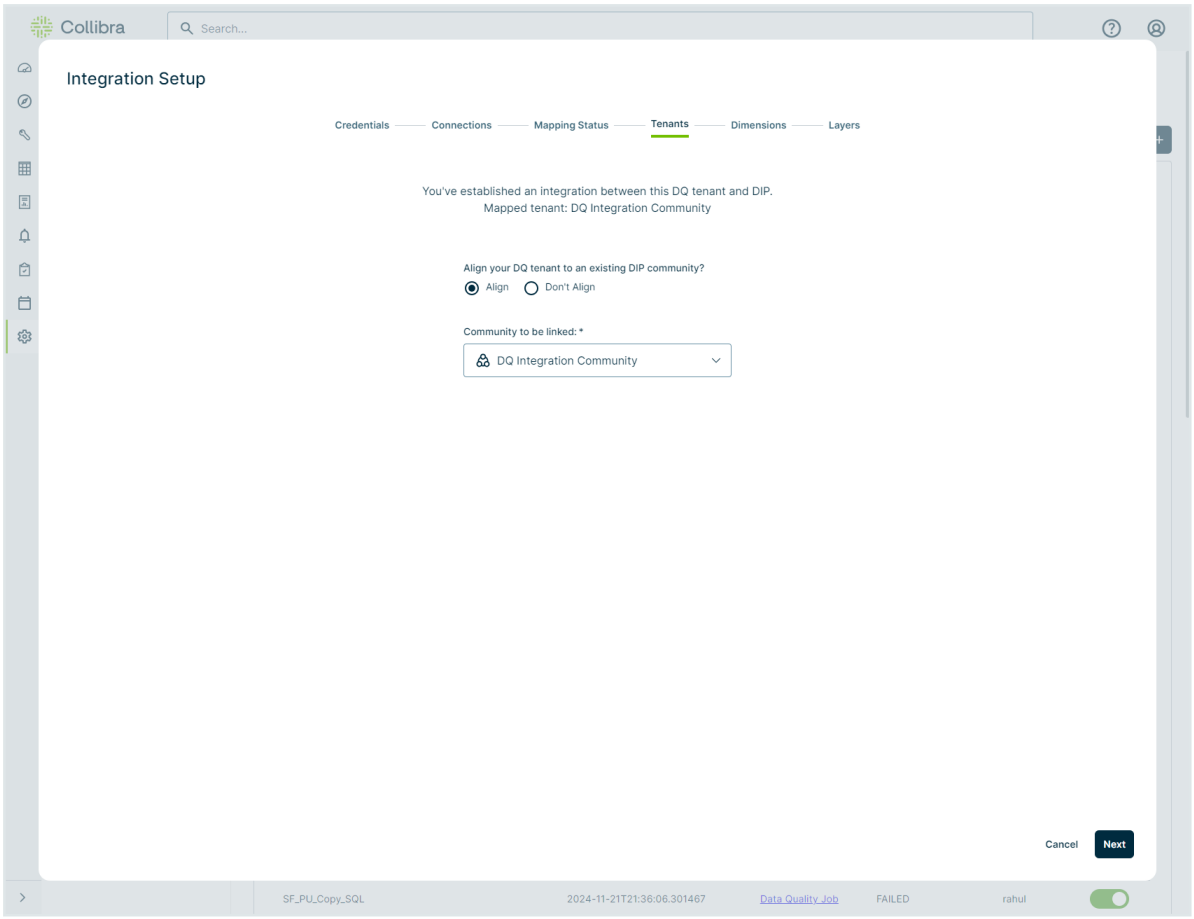
g. Optionally click **Add New** to map another connection and then repeat this process.

6. Click **Next**.
   » The Mapping Status step opens.
7. On the **Mapping Status** step, optionally review mapping details.
   Mapping Status allows you to view mapping details, such as the number of available tables in a schema are mapped and when a connection mapping fails.

Column	Description
Connection Name	The name of the data source included in the mapping.
Schema Name	The name of the schema included in the mapping.
Status	The state of completion of the mapping.
Start Time	The time stamp, in yyyy-MM-dd hh:mm:ss format, of when the mapping began.
End Time	The time stamp, in yyyy-MM-dd hh:mm:ss format, of when the mapping stopped.
Tables Mapped	The number of tables that mapped out of the total tables available in the schema.
Exception	When an error during the mapping occurs, a message describes the issue. If the error message is truncated, you can hover your cursor over it to view the full message in a tooltip. When an exception did not occur, the value in this column is N/A.
Actions	When a mapping job is in progress, a option appears in this column to allow you to cancel the job if necessary.



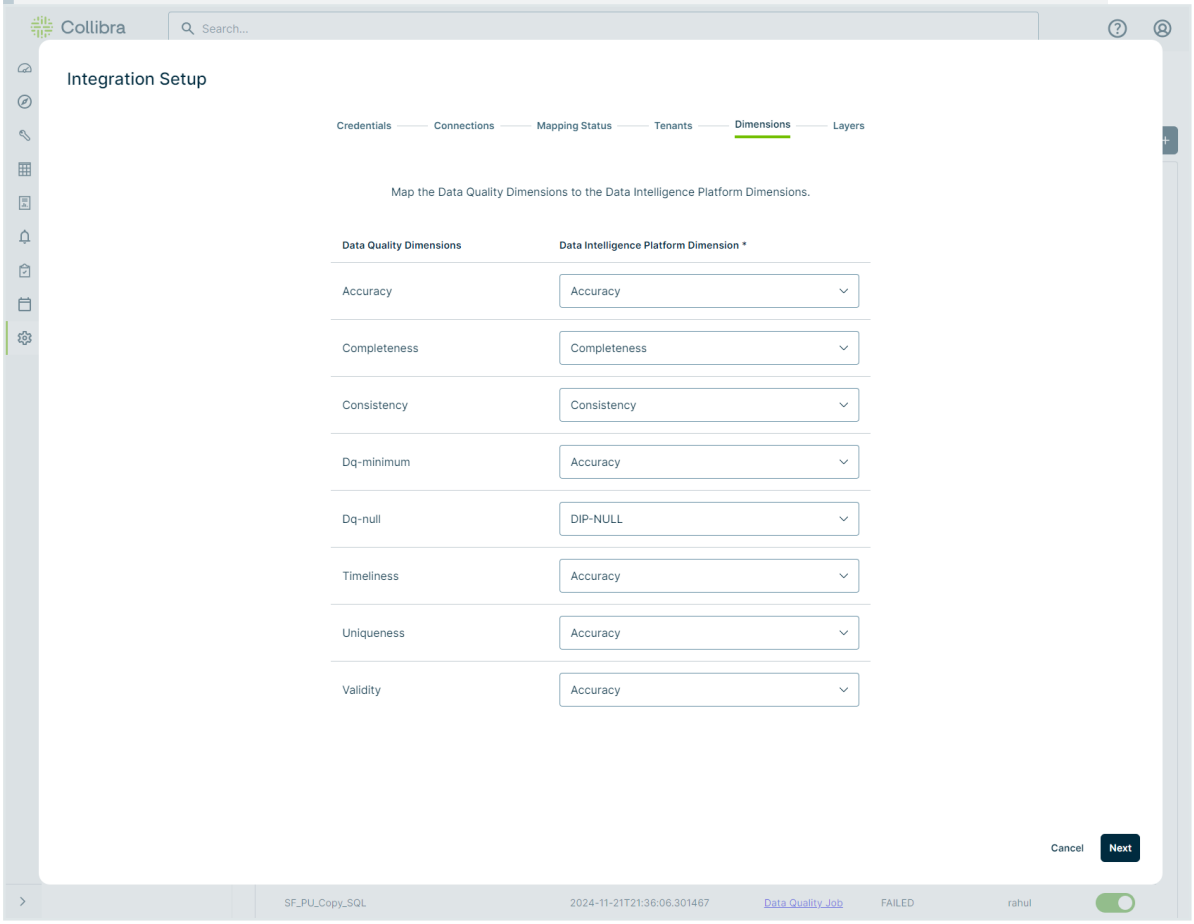
8. Click **Next**.
   » The Tenants step opens.
9. On the **Tenants** step, enter the required information.
   a. Optionally, choose to align your Collibra DQ tenant with an existing Collibra Platform community.
      i. Align
         - Select a Collibra Platform community from the drop-down list with which to align your tenant.
      ii. Don't Align,
         - Enter the name of a new community with which to align your tenant.



> **Note** Beneath each tenant-level community, a sub-community is created for each Business Unit that has datasets assigned to it in Collibra Platform. If a dataset is not assigned to a Business Unit, it is added to a community called *Unassigned DQ Job* by default.

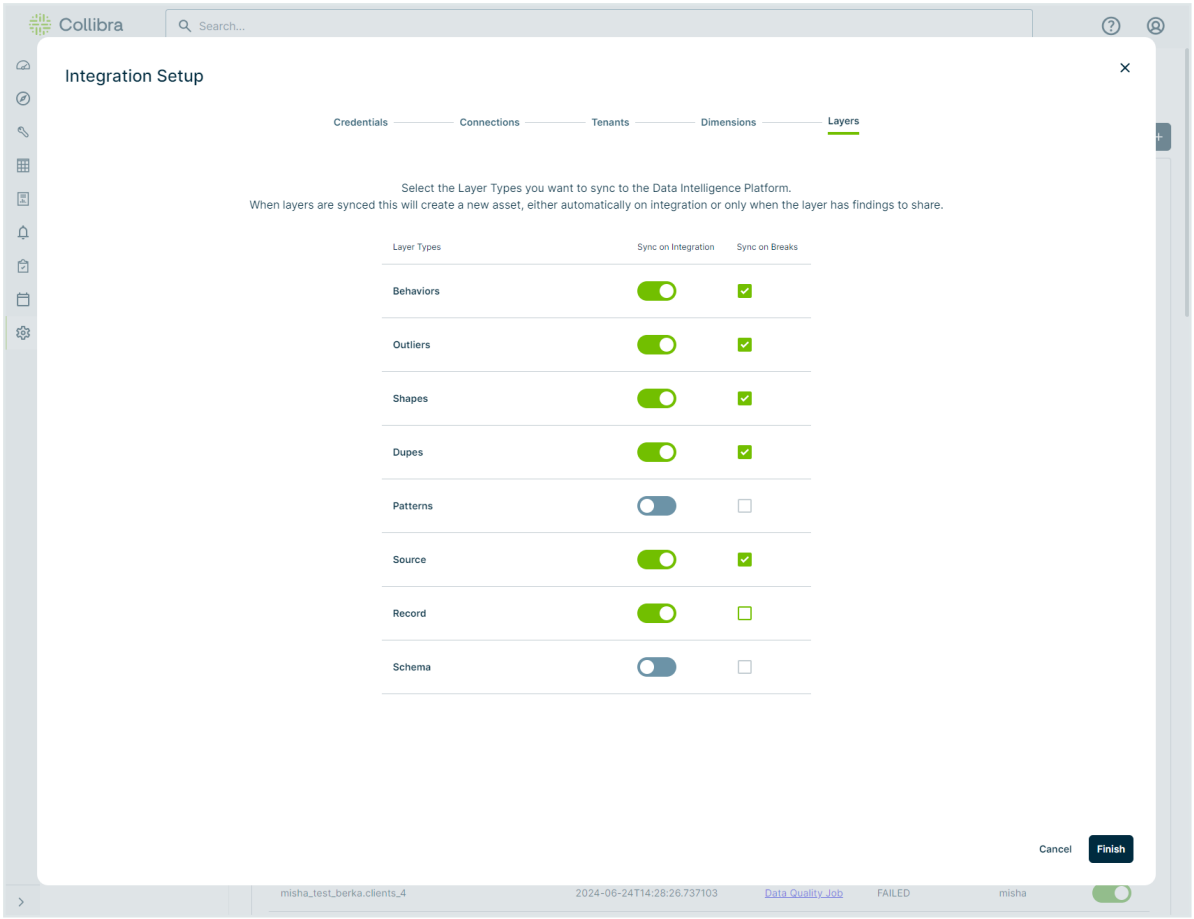
10. Click **Next**.
    » The Dimensions step opens.
11. On the **Dimensions** step, enter the required information.
    a. Select a dimension from the drop-down list of the Data Intelligence Platform Dimension column that corresponds with the Data Quality Dimension you want to identify as a match.

Note All Data Quality Dimensions from your Collibra Platform environment are available to map to Data Quality Dimensions in Collibra DQ, including custom dimensions. However, Data Quality Dimensions from Collibra DQ are not customizable at this time.



12. Click **Next**.
    » The Layers step opens.
13. On the **Layers** step, select which layers to synchronize to Collibra Platform.
    a. Select a layer in the **Sync on Integration** column to automatically create a new asset in Collibra Platform when the integration runs.
    b. Select the **Sync on Breaks** option to create a new asset in Collibra Platform only

when the layer has a finding.



14. Click **Finish**.

Known limitations

The integration only processes the primary schema or table and their associated columns. This is because the columns of the Collibra Platform table used by the integration map to the Data Quality & Observability Classic rules that contain them. A possible workaround to map columns from other tables referenced in the load query is to use a view to join multiple tables, then use that view as the Data Quality & Observability Classic data source.

What's next?

Enable a dataset for integration and start ingesting metadata from Collibra DQ into Collibra Platform.

Enabling dataset integrations

This section shows you how to integrate Data Quality & Observability Classic metadata from datasets into Collibra Platform.

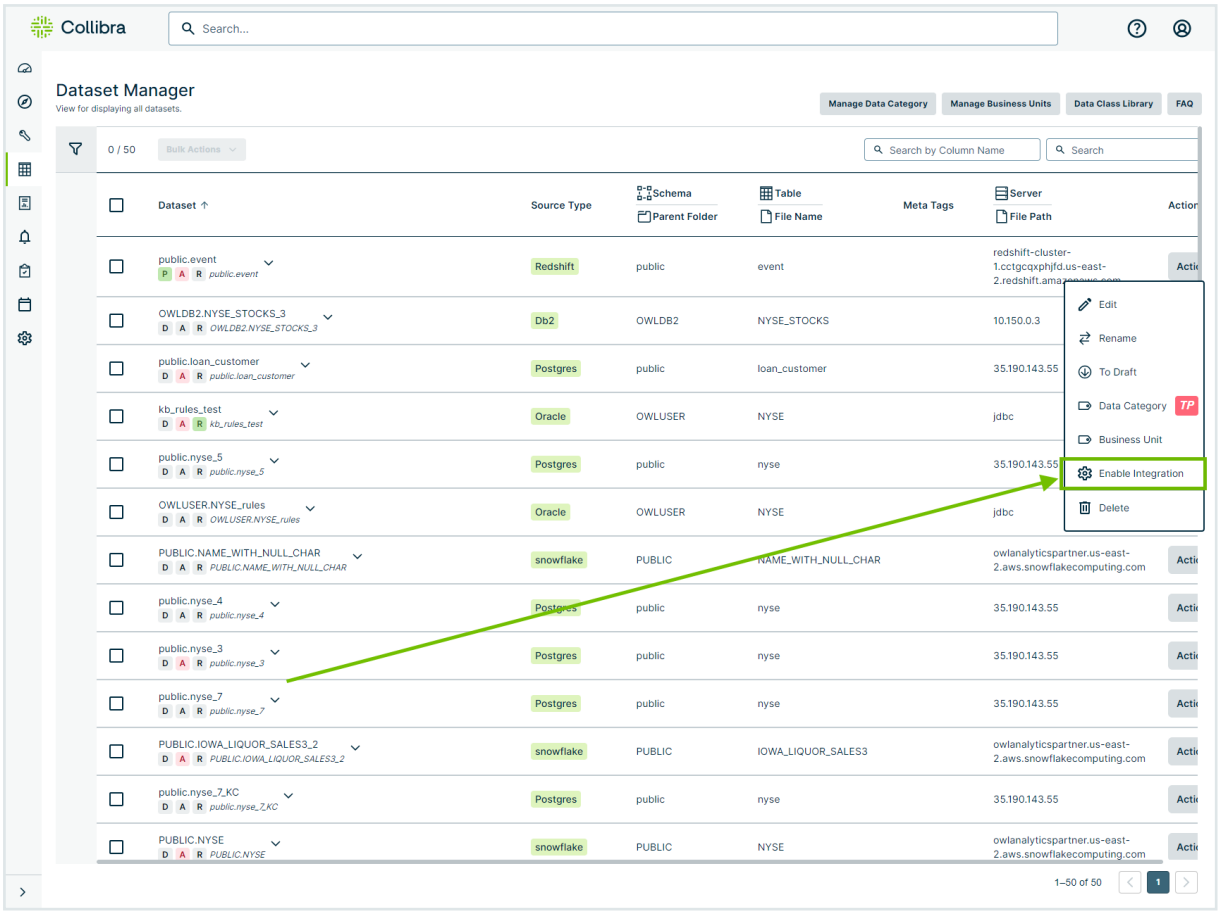
Prerequisites

- You have set up an integration between Collibra DQ and Collibra Platform.
- You have Admin privileges.

Steps

You can enable the integration of a dataset from the following pages.

1. From the sidebar menu, click to open Dataset Manager.
   » The Dataset Manager page opens.
2. In the Actions column, click Action, then click **Enable Integration**.
   » Collibra Platform ingests the assets that pertain to your dataset in Collibra DQ.

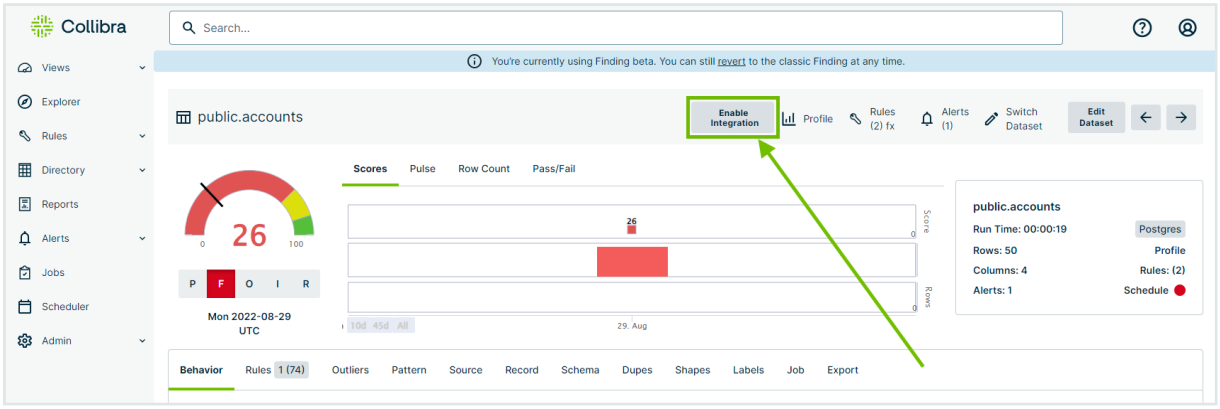


Important To enable an integration from the Findings page, React must be on.

1. There are three ways you can access the Findings page.
    a. From the Profile page:
        i. In the metadata ribbon below the score trend graph, click ⋮ on the right, and then click **View Findings**.
            » The Findings page opens.
    b. From the Jobs page:
        i. Click the name of your dataset in the Dataset column.
            » The Findings page opens.
    c. From the Dataset Manager page:
        i. Click the name of your dataset in the Dataset column.
            » The Profile page opens.

ii. In the metadata ribbon below the score trend graph, click ⋮ on the right, and then click **View Findings**.
   » The Findings page opens.

2. In the metadata ribbon at the top of the page, click **Enable Integration**.
   » Collibra Platform ingests the assets that pertain to your dataset in Collibra DQ.



What's next?

You may find that you do not want Collibra Platform to ingest a particular dataset, or no longer want an old dataset to continue to integrate when it runs in Collibra DQ. If this is the case, see how you can optionally disable dataset integrations.

Disabling dataset integrations

This section shows you how to disable automatic integrations of metadata from datasets in Collibra DQ into Collibra Platform.

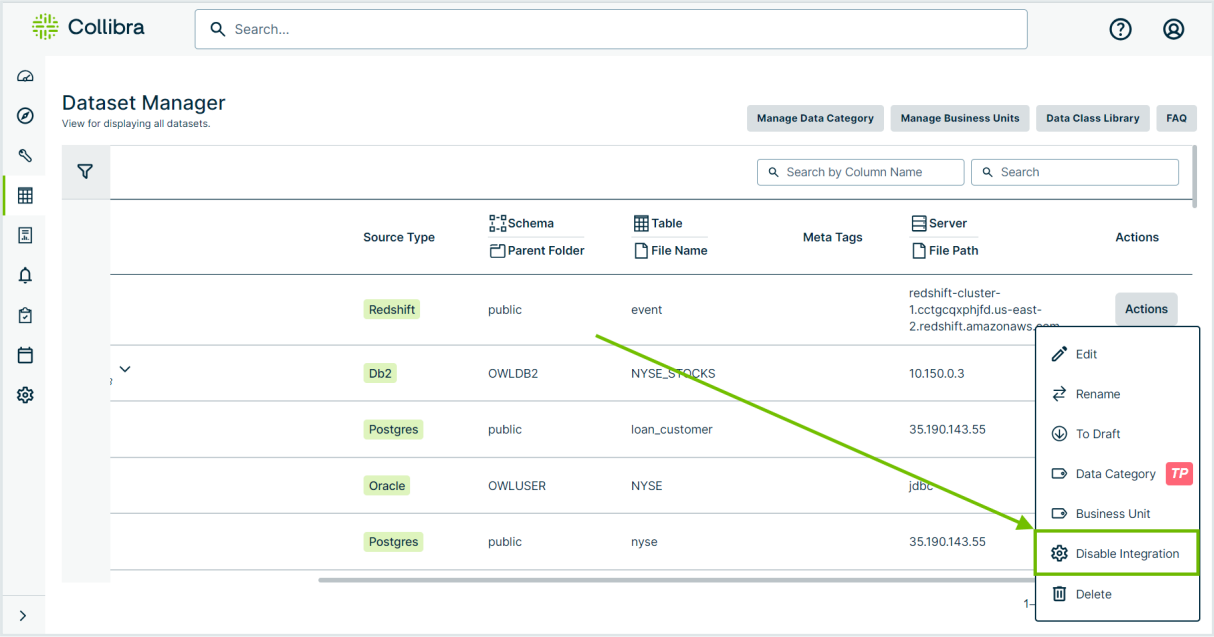
Steps

You can optionally disable integrations of a dataset from the following pages.

1. From the sidebar menu, click to open Dataset Manager.
   » The Dataset Manager page opens.

2. In the Actions column, click Action, then click **Enable Integration**.
   » Collibra Platform no longer automatically ingests the metadata from your dataset when it runs in Collibra DQ.

> **Note** Disabling an integration does not delete Data Quality Job assets in Collibra Platform.



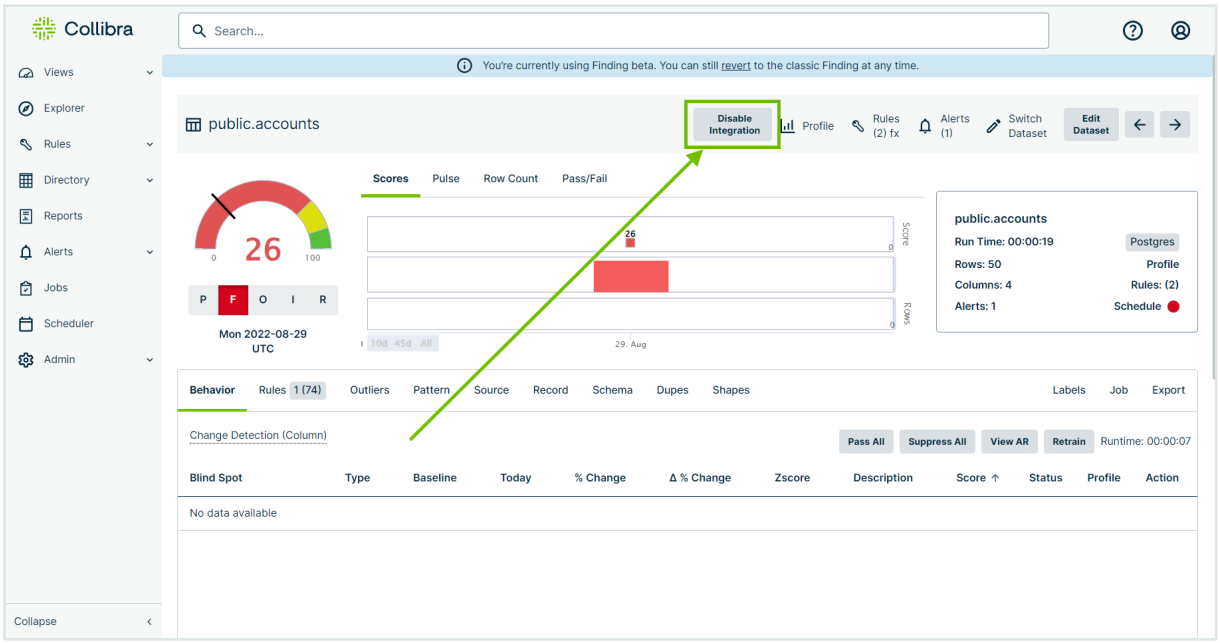
> **Important** To enable an integration from the Findings page, React must be on.

1. There are three ways you can access the Findings page.
   a. From the Profile page:
      i. In the metadata ribbon below the score trend graph, click ⋮ on the right, and then click **View Findings**.
         » The Findings page opens.
   b. From the Jobs page:
      i. Click the name of your dataset in the Dataset column.
         » The Findings page opens.

c. From the Dataset Manager page:
   i. Click the name of your dataset in the Dataset column.
      » The Profile page opens.
   ii. In the metadata ribbon below the score trend graph, click ⋮ on the right, and then click **View Findings**.
      » The Findings page opens.

2. In the metadata ribbon at the top of the page, click **Disable Integration**.
   » Collibra Platform no longer automatically ingests the metadata from your dataset when it runs in Collibra DQ.

> **Note** Disabling an integration does not delete Data Quality Job assets in Collibra Platform.



Modifying an existing integration

This section shows you how to use the Integration Configuration Wizard to modify an existing integration of Data Quality & Observability Classic with Collibra Platform.

Steps

1. Hover your cursor over the ⚙ icon and click **Integrations**.
   » The Integrations page opens.
2. On the right side of the status table, click the ✎ icon.
   » The Update Integration Credentials wizard appears.
3. Update the information on as many tabs as necessary.
4. Click **Save**.

Integration FAQ

The following table contains the most frequently asked questions about the Data Quality & Observability Classic Integration.

Question	Answer
What roles are required in Collibra DQ and Collibra Platform?	In **Collibra DQ**: • `ROLE_ADMIN` is required to complete the configuration wizard steps, which includes adding Collibra Platform credentials and the mappings of connections, tenants, and dimensions. • `ROLE_ADMIN` or `ROLE_DATASET_MANAGER` is required to enable or disable individual datasets for integration. In **Collibra Platform**: • The credentials shared in the configuration wizard in Collibra DQ must have `Sysadmin` and `DataSteward` permissions in Collibra Platform. • Any user can view DQ assets in Collibra Platform.

Question	Answer

What assets are created in Collibra Platform based on Collibra DQ metadata?	• **Data Quality Job Assets**
	  • These are summaries of the Collibra DQ dataset with systemic relations to the included tables.
	• **Data Quality Rule Assets**
	  • These are definitions of the data points of a Collibra DQ dataset that are being assessed for data quality and include systemic relations to the column (s) included in the rule.
	• **Data Quality Metric Assets**
	  • These are the quantitative outputs of a rule that include the numeric score used to calculate its data quality. Additionally, the following sub-asset types of data quality metrics are created based on the rule type:
	  • User Defined Rule Score
	  • Adaptive Rule Score
	  • Shapes Score
	  • Duplicates Score
	  • Outlier Score
	  • Record Score
	  • Schema Score
	  • Source Score
	  • Pattern Score

Will changes made to data quality assets in Collibra Platform be reflected in Collibra DQ?	No. In its current implementation, metadata only flows from Collibra DQ to Collibra Platform. Changes you make to DQ rules, jobs, or metric assets in Collibra Platform will not be reflected in Collibra DQ.
How are data quality scores calculated in Collibra Platform? Can I configure this?	We use the average of all rule (metric) level scores associated with a given asset to calculate data quality scores in Collibra Platform. While this is not currently configurable, if you have specific use case or methodology that you'd like to see implemented in the future, please submit a request via the Ideation platform for the Collibra DQ Product team to review.

Question	Answer
Which DQ dimensions does Collibra DQ share with Collibra Platform? Is this configurable?	Collibra DQ shares all DQ dimensions with Collibra Platform, which includes the following: • User Defined Rules • Adaptive Rules • Shapes • Source • Schema • Outliers • Duplicates • Records • Patterns This is not configurable.
I deleted a data quality rule in Collibra DQ, but I still see the DQ Rule asset in Collibra Platform. Does this affect the quality score of an asset?	No, it will not. When you or someone in your organization deletes a DQ rule from Collibra DQ, the DQ Rule asset in Collibra Platform does not automatically delete. However, DQ rules deleted in Collibra DQ update to "Suppressed" in Collibra Platform to automatically reflect a status change. A Suppressed DQ Rule does not impact the overall quality score of a given asset. To delete a DQ Rule asset in Collibra Platform, you need to perform a manual clean-up of the assets in Collibra Platform by following the steps outlined in the Delete an asset documentation.
I want to try the integration but I don't see the Integrations tab in the Admin Console in Collibra DQ. Do I need to upgrade my environment?	As long as your Collibra DQ environment is at least version 2023.07, you have access to the integration. Because the integration user interface components are only available when React is enabled, ensure your settings are properly configured to leverage the integration functionality.

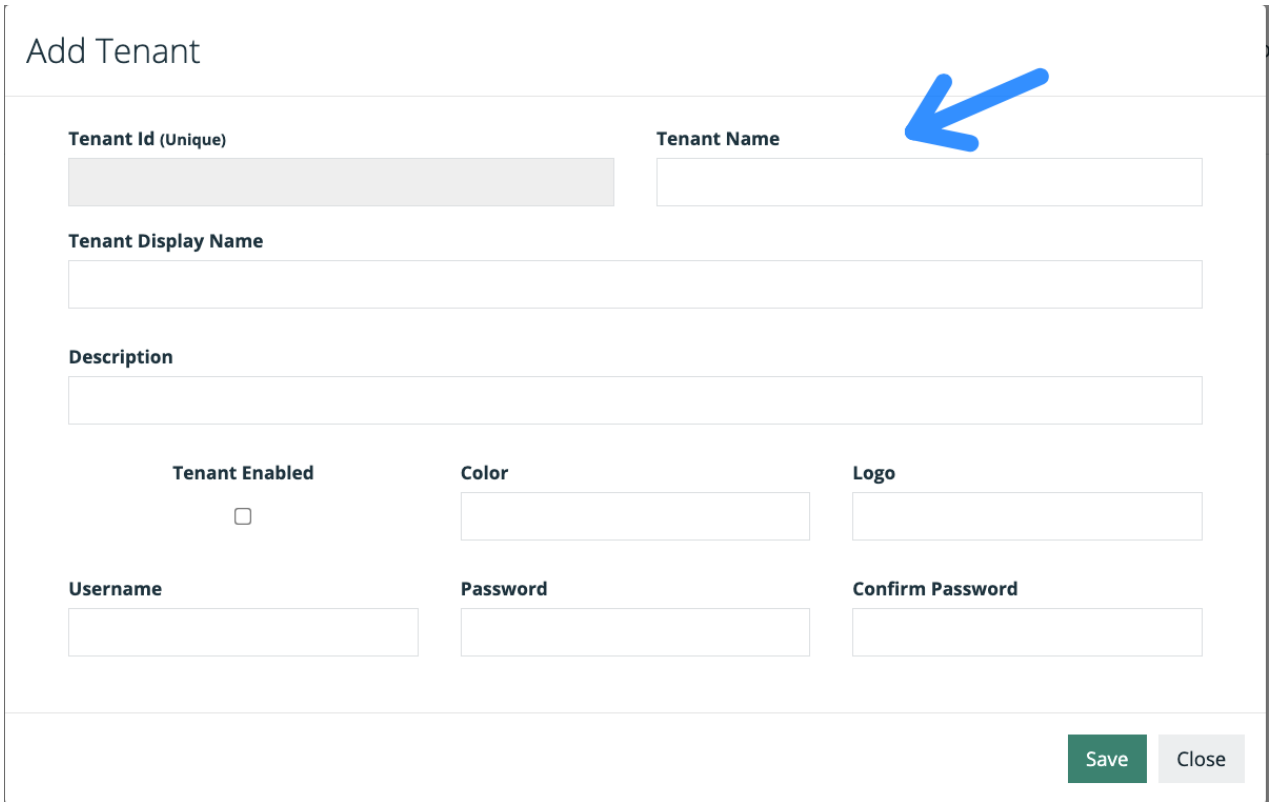
Question	Answer
I integrated my Collibra DQ dataset but the Quality tab/dashboard in Collibra Platform is missing. How do I access these views?	As with the legacy DQ Connector, to see the Quality tab/dashboard on asset pages in Collibra Platform, you need to configure and assign data quality rules from the Collibra settings page. For more information about how to create data quality rules, go to Data quality rules.

Solutions

Best Practices

Multi Tenant Names

> **Note** Tenant names should be lower case only.



Understanding Collibra DQ activities and what the key/date columns mean for each

- Starting with profile and expanding to rules and then other advanced capabilities.
- Training with DQ-team Zoom/Onsite support.
- Running with sample data.
- Introducing anomalies on sample data and running an owlcheck to see the anomalies.

Using the tool with practical scenarios

- Having Well Defined Use Cases
    - Determine a single table (dataset) that you would like to scan.
    - Have an expectation of what you would expect DQ to find in this dataset.
    - Understand which activities would capture the expected findings.
- Target internal datasets with known data issues.
- Historical Comparisons:
    - If pre-cleaned data is available with data findings that have been cleaned via legacy methods such as internal rules, run these datasets and compare the results from DQ to Internal findings.
- Work with data owners to understand findings or review expected findings.

Explorer

- The date selected with the calendar widget in the Scope (home) tab should align with the calendar widget assigned on the final (Save/Run) tab.
- If you elect to Unlock the cmd line and override the final parameters, do not re-lock or the changes will be overwritten. In general, only advanced users should override the guided settings.
- Pushdown and parallel JDBC cannot be used together. If you are using pushdown, do not select the parallel JDBC option.

Files

- File paths should not contain spaces or special characters.
- Backrun (replay) and advanced features are best suited for JDBC connections. Some features are unavailable if file and storage naming conventions do not consistently contain a date signature.

Connection Pool

If you see this message, update the agent configs in owl-env.sh or agent confg map for k8 deployments.

```
Failed to obtain JDBC Connection; nested exception is org.a-
pache.tomcat.jdbc.pool.PoolExhaustedException: [pool-29-thread-
2] Timeout: Pool empty. Unable to fetch a connection in 0
seconds, none available[size:2; busy:1; idle:0; lastwait:200].
```

Adjust these configs to modify the connection pool available.

```
export SPRING_DATASOURCE_POOL_MAX_WAIT=500
export SPRING_DATASOURCE_POOL_MAX_SIZE=10
export SPRING_DATASOURCE_POOL_INITIAL_SIZE=5
```

Freeform Agent Configs

When configuring the DQ Agent and using the Free Form Parameters at the bottom of the dialogue, you need to comma separate multiple -conf key/value pairs. Use this format: "-conf some.key=x, some.other.key=y".

K8 Secrets

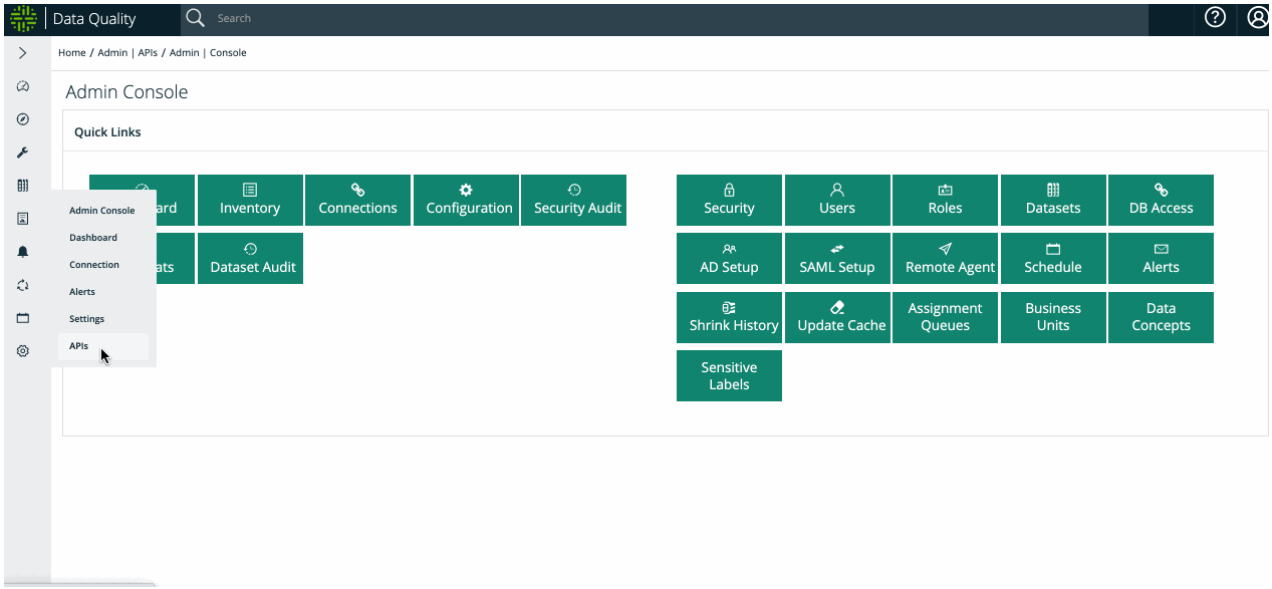
The following Env Vars are now managed as a Secret instead of as a Configmap:

LICENSE_KEY LIVY_SSL_KEY_PASS SERVER_SSL_KEY_PASS SPRING_AGENT_ DATASOURCE_PASSWORD SPRING_AGENT_DATASOURCE_USERNAME SPRING_ DATASOURCE_PASSWORD SPRING_DATASOURCE_USERNAME

Chapter 12

APIs

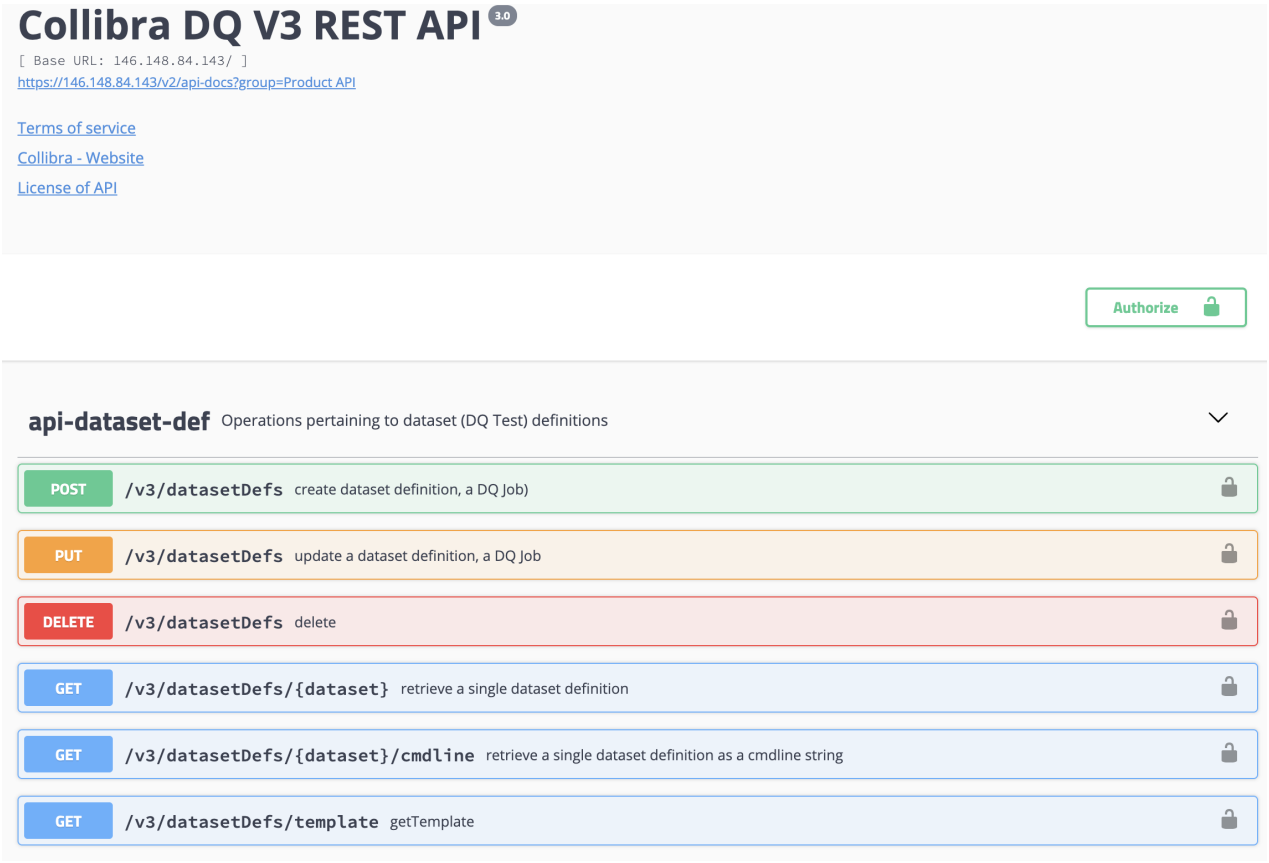
Rest



All REST APIs are available inside the application under admin section. The APIs can be used against the application in live working mode, which is preferred over documentation of APIs because it means the API works and was tested at compile time versus documentation time.

Product API

The product API is for end-users who want to interact with the official and supported API. You can also generate a client side SDK from the API with four steps below.




```
#psuedo code example REST API

dataset = 'public.nyse'
runId = '2021-03-05'

#SAVE datasetDef
dataset = POST /v3/datasetDefs/ {json_datasetDef}

#UPDATE datasetDef
dataset = PUT /v3/datasetDefs/ {json_datasetDef}

#RUN JOB
jobId = POST /v3/jobs/run/{dataset},{runDate}

#CHECK STATUS
status = /v3/jobs{jobId}/status

#GET DQ FINDINGS
findings = /v3/jobs/{jobId}/findings
```

JWT Token For Auth

```
import requests
import json
url = "http://localhost:9000/auth/signin"
payload = json.dumps({
    "username": "<user>",
    "password": "<pass>",
    "iss": "public"
})
headers = {
    'Content-Type': 'application/json'
}
response = requests.request("POST", url, headers=headers, data-
a=payload)
print(response.text)
```

```
curl --location --request POST 'http://-
localhost:9000/auth/signin' \
--header 'Content-Type: application/json' \
--data-raw '{
    "username": "<user>",
    "password": "<pass>",
    "iss": "public"
    }'
```

Python Example

Alternatively, you can use the rest endpoints directly. This example shows how it can be done with Python.

1. Create a dataset def:
    a. Using the UI (Explorer) or
    b. Using the dataset-def-api (https://<ip>/swagger-ui.html#/dataset-def-api)
2. Confirm your Python environment has the appropriate modules and imports.
3. Fill-in the variables and customize to your preference:
    a. url, user and pass
    b. dataset, runDate, and agentName

```
import requests
import json

# Authenticate
owl = "https://<url>"
url = "https://<url>/auth/signin"
payload = json.dumps({
  "username": "<user>", # Edit Here
  "password": "<pass>", # Edit Here
  "iss": "public"       # Edit Here
})
headers = {
  'Content-Type': 'application/json'
}
response = requests.request("POST", url, headers=headers, data-
a=payload, verify=False)
owl_header = {'Authorization': 'Bearer ' + response.json()
['token']}


# Run
dataset = '<your_dataset_name>'    # Edit Here
runDate = '2021-08-08'             # Edit Here
agentName = '<your_agent_name'     # Edit Here

response = requests.post(
    url = owl +
'/v3/jobs/run?agentName='+agentName+'&dataset='+dataset+'&run-
Date='+runDate,
    headers=owl_header,
    verify=False
)

jobId = str(response.json()['jobId'])


# Status
for stat in range(100):
    time.sleep(1)

    response = requests.get(
        url = owl + '/v3/jobs/'+jobId,
        headers=owl_header,
        verify=False
    )

    job = response.json()
```

```
    if job['status'] == 'FINISHED':
        break


# Results
response = requests.get(
    url = owl + '/v3/jobs/'+jobId+'/findings',
    headers=owl_header,
    verify=False
)

print(response.json())
```

This assumes you have created a dataset definition using the UI or from the template.

Command Line instead of JSON dataset def

You can run a similar job submission using the cmd line. Please note it is easiest to get the saved command line from the dataset-def-api **/v3/datasetDefs//cmdline** (with proper escaping) and passed to the **/v3/jobs/runCmdLine**.

Breaking Down The Sections

Submit the Job

Send in a data set name, date and agent to submit the job. This kicks off the engine to go do the work.

```
# Run
dataset = 'API_V3'
runDate = '2021-08-08'
agentName = 'owldq-owl-agent-owldq-dev-0'

response = requests.post(
    url = owl +
'/v3/jobs/run?agentName='+agentName+'&dataset='+dataset+'&run-
Date='+runDate,
    headers=owl_header
)

jobId = str(response.json()['jobId'])
```

Get the Status

Using the jobId returned from the job submission, you can check the status. In the example above, there is an interval to wait for the job to complete. You can create your own logic and orchestrate more precisely.

```
response = requests.get(
    url = owl + '/v3/jobs/'+jobId,
    headers=owl_header
)
```

Get the Results

Using the same jobId returned from the job submission, you can check the results. You will get a detailed json object with all the capabilities and detections in one payload. This is where you would decision, based on your organization and process.

```
response = requests.get(
    url = owl + '/v3/jobs/'+jobId,
    headers=owl_header
)
```

Python Example Raw

```
import requests
import json

# Variables
owl = 'https://<ip_address>'                #Edit
user = '<user>'                             #Edit
password = '<password>'                     #Edit
tenant = 'public'                           #Edit
dataset = '<your_dataset_name>'             #Edit
runDate = '2021-08-08'                      #Edit
agentName = 'your_agent_name'               #Edit

# Authenticate
url = owl+'/auth/signin'
payload = json.dumps({"username": user, "password": password,
"iss": tenant })
headers = {'Content-Type': 'application/json'}
response = requests.request("POST", url, headers=headers, data-
a=payload, verify=False)
owl_header = {'Authorization': 'Bearer ' + response.json()
['token']}

# Run
response = requests.post(url = owl +
'/v3/jobs/run?agentName='+agentName+'&dataset='+dataset+'&run-
Date='+runDate, headers=owl_header, verify=False)
jobId = str(response.json()['jobId'])

# Status
for stat in range(100):
    time.sleep(1)

    response = requests.get(url = owl + '/v3/jobs/'+jobId, head-
ers=owl_header, verify=False)
    status = response.json()['status']

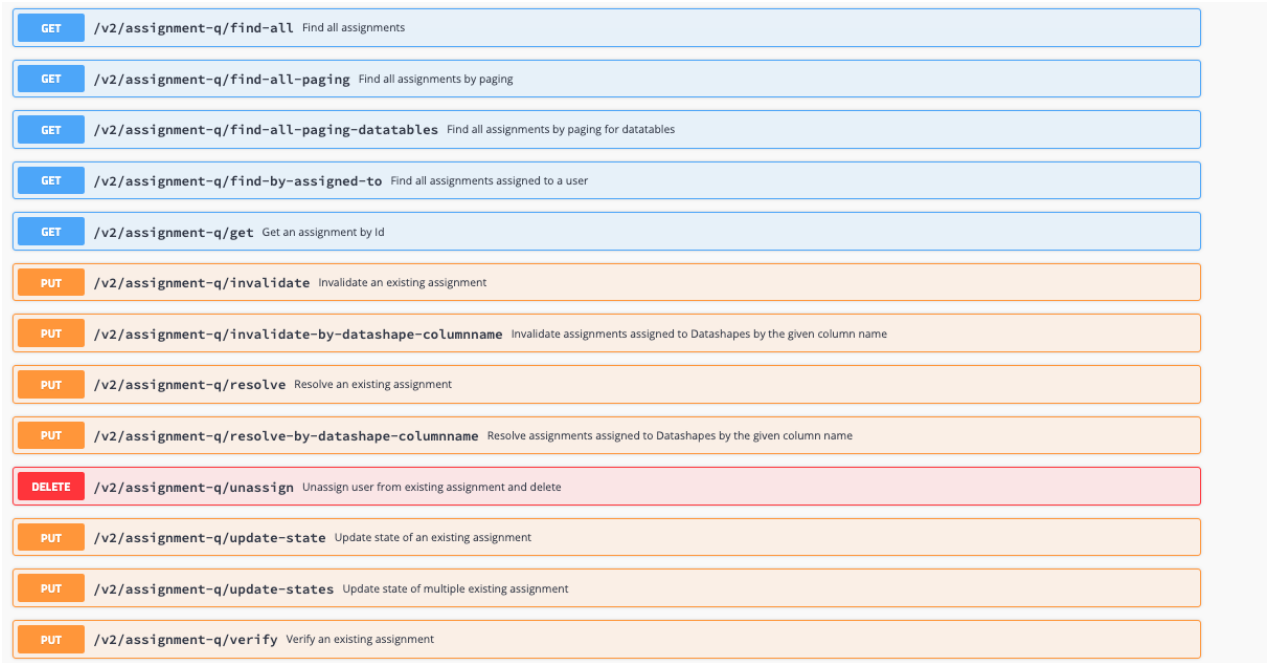
    if status == 'FINISHED':
        break

# Results
response = requests.get(url = owl + '/v3/jobs/'+jobId+'/find-
ings', headers=owl_header,  verify=False)
```

Internal API

Collibra DQ also exposes the internal API so that all potential operations are available. The caveat is that these calls may change over time or expose underlying functionality.





Dataset Definition

The JSON for the full dataset definition. It can be more terse to send in the cmdline string of just the variables you use for your DQ Job.

```
-df "yyyy/MM/dd" -owluser <user> -numexecutors 1 -executormemory
1g \
-f s3a://s3-datasets/dataset.csv -h <host>:5432/dev?-
currentSchema=public \
-fq "select * from dataset" -drivermemory 1g -master k8s:// -ds
dataset_csv_1 \
-deploymode cluster -bhlb 10 -rd "2021-04-01" -fullfile -log-
level INFO -cxn s3test5 \
-sparkprinc user2@CW.COM -sparkkeytab /tmp/user2.keytab
```

```
{
     "dataset": "",
     "runId": "",
     "runIdEnd": "",
     "runState": "DRAFT",
     "passFail": 1,
     "passFailLimit": 75,
     "jobId": 0,
     "coreMaxActiveConnections": null,
     "linkId": null,
     "licenseKey": "",
     "logFile": "",
     "logLevel": "",
     "hootOnly": false,
     "prettyPrint": true,
     "useTemplate": false,
     "parallel": false,
     "plan": false,
     "dataPreviewOff": false,
     "datasetSafeOff": false,
     "obslimit": 300,
     "pgUser": "",
     "pgPassword": "",
     "host": null,
     "port": null,
     "user": "anonymous : use -owluser",
     "alertEmail": null,
     "scheduleTime": null,
     "schemaScore": 1,
     "optionAppend": "",
     "keyDelimiter": "~~",
     "agentId": null,
     "load": {
       "readonly": false,
       "passwordManager": null,
       "alias": "",
       "query": "",
       "key": "",
       "expression": "",
       "addDateColumn": false,
       "zeroFillNull": false,
       "replaceNulls": "",
       "stringMode": false,
       "operator": null,
       "dateColumn": null,
       "transform": null,
       "filter": "",
       "filterNot": "",
```

```
      "sample": 1,
      "backRun": 0,
      "backRunBin": "DAY",
      "unionLookBack": false,
      "cache": true,
      "dateFormat": "yyyy-MM-dd",
      "timeFormat": "HH:mm:ss.SSS",
      "timestamp": false,
      "filePath": "",
      "fileQuery": "",
      "fullFile": false,
      "fileHeader": null,
      "inferSchema": true,
      "fileType": null,
      "delimiter": ",",
      "fileCharSet": "UTF-8",
      "skipLines": 0,
      "avroSchema": "",
      "xmlRowTag": "",
      "flatten": false,
      "handleMaps": false,
      "handleMixedJson": false,
      "multiLine": false,
      "lib": "",
      "driverName": null,
      "connectionName": "",
      "connectionUrl": "",
      "userName": "",
      "password": "",
      "connectionProperties": {},
      "hiveNative": null,
      "hiveNativeHWC": false,
      "useSql": true,
      "columnName": null,
      "lowerBound": null,
      "upperBound": null,
      "numPartitions": 0,
      "escapeWithBackTick": false,
      "escapeWithSingleQuote": false,
      "escapeWithDoubleQuote": false,
      "escapeCharacter": "",
      "hasHeader": true
    },
    "outliers": [
      {
        "id": null,
        "on": false,
        "only": false,
```

```
        "lookback": 5,
        "key": null,
        "include": null,
        "exclude": null,
        "dateColumn": null,
        "timeColumn": null,
        "timeBin": "DAY",
        "timeBinQuery": "",
        "categorical": true,
        "by": null,
        "limit": 300,
        "minHistory": 3,
        "historyLimit": 5,
        "score": 1,
        "aggFunc": "",
        "aggQuery": "",
        "query": "",
        "q1": 0.15,
        "q3": 0.85,
        "categoricalColumnConcatenation": false,
        "limitCategorical": null,
        "measurementUnit": "",
        "multiplierUpper": 1.35,
        "multiplierLower": 1.35,
        "record": true,
        "filter": null,
        "combine": true,
        "categoricalConfidenceType": "",
        "categoricalTopN": 3,
        "categoricalBottomN": 2,
        "categoricalMaxConfidence": 0.02,
        "categoricalMaxFrequencyPercentile": 0.25,
        "categoricalMinFrequency": 1,
        "categoricalMinVariance": 0,
        "categoricalMaxCategoryN": 1,
        "categoricalParallel": true,
        "categoricalAlgorithm": "",
        "categoricalAlgorithmParameters": {}
      }
    ],
    "outlier": {
      "id": null,
      "on": false,
      "only": false,
      "lookback": 5,
      "key": null,
      "include": null,
      "exclude": null,
```

```
      "dateColumn": null,
      "timeColumn": null,
      "timeBin": "DAY",
      "timeBinQuery": "",
      "categorical": true,
      "by": null,
      "limit": 300,
      "minHistory": 3,
      "historyLimit": 5,
      "score": 1,
      "aggFunc": "",
      "aggQuery": "",
      "query": "",
      "q1": 0.15,
      "q3": 0.85,
      "categoricalColumnConcatenation": false,
      "limitCategorical": null,
      "measurementUnit": "",
      "multiplierUpper": 1.35,
      "multiplierLower": 1.35,
      "record": true,
      "filter": null,
      "combine": true,
      "categoricalConfidenceType": "",
      "categoricalTopN": 3,
      "categoricalBottomN": 2,
      "categoricalMaxConfidence": 0.02,
      "categoricalMaxFrequencyPercentile": 0.25,
      "categoricalMinFrequency": 1,
      "categoricalMinVariance": 0,
      "categoricalMaxCategoryN": 1,
      "categoricalParallel": true,
      "categoricalAlgorithm": "",
      "categoricalAlgorithmParameters": {}
    },
    "pattern": {
      "id": null,
      "only": false,
      "lookback": 5,
      "key": null,
      "dateColumn": null,
      "include": null,
      "exclude": null,
      "score": 1,
      "minSupport": 0.000033,
      "confidence": 0.6,
      "limit": 30,
      "query": "",
```

```
      "filter": null,
      "timeBin": "DAY",
      "on": false,
      "match": true,
      "lowFreq": false,
      "bucketLimit": 450000,
      "deDupe": true
    },
    "patterns": [
      {
        "id": null,
        "only": false,
        "lookback": 5,
        "key": null,
        "dateColumn": null,
        "include": null,
        "exclude": null,
        "score": 1,
        "minSupport": 0.000033,
        "confidence": 0.6,
        "limit": 30,
        "query": "",
        "filter": null,
        "timeBin": "DAY",
        "on": false,
        "match": true,
        "lowFreq": false,
        "bucketLimit": 450000,
        "deDupe": true
      }
    ],
    "dupe": {
      "on": false,
      "only": false,
      "include": null,
      "exclude": null,
      "depth": 0,
      "lowerBound": 99,
      "upperBound": 100,
      "approximate": 1,
      "limitPerDupe": 15,
      "checkHeader": true,
      "filter": null,
      "ignoreCase": true,
      "score": 1,
      "limit": 300
    },
    "profile": {
```

```
      "on": true,
      "only": false,
      "include": null,
      "exclude": null,
      "shape": true,
      "correlation": null,
      "histogram": null,
      "semantic": null,
      "limit": 300,
      "histogramLimit": 0,
      "score": 1,
      "shapeTotalScore": 0,
      "shapeSensitivity": 0,
      "shapeMaxPerCol": 0,
      "shapeMaxColSize": 0,
      "shapeGranular": null,
      "behavioralDimension": "",
      "behavioralDimensionGroup": "",
      "behavioralValueColumn": "",
      "behaviorScoreOff": false,
      "behaviorLookback": 10,
      "behaviorMinSupport": 4,
      "profilePushDown": null,
      "behaviorRowCheck": true,
      "behaviorTimeCheck": true,
      "behaviorMinValueCheck": true,
      "behaviorMaxValueCheck": true,
      "behaviorNullCheck": true,
      "behaviorEmptyCheck": true,
      "behaviorUniqueCheck": true,
      "adaptiveTier": null
    },
    "source": {
      "on": false,
      "only": false,
      "validateValues": false,
      "matches": false,
      "sourcePushDownCount": false,
      "include": null,
      "exclude": null,
      "includeSrc": null,
      "excludeSrc": null,
      "key": null,
      "map": null,
      "score": 1,
      "limit": 30,
      "dataset": "",
      "driverName": "",
```

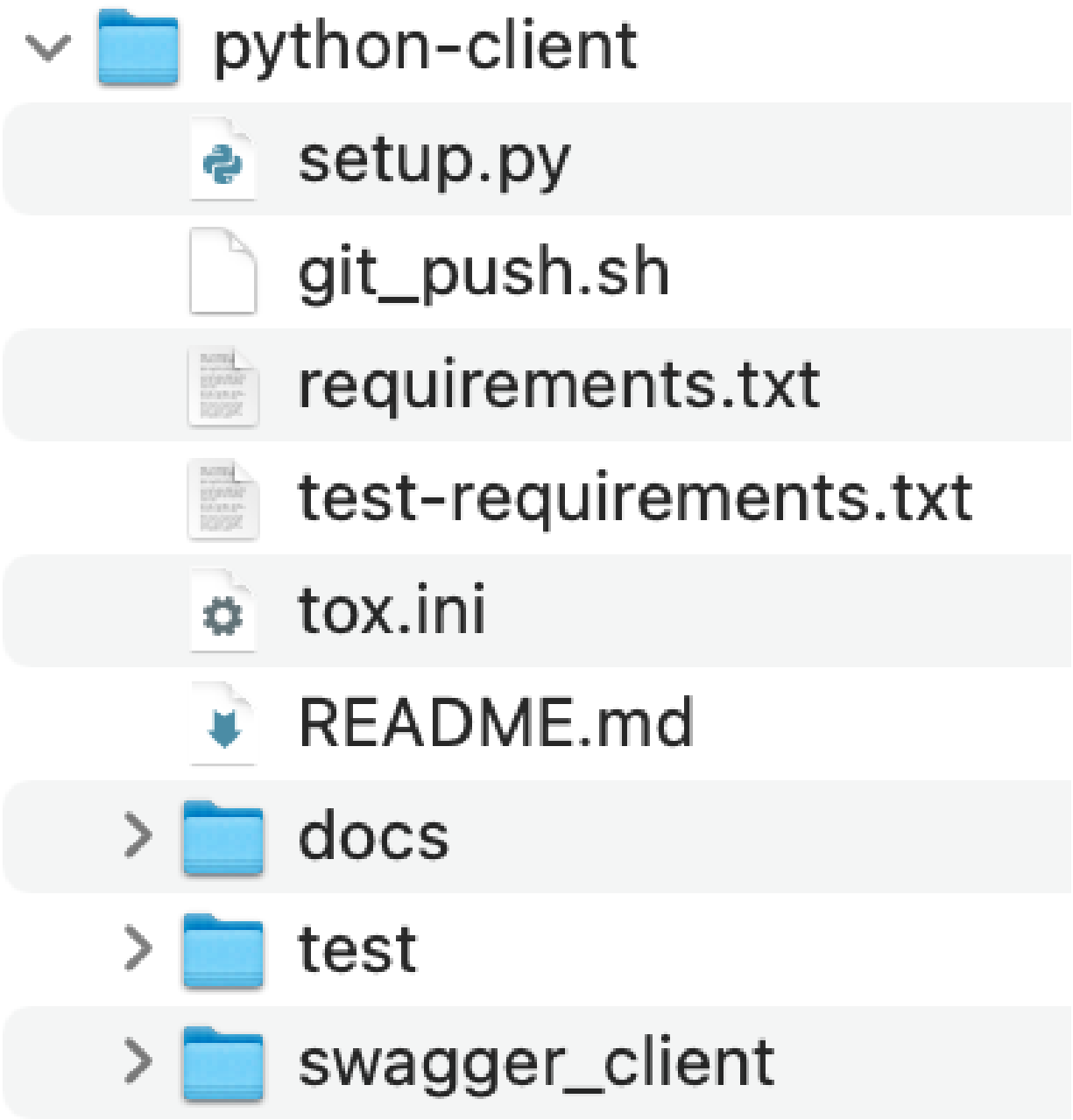
```
      "user": "",
      "password": "",
      "passwordManager": "",
      "connectionName": "",
      "connectionUrl": "",
      "query": "",
      "lib": "",
      "checkType": true,
      "checkCase": false,
      "validateValuesFilter": "",
      "validateSchemaOrder": false,
      "connectionProperties": {},
      "filePath": "",
      "fileQuery": "",
      "fullFile": false,
      "header": null,
      "skipLines": 0,
      "inferSchema": true,
      "fileType": null,
      "delimiter": ",",
      "fileCharSet": "UTF-8",
      "avroSchema": "",
      "xmlRowTag": "",
      "flatten": false,
      "handleMaps": false,
      "handleMixedJson": false,
      "multiLine": false,
      "hasHeader": true
    },
    "rule": {
      "on": true,
      "only": false,
      "lib": null,
      "name": "",
      "absoluteScoring": false,
      "ruleBreakPreviewLimit": 6
    },
    "colMatch": {
      "colMatchParallelProcesses": 3,
      "colMatchDurationMins": 20,
      "colMatchBatchSize": 2,
      "level": "exact",
      "fuzzyDistance": 1,
      "connectionList": []
    },
    "spark": {
      "numExecutors": 3,
      "driverMemory": "",
```

```
      "executorMemory": "",
      "executorCores": 1,
      "conf": "",
      "queue": "",
      "master": "local[*]",
      "principal": "",
      "keyTab": "",
      "deployMode": "",
      "jars": null,
      "packages": null,
      "files": null
    },
    "env": {
      "jdbcPrincipal": "",
      "jdbcKeyTab": ""
    },
    "record": {
      "on": false,
      "in": "",
      "notIn": "",
      "include": null,
      "percDeltaLimit": 0.1,
      "score": 1
    },
    "transforms": [],
    "pipeline": []
  }
```



Generate Client SDK

1. Go to https://editor.swagger.io/.
2. Click File Import URL.
3. Paste a URL that looks like this https://<host>/v2/api-docs?group=Product%20API.
4. Click generate client (python, java, scala, C#).



```
#Python SDK Example

#GET CMDLINE
cmdLine = get_job_cmdline(dataset)

#SUBMIT JOB
job_id = run(dataset, run_date)

#CHECK STATUS
status = get_job_status(job_id)

#GET DQ ISSUES
status = get_job_findings(dataset, run_date)
```

Code Migration using Product APIs

This document describes the process of migrating Collibra Data Quality code across environments using Product APIs. It supports both pushdown and pullup datasets.

> **Note** You must be logged in to Collibra Data Quality & Observability in order to access the Swagger API. Open Swagger from either the Admin Console or by clicking the ⓘ icon in the upper right corner of the page.

The migration process consists of the following steps:

1. Get definitions from the source environment
2. Update payloads for the target environment
3. Post definitions to target environment
4. Run job using the definition

The following sections describe these steps in more detail.

Get definitions from the source environment

Use the following endpoints to get the data that you want to migrate.

Get dataset definitions

`GET /v3/datasetDefs/{dataset}`

This endpoint returns the dataset definition for the specified dataset.

Get rules

`GET /v3/rules`

This endpoint returns definitions of all the custom rules associated with the dataset specified.

Update payloads for the target environment

After retrieving the dataset and rules definitions, copy them to a text editor. You can then prepare them for use as payloads in the subsequent POST endpoints.

> **Note** The following table provides the minimum number of properties to be updated. For a comprehensive list, create a sample dataset in the target environment using the UI and compare the JSON definitions.

Dataset payload

In the dataset payload, update the following fields for the target environment:

Field	Description
host	Change to the host name of the target environment. Use the endpoint `v2/getremotehost` if necessary.
user	The Collibra DQ user that will be used to run the newly migrated datasets.
agentId	Change the id and uuid values under agentId to match the target environment. Use the endpoint `v2/getagents` if necessary.
lib	Verify that the library specified matches the target environment.

Field	Description
master	Verify that the master URL (specified under `spark`) matches the target environment.

Rules payload

In the rules payload, verify that the userNM aligns with the target environment. Update the payload for each rule that you want to migrate.

Post definitions to target environment

Use the payloads with following endpoints to migrate code to the target environment.

Post dataset definitions

```
POST /v3/datasetDefs
```

Post rules

```
POST /v3/rules/{dataset}
```

This endpoint can accept multiple rule definitions (i.e., an array of rules). If the rule already exists, the endpoint will overwrite the rule with the new values.

Run job

After migrating the datasets and rules, use the following endpoint to run the job:

```
POST /v3/jobs/run
```

Provide the following parameters:

Parameter	Type	Description
dataset	string	Specify the name of the dataset that you created.

Parameter	Type	Description
runDate	string ($date-time)	Specify the run date.
runDateEnd	string ($date-time)	Required only for datasets that have a date range.
agentName	string	Specify the agentName. Use the endpoint `v2/getagents` if necessary to obtain the agentName. Required only for Pullup datasets.

You can check the status of the job from the Jobs page.

Using the Export and Import API

> **Important** The export/import process described in this document is intended for Pullup datasets only. If you are configuring a new migration process or migrating Pushdown datasets, use the migration process described in Migrating Data Across Environments.

Use the Export and Import API to promote and move datasets across environments. The export/import process supports the following tables:

- rule_repo
- owl_rule
- owl_check_repo
- job_schedule
- alert_cond
- alert_output
- dataset_scoring
- dataset_schema

The export/import process consists of the following steps:

- Add the required stored procedures (prerequisite)
- Export content from source
- Import content into target

The following sections describe these steps in more detail.

Add the required stored procedures (prerequisite)

Add the following stored procedure to the Collibra DQMetastore in the target and source environments. Replace `<schema>` with the tenant name. Note that this procedure is supported in CRDB version 23.2.16 and later.

```
CREATE OR REPLACE FUNCTION <schema>.dump(p_schema text, p_table
text, p_where text)
 RETURNS SETOF text
 LANGUAGE plpgsql
AS $function$
 DECLARE
     dumpquery_0 text;
     dumpquery_1 text;
     selquery text;
     selvalue text;
     valrec record;
     colrec record;
 BEGIN

     -- ------ --
     -- GLOBAL --
     --   build base INSERT
     --   build SELECT array[ ... ]
     dumpquery_0 := 'INSERT INTO ' ||  quote_ident(p_schema) ||
'.' || quote_ident(p_table) || '(';
     selquery    := 'SELECT array[';

     <<label0>>
     FOR colrec IN SELECT table_schema, table_name, column_
name, data_type
                   FROM information_schema.columns
                   WHERE table_name = p_table and table_schema
= p_schema
                   ORDER BY ordinal_position
     LOOP
         dumpquery_0 := dumpquery_0 || quote_ident(colrec.-
column_name) || ',';
         selquery    := selquery    || 'CAST(' || quote_ident
(colrec.column_name) || ' AS TEXT),';
```

```
     END LOOP label0;

     dumpquery_0 := substring(dumpquery_0 ,1,length(dumpquery_
0)-1) || ')';
     dumpquery_0 := dumpquery_0 || ' VALUES (';
     selquery    := substring(selquery    ,1,length(selquery)-
1)    || '] AS MYARRAY';
     selquery    := selquery    || ' FROM ' ||quote_ident(p_
schema)||'.'||quote_ident(p_table);
     selquery    := selquery    || ' WHERE '||p_where;
     -- GLOBAL --
     -- ------ --

     -- ----------- --
     -- SELECT LOOP --
     --   execute SELECT built and loop on each row
     <<label1>>
     FOR valrec IN  EXECUTE  selquery
     LOOP
         dumpquery_1 := '';
         IF not found THEN
             EXIT ;
         END IF;

         -- ----------- --
         -- LOOP ARRAY (EACH FIELDS) --
         <<label2>>
         FOREACH selvalue in ARRAY valrec.MYARRAY
         LOOP
             IF selvalue IS NULL
             THEN selvalue := 'NULL';
             ELSE selvalue := quote_literal(selvalue);
             END IF;
             dumpquery_1 := dumpquery_1 || selvalue || ',';
         END LOOP label2;
         dumpquery_1 := substring(dumpquery_1 ,1,length
(dumpquery_1)-1) || ');';
         -- LOOP ARRAY (EACH FIELD) --
         -- ----------- --

         -- debug: RETURN NEXT dumpquery_0 || dumpquery_1 || '
--' || selquery;
         -- debug: RETURN NEXT selquery;
         RETURN NEXT dumpquery_0 || dumpquery_1;

     END LOOP label1 ;
     -- SELECT LOOP --
     -- ----------- --

 RETURN ;
```

```
  END
  $function$
;
```

Add the following assignment:

```
alter function dump(text, text, text) owner to <ownername>;
```

Export content from source

Export content from the source schema. Use the following endpoint:

`GET /v2/db-export`

This endpoint is available in Swagger under the **UI Internal** definition.

Parameters

Parameter	Type	Description
`datasets*`	List of strings	List of datasets to export. You need to give at least one **valid** dataset name into the list.
`schema`	String	Name of the schema/tenant where you want to perform the export. \ *Default value:* **public**
`tables`	String	List of tables to export on the given schema & dataset(s). \ If you leave it empty, the following tables will be exported altogether: **rule_repo, owl_catalog, owl_rule, alert_cond, owl_check_repo, job_schedule, alert_output**

Response

The endpoint returns **200:OK** if the export was successful. The response includes a list of SQL INSERT statements in JSON format. Copy this response. It is required for the next step in the

process.

Import content into target

Import content into the target environment. Use the following endpoint:

`POST /v2/db-import`

This endpoint is available in Swagger under the **UI Internal** definition.

Request body

In the request body, paste in the JSON response from the export endpoint (`/v2/db-export`). Be sure to check the JSON parameters for differences in connections, agents, spark and environment configurations. Migrating to different environments typically requires the payload to be modified. If importing into the same environment, edit the payload to avoid key constraint conflicts.

> **Note** The import is rung on non-transactional mode, any error happens in the middle, the saved items will be left in the database as is.

Response

The endpoint returns **200:OK** if the import was successful. At this point, you may open the user interface and verify that the rules were successfully transferred.

Integration API

The Integration API lets you send Collibra DQ rules, ML layer findings, and associated data quality scores to Collibra Platform without a dependency on Edge. When you configure the integration, your data quality metrics and scores update in real-time within Collibra Platform, empowering you with a robust view of the health of your organization's data.

Prerequisites

- Your Collibra Platform profile must have Sysadmin and DataSteward global roles to manage all assets and objects of the integration.
- Both Collibra Platform and Data Quality & Observability Classic must be version 2023.05+.

Add connection credentials

POST `/v3/dgc/integrations/connections/credentials`

Creates a new integration of your Collibra Platform instance with Data Quality & Observability Classic.

Request

Path parameters

Enter the parameters of your Collibra Platform instance and then click **Execute**.

> **Note** You can execute the GET /v3/dgc/integrations/connections/credentials endpoint to return the details that these parameters require.

The following table describes the parameters available in Swagger.

Parameter	Type	Description	Required
serviceType	string	Data Intelligence Cloud	✓ Yes
serviceUrl	string	The URL of your Collibra Platform instance. For example, `https://dg-example.collibra.com/`	✓ Yes
authMode	string	Is always `basic`	✓ Yes

Parameter	Type	Description	Required
serviceAccount	string	The account name of your Collibra Platform instance. For example, `integration.user`	✓ Yes
password	string	The password of your Collibra Platform instance.	✓ Yes

Response examples

```
{
  "status": "success",
  "msg": "Credentials stored successfully."
}
```

```
{
  "status": "failed",
  "msg": "message"
}
```

Get connection credentials

GET `/v3/dgc/integrations/connections/credentials`

Returns the connection credentials of your Collibra Platform instance.

Request

Path parameters

There are no parameters for this request.

Click **Execute** to return your Collibra Platform connection credentials.

Response examples

The following table shows examples of the properties that return when you execute the request.

Properties	Type	Description
id	string	The Id of your Collibra Platform instance.
serviceType	string	Data Intelligence Cloud
serviceUrl	string	The URL of your Collibra Platform instance.
defaultFields	string	""
authenticationMode	string	The authentication method.
username	string	The username of your Collibra Platform account.


```
[
  {
    "id": "your-id",
    "serviceType": "Data Intelligence Cloud",
    "serviceUrl": "https://dg.collibra.com/",
    "proxyServiceUrl": "https://dg.collibra.com/",
    "defaultFields": "",
    "authenticationMode": "basic",
    "username": "Admin"
  }
]
```


Integrate a dataset into Collibra Platform

POST `/v3/dgc/integrations/import/dgcjson`

Adds a new integration of a given Data Quality & Observability Classic dataset into Collibra Platform.

Request

Path parameters

Enter the name of the dataset to integrate with Collibra Platform and then click **Execute**. The following table describes the parameters available in Swagger.

Parameter	Type	Description	Required
dataset	string	The name of your dataset in Collibra DQ.	✓ Yes
runid	string	Optional. The run Id of your dataset in date time format. Import quality rules from a specific past run.	✗ No

Response examples

```
{
  "msg": "Import to DGC completed successfully......",
  "status": "success"
}
```

```
{
  "status": "failed",
  "msg": "error message",
}
```

Integration toggle

POST `/v3/dgc/integrations/connections/{id}/credentials/toggle`

Turns your Collibra Platform instance credentials on and off. If your credentials are set to false, no data flows between Collibra Platform and Data Quality & Observability Classic.

Request

Path parameters

Enter your Collibra Platform Id and then click **Execute**.

> **Note** You can execute the GET /v3/dgc/integrations/connections/credentials endpoint to return the details that this parameter requires.

The following table describes the path parameter available in Swagger.

Parameter	Type	Description	Required
id	string	The Collibra Platform instance Id with which to integrate Data Quality & Observability Classic.	✓ Yes

Response examples

> **Note** A successful request only returns a response header.

```
 cache-control: no-cache,no-store,max-age=0,must-revalidate
 connection: keep-alive
 content-length: 0
 date: Wed,17 May 2023 03:21:49 GMT
 expires: 0
 keep-alive: timeout=60
 pragma: no-cache
 strict-transport-security: max-age=31536000 ; includeSubDomains
 vary: Origin,Access-Control-Request-Method,Access-Control-
Request-Headers
 x-content-type-options: nosniff
 x-frame-options: SAMEORIGIN
 x-xss-protection: 1; mode=block
```

Error: response status is 500

```
{
  "message": "No value present"
}
```

Cookie

Use cookies file to run Collibra DQ CURL commands.

```
curl -i -X POST -d username=<username> -d password=<password>
http://localhost:9000/login -c cookies.txt

curl -i --header "Accept:application/json" -X GET -b cookies.txt
"http://localhost:9000/v2/getsecuritymap"
```

Multi-Tenant without subdomain in URL (tenant parameter required):

```
curl -i -X POST -d username=<username> -d password=<password> -d
tenant=<tenant> -d tenant=public http://localhost:9000/login -c
cookies.txt

curl -i --header "Accept:application/json" -X GET -b cookies.txt
"http://localhost:9000/v2/getsecuritymap"
```

JWT

This section shows you how to use /auth/signin endpoints to generate JSON web tokens (JWTs) so you can run Collibra DQ CURL commands in your environment.

The endpoints that let you generate JWTs include the following:

- **/v3/auth/signin**
- **/v2/auth/signin**
- **/auth/signin**

> **Tip** It is best practice to use the /v3/auth/signin endpoint to generate a JWT.

After you have a JWT, you can use it to authorize any API call in Collibra DQ, depending on your user-based role.

> **Note** JWT generated in Collibra DQ use a HMACSHA256 keyed hash algorithm.

Single-tenant

To get a JWT to authorize calls in a single-tenant environment, execute the following request:

```
TOKEN=$(curl -s -X POST http://localhost:9000/auth/signin -H
"Content-Type:application/json" -d "{\"user-
name\":\"<username>\", \"password\":\"<password>\"}"| jq -r
'.token')

curl -i -H 'Accept: application/json' -H "Authorization: Bearer
${TOKEN}" http://localhost:9000/v2/getsecuritymap
```

Multi-tenant

To get a JWT to authorize calls in a multi-tenant environment, execute the following request without a sub-domain in the URL:

> **Note** The `iss` tenant parameter is required.

```
TOKEN=$(curl -s -X POST http://localhost:9000/auth/signin -H
"Content-Type:application/json" -d "{\"user-
name\":\"<username>\", \"password\":\"<password>\", \"is-
s\":\"<tenant>\"}"| jq -r '.token')

curl -i -H 'Accept: application/json' -H "Authorization: Bearer
${TOKEN}" http://localhost:9000/v2/getsecuritymap
```

Without headers and jq display

```
curl -H 'Accept: application/json' -H "Authorization: Bearer
${TOKEN}" http://localhost:9000/v2/getsecuritymap | jq '.' | cat
```

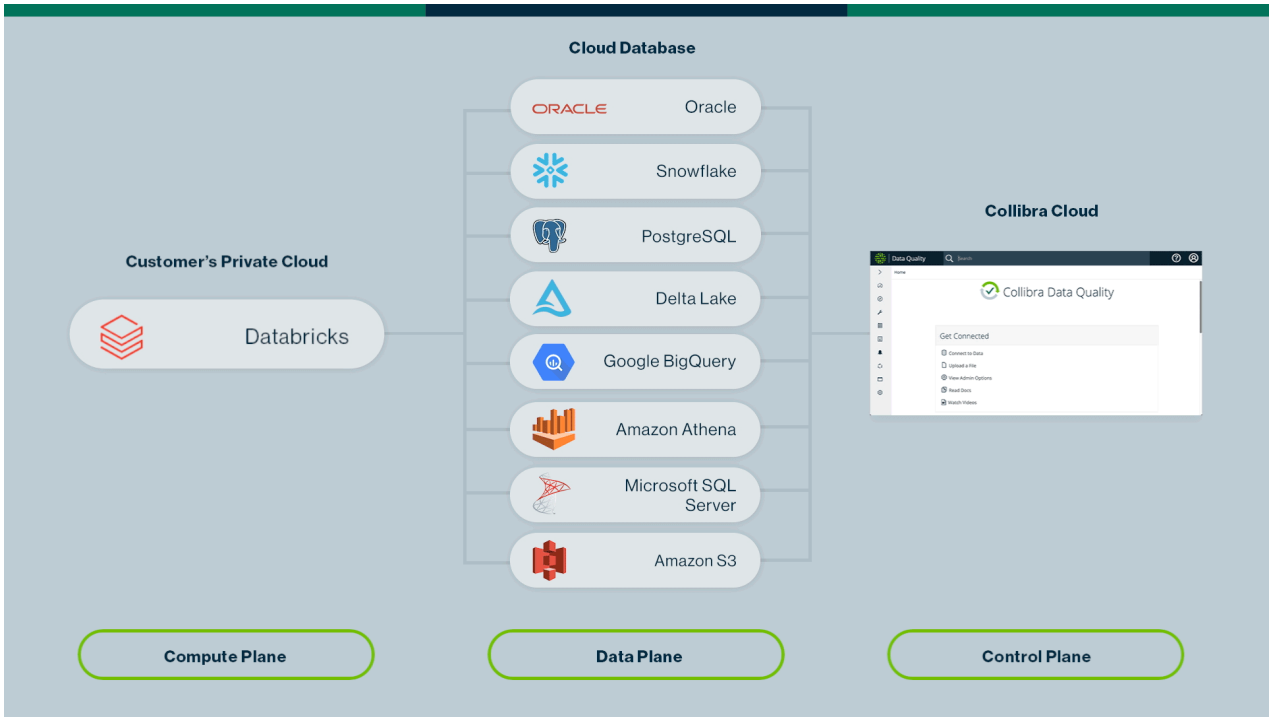
Notebook

Collibra DQ + Databricks

Introduction

This page provides guidance to help you upload Collibra DQ jars to a Databricks cluster and run a Collibra DQ job by invoking Collibra DQ APIs. Some examples are provided in this page.

Architecture



Collibra DQ Environment Setup

This section explains the steps involved in setting up your Collibra DQ environment in Databricks. This is the first step towards invoking Collibra DQ APIs in Databricks.

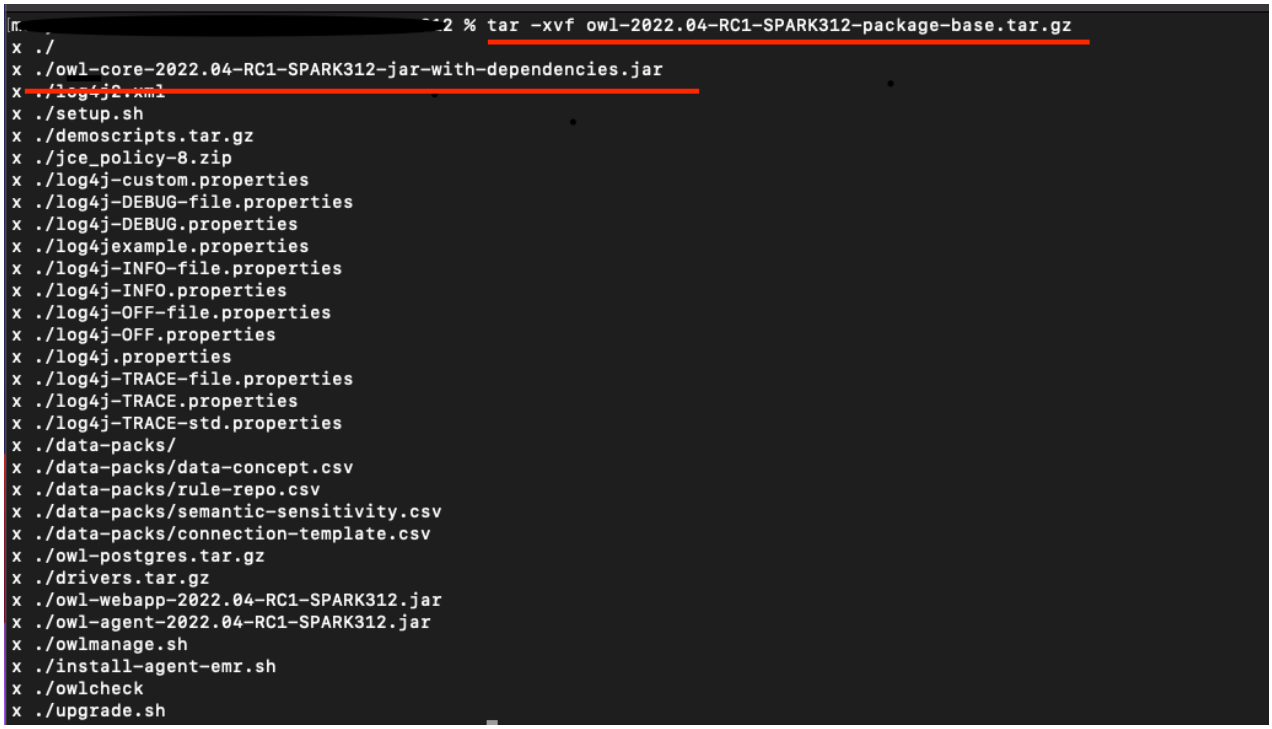
Step 1: Extract the Collibra DQ core jar from Collibra DQ package zipped file.

The first step is to get the CDQ jar file. Once you have the cdq jar package file, you can get the jars by running the following commands:

```
tar -xvf package.tar.gz
```

For example, `tar -xvf owl-2022.04-RC1-default-package-base.tar.gz`

Running this command instructs tar to extract the files from the zipped file. From the list of the files, you need to upload the owl-core-xxxx-jar-with-dependancies.jar to our Databricks file system which will be explained in the next section.



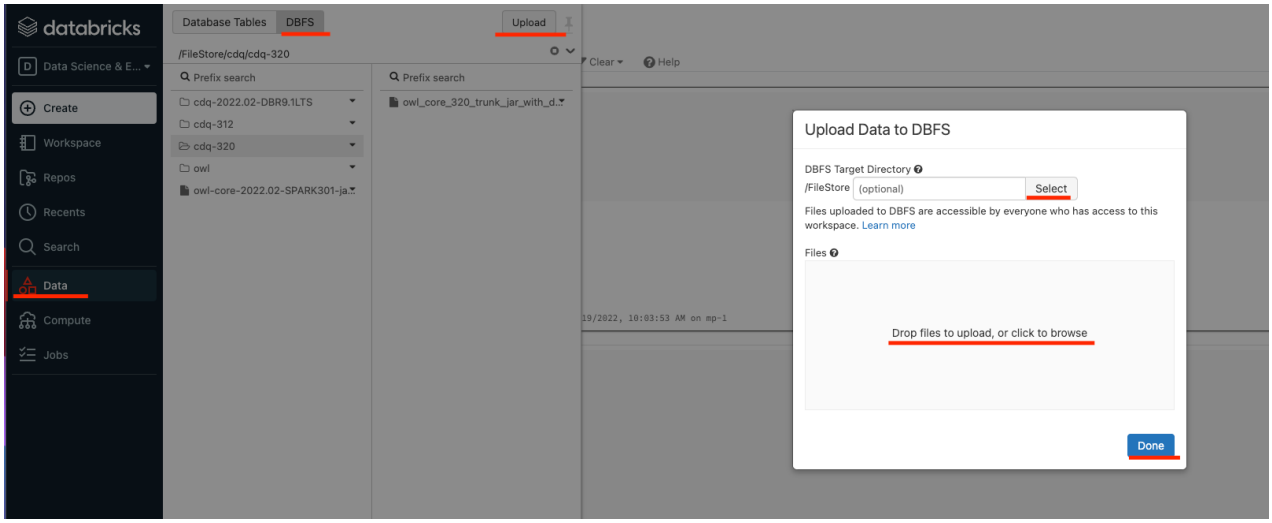
Step 2: Upload the Collibra DQ core jar file to Databricks file system using UI

The jars should be manually uploaded in Databricks file system. Below is the quick summary of the steps. For more information on uploading files in Databricks, refer to the official Databricks documentation.

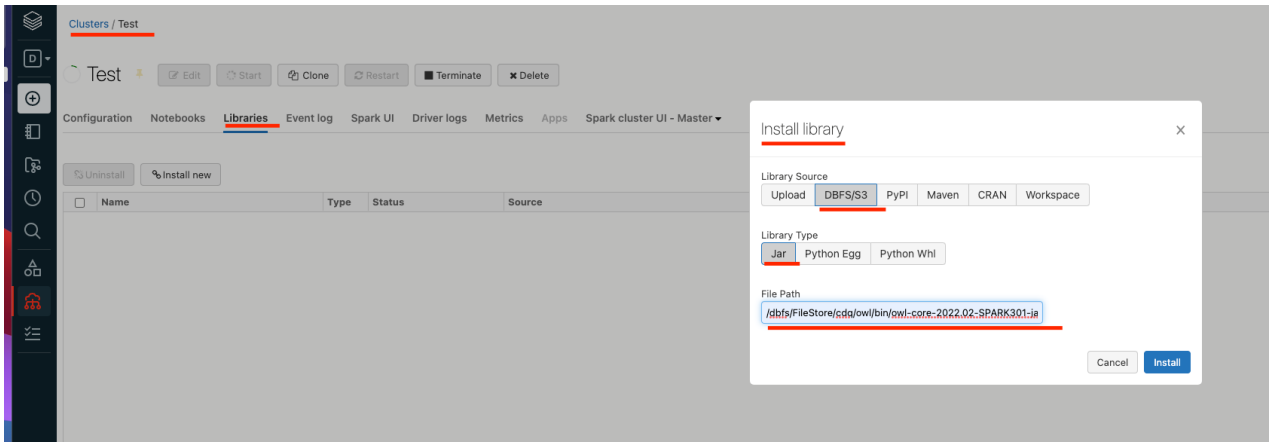
a. Login to your Databricks account.

b. Click  **Data** in the sidebar.

c. Click **DBFS** at the top of the page.
d. Upload the owl-core-xxxx-jar-with-dependancies.jar to your desired path.



Step 3: Install Collibra DQ library in your Databricks cluster



Once this step is completed, you can create a workspace and start using Collibra DQ APIs.

Step 4: Update the Spark Config in your Databricks cluster

> **Note** This step is required if your cluster uses Spark 3.2.1 and onward.

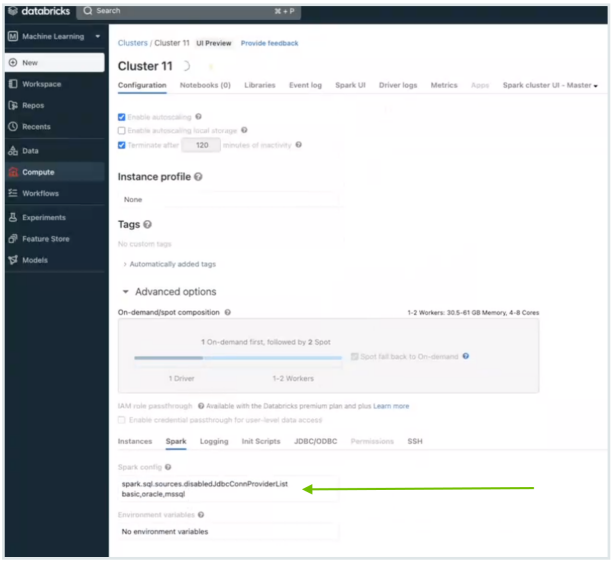
When you bring Collibra DQ jars into Databricks, you are required as of the Collibra DQ 2023.01 release to set the property

`spark.sql.sources.disabledJdbcConnProviderList='basic,oracle,mssql'` at either the Spark Cluster-level or the SparkSession-level before using Collibra DQ's set of functions for Spark profiles 3.2.1 and onwards.

Setting the property in the Spark Config of your Databricks cluster

If you have an active Databricks cluster, you can set the property in the Spark Config section of your Databricks cluster.

1. On your Databricks cluster configuration page, click the **Advanced Options** toggle and select the **Spark** tab.
2. In the **Spark Config** section, enter the following configuration property as one key-value pair per line:
   `spark.sql.sources.disabledJdbcConnProviderList basic,oracle,mssql`



Setting the property in a SparkSession

If you have your own Spark data source, you can set the property in the SparkSession. For example:
`SparkSession.getActiveSession.get.conf.set`

```
("spark.sql.sources.disabledJdbcConnProviderList",
"basic,oracle,mssql")
```

Step 5 (Optional): Update datasource pool size

> **Note** This step is necessary if you get *PoolExhaustedException* when you call Collibra DQ APIs.

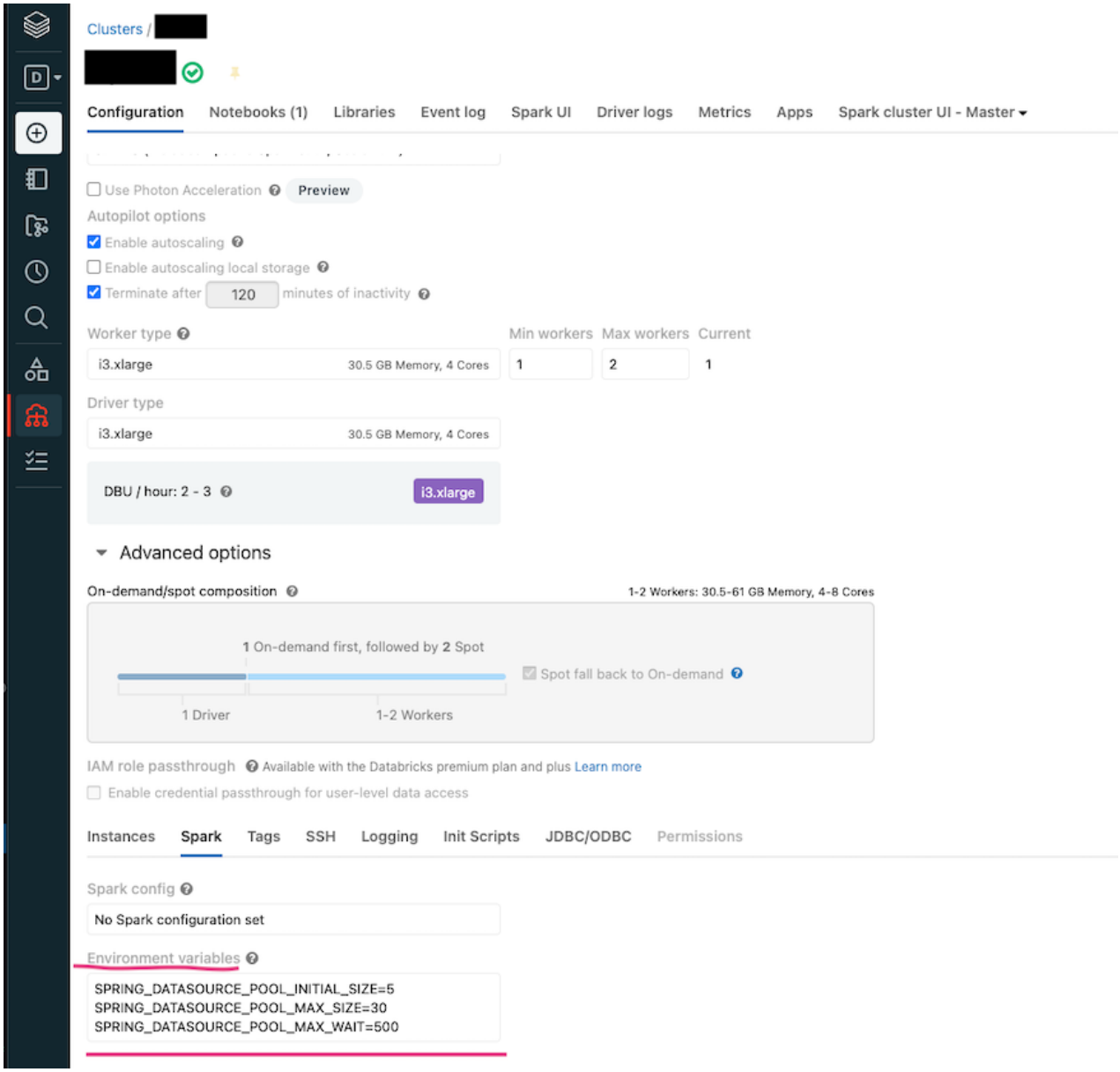
Update the connection pool size in the Spark environment using the following environment variables:

```
SPRING_DATASOURCE_POOL_MAX_WAIT=500

SPRING_DATASOURCE_POOL_MAX_SIZE=30

SPRING_DATASOURCE_POOL_INITIAL_SIZE=5
```

For more information on setting up Databricks environment variables, refer to the official Databricks documentation.



Enabling multi-tenancy for notebook APIs

The following Scala code enables multi-tenancy for notebook APIs by setting up database connection parameters and assigning them to the opt object. In this example, "finance" is the name of the non-public tenant for Collibra DQ.

```
// Enable Multi Tenancy for Notebook API
val tenantName = "finance"
val pgHost = "xxxx.amazonaws.com"
val pgDatabase = "postgres"
val pgSchema = "public"
val pgUser = "???????"
val pgPass = "????"
val pgPort = "0000"

// Assign these values to the opt object
val opt = new OwlOptions()
opt.port = s"jd-
bc:-
post-
gresql://$pgHost:$pgPort/$pgDatabase?currentSchema=$tenantName"
```

After you run the job, check the specific tenant's job page for the results.

Collibra DQ Working Example in DataBricks

Import Collibra DQ library

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
import scala.collection.JavaConverters._
import java.util.Date
import java.time.LocalDate
import java.text.SimpleDateFormat
import spark.implicits._
import java.util.{ArrayList, List, UUID}
// CDQ Imports
import com.owl.core.Owl
import com.owl.common.options._
import com.owl.common.domain2._
import com.owl.core.util.OwlUtils
spark.catalog.clearCache
```

Bringing customer data from another database

Bringing customer data from a file

```
val df = (spark.read
.format("csv").option("header", true).option("delimiter", ","
.load(dbfs:/FileStore/nyse.csv")
)
```

Bringing customer data from a database

```
val connProps = Map(
"driver" -> "org.postgresql.Driver",
"user" -> "your-username",
"password" -> "your-password",
"url" -> "jdbc:postgresql://abc:1234/postgres",
"dbtable" -> "public.example_data")
//--- Load Spark DataFrame ---//
val df = spark.read.format("jdbc").options(connProps).load dis-
play(df)
display(df) // view your data
```

Variables to set up Collibra DQ Metastore database location

```
val pgHost = "xxxx.amazonaws.com"
val pgDatabase = "postgres"
val pgSchema = "public"
val pgUser = "???????"
val pgPass = "????"
val pgPort = "0000"
```

Create a Collibra DQ Test (Rules) and detect breaks

> **Note** If the rules are already created and assigned to a dataset from the UI, calling owlcheck() automatically executes all the rules associated with the given dataset and there is no need to recreate the rule from notebook.

```
val dataset = "cdq_notebook_db_rules"
var date = "2018-01-11"

// Options
val opt = new OwlOptions()
opt.dataset = dataset
opt.runId = date
opt.host = pgHost
opt.port = pgPort
opt.pgUser = pgUser
opt.pgPassword = pgPass

opt.setDatasetSafeOff(false) // to enable historical overwrite
of dataset

// Create a simple rule and assign it to dataset
val simpleRule = OwlUtils.createRule(opt.dataset)
      simpleRule.setRuleNm("nyse-stocks-symbol")
      simpleRule.setRuleValue("symbol == 'BHK'")
      simpleRule.setRuleType("SQLG")
      simpleRule.setPerc(1.0)
      simpleRule.setPoints(1)
      simpleRule.setIsActive(1)
      simpleRule.setUserNm("admin")
      simpleRule.setPreviewLimit(8)

// Create a rule from generic rules that are created from UI:
val genericRule = OwlUtils.createRule(opt.dataset)
    genericRule.setRuleNm("exchangeRule") // this could be any
name
    genericRule.setRuleType("CUSTOM")
    genericRule.setPoints(1)
    genericRule.setIsActive(1)
    genericRule.setUserNm("admin")
    genericRule.setRuleRepo("exchangeCheckRule"); // Validate
the generic rule name //from UI
    genericRule.setRuleValue("EXCH") // COLUMN associate with
the rule


// Pre Routine
val cdq = com.owl.core.util.OwlUtils.OwlContext(df, opt)
cdq.removeAllRules(opt.dataset)
.register(opt)
.addRule(simpleRule)

// Scan
cdq.owlCheck()
```

```
val results = cdq.hoot() // returns object Hoot, not a DataFrame
//See Json Results(Option for downstream processing)
println("--------------Results:----------------\n")
println(results) //optional

//Post Routine, See DataFrame Results (Option for downstream pro-
cessing)
val breaks = cdq.getRuleBreakRows("nyse-stocks-symbol")
println("--------------Breaks:----------------\n")
display(breaks)

// Different Options for handling bad records
val badRecords = breaks.drop("_dataset","_run_id", "_rule_name",
"owl_id")
display(badRecords)

val goodRecords = df.except(badRecords)
display(goodRecords)
```

Write the breaks (bad records) DataFrame to a Parquet file

```
// Remove the file if it exists
dbutils.fs.rm("/tmp/databricks-df-example.parquet", true)
breaks.write.parquet("/tmp/databricks-df-example.parquet")
```

The following image shows the code snippet and the result in Databricks:



```
opt.pgUser = pgUser
opt.pgPassword = pgPass

// Create a template rule
val templateRule = OwlUtils.createRule(opt.dataset)
      templateRule.setRuleNm("rule1_nyse-stocks-symbol")
      templateRule.setRuleValue("symbol == 'BHK'") /* if the symbol == BHK break the rule */
      templateRule.setRuleType("SQLG")
      templateRule.setPoints(5)  // Whole int 1 and greater
      templateRule.setIsActive(1)
      templateRule.setUserNm("admin")
      templateRule.setPreviewLimit(8)

// Create a second template rule
val templateRule1 = OwlUtils.createRule(opt.dataset)
      templateRule1.setRuleNm("rule2_nyse-stocks-symbol")
      templateRule1.setRuleValue("symbol == 'MSP'") /* if the symbol == MSP break the rule: */
      templateRule1.setRuleType("SQLG")
      templateRule1.setPoints(5)  // Whole int 1 and greater
      templateRule1.setIsActive(1)
      templateRule1.setUserNm("admin")
      templateRule1.setPreviewLimit(8)

// Scan
val cdq = com.owl.core.util.OwlUtils.OwlContext(df, opt)
.register(opt)
.addRule(templateRule).addRule(templateRule1)
cdq.owlCheck()

// create another dataset and re-use the previous rules from the other dataset
val dataset2 ="nyse-dataset"
var opt1 = new OwlOptions()
opt1.dataset = dataset2
opt1.runId = date
opt1.host = pgHost
opt1.port = pgPort
opt1.pgUser = pgUser
opt1.pgPassword = pgPass
opt1.setDatasetSafeOff(false)

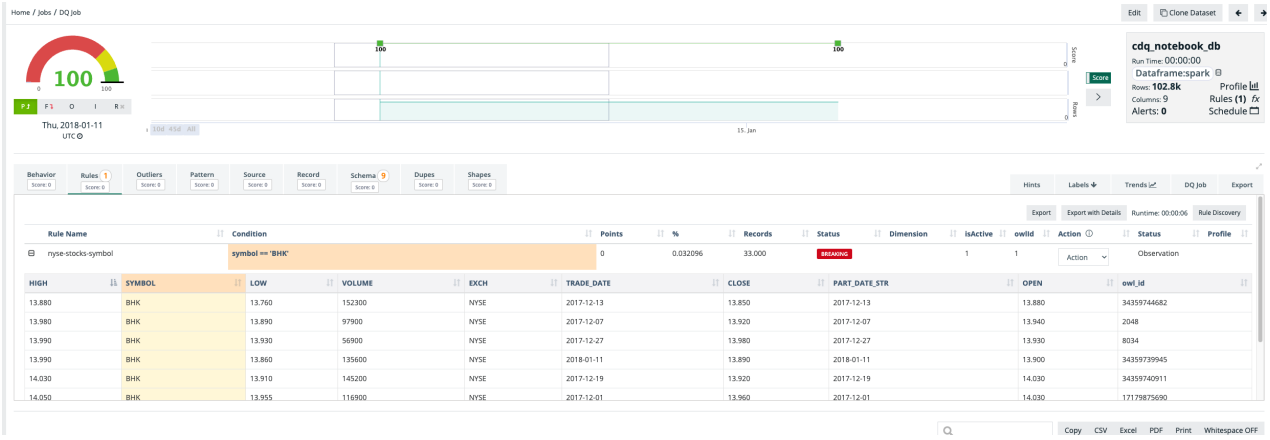
val cdq1 = OwlUtils.OwlContext(df, opt1)
cdq1.register(opt1)
// get the rules from "master_template_rules_dataset"
val rules = cdq1.getRulesByDataset(dataset)
if (rules.size > 0)
  cdq1.removeAllRules(dataset2).addRules(rules, dataset2)
cdq1.owlCheck()
val breaks = cdq1.getRuleBreakRows()

//display(breaks)

breaks.show()
if(breaks.count() > 0 ) {
  //val badRecords = breaks.drop("_dataset", "_runId", "_rulename", "owl_id")
  val badRecords = breaks.drop("_dataset", "_runId", "_rulename", "owl_id")
  display(badRecords)
  val goodRecords = df.except(badRecords)
  display(goodRecords)
 }
```

Steps to reassign the rules of one dataset to another via the API.

The breaks and the rules can also be viewed in Collibra DQ web.



Create a Collibra DQ Test (Profile)

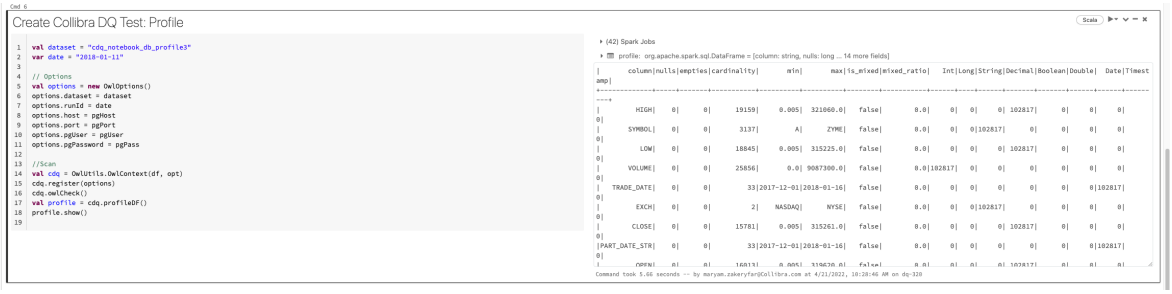
```
val dataset = "cdq_notebook_nyse_profile"

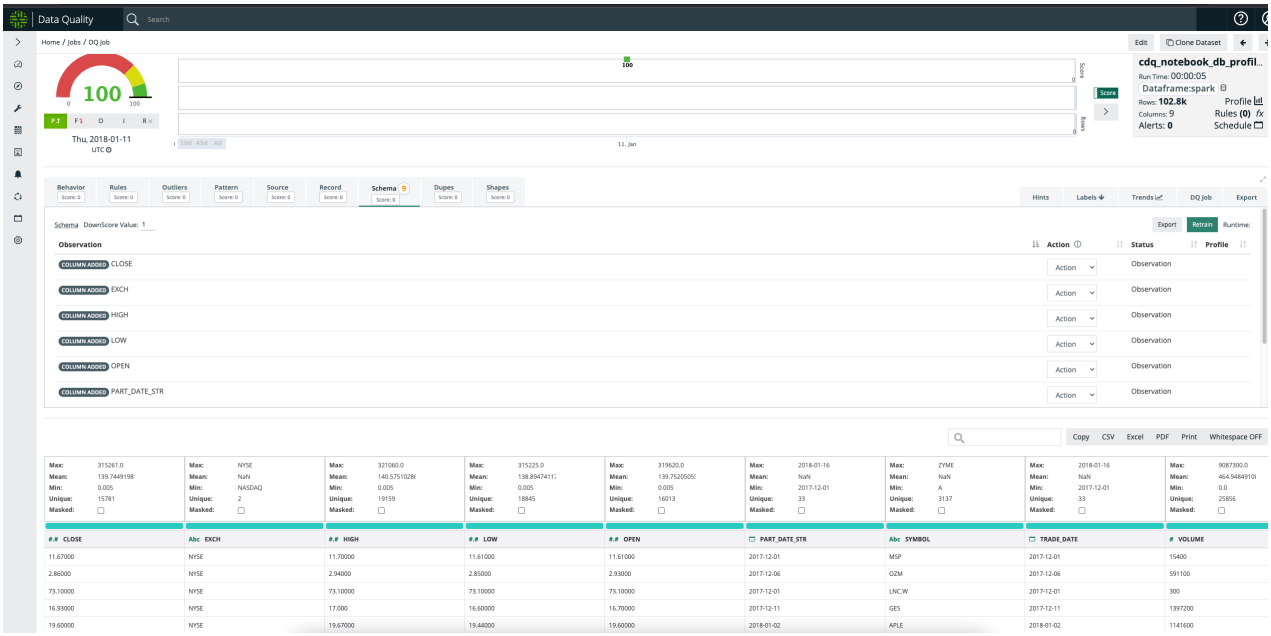
val runList = List("2018-01-01", "2018-01-02", "2018-01-03",
"2018-01-04", "2018-01-05"
for(runID <- runList {
// Options
val options = new OwlOptions()
options.dataset = dataset
options.host = pgHost
options.port = pgPort
options.pgUser = pgUser
options.pgPassword = pgPass

//Scan
val profileOpt = new ProfileOpt
profileOpt.on = true
profileOpt.setShape(true)
profileOpt.setShapeSensitivity(5.0)
profileOpt.setShapeMaxPerCol(10)
profileOpt.setShapeMaxColSize(10)
profileOpt.setShapeGranular(true)
profileOpt.behaviorEmptyCheck = true
profileOpt.behaviorMaxValueCheck = true
profileOpt.behaviorMinValueCheck = true
profileOpt.behaviorNullCheck = true
profileOpt.behaviorRowCheck = true
profileOpt.behaviorMeanValueCheck = true
profileOpt.behaviorUniqueCheck = true
profileOpt.behaviorMinSupport = 5 // default is 4
profileOpt.behaviorLookback = 5
options.profile = profileOpt

var date = runId
var df_1 = df.where($"TRADE_DATE"===s"$date")

//Scan
val cdq = OwlUtils.OwlContext(df_1, options)
cdq.register(opt)
cdq.owlCheck()
val profile = cdq.profileDF()
profile.show()
}
```


Create Collibra DQ Test: Profile
Scala
val dataset = "cdq_notebook_db_profile3"
var date = "2018-01-11"
// Options
val options = new DwlOptions()
options.dataset = dataset
options.runId = date
options.host = pgHost
options.port = pgPort
options.pgUser = pgUser
options.pgPassword = pgPass
//Scan
val cdq = DwlUtils.DwlContext(df, opt)
cdq.register(options)
cdq.owlCheck()
val profile = cdq.profileDF()
profile.show()
(42) Spark Jobs
profile: org.apache.spark.sql.DataFrame = [column: string, nulls: long ... 14 more fields]
column	nulls	empties	cardinality	min	max	is_mixed	mixed_ratio	Int	Long	String	Decimal	Boolean	Double	Date	Timestamp
HIGH	0	0	19159	0.005	321060.0	false	0.0	0	0	0	102817	0	0	0
SYMBOL	0	0	3137	A	ZYME	false	0.0	0	0	102817	0	0	0	0
LOW	0	0	18845	0.005	315225.0	false	0.0	0	0	0	102817	0	0	0
VOLUME	0	0	25856	0.0	9087300.0	false	0.0	102817	0	0	0	0	0	0
TRADE_DATE	0	0	33	2017-12-01	2018-01-16	false	0.0	0	0	0	0	0	0	102817
EXCH	0	0	2	NASDAQ	NYSE	false	0.0	0	0	102817	0	0	0	0
CLOSE	0	0	15781	0.005	315261.0	false	0.0	0	0	0	102817	0	0	0
PART_DATE_STR	0	0	33	2017-12-01	2018-01-16	false	0.0	0	0	0	0	0	0	102817
Command took 5.66 seconds -- by maryam.zakeryfar@Collibra.com at 4/21/2022, 10:28:46 AM on dq-320


Data Quality
Search
Home / Jobs / DQ Job
Edit
Clone Dataset
100
Thu, 2018-01-11
UTC
cdq_notebook_db_profil...
Run Time: 00:00:05
Dataframe:spark
Rows: 102.8k
Columns: 9
Alerts: 0
Profile
Rules (0)
Schedule
Behavior
Rules
Outliers
Pattern
Source
Record
Schema 9
Dupes
Shapes
Hints
Labels
Trends
DQ Job
Export
Schema DownScore Value: 1
Export
Retrain
Runtime:
Observation
Action
Status
Profile
COLUMN ADDED CLOSE
COLUMN ADDED EXCH
COLUMN ADDED HIGH
COLUMN ADDED LOW
COLUMN ADDED OPEN
COLUMN ADDED PART_DATE_STR
Observation
Copy
CSV
Excel
PDF
Print
Whitespace OFF
CLOSE
EXCH
HIGH
LOW
OPEN
PART_DATE_STR
SYMBOL
TRADE_DATE
VOLUME
11.67000 NYSE 11.70000 11.61000 11.61000 2017-12-01 MSP 2017-12-01 15400
2.86000 NYSE 2.94000 2.85000 2.93000 2017-12-06 OZM 2017-12-06 591100
73.10000 NYSE 73.10000 73.10000 73.10000 2017-12-01 LNC.W 2017-12-01 300
16.93000 NYSE 17.000 16.60000 16.70000 2017-12-11 GES 2017-12-11 1397200
19.60000 NYSE 19.67000 19.44000 19.60000 2018-01-02 APLE 2018-01-02 1141600

Create a Collibra DQ Test (Dupes)

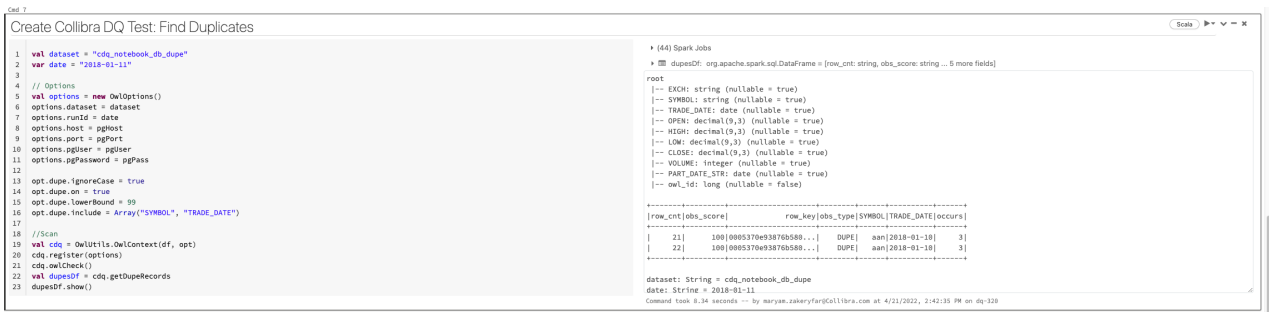
```
val dataset = "cdq_notebook_db_dupe"
var date = "2018-01-11"

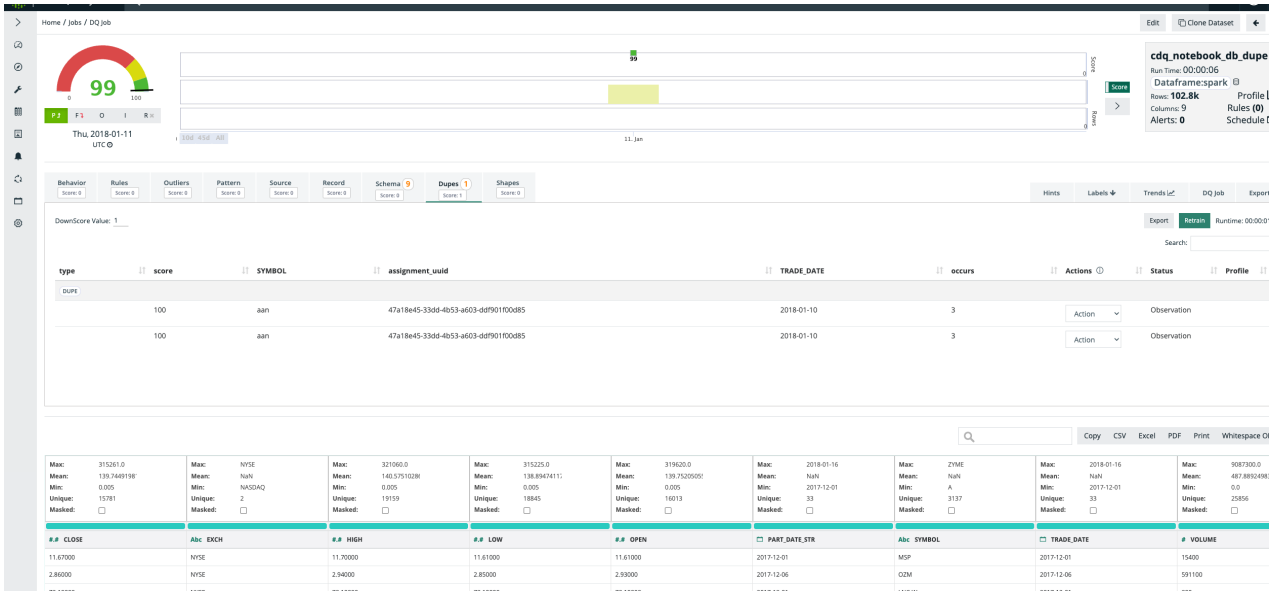
// Options
val options = new OwlOptions()
options.dataset = dataset
options.runId = date
options.host = pgHost
options.port = pgPort
options.pgUser = pgUser
options.pgPassword = pgPass

opt.dupe.ignoreCase = true
opt.dupe.on = true
opt.dupe.lowerBound = 99
opt.dupe.include = Array("SYMBOL", "TRADE_DATE")

//Scan
val cdq = OwlUtils.OwlContext(df, opt)
cdq.register(options)
cdq.owlCheck()

val dupesDf = cdq.getDupeRecords
dupesDf.show()
```




Home / Jobs / DQ Job
Edit
Clone Dataset
99
Thu, 2018-01-11
UTC
cdq_notebook_db_dupe
Run Time: 00:00:06
Dataframe:spark
Rows: 102.8k
Columns: 9
Alerts: 0
Rules (0)
Behavior
Rules
Outliers
Pattern
Source
Record
Schema 9
Dupes 1
Shapes
Hints
Labels
Trends
DQ Job
DownScore Value: 1
Export
Retrain
Search:
type
score
SYMBOL
assignment_uuid
TRADE_DATE
occurs
Actions
Status
Profile
DUPE
100
aan
47a18e45-33dd-4b53-a603-ddf901f00d85
2018-01-10
3
Action
Observation
Copy
CSV
Excel
PDF
Print
CLOSE
EXCH
HIGH
LOW
OPEN
PART_DATE_STR
SYMBOL
TRADE_DATE
VOLUME
11.67000
NYSE
11.70000
11.61000
11.61000
2017-12-01
MSP
2017-12-01
15400
2.86000
NYSE
2.94000
2.85000
2.93000
2017-12-06
OZM
2017-12-06
591100

Create a Collibra DQ Test (Outlier)

```
import scala.collection.JavaConverters._
import java.util
import java.util.{ArrayList, List, UUID}

val dataset = "cdq_notebook_db_outlier"
var date = "2018-01-11"

// Options
val options = new OwlOptions()
options.dataset = dataset
options.runId = date
options.host = pgHost
options.port = pgPort
options.pgUser = pgUser
options.pgPassword = pgPass

opt.dupe.on = false

val dlMulti: util.List[OutlierOpt] = new util.ArrayList[Out-
lierOpt]
val outlierOpt = new OutlierOpt()
outlierOpt.combine = true
outlierOpt.dateColumn = "trade_date"
outlierOpt.lookback = 4
outlierOpt.key = Array("symbol")
outlierOpt.include = Array("high")
outlierOpt.historyLimit = 10
dlMulti.add(outlierOpt)

opt.setOutliers(dlMulti)

val cdq = OwlUtils.OwlContext(df, opt)
        .register(opt)

cdq.owlCheck
val outliers = cdq.getOutliers()
outliers.show
outliers.select("value")
```

Create Collibra DQ Test (ValidateSource)

```
import com.owl.common.options.SourceOpt
import java.util
import java.util.{ArrayList, List}

var opt = new OwlOptions()
val dataset = "weather-validateSrc"
opt.setDataset(dataset)
opt.runId = "2018-02-23"
clearPreviousScans(opt)
val src = Seq(
  ("abc", true, 55.5, "2018-02-23 08:30:02"),
  ("def", true, 55.5, "2018-02-23 08:30:02"),
  ("xyz", true, 55.5, "2018-02-23 08:30:02")
).toDF("name", "sunny", "feel-like-temp", "d_date")

val target = Seq(
  ("abc", 72, false, 55.5, "2018-02-23 08:30:02"), // true to
false
  ("xyz", 72, true, 65.5, "2018-02-23 09:30:02") // 08 to 09
).toDF("name", "temp", "sunny", "feel-like-temp", "d_date")

val optSource = new SourceOpt
optSource.on = true
optSource.dataset = dataset
optSource.key = Array("name")
opt.setSource(optSource)

//scan
val cdq = OwlUtils.OwlContext(src, target, opt)
  .register(opt)
cdq.owlCheck

val breakCountDf = cdq.getSourceBreaksCount()
breakCountDf.show()
```

Known API Limitations

Collibra DQ activities cannot currently be called independently. DQ Check() function should be called before calling any of the activities. For example, to get the profile DataFrame you should call the following code snippet:

```
cdq.owlCheck()
cdq.getProfileDF()
```

DQ-Databricks Submit

Introduction

In this page we will demonstrate two paths to run a spark submit job on Databricks's cluster. The first approach is to run a DQ spark submit job using Databricks UI and the second approach is by invoking Databricks rest API.

> **Note** These are only examples to demonstrate how to achieve DQ spark submit to Databricks's cluster. These paths are **not** supported in production and DQ team does **not** support any bug coverages or professional services or customer questions for these flows.

Limitations

There are a few limitation to spark-submit jobs in Databricks listed in this section: https://docs.databricks.com/jobs.html#create-a-job. Also, spark-submit is only on new clusters from both the UI via Jobs or calling the REST APIs. See Step 4 in: https://docs.databricks.com/jobs.html#create-a-job where it lists that spark-submit is handled by new clusters only.

Steps to create and run a spark submit job from Databricks UI:

1. Grant Collibra DQ Database access to your instance of Databricks.
2. Upload DQ jars in Databricks File System (DBFS).
3. Set up environment variables for your new cluster.
4. Prepare the DQ JSON payload.
5. Create and Run your job.
6. View the status and result of your job from the DQ Jobs page.

Database access

To begin, ensure that sure your Databricks instance has access to the DQ Database.

The entire subnet must be whitelisted to connect to the database. As specified in Databricks' documentation on subnets, Databricks must have access to at least two subnets for each database. To connect to the two Databricks subnets where the nodes will be instantiated, you must allow AWS to whitelist your IP address range.

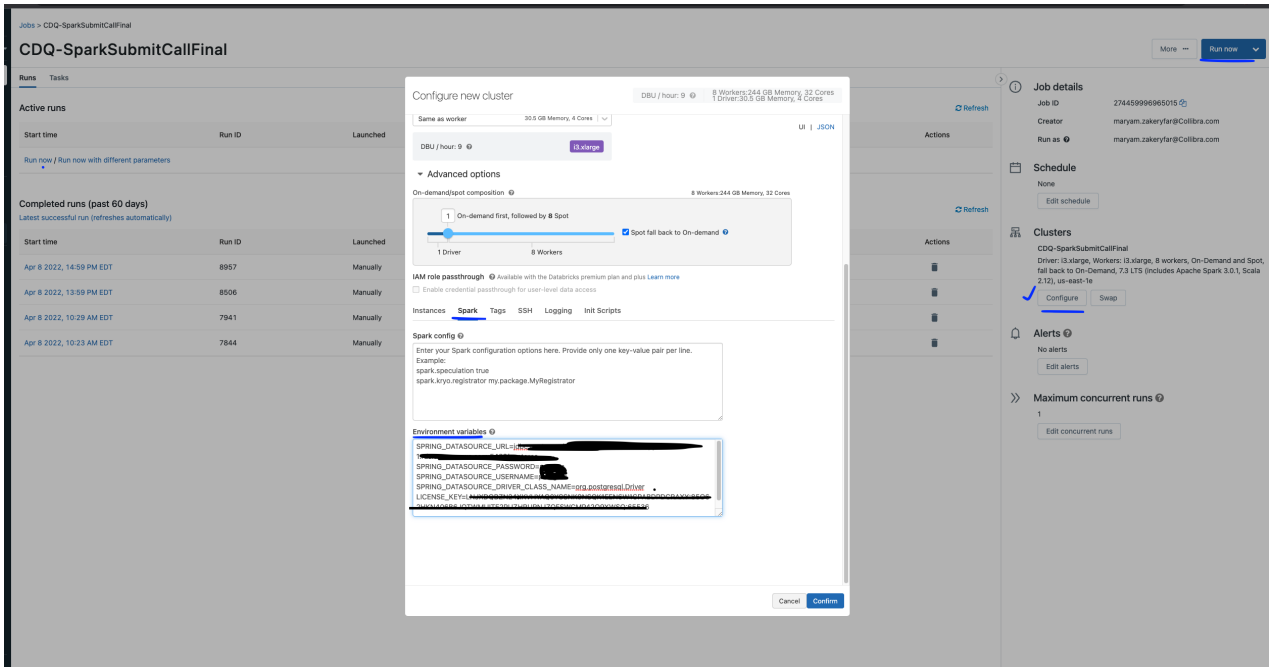
Upload DQ's jars in DBFS

The jars should be manually uploaded in Databricks file system. The steps can be found on Databricks website: https://docs.databricks.com/data/databricks-file-system.html#access-dbfs.

Environment variables for the new cluster:

Here is the documentation from Databricks about how to set up environment variables: https://docs.databricks.com/clusters/configure.html#environment-variables

These CDQ environment variables should be set on the new cluster : `SPRING_DATASOURCE_URL=xx\SPRING_DATASOURCE_USERNAME=xx\SPRING_DATASOURCE_DRIVER_CLASS_NAME=xx\LICENSE_KEY=xx // This is DQ's license key`



JSON payload

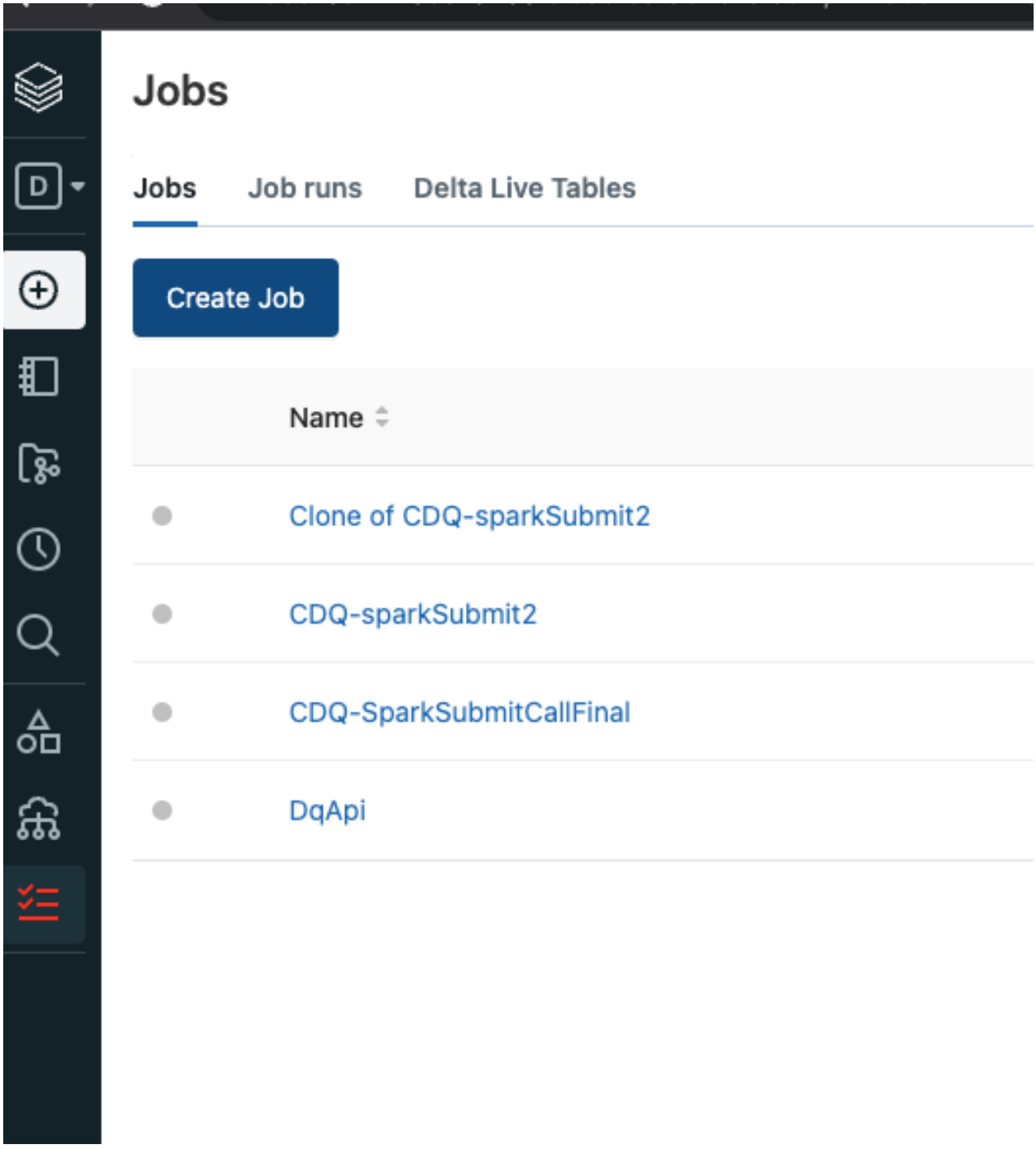
Once the above steps are completed, you can submit a spark submit job with DQ's parameters. Payload parameters can be from DQ's web Run command. You can copy and paste the parameters to prepare a JSON payload. Here is one sample:

```
                    "--class",
                    "com.owl.core.cli.OwlCheck",
                    "dbfs:/FileStore/cdq/owl-core-2022.02-
SPARK301-jar-with-dependencies.jar",
                    "-lib",
                    "dbfs:/FileStore/cdq/owl/drivers/postgres",
                    "-q",
                    "select * from xx.xxx",
                    "-bhlb",
                    "10",
                    "-rd",
                    "2022-03-16",
                    "-driver",
                    "owl.com.org.postgresql.Driver",
                    "-drivermemory",
                    "4g",
                    "-cxn",
                    "metastore",
                    "-h",
                    "xxxx.xxxxxx.amazonaws.com:xxxx/postgres",
                    "-ds",
                    "public.agent_2",
                    "-deploymode",
                    "cluster",
                    "-owluser",
                    "admin"
                ]
```

Run the job

Once you have completed the above steps, you can create a spark submit job through Databricks UI.

You can then add the environment variables to the cluster and click Run on Databricks UI.



Check the result in DQ web:

Once the job is submitted, you can login to your DQ web instance and check the job in the Jobs page.

Spark submit by invoking Databricks REST API

There are public REST APIS available for the Jobs API, including the latest version.

For this path we need to do the steps 1-4 of the the previous section and then call directly the REST API using Postman, or your preferred API testing tool. We assume that as per step 2, CDQ jars are uploaded to the DBFS path in the location dbfs:/FileStore/cdq. Also JDBC postgres driver should be uploaded to DBFS. For example: dbfs:/`FileStore/cdq/owl/drivers/postgres`

Steps:

1. Prepare the DQ JSON Payload.
2. Authenticate the Databricks REST API.

JSON payload

Sample JSON payload:

```
POST /api/2.1/jobs/runs/submit HTTP/1.1Host: xxxxxx.cloud.databricks.com\ `Content-Type: application/json`\ `Authorization: Bearer ~~xxxxxxxxxxxx~~\ Cache-Control: no-cache\ Postman-Token: xxxxxxxx`
```

```
{
  "tasks": [
    {
      "task_key": "CDQ-SparkSubmitCallFinal",
      "spark_submit_task": {
        "parameters": [
          "--class",
          "com.owl.core.cli.OwlCheck",
          "dbfs:/FileStore/cdq/owl-core-2022.02-SPARK301-jar-
with-dependencies.jar",
          "-lib",
          "dbfs:/FileStore/cdq/owl/drivers/postgres",
          "-q",
          "select * from public.agent",
          "-bhlb",
          "10",
          "-rd",
          "2022-03-16",
          "-driver",
          "owl.com.org.postgresql.Driver",
          "-drivermemory",
          "4g",
          "-cxn",
          "metastore",
          "-h",
          "xxxs.amazonaws.com:xxx/postgres",
          "-ds",
          "public.agent_2",
          "-deploymode",
          "cluster",
          "-owluser",
          "admin"
        ]
      },
      "new_cluster": {
        "cluster_name": "",
        "spark_version": "7.3.x-scala2.12",
        "aws_attributes": {
          "zone_id": "us-east-1e",
          "first_on_demand": 1,
          "availability": "SPOT_WITH_FALLBACK",
          "spot_bid_price_percent": 100,
          "ebs_volume_count": 0
        },
        "node_type_id": "i3.xlarge",
        "spark_env_vars": {
          "SPRING_DATASOURCE_URL": "jdbc:postgresql://xxx-xx-
xxs.amazonaws.com:xx/postgres",
```

```
            "SPRING_DATASOURCE_PASSWORD": "xxx",
            "SPRING_DATASOURCE_USERNAME": "xxx",
            "SPRING_DATASOURCE_DRIVER_CLASS_NAME": "org.-
postgresql.Driver",
            "LICENSE_KEY": "xxxx"
          },
          "enable_elastic_disk": false,
          "num_workers": 8
        },
        "timeout_seconds": 0
      }
    ]
}
```

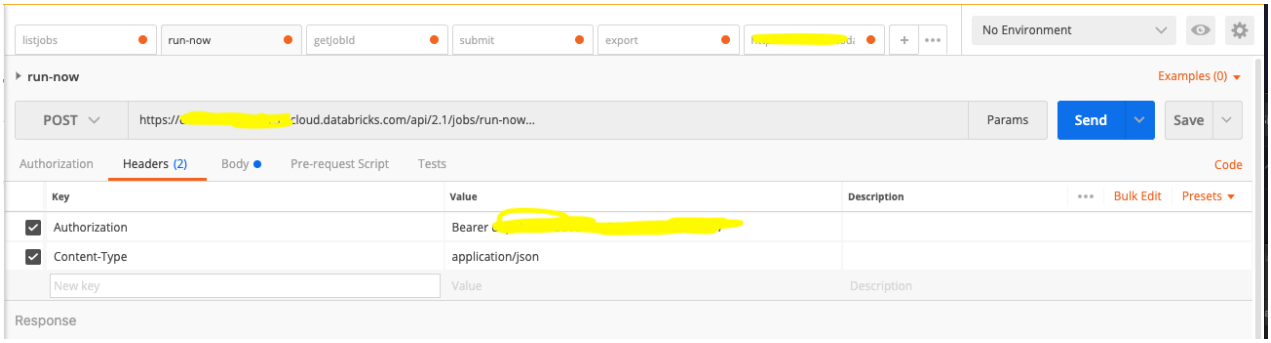
Values to be updated in above payload are:

Cluster variables:`"SPRING_DATASOURCE_URL":"SPRING_DATASOURCE_PASSWORD":"SPRING_DATASOURCE_USERNAME":"LICENSE_KEY": //CDQ License key``

CDQ variables\ Users can customize the variables based on the activity they choose from CDQ Web. They can copy the variables from Run CMD option of their DQ job and paste it in their Json message. ``

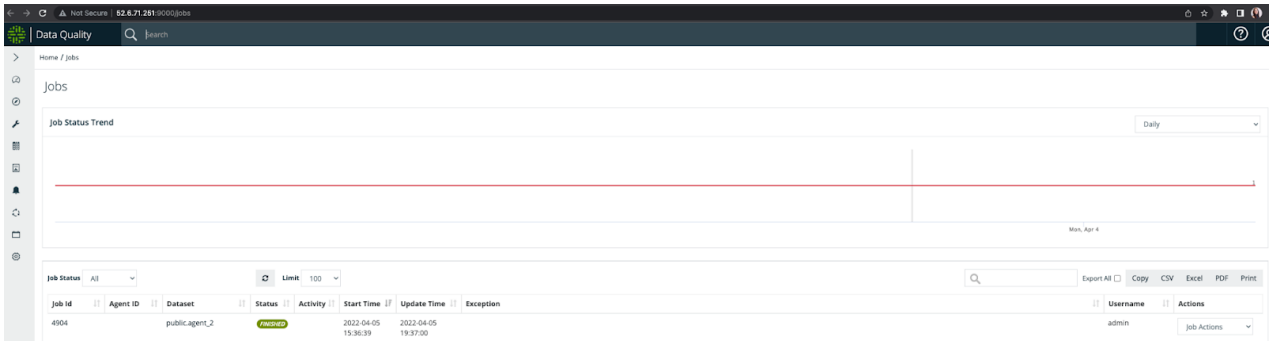
Authenticate the Databricks REST API

Here is the Databricks documentation about how to create a personal access token: https://docs.databricks.com/dev-tools/api/latest/authentication.html



View the job's result in DQ web

You can view the result of your job run by navigating to the DQ Jobs page.



Examples

Programmatic DQ

Don't like leaving your notebook? Want to build DQ into your in-house data quality pipeline? Collibra DQ can do both!

Simple

Load Table use SparkJDBC

```
//--- Configure Table From Database ---//
val connProps = Map (
  "driver"   -> "org.postgresql.Driver",
  "user"     -> s"${user}",
  "password" -> s"${pass}",
  "url"      ->
s"jdbc:postgresql://${host}:${port}/${database}",
  "dbtable"  -> "cdq_test.nyse"
)

//--- Load Spark DataFrame ---//
val jdbcDF = spark.read.format("jdbc").options(connProps).load
jdbcDF.show
```

Configure Collibra DQ Options

Connect to DQ's Metadata Database and control DQ scan options. Wrap sparkDF with DQ context.

```
import com.owl.common.options._
import com.owl.core.util.OwlUtils

val opt = new OwlOptions
//--- Collibra DQ Metastore ---//
opt.host = s"$cdqHost"
opt.port = s"xxxx/postgres?currentSchema=public"
opt.pgUser = s"$cdqUser"
opt.pgPassword = s"$cdqPass"
//--- Run Options ---//
opt.dataset = "cdq_test.nyse"
opt.runId = "2018-01-10"
opt.datasetSafeOff = true

val cdq = OwlUtils.OwlContext(jdbcDF, opt)
```

Register with Catalog and Run Profile

```
//--- Register with Collibra DQ Catalog ---//
cdq.register(opt)
cdq.owlCheck()
//--- Profile Dataset ---//
val profile = cdq.profileDF
profile.show
```

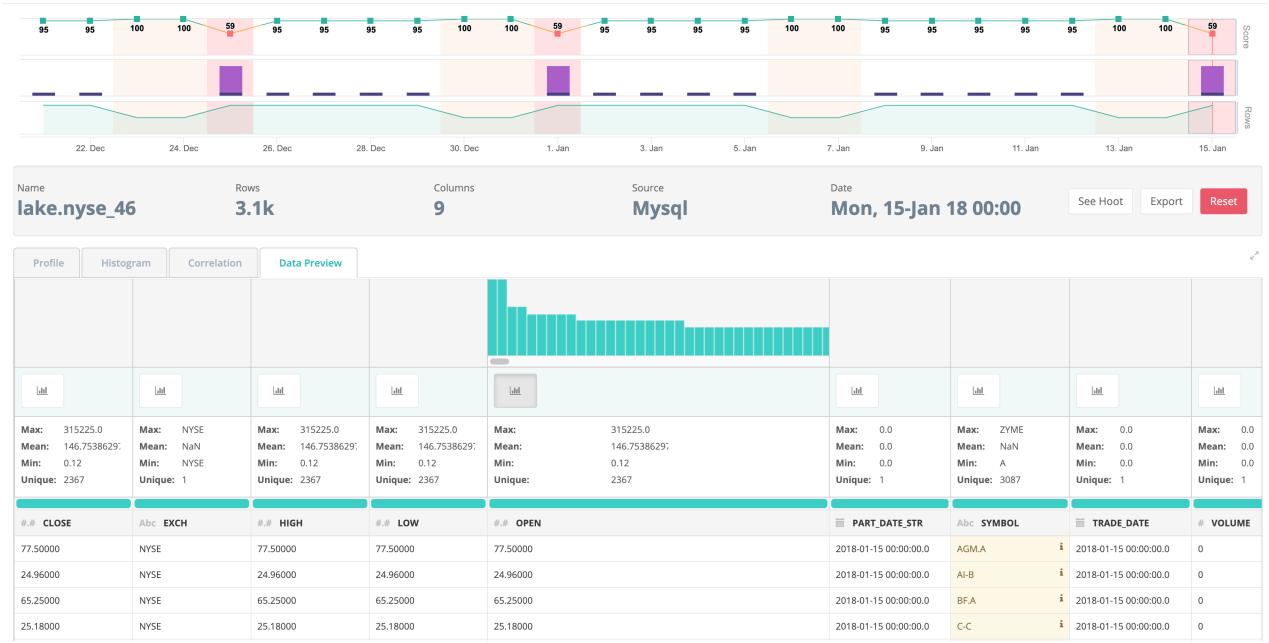
Notice that DQ returns results as Dataframes. This is a fantastic abstraction that allows you to ignore all domain objects and custom types and interact with a scaleable generic result set using common protocols like "where" or "filter" or "save" or "write" all with parallel operations.

```
+--------------+-----+-------+-----------+---+---+--------+-----
------+------+----+------+-------+-------+------+----+---------+
|        column|nulls|empties|cardinality|min|max|is_mixed|-
mixed_
ratio|   Int|Long|String|Decimal|Boolean|Double|Date|Timestamp|
+--------------+-----+-------+-----------+---+---+--------+-----
------+------+----+------+-------+-------+------+----+---------+
|     tenant_
id|
0
|
0
|
60
|
0
|
9
|
false
|
0.0|100000|   0|     0|      0|      0|     0|   0|         0|
|           a11|
0
|
0
|
1
|a11|a11|
false
|
0.0|     0|   0|100000|      0|      0|     0|   0|         0|
|           a10|
0
|
0
|
1
|a10|a10|
false
|
0.0|     0|   0|100000|      0|      0|     0|   0|         0|
|  account_type|    0|      0|          3| 02|
06
|
false
|
0.0|100000|   0|     0|      0|      0|     0|   0|         0|
```

```
|           a13|
0
|
0
|
1
|a13|a13|
false
|
0.0|     0|   0|100000|      0|      0|     0|   0|         0|
|security_
alias|
0
|
0
|
3
|
0
|
2
|
false
|
0.0|100000|   0|     0|      0|      0|     0|   0|         0|
|           a12|
0
|
0
|
1
|a12|a12|
false
|
0.0|     0|   0|100000|      0|      0|     0|   0|         0|
+--------------+-----+-------+----------+---+---+--------+-----
------+------+----+------+-------+-------+------+----+---------+
```

Profile UI

While the spark DF.show() is a nice and convenient output format, you may prefer a rich UI visual that tracks the data tests over time. The UI provides trend analysis, data drift, data relationships and more.



Duplicates

Take duplicate detection for example. A common use case where a business wants to make sure they do not have repeated or duplicate records in a table. Set the lowerBound to the percent fuzzy match you are willing to accept, commonly 87% or higher is an interesting match. You might also want to target a single day or week or month that you shouldn't have dupes within. Notice the .where function and then pass in a custom dataframe to the DQ context.

```
opt.dupe.on = true
opt.dupe.lowerBound = 99
opt.dupe.include = Array("SYMBOL", "EXCH")

val df1Day = jdbcDF.where("TRADE_DATE = '2018-01-10' ")
val cdq = OwlUtils.OwlContext(df1Day, opt)
cdq.owlCheck()
val dupesPreview = cdq.getDupesPreview()
val records = cdq.getRecordsDF
dupes.show

// rdd collect
dupes.rdd.collect.foreach(println)

// records linked together for remediation
cdq.getDupeRecords.show
```

Outliers

Gaining and understanding of your outliers is a commonly desired DQ function. DQ has several configurations to help find the most meaningful outliers in your dataset and over time. Below compares the current day to a baseline of days in the historical dataframe.

```
val outlierOpt = new OutlierOpt()
outlierOpt.combine = true
outlierOpt.dateColumn = "trade_date"
outlierOpt.lookback = 4
outlierOpt.key = Array("symbol")
outlierOpt.timeBin = OutlierOpt.TimeBin.DAY
outlierOpt.historyLimit = 10
dlMulti.add(outlierOpt)
opt.setOutliers(dlMulti)
val cdq = OwlUtils.OwlContext(df, opt)
.register (opt)
cdq.setHistoricalDF(dfHistory)
cdq.owlCheck()
val outliers = cdq.getOutliers()
outliers.show
```

Advanced

Programmatic DQ

Don't like leaving your notebook? Want to build data quality into your in-house data quality pipeline? Collibra DQ can do both!

Real World Examples

Rules

Let's assume we were provided a file named "atm_cust_file" and want to load it into a database table as well as scan it for all possible errors. We want to provide a couple levels of protection. 1) A business rule checking if *a customer joined before before the company was founded*. 2) Check if the file 100% matches to the DataFrame or db table we've created. 3) Check for all

possible outliers or anomalies in the dataset. Each one of these 3 issues had a different impact to the business and causes a different flow to trigger in our pipeline.

Add Rule

Let's create a simple rule and assign points to the overall scoring system for later delegation.

```
val rule = new Rule
rule.setRuleNm("customer_before_company")
rule.setRuleValue("customer_since_date < '1956-11-01'")
rule.setPerc(1.0)
rule.setPoints(1)
rule.setIsActive(1)
rule.setUserNm("User")
rule.setDataset("ATM_CUSTOMER3")
OwlUtils.addRule (rule)
```

Now let's chain together the remaining two items that were part of our original requirement. Note that DQ has six additional ML DQ features that we did not turn on in this case.

```
val df_source = (spark.read
.format ("csv").option("header", true).option("delimiter", ",")
.load("atmCustFile.csv")
)
val dataset = "atm_customer"
var date = "2018-01-16"

val optSource = new SourceOpt
optSource.on = true
optSource.dataset = dataset
optSource.validateValues = true
optSource.key = Array("name")
opt.setSource(optSource)
//Dataframe for the table created from the file

val connProps = Map (
"driver"    -> "org.postgresql.Driver",
"user"        -> "postgres",
"password"    -> "xxx",
"url"        -> "jdbc:postgresql://xxx:xxx/postgres",
"dbtable"    -> "public.atmCustomer" )
val df_target = spark.read.format("jdbc").options(con-
nProps).load
val cdq = OwlUtils.OwlContext(df_source, df_target, opt)
.register (opt)
cdq.owlCheck
// first register with catalog if not registered
cdq.register (props)
// Check if dataframe matches the source file 'atm_cust_file'
val breakCountDF = cdq.getSourceBreaksCount()
breakCountDf.show()
if (breakCountDf.count() > 1) {
  // create service now ticket and exit with fail based on not
matching to original file
}
val ruleBreaks = cdq1.getRuleBreakRows()
if (ruleBreaks.count() > 1) {
  if (ruleBreaks.where($"score" > 5).count > 1) {
    //create service now ticket and exit with fail based on
rules
  }
}
val outliers = cdq.getOutliers()
if (outliers.where($"confidence" < 10).count > 3) {
  // CDQ email Alert to business group for attention
  // where 3 outliers have a confidence below 10
}
```

Ingesting Intraday Files

Here we illustrate an example of how to work with files when using DQ programmatically. This can be implemented in both a Notebook setting and in your own codebase.

```
//////////////////////////////////////////////////////////////-
//////////
    //                  USE CASE - Ingesting Intraday
Files                  //

//////////////////////////////////////////////////////////////-
//////////

    // Part of your pipeline includes the ingestion of files
that have the date
    // and hour encoded in the file name. How do you process
those files using Collibra DQ?
    //
    // Format: <name>_<year>_<month>_<day>.csv
    //
    // Build up a data structure containg the files you want to
process (here we
    // just use a simple list, but you may want to be pulling
from a pubsub
    // queue, AWS bucket, etc...). Here we just use a simple
file list of 6
    // hours of trade position data.
    val position_files = List( new File(getClass.getResource
("/position_file_2019_11_03_08.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
09.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
08.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
09.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
10.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
11.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
12.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
13.csv").getPath),
      new File(getClass.getResource("/position_file_2019_11_03_
14.csv").getPath))

    // Create your spark session.
    val spark = SparkSession.builder
      .master("local")
      .appName("test")
      .getOrCreate()

    // Configure Collibra DQ
```

```
    val opt = new OwlOptions
    opt.dataset = "positions"
    opt.load.delimiter = ","
    opt.spark.master = "local[1]"
    opt.dupe.on = true
    opt.dupe.include = Array("ticker", "cid")
    opt.outlier.on = true
    opt.outlier.key = Array("cid")
    opt.outlier.timeBin = TimeBin.HOUR
    // Customize this to only process a subset of the data.
    opt.load.fileQuery = "select * from dataset"

    position_files.foreach { file: File =>
      // Tell Collibra DQ where to find the file.
      opt.load.filePath = file.getPath

      // Parse the filename to construct the run date (-rd) that
will be passed
      // to Collibra DQ.
      val name = file.getName.split('.').head
      val parts = name.split("_")
      val date = parts.slice(2, 5).mkString("-")
      val hour = parts.takeRight(1).head

      // Must be in format 'yyyy-MM-dd' or 'yyyy-MM-dd HH:mm'.
      val rd = s"${date} ${hour}"

      // Tell Collibra DQ to process data
      opt.runId = rd

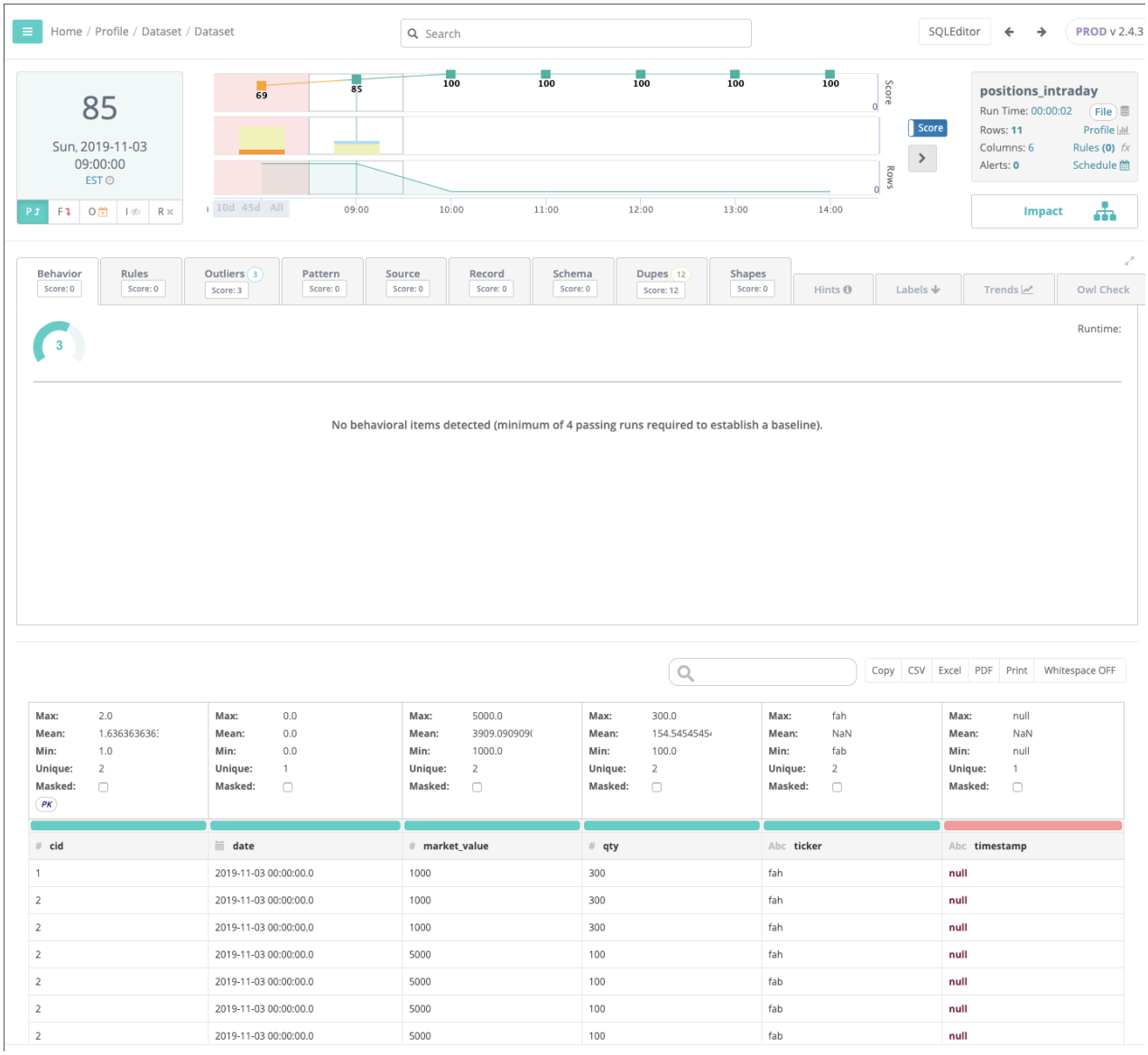
      // Create a DataFrame from the file.
      val df = OwlUtils.load(opt.load.filePath, opt.-
load.delimiter, spark)

      // Instantiate an OwlContext with the dataframe and our
custom configuration.
      val cdq = OwlUtils.OwlContext(df, spark, opt)

      // Make sure Collibra DQ has catalogued the dataset.
      cdq.register(opt)

      // Let Collibra DQ do the rest!
      cdq.owlCheck()

    }
```



All Pipeline Activities in One Line

For brevity and convenience, DQ allows a DF to be loaded in the constructor and in one line run all nine dimensions of data quality "cdq.owlcheck()". To adjust the DQ dimensions you simply set the properties in the props object.

```
val cdq = Util.OwlContext(df, atmCustFileDF, opts)
cdq.owlCheck()
```

Example of some common property settings

```
val props = new Props()
props.filePath = s"${filePath}/atm_customer_${rd.replace("-","_
")}.csv"
props.runId = rd
props.dateCol = "OWL_RUN_ID"
props.dataset = "ATM_CUSTOMER3"
props.del = ","
props.datasetSafety = false
props.calculateBoundaries = true
props.fileLookBack = true
props.timeBin = "DAY"

// outlier, missing records
props.dl = true
props.dlKey = "customer_id"
props.dlLb = 4

// pattern mining
props.freqPatternMiningByKey = true
props.fpgKey = "customer_id"
props.fpgLookback = 4
props.fpgDateCol = "OWL_RUN_ID"
props.fpgCols = "card_number,first_name,last_name,checking_sav-
ings"
props.fpgLowFreq = true

// validate Src
props.validateSrc = true
props.valSrcKey = "customer_id"

// fuzzy match
props.isDupe = true
props.dupeCutOff = 88
props.depth = 3
props.dupeExcludeCols = "customer_id,card_number,customer_since_
date,OWL_RUN_ID"
```

README.md

OwlSpark DQ Pipelines

Build your own rich data quality pipelines directly on dataframes using Owl's DQ suite

Owl Predictive Data Science + Spark Scale

Auto Clean

first	last	age	ssn	phone
Kirk	brian	11	222-33-4444	410-489-8095
kirk	kirk	22	111-33-4444	410-489-8095
bryan	Kirk	11	555-33-4444	443-885-8989
Dan	dan	33	666-33-4444	(212) 333-4444
	brian	11	222-33-4444	(212) 333-4444
kirk	kirk	21	111-33-4444	(212) 333-4444
brianne	kirk	18	555-33-4444	3015859966
dan	dan	31		
kirk	brian	66		
kirk	null	age-27		
brianne	Kirt	13		
dan	DAN	35		
kirk	brian	16		

first	last	age	ssn	phone
Kirk	brian	11	222-33-4444	4104898095
kirk	kirk	22	111-33-4444	4104898095
bryan	Kirk	11	555-33-4444	4438858989
Dan	dan	33	666-33-4444	2123334444
	brian	11	222-33-4444	2123334444
kirk	kirk	21	111-33-4444	2123334444
brianne	kirk	18	555-33-4444	3015859966
dan	dan	31	666-33-4444	3015859966
kirk	brian	66	222-33-4444	3015859966
kirk	null	27	111-33-4444	3015859966
brianne	Kirt	13	555-33-4444	3015859966
dan	DAN	35	666-33-4444	3015859966
kirk	brian	16	222-33-4444	4104898095

```
val df = Seq(
  ("Kirk",  "brian",  "11",  "222-33-4444",  "410-489-8095"),
  ("kirk",  "kirk",   "22",  "111-33-4444",  "410-489-8095"),
  ("bryan", "Kirk",   "11",  "555-33-4444",  "443-885-8989"),
  ("Dan",   "dan",    "33",  "666-33-4444",  "(212) 333-4444"),
  ("",      "brian",  "11",  "222-33-4444",  "(212) 333-4444"),
  ("kirk",  "kirk",   "21",  "111-33-4444",  "(212) 333-4444"),
  ("brianne","kirk",  "18",  "555-33-4444",  "3015859966"),
  ("dan",   "dan",    "31",  "666-33-4444",  "3015859966"),
  ("kirk",  "brian",  "66",  "222-33-4444",  "3015859966"),
  ("kirk",   null,    "age-27", "111-33-4444","3015859966"),
  ("brianne","Kirt",  "13",  "555-33-4444",  "3015859966"),
  ("dan",   "DAN",    "35",  "666-33-4444",  "3015859966"),
```

Collibra DQ Rules

In this section, you can learn how to work with Rules in Notebooks written in Scala.

Instantiation

Code	Description
val = new Rule()	Instantiating a Rule object
rule.setDataset (<DATASET>)	Adding the name of the dataset
rule.setRuleNm (<RULE_NAME>)	Adding the name of the given rule
rule.setRuleValue (<RULE_ EXPRESSION>)	Setting the simple RULE_EXPRESSION
rule.setRuleType (<RULE_TYPE>)	Setting the rule type
rule.setPerc (<RULE_ PERCENTAGE>)	Setting the percentage
rule.setPoints (<RULE_POINT>)	Setting how many points will be deducted from total for each percentage
rule.setIsActive (<RULE_IS_ ACTIVE>)	Making rule active/inactive Possible values: ACTIVE: 1 / true INACTive: 0 / false
rule.setUserNm (<RULE_OWNER_ USER_NAME>)	Adding the owner

Rule Types

Requirements

Required imports

```
import com.owl.core.Owl
import com.owl.core.util.OwlUtils
import com.owl.common.options.OwlOptions
import com.owl.common.{Props, Utils}

import com.owl.common.domain2.Rule

import org.apache.spark.sql.funtions._
```

SparkSession initialization

Databricks and other notebook execution frameworks are working with managed/shared Spark sessions, therefore, Collibra DQ recommend using this code snippet in your notebook to initialize the current Spark session properly.

```
// ----- Init Spark ----- //
def sparkInit(): Unit = {
  sparkSession = SparkSession.builder
    .master("local")
    .appName("test")
    .getOrCreate()
}
```

Warning Do not call **spark.stop** at any of your notebooks. Otherwise, the execution engine will exit immediately.

Important Make sure OwlContext is already created before using any method from OwlUtils.

Global rules

Distributed with the application by default and you can use them from UI/CLI/Notebooks.

Rule types

- Invalid_Email_Check
- Invalid_Phone_Num_Check
- Invalid_Zip_Code_Check
- Invalid_SSN_Check
- Invalid_IP_Address_Check
- Invalid_Gender_Check
- Invalid_EIN_Check
- Invalid_State_Check
- Invalid_Credit_Card_Check
- Valid_Email_Check
- Valid_Phone_Num_Check
- Valid_Zip_Code_Check
- Valid_SSN_Check
- Valid_IP_Address_Check
- Valid_Gender_Check
- Valid_EIN_Check
- Valid_State_Check
- Valid_Credit_Card_Check
- Percent_Move_5 Percent_Move_10
- Percent_Move_20 Percent_Move_50
- Not_In_Previous_run
- Not_In_Current_run
- Having_Count_Greater_Than_One

SQL based rules

Simple rule

You can use simple rules to filter a condition on a single column in a single table.

Example #1

This example shows you how to create a simple SQL rule named **simple_sql_rule**.

Code	Description
rule.setRuleNm("**simple_sql_rule**")	Add the name of the given rule.
rule.setRuleValue("**startDate < '2011-11-01'"**)	Set the simple SQL expression. **JOIN** is not allowed between tables.
rule.setRuleType("**SQLG**")	Set the rule type.

Code

example_simple_sql_rule.scala

```
import com.owl.core.Owl
import com.owl.core.util.OwlUtils
import com.owl.common.bll.{RuleBll, RuleTemplateBll}
import com.owl.common.domain2.Rule
import com.owl.common.options.{LoadOpt, OwlOptions}

import org.junit.{Assert, Test}

import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._

val loadOptions = new LoadOpt {
  pghost = "localhost:5432/postgres"
  pguser = "username"
  pgpassword = "password"
}

//----- Init Spark ----- //
def sparkInit(): SparkSession = {
  val sparkSession = SparkSession.builder
    .master("local")
    .appName("test")
    .getOrCreate()

  sparkSession
}

def simpleRuleNotebook(): Unit = {

  // Arrange
  val spark = sparkInit()
  import spark.implicits._

  val headers = "firstName,lastName,startDate"
  val source = Seq(
    ("Thomas", "Martinez", "2010-11-01"),
    ("Harry", "Williams", "2012-05-01"),
    ("Ethan", "Davis", "2009-08-01")
  )
  val arr = headers.split(",")
  val df = source.toDF(arr: _*)

  val opt = new OwlOptions {
    runId = "2019-09-20"
    dataset = "simple_sql_rule_ds"
     load = loadOptions
  }
```

```
  val rule = new Rule {
    setDataset(opt.dataset)
    setRuleNm("simple_sql_rule")
    setRuleValue("startDate < '2011-11-01'")
    setRuleType("SQLG")
    setPerc(1.0)
    setPoints(1)
    setIsActive(1)
    setUserNm("admin")
  }

  val cdq = OwlUtils.OwlContext(df, opt)
    .register(opt)

  OwlUtils.addRule(rule)

  // Act
  cdq.owlCheck()

  // Assert
  import scala.collection.JavaConversions
  val hootRule = JavaConversions.asScalaBuffer
(owl.hoot.rules).find(x => rule.getRuleNm.equals
(x.getRuleNm)).orNull

}

// Execute notebook
simpleRuleNotebook()
```

You can do multiple assertions on the result of the DQ Check process.
The **owl.hoot** parameter will provide access to the execution results, in this case for the rule.

Example #2

This example shows you how to create a simple SQL rule with templates named **simple_sql_rule_with_template**.

Steps

1. Create the rule template, where the template column name should be marked with **$colNm** string.\

```
val ruleTemplate = RuleTemplateBll.createRuleTemplate(
    "not_null_or_empty",
    "Column cannot contain null or empty values",
    " $colNm is null or $colNm = \'\' or $colNm  = \'null\'
"
)
```

2. Create the Rule instance, where value of **RuleValue** will be used to replace **$colNm** in the template expression.\

```
val rule = RuleBll.createRule(opt.dataset)
rule.setRuleNm("is_city_not_null_or_empty")
rule.setRuleValue("city")
rule.setRuleType("CUSTOM") // legacy type required to look
into rule repo
rule.setRuleRepo("not_null_or_empty") // custom rule name
to pull rule value from rule repo
rule.setPerc(1.0)
rule.setPoints(1)
rule.setIsActive(1)
rule.setUserNm("admin")
```

Code

```
import com.owl.core.Owl
import com.owl.core.util.OwlUtils
import com.owl.common.bll.{RuleBll, RuleTemplateBll}
import com.owl.common.domain2.Rule
import com.owl.common.options.{LoadOpt, OwlOptions}

import org.junit.{Assert, Test}

import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._

val loadOptions = new LoadOpt {
  pghost = "localhost:5432/postgres"
  pguser = "username"
  pgpassword = "password"
}

//----- Init Spark ----- //
def sparkInit(): SparkSession = {
  val sparkSession = SparkSession.builder
    .master("local")
    .appName("test")
    .getOrCreate()

  sparkSession
}

 def simpleRuleWithTemplate(): Unit = {

  // Arrange
  val spark = sparkInit()
  import spark.implicits._

  val headers = "firstName,lastName,city"
  val source = Seq(
    ("Thomas", "Martinez", ""),
    ("Harry", "Williams", null),
    ("Ethan", "Davis", "Los Angeles")
  )
  val arr = headers.split(",")
  val df = source.toDF(arr: _*)

  val opt = new OwlOptions {
    runId = "2019-09-20"
    dataset = "simple_sql_rule_with_template_ds"
    onReadOnly = false
    load = loadOptions
  }
```

```
  // create a generic rule
  val ruleRepoName = "not_null_or_empty"
  val ruleRepo = OwlUtils.createRuleTemplate(ruleRepoName,
"Column cannot contain null or empty values", "$colNm is null or
$colNm = \'\' or $colNm  = \'null\'")
  Util.RuleRepoDaoFactory().delete(ruleRepo)
  val cdq = OwlUtils.OwlContext(df, opt)

  cdq.addRuleTemplate(ruleRepo)

  val rule = OwlUtils.createRule(opt.dataset)
  rule.setRuleNm("is_city_not_null_or_empty")
  rule.setRuleValue("city")
  rule.setRuleType("CUSTOM") // legacy type required to look
into rule repo
  rule.setRuleRepo("not_null_or_empty") // custom rule name to
pull rule value from rule repo
  rule.setPerc(1.0)
  rule.setPoints(1)
  rule.setIsActive(1)
  rule.setUserNm("admin")

  val cdq = OwlUtils.OwlContext(df, opt)
    .register(opt)

  cdq.addRule(rule)
  // Act
  owl.owlCheck()

  // Assert
  import scala.collection.JavaConversions
  val hootRule = JavaConversions.asScalaBuffer
(owl.hoot.rules).find(x => rule.getRuleNm.equals
(x.getRuleNm)).orNull
   println(hootRule.getScore)
}

// Execute notebook
simpleRuleWithTemplate()
```

Freeform SQL

Freeform SQL allows you to apply a complex condition across multiple tables or columns. You can also use it when you need more flexibility or customization of your rules.

> **Note** Use the built-in Spark functions for non-Pushdown connections. Native SQL will not work, unless you switch to Native SQL rules. See (https://spark.apache.org/docs/2.3.0/api/sql/) for Simple and Freeform SQL rules.

> **Important** Column name references on the Spark dataframe do not necessarily match the column names.

Individual statement

Syntax

```
SELECT * FROM @<dataset_name> <table_alias>
WHERE <filter_expression>
GROUP BY <group_by_expression>
HAVING <having_expression>
```

The base of the statement is given with **@<dataset_name>** style. In general, the <dataset_name> is the same, where the rule is attached to, but you can generally use any valid dataset name in the expression.

Examples

Simple rule expression

```
opt.dataset = "example_ds"

val rule = OwlUtils.createRule(opt.dataset)
rule.setRuleNm("is_city_not_null_or_empty")
rule.setRuleValue("select * from @example_ds t where t.amount >
'5000'")
rule.setRuleType("SQLF")
rule.setPerc(1.0)
rule.setPoints(1)
rule.setIsActive(1)
rule.setUserNm("admin")
```

Complex rule expression

```
opt.dataset = "unique_rule_ds"

val rule = RuleBll.createRule(opt.dataset)
rule.setRuleNm("unique_rule")
rule.setRuleValue("select * from ( select count(*) as cnt, cus-
tomer_id from @unique_rule_ds group by customer_id ) having cnt
> 1")
rule.setRuleType("SQLF")
rule.setPerc(1.0)
rule.setPoints(1)
rule.setIsActive(1)
rule.setUserNm("admin")
```

RegExp expression

```
opt.dataset = "regexp_rule_ds"

val rule = RuleBll.createRule(opt.dataset)
rule.setRuleNm("LIKE_rule")
rule.setRuleValue("select * from @regexp_rule_ds.SYMBOL rlike
'^ABB+'  ")
rule.setRuleType("SQLG")
rule.setPerc(0.02)
rule.setPoints(1)
rule.setIsActive(1)
rule.setUserNm("admin")
```

Join statements

Available join types between multiple datasets

- WHERE tableA.id = tableB.id style
- INNER JOIN
- LEFT <OUTER> JOIN
- RIGHT <OUTER> JOIN

Joining other datasets

A @t1 rule is an out-of-the-box rule type that creates a dataset using the previous run's date as the run date to read data from a multi-day run period so that data from the current and previous days can be evaluated.

To get the historical state of the same dataset:

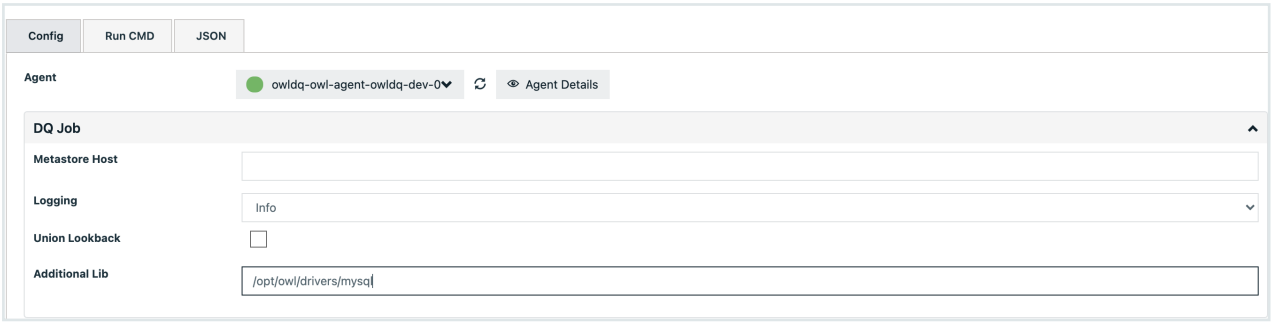
Syntax: @t<n>, where the parameter *n* means, the historical run (number) of the base dataset (marked with @<dataset_name>) to instruct the temporary job created at runtime to target. The following table demonstrates what each of the following @t*n* value examples mean:

@t*n* value	Description
@t1	Loads the data from the previous scan.
@t2	Loads the data from the second to last scan.
@t3	Loads the data from the third to last scan.
@t4	Loads the data from the fourth to last scan

To get a different dataset:
Syntax: @<other_dataset_name>

> **Note** When you use a rule with @t1 syntax, a temporary job in the Jobs queue appears. This temporary job displays a LOAD status to allow the run that the @t1 is targeting to load and use it for the rule.

Cross-Connection Libraries



> **Note** When applying cross-connection rules please use the -addlib to submit the job with the appropriate jar files. In this example, a secondary set of jars is added through the Explorer. These files are located in the /opt/owl/drivers/mysql directory. The path should not contain double quotes or single quotes. It should point to a directory without spaces in the path.

WHERE style

Lookback dataset

```
SELECT * FROM @<dataset_name> <table_alias>, @t1 [<history_
table_alias>]
WHERE <join_expression> AND <filter_expression>
```

Example

```
opt.dataset = "example_ds"

val rule = OwlUtils.createRule(opt.dataset)
rule.setRuleValue("select * from @example_ds t, @t1  where t.cus-
tomer_id = t1.customer_id  and t.card_number <> t1.card_number
")
rule.setRuleType("SQLF")
```

Different dataset

```
SELECT * FROM @<dataset_name> <table_alias>, @<other_dataset_
name> [<other_alias>]
WHERE <join_expression> AND <filter_expression>
```

Example

```
opt.dataset = "example_ds"
opt2.dataset = "other_ds"

val rule = RuleBll.createRule(opt.dataset)
rule.setRuleValue("select * from @example_ds t, @other_ds ds2
where t.customer_id = ds2.customer_id  and t.card_number <> ds2.-
card_number ")
rule.setRuleType("SQLF")
```

LEFT JOIN

Example

```
opt.dataset = "example_ds"

val rule = OwlUtils.createRule(opt.dataset)
rule.setRuleNm("not_back2back_days")
rule.setRuleValue(" select * from @example_ds A LEFT OUTER JOIN
@t1 B ON A.customer_id = B.customer_id where A.customer_id is
not null and B.customer_id is null  ")
rule.setRuleType("SQLF")
rule.setPerc(1.0)
rule.setPoints(1)
rule.setIsActive(1)
rule.setUserNm("admin")
```

Known Limitations

Freeform rules with fully qualified column names are currently unsupported when they use the following syntax:

```
select <column name> FROM @<dataset name> WHERE @<dataset
name>.<column name> condition
```

A possible workaround to this limitation is to use aliasing instead.

Data type rules

Simple checks for individual columns.

> **Note** See also: Data type.

Rule types

Empty check

Rule type: EMPTYCHECK
Description: Checks whether or not the target column has empty values.

Null check

Rule type: NULLCHECK
Description: Checks whether or not the target column has *NULL* values.

Date check

Rule type: DATECHECK
Description: Checks whether or not the target column has **only** DATE values.

Integer check

Rule type: INTCHECK
Description: Checks whether or not the target column has **only** INTEGER values.

Double check

Rule type: DOUBLECHECK
Description: Checks whether the target column has **only** DOUBLE values or not.

String check

Rule type: STRINGCHECK
Description: Checks whether the target column has **only** STRING values or not.

Mixed datatype check

Rule type: DATATYPECHECK
Description: Checks whether the target column has values with different data types, such as DATE, INTEGER, DOUBLE, and STRING.

Syntax

- **<rule_type>** - Fixed key to the rule type
- **<column_name>** - Column to apply the rule
- **<rule_name>** - Custom name of the rule

```
opt.dataset = "example_ds"

val rule = OwlUtils.createRule(opt.dataset)
rule.setRuleNm("<rule_name>")
rule.setRuleValue("<column_name>")
rule.setRuleType("<rule_type>")
rule.setPerc(1.0)
rule.setPoints(1)
rule.setIsActive(1)
rule.setUserNm("admin")
```

Outliers

This real life use-case is when you have a large file or data frame with many days of data but you want the run profile to be the current day so that it trends properly overtime. Another nuance to this use-case is that the customer_id is a unique field to the user and it should not show up in the analytics i.e. an outlier. But the customer_id should be available when the user wants to query the rest api end points. The customer_id is then used to link back the users original dataset. A bloomberg_Id (BB_ID) is a common example.

CSV File

```
fname,app_date,age,customer_id
Kirk,2018-02-24,18,31
Kirk,2018-02-23,11,4
Kirk,2018-02-22,10,3
Kirk,2018-02-21,12,2
Kirk,2018-02-20,10,1
```

Notebook Code (Spark Scala)

```
val filePath = getClass.getResource("/notebooktest.csv").getPath

val spark = SparkSession.builder
  .master("local")
  .appName("test")
  .getOrCreate()
val opt = new OwlOptions()
opt.dataset = "dataset_outlier"
opt.runId = "2018-02-24"
opt.outlier.on = true

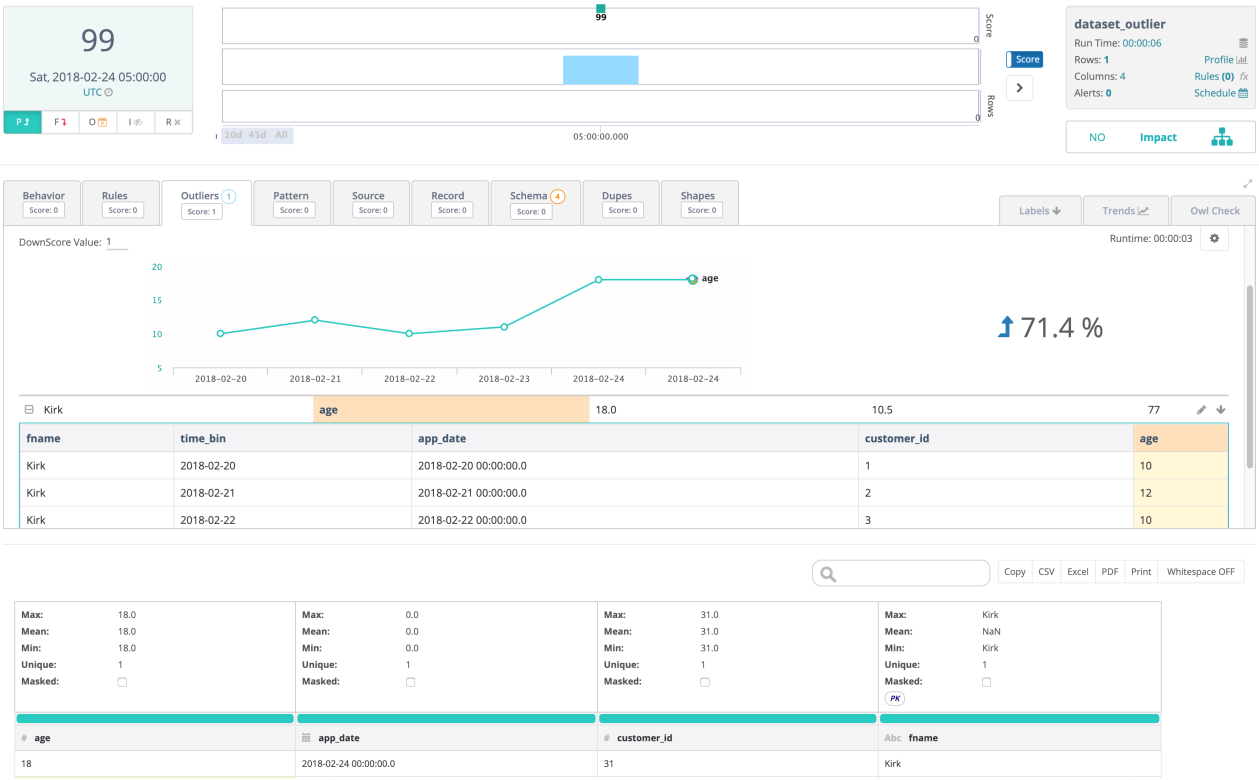
 val dlMulti: util.List[OutlierOpt] = new util.ArrayList[Out-
lierOpt]
 val outlierOpt = new OutlierOpt()
 outlierOpt.combine = true
 outlierOpt.dateColumn = "app_date"
 outlierOpt.lookback = 5
 outlierOpt.key = Array("fname")
 outlierOpt.exclude = Array("customer_id")
 outlierOpt.historyLimit = 10
 outlierOpt.timeBin = OutlierOpt.TimeBin.DAY
 dlMulti.add(outlierOpt)
 opt.setOutliers(dlMulti)

val dfHist = OwlUtils.load(filePath = filePath, delim = ",",
sparkSession = spark)
val dfCurrent = dfHist.where(s"app_date = '${opt.runId}' ")

val cdq = OwlUtils.OwlContextWithHistory(dfCurrent=dfCurrent,
dfHist=dfHist, opt=opt)
cdq.register(opt)
cdq.owlCheck()
```

Collibra DQ Web UI

Score drops from 100 to 99 based on the single outlier in the file. Row count is 1 because there is only 1 row in the current data frame. The historical data frame was provided for context and you can see those rows in the outlier drill-in. The customer_id is available in the data preview and can be used as an API hook to link back to the original dataset.



After you run an DQcheck using owl.owlcheck you might want to check individual scores to see what type of issues were in the data. Collibra DQ sends back the records with issues in the format of a DataFrame using the notebook cmds or JSON from the REST api.

```
val hoot = cdq.hoot

println(s"SHAPE   ${hoot.shapeScore} ")
println(s"DUPE    ${hoot.dupeScore} ")
println(s"OUTLIER ${hoot.outlierScore} ")
println(s"PATTERN ${hoot.patternScore} ")
println(s"RECORD  ${hoot.recordScore} ")
println(s"SCHEMA  ${hoot.schemaScore} ")
println(s"BEHAVIOR${hoot.behaviorScore} ")
println(s"SOURCE  ${hoot.sourceScore} ")
println(s"RULES   ${hoot.ruleScore} ")

if (hoot.shapeScore > 0) {
  cdq.getShapeRecords.show
}
if (hoot.dupeScore > 0) {
  cdq.getDupeRecords.show
}
```

```
+-------+---------+--------------------+--------+-----------+---
----+-------+
|row_cnt|obs_score|             row_key|obs_type|customer_
id|  fname|owl_id|
+-------+---------+--------------------+--------+-----------+---
----+-------+
|     21|       46|afa89984ce472a409...|    DUPE|         32|   -
Kirk|     1|
|     22|       46|afa89984ce472a409...|    DUPE|         31|Kir-
k's.|     2|
|     23|       60|41ea2d828b1a5fbf2...|    DUPE|         30|   -
 Dan|     3|
|     24|       60|41ea2d828b1a5fbf2...|    DUPE|         27|   -
 Dan|     6|
```

```
+--------------+--------------------+--------+---------+------
--------+--------+-------+-------+---+--------------------+-----
------+-------+------+---------+
|        dataset|              run_id|col_name|col_format|col_
format_cnt|owl_rank|row_cnt|row_key|age|            app_date|-
customer_id|  fname|owl_id|time_bin|
+--------------+--------------------+--------+---------+------
--------+--------+-------+-------+---+--------------------+-----
------+-------+------+---------+
|dataset_outlier|2018-02-24
00:00:...|   fname|   xxxx'x.|             1|       1|      2|xx-
xx'x.| 18|2018-02-24
00:00:...|         31|Kirk's.|     2|    null|
+--------------+--------------------+--------+---------+------
--------+--------+-------+-------+---+--------------------+-----
------+-------+------+---------+
```

GET http://$host/v2/getoutlier?dataset=dataset_outlier&runId=2018-02-24

GetOutlier

Parameters

Path

dataset	string	name of dataset
dataset	string	yyyy-MM-dd format can include time and timezone

Responses

- 200


```
{
    confidence: 77
    dataset: "dataset_outlier"
    keyArr: null
    lb: 0
    outColumn: "age"
    outKey: "Kirk"
    outMedian: "10.5"
    outValue: "18.0"
    runId: "2018-02-24T05:00:00.000+0000"
    ub: 0
    }
```


GET http://$host/v2/getdatashapes?dataset=dataset_outlier&runId=2018-02-24

GetShape

Parameters

Path

dataset	string	name of dataset
runId	string	yyyy-MM-dd format can include time and timezone

Responses

- 200

Options API

Field mappings

Field name	CLI prop	Description
dataset	*ds*	dataset name, example: userTable or users or user_file

Field name	CLI prop	Description
rundId	*rd*	run date, must be in format **yyyy-MM-dd** or for incremental use Hours or Minutes **yyyy-MM-dd HH:mm**
rundIdEnd	*rdEnd*	end date for query ranges **t_date >= $ and t_date < $**, must be in format **yyyy-MM-dd** or for incremental use Hours or Minutes **yyyy-MM-dd HH:mm**
passFail	*passfaillimit*	Limit for passing or failing runs
jobId		
onReadOnly	*readonly*	Do not persist results to the DQ metastore - useful during testing.

Load Options

Field name	CLI prop	Description
fullFile	*fullfile*	use entire file for lookbacks instead of just filequery
fileQuery		
header		
headerSrc		
hasHeader		
isParallel		
isJson		
isMixedJson		
isMapsJson		
flatten		
isMultiLine		
sparkprinc		

Field name	CLI prop	Description
sparkkeytab		
jdbcprinc		
jdbckeytab		
srcpwdmgr		
pwdmgr		
pguser	pguser	
pgpassword	pgpassword	
pghost	host	
pgport	port	
executorcores		
isParquet		
isAvro		
avroSchema		
isXml		
xmlRowTag		
isOrc		
dateFmt		
timeFmt		
datasetSafety		
filePath		
delimiter		
fileLookBack		

Field name	CLI prop	Description
dbLookBack		
connectionURL		
userName		
password		
sqlQuery		
connectionProps		
zkHost	~~~~	***Deprecated***
zkPort	~~~~	***Deprecated***
zkPath	~~~~	***Deprecated***

Outlier Options

Field name	CLI prop	Description
on	dl	Deep learning.
lookback	dllb	A deep learning lookback. For example, a value 5 for a lookback of 5 days. **Note** Lookback periods directly correlate to `-dllb`.
key	dlkey	A comma-delimited deep learning key. comma delim key ex: symbol,date
dateField		
bin		
includes	dlinc	A deep learning col limit. For example, open,close,high,volume.
excludes	dlexc	A deep learning col exclusion. For example, open,close,high,volume.

Field name	CLI prop	Description
categorical		
by		
limit		
historyLimit		
minhistory	dlminhist	An automatically generated flag based on the outlier lookback setting –`dllb` that defines the minimum number of days before DQ flags data as potential outliers. –`dlminhist`ensures that the number of days in the algorithm is relative to the total scope of the lookback period. It is not recommended that you override this flag from the command line. > **Note** Lookback periods do not directly correlate to –`dlminhist`.
score		

FPG Options

Field name	CLI prop	Description
on	fpgon	pattern mining
lookback	fpglb	lookback interval for pattern mining. Ex: **-fpglb 5**
key	fpgkey	natural key for pattern mining activity
dateField	fpgdc	date column for pattern mining. Ex: **-fpgdc date_col**
lowFreq		*Deprecated*
includes	fpginc	pattern mining is expensive use this input to limit the observed cols

Field name	CLI prop	Description
excludes	fpgexc	pattern mining is expensive use this input to limit the observed cols
timeBin	fpgtbin	time bin for pattern mining. Ex: **-fpgtbin DAY**
score	fpgscore	score for pattern mining records
minSupport	fpgsupport	
confidence	fpgconfidence	

Dupe Options

Field name	CLI prop	Description
on	dupe	duplicate record detection
includes	dupeinc	duplicate record detection, column inclusion list
excludes	dupeexc	duplicate record detection, column exclusion list
depth		
lowerBound	dupelb	duplicate lower bounds on percent match default [85]
upperBound		
blocksize		
useCache		
checkHeader		
exactMatch		
ignoreCase	dupenocase	duplicate record detection, column exclusion list
score	dupescore	
limit	dupelimit	Limit for dupe rows stored

Profile Options

Field name	CLI prop	Description
on		
includes		
excludes		
dataShapeOn		
statsOn		
correlationOn		
histogramOn		
cardinalityOn		
dataShapeColsInc		
dataShapeColsExc		

Source Options

Field name	CLI prop	Description
on		
includes		
excludes		
key		
fileQuery		
map		
score		
datasetSrc		

Field name	CLI prop	Description
driverSrc		
userNameSrc		
passwordSrc		
connectionURLSrc		
sqlQuerySrc		
connectionPropsSrc		

Rule Options

Field name	CLI prop	Description
on		
rulesOnly		
semantic		

ColMatch Options

Field name	CLI prop	Description
colMatchParallelProcesses		
colMatchDurationMins		
colMatchBatchSize		
connectionList		

Spark Options

Field name	CLI prop	Description
numExecutors		
executorMemory		
driverMemory		
executorCores		
master		
jars		
libs		
driver		

Misc Options

Field name	CLI prop	Description
obslimit		
nullValue		

Classes

Chapter 12

Load

```
package com.owl.common.options;

import org.apache.commons.lang3.StringUtils;

import java.util.Properties;

/**
 * Collibra DQ Options related to data loading
 */
public class LoadOpt {
    // Options order: "unsorted",
    // "dataset scope columns", "dataset scope rows", "look
back",
    // "common options for both data sources", "file as data
source", "db as data source"

    public static final String SINGLE_QUOTE = "'";
    public static final String DOUBLE_QUOTE = "\"";
    public static final String BACK_TICK = "`";

    /**
     * If true, don't save any metadata
     * TODO confirm if this is correct
     */
    public Boolean readonly = false;

    /**
     * The Password manager.
     */
    public String passwordManager = null;

    /**
     * Catalog alias (Catalog name)
     */
    public String alias = StringUtils.EMPTY;

    // --- Dataset Scope Column specifications ------- //
    // Properties that select columns for Dataset activities or
modifies (data type or new columns)
    // prior and/or during loading into Spark DF
    /**
     * Dataset scope query. (IMPORTANT)
     * The query should contain all the columns necessary to run
the activities.
     * TODO: figure out if this gets used when using files
     */
    public String query = StringUtils.EMPTY;
```

```
    /**
     * Concatenated column names (sep = ",") for columns that
are keys
     * TODO: confirm
     */
    public String key = StringUtils.EMPTY;

    /**
     * SELECT expression to transform expressions with assign-
ment by "=" and delimited by "|".
     * e.g. colname=cast(colname as string)|colname2=colname2
(cast as date)
     */
    public String expression = StringUtils.EMPTY;

    /**
     * Add "OWL_RUN_ID" UNIX timestamp (s) column to Spark DF
usng the OwlOptions.runId.
     * Does not obey timeStampDivisor (timestamp in seconds
because Spark)
     */
    public Boolean addDateColumn = false;

    /**
     * Fill null values in Spark DF with 0 (numeric columns
only)
     */
    public Boolean zeroFillNull = false;

    /**
     * A string that indicates a null value; any value matching
this string will be set as nulls in the Spark DF
     * Default: "" -> NULL
     * Example: 'null' -> NULL
     * --
     * Note: to emptyStirngFillNull (replace String column null
-> "", use expression
     */
    public String replaceNulls = StringUtils.EMPTY;

    /**
     * All data types forced to strings for type safe pro-
cessing.
     * Not implemented in activity (yet)
     */
    public Boolean stringMode = false;

    // --- Dataset Scope Row specifications ------- //
```

```
    // Properties that filter rows for Dataset activities
    // prior and/or during loading into Spark DF
    /**
     * Convert row into string and only use rows containing this
value.
     * Strict matching only.
     */
    public String filter = StringUtils.EMPTY;

    /**
     * Convert row into string and only use rows containing this
value.
     * Strict matching only.
     */
    public String filterNot = StringUtils.EMPTY;

    // --- Look back ------- //
    // For Look back feature
    /**
     * Build up history of DQ Checks. Does not include current
DQ Check.
     * TODO: Document the relationship with unionLookBack
     */
    public Integer backRun = null;

    /**
     * Whether to load data for looking back in history.
     * How much historical data to load is based on Out-
lierOpt.lookback and PatternOpt.lookback.
     */
    public Boolean unionLookBack = false;

    // --- Shared Data Loading Options ------- //
    // Properties that affect data loading & pre-processing for
both files and db as source
    /**
     * Whether to use cached data for activities
     */
    public Boolean cache = true;

    /**
     * The year, month, and day format of date columns in the
dataset for loading the data only.
     * Default = "yyyy-MM-dd"
     */
    public String dateFormat = "yyyy-MM-dd";

    /**
```

```
     * The hour, minute, second, and milisecond format of date
columns in the dataset for loading the data only/
     * Default = "HH:mm:ss.SSS"
     * Not used. Questionably why separate timeFormat variable
exists when dateFromat can represent hms as well.
     */
    public String timeFormat = "HH:mm:ss.SSS";

    /**
     * Whether to convert date columns (specified by activity
opts) in dataset
     * into timestamp in ms (to make it seconds, set Props.-
timeStampDivisor = "s")
     * TODO: Needs LoadOpt.timeStampDivisor and fix Utils.scala
date2Timestamp
     */
    public Boolean timestamp = false;

    /* TODO add timeStampDivisor here and map between owl props?
     public String timeStampDivisor = "ms"
     */

    // --- Using file as data source ------- //
    // Properties that control where & how static file is read
    /**
     * Full path to the file.
     * If hdfs, then "hdfs://...".
     * If s3, then "s3://...", "s3a://...", or "s3n://...".
     * If parquet, then "...parquet" or "...PARQUET"
     */
    public String filePath = StringUtils.EMPTY;

    /**
     * SQL query used on file.
     * owl_id is added if not included in select clause.
     * If empty, then defaults to full file query.
     * (Does not update LoadOpts.fullFile to true).
     */
    public String fileQuery = StringUtils.EMPTY;

    /**
     * Whether to use full file (i.e. use all columns) on data
load
     */
    public Boolean fullFile = false;

    /**
     * File column names, comma separated
```

```
     */
    public String fileHeader = null;

    /* TODO checkHeader needs to be moved here from DupeOpt
     public Boolean checkHeader = true;*/

    /**
     * Whether to have Spark infer the schema of data source
     * If props.profile2 == true, this is overwritten to false!
     * If xml file, this is ignored and schema is always
inferred by Spark on xml data load.
     * If avro file, this value is respected (but may get over-
written by props.profile2)
     * (see activity2.Load.file)
     */
    public Boolean inferSchema = true;

    /**
     * Sample without replacement from file. Valid value is a
fraction [0, 1.0].
     * Only affects when filetype is xml or unspecified (and
therefore assumed to be delimited table)
     */
    public Double sample = 1.0;

    /**
     * Filetype (avro, json, orc, parquet, xml). Unspecified
file
     */
    public FileType fileType = null;

    /**
     * Delimiter for file. If number of characters after repla-
cing "\" with "" is 2 or more character
     * (e.g. compound delimiters like \t\t), then defaults to
"\t" and attempts to read file as tsv
     * See Activity2.load.file for details
     */
    public String delimiter = ",";

    /**
     * File character encoding
     */
    public String fileCharSet = "UTF-8";

    /**
     * The Avro schema for relevant avro file. Ignored if empty
string
```

```
     */
    public String avroSchema = StringUtils.EMPTY;

    /**
     * The Xml row tag for xml file. Ignored if empty string.
     */
    public String xmlRowTag = StringUtils.EMPTY;

    /**
     * Whether to flatten arrays in nested schema
     * TODO explain better. Does this only affect JSON file?
     */
    public Boolean flatten = false;

    /**
     * Whether data contains maps in json that requires extra
handling"
     * TODO explain better. Does this only affect JSON file?
     */
    public Boolean handleMaps = false;

    /**
     * Whether to handle mixed json.
     * TODO explain better. Does this only affect JSON file?
     */
    public Boolean handleMixedJson = false;

    /**
     * Spark.read option multiline, for JSON file only
     */
    public Boolean multiLine = false;

    // --- Using database as data source ------ //
    /**
     * Path to DB Driver. (e.g. /opt/owl/driver/postgres)
     */
    public String lib = StringUtils.EMPTY;

    /**
     * DB Driver name (Java namespace, e.g. org.-
postgresql.Driver).
     * Leave as null (default) and LoadOpts.connectionURL will
resolve the driver name.
     */
    public String driverName = null;

    /**
```

```
     * Connections name in metastore DB (pub-
lic.connections.aliasname).
     * Does not refer to the "name" of the database. Refers to
"aliasname" that the user set when
     * uploading connection config to Collibra DQ.
     */
    public String connectionName = StringUtils.EMPTY;

    /**
     * The Connection url, prefixed by jdbc.
     * e.g. "jdbc:postgresql://localhost:5432"
     */
    public String connectionUrl = StringUtils.EMPTY;

    /**
     * DB username
     */
    public String userName = StringUtils.EMPTY;

    /**
     * DB password
     */
    public String password = StringUtils.EMPTY;

    /**
     * JDBC Connection properties (e.g. fetchsize)
     */
    public Properties connectionProperties = null;

    /**
     * Whether data source is Hive Native (not using JDBC)
     * TODO: Why is the default null as opposed to false?
     */
    public Boolean hiveNative = null;

    /**
     * Whether data source is Hive Hadoop Web Cluster (not using
JDBC)
     */
    public Boolean hiveNativeHWC = false;

    // --- Parallel JDBC ------- //
    /**
     * When running parallel JDBC, use LoadOpts.query and Owl0p-
tions.dataset as base table
     */
    public Boolean useSql = true;
```

```
    /**
     * When running parallel JDBC, specify column name
     * ?? Activity2.Load and web has hard-coded magic string
"OWLAUTOJDBC"
     */
    public String columnName = null;

    /**
     * When running parallel JDBC, the upper bound for partition
column.
     * (e.g. "1000000")
     */
    public String lowerBound = null;

    /**
     * When running parallel JDBC, the upper bound for partition
column.
     * (e.g. "5000000")
     */
    public String upperBound = null;

    /**
     * When running parallel JDBC, the number of partitions
used.
     * If 0, then numPartitions used is based on the number of
available Spark Executor (1/2 ~ 2/3)
     * If > 20, then overwritten to 20 (no more than 20 con-
current connections to a database on a single dataset)
     */
    public Integer numPartitions = 0;

    // --- SQL Query properties ---------- //
    // TODO: does this effect DB as source or file as source as
well?
    /**
     * Whether the escape character would be back tick (`).
     * Ignored if escapeCharacter is non-empty (if using DQ
Check from Options).
     * Marked as true if props.escapeCharacter is a tick
     * (to preserve bijection between props and opts, and vice
versa).
     */
    public Boolean escapeWithBackTick = false;
    /**
     * Whether the escape character would be single quote (').
     * Ignored if escapeCharacter is non-empty (if using DQ
Check from Options).
     * Marked as true if props.escapeCharacter is a tick
```

```
     * (to preserve bijection between props and opts, and vice
versa).
     */
    public Boolean escapeWithSingleQuote = false;
    /**
     * Whether the escape character would be double quote (").
     * Ignored if escapeCharacter is non-empty(if using DQ Check
from Options).
     * Marked as true if props.escapeCharacter is a tick
     * (to preserve bijection between props and opts, and vice
versa).
     */
    public Boolean escapeWithDoubleQuote = false;

    /**
     * Specify custom escape character. This takes precedence
over all other escapeWithXYZ options.
     * i.e. if non-empty, then other escapeWithXYZ options are
ignored.
     * If empty (default), no escaping attempt is made (and SQL
query may fail if it contains reserved word)
     *
     * @deprecated Access level of this field will be changed to
private. Please use {@link #setEscapeCharacter(String)} instead.
     */
    @Deprecated
    public String escapeCharacter = StringUtils.EMPTY;


    /**
     * The enum File type.
     */
    public enum FileType {
        /**
         * Avro file type.
         */
        avro,
        /**
         * Json file type.
         */
        json,
        /**
         * Orc file type.
         */
        orc,
        /**
         * Parquet file type.
         */
        parquet,
```

```
        /**
         * Xml file type.
         */
        xml
    }
}
```

Profile

```
public class ProfileOpt {

    public Boolean on = true;           //Whether to run profile
    public Boolean only = false;        //Whether to run only pro-
file
    public String[] include;            //Which columns to include
    public String[] exclude;            //Which columns to exclude
    public Boolean shape = true;        //Disable shape detection
    public Boolean correlation = null;//On/Off Pearsons Cor-
relation, null=auto
    public Boolean histogram = null;  //On/Off Histograming,
null=auto
    public Boolean semantic = null;   //On/Off Semantic dis-
covery, null=auto

    public Integer limit = 300;
    public Integer histogramLimit = 0;
    public Double score = 1.0;          //downscore points per
Shape issue
    public Integer shapeTotalScore = 0;
    public Double shapeSensitivity = 0.00;
    public Integer shapeMaxPerCol = 0;
    public Integer shapeMaxColSize = 0;
    public String behavioralDimension = StringUtils.EMPTY;
    public String behavioralDimensionGroup = StringUtils.EMPTY;
    public String behavioralValueColumn = StringUtils.EMPTY;
    public Boolean behaviorScoreOff = false;  // disable beha-
vior scoring
```

Dupe

```
package com.owl.common.options;

/**
 * Options for Dupe Activity
 */
public class DupeOpt {

    /**
     * Whether to run Dupe Activity
     */
    public Boolean on = false;           // --dupe

    /**
     * @deprecated Unused for Activity2
     */
    public Boolean only = false;         // --dupeonly

    /**
     * Column names to include Dupe Activity
     */
    public String[] include;             // -dupeinc

    /**
     * Column names to exclude Dupe Activity
     */
    public String[] exclude;             // dupeexc

    /**
     * Indicator for complexity. See Activ-
ity2.Dupe.Scala.execute()
     * depth == 0 : exact match (sets props.dupeExactMatch =
TRUE downstream)
     */
    public Integer depth = 2;            // -depth

    /**
     * The minimum dupe scores between two duplicates. (cur-
rently calculated as "edit distance", out of upperBound)
     * Two values with dupe score less than this is lowerBound
are not duplicates (i.e. "truly" different values)
     */
    public Integer lowerBound = 80;      // -dupelb, -dupecutoff

    /**
     * The maximum possible dupe score for duplicate records
(for a given dupe detection method).
     * Currently assumed to be 100.
     */
```

```
    public Integer upperBound = 100;    // -dupeub, -dupeper-
matchupperlimit

    /**
     * Approximate dupe score used to create block index (when
DF is large)
     */
    public Integer approximate = 1;     // -dupeapprox

    /**
     * Number of observations per unique duplicate
     */
    public Integer limitPerDupe = 15;

    /**
     * Whether to process column headers when data load uses
manual column names (LoadOpts.fileHeader)
     * TODO this belongs in LoadOpts, not DupeOpts
     */
    public Boolean checkHeader = true;

    /**
     * TODO remove
     *
     * @deprecated not used;
     */
    public String filter;

    /**
     * If true, dupe activity is case insensitive. If false,
dupe activity is case sensitive.
     * Convenience feature for upper and lower set to 100
     */
    public Boolean ignoreCase = false;      //-dupenocase

    /**
     * Number of points each duplicate contributes to the total
schema score (on the Findings page)
     */
    public Double score = 1.0;                //-
dupescore  points per duplicate found default 1

    /**
     * Number of unique duplicates to compute during dupe activ-
ity
     */
    public Integer limit = 300;               //-
dupelimit  default 300
```

Source

```
public class SourceOpt {

    public Boolean on = false;                //-vs
    public Boolean only = false;              //-sourceonly
    public Boolean validateValues = false;    //-validatevalues
    public Boolean matches = false;           //-matches

    public String[] include;                 //-valinc
    public String[] exclude;                 //-valexc
    public String[] key;                      //valkey
    public Map<String, String> map;           //
    public Double score = 1.0;                // points per val-
idate source found, default 1-5
    public Integer limit = 30;                //-valsrclimit
    public String delimiter = ",";            //-srcdel
    public String fileCharSet = "UTF-8";      //-srcencoding
    public String filePath = StringUtils.EMPTY; //--srcfile

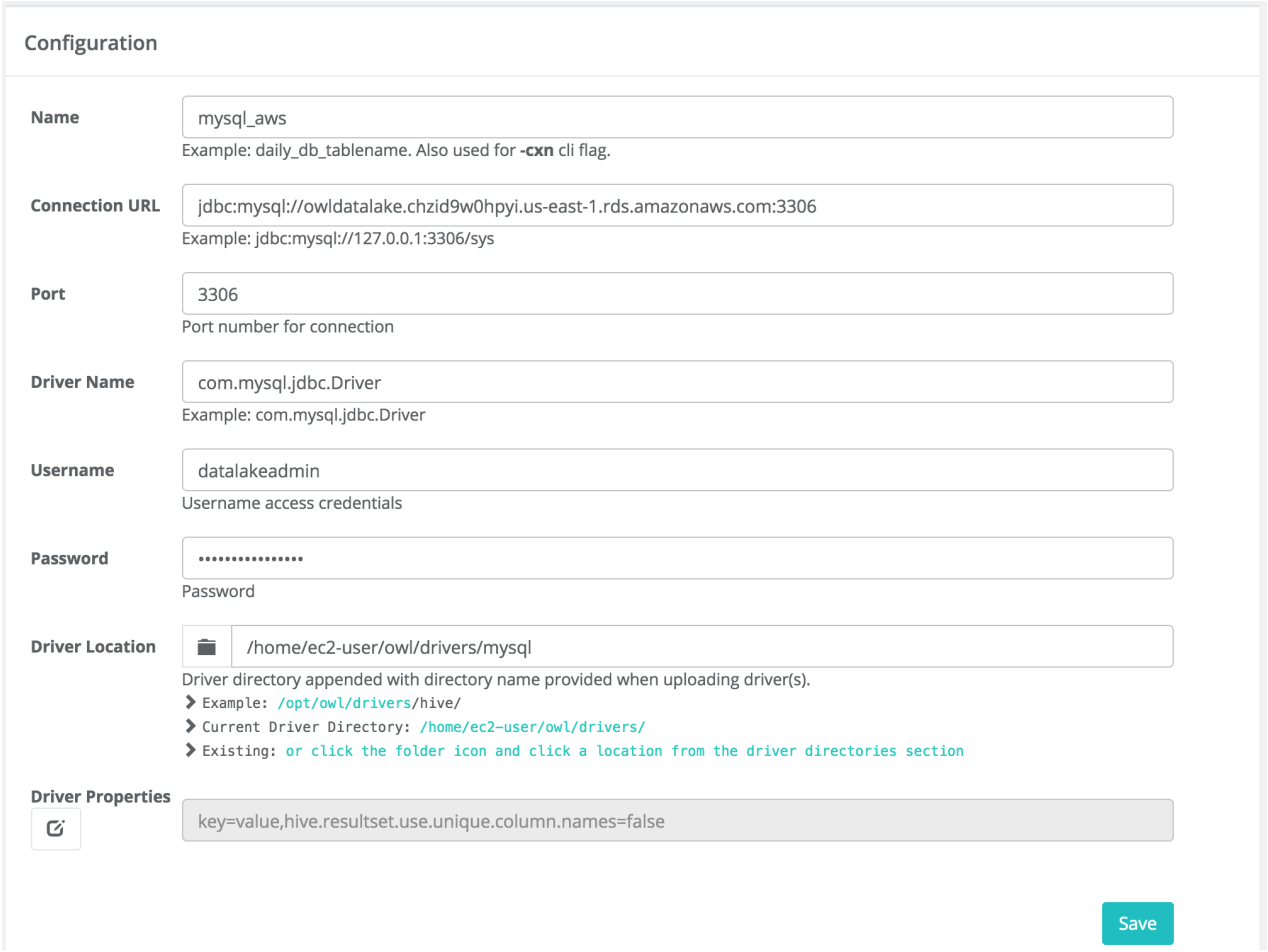
    public String header = null;              //-srcheader
    public String dataset = StringUtils.EMPTY;
    public String driverName = StringUtils.EMPTY;
    public String user = StringUtils.EMPTY;
    public String password = StringUtils.EMPTY;
    public String passwordManager = StringUtils.EMPTY;
    public String connectionName = StringUtils.EMPTY;
    public String connectionUrl = StringUtils.EMPTY;
    public String query = StringUtils.EMPTY;
    public String fileQuery = StringUtils.EMPTY;
    public String lib = StringUtils.EMPTY;
    public Properties connectionProperties;
```

Command Line

DQ Job JDBC

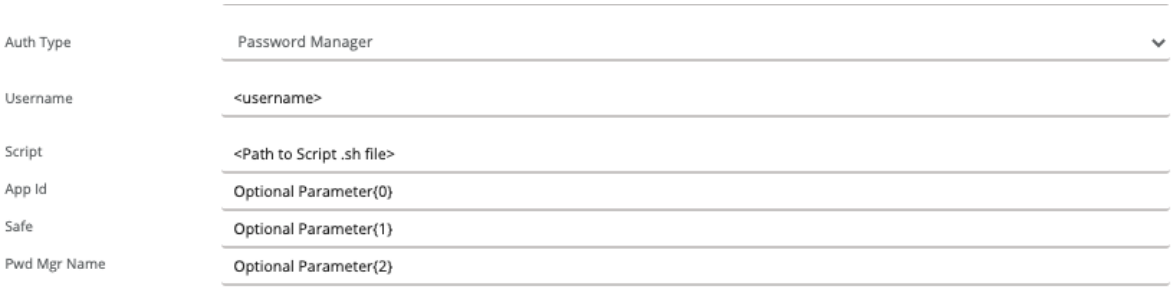
Connect to any database via JDBC.

```
-q "select * from lake.stock_eod where date = '2017-01-20' " \
-u username -p password \
-c "jdbc:mysql://instance.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306" \
-rd "2017-01-20" \
-dc "date" \
-ds "stocks" \
-driver com.mysql.jdbc.Driver \
-lib "/home/ec2-user/owl/drivers/mysql/"
```



Password Manager

You can configure Collibra DQ to call a script file to retrieve a password from a password manager vault or other storage container. The customer is responsible for generating a script to pull just the password and DQ uses that value dynamically when the connection is needed for the UI or when kicking off an DQCheck.



In the connection dialog, select Password Manager from the Auth Type dropdown, and supply a user name. The script is the path to the .sh script on the machine where the web application is running, and the user account that runs DQ-web should be allowed to execute the script. You can either use the optional parameters or pass any parameters your script needs directly inline on the Script value.

Fetch Size

It is important to consider the drivers fetch size when loading greater than 1 Million rows across the network. DQ allows you to set this driver property in the WebApp but this is only for web interaction therefore "fetchsize" will not help here. DQ also allows fetchsize in the DQCheck by passing in a connection property.

CMD line

```
-connectionprops "fetchsize=3000"
```

Notebook

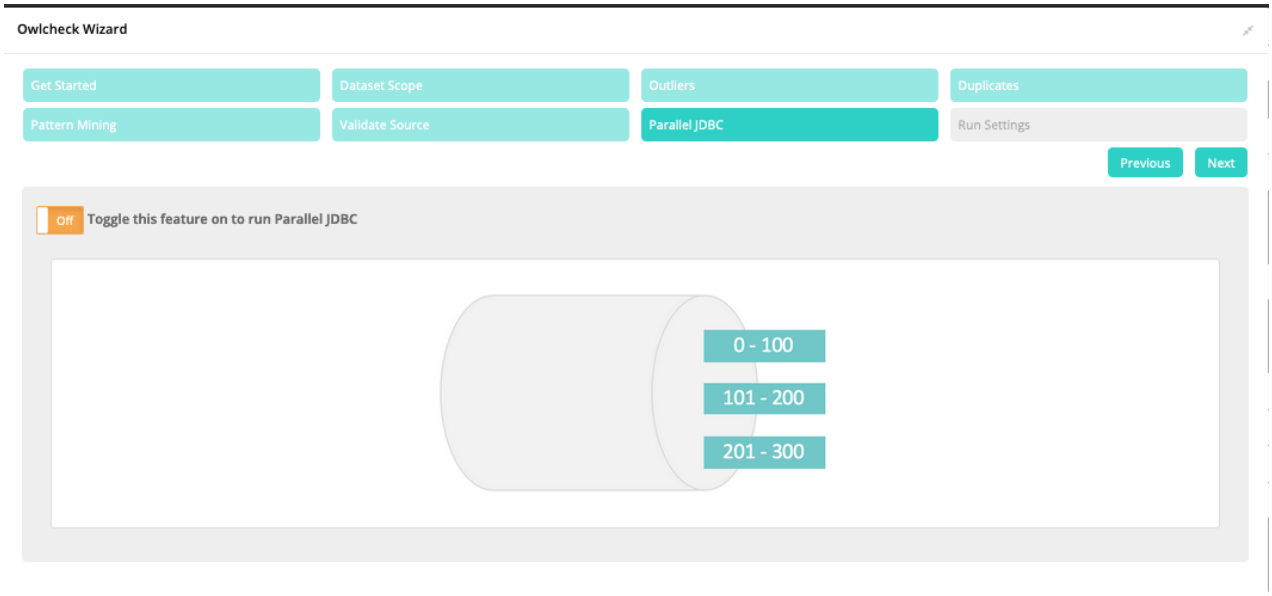
```
props.connectionProps.put("fetchsize", "3000")
```

Parallel JDBC

For greater performance or moving large datasets across a network DQ supports parallel JDBC, which can be enabled by passing `numpartitions` to DQCheck. This can be a 2-5X improvement in many cases.

```
-lib "/opt/owl/drivers/mysql8/"
-cxn mysql
-q "select * from lake.nyse where trade_date = '${rd}' "
-rd 2018-01-01
-ds nyse
-columnname volume
-numpartitions 4
-lowerbound "0"
-upperbound "5000000000"
-usesql
```

DQ also supports auto parallelization, which will configure the `numPartitions` parameter for you based on the size of your data. This is enabled in the UI when you create a dataset using the DQCheck wizard.



DQ Job Hive

Run a data quality check on a Hive table. Use the -hive flag for a native connection via the HCat, this does not require a JDBC connection and is optimized for distributed speed and scale.

Hive Native, no JDBC Connection

Open source platforms like HDP, EMR and CDH use well known standards and because of this DQ takes advantage of things like HCat and it removes the need for JDBC connection details

as well as offers optimum data read speeds. DQ recommends and supports this with the -hive flag.

```
./owlcheck -ds hive_table -rd 2019-03-13 \
-q "select * from hive_table" -hive
```

Example output. A hoot is a valid JSON response

```
{
  "dataset": "hive_table",
  "runId": "2019-02-03",
  "score": 100,
  "behaviorScore": 0,
  "rows": 477261,
  "prettyPrint": true
}
```

Hive JDBC

1. You need to use the hive JDBC driver, commonly org.apache.hive.HiveDriver.
2. You need to locate your driver JDBC Jar with the version that came with your EMR, HDP or CDH
    a. This jar is commonly found on an edge node under /opt/hdp/libs/hive/hive-jdbc.jar etc...

```
./owlcheck -rd 2019-06-07 -ds hive_table \
-u <user> -p <pass> -q "select * from table" \
-c "jdbc:hive2://<HOST>:10000/default" \
-driver org.apache.hive.HiveDriver \
-lib /opt/owl/drivers/hive/ \
-master yarn -deploymode client
```

HDP Driver - org.apache.hive.HiveDriver

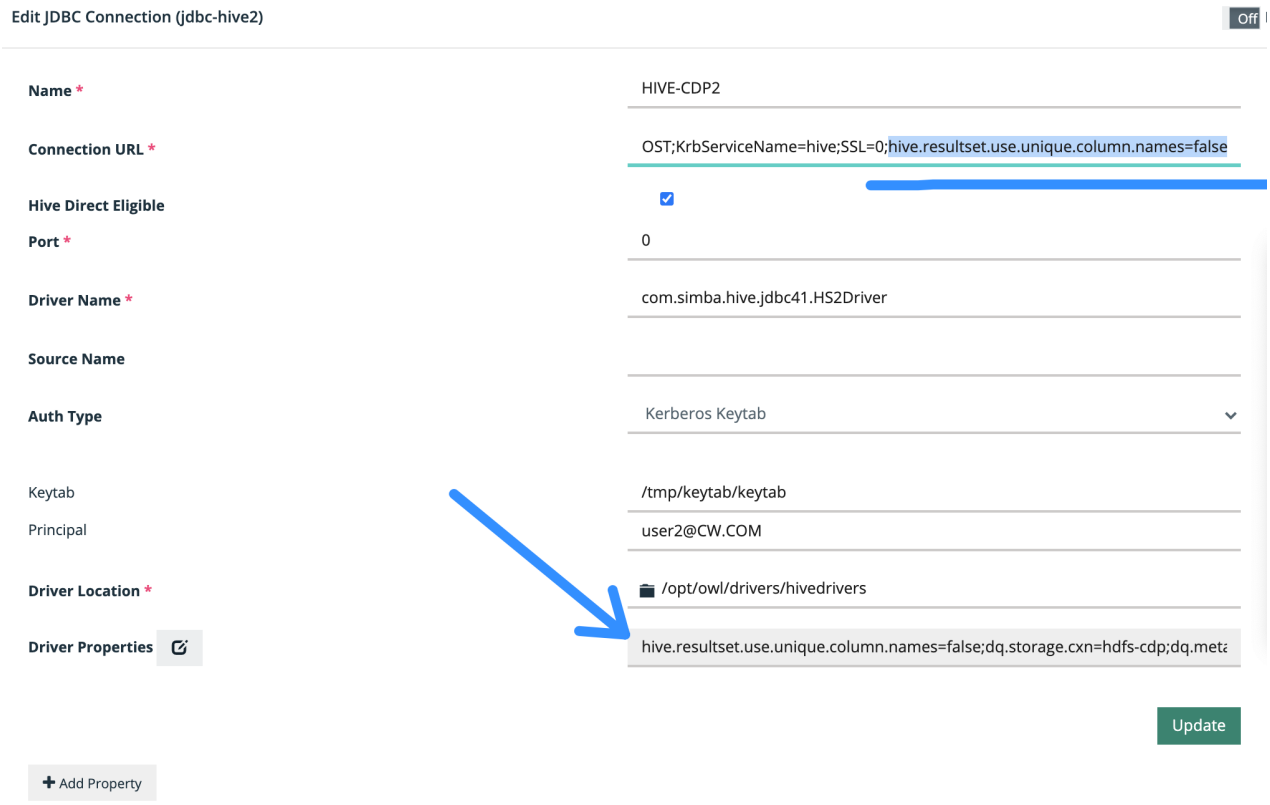
CDH Driver - com.cloudera.hive.jdbc41.Datasource

For CDH all the drivers are packaged under, HiveJDBC41_cdhversion.zip.

Troubleshooting

A common JDBC connection is hive.resultset.use.unique.column.names=false.

This can be added directly to the JDBC connection url string or to the driver properties section.



Test your hive connection via beeline to make sure it is correct before going further.

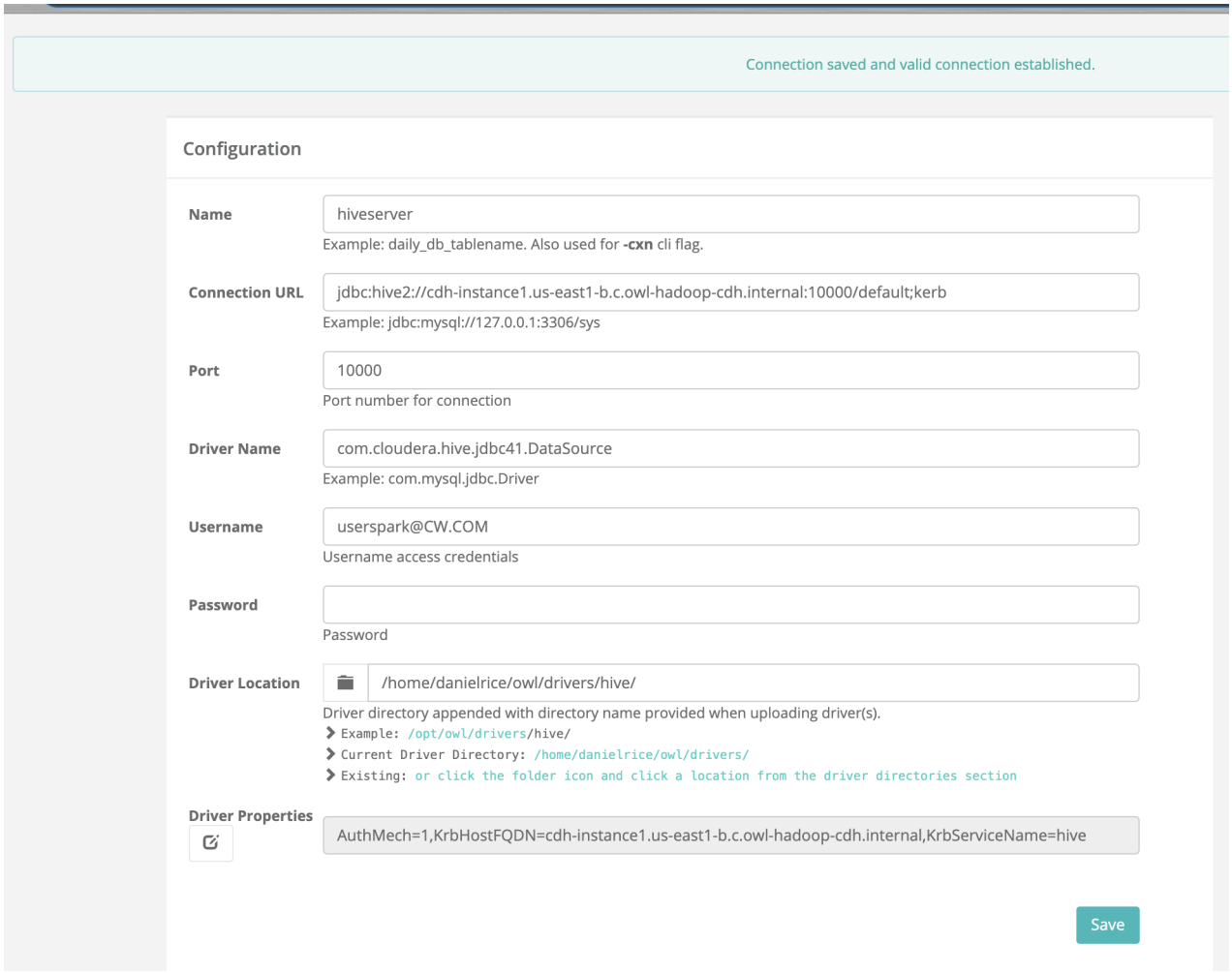
```
beeline -u 'jdb-
c:hive2://<HOST>:10000/default;principal=hive/cdh-instance1.us-
east1-b.c.company-hadoop-cdh.internal@CW.COM;useSSL=true' -d
org.apache.hive.jdbc.HiveDriver
```

Kerberos Example

```
jdbc:hive2://<HOST>:10000/default;principal=hive/cdh-
instance1.us-east1-b.c.company-hadoop-cdh.in-
ternal@CW.COM;useSSL=true
```

Connecting DQ WebApp to Hive JDBC

Notice the driver properties for kerberos and principals.



In very rare cases where you can't get the jar files to connect properly one workaround is to add this to the DQ-web startup script.

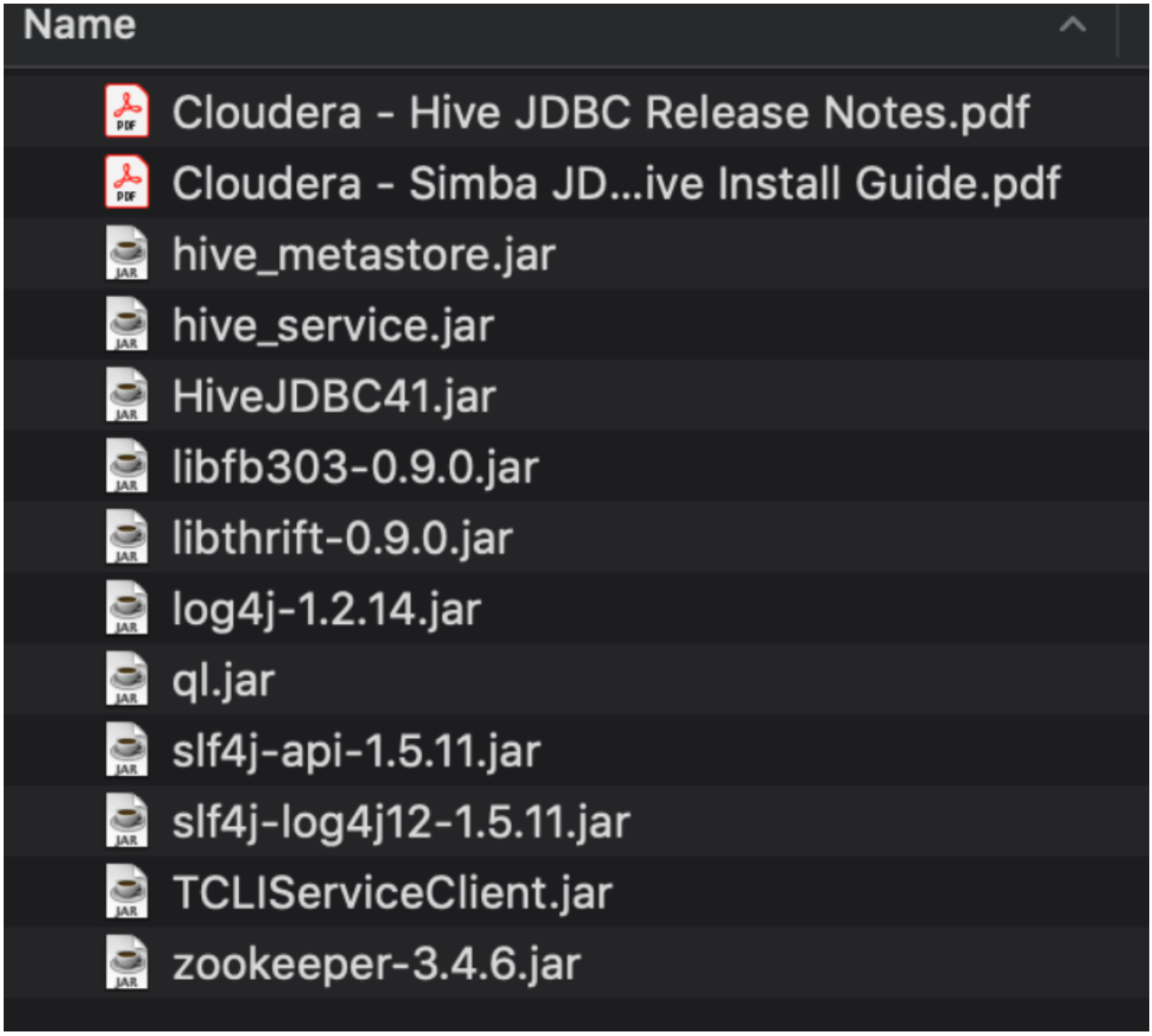
```
$JAVA_HOME/bin/java -Dload-
er.path=lib,/home/danielrice/owl/drivers/hive/ \
-DowlAppender=owlRollingFile \
-DowlLogFile=owl-web -Dlog4j.configurationFile=file://$INSTALL_
PATH/config/log4j2.xml \
$HBASE_KERBEROS -jar $owlweb $ZKHOST_KER \
--logging.level.org.springframework=INFO $TIMEOUT \
--server.session.timeout=$TIMEOUT \
--server.port=9001 > $LOG_PATH/owl-web-app.out 2>&1 & echo $!
>$INSTALL_PATH/pids/owl-web.pid
```

Class Not Found apache or client or log4j etc...

Any class not found error means that you do not have the "standalone-jar" or you do not have all the jars needed for the driver.

Hive JDBC Jars

It is common for Hive to need a lot of .jar files to complete the driver setup.



Java jar cmds

Sometimes it is helpful to look inside the jar and make sure it has all the needed files.

```
jar -tvf hive-jdbc.jar
```

DQ Job Files

For example, a large file `transaction_2021-01-01.csv` might contain the following transaction data with two transaction per day spanning all of January.

transaction_id	account_id	date	amount
1	1	2021-01-01	100
2	2	2021-01-01	120
3	1	2021-01-02	90
4	2	2021-01-02	115
...		...	...
61	1	2021-01-31	100
62	2	2021-01-31	999

and this file might be located on the directory `~/customer/transaction-2021-01-01/`.

```
~/customer
   ├── transaction-2021-01-01
   │   └── transaction_2021-01-01.csv
   ├── transaction-2021-02-01
   │   └── transaction_2021-02-01.csv
   ... # folders for 2021-03-01 to 2021-11-01 ommitted
   ├── transaction-2021-12-01
   │   └── transaction_2021-12-01.csv
```

Other folders with similar pattern may exist in your directory, such as `~/customer/transaction-2021-02-1`. Note that February data is located in a separate directory with a similar pattern for all the months of 2021. This dataset could similarly have 2 account IDs and 1 transaction per account per day (= 28 x 2 = 56 rows of data). For this example, let's assume this is the case for all the files.

To run an DQCheck on this single file containing multiple dates, you have the following choices.

DQChecks with **file**

1. Run an DQCheck on all the rows in a single file.

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset"
    ... # other relevant options
```

Here we assume that run date (`-rd`) is "2021-01-01" because it is currently January 1, 2021. The above command would lead to an DQCheck on 62 rows of data spanning all of January 2021 from a single file located at `~/customer/transaction-2021-01-01/transaction_20210101.csv`. If you were to schedule a job to run this job monthly and next job ran on February 1st, 2021, then same DQ checks will be performed on the **same set of 62 rows** with same score as your DQCheck run from January 1, 2021. For example, the follow-up scheduled job running on February 1st, 2021 would be:

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-02-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset"
    ... # other relevant options
```

This type of DQCheck on a single file is suitable if you are verifying a static file `~/customer/transaction-2021-01-01/transaction_20210101.csv` that does not change over time and expect the score to be the same every run. Hence, it is suggested to name the dataset that reflect this, such as `DQCheck_transaction_jan21` to reflect the idea that this dataset is checking the Data Quality of transaction table containing January 2021 data. Similar DQCheck for February 2021 data would then be a separate and independent dataset named `DQCheck_transaction_feb21` This type of DQCheck can also be used if `~/customer/transaction-2021-01-01/transaction_20210101.csv` is changing (the rows are changing values or new rows are being added) and want to detect data quality changes. Transaction file doesn't fit with this scenario, but the idea is that the above command

specifies Data Quality DQChecks on the entirety of the file. The run date is a date that you choose to assign for that DQCheck. It is conventional to have one-to-one mapping between run date and the date corresponding to the date that DQ checks are being performed. Run date does not have to match with the data underlying the file.

2. Run an DQCheck on subset of rows from a single file

The single file contains daily data for January of 2021. To run Data Quality checks on January 1st, January 2nd, … , and January 31st, you need to run 31 DQChecks, each with subset of rows from the file. Note the `where` clause in `-fq` matching with the run date `-rd`

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-01'"
    ... # other relevant options

./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-02"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-02'"
    ... # other relevant options

... # DQ checks for -rd 2021-01-03 to 2021-01-30 ommitted

./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-31"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-31'"
    ... # other relevant options
```

By using the same dataset name `-ds`, all 31 DQCheck will appear under one dataset `DQCheck_transaction_jan21` on the Findings page.

A convenient way to parameterize this run date is to use `${rd}` in the query.

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-01"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '${rd}'"
    ... # other relevant options
```

A daily scheduled job starting on January 1st, 2021 to January 31, 2021 will automatically replace the `${rd}` with "2021-01-01", "2021-01-02", ... , and "2021-01-31" for the respective run date.

3. Run an DQCheck on subset of rows from a single file with day lookback

For certain core components like **Outlier**, a set of rows corresponding to historical training data can be used to establish a baseline. For example, the row with `transaction_id` 62 has amount of 999. This looks like an outlier that we want to catch. This value of 999 seems to be an outlier because past transaction amounts for `account_id`2 are in the 100s range. We can use historical data from January 15th to January 30th and use that info to see if January 31st data contains any outliers. In this scenario, our single file `~/customer/transaction-2021-01-01/transaction_2021-01-01.csv` contains such historical data because that file contains all the data for all of January. How do we use the same file for both current data (January 31st) and historical (January 15th to January 30th) data? You do not have to split the files into two. You can simply do exactly what you would do for DQCheck on "2021-01-31" with a `-fullfile` flag. The `-fullfile` flag tells the DQCheck that "the file in `-f` contains the historical data. Construct a query and subset those rows for me".

```
./owlcheck
    -ds DQCheck_transactions_jan21
    -rd "2021-01-31"
    -f "~/customer/transaction-2021-01-01/transaction_2021-01-
01.csv"
    -fq "select * from dataset where date = '2021-01-31'"
    -fullfile
    # outlier options
    -dc "date"
    -dl
    -tbin "DAY" # look back time bin is day
    -dllb 15 # look back up to 15 days
    ... # other relevant options
```

DQChecks with *multiple files*

4. Run an DQCheck on a single file with lookback using series of file

Recall our folder structure:

```
~/customer
   ├── transaction-2021-01-01
   │   └── transaction_2021-01-01.csv
   ├── transaction-2021-02-01
   │   └── transaction_2021-02-01.csv
   ... # ommitted for space
   ├── transaction-2021-12-01
   │   └── transaction_2021-12-01.csv
```

If we want to run an DQCheck for December of 2021 and use July of 2021 to November of 2021 as our historical training dataset, how can we load *multiple files*? Just like how `-fullfile` provides a convenient way to create historical training dataset on a single file, `-fllb` (file lookback) provides a convenient way to load *series of files* with patterns while still pointing to the target file (December file) in `-f`

```
./owlcheck
    -ds DQCheck_transactions_dec21
    -rd "2021-12-01"
    -f "~/customer/transaction-2021-12-01/transaction_2021-12-
01.csv"
    -fq "select * from dataset"
    -fllb
    # outlier options
    -dc "date"
    -dl
    -tbin "MONTH"
    -dllb 5 # look back up to 5 months
    ... # other relevant options
```

One caveat to this `-fllb` method __ is that the DQCheck history must be "primed" first so that the DQ knows the file path of the past series of files. In fact, `-fllb` does not use the file path provided in `-f` and loads different files from different folders. *It relies on the DQCheck history* under the same `-ds` name. `-fllb` means lookback up to N number of past consecutive DQChecks. For each of those past DQCheck, look up the file path `-f` used in the past and follow those paths. The number N is determined by the maximum number of lookbacks from

Outlier (`-dllb`) and Patterns (`-fpglb`). In the DQCheck above, because `-dllb 5` is provided along with `-fllb`, it means "Look up 5 past DQChecks and load those files as historical dataset". In summary, in order to run an DQCheck on "2021-12-01" and have that DQCheck for that date "look up" the files in `~/customer/transaction-2021-07-01/transaction_2021-07-01.csv,~/customer/transaction-2021-08-01/transaction_2021-08-01.csv,~/customer/transaction-2021-09-01/transaction_2021-09-01.csv,~/customer/transaction-2021-10-01/transaction_2021-10-01.csv,` and `~/customer/transaction-2021-11-01/transaction_2021-11-01.csv`, you need to have ran DQChecks for "2021-07-01", "2021-08-01", ... , and "2021-11-01" under the same dataset name. Therefore, it would be more logical, best-practice is to name the dataset `-ds DQCheck_transaction_2021` and run series of monthly owlchecks up to "2021-12-01" (but the name of the dataset is up to you)/

```
# Prime past DQ checks so that "2021-12-01" knows the file path
of past months
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-07-01"
    -f "~/customer/transaction-2021-07-01/transaction_2021-07-
01.csv"
    -fq "select * from dataset"

./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-08-01"
    -f "~/customer/transaction-2021-08-01/transaction_2021-08-
01.csv"
    -fq "select * from dataset"


./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-09-01"
    -f "~/customer/transaction-2021-08-01/transaction_2021-09-
01.csv"
    -fq "select * from dataset"

./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-10-01"
    -f "~/customer/transaction-2021-08-01/transaction_2021-10-
01.csv"
    -fq "select * from dataset"

./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-11-01"
    -f "~/customer/transaction-2021-11-01/transaction_2021-11-
01.csv"
    -fq "select * from dataset"

# Priming 5 past DQ checks complete. Now run the 2021-12-01
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-12-01"
    -f "~/customer/transaction-2021-12-01/transaction_202-11-
201.csv"
    -fq "select * from dataset"
    -fllb
    # outlier options
    -dc "date"
    -dl
```

```
    -tbin "MONTH"
    -dllb 5 # look back up to 5 months
    ... # other relevant options
```

In this scenario, since the folder paths have a pattern, we can use `-br` for priming in one command instead of writing five DQCheck commands. The flag `-br` runs DQChecks consecutively from the past and increments by monthly if `-tbin "MONTH"` (by default `-tbin DAY`so the default behavior is to increment daily). The different folder paths on each past consecutive run dates are replaced with `${rd}`. The below command is identical to the above:

```
# Prime past DQ checks so that "2021-12-01" knows the file path
of past months
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-12-01"
    -f "~/customer/transaction-${rd}/transaction_${rd}.csv"
    -fq "select * from dataset"
    -br 5 # run 5 runs to in the past
    -tbin "MONTH" # <-- required since we want a MONTHLY back-
run. Default "DAY"

# Priming complete. Now run the 2021-12-01
./owlcheck
    -ds DQCheck_transactions_2021
    -rd "2021-12-01"
    -f "~/customer/transaction-2021-12-01/transaction_2021-12-
01.csv"
    -fq "select * from dataset"
    -fllb
    # outlier options
    -dc "date"
    -dl
    -tbin "MONTH"
    -dllb 5 # look back up to 5 months
    ... # other relevant options
```

This pattern is designed so that a single DQCheck command can be scheduled and `${rd}` be used to replace the folder & file path. Your `~/customer` folder could contain transactions for all the years, spanning all the way back to 1992 and into the future like so:

```
~/customer

   ├── transaction-1992-01-01
   │   └── transaction_1992-01-01.csv
   ├── transaction-1992-01-01
   │   └── transaction_1992-01-01.csv
   ... # ommitted for space
   ├── transaction-2021-01-01
   │   └── transaction_2021-01-01.csv
   ├── transaction-2021-02-01
   │   └── transaction_2021-02-01.csv
   ... # ommitted for space
   ├── transaction-2021-12-01
   │   └── transaction_2021-12-01.csv
   ... # hasn't happened yet!
```

In this scenario, a monthly scheduled job would get rid of the need to "prime" the DQCheck history, since your past scheduled jobs would have already ran the past DQChecks.

DQ Job JSON

Files

Run against a file using -json. Additionally, options are available for -flatten and -multiline. This is helpful for nested and various formats.

```
-ds json_file_example \
-f s3a://bucket_name/file.json \
-h instance.us-east4-c.c.owl-node.internal:5432/postgres \
-master spark://instance.us-east4-c.c.owl-node.internal:7077 \
-json \
-flatten \
-multiline
```

> **Note** Automatic flattening will infer schema and explode all structs, arrays, and map types.

Using Spark SQL

```
-ds public.json_sample \
-lib "/opt/owl/drivers/postgres/" \
-h instance.us-east4-c.c.owl-node.internal:5432/postgres \
-master spark://instance.us-east4-c.c.owl-node.internal:7077
-q "select * from public.jason"
-rd "2021-01-17"
-driver "org.postgresql.Driver"
-cxn postgres-gcp
-fq "select  \
get_json_object(col_3, '$.data._customer_name') AS  `data_cus-
tomer_name` , \
get_json_object(col_3, '$.data._active_customer') AS  `data_act-
ive_customer` , \
from dataset "
```

Note Pass in the path to Collibra DQ's -fq parameter. This is great for mixed data types within a database. For example, if you store JSON data as a string or a blob among other data.

```
// Flatten
val colArr = new JsonReader().flattenSchema(df.schema)
colArr.foreach(x => println(x))
```

Note This Collibra DQ utility traverses the entire schema and print the proper get JSON object spark sql strings. You can use this instead of typing each query statement into the command line -fq parameter as seen above.

Using DQ Libraries

```
import com.owl.common.options._
import com.owl.core.util.OwlUtils
import com.owl.core.activity2.JsonReader

val connProps = Map (
  "driver"   -> "org.postgresql.Driver",
  "user"     -> "user",
  "password" -> "password",
  "url"      -> "jdbc:postgresql://10.173.0.14:5432/postgres",
  "dbtable"  -> "public.data"
)

// Spark
var rdd = spark.read.format("jdbc").options
(connProps).load.select($"col_name").map(x=>x.toString()).rdd
var df = spark.read.json(rdd)

// Flatten
val colArr = new JsonReader().flattenSchema(df.schema)
val flatJson = df.select(colArr: _*)
flatJson.cache.count

// Opts
val dataset = "json_example"
val runId = s"2021-01-14"
val opt = new OwlOptions()
opt.dataset = dataset
opt.runId = runId
opt.datasetSafeOff = true

// DQ check
OwlUtils.resetDataSource("instance.us-east4-c.c.owl-node.in-
ternal","5432/postgres","user","pass", spark)
val owl = OwlUtils.OwlContext(flatJson, opt)
owl.register(opt)
owl.owlCheck
```

Note
JsonReader()
This uses DQ's JsonReader to do the heavy lifting.

Known Limitations

- Array and nested array datatypes in JSON files are not supported.

DQ Job S3

S3 permissions need to be setup appropriately.

> **Note** S3 connections should be defined using the root bucket. Nested S3 connections are not supported.

Example Minimum Permissions

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketMultipartUploads",
                "s3:ListBucket",
                "s3:ListMultipartUploadParts",
                "s3:GetObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:athena:*:<AWSAccountID>:workgroup/prima-
ry",
                "arn:aws:s3:::<S3 bucket name>/*",
                "arn:aws:s3:::<S3 bucket name>",
                "arn:aws:glue:*:<AWSAccountID>:catalog",
                "arn:aws:glue:*:<AWSAccountID>:database/<databas-
e name>",
                "arn:aws:glue:*:<AWSAccountID>:table/<database
name>/*"
            ]
        }
    ]
}
```

(Needs appropriate driver) http://central.maven.org/maven2/org/apache/hadoop/hadoop-aws/ Hadoop AWS Driver hadoop-aws-2.7.3.2.6.5.0-292.jar

```
-f "s3a://s3-location/testfile.csv" \
-d "," \
-rd "2018-01-08" \
-ds "salary_data_s3" \
-deploymode client \
-lib /home/ec2-user/owl/drivers/aws/
```

Databricks Utils Or Spark Conf

```
val AccessKey = "xxx"
val SecretKey = "xxxyyyzzz"
//val EncodedSecretKey = SecretKey.replace("/", "%2F")
val AwsBucketName = "s3-location"
val MountName = "kirk"

dbutils.fs.unmount(s"/mnt/$MountName")

dbutils.fs.mount(s"s3a://${AccessKey}:${SecretKey}@${AwsBuck-
etName}", s"/mnt/$MountName")
//display(dbutils.fs.ls(s"/mnt/$MountName"))

//sse-s3 example
dbutils.fs.mount(s"s3a://$AccessKey:$SecretKey@$AwsBucketName",
s"/mnt/$MountName", "sse-s3")
```

Databricks Notebooks using S3 buckets

```
val AccessKey = "ABCDED"
val SecretKey = "aaasdfwerwerasdfB"
val EncodedSecretKey = SecretKey.replace("/", "%2F")
val AwsBucketName = "s3-location"
val MountName = "abc"

// bug if you don't unmount first
dbutils.fs.unmount(s"/mnt/$MountName")

// mount the s3 bucket
dbutils.fs.mount
(s"s3a://${AccessKey}:${EncodedSecretKey}@${AwsBucketName}",
s"/mnt/$MountName")
display(dbutils.fs.ls(s"/mnt/$MountName"))

// read the dataframe
val df = spark.read.text(s"/mnt/$MountName/atm_customer/atm_cus-
tomer_2019_01_28.csv")
```

Advanced

A DQ Check is a bash script that is essentially the launch point for any DQ Job to scan a data set. A data set can be a flat file, such as textfile, json file, parquet file, etc, or a table from any number of databases, such as Oracle, Postgres, Mysql, Greenplum, DB2, SQLServer, Teradata, etc.

Example Run a data quality check on any file by setting the file path.

```
./owlcheck -ds stock_trades -rd 2019-02-23 -f /path/to/file.csv
-d ,
```

Example output below. A hoot is a valid JSON response

```
{
  "dataset": "stock_trades",
  "runId": "2019-02-03",
  "score": 100,
  "behaviorScore": 0,
  "rows": 477261,
  "passFail": 1,
  "peak": 1,
  "dayOfWeek": "Sun",
  "avgRows": 0,
  "cols": 5,
  "activeRules": 0,
  "activeAlerts": 0,
  "runTime": "00:00:23",
  "dqItems": {},
  "datashapes": [],
  "validateSrc": [],
  "alerts": [],
  "prettyPrint": true
}
```

Monthly Data

Sometimes you may want to run monthly profiles with aggregated data. In this case, the scheduling tool can supply the $ as a variable such as $runDate and the end date as $endDate. 1 line examples for bash or shell below.

```
echo "Hello World Collibra DQ"

runDate=$(date +"%Y-%m-%d")
endDate=$(date -d "$runDate +1 month" +%Y-%m-%d)

echo $runDate
echo $endDate

./owlcheck \
-q "select * from table where date >= '$runDate' and date <
'$endDate' " \
-ds example \
-rd $runDate \
-tbin MONTH
```

Monthly BackRun (Using Collibra Data Quality's built-in Monthly)

Collibra Data Quality has 2 convenient features here:

1. The use of built-in $ and $ removes the need for any shell scripting.
2. Using -br, DQ will replay 20 months of data using this template automatically.

```
./owlcheck \
-q "select * from table where date >= '${rd}' and date <
'${rdEnd}' " \
-ds example
-rd 2019-01-01
-rdEnd 2019-02-01
-tbin MONTH
-br 20
```

Daily Data

One of the most common examples is data loading or running once a day. A job control framework can pass in this value or you can pull it from shell.

```
echo "Hello World Collibra DQ"

runDate=$(date +"%Y-%m-%d")
echo $runDate

./owlcheck \
-q "select * from table where date = '$runDate' " \
-ds example \
-rd $runDate \
-tbin DAY
```

Daily Data (Using Collibra Data Quality's built-in Daily)

```
./owlcheck \
-q "select * from table where date = '${rd}' " \
-ds example \
-rd 2019-03-14
```

Daily Data with Timestamp instead of Date

```
./owlcheck \
-q "select * from table where TS >= '${rd} 00:00:00' and TS <=
'${rd} 23:59:59' " \
-ds example \
-rd 2019-03-14
```

OR Timestamp using $

```
./owlcheck \
-q "select * from table where TS >= '${rd} 00:00:00' and TS <
'${rdEnd} 00:00:00' " \
-ds example \
-rd 2019-03-14 \
-rdEnd 2019-03-15 \
-tbin DAY
```

Hourly Data

```
./owlcheck \
-q "select * from table where TS >= '${rd}' and TS < '${rdEnd}'
" \
-ds example \
-rd    "2019-03-14 09:00:00" \
-rdEnd "2019-03-14 10:00:00" \
-tbin HOUR
```

DQ Check Template with Service Hook

The best practice is to make a generic job that would be repeatable for every DQ Check. Below is an example that first hits Collibra Data Quality using a REST call and then runs the response.

```
curl -X GET "http://$host/v2/-
getowlchecktemplate?dataset=lake.loan_customer" \
-H "accept: application/json"
```

The above REST call returns the below DQ Check. It is left up to the Job Control to replace the $ with the date from the Job Control system. You can use Collibra DQ's built-in scheduler to save these steps.

```
./owlcheck \
-lib "/home/danielrice/owl/drivers/mysql/" \
-cxn mysql \
-q "select * from lake.loan_customer where load_dt = '${rd}' " \
-key post_cd_num -ds lake.loan_customer \
-rd ${rd} \
-dc load_dt -dl -dlkey usr_name,post_cd_num -dllb 5 \
-tbin DAY -by DAY -dupe -dupeinc ip_address_home,usr_name -
dupecutoff 85 \
-fpgon -fpgkey usr_name,post_cd_num -fpgdc load_dt -fpglb 5 -
fpgtbin DAY \
-loglevel INFO \
-h $host:5432/owltrunk \
-owluser {user}
```

REST API End Point

The easiest option is to use the **runtemplate** end point API call to make requests to from cmdLine or JobControl System. This endpoint gets the DQ Check saved in Collibra instead of the client needing to know the DQ Check details.

POST https://$host**/v2/runtemplate?dataset=lake.spotify**

RunTemplate

Parameters

Path

dataset	string	name of dataset. -ds OR opt.dataset
rd	string	yyyy-MM-dd format can add time or timezone. if note passed in it will use the current day

rdEnd	string	yyyy-MM-dd format can add time or timezone. if not passed it will not be used

Responses

- 200

```
{
  "msg": "Success, DQ Check is Running as process 13996",
  "pid": "13996",
  "runid": "2017-01-01",
  "starttime": "Thu Oct 17 13:27:01 EDT 2019",
  "cmd": "cmd": "-ds lake.spotify -rd 2019-10-17   -q \"select *
from lake.spotify\" -cxn mysql -lib /opt/owl/drivers/mysql/ -
drivermemory 2G -histoff   -owluser {user}",
  "dataset": "lake.spotify"
}
```

Curl example for the above Rest Call

```
TOKEN=$(curl -s -X POST http://$host/auth/signin -H "Content-
Type:application/json" -d "{\"username\":\"$username\", \"pass-
word\":\"$password\"}" | jq -r '.token')

curl -i -H 'Accept: application/json' \
  -H "Authorization: Bearer ${TOKEN}" \
  http://$host/v2/runtemplate?dataset=lake.spotify
```

Bash Script

A generic and repeatable DQCheck script for job schedulers, that hooks into Collibra to get the template.

```
#1 authenticate
curl -sb -X POST -d username={user} -d password={password}
http://$OWL_HOST/login -c cookies.txt

#2 get template
owlcheck_args=$(curl -b cookies.txt -H "accept: applic-
ation/json" -X GET http://$OWL_HOST/v2/-
getowlcheckcmdlinebydataset\?dataset=insurance | sed 's/.*\[\
(.*\)\]/\1/' | sed -e "s/^\"//" -e "s/\"$//"  | sed 's/\\\"\
(.*\)\\\"/\x27\1\x27/')

#3 replace ${rd} with job_run_date
job_run_date="2019-03-14 10:00:00"
owlcheck_args=${owlcheck_args//'${rd}'/$job_run_date}

#4 run owlcheck
eval owlcheck $owlcheck_args
```

For more Information on Collibra Data Quality's Scheduler, visit the DQ Job Cron page.

DQ Job Back Run

How to Replay a Data Test

Many times you will want to see how a dataset plays out over time. This could be five days or five months. Using this slider the tool will automatically create training sets and profiles as well as run any rules or outliers you've put in place.

Quickly Replay 30 days of data, -br 30

Add -br to any DQCheck and replay in time order. Jan 1st, Jan 2nd, Jan 3rd...To do this we need to use the ${rd} variable that DQ provides as a run_date replacement for job control and templates. Also note that if you run from the cmdline you need to escape "$"s. So use \${rd}. If you are running from a Notebook or Java or Scala or the Rest API you do not need to escape the ${rd} variable.

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 -rd "2018-01-14" \
-lib "/opt/owl/drivers/db2/" \
-cxn db2 \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE =
'\${rd}'" \
-br 4
```

Replay 4 Months of data, -br 4 -tbin MONTH

In situations where your data rolls up into Months, you may want to re-run several months of data but not a day at a time. In this case we will use -br with -tbin.

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 \
-rd "2018-01-01" \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE =
'\${rd}'" \
-br 4 \
-tbin MONTH \
-lib "/opt/owl/drivers/db2/" \
-cxn db2
```

Monthly using a range for the entire Month

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 \
-rd "2018-01-01" \
-rdEnd "2018-02-01" \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE >= '${rd}'
and TRADE_DATE < '${rdEnd}'" \
-br 4 \
-tbin MONTH
-lib "/opt/owl/drivers/db2/" \
-cxn db2
```

Monthly on the last day of the month

To create a back run for the last day each month, ensure your run date is the 31st of a month with 31 days in it and the back run time bin is set to `MONTH`. If a month included in the back run does not have 31 days, the back run job will run on the next available day. For example, if September (a month with only 30 days) is included in the scope of the back run, a back run will occur on September 30.

```
./owlcheck \
-ds OWLDB2.NYSE_STOCKS3 \
-rd "2024-12-31" \
-q "select * from OWLDB2.NYSE_STOCKS\
-br 4 \
-brbin MONTH
-lib "/opt/owl/drivers/db2/" \
-cxn db2
```

DQ Job Cron

Template for Job Control

Cron / Autosys / Control M / Oozie

It is common for organization to need to run jobs on a schedule. Below are a few shell tricks to get a date from bash and use an DQCheck with template variables.

Kinit and get run_date from shell or job control variable, pass it into Collibra DQ using $run_ date.

```
%sh
run_date=$(date +%Y-%m-%d)
run_date=$(date '+%Y-%m-%d %H:%M')
echo $run_date

#kinit
echo "password" | kinit  userabc@CW.COM

~/owl/bin/owlcheck -q "select * from lake.stock_eod where date =
'$run_date' " \
-u user -p pass \
-c "jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306" \
-rd "$run_date" \
-dc "date" \
-dl \
-dllb 7 \
-dlminhist 2 \
-tbin DAY \
-dlkey sym,exch \
-ds "lake.stock_nasdaq" \
-driver "com.mysql.jdbc.Driver" \
-lib "/home/ec2-user/owl/drivers/mysql/" \
-master yarn -deploymode client -numexecutors 1 -executormemory
1g \
-loglevel DEBUG
```

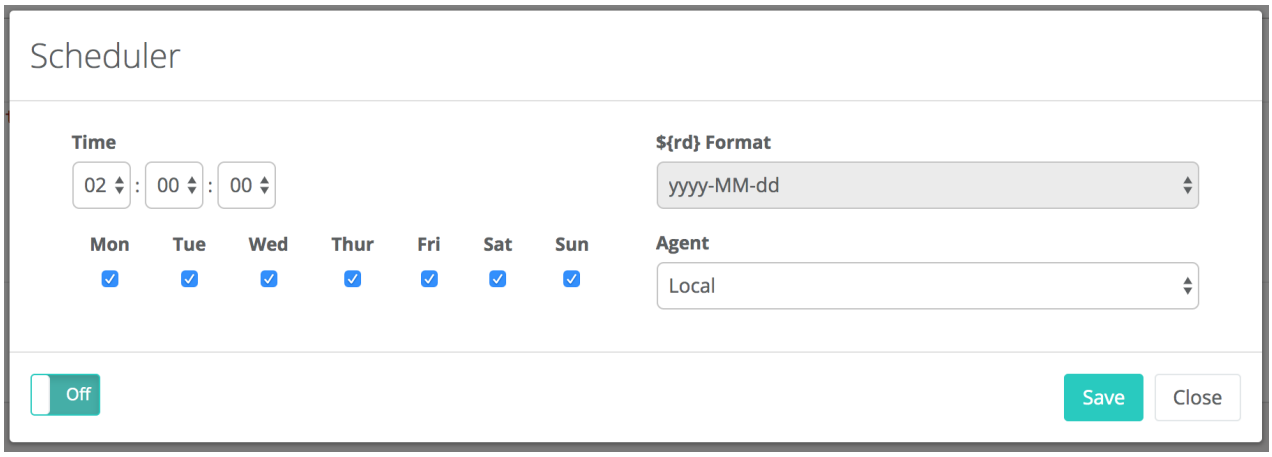
Template

You can also use -template to use DQ as a service hook and remove the need to pass in almost anything. In this case, DQ looks up the template automatically from either a previous run or if you've saved a template, and use these variables. Any variable at the cmdline will override and win/replace. This is a great way to remove connection and other information from being hard coded into the job control framework and allows edit ability from DQ Webapp.

```
%sh
~/owl/bin/owlcheck -usetemplate -ds lake.stock_nasdaq -rd $run_
date
```

Collibra DQ Scheduler - Built In

A quick option is to use DQ's built in scheduler. DQ automatically substitutes the runtime variables like $ into the job. This also gives you control to edit the DQCheck.

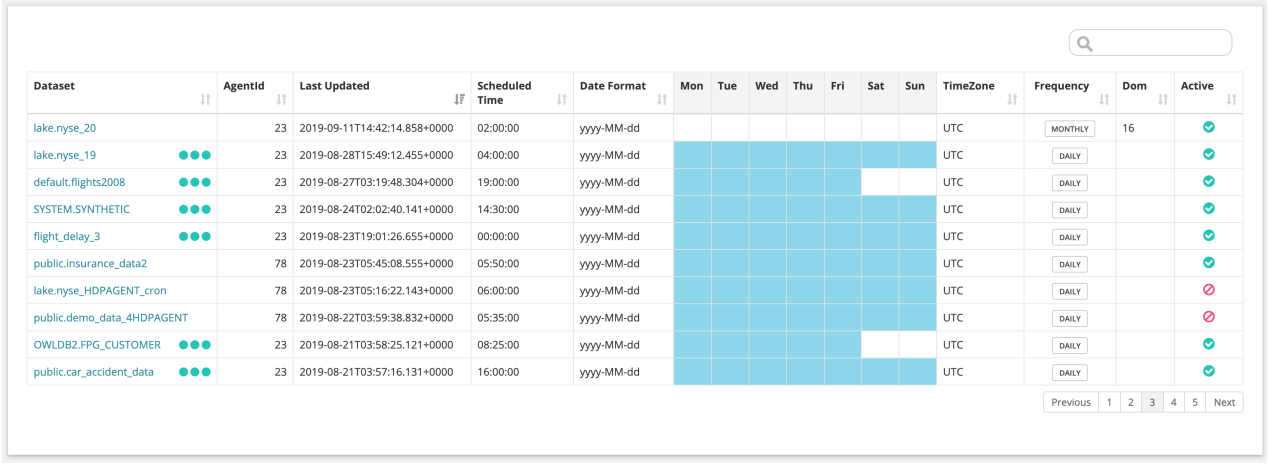


The schedule is based on the DQCheck Template. This way the runtime variables are replaced in each run. Notice the $ below.



All Scheduled Jobs in One Place

Under the jobs dashboard you can see an overview schedule with all running jobs and their status.



Dataset	AgentId	Last Updated	Scheduled Time	Date Format	Mon	Tue	Wed	Thu	Fri	Sat	Sun	TimeZone	Frequency	Dom	Active
lake.nyse_20	23	2019-09-11T14:42:14.858+0000	02:00:00	yyyy-MM-dd								UTC	MONTHLY	16	✓
lake.nyse_19	23	2019-08-28T15:49:12.455+0000	04:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
default.flights2008	23	2019-08-27T03:19:48.304+0000	19:00:00	yyyy-MM-dd	■	■	■	■	■			UTC	DAILY		✓
SYSTEM.SYNTHETIC	23	2019-08-24T02:02:40.141+0000	14:30:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
flight_delay_3	23	2019-08-23T19:01:26.655+0000	00:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
public.insurance_data2	78	2019-08-23T05:45:08.555+0000	05:50:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓
lake.nyse_HDPAGENT_cron	78	2019-08-23T05:16:22.143+0000	06:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		⊘
public.demo_data_4HDPAGENT	78	2019-08-22T03:59:38.832+0000	05:35:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		⊘
OWLDB2.FPG_CUSTOMER	23	2019-08-21T03:58:25.121+0000	08:25:00	yyyy-MM-dd	■	■	■	■	■			UTC	DAILY		✓
public.car_accident_data	23	2019-08-21T03:57:16.131+0000	16:00:00	yyyy-MM-dd	■	■	■	■	■	■	■	UTC	DAILY		✓

Previous 1 2 3 4 5 Next

DQ Job Link IDs

Link ID is an out-of-the-box feature that lets you link the findings of a DQ Job back to the source record, or key, for remediation outside the application. The link ID should be unique and is most commonly the primary key. Composite primary key is also supported.

For more information on Link IDs, go to Working with Link IDs.

Notebook

To define the Link ID in a notebook, use the following command:

```
val opt = new OwlOptions()
opt.runId = "2018-02-24"
opt.dataset = "orders"
opt.linkId = Array("transaction_id", "trans_time")
```

Command Line

To define the Link ID using the command line, use the following command:

```
./owlcheck -ds orders \
-rd "2018-02-24" \
-linkid transaction_id,trans_time
```

> **Note** For rules to use linkID, the columns need to be present in the select statement (either select * or select specific column names). All Simple rules are eligible for linkID and Freeform rules need to contain the columns in the projection part of the SQL statement.

Notebook API Example

```
+-----------+---------+------+------+-----+----------------
+--------------+
|     dataset|     runId|fieldNm|
format|count|          percent| transaction_id|
+-----------+---------+------+------+-----+----------------
+--------------+
|      order |2018-02-
24|  fname|xxxx'x.|    1|7.142857142857142|t-1232         |
+-----------+---------+------+------+-----+----------------
+--------------+
```

```
owl.getShapesDF
```

Rest API Example

When supplying a linkID, Collibra DQ naturally excludes this field from most activities, meaning a unique ID or primary key column can not be duplicative or it would not be the primary key. Because of this, it is not evaluated for duplicates. The same is true for Outliers and Shapes, as a large sequence number or other variations might trigger a false positive when this column is denoted to be simply for the purpose of linking uniquely back to the source. If you also want to evaluate this column and link it, create a derived column with a different name and Collibra DQ will naturally handle both cases.

```
owl.getShapes
owl.getDupes
owl.getOutliers
owl.getRuleBreaks
owl.getSourceBreaks
```

getRules()

```
----Rules----
+-----------------+----------+--------------------+-------------
-----+------+
|          dataset|     runId|              ruleNm|         rule-
Value|linkId|
+-----------------+----------+--------------------+-------------
-----+------+
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|fname like
'Kirk' |  c-41|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|fname like
'Kirk' |  c-42|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|fname like
'Kirk' |  c-43|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|fname like
'Kirk' |  c-44|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|fname like
'Kirk' |  c-45|
|dataset_outlier_3|2018-02-24|if_email_is_
valid...|             email|  c-31|
|dataset_outlier_3|2018-02-24|if_email_is_
valid...|             email|  c-33|
|dataset_outlier_3|2018-02-24|if_zip_is_valid_
Z...|               zip|  c-40|
+-----------------+----------+--------------------+-------------
-----+------+
```

getDupes()

First split on ~~ then if you have a multiple part key split on ~|.

```
----Dupes----
+-----------------+----------+-----+--------------------+-------
---+
|          dataset|     runId|score|                 key|    lin-
kId|
+-----------------+----------+-----+--------------------+-------
---+
|dataset_outlier_3|2018-02-24|  100|9ec828d5194fa397b...|c-
45~~c-36|
|dataset_outlier_3|2018-02-24|  100|1f96274d1d10c9f77...|c-
45~~c-35|
|dataset_outlier_3|2018-02-24|  100|051532044be286f99...|c-
45~~c-44|
|dataset_outlier_3|2018-02-24|  100|af2e96921ae53674a...|c-
45~~c-43|
|dataset_outlier_3|2018-02-24|  100|ad6f04bf98b38117a...|c-
45~~c-42|
|dataset_outlier_3|2018-02-24|  100|1ff7d50a7a9d07d02...|c-
45~~c-41|
|dataset_outlier_3|2018-02-24|  100|6ed858ed1f4178bb0...|c-
45~~c-40|
|dataset_outlier_3|2018-02-24|  100|d2903703b348fb4cb...|c-
45~~c-39|
|dataset_outlier_3|2018-02-24|  100|24bf54412de1e720d...|c-
45~~c-38|
|dataset_outlier_3|2018-02-24|  100|7a7ce0beb41b39564...|c-
45~~c-37|
+-----------------+----------+-----+--------------------+-------
---+
```

getRuleBreaks()

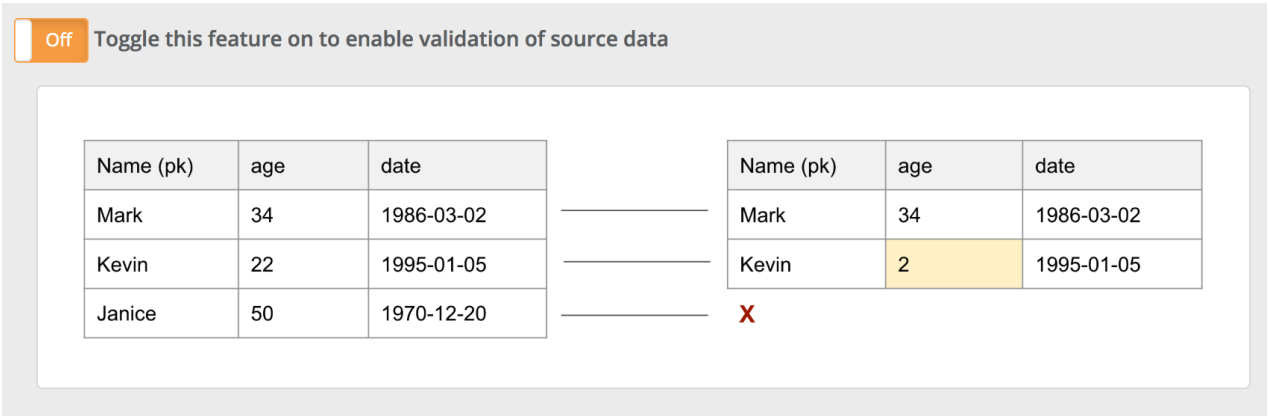
The getRuleBreaks endpoint retrieves all broken records within your dataset. There is no size limit to this API.

```
----Rule-Breaks----
+-----------------+----------+--------------------+------+
|          dataset|     runId|              ruleNm|linkId|
+-----------------+----------+--------------------+------+
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|  c-41|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|  c-42|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|  c-43|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|  c-44|
|dataset_outlier_3|2018-02-24|     fname_like_Kirk|  c-45|
|dataset_outlier_3|2018-02-24|if_email_is_valid...|  c-31|
|dataset_outlier_3|2018-02-24|if_email_is_valid...|  c-33|
|dataset_outlier_3|2018-02-24|if_zip_is_valid_Z...|  c-40|
+-----------------+----------+--------------------+------+
```

DQ Job Validate Source

Reconciliation

Commonly data driven organizations have a need to ensure that two tables or a table and file match. This match might be a daily reconciliation or any snapshot in time. Collibra DQ calls this Source to Target or Left to Right matching. It covers row differences, schema differences and all cell values.



Impala/Hive -> DB2

Below is an example of comparing a table in DB2 to the same table in Impala.

```
./owlcheck \
-lib "/home/install/owl/drivers/db2" \
-cxn db2 \
-q "select * from OWLDB2.NYSE_STOCKS where TRADE_DATE = '${rd}'
" \
-ds NYSE_STOCKS_VS \
-rd "2018-01-10" \
-vs \
-valsrckey SYMBOL \
-validatevalues \
-h $host/owltrunk \
-srcq "select * from nyse where TRADE_DATE = '${rd}' " \
-srccxn impala-jdbcuser \
-libsrc /home/isntall/owl/drivers/hivedrivers \
-jdbcprinc jdbcuser@CW.COM -jdbckeytab /tmp/jdbcuser.keytab \
-owluser admin \
-executorcores 4 -numexecutors 6 -executormemory 4g -driver-
memory 4g -master yarn -deploymode cluster \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM
```

DB2 -> Hive (Native)

Most databases only expose data through a JDBC connection but Hive offers a second path which does not require a JDBC connection. Hive has the ability to push down its processing to the local worker nodes and read directly from disk in the case when the processing is happening locally on a cluster. If your processing is not happening local to the cluster then you must use HiveJDBC. Take note of the -hive flag.

```
./owlcheck \
-hive \
-q "select * from nyse" \
-ds hiveNativeNyse \
-rd "2019-10-01" \
-vs \
-valsrckey exch,symbol,trade_date \
-validatevalues \
-srcq "select * from OWLDB2.NYSE_STOCKS" \
-srccxn db2 -libsrc /home/install/owl/drivers/db2 \
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
-sparkkeytab /home/install/owl/bin/user2.keytab -sparkprinc
user2@CW.COM
```

MySQL -> Oracle

This example compares the entire table instead of just a single day. Notice the 3 part valsrckey EXCH,SYMBOL,TRADE_DATE. Adding the date field ensures our key is unique and won't create a cartesian product. If the goal was to compare day over day with Oracle make sure to add TO_DATE('YYYY-MM-DD', '2019-10-01') to the where clause.

```
./owlcheck \
-lib /home/install/owl/drivers/mysql/ \
-cxn mysql \
-q "select * from lake.nyse" \
-ds lake.nyse \
-rd 2019-10-01 \
-vs \
-valsrckey EXCH,SYMBOL,TRADE_DATE \
-validatevalues \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM \
-srcq "select * from SYSTEM.NYSE" \
-srccxn oracle \
-libsrc /home/danielrice/owl/drivers/oracle/
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
```

File -> MySQL Table

Taking a file and loading it into a staging table or final table is a common part of every ETL process. However it is extremely common that the file values do not match or coherence into the table properly and these silent errors are usually not caught until a business user sees the data far long down stream.

```
./owlcheck \
-ds lake.nyse \
-rd 2019-10-01 \
-cxn "mysql" \
-q "select * from lake.nyse" \
-vs \
-valsrckey EXCH,SYMBOL,TRADE_DATE \
-validatevalues \
-srcfile "hdfs:///user/source/nyse.csv" \
-srcd "," \
-lib /home/install/owl/drivers/mysql/ \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM \
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
```

File -> File

DQ compares a File to a File. This is common in landing zones and staging areas where a file might be moved or changed and you need to know if anything changed or is incorrect.

```
./owlcheck \
-ds lake.nyse \
-rd 2019-10-01 \
-f "hdfs:///user/target/nyse.csv" \
-d "," \
-vs \
-valsrckey EXCH,SYMBOL,TRADE_DATE \
-validatevalues \
-srcfile "hdfs:///user/source/nyse.csv" \
-srcd "," \
-sparkkeytab /home/install/owl/bin/user2.keytab \
-sparkprinc user2@CW.COM \
-numexecutors 2 -executormemory 5g -drivermemory 4g -master yarn
-deploymode cluster \
```

Add Date Column

Example

```
./owlcheck \
-ds "datataset_date_column" \
-rd "2019-07-01" \
-f "/Users/Downloads/csv2/2019010.csv" \
-adddc
```

> **Note** Add date column will use the run date supplied and add a date column named DQ_RUN_ID.

```
-adddc
```

This is used when you are using datasets that do not contain a date column or a malformed date string.

Known Limitations

This feature is only available for files, not for database tables.

Cloudera CLASSPATH

What is a CLASSPATH?

A CLASSPATH is essentially a list of jars that get injected into a JVM on the start of a job execution. Like many applications, Spark can have jars injected when a job is run. Cloudera has defined a list of predefined jars (rightfully called classpath.txt):

```
/etc/spark2/conf/classpath.txt
```

That will get injected whenever Spark is called. Here is an example list of jars as defined within a cluster we have stood up @ Collibra DQ:

```
[danielrice@cdh-edge ~]$ cat /etc/spark2/conf/classpath.txt
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/activation-1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/aopalliance-1.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/apacheds-
i18n-2.0.0-M15.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/apacheds-
kerberos-codec-2.0.0-M15.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/api-asn1-
api-1.0.0-M20.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/api-util-
1.0.0-M20.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/asm-3.2.-
jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/avro-
1.7.6-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/aws-java-
sdk-bundle-1.11.134.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/azure-
data-lake-store-sdk-2.2.9.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
beanutils-1.9.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
beanutils-core-1.8.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
codec-1.4.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
codec-1.9.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
configuration-1.6.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
daemon-1.0.13.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
digester-1.8.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
el-1.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
logging-1.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
math-2.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
math3-3.1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/commons-
net-3.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/core-
3.1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/curator-
client-2.7.1.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/curator-
framework-2.7.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/curator-
recipes-2.7.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/disruptor-3.3.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/findbugs-
annotations-1.3.9-1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/guava-
11.0.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/guava-
12.0.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/guice-
3.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
annotations-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
ant-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
archive-logs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
archives-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
auth-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
aws-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
azure-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
azure-datalake-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
common-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
datajoin-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
distcp-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
extras-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
gridmix-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
hdfs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
hdfs-nfs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-app-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-common-2.6.0-cdh5.16.1.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-core-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-hs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-hs-plugins-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-jobclient-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-nativetask-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-client-shuffle-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
mapreduce-examples-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
nfs-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
openstack-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
rumen-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
sls-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
streaming-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-api-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-applications-distributedshell-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-applications-unmanaged-am-launcher-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-client-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-common-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-registry-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-applicationhistoryservice-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-common-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-nodemanager-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-resourcemanager-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hadoop-
yarn-server-web-proxy-2.6.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hamcrest-
core-1.3.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
annotations-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-cli-
ent-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-com-
mon-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
examples-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
external-blockcache-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
hadoop-compat-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
hadoop2-compat-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-it-
1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-pre-
fix-tree-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-pro-
cedure-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-pro-
tocol-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
resource-bundle-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
rest-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
rsgroup-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
server-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
shell-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hbase-
thrift-1.2.0-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/high-
scale-lib-1.1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hsqldb-
1.8.0.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/htrace-
core-3.2.0-incubating.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/htrace-
core4-4.0.1-incubating.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/httpclient-4.2.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/httpcore-
4.2.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/hue-plu-
gins-3.9.0-cdh5.16.1.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
annotations-2.2.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
core-2.2.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
core-asl-1.8.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
databind-2.2.3-cloudera.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
jaxrs-1.8.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
mapper-asl-1.8.10-cloudera.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jackson-
xc-1.8.10.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jamon-
runtime-2.4.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jasper-
compiler-5.5.23.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jasper-
runtime-5.5.23.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/java-xml-
builder-0.4.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/javax.inject-1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jaxb-api-
2.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jaxb-api-
2.2.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jaxb-
impl-2.2.3-1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jcodings-
1.0.8.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jets3t-
0.9.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jettison-
1.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jettison-
1.3.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jline-
2.11.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/joni-
2.1.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jruby-
cloudera-1.0.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsch-
0.1.42.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsp-2.1-
6.1.14.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsp-api-
2.1-6.1.14.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsp-api-
2.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/jsr305-
3.0.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/leveldbjni-all-1.8.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/log4j-
1.2.16.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/log4j-
1.2.17.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/metrics-
core-2.2.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/metrics-
core-3.0.2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/microsoft-windowsazure-storage-sdk-0.6.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/mockito-
all-1.8.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/netty-
3.10.5.Final.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/netty-
all-4.0.50.Final.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/okhttp-
2.4.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/okio-
1.4.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/paranamer-2.3.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/protobuf-
java-2.5.0.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/slf4j-
api-1.7.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/slf4j-
log4j12-1.7.5.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/snappy-
java-1.0.4.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/spark-
1.6.0-cdh5.16.1-yarn-shuffle.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/spymemcached-2.11.6.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/stax-api-
1.0-2.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/xercesImpl-2.9.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/xml-apis-
1.3.04.jar
```

```
/opt/cloudera/parcels/CDH-5.16.1-1.cdh5.16.1.p0.3/jars/xmlenc-
0.52.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/jars/zookeeper-3.4.5-cdh5.16.1.jar
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/lib/hadoop/LICENSE.txt
/opt/cloudera/parcels/CDH-5.16.1-1.cd-
h5.16.1.p0.3/lib/hadoop/NOTICE.txt
/usr/java/jdk1.8.0_131/lib/tools.jar
```

At each execution of any Spark job (including the use of spark-submit) this list of jars above will automatically get loaded.

What is a JAR?

A jar file is essential a compressed list of classes and methods. It is important to note that when jar files are built they will typically have an associated version number.

Someone can look at the contents of a jar file by executing:

```
jar -tvf phoenix-4.13.1-HBase-1.3-client.jar
```

Or you can wrap the above in a for loop that will look at the contents of every jar that might contain a method you are looking for.

```
for i in `ls -1 *.jar`;do jar -tvf $i | grep -i
htrace/trace;echo $i;done;
```

Common issues that can occur with CLASSPATH's

> **Note** **Caused by: java.lang.NoClassDefFoundError: org/apache/htrace/Trace** at org.apache.hadoop.hbase.zookeeper.RecoverableZooKeeper.exists (RecoverableZooKeeper.java:218) at org.apache.hadoop.hbase.zookeeper.ZKUtil.checkExists(ZKUtil.java:481) at org.apache.hadoop.hbase.zookeeper.ZKClusterId.readClusterIdZNode (ZKClusterId.java:65) at org.apache.hadoop.hbase.client.ZooKeeperRegistry.getClusterId (ZooKeeperRegistry.java:86) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.retrieveClusterId(ConnectionManager.java:850) at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.<init>(ConnectionManager.java:635)

When a situation like this occurs it means that a method cannot be found in the classpath for the job that is trying to execute. This can indicate a couple things:

1. The job cannot find a jar file that contains the method flagged (in the example above the org/apache/htrace/Trace method).
2. Sometimes different versions of the same jar file gets loaded and the first jar loaded will always win. Older jars that get loaded first may not have a method defined in new jars.

At DQ, we have solved CLASSPATH / CLASSLOAD issues by automatically injecting jars defined in our ***owl/libs*** directory, and allowing users the ability to simply toggle loading them or not.

Column Matching

How much is your redundant data costing you?

Reclaim Gigabytes of Redundant Data

As data engineers, first we copy files into a landing zone, next we load the files into a staging area. After that we transform (ETL) the data into the final table. Soon that same data is copied to a lake for other groups to run analytics on. Eventually a group of analysts will need the data

in another format and a data engineer will copy the data in a newly joined or transposed fashion. Sounds familiar?

The result is the same data or similar columns of the same data being copied many times. **The answer**: *Buy more hardware*... could be OR *run a Collibra DQ health report* and gain an understanding of how much data could be removed, reclaiming disk space and instantly seeing a return on investment after clicking the button.

Tabular breakdown of percentage of fingerprint matches

dataset_1	dataset_2	col_1	col_2	perMatch
silo.account	silo.user_account	networth	acc_branch	16
silo.account	silo.user_account	networth	networth	100
silo.account	silo.user_account	acc_name	acc_name	88
silo.account	silo.user_account	acc_name	acc_branch	0
silo.account	silo.user_account	acc_name	networth	0
silo.account	silo.user_account	acc_branch	acc_name	0
silo.account	silo.user_account	acc_branch	acc_branch	100
silo.account	silo.user_account	acc_branch	networth	14
silo.user_account	silo.user_account	acc_name	acc_branch	0
silo.user_account	silo.user_account	acc_name	networth	0
silo.user_account	silo.user_account	acc_branch	networth	14

Sometimes its not as simple as comparing two tables from the same database. Collibra DQ allows a technical user to setup multiple DB connections before executing health check.

```
import com.owl.common.Props
import com.owl.core.Owl

val c1 = new Connection()
c1.dataset = "silo.account"
c1.user = "user"
c1.password = "pass"
c1.query = "select id, networth, acc_name, acc_branch from
silo.account limit 200000"
c1.url = "jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306"

val c2 = new Connection()
c2.dataset = "silo.user_account"
c2.user = "user"
c2.password = "pass"
c2.query = "SELECT acc_name, acc_branch, networth FROM silo.ac-
count limit 200000"
c2.url = "jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306"

val props = new Props()
props.dataset = "colMatchTest1"
props.runId = "2017-02-04"
props.connectionList = List(c1,c2).asJava
props.colMatchBatchSize = 2
props.colMatchDurationMins = 3

val matchDF = new Owl(props).colMatchDF
matchDF.show

matchDF.createOrReplaceTempView("matches")
```

High level view of data overlap



Date Time Variable Options

Date Time Variables can enhance your query and file date templating and variable options. This allows easier scheduling and programmatic templating for common date variables.

> **Important** File names must use the `{yyyy}-{MM}-{dd}` format in file date templates.

Key	Function
${rd}	Replaces with -rd values in CMD. For example, 2023-01-20.
${rdEnd}	Replaces with -rdEnd values in CMD.
${yyyy}	Replaces with 4-digit year portion of -rd values in CMD. For example, 2023.
${yy}	Replaces with 2-digit year portion of -rd values in CMD. For example, 23.
${M}	Replaces with 1-digit month portion of -rd values in CMD.
${MM}	Replaces with 2-digit month portion of -rd values in CMD.
${MMM}	Replaces with 3 letter month name portion of -rd values in CMD For example, Jan, Jul, Dec.
${MMMM}	Replaces with long month name portion of -rd values in CMD For example, January, July, December.
${d}	Replaces with 1 digit day portion of -rd values in CMD.
${dd}	Replaces with 2 digit day portion of -rd values in CMD.
${HH}	Replaces with 2 digit hour portion of -rd values in CMD.
${KK}	Replaces with 1 digit hour portion of -rd values in CMD.
${mm}	Replaces with 2 digit minute portion of -rd values in CMD.
${ss}	Replaces with 2 digit second portion of -rd values in CMD.

Deploy Mode

Yarn and Cluster

Deploy Mode Client

```
--deploy-mode client
```

Deploy Mode Cluster

```
--deploy-mode cluster
```

Job Stuck in ACCEPTED State

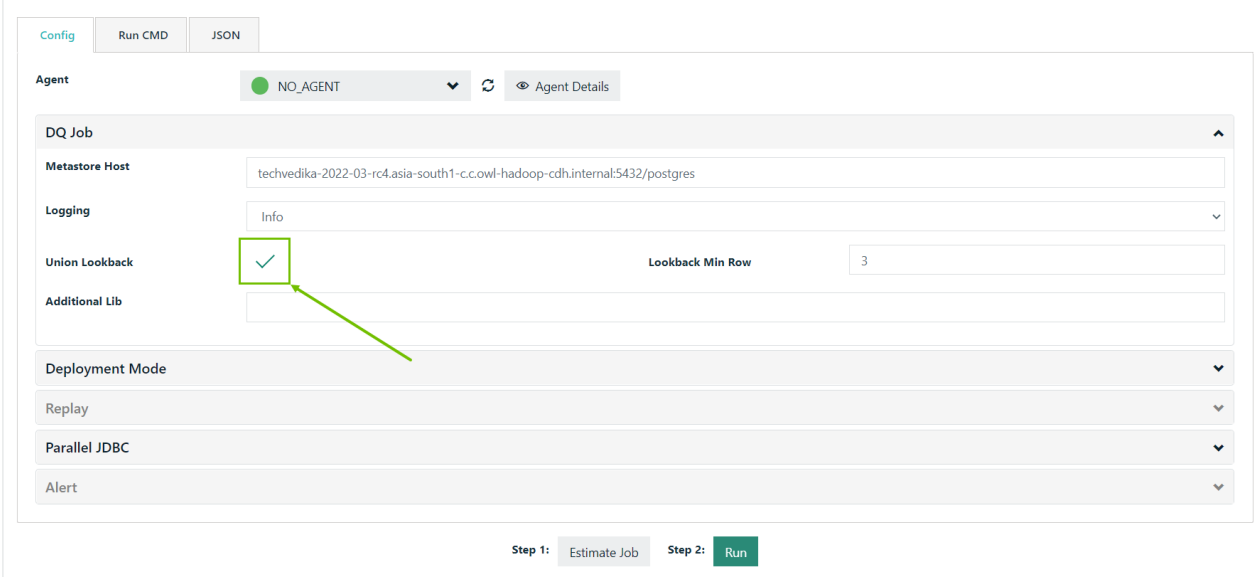
yarn.Client: Application report for application_1557720962505_0085 (state: ACCEPTED)
yarn.Client: Application report for application_1557720962505_0085 (state: ACCEPTED).

If running in cluster mode make sure you are passing in the below.

```
--deploy-mode cluster
--master yarn          # or spark master
-h 123.45.6.77:2181    # host to owl metastore
```

Lookback

Lookback lets you specify a historical period of past DQ scans to include in a DQ Job. For example, a lookback period of 5 includes DQ checks over the last 5 days of available data. By using the date column of the table, file, or view to scan the history of your dataset, lookback can help you identify potential outliers, patterns, and behavioral anomalies.



Important When a job that includes a lookback period runs, a temporary job appears in the Jobs queue. The temporary job will not resolve until all other lookback jobs are complete.

Lookback options

There are two lookback options to consider.

Option	Description
Union Lookback	Loads a specified number of historical runs of files on certain dates under the same dataset name and includes them in the DQ scans.
Full File Lookback	Includes the historical context of a single file in the outlier and/or patterns scans. To string together historical files that contain a single timeslice and include a larger historical window, use union lookback instead.

Union Lookback (-fllb)

Union Lookback, or File Lookback (-fllb) as it is also known, is used with deep learning and pattern matching. In the example below, it is used with deep learning.

File Lookback is used to check DQ Check history for previous files.

```
-fllb
```

This is often used with files and in conjunction with -adddc in cases where a date column is not in an ideal format or you do not have a date column on the given dataset.

Despite the name, this can be used with file or database storage formats.

> **Note** File look back (-fllb) should only be used when a SQL layer is not available. This is considered for advanced use cases, but may not be suitable for all file types and folder structures. Best practice is to expose a date signature somewhere in the file or directory naming convention.

Example

```
-ds "demo_lookback" \
-rd "2017-07-29" \
-lib "/opt/owl/drivers/mysql" \
-cxn "mysql" \
-q "select  * from lake.dateseries where DATE_COL = '2017-07-29'
" \
-dc DATE_COL \
-dl \
-dlkey sym \
-dllb 4 \
-fllb
```

> **Note** This look back will load your past 4 runs as your historical training set

Full File Lookback (-fullfile)

Like Union Lookback, Full File Lookback (-fullfile) is used with deep learning and pattern matching.

Fullfile Lookback uses the entire file for lookbacks instead of just filequery.

Understanding lookback command line flags

When you look at the DQ Job command line with lookback enabled, you may see several different lookback flags. The following table describes what each lookback flag means and to which DQ layers they apply.

Flag	Description	Layer
`-fllb`	Union lookback loads the DQ check history of files or database tables on certain dates under the same dataset name and includes them in the DQ Job.	Outliers and Patterns
`-fllbminrow`	The minimum number of rows in a dataset before it is included in a file lookback. This is automatically applied when Union Lookback is selected.	Outliers and Patterns
`-dllb`	Deep learning lookback loads DQ checks over a specified period of days and includes them in the scan for outliers.	Outliers
`-dlminhist`	An automatically generated flag based on the outlier lookback setting -dllb that defines the minimum number of days before DQ flags data as potential outliers. -dlminhistensures that the number of days in the algorithm is relative to the total scope of the lookback period.	Outliers
`-fullfile`	Includes the historical context of a single file in the outlier and/or patterns scans.	Outliers and Patterns
`-bhlb`	Behavior lookback loads a specified period of past DQ checks to include in a DQ Job. This controls the baseline profiling of a dataset.	Behavior
`-fpglb`	Pattern lookback loads DQ checks over a specified period of days and includes them in the scan for patterns.	Patterns

Filter & Filter Not

Filter & Filter Not is similar to a grep for limiting a dataframe to rows containing a substring.

> **Note** This feature should only be used when -q (query) and -fq (filequery) are not applicable. This feature is primarily used with raw files for limited filtering and not advanced conditional logic.

Example

```
./owlcheck \
-ds "dataset_name" \
-rd "2018-07-23 10" \
-d "," \
-f "/Users/Documents/file.csv" \
-filter "2018-07-23"
```

> **Note** **-filter "2018-07-23"**
> If file.csv contained multiple strings, but you only wanted rows containing "2018-07-23".

The inverse

> **Note** **-filternot "2018-07-23"**
> To exclude rows containing "2018-07-23".

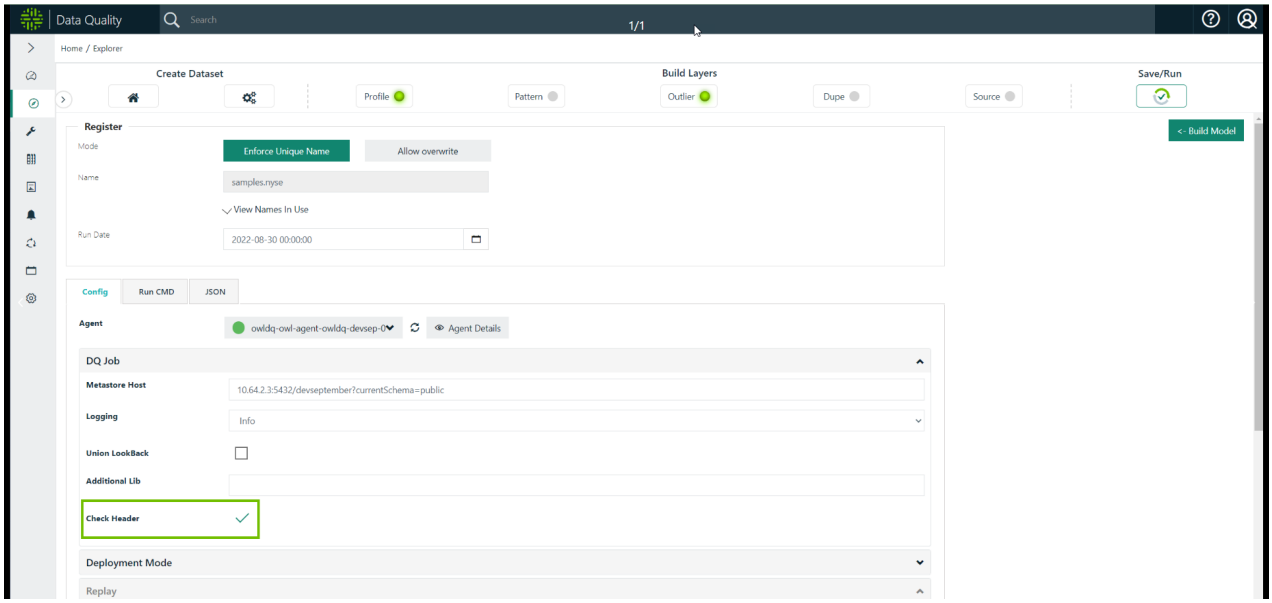
Header Check

Header Check lets you toggle column name detection on and off so column names containing special characters are not detected as schema changes. This is configurable with the **Check Header** checkbox or from the command line.

Configuring with the Check Header checkbox

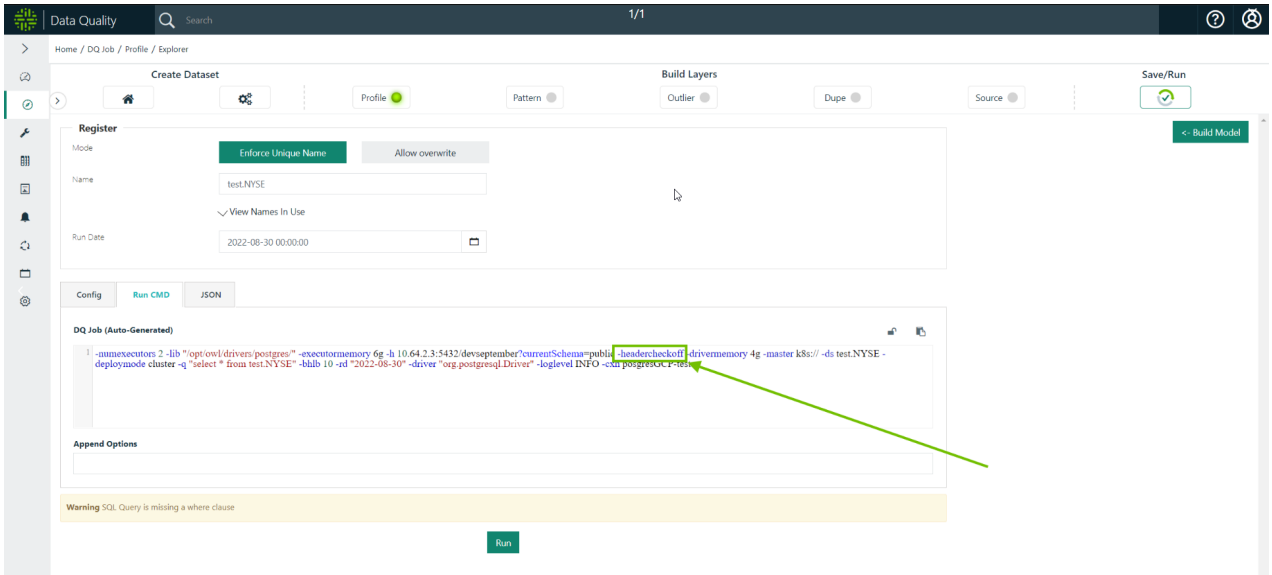
The Check Header checkbox is checked by default. When it is checked, schema findings do not display when detected.

To disable the header check, uncheck the Check Header checkbox. This allows the schema findings to display when detected.



Configuring from the command line

Header checks are enabled by default and do not appear in the command line when enabled. To disable header checks from the command line, click the lock icon to unlock the command line and use the -headercheckoff variable, as shown in the screenshot below. When you are done editing the command line, click the lock icon again to lock the command line; then, click **Run**.



Multiple Pattern Relationships

Run more than one relationship through pattern matching.

Example

```
./owlcheck \
-ds "fpg_multiple" \
-rd "2018-10-04" \

-cxn "postgres" \
-lib   "/opt/owl/drivers/postgres/" \
-q "select * from public.fpg_accounts where d_date = '2018-10-
04'" \

-fpgon \
-fpgdc "d_date" \
-fpglb "4" \
-fpgmulti "id,ssn_num=first_name,email|id,ssn_num=first_name,-
gender|id,ssn_num=last_name"
```

> **Note** Instead of -fpgkey and -fpgcol
> key1=cols1|keys2=cols2
> Enter multiple key=cols combinations separated by a pipe.

```
-fpgmulti "id,ssn_num=first_name,email|id,ssn_num=first_name,-
gender"
```

Nulls in Datasets

Example

```
./owlcheck \
-ds "datataset_date_column" \
-rd "2019-07-01" \
-f "/Users/Downloads/csv2/2019010.csv" \
-zfn \
-nulls "N.A."
```

Zero if null will replace null values with zero.

```
-zfn
```

To replace characters that represent a null to actual null values.

```
-nulls "N.A."
```

Spark-shell Sample

```
./bin/spark-shell --jars /opt/owl/bin/owl-core-trunk-jar-with-dependencies.jar,/opt/owl/drivers/postgres/postgresql-42.2.5.jar --deploy-mode client --master local[*]
```

Import lib's, if you get a dependency error, please import a second time.

```
import com.owl.core.util.{OwlUtils, Util}

import com.owl.common.domain2.OwlCheckQ

import com.owl.common.options._
```

Set up connection parameters to the database we want to scan if you don't already have a dataframe.

```
val`` ``url`` ``= "jdbc:postgresql://xxx.xxx.xxx.xxx:xxxx/db?currentSchema=schema"
val`` ``connProps``=``Map("driver" -> "org.postgresql.Driver","user" -> "user","password" -> "pwd","url" ->`` ``url","dbtable" -> "db.table")
```

Create a new OwlOptions object so we can assign properties.

```
val opt = new OwlOptions()
```

Set up variables for ease of re-use.

```
val dataset = "nyse_notebook_test_final"

val runId = "2017-12-18"
```

```
var date = runId

var query = s"""select * from <table> where <date_col> = '$date' """

val pgDatabase = "dev"val pgSchema = "public"
```

Set OwlOptions values to the metastore.

```
opt.dataset`` ``= datasetopt.runId`` ``= runIdopt.host`` ``=
"xxx.xxx.xxx.xxx"opt.pgUser`` ``= "xxxxx"opt.pgPassword`` ``=
"xxxxx"opt.port`` ``= s"5432/$pgDatabase?currentSchema=$pgSchema"
```

Create a connection, build the dataframe, register and run.

With inline processing you will already have a dataframe so you can skip down to setting the OwlContext.

```
val conn =`` ``connProps`` ``+ ("dbtable" -> s"($query)``
``$dataset")val df =`` ``spark.read.format("jdbc").options
(conn).load

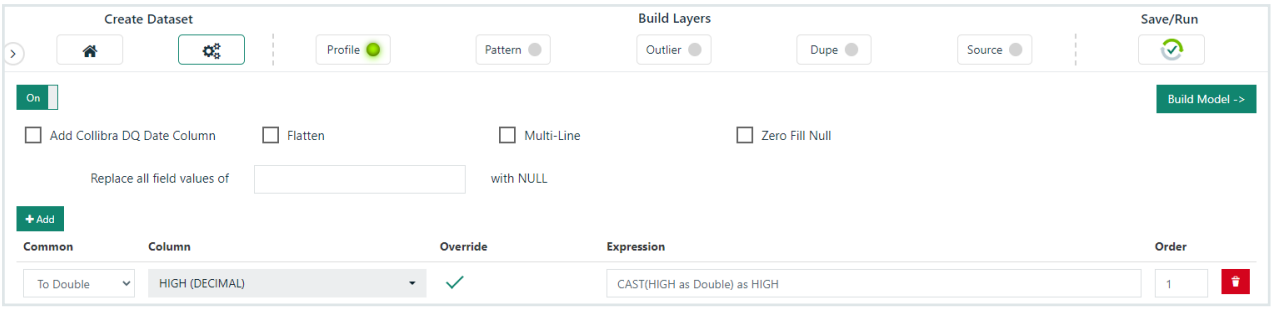
val owl = OwlUtils.OwlContext(df, opt)owl.register(opt)owl.owlCheck
```

Transform

Transform Date

When you create a dataset, you can transform columns such as Dates and Numbers to preferred formats. It is a common need to replace N.A. with nulls or empty white space.

> **Note** Because the Transform option defaults to a most common use case of `to_date`, you need to determine the proper transform function required for the data source in use.



Option	Description
Add Collibra DQ Date Column	Adds a synthetic run date column. **Tip** This can be helpful when your file does not contain dates that are easy to handle.
Flatten	Refers to the level of nesting in your file.
Multi-line	Refers to the content of the file being more than one line.
Zero Fill Null	Replaces NULL values with 0.
Replace all field values of [value] with NULL	Replaces all values of the column you specify with NULL.

Example

```
./owlcheck \
-ds  "dataset_transform" \
-rd  "2018-01-31" \
-f   "/Users/Documents/file.csv" \
-transform "purch_amt=cast(purch_amt as double)|return_amt=cast
(return_amt as double)"
```

Note Submit an expression to transform a string to a particular type.
In this example, transform the purch_amt column to a double.

```
-transform "purch_amt=cast(purch_amt as double)"
```

> **Note** Example of converting a string to a date.

```
-transform "RECEIVED_DATE=to_date(CAST(RECEIVED_DATE AS STRING),
'yyyyMMdd') as RECEIVED_DATE"
```

Converting an integer data type to a string

When you transform your target column is an integer data type, such as 20230414, you need to convert it to a string data type before you can convert it to a date.

1. From the Explorer page, select the **Transform** option.
2. In the **Expression** input field, enter the following expression:

   ```
   date_parse(cast(<your_column_name> as varchar), '%Y%m%d')
   ```

3. Click **Save**.

   > **Note** If your query does not contain a ${rd} in the where clause, you need to wrap it with the DATE() function in the command line. For example, `select * from default.dataset where date_parse(CAST(example_date_column as varchar), '%Y%m%d') = DATE('${rd}')`

Known Limitations

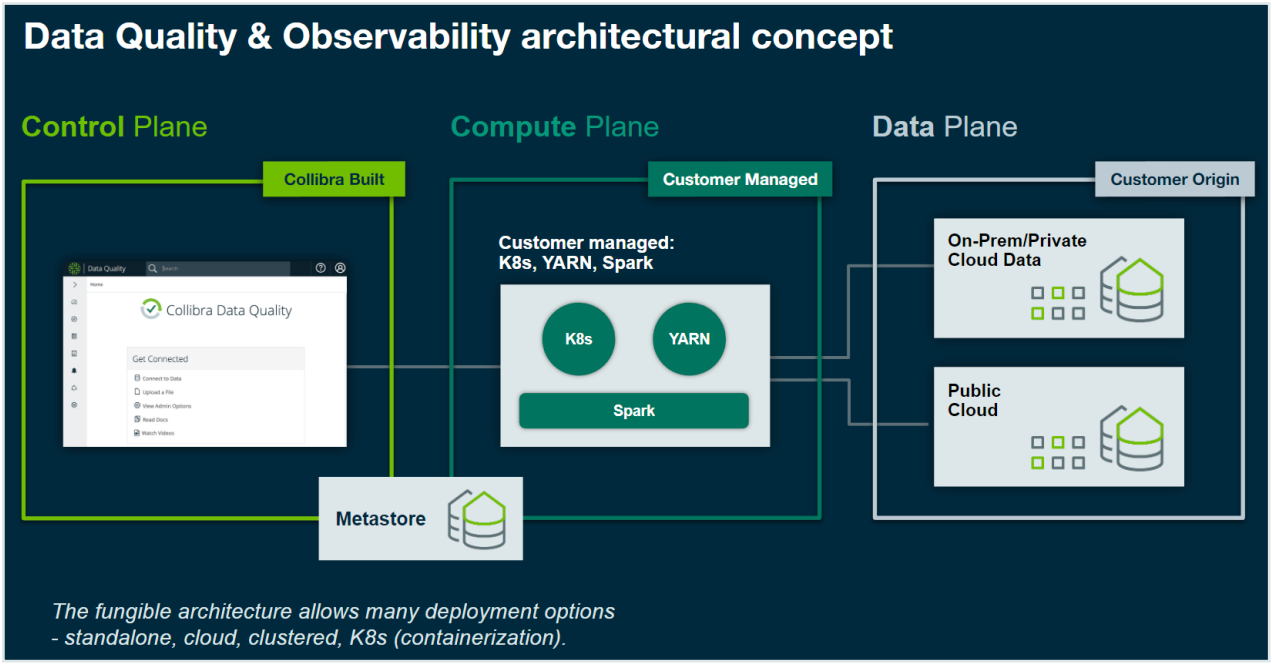
- The UI does not run multi-line. A workaround is to supply these options manually in the command line.
- Explorer can flatten nested JSON array, but DQ Core can only flatten nested JSON with JSON object. Because of this limitation, you may encounter a scenario where the results in Explorer are flattened, but not in DQ Core.

Chapter 13

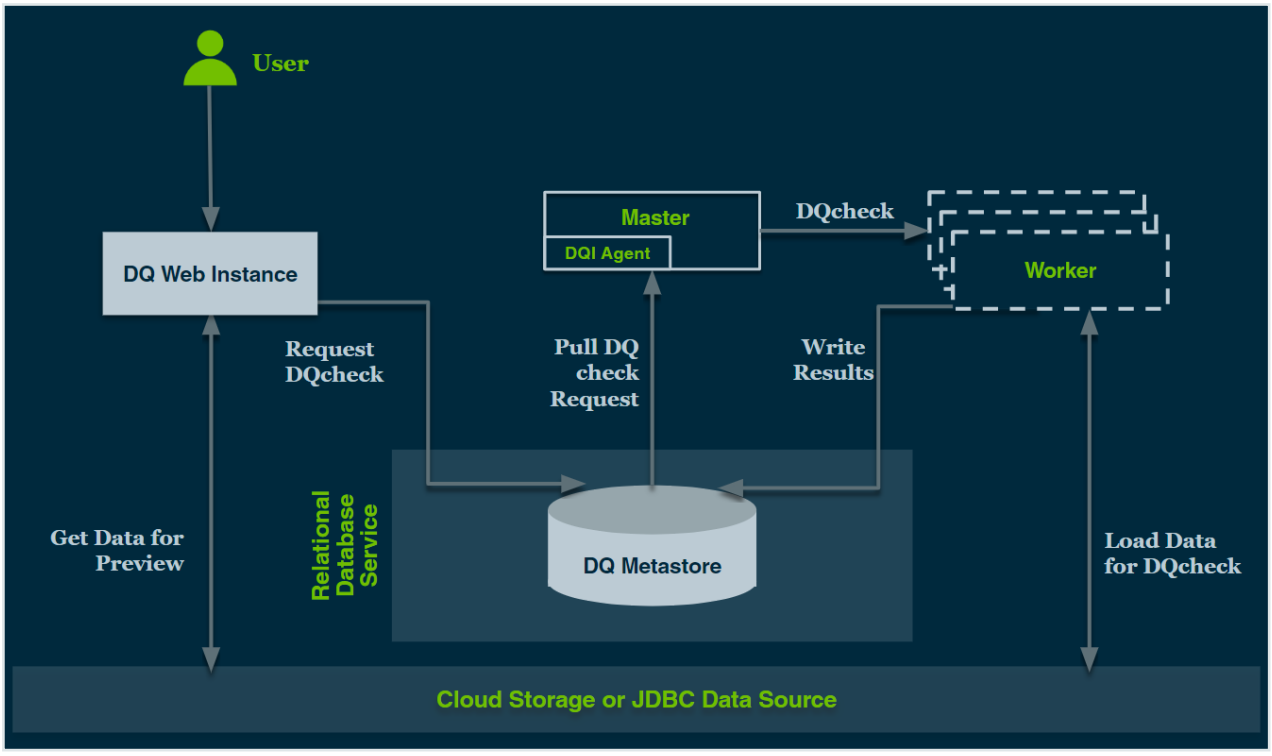
Architecture

Architecture Diagram

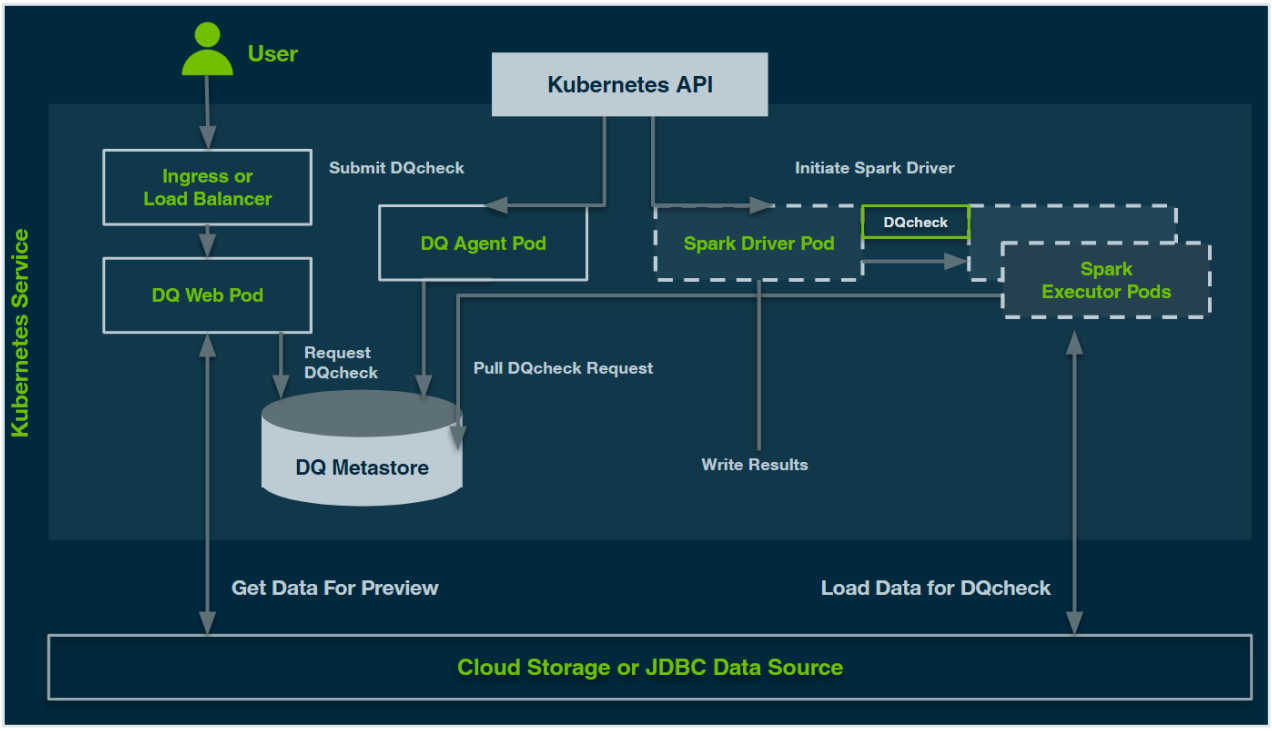
Collibra DQ Architecture



Collibra DQ Hadoop Deployment Diagram



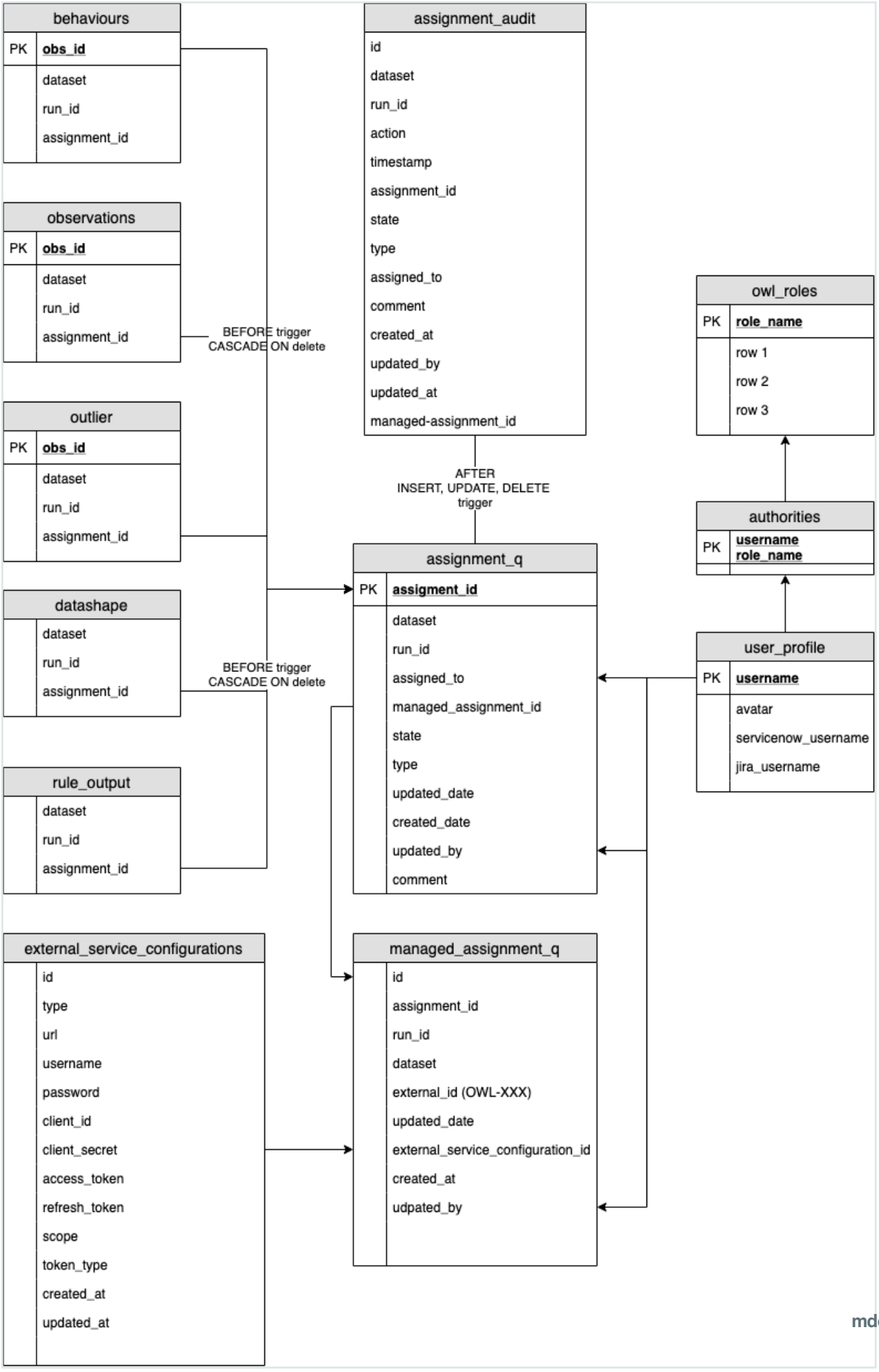
Collibra DQ Kubernetes Deployment Diagram



Note For Kubernetes deployments of Collibra DQ should use Auto Scaling and Spot instances to further increase efficiency and reduce cost.

ERD

Please note there are over 110 tables in the underlying Postgres database and many are for application settings.


behaviours
PK obs_id
dataset
run_id
assignment_id
observations
PK obs_id
dataset
run_id
assignment_id
outlier
PK obs_id
dataset
run_id
assignment_id
datashape
dataset
run_id
assignment_id
rule_output
dataset
run_id
assignment_id
BEFORE trigger
CASCADE ON delete
BEFORE trigger
CASCADE ON delete
assignment_audit
id
dataset
run_id
action
timestamp
assignment_id
state
type
assigned_to
comment
created_at
updated_by
updated_at
managed-assignment_id
AFTER
INSERT, UPDATE, DELETE
trigger
assignment_q
PK assigment_id
dataset
run_id
assigned_to
managed_assignment_id
state
type
updated_date
created_date
updated_by
comment
owl_roles
PK role_name
row 1
row 2
row 3
authorities
PK username
role_name
user_profile
PK username
avatar
servicenow_username
jira_username
external_service_configurations
id
type
url
username
password
client_id
client_secret
access_token
refresh_token
scope
token_type
created_at
updated_at
managed_assignment_q
id
assignment_id
run_id
dataset
external_id (OWL-XXX)
updated_date
external_service_configuration_id
created_at
udpated_by

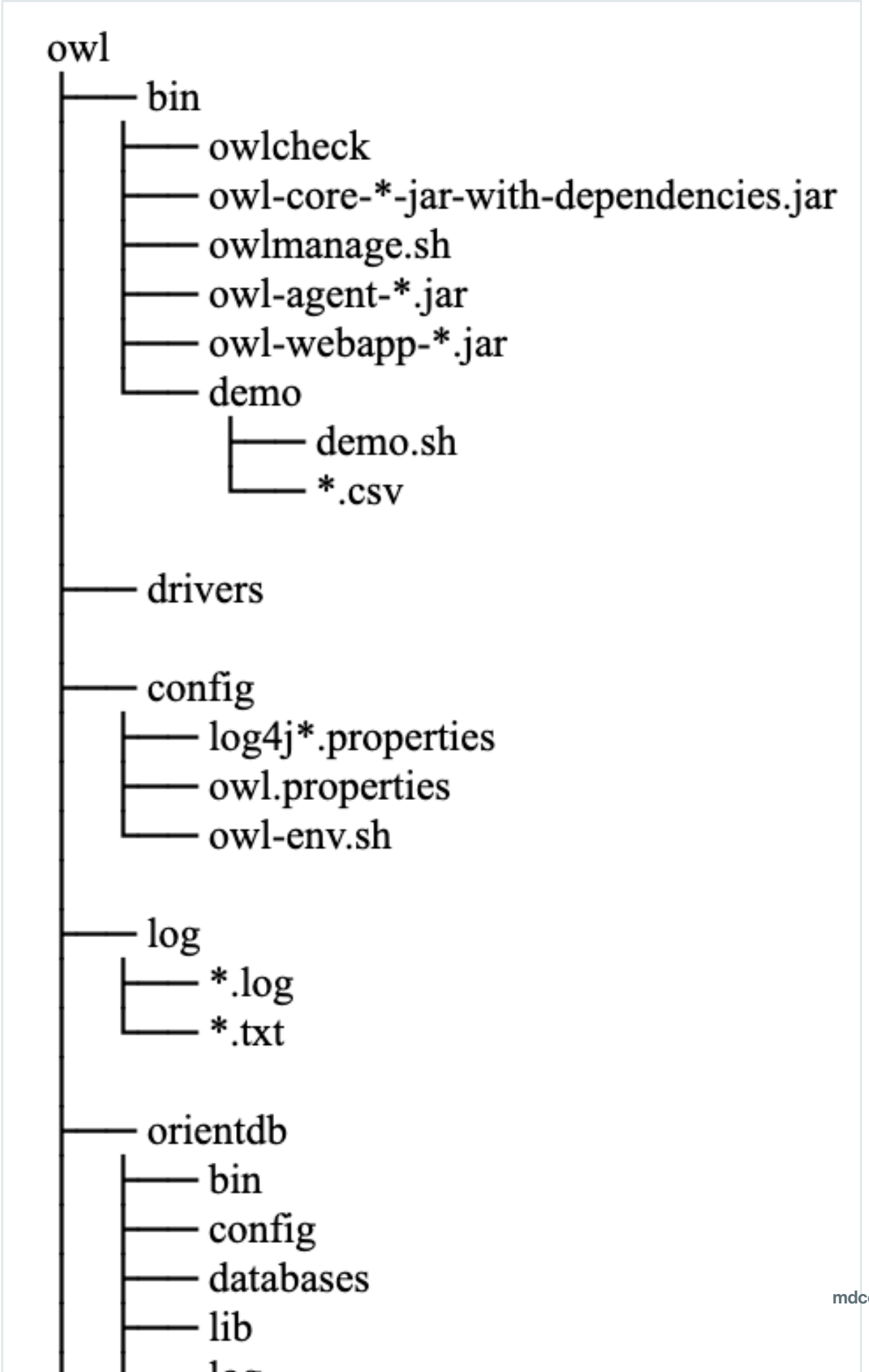
Build Versions

Generally available build versions

- Spark 3.5.3 (default as of 2025.02)
- Spark 3.4.1
- Spark 3.2.2
- Spark 3.1.2
- Spark 3.0.1
- Spark 2.3.0
- HDP 3
- CDH5
- CDH6
- CDH6-NOLOG
- K8s

Data Quality & Observability Classic Directory Structure

The Collibra DQ installation process creates the following directory structure:


owl
bin
owlcheck
owl-core-*-jar-with-dependencies.jar
owlmanage.sh
owl-agent-*.jar
owl-webapp-*.jar
demo
demo.sh
*.csv
drivers
config
log4j*.properties
owl.properties
owl-env.sh
log
*.log
*.txt
orientdb
bin
config
databases
lib

Documentation Archive

> **Important**
> This documentation has been archived and is unlikely to be updated. Some features in this section are no longer supported in the Collibra DQ application.

Cloud

> **Warning** We have decommissioned DQ Cloud.

Collibra DQ Cloud is delivered as a Software-as-a-Service offering. This section describes the installation process and details the Edge component for Collibra DQ Cloud.

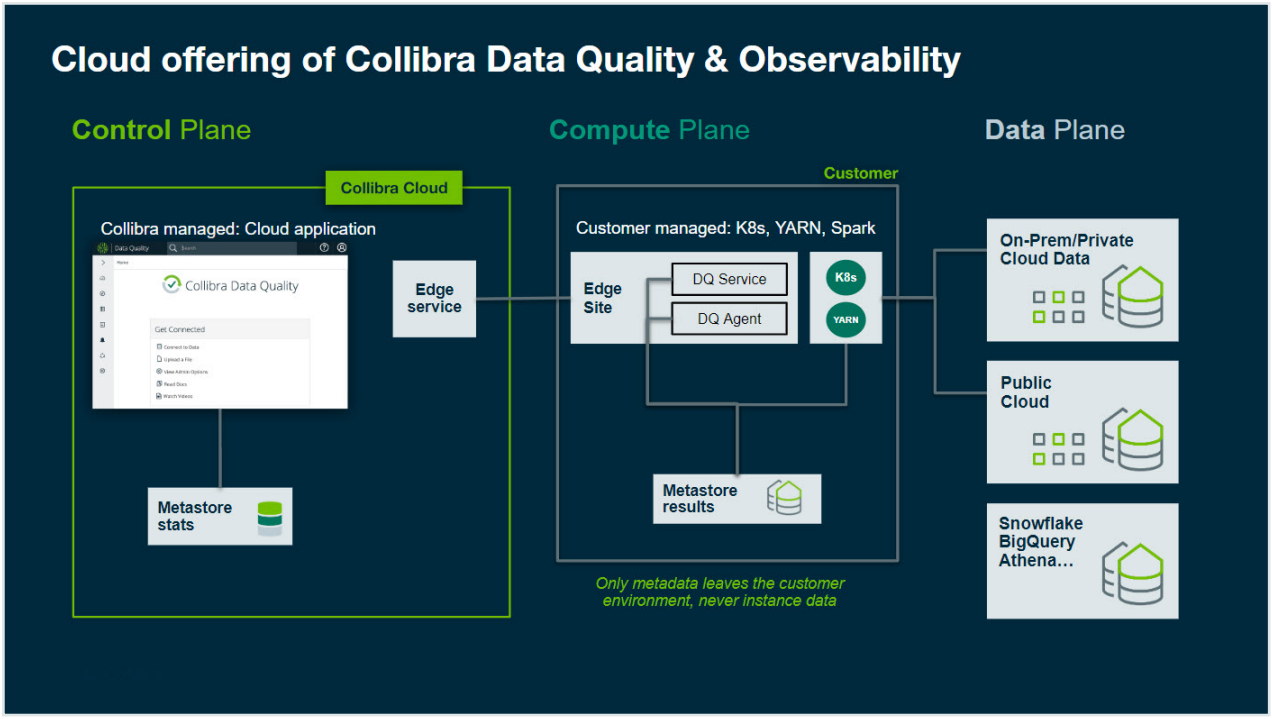
Requirements

To use Collibra DQ Cloud, you must ensure that the following system requirements are met.

Resource	Notes	Provisioned by
Collibra DQ	Version 2022.02 or newer with Edge mode enabled.	Collibra
Collibra Edge site	Version 2022.02 or newer.	Customer
Postgres	Version 11 or newer.	Customer

Diagram

The following diagram shows the role of Edge and how it connects to Collibra DQ. Edge connects to your data sources and then provides data from them back to Collibra DQ, where you can scan and profile your data for quality issues.



Prerequisites

You need the following minimum hardware and software requirements:

VM

This is where your Edge site is installed. You need:

- RedHat 8 or CentOS 8
- SSH access
- 55 GB of free storage
  Verification command...
  Run the following command to verify:

```
df -B G
```

- 64 GB memory
  Verification command...
  Run the following command to verify:

```
free -m
```

- 16 cores
  Verification command...
  Run the following command to verify:

```
cat /proc/cpuinfo
```

- Egress (outbound) network access on port 443
  Verification command...
  Run the following command to verify:

```
wget https://google.com
```

- Network access to Postgres installed in step 2
  Verification command...
  Run the following command to verify:

```
telnet 10.64.2.3 5432
```

> **Note** For medium to large workloads of more than 100M rows by 100 columns, we recommend that your VM has a minimum of 32 cores, 128 GB memory, and 500 GB of free storage.

Edge installation requirements can be found here.

Postgres

This is where your DQ Job results are stored. You need:

- Version 11 or newer
- A minimum of 100 GB of free storage
- A minimum of 4 cores

- Network access to and from the VM where Edge is installed
- User with ownership rights over the target database

Steps

1. Obtain a Secure Collibra DQ Web URL

This is provisioned by Collibra. Along with the URL, credentials will be provided to access your instance.

2. Install Postgres

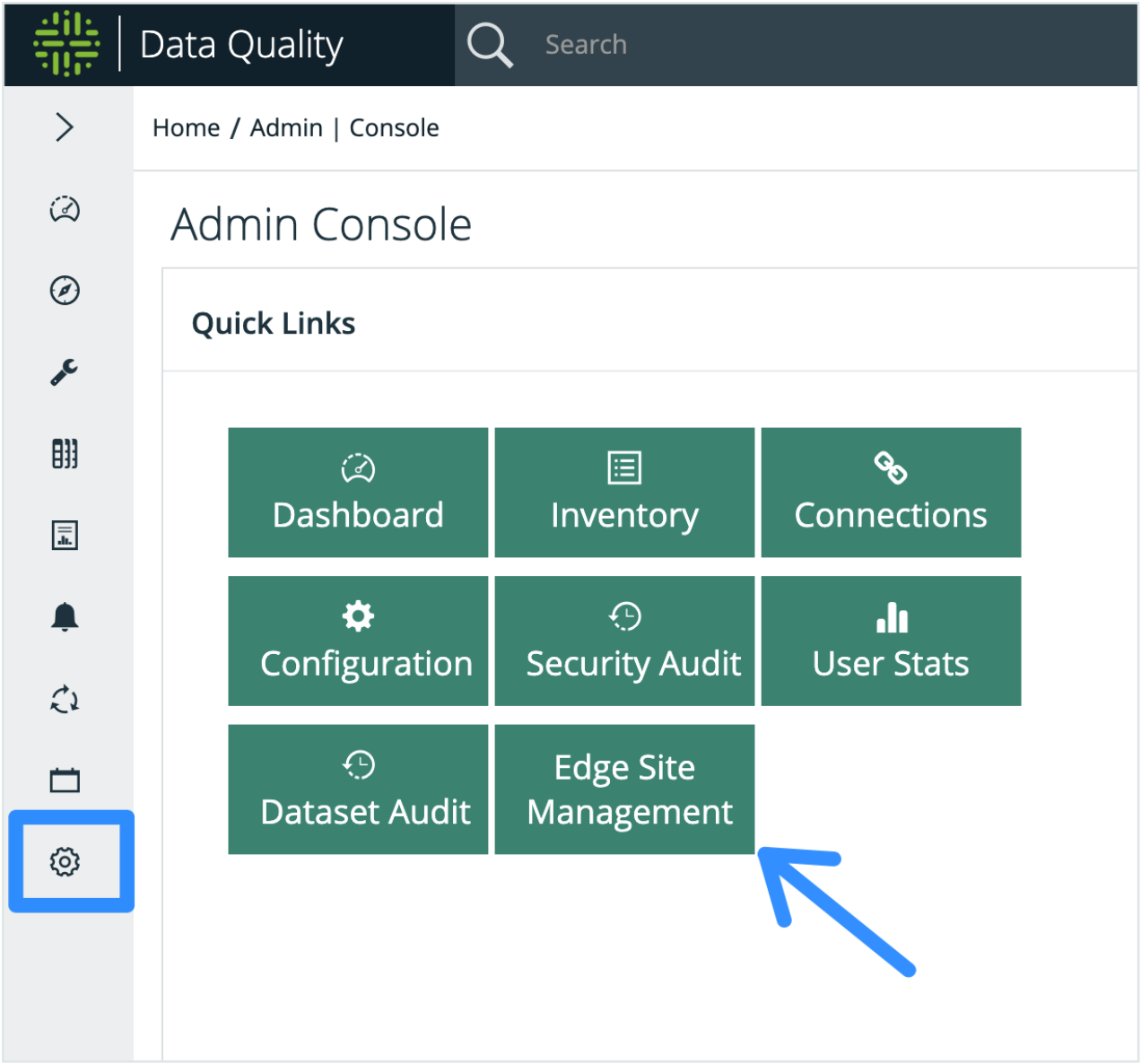
This is provisioned by the customer. There are several ways to install Postgres. You should follow your existing company process to provision a Postgres instance, such as RDS, Azure SQL, Cloud SQL, or standard install with a package manager. Ensure your version is 11 or newer.

> **Important** Remember your Postgres IP and login credentials. This is required when deploying the Edge site.

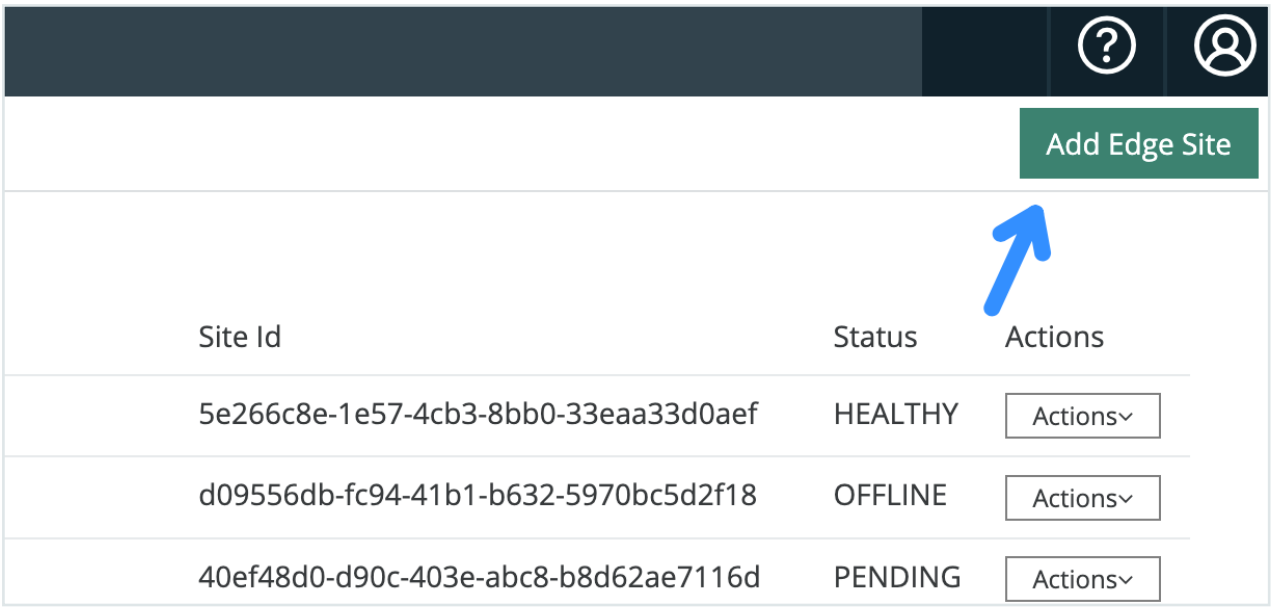
3. Install Edge

Refer to Edge documentation for system requirements.

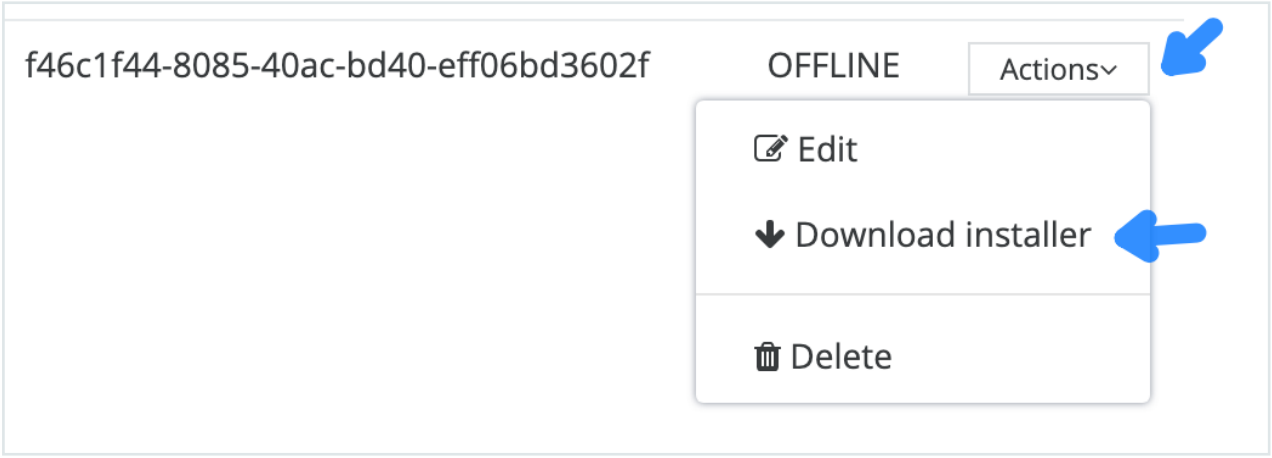
Navigate to the Edge site Management panel in the Admin Console



Add an Edge site and provide a name and description



Using the Actions drop-down, download the Edge installer package locally



> **Warning** Because connections must have an exact relationship between the Edge site and the datasource hostname, do not delete your Edge site from the Edge site Management page.

Upload the Edge installer package to your VM that meets the Edge system requirements above. An example scp command is below, but you can do this several ways.

```
scp -i ~/Downloads/vm-key.pem ~/Downloads/<installer>.tgz user-
@<host-or-ip>:/home/user/<installer>.tgz
```

SSH to your VM after uploading the installer package. Untar the .tgz

```
tar -xvf <installer>.tgz
```

Install prerequisite Edge packages.

```
sudo yum install -y container-selinux selinux-policy-base

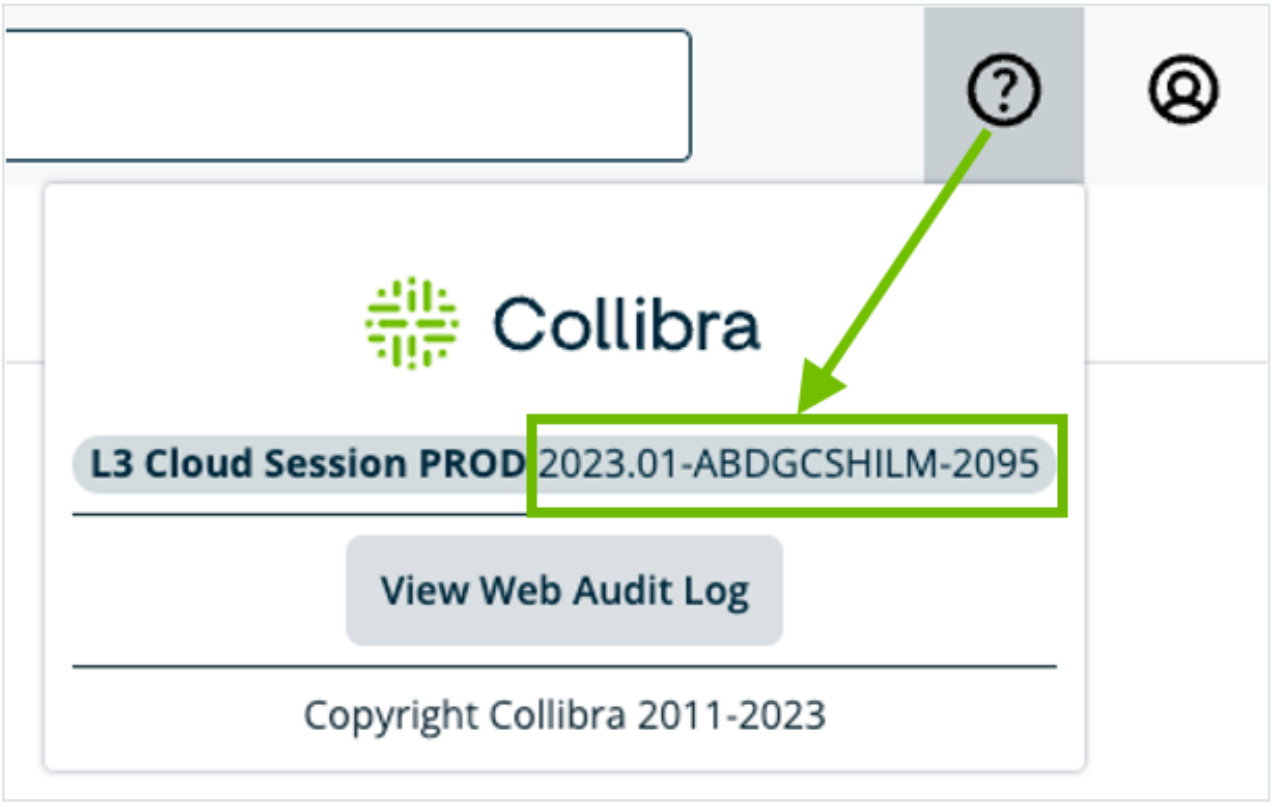
sudo yum install -y sudo yum install -y https://rp-
m.rancher.io/k3s/stable/common/centos/8/noarch/k3s-selinux-
1.2.2.el8.noarch.rpm

sudo firewall-cmd --zone=trusted --add-interface=lo --permanent

sudo firewall-cmd --zone=trusted --add-interface=cni0 --per-
manent

sudo firewall-cmd --reload
```

Confirm you have the correct Collibra DQ version pointer from your Cloud instance, for example **2023.01-ABDGCSHILM-2095**.



Remember your Postgres IP and credentials from the previous step.

Install Edge w/ DQ w/ the correct parameters

```
sudo /home/centos/install-master.sh --storage-path /var/edge
properties.yaml -r registries.yaml --set collibra_edge.-
collibra.dq.enabled=true,collibra_edge.-
collibra.dq.targetRevision=2023.01-ABDGCSHILM-2095,collibra_
edge.collibra.dq.sparkVersion=3.4.1,collibra_edge.-
collibra.dq.metastoreUrl=jdbc:postgresql://<your-postgres-
ip>:5432/postgres,collibra_edge.collibra.dq.metastoreUser=<your-
postgres-user>,collibra_edge.collibra.dq.metastorePass=<your-
postgres-password>
```

The snippet below is the same as the code block above.

The bold sections are the areas you will edit

sudo /home/**<your-directory>**/install-master.sh --storage-path /var/edge properties.yaml -r registries.yaml --set collibra_edge.collibra.dq.enabled=true,collibra_edge.collibra.dq.targetRevision=**<dq-version>**,collibra_edge.collibra.dq.sparkVersion=**<spark-version>**,collibra_edge.collibra.dq.metastoreUrl=jdbc:postgresql://**<postgres-ip>**:**<postgres-port>**/**<postgres_database_name>**,collibra_edge.collibra.dq.metastoreUser=**<postgres-user>**,collibra_edge.collibra.dq.metastorePass=**<postgres-password>**

> **Warning** The $ symbol is not a supported special character in your Postgres Metastore password.

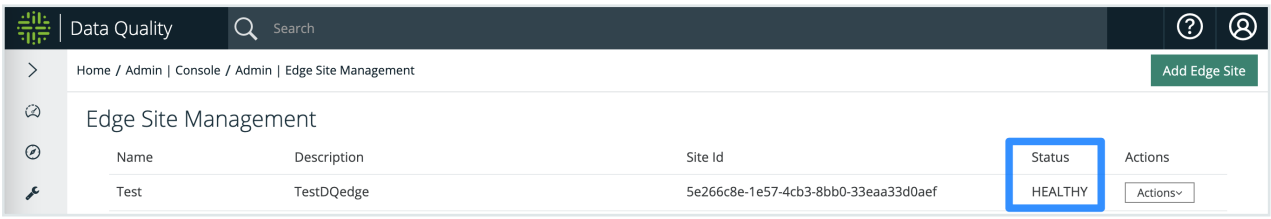
> **Note** Because Collibra DQ automatically derives its version from the Collibra Platform version, you do not need to specify the version of Collibra DQ for successful automatic upgrades.

Check that all the processes are running / completed

```
sudo /usr/local/bin/kubectl get pods --all-namespaces
```

```
[centos@test-matt-mearns-edge ~]$ sudo /usr/local/bin/kubectl get pods --all-namespaces
NAMESPACE             NAME                                                READY   STATUS      RESTARTS   AGE
kube-system           local-path-provisioner-5ff76fc89d-b8vhj             1/1     Running     0          48m
kube-system           coredns-6488c6fcc6-vrwsd                            1/1     Running     0          48m
kube-system           metrics-server-86cbb8457f-z25lc                     1/1     Running     0          48m
collibra-edge         argo-cd-argocd-redis-648c968585-hzhlk               1/1     Running     0          47m
collibra-edge         argo-cd-argocd-application-controller-656bc485c9-w8jx6   1/1     Running     0          47m
collibra-edge         argo-cd-argocd-repo-server-5f796d76f4-ftrmx          1/1     Running     0          47m
collibra-edge         argo-cd-argocd-server-6bd4b4c7f7-8h62m               1/1     Running     0          47m
collibra-edge         collector-metrics-5989b8fc69-vgtm8                  1/1     Running     0          46m
collibra-edge         eventrouter-edge-eventrouter-58cf94d7f6-8c8gh       1/1     Running     0          46m
collibra-edge         metricscrape-fb6b6499c-jj2bm                        1/1     Running     0          46m
collibra-edge         agent-vkzd9                                         1/1     Running     0          46m
collibra-edge         signalfx-edge-signalfx-kvtcc                        1/1     Running     0          46m
collibra-edge         workflows-workflow-controller-6c9886bcdf-f7rv7      1/1     Running     0          45m
collibra-edge         minio-3                                             1/1     Running     0          45m
collibra-edge         minio-2                                             1/1     Running     0          45m
collibra-edge         minio-0                                             1/1     Running     0          45m
collibra-edge         minio-1                                             1/1     Running     0          45m
collibra-edge         workflows-server-5fcdbd6455-s59pl                   1/1     Running     0          45m
collibra-edge         collibra-edge-proxy-846dd4bdc5-9ssqj                1/1     Running     0          46m
collibra-edge         edge-agent-964pn                                    1/1     Running     0          45m
collibra-edge         owldq-owl-agent-owldq-96beafd3-0                    1/1     Running     0          45m
collibra-edge         datadog-54r7v                                       1/1     Running     0          45m
collibra-edge         owldq-owl-web-owldq-96beafd3-0                      1/1     Running     0          45m
collibra-edge         collibra-edge-controller-675d797cc8-rb458           1/1     Running     1          46m
edge-kube-installer   edge-kube-installer-job-2xgh5-fqvcc                 0/1     Completed   0          48m
```

Your Edge site will appear as HEALTHY upon successful installation



Uninstall Edge if there were mistakes or typos in the process

```
sudo /usr/local/bin/uninstall-edge.sh --force
```

> **Note** Do not delete an Edge site from the UI. You can safely uninstall an Edge site with this command.

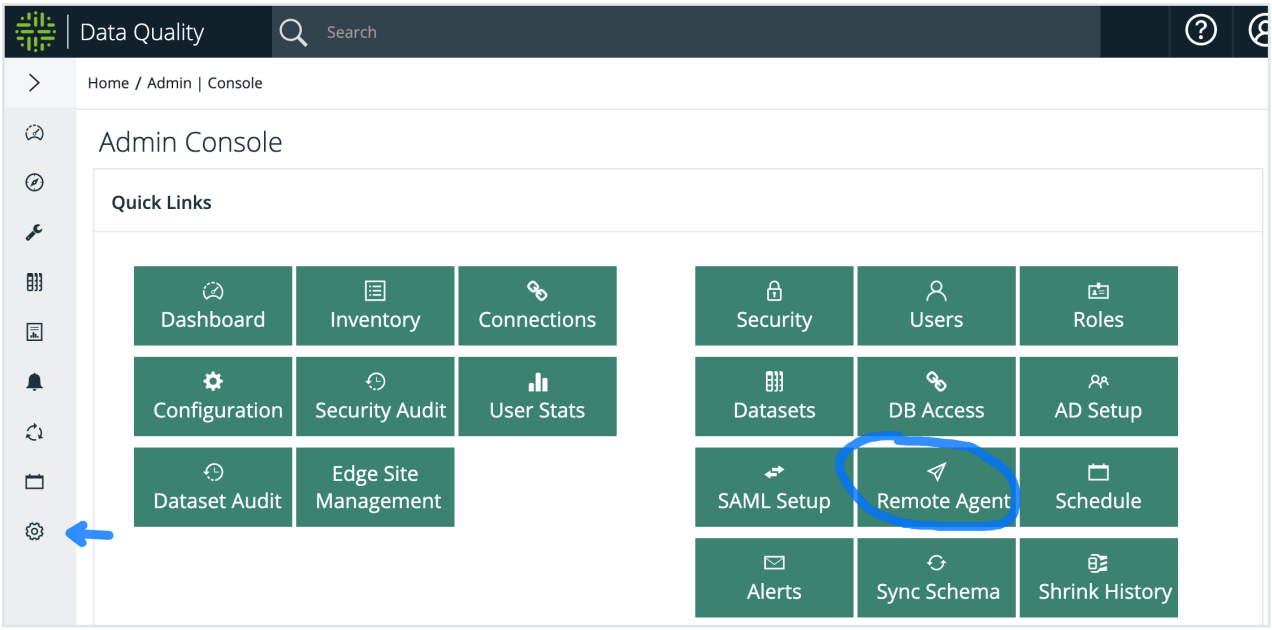
Reinstall the prerequisites if you perform the uninstall

```
sudo yum localinstall --skip-broken -y https://rp-
m.rancher.io/k3s/stable/common/centos/8/noarch/k3s-selinux-
1.2.2.el8.noarch.rpm
```

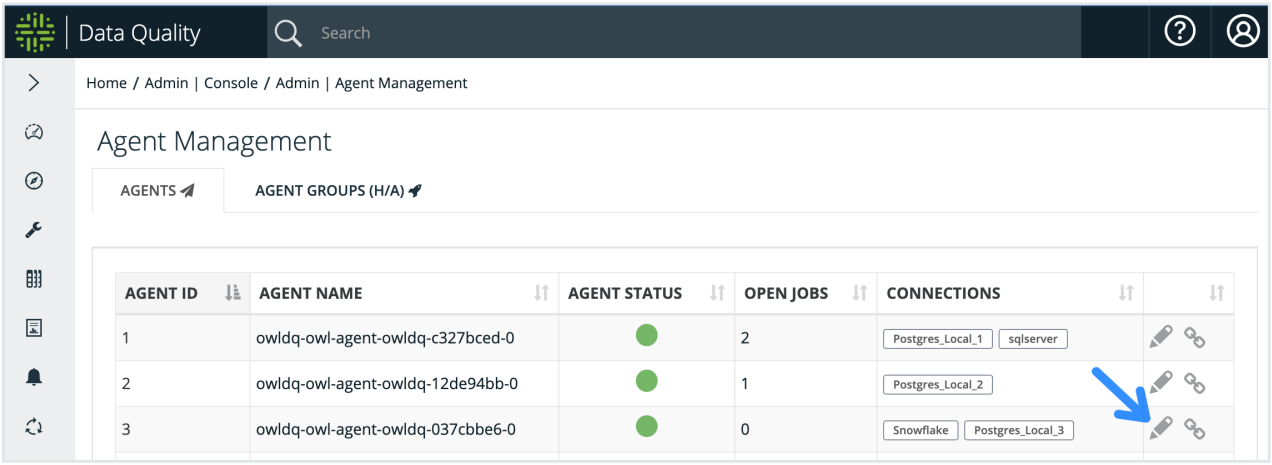
> **Warning** To avoid orphaned records, do not delete an Edge using the UI.

4. Configure an Agent

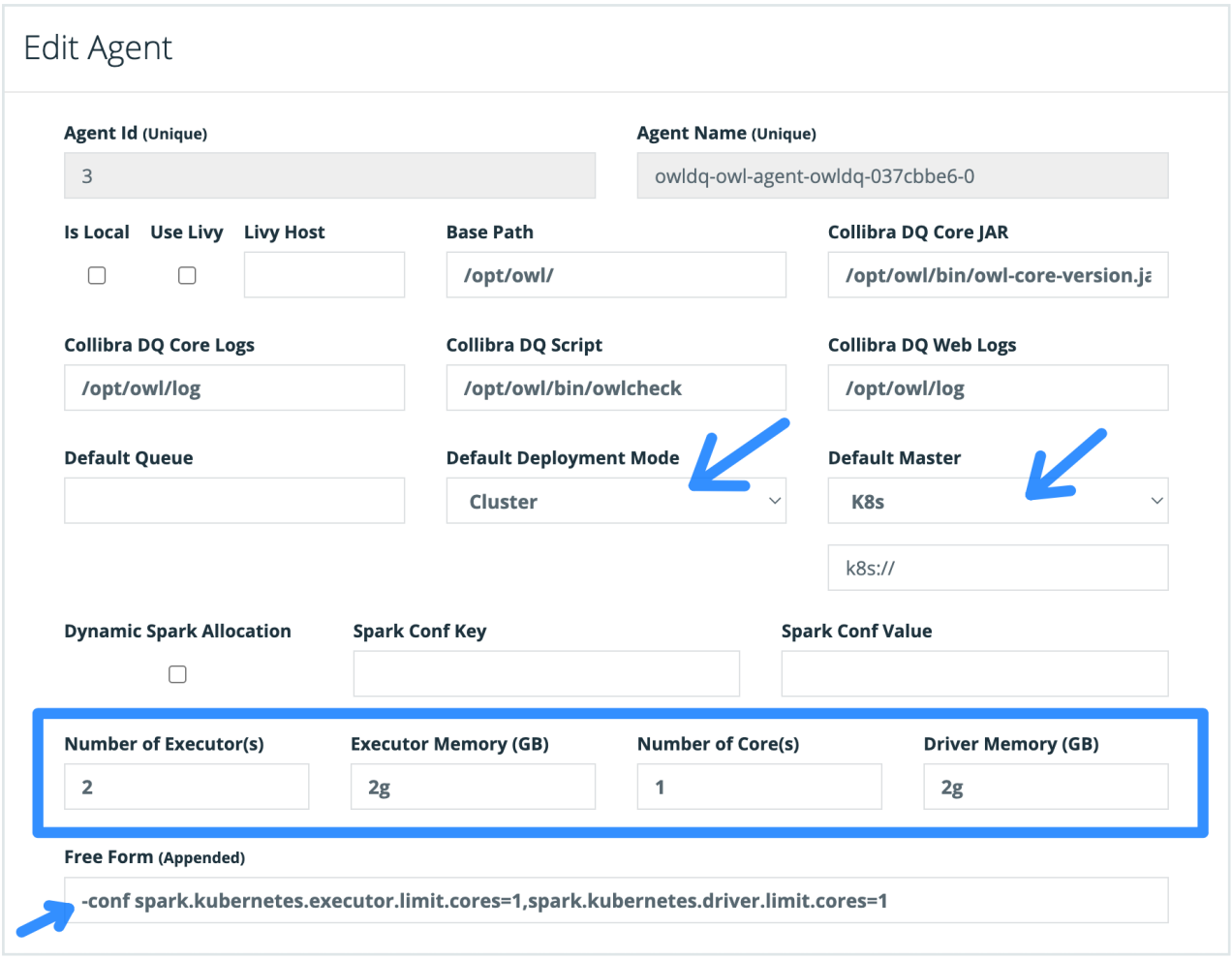
Go to the Remote Agent panel in the admin console



Upon completion of the Edge installation, you'll find an agent available from each respective Edge site. Click the pencil icon to configure the agent.



Change the Default Deploy Mode to Cluster, the Default Masters to K8s and input defaults for resource assignment. Also add freeform append Spark confs as shown here.



Use the spark confs in the code block below.

```
-conf spark.kuber-
netes.ex-
ecutor.limit.cores=1,spark.kubernetes.driver.limit.cores=1
```

Note The DQ Job (Spark) compute will take place locally on Edge K3s. Increase the size of your VM to vertically scale for more resources (.e.g. 32 cores, RAM, etc.). This is the preferred option in preview. Hadoop compute is supported if customer chooses that path and uses their Dataproc or EMR cluster.

Note Make note of the agent name that as created. In the following step you will create a connection and select (link) the agent to your connection.

> **Warning** Do not delete an Agent from the UI, to avoid any orphaned records.

5. Set Job Limits

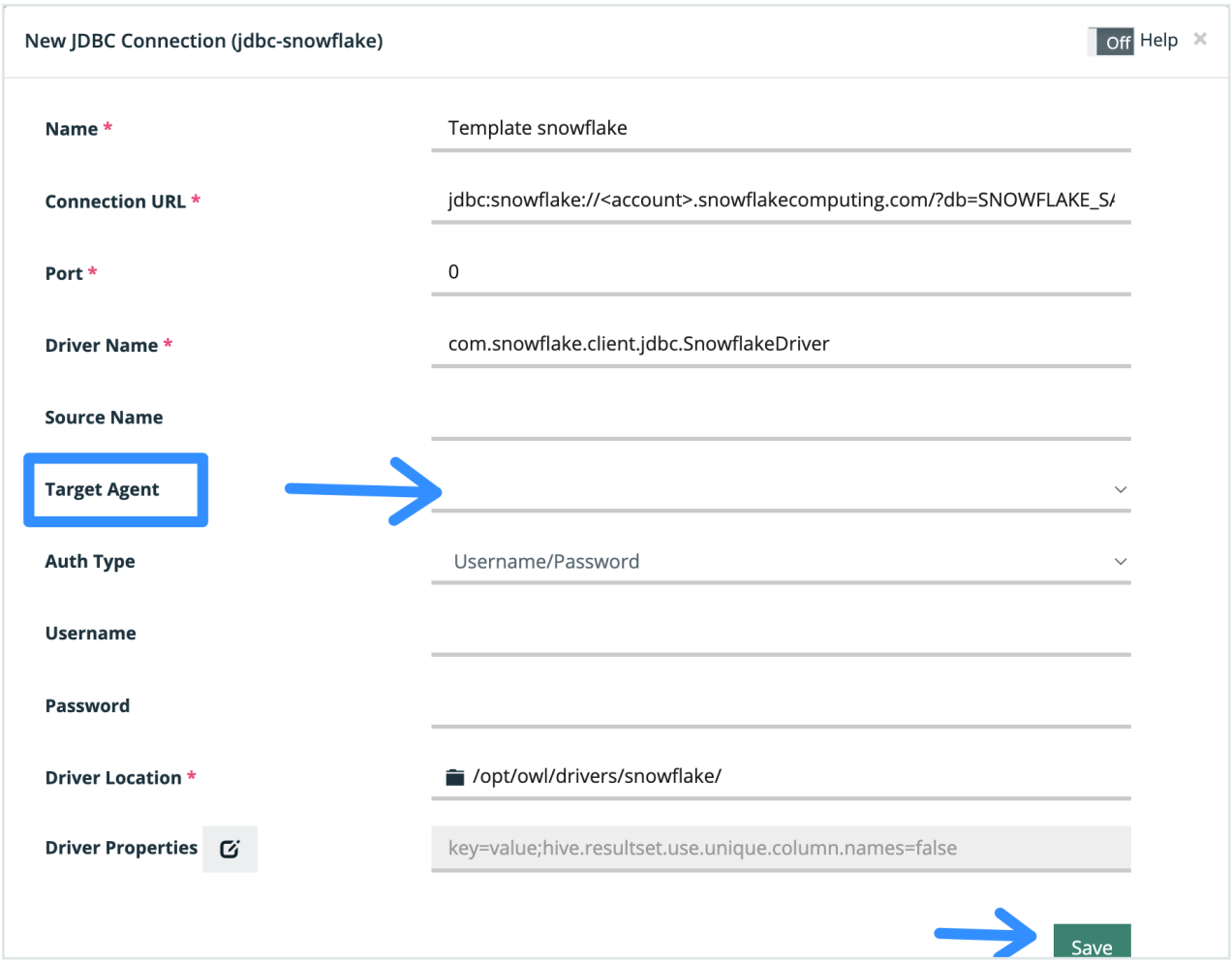
Set max cores to 1 in the job limit settings.

Refer to this link for configuring job limits.

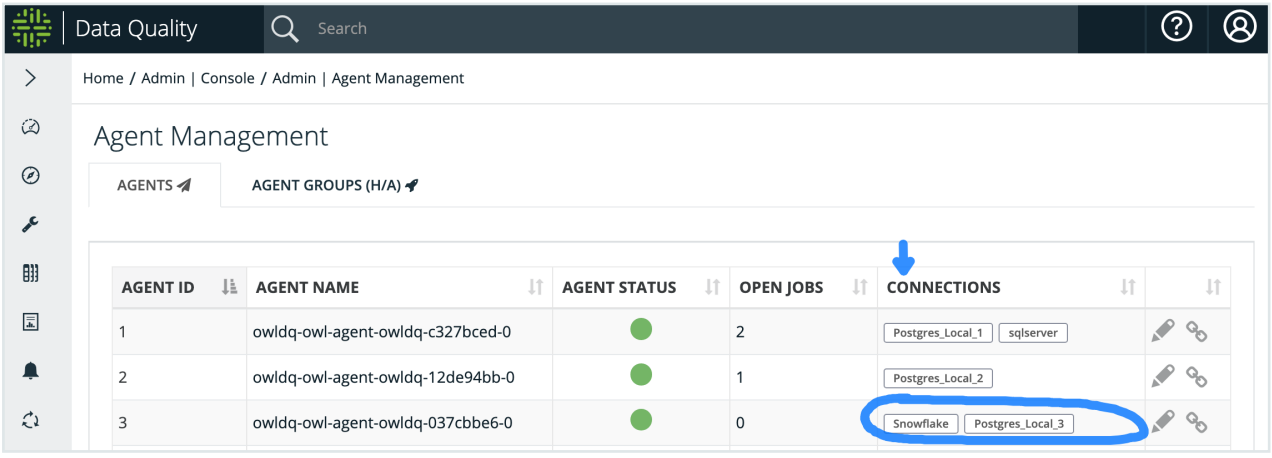
6. Add a Connection

This is the same process of adding a connection found here with one difference. You will map the connection to your agent upon establishing a connection. **This is different than mapping a connection and an agent in the self-hosted application.**

Select your target agent using the **Target Agent drop-down**. This drop-down will populate with existing agents. Here is where you will select the agent name from the previous step.



Afterward, you do not need to assign the connection to the agent. It will be automatically mapped.



> **Note** To map a connection to another agent, you need to re-save the connection and select another agent from the drop-down list.

7. Run a DQ Job

Run a DQ Job to validate the installation. Use the Explorer to onboard a table and check the Jobs page as normal to see the status.

> **Note** If the DQ Job does not succeed, please check your Agent settings and system prerequisites

Notes

Edge Capability Resource Requirements: If insufficient resources, your capabilities will not perform properly.

Installer: Please beware, downloading new installer will invalidate previous installer.

Volume: /var/lib/rancher/k3s path must have 50gb available

Root access: root access is needed, though future revisions will follow the least privileged user access policies.

The private preview is designed to let customers 1) complete the installation 2) confirm successful DQ jobs can be run and 3) validate their security requirements whereby no sensitive data is stored outside their custody.

Helpful Commands

```
# Get all pods running
sudo /usr/local/bin/kubectl get pods --all-namespaces

# Get shell access to pod
sudo /usr/local/bin/kubectl exec -it <dq-web-pod> -n collibra-
edge -- bash

# Get shell access to pod
sudo /usr/local/bin/kubectl exec -it collibra-edge-controller-
<pod-name> -n collibra-edge -- sh

# Check network connectivity to database
curl telnet://<rds-host>:<port>

# Delete jobs
sudo /usr/local/bin/kubectl delete pod <pod-name> -n collibra-
edge
```

Known limitations

- DQ Cloud does not currently support SAML configuration.
- When reviewing the Completeness Report, new data only displays correctly after you upgrade your Collibra DQ Cloud instance to version 2023.05.2 or later.
- If an Edge site needs to be reinstalled, you must use the original PostgreSQL metastore database or metastore corruption may occur in Collibra DQ. If necessary, restore the metastore database from a backup before reinstalling. Ensure that the installation command line parameter `collibra_edge.collibra.dq.metastoreUrl` points to the correct database.
- Instance Profile is not supported for S3 connections.
- When using the Completeness Report, data only appears after upgrading to 2023.06 or later.
- When using the Findings page, you currently cannot drill into a rule break record. While there is no workaround for this limitation, a fix is planned for the 2023.06 release.
- When using the Findings page, you currently cannot tag job runs as off-peak. This will be fixed in the 2023.07 release.

FAQ

What network access is needed?

- The Edge site and Postgres need to communicate with each other.
- Additionally, logging and heartbeat requires outbound access to several services. Please refer to Edge documentation for specific services that are used.

How can a user check the install?

- Time: The install should complete in around ~5 minutes; if not, there is likely an issue.
- Check that the pods
- `sudo /usr/local/bin/kubectl get pods --all-namespaces`

Is there a way to get more checks / more logs?

- `sudo /usr/local/bin kubectl describe`

How to verify successful install?

- In your Collibra DQ instance, navigate to the Edge site Management panel in the Admin Console and confirm a HEALTHY status
- Support can confirm via Datadog, the Edge site will send heartbeats

How to locate my Edge site in Datdog?

- Send your Edge site ID to Support to check the health status.

Do customers have access to Datadog?

- Only Collibra has access to Datadog logging.

Can all my Collibra DQ and other capabilities run on the same Edge site?

- There are not technical reasons preventing other capabilities and Collibra DQ from running on the same Edge site.
- The guidance for the preview is to have DQ Edge separate from DGC Edge capabilities and simply use two Edge sites.

Are there any limitations with Collibra DQ Cloud in terms of features or functionality?

- While remote files are supported, local files and uploaded files are not supported due to security restrictions

- Specific drivers are not available in the preview, though the most common data sources are available.

What are the benefits of installing with Edge vs. a stand-alone, self-hosted application?

- The primary benefits are managed upgrades, maintenance, and reducing the ownership costs of an entirely self-hosted set of components.
- In addition, this design allows customers to take advantage of containers and cloud technologies without deep technical skillset requirements.
- This installation pattern was intentionally developed to not compromise any security requirements and give the customer complete custody of their data.
- Lastly, this aligns the Collibra architecture standards so support and services teams will benefit from normalized deployment models. In particular, when it comes to installation, configuration, and troubleshooting.

Troubleshooting

When upgrades of DQ Edge sites are required, you can leverage a utility script to update the Edge DQ version without reinstalling the Edge site.

> **Important**
> You must run the override-dq.sh script from a machine that is configured to manage Kubernetes where the DQ Edge site is running. For example, the same user and VM where DQ Edge is installed.

Steps

1. Download the override-dq.sh script.
2. Open a terminal session and execute the following shell command:

```
# Give your .sh file execute permission
chmod +x override-dq.sh

# Run the script
./override-dq.sh
```

3. Choose one of the following options:
   a. (Recommended) Remove the setting that allows your Edge site to update according to the DQ Cloud version.

```
# Run the script
 $ ./override-dq.sh

 # Unset the DQ service version override at the <span
class="CollibraProducts.EdgeSite mc-variable">Edge
site</span>
 --unset
```

b. Set the version to a specific Collibra DQ version.
   a. Click ⓘ in the upper right corner of the application, then copy the **App Version**, for example, 2023.08-ABDGCSHILM-3280.
   b. Paste the copied App Version into the version.

```
# Run the script
 $ ./override-dq.sh

 # Update the target DQ version to set at the <span
class="CollibraProducts.EdgeSite mc-variable">Edge
site</span>
 --dq-target-version <20XX.XX-ABDGCSHILM-XXX>
```

> **Tip** You can use either the `--help` or `-h` command to display the following message: `Note Either --dq-target-version or --unset should be set.`

Use Cases

> **Warning** This documentation is archived and is no longer maintained.

Data Projects

> **Warning** This documentation is archived and is no longer maintained.

Builds a Better DQ Dashboard

> **Warning** This documentation is archived and is no longer maintained.

- **DQ Dashboards.** Many DQ problems result from an improper or a too slow observation of business rules related to the data. What is not caught by handmade visual inspection or a potentially outdated man-made rule can only be flagged by AI Machine Learning. Conversely, what does get flagged should also be easily triaged and then immediately fixed with the aid of AI. The most important metric for a DQ Dashboard is the time to fix, not simply the overall DQ score.

Ensures CCPA & GDPR

> **Warning** This documentation is archived and is no longer maintained.

Aggregation from hundreds of locations: The dashboard for what is within spec based on AI observation – not handmade rules.

Then push-down fix: The rules are created and then immediately applied via self-service. The problem is immediately identified and a fix (recommendation engine) is applied.

The value of this for both companies: The DQ problem never corrupts the whole. The longer the bad quality exists the bigger problem it can create.

Collibra DQ improves your data lake

> **Warning** This documentation is archived and is no longer maintained.

Data lakes support analytics, which will ultimately drive actions that increase revenue, support compliance, prevent churn, etc. However, whether that action is near to real-time or not, none of those can be performed without first performing a DQ check. For example, can you trigger an action before first checking the “GDPR Remove” list? A Data Quality check must always be

the first step in any action. Collibra DQ with Schema Learned can perform 100+ DQ checks. However beyond simply those checks, it is Collibra DQ's unique Spark-based architecture listed below that enables innovation. Churn, credit check, AML, infosec checks developed in the data lake could be added as part of DQ check on the streaming data.

- **Data and Privacy in Place.** Data never has to move for a DQ check. The latency saved from operating in place, the added hybrid flexibility, the privacy maintained serves many new use cases that were not possible before. It also removes any unnecessary consolidation for the sake of simply consolidation. DQ doesn't have to start by first moving it into a data lake.
- **DQ or Any Rules applied in the Stream.** The DQ rules learned by DQ can be applied back to the source on data in the stream. However, other non-DQ rules learned in the data lake can also be added to the DQ check.
- **Self-Service and DQ push-down fix.** DQ can enable a self-service push-down fix (recommendation engine) to anything flagged at the source. The best time to fix DQ is when and where the problem started. This enables tighter integration with Data Governance tools since DQ is maintained at the source once, not downstream where corruption beyond just the data can occur.
- **Multi-cloud/On-prem/Hybrid.** Collibra DQ can scan/alert/report at the source or can operate natively on the target data lake such as Databricks Delta in Azure or Snowflake on AWS, or Qubole on GCP. Why compromise DQ just because your data is not in one place? Why settle on a DQ strategy that only works if the data is first migrated or moved?
- **DQ Dashboards.** Many DQ problems result from an improper or a too slow observation of business rules related to the data. What is not caught by handmade visual inspection or a potentially outdated man-made rule can only be flagged by AI Machine Learning. Conversely, what does get flagged should also be easily triaged and then immediately fixed with the aid of AI. The most important metric for a DQ Dashboard is the time to fix, not simply the overall DQ score.

Speeds Migrations/Enables Replications

> **Warning** This documentation is archived and is no longer maintained.

- **Speed Migrations/Enable Replications.** Batch collection with subsequent excel compare is very problematic. Instead, rules are generated by the data itself, and anomalies are

triggered on the fly at the source system. This type of schema learned approach will speed migration and enable replications with much less overhead.

Collibra DQ assists data aggregation

> **Warning** This documentation is archived and is no longer maintained.

- **Greatly Reduce Support Costs.** Apply AI-generated rules to maintain consistency across all accounts, rather than apply hand-made rules per every account. You are not managing 100+ variations of rules, but a consistent set that is learned from all.
- **Capitalize on Real-time.** Batch collection with excel compare will never support real-time. Instead, rules are generated by the data itself, and anomalies are triggered on the fly at the source system.
- **Rapid Onboarding.** Universal and already tested scans are applied quickly.
- **Improve Customer Satisfaction.** Monitor the real-time speed of the DQ pushdown fix, not just the overall DQ score over time.
- **Improved SLAs.** When DQ is fixed immediately, all SLAs can be improved not just DQ SLAs.

Create a data quality pipeline

> **Warning** This documentation is archived and is no longer maintained.

Organizations that leverage data as an asset to make decisions into their future must entrust the data, from which important business decisions are derived. While almost all businesses leverage their own collected or generated data (or plan to) for internal use, how many actually scrutinize their data? Companies that sell a product must ensure that it is run through some sort of quality assurance suite of tests/runs before it is available to a customer. So organizations that use data internally as their own asset/product should have the same or more confidence in the quality of their data. Collibra DQ is a data quality product that observes the data to surface behaviors, patterns, outliers, schema changes, record changes, and more.

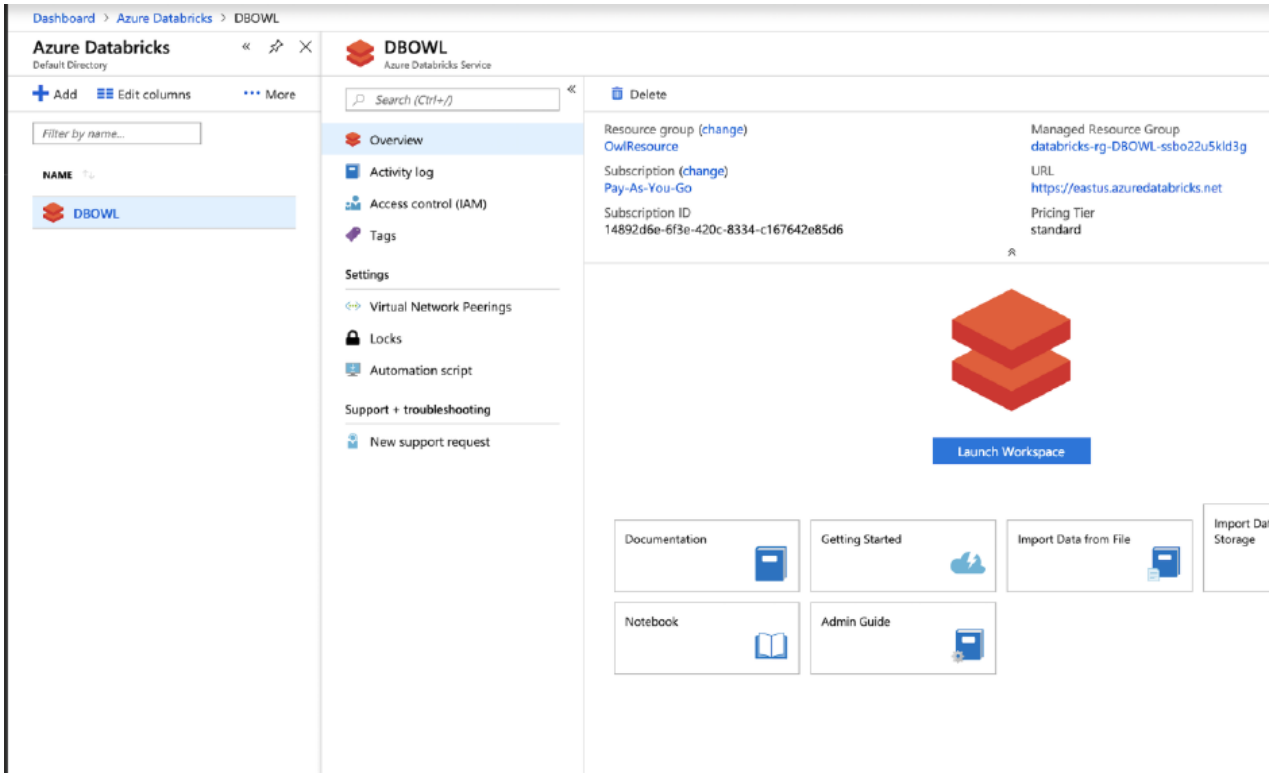
Data Scientists are trying to find insights in data to solve complex problems for their business. However, 80 percent of their time is spent discovering the data to cleanse it to make it ready for the model. Over time, models deteriorate as data underneath changes or new trends/ patterns arise. Leveraging Collibra DQ to validate the quality of the data during the data pipeline and before the data is presented to the Data Scientist reduces time to value for business insights as Data Scientists get time back, not cleaning / prepping the data, and helping the model maintain a longer life.

Azure Databricks allows the ability for Scala code to be written in a Jupyter Notebook against an Azure backed Databricks cluster to scale the work out to more nodes. This is to support the model and the amount of data being crunched by a business's Data Scientists. The simplistic nature of Azure and Databricks and the unification of Spark and Jupyter Notebooks, on top of a robust infrastructure (from storage to compute), allow for Collibra DQ Data Qualified pipelines to be built and executed seamlessly. This reduces the time it usually takes to obtain valuable insights.

Here is how you can build such great DQ pipelines.

Step 1: Build a Databricks Cluster in Azure.

Within the Azure portal find Azure Databricks Service and create a cluster, after the cluster is built you should be able to launch the Workspace as shown below.

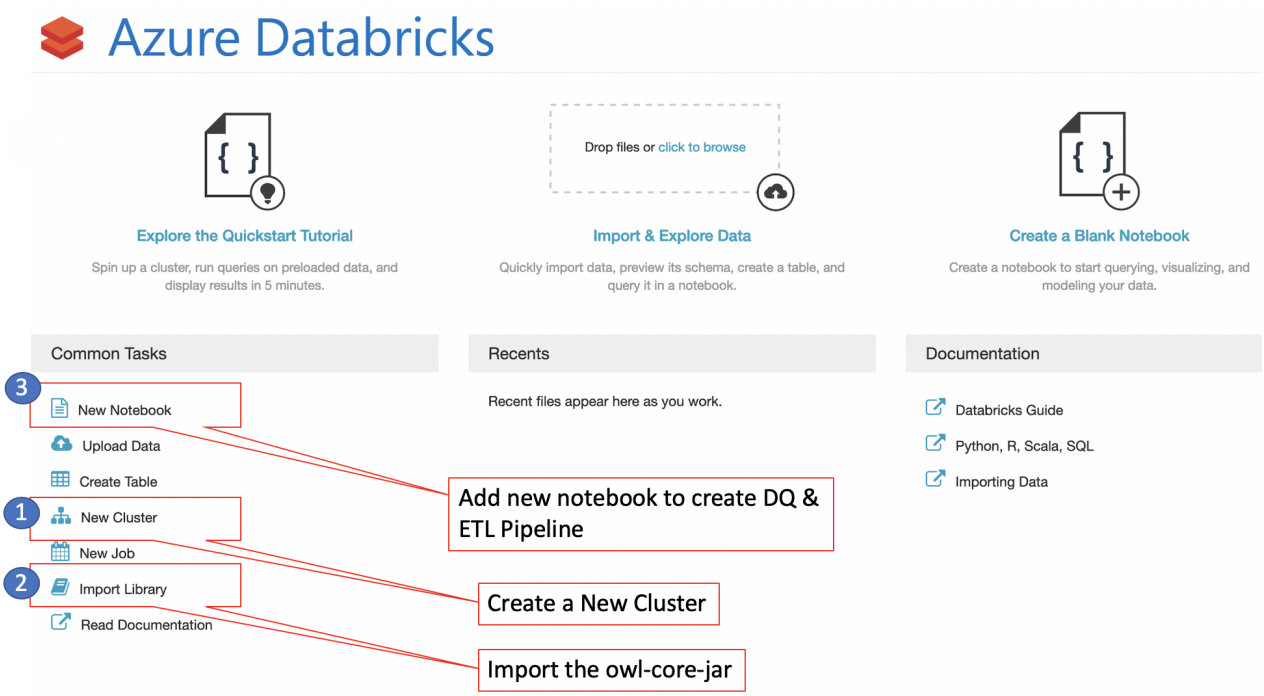


Step 2: Create a Cluster, add the DQ jar and create a Notebook.

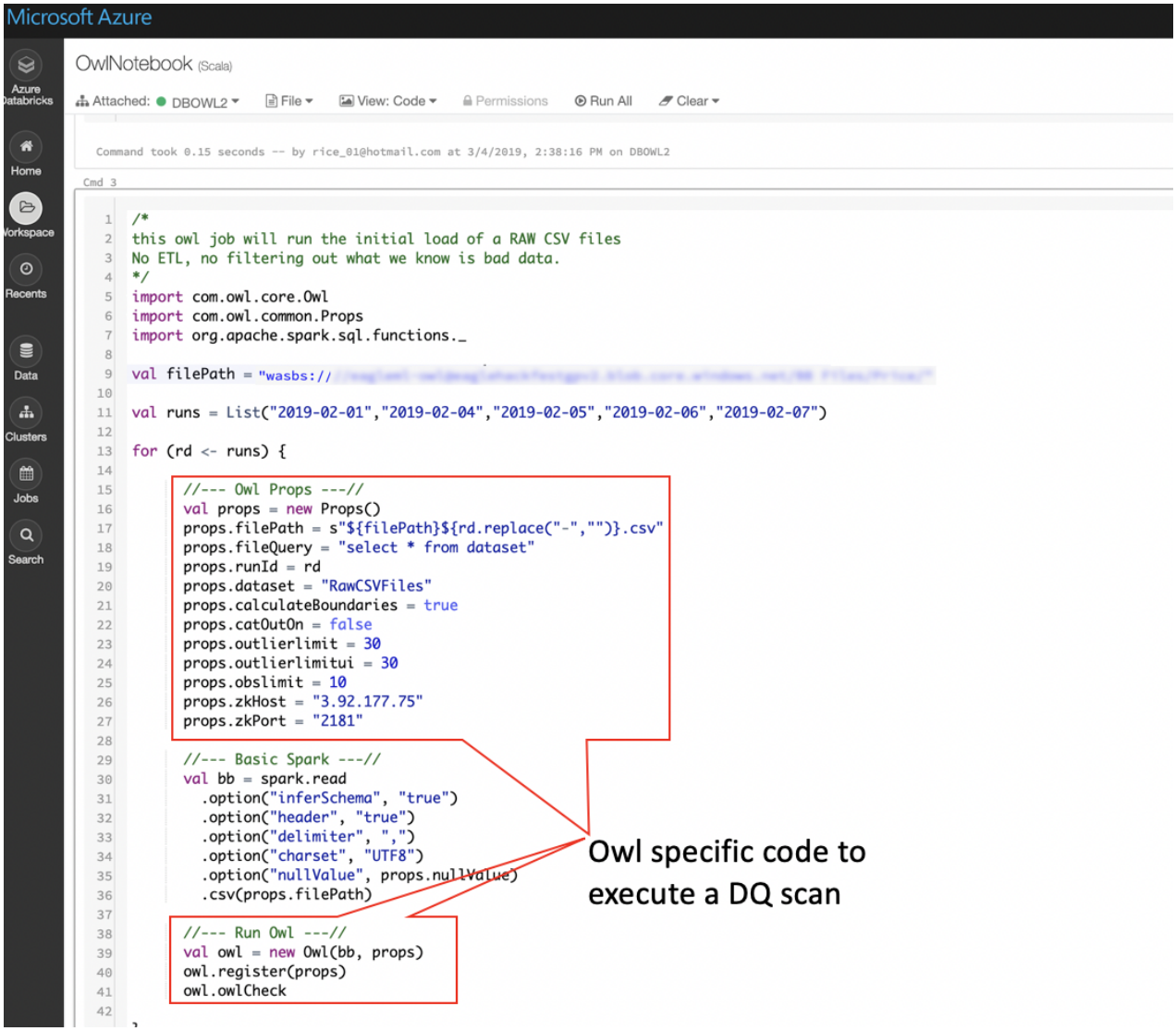
1.) Inside the Azure Databricks UI, create a cluster, provide a cluster name (in this example we will be using DBOWL2 as the cluster name) and select the Databricks Runtime Version that DQ currently supports.

2.) After the cluster is created, make sure to import DQ's jar file onto the cluster so that the Notebook can access the methods exposed in the jar file.

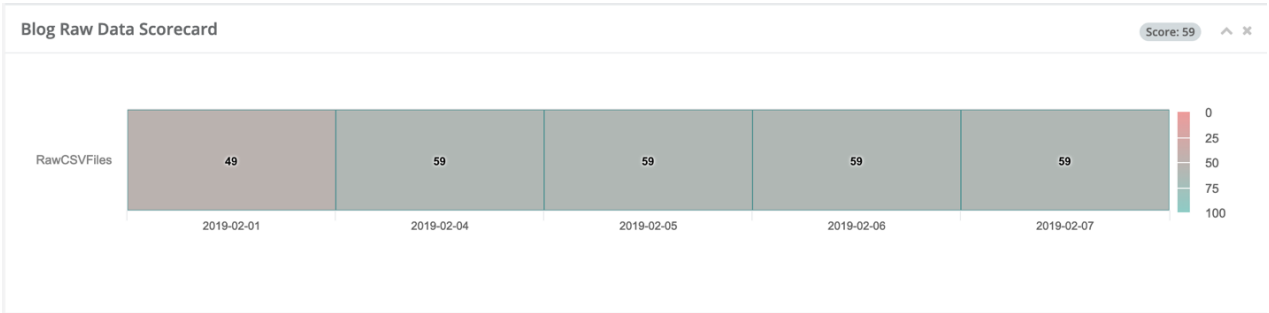
3.) Now that the Jar file has been added, create the Notebook and attach it to the cluster.



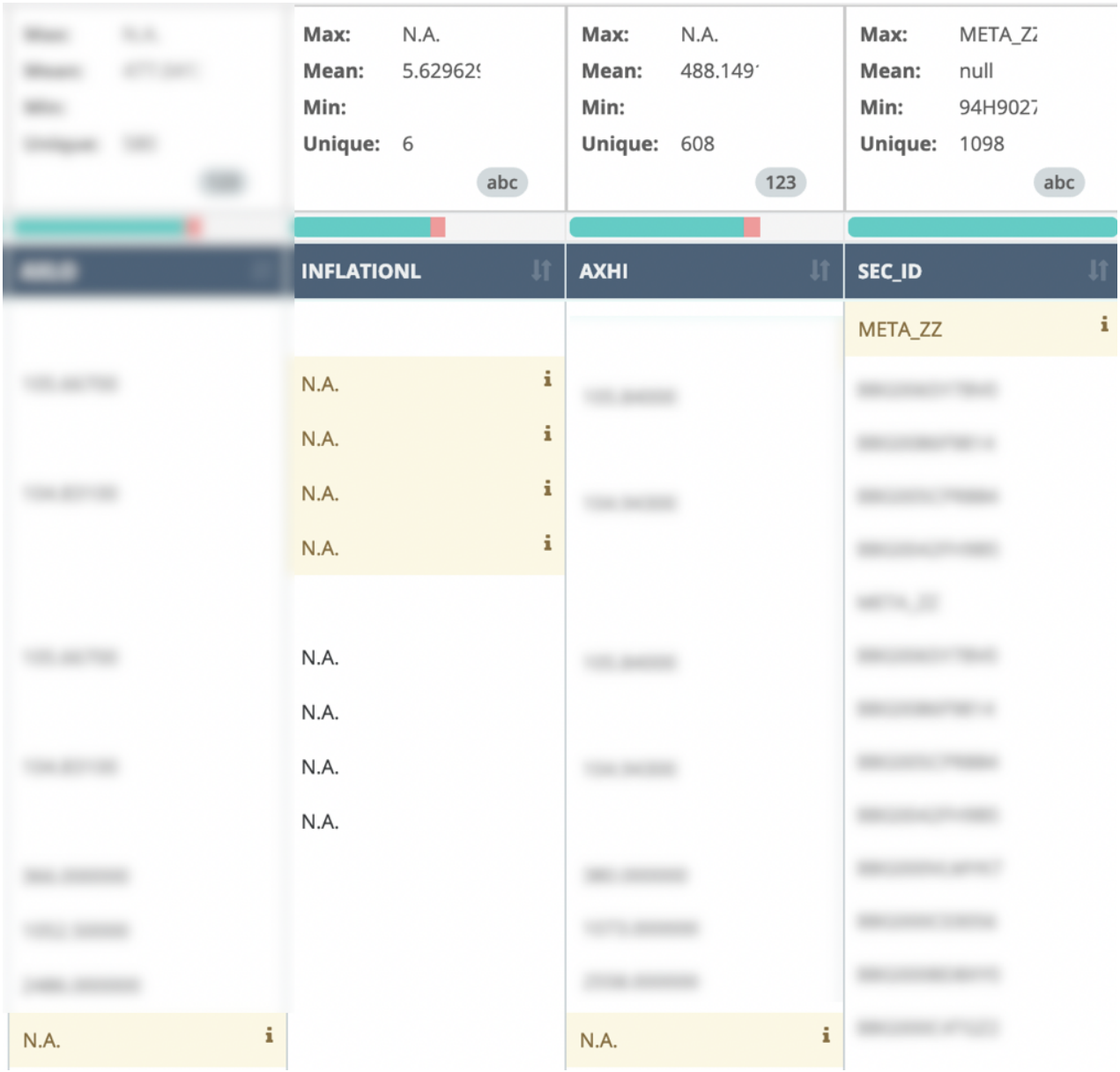
Now the cluster is running with the DQ jar loaded on the cluster. Open the Jupyter notebook attached to the cluster and begin looking at a dataset as a Data Engineer would, prepping the data for use by a Data Scientist by leveraging a DQ Pipeline as shown in the following screenshot.



This Scala code imports the DQ jar and loops through the dates residing in files on Azure blob storage, pulls them into a Spark Data Frame (DF), and executes a DQ job to scan for the quality issues on the Spark DF. Once the scan is completed, the results are stored into the metadata repository under DQ's web application and visible through your browser.



The reason for a score of 49 on the raw data, as shown in the above screenshot, is due to the file having string values sprinkled in the file when something is Not Applicable (N.A.). When reading data in a column of a file that has a mix of numeric and string values the column will automatically conform to a string regardless if the majority class are integers. Also, within the files there is a single record in this file that has meta data information about the file "META_ZZ" this is also adding empty strings for all other columns. This record will also cause all columns to conform to strings.



You should now have an understanding of the raw file and how you need to conform it before analysts can start to glean business value from the contents itself. First, ETL or cleanse the

data that you discovered as being in error by filtering out the erroneous record and flipping all the N.A. values to null as the next step in our ETL and DQ pipeline.

```
Command took 0.15 seconds -- by rice_01@hotmail.com at 3/4/2019, 2:38:16 PM on DBOWL2

Cmd 3

/*
this owl job will use native spark to filter out bad data replacing N.A. with nulls
and filtering out the a column with value META_ZZ...once ETL'd the data will get
stored in a cleaned directory and rerun with Owl
*/
import com.owl.core.Owl
import com.owl.common.Props
import org.apache.spark.sql.functions._

val filePath = "wasbs://

val runs = List("2019-02-01","2019-02-04","2019-02-05","2019-02-06","2019-02-07")

for (rd <- runs) {

    //--- Owl Props ---//
    val props = new Props()
    props.filePath = s"${filePath}${rd.replace("-","")}.csv"
    props.fileQuery = "select * from dataset"
    props.runId = rd
    props.dataset = "CleanCSVFiles"
    props.calculateBoundaries = true
    props.catOutOn = false
    props.nullValue = "N.A."

    //--- Basic Spark ---//
    val bb = spark.read
                .option("inferSchema", "true")
                .option("header", "true")
                .option("delimiter", ",")
                .option("charset", "UTF8")
                .option("nullValue", props.nullValue)
                .csv(props.filePath)

    //--- Spark File ETL ---//
    val bb1 = bb.filter(row => row.getAs("SEC_ID") != "META_ZZ")
              .withColumn("OWL_RUN_ID", lit( props.runId) )
              .withColumn("OWL_RUN_ID", to_date($"OWL_RUN_ID") )

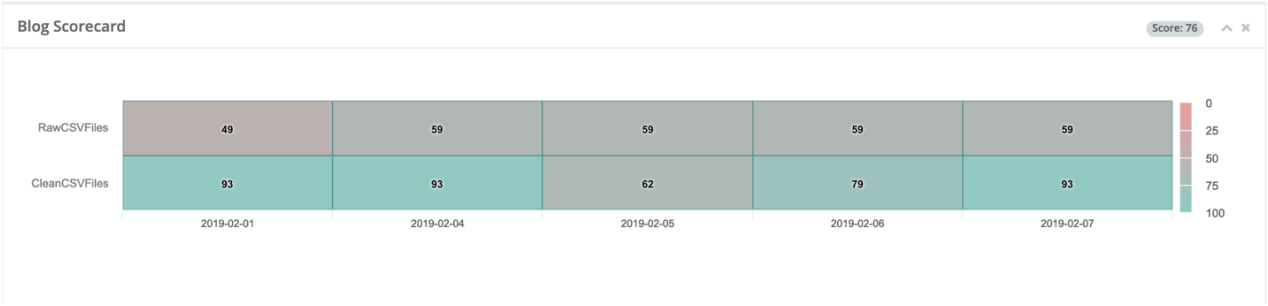
    // file staging area
    props.filePath = s"${filePath}/cleaned/${props.runId}"
    bb1.write
       .mode("overwrite")
       .format("com.databricks.spark.csv")
       .option("header", "true")
       .save( props.filePath)

    val df = spark.read
                .option("inferSchema", "true")
                .option("header", "true")
                .option("delimiter", ",")
                .option("charset", "UTF8")
                .option("nullValue", props.nullValue)
                .csv( props.filePath)

    //--- Run Owl ---//
    val owl = new Owl(df, props)
    owl.register(props)
    owl.owlCheck
}
```

Spark ETL
after cleanse
write to staging then
run another Owl DQ

The DQ block of code is essentially the same, however, there is a new DQ property added to auto filter values “props.nullValue = ‘N.A’”. This finds every cell that has the value of N.A. and conforms it to a “null”. Once the file is read into a Spark DF, you use Spark to “Filter” out the erroneous record on line 36 in the code snippet above. Notice we are also adding an Owl_Run_ID date as this dataset did not have a date that conforms easily. After the ETL process cleanses the data, you then have Collibra DQ's data quality engine scan the newly processed Spark DF, storing the results into a dataset called CleanCSVFiles.



Notice the composite scores in the boxes are substantially better for the CleanCSVFiles dataset than what they are for the original RawCSVFiles. In the next article, we will look deeper at the intelligence a DQ scan garners on a dataset when run over several days and how DQ surfaces different patterns, behaviors, trends and more in the data itself.

Bank Loans

> **Warning** This documentation is archived and is no longer maintained.

It is common for banks to lend money in return for monthly payments with interest. However, to do so a bank must make sure that the applications are valid and well formed to begin the underwriting and approval process. The following list comprises some basic lending concepts to Collibra DQ.

1. Credit Score Validation
2. SSN Validation
3. Loan to Value Validation
4. Interest Rate Validation
5. Duplicate Loan Applications
6. Loan Amount Validation
7. Loan Completeness Validation



member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	credit_rating	rate	grade	sub_grade	emp_title	emp_length	home_ownership	annual_inc	verification_status	issue_d	loan_status
1	2500	2500	2500	36 months	13.56	84.92	FAIR	6.5	C	C1	Chef	10+ years	RENT	55000	Not Verified	18-Dec	Current
1	2500	2500	2500	36 months	13.56	84.92	FAIR	6.5	C	C1	Chef	10+ years	RENT	55000	Not Verified	18-Dec	Current
2	30000	30000	30000	60 months	0	777.23	POOR	8.5	D	D2	Unemployed	10+ years	MORTGAGE	90000	Source Verified	18-Dec	Current
2	30000	30000	30000	60 months	0	777.23	POOR	8.5	D	D2	Unemployed	10+ years	MORTGAGE	90000	Source Verified	18-Dec	Current
2	30000	30000	30000	60 months	18.94	777.23	POOR	8.5	D	D2	Postmaster	10+ years	MORTGAGE	90000	Source Verified	18-Dec	Current
2	30000	30000	30000	60 months	18.94	777.23	POOR	8.5	D	D2	Postmaster	10+ years	MORTGAGE	90000	Source Verified	18-Dec	Current
3	5000	5000	5000	36 months	0	180.69	POOR	8.5	D	D1	Administrative	6 years	MORTGAGE	59280	Source Verified	18-Dec	Current
3	5000	5000	5000	36 months	17.97	180.69	POOR	8.5	D	D1	Administrative	6 years	MORTGAGE	59280	Source Verified	18-Dec	Current
3	5000	5000	5000	36 months	17.97	180.69	POOR	8.5	D	D1	Administrative	6 years	MORTGAGE	59280	Source Verified	18-Dec	Current
4	4000	4000	4000	36 months	0	146.51	POOR	8.5	D	D2	IT Supervisor	10+ years	MORTGAGE	92000	Source Verified	18-Dec	Current

1. Credit Score

Business Check	Collibra DQ Feature	Manual vs Auto
Is the credit score a whole number?	BEHAVIOR	AUTO
Is the credit score within a valid range? (between 300 - 850)	RULE	credit_score between 300 and 850
Is the credit score NULL or Missing?	BEHAVIOR	AUTO

2. SSN Validation

Business Check	Collibra DQ Feature	Text
Is the SSN format valid?	RULE	AUTO-SSN detection

Business Check	Collibra DQ Feature	Text
SSN is PII.	SENSITIVITY	AUTO-SSN labeled
Is the SSN NULL or Missing?	BEHAVIOR	AUTO
Does the SSN belong to the Applicant?	PATTERN	SSN -> first_name, last_name

3. Loan to Value

Business Check	Collibra DQ Feature	Text
Are loan amount and asset value (home or auto) valid numbers?	BEHAVIOR	AUTO
95% loan to value ratio to approve?	RULE	loan / asset_value < .95

4. Interest Rate

Business Check	Collibra DQ Feature	Text
Interest rate between min and max allowable range for the loans credit rating.	RULE COMPLEX	loan l join rates r on l.credit_ rating = r.credit_rating where l.rate between r.min_rate and r.max_rate

5. Duplicate Loan Applications

Business Check	Collibra DQ Feature	Manual vs Auto
Ensure we don't issue the same loan twice.	DUPE	first_n, last_n, SSN, Address

6. Loan Amount

Business Check	Collibra DQ Feature	Manual vs Auto
Loan amount within lendable range	OUTLIER	AUTO
Loan amount within lendable range. Only lend money between 50K and 3M.	RULE	loan_amount between 50000 and 3000000

Resulting DQ Check

```
-lib "/home/opt/owl/drivers/postgres" \
-cxn postgres-gcp \
-q "select * from public.loan_risk_grade where last_pymnt_d =
'2019-04-01'" \
-key member_id -alias loan_risk \
-ds public.loan \
-rd "2019-04-01" \
-dl -loglevel INFO \
-h 10.142.0.29:5432/owltrunk \
-numexecutors 10 -executormemory 1g -drivermemory 4g \
-master yarn -deploymode cluster \
-sparkprinc user2@CW.COM \
-sparkkeytab /tmp/user2.keytab -tbin MONTH \
-dupe -dupeinc purpose -fpgon -fpgkey grade \
-fpginc grade,sub_grade -fpglb 365 -fpgdc last_pymnt_d \
-record member_id -dupecutoff 60 -dupepermatchupperlimit 99
```

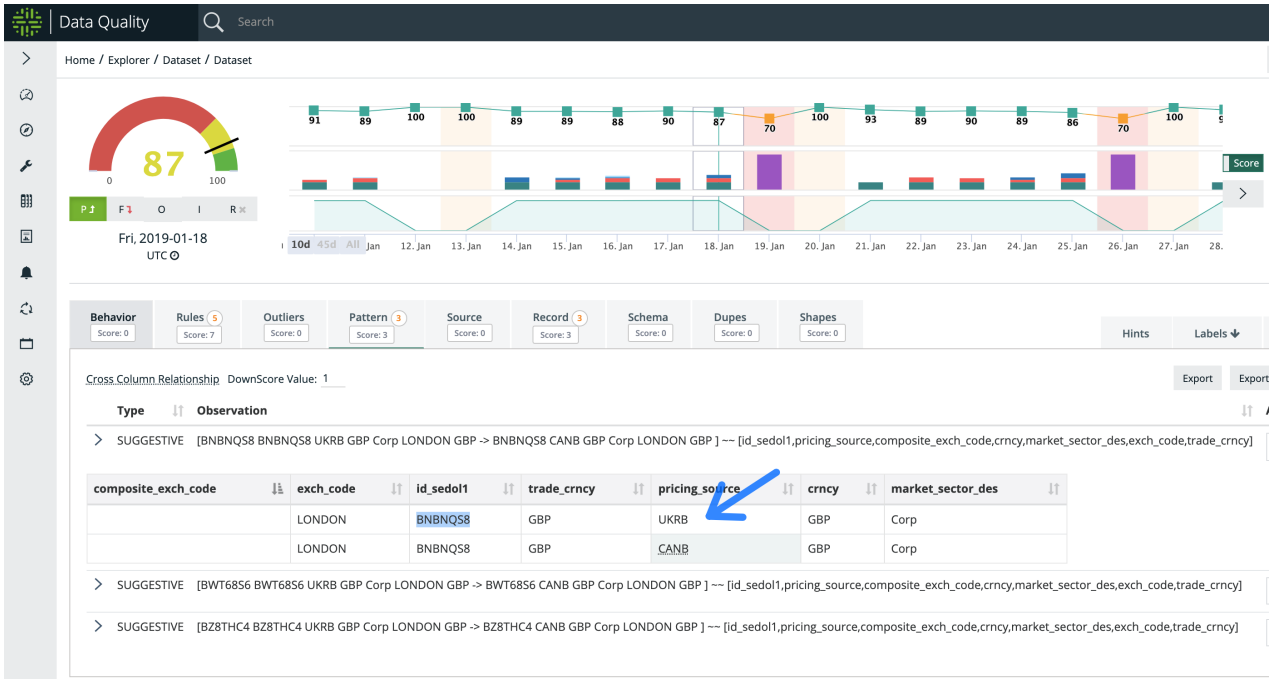
Which components did we use?

We made use of Profiles, Duplicates, Outliers and Rules in this example. The experiments were automatically cataloged and put on a job scheduler. The next time a loan issue arises, we will be able to take remediation action using the workflow Q. Over time we can see how the bank loan program is running via the report section.

Bloomberg Data

> **Warning** This documentation is archived and is no longer maintained.

Collibra DQ finds over 50 data quality issues per day in common market data through pattern mining. Sequential pattern mining is a tool that finds statistically relevant patterns between data examples where the values are delivered in a sequence.



Cyber Anomalies in Real-Time

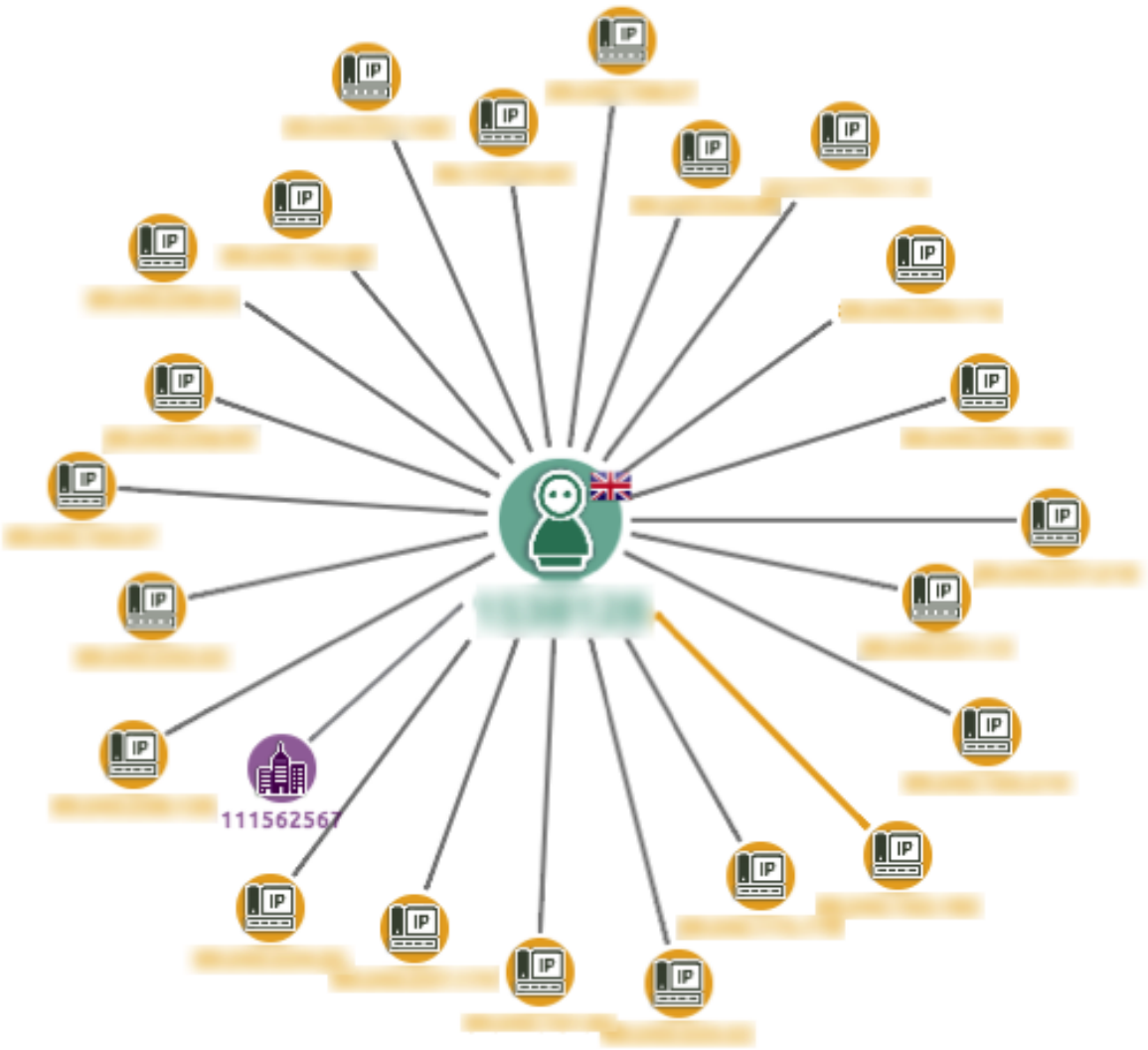
> **Warning** This documentation is archived and is no longer maintained.

With an increasing number of cyber threats, most of the cyber security team doesn't have the capacity to manually detect, monitor, and defend against all of them. Effective cyber threat management requires leveraging automation to inform decisions.

The Collibra DQ framework, provides organizations the ability to load and process diverse security data feeds at scale to detect network data anomalies. The DQ alerts enable network admins to respond to these events in timely manner.

The following scenario demonstrates how to detect anomalies with network traffic datasets.

1. Perform IP address validation.
2. Detect the unusual network traffic patterns based on locations.
3. Identify the suspicious packets based on size.
4. Detect the malicious activity based on source and destination IP addresses.



Infosec dataset preview

Data set contains Timestamp, Source Workgroup, Source IP, Source Port, Destination Workgroup, Destination IP, Destination Port, Application, Protocol and Packet size information.

Timestamp	Source Workgroup	Source IP	Source Port	Destination Workgroup	Destination IP	Destination Port	Application	Protocol	Packet Size(B)
2010-12-01 00:00:00	Atlanta	10.0.0.4	53323	SFO	10.0.0.18	443	HTTPS	TCP	122849
2010-12-01 00:00:00	Atlanta	10.0.0.4	52419	SFO	10.0.0.18	22	FTP	TCP	81055
2010-12-01 00:00:00	Atlanta	10.0.0.1	51618	SFO	10.0.0.17	22	FTP	TCP	98392
2010-12-01 00:00:00	Atlanta	10.0.0.5	58518	SFO	10.0.0.18	990	FTPS	TCP	245533
2010-12-01 00:00:00	Atlanta	10.0.0.1	58329	Manhattan	10.0.0.13	989	FTPS	TCP	250059
2010-12-01 00:00:00	Atlanta	10.0.0.5	49765	SFO	10.0.0.21	22	FTP	UDP	29012
2010-12-01 00:00:00	Atlanta	10.0.0.2	63277	SFO	10.0.0.17	22	FTP	UDP	27761
2010-12-01 00:00:01	Atlanta	10.0.0.4	63870	SFO	10.0.0.19	990	FTPS	TCP	189328
2010-12-01 00:00:01	Atlanta	10.0.0.4	63340	Manhattan	10.0.0.12	990	FTPS	TCP	180320
2010-12-01 00:00:01	Atlanta	10.0.0.4	60839	SFO	10.0.0.18	989	FTPS	TCP	230823
2010-12-01 00:00:02	Atlanta	10.0.0.1	59764	SFO	10.0.0.17	443	HTTPS	TCP	169610
2010-12-01 00:00:03	Atlanta	10.0.0.4	58744	Manhattan	10.0.0.9	443	HTTPS	TCP	141319
2010-12-01 00:00:04	Atlanta	10.0.0.4	58895	Manhattan	10.0.0.14	22	FTP	TCP	21691
2010-12-01 00:00:04	Atlanta	10.0.0.2	49241	SFO	10.0.0.21	22	SSH	TCP	17662
2010-12-01 00:00:04	Atlanta	10.0.0.4	65170	Manhattan	10.0.0.14	443	HTTPS	TCP	115949
2010-12-01 00:00:04	Atlanta	10.0.0.1	56006	Manhattan	10.0.0.12	990	FTPS	TCP	142744
2010-12-01 00:00:06	Atlanta	10.0.0.5	50528	SFO	10.0.0.21	990	FTPS	TCP	132937
2010-12-01 00:00:06	Atlanta	10.0.0.2	52919	Manhattan	10.0.0.10	22	SSH	TCP	14111
2010-12-01 00:00:06	Atlanta	10.0.0.4	58538	SFO	10.0.0.17	80	HTTP	TCP	121021
2010-12-01 00:00:06	Atlanta	10.0.0.6	56345	SFO	10.0.0.17	989	FTPS	TCP	126612
2010-12-01 00:00:06	Atlanta	10.0.0.6	59640	Manhattan	10.0.0.13	443	HTTPS	TCP	165096
2010-12-01 00:00:06	Atlanta	10.0.0.2	54127	SFO	10.0.0.21	22	SSH	TCP	19616

IP Address format validation

Business Check	Collibra DQ Feature	Text
Is IP a valid format?	RULE	AUTO-IP detection
Is the IP address NULL or Missing?	BEHAVIOR	AUTO

Source and destination workgroups

Business Check	Collibra DQ Feature	Text
Is it usual network traffic based on locations?	PATTERN	Source_Workgroup -> Destination_Workgroup

Source and Destination IP Address validation

Business Check	Collibra DQ Feature	Text
Is it usual network traffic based on source and destination IP?	PATTERN	Source_IP -> Destination_IP

Packet Size

Business Check	Collibra DQ Feature	Text
Is the Packet Size NULL or Missing?	BEHAVIOR	AUTO
Packet Size within normal range?	PATTERN	Source_IP -> Packet_SizeB

Resulting DQ Check

```
-f file:///home/danielrice/owl/bin/demos/infosec/ -d tab \
-fullfile -fq "select * from dataset" -encoding UTF-8 -ds
infosecv2 \
-rd "2020-04-04" -dl -dlinc Destination_IP,Packet_SizeB,Source_
IP \
-dlkey Source_IP -fpgon -fpginc Destination_Workgroup -fpgkey
Source_Workgroup \
-df "yyyy-MM-dd" -loglevel INFO -h 10.142.0.29:5432/owltrunk -
owluser admin \
-fpgsupport .000000001 -fpgconfidence 0.4
```

Which components did we use?

DQ addresses the issue of efficient network traffic classification by performing unsupervised anomaly detection and uses this information to create dynamic rules that classify huge amounts of Infosec data in real time.

By providing Infosec datasets, along with anomaly records DQ outlier and pattern algorithms found the anomaly in the network traffic. It mainly detects the following anomalies:

1. Traffic between Atlanta->Texas.
2. The packet size extremely low between Atlanta->Texas.
3. Atlanta source IP and Texas Destination IP.

Realtime DQ provides the alerts on network traffic anomalies, which can help network admins to do further deep analysis and takes preventative measure, which is a daunting task with huge amount of data.

Sample Dataset

infosec-anomaly.csv
Binary

Financial FxRate Data

> **Warning** This documentation is archived and is no longer maintained.

Collibra DQ Automatically Alerts to Incorrect Foreign Exchange (FX) Rate Data without a Single Rule

FX Rate data commonly looks like the below table. Often you have a TO currency and a FROM currency with the RATE being the multiplier column for conversion. For example, in March you would need to spend $1 US Dollar and 18 US cents to receive $1 Euro. "In exchange for":

TO_CURR	FROM_CURR	RATE	DATE
USD	EUR	0.82	2019-03-12
EUR	USD	1.18	2019-03-12
USD	YEN	111.0	2019-03-12

28,000 Currency Pairs

There are roughly 28,000 currency pairs and the exchange rates change throughout the day but at a minimum most banks are concerned with the daily close of the FX Rate. Now imagine trying to write a rule for each currency pair. You'd have to know the relationship and adjust a static threshold for each of the 28K pairs every couple of days to keep the rule intact. Our minds quickly jump to a conclusion that we might be able to solve this with simple math. We can get closer using averages or percent change formulas but these formulas quickly come up short when some currencies commonly fluctuate more than others. Our minds then quickly graduate from stats 101 to 201 and we could consider the individual variance of every combination. But even this only gets us so far as time is an important dimension, the length of time or window can often be tricky to calculate. The problem gets harder when you run your basic stat model and receive multiple false positive alerts. Signal-to-noise ratio is important, confidence factors are important and down-training individual foreign currencies that don't seem to fit your statical model are important. Knowing if you copied the data incorrectly, truncated nine levels of precision on the decimal, or if the source provider sent the wrong information is important. Needing the ability to flag exceptions in production on a single currency pair while *not* flagging the other 27,445 pairs. Using a feedback loop so that the data steward interactions are captured and learned from vs having to take the same corrective action over and over. What happens when there is a typo in the currency pair or a single pair goes missing? The answer is that rules don't scale and we need much more than just one off statistical metrics to have a robust and trust worthy data quality program.

Consistency

Even when it is possible to deploy a team of smart people to build a solution to handle this use case, the question then becomes "but what about all my other data, don't I want similar yet different controls on everything?" Especially since FX Rate data by itself doesn't mean that much and is often combined with a number of other data sets to produce value. What if those data sets aren't accurate either? But those data sets have very different columns, different relationships and different time windows. DQ takes an auto-learning approach whereby it interrogates and runs fitness tests against each data set individually to devise the best statistical and learning approach. The goal being to provide an automated and elegant way to have consistent controls across all your data sets.

Auto Adapting DQ Checks for FxRate Data

```
-ds fx_rate \
-rd $rd \
-dl \
-dlkey TO_CURR,FROM_CURR \
-q "select * from FX_RATE where date = '${rd}'" \
-cxn dbConnName \
-dupe -dupeinc TO_CURR,FROM_CURR -depth 0
```

What this DQ Check Does

- Automatic correlation and relationship analysis
- Histogramming and segmentation analysis
- Anomaly detection
- Currency pair tracking
- Schema evolution
- Removes 28K static rules
- Duplicate detection for redundant currency pairs

Healthcare Data Quality

> **Warning** This documentation is archived and is no longer maintained.

Collibra DQ connects all members of the healthcare continuum with trustworthy, timely, and meaningful patient data, while reducing the time, expense, and effort required by 70 percent.

Poor data quality in healthcare is the leading problem that maligns patient outcomes. Hospitals and health information exchanges (HIEs) still struggle with patient matching issues, with most citing data quality problems and poor algorithms as top barriers to patient matching. Correctly linking patient data across organizations is a key element of value-based care, patient safety, and care coordination. Duplicate or mismatched records can result in privacy risks, claim denials, redundant medical tests or procedures, and reporting errors.

The lack of accurate and reliable data quality in healthcare leads to dire consequences that are completely preventable, as shown in DQ's troponin example below. Complete and accurate data is a vital component of our complex health system, and anything less is an unacceptable

risk. DQ provides the predictable data quality that healthcare organizations need to deliver high-quality care that we all strive to achieve.

Health Insurance Claims Data

> **Warning** This documentation is archived and is no longer maintained.

Poor data quality is the primary cause for diagnostic providers receiving incomplete health care payments during Revenue Cycle Management (RCM).

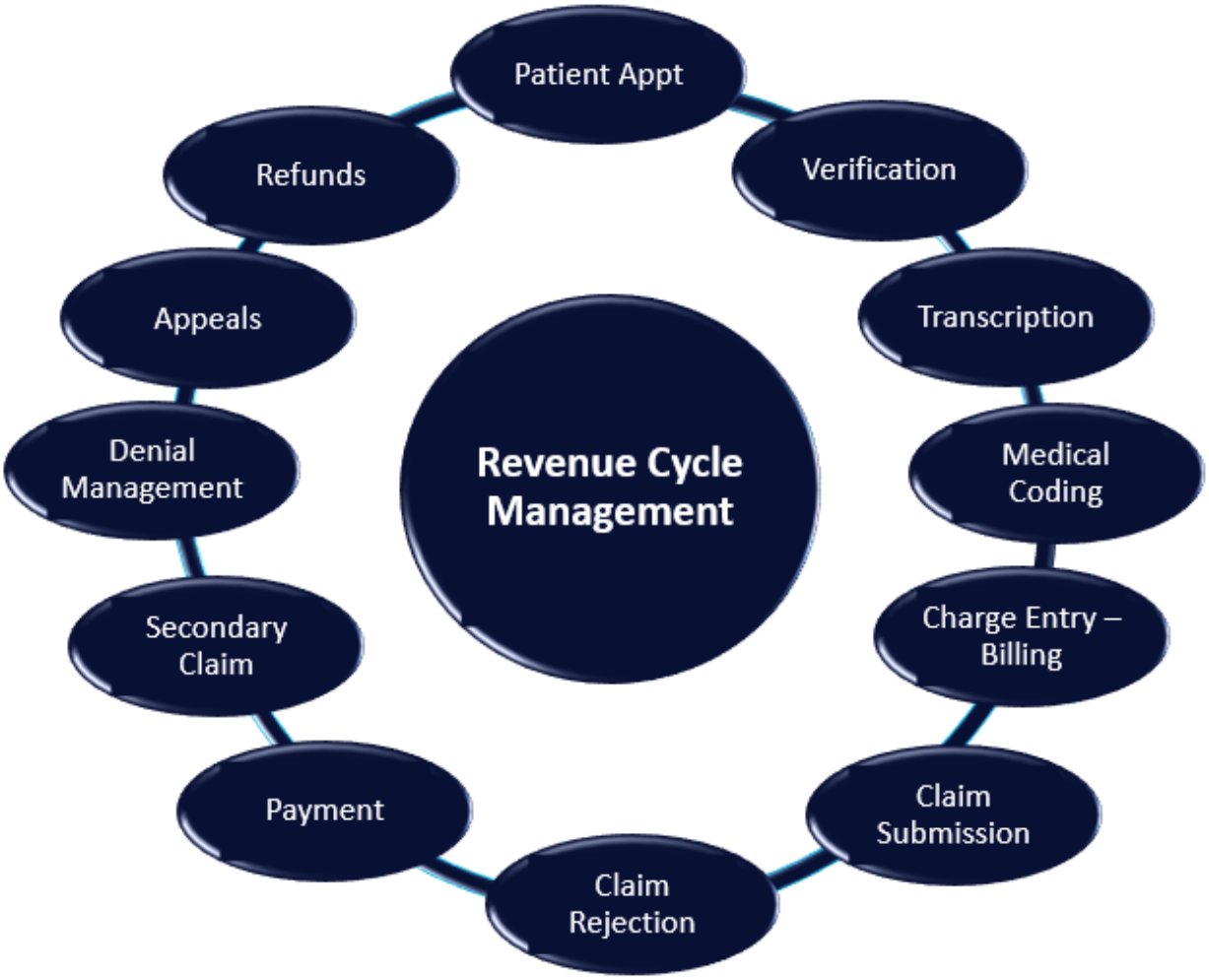
Revenue Cycle Management is the process of identifying, collecting and managing the practice's revenue from payers based on the services provided. A complete RCM process is critical for a healthcare practice to maintain financial viability and continue to provide quality care for their patients.

Inaccurate claims data, is the primary cause for diagnostic providers receiving incorrect payments for their services. Most providers struggle with the quality of the data that they receive, and without direct access to the patients, it can be an expensive, laborious process to correct incomplete, or missing data that is required for claim reimbursement.

Cleaning up or correcting incomplete data is not a step in the claims process that can be skipped. It must be done to assure the reimbursement process is accurate, and complete in the agreed time frame. Automating the data quality during intake is the key to the timely completion of the reimbursement process, and saving the cost and effort of correcting the data down stream.

```
Increase revenue from insurance and patient payments
Spend less time tracking down missing patient information
Lower error processing rates
Reduce operating costs
Improve claim processing speed
```

The revenue cycle includes all the administrative and clinical functions that contribute to the capture, management and collection of patient service revenue, according to the Healthcare Financial Management Association.




```
Preregistration - Collecting preregistration information
Verification - Patient eligibility and benefit is verified
Transcription - recording the diagnoses and procedure
Medical Coding - Properly coding diagnoses and procedures.
Charge capture - Medical services into billable charges.
Claim submission - Submitting claims to insurance companies.
Claim Rejection - when necessary
Payment Posting - Determining patient balances, collection
Secondary Claim Submission
Denial Management - Applying or rejecting payments remittance
Medical Appeals - Examining the necessity of medical services.
Refund - where aplicable
```

Intraday Positions

> **Warning** This documentation is archived and is no longer maintained.

It is common for financial organizations to receive a steady stream of files that have hourly or minutely data. The files might trail the market in a near real-time fashion. Below is an example:

```
--positions/
   |--2019/
     |--01/
       |--22/
         position_2019_01_22_09.csv
         position_2019_01_22_10.csv
         position_2019_01_22_11.csv
         position_2019_01_22_12.csv
```

File Contents @ 9am

TIME	COMPANY	TICK	SIDE	QTY
2019-01-22 09:00	T&G	xyz	LONG	300
2019-01-22 09:00	Fisher	abc	SHORT	20
2019-01-22 09:00	TradeServ	def	LONG	120

File Contents @ 10am

TIME	COMPANY	TICK	SIDE	QTY
2019-01-22 10:00	T&G	xyz	LONG	280
2019-01-22 10:00	BlackTR	ghi	SHORT	45

Notice that during the day you may or may not have a position for every company recorded. We need a way to link the "company" to its position throughout the day but not alert in cases where they simply did not trade or adjust their position. Collibra DQ offers real-time outlier detection for this scenario (see code snippet below). We also need to ensure that each

company's position is only represented once per file (per hour in this case) because positions are already the aggregate view of the trades, so they should be unique. DQ offers duplicate detection (see code snippet below).

Chapter 13

Collibra DQ Pipeline

```
// Part of your pipeline includes the ingestion of files that
have the date
// and hour encoded in the file name. How do you process those
files using Collibra DQ?
//
// Format: <name>_<year>_<month>_<day>.csv

val filePath = // <set this> positions/2019/01/22/positions_
2019-01-22_09.csv

// Configure Collibra DQ.
val opt = new OwlOptions
opt.dataset = "positions"
opt.load.delimiter = ","
opt.load.fileQuery = "select * from dataset"
opt.load.filePath = file.getPath

opt.outlier.on = true
opt.outlier.key = Array("COMPANY")
opt.outlier.timeBin = TimeBin.HOUR

opt.dupe.on = true
opt.dupe.include = Array("COMPANY", "TICK")
opt.dupe.exactMatch = true

// Parse the filename to construct the run date (-rd) that will
be passed
// to Collibra DQ.
val name = file.getName.split('.').head
val parts = name.split("_")
val date = parts.slice(2, 5).mkString("-")
val hour = parts.takeRight(1).head

// Must be in format 'yyyy-MM-dd' or 'yyyy-MM-dd HH:mm'.
val rd = s"${date} ${hour}"

// Tell Collibra DQ to process data
opt.runId = rd

// Create a DataFrame from the file.
val df = OwlUtils.load(opt.load.filePath, opt.load.delimiter,
spark)

// Instantiate an OwlContext with the dataframe and our custom
configuration.
val owl = OwlUtils.OwlContext(df, spark, opt)

// Make sure Collibra DQ has catalogued the dataset.
```

```
owl.register(opt)

// Let Collibra DQ do the rest!
owl.owlCheck
```

DQ Coverage for Position data

- Schema evolution
- Profiling
- Correlation analysis
- Segmentation
- Outlier detection
- Duplicate detection
- Pattern mining

Security Reference Data

> **Warning** This documentation is archived and is no longer maintained.

Pattern recognition for cross column, categorical, and conditional relationships.

Given the interconnected, automated nature of the data generated by reporting, exchanges, and source systems - hidden patterns go unnoticed.

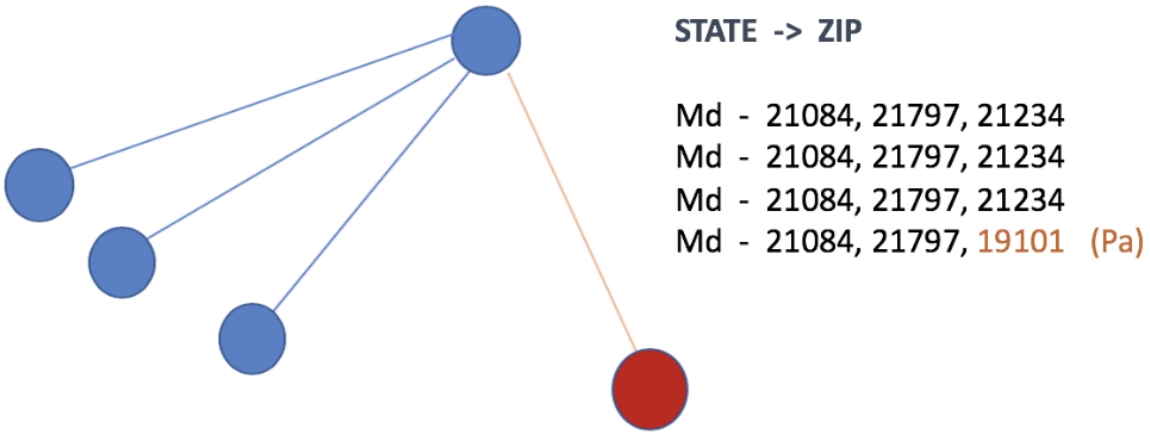
Financial firms of all shapes and sizes ingest financial data for a variety of reasons. A few vendors include Bloomberg, Thomson Reuters, ICE Data Services or SIX Financial Information.

> **Note** **In no uncertain terms, critical business decisions rely on the accuracy of this data.**

```
|-- RECEIVED_DATE: integer (nullable = true)
|-- B_SECURITY_ID: string (nullable = true)
|-- B_STATUS: integer (nullable = true)
|-- NUM_FIELDS: integer (nullable = true)
|-- PX_LAST: double (nullable = true)
|-- PX_FIXING: double (nullable = true)
|-- PX_BID: double (nullable = true)
|-- PX_MID: double (nullable = true)
|-- PX_ASK: double (nullable = true)
|-- ID_SEDOL1: string (nullable = true)
|-- ID_CUSIP: string (nullable = true)
|-- ID_BB_UNIQUE: string (nullable = true)
|-- LAST_UPDATE_DT: integer (nullable = true)
|-- PX_OPEN: double (nullable = true)
|-- PX_HIGH: double (nullable = true)
|-- PX_LOW: double (nullable = true)
|-- PX_VOLUME: integer (nullable = true)
|-- PRICING_SOURCE: string (nullable = true)
|-- COMPOSITE_EXCH_CODE: string (nullable = true)
|-- CRNCY: string (nullable = true)
|-- ID_MIC_PRIM_EXCH: string (nullable = true)
|-- MARKET_SECTOR_DES: string (nullable = true)
|-- EXCH_CODE: string (nullable = true)
|-- INFLATION_LINKED_INDICATOR: string (nullable = true)
|-- INFLATION_LAG: integer (nullable = true)
|-- IDX_RATIO: double (nullable = true)
|-- TRADE_CRNCY: string (nullable = true)
```

This data is not monolithic and most real-world data easily consists of over 100 columns. Maintaining the quality can be challenging, given the variety of sources feeding into just a single feed. Even the most simple quality checks can snowball into a daunting task. Everything from tickers, sedols, cusips, products, sub-products, issuers, and issuing countries can further complicate the problem. Identifying anomaly values earlier in the data ingestion process can significantly reduce downstream complexity. Furthermore, finding improbable patterns before they're used for making decisions can save costly remediation efforts.

An easy way to think about Pattern Analysis. Columns that belong 'together'.



A common application of Collibra DQ is to identify securities that are violating historical patterns. Conditional dependencies can be discovered and used as a guide to highlight misreporting. Rather than defining look-up tables, reference datasets, and predefining exact conditions - security specific patterns can be derived using probabilistic inferences.

By clicking columns that belong together, a robust framework for cross-column pattern analysis can be generated. The equivalent of dozens of multi-column, conditional rule checks can be applied with just a few clicks. In creating a confidence and probability weighted classification algorithm, this is both adaptive and granular.

Column	Key	Date	Data Type
☐ b_security_id	☑	☐	text
☐ b_status	☐	☐	int4
☑ composite_exch_code	☐	☐	text
☑ crncy	☐	☐	text
☑ exch_code	☐	☐	text

Using this technique, DQ reduces false positives, scales coverage, and quickly models a more complex series of checks than domain experts would want to develop.

Validating Data Movement

> **Warning** This documentation is archived and is no longer maintained.

Validate Data Integrity between distinct storage systems.

Record-for-Record Reconciliation

When you're copying or moving data between distinct storage systems, such as multiple HDFS clusters or between non-HDFS storage and cloud storage, it's a good idea to perform some type of validation to guarantee data integrity. This validation is essential to be sure data wasn't altered during transfer.

Detect potential data corruption caused, for example, by older versions of drivers, parsing errors, connection limits, noisy network links, memory errors on server computers and routers along the path, or software bugs (such as in a library that customers use).

Common Data Copying/Movement Scenarios

- Landing, Loading, Persisting third-party files:
  - Landing daily files.
  - Loading daily files into staging location.
  - Persisting data in lake or warehouse.
- Cloud Migrations:
  - Between existing database storage to optimized cloud storage formats.
  - Between local file systems and cloud relational database.
- Data Lake or Data Warehouse:
  - Migrating data from single storage system to distributed storage.
  - Consolidating storage systems to a single lake or warehouse.
- Same Storage, Different Environments:
  - Copying same data between Dev, QA, and Prod environments.

Shortcomings of Existing Validation Checks

- Low-level integrity checks, such as row counts and column counts, may not be sufficient.
- No easy way to reconcile between across non-HDFS files and database.

- Chunk verification requires storage size, format, and metadata to be exactly equal.
- Different data types in two distinct databases (Oracle and Teradata) do not reconcile.
- There are two different copies of the same files in HDFS, but with different per-file block sizes configured.
- There are two different instances of HDFS with different block or chunk sizes configured.
- Copying across non-HDFS Hadoop-compatible file systems.
- (HCFS), such as Cloud Storage.

Explicit end-to-end data integrity validation adds protection for cases that may go undetected by typical in-transit mechanisms.

Collibra DQ Integrity Validation

To ensure and protect against target systems getting out of sync or not matching the originating source, turn on `-vs` to validate that the source matches the target. For more information, go to DQ Job Validate Source.

Complete row, column, conformity, and value checks between any two distinct storage systems can be run against high-dimension or low-dimension datasets. This works between Files and/or Database storage, on-premise, or across cloud environments.

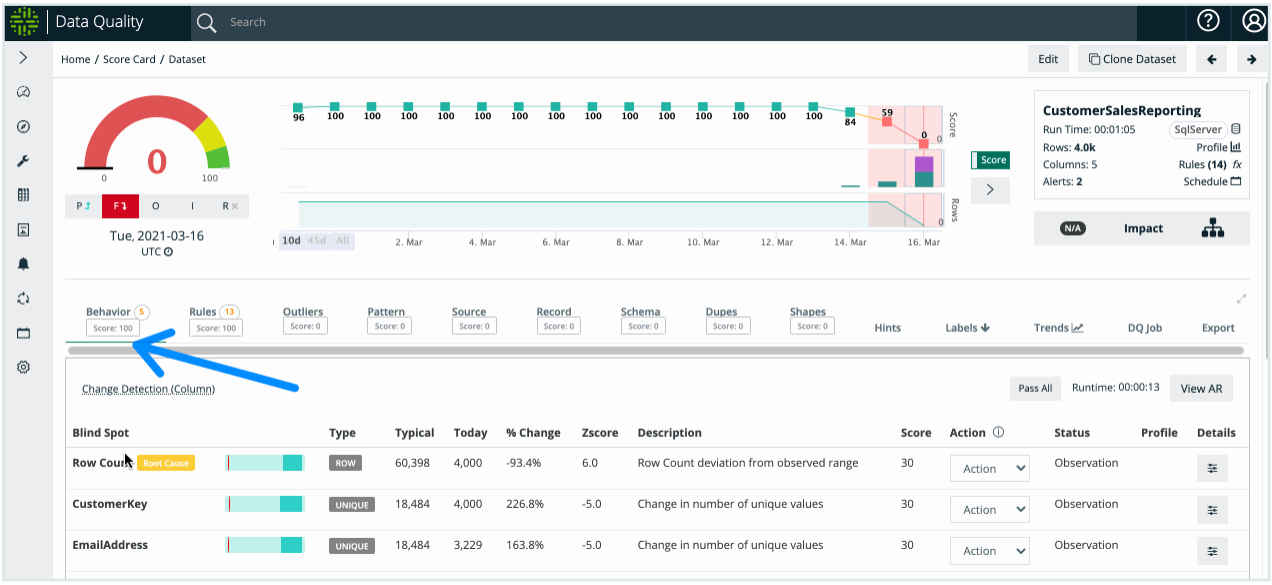
Data Quality Monitors

When specific data quality challenges require specific data quality detection techniques, Data Quality & Observability Classic offers a wide variety of advanced functionality. While Schema and Shapes utilize auto-discovery, other detection algorithms are best suited for users that understand their data and have specific use-cases in mind.

Behavior (automatic)

This is commonly referred to as statistical change detection or data observability.

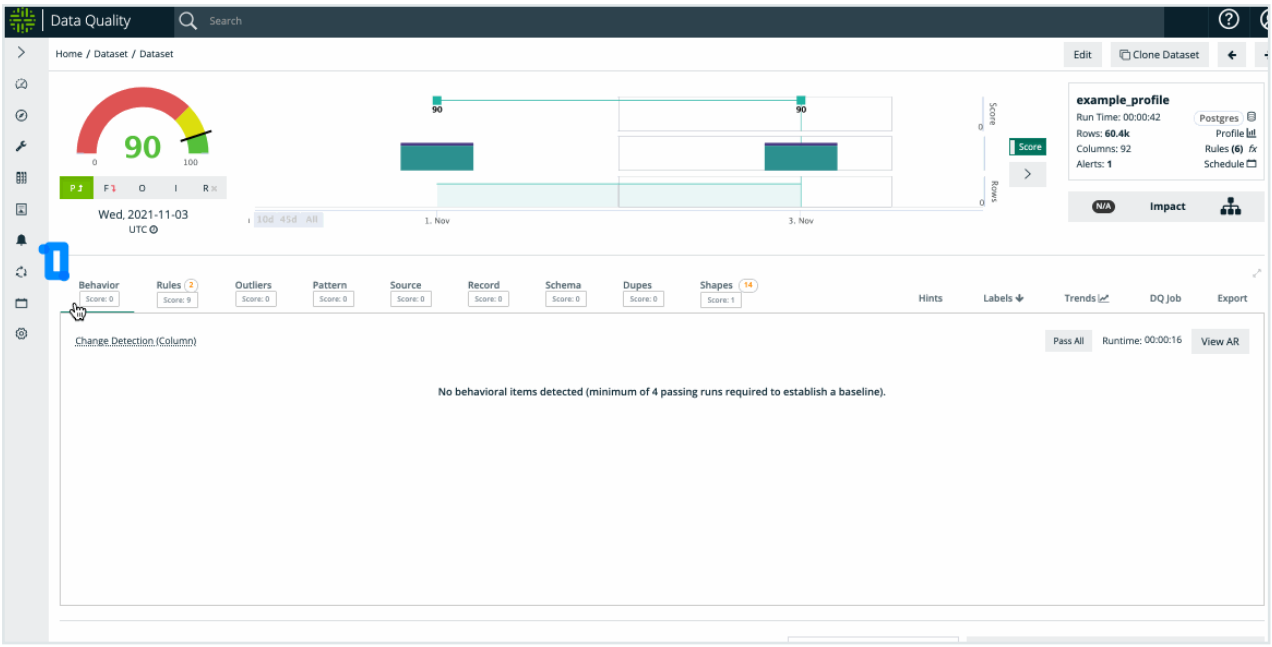
> **Note** Results are found under the Behavior tab (short for behavioral analytics). This tracks changes in the heuristics of the underlying data profiling metrics. The adaptive rules (AR) modal displays a complete list of monitoring types and criteria.



Evolution of Rule-based Data Quality

The main goal of Collibra Data Quality is to provide enterprise data quality insight while greatly reducing the volume of Rules that need to be written manually. When a dataset is brought under management, Collibra profiles the data and builds a model for each dataset. This allows Collibra to learn what "normal" means within the context of each data set. As the data changes, the definition of "normal" also changes. Instead of requiring you to adjust rule settings, Collibra continues to adjust its model. This approach enables Collibra to provide automated, enterprise-grade data quality coverage that removes the need to write dozens or even hundreds of rules per dataset.

> **Note** Behaviors is turned on by default. Monitoring will calibrate and detect DQ observations, based on the profiling activity.



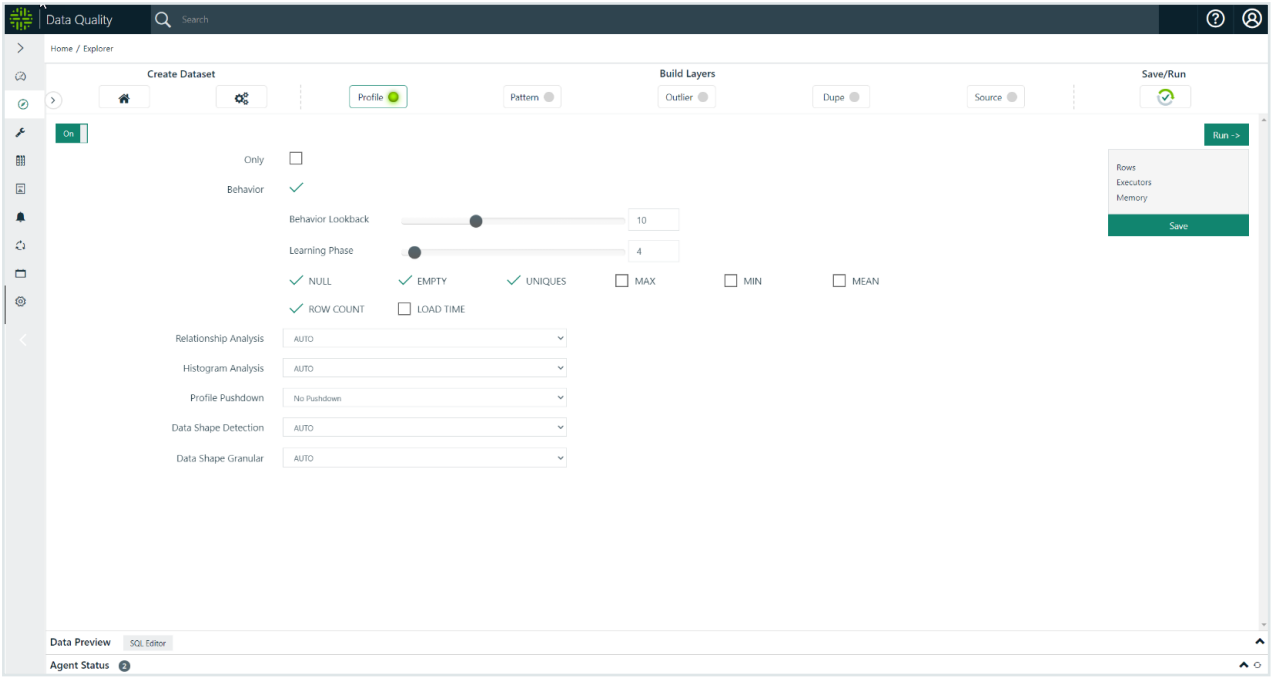
Using Behavioral Analytics (Change Detection)

Typically, data quality checks are scheduled to run on a given dataset daily. Behavior data quality, or change detection, is built on top of data calculated by the Profile activity. The default settings will often work just fine, however, Collibra lets you specify two key parameters:

Parameter	Description
Behavior Lookback	The number of DQ checks that the model encompasses. For example: A lookback of 10 means that the model is based on the combined statistics from the last 10 DQ checks of a dataset.
Learning Phase	The minimum number of DQ checks that are required before behavioral scoring is applied. Collibra DQ does not attempt to apply Behavioral scoring to a dataset until at least this many DQ checks are run on it.

Applicable Behavioral Factors

You can choose to forego the application of any of the above factors to the scoring of the model. For example, you can instruct Collibra Data Quality to not track MIN and MAX ranges of values in columns by unchecking the MIN and MAX checkbox. This prevents Collibra from detecting any extreme values in any column of the dataset using the Behavioral model.



With each run, Collibra Data Quality profiles the dataset at the column-level and begins to establish a model for the dataset. Initially, there is no need for any manual intervention, just keep the data coming. Within a few runs, the model becomes sufficiently robust to begin detecting data quality issues that are otherwise covered by manual rules. For example, Collibra may detect that a particular column experienced a spike in the number of NULL values (typically a manually defined rule).

> **Note** Collibra Data Quality's behavioral model consists of the following factors:
>
> - NULL values
> - Empty values
> - Cardinality
> - Datatype shifting
> - Row counts
> - Load time
> - Minimum value
> - Maximum value
> - Mean value

Over time, the definition of normal for any given column within the dataset can change. The data may legitimately become more sparse or decrease in volume. Collibra Data Quality continues to learn and adjust the model throughout the life of the dataset. However, if there is a drastic (but legitimate) change in the data, this could still mean several days of unnecessary

alerts while the model is adjusting. To accelerate model adjustment, Collibra DQ provides the ability to adjust the acceptable range for a given behavioral finding.

For example, Collibra DQ learned that a particular column typically has between 10% and 20% Empty values. Today, the column is 80% Empty values. Collibra raises a data quality issue and subtracts a proportional amount of points from the quality score of today's DQ run. You may review the finding and realize that there is a legitimate business reason for why that column has more empty values. With a few clicks, you can adjust the acceptable range for that finding. Collibra incorporates the user-defined inputs into the model and adjusts the current day's quality score. Collibra Data Quality would have eventually arrived at the correct range without any input, but without user input, it may have taken a few runs to get there.

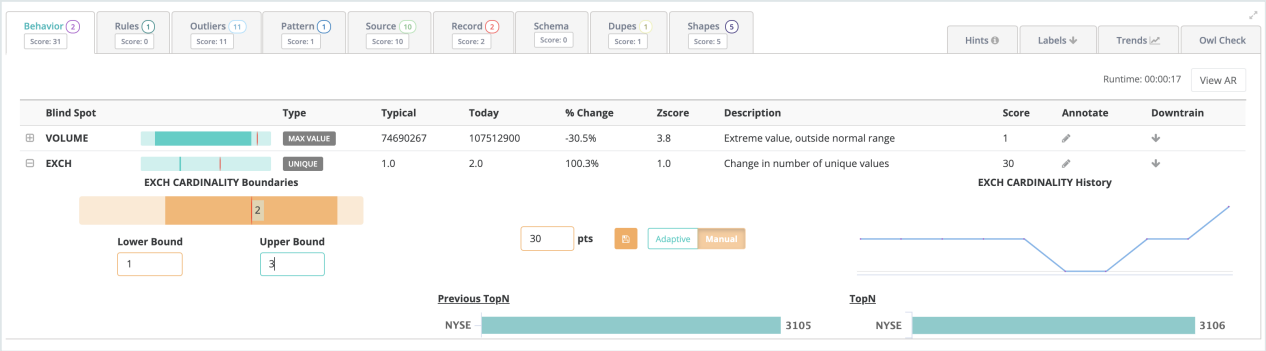
Drill-in to see the predicted range of valid values

Automatic flagging of break records with erroneous data.



The screenshot above shows some of the controls and visualizations that can be used to tune the Behavioral model. In this specific example, Collibra has detected that the cardinality of the EXCH field has doubled from 1 to 2 unique values. However, you can instruct Collibra to disregard this finding and adjust the model by manually specifying the range of values

acceptable in this column. To assist the user, Collibra provides a line chart and a historical topN visualization of this column's cardinality.



If you want to instruct Collibra that there can be as many as 3 valid values in the EXCH column, click the "Manual" button and adjust the upper bound from 1 to 3, then click the save button.



Collibra adjusts the Behavioral model's baseline, removes the finding, and adjusts the quality score. From that point on, Collibra knows the acceptable range for unique values in the EXCH column is between 1 and 3.

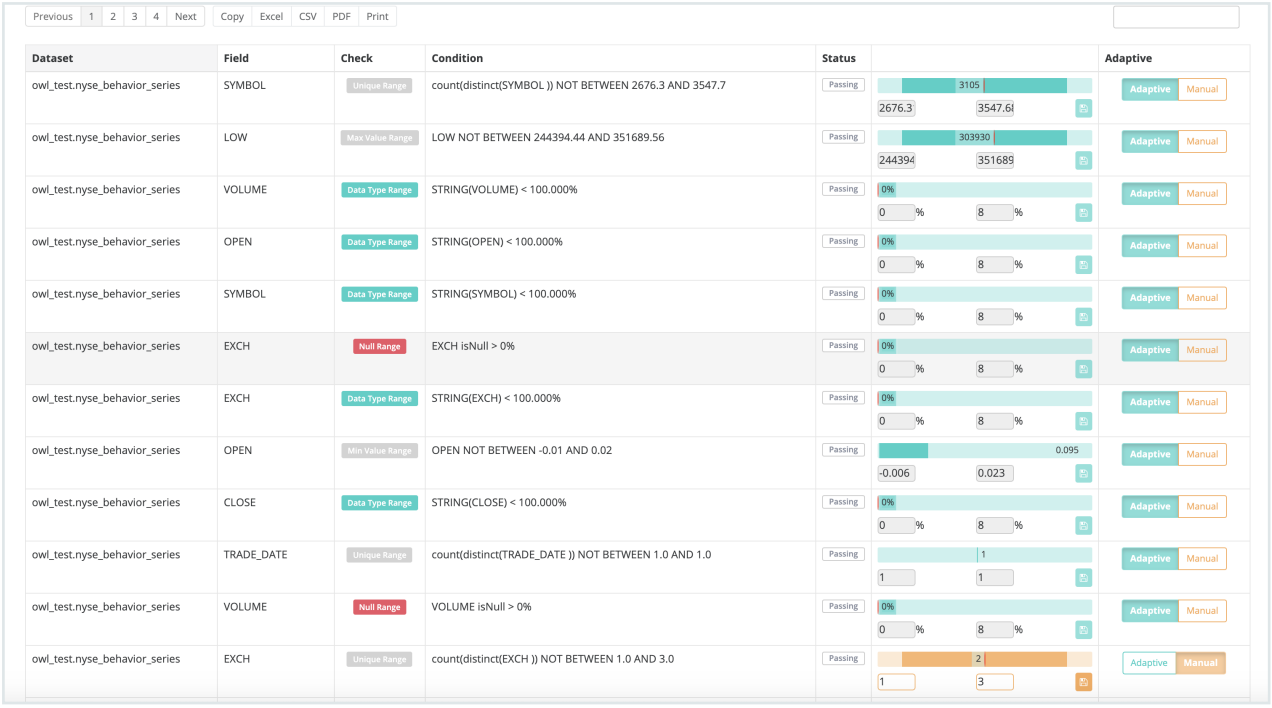
Item	Description
Blind Spot	The name of the column with a possible change detected during a DQ check.
Type	The type of DQ check performed on a given column. For example, Unique (Range) is the number of unique values that fall outside a given range.
Baseline	The baseline value is the mean of the preceding number of scans determined by the value selected for Behavior Lookback in the Profile section of the Explorer page.
Today	

Item	Description
% Change	The percent change from the value of one row to another. $x = \frac{V_2 - V_1}{\lvert V_1 \rvert} * 100$
Δ % Change	The delta percent change from the value of one row to another.
Zscore	The number of standard deviations away from the expected baseline value.
Description	The description of the type of DQ check performed on a given column.
Score	The value subtracted from your overall DQ score. The distance from the expected ranges set by the variance and boundaries of the baseline value. Expected ranges are also visible in the AR panel with graphs available in the Details panel for each line item.
Action	The item labels you can apply to an observation that let you train the behavioral model on future runs. Available options are Validate, Invalidate, and Resolve.
Status	The status of a DQ item, for example, Observation.
Profile	
Details	By clicking the Details button, a line graph lets you drill into changes over time.

Adaptive Rules

As Collibra Data Quality builds and evolves the behavioral model, it exposes all of the "Adaptive Rules" that it learns about. The above example demonstrates how Collibra learns and automatically applies rules. You have control but if left alone, Collibra learns what "Normal" means for a given dataset and scores it accordingly. This results in a large set of rules that are automatically applied and adapted as the dataset changes over time.

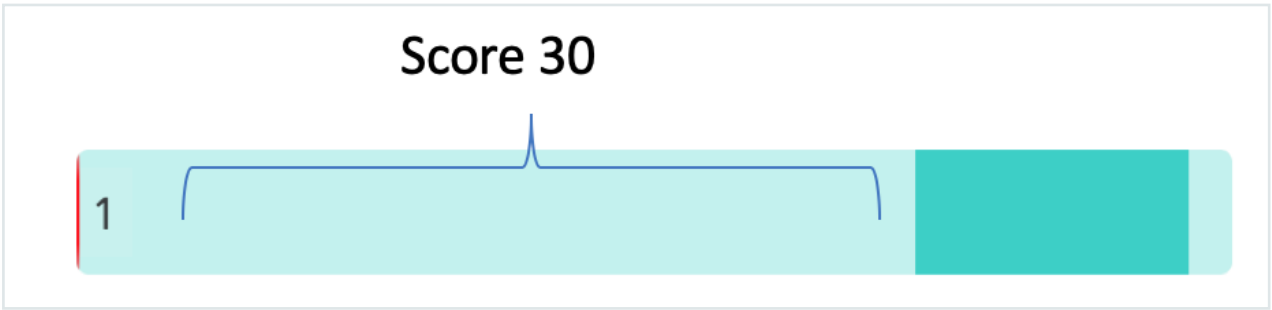
To view or modify Adaptive Rules, navigate to the Behavior tab on the findings page for the desired dataset and click the "View AR" button on the right side of the screen. This brings up a full list of Adaptive Rules.



The Adaptive Rules also provide you with the ability to adjust ranges derived from the behavior model. You can manually adjust the tolerance range and score of any Adaptive Rule. While this may at times be convenient, it is also just fine to let Collibra Data Quality handle the model tuning through its own learning process.

Scoring

In adaptive mode Collibra Data Quality automatically generates a DQ item score based on the egregiousness of the line item. This measurement is directly proportional to the distance from the green range to the red line. Example below.



The score can range from 0-30. This ties to the percent change and Z-Score. In cases when the Z-Score ranges from 0.0 - 6.0.

FAQ

Q: Which Collibra DQ API contains all behavioral checks (passing and breaking)?

- /v2/getdqchecksdetails.

Q: How is 'Mean' defined in the Behavioral Modal chart?

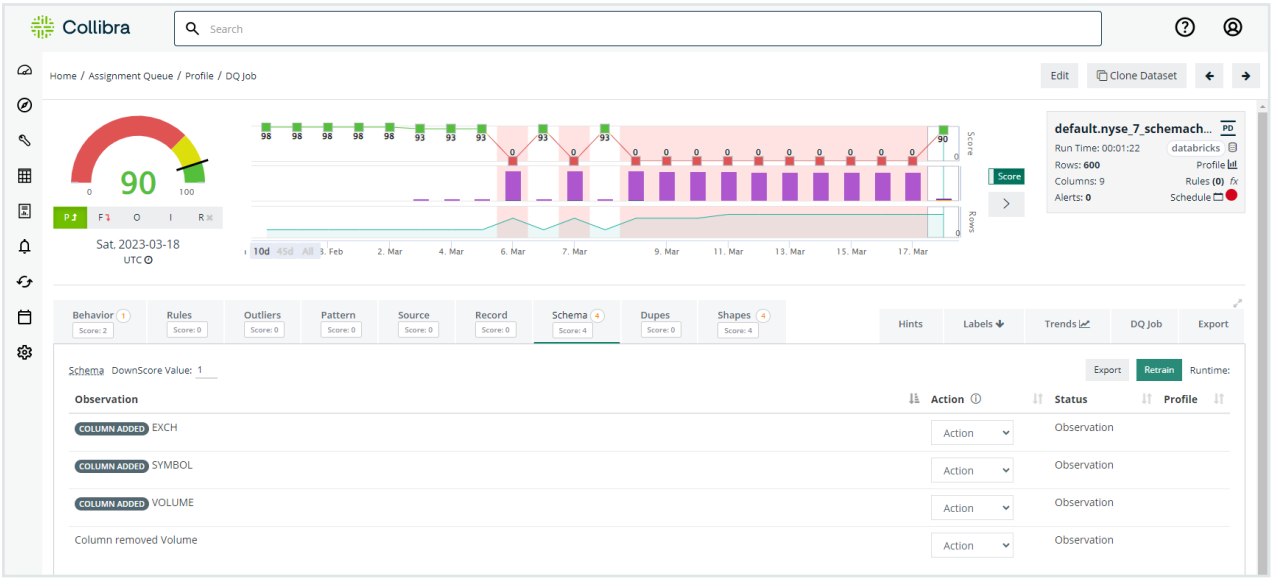
- Mean represents the average of behavioral lookback window e.g. if today is the 11th of the month, and the bhlb is set at 10, the mean will be the average of the 1st to the 10th, and the statistic on the 11th day will represent the change to that mean.
- Also of note: the mean only includes passing rules, not failed runs.

Schema (automatic)

Detect schema evolution and unexpected schema changes.

Dataset schemas are the columns or fields that define the dataset. They are often located in the header row of a tabular file or database table. However, JSON and XML are two examples of formats that include schema columns that are not in the header but rather nested throughout the document. Collibra DQ automatically without needing to turn on any features detects the schema columns as well as reads or infers their data types (varchar, string, double, decimal, int, date, timestamp, etc.). Collibra DQ observes each dataset so if a column is ever altered, removed or added it will automatically raise the event via its standard composite scoring system.

Scoring... Alerting... Schema Detection... Automatically



Schema Evolution is one of Collibra DQ's 9 DQ dimensions. It can be an important measurement for data stewards to understand how the dataset is changing overtime. The orange bar on the chart shows a change in schema and allows for drilling in over time.

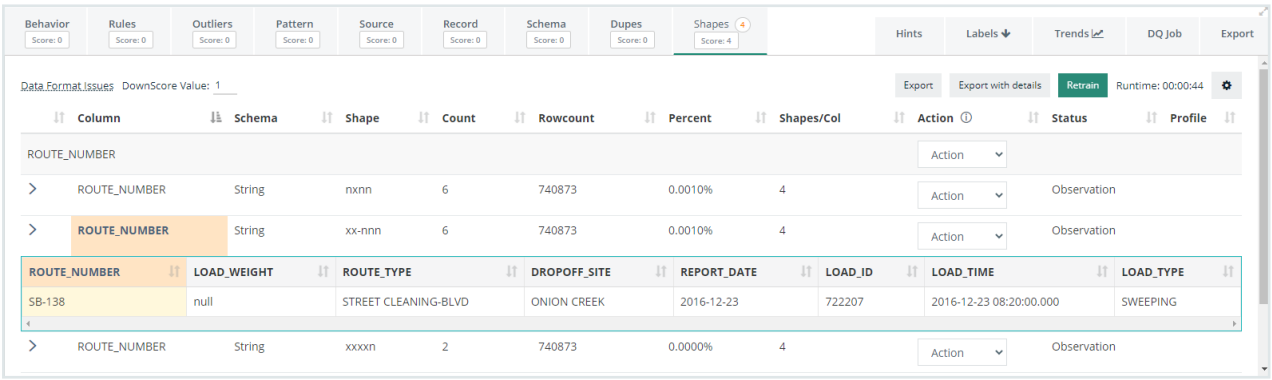
Shapes (automatic)

Collibra DQ automatically detects inconsistencies in data formats. These inconsistencies are where Data Scientists spend an enormous amount of time cleaning the data before building a ML model. Many reports have documented that over 80% of the time it takes to build a credible model comes from first understanding all the different formats and then writing munging or ETL style code to clean it before processing. What about all the patterns the process or person doesn't even know about?

auto_year	phone_num	id
2004	212-88-9878	3
1994	212 88-9880	5
1992	212 888 882	7
1996	212.888.88888	10
1993	212 888 9876	1

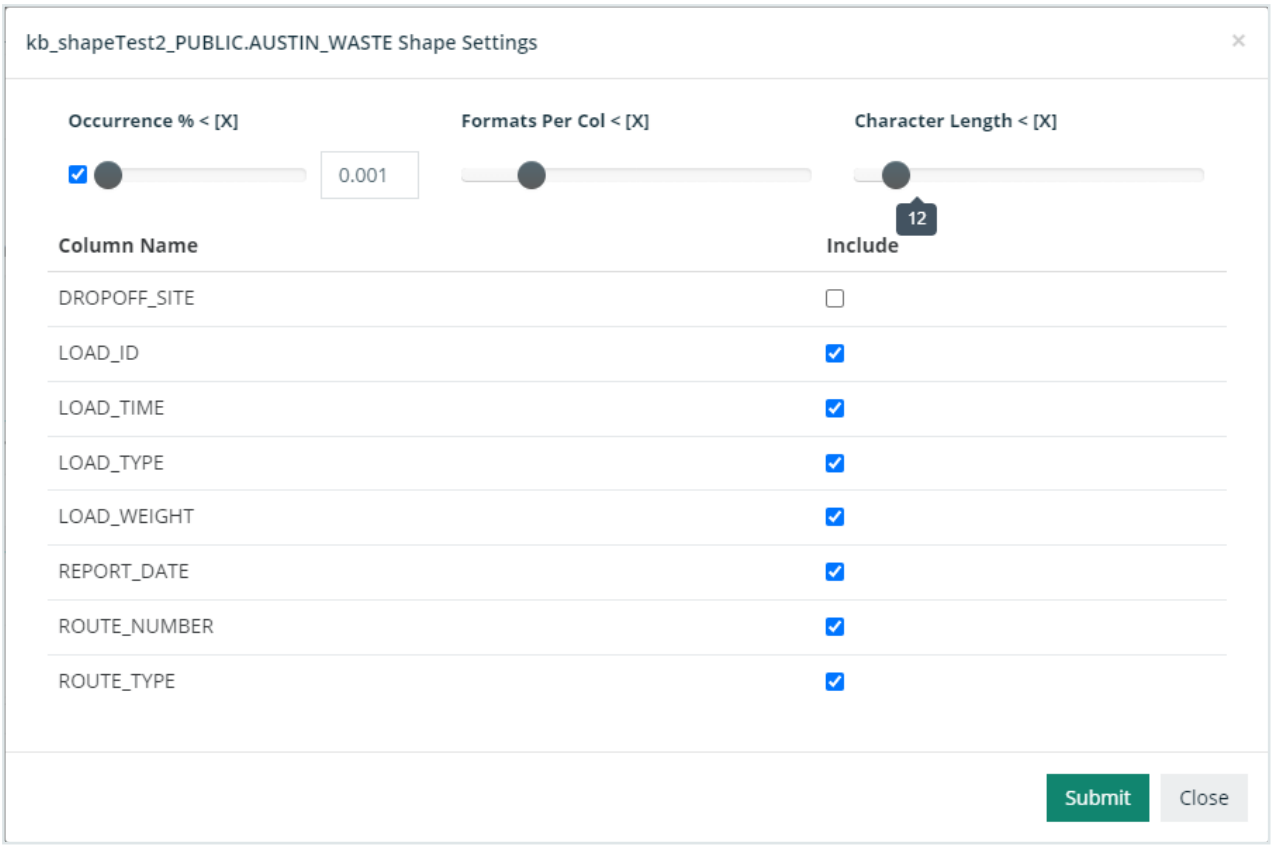
Drill-in to any Shape anomaly and see a visual example

See an itemized list view of the most infrequent or odd shapes in your datasets.



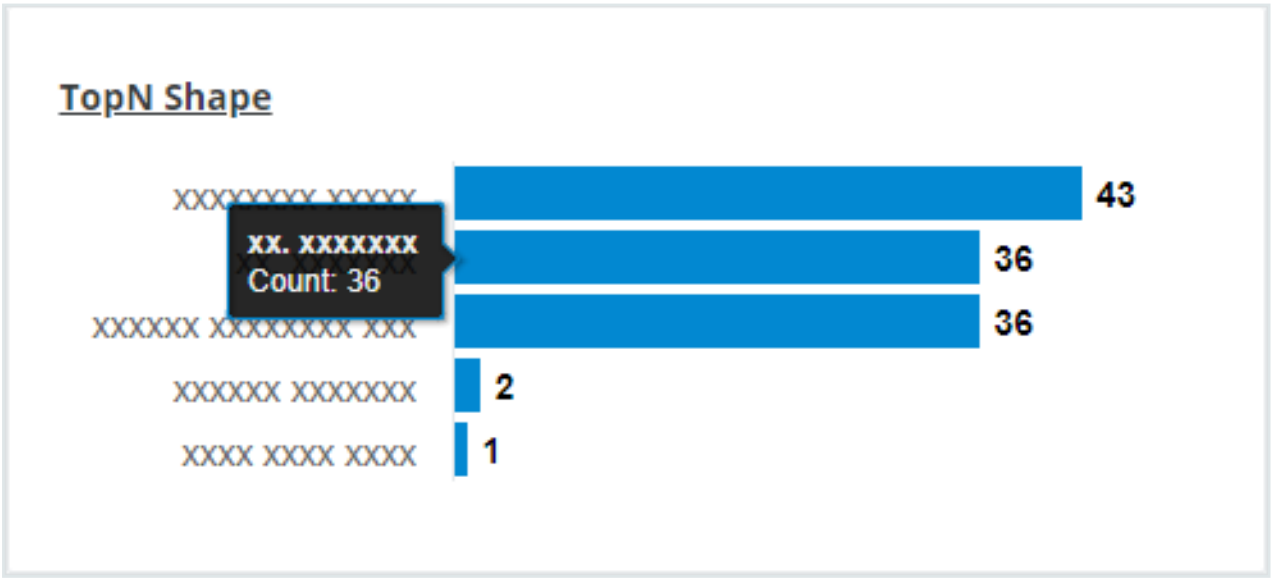
Shape Tuning

Shapes detection is on by default. Click the ⚙ on the Findings page to manually tune the Shape Settings. Click and drag the sliders to adjust the tuning of **Occurrence %**, **Format per Column**, and **Character Length**.



TopN Shape

Beneath the TopN Values chart on the Profile page is the **TopN Shape** chart. The TopN Shape chart represents the exceptional shapes for a particular column on the Shapes tab.



Dupes (advanced)

Dupes, or Duplicates, are values that match other existing values in columns and can be set to detect exact and fuzzy matches. Both exact and fuzzy support case-insensitive matching.

Exact matches detect entries that may match despite formatting issues. For example, Collibra DQ flags `John Doe`, `JOHN DOE`, and `john doe` as exact match dupes of each other. Exact match groups the same dupes and displays their counts in the **occurs** column on the Findings page.

Fuzzy matches detect entries that may match despite misspellings, common spelling variations, and more. For example, Collibra DQ flags `John Doe`, `Johnny Doe`, and `Johhnn Doe` as fuzzy match dupes of each other. The fuzzy match algorithm complexity is O^2 and compares every two fields in the dataset.

When fuzzy match runs on a dataset that is expected to have only exact matches, the dupes result will be greater than the size of the dataset because it compares every field with the rest of the fields on the list.

General Ledger. Accounting use-case

Whether you're looking for a fuzzy matching percent or single client cleanup, Collibra DQ's duplicate detection can help you sort and rank the likelihood of duplicate data.

Double Booked Entries

Journal entries that are booked twice in the ledger

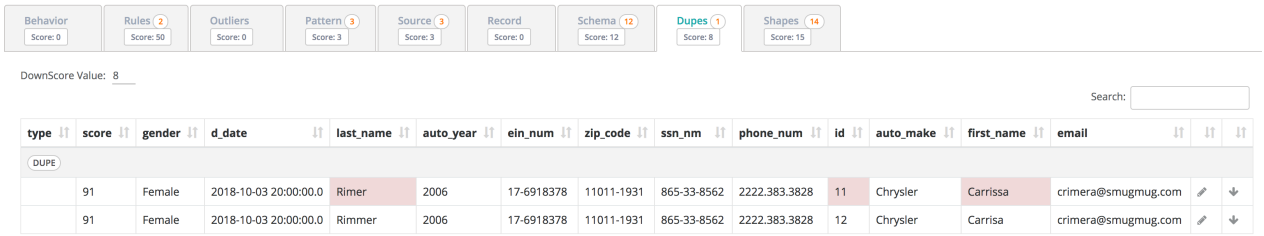
Owl's duplicate detection and fuzzy matching engine can track down hard to find nuanced bookings that are actually the same root booking.

JOURNAL_ID	ITEM	ACCOUNT	AMOUNT
JID-123	Swa flight-147	W2043	724.34
JID-629	Marriott	W2133	234.75
JID-456	Swa flight-147	W2043	724.34
JID-9022	Swa flight-822	W2043	1002.43

```
-f "file:///home/ec2-user/single_customer.csv" \
-d "," \
-ds customers \
-rd 2018-01-08 \
-dupe \
-dupenocase \
-depth 4
```

User Table has duplicate user entry

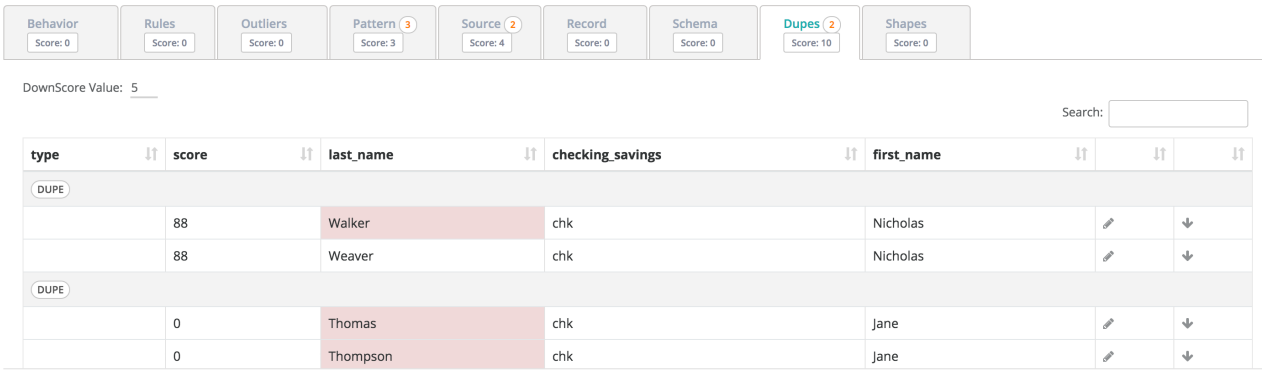
Carrisa Rimmer vs Carrissa Rimer



type	score	gender	d_date	last_name	auto_year	ein_num	zip_code	ssn_nm	phone_num	id	auto_make	first_name	email
DUPE													
	91	Female	2018-10-03 20:00:00.0	Rimer	2006	17-6918378	11011-1931	865-33-8562	2222.383.3828	11	Chrysler	Carrissa	crimera@smugmug.com
	91	Female	2018-10-03 20:00:00.0	Rimmer	2006	17-6918378	11011-1931	865-33-8562	2222.383.3828	12	Chrysler	Carrisa	crimera@smugmug.com

ATM customer data with only a 88% match

As you can see below, less than a 90% match in most cases is a false positive. Each dataset is a bit different, but in many cases you should tune your duplicates to roughly a 90+% match for interesting findings.



type	score	last_name	checking_savings	first_name
DUPE				
	88	Walker	chk	Nicholas
	88	Weaver	chk	Nicholas
DUPE				
	0	Thomas	chk	Jane
	0	Thompson	chk	Jane

Simple DataFrame Example

```
val l = Seq(
  ("Kirk", "Haslbeck", "2018-02-24 8:30:45", 18),
  ("Kirk", "Hasselbach", "2018-02-23 8:30:45", 11),
  ("Kirt", "Haslbeck", "2018-02-22 8:30:45", 10),
  ("Kirk", "hasselback", "2018-02-21 8:30:45", 12),
  ("kirk", "Haslbeck", "2018-02-20 8:30:45", 10)
)

val df = l.toDF("fname","lname", "app_date", "price")
val owl = new Owl(df).dupesDF
```

distance	records
0.16	DUPE [score: 83][Kirk,Hasselbach,2018-02-23 8:30:45] [Kirt,Haslbeck,2018-02-22 8:30:45]
0.083	DUPE [score: 91][Kirk,Haslbeck,2018-02-24 8:30:45] [Kirt,Haslbeck,2018-02-22 8:30:45]
0.083	DUPE [score: 91][Kirk,Haslbeck,2018-02-24 8:30:45] [Kirk,Hasselbach,2018-02-23 8:30:45]

Known Limitations

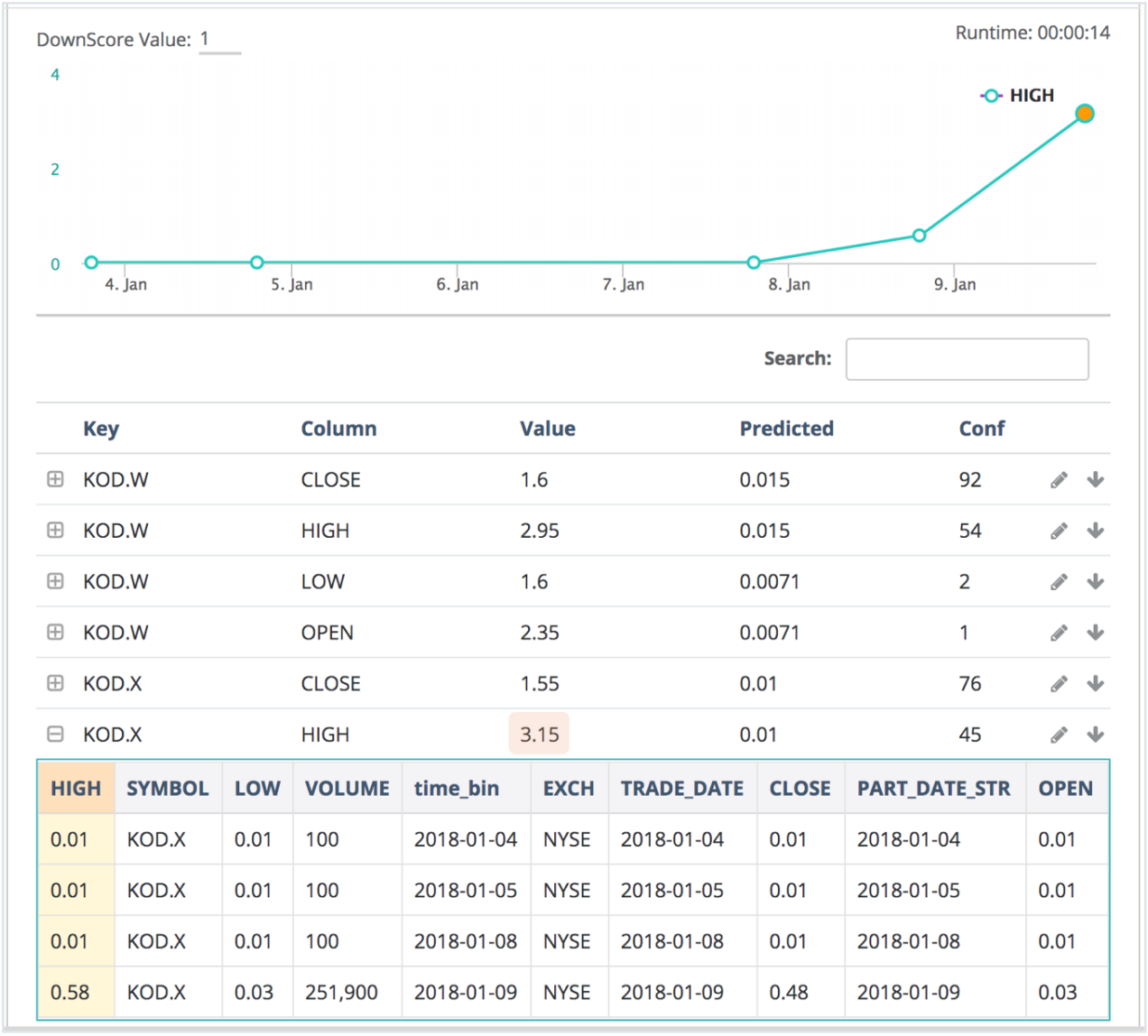
There is a known limitation where you cannot perform a dupes check on all columns or more than 10 columns at once.

Outliers (advanced)

> **Note** This is an advanced opt-in feature.

Numerical Outliers

Kodak Coin! In 2018 Kodak announced themselves as Kodak Coin and witnessed a steep change in their stock price. Collibra DQ automatically captured this event and provided the ability to drill into the item.

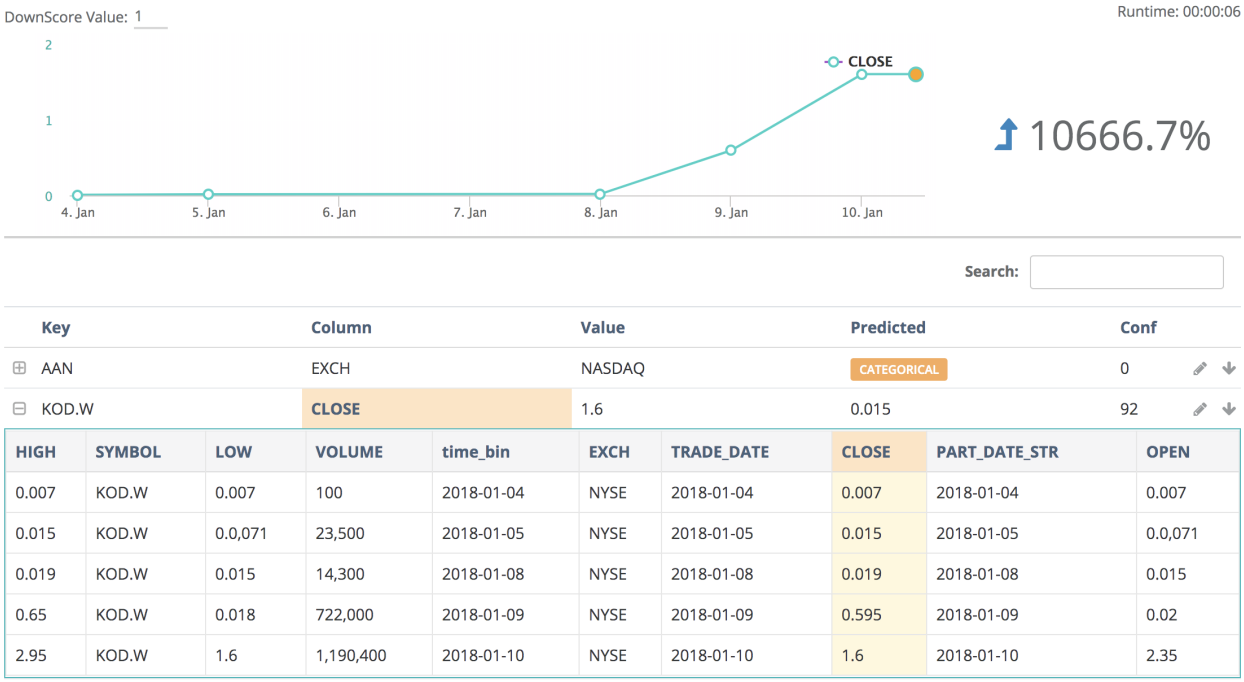


Key	Column	Value	Predicted	Conf
KOD.W	CLOSE	1.6	0.015	92
KOD.W	HIGH	2.95	0.015	54
KOD.W	LOW	1.6	0.0071	2
KOD.W	OPEN	2.35	0.0071	1
KOD.X	CLOSE	1.55	0.01	76
KOD.X	HIGH	3.15	0.01	45

HIGH	SYMBOL	LOW	VOLUME	time_bin	EXCH	TRADE_DATE	CLOSE	PART_DATE_STR	OPEN
0.01	KOD.X	0.01	100	2018-01-04	NYSE	2018-01-04	0.01	2018-01-04	0.01
0.01	KOD.X	0.01	100	2018-01-05	NYSE	2018-01-05	0.01	2018-01-05	0.01
0.01	KOD.X	0.01	100	2018-01-08	NYSE	2018-01-08	0.01	2018-01-08	0.01
0.58	KOD.X	0.03	251,900	2018-01-09	NYSE	2018-01-09	0.48	2018-01-09	0.03

Complex outliers made Simple

Even though Collibra DQ uses complex formulas to identify the correct outliers in a dataset, it uses simple terms when displaying them. If you notice below the change happened gradually, therefore if you only compared averages or previous values you would not understand the full impact of this price change. 0% changed from yesterday and its moving/trailing average would have caught up.



Key	Column	Value	Predicted	Conf
AAN	EXCH	NASDAQ	CATEGORICAL	0
KOD.W	CLOSE	1.6	0.015	92

HIGH	SYMBOL	LOW	VOLUME	time_bin	EXCH	TRADE_DATE	CLOSE	PART_DATE_STR	OPEN
0.007	KOD.W	0.007	100	2018-01-04	NYSE	2018-01-04	0.007	2018-01-04	0.007
0.015	KOD.W	0.0,071	23,500	2018-01-05	NYSE	2018-01-05	0.015	2018-01-05	0.0,071
0.019	KOD.W	0.015	14,300	2018-01-08	NYSE	2018-01-08	0.019	2018-01-08	0.015
0.65	KOD.W	0.018	722,000	2018-01-09	NYSE	2018-01-09	0.595	2018-01-09	0.02
2.95	KOD.W	1.6	1,190,400	2018-01-10	NYSE	2018-01-10	1.6	2018-01-10	2.35

Dynamic history options

Data may not always enter your data pipeline on time and as expected due to weekend, holidays, errors, etc. To help capture outliers in spite of gaps, there are two main options:

- 1) Extend the lookback period (to 10 days from 5 days, for example)
- 2) Utilize additional flags per below (fllbminrow new as of 2021.11)

Flag	Description	Example
fllbminrow	File Lookback Minimum Rows: determines minimum number of rows that a previous file scan needs to be counted as file lookback	-fllbminrow 1 (counts nay DQ scans with 1 or more row in minimum history) -fllbminrow 0 (default behavior, row count does not matter)
dllb	Date Lookback: determines how many days of learning	-dllb 5 (5 days)

Upper & Lower Bound Limits

Collibra DQ automatically detects and flags data that falls outside of preset **Upper Bound** or **Lower Bound** limits. If data is detected outside of these limits, an alert is generated to notify of an outlier. Setting these limits allows you to finetune your ability to identify outliers.

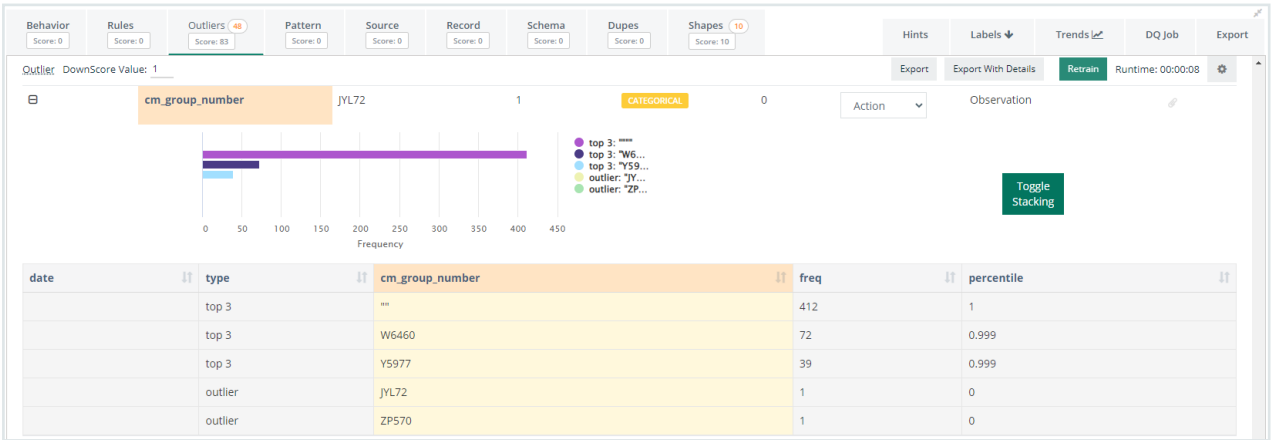
Categorical Outliers

Categorical Outliers are much different than numerical outliers and require separate techniques to automatically capture meaningful anomalies. The details regarding Collibra DQ's methodology and testing can be found below, 3 minute read on the topic.

Categorical Outliers Don't Exist

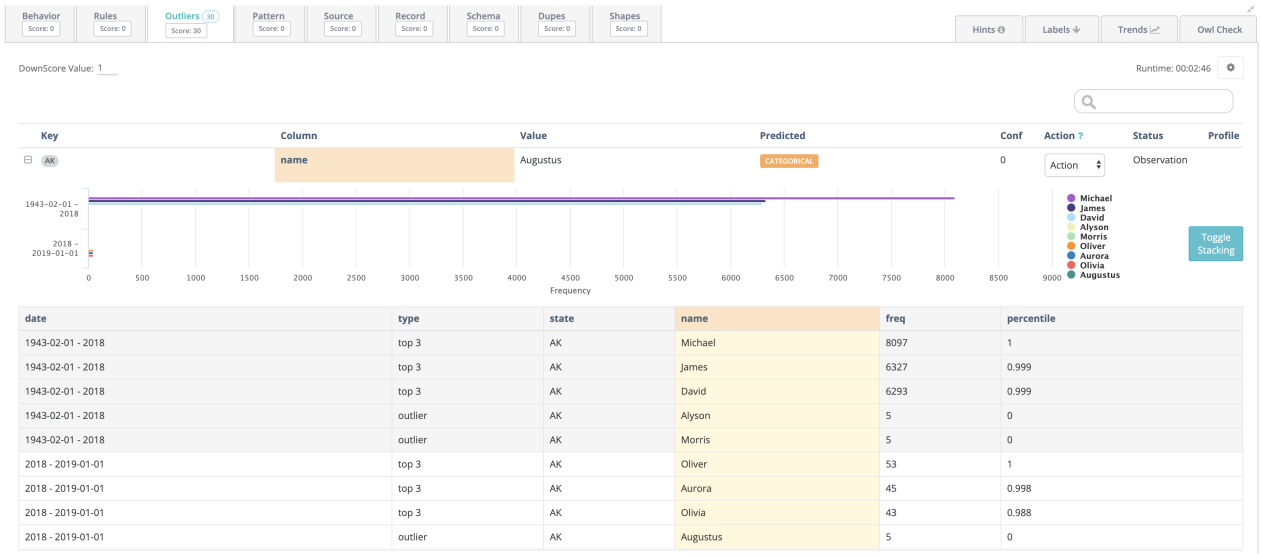
Collibra DQ will automatically learn the normal behavior of your String and Categorical attributes such as STOCK,OPTION,FUTURE or state codes such as MD,NC,D.C. When a strange pattern occurs (e.g NYC instead of NY), Collibra DQ will show this as a categorical outlier.

Collibra DQ is able to detect Categorical Outliers both with and without taking time into account. If a time dimension is not provided, Collibra DQ will calculate the distribution of categorical values within the available data, and identify the values that fall into the most infrequent percentile (configurable).



If a time dimension is provided, Collibra DQ will first identify infrequent categories in the historical context and then in the context of the current DQ Check. Only values that are

historically infrequent or non-existent, and are infrequent in the current run will be considered Outliers.

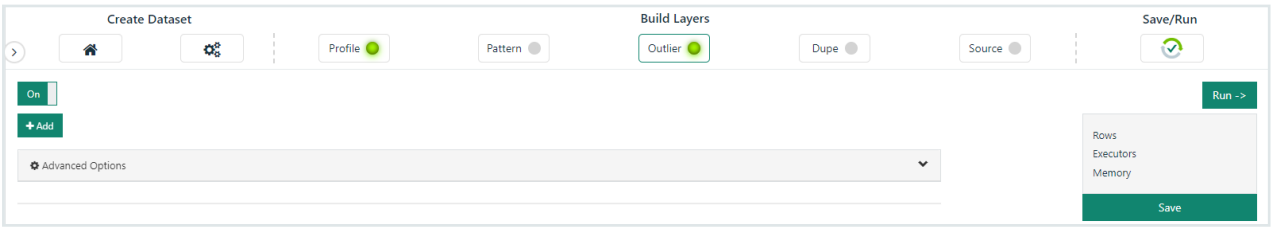


date	type	state	name	freq	percentile
1943-02-01 - 2018	top 3	AK	Michael	8097	1
1943-02-01 - 2018	top 3	AK	James	6327	0.999
1943-02-01 - 2018	top 3	AK	David	6293	0.999
1943-02-01 - 2018	outlier	AK	Alyson	5	0
1943-02-01 - 2018	outlier	AK	Morris	5	0
2018 - 2019-01-01	top 3	AK	Oliver	53	1
2018 - 2019-01-01	top 3	AK	Aurora	45	0.998
2018 - 2019-01-01	top 3	AK	Olivia	43	0.988
2018 - 2019-01-01	outlier	AK	Augustus	5	0

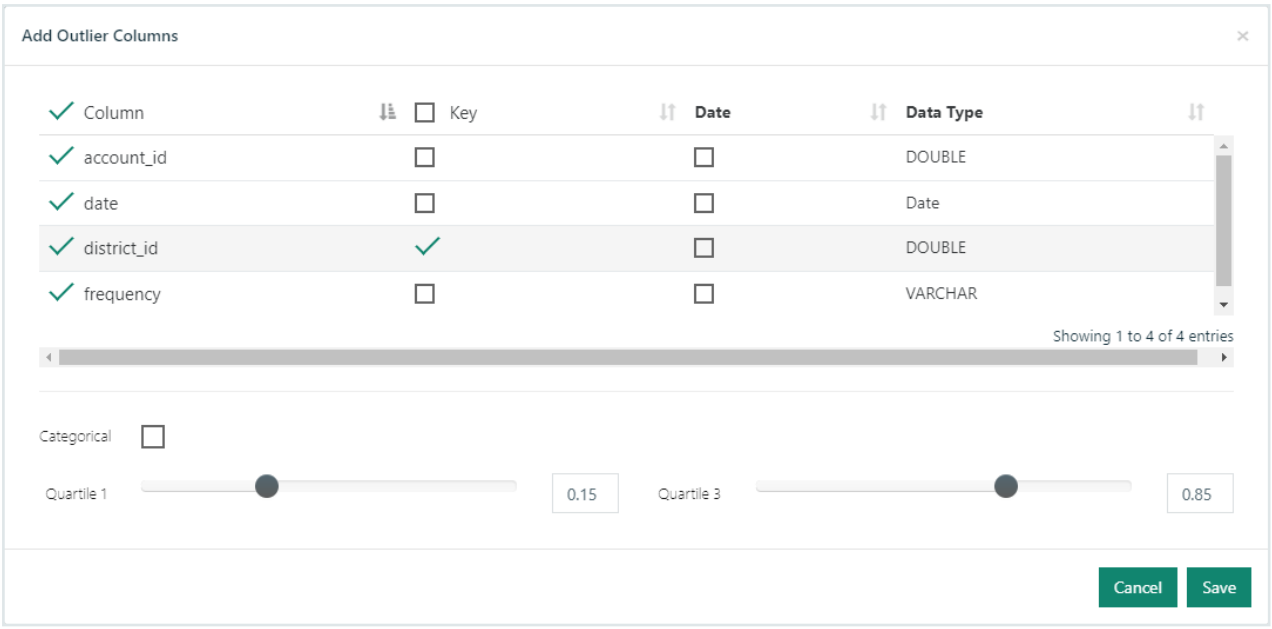
Training Outlier Detection Model

Although Collibra DQ uses different techniques to detect Numerical and Categorical Outliers, the training process is very similar.

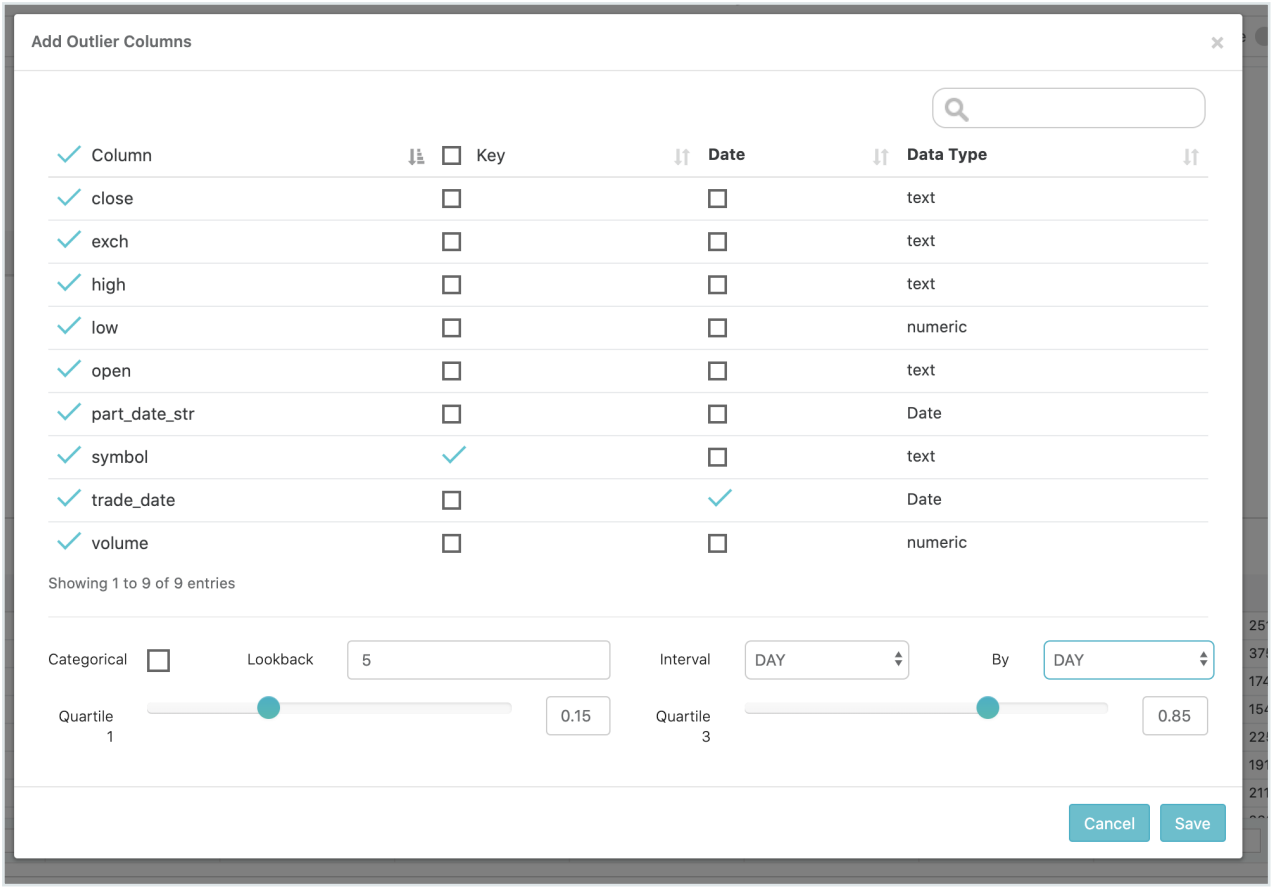
At a minimum, Collibra DQ requires historical data that can be used as the training dataset. If no other input is provided, Collibra DQ will calculate the normal range for each selected column and look for numerical and categorical outliers within the training dataset without any further context. The output will essentially consist of infrequent values that fall outside the normal range of each column.



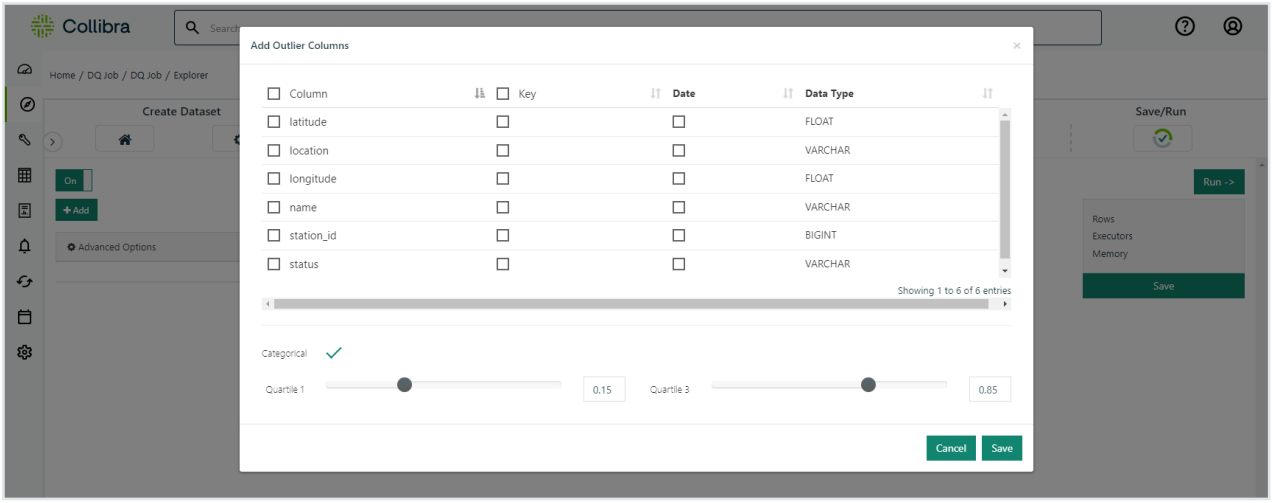
To obtain more targeted results, the Collibra DQ requires a "key" column. This column will be used to provide context by grouping each column by the key column. Defining a good key column tends to provide results that are a better indicators of actual data quality issues instead of simply infrequent values.



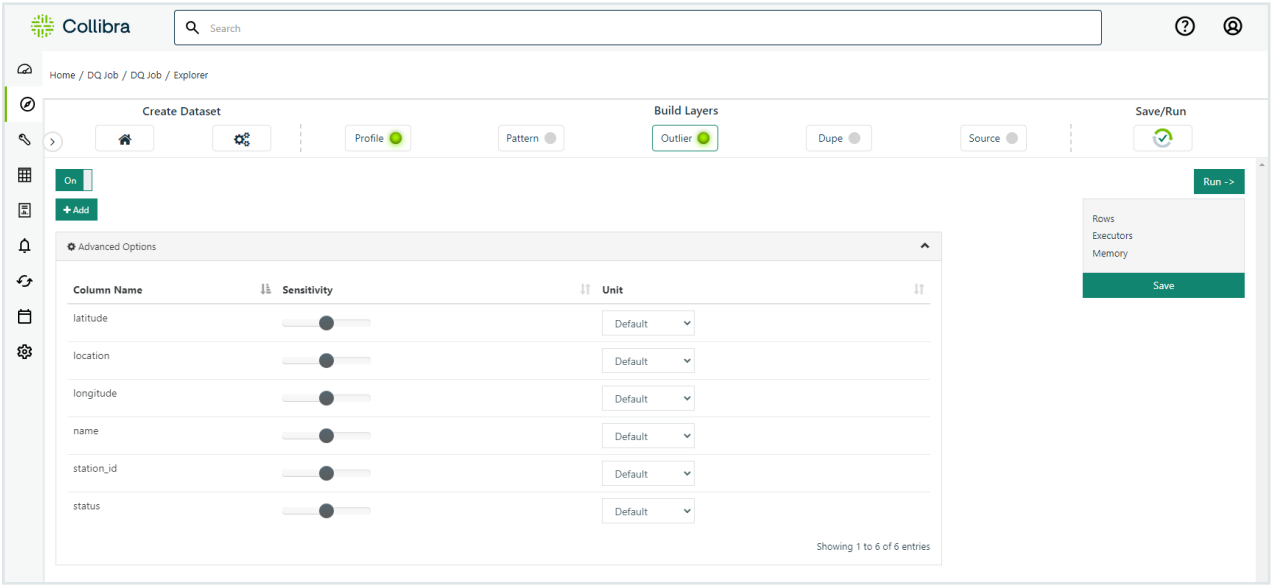
Another input that can make outlier detection more precise is a date/time column and a look back period. This enables a more precise calculation of the normal range for a column and in the case of numerical outliers, makes it possible for Collibra DQ to establish a trend. Given a time column and key column, Collibra DQ will not only identify numerical outliers, it will plot the historical trend of the column value trailing the outlier.



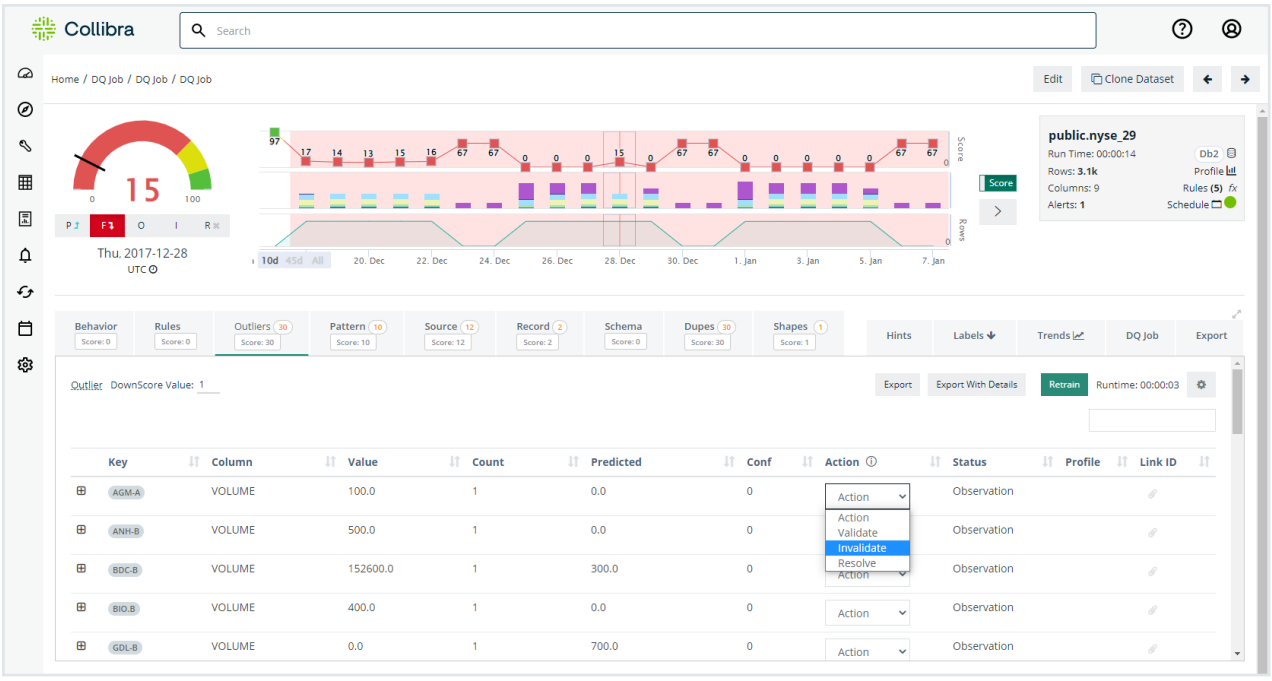
Collibra DQ also allows further refinement of the time dimension by defining time bins and processing intervals. By default, when given a time column, Collibra DQ will bin the data into days and process the data in daily interval. However, if the data is high frequency, day bins and day intervals might be too coarse grained. In this case, it might make more sense to group the data into bins on the minute and process the data in hour or minute intervals. The same concept applies in the other direction. What if the data is already aggregated on the month or year? In this case, it makes more sense to set the bins and intervals to month by month or year by month.



Some data may be measured in really small or large units or contain a lot of noise. In this case, Collibra DQ allows the user to adjust the sensitivity level and unit of measure for outlier detection on each column. Click the advanced tab to make these adjustments.



Once Outlier detection is complete for a given run, it's time to tune the scoring of the model. Collibra DQ allows the user to label any outlier findings as legitimate, thus preventing that outlier from being detected in the future or effecting the score of the current run. In addition, it is possible to define the significance of an outlier finding to a given dataset. This can be accomplished by setting how many quality points should be deducted for each outlier finding on any given run on that dataset. It is also possibly to adjust sensitivity and unit of measure of future runs by clicking on the small gear icon on the far left of the screen.



Spark DataFrame Example

Categorical Outliers

```
val assetClass =
Seq("STOCK","OPTION","FUTURE","STOCK","OPTION","FUTURE","STOCK","STOCK", "FUTURE",
    "STOCK","OPTION","FUTURE","STOCK","OPTION","FUTURE","STOCK","STOCK", "FUTURE",
    "STOCK","OPTION","FUTURE","STOCK","OPTION","FUTURE","STOCK","STOCK", "FUTURE",
    "STOCK","OPTION","FUTURE","STOCK","OPTION","FUTURE","STOCK","STOCK", "FUTURE",
    "STOCK","OPTION","FUTURE","STOCK","OPTION","FUTURE","STOCK","STOCK", "FUTURE",
    "STOCK","OPTION","FUTURE","STOCK","OPTION","FUTURE","STOCK","STOCK", "FUTURE",
    "SWAP", "OPTION", "STOCK","SWAP", "SWAP", "FUTRS", "STOCK", "OPTION","FUTURE",
    "OPTION", "STOCK" )

val df = assetClass.toDF("ASSET")
val outliers = new Owl(df).outliersDF("ASSET")
```

Key	Column	Value	Prediction	Confidence
	ASSET	FUTRS	FUTURES	

Real World Example

Imagine you are the data manager at Iowa Department of Commerce, Alcoholic Beverage Division. As part of the Department's open data initiative, the monthly Iowa liquor sales data are available to the public for analysis. (Thank you Iowa!)

An Iowan data analyst emails you about a data quality issue with **address** for store #2508 in the year 2016. You quickly run a SQL query on your data warehouse to see what is going on.

```
-- Assuming Postgres DB
select date_trunc('MONTH', "date") "date_month", address, count
(*) "sales_count"
from iowa_liquor_sales
where "date" >= '2016-01-01' and "date" < '2017-01-01' and
store_number = '2508'
group by date_trunc('MONTH', "date"), address
order by date_month, address
```

date_month	address	sales_count
2016-01-01 00:00:00	1843 JOHNSON AVENUE, N.W.	422
2016-02-01 00:00:00	1843 JOHNSON AVENUE, N.W.	451
2016-03-01 00:00:00	1843 JOHNSON AVENUE, N.W.	579
2016-04-01 00:00:00	1843 JOHNSON AVENUE, N.W.	404
2016-05-01 00:00:00	1843 Johnson Avenue, N.W.	625
2016-06-01 00:00:00	1843 Johnson Avenue, N.W.	695
2016-07-01 00:00:00	1843 Johnson Avenue, N.W.	457
2016-08-01 00:00:00	1843 Johnson Avenue, N.W.	744
2016-09-01 00:00:00	1843 Johnson Avenue, N.W.	681
2016-10-01 00:00:00	1843 Johnson Avenue, N.W.	728
2016-11-01 00:00:00	1843 Johnson Avenue, N.W.	1062
2016-12-01 00:00:00	1843 Johnson Avenue, N.W.	992

Because `store_number` is an unique number assigned to the store who ordered the liquor, the inconsistent `address` values for the same store pose data quality problem. But `address` is a string value that can take many forms. For store #2508, the reported address value has a shifted behavior from all capital letters starting on May 2016. For other cases, it might be completely different behavior change that you would have to manually check one by one. With

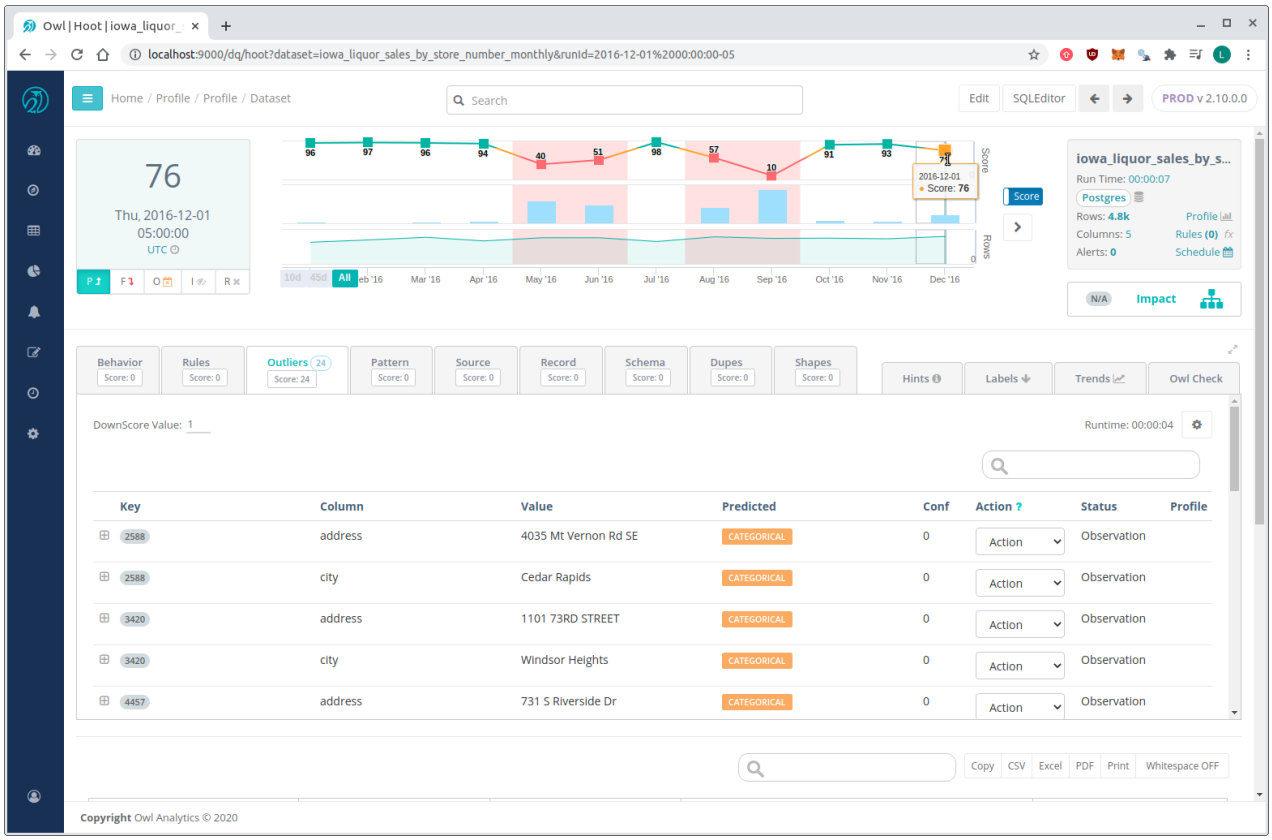
over 2,000 unique stores, 19 million rows, and 8 years of data, you need an automated way to detect meaningful categorical outliers.

The following command shows an example of running monthly DQ Checks, from the month of Jan 2016 to the month of December 2016. Each monthly run looks back 3 months of data to establish a baseline for categorical columns that you suspect would have similar data quality issues: `store_name`, `address`, and `city`.

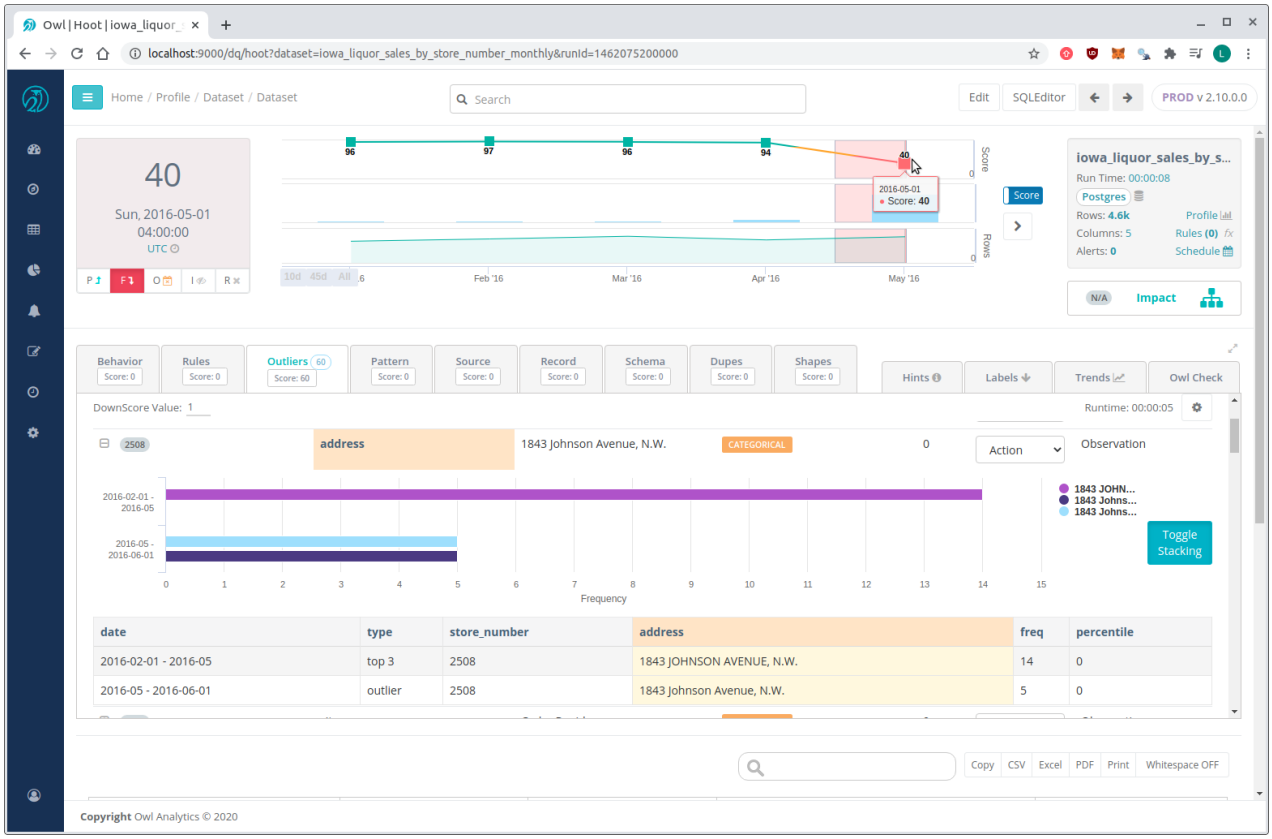
```
/opt/owl/bin/owlcheck
  # connection information to data
  -lib "/opt/owl/drivers/postgres/" -driver "org.-
postgresql.Driver"
  -c, "jdbc:postgresql://localhost:5432/postgres"
  -u, "postgres", "-p", "password"
  # Specify dataset name
  -ds "iowa_liquor_sales_by_store_number_monthly"
  # Specify date filter for the last filter, e.g. date >= '2016-
12-01' and date < '2017-01-01'
  -rd "2016-12-01" -rdEnd "2017-01-01"
  # SQL query template (${rd} and ${rdEnd} matches with -rd and
-rdEnd
  -q "select distinct on (date, store_number) date, store_num-
ber, store_name, address, city
       from iowa_liquor_sales where date >= '${rd}' and date <
'${rdEnd}' "
  # Turn on Outliers
  -dl
  # Group on store_number (optional if no grouping)
  -dlkey "store_number"
  # Specify column that is of date type (optional, if running DQ
Check without time context)
  -dc "date"
  # Specify columns to run Outlier analysis (if not specified,
all the columns in query are included in analysis)
  -dlinc "store_name,address,city"
  # Specify 3 month lookback for each DQ Check
  -dllb 3
  # Run Monthly DQ Check
  -tbin "MONTH"
  # "backrun" Convenient way to run 12 preceding MONTHly DQ
check
  -br 12
```

Results

The `-br 12` option ran 12 monthly DQ Checks for every month of 2016. The figure below shows the findings page for the latest run of dataset `iowa_liquor_sales_by_store_numbers_monthly`. The Findings page shows that DQ Check identified 24 Outliers among 4.8k rows of unique date x store_number for month of December, 2016.



Since the original data quality issue that inspired us to run DQ Check is from May 2016, we can navigate to specific run date 2016-05-01 by click on the line graph on top. Then searching for store #2508 on the **key** column shows outlier detected for **column** `address`. Press [+] for that row to see contextual details about this detected value.



We can verify that DQ Check identified the outlier of interest among other 60 data quality issues. Using DQ Check, you can identify issues at scale for past data (using backrun), current (using simple DQ Check), and future (using scheduled jobs).

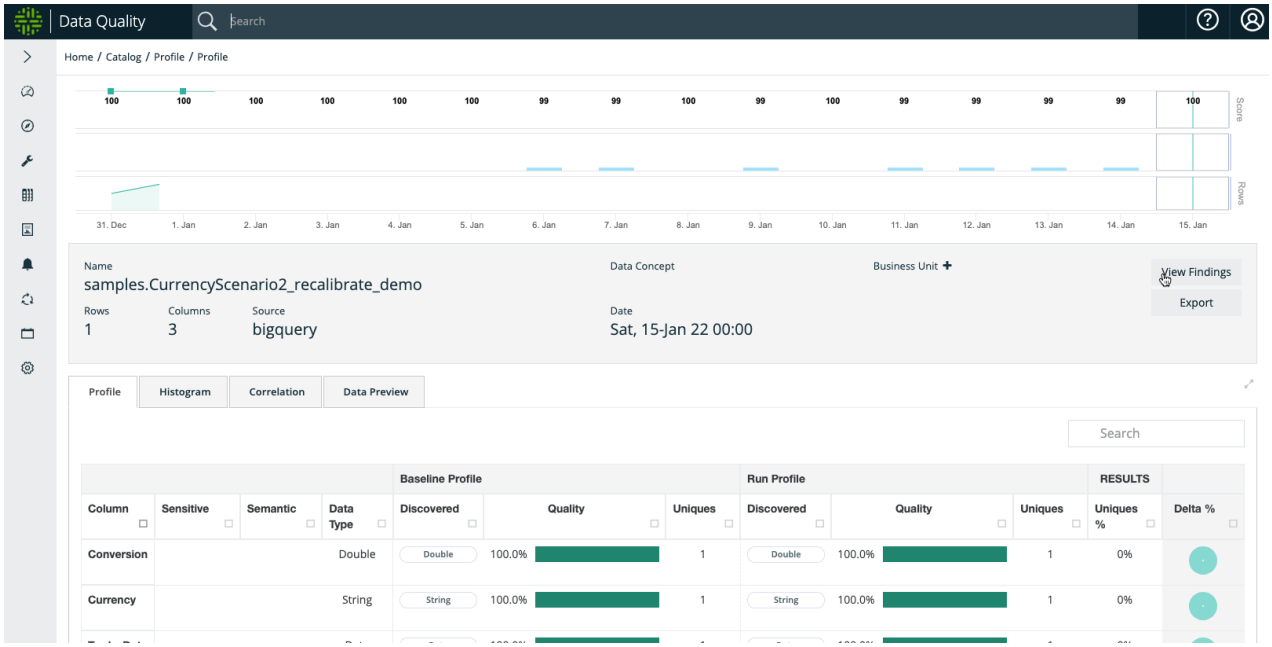
Tech Preview [TP] Outlier Calibration

Use Case

When A) using Outliers and B) faced with an event such as a stock split or currency devaluation, it may be helpful to calibrate the outlier boundaries within Collibra DQ to avoid surfacing non-issues for a period of time.

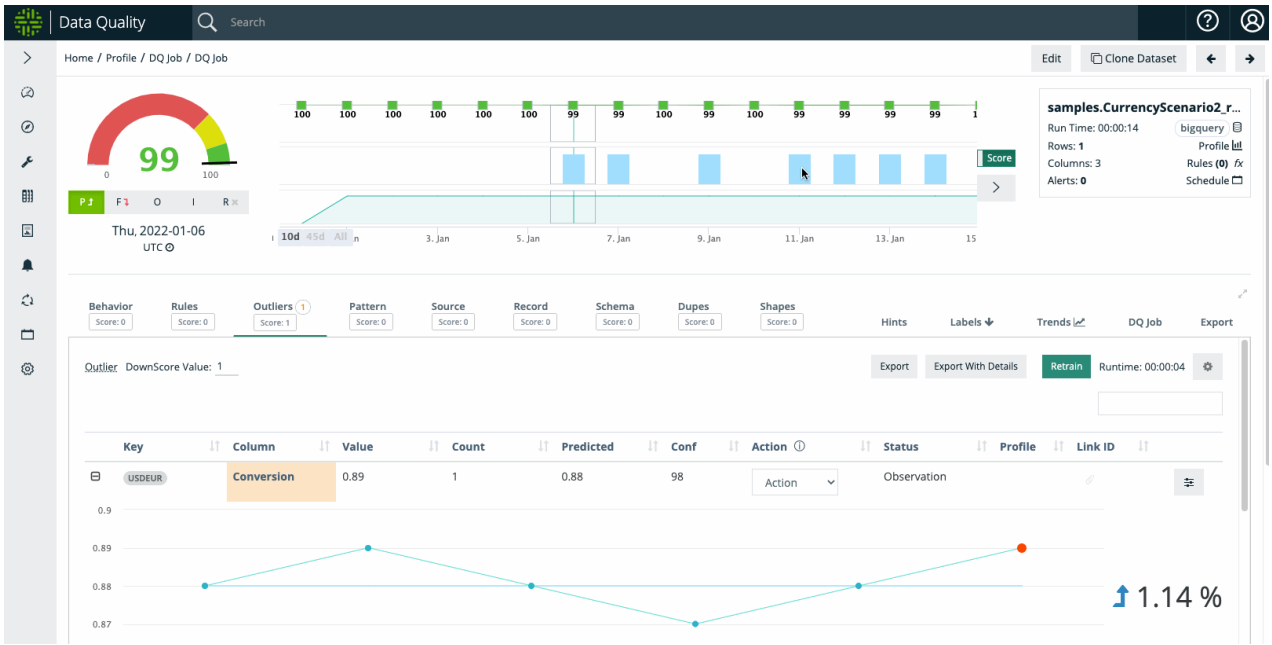
Example Step #1: No Action Necessary

In the video below, Collibra DQ Outliers were set to a high sensitivity. The USDEUR conversion rate on January 6th in the sample dataset may be considered reasonable and the user can 1) rerun the dataset with lower sensitivity or 2) downtrain the unintended Outlier anomalies.



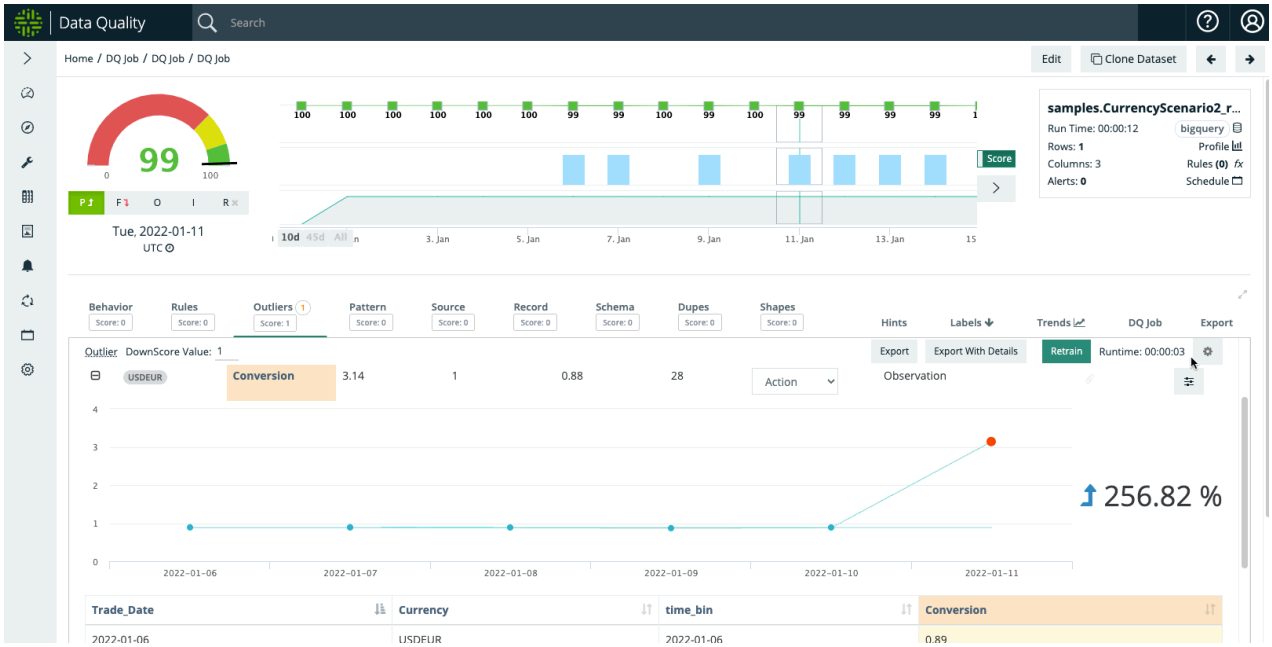
Example Step #2: Macro Event That User Understands e.g. Currency Devaluation or Stock Split

When examining the outlier on January 11th, the dataset depicts that the USDEUR conversion shot up to 3.14, which in our hypothetical example coincides with an explainable macroeconomic phenomenon. As such, the user may not want Outlier anomalies to trigger for a period of time.



Example Step #3: User Wants To Suppress Outliers

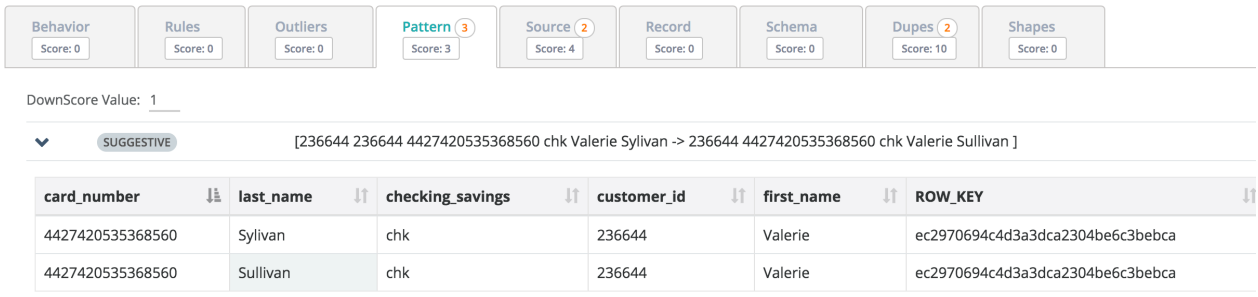
Once Outlier Calibration is enabled, a user can select the boundaries and duration of the 'suppression' period. And once the DQ Job is re-run for the selected date(s), the outliers will not trigger an anomaly / downscore.



Patterns (advanced)

> **Note** This is an advanced opt-in feature.

Collibra DQ uses the latest advancements in data science and ML to find deep patterns across millions of rows and columns. In the example below it noticed that Valerie is likely the same user as she has the same customer_id and card_number but recently showed up with a different last name. Possible misspelling or data quality issue?

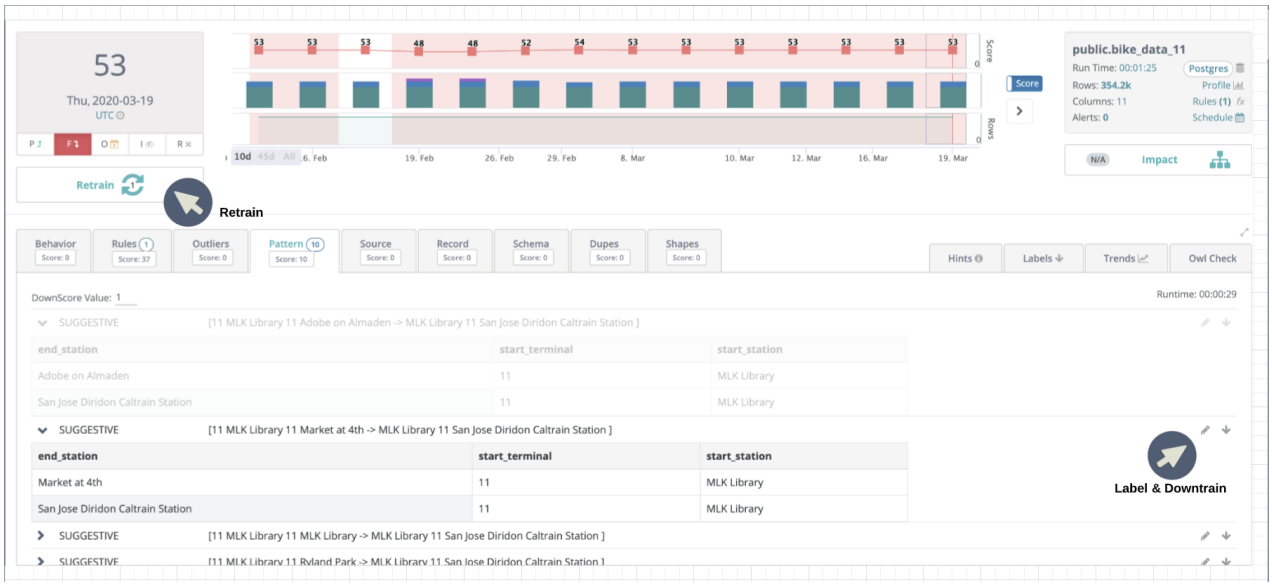


Training Anti-Pattern Detection Model

When the Patterns feature is enabled, DQ builds a collection of patterns that it identifies within the data. It will then use that collection to identify values that break established patterns. For example, in the image below, DQ learned that a bike route that starts at "MLK library" will end at "San Jose Diridon Caltrain Station". However, when the current day's data cross referenced against this pattern, DQ detects an anti-pattern where a trip starts at "MLK Library" but ends at "Market at 4th". DQ raises this anti-pattern as a data quality issue and highlights the what it believes the "end_station" value should have been.

To build a Pattern model, DQ requires historical data that contains the valid patterns and if possible, a date/time column. The user can then needs to define the date/time column, the look back period, and what columns make up the pattern. In the image below, the pattern was composed of "end_station", "start_terminal", "start_station".

It is possible that an apparent anti-pattern finding is actually valid data and not a data quality issue. In this case, DQ allows the user to further instruct the existing Patterns model on how to properly score and handle the findings. For example, if it turns out that "Market at 4th" is actually a valid "end_station" for a bike trip, the user can negate the identified anti-pattern by labeling it as valid. This action instructs DQ to not raise this particular anti-pattern again.
DQ also rescores the current DQ check results to reflect the user's feedback. In addition, it is possible to define the weight of an anti-pattern finding on the current dataset by setting the numerical value to deduct per finding.



Fraud Detection?

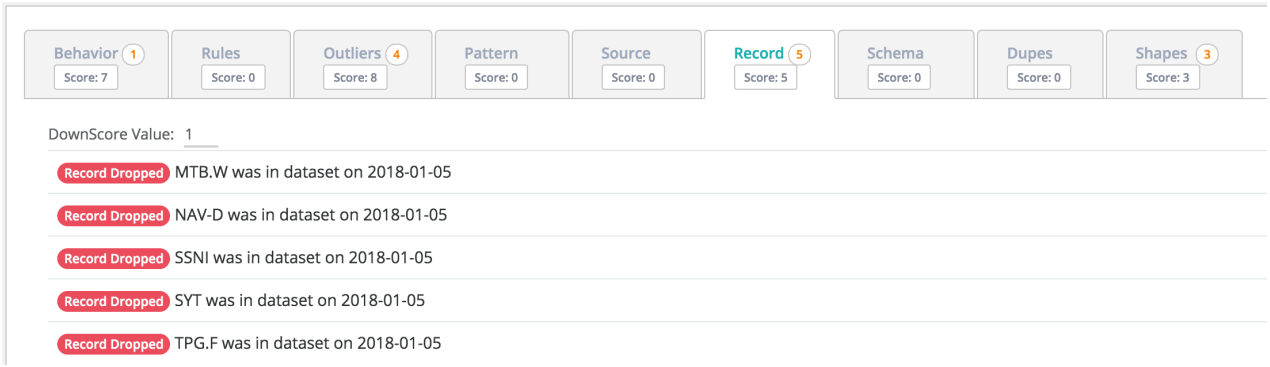
Think about a scenario where a dataset has a SSN column along with FNAME, LNAME and many others. What if your traditional rules engine passes because one of the rows has a valid SSN and a valid Name but the SSN doesn't belong to that person (his or her name and address, etc.)? This is where data mining can derive more sophisticated insights than a rules based approach.

Records (advanced)

> **Note** This is an advanced opt-in feature.

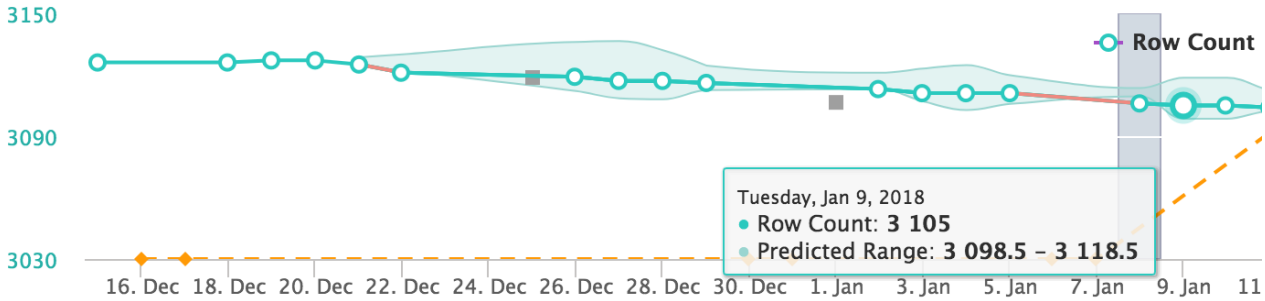
Where did my rows go?

Collibra DQ is constantly learning which records or rows in a dataset are most common. In the case below the NYSE had a reasonable dataset volume (row count).



Row Count Trend

We can see the rows dipping just slightly outside their predicted range. Arguably a subtle drop, yet abnormal to not represent these companies that typically do trade on the NYSE. Were they de-listed?



Source (advanced)

> **Note** This is an advanced opt-in feature.

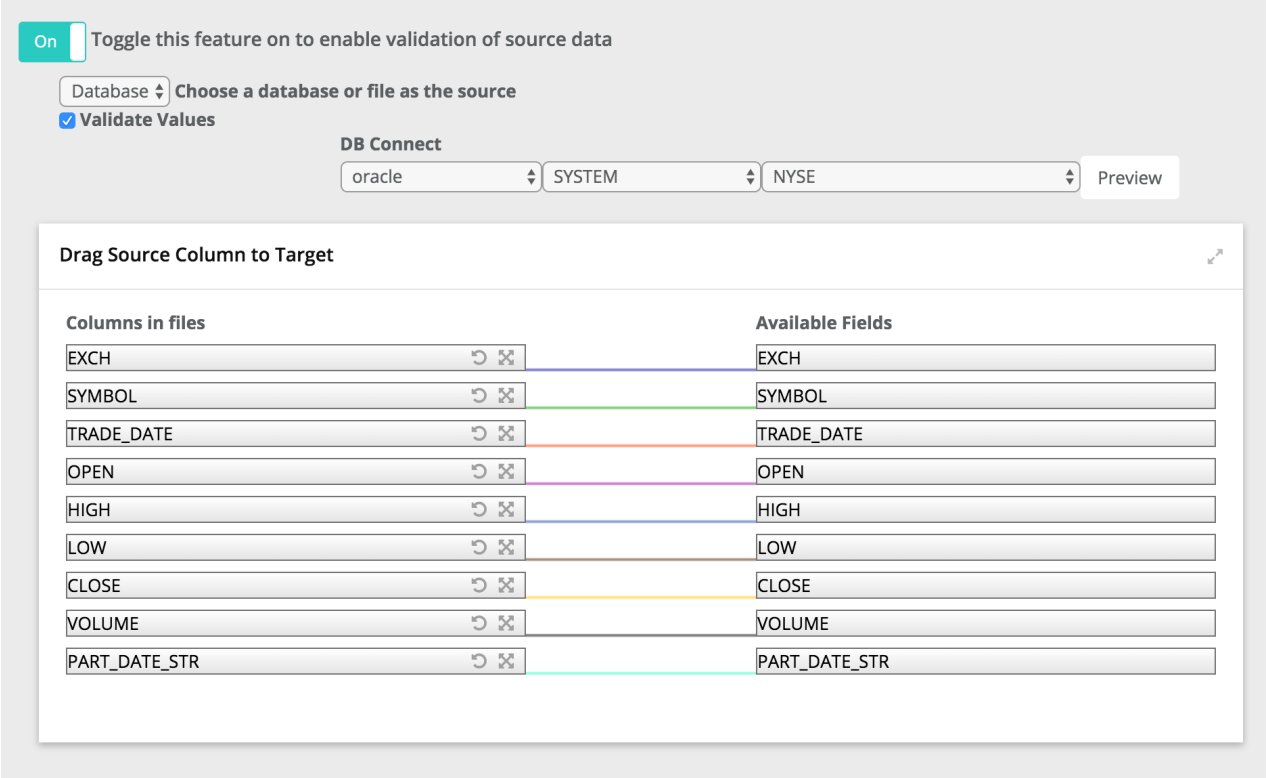
Does your data-lake reconcile with your upstream system?

Copying data from one system to another is probably the most common data activity to all organizations. Collibra DQ refers to this as source to target. As simple as this activity sounds, DQ has found that most of the time files and database tables are not being copied properly. To ensure and protect against target systems getting out of sync or not matching the originating source, turn on `-vs` to validate that the source matches the target.

A row count is not enough...

The most common check we encounter is a row count. However, a row count does not account for:

- Schema differences - Boolean to Int, Decimal to Double with precision loss, Timestamps and Dates.
- Value differences - Char or Varchars with whitespace vs Strings, null chars, delimiter fields that cause shifting, and much more.



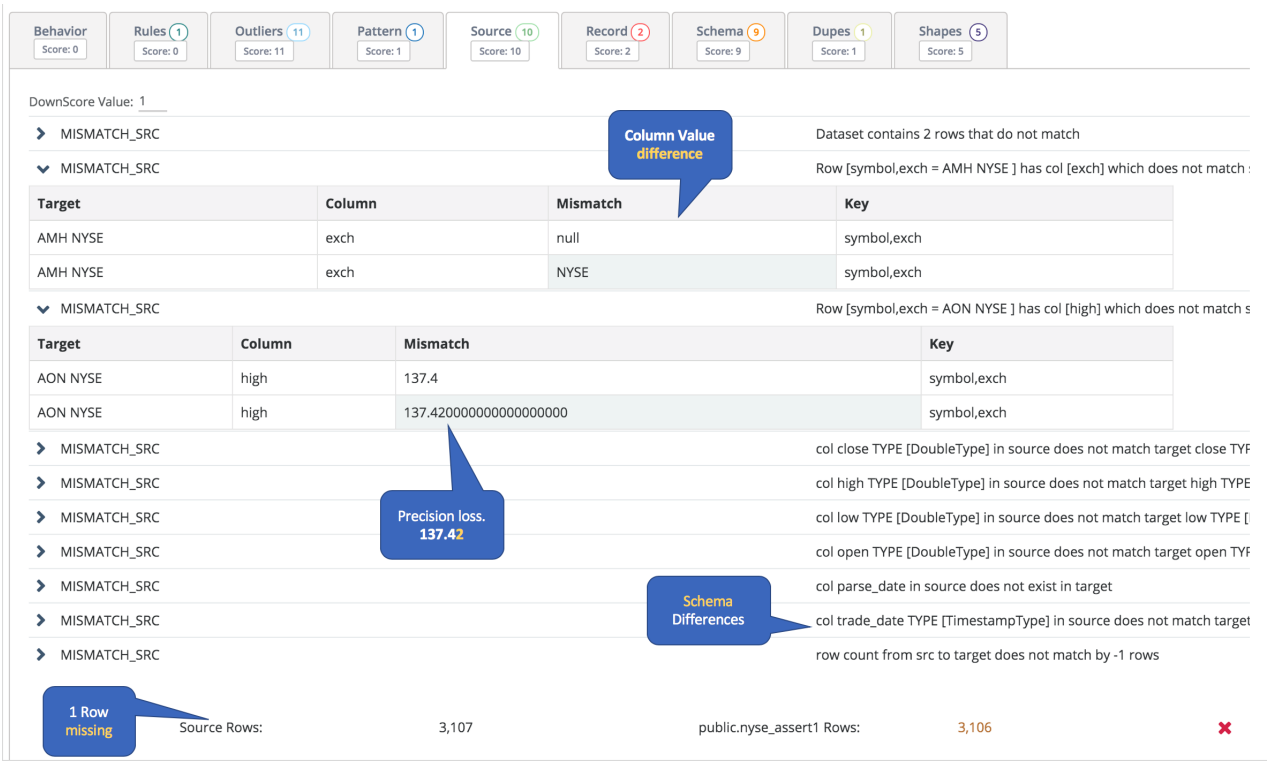
DQ Check Created from Wizard

The DQ Wizard GUI creates the below DQ Check which it can execute from the GUI by clicking RUN or by pasting at the cmdline.

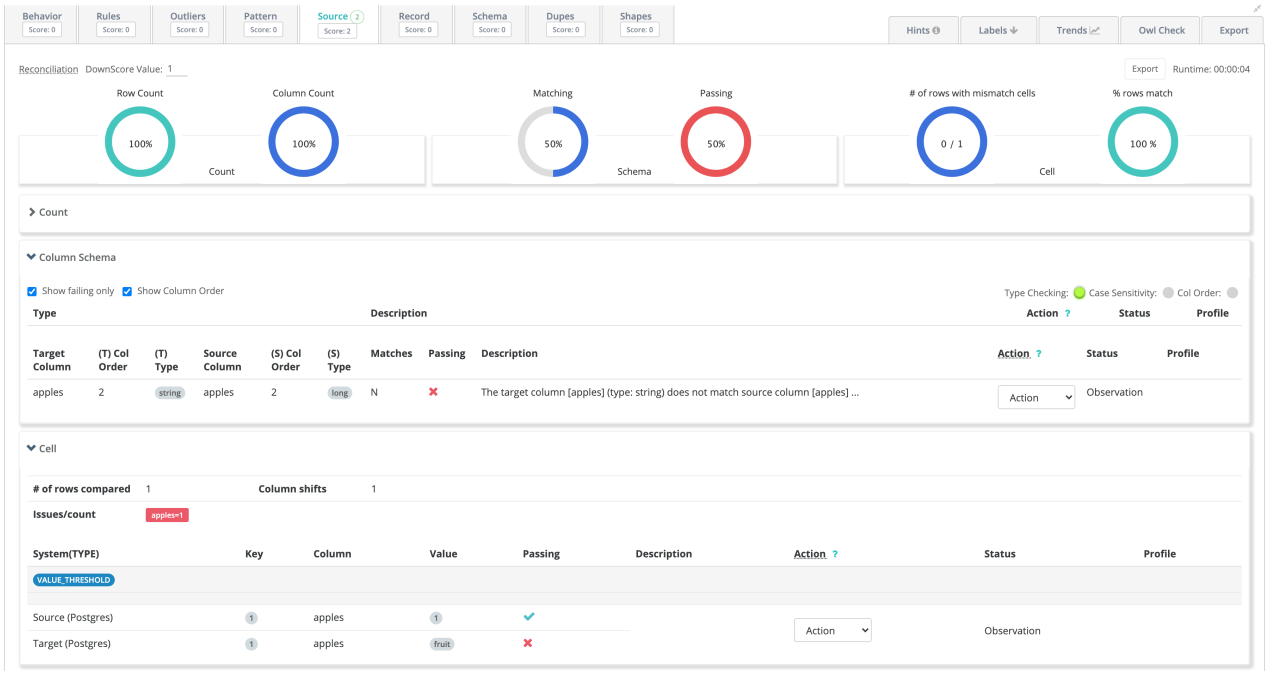
```
-lib /home/ec2-user/owl/drivers/valdrivers \
-driver org.postgresql.Driver \
-u user -p password \
-c "jdbc:postgresql://ec2-34-227-151-67.compute-1.amazon-
aws.com:5432/postgres" \
-q "select * from public.dateseries4" \
-ds psql_dateseries2 -rd 2018-11-07 \
-srcq select dz, sym as symz, high as highz, low as lowz, close
as closez, volume as volumez, changed as changedz, changep as
changepz, adjclose as adjclosez, open as openz from lake.d-
ateseries \
-srcu user \
-srcp password \
-srcds mysqlSYMZ \
-srcd com.mysql.cj.jdbc.Driver \
-srcc "jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306/lake" \ -valsrckey "SYMZ" \
-vs \
-valsrcinc "dz,symz,open-
z,highz,lowz,closez,volumez,changedz,changepz,adjclosez"
```

End of Day Stock Data from Oracle to Mysql

In this example we loaded NYSE_EOD data in both Oracle and Mysql and then used Collibra DQ's Source Validation feature. We see 3 main classes of issues. 1) The row count is off by 1 row, this means a row was dropped or went missing when the data was copied. 2) The schemas don't exactly match. 3) In 2 cases the values are different at the cell level. NULL vs NYSE and 137.4 vs 137.42.



Latest View in 2.13+



Known Limitations

- Datasets with column names that contain spaces or non-standard characters are unsupported.
    - A possible workaround for SQL Server connections is to set the dataset with column names that contain spaces or non-standard characters as the source and use bracket syntax to work with specific column names.