Collibra

Collibra Platform Self-Hosted

Collibra Data Quality & Observability

Version 2024.02

Collibra

Collibra Platform Self-Hosted - Collibra Data Quality & Observability

Release date: February 4, 2024 / Revision date: February 26, 2024

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

https://productresources.collibra.com/doc/collibra/latest/Content/DataQuality/to_data-quality.htm

Contents

Contents ii

Introducing Collibra Data Quality vi

Getting Started ix

Release Notes xi

Release 2024.02 xi

Release 2024.01 xxi

Release 2023.11 xxxiii

Release 2023.10 lii

Release 2023.09 lxii

Builds cclix

APIs cclxviii

Collibra DQ Installation cclxxi

Collibra DQ Connections ..... cdxiii

Before you begin ..... cdlvi

Prerequisites ..... cdlvi

Steps ..... cdlvii

Workload Identity ..... cdlviii

Prerequisites ..... cdlix

1. Authenticate to API ..... cdlix

2. Create a connection ..... cdlx

3. Verify the connection ..... cdlxi

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

Collibra DQ Features ..... dlxiii

Requirements ..... dlxxxix
Configuring Pushdown for Databricks ..... dxc
Available features ..... dxcii
What's next? ..... dxciii
Running the Snowflake setup script ..... dxciii
Setting up the Snowflake Connection template ..... dxcvii
Available features ..... dxcix
Known limitations ..... dc
What's next? ..... dc
Requirements ..... dci
Configuring Pushdown for BigQuery ..... dci
Available features ..... dciv
What's next? ..... dciv
Requirements ..... dcv
Configuring Pushdown for Athena ..... dcv
Available features ..... dcvii
What's next? ..... dcviii
Known limitations ..... dcviii
Requirements ..... dcix
Configuring Pushdown for Redshift ..... dcix
Available features ..... dcxi
What's next? ..... dcxi
Known limitations ..... dcxii
Requirements ..... dcxii
Configuring Pushdown for Trino ..... dcxiii
Available features ..... dcxv

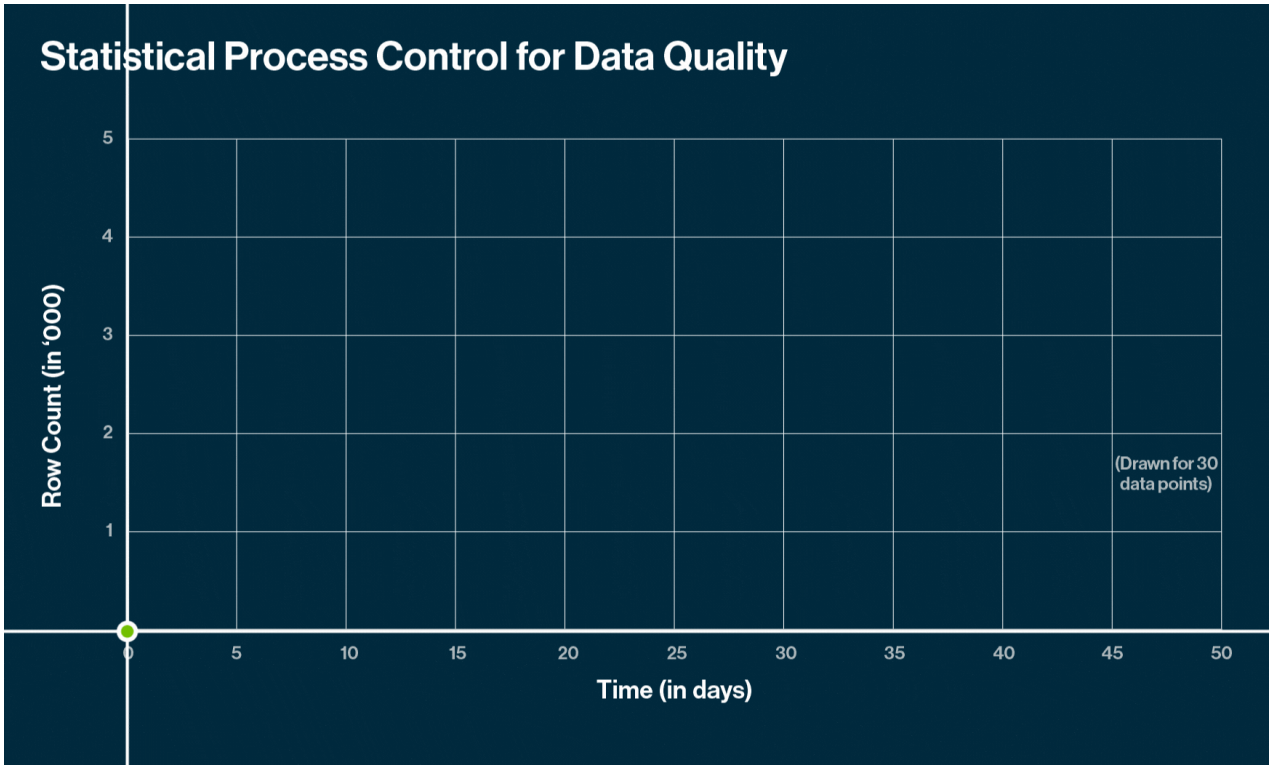
What's next? ..... dcxvi
Known limitations ..... dcxvi
Steps ..... dclxxxv
Known limitations ..... dclxxxviii
Prerequisites ..... dccvi
Using Parallel JDBC ..... dccvi
Known limitations ..... dccviii
Break record details ..... dccx
Prerequisites ..... dccxi
Steps ..... dccxiii
Setting up break record archival ..... dccxv
Viewing break record details ..... dccxix
Prerequisites ..... dcccv
Using [[[Undefined variable DataQuality.SQL assistant]]] ..... dcccvi
FAQ ..... dcccxvii
Known Limitations ..... dcccxxii
Cross-Data Set Rules ..... dcccxxiii
In-Clause (Single Column) ..... dcccxxiii
Except (Multi-Column) ..... dcccxxiii
Referencing secondary data sets ..... dcccxxiv
Joins ..... dcccxxiv
Sample Results ..... dcccxxv
Known Limitations ..... dcccxxvi
Data Type ..... dcccxxxi
Running a Full Scan ..... mxx
Running a Partial Scan ..... mxxi

Scanning with SQL Query ..... mxxii
Collibra DQ Workflows ..... mxxiii
Collibra DQ DIC Integration ..... mxl
Collibra DQ Solutions ..... mlxx
Collibra DQ Benchmarks ..... mcxix
Collibra DQ APIs ..... mcxxxi
Rule types ..... mccxxii
Simple rule ..... mccxxiii
Example #1 ..... mccxxiii
Example #2 ..... mccxxvi
Freeform SQL ..... mccxxxii
Individual statement ..... mccxxxii
Join statements ..... mccxxxiii
Data type based rules ..... mccxxxvi
Rule types ..... mccxxxvi
Syntax ..... mccxxxvii
Outliers ..... mccxxxviii
Collibra DQ Architecture ..... mcccxxxv
Collibra DQ Admin ..... mcccli
What is Livy? ..... mccclxxviii
Livy architecture within Collibra DQ ..... mccclxxx
Supported deployment types ..... mccclxxxii
Supported remote file data sources ..... mccclxxxii
Troubleshooting Livy ..... mccclxxxv
Collibra DQ Security Configuration ..... mcccxcv
Collibra DQ Legal ..... mcdxlviii

Introducing Collibra Data Quality

Automatic data quality without the need for rules. Collibra Data Quality & Observability provides a fast and elegant way to manage your datasets by learning through observation rather than human input.

Collibra DQ applies the latest advancements in data science and machine learning to the problem of data quality, surfacing data issues in minutes instead of months.


Statistical Process Control for Data Quality
Row Count (in '000)
5
4
3
2
1
(Drawn for 30 data points)
0
5
10
15
20
25
30
35
40
45
50
Time (in days)

Getting Started with Collibra DQ

A Pluggable and Complete Data Quality Framework

If you are adding data quality to your data pipelines the below visual illustrates the number of products and pieces you will need consider to successfully complete your overall governance program. The Collibra DQ suite allows you to use either native Collibra DQ components or integrate the 3rd party components of your choice. By using our best practice guide and framework you can easily complete the DQ lifecycle.

Data Pipelines that Tie into the DQ Framework

Collibra DQ offers a coding framework for developers or ETL designers that want to built real-time data quality into their broader data pipeline. This provides the same algos and DQ checks as the Collibra DQ UI Wizard but with direct access into your code points. Consistency is a must to have a program you trust.

Getting Started

Standard Install

Standard	The install script for a standard install. *All software and data resides in your VPC.* Install time: `7 minutes`	Click Here
Containerized	Uses Helm Charts and K8s. For users who want WebApp and Compute components in containers. Compute pools are supported. *All software and data resides in your VPC.* Install time: `12 minutes`	Click Here

GCP Install

Marketplace	The Google Marketplace option is a simple, 1-click image installation. This is the quickest option for single-server install in GCP. *All software and data resides in your VPC.* Install time: `5 minutes`	Click Here
Google Cloud Deployment Manager	Hooks into GCP cloud services like GKE for ephemeral compute and RDS for cloud database. *All software and data resides in your VPC.*	Coming soon

AWS Install

Marketplace	The Amazon Marketplace option is a simple, 1-click image installation. This is the quickest option for single-server install in AWS Cloud. *All software and data resides in your VPC*. Install time: `5 minutes`	Click Here
CloudFormation	Hooks into AWS cloud services like EKS for ephemeral compute and RDS for cloud database. *All software and data resides in your VPC*. Install time: `12 minutes`	Click Here

Please see the Agreements for the terms and conditions on Collibra's evaluation offerings.

> Tip For more information, please contact **info@collibra.com**

Release Notes

Important
Disclaimer - Failure to upgrade to the most recent release of the Collibra Service may adversely impact the security, reliability, availability, integrity, performance or support (including Collibra's ability to meet its service levels) of the Service. Collibra hereby disclaims all liability, express or implied, for any reduction in the security, reliability, availability, integrity, performance or support of the Service to the extent the foregoing would have been avoided had you allowed Collibra to implement the most current release of the Service when scheduled by Collibra. Further, to the extent your failure to upgrade the Service impacts the security, reliability, availability, integrity or performance of the Service for other customers or users of the Service, Collibra may suspend your access to the Service until you have upgraded to the most recent release.

Release 2024.02

- Highlights
- Migration Updates
- Enhancements
- Beta Features
- Fixes
- Known Limitations
- DQ Security

Release Information

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

Highlights

Archive Break Records

[[[Undefined variable DataQuality.Pullup Spark processing]]]
When rule breaks are stored in the PostgreSQL [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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

Old Key	Renamed Key
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 [[[Undefined variable DataQuality.DQEnv]]] 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 [[[Undefined variable DataQuality.DQ 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.

Beta Features

Collibra AI

- [[[Undefined variable DataQuality.SQL assistant]]] (beta) 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 Data Intelligence Cloud for private beta 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.

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 Data Intelligence Cloud. (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

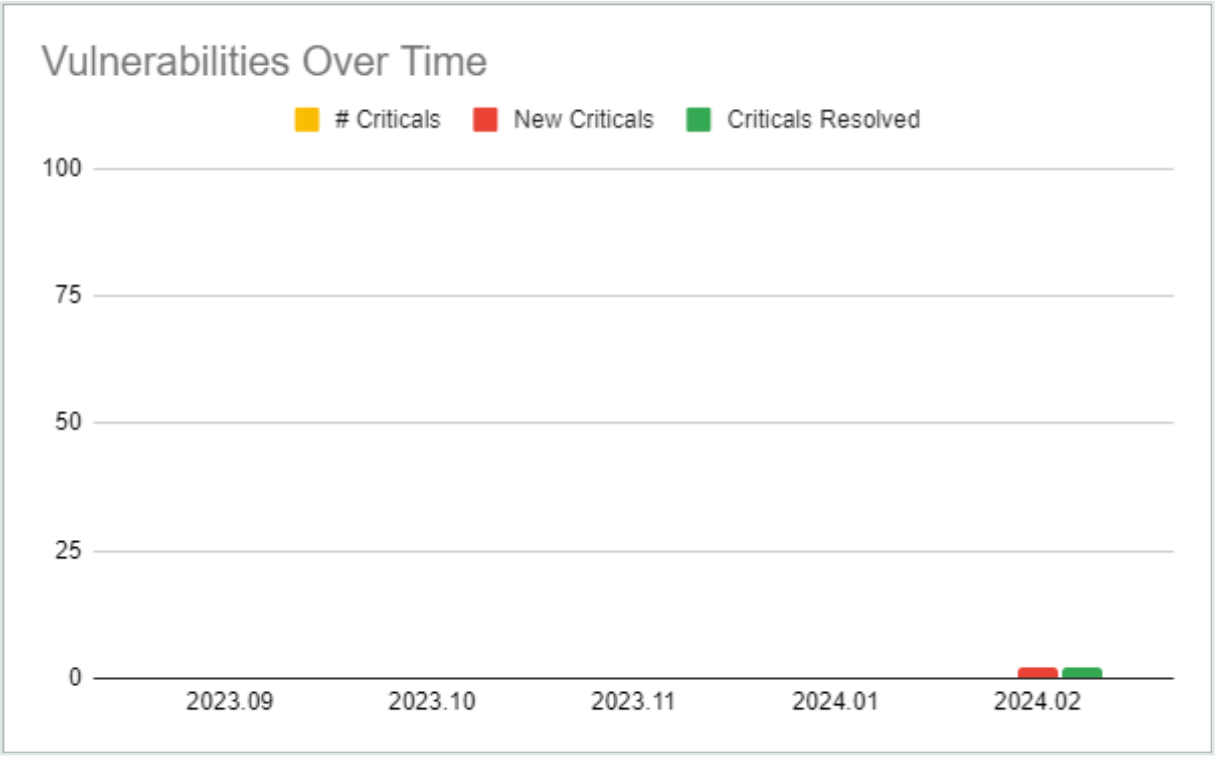
> Note If your current Spark version is 3.2.2 or older, we recommend upgrading to Spark 3.4.1 to address various critical vulnerabilities present in the Spark core library, including Log4j.

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 Collibra Data Quality & Observability 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 Data Intelligence Cloud for private beta 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 [[[Undefined variable DataQuality.DQEnv]]] 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 [[[Undefined variable DataQuality.DQEnv]]] 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.
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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 Data Intelligence Cloud.
- When a User Defined Rule is deactivated or deleted from a dataset in Collibra DQ, then the Rule and Score status in Collibra Data Intelligence Cloudwill 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

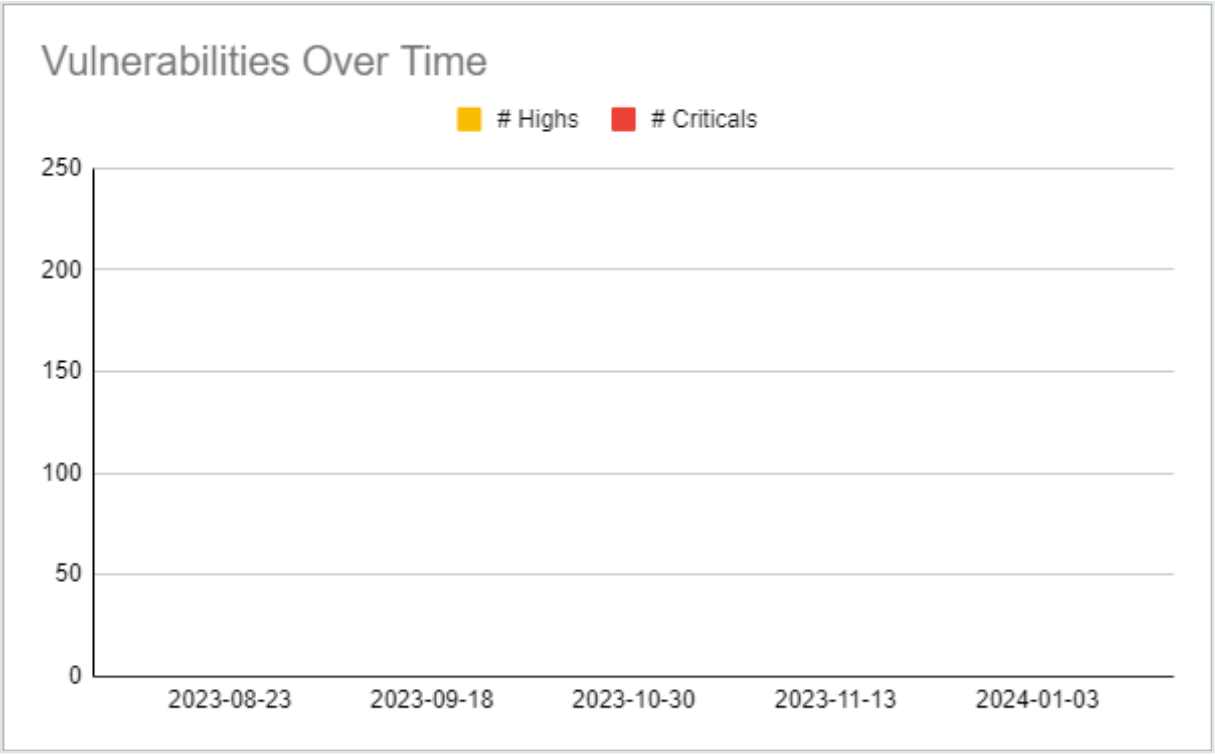
- When configuring `dupelimitui`, values above 2000 result in some UI lag. Because of this, we recommend using `dupelimitui` values under 2000.

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 Collibra Data Quality & Observability 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 beta 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 beta.

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 [[[Undefined variable DataQuality.DQEnv]]] 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 Data Intelligence Cloud. 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 Data Intelligence Cloud is enabled, you can now view the Community, Sub-Community, and Domain hierarchy to which the integration is mapped in Collibra Data Intelligence Cloud from the metadata bar on all dataset-level pages. This increases the transparency of where Assets are created in Collibra Data Intelligence Cloud 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 Data Intelligence Cloud.
- When running Pushdown jobs with integrations enabled, the results now integrate into Collibra Data Intelligence Cloud 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 Data Intelligence Cloud.
- 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 Data Intelligence Cloud 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 Data Intelligence Cloud, the columns now display correctly in Collibra Data Intelligence Cloud 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.

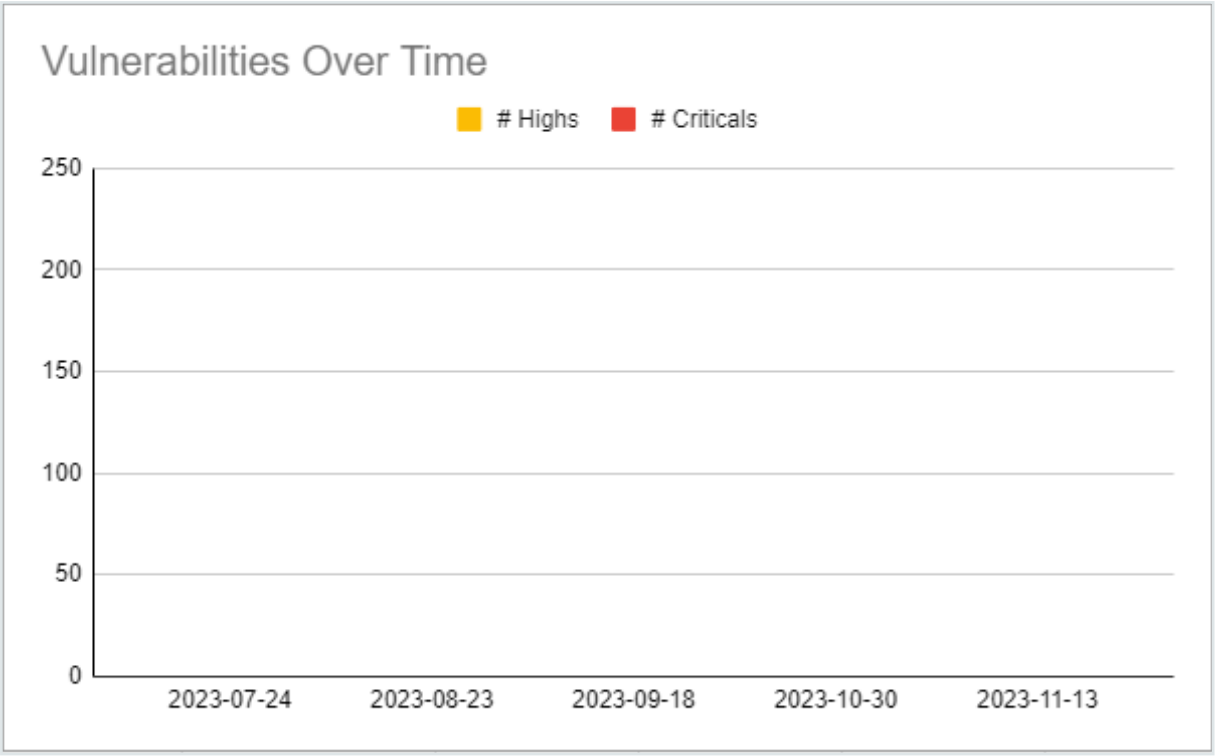
DQ Security

> 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 Data Intelligence Cloud. 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 beta 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 beta 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 beta 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

Component	Description	Available in Classic
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
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

Component	Description	Available in Classic
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 beta 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
- Beta UI
- Beta UI Limitations

Release Information

- Expected release date of Collibra Data Quality & Observability 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 beta 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 [[[Undefined variable DataQuality.DQEnv]]] 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 Data Intelligence Cloud 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.

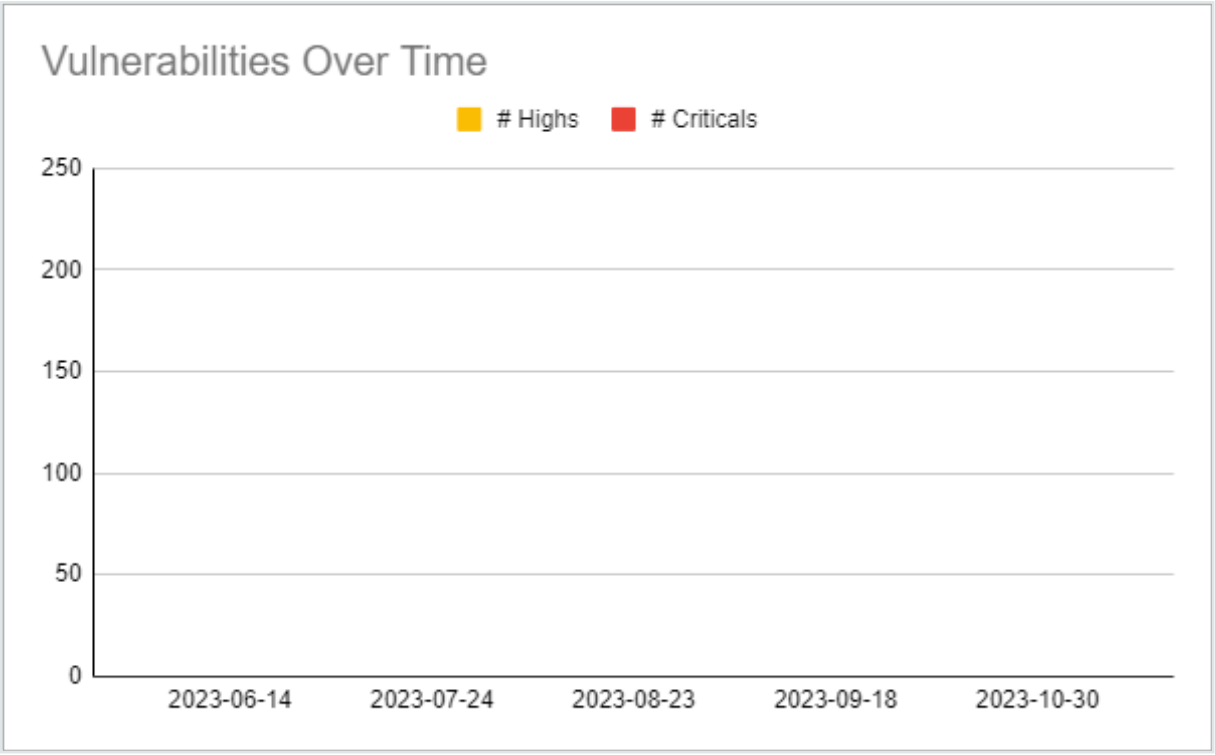
DQ Security

> 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.



Beta UI

Beta UI Status

The following table shows the status of the Beta 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

Page	Location	Status
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
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 Beta UI.

Beta 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
- Beta UI
- Beta UI Limitations

Release Information

- Expected release date of Collibra Data Quality & Observability 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 betas! 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 [[[Undefined variable DataQuality.DQEnv]]] 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 Data Intelligence Cloud 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 Data Intelligence Cloud integration, you now correctly see database views from Collibra DQ to the tables and columns to which they relate in Collibra Data Intelligence Cloud. (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.

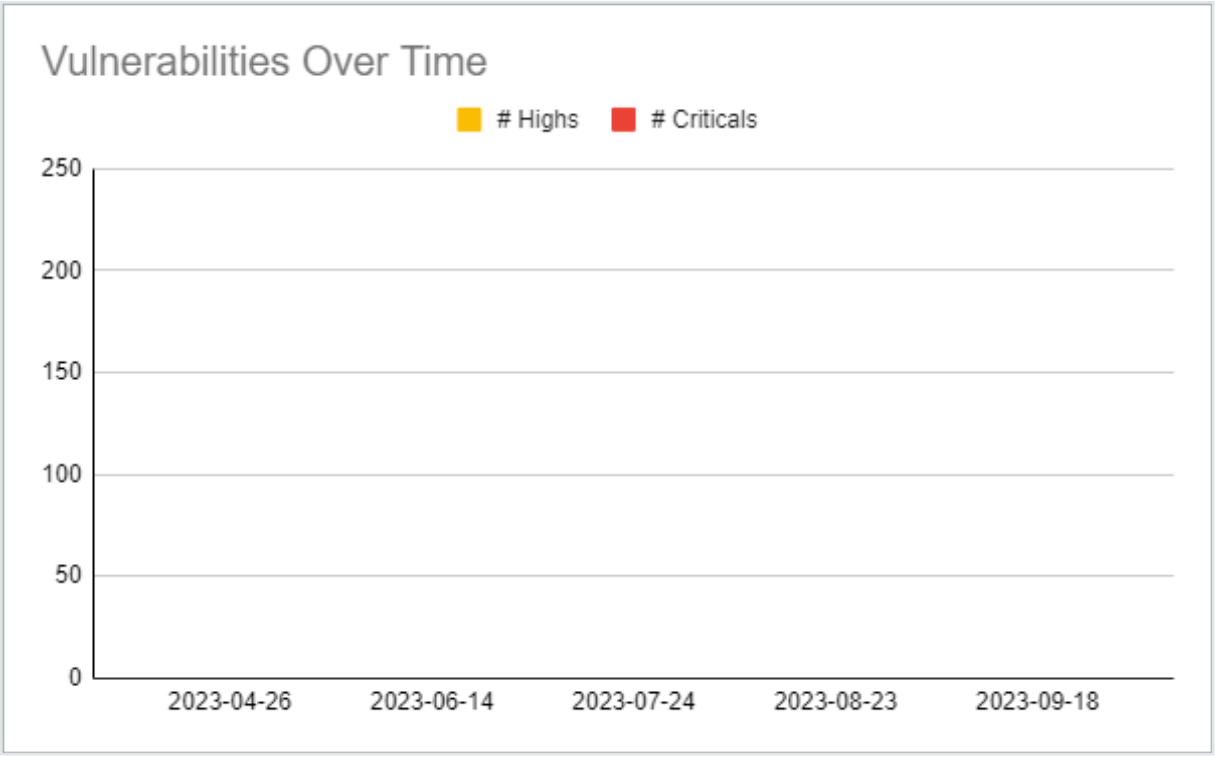
> Warning We found 1 critical and 1 high CVE in our JFrog scan. Upon investigation, these CVEs are disputed by Red Hat and no fix is available. For more information, see the official statements from Red Hat:
> https://access.redhat.com/security/cve/cve-2023-0687 (Critical)
> https://access.redhat.com/security/cve/cve-2023-27534 (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.



Beta UI

Beta UI Status

The following table shows the status of the Beta 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

Page	Location	Status
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
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 Beta UI.

Beta 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.

Collibra Data Quality & Observability release archive

Release 2024.02

- Highlights
- Migration Updates

- Enhancements
- Beta Features
- Fixes
- Known Limitations
- DQ Security

Release Information

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

Highlights

Archive Break Records

[[[Undefined variable DataQuality.Pullup Spark processing]]]
When rule breaks are stored in the PostgreSQL [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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

Old Key	Renamed Key
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 [[[Undefined variable DataQuality.DQEnv]]] 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 [[[Undefined variable DataQuality.DQ 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.

Beta Features

Collibra AI

- [[[Undefined variable DataQuality.SQL assistant]]] (beta) 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 Data Intelligence Cloud for private beta 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.

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 Data Intelligence Cloud. (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

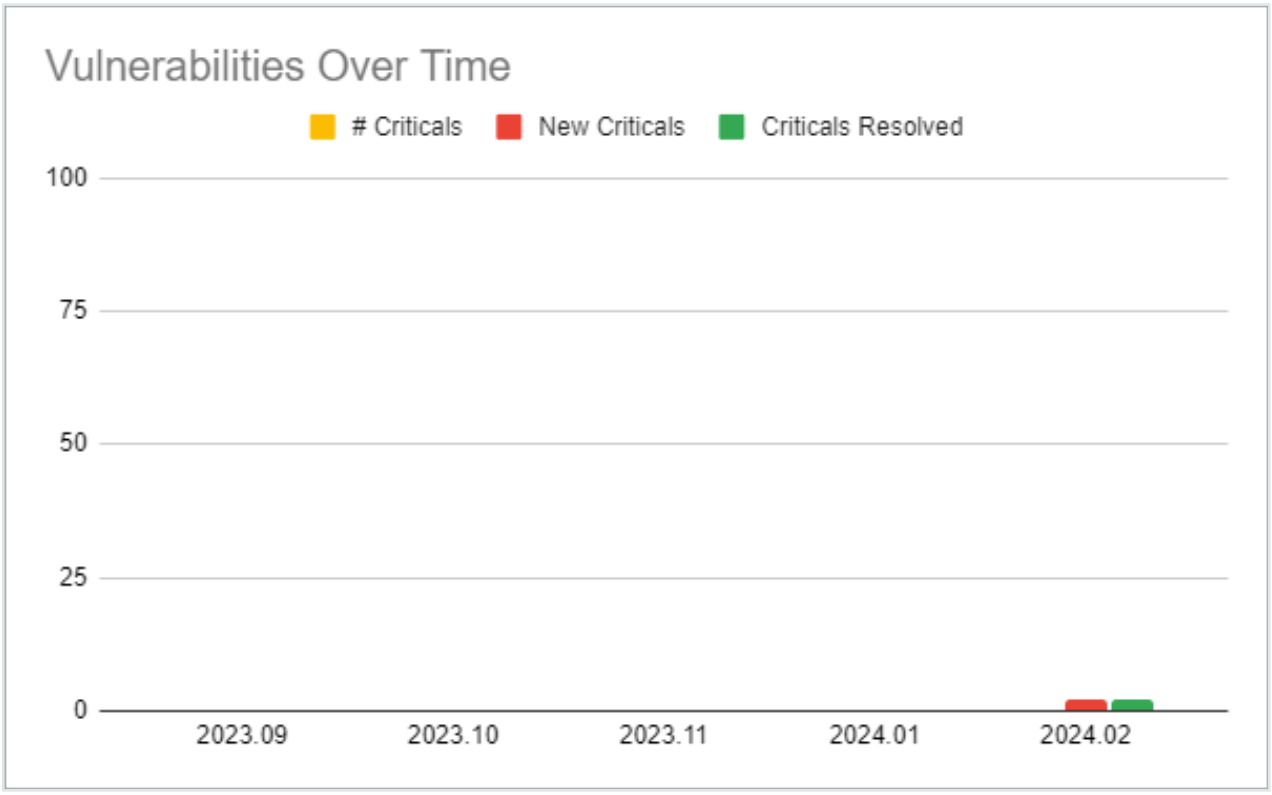
> Note If your current Spark version is 3.2.2 or older, we recommend upgrading to Spark 3.4.1 to address various critical vulnerabilities present in the Spark core library, including Log4j.

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 Collibra Data Quality & Observability 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 Data Intelligence Cloud for private beta 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 [[[Undefined variable DataQuality.DQEnv]]] 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 [[[Undefined variable DataQuality.DQEnv]]] 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.

Env variable	Variable name	Default	Usage
VALIDATION_PATTERN_CONN_NAME	Connection name	Alphanumeric characters (letters and digits), underscores (_), and hyphens (-).	For overriding defaults of connection names in Collibra DQ.
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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 Data Intelligence Cloud.
- When a User Defined Rule is deactivated or deleted from a dataset in Collibra DQ, then the Rule and Score status in Collibra Data Intelligence Cloudwill 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

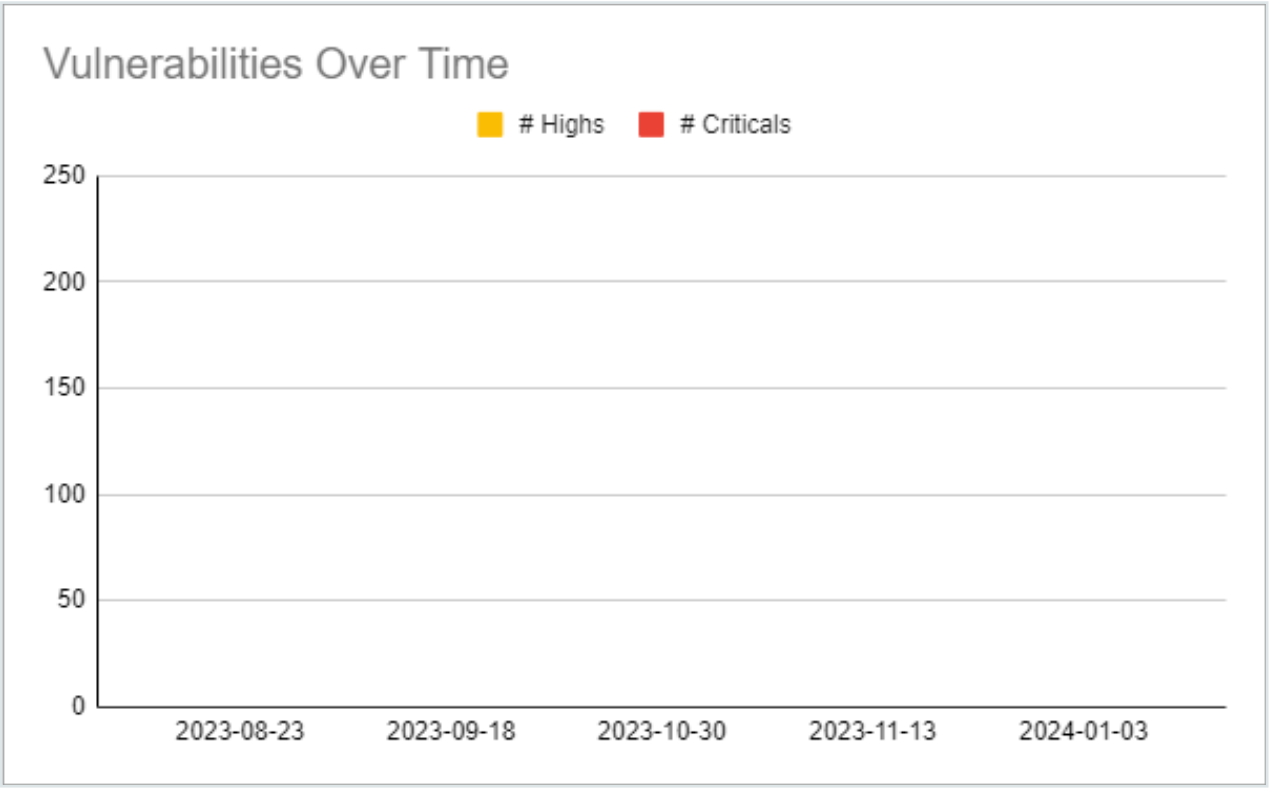
- When configuring `dupelimitui`, values above 2000 result in some UI lag. Because of this, we recommend using `dupelimitui` values under 2000.

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 Collibra Data Quality & Observability 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 beta 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 beta.

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 [[[Undefined variable DataQuality.DQEnv]]] 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 Data Intelligence Cloud. 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 Data Intelligence Cloud is enabled, you can now view the Community, Sub-Community, and Domain hierarchy to which the integration is mapped in Collibra Data Intelligence Cloud from the metadata bar on all dataset-level pages. This increases the transparency of where Assets are created in Collibra Data Intelligence Cloud 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 Data Intelligence Cloud.
- When running Pushdown jobs with integrations enabled, the results now integrate into Collibra Data Intelligence Cloud 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 Data Intelligence Cloud.
- 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 Data Intelligence Cloud 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 Data Intelligence Cloud, the columns now display correctly in Collibra Data Intelligence Cloud 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.

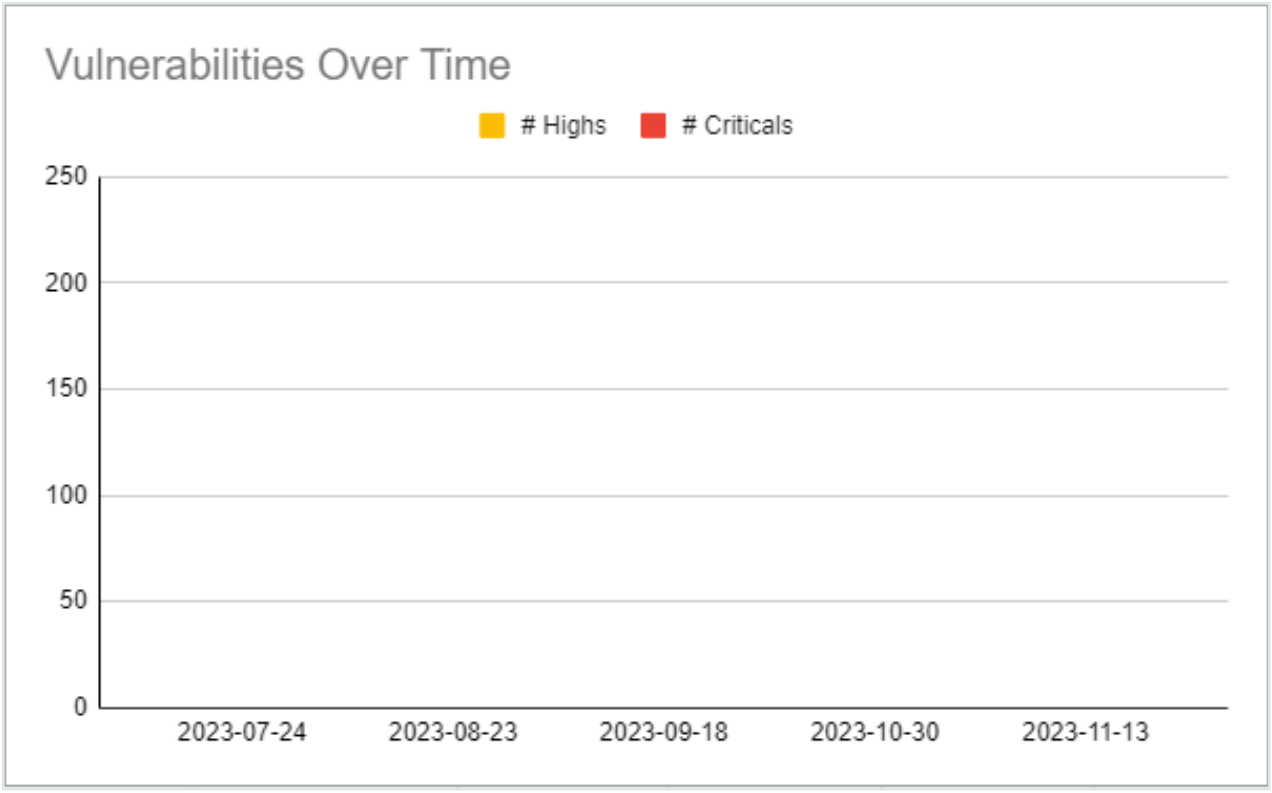
DQ Security

> 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 Data Intelligence Cloud. 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 beta 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 beta 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 beta 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 beta 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
- Beta UI
- Beta UI Limitations

Release Information

- Expected release date of Collibra Data Quality & Observability 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 beta 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 [[[Undefined variable DataQuality.DQEnv]]] 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 Data Intelligence Cloud 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

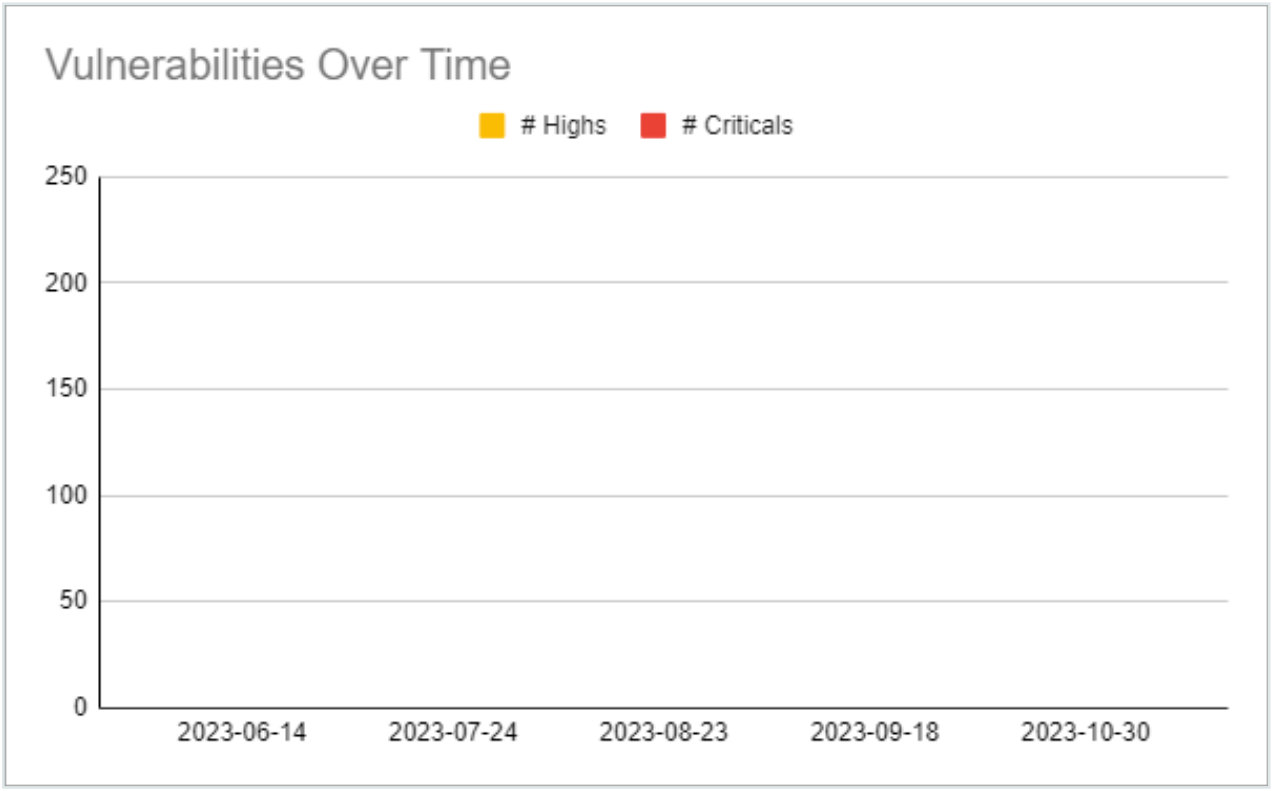
within containers, you need to modify them to use different mechanisms, such as dev/tcp socket, for the same functions.

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.



Beta UI

Beta UI Status

The following table shows the status of the Beta 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

Page	Location	Status
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 Beta UI.

Beta 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
- Beta UI
- Beta UI Limitations

Release Information

- Expected release date of Collibra Data Quality & Observability 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 betas! 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 [[[Undefined variable DataQuality.DQEnv]]] 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 Data Intelligence Cloud 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 Data Intelligence Cloud integration, you now correctly see database views from Collibra DQ to the tables and columns to which they relate in Collibra Data Intelligence Cloud. (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,

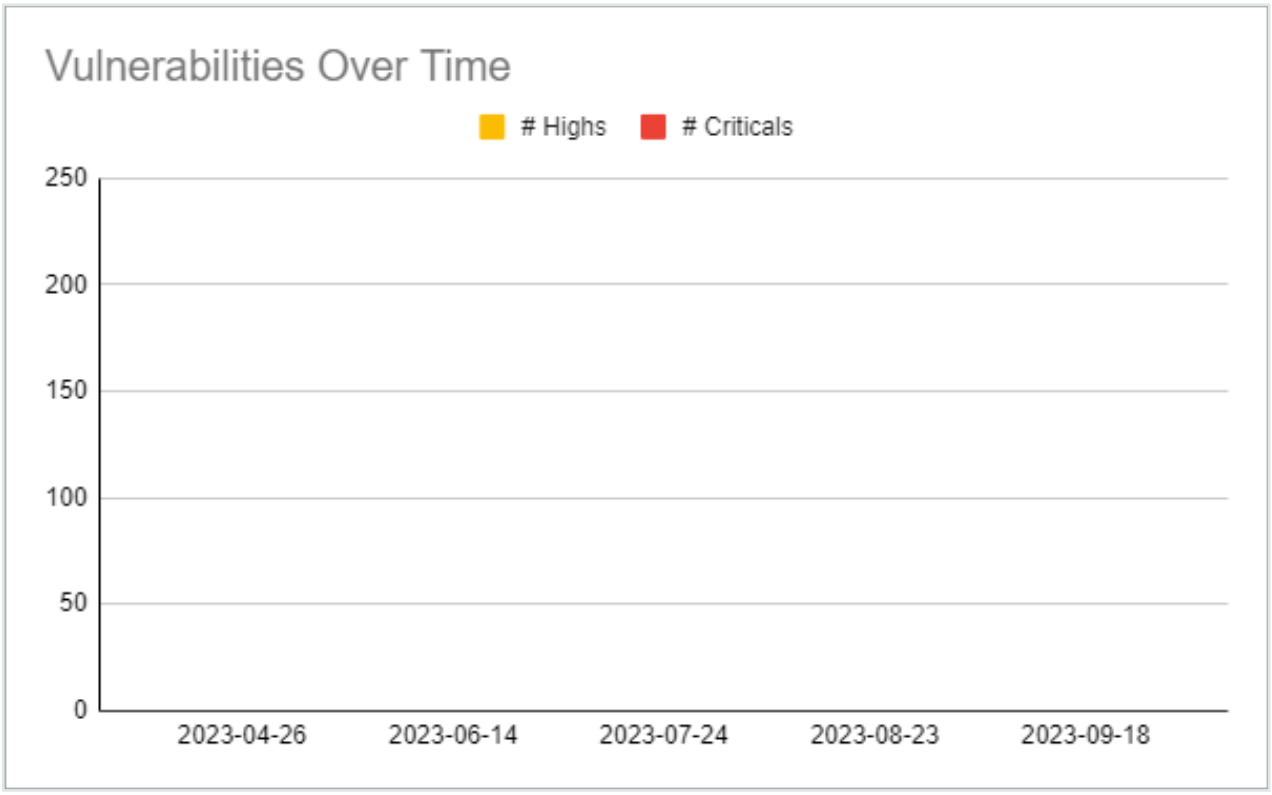
> see the official statements from Red Hat:
> https://access.redhat.com/security/cve/cve-2023-0687 (Critical)
> https://access.redhat.com/security/cve/cve-2023-27534 (High)

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.



Beta UI

Beta UI Status

The following table shows the status of the Beta 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

Page	Location	Status
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 Beta UI.

Beta 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
- Beta UI Redesign
- Beta 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 beta 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 Collibra Data Quality & Observability 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

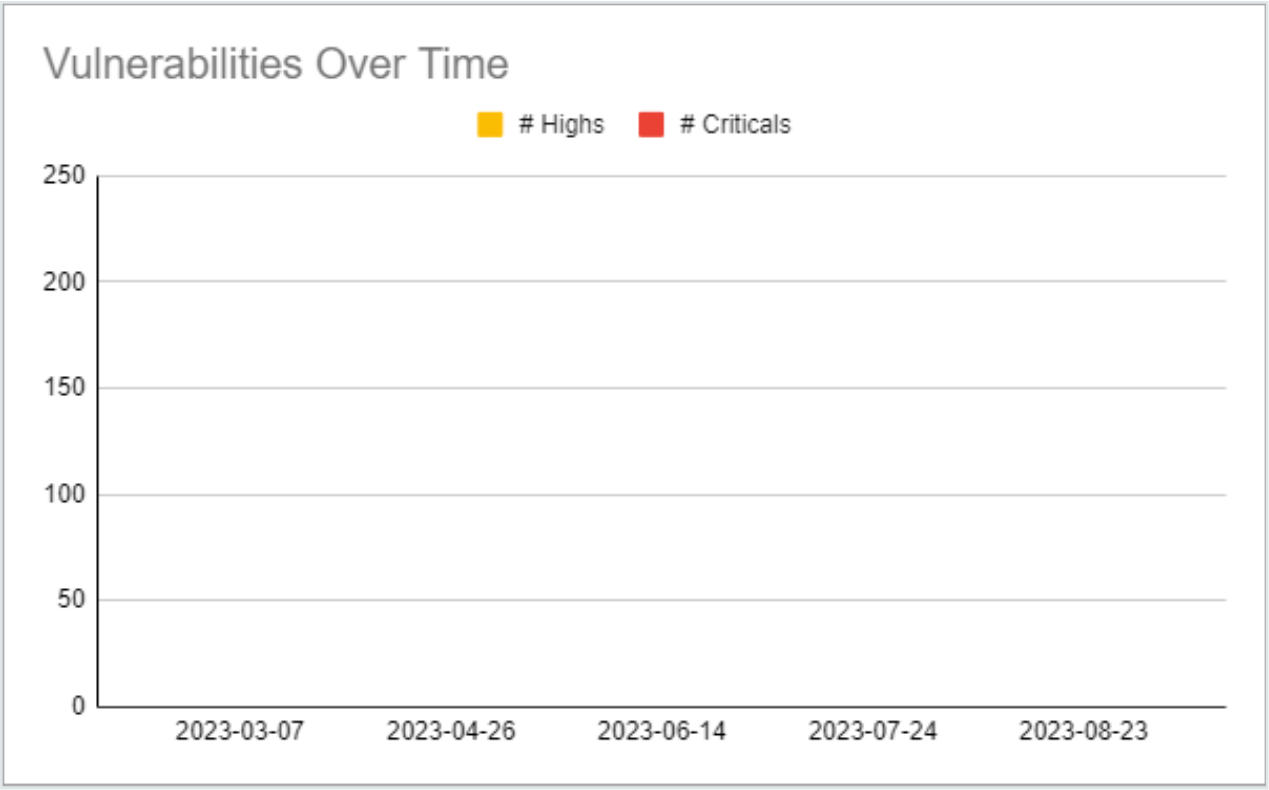
> Note The medium, high, and critical vulnerabilities of the DQ Connector are now resolved.

> 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.



Beta UI Redesign

The following table shows the status of the Beta 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 Beta 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

Page	Location	Status
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	✕ 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 Beta UI.

Beta 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 Data Intelligence Cloud, 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 Data Intelligence Cloud.
- Align DQ tenants with Collibra Data Intelligence Cloud communities.
- Map DQ dimensions to their corresponding dimensions in Collibra Data Intelligence Cloud.

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 Collibra Data Quality & Observability.

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

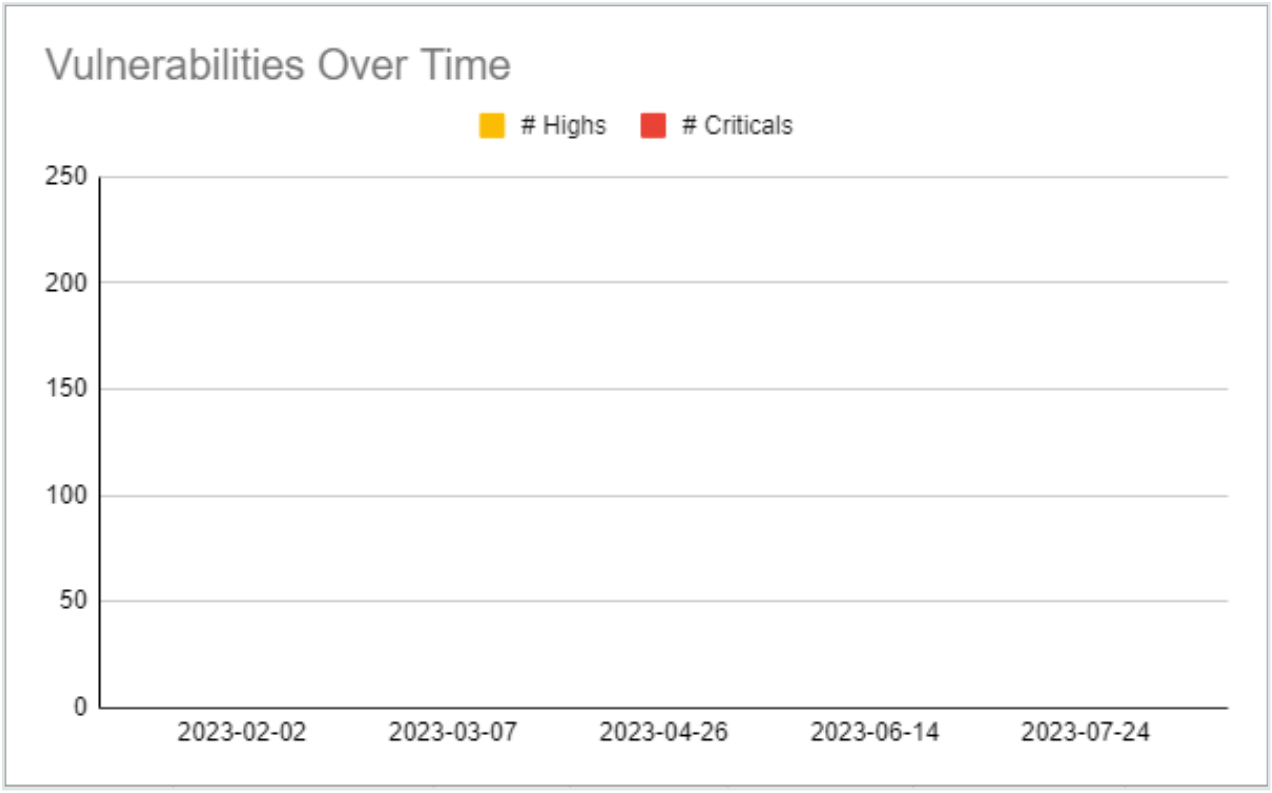
> Note The medium, high, and critical vulnerabilities of the DQ Connector are now resolved.

> 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

Page	Location	Status
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
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 beta 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 Data Intelligence Cloud 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 Collibra Data Quality & Observability 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 username, 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 [[[Undefined variable DataQuality.BigQuery abbreviated]]] 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.

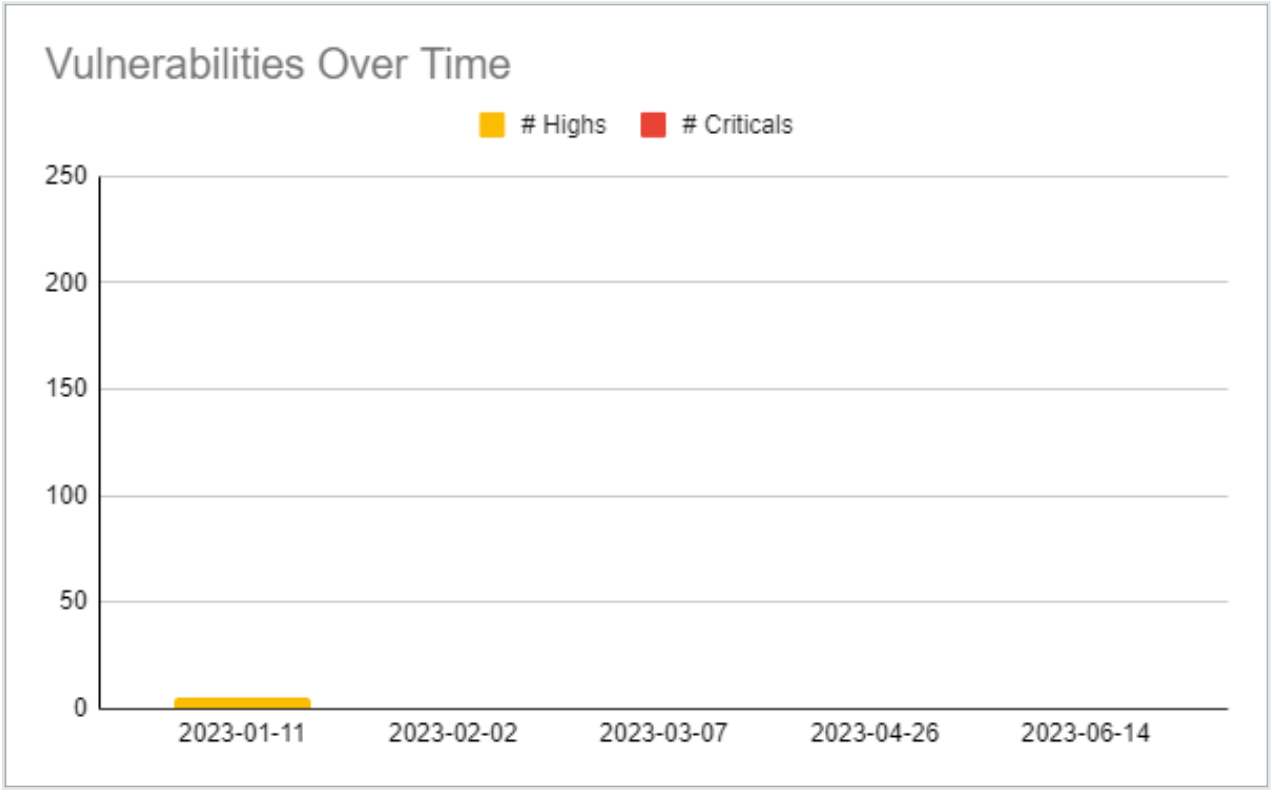
      > Note If you encounter any difficulties locating the necessary jar file on the Apache Downloads page, contact your CS or SE for assistance.

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
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

Page	Location	Status
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
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 beta! 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 Data Intelligence Cloud. With data quality details in Collibra Data Intelligence Cloud, 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 beta 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

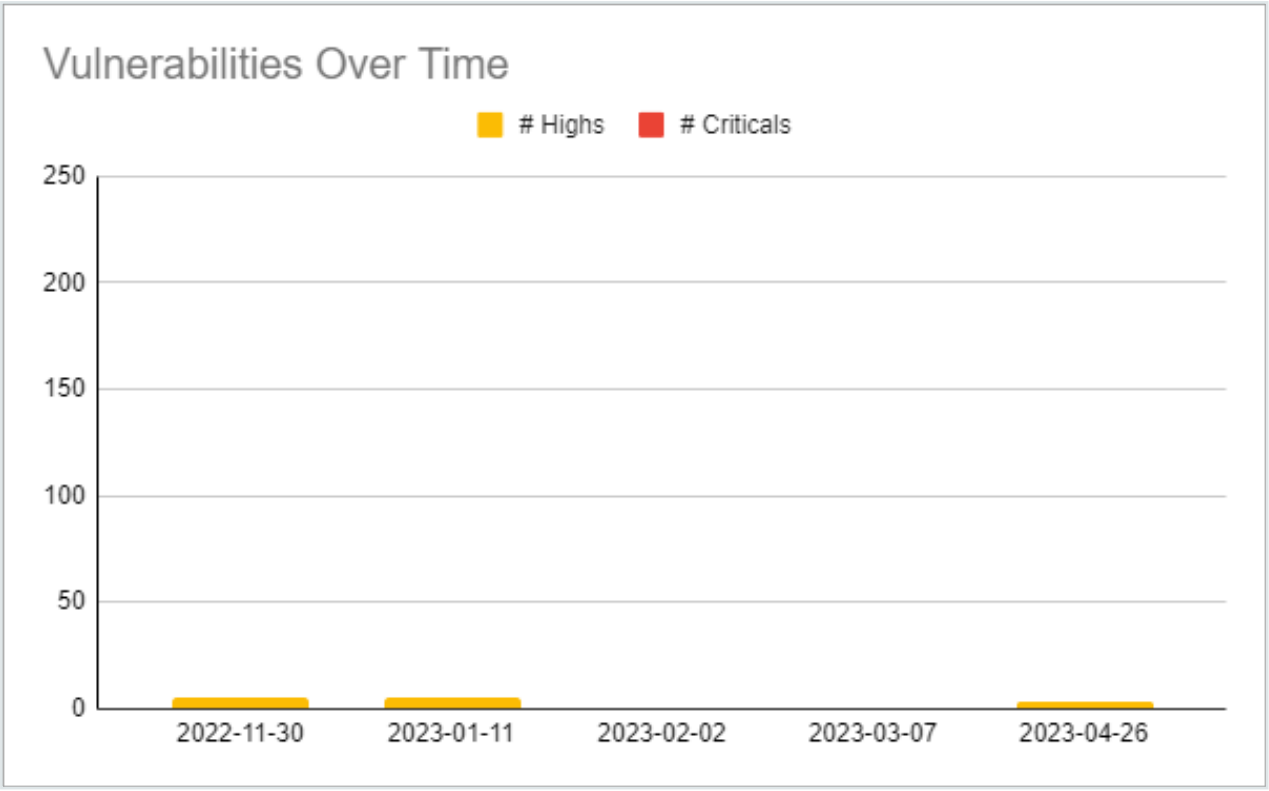
- When scanning for dupes, fuzzy match duplicates display incorrectly as "NULL" values on the Findings page. This will be fixed in the 2023.06 release.

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

Page	Location	Status
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	✕ In Progress
Rule Builder	Rules	✕ In Progress
Data Classes	Rules	✓ Done

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

2023.05.1

- 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 dropdown 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.

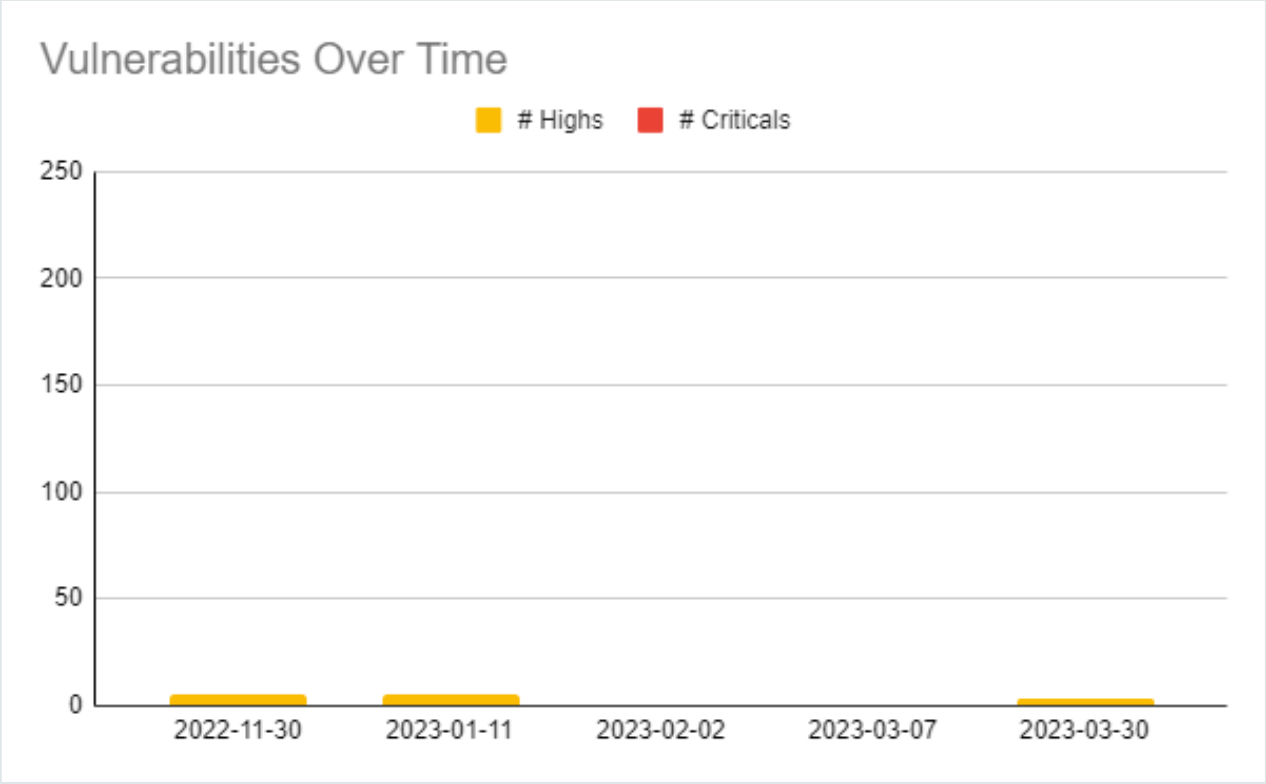
DQ Connector

- 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 Data Intelligence Cloud.
  - A possible workaround until a fix is available is to update `export CSRF_TOKEN_ENABLED=false` in the [[[Undefined variable DataQuality.DQEnv]]] 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
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	✕ In Progress
Rule Workbench	Rules	✕ In Progress
Rule Builder	Rules	✕ In Progress
Data Classes	Rules	✕ To Do
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
> ```

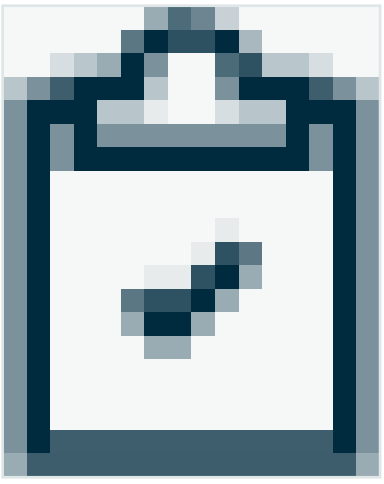
New Features

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 (beta)

- 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 (beta)

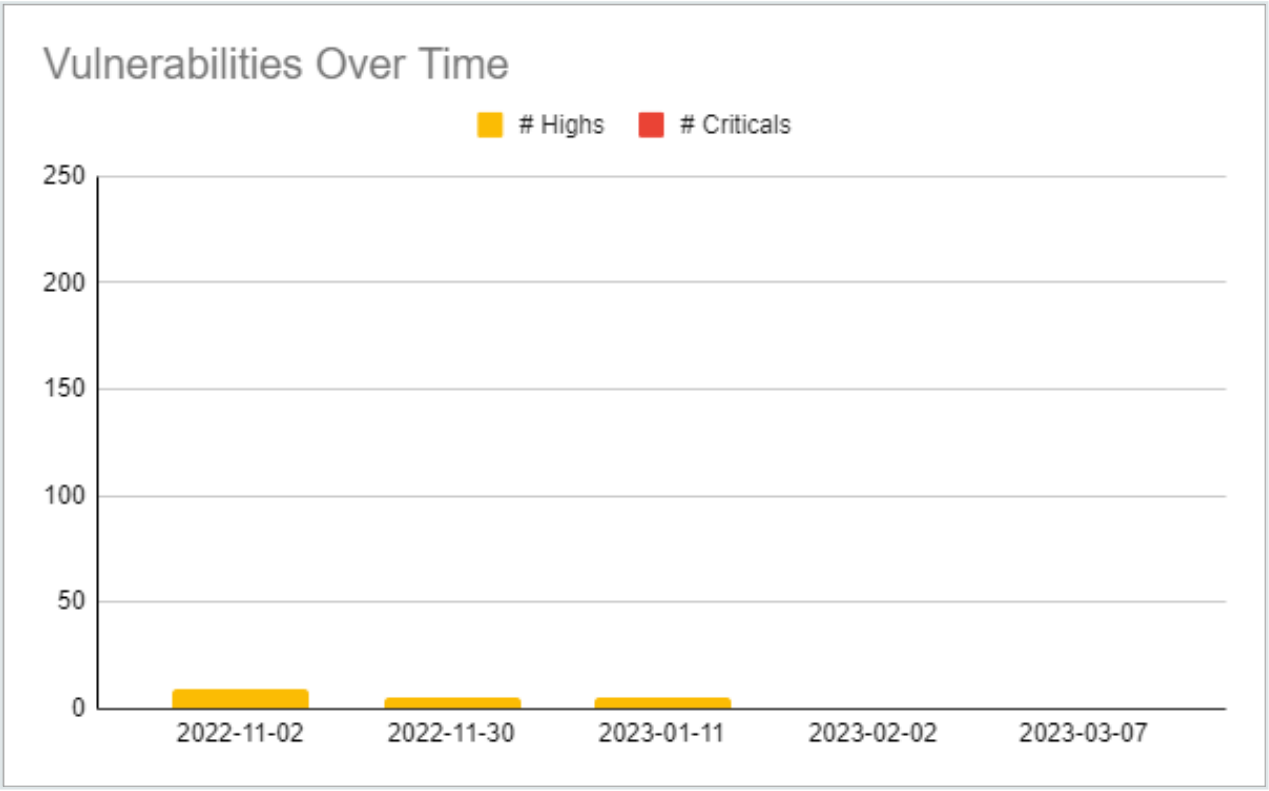
- Mixed datatype rules are not supported in Pushdown.

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 Data Intelligence Cloud.
    - A possible workaround until a fix is available is to update `export CSRF_TOKEN_ENABLED=false` in the [[[Undefined variable DataQuality.DQEnv]]] 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 (beta)

- 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 (beta)

- 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

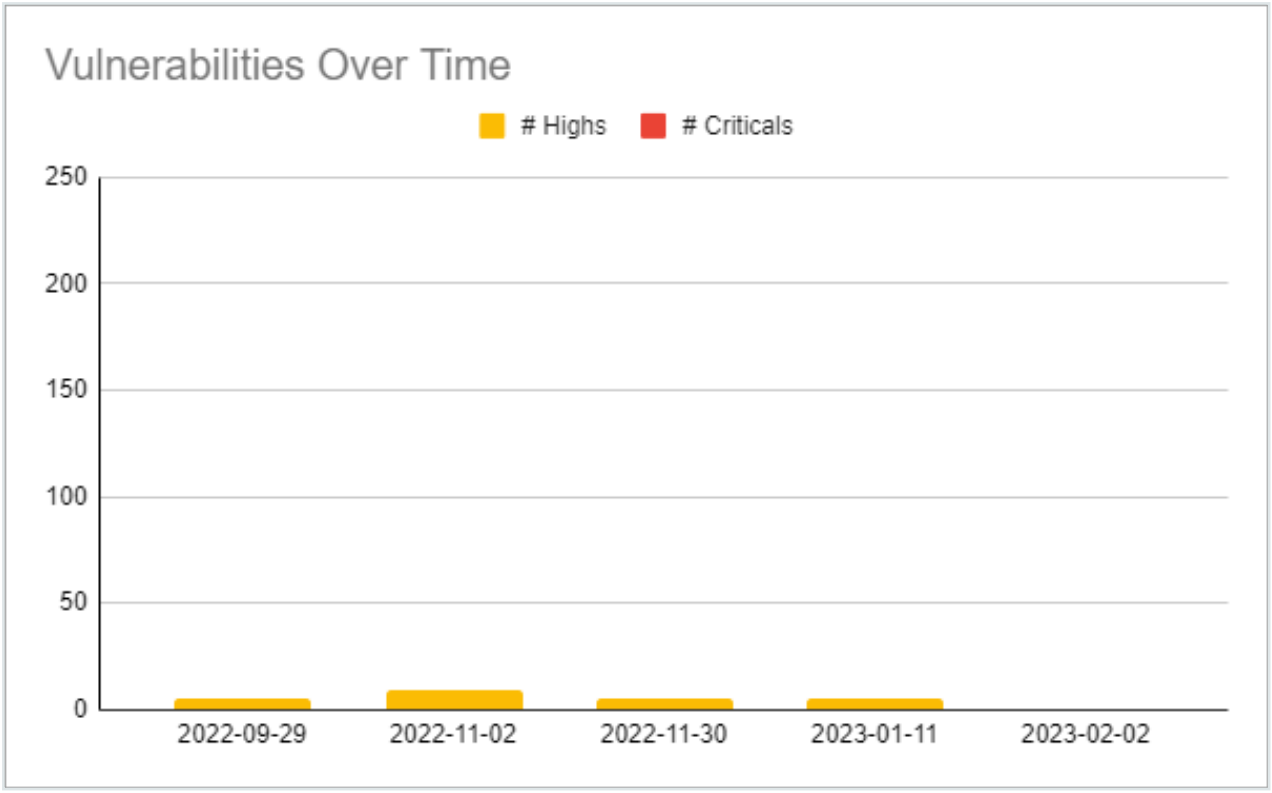
- The $daysWithoutData stat rule has a limitation where rules that are not named "daysWithoutData" do not display their Pulse View charts when you drill into the rule on the Findings page. A workaround for this is to name your stat rule exactly as it appears here: `daysWithoutData`.

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 Data Intelligence Cloud 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

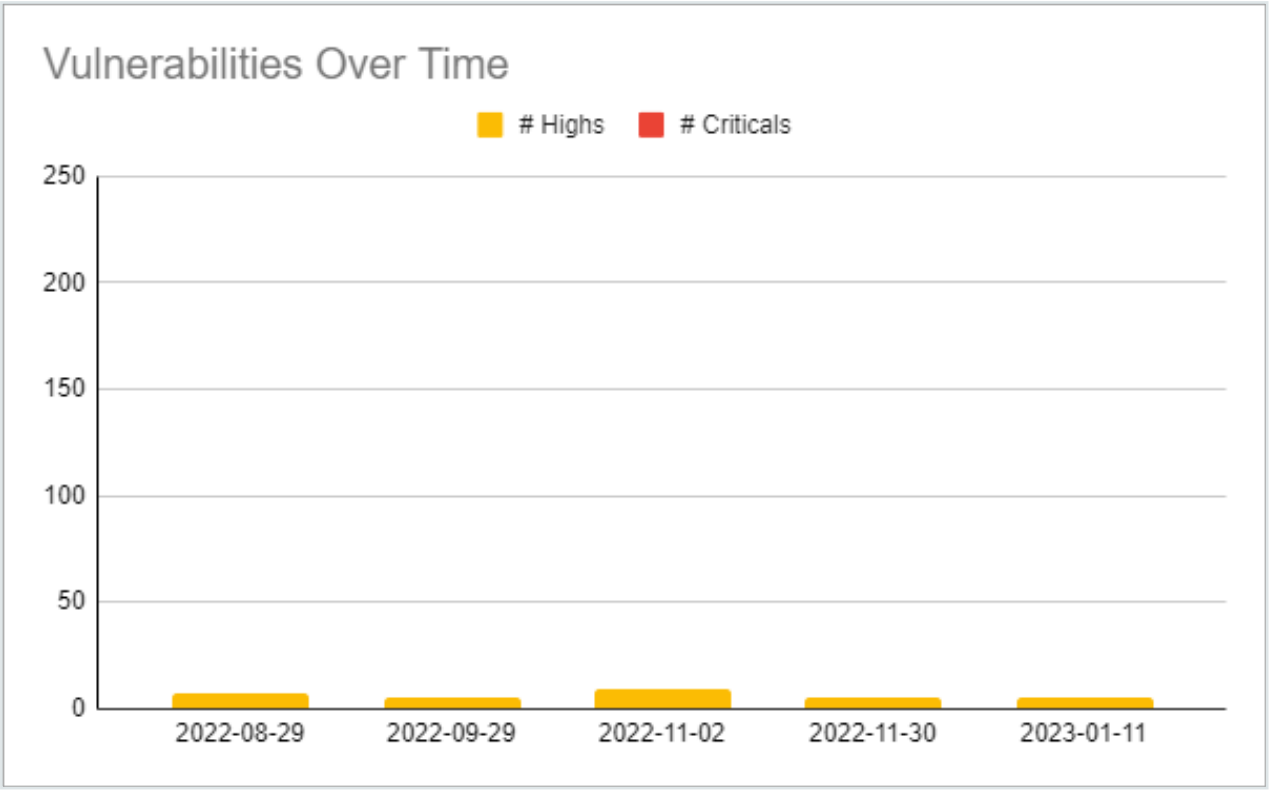
Rules

- 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

Collibra Data Quality & Observability 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 (beta)

- 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

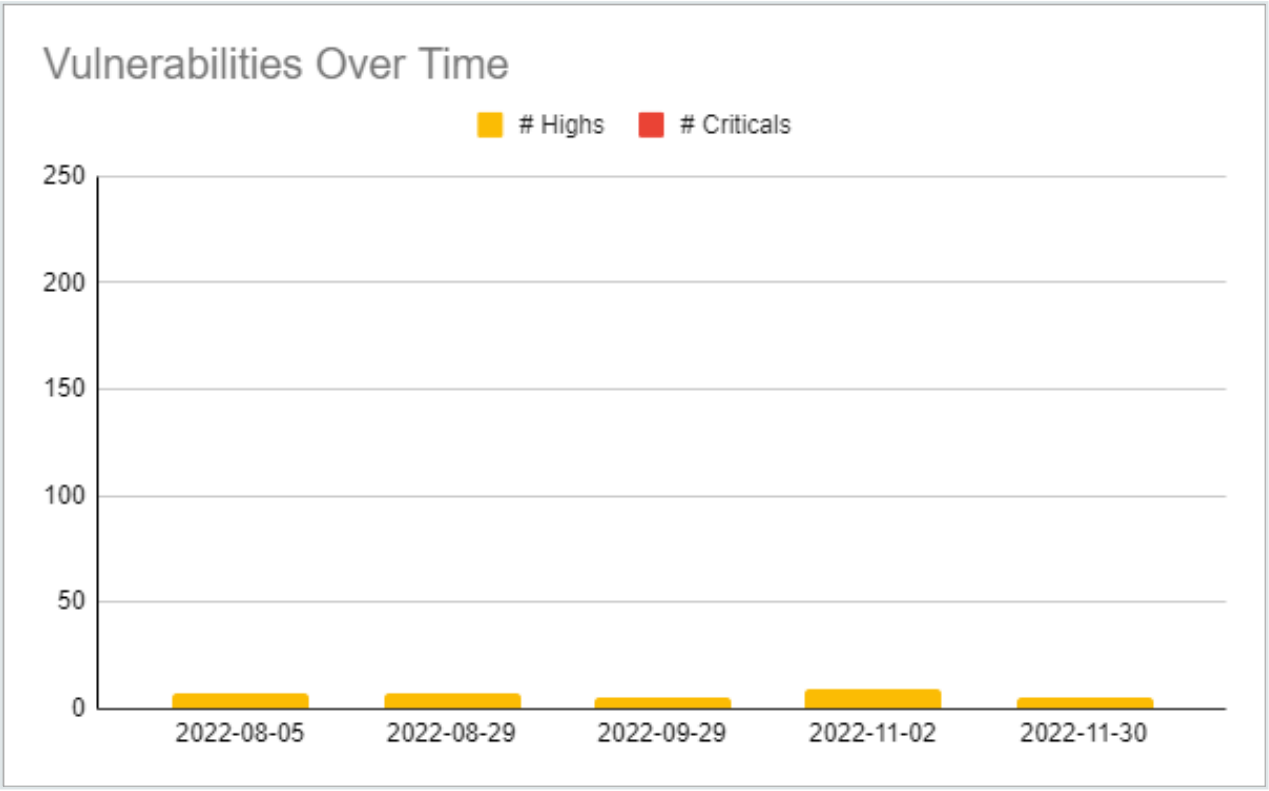
Rules

- 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

Collibra Data Quality & Observability 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 semi-colon 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 (beta)

- 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.

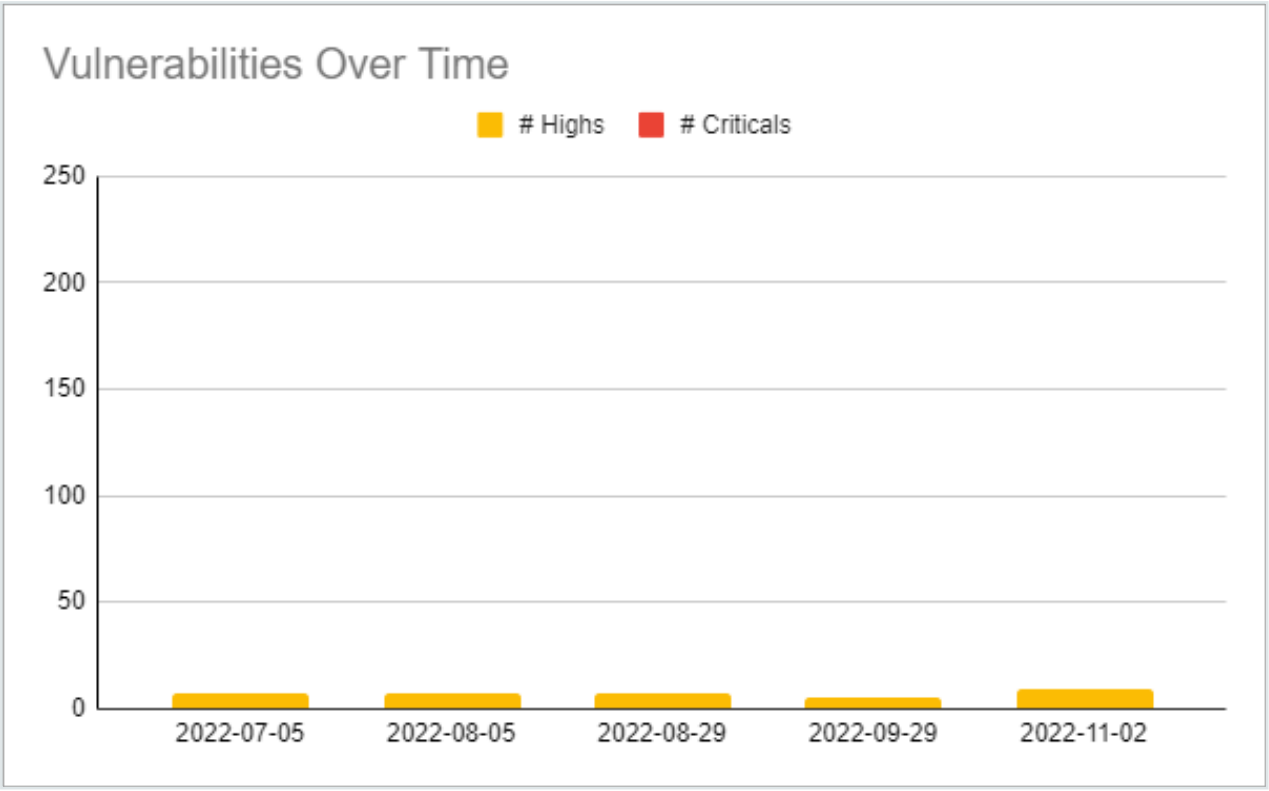
Validate Source

- 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 check-box 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 `-post-clearcache` 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 (beta)

- 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.

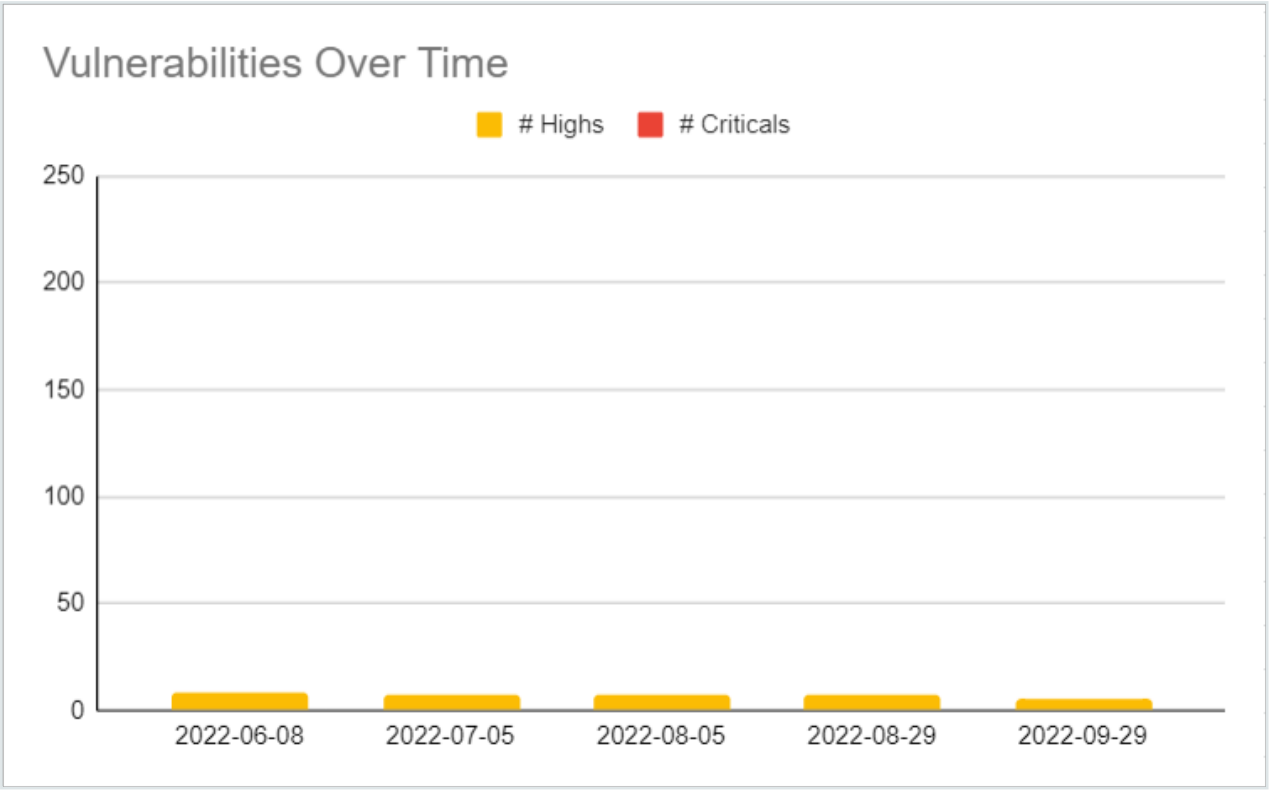
Catalog

- 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 (beta)

- 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 2024.02 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

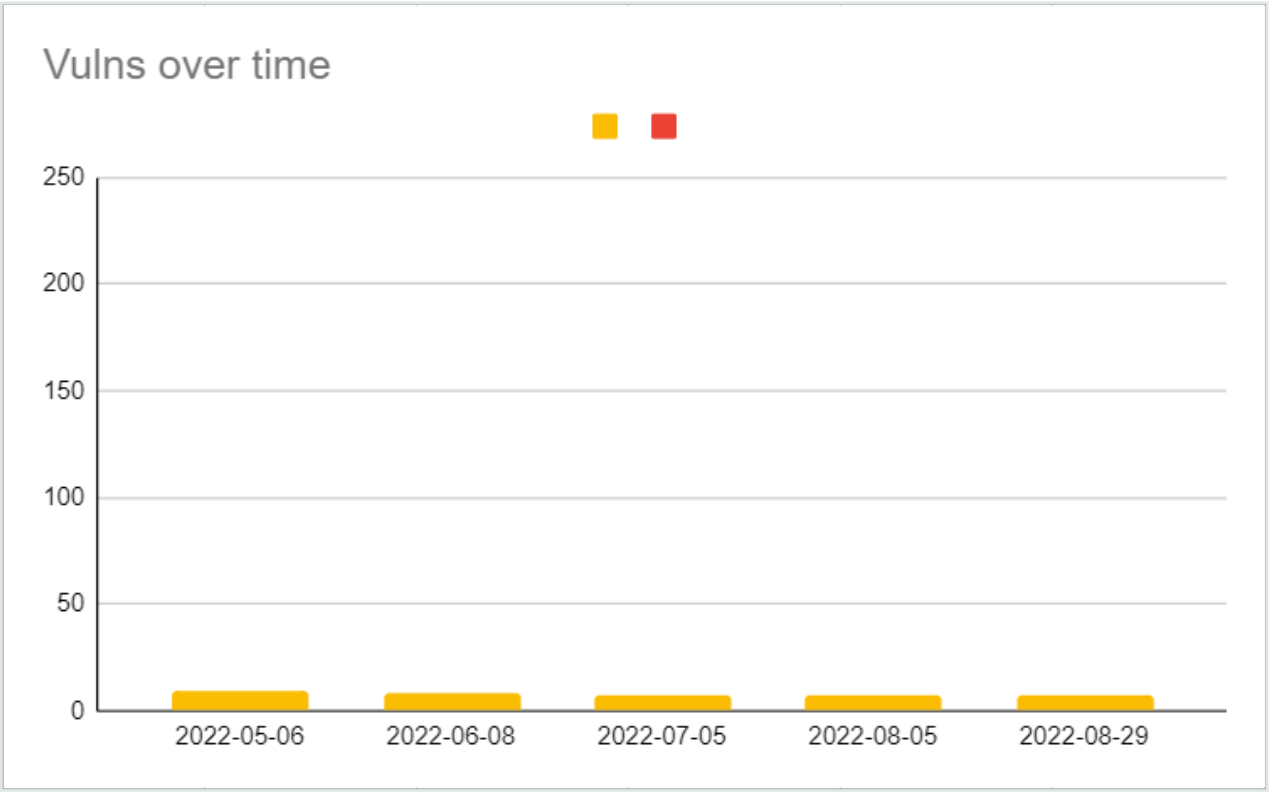
- When alert recipient email addresses are separated by semicolons `;`, alerts emails are not sent to the intended recipients.

    - 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 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.

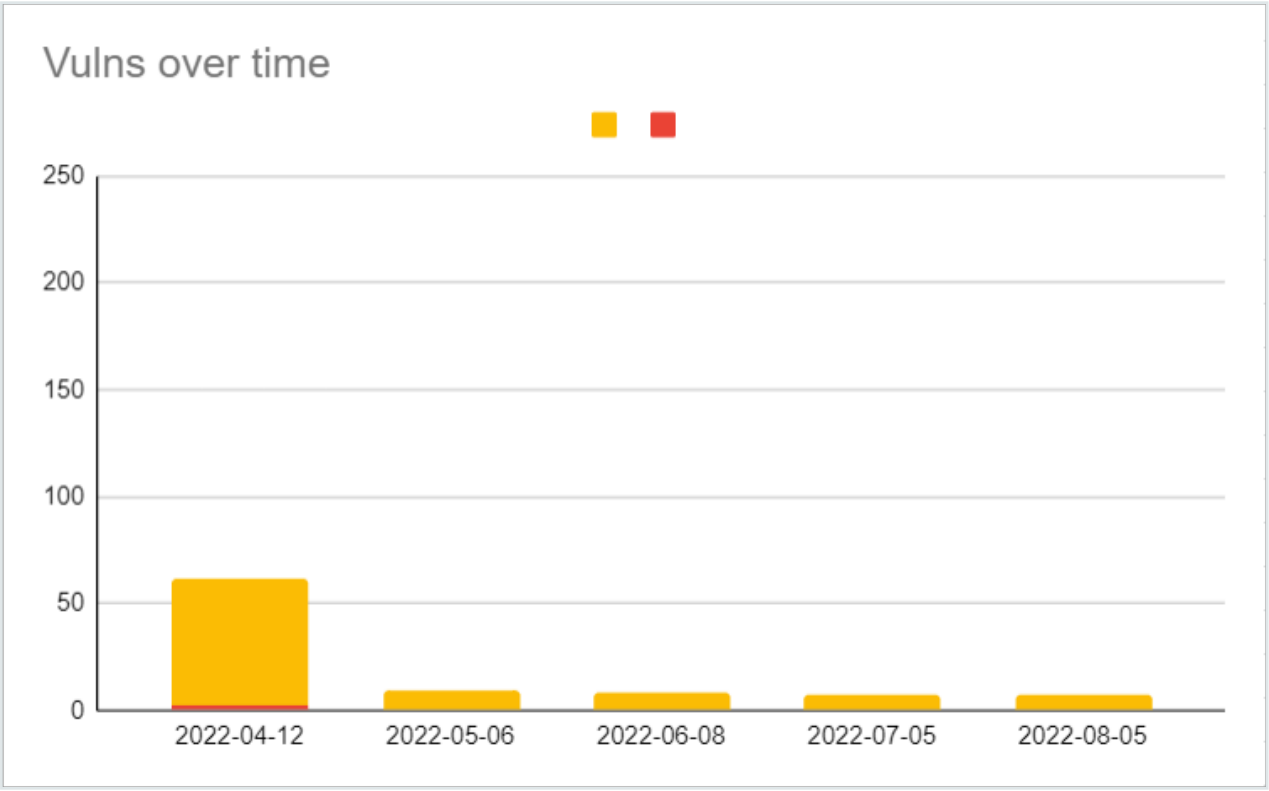
Beta features

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 beta feature only available by request. Since this is a beta 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 vulnerability 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.

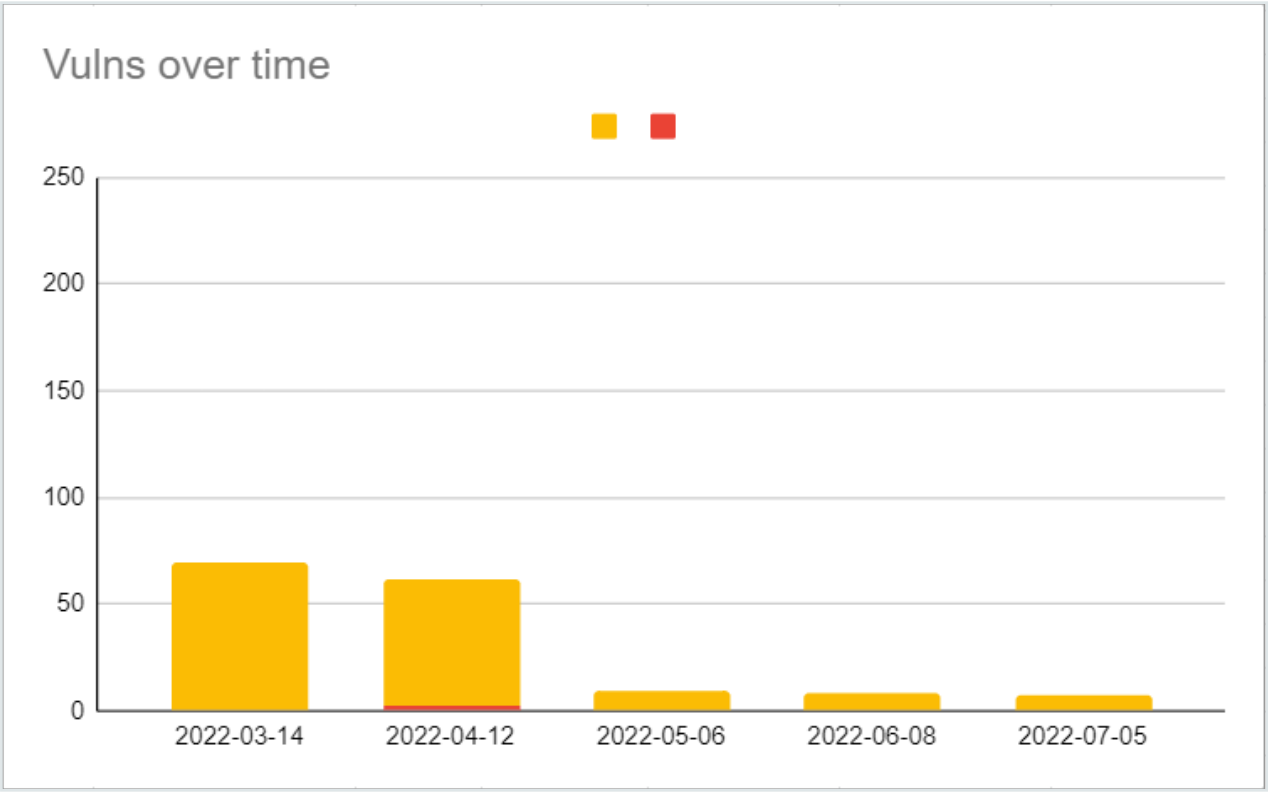
        - A UI integration is planned for a future release.
- Schedule
    - Added an Active column to the scheduler export.
        - The RunJob column was removed. (#88799)
- Reporting
    - Fixed an issue that created misalignment of column headers in PDF exports. (#89739)

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 Resol	# 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.

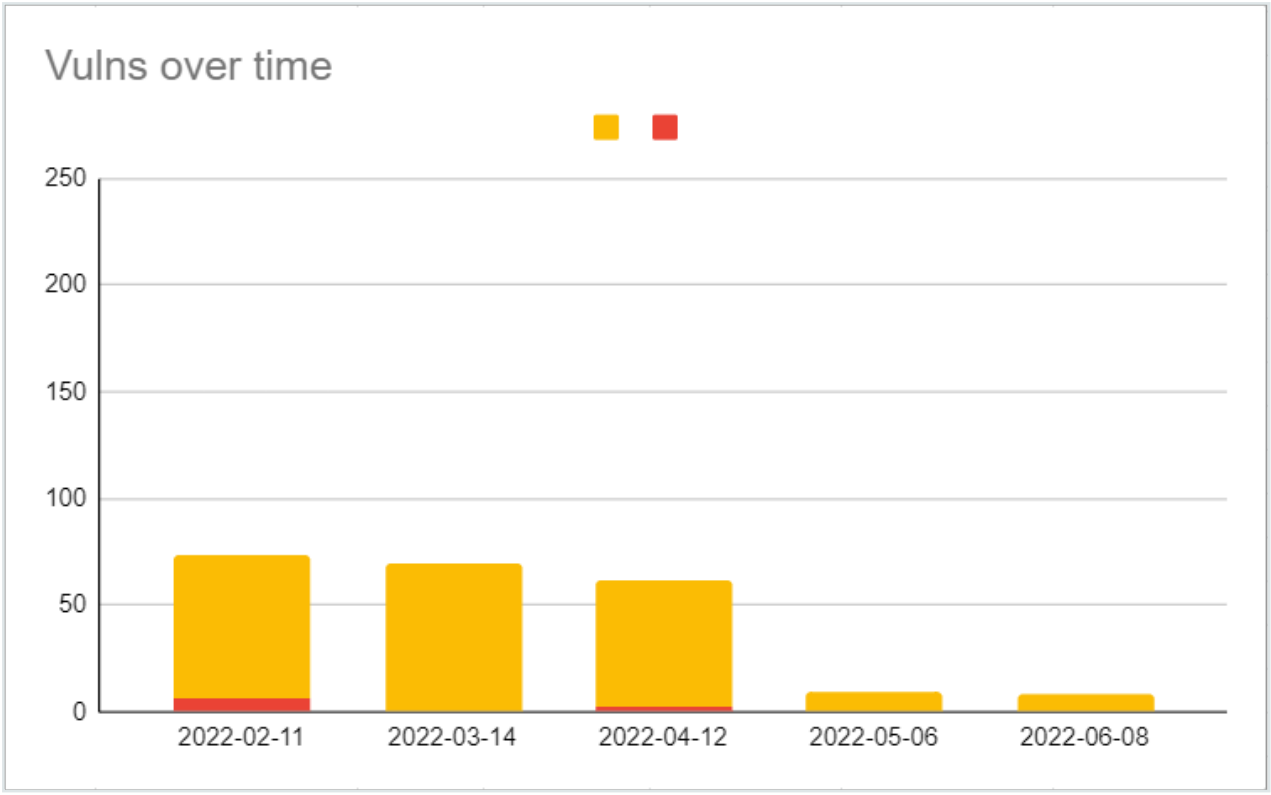
- Legal
  - Added a disclaimer to the DQ login page with a link to the Collibra Evaluation Agreement.

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 (configurable via app config).
        - 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 (ubdate users table "accountNonLocked" colun 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.

Release 2022.03

Fixes / Enhancements

- DQ Job
  - The -validatevaluesshowmissingkeys options now allows the extrapolation of missing keys between target and source.
  - Newly created jobs will no longer be marked incorrectly with enclosing double quotes.
  - File names with spaces are now handled with double quotes within the application.
- Alerts
  - Email notifications now have Collibra branding and terminology.
  - Fixed Cancel Action for Delete functionality on Alert page.
- Outliers
  - Fixed the issue where Numerical Outlier drill in graph wasn't displaying when perChange is NaN.
- Rules
  - Added additional HealthCare Data Classes to Rule Library.
  - Fixed input validation rule of POST - /v3/rules/ endpoints. The following validation rules have been applied to RuleDTO.ruleName field:
    - Maximum size is 100.
    - Must comply with the following regular expression: ^[a-zA-Z0-9_]+$
  - The rules on the Hoot page now show the correct exception data when expanded if there are two or more rules with exceptions attached to the dataset.

- Security
  - Vulnerabilities identified by Jfrog
    - Vulns 0, critical, 6 high vulnerabilities
  - Password length has increased to a maximum of 72 characters.
  - Forgot password screen will now always show success message in UI regardless of success or failure.
  - Fixed an issue of a throwing error message when adding/editing user roles.
  - Added error checks if the password manager script throws any errors.
  - Added the helper text "Enforce user roles to run the job" to DQ Job Security row.
  - User password field removed while updating user in user management screen.
    - Admin can only set password for another user wile creating new user, but not while updating/modifying them.
    - To change a password, users can now use either the profile page or the self-service (Forgot password) feature.
  - XSS security
    - Fixed the vulnerability on scorecard, jobs, rules and catalog pages.
    - Fixed the vulnerability via remote connection.
  - Mitigated the endpoint "/v2/getrawpreview" vulnerable to Local File inclusion attack.
  - DQ HTTP session cookie is now secured by default when HTTPS is enabled.

Rule Discovery Terminology Alignment

Data Concepts => Data Categories

Semantics => Data Classes

- Profile
  - Precision and Scale metrics are correct when using multi executors.
- Admin
  - Edge download page within Admin Console (for Cloud customers).
- Validate Source
  - *Tech Preview* [TP] Update Source Scope.
    - Added "Update Source Scope" in the Query section of the Source tab.
- Connection
  - Added handling for errors during log cleanup process.

- API
  - Improved API calls for the Save function.

Known Limitations

- Validate Source
  - *Tech Preview* [TP] Update Source Scope.
    - Only works for JDBC connections. Feature is hidden for remote, temp, local files.
    - Valsrc query won't be updated automatically when modifying column mappings. Use 'Preview' button to reset the feature if column mappings need to be changed.

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] 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 get-catalogandconnsrcnamebydataset 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-S="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/creat-erule) 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 configuration settings are available when the Load Balancer is set for SSL Termination.
    - You can now set theMulti-tenancy 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

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 packaged.

> Note **The optional drivers only impact container versions of Collibra DQ and do not impact Standalone installation packages.**

Available Containers

2024.02

Collibra Data Quality & Observability

- 2024.02-ABDGCSHILM-204
- 2024.02-ABDGCSHILMH-205
- 2024.02-206

Spark

- 3.4.1-2024.02-ABDGCSHILM-204
- 3.4.1-2024.02-ABDGCSHILMH-205
- 3.4.1-2024.02-206

2024.01

Collibra Data Quality & Observability

- 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

2023.11

> Note As of Collibra Data Quality & Observability 2023.11, the default Spark version for cloud native deployments is 3.4.1.

Collibra Data Quality & Observability

- 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

2023.10

Collibra Data Quality & Observability

- 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

2023.09

> 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.

Collibra Data Quality & Observability

- 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

2023.08

Collibra Data Quality & Observability

- 2023.08-ABDGCSHILM-3292
- 2023.08-3298

Spark

- 3.2.2-2023.08-ABDGCSHILM-3292
- 3.2.2-2023.08-3298

2023.07.4

Collibra Data Quality & Observability

- 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

Collibra Data Quality & Observability

- 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
Collibra Data Quality & Observability

- 2023.07-ABDGCSHILM-3156
- 2023.07-3162

Spark

- 3.2.2-2023.07-ABDGCSHILM-3156
- 3.2.2-2023.07-3162

2023.06
Collibra Data Quality & Observability

- 2023.06-ABDGCSHILM-2957
- 2023.06-2958

Spark

- 3.2.2-2023.06-ABDGCSHILM-2957
- 3.2.2-2023.06-2958

2023.05.1
Collibra Data Quality & Observability

- 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
Collibra Data Quality & Observability

- 2023.05-ABDGCSHILM-254
- 2023.05-253

Spark

- 3.2.2-2023.05-ABDGCSHILM-254
- 3.2.2-2023.05-253

2023.04.2

Collibra Data Quality & Observability

- 2023.04.2-ABDGCSHILM-235

Spark

- 3.2.2-2023.04.2-ABDGCSHILM-235

2023.04.1

Collibra Data Quality & Observability

- 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

Collibra Data Quality & Observability

- 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

2023.03

Collibra Data Quality & Observability

- 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

2023.02

Collibra Data Quality & Observability

- 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

2023.01

Collibra Data Quality & Observability

- 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

2022.12

Collibra Data Quality & Observability

- 2022.12-LM-1960
- 2022.12-ABDGCSILM-1959
- 2022.12-ABDGCSHILM-1958
- 2022.12-1964

Spark

- 3.2.2-2022.12-LM-1960
- 3.2.2-2022.12-ABDGCSILM-1959
- 3.2.2-2022.12-ABDGCSHILM-1958
- 3.2.2-2022.12-1964

> Note The default build is considered the Secure Build with no optional drivers included and no critical vulnerabilities.

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 Collibra Data Quality & Observability 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 Packaged

- 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

APIs

Rest

Please see the REST APIs section for more details.

```
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/-
job-
s/run-
?agentName='+agentName+'&dataset='+dataset+'&runDate='+runDate,
headers=owl_header, verify=False)
jobId = str(response.json()['jobId'])

# Wait
waitForJobCompletion(100)

# Results
response = requests.get(url = owl + '/v3/-
jobs/'+jobId+'/findings', headers=owl_header,  verify=False)
```

Notebook

Please see the Databricks example for more information

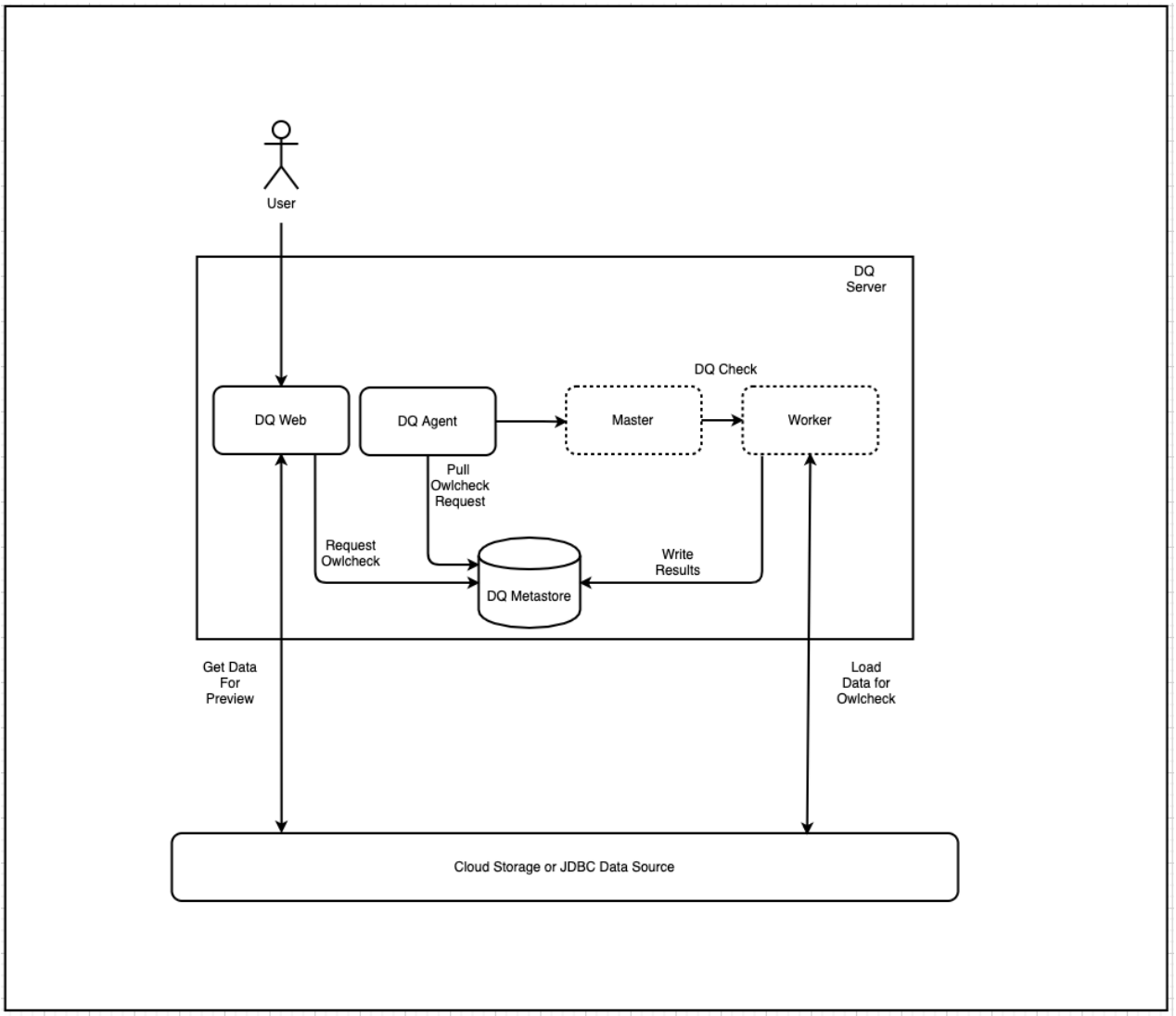
```
val dataset = "cdq_notebook"
var date = "2018-01-11"

// Options
val opt = new OwlOptions()
opt.dataset = dataset
opt.runId = date
opt.host = pgHost
opt.port = pgPort
opt.pgUser = pgUser
opt.pgPassword = pgPass

// Pre Routine
val cdq = com.owl.core.util.OwlUtils.OwlContext(df, opt)
.register(opt)

// Scan
cdq.owlCheck()
val results =  cdq.hoot()
```

Chapter 5

Collibra DQ Installation

Standalone

When large scale and high concurrency checks are not required, DQ can be installed and operated entirely on a single host. In this mode, DQ will leverage a **Spark Standalone pseudo cluster** where the master and workers run and use resources from the same server. DQ also requires a Postgres database for storage and Java 8 for running the DQ web application. It is possible to install each of the Spark, Postgres, and Java 8 components separately and install DQ on top of existing components. However, we offer a full installation package that installs these components in off-line mode and install DQ in one server.



Standalone Install

1. Setup Tutorial Assumptions

We assume that a server running CentOS 7/CentOS 8 or RHEL 7/RHEL 8 is set up and ready to install Collibra DQ in the home directory (base path: `OWL_BASE`) under the subdirectory `owl` (install path: `$OWL_BASE/owl`). There is no requirement for Collibra DQ to be installed in the home directory, but the Collibra DQ Full Installation script may lead to permission-denied issues during local PostgreSQL server installation if paths other than the home directory are used. If these issues occur, please adjust your directory permission to allow the installation script a write access to the PostgreSQL data folder.

> Note Collibra DQ is supported to run on RHEL 8. You can use your PostgreSQL database instance.

This tutorial assumes that you are installing Collibra DQ on a brand new compute instance on Google Cloud Platform. Google Cloud SDK setup with proper user permission is assumed. This is optional, as you are free to create Full Standalone Installation setup on any cloud service provider or on-premise.

Please refer to the **GOAL** section for the intended outcome of each step and modify accordingly.

> Note The full install package supports CentOS 7/CentOS 8 and RHEL 7/RHEL 8. If another OS is required, please follow the basic install process. You must have a PostgreSQL database installed and accessible, prior to running an RHEL 8 installation package.

> Note Collibra DQ Metastore supports AWS RDS or AWS Aurora.

```
# Create new GCP Compute Instance named "install"
gcloud compute instances create install \
    --image=centos-7-v20210701 \
    --image-project=centos-cloud \
    --machine-type=e2-standard-4

# SSH into the instance as user "centos"
gcloud compute ssh --zone "us-central1-a" --project "gcp-
example-project" "centos@full-standalone-installation"
```

GOAL

1. Create a new compute instance on a cloud provider (if applicable).
2. Access the server where Collibra DQ will be installed.

2. Download DQ Full Package

Download full package tarball using the signed link to the full package tarball provided by the Collibra DQ Team. Replace `<signed-link-to-full-package>` with the link provided.

```
### Go to the OWL_BASE (home directory of the user is most com-
mon)
### This example uses /home/owldq installing as the user owldq

cd /home/owldq

### Download & untar
curl -o dq-full-package.tar.gz "<signed-link-to-full-package>"
tar -xvf dq-full-package.tar.gz

### Clean-up unnecessary tarball (optional)
rm dq-full-package.tar.gz
```

GOAL

1. Download the full package tarball and place it in the `$OWL_BASE` (home directory). Download via `curl` or upload directly via FTP. The tarball name is assumed to be `dq-full-package.tar.gz` for the sake of simplicity.
2. Untar `dq-full-package.tar.gz` to `OWL_BASE`.

3. Install Collibra DQ + PostgreSQL + Spark

First set some variables for `OWL_BASE` (where to install Collibra DQ. In this tutorial, you are already in the directory that you want to install), `OWL_METASTORE_USER` (the PostgreSQL username used by DQ Web Application to access PostgreSQL storage), and `OWL_METASTORE_PASS` (the PostgreSQL password used by DQ Web Application to access PostgreSQL storage).

```
### The base path that you want Collibra DQ installed under.
Trailing spaces are not permitted.

export OWL_BASE=$(pwd)
export OWL_METASTORE_USER=postgres
# minimum complexity recommended (18 length, upper, lower, num-
ber, symbol)
# example below
export OWL_METASTORE_PASS=H55Mt5EbXh1a%\$aiX6
# The default Spark version is 3.4.1, but you can opt to install
Spark 3.2.2 or 3.0.1 instead.
# Spark 3.4.1
export SPARK_PACKAGE=spark-3.4.1-bin-hadoop3.2.tgz
# Spark 3.2.2
export SPARK_PACKAGE=spark-3.2.2-bin-hadoop3.2.tgz
```

> Warning The `$` symbol is not a supported special character in your PostgreSQL Metastore password.

`dq-package-full.tar.gz` that you untarred contains installation packages for Java 9 or Java 11, PostgreSQL 11, and Spark. There is no need to download these components. These off-line installation components are located in `$(pwd)/package/install-packages`.

```
[centos@install ~]$ pwd
/home/centos
[centos@install ~]$ ls packages/install-packages/
postgresql11-11.9-1PGDG.rhel7.x86_64.rpm          postgresql11-libs-11.9-1PGDG.rhel7.x86_64.rpm    spark-3.0.1-bin-hadoop3.2.tgz
postgresql11-contrib-11.9-1PGDG.rhel7.x86_64.rpm  postgresql11-server-11.9-1PGDG.rhel7.x86_64.rpm  spark-extras.tar.gz
[centos@install ~]$
```

One of the files extracted from the tarball is `setup.sh`. This script installs Collibra DQ and the required components. If a component already exist (for example, Java 8 is already

installed and `$JAVA_HOME` is set), then that component is not installed (i.e. Java 8 installation is skipped).

To control which components are installed, use the `-options=...` parameter. The argument provided should be a comma-delimited list of components to install (valid inputs: `spark`, `postgres`, `owlweb`, and `owlagent`. `-options=postgres,spark,owlweb,owlagent` means "install PostgreSQL, Spark pseudo cluster, DQ Web Application, and DQ Agent". Note that Java is not part of the options. Java 8 or Java 11 installation is automatically checked and installed/skipped depending on availability.

At a minimum, you must specify `-options=spark,owlweb,owlagent` if you independently installed PostgreSQL or use an external PostgreSQL connection (as shown in Step 4, if you choose that installation route).

```
### The following installs PostgresDB locally as part of Col-
libra DQ install

./setup.sh \
    -owlbase=$OWL_BASE \
    -user=$OWL_METASTORE_USER \
    -pgpassword=$OWL_METASTORE_PASS \
    -options=postgres,spark,owlweb,owlagent
```

> Warning The `$` symbol is not a supported special character in your PostgreSQL Metastore password.

> Note If you are prompted to install Java 8 or Java 11 because you do not have one of them installed, accept to install from a local package.

You are prompted to select a location to install PostgreSQL, as shown below:

```
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 ] =
```

If the data files for the PostgreSQL database need to be hosted at a specific location, provide it during this prompt. **Ensure the directory is writable**. Otherwise, press <Enter> to install the data files into `$OWL_BASE/owl/postgres/data`. The default suggested path does not have permission issues if you use `OWL_BASE` as the home directory.

If no exceptions occur and the installation is successful, then the process completes with the following output:

```
installing owlweb
starting owlweb
starting owl-web
installing agent
not starting agent
install complete
please use owl owlmanage utility to configure license key and
start owl-agent after owl-web successfully starts up
```

GOAL

1. Specify `OWL_BASE` path where Collibra DQ will be installed and specify PostgreSQL environment variables.
2. Install DQ Web with PostgreSQL and Spark linked to DQ Agent (**all files are in the `$OWL_BASE/owl` sub-directory**) using `setup.sh` script provided. The location of `OWL_BASE` and PostgreSQL are configurable, but we advise you take the defaults.

4. Install DQ + Spark and use existing PostgreSQL (advanced)

> Note Skip Step 4 if you opted to install PostgreSQL and performed Step 3 instead.
>
> We recommend Step 4 over Step 2 for advanced DQ Installer.

If you already installed DQ in the previous step, then skip this step. This is only for those who want to use external PostgreSQL (e.g. use GCP Cloud SQL service as the PostgreSQL metadata storage). If you have an existing PostgreSQL installation, then everything in the previous step applies except the PostgreSQL data path prompt and the `setup.sh` command.

Refer to Step 3 for details on `OWL_BASE`, `OWL_METASTORE_USER` , and `OWL_METASTORE_PASS`.

```
# The base path that you want Collibra DQ installed. No trailing

export OWL_BASE=$(pwd)
export OWL_METASTORE_USER=postgres

# minimum complexity recommended (18 length, upper, lower, num-
ber, symbol)
# example below
export OWL_METASTORE_PASS=H55Mt5EbXh1a%$aiX6
```

Run the following installation script. Note the missing "Postgres" in `-options` and new parameter `-pgserver`. This `-pgserver` could point to any URL that the standalone instance has access to.

```
# The following does not install PostgresDB and
# uses existing PostgresDB server located in localhost:5432 with
"postgres" database
./setup.sh \
    -owlbase=$OWL_BASE \
    -user=$OWL_METASTORE_USER \
    -pgpassword=$OWL_METASTORE_PASS \
    -options=spark,owlweb,owlagent \
    -pgserver="localhost:5432/postgres"
```

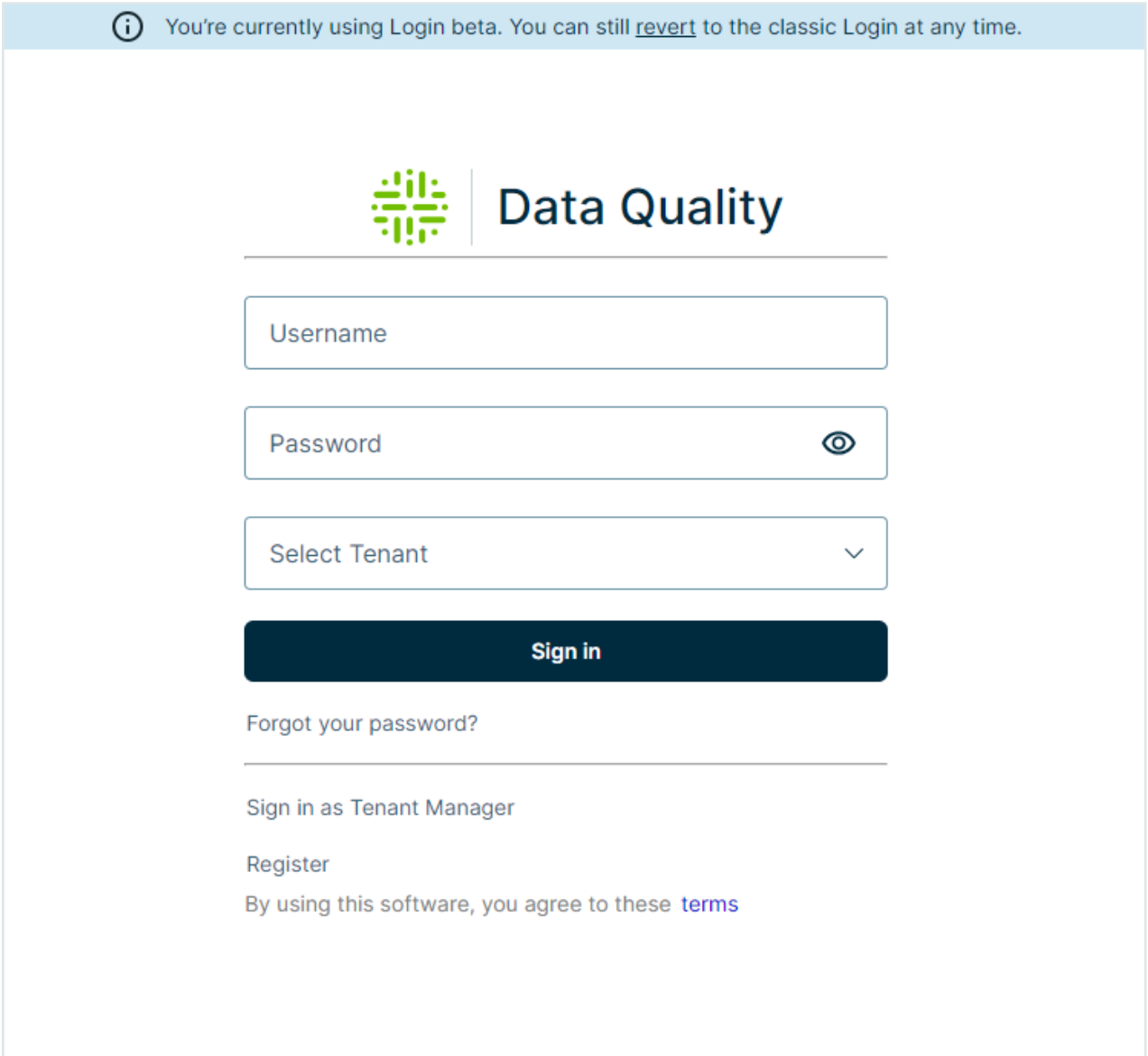
The database named `postgres` is used as the default DQ metadata storage. Changing this database name is out-of-scope for Full Standalone Installation. Contact the Collibra DQ team for assistance.

GOAL

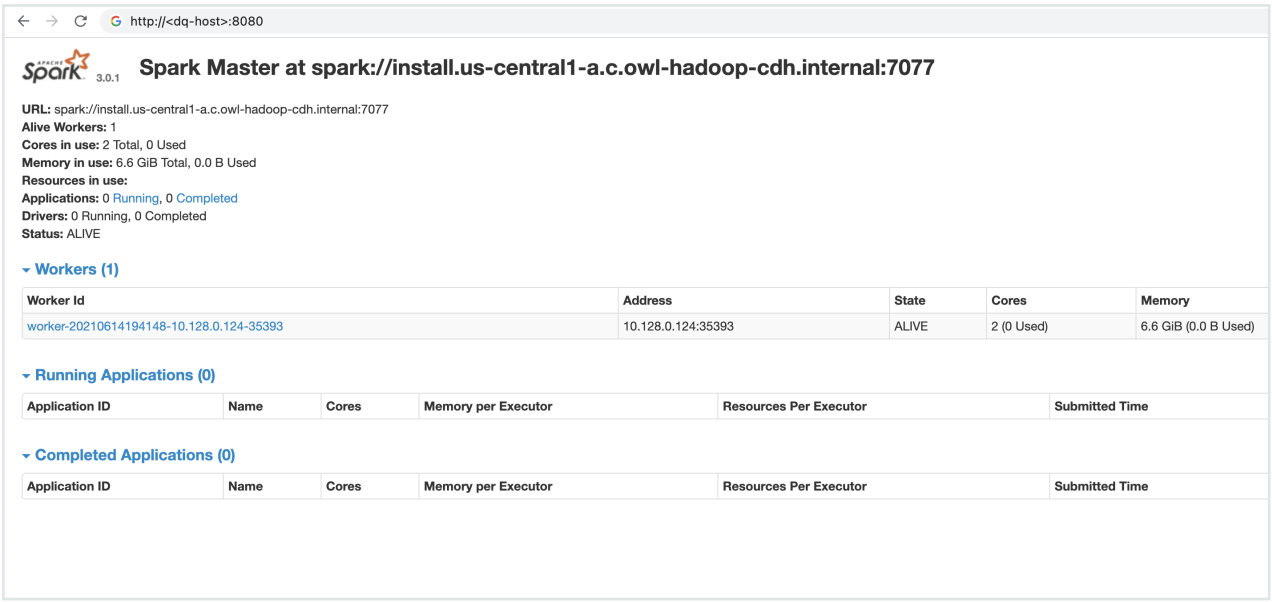
1. Specify `OWL_BASE` path where Collibra DQ will be installed and specify PostgreSQL environment variables
2. Install DQ Web and Spark linked to DQ Agent (**all files will be in `$OWL_BASE/owl` sub-directory**) using `setup.sh` script provided and link DQ Web to an existing PostgreSQL server.

5. Verify DQ and Spark Installation

The installation process will start the DQ Web Application. This process initializes the PostgreSQL metadata storage schema in PostgreSQL under the database named `postgres`. This process must complete successfully before the DQ Agent can be started. Wait approximately 1 minute for the PostgreSQL metadata storage schema to be populated. If you can access DQ Web using `<url-to-dq-web>:9000` using a Web browser, then this means you have successfully installed DQ.



Next, verify that the Spark Cluster has started and is available to run DQ checks using `<url-to-dq-web>:8080`. Take note of the Spark Master URL starting with `spark://...`. This is required during DQ Agent configuration.



6. Set License Key

In order for Collibra DQ to run checks on data, the DQ Agent must be configured with a license key. Replace `<license-key>` with a valid license key provided by Collibra.

> Note Your license key is the value following `YOUR KEY IS =` in the license provision email.

```
cd $OWL_BASE/owl/bin
./owlmanage.sh setlic=<license-key>

# expected output:
# > License Accepted new date: <expiration-date>
```

7. Set License Name

It is required that you set a license name upon your initial deployment of Collibra DQ.

Replace `<your-license-name>` with a valid license name provided by Collibra.

> Note Your license name is the value following `YOUR NAME IS =` in the license provision email.

```
vi /<install-dir>/owl/config/owl-env.sh
export LICENSE_NAME=<your-license-name>
```

8. Set DQ Agent Configuration

Start the DQ Agent process to enable processing of DQ checks.

```
# 1 Start the agent to create the agent.properties file.
cd $OWL_BASE/owl/bin
./owlmanage.sh start=owlagent

# 2 Stop the agent and add this line to agent.properties:
./owlmanage.sh stop=owlagent

# 3 Add this line to agent.properties:
sparksubmitmode=native
sparkhome=</your/spark/home/folder>

# 4 Start the agent again.
./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
```

When the script successfully runs, the `$OWL_BASE/owl/config` folder contains a file called `agent.properties`. This file contains agent ID and the number of agents installed in your machine. Since this is the first non-default agent installed, the expected agent ID is 2. Verify that the `agent.properties` file is created. Your`agent.properties` is expected to have a different timestamp, but you should see `agentid=2`.

```
cd $OWL_BASE/owl/config
cat agent.properties

# expected output:
> #Tue Jul 13 22:26:19 UTC 2021
> agentid=2
```

Once the DQ Agent starts, it needs to be configured in DQ Web in order to successfully submit jobs to the local Spark (pseudo) cluster.

The new agent is set up with the template base path `/opt` and install path `/opt/owl`. The `owlmanage.sh start=owlagent` script does not respect `OWL_BASE` environment. **You must edit the Agent Configuration to follow the** `OWL_BASE`.

Follow the steps in the Agent section to configure the newly created DQ Agent and edit the following parameters in DQ Agent #2.

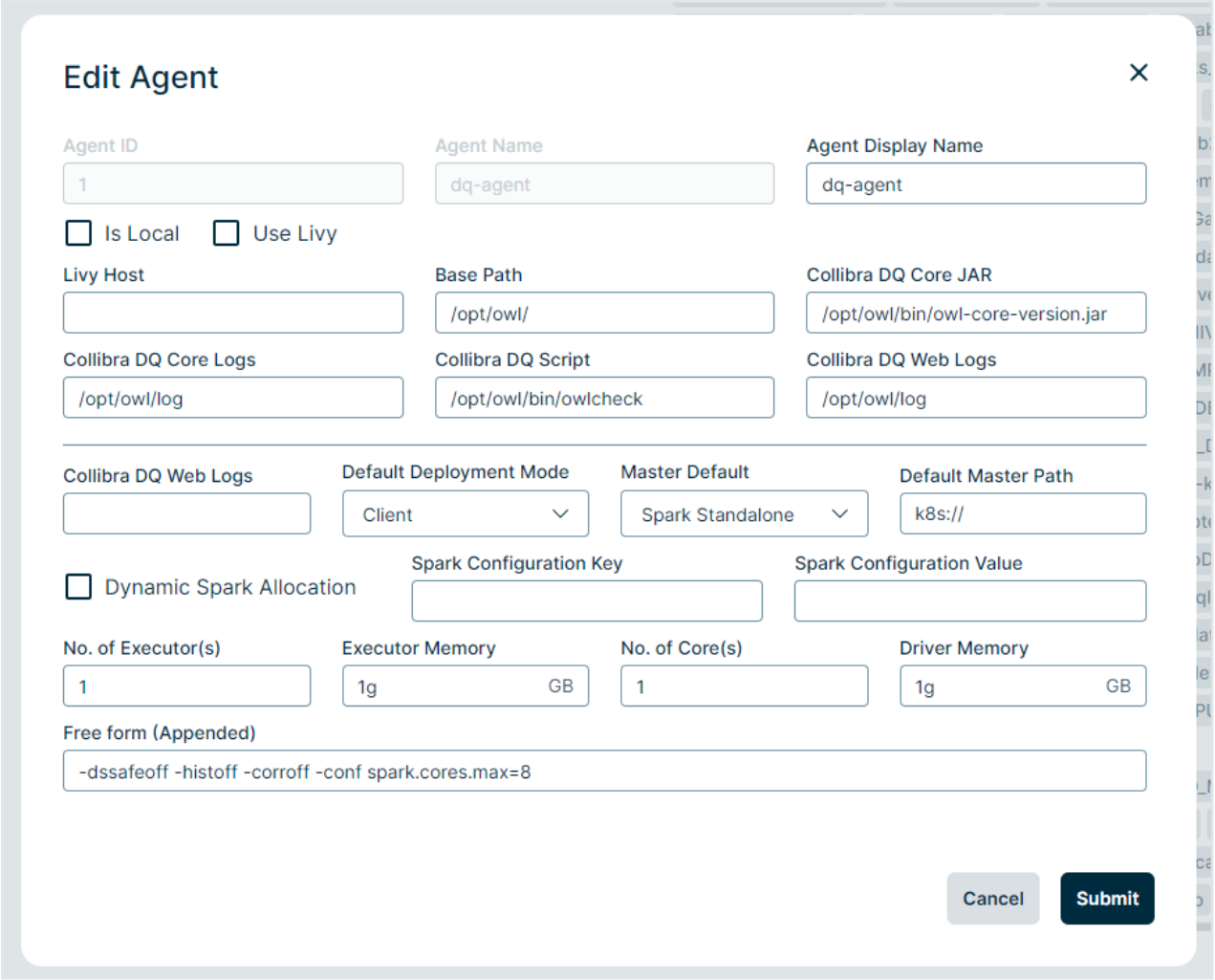
- Replace all occurrence of `/opt/owl` with your `$OWL_BASE/owl/`in **Base Path**, **Collibra DQ Core JAR**, **Collibra DQ Core Logs**, **Collibra DQ Script**, and **Collibra DQ Web Logs**.
    - Note that **Base Path** here does not refer to `OWL_BASE`
- Replace **Default Master** value with the Spark URL from Fig 3
- Replace **Number of Executors(s)**, **Executor Memory (GB)**, **Driver Memory (GB)** to a reasonable default based on the size of your instance.

Refer to the Agent section for descriptions of the parameters.

Specify the **Number of Core(s)**.

To limit Spark cores from being used for each job, a common configuration for the **Free Form (Appended)** field is `-conf spark.cores.max=8`.

Set the **Default Deployment Mode** option to **Client** for a Spark Standalone master.



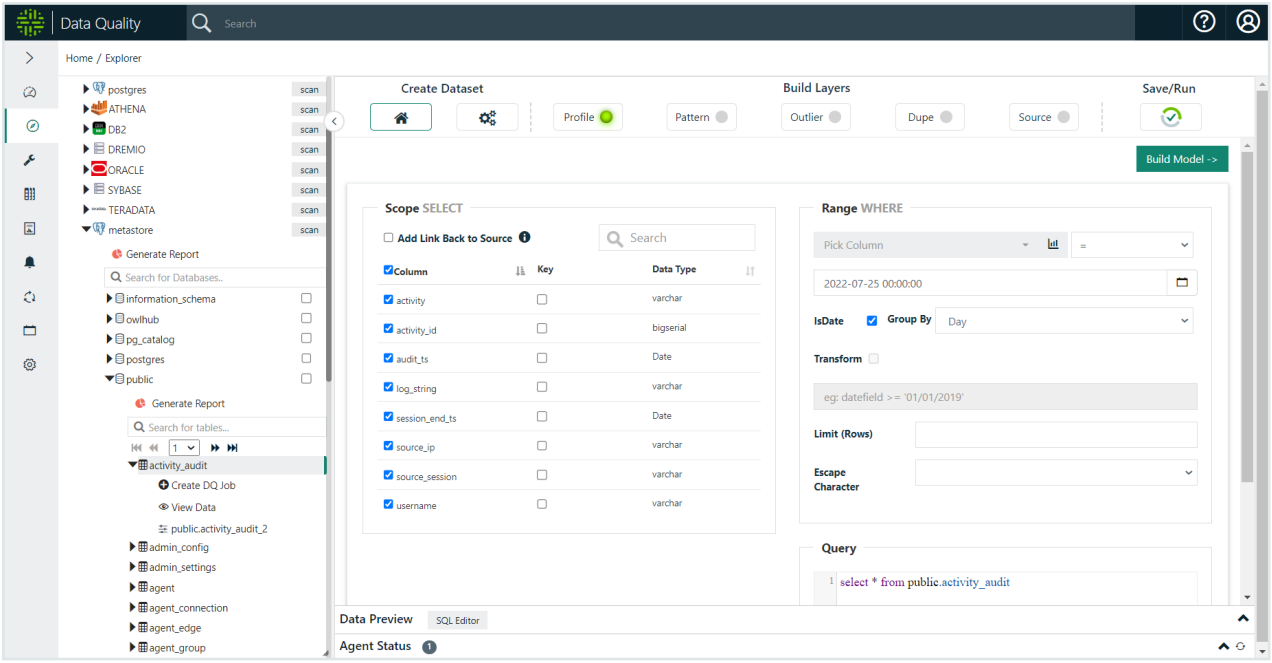
9. Create DB Connection for DQ job

Follow the steps on Agent to add metastore database connection.

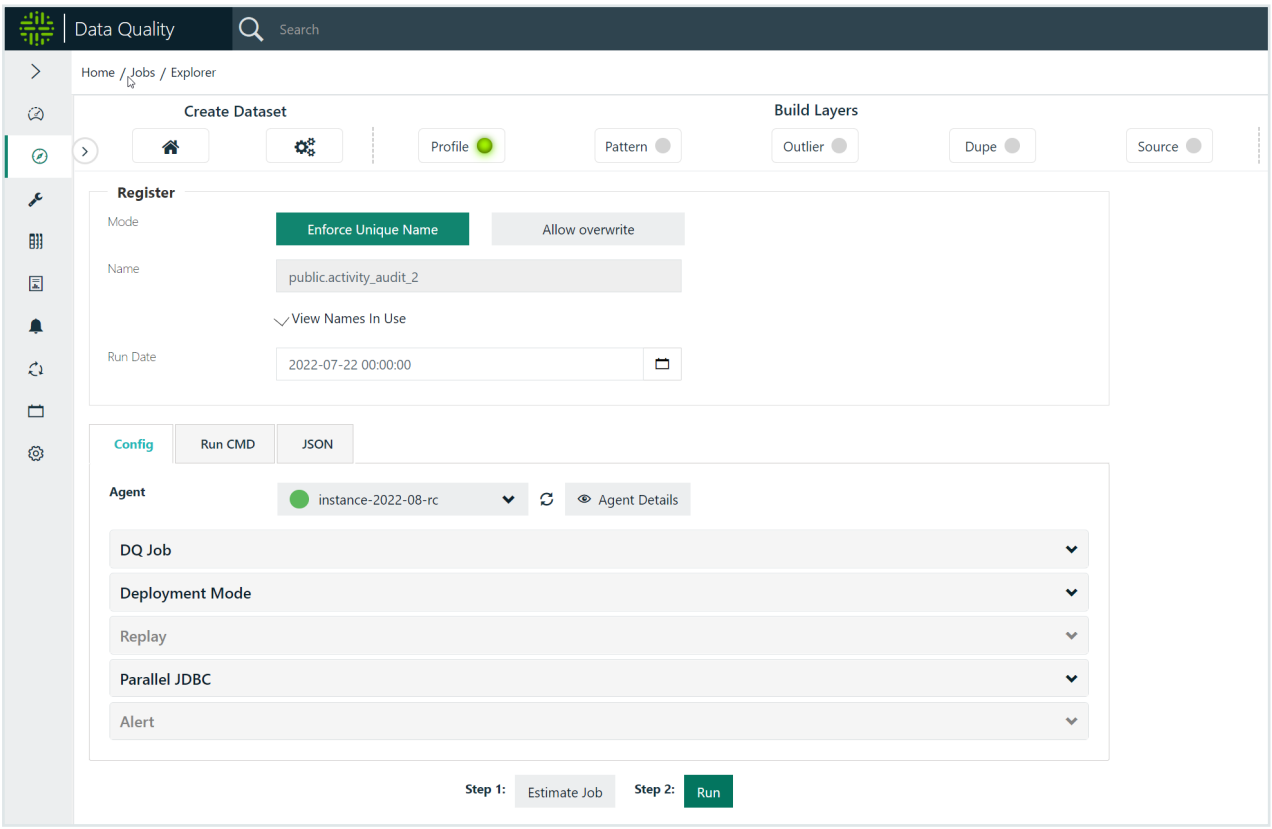
> Note In the following examples, a DQ Job is run against local DQ Metadata Storage.

Follow the steps on Agent section to configure newly created DQ Agent.

Click the compass icon in the navigation pane to open the Explorer page. Click the **metastore** connection, select the **public** schema, and select the first table in the resulting list of tables. From the Preview and Scope page, click **Build Model**. When the Profile page populates, click **Save/Run**.

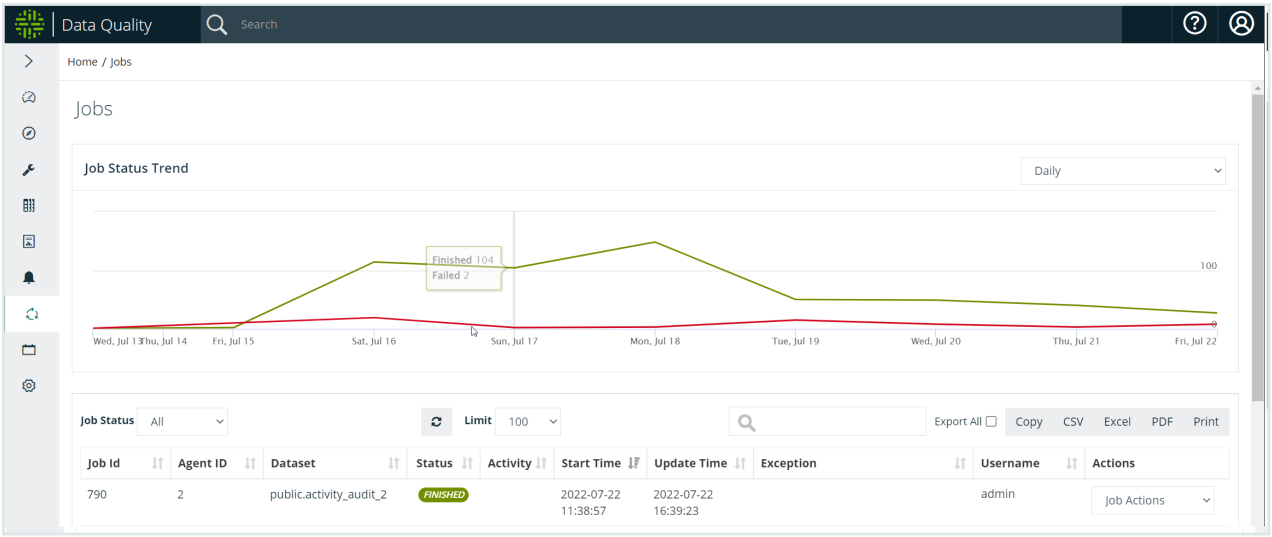


On the Run page, click **Estimate Job**, acknowledge the resource recommendations, and click **Run**.



Click the revolving arrows icon in the left navigation panel to open the Jobs page.

Wait 10 seconds and then click the refresh button above the Status column until the status shows that the DQ job is Finished. We recommend refreshing several times, pausing for a few seconds in between clicks. While a job runs, the Activity column tracks the sequence of activities DQ performs before it completes a job. A successful job shows its status as Finished last.



Troubleshooting + Helpful Commands

```
### Setting permissions on your pem file for ssh access

chmod 400 ~/Downloads/ssh_pem_key
```

Make sure 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
```

Changing 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.

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`

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
```

Getting the hostname of the instance

```
### Getting the hostname of the instance

hostname -f
```

Increase Thread pool / Thread Pool Exhausted

Adding to the owl-env.sh script

```
# vi owl-env.sh
# modify these lines

export SPRING_DATASOURCE_POOL_MAX_WAIT=500
export SPRING_DATASOURCE_POOL_MAX_SIZE=30
export SPRING_DATASOURCE_POOL_INITIAL_SIZE=5

# restart web and agent
```

Adding to the Spark agent configurations

If you see the following message, update the agent configurations within load-spark-env.sh:

```
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 the following configurations to modify the connection pool available:

```
export SPRING_DATASOURCE_POOL_MAX_WAIT=1000
export SPRING_DATASOURCE_POOL_MAX_SIZE=30
export SPRING_DATASOURCE_POOL_INITIAL_SIZE=5
```

> Note The load-spark-env.sh file is located in the $SPARK_HOME/bin folder.

Adding to the owl.properties file

Depending on client vs. cluster mode and cluster type, you may also need to add the following configurations in the owl.properties file:

```
spring.datasource.tomcat.initial-size=5
spring.datasource.tomcat.max-active=30
spring.datasource.tomcat.max-wait=1000
```

Active Database Queries

```
select * from pg_stat_activity where state='active'
```

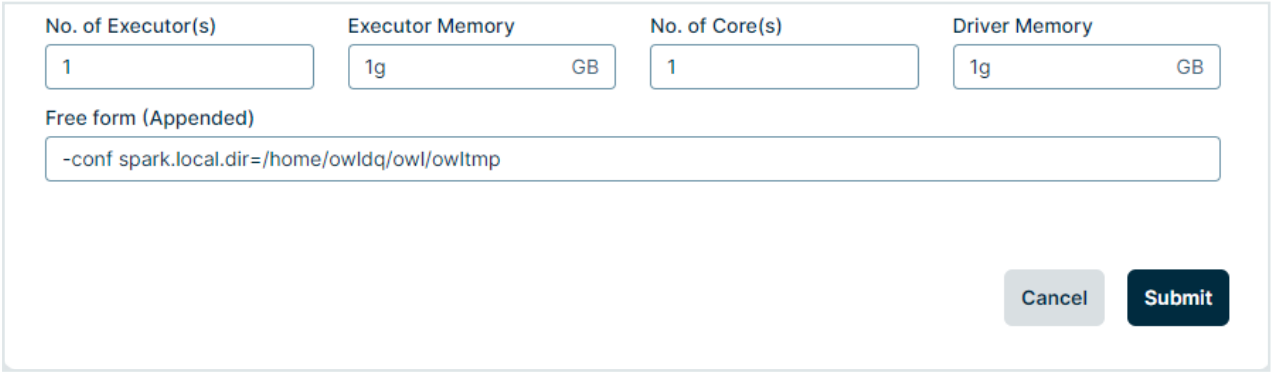
Too many open files configuration

```
### "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)
```

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
```

Tip: 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
```

Check 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
```

Starting Components

```
### Restart different components

cd /home/owldq/owl/bin/
./owlmanage.sh start=postgres
./owlmanage.sh start=owlagent
./owlmanage.sh start=owlweb

cd /home/owldq/owl/spark/sbin/
./stop-all.sh
./start-all.sh
```

Configuration Options

Setup.sh arguments

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 -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/`

Example installing just the agent (perhaps on an Edge node of a hadoop cluster):

`./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/`

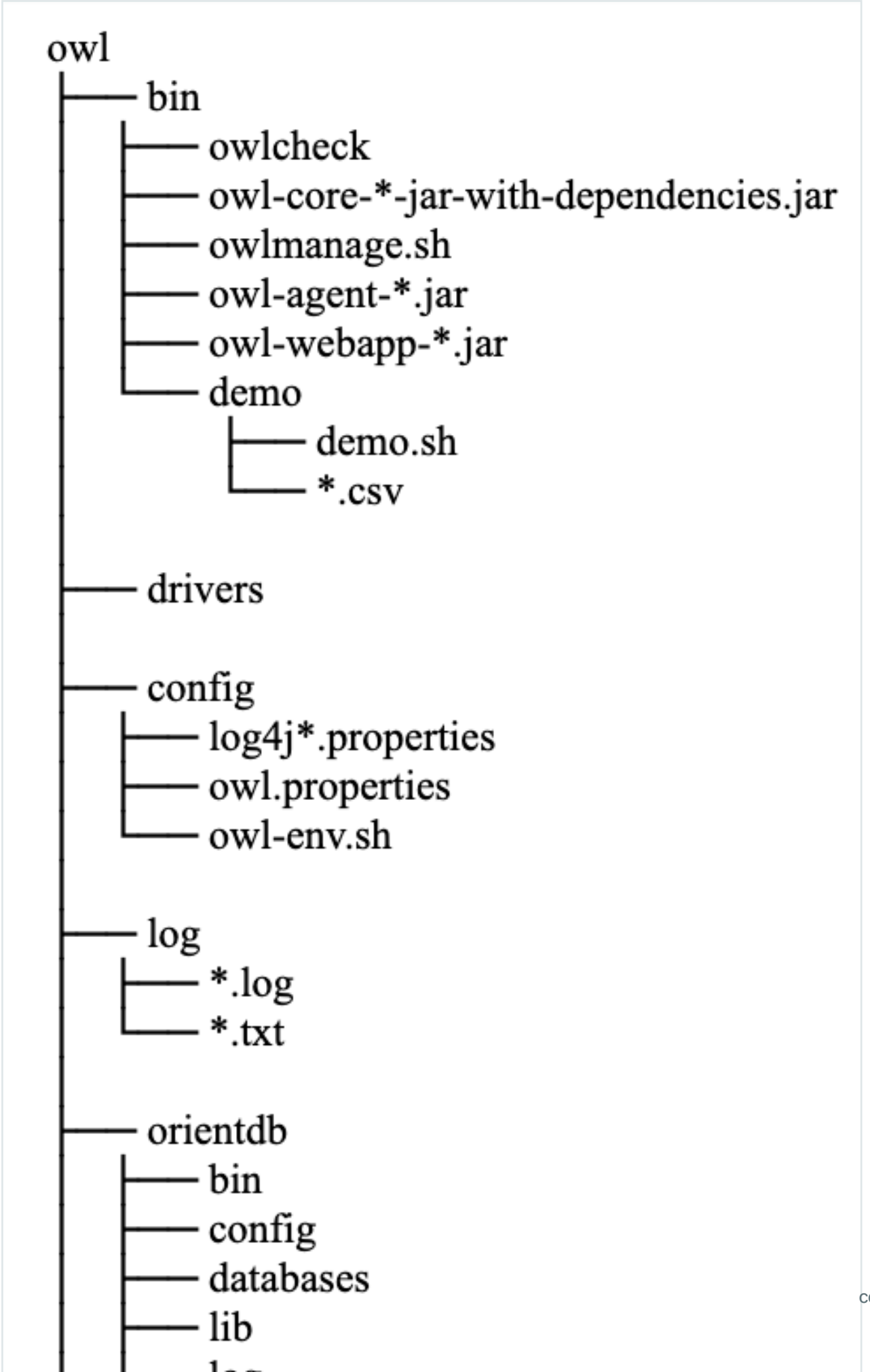
When installing different features, the following questions are asked:

- postgres = Postgres DBPassword needs to be supplied.
- If postgres is not being installed (such as agent install only) postgres metastore server name needs to be supplied.

Launching and Administering Collibra DQ

When the setup.sh script finishes by default software is automatically started. The setup.sh also creates the owlmanage.sh script which allows for stopping and starting of all owl services or some components of services. The setup script will also generate an owl-env.sh script that will hold the main variables that are reused across components (see owl-env.sh under the config directory).

Collibra DQ Directory Structure after running setup.sh


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

ccxcv

Configuration of ENV settings within owl-env.sh

Contents of the owl-env.sh script and what the script is used for.

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.
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.

owl-env.sh Scripts	Description
export ext_pass_script="/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.
PORT=9003 #owl-web port NUMBER	The default port to use for owl-web.
SPRING_LOG_LEVEL=ERROR	The logging level to be displayed in the owl-web.log
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 DQ Postgres metastore.
export SPRING_DATASOURCE_USERNAME-E=collibra	Collibra DQ PostgreSQL username.
export SPRING_DATASOURCE_PASSWORD-D=3+017wfY1I1vmsvGYAyUcw5zGL	Collibra DQ PostgreSQL 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
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

owl-env.sh Scripts	Description
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=t2IMFWEHsQha3QaWnNaR8ALaFPH15Mh9	Certificate key password.
export SERVER_SSL_KEY_ALIAS=owl	Certificate key alias.
export SERVER_REQUIRE_SSL=true	Override HTTP on and force HTTPS regard-less of HTTP settings.
export MULTITENANTMODE=FALSE	Flipping to TRUE will enable multi tenant support.
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**.

owl-env.sh Scripts	Description
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>**.

Configuration of owl.properties file

Example	Description
key=XXXXXX	The license key.
spring.data-source.url=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.
spring.agent.datasource.url	jdb-c:postgresql://$host:$port/owltrunk
spring.agent.datasource.username	{user}
spring.agent.datasource.passwords	{password}
spring.agent.datasource.driver-class-name	org.postgresql.Driver

Starting Spark

Spark Standalone Mode - Spark 3.2.0 Documentation

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.
`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.

Shell Script	Description
`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 Spark Standalone

cd /home/owldq/owl/spark/sbin
./start-all.sh

### Stopping Spark
cd /home/owldq/owl/spark/sbin
./stop-all.sh
```

```
### Starting Spark 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
```

Standalone Install (Script)

Requirements

Resource	Notes	Provided by
OS	Red Hat 7/Red Hat 8, or CentOS 7/CentOS 8 > Note For RHEL 8 or CentOS 8, Postgres is automatically installed when you run the script.	Customer
Memory	16 GB Ram	
Cores	8	
Storage	50 GB Disk	
Permission	`sudo`	
Install Script	Download using the curl command below	Collibra

Access the machine through either a cloud shell or SSH. The snippet below is an example SSH command.

```
ssh -i <your_key> <user>@<public-ipv4-ip-address>
```

```
ssh -i ~/.ssh/abc.pem centos@22.000.111.3333
```

Your pem file should have correct permissions. Confirm you have the correct permissions with the command: `sudo chmod 400 <your_key>`

Steps

After you enter your SSH into the Centos or Redhat VM, run these commands from the command line:

Step 1: Download the script

```
curl -o cdq_install.sh https://owl-pack-
ages.s3.amazonaws.com/MP/cdq_install.sh
```

Step 2: Modify script permission

```
sudo chmod +x cdq_install.sh
```

Step 3: Run the script

```
echo | ./cdq_install.sh
```

> Note The most common directory to use for installation is your user directory (/home/<user>). You do not need to create any directories because the install script creates the correct directory structure. You should have `sudo` access to perform the installation and the directory should not be a restricted system directory.

> Note If you run the setup script in a non-interactive mode, such as an automated deployment, you must export the following variables for the script to run correctly:
>
> ```
> export DQ_ADMIN_USER_PASSWORD
> export DQ_ADMIN_USER_EMAIL
> ```

In this step, you are prompted to enter a default admin user email and password associated with the default admin user 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).

> Warning As part of Collibra's password policy, you cannot proceed with installation until you enter a valid password.

Step 4: Click the URL

Sign in to the application with the username `admin` and the default admin password you provided in the setup script.

```
http://<server_name/ip>:9000
```

> Note Make sure you have access to the server and port. Adding the correct security group or whitelisting your IP address is a common step to be able to access an application running on a cloud server.

BYOL (Optional)

Using the same command, you can bring your own license or use a different download link.

```
./cdq_install.sh "<Installer Download Link>" "<License Key>"
```

Confirmation (Optional)

Once the installation script is complete, check the details of the processes on the server.

```
ps -ef | grep -i spark
ps -ef | grep -i owl-web
ps -ef | grep -i owl-agent
ps -ef | grep -i postgres
```

When all of the processes are up and running, sign in to your standalone instance of Collibra Data Quality & Observability.

Configure your Agent and Adding Connections as normal.

Troubleshooting

Common error messages

If you receive an error with the message, "Application already running on port 8080," enter the following command:

```
sudo netstat -plten |grep java
```

You can then use a `kill` command to kill the process.

```
kill -9 <appId>
```

> Note If you receive a "permission denied" error message, make sure that you are using `sudo`.

Known Limitations

The dollar sign `$` is not a supported character for passwords created via Helm during the admin password creation process in Step 3. Because passwords require at least one special character, use one of the following supported characters instead: `!@#%^&*?_~`.

Getting Started with AWS

Prerequisites

To get started, you need a Collibra Data Quality account.

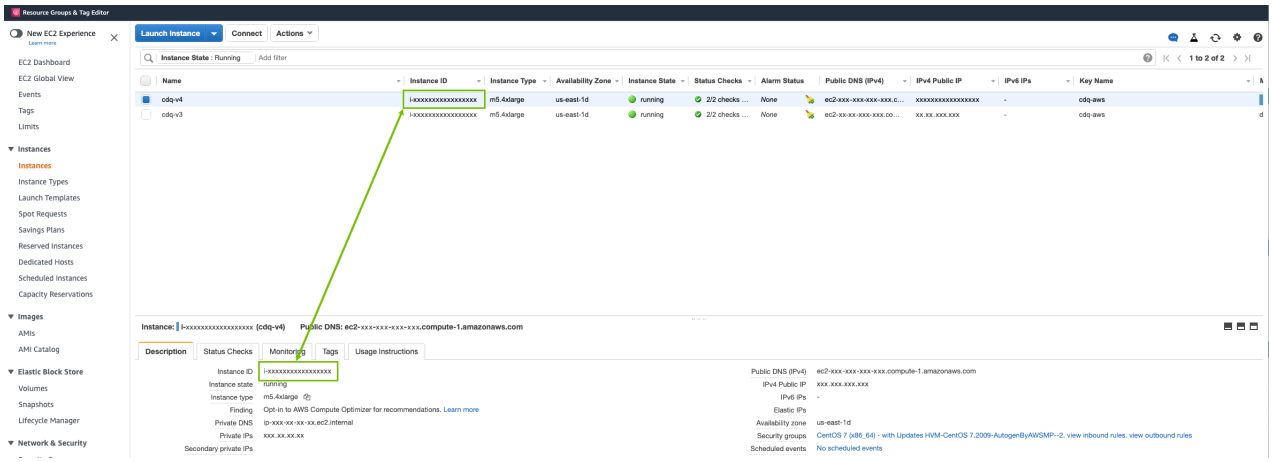
Connecting to Collibra Data Quality

1. To connect to your environment, enter your URL.

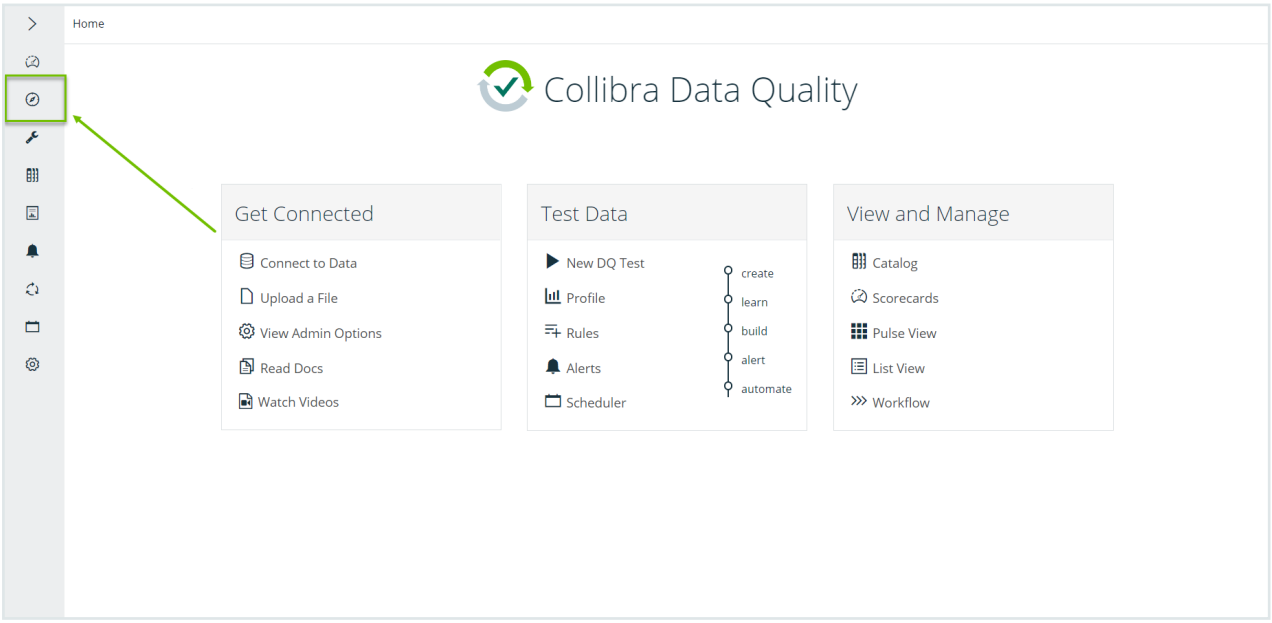
Example: https://<your_host_name>:9000

2. Enter "admin" as your **Username** and your unique Instance ID as your **Password**.

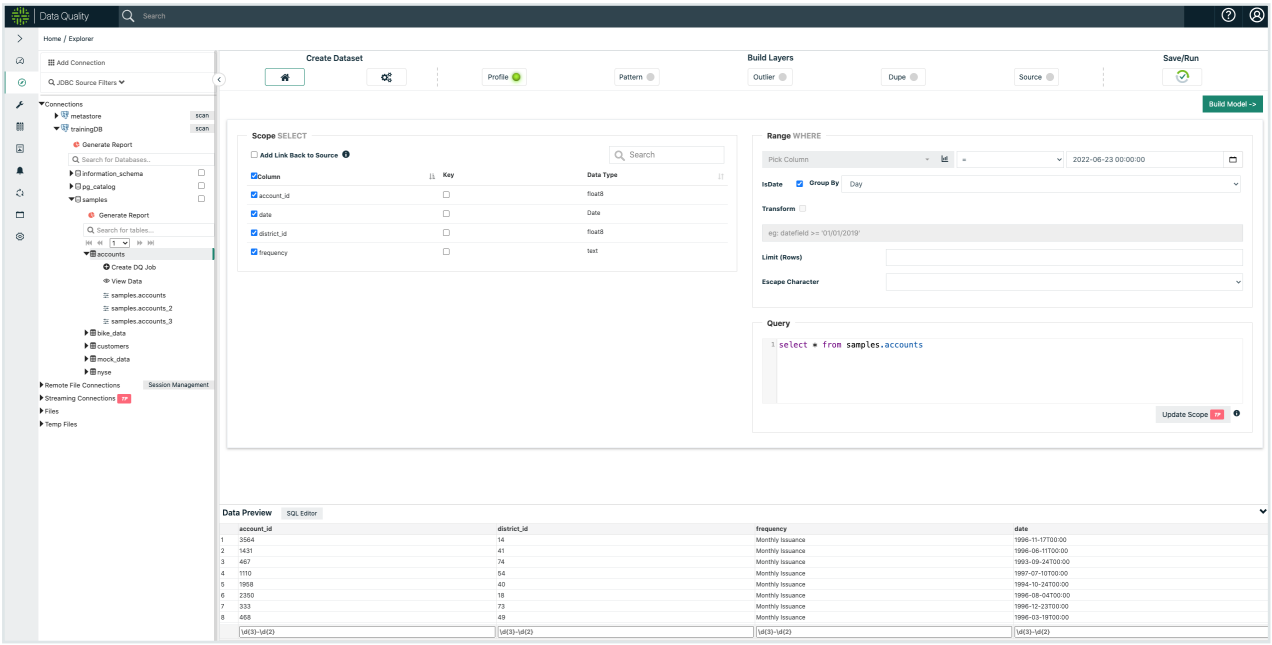
> Note Username: admin Password: CDQ<your_instance_id>



3. From the home page, click the **Explorer** tab.



4. From Explorer, select a connection from the **Connections** dropdown to access your data set.



What's next?

- Visit the Explorer (no-code) for more information on quickly connecting to your data sets.
- Visit Collibra Data Quality's Youtube channel for more tutorials.
- Register for Collibra Data Quality's Product Showcase to learn more about Collibra Data Quality.

Standalone Install (AWS CloudFormation)

This section describes how to install and configure Collibra Data Quality using the AWS CloudFormation stack.

Prerequisites

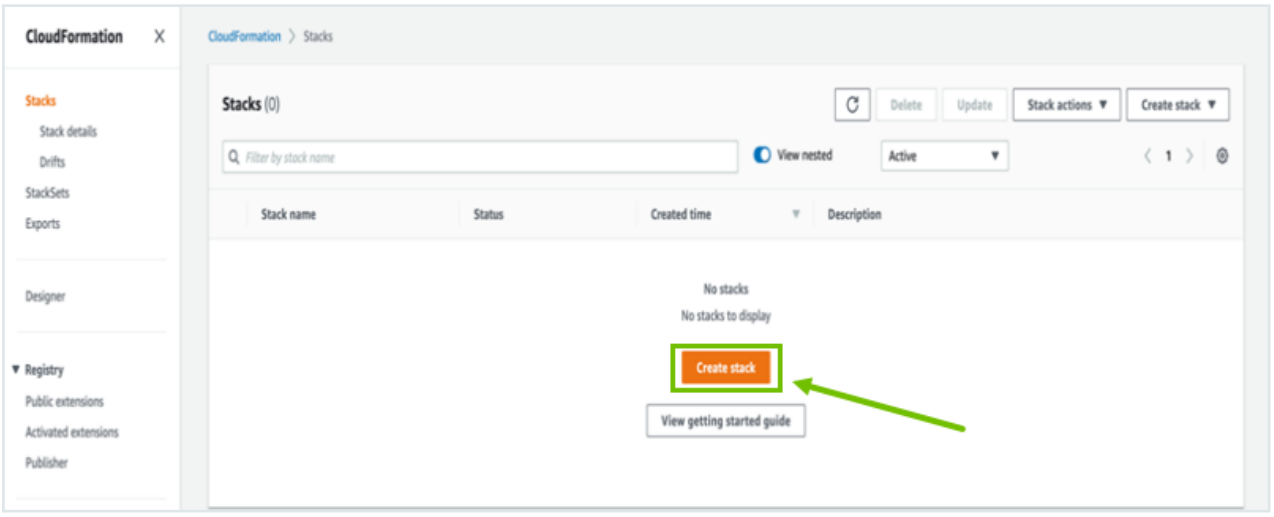
To install Collibra Data Quality using the AWS CloudFormation stack, you need an AWS user account with permissions to provision AWS resources.

Steps

1. Login to your AWS account and navigate to **CloudFormation** from the search bar.

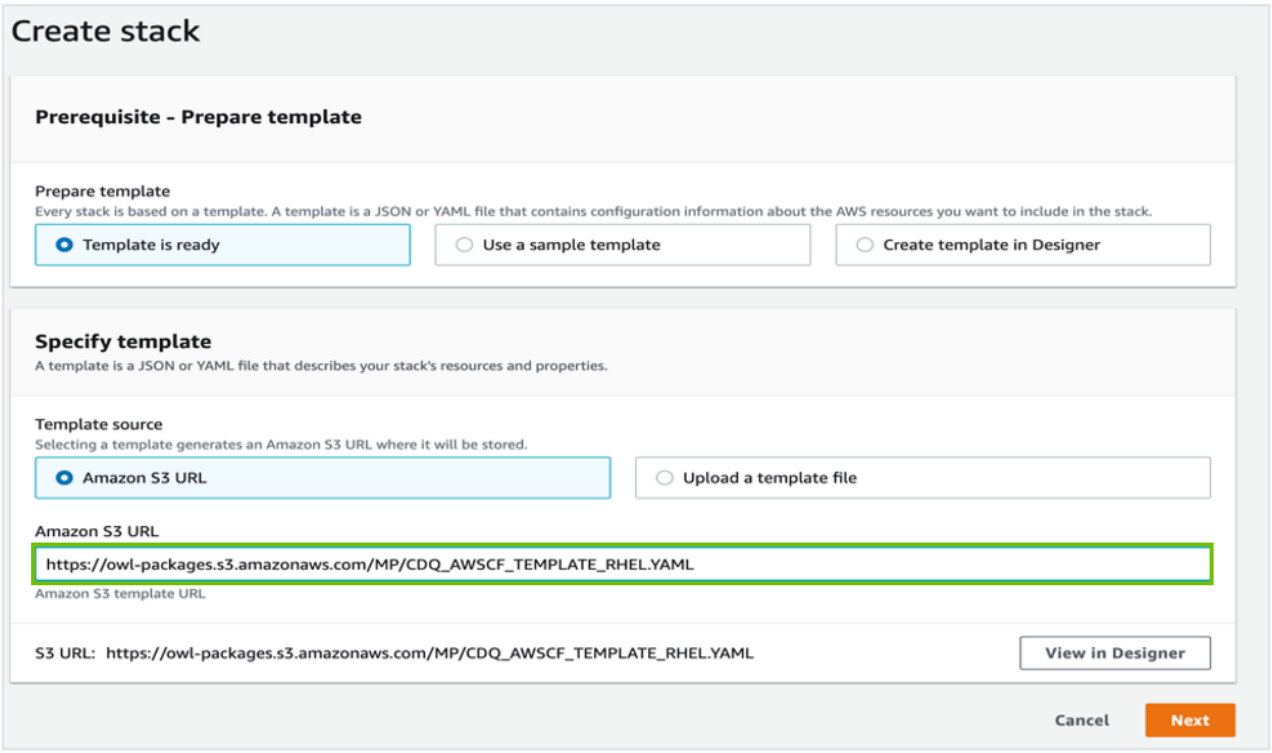
2. In CloudFormation, click the **Create stack** button and provide the S3 location for the template.

```
https://owl-packages.s3.amazonaws.com/MP/CDQ_AWSCF_TEMPLATE_
RHEL.YAML
```



3. Click **Next**.

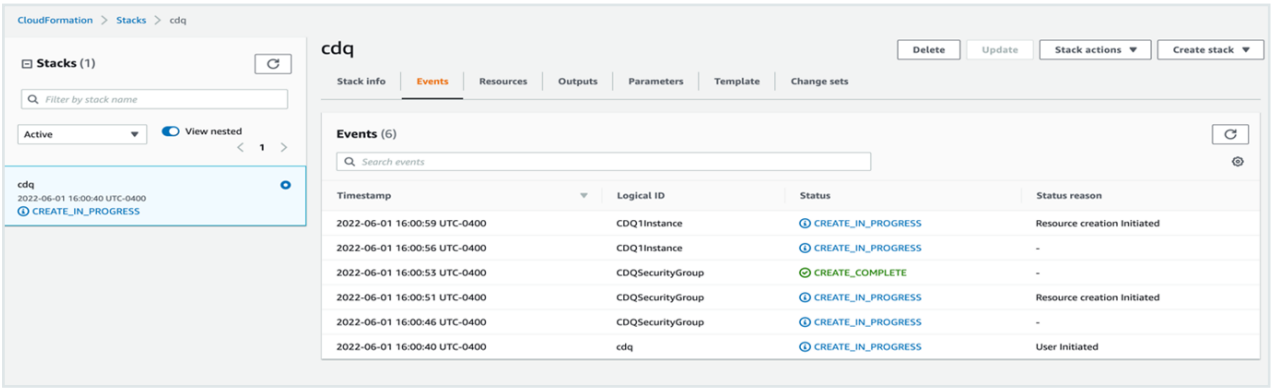
4. Follow the prompts and select the appropriate instance type, VPC, subnet, and key values based on your AWS account. Rest accepts everything by default.



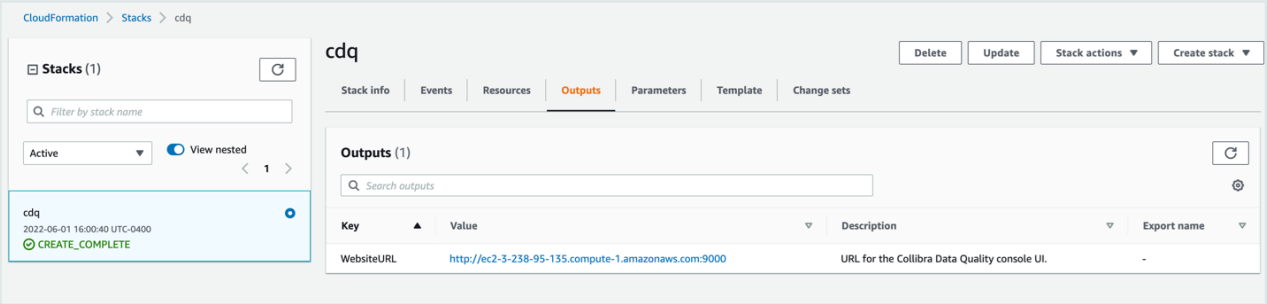
5. Click **Next**.

> Note This process takes 10-15 minutes to spin up the EC2 instance and deploy Collibra DQ.

6. In the **Events** tab, you can monitor status of your stack.



7. Once your stack is created, click the **Outputs** tab to access the CDQ Login URL.



8. In the **Value** column, click the CDQ Login URL.

> Note Your CDQ instance comes with sample data and prewired DB connections.

9. To uninstall and release the resource, delete the stack.

Standalone Install (Google Cloud Platform)

This section provides information on how to deploy Collibra Data Quality on Google Cloud Platform.

Prerequisites

You have:

- A Google user account.
- A project for Google Cloud Platform to deploy Collibra Data Quality.
- Deployment Manager permissions provisioned by an admin.

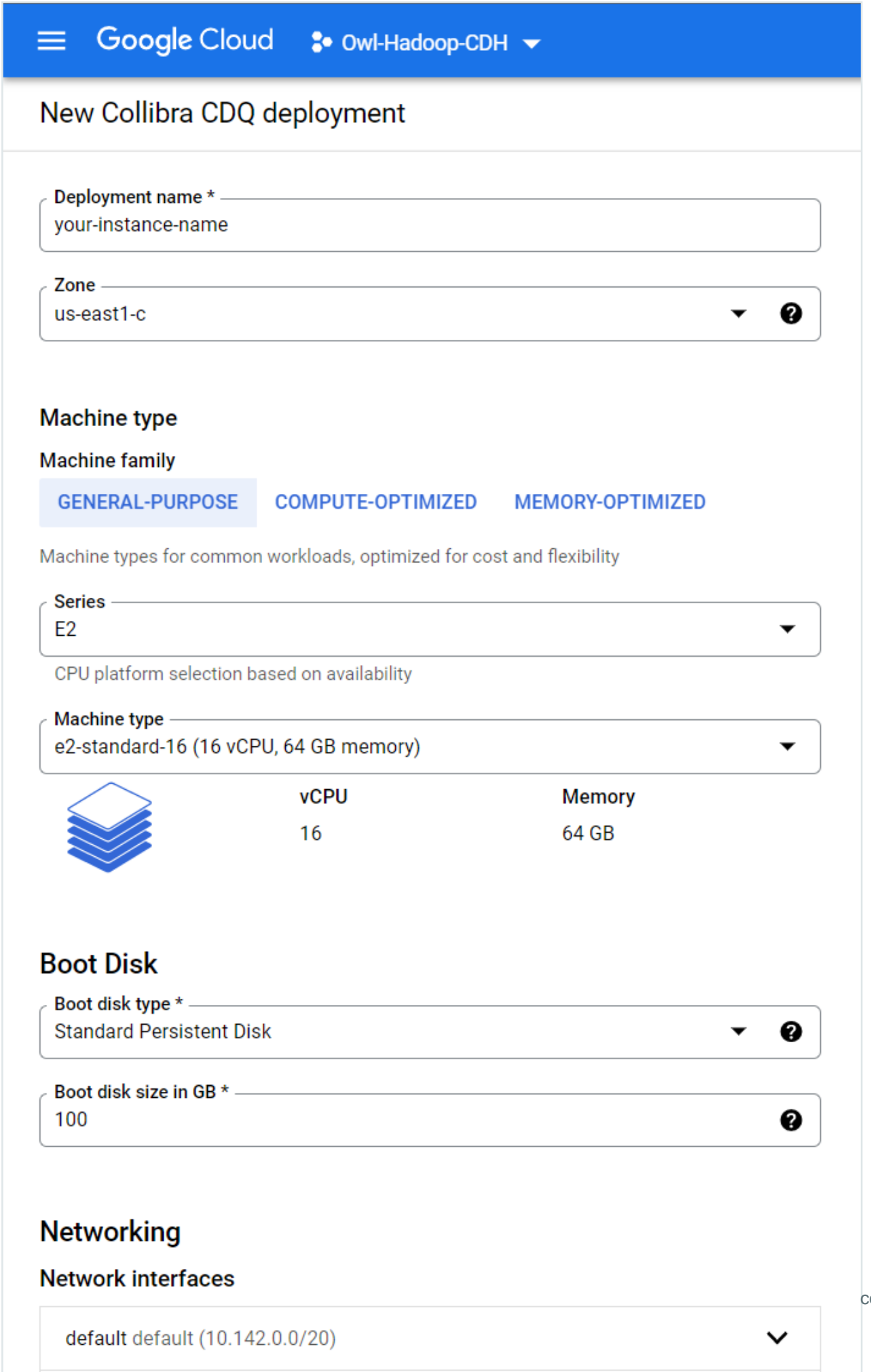
Steps

1. Step 1: Create a new Collibra Data Quality deployment
2. Step 2: Launch a new Collibra Data Quality deployment

Step 1: Create a new Collibra Data Quality deployment

1. Sign in to your Google Cloud Marketplace account and choose your **working project** for Collibra Data Quality deployment.

2. In the search bar, search for ***Collibra*** and press enter.
   >> The search results appear.
3. Select **Collibra CDQ**.
   >> The Collibra CDQ product page opens.
4. Select **Launch**.
   >> The New Collibra CDQ deployment page opens.
5. On the **New Collibra CDQ deployment** page, specify the following information:


Google Cloud
Owl-Hadoop-CDH
New Collibra CDQ deployment
Deployment name *
your-instance-name
Zone
us-east1-c
Machine type
Machine family
GENERAL-PURPOSE
COMPUTE-OPTIMIZED
MEMORY-OPTIMIZED
Machine types for common workloads, optimized for cost and flexibility
Series
E2
CPU platform selection based on availability
Machine type
e2-standard-16 (16 vCPU, 64 GB memory)
vCPU
16
Memory
64 GB
Boot Disk
Boot disk type *
Standard Persistent Disk
Boot disk size in GB *
100
Networking
Network interfaces
default default (10.142.0.0/20)

Field	Description
Deployment name	The name of your Collibra Data Quality deployment.
Zone	Select the zone closest to your region.
Series	The default is E2.
Machine type	The default is e2-standard-16 (16 vCPU, 64 GB memory).
Boot disk type	The default is Standard Persistent Disk.
Boot disk size in GB	The default is 100.

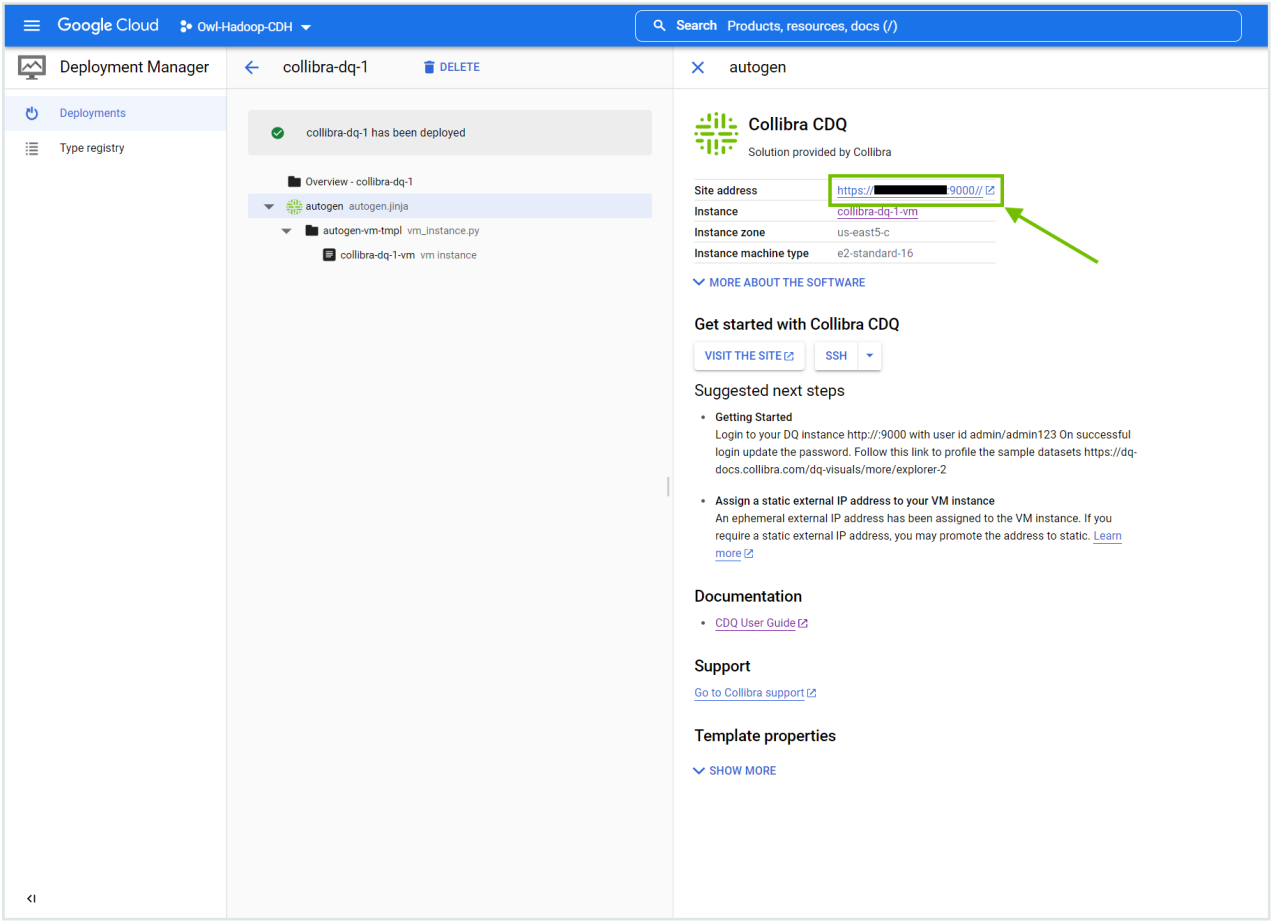
6. Read and accept the **Terms of Service**.

7. Select **Deploy**.
>> The **Deployment Manager** page opens.

> Note GCP also sends you a confirmation email containing a direct link to the Deployment Manager Page.

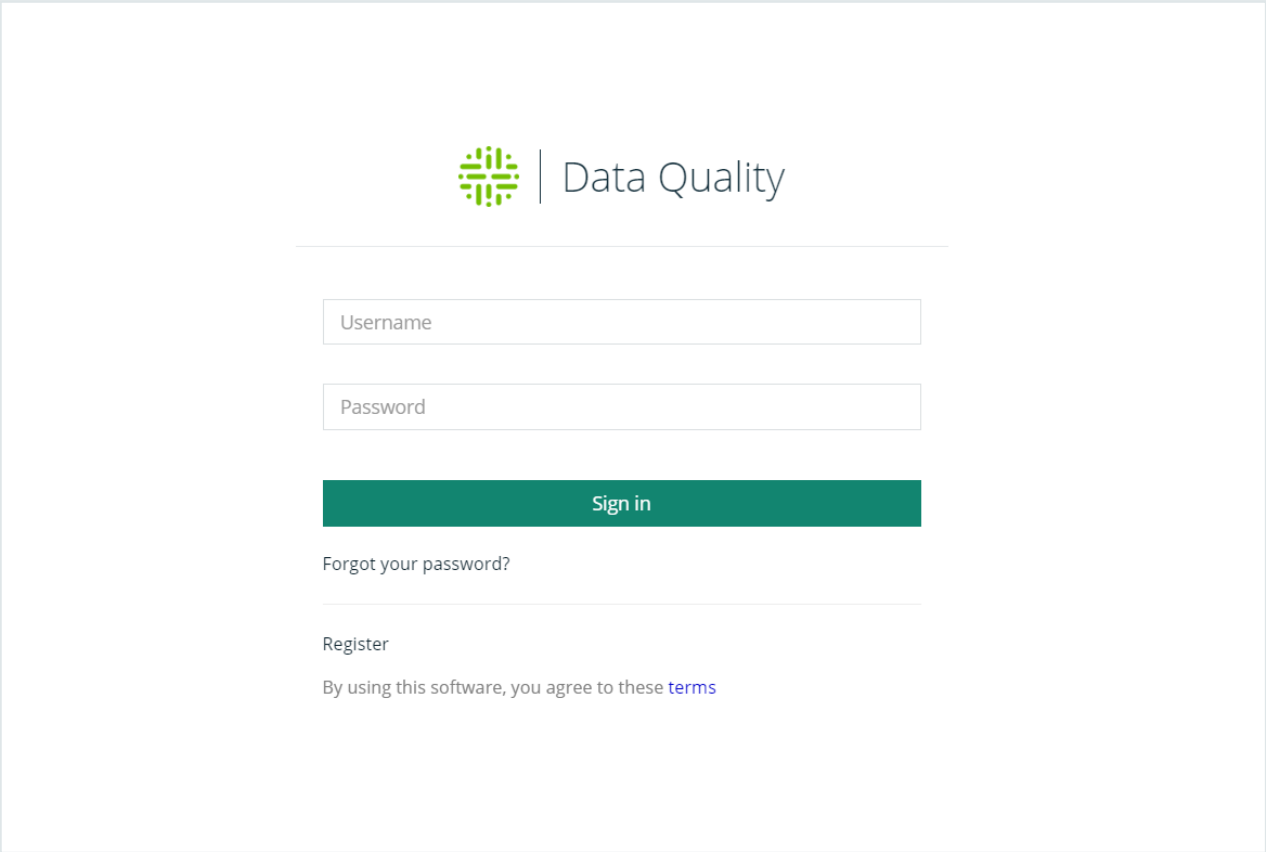
Step 2: Launch a new Collibra Data Quality deployment

1. From the **Deployment Manager**, select the **site address** to sign in to your Collibra Data Quality instance.



2. Sign in to your instance using the following one-time username and password: `admin / admin123` >> The Collibra Data Quality landing page opens.

> Note You must change your password after successfully signing in for the first time. Select the avatar in the upper right of your screen and select the Change Password tab and follow the prompts to change your password.



Troubleshooting your deployment

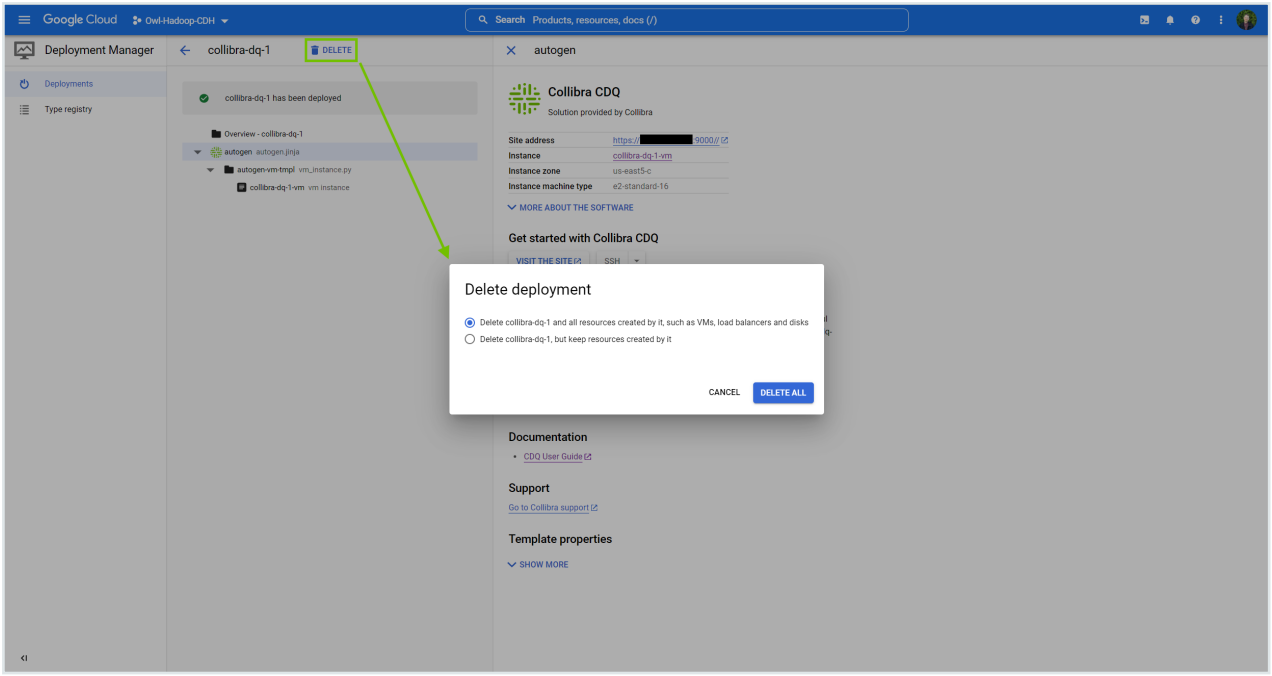
After a successful deployment of Collibra Data Quality on GCP, it is possible that you receive an error message when you select your site address. If this happens, you can:

- Check your network access and verify that you have the appropriate network tags.
- Check that the firewall entry is properly defined for your install.
- Check the URL and remove the **s** from http**s**. Also remove the second trailing forward slash / after :9000. Correct: http://<your.instance>:9000/

Deleting your deployment

To delete your deployment, go to the **Deployment Manager**. Select **Delete** at the top of the screen and then from the dialog box, choose to either

- Delete your deployment and all of its resources.
- Delete your deployment but keep its resources.



When you select a deletion method, your deployment is permanently removed from the list of deployments on the Deployment Manager page.

What's next?

- Visit the Explorer (no-code) for more information on quickly connecting to your data sets.
- Visit Collibra Data Quality's Youtube channel for more tutorials.

Standalone upgrade

> 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.

Download the Collibra DQ Upgrade Package

> Note Starting with the December 2021 release, 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.

Download the tarball using the signed link to the full package tarball provided by Collibra. Replace `<signed-link-to-full-package>` with the link provided.

```
### Go to the OWL_BASE (home directory of the user is most com-
mon)
### This example we will use /home/owldq installing as the user
owldq

cd /home/owldq

### Download & untar
curl -o dq-full-package.tar.gz "<signed-link-to-full-package>"
tar -xvf dq-full-package.tar.gz

### Clean-up unnecessary tarball (optional)
rm dq-full-package.tar.gz
```

Upgrade steps

Upgrading from a Collibra DQ version older than 2023.09 to a different version older than 2023.09

This section provides steps to upgrade standalone deployments of Collibra DQ from a version older than 2023.09 to a different version older than 2023.09. To upgrade from a version before 2023.09 to one from 2023.09 or later, follow the steps outlined in the next section instead.

1. Copy the contents of the provided package, for example, owl-<new-version>-<SPARK301>-package-full.tar.gz to the system being upgraded (extract

contents).

> Tip Untar the contents into a uniquely named folder, for example, *2023-05-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 owl-webapp-<oldversion>-<spark301>.jar /tmp
mv owl-agent-<oldversion>-<spark301>.jar /tmp
mv owl-core-<oldversion>-<spark301>.jar /tmp
```

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

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

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

Upgrading Collibra DQ from a version older than 2023.09 to version 2023.09 or newer

This section provides steps to upgrade standalone deployments of Collibra DQ from a version older than 2023.09 to version 2023.09 or newer.

1. Copy the contents of the provided package, for example, dq-<new-version>-<SPARK301>-package-full.tar.gz to the system being upgraded (extract contents).

> Tip Untar the contents into a uniquely named folder, for example *2023-09-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 owl-webapp-<oldversion>-<spark301>.jar /tmp
mv owl-agent-<oldversion>-<spark301>.jar /tmp
mv owl-core-<oldversion>-<spark301>.jar /tmp
```

5. 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
```

6. 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`.

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
```

Upgrading Collibra DQ from version 2023.09 or newer to a version newer than 2023.09

This section provides steps to upgrade standalone deployments of Collibra DQ from version 2023.09 or newer to a version newer than 2023.09.

1. Copy the contents of the provided package, for example, dq-<new-version>-<SPARK301>-package-full.tar.gz to the system being upgraded (extract contents).

   > Tip Untar the contents into a uniquely named folder, for example *2023-11-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>-<spark301>.jar /tmp
mv dq-agent-<oldversion>-<spark301>.jar /tmp
mv dq-core-<oldversion>-<spark301>.jar /tmp
```

5. 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
```

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

Additional Log4J notes

Place Log4j-1.2-api-2.17.1.jar (as of 2022.02) into /<install-home>/owl/spark/jars

> Note For versions 2021.12 and 2022.01, the name of the package is **Log4j-1.2-api-2.17.0.jar**.

> Action For all Collibra DQ customers, particularly those leveraging CLI mode.

1. Go to the same folder where you extracted the Collibra-provided upgrade package.
2. Go to **<location_of_2022-02-dq-upgrade>/packages/install-packages**.
3. Extract **log4j-1.2-api-2.17.1.jar** with the following command:

```
tar -xvf spark-extras.tar.gz spark-extras/log4j-1.2-api-2.17.1.jar
```

4. Move the **log4j-1.2-api-2.17.1.jar** file to the **/<install-path>/spark/jars** folder.

FAQ

Q: (When) do I need to move **log4j-1.2-api-2.17.1.jar** before or after swapping the main Collibra DQ jars?

- A: The sequence does not matter.

Q: (What) if I don't follow these additional upgrade steps?

- A: If your `SPARK_SUBMIT_MODE` within owl-env.sh is set to `SPARK_SUBMIT_MODE-E=native`, Collibra DQ will function properly without the above additional upgrade step, with the exception of CLI mode.

Remove a legacy properties file

> Action Only for Collibra DQ customers upgrading Agents installed on Cloudera CDP Hadoop Edge Nodes.

1. Go to **/<agenthome>/owl/config/**.
2. Remove the `log4j-cluster.properties` file.

FAQ

Q: (When) do I need to remove **log4j-cluster.properties** before or after swapping the main Collibra DQ jars?

- A: Remove the file before restarting **owl-agent**. Otherwise, stop owl-agent again, remove the file, then restart **owl-agent**.

Q: (What) if I don't follow these additional steps?

- A:If you use agents on Hadoop edge nodes, you will receive errors when running DQ Jobs as a result of engaging a method that no longer exists.

Q: What should I do if I am not a vendor-supported Cloudera CDP version?

- A: Our testing and guidance mainly applies to vendor-supported (non-EOL) Cloudera CDP versions. Other Hadoop variants may handle logging differently and may require the legacy properties file. In short, feel free to first upgrade without this

step, then remove the **log4j-cluster.properties** file if DQ Jobs are running into issues.

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. Confirm with Collibra Support which drivers need to be updated.
2. From the previously extracted tarball provided to you by Collibra, locate the **driver-s.tar.gz** file and extract the contents into a new directory called *drivers*.
3. Replace the drivers: Replace **OWL_BASE/owl/drivers/<old-driver>** with the new drivers extracted from the tarball **OWL_BASE/owl/drivers/<new-driver>**.

   > Example If you replace an old Databricks driver with a new one, the file path might look like **OWL_BASE/owl/drivers/databricks**.

Standalone Sizing

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

Component	RAM	Cores
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

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 Performance Tuning for more information.

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

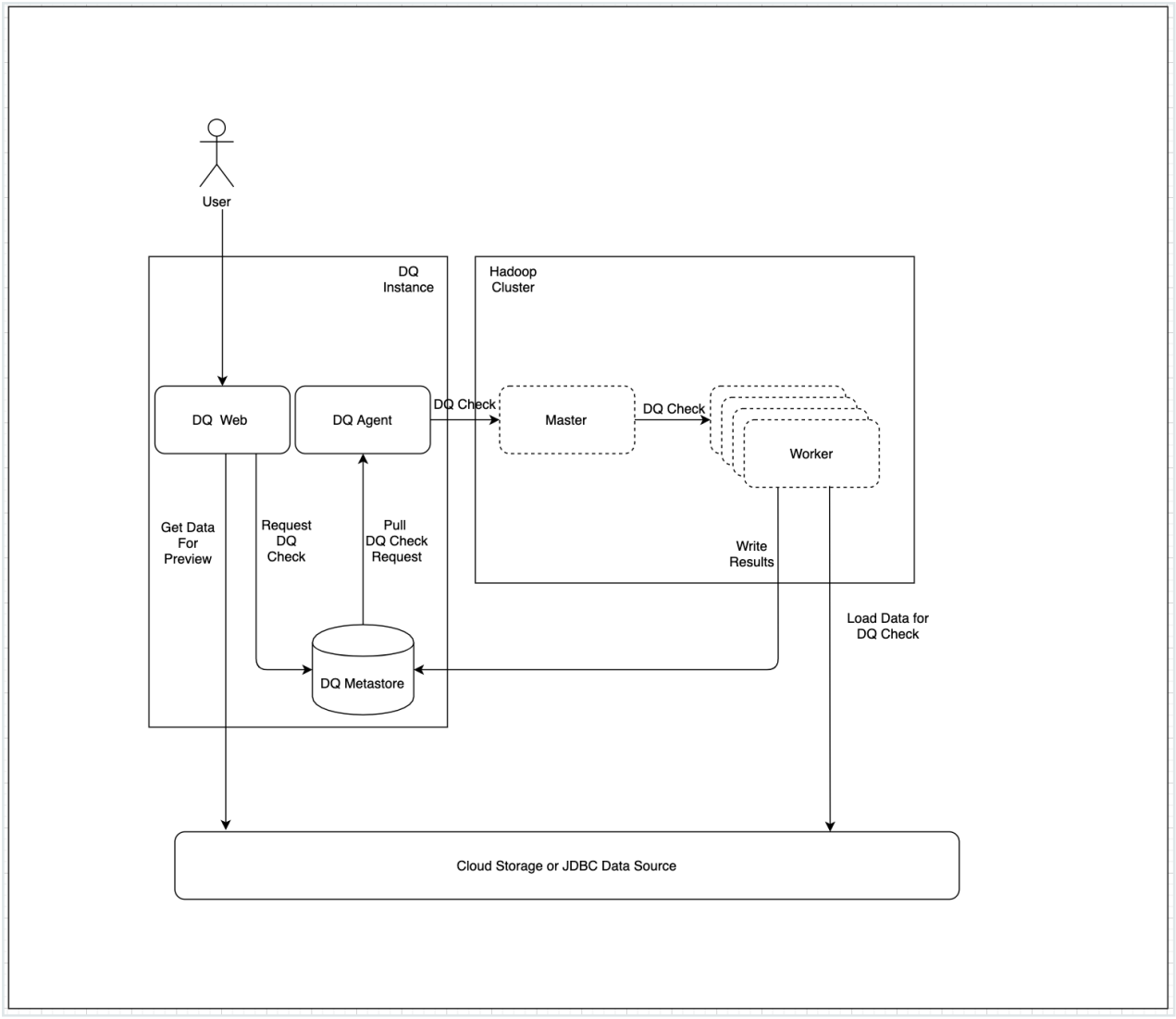
Bytes per Cell	Rows	Columns	Gigabytes	Gigabytes for Spark (3x)
16	10,000,000.00	100	16	48
16	1,000,000,000.00	100	1600	4800
16	100,000,000.00	100	160	480
16	1,000,000.00	200	3.2	9.6
16	10,000,000.00	200	32	96
16	100,000,000.00	200	320	960
16	1,000,000,000.00	200	3200	9600

Cluster

If your program requires more horsepower or (spark) workers than the example tiers above which is fairly common in Fortune 500 companies then you should consider the horizontal and ephemeral scale of a cluster. Common examples are Amazon EMR, Cloudera CDP, etc. Collibra DQ is built to scale up horizontally and can scale to hundreds of nodes.

Hadoop

For large scale processing and concurrency, a single vertically scaled Spark server is not enough. To address large scale processing, DQ has the ability to push compute to an external Hadoop cluster. This page describes the process by which the DQ Agent can be configured to push DQ jobs to Hadoop.



Hadoop Install

> Note 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.

Hadoop Config Setup

Hadoop configuration can be incredibly complex. There can be hundreds of "knobs" across dozens of different components. However, DQ's goal is to simply leverage Hadoop to allocate compute resources in order to execute DQ checks (Spark jobs). This means that the only client side configurations required are:

- Security protocol definition
- Yarn Resource Manager endpoints
- Storage service (HDFS or Cloud storage).

Once the Hadoop client configuration is defined, it is only a matter of pointing 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.

> Note 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.

Create the Config Folder

```
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
```

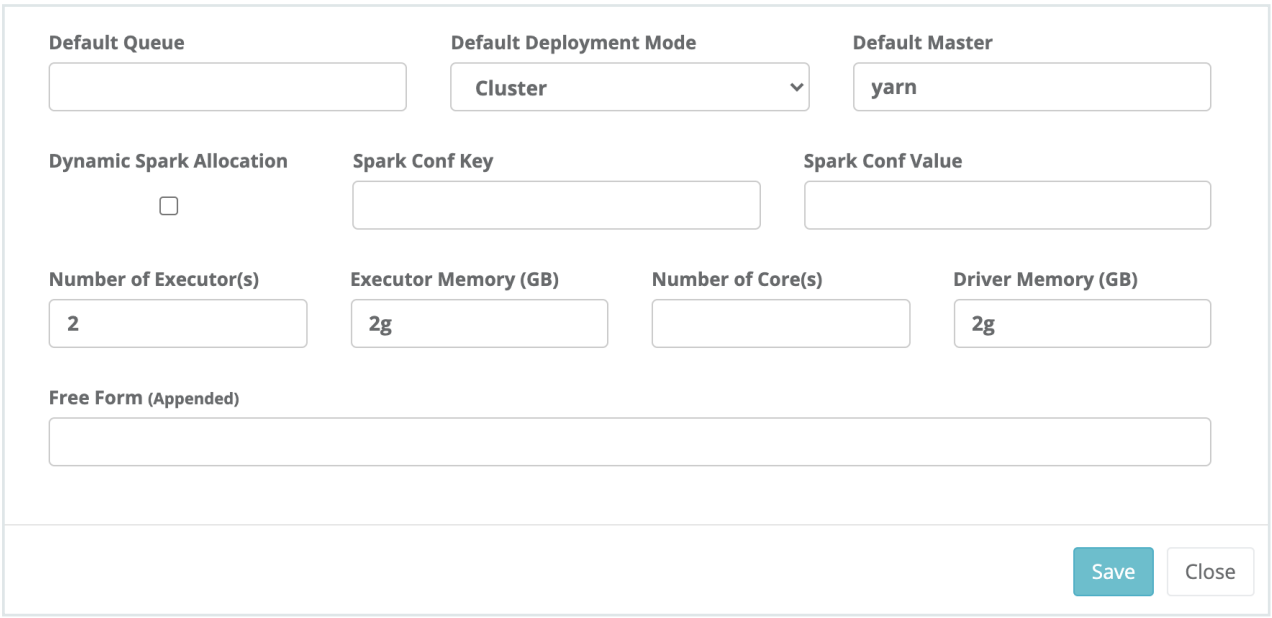
Minimum Config with Kerberos and TLS Disabled

This configuration would typical only be applicable in Cloud Hadoop scenarios (EMR/Dataproc/HDI). Cloud Hadoop clusters are ephemeral and do not store any data as the data is stored in and is secured by Cloud Storage.

```
export RESOURCE_MANAGER=<yarn-resoruce-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
```

> Note 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. To achieve this, create a new storage bucket and ensure that both the Hadoop cluster and the instance running DQ Agent have access to it. This is usually accomplished using a Role that is attached to the infrastructure. For example, AWS Instance Role with bucket access policies. Then, set "fs.defaultFS" in core-site.xml to the bucket path instead of HDFS.

Once the Hadoop client configuration is created, go to Agent Management console from the Admin Console and configure the agent to use Yarn (Hadoop resource scheduler) as the Default Master and set the Default Deployment Mode to "Cluster".

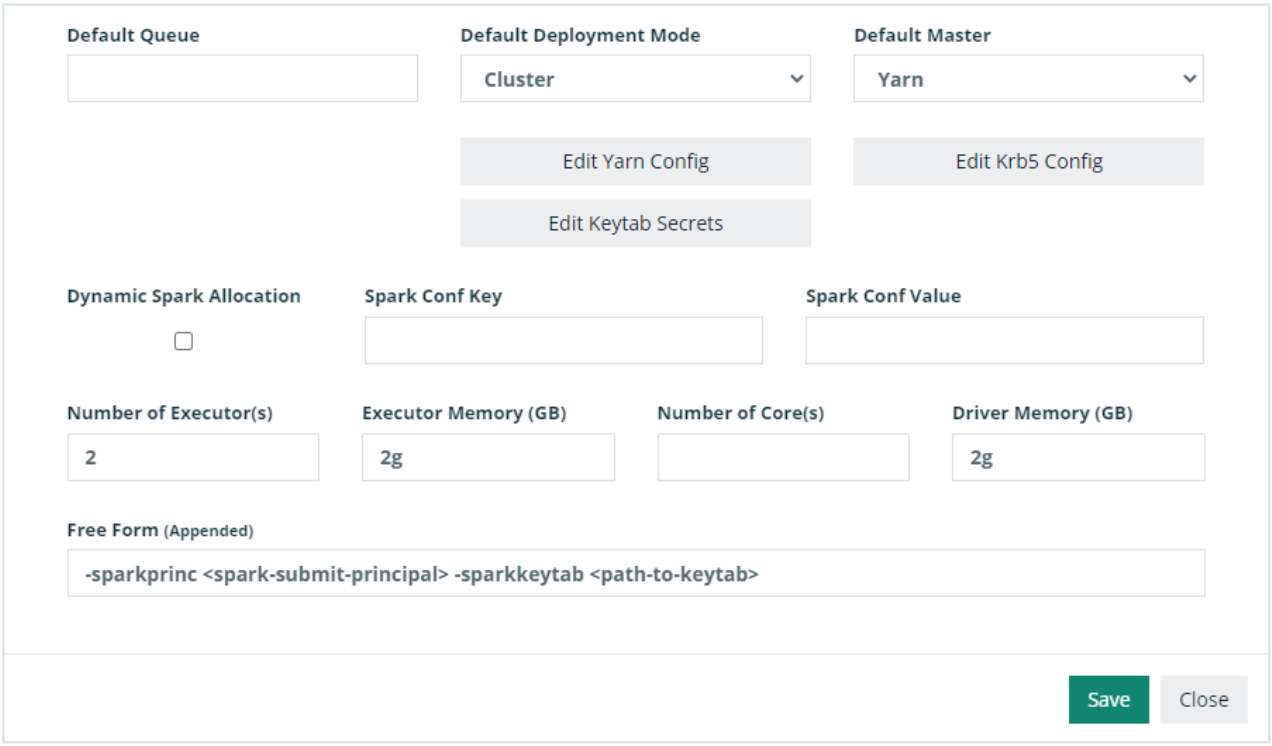


Kerberos Secured with Resource Manager TLS enabled

Typically, Hadoop clusters that are deployed on-premises are multi-tenant and not ephemeral. This means they must be secured using Kerberos. Additionally, all endpoints with HTTP endpoints will have TLS enabled. 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
```

When the target Hadoop cluster is secured by Kerberos, DQ checks 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. Access the DQ Agent configuration page from the Admin Console and configure the "Freeform Append" setting with the -sparkprinc <spark-submit-principal> -sparkkeytab <path-to-keytab>.



EMR / Dataproc / HDI

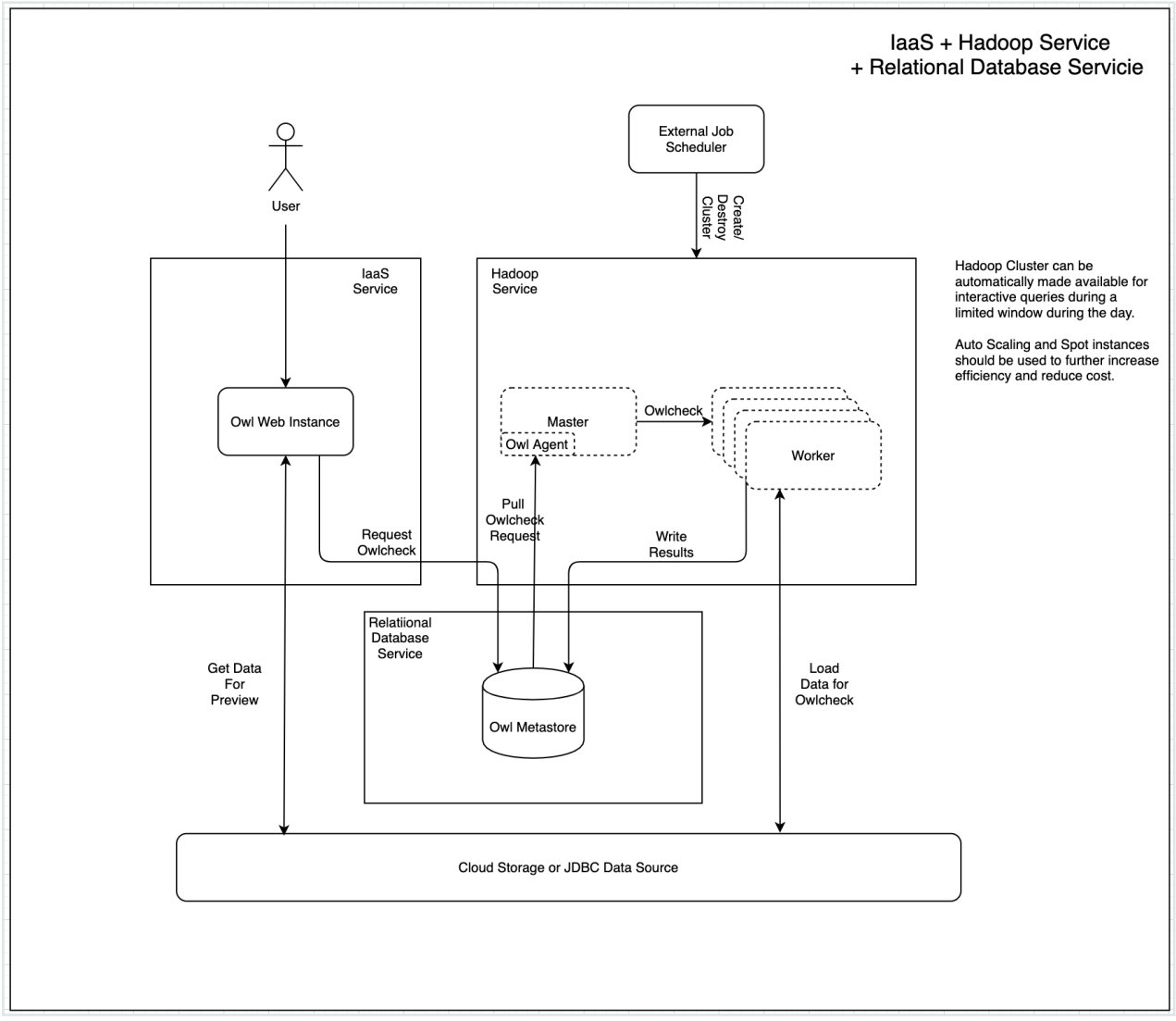
Running Apache Spark on Kubernetes differs from running this on virtual machine-based Hadoop clusters, which is the current mechanism provided by the existing CloudProc Dataproc service or competitive offerings like Amazon Web Services (AWS) Elastic MapReduce (EMR) and Microsoft's Azure HDInsight (HDI).

Each cloud provider has unique steps and configuration options. For more details on enabling agents for this deployment option refer to the Hadoop Integration section.

A detailed guide for EMR is provided below.

Collibra Data Quality & Observability on EMR Architecture

Collibra DQ is able to use EMR as the compute space for data quality jobs (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**:

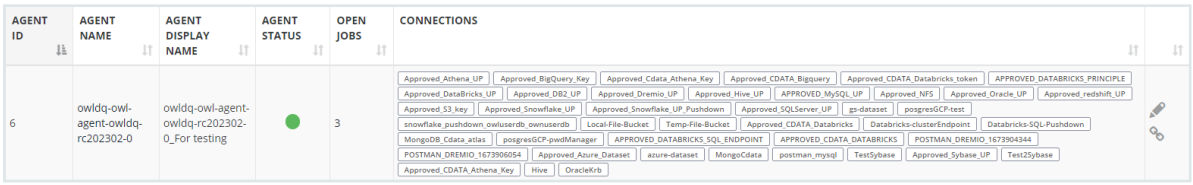
```
aws s3 cp s3://${OWL_PACKAGE_LOCATION}/dq-$OWL_VERSION-package-full.tar.gz ./
tar -xvf dq-$OWL_VERSION-package-full.tar.gz
```

```
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\"} \
            ]"
```

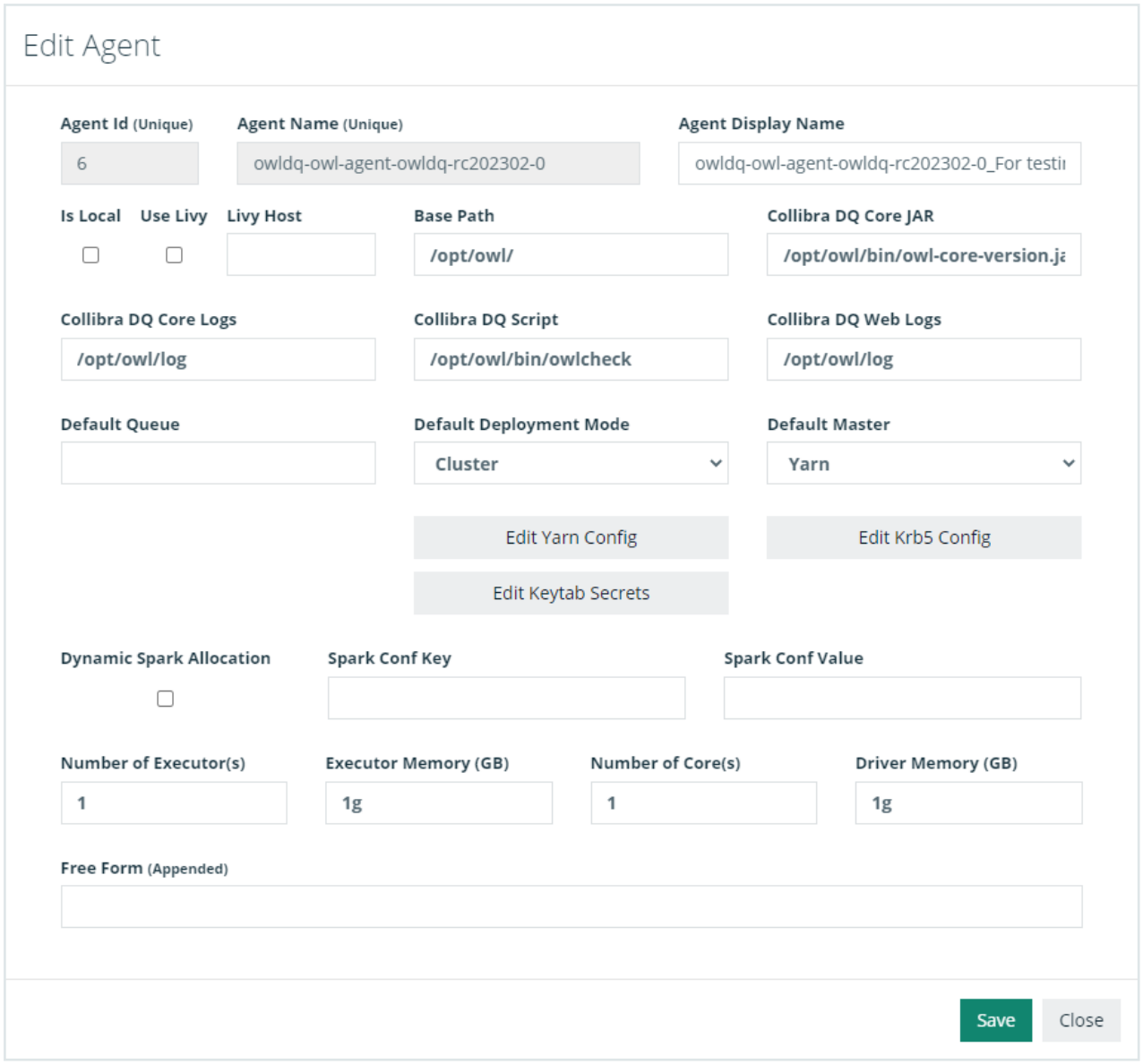
Configure Agent

Once the EMR cluster and the DQ Agent is deployed, it needs to be configured in DQ Web.

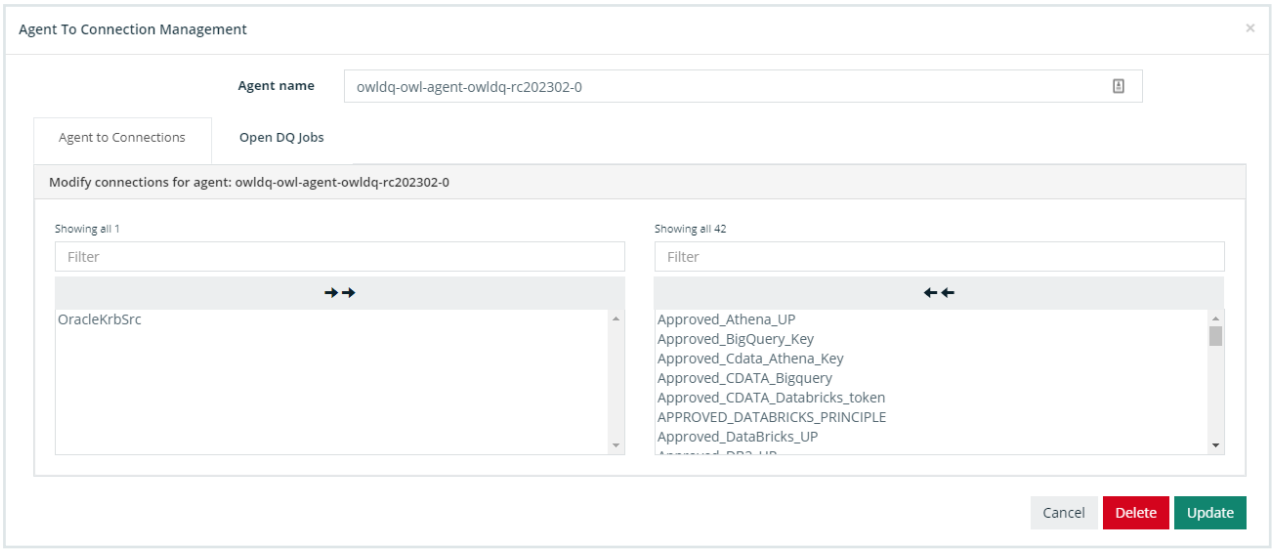
1. Sign in to your web instance of Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
3. Click the **Remote Agent** tile.
   » The **Agent Management** page opens.
   » In the following example, the newly created agent has a green indicator to 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. From the **Default Deploy Mode** drop-down list, select **Cluster**.
6. From the **Default Master** drop-down list, select **Yarn**.



7. Click **Save**.
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. Refer to the Explorer (no-code) section for more details.

Cloud native

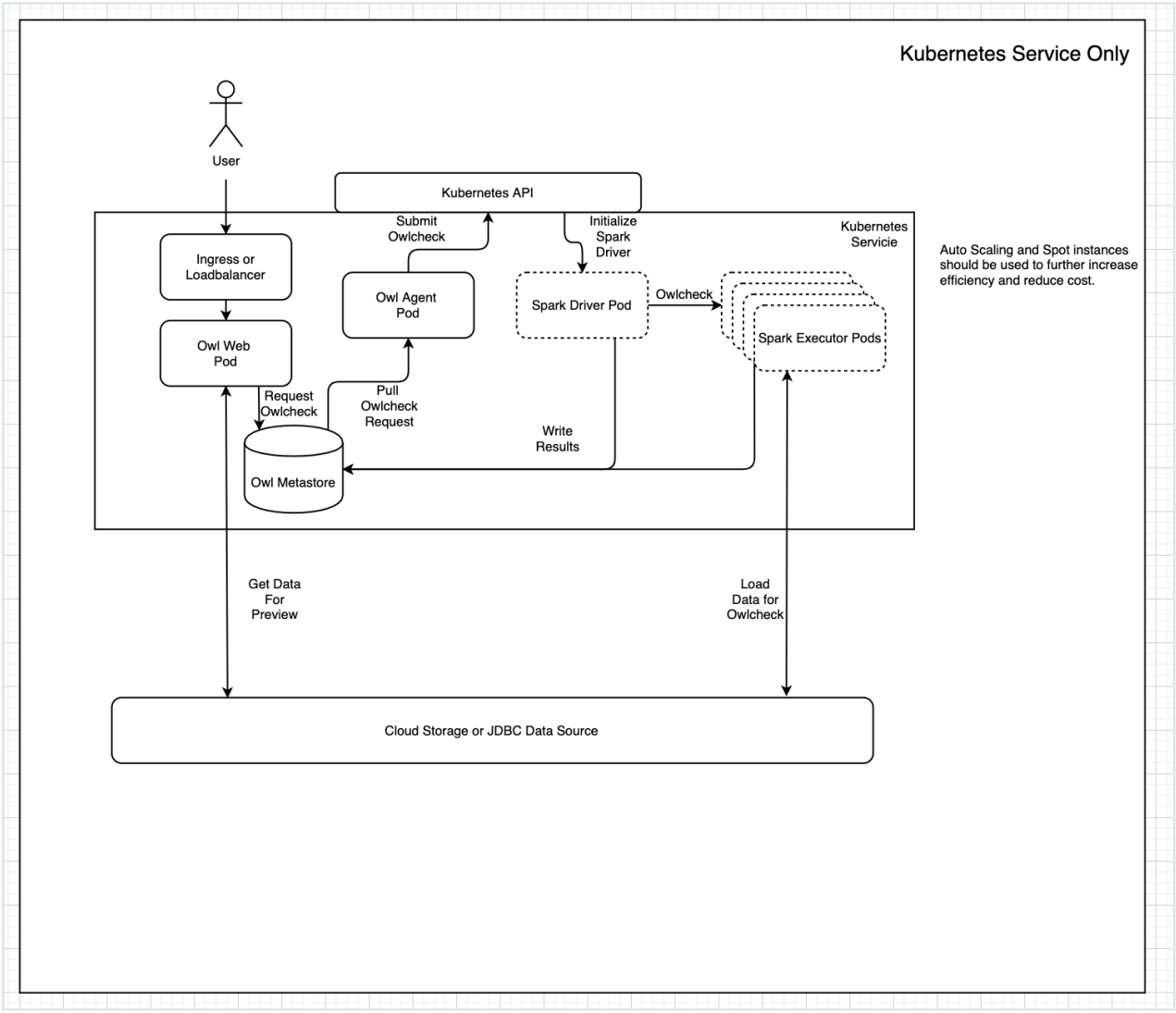
Introduction to cloud native architecture

According to the Cloud Native Computing Foundation ("CNCF") Charter:

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

Collibra Data Quality & Observability wholeheartedly embraces these principles in its design and deployment. The diagram below depicts Collibra Data Quality & Observability'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.

- **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** - 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** - This is 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.
- **History Server** - Collibra DQ relies on Apache Spark to actually scan data and perform the bulk of data quality activities. To facilitate troubleshooting and performance tuning of DQ jobs, Collibra DQ uses an instance of Spark History Server to enable easy access to Spark logs.
- **Spark** - Apache Spark is 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.

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.

> Note For complete details on how to install Collibra DQ on Kubernetes with Docker containers, see Cloud native install.

> Note As of the Collibra DQ 2024.02 version, Collibra DQ supports Kubernetes versions 1.21- 1.29. Collibra DQ versions before 2024.02 support Kubernetes versions 1.21 - 1.27.

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.

For example, the command below deploys Collibra DQ with all of the components depicted in the Introduction to cloud native architecture into Google Kubernetes Engine with Google Cloud Storage (GCS) as the storage location for Spark logs. The only prerequisite is that the image pull secret, representing credentials to access the container registry, and secret containing the credentials for a service account with access to GCS are already deployed to the namespace.

```
helm upgrade --install --namespace <namespace> \
--set global.version.owl=<cdq-version> \
--set global.version.spark=<cdq-spark-version> \
--set global.configMap.data.license_key=<cdq-license-key> \
--set global.spark_history.enabled=true \
--set global.spark_history.logDirectory=gs://logs/spark-history/
\
--set global.cloudStorage.gcs.enableGCS=true \
<deployment-name> \
/path/to/chart/owldq
```

> Note The full universe of possible customizations is quite extensive and provides a great deal of flexibility in order to be applicable in a wide variety of platforms. However, when deploying on a known platform (EKS, GKE, AKS), the number of required inputs is quite limited. In common cases, you 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.

Cloud native requirements

Minimum requirements

You need a machine with the following files and packages to run the installation. You can run these from a laptop or separate VM and they do not need to be issued on the Kubernetes cluster itself.

> Note For complete details on how to install Collibra DQ on Kubernetes with Docker containers, see Cloud native install.

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.

Files

- The helm chart.
- JKS files with secrets created in kubectl:
  - *dq-ssl-secret*
  - *dq-pull-secret**
- A *spark-gcs-secret* you create from your service account file or token.

> Note * Available upon request from Collibra.

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	-

> 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.

Obtaining 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

Collibra Data Quality & Observability 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 to the default Edit role. Edit is a role that is generally available in a Kubernetes cluster. If the Edit role is not available, you can manually create it.

Accessing the 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.

Preparing secrets

Once access to the target platform is confirmed, you can begin the preparation of the namespace. Typically the namespace that Collibra DQ is going to be deployed into is pre-allocated by the platform team.

```
kubectl create namespace <namespace>
```

> Note There is a lot more that can go into namespace creation such as resource quota allocation, but that is generally a task for the platform team.

Create an SSL keystore secret

> Note For complete details on how to install Collibra DQ on Kubernetes with Docker containers, see Cloud native install.

Create a container pull secret

> Note For complete details on how to install Collibra DQ on Kubernetes with Docker containers, see Cloud native install.

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.

> Note For complete details on how to install Collibra DQ on Kubernetes with Docker containers, see Cloud native install.

Cloud native install

Install Collibra DQ on Kubernetes with Docker Containers

Collibra DQ provides the following Core Docker containers:

- **dq-agent**: Launches the Apache Spark Jobs
- **dq-web**: The Collibra DQ web application itself
- **Apache Spark**: The runtime analytics engine
- **Postgres** (persistent volume needed): The Collibra DQ metastore
- **Apache Livy**:
  - Session Manager: How Collibra DQ can browse the HDFS, S3, GCS, or Azure Data Lake (ADL)
  - Interacts with Object Stores, similar to JDBC sources Explorer (estimate Jobs, get days with data, Filtergrams, etc.)

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.

Steps

To install Collibra DQ on Kubernetes with Docker containers, follow these steps.

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 code snippet:

```
kubectl create namespace <owldq>
```

Pull images from the Collibra registry

Collibra DQ containers are located in the Docker Hub registry. Collibra provides a repo-key to access Collibra images in a .json file, which can be stored locally and used to login.

1. Download the Docker .json repo-key.
2. Run the following command:

```
docker login -u _json_key -p "$(cat repo-key.json)"
https://gcr.io
```

> Note Image names with their versions are provided by Collibra.

3. 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 Docker 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

```
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.

1. To create a pull secret, use the following code snippet:

```
kubectl create secret docker-registry dq-pull-secret
--docker-server=https://gcr.io
--docker-username=_json_key
--docker-email=<email of customer>
--docker-password="$(cat repo-key.json)"
--namespace <dq>
```

> 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.

Helm chart

For more detailed information about the Helm Chart, see Cloud native.

Unzip and store the Helm charts provided by Collibra on a Linux compatible deployment location.

> Note Once you have your Collibra DQ license, you will receive an email from Collibra that includes the Helm Charts as zip files.

There should be two folders and two files:

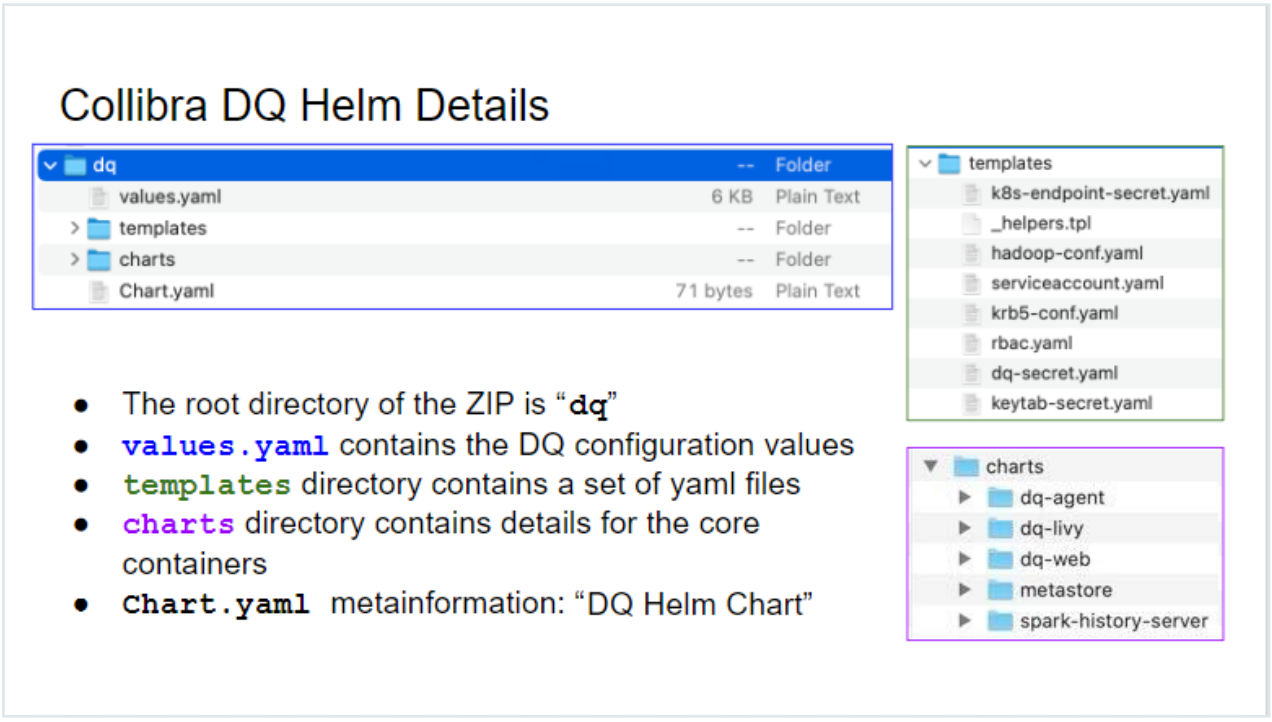
- drwxrwxr-x -- templates
- drwxrwxr-x -- charts

- -rw-rw-r-- Chart.yaml
- -rw-rw-r-- values.yaml

There are two ways of passing parameters. While deploying, parameter values can be passed:

2. Using the values.yaml file or,
3. Using the Helm set commands.

> Note The set commands take precedence over the values.yaml file.



Deploy Collibra Data Quality & Observability

Once you have created a namespace and added all of the required secrets, you can begin the deployment of Collibra DQ.

Minimal install

Install Web, Agent, and 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 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 in the Sign in to the Kubernetes cluster step.
<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.
<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 Create a pull secret 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.

Externally Accessible Service

Perform the Minimal install and add 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
```

Externally Accessible with SSL Enabled

Perform the install with external 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
```

Externally Accessible and History Server for GCS Log Storage

Perform the install with external service and Spark History Server enabled. In the following 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 Cloud storage credentials.

```
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
```

Externally Accessible and History Server for S3 Log Storage

Perform the install with external 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, makes 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 Cloud storage credentials.

```
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
```

Externally Accessible with External Metastore

Perform the install with external 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.

Externally Accessible and History Server for Windows Azure Storage Blob (WASB) Log Storage

Perform the install with external service and Spark History Server enabled.

Note 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 (e.g., ADLS, S3, or GCS).

```
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
```

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.

Troubleshooting and Helpful Commands

This guide is to provide the most common commands to run when troubleshooting a DQ environment that is deployed on Kubernetes. For a basic overview of Kubernetes and other relevant knowledge.

```
### 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 & 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 '*' '*'
```

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:

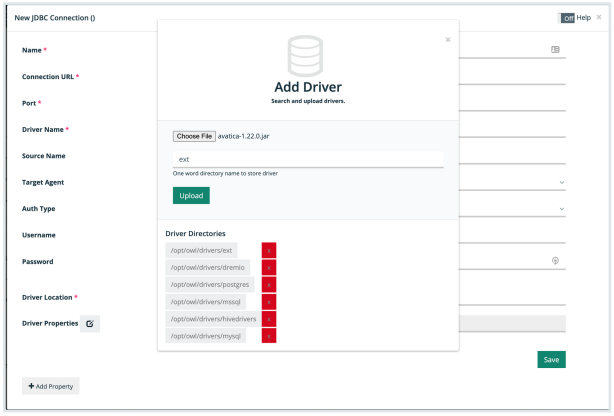
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-pv
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  nfs:
    server: nfs-server.nfs.svc.cluster.local
    path: "/"

---
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.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,s-
park.kubernetes.executor.volumes.persistentVolumeClaim.nfs-
pv.options.claimName=nfs-
pvc,s-
park.kubernetes.executor.volumes.persistentVolumeClaim.nfs-
pv.-
option-
s.s-
izeLim-
it=10Gi,-
spark.kuber-
netes.executor.volumes.persistentVolumeClaim.nfs-
pv.-
moun-
t.path-
=/o-
pt/owl/driver-
s/ex-
t,s-
park.kubernetes.executor.volumes.persistentVolumeClaim.nfs-
pv.-
moun-
t.readOnly-
=false,-
,spark.kubernetes.driver.volumes.persistentVolumeClaim.nfs-
pv.options.claimName=nfs-
pvc,s-
park.kubernetes.driver.volumes.persistentVolumeClaim.nfs-pv.-
option-
s.s-
izeLim-
it=10Gi,-
,spark.kubernetes.driver.volumes.persistentVolumeClaim.nfs-
pv.-
moun-
t.path-
=/o-
pt/owl/driver-
```

```
s/ex-
t,spark.kubernetes.driver.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.

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).

> 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 **Remote Agent**.
   » The **Agent Management** page opens.
6. Click the pencil icon.
   » The **Edit Agent** modal opens.
7. In the **Free Form (Appended)** 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-template.yml
```

> Tip If you add any additional configuration properties based on your specific requirements, separate them with commas.

8. 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.

EKS / GKE / AKS

For organizations that are familiar with containers and managed Kubernetes services, Collibra Data Quality & Observability offers a cloud native deployment option. Please refer to the Cloud Native section for more detail.

Cloud native

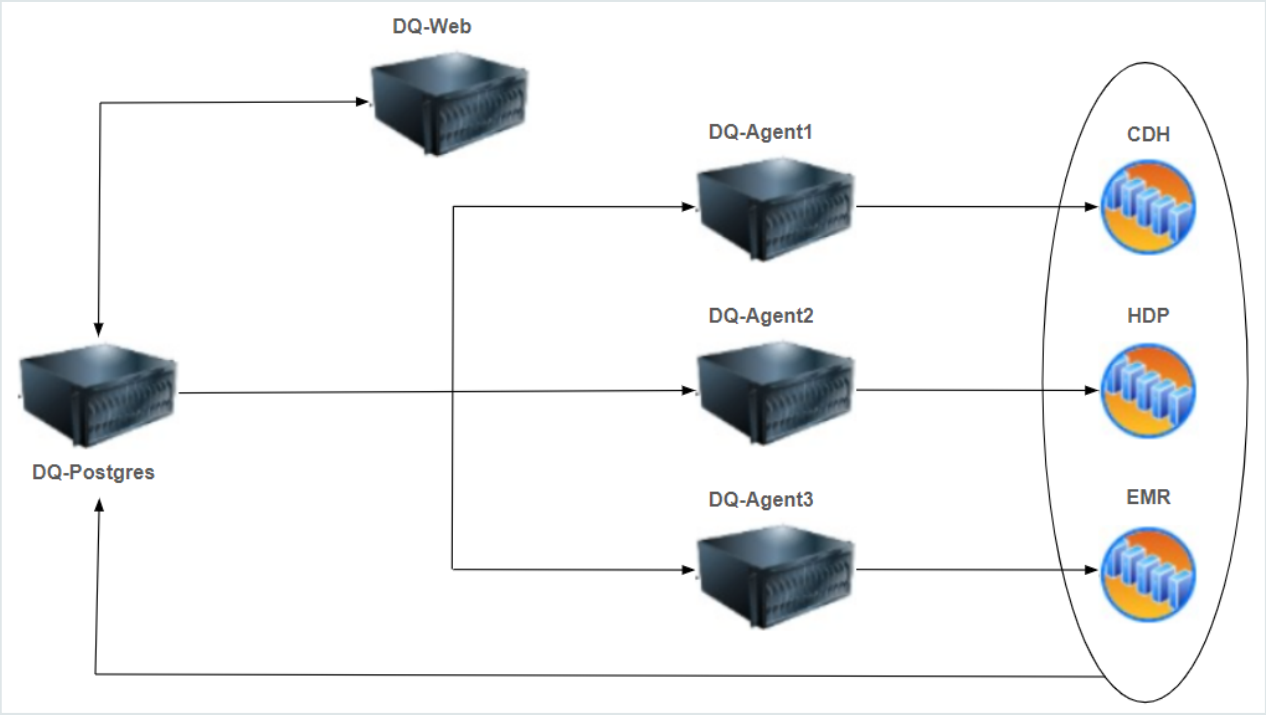
Cloud native requirements

Cloud native install

> Note The containers are part of a private repository which requires access. Please contact Collibra directly for more information.

Agent

Diagram



The diagram above 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 DQ 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.

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 DQ Metastore. The results then display in the DQ Web App, are exposed as REST API, and become available for direct SQL query against the DQ Metastore.

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

Steps

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 Storage 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:postgresql://{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

Steps

1. On the Collibra DQ home page, hover your cursor over ⚙ **Settings** and select **Admin Console**.
   » The Admin Console opens.
2. Click Remote Agent.
   » The **Agent Management** page opens.
3. In the last column of the Agents table, to the right, click the pencil icon 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.

Field	Description
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/`
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`

Field	Description
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.
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.

Field	Description
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.
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.

Field	Description
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`.
Default Queue	The default resource queue to submit jobs.

Field	Description
Default Deployment Mode	The Spark deployment mode for Yarn is **Cluster**.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Default Master</td><td>Set to <code>Yarn</code>.

Click Edit Yarn Config to ensure you have the necessary Hadoop xml files. Edit the file templates as necessary:

<table>
<tr><th>XML File</th><th>Description</th></tr>
<tr><td>core-site.xml</td><td>Contains information about where authentication protocol, HDFS_RPC_PROTECTION, and the NAME_NODE run in the Hadoop cluster.

Show example template

<pre>
<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>
</pre>
</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>XML File</th><th>Description</th></tr>
<tr><td>hdfs-site.xml</td><td>Contains the configuration settings for authentication protocol, the NAME_NODE, and DATA_NODE.

Show example template

<pre>
<configuration>
  <property>
    <name>hadoop.security.authentication</name>
    <value>HDFS/_HOST@$KERBEROS_DOMAIN</value>
  </property>
</configuration>
</pre></td></tr>
<tr><td>yarn-site.xml</td><td>Contains the Yarn resource manager settings.

Show example template

<pre>
<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>
</pre></td></tr>
</table>
</td></tr>
</table>

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**.

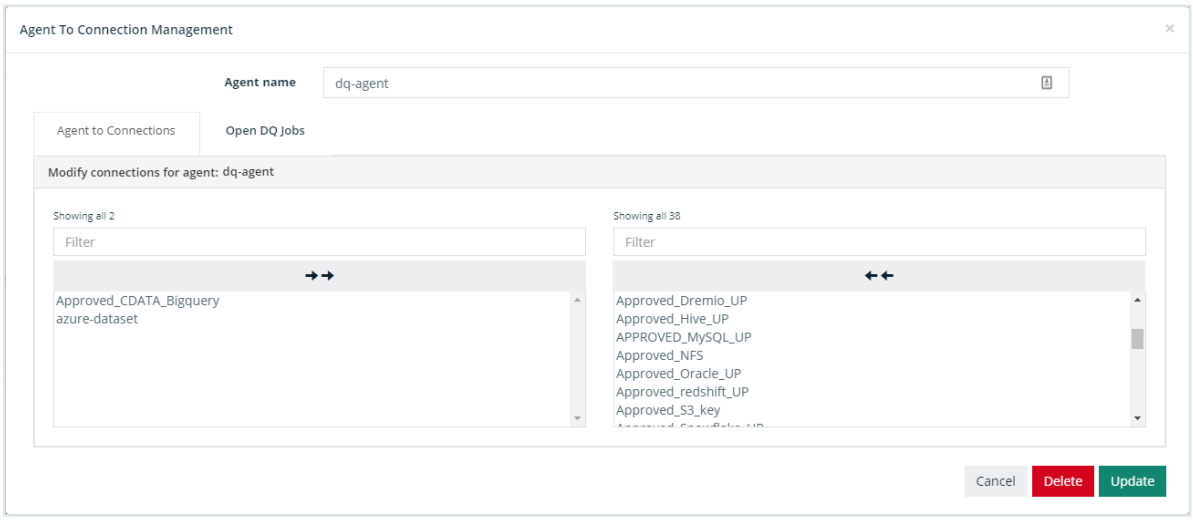
Linking data sources to the DQ Agent from the Admin Console

When you add new Data Sources, the DQ Agent requires permission to run DQ Jobs with them.

Steps

1. On the Collibra DQ home page, hover your cursor over ⚙ **Settings** and select **Admin Console**.
   » The Admin Console opens.
2. Click Remote Agent.
   » The **Agent Management** 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.

Adding a connection to a DQ Agent

Upgrade Spark

This section details how to upgrade on-premises Apache Spark versions.

> 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.

Steps

1. Run the following commands to stop Spark and Collibra DQ services.
   a. Stop Spark Master:

```
cd $OWL_HOME/spark/sbin
./stop-master.sh
```

   b. Stop Spark Worker:

```
cd $OWL_HOME/spark/sbin
./stop-worker.sh
```

   c. Stop DQ Web:

```
cd $OWL_HOME/bin
 ./owlmanage.sh stop=owlweb
```

d. Stop DQ Agent:

```
cd $OWL_HOME/bin
./owlmanage.sh stop=owlagent
```

e. Verify that all processes are stopped:

```
ps -ef | grep owl

## No DQ processes should return as running. ##
```

2. Create backups of the following directories and files:
   - spark
   - spark-extras
   - dq-core-*-dependencies.jar
   - dq-webapp*.jar
   - dq-agent*.jar
3. Remove the directories and files listed in the previous step.
4. Create a backup of the following files from the $OWL_HOME/bin folder if you intend to replace them with the latest, otherwise, skip steps 4 and 5:
   - owlcheck
   - owlmanage.sh
5. Remove the files listed in the previous step.
6. Download the Collibra DQ installer packaged with the necessary Spark version from the Collibra Data Quality & Observability Downloads page in the Product Resource Center.
7. Run the following command to go to the "packages/install-packages" directory.

```
cd <OWL_INSTALLER_DIRECTORY>/packages/install-packages
```

8. Run the following command to extract the installer:

```
tar -xvf "new_installer.tar.gz"
```

> Example To extract the recommended Spark 3.4.1, the command would be `tar -xvf spark-3.4.1-bin-hadoop3.2.tgz`.

9. Run the following to extract the "spark-extras.tar.gz":

```
tar -xvf spark-extras.tar.gz
```

10. Run the following command to copy the "spark-extras" directory from the installer extract directory to "$OWL_HOME/spark-extras" directory:

```
cp -r <INSTALLER_DIRECTORY>/packages/install-packages/spark-
extras
$OWL_HOME/spark-extras
```

11. Run the following command to copy the "spark-3.4.1-bin-hadoop3" directory from the installer extract directory to the "$OWL_HOME/spark" directory:

```
cp -r <INSTALLER_DIRECTORY>/packages/install-packages/spark-
3.4.1-bin-hadoop3
$OWL_HOME/spark
```

12. Run the following command to copy the "spark-extras" directory into the "spark/jars folder:

```
cp -r <INSTALLER_DIRECTORY>/packages/install-packages/spark-
extras
$OWL_HOME/spark/jars
```

13. Run the following commands to copy the Collibra DQ jar files.
    a. Copy the dq-core jar file:

```
cp -r <INSTALLER_DIRECTORY>/dq-core-*-SPARK341-jar-
with-dependencies.jar
$OWL_HOME/bin
```

    b. Copy the dq-webapp jar file:

```
cp -r <INSTALLER_DIRECTORY>/dq-webapp-*-SPARK341.jar
$OWL_HOME/bin
```

c. Copy the dq-agent jar file:

```
cp -r <INSTALLER_DIRECTORY>/dq-agent-*-SPARK341.jar
$OWL_HOME/bin
```

d. Copy the owlcheck:

```
cp -r <INSTALLER_DIRECTORY>/owlcheck
$OWL_HOME/bin
```

e. Copy the owlmanage.sh file:

```
cp -r <INSTALLER_DIRECTORY>/owlmanage.sh
$OWL_HOME/bin
```

14. 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. Start DQ Agent:

```
cd $OWL_HOME/bin
./owlmanage.sh start=owlagent
```

15. Visit the Spark page on port 8080 to confirm that the Spark version is the one you installed and the worker is alive.
16. 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 Collibra Data Quality & Observability

This section describes how to upgrade from an older Collibra Data Quality & Observability 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.

User permissions

- Installation is completed via tarball.
- You must be able to create directories, launch scripts, and start Java processes.
- Use the same service account that you used during the initial installation and is currently running Collibra DQ processes.
- ULIMIT settings of 4096 or higher
    - All Collibra DQ services typically consume about 428 threads.
    - During each DQ Job, about 400 additional threads are consumed.
    - Thus 4096 threads can allow for about 9 concurrent DQ Jobs. If more are needed, plan accordingly.

System requirements

Supported operating systems

- Red Hat Enterprise Linux 7.x
- Red Hat Enterprise Linux 8.x
- CentOS 7.x
- CentOS 8.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 Performance Tuning for more information.

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

Bytes per Cell	Rows	Columns	Gigabytes	Gigabytes for Spark (3x)
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

- 5432 - PostgreSQL
- 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 version

As of the Collibra DQ 2024.02 version, Collibra DQ supports Kubernetes versions 1.21 - 1.29. Collibra DQ versions before 2024.02 support Kubernetes versions 1.21 - 1.27.

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 [[[Undefined variable DataQuality.DQ Metastore]]]	1 core	2 GB	10 GB PVC
Spark*	2 cores	2 GB	-

> 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

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 [[[Undefined variable DataQuality.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 [[[Undefined variable DataQuality.DQ Metastore]]]
- Upgrade Collibra Data Quality & Observability

Create and restore a backup

This section details how to create a backup of the [[[Undefined variable DataQuality.DQ Metastore]]] prior to an upgrade.

Internal vs. external [[[Undefined variable DataQuality.DQ Metastore]]]

When you create a backup of the [[[Undefined variable DataQuality.DQ Metastore]]], it's important to consider whether your deployment of Collibra DQ uses the internal [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ Metastore]]], such as GCP Cloud SQL or AWS RDS, some Standalone deployments use an internal [[[Undefined variable DataQuality.DQ Metastore]]]. Regardless of your [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ Metastore]]], copy the contents within the `owl/postgres` folder created during the installation.

Restore a backup

To restore a snapshot of an internal [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ Metastore]]]s are the most commonly used [[[Undefined variable DataQuality.DQ Metastore]]]s for both Cloud Native and Standalone deployments of Collibra DQ. Because the procedure for backing up external [[[Undefined variable DataQuality.DQ Metastore]]]s 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 [[[Undefined variable DataQuality.DQ Metastore]]]s.

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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ Metastore]]] to provide additional fault-tolerance.

Restore a backup

As external [[[Undefined variable DataQuality.DQ Metastore]]]s 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 [[[Undefined variable DataQuality.DQ 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.

Download the Collibra DQ Upgrade Package

> Note Starting with the December 2021 release, 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.

Download the tarball using the signed link to the full package tarball provided by Collibra. Replace `<signed-link-to-full-package>` with the link provided.

```
### Go to the OWL_BASE (home directory of the user is most
common)
### This example we will use /home/owldq installing as the
user owldq

cd /home/owldq

### Download & untar
curl -o dq-full-package.tar.gz "<signed-link-to-full-
package>"
tar -xvf dq-full-package.tar.gz

### Clean-up unnecessary tarball (optional)
rm dq-full-package.tar.gz
```

Upgrade steps

> Important Verify the Collibra Data Quality & Observability 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. Select an option from the

abs below depending on the version from which you are upgrading and the
ersion to which you plan to upgrade.

Upgrading Collibra DQ from version 2023.09 or newer to a version newer than 2023.09

This section provides steps to upgrade standalone deployments of Collibra DQ from version 2023.09 or newer to a version newer than 2023.09.

1. Copy the contents of the provided package, for example, dq-<new-version>-<SPARK301>-package-full.tar.gz to the system being upgraded (extract contents).

   > Tip Untar the contents into a uniquely named folder, for example *2023-11-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>-<spark301>.jar /tmp
   mv dq-agent-<oldversion>-<spark301>.jar /tmp
   mv dq-core-<oldversion>-<spark301>.jar /tmp
   ```

5. 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
```

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

Upgrading Collibra DQ from a version older than 2023.09 to version 2023.09 or newer

This section provides steps to upgrade standalone deployments of Collibra DQ from a version older than 2023.09 to version 2023.09 or newer.

1. Copy the contents of the provided package, for example, dq-<new-version>-<SPARK301>-package-full.tar.gz to the system being upgraded (extract contents).

> Tip Untar the contents into a uniquely named folder, for example *2023-09-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 owl-webapp-<oldversion>-<spark301>.jar /tmp
mv owl-agent-<oldversion>-<spark301>.jar /tmp
mv owl-core-<oldversion>-<spark301>.jar /tmp
```

5. 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
```

6. 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`.

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
```

Upgrading from a Collibra DQ version older than 2023.09 to a different version older than 2023.09

This section provides steps to upgrade standalone deployments of Collibra DQ from a version older than 2023.09 to a different version older than 2023.09. To upgrade from a version before 2023.09 to one from 2023.09 or later, follow the steps outlined in the next section instead.

1. Copy the contents of the provided package, for example, owl-<new-version>-<SPARK301>-package-full.tar.gz to the system being upgraded (extract contents).

> Tip Untar the contents into a uniquely named folder, for example, *2023-05-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 owl-webapp-<oldversion>-<spark301>.jar /tmp
mv owl-agent-<oldversion>-<spark301>.jar /tmp
mv owl-core-<oldversion>-<spark301>.jar /tmp
```

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

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

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

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:2023.11-
ABDGCSHILM-123
docker pull gcr.io/owl-hadoop-cdh/dq-agent:2023.11-
ABDGCSHILM-123
docker pull gcr.io/owl-hadoop-cdh/dq-livy:3.4.1-2023.11-
ABDGCSHILM-123
docker pull gcr.io/owl-hadoop-cdh/dq-spark:3.4.1-2023.11-
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.

```
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.
${password}	The default admin user password for the admin account. The password must adhere to the following password policy: • A minimum of 8 characters. • A maximum of 72 characters. • At least one upper-case character. • At least one numeric character. • At least one supported special character (`!@#%$^&*?_~`). • 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.

Value	Description
<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.

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
```

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 '*' '*'
```

Chapter 6

Collibra DQ Connections

Supported Connections

This section shows the supported data source connections in Collibra Data Quality & Observability. 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.

Data Source	Packaged	Certified	Supports Push-down
Amazon Redshift	✓ Yes	✓ Yes	✓ Yes Note Pushdown for Amazon Redshift is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.

Data Source	Packaged	Certified	Supports Push-down
Apache Impala	✓ Yes	✓ Yes	✕ No
Amazon Athena	✓ Yes	✓ Yes	✓ Yes Note Pushdown for Amazon Athena is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.
CockroachDB	✓ Yes	✓ Yes	✕ No
Databricks	✓ Yes	✓ Yes	✓ Yes
Dremio	✕ No	✕ No	✕ No
Google BigQuery	✓ Yes	✓ Yes	✓ Yes
Hive	✕ No	✓ Yes	✕ No
Db2	✓ Yes	✓ Yes	✕ No
MongoDB	✓ Yes	✓ Yes	✕ No
MySQL	✓ Yes	✓ Yes	✕ No

Data Source	Packaged	Certified	Supports Push-down
Oracle	✓ Yes	✓ Yes	✗ No
PostgreSQL	✓ Yes	✓ Yes	✗ No
SAP HANA	✓ Yes	✓ Yes	✗ No
Snowflake	✓ Yes	✓ Yes	✓ Yes
SQL Server	✓ Yes	✓ Yes	✗ No
Sybase	✓ Yes	✓ Yes	✗ No
Teradata	✓ Yes	✓ Yes	✗ No
Trino	✗ No	✓ Yes	✓ Yes Note Pushdown for Trino is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.

> Note While many of our certified JDBC data sources are compatible with Java 8 and earlier Spark versions, we only perform regression testing on the Java 11 and Spark 3.2.2 versions.

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	

Field	Description
Pushdown	✓ Yes Note Pushdown for Redshift is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.
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 Redshift data into Collibra Data Quality & Observability, 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. Example `jdbc:redshift://<host>:5439/dev`
✓ Yes	Driver Name	String	The driver class name of your Redshift connection. `com.amazon.redshift.jdbc42.Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `5439`
✕ 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.

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Username	The username of your Redshift account.
Password	The password of your Redshift account.
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/redshift_pwd_mgr.sh
Param $1	Optional. An additional parameter to authenticate your Redshift connection.
Param $2	Optional. An additional parameter to authenticate your Redshift connection.
Param $3	Optional. An additional parameter to authenticate your Redshift connection.

Locating the JDBC URL

Before you can connect to your Amazon 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 subprotocol that specifies to use the Amazon Redshift driver to connect to the database.
`endpoint`	The endpoint of the Amazon 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

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	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Impala data into Collibra Data Quality & Observability, 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.
✓ Yes	Connection URL	String	The connection string path of your Impala connection. 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`

Required	Connection Property	Type	Value
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `21050`
✗ 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. Optionally add the following driver property to remove the enforcement of unique column names: `hive.resultset.use.unique.column.names=false`

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Principal	The Kerberos entity to authenticate and grant access to your connection.
Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. Example /tmp/keytab/impala_user.keytab
Password	The secret credential associated with your Kerberos principal.
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
Param $1	Optional. Additional Kerberos parameter.
Param $2	Optional. Additional Kerberos parameter.
Param $3	Optional. Additional Kerberos parameter.
TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

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	

Field	Description
Pushdown	✓ Yes Note Pushdown for Athena is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.
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 Collibra Data Quality & Observability, 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.
			
			Example
			`jdbc:awsathena://AwsRegion=${region};User=${key};Password=${secret};S3OutputLocation=s3://${output_location};MetadataRetrievalMethod=Query`
			`jdbc:cdata:amazonathena:AwsRegion=northernvirginia;S3StagingDirectory=s3://<bucket name>;database=default;`

Required	Connection Property	Type	Value

✓ 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	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.

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

Username/Password

Field	Description
Username	The username of your Athena service account.
Password	The password of your Athena service account.

> Tip
> To use the Access Key + Secret Key combination, enter your Access Key as the Username and your Secret Key as the Password.
>
> To use Instance Profile, leave the Username and Password input fields blank.

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://`
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	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your CockroachDB data into Collibra Data Quality & Observability, 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.

Required	Connection Property	Type	Value
✓ Yes	Connection URL	String	The connection string path of your CockroachDB connection. 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	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.

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Username	The username of your CockroachDB account.
Principal	The Kerberos entity to authenticate and grant access to your connection.
Password	The password of your CockroachDB account.
Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. Example /tmp/keytab/impala_user.keytab
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
Param $1	Optional. An additional parameters to authenticate your CockroachDB connection.
Param $2	Optional. An additional parameter to authenticate your CockroachDB connection.

Field	Description
Param $3	Optional. An additional parameter to authenticate your CockroachDB connection.
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

Field	Description
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 Collibra Data Quality & Observability, 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];...
			
			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
			
			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.

Required	Connection Property	Type	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`

✓ Yes	Driver Name	String	The driver class name of your Databricks connection.
			`com.databricks.client.jdbc.Driver`
			`cdata.jdbc.databricks.DatabricksDriver`

✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `0`
✗ 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.
✓ Yes	Driver Properties	String	The configurable driver properties for your connection. When you specify `${access_token}` in the Connection URL, set the following properties on the **Properties** tab. **Property Name** is set to `access_token`. **Value** is the password.

Authentication

Username/Password

Field	Description
Username	The username of your Databricks service account. Set Username to `token`.

Field	Description
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.

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.

Connecting to Dremio

This section contains details for Dremio connections.

General information

Field	Description
Data source	Dremio
Supported versions	24.3.0
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 Collibra Data Quality & Observability, 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. 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	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	Driver Properties	String	The configurable driver properties for your connection.

Authentication

Username/Password

Field	Description
Username	The username of your Dremio account.
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 [[[Undefined variable DataQuality.full Google BigQuery]]] connections.

Select an option from the dropdown menu to display information for a particular driver class.

General information

Field	Description
Data source	[[[Undefined variable DataQuality.full 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
Spark agent	✓ Yes
Yarn agent	✓ Yes ✕ No
Parallel JDBC	✓ Yes

Field	Description
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

To bring your [[[Undefined variable DataQuality.BigQuery abbreviated]]] data into Collibra Data Quality & Observability, you need the following permissions:

- Read access on your [[[Undefined variable DataQuality.BigQuery abbreviated]]] data-set tables.
- [[[Undefined variable DataQuality.BigQuery abbreviated]]] 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.

✓ Yes	Connection URL	String	The connection string path of your [[[Undefined variable DataQuality.BigQuery abbreviated]]] connection.
			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="owl-hadoop-cdh";InitiateOAuth="GETANDREFRESH";OAuthJWTCert="/tmp/keytab/bigquery.json";OAuthJWTCertType="GOOGLEJSON";OAuthJWTIssuer="iam.service.account";`

✓ Yes	Driver Name	String	The driver class name of your [[[Undefined variable DataQuality.BigQuery abbreviated]]] 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	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 differ 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.

Authentication

Select an authentication type from the dropdown menu.

Field	Description
Username	The username of your [[[Undefined variable DataQuality.BigQuery abbreviated]]] service account. Set Username to any value.

Field	Description
Password	The password of your [[[Undefined variable DataQuality.BigQuery abbreviated]]] 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.

Field	Description
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.
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.

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_2.12-0.18.1.jar
3. Ensure these JARs present in <INSTALL_PATH>/owl/drivers/bigquery/:
   ****animal-sniffer-annotations-1.19.jargoogle-api-services-bigquery-v2-rev20201030-1.30.10.jargrpc-google-cloud-bigquerystorage-v1beta1-0.106.4.jarlistenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jarannotations-4.1.1.4.jargoogle-auth-library-credentials-0.22.0.jargrpc-google-cloud-bigquerystorage-v1beta2-0.106.4.jaropencensus-api-0.24.0.jarapi-common-1.10.1.-jargoogle-auth-library-oauth2-http-0.22.0.jargrpc-grpclb-1.33.1.jaropencensus-contrib-http-util-0.24.0.jarauto-value-annotations-1.7.4.jarGoogleBigQueryJDBC42.jargrpc-netty-shaded-1.33.1.jarperfmark-api-0.19.0.jaravro-1.10.0.jargoogle-cloud-bigquery-1.125.0.jargrpc-protobuf-1.33.1.jarprotobuf-java-3.13.0.jarchecker-compat-qual-2.5.5.jargoogle-cloud-bigquerystorage-1.6.4.jargrpc-protobuf-lite-1.33.1.jarprotobuf-java-util-3.13.0.jarcommons-codec-1.11.jargoogle-cloud-core-1.93.10.jargrpc-stub-1.33.1.jarproto-google-cloud-bigquerystorage-v1-1.6.4.jarcommons-compress-1.20.-jargoogle-cloud-core-http-1.93.10.jargson-2.8.6.jarproto-google-cloud-bigquerystorage-v1alpha2-0.106.4.jarcommons-lang3-3.5.jargoogle-http-client-1.38.0.jarguava-23.0.jarproto-google-cloud-bigquerystorage-v1beta1-0.106.4.jarcommons-logging-1.2.jargoogle-http-client-apache-v2-1.38.0.jarhttpclient-4.5.13.jarproto-google-cloud-bigquerystorage-v1beta2-0.106.4.jarconscrypt-openjdk-uber-2.5.1.jargoogle-http-client-appengine-1.38.0.jarhttpcore-4.4.13.jarproto-google-common-protos-2.0.1.jarcoregoogle-http-client-jackson2-1.38.0.jarj2objc-annotations-1.3.jarproto-google-iam-v1-1.0.3.jarerror_prone_annotations-2.4.0.jargoogle-oauth-client-1.31.1.jarjackson-annotations-2.11.0.jargrpc-alts-1.33.1.jarjackson-core-2.11.3.jarslf4j-api-1.7.30.jarfailureaccess-1.0.1.jargrpc-api-1.33.1.jarjackson-databind-2.11.0.jargax-1.60.0.jargrpc-auth-1.33.1.jarjavax.annotation-api-1.3.2.jarthreetenbp-1.5.0.jargax-grpc-1.60.0.jargrpc-context-1.33.1.jarjoda-time-2.10.1.jargax-httpjson-0.77.0.jargrpc-core-1.33.1.jarjson-20200518.jargoogle-api-client-1.31.1.jargrpc-google-cloud-bigquerystorage-v1-1.6.4.jarjsr305-3.0.2.jar

4. You may get a CLASSPATH conflict regarding the JAR files.
5. Ensure the BigQuery connector Scala version matches your Spark Scala version.

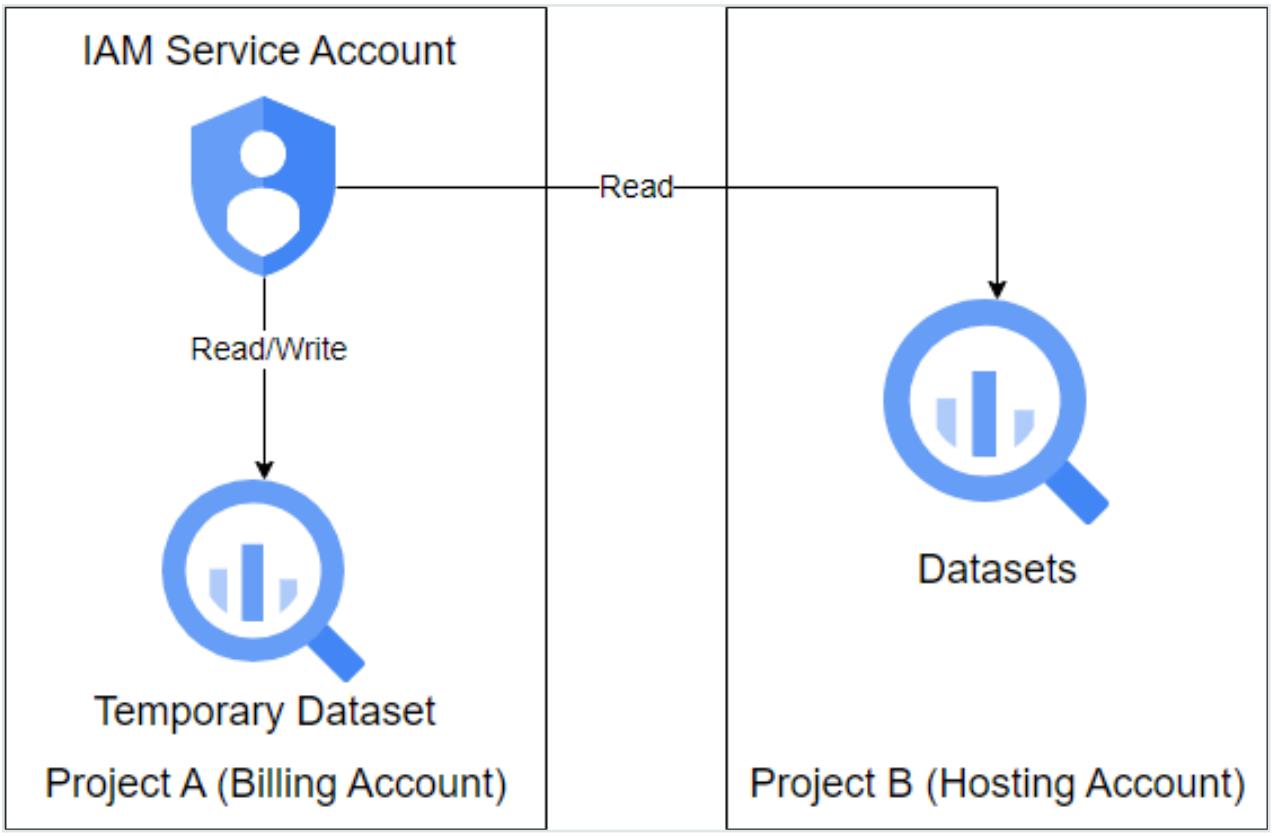
Known limitations

- 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.

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.

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 Collibra DQ and click **Admin** in the sidebar menu, then click **Connections**.
   » The Connections page opens.
2. Click the **BigQuery** tile.
   » The Add/Edit Connection modal appears.
3. Append the following properties to the Connection URL:

```
AdditionalProjects=project-b;QueryProperties=dataset_project_id=project-b
```

> Note Set the value of `project-b` according to your specific requirements.

4. 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.

5. Click **Save**.
6. Click **Explorer** and verify that the records in Explorer are the same as in your new BigQuery connection.
7. 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	2.6.19.1022
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 Collibra Data Quality & Observability, 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	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 Hive connection. 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.jdbc41.HS2Driver`
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `10000`
✕ 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. Optionally add the following driver property to remove the enforcement of unique column names: `hive.resultset.use.unique.column.names=false`

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Principal	The Kerberos entity to authenticate and grant access to your connection.
Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. Example /tmp/keytab/hive_user.keytab
Password	The secret credential associated with your Kerberos principal.
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
Param $1	Optional. Additional Kerberos parameter.
Param $2	Optional. Additional Kerberos parameter.
Param $3	Optional. Additional Kerberos parameter.

Field	Description
TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

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

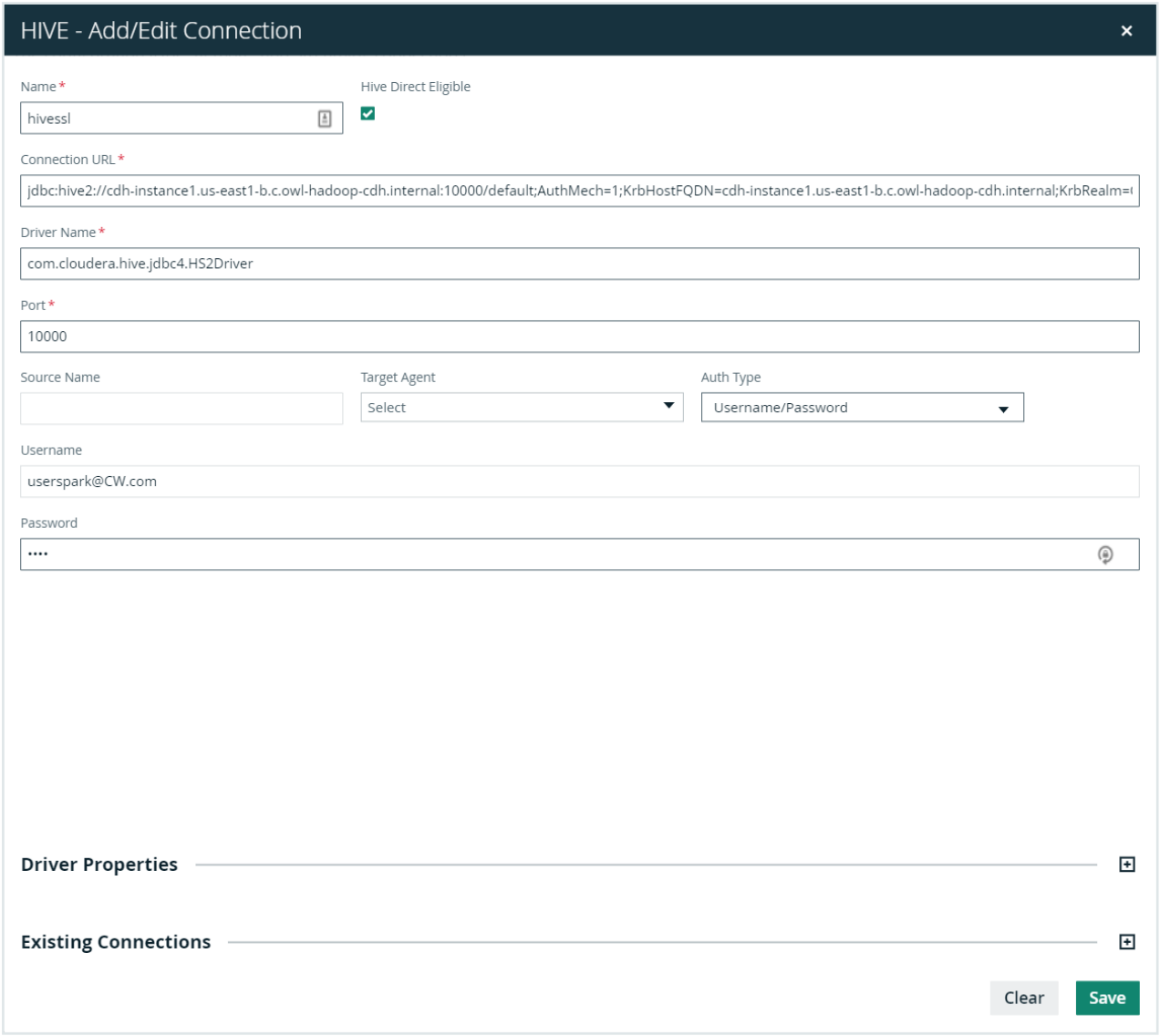
The SSL KeyStore to be used to = SSLKeyStore=/opt/cloudera/security/pki/cdh-instance1.us-east1-b.c.owl-hadoop-cdh.internal-server.jks *(Could use SSLTrustStore also)*

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 (See ScreenShot below)



Name = hivessl

Connection URL = 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

Port = 10000

Driver Name = com.cloudera.hive.jdbc4.HS2Driver

Username = userspark@CW.COM

Password = password

Connecting to Db2

This section contains details for IBM Db2 connections.

General information

Field	Description
Data source	IBM Db2
Supported versions	4.21.29
Connection string	`jdbc:db2://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	
Pushdown	✗ No
Spark agent	✓ Yes

Field	Description
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 Collibra Data Quality & Observability, 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.
✓ Yes	Connection URL	String	The connection string path of your Db2 connection. Example `jdbc:db2://$host:50000:owldb2`
✓ Yes	Driver Name	String	The driver class name of your Db2 connection. `com.ibm.db2.jcc.DB2Driver`

Required	Connection Property	Type	Value
✓ Yes	Port	Integer	The port number to establish a connection to the datasource. The default port is `50000`
✕ 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.

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Username	The username of your Db2 account.
Password	The password of your Db2 account.

Field	Description
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
Param $1	Optional. An additional parameters to authenticate your Db2 connection.
Param $2	Optional. An additional parameter to authenticate your Db2 connection.
Param $3	Optional. An additional parameter to authenticate your Db2 connection.

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

Field	Description
Supported version	3.12.1122.0.8326.0
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 Collibra Data Quality & Observability, 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. Example `jdbc:mongodb://cluster0-shard-00-01.upozn.mongodb.net?AuthDatabase=admin&AuthScheme=SCRAM-SHA-1&UseSSL=true&database=<database_name>`

✓ 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`
✗ 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	Driver Properties	String	The configurable driver properties for your connection.

Authentication

Username/Password

Field	Description
Username	The username of your MongoDB account.
Password	The password of your MongoDB account.

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

Field	Description
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	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your MySQL data into Collibra Data Quality & Observability, 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. `jdbc:mysql://$host:3306/{database_name}?useSSL-L=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`
✕ 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.

Authentication

Username/Password

Field	Description
Username	The username of your MySQL account.
Password	The password of your MySQL account.

Connecting to Oracle

This section contains details for Oracle connections.

General information

Field	Description
Data source	Oracle
Supported versions	12.2.0.1.0
Connection string	`jdbc:oracle:thin`
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	✓ Yes
Java Platform version compatibility	
JDK 8	✓ Yes
JDK 11	✓ Yes

Minimum user permissions

In order to bring your Oracle data into Collibra Data Quality & Observability, 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. Example `jdbc:oracle:thin:@$host:SID`

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	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.

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`

Name	Short Name	Type	Description
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")
includeSynonyms	synonyms	String (containing boolean value)	"true" to include column information from predefined "synonym" SQL entities when you execute a `DataBaseMetaData getColumns()` call; equivalent to connection `setIncludeSynonyms()` call (default value is "false")
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

Use the dropdown menu to select an authentication type.

Field	Description
Username	The username of your Oracle account.
Password	The password of your Oracle account.
Principal	The Kerberos entity to authenticate and grant access to your connection. Example test_user@example.collibra.com
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
Password	The secret credential associated with your Kerberos principal.

Field	Description
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
Param $1	Optional. Additional Kerberos parameter.
Param $2	Optional. Additional Kerberos parameter.
Param $3	Optional. Additional Kerberos parameter.
TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

Known 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 uppercase, Oracle also treats lower case characters as special characters. A possible workaround 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\" "`

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 Collibra Data Quality & Observability, 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. 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`
✗ 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	Driver Properties	String	The configurable driver properties for your connection.

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Username	The username of your PostgreSQL account.
Password	The password of your PostgreSQL account.
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
Param $1	Optional. An additional parameter to authenticate your PostgreSQL connection.
Param $2	Optional. An additional parameter to authenticate your PostgreSQL connection.
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://`
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 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 Collibra Data Quality & Observability, you need the following permissions.

- Read access on your SAP HANA 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 SAP HANA connection. Example `jdbc:sap://$host:30015`
✓ 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`
✕ 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.

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Username	The username of your SAP HANA account.
Password	The password of your SAP HANA account.
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
Param $1	Optional. An additional parameter to authenticate your SAP HANA connection.
Param $2	Optional. An additional parameter to authenticate your SAP HANA connection.
Param $3	Optional. An additional parameter to authenticate your SAP HANA connection.

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	3.14.4
Connection string	`jdbc:snowflake://`
Packaged?	✓ Yes
Certified?	✓ Yes
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	

Field	Description
Pushdown	✕ No Important To see information for Snowflake Pushdown, click the Pushdown tab above.
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 Snowflake data into Collibra Data Quality & Observability, 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.

✓ Yes	Connection URL	String	The connection string path of your Snowflake connection.
			Example
			`jdbc:snowflake://<account>.snowflakecomputing.com/?db=SNOWFLAKE_SAMPLE_DATA`
			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`
✗ 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	Pushdown	Option	Switches Pushdown processing on or off. Important This option should not be selected for Pullup connections.
✕ No	Driver Properties	String	The configurable driver properties for your connection. 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

Username/Password

Field	Description
Username	The username of your Snowflake service account.
Password	The password of your Snowflake service account.

> Tip To use JWT authentication, enter any value as the Username and your token value as the Password.

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.

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	3.14.4
Connection string	`jdbc:snowflake://`
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 Collibra Data Quality & Observability, 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.
			Example
			`jdbc:snowflake://<account>.snowflakecomputing.com/?db=SNOWFLAKE_SAMPLE_DATA`
			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`
✕ 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.

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	Pushdown	Option	Switches Pushdown processing on or off. Important This option must be selected for Pushdown processing to work correctly.
✗ 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 Properties	String	The configurable driver properties for your connection. 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

Username/Password

Field	Description
Username	The username of your Snowflake service account.
Password	The password of your Snowflake service account.

> Tip To use JWT authentication, enter any value as the Username and your token value as the Password.

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.
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.

Connecting to SQL Server

This section contains details for SQL Server connections.

General information

Field	Description
Data source	SQL Server
Supported versions	N/A
Connection string	`jdbc:sqlserver://`
Packaged?	✓ Yes

Field	Description
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	✗ No

Minimum user permissions

In order to bring your SQL Server data into Collibra Data Quality & Observability, 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.
✓ Yes	Connection URL	String	The connection string path of your SQL Server connection. Example `jdbc:sqlserver://$host:1433;databaseName=ContosoRetailDW`
✓ 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	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.

Authentication

Use the dropdown menu to select an authentication type.

Field	Description
Username	The username of your SQL Server account.
Password	The password of your SQL Server account.
Principal	The Kerberos entity to authenticate and grant access to your connection. Example test_user@example.collibra.com
Keytab	The file path of the keytab file that contains the encrypted key for a Kerberos principal. Example /tmp/keytab/sqlserver_user.keytab
Password	The secret credential associated with your Kerberos principal.
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
Param $1	Optional. Additional Kerberos parameter.
Param $2	Optional. Additional Kerberos parameter.

Field	Description
Param $3	Optional. Additional Kerberos parameter.
TGT Cache	The ticket-granting ticket cache that stores the TGT to authenticate your connection.

> Note Collibra DQ does not currently support Azure service principal authentication.

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
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 Collibra Data Quality & Observability, 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. Example `jdbc:sybase:Tds:$host:5000/template1?ENCRYPT_PASSWORD=false`
✓ 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	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	Driver Properties	String	The configurable driver properties for your connection.

Authentication

Username/Password

Field	Description
Username	The username of your Sybase service account.
Password	The password of your Sybase service account.

Connecting to Teradata

This section contains details for Teradata connections.

General information

Field	Description
Data source	Teradata
Supported versions	16.20.00.13
Connection string	`jdbc:teradata://`
Packaged?	✓ Yes

Field	Description
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 Collibra Data Quality & Observability, 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. 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`
✕ 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. Add the following driver property: `RBAC=USR_RBAC_CCB_PDQOWL_INTL`

Authentication

Username/Password

Field	Description
Username	The username of your Teradata account.
Password	The password of your Teradata account.

Connecting to Trino

This section contains details for Trino connections.

General information

Field	Description
Data source	Trino
Supported versions	405.0
Connection string	`jdbc:trino://`
Packaged?	✓ Yes
Certified?	✗ No
Supported features	
Estimate job	✓ Yes
Analyze data	✓ Yes
Schedule	✓ Yes
Processing capabilities	

Field	Description
Pushdown	✓ Yes Note Pushdown for Trino is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.
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 Collibra Data Quality & Observability, 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.
			Example
			`jdbc:trino://${host}:443/<catalog>/<schema>&source=jdbc:presto`
			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`
✗ 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.
✓ Yes	Driver Properties	String	The configurable driver properties for your connection.

Authentication

Username/Password

Field	Description
Username	The username of your Trino account.
Password	The password of your Trino account.

Known limitations

- Trino connection URLs must contain the following substring: `&source=jdbc:presto`
  - This requirement will be addressed in a future release.

Supported remote files

This section includes the connection details of remote files supported by Collibra DQ.

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

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 Collibra Data Quality & Observability 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
Supported versions	N/A
Connection string	`s3://`
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 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.

Required	Connection Property	Type	Value
✕ No	HTTPS	Option	Converts the Connection URL field from an `s3://` to `https://` URI path.
✓ 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**

✓ 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.

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. 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.

Authentication

Select an authentication type from the dropdown menu.

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.

Field	Description
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.
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.

Field	Description
Password	Note Password is not required when you use Password Manager.
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 end-point URI	**s3-endpoint=s3**
NetApp	**s3-endpoint=netapp**

Known 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://`

Field	Description
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>.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	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.

Connection URL elements

Field	Description
File scheme	The `wasbs` protocol is used as the scheme identifier.

Field	Description
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.

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.

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	`abfs://`
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 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.
✓ Yes	Connection URL	String	The connection string value of your ADLS connection. `abfs://<file system>@<account name>.d-fs.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.

Connection URL elements

Field	Description
File scheme	The `abfs` 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.

Field	Description
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.

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.

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	Driver Properties	String	The configurable driver properties for your connection. 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.

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	✗ 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 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.

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-----\encodedstring\n-----END PRIVATE KEY-----\n", "client_email": "<your-service-account>@developer.gserviceaccount.com", "client_id": "your-service-account-id", "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 This JSON file is typically available to download in GCP when you create a service account in your IAM profile.

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

Field	Description
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.
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.

Authentication

Auth Type must be set to **NFS**.

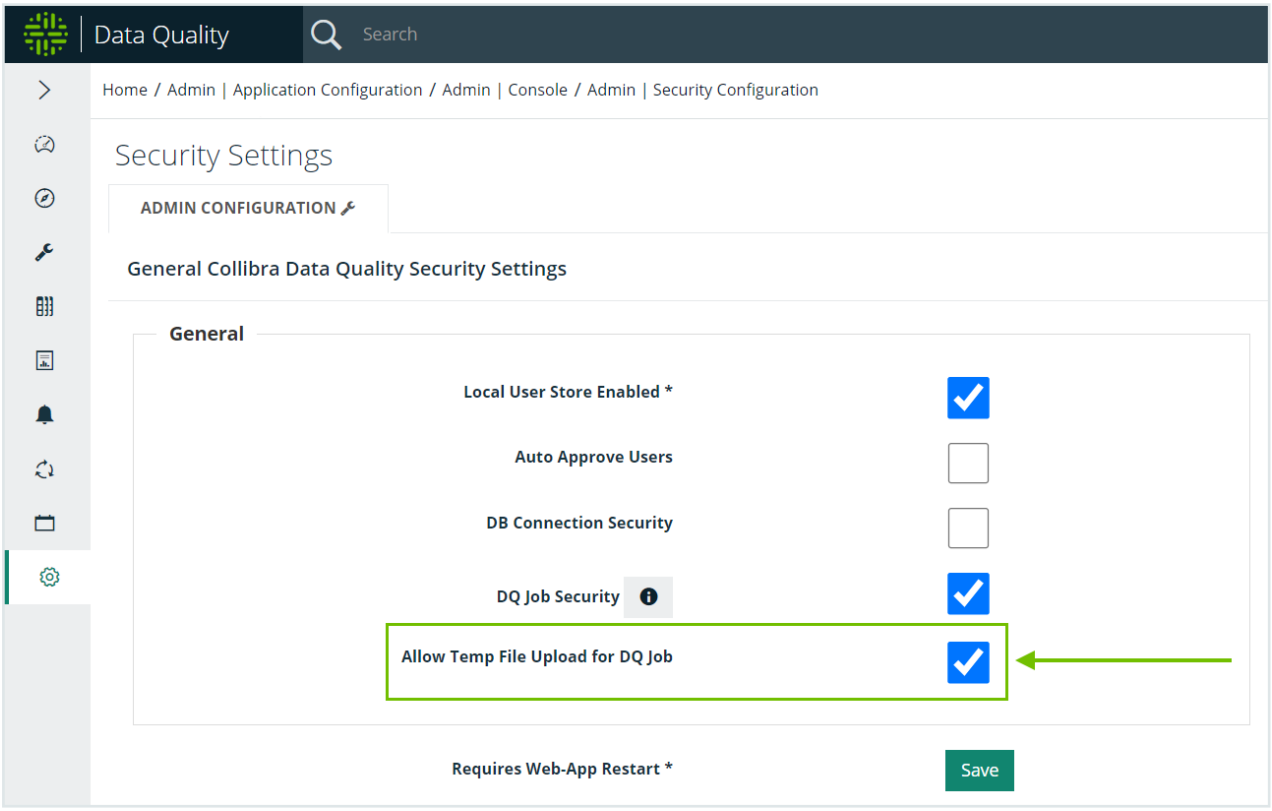
Connecting to Temp Files

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

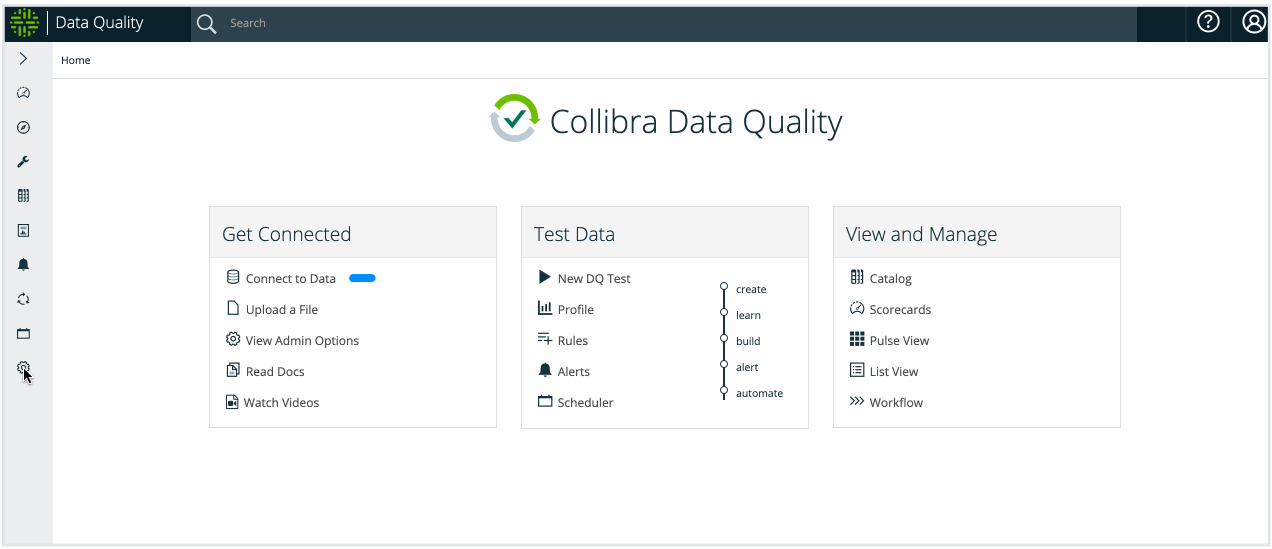
1. In the main menu, click the **Explorer** button.
   » The Explorer expands.
2. Click **Temp Files**.
   » Temp Files expands.
3. Click **Add Temp File**.
   » Upload Temp File alert opens.
4. Click **Choose File**.
5. Select a file from your local drive.
6. Click **Open**.
   » Your file loads into 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.

Adding Connections



Prerequisites

To add a connection, you need:

- A Collibra DQ account with ROLE_ADMIN permissions.
- Access to the account credentials of your connection, for example, PostgreSQL.

Adding a JDBC connection

This section shows you how to connect to a JDBC connection from the Connections Management page.

1. Sign into Collibra DQ.
2. Hover your cursor over the ⚙ icon and click **Admin Console**.
   » The Admin Console opens.
3. Click **Connections**.
   » The **Connections Management** page opens.
4. Click **Add** next to a JDBC connection.
   » The **New JDBC Connection** template appears.
5. Enter the required information.

Required	Connection Property	Type	Value
✓ 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:<data-source>://$host:<-port>/<AdditionalProperties>` Replace the Connection URL to point to the database server to run DQ Jobs against.
✓ Yes	Driver Name	String	The driver class name. For example, a Google BigQuery JDBC driver may use the following convention: `com.simba.googlebigquery.jdbc42.Driver`
✓ Yes	Port	Integer	The port for the connection to the data source host.
✗ 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 on the Catalog page and 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.

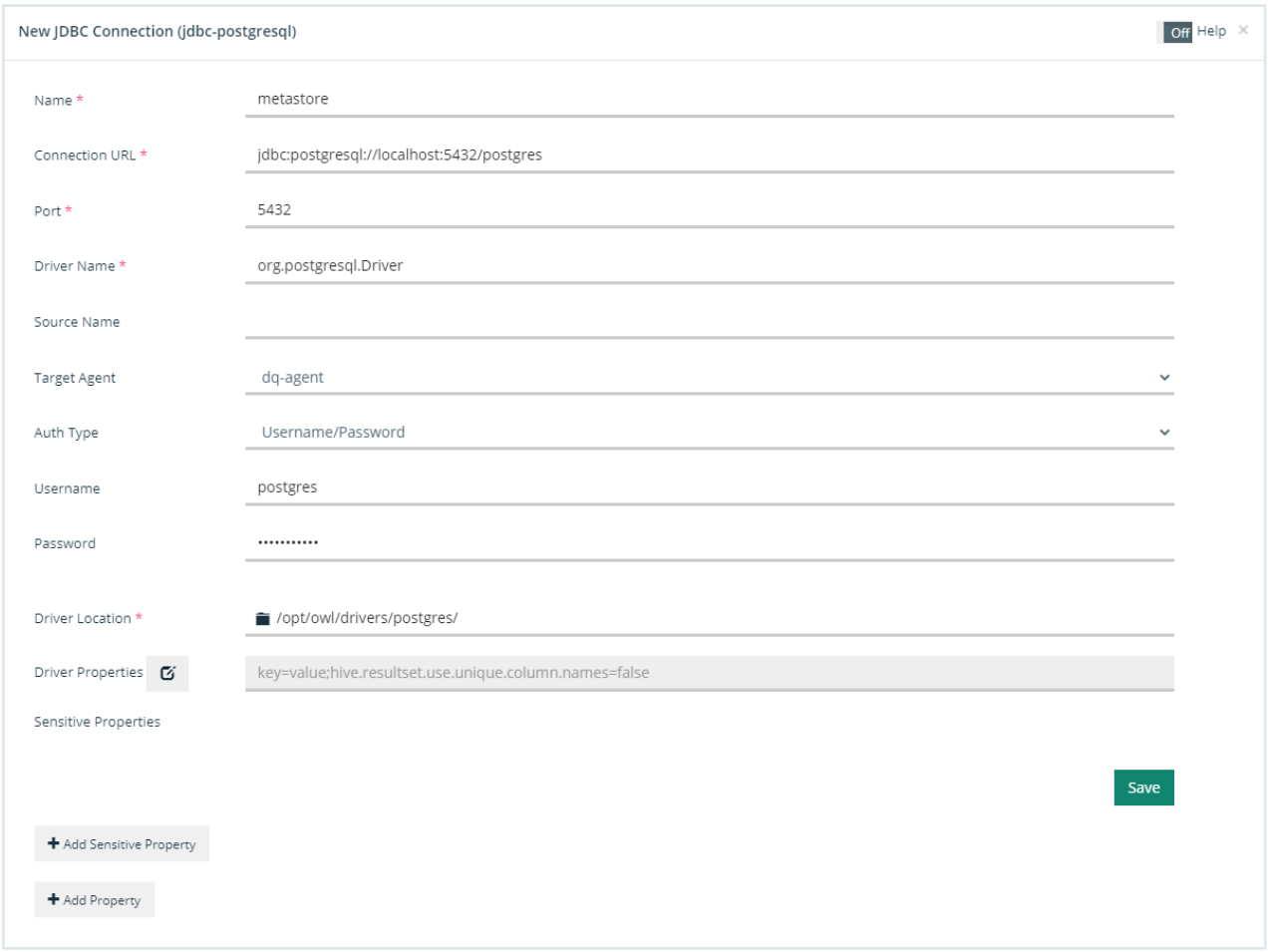
Required	Connection Property	Type	Value
✕ No	Target Agent	String	The Agent used to submit your DQ Job.
✕ No	Pushdown	Option	Enables Pushdown processing. This option is required for Pushdown Job submission. Note This option is currently only available for Snowflake and Databricks (beta) connections.
✓ Yes	Auth Type	Option	The method to authenticate your connection. Note The configuration requirements differ depending on the Auth Type you select. See Adding Connections for more details on available authentication types.
✓ 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.
✕ No	Driver Properties	String	The configurable driver properties of your connection.

<table>
<tr><th>Required</th><th>Connection Property</th><th>Type</th><th>Value</th></tr>
<tr><td>✕
No</td><td>Sensitive Properties</td><td>String</td><td>Define sensitive properties to hide in the Connection URL.
Click Add Sensitive Property, then enter your information in the following format:
<table>
<tr><th>Sensitive Property</th><th>Description</th></tr>
<tr><td>property name</td><td>The variable to display in place of the sensitive information. This can be any property, such as <code>Password</code>
In the Connection URL, replace the sensitive information with this value. In the following example, <code>dquser</code> and <code>dqpassword</code> are the property names that replace the sensitive property values:
<code>jdbc:postgresql://localhost:5432/postgres?user=${dquser}&password=${dqpassword}</code></td></tr>
<tr><td>property value</td><td>The sensitive value to hide in the Connection URL. For example, if the password value of the connection credential is password123, enter <code>password123</code> in the property value field.</td></tr>
</table>
Tip If you need to redefine the value of a sensitive property, click the 🗑 next to a saved sensitive property, then re-add it with the updated property value.</td></tr>
</table>

6. Click **Save**.

PostgreSQL connection template example

> **Note** This example uses a connection named `metastore` that connects to a local PostgreSQL server (`localhost:5432/postgres`)



Link Connection to Agent

> **Note** Make sure to link a DB connection to a DQ Agent, if required.

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.

Remote file connections

About remote file connections

This section gives an overview of the supported file formats and the limitations when connecting to a remote file.

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

Type	File structure	Notes
JSON (.json)	Semi-structured	N/A
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

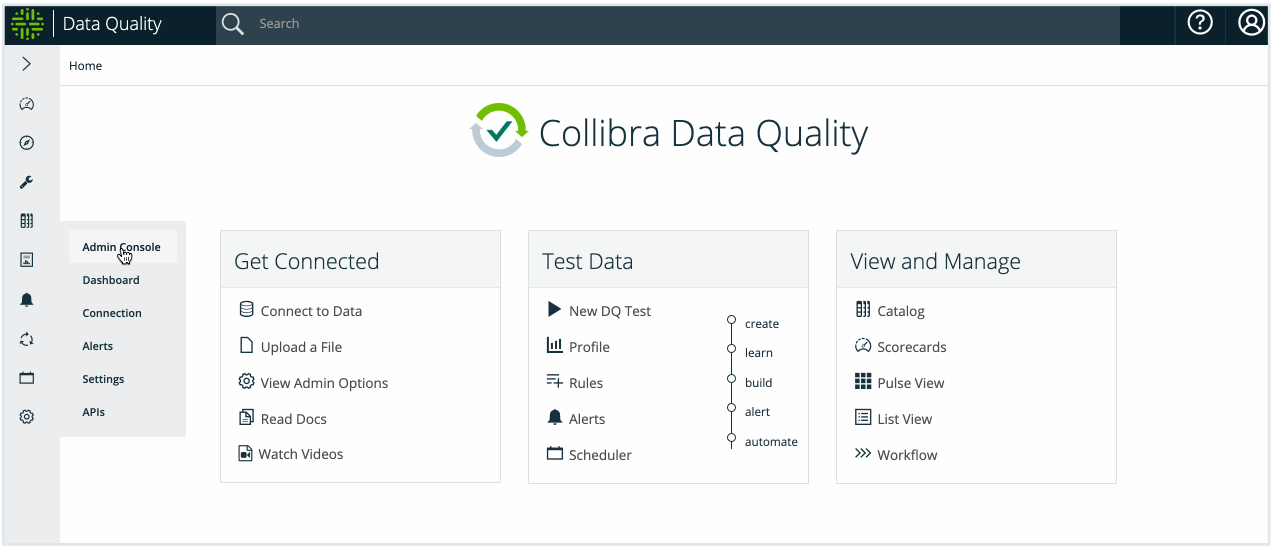
The following table is a list of supported delimiters available in the Delimiter dropdown menu.

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

- 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.

Adding a connection to a DQ Agent

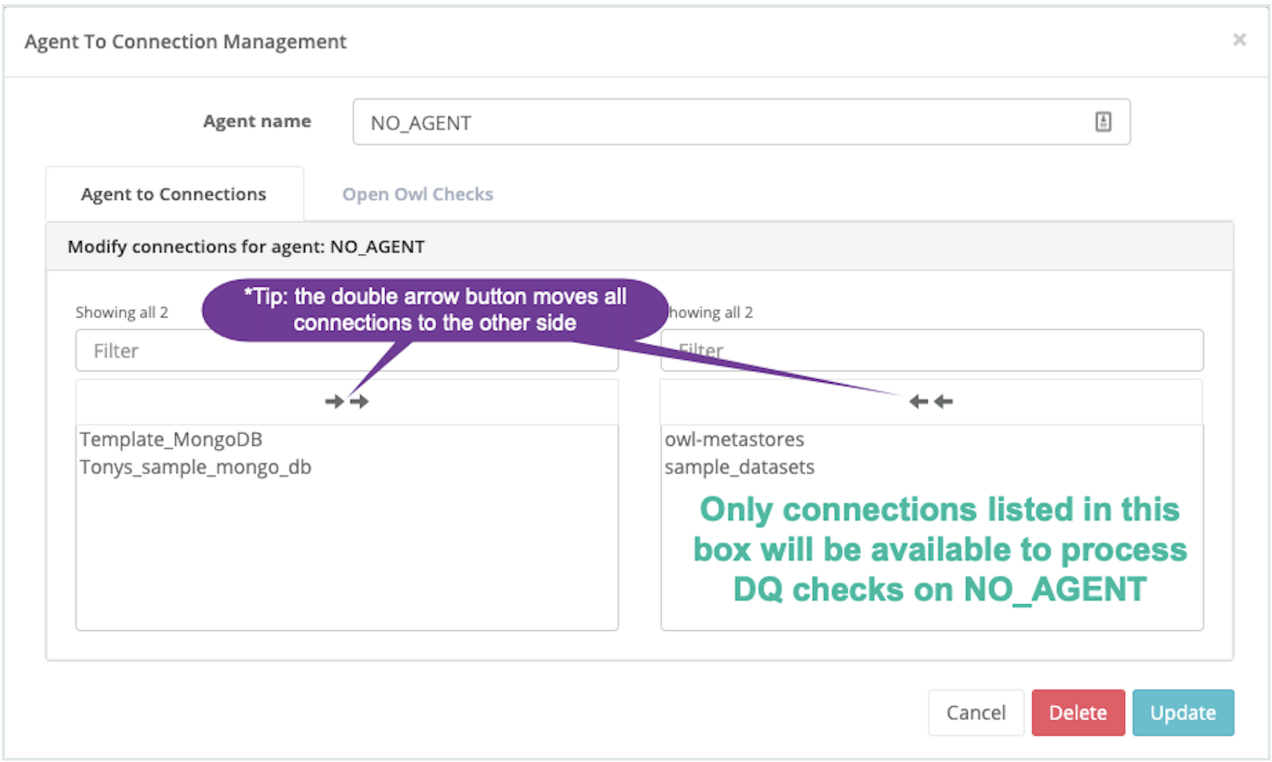


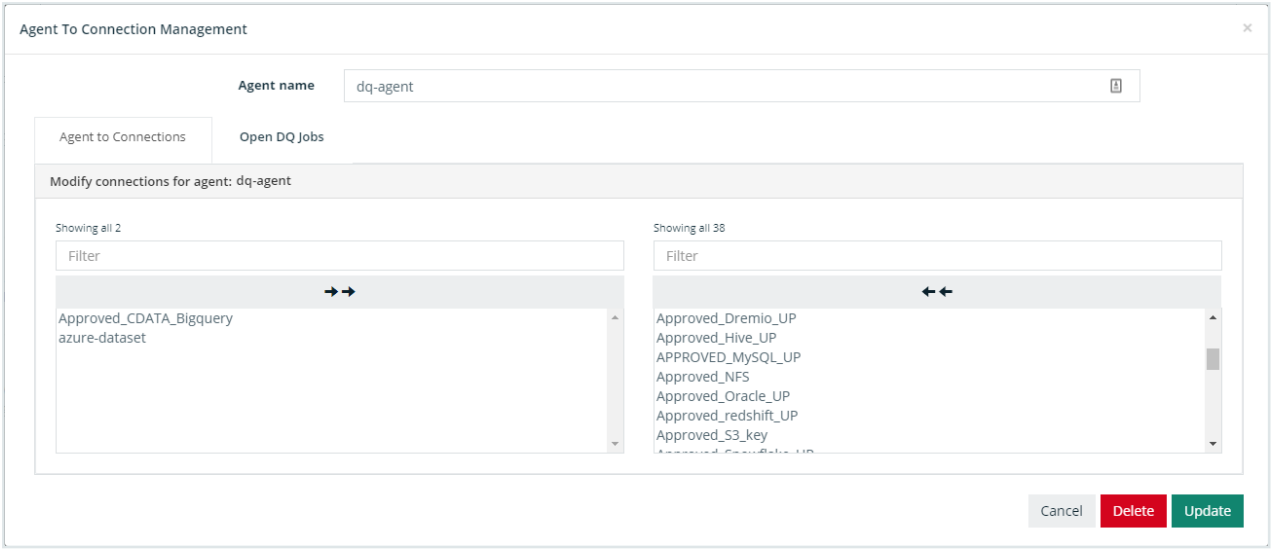
How to link a DB connection to an Agent via the UI

When you add a new database connection, the DQ Agent must be given the permission to run DQ job via the specified agent.

From Fig 3, select the chain link icon next to the DQ Agent to establish a link to the DB connection. A modal to add give that agent permission to run DQ jobs by DB connection name will show (Fig 5). The left-side panel is the list DB connection names that has not been linked to the DQ Agent. The right-side panel is the list of DB connection names that has the permission to run DQ job.

Double click the DQ connection name to move from left to right. In Fig 5, DB connection named "metastore" is being added to DQ Agent. Click the "Update" button to save the new list of DB Connections. Fig 5: Adding DB Connection named "metastore" to the DQ Agent. Fig 6: How to add all connections to the selected DQ Agent





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 Kubernetes and YARN deployments of Collibra DQ.

> Note Kerberos authentication is not supported on Standalone deployments of Collibra DQ.

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
```

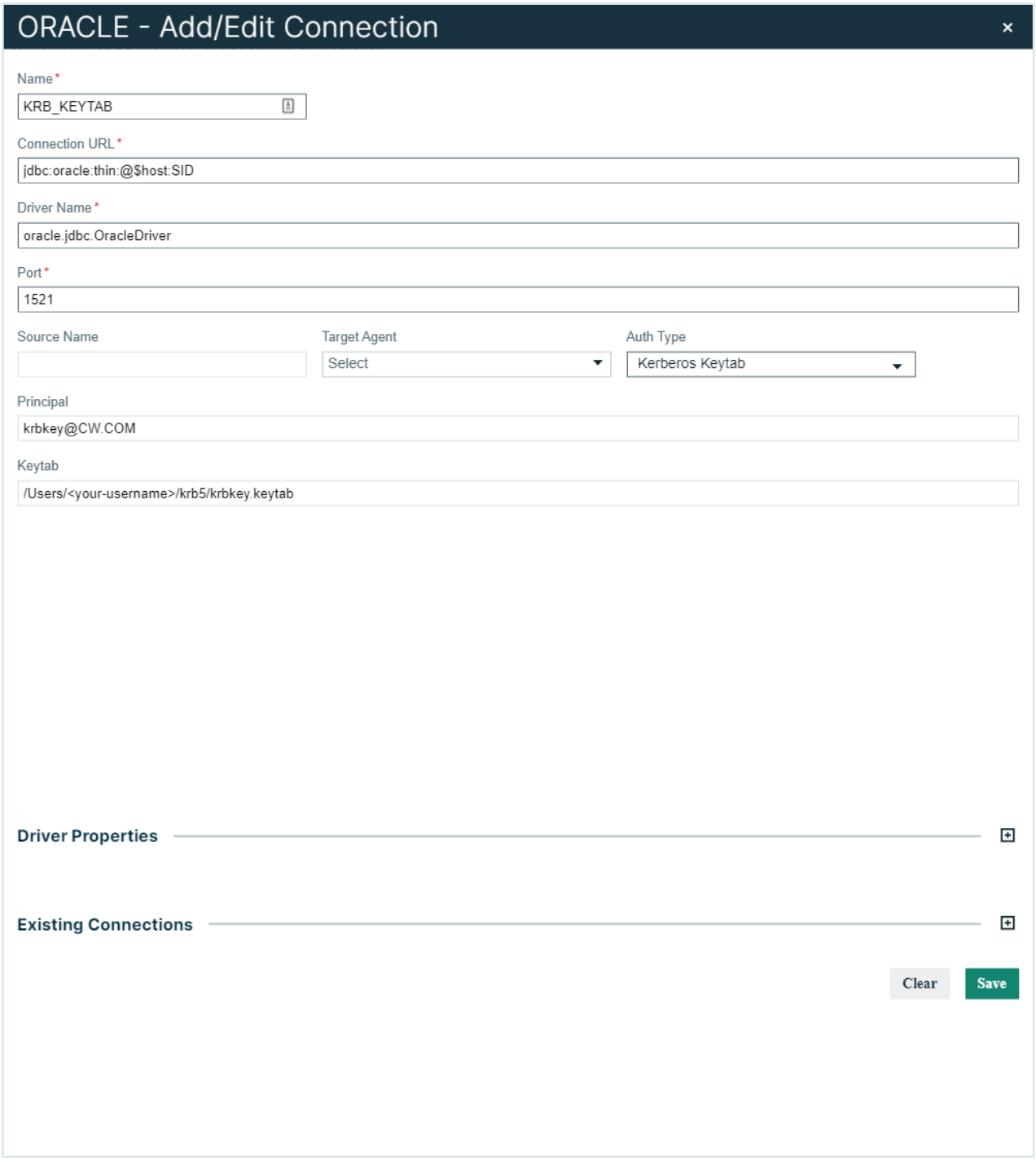
> Note You can define the location of your file, for example `/tmp/`.

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
```

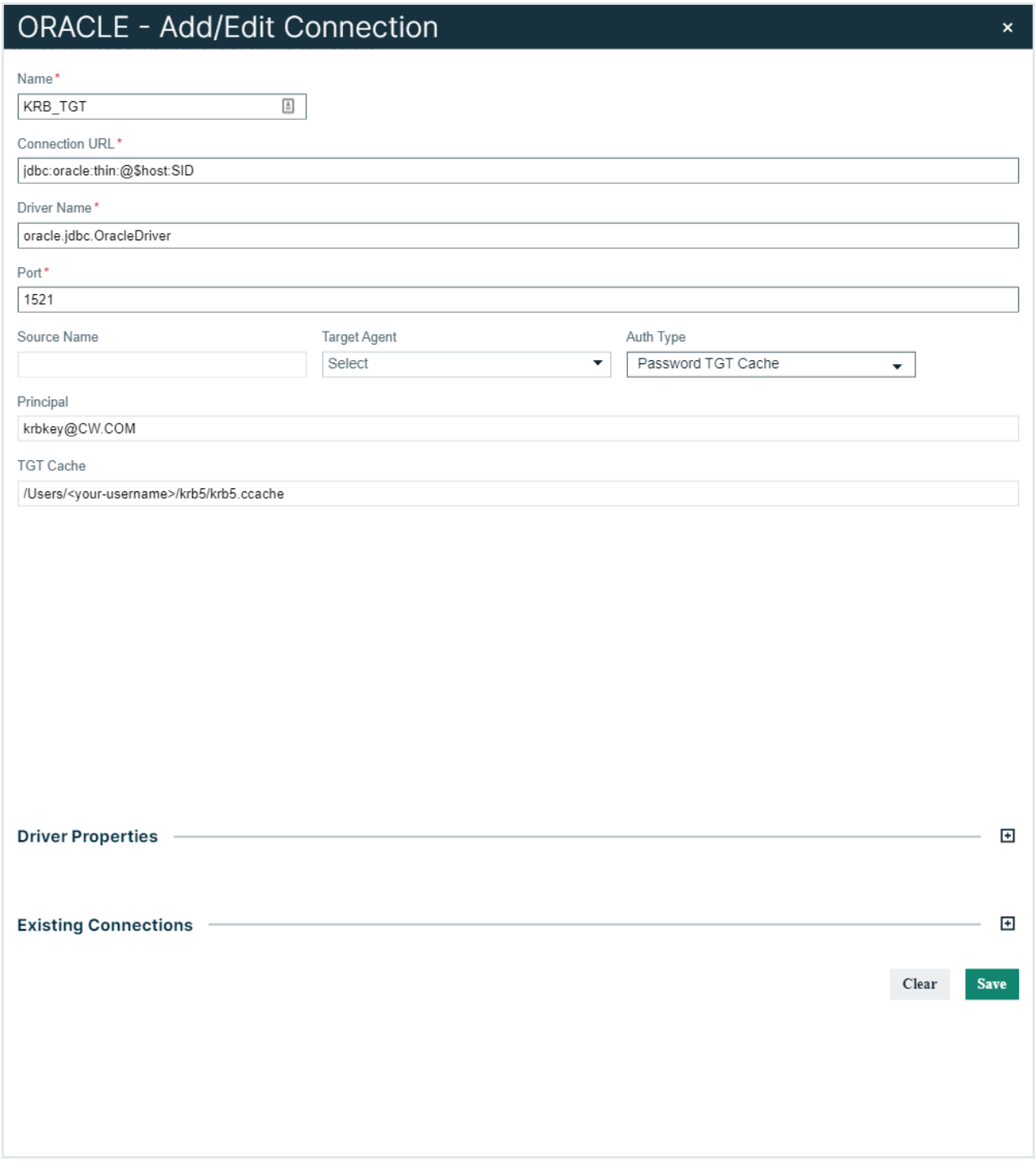
> Note You can define the location of your file, for example `/tmp/`.

5. Run the following command with your username to view the details of your TGT file:

```
klist <your-username>
```

View example screenshot...[1]

1



Collibra DQ Features

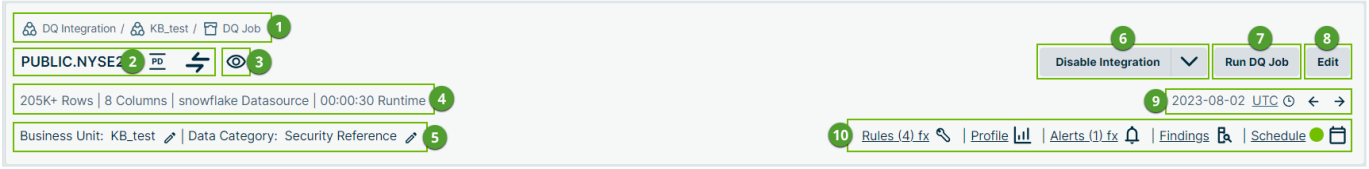
Navigation

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 Data Intelligence Cloud is enabled, the breadcrumbs show the community hierarchy to which the integration is mapped. In order, the breadcrumbs show the Collibra Data Intelligence Cloud 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 Data Intelligence Cloud.
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 Data Intelligence Cloud Data Category lets you assign your dataset to a data category.
6	Integration options	Enable, disable, or reset an integration within Collibra Data Intelligence Cloud. Note ROLE_ADMIN and ROLE_DATASET_MANAGER are required to use this.

No.	Component	Description
7	Run DQ Job	Submits your job for processing. When a job runs, you can find it on the Jobs page.
8	Edit	Opens Explorer so you can edit your dataset.
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

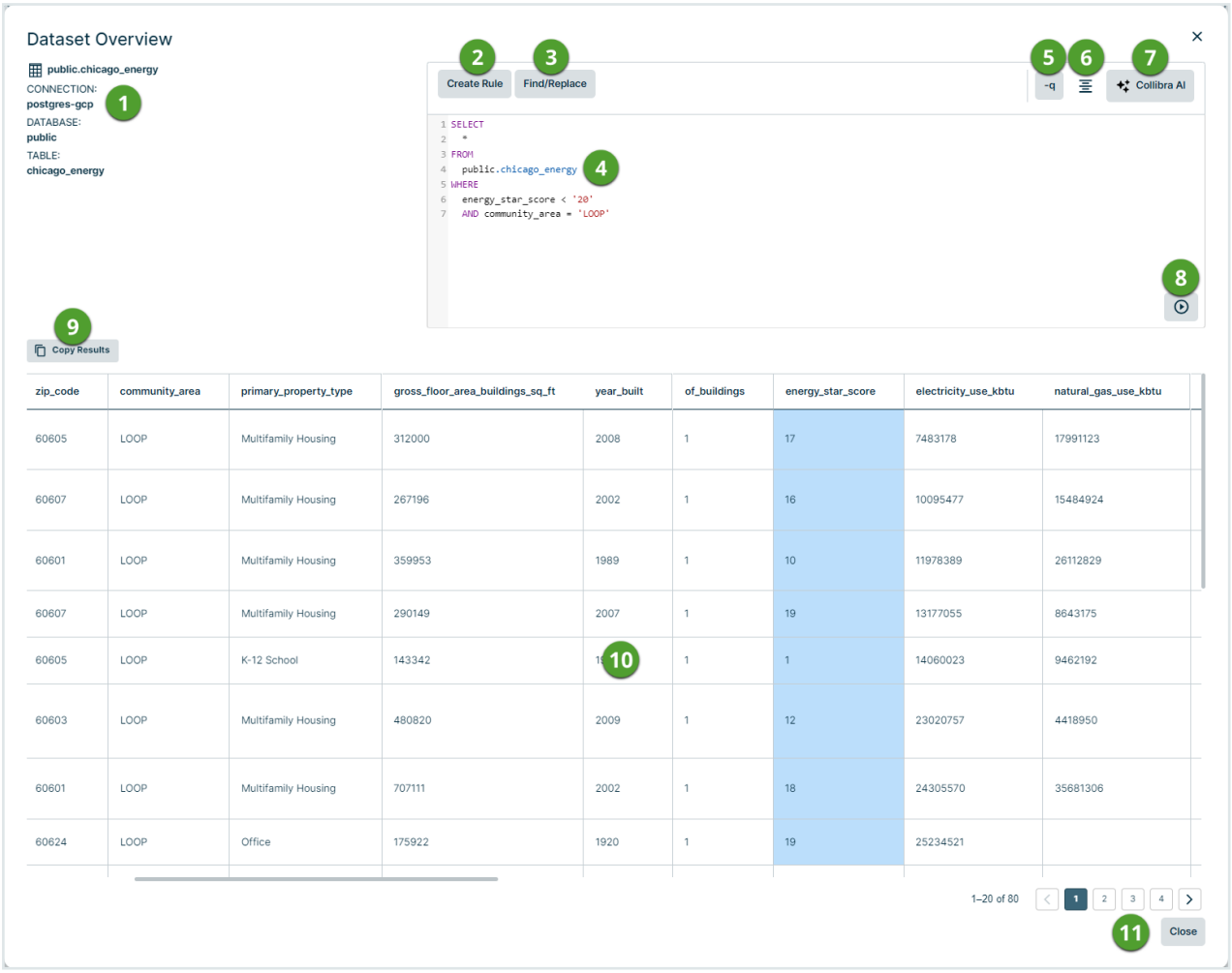
Dataset Overview allows you to execute SQL queries, instantly view results, and conduct initial exploration of new datasets. Additionally, you can directly create and add rules to your jobs. With the power to write SQL to query your dataset, you can accelerate the data discovery process and extrapolate important insights in real time.

From the Metadata Bar on any dataset-level page, click the to open Dataset Overview.

Prerequisites

You have either ROLE_ADMIN or ROLE_VIEW_DATA assigned to your user account.

About Dataset Overview



No.	Component	Description
1	Dataset information	Displays the number of rows and columns in your dataset, as well as the name of the datasource.

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	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.
5	-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.
6	Pretty print	Formats your SQL query in a more readable layout.

No.	Component	Description
7	Collibra AI (beta)	Collibra AI allows 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 Collibra AI. Tip If you are interested in participating in the Collibra AI private beta, please contact your Collibra CSM for more details.
8	Run	Runs the SQL in the editor and updates the overview table.
9	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.
10	Overview table	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 in 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.
11	Pagination and Close	Click the available pages or use the arrows to navigate datasets with more than 20 rows. Click Close to exit the Dataset Overview modal.

Using the SQL editor

The SQL editor helps you quickly navigate and extract the information you need from your dataset.

1. If your organization participates in the Collibra AI private beta, 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 in 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

There is a limitation where the rule name is not checked to already exist. When you name a rule that matches an existing rule name, the previous rule is replaced by the new one.

[[[Undefined variable DataQuality.SQL assistant]]] (Beta)

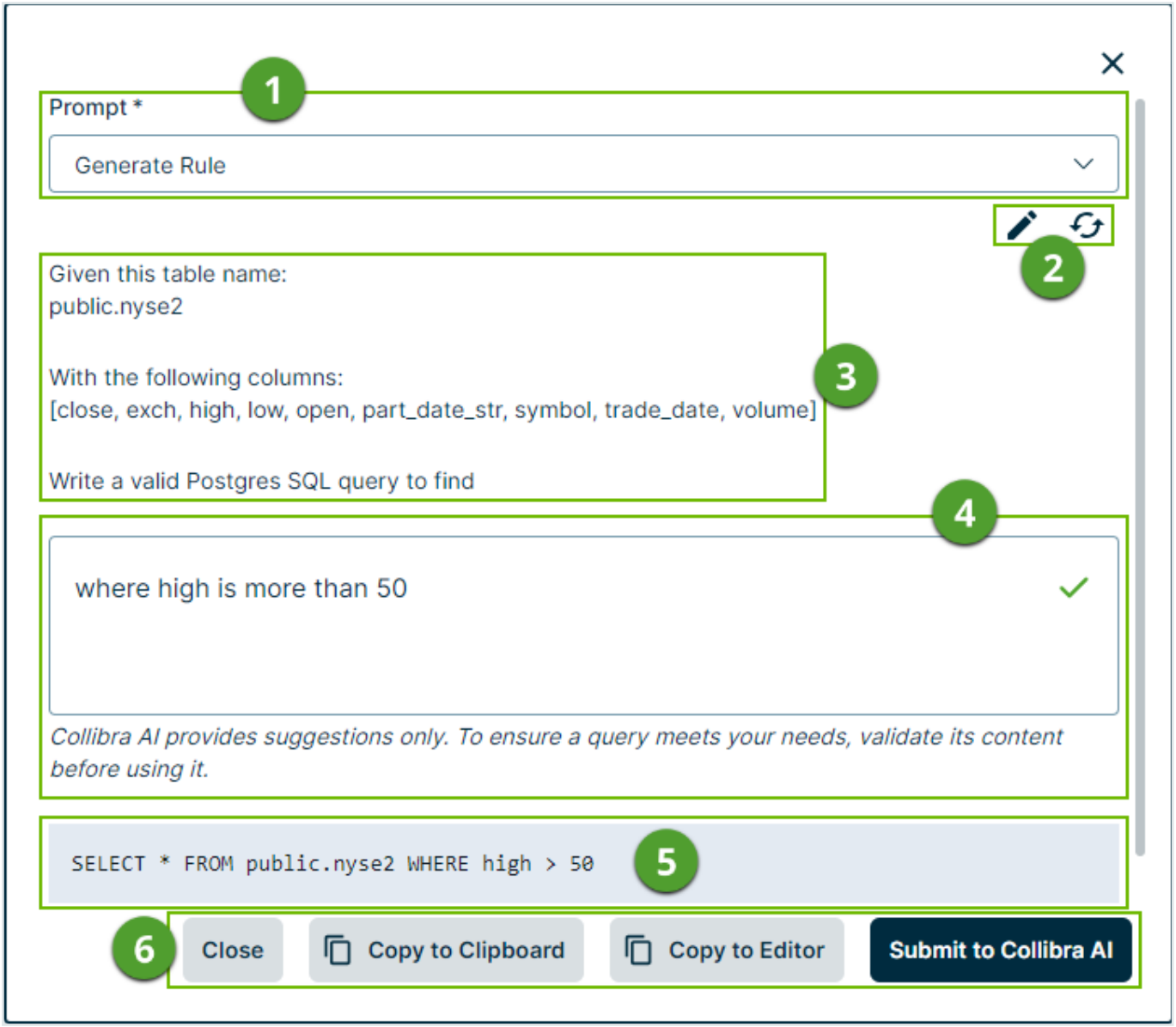
> Note This feature is in Beta testing and is powered by Collibra AI. If you would like to share your feedback with our product team, please go to the Collibra beta platform.

[[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]], 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 [[[Undefined variable DataQuality.SQL assistant]]] allow you to elegantly correct syntactical mistakes, resolving the once tedious SQL syntax reviews of the past.

You can use [[[Undefined variable DataQuality.SQL assistant]]] to assist with SQL rule writing and troubleshooting on the following pages:

- Dataset Overview
- Rule Workbench



No.	Component	Description
1	Collibra AI options	Click the dropdown menu and select an option. • **Generate Rule** creates SQL suggestions based on your plain-text input in the request prompt. • **Troubleshoot Rule** provides suggestions to fix and rewrite 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.
2	Edit and reset prompt	Optionally click ✎ to manually edit the default request and request prompt, or click ↻ to refresh your request prompt.
3	Default request	The dataset metadata that instructs [[[Undefined variable DataQuality.SQL assistant]]] on what to include in the query. This is typically all of the details before the `WHERE` clause in a SQL query.
4	Request prompt	An input field where you can enter a prompt for [[[Undefined variable DataQuality.SQL assistant]]] in plain-text format.
5	Results	The results that [[[Undefined variable DataQuality.SQL assistant]]] returns based on the content of your prompt.
6	Action buttons	• 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 on the Dataset Overview modal. • 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.

Prerequisites

To use [[[Undefined variable DataQuality.SQL assistant]]] in Collibra DQ, you first need to:

- Ensure that the AI_TENANT application configuration is set to **TRUE** in the Admin Console ▸ Configuration Settings.
- Safelist network egress for **oauth2.googleapis.com** and **us-central1-aiplatform.googleapis.com** over port 443.
- Have one of the following Collibra DQ roles assigned to your user account:
  - ROLE_ADMIN
  - ROLE_GENAI_USER

> Note Collibra DQ administrators can add or update roles from the Admin Console ▸ Role Management ▸ Roles.

Using [[[Undefined variable DataQuality.SQL assistant]]]

[[[Undefined variable DataQuality.SQL assistant]]] uses Collibra AI to automatically write and troubleshoot SQL queries. When working on a Pullup dataset, the type of SQL that [[[Undefined variable DataQuality.SQL assistant]]] generates depends on the rule type. For instance, if your rule type is Freeform SQL, the SQL generated by [[[Undefined variable DataQuality.SQL assistant]]] 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 following steps show you how to use [[[Undefined variable DataQuality.SQL assistant]]] and from where you can access it.

From Dataset Overview

1. From the Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

» The Dataset Overview modal appears.

2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] 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.

Examples

- "high is more than 50"
- "create a complex rule that is 25 lines long and returns only 1 record"
- "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**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] to automatically troubleshoot and fix the error.

1. From the Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Troubleshoot Rule**.

   » The [[[Undefined variable DataQuality.SQL assistant]]] 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**.

   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Categorical**.
4. In the query input field, enter the name of the column in your dataset to include in the SQL query.
5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:

   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Dupe**.
4. In the query input field, enter the name of the column in your dataset to include in the SQL query.
5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Record**.

4. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

> Important You must specify one column with numeric data and one date column.

5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder
   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Pattern**.
4. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

From the Rule Workbench

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 [[[Undefined variable DataQuality.SQL assistant]]] 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.

Examples

- "high is more than 50"
- "create a complex rule that is 25 lines long and returns only 1 record"
- "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**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
6. From the Prompt dropdown menu, select **Troubleshoot Rule**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] 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**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL rule 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Categorical**.
5. In the query input field, enter the name of the column in your dataset to include in the SQL query.
6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Dupe**.
5. In the query input field, enter the name of the column in your dataset to include in the SQL query.

6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Record**.
5. In the query input field, enter the names of the columns in your dataset to include in the SQL query.
6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Pattern**.
5. In the query input field, enter the names of the columns in your dataset to include in the SQL query.
6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

FAQ

- What models are used?
  The generative AI features use foundational models from Google Model Garden on Vertex AI. The 2 models that we use are PaLM 2 for Text (text-bison) and

Embeddings for text (textembedding-gecko).

- Are the models fine-tuned or retrained in any way?
  No, we do not fine-tune or retrain them in the beta implementation.
- Where, regionally, are the models located?
  The models are located in the US Central region.
- Where, regionally, are the other components located?
  The US Central region.
- How is data secured when it is sent from the proxy to the Vertex Models?
  All data is encrypted in transit using HTTPS with TLS protocols.
- What data is sent to Vertex AI?
  When a user submits a request to Collibra AI, the following info is sent to Vertex AI:
  - Generate Rule
    - Table name
    - Column names
    - User-generated plain language query description
  - Troubleshoot Rule
    - Table name
    - Column names
    - Current query
    - Contents of error message from SQL engine
- What data is stored in Vertex AI?
  The only data stored in Vertex AI is metadata regarding user queries, which does not currently include any user query or model output. This includes metadata such as total billable characters, character count, and latency metrics. Additionally, Google has made it clear that, in our current setup, no prompts are used to improve their foundational models. We do foresee a scenario where user queries might be stored in the future, but we will engage the Legal team and obtain approval prior to any changes to this.

DQ 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.

[[[Undefined variable DataQuality.Pullup Spark processing]]] processing

[[[Undefined variable DataQuality.Pullup Spark processing]]] 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. [[[Undefined variable DataQuality.Pullup Spark processing]]] 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, [[[Undefined variable DataQuality.Pullup Spark processing]]] 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 [[[Undefined variable DataQuality.Pullup Spark processing]]]

Before running [[[Undefined variable DataQuality.Pullup Spark processing]]] 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 Snowflake and Databricks features 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.

Pushdown is currently supported for the following data sources:

- Snowflake
- Databricks
- BigQuery
- Athena (beta)
- Redshift (beta)
- Trino (beta)

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 do one of the following depending on the connection:

- Enable Pushdown from the Snowflake connection template in Admin Console ▸ Connections.

  > Tip 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.

- Enable Pushdown from the Databricks connection template in Admin Console ▸ Connections.
- Enable Pushdown from the BigQuery connection template in Admin Console ▸ Connections.
- Enable Pushdown from the Athena connection template in Admin Console ▸ Connections.
- Enable Pushdown from the Redshift connection template in Admin Console ▸ Connections.
- Enable Pushdown from the Trino connection template in Admin Console ▸ Connections.

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 Collibra DQ 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'`
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 @dataset where fname = 'John'`	`select * from (select * where d_date='2023-01-01') where fname = 'John'`

Known limitations

- Pushdown is currently only available for on-premises deployments of Collibra DQ.
- You cannot currently run a job from the command line.
- Mixed data type rules are not supported in Pushdown.

- Cross-connection rules are not supported in Pushdown. However, support for this capability when connections belong to the same Snowflake account is planned for an upcoming version.
- 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 a 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.
✓ 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**.

Available features

The following table contains an overview of the data quality features currently available for Pushdown for Databricks.

Feature	Available
Profile	✓ Yes
Rules	✓ Yes
AdaptiveRules	✓ Yes
Shapes	✓ Yes
Outliers	✓ Yes
Patterns	✗ No
Dupes	✓ Yes
Schema	✓ Yes
Archive Break Records	✓ Yes

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 the Snowflake setup script
- Setting up the Snowflake Connection template

Running the Snowflake setup script

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.

Step	Details
1	Open a terminal session.

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.
	Warning Do not update the variables for Collibra DQ.

3	Run as an admin user. `USE ROLE ACCOUNTADMIN;`

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);`

Step	Details
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;`
6	Assign privileges to the Collibra DQ warehouse. `GRANT OPERATE, USAGE, MONITOR ON WAREHOUSE identifier($dq_warehouse_name) TO ROLE identifier($dq_role_name);`
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);`

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);`

Warning Please ensure the SQL variables are updated in the above script before proceeding.

Setting up the Snowflake Connection template

The following steps show you how to set up the Snowflake connection template with Pushdown enabled.

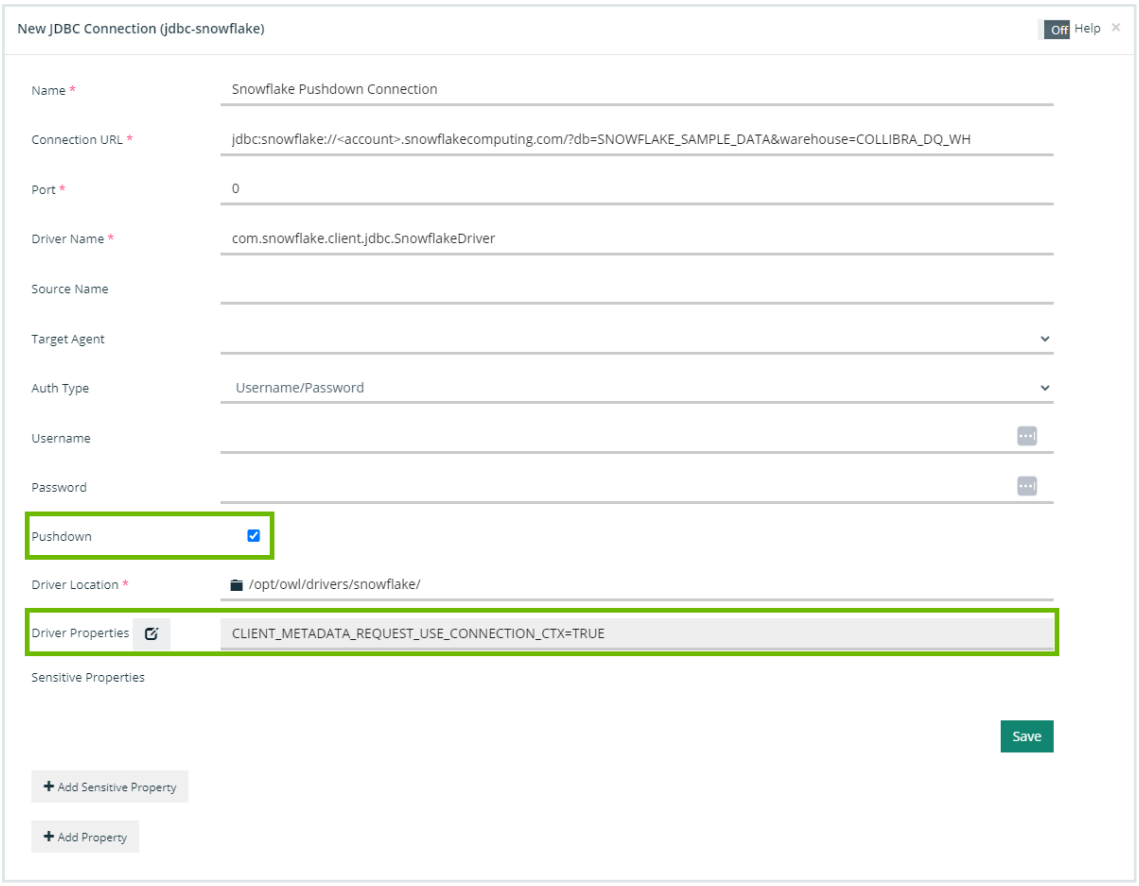
1. On the main menu, click ⚙, and then click **Connections**.
2. Click the Databricks tile, then enter the required information.

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 turned on and additional Driver Properties.



3. Click **Save**.

Available features

The following table contains an overview of the data quality features currently available for Pushdown for Snowflake.

Feature	Available
Profile	✓ Yes
Rules	✓ Yes
AdaptiveRules	✓ Yes

Feature	Available
Shapes	✓ Yes
Outliers	✓ Yes
Patterns	✕ No
Dupes	✓ Yes
Schema	✓ Yes
Archive Break Records	✓ Yes

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.

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.
✗ 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.

Required	Property	Type	Description
✓ 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. > Note As of Collibra DQ version 2023.08, only rule break records are eligible for source archival.
✓ 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**.

Available features

The following table contains an overview of the data quality features currently available for Pushdown for BigQuery.

Feature	Available	Notes
Profile	✓ Yes	
Rules	✓ Yes	
AdaptiveRules	✓ Yes	
Shapes	✓ Yes	
Outliers	✕ No	
Patterns	✕ No	
Dupes	✕ No	
Schema	✕ No	
Archive Break Records	✓ Yes	Currently, only break records from Rules can be archived.

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.

> Note Pushdown for Athena is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.

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.
✓ 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**.

Available features

The following table contains an overview of the data quality features currently available for Pushdown for Athena.

Feature	Available
Profile	✓ Yes
Rules	✓ Yes
AdaptiveRules	✓ Yes
Shapes	✕ No
Outliers	✕ No

Feature	Available
Patterns	✕ No
Dupes	✕ No
Schema	✓ Yes
Archive Break Records	✕ No

What's next?

You can now create a Pushdown job on your Athena connection.

Known limitations

- As of the Collibra DQ 2023.09, only AdaptiveRules are available from the Add Layers step.
- Archive Break Records is not currently supported.

This section shows you how to set up Pushdown for a Redshift connection.

> Note Pushdown for Redshift is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.

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**.

Available features

The following table contains an overview of the data quality features currently available for Pushdown for Redshift.

Feature	Available
Profile	✓ Yes
Rules	✓ Yes
AdaptiveRules	✓ Yes
Shapes	✕ No
Outliers	✕ No
Patterns	✕ No
Dupes	✕ No
Schema	✓ Yes
Archive Break Records	✓ Yes

What's next?

You can now create a Pushdown job on your Redshift connection.

Known limitations

- As of the Collibra DQ 2023.09, only AdaptiveRules are available from the Add Layers step.

This section shows you how to set up Pushdown for a Trino connection.

> Note Pushdown for Trino is currently available as a public beta. Since this is a beta feature, some capabilities may be limited. For more information on our beta program, refer to Public betas.

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.
✓ 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.

Required	Property	Type	Description
✓ 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. > Note As of Collibra DQ version 2023.10, only rule break records are eligible for source archival.
✓ 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**.

Available features

The following table contains an overview of the data quality features currently available for Pushdown for Trino.

Feature	Available
Profile	✓ Yes
Rules	✗ No
AdaptiveRules	✓ Yes
Shapes	✗ No
Outliers	✗ No
Patterns	✗ No

Feature	Available
Dupes	✕ No
Schema	✓ Yes
Archive Break Records	✕ No

What's next?

You can now create a Pushdown job on your Trino connection.

Known limitations

- As of the Collibra DQ 2023.10, only AdaptiveRules are available from the Add Layers step.
- Archive Break Records is not currently supported.

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.
3	Select your columns	Select the columns to include in your DQ job. Optionally assign Key columns and Link IDs. > 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 transforms	Modify or create new columns to meet specific formatting requirements at the Spark level.
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.

Step	Action	Description
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.

Step	Action	Description
3	Select your columns	Select the columns to include in your DQ job. Optionally assign Key columns and Link IDs. > **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 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.

Step	Action	Description
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]

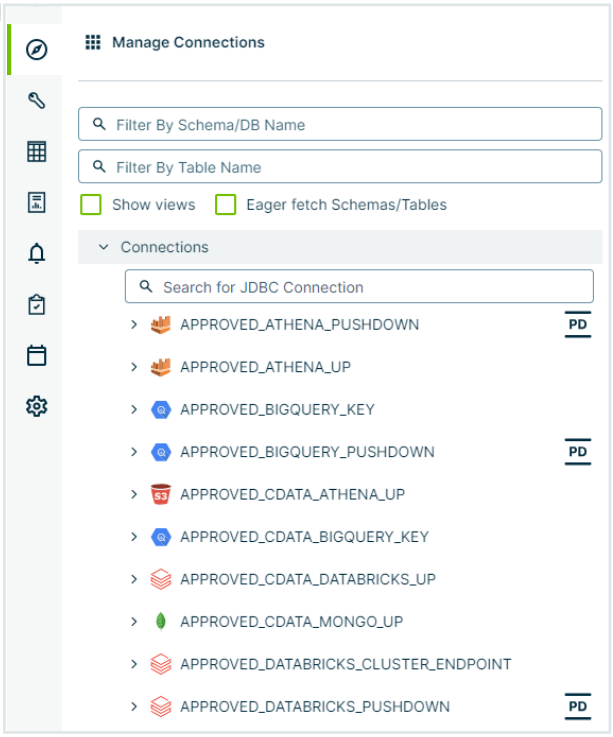
What's next?

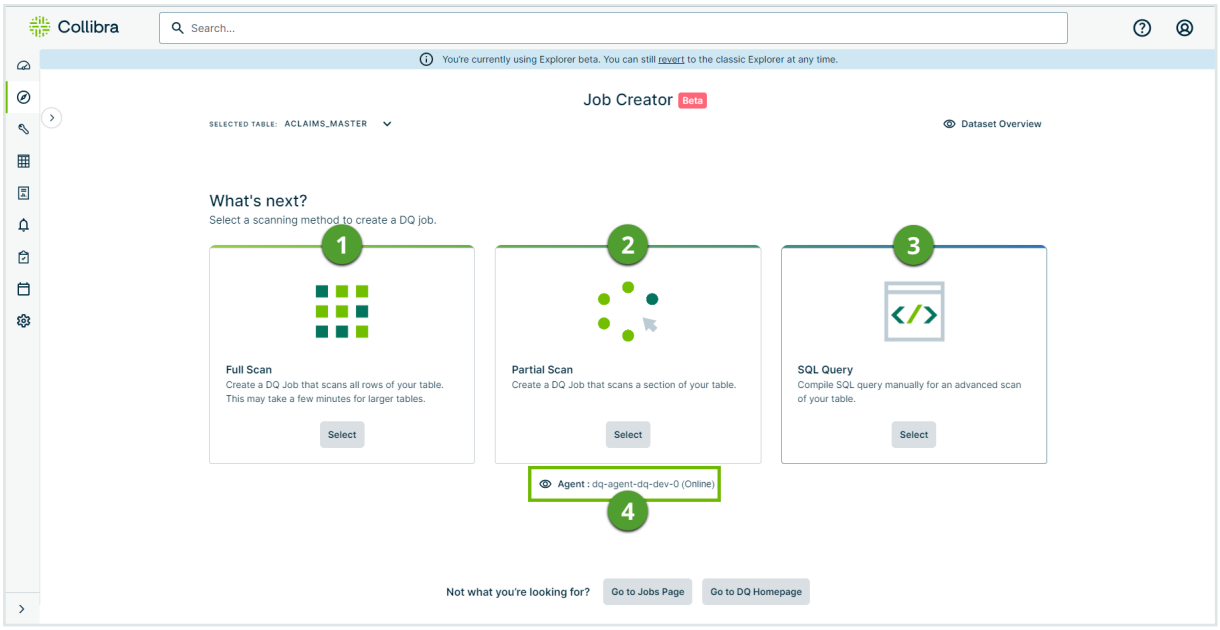
Select a scanning method to get started.

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.

1

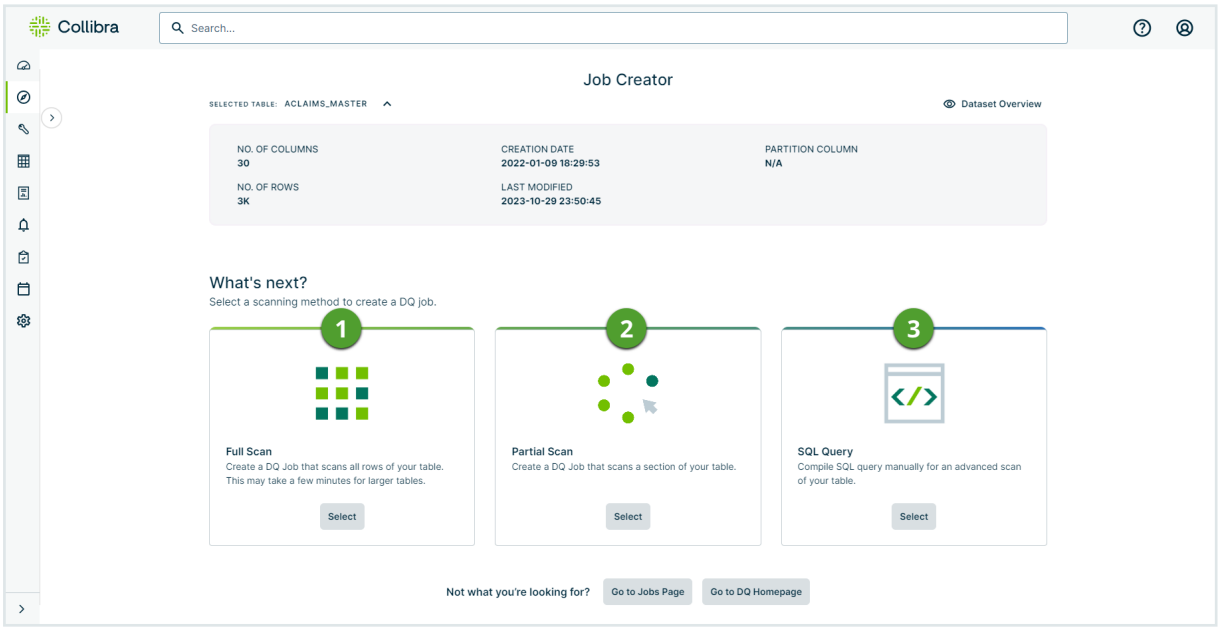




The following table shows the available scanning methods. Each option has a number that corresponds with the number in the image above.

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 Collibra Data Quality & Observability 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.
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 Collibra Data Quality & Observability 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	Description
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 given column. This column is autofilled and cannot be edited or selected. Collibra DQ supports all standard data types. Collibra DQ also supports custom data types, but this may vary depending on the datasource, for example, Snowflake.
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.

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 As of Collibra DQ 2023.04 version, you can only apply one Row Filter from the UI. However, the ability to apply multiple Row Filters from the UI is planned for a future release. To add an additional filter, you can 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. You can click the three dots menu in the upper right corner of the Filter Type window to switch to SQL View. Click the three dots again to switch back to Standard View.

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.

Option	Description
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.
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 As of the April 2023 release version, you can only apply one Row Filter from the UI. However, the ability to apply multiple Row Filters from the UI is planned for a future release. To add an additional filter, you can 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

You can click the three dots menu in the upper right corner of the Filter Type window to switch to SQL View. SQL View allows advanced SQL users to write and compile SQL queries manually. 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. If you are using Pushdown mode, proceed to the next available step.

Mapping

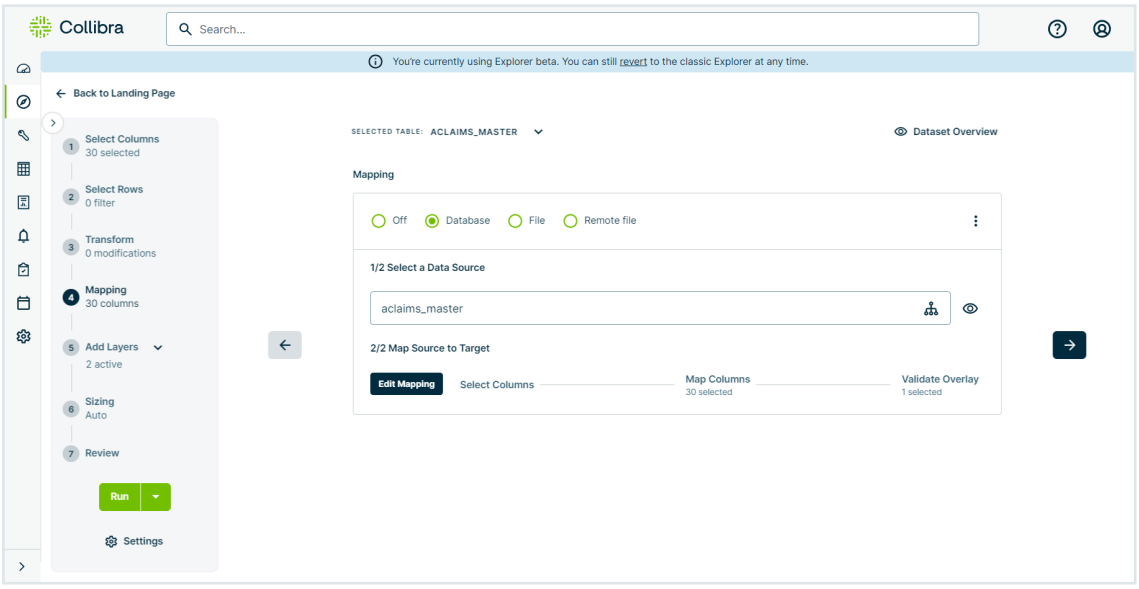
Mapping allows you to discover inconsistencies between the columns from one database, file, or remote file and those of another. Mapping checks for inconsistencies such as row count, schema, and value differences in data copied from data sources to target tables. By

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 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 **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.

e. 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.
f. Select the columns to include in the mapping using the checkbox options next to each column name.

   > Note All columns are selected by default.

g. 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 databases.

h. 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.
i. 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.

j. Optionally select the **Key** checkbox at the top of any of the columns to assign them as key columns.
k. Click **Validate**.
   » The main Mapping screen displays the number of mapped columns in the stepper.

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
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

Option	Description	.csv	.txt	.orc	parquet	avro	json	delta	xml	hudi
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
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 drop-down 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.

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.

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.
Check case	Checks for case sensitivity when comparing source and target schemas.
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. This requires a large amount of computational effort.
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.

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 depending on whether you are using Pullup or Pushdown mode.

About AdaptiveRules

AdaptiveRules are rules that use machine learning to automatically observe and adapt to changes in numeric representations of data over time and down-score any values outside defined boundaries.

Configuring AdaptiveRules

The following table shows details about Collibra DQ's AdaptiveRules and how to configure them.

AdaptiveRule	AdaptiveRule type	Description	Default
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

AdaptiveRule	AdaptiveRule type	Description	Default
	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.	
	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

AdaptiveRule	AdaptiveRule type	Description	Default
	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			
	Data Type Check (Integer, String, Date)	Monitors columns that shift from one data type to another.	✕ No
	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
< 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		
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

Option	Description	Default Value
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
Data Shape Granular		

Option	Description	Default Value
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.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Column</td><td>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.</td></tr>
<tr><td>Type</td><td>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:

<table><tr><th>Data type</th><th>Description</th></tr><tr><td>Date</td><td>A date.</td></tr><tr><td>DOUBLE</td><td>A normal-size floating point number.</td></tr><tr><td>VARCHAR</td><td>A VARIABLE length string that can contain letters, numbers, and special characters.</td></tr><tr><td>text</td><td>A variable width character string.</td></tr><tr><td>numeric</td><td>Fixed precision and scale numbers.</td></tr></table></td></tr>
</table>

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 the three dots menu icon 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	
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.
Unit	The unit of outlier detection sensitivity. Select an option from the dropdown menu.

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.

Option	Description
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.
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 the three dots menu icon 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
	The column(s) used to detect dupes. You can select multiple columns for dupes detection. Required. 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.

Option	Description
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.
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**.

About AdaptiveRules

AdaptiveRules are rules that use machine learning to automatically observe and adapt to changes in numeric representations of data over time and down-score any values outside defined boundaries.

Configuring AdaptiveRules

The following table shows details about Collibra DQ's AdaptiveRules and how to configure them.

AdaptiveRule	AdaptiveRule type	Description	Default
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.	

AdaptiveRule	AdaptiveRule type	Description	Default
	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

AdaptiveRule	AdaptiveRule type	Description	Default
	Max	Monitor columns with max values outside the normal range. Select this option to include max value observations in your DQ check.	✕ No
Schema			
	Data Type Check (Integer, String, Date)	Monitors columns that shift from one data type to another.	✕ No
	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
< 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		
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

Option	Description	Default Value
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
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.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Column</td><td>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.</td></tr>
<tr><td>Type</td><td>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:

<table><tr><th>Data type</th><th>Description</th></tr><tr><td>Date</td><td>A date.</td></tr><tr><td>DOUBLE</td><td>A normal-size floating point number.</td></tr><tr><td>VARCHAR</td><td>A VARIABLE length string that can contain letters, numbers, and special characters.</td></tr><tr><td>text</td><td>A variable width character string.</td></tr><tr><td>numeric</td><td>Fixed precision and scale numbers.</td></tr></table></td></tr>
</table>

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 the three dots menu icon 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	
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.
Unit	The unit of outlier detection sensitivity. Select an option from the dropdown menu.

> Warning Patterns are not yet available for Pushdown. Patterns will be available in an upcoming Collibra DQ release.

About Patterns

Patterns detect similarities among string values across columns and downscore any observations.

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
	The column(s) used to detect dupes. You can select multiple columns for dupes detection. Required. 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.

Option	Description
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.
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

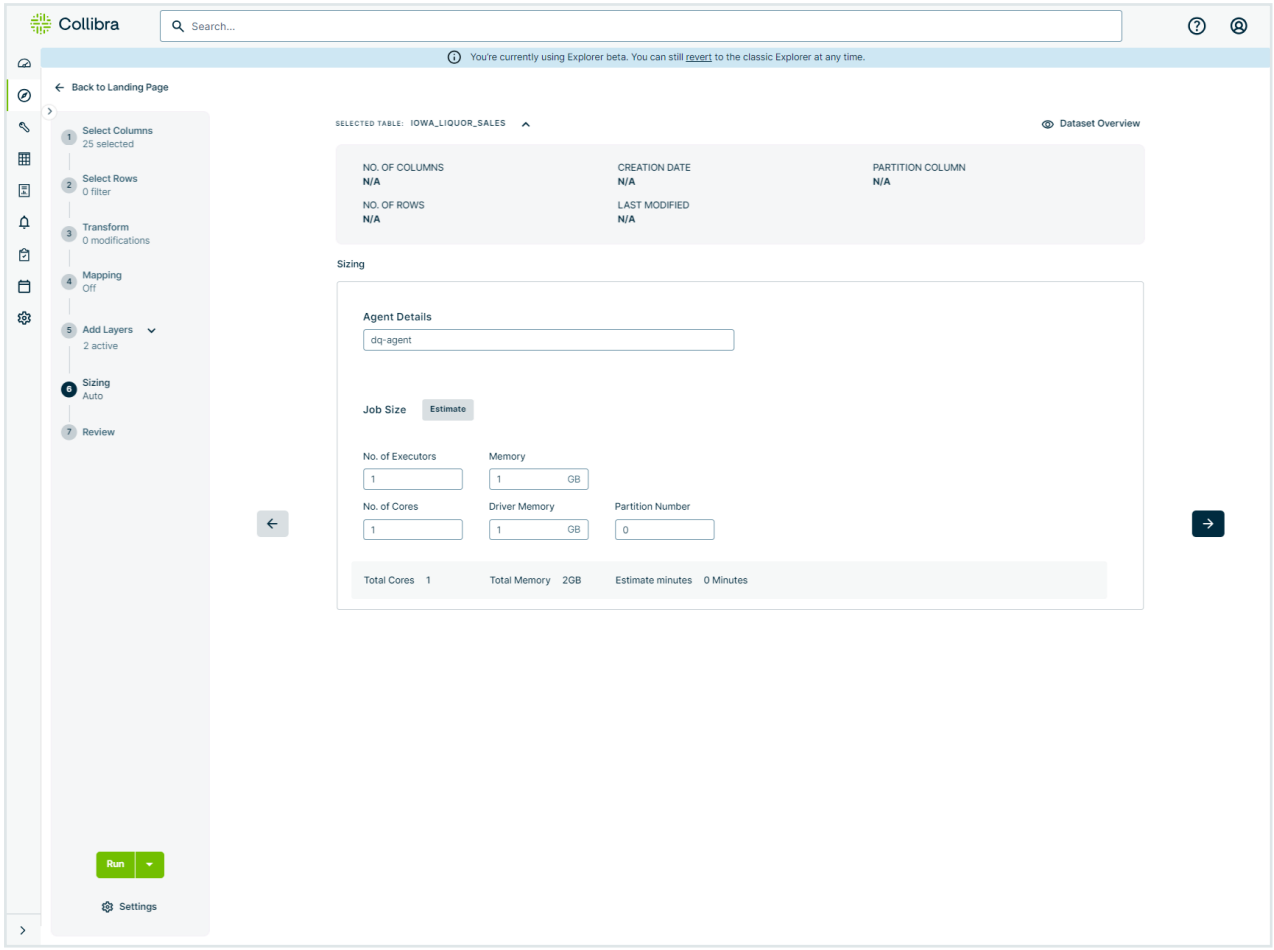
your job. These resources include:

- 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.

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

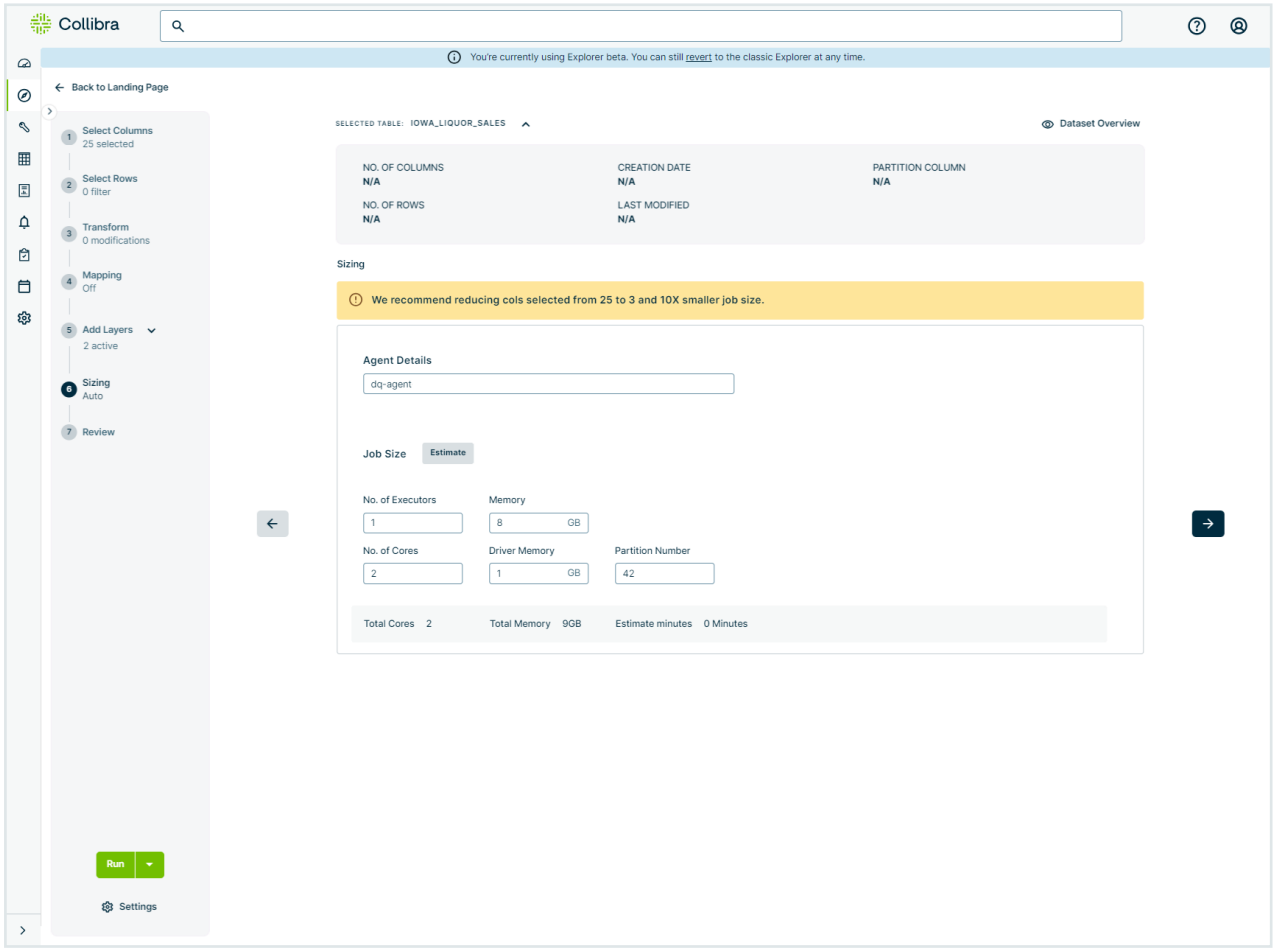
Spark Resource	Description	Default Value
Partition Number	The number of chunks of the dataset to split evenly across the Spark executors.	0

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.

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

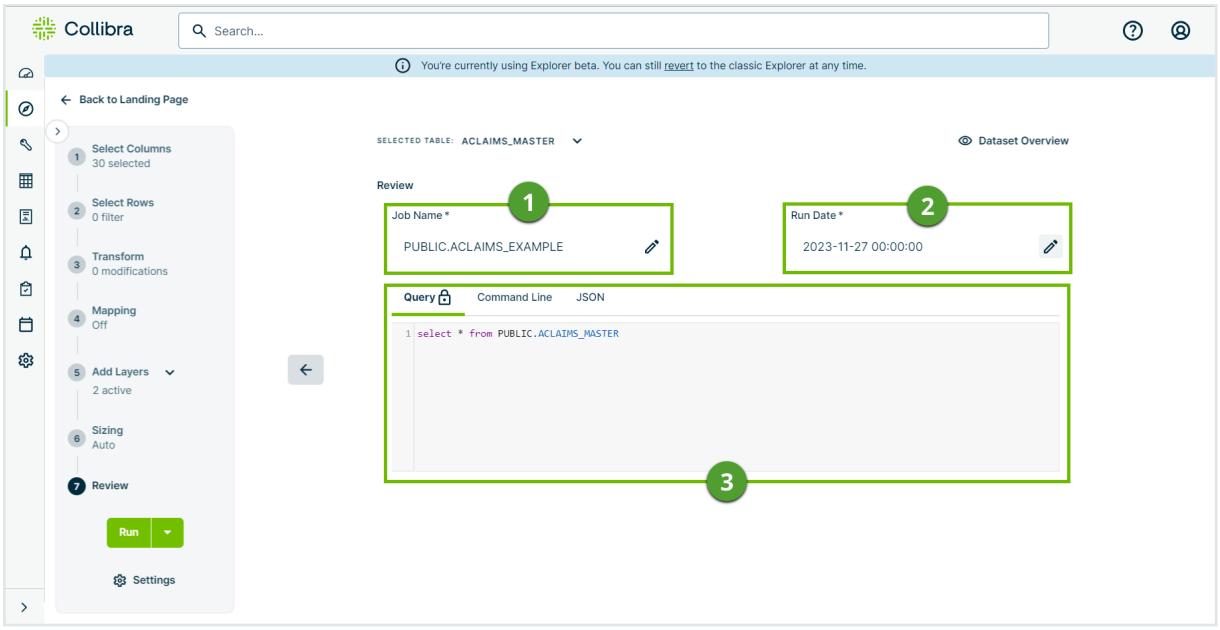
Memory	`-executormemory`	To determine executor memory, you first need to determine your dataset size. To find the size of your dataset, calculate the following using inputs from your environment:
		`(number of rows * number of columns * 10) / 1024^3`
		Next, determine the upper bound of maxworkermemory 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.

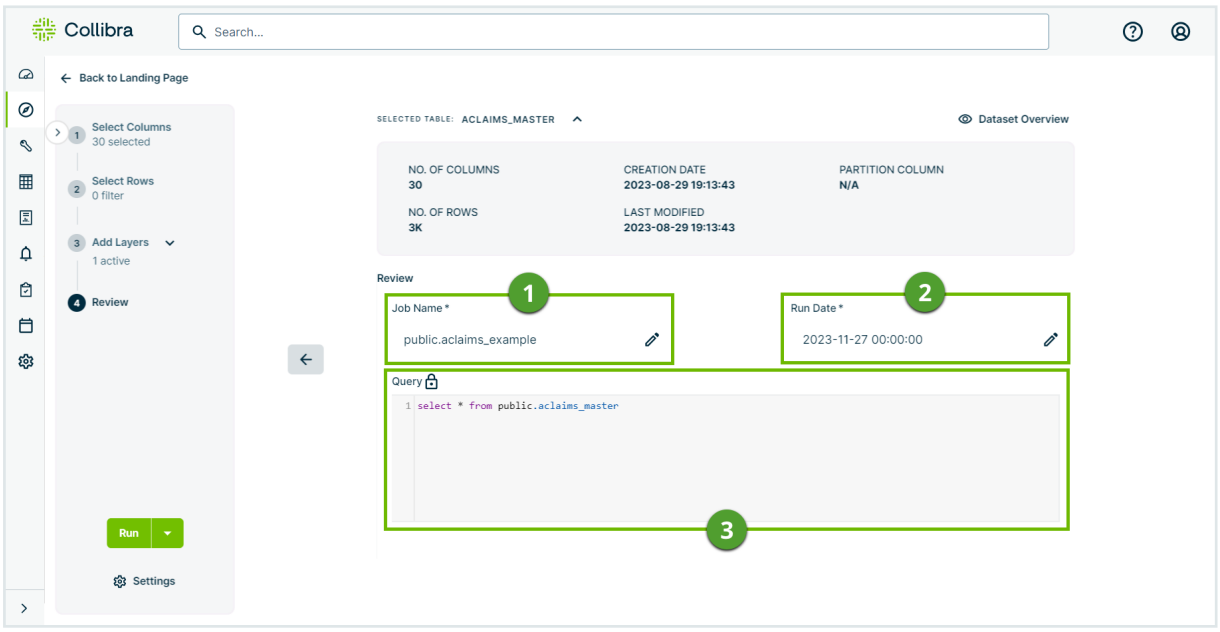
Review

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 [[[Undefined variable DataQuality.Pullup Spark processing]]] mode. Select from one of the options below depending on whether you are using Pullup or Pushdown mode.



No.	Option	Description
1	Job Name	The name of your DQ job. Click the ✎ to update the name. Click the ✔ to save your changes.
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.

No.	Option	Description
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. **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
1	Job Name	The name of your DQ job. Click the ✎ to update the name. Click the ✔ to save your changes.
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.

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 time that will replay from a specified number. 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 [[[Undefined variable DataQuality.Pullup Spark processing]]] 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.
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 default value is 10.

Option	Description
Learning Phase	The minimum number of DQ Job runs required for behavioral scoring begins to calculate. The default value is 4.
Parallel JDBC	
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.

Option	Description
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.
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 default value is 10.
Learning Phase	The minimum number of DQ job runs required for behavioral scoring begins to calculate. The default value is 4.
Archive Break Records	

Option	Description
Data Preview from Source	Prevents data preview records from storing on the PostgreSQL [[[Undefined variable DataQuality.DQ Metastore]]]. When you select this option: • All data preview records are removed from the PostgreSQL [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ Metastore]]].
Archive Dupes Break Records	Allows the storage of dupe break records to the source system instead of the PostgreSQL [[[Undefined variable DataQuality.DQ Metastore]]].
Archive Outliers Break Records	Allows the storage of outlier break records to the source system instead of the PostgreSQL [[[Undefined variable DataQuality.DQ Metastore]]].

Option	Description
Archive Rules Break Records	Allows the storage of rule break records to the source system instead of the PostgreSQL [[[Undefined variable DataQuality.DQ Metastore]]].
Archive Shapes Break Records	Allows the storage of shapes break records to the source system instead of the PostgreSQL [[[Undefined variable DataQuality.DQ 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	
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.

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. The default value is 2.
Run Date	
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 [[[Undefined variable DataQuality.Pullup Spark processing]]] 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.

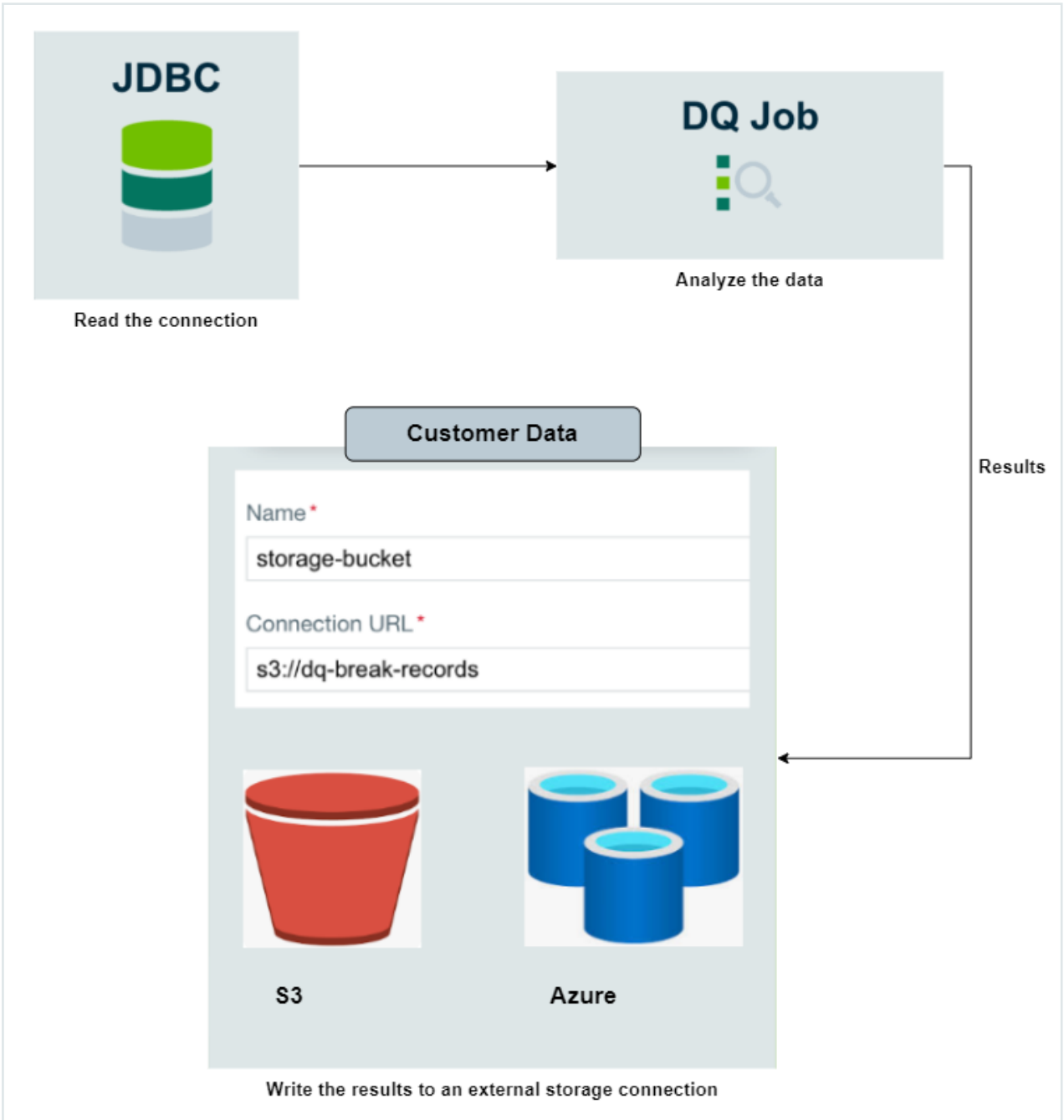
Archiving break records from Pullup jobs

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



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 Collibra Data Quality & Observability.

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

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.
Azure Blob	• An Azure Blob connection. • Read and write access on your Azure Blob 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/*"
            ]
        }
    ]
}
```

Steps

1. From Explorer, connect to a data source.
2. In the Scope Select, click **Add Link Back to Source** and assign a **Link ID** to a column.
3. Finish setting up your DQ job, then click **Save/Run**.
4. On the Config tab, click **DQ Job**.
5. Select the **Archive Breaks** option.
6. Select an external storage output option from the **Archive Connection** dropdown menu.
7. Click **Estimate Job**.
8. Click **Run**.
9. When a record breaks, its metadata exports automatically to your external storage service.

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 [[[Undefined variable DataQuality.Pushdown processing]]] jobs

Collibra Data Quality & Observability [[[Undefined variable DataQuality.Pushdown processing]]] uses SQL queries native to the data source in order to connect directly to the

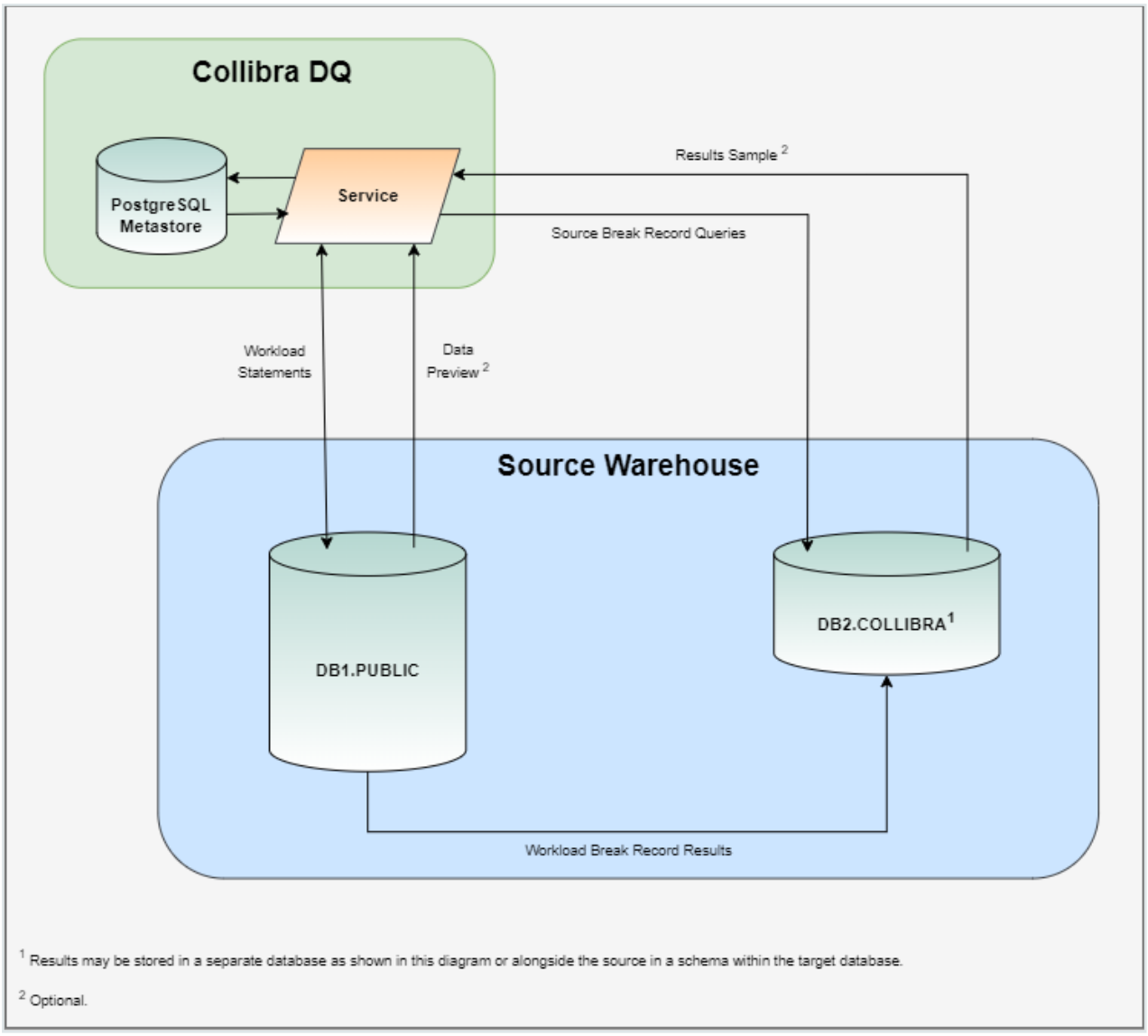
source system and submit the data quality workload. This gives Collibra DQ the unique opportunity to store break records that are typically written to the PostgreSQL [[[Undefined variable DataQuality.DQ Metastore]]] in a database or schema specifically allocated for Collibra DQ. By archiving break records 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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ Metastore]]] to power the DQ web application. To prevent break record storage in the [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.DQ 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 [[[Undefined variable DataQuality.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 [[[Undefined variable DataQuality.Pushdown processing]]] connections

The following [[[Undefined variable DataQuality.Pushdown processing]]] connections support the archival of break record:

- Snowflake
- Databricks
- BigQuery

Prerequisites

- You have an active [[[Undefined variable DataQuality.Pushdown processing]]] connection.
- The service account user has write access to archive break records to the database or schema.

  > Important
  > Data sources may have different combination 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.

Steps

1. Click the ⚙ icon and then click **Connections**.
   » The Connections Management page opens.
2. Select a Pushdown compatible data source.
   » The Add/Edit Connection modal opens.
3. Click Existing Connections at the bottom of the connection template of your data source, then select your connection.
4. Select the **Archive Break Records** option.
   » The Archive Break Records Location appears below it.

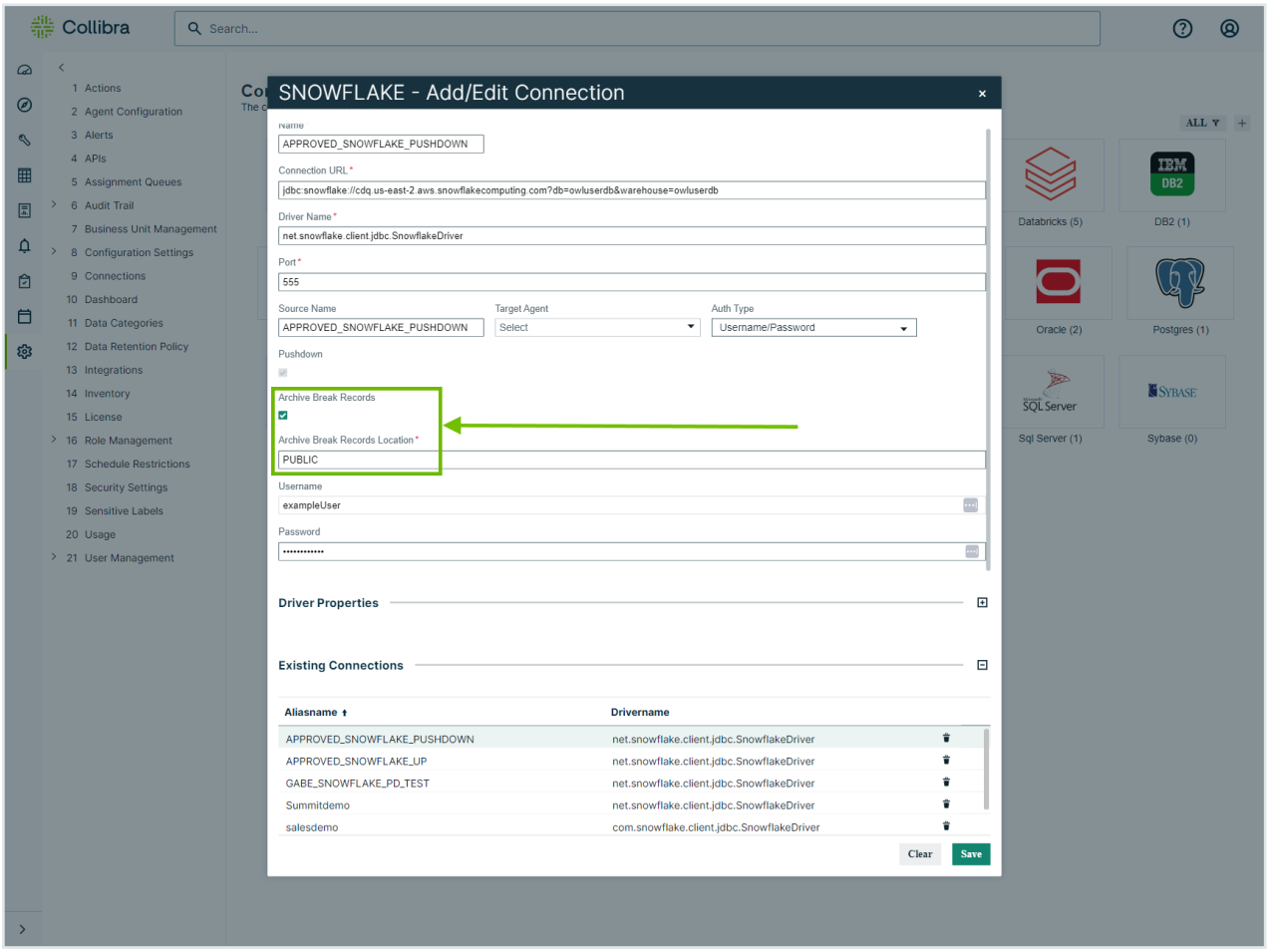
5. In the **Archive Break Records Location** input field, specify a schema output location in your data source for break records to send.

> Note If you enter an incorrect schema, Collibra DQ automatically uses the default output location, PUBLIC, instead.

> Tip
> You can enter database.schema when your schema resides outside of the current database. For example, db1.PUBLIC, db2.marketing, and db3.sales.
>
> Because it is optional to specify a database, if you only specify the schema, the database defaults to your current database.

Example screenshot...[1]

1

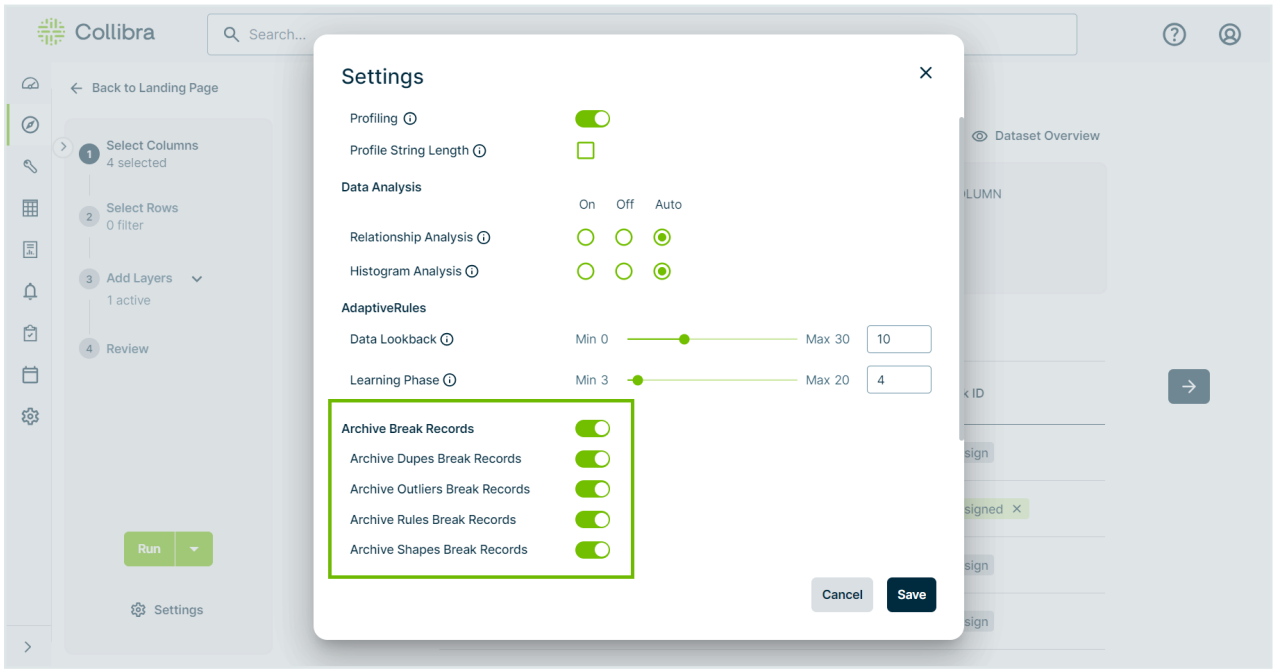


6. Click **Save**.
7. Open Explorer and connect to your [[[Undefined variable DataQuality.Pushdown processing]]] data source.
8. Select a scanning method.
9. In the Select Columns step, assign a **Link ID** to a column.

   > 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.

10. In the lower left corner, click **Settings**.
11. Select the **Archive Break Records** option.
    » By default, turning on Archive Break Records enables the export of break record for all available layers.
    Example screenshot...[1]
12. Click **Save**.
13. Set up and run your DQ job.
14. 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 layers of [[[Undefined variable DataQuality.Pushdown processing]]] jobs:

- Rules
- Outliers
- Shapes
- Duplicates

The records for each of these layers contains 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.

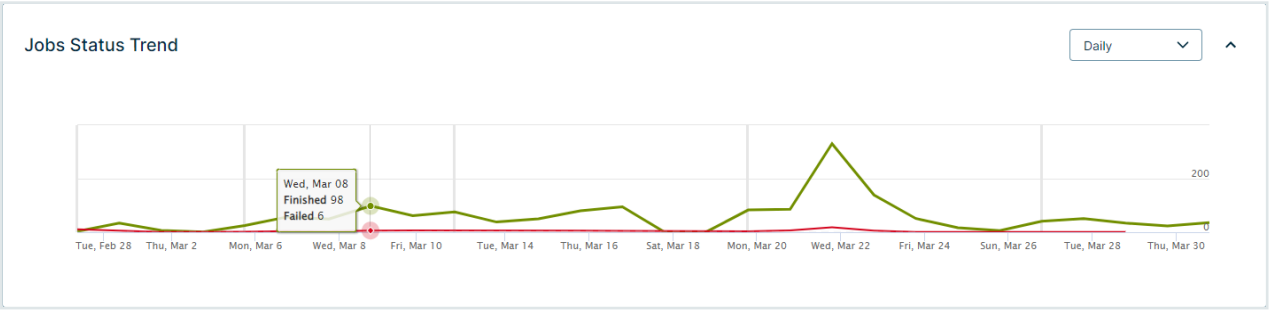
Unlike in [[[Undefined variable DataQuality.Pullup Spark processing]]] mode, when Collibra DQ detects a duplicate value in a [[[Undefined variable DataQuality.Pushdown processing]]] job with Link IDs assigned, all columns included in the source query of the row where the break is detected display on the Findings page. In [[[Undefined variable DataQuality.Pullup Spark processing]]] mode, only Link ID columns display on the Findings page when a duplicate value is detected, because the data_preview [[[Undefined variable DataQuality.DQ Metastore]]] table only stores Link ID columns.

Managing the Jobs page

The Jobs page provides details about all of the DQ Jobs with attempts to run. When a 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 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 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.

Column	Description
Data Set	The name of your data set.
Status	The completion status of your DQ Job. Use the Job Status filter to filter by statuses as they appear in this column.
Activity	The task that the system performs while it processes the Job.
Start Time	The timestamp when the Job starts to run. The format is YYYY-MM-DD HH:MM:SS.
Update Ts	The timestamp of the last processing activity. The format is YYYY-MM-DD HH:MM:SS.
Job Exception	When an error or exception occurs, a helpful description of the event displays in this column.
Username	The username of the person in your organization who ran the Job.
Actions	The action you can take to troubleshoot or terminate your Job.

> 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 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

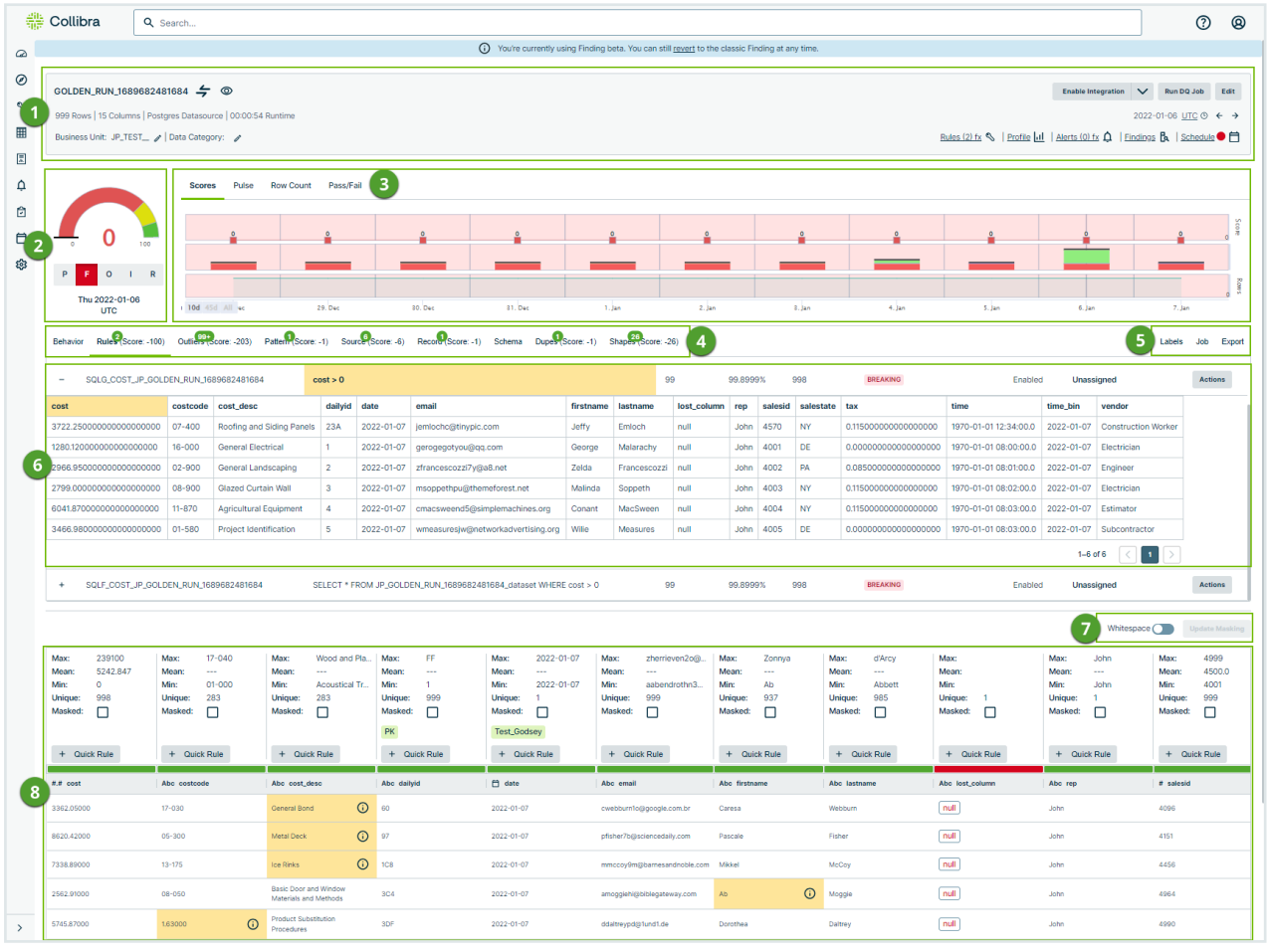
Findings

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.

<table>
<tr><th>No.</th><th>Component</th><th>Description</th></tr>
<tr><td>2</td><td>Data quality score</td><td>
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.

<blockquote>Note The findings from each data quality dimension where Collibra DQ detects potential data quality issues contribute to the overall data quality score.</blockquote>
Below the score meter are five clickable options.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>P</td><td>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.</td></tr>
<tr><td>F</td><td>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.</td></tr>
<tr><td>O</td><td>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.</td></tr>
<tr><td>I</td><td>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.</td></tr>
<tr><td>R</td><td>Remove (R) indicates a run that triggered by mistake and should be removed.</td></tr>
<tr><td>Timestamp</td><td>The run date of the current job. You can click this option to set the runId to a specific timezone.</td></tr>
</table>
</td></tr>
</table>

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.
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.

No.	Component	Description
5	Labels, Job, and Export tabs	These three tabs let you do the following: • Apply labels and annotations. • View and edit the run command line of the current and previous job. • Export individual or all data quality findings as .xlsx files.
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.

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.

Exploring data quality findings

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 it detects the issue. Drill down into the various data quality dimension tabs to better understand the quality and reliability of your dataset.

This section shows the available dimensions and what they mean when you drill into a specific finding.

Behaviors

Collibra DQ learns from column-level profiling to create AdaptiveRules, which contribute to the overall Behavior score. AdaptiveRules are rules that automatically observe and adapt to changes in numeric representations of data over time and down-score any values outside defined boundaries.

Behavioral anomalies most commonly appear when you use Replay or behavior lookback. When you use Replay, Collibra DQ learns the behavior of a dataset over a period of time.

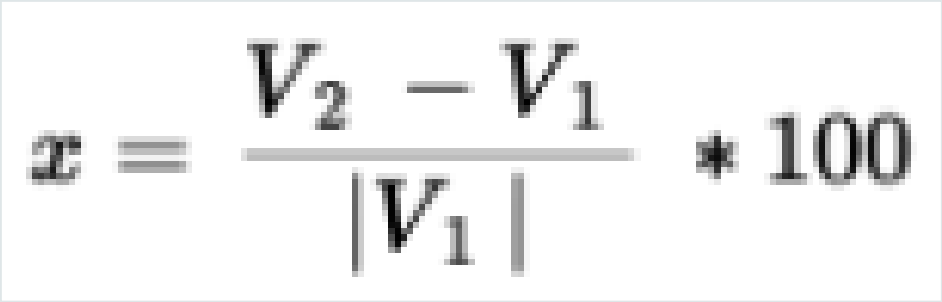
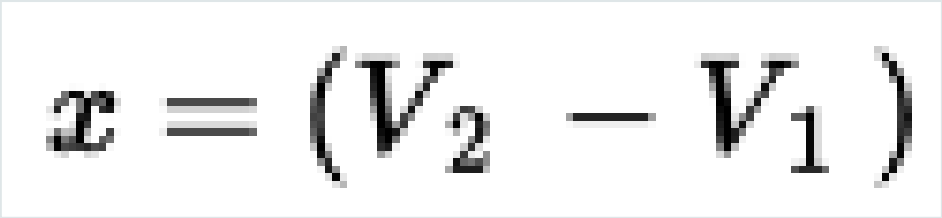
The following table describes the information available on the Behaviors tab of the Findings page.

Column	Description
Blind Spot	The column where a behavioral anomaly is detected.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Type</td><td>The type of AdaptiveRule that the behavioral model detects from the profiling activity on a given column.

The following table shows the possible AdaptiveRules:

<table>
<tr><th>AdaptiveRule</th><th>Description</th></tr>
<tr><td>Row Count</td><td>Row count change in your table.</td></tr>
<tr><td>Loading Time</td><td>Loading time changes.</td></tr>
<tr><td>Uniqueness</td><td>Cardinality changes to a column within the range of previous DQ Jobs.</td></tr>
<tr><td>Null Values</td><td>Null values detected in a column.</td></tr>
<tr><td>Empty Values</td><td>Empty data detected in a column.</td></tr>
<tr><td>Min</td><td>Columns with min values outside the normal range.</td></tr>
<tr><td>Mean</td><td>Columns with mean values outside the normal range.</td></tr>
<tr><td>Max</td><td>Columns with max values outside the normal range.</td></tr>
<tr><td>Data Type Check (Integer, String, Date)</td><td>Columns that shift from one data type to another.</td></tr>
<tr><td>Schema Change (Add, Alter, Delete)</td><td>Schema evolution changes, such as columns that are added or dropped from a data set.</td></tr>
</table>
</td></tr>
<tr><td>Baseline</td><td>The mean of the preceding number of scans determined by the Data Lookback value on the Settings modal in Explorer.</td></tr>
<tr><td>Today</td><td>The value of the behavioral observation on the day that Collibra DQ detects it.</td></tr>
</table>

Column	Description
% Change	The percent change from the value of one row to another. 
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.  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.
Action	The Item Labeling 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 your data quality item, for example, Observation.

Column	Description
Profile	N/A
Details	Click to open the Change Detection modal.

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 Rule Check Details of a DQ Job.

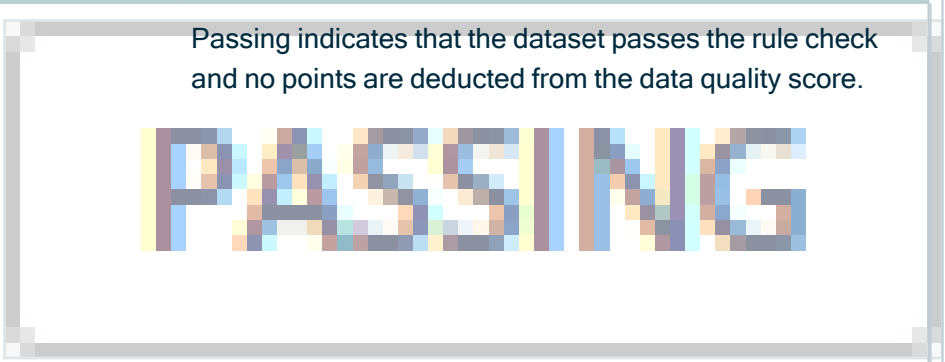
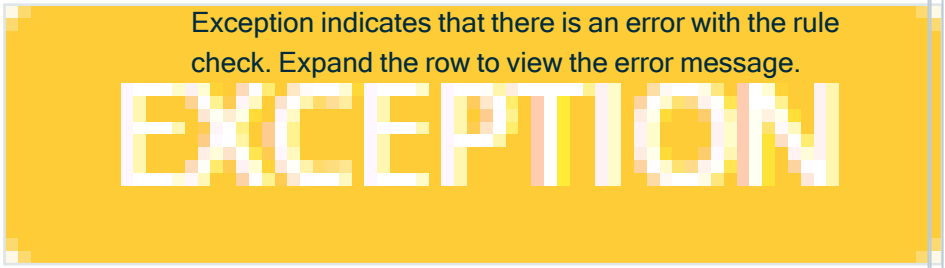
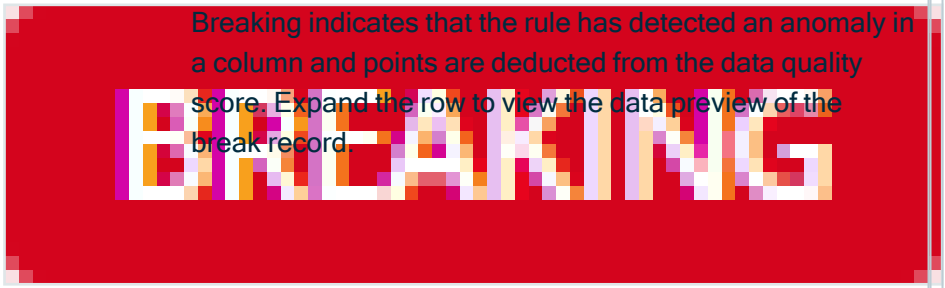
Collibra DQ profiles the data and builds a model for each dataset it scans. This allows Collibra DQ to learn what *normal* means within the context of each dataset. As the data changes, the definition of normal also changes. Instead of requiring you to adjust rule settings, Collibra DQ continues to adjust 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.

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 following table describes the information available on the Rules tab of the Findings page.

Column	Description
Rule Name	The name of your rule.
Condition	The rule condition that is defined in the rule. An alert generates when the condition is met.
Points	The number of points to deduct from the DQ Score for a given rule break.
%	The percentage of your overall DQ Score that are deducted because of a Breaking rule.
Records	The number of rows with Breaking records. If the status of the rule is Exception or Passing, then the value should be 0.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Status</td><td>The status of the rule. The following table shows the possible statuses:
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td> 
</td><td>Passing indicates that the dataset passes the rule check and no points are deducted from the data quality score.</td></tr>
<tr><td> 
</td><td>Exception indicates that there is an error with the rule check. Expand the row to view the error message.</td></tr>
<tr><td> 
</td><td>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.</td></tr>
</table></td></tr>
<tr><td>Dimension</td><td>The DQ Dimension that is identified in the rule finding, for example, Completeness.</td></tr>
<tr><td>isActive</td><td>Shows if a rule is active. When the value is 1, the rule is active. When the value is 0, the rule is inactive.</td></tr>
<tr><td>Status</td><td>The status of your data quality item, for example, Observation.</td></tr>
</table>

Column	Description
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	**Rule 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 [[[Undefined variable DataQuality.Pullup Spark processing]]] mode, you can download a CSV file containing details of the break records. • In [[[Undefined variable DataQuality.Pushdown processing]]] mode, you can download either a CSV or JSON file containing details of the break records.

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** and 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 [[[Undefined variable DataQuality.DQ Metastore]]]

When a rule returns breaking records, the following query inserts unique records into the PostgreSQL [[[Undefined variable DataQuality.DQ 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.

Outliers

The Outliers activity detects 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.

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.
Value	The value of the column of the detected outlier.
Count	The number of potential outliers in the column.
Predicted	The value that Collibra DQ predicts for a given run. This value is based on the observed values of previous runs.
Conf	The confidence score. A low confidence score starts at 0.

Column	Description
Status	The status of your data quality item, for example, Observation. Status also 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 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.
Link ID	Links back to the detected record for remediation. > Note Link ID is not available for categorical outliers.
Action	In [[[Undefined variable DataQuality.Pushdown processing]]] mode, you can download either a CSV or JSON file containing details of the break records.

> Note When you assign a Date and Key column in an Outlier configuration, Collibra DQ may also discover Record finding.

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** 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.

Patterns

Patterns detect similarities among string values across columns and down-score any uncommon observations.

The following table describes the information available on the Patterns tab of the Findings page.

Column	Description
V	A data preview of the actual value and the predicted value. Click V 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.

Column	Description
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.
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 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

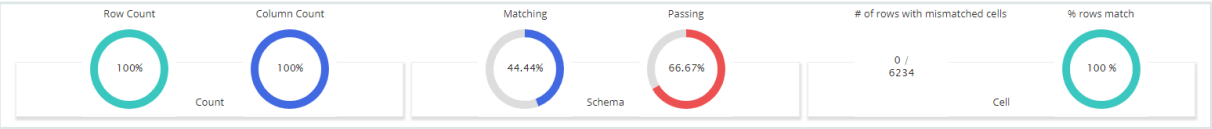
There are two options to export the details of your rule break records as .xlsx files:

- **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

Mapping 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.

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

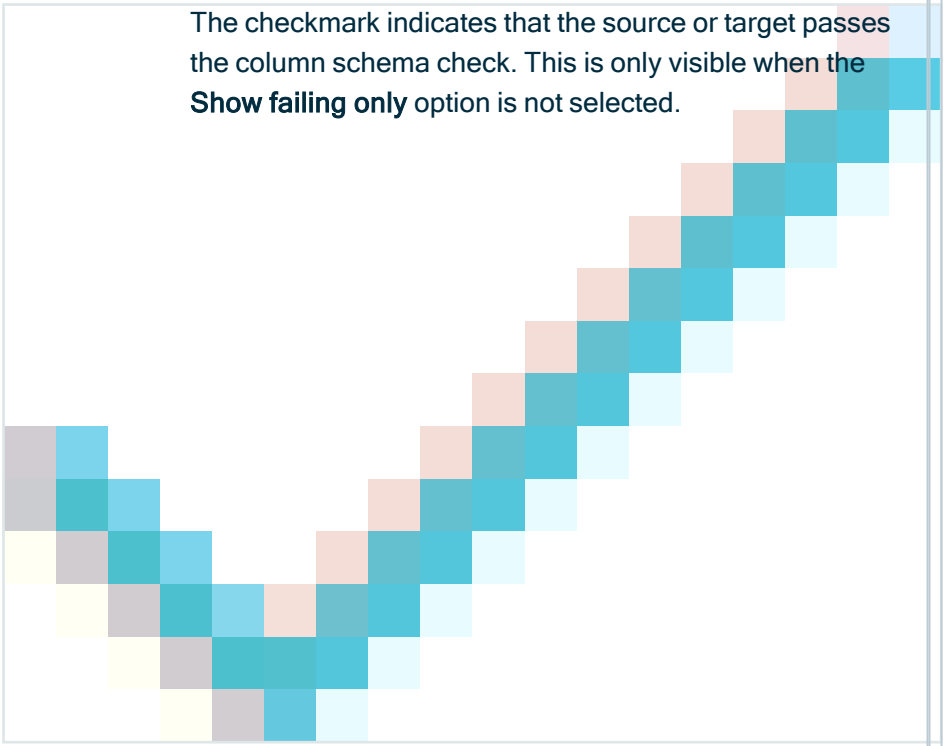
Column	Description
Type	The type of count issue, for example, Dataset Column Count and Dataset Row Count.

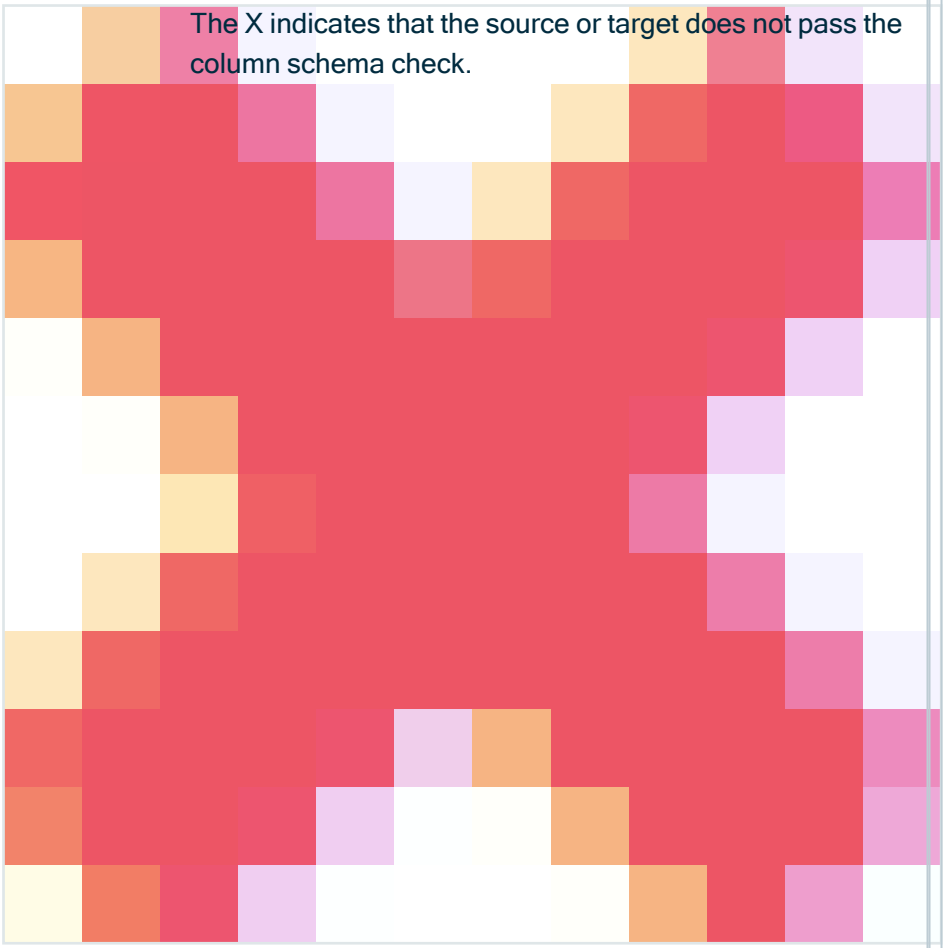
Column	Description
Target Count	The number of columns or rows that Collibra DQ counts in the target dataset.
Source Count	The number of columns or rows that Collibra DQ counts in the source dataset.
Change (%)	The percentage count difference between the target and source datasets.
Description	The description 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.
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.

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.
(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.
(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.

Column	Description
Passing	There are two possible options: **Option** — **Description** 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.



Column	Description
	<table><tr><th>Option</th><th>Description</th></tr><tr><td>  </td><td>The X indicates that the source or target does not pass the column schema check.</td></tr></table>
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.

Cell

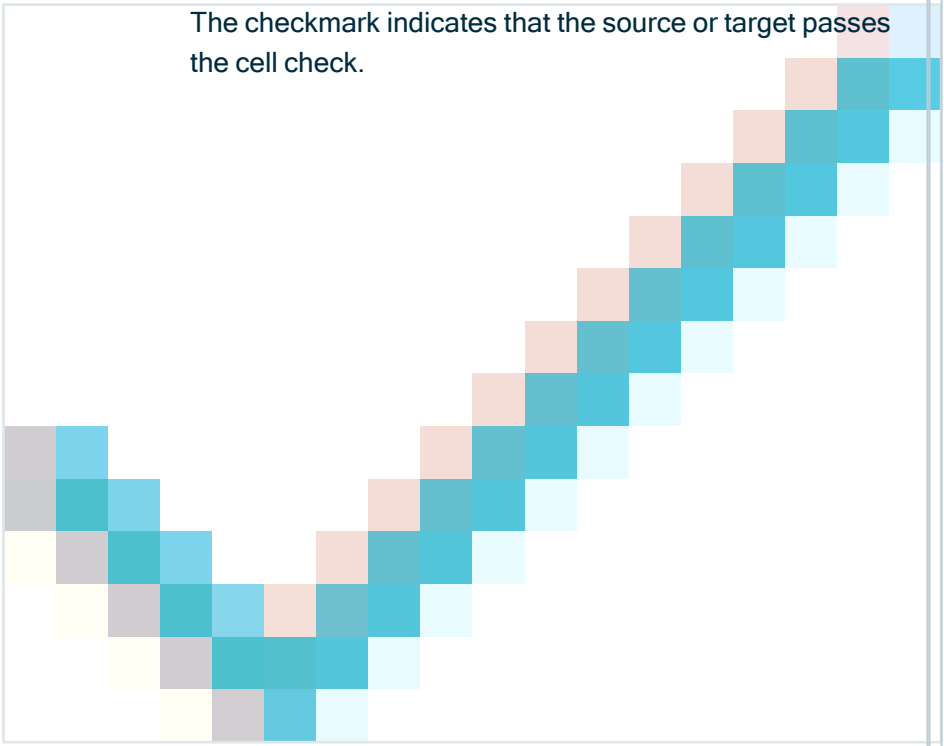
# of rows compared are the total number of rows from the target and source datasets that Collibra DQ includes in its cell check.

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.
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
	<table><tr><th>Option</th><th>Description</th></tr><tr><td></td><td>The X indicates that the source or target does not pass the cell check.</td></tr></table>
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 activity 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.
Action	The available dropdown menu options are Validate 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. **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 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

Schema 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.
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

Dupes are 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. 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	The status of your data quality item, for example, Observation. Status also 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 [[[Undefined variable DataQuality.Pushdown processing]]] mode, you can download either a CSV or JSON file containing details of the break records.

Exporting dupes records

Click **Export** above the drill-in table to generate an Excel file with the details from the drill-in.

Shapes

Shapes detect rare or inconsistent data formats in string columns.

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.

Column	Description
Status	The status of your data quality item, for example, Observation. Status also 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 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 [[[Undefined variable DataQuality.Pushdown processing]]] mode, you can download either a CSV or JSON file containing details of the break records.

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
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

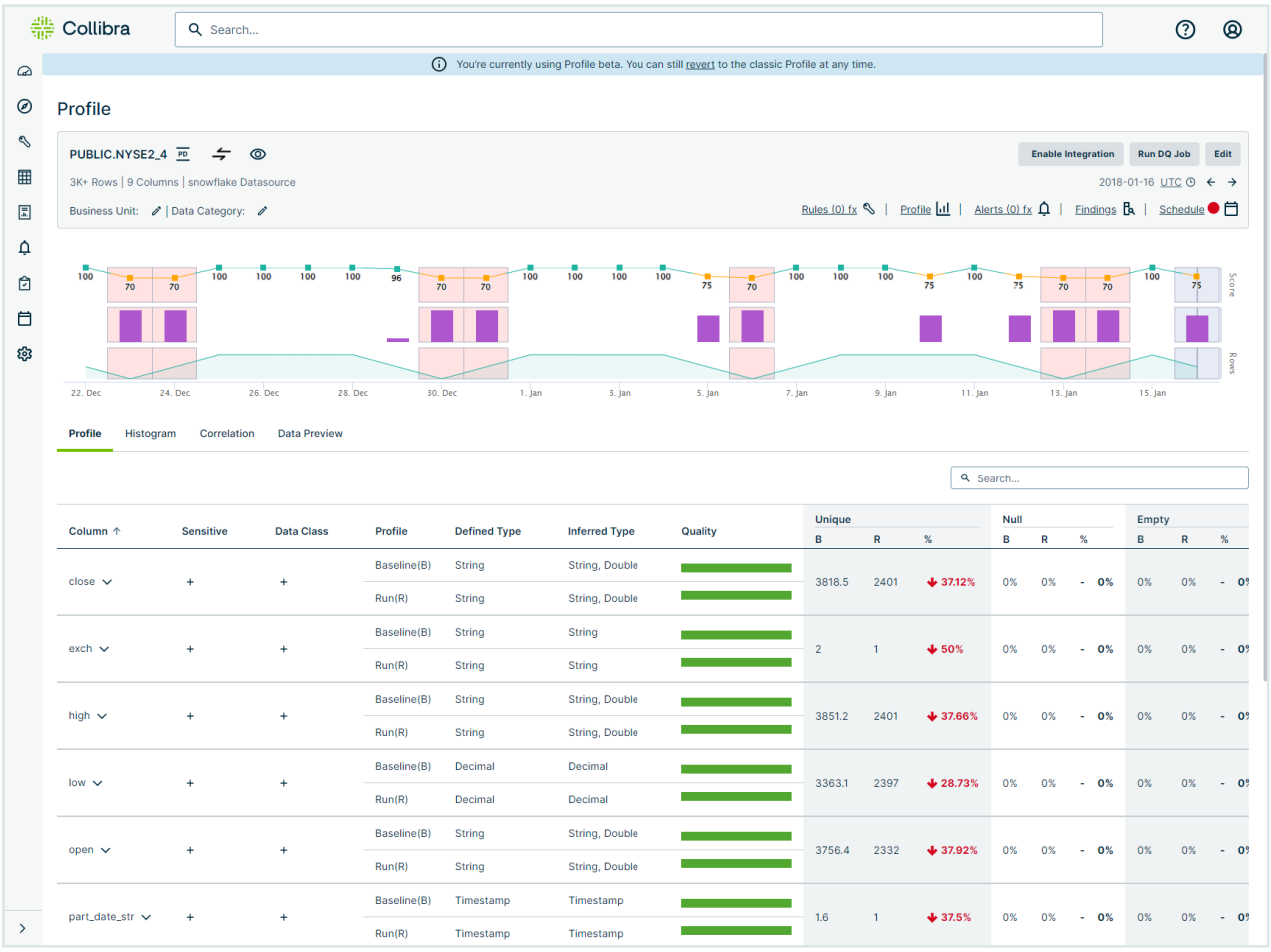
Profile

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.

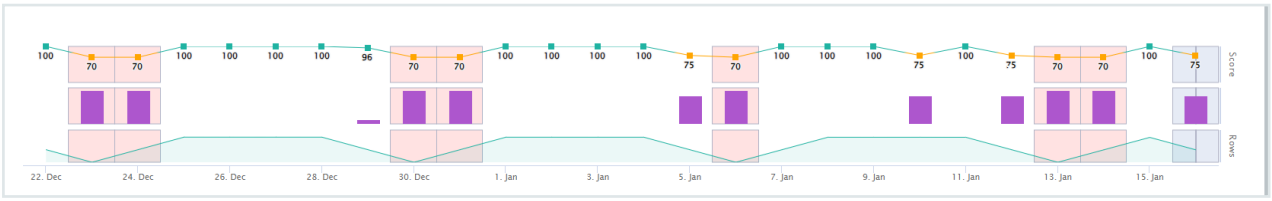
Collibra DQ automatically profiles datasets over time to allow you to drill down into its columns for detailed insights and automated data quality. The creation of a dataset profile is just the first step towards an amazing amount of auto-discovery. Visualize segments of the dataset and how the dataset 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.

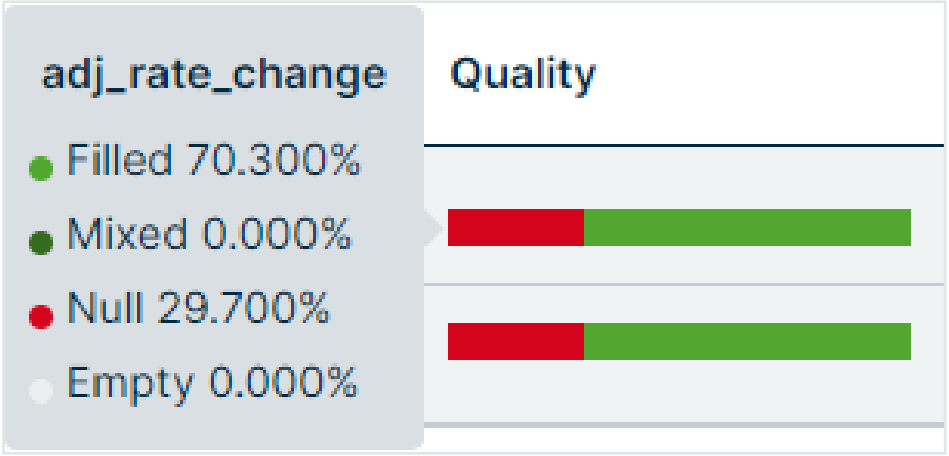
Dataset profile table

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 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 clickable + 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.

Column	Description
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.
Profile	**Baseline (B)** is the historical profile of a dataset.
	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.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Quality</td><td>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.
<table>
<tr><th>Metric</th><th>Type</th><th>Description</th></tr>
<tr><td>Filled</td><td>[1] Integer</td><td>The percentage of data that is numeric (or non-numeric) in a numeric (or non-numeric) discovered column.</td></tr>
<tr><td>Mixed</td><td>[String] Integer</td><td>The percentage of data that is non-numeric (or numeric) in a numeric (or non-numeric) discovered column.</td></tr>
<tr><td>Null</td><td>[]</td><td>The percentage of data that has no value at all.</td></tr>
<tr><td>Empty</td><td>[""]</td><td>The percentage of data that has a string instance of zero length.</td></tr>
</table>
See an example

</td></tr>
</table>

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

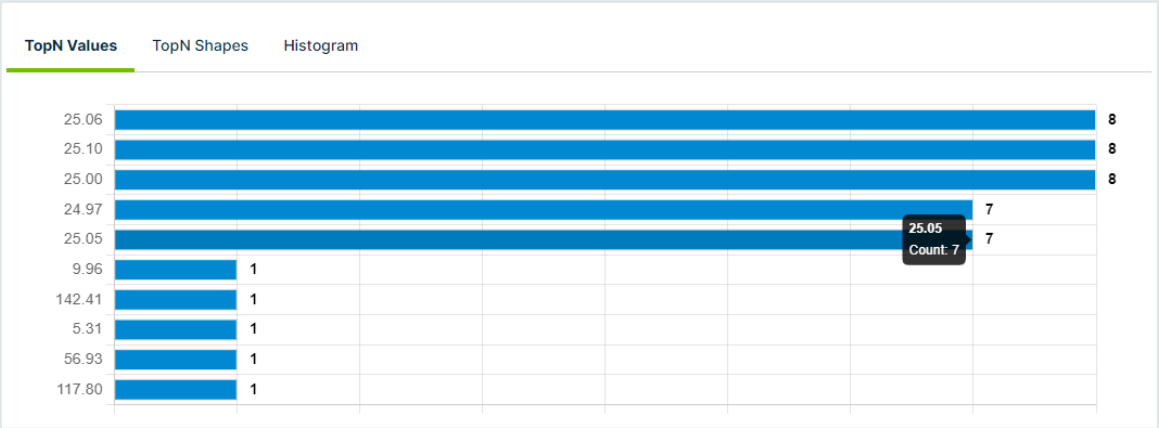
Column insights become visible when you drill down into a column. These insights include key statistical data points, data visualization tools, and overviews of Adaptive Rules and Rules. Click the expand icon to the right of the column name to drill down into the column and reveal insights.

Column-level statistics

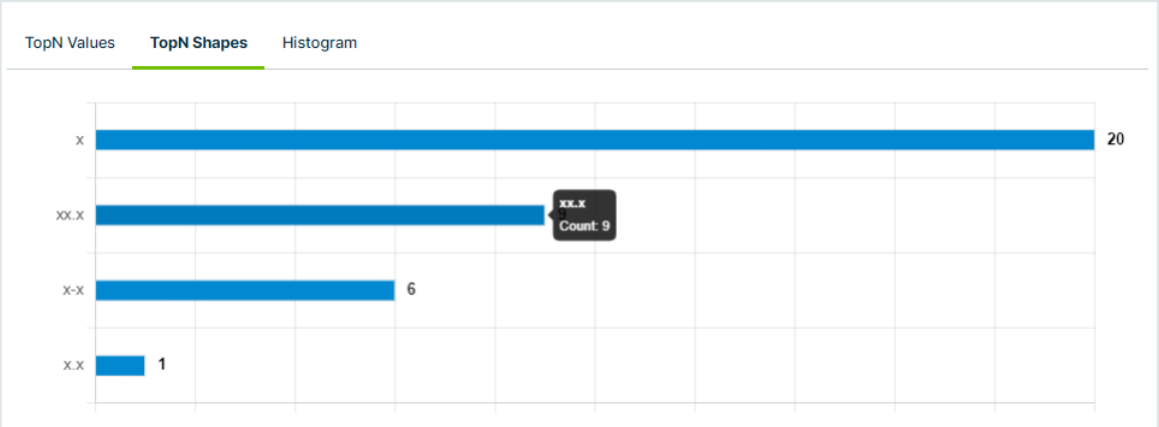
When data is available, a drill down into a column reveals four sections on the left:

- The **Stats** section shows high-level statistics, 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.
- The **Completeness** section shows the following donut charts:
  - 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.

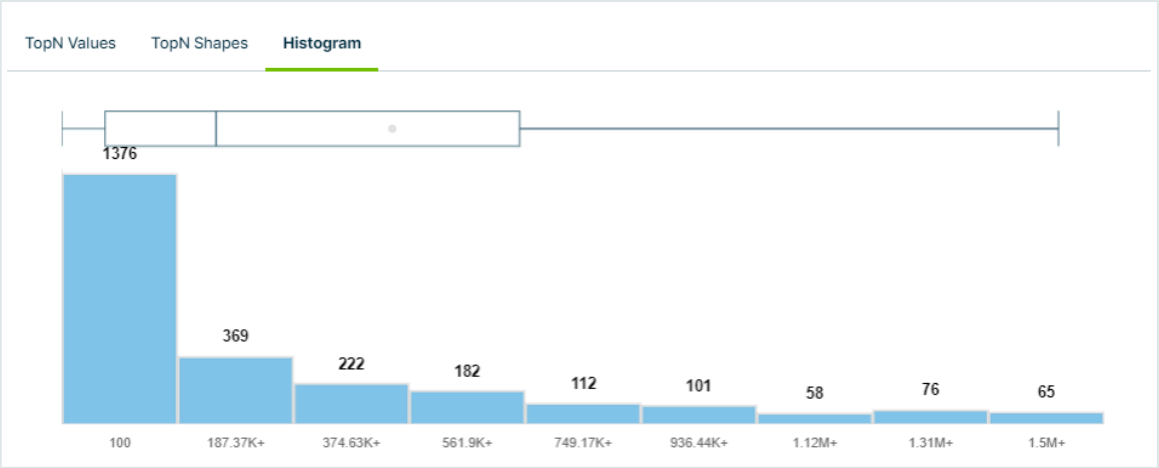
Column-level data visualization



The TopN Values tab shows the cell values that appeared the most and least frequently in the column of reference. 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. 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. This provides insights into important data points, including:

- Minimum value
- Lower and upper quartiles
- Median
- Mean
- Maximum value

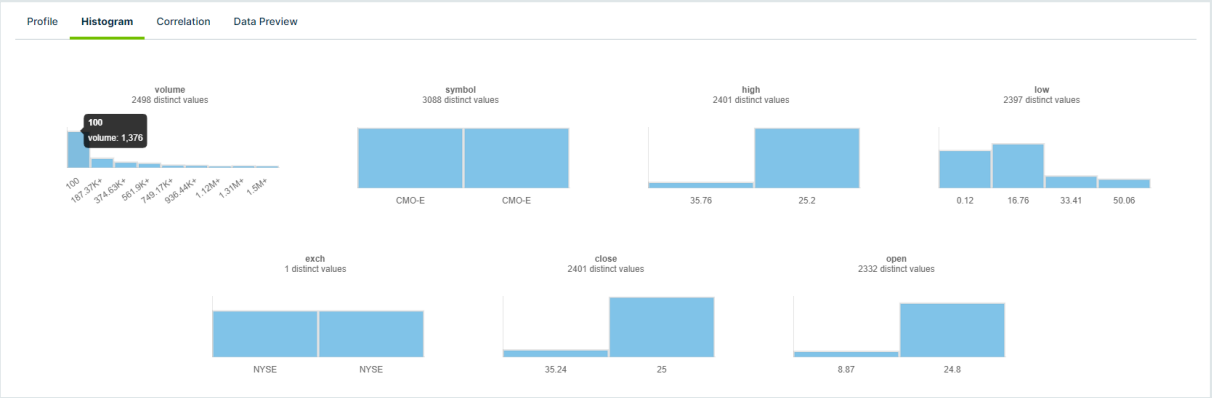
Hover your cursor over the box and whisker plot to see the above data points, or hover over an individual bar to see the values it represents.

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.

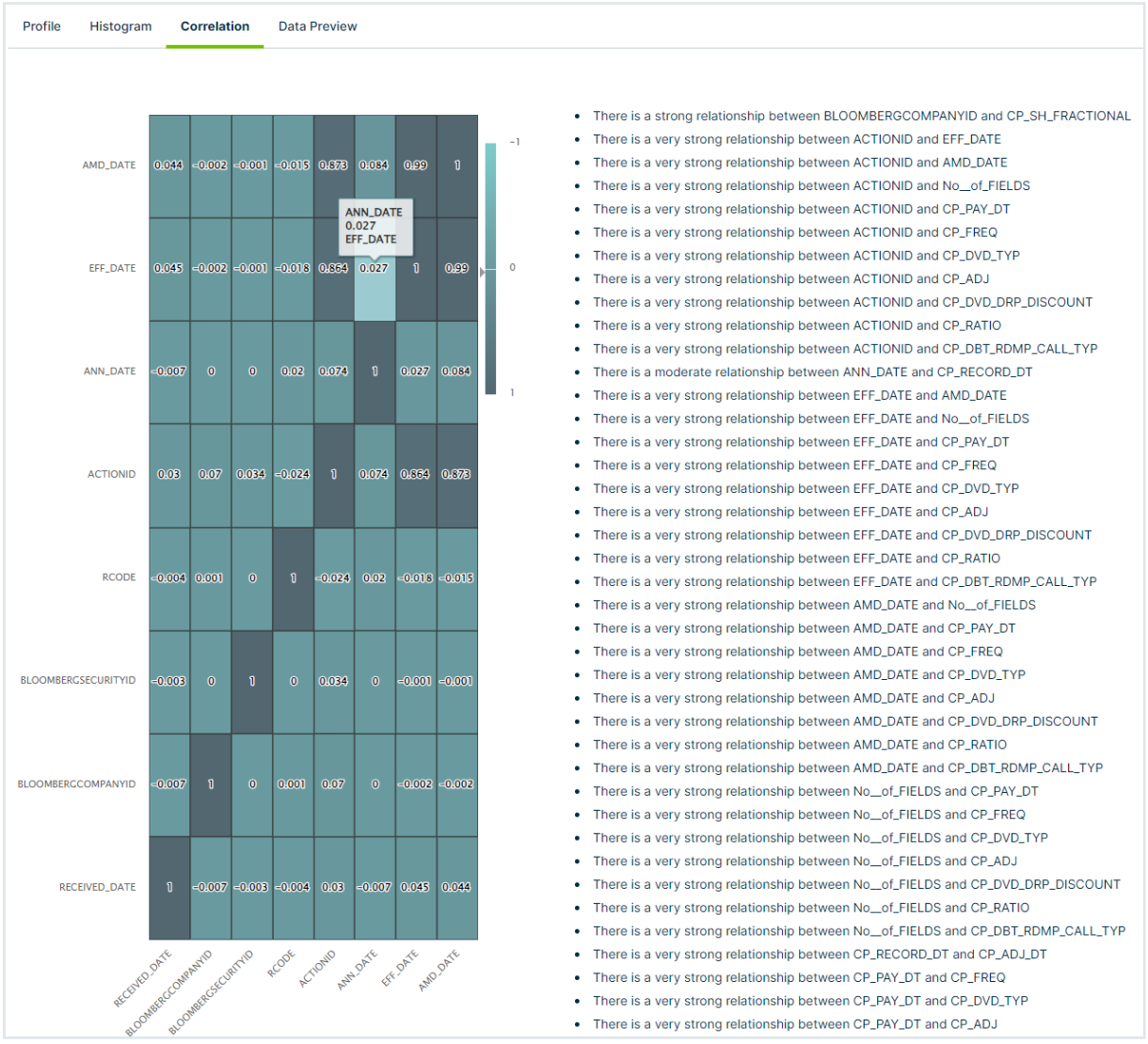
Column	Description
Rule Name	The name of your column-level rule.
Value	The condition of your column-level rule.
Valid	Shows whether a rule passes validation. ✓ indicates that a rule is valid. ⚠ indicates that a rule is invalid.
Repo	The data class or template from which a rule is created. This only applies 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 histogram overview of all of the columns in your dataset where histogram data is available. From this tab, you can also see the number

of distinct values that a column contains, as well as the frequency appears in a column when you hover your cursor over a histogram bar, as shown in the screenshot above.

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

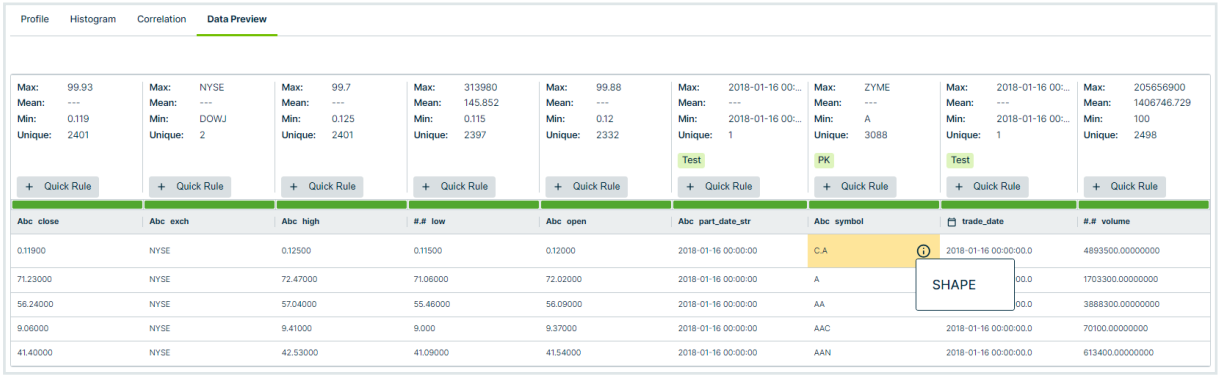
relationships in your dataset and measure their strength. Using just a few examples from the above screenshot, Collibra DQ observed the following relationships:

- 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.

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.

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.
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.

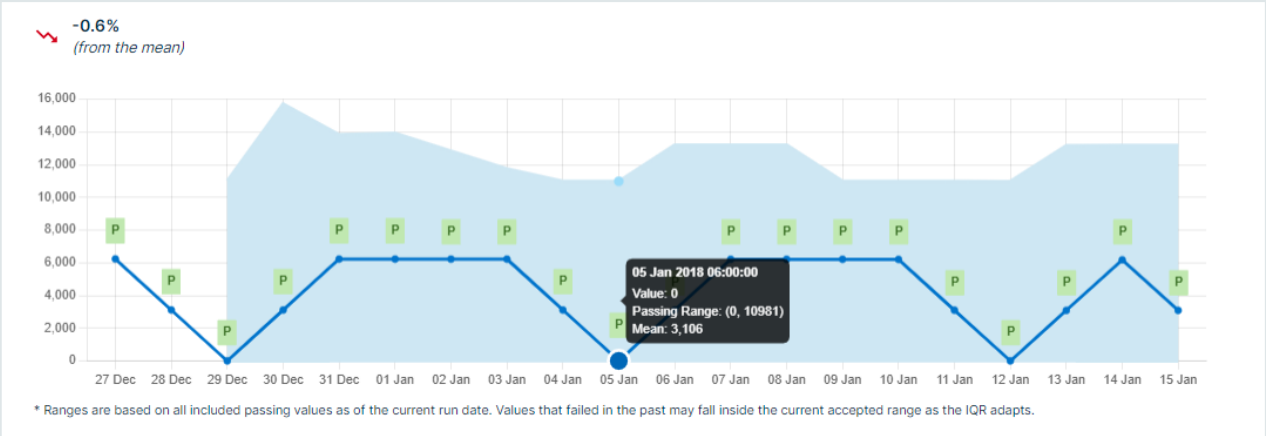
Adaptive level	Threshold	Impacts
Cardinality (Standard Deviation)		
Strict	Cardinality standard deviation threshold (ABS): 2.1 Cardinality standard deviation min/max threshold (z-score for min/max drift): 3.1	• Uniques (Cardinality) • Mean (Standard Deviation)
Neutral/System Detected	Cardinality standard 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)		
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)		

Adaptive level	Threshold	Impacts
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		
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.

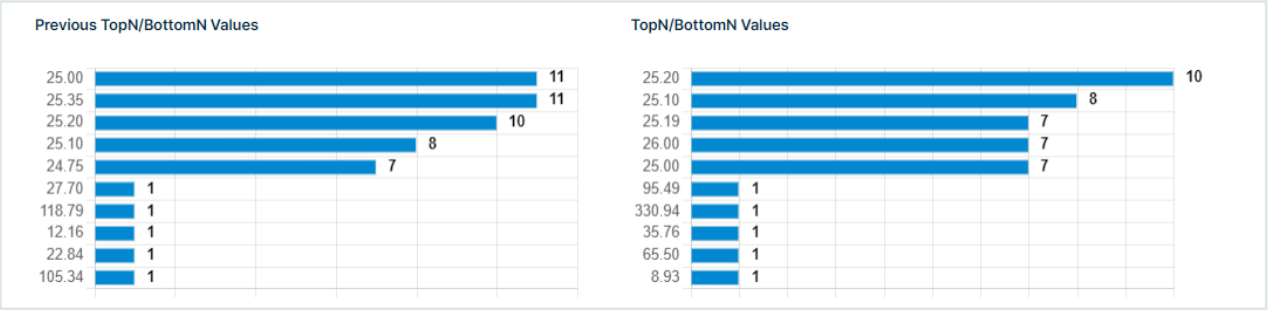
Turns off alerts for unique findings in the selected column.

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
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

Setting	Description	Default Value
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

Setting	Description	Default Value
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
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>
<tr><td>Dataset</td><td>The name of the dataset that this unique rule queries.

Note This field is not editable.</td><td>N/A</td></tr>
</table>

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.
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

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.
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.

Option	Description
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 Collibra Data Quality & Observability 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 Collibra Data Quality & Observability 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.

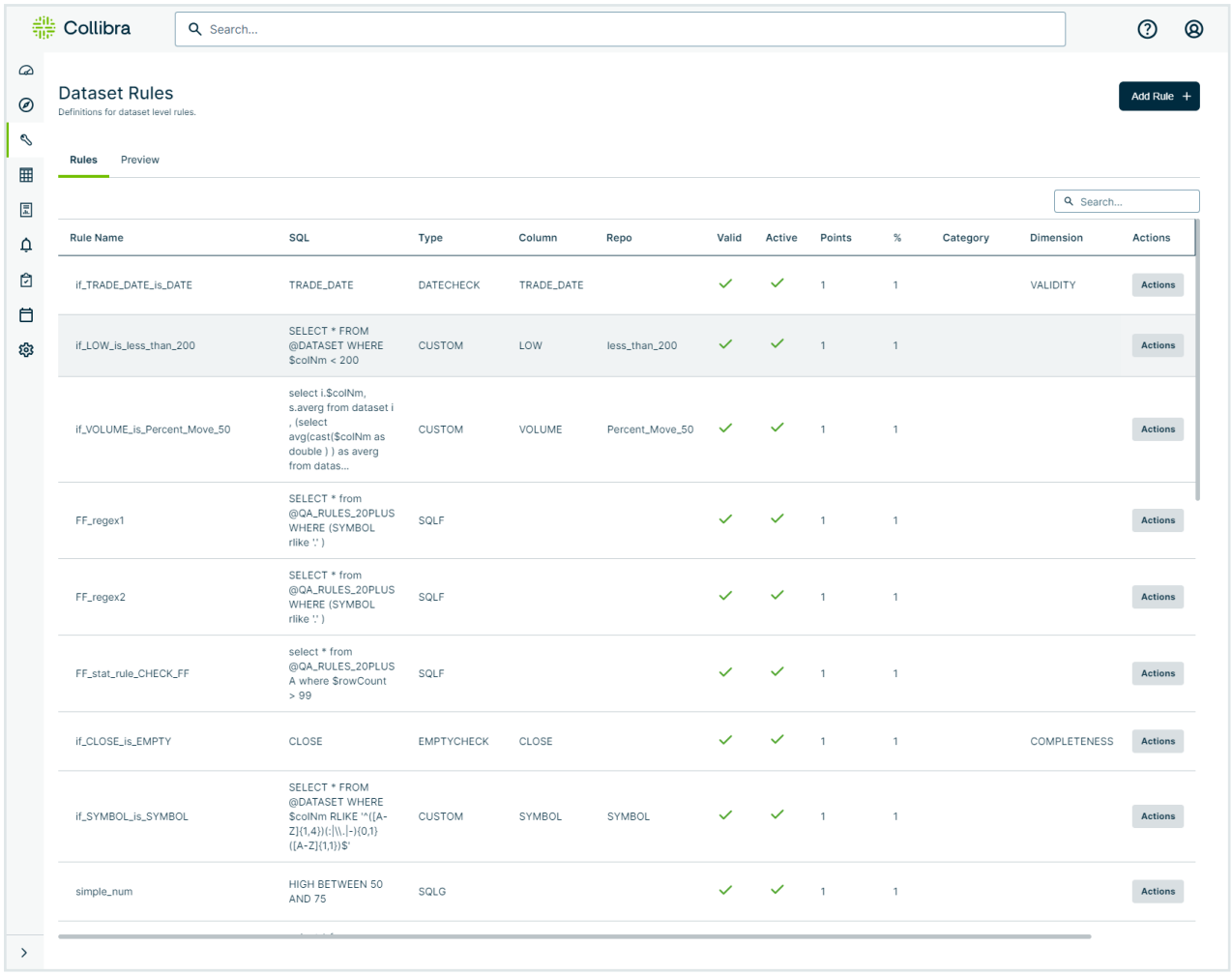
Prerequisites

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 your rule. Hover your cursor over the rule name and click ✎ to edit your rule.
SQL	The SQL condition of your rule.
Type	The type of rule.
Column	The primary column that your rule queries.
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.
Valid	Shows whether your rule passes rule validation. ✓ shows that your rule passes validation. ⚠ shows that your rule does not pass validation. > Note If you see ⚠, you can click it to view more details about why your rule does not pass validation.
Active	Shows whether your rule is active for future runs of the dataset. Click the icon to change the active status of your rule. ✓ shows that your rule is active. ⊗ shows that your rule is inactive.
Points	The number of points that Collibra DQ deducts from your data quality score when data breaches the conditions of your rule. You can set this value on the Rule Details modal of the Rule Workbench. If you do not customize this value, then Collibra DQ uses the default value of 1.
%	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 your rule on the Workbench.

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 or delete your rule.

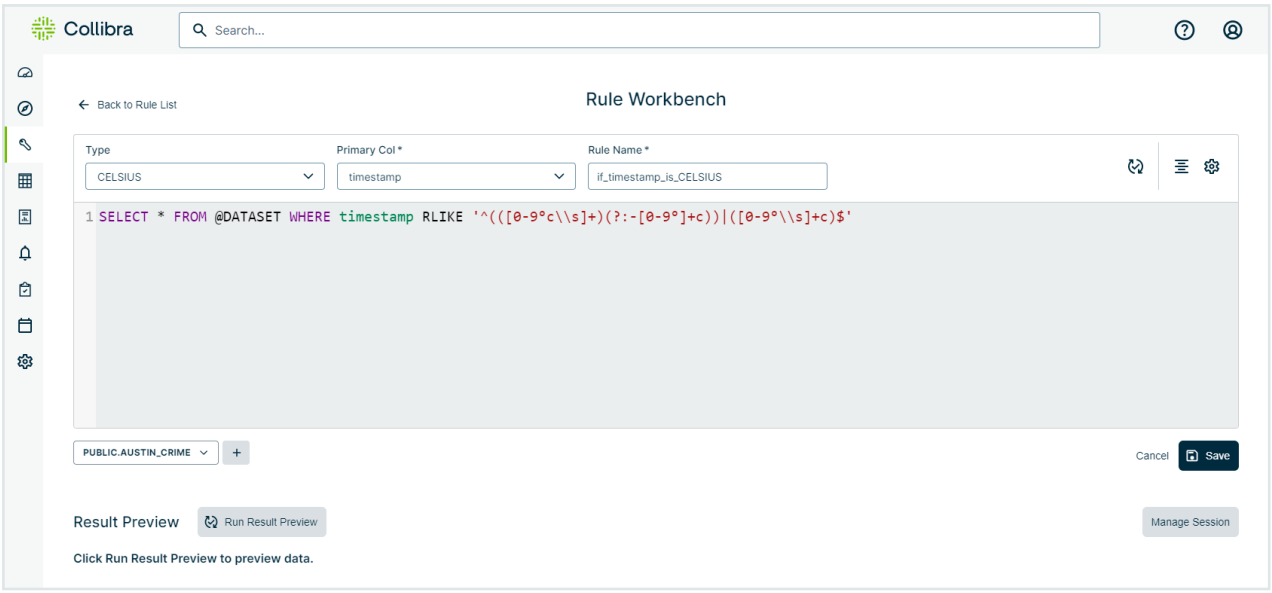
Adding new rules

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.



Steps

Select a navigation method from the following tabs.

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.

Option	Description
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 expressions 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 Because the rules of Pushdown jobs process in the native database, such as Snowflake, Native SQL rules are ineligible for Pushdown jobs.
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.
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.

Option	Description
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.

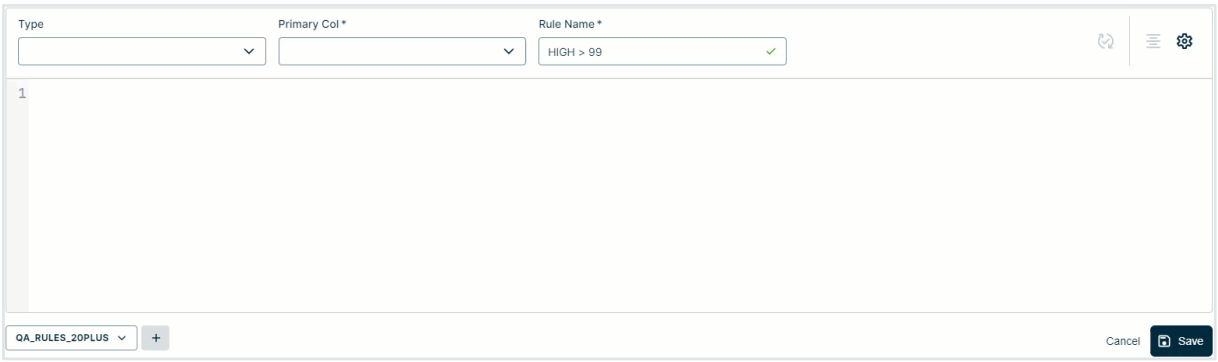
4. Enter a name for your rule in the **Rule Name** input field.

   Important Rule names cannot contain spaces.

5. Select a primary column from the Primary Col dropdown menu.

   Note Primary Column is only required when you use a Data Type, Data Class, or Template rule.

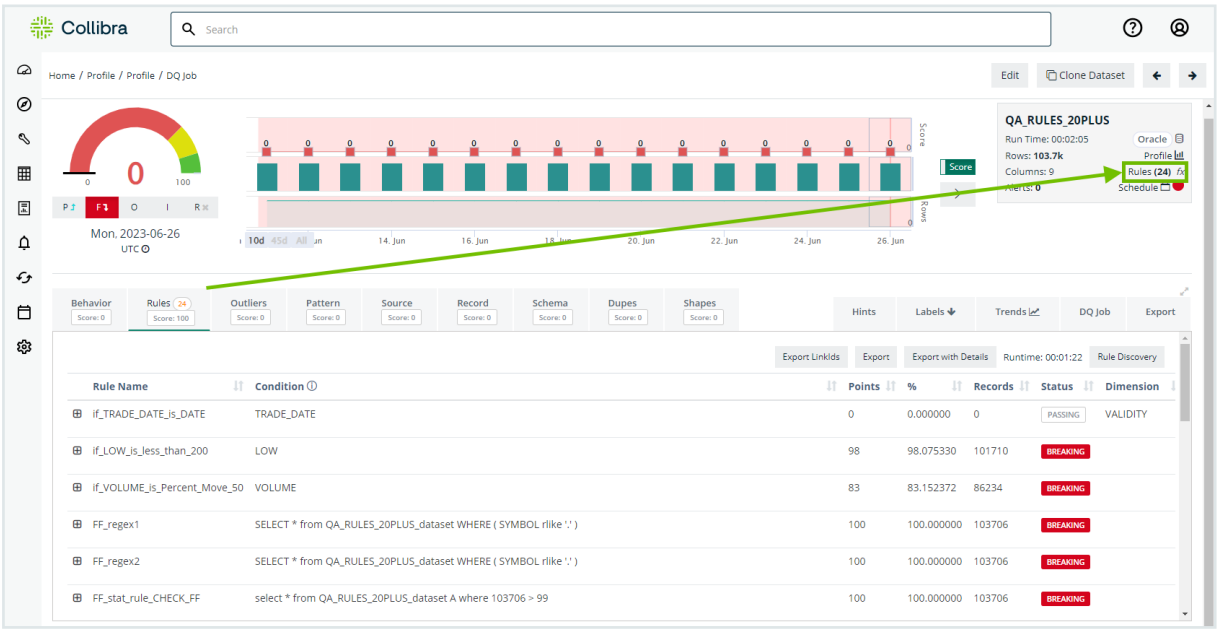
6. Optionally click + to add a secondary dataset.
   a. Enter the name of your secondary dataset in the search bar when the **Add Secondary Dataset** dialog appears, then click **Add Secondary Dataset**.
7. Write your SQL query. When you select the Freeform rule type, as shown in the GIF below, the Workbench 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 For a complete list of Spark SQL operators and functions available for use on SQLG and SQLF rule types, see the official Spark documentation.

8. Click **Validate** to validate your expression.
9. Optionally click **Beautify** to format your SQL query in a more readable layout.
10. Click **Save** to apply your rule to your dataset.

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 expressions 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 Because the rules of Pushdown jobs process in the native database, such as Snowflake, Native SQL rules are ineligible for Pushdown jobs.
Data Type	Data Type rules check for nulls, empty values, or specific data types.

Option	Description
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.
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.

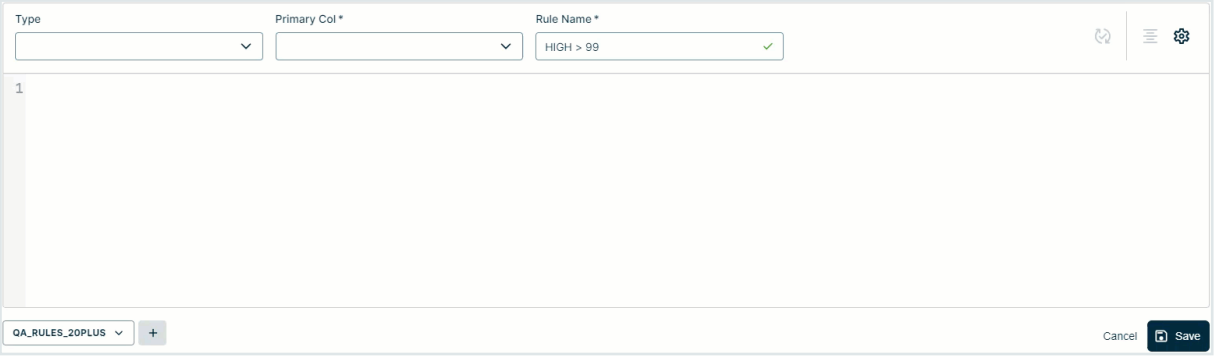
> Important Rule names cannot contain spaces.

5. Select a primary column from the Primary Col dropdown menu.

> Note Primary Column is only required when 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**.

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 For a complete list of Spark SQL operators and functions available for use on SQLG and SQLF rule types, see the official Spark documentation.

8. Click **Validate** to validate your expression.
9. Optionally click **Beautify** to format your SQL query in a more readable layout.
10. Click **Save** to apply your rule to your dataset.

Configuring Rule Details

To configure rule details, click the cogwheel 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.

<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>DQ Dimension</td><td>Associate your rule with a DQ Dimension, as shown in the following table. This can be especially useful when you use the DQ Connector and custom reporting.
<table>
<tr><th>Dimension</th><th>Description</th></tr>
<tr><td>Completeness</td><td>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.</td></tr>
<tr><td>Timeliness</td><td>Timeliness refers to the degree to which data is relevant and up-to-date at a certain point in time.</td></tr>
<tr><td>Uniqueness</td><td>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.</td></tr>
<tr><td>Validity</td><td>Validity refers to the degree to which data conforms to its defining constraints or conditions, which can include data type, range, or format.</td></tr>
<tr><td>Consistency</td><td>Consistency refers to the degree to which data contains differing, contradicting, or conflicting entries.</td></tr>
<tr><td>Accuracy</td><td>Accuracy refers to the degree to which data correctly reflects its intended values.</td></tr>
</table>
</td></tr>
<tr><td>Category</td><td>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".</td></tr>
<tr><td>Description</td><td>Give a detailed description of your rule.</td></tr>
<tr><td>Preview Limit (Rows)</td><td>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.</td></tr>
</table>

<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>Run Time Limit (Minutes)</td><td>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.</td></tr>
<tr><td>Score</td><td>
<table>
<tr><th>Score</th><th>Description</th></tr>
<tr><td>Low</td><td>Downscores 1 point for every 1% of records that violate your rule. For example, when 25 of 100 records break your rule, then 25 points are deducted from the total DQ score.</td></tr>
<tr><td>Medium</td><td>Downscores 5 points for every 1% of records that violate your rule. For example, when 2 of 100 records break your rule, then 10 points are deducted from the total DQ score.</td></tr>
<tr><td>High</td><td>Downscores 25 points for every 1% of records that violate your rule. For example, when 3 of 100 records break your rule, then 75 points are deducted from the total DQ score.</td></tr>
</table>
</td></tr>
<tr><td>Scoring Type</td><td>
<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td>Percent</td><td>Default. 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.</td></tr>
<tr><td>Absolute</td><td>Deducts points according to the scoring threshold you set in the Points input field. When your rule breaks, the scoring model deducts the total number of points you set. For example, if you set the value of the Points input field to 10 and your rule breaks, then a total of 10 points are deducted from your overall DQ score.</td></tr>
</table>
</td></tr>
<tr><td>Points</td><td>The number of points deducted from the total DQ score when your rule breaks. You can either use the Score buttons to automatically update this input field, or enter a custom value to meet your business needs.</td></tr>
</table>

Setting	Description
Percent	You can either use the Score buttons to automatically update this input field, or enter a custom value to meet your business needs. Note If you select Absolute as the scoring type, then this option is not available.

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.

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.

[[[Undefined variable DataQuality.SQL assistant]]] (Beta)

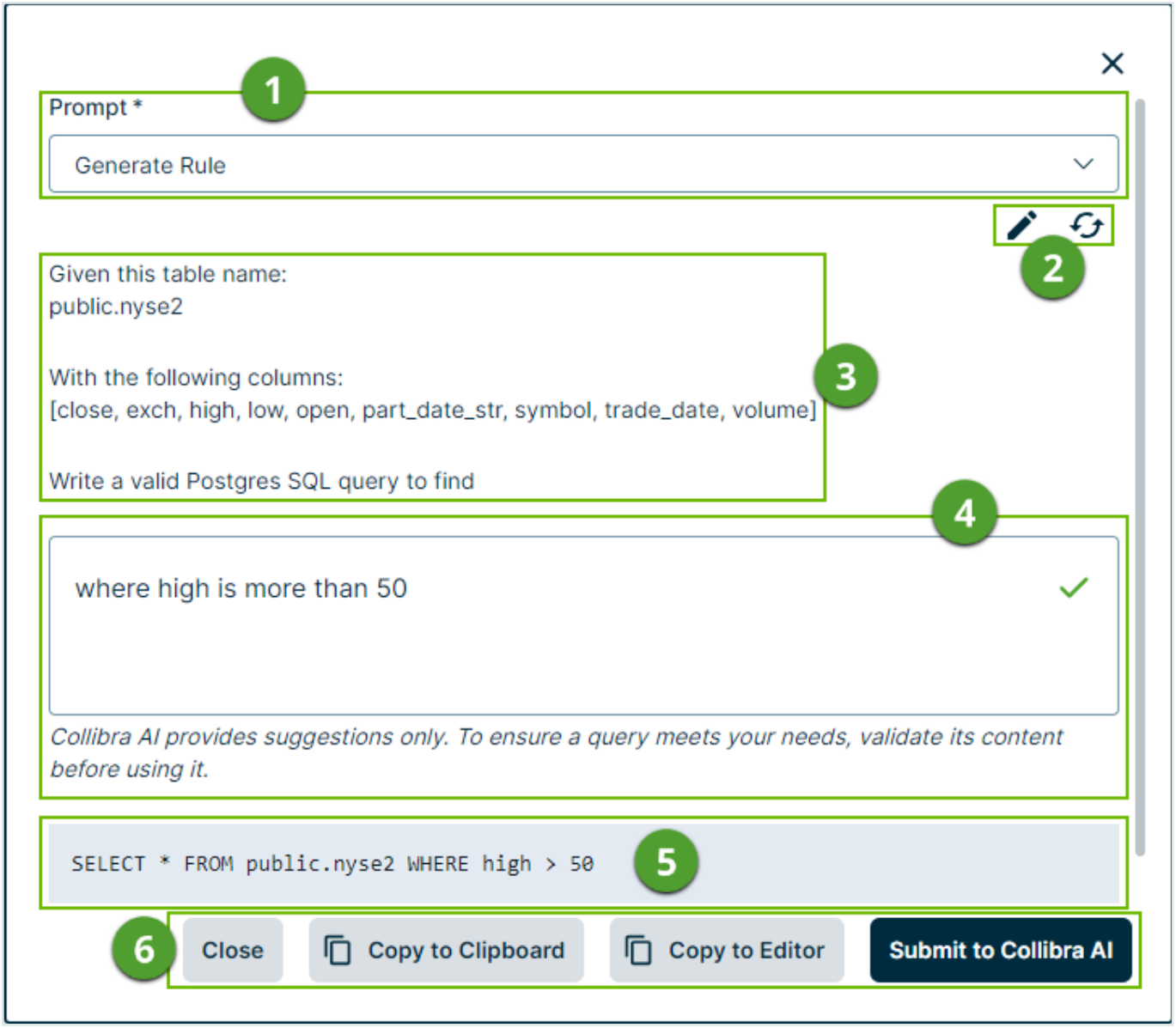
> Note This feature is in Beta testing and is powered by Collibra AI. If you would like to share your feedback with our product team, please go to the Collibra beta platform.

[[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]], 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 [[[Undefined variable DataQuality.SQL assistant]]] allow you to elegantly correct syntactical mistakes, resolving the once tedious SQL syntax reviews of the past.

You can use [[[Undefined variable DataQuality.SQL assistant]]] to assist with SQL rule writing and troubleshooting on the following pages:

- Dataset Overview
- Rule Workbench


Prompt *
1
Generate Rule
2
Given this table name:
public.nyse2
With the following columns:
[close, exch, high, low, open, part_date_str, symbol, trade_date, volume]
3
Write a valid Postgres SQL query to find
4
where high is more than 50
Collibra AI provides suggestions only. To ensure a query meets your needs, validate its content before using it.
SELECT * FROM public.nyse2 WHERE high > 50
5
6
Close
Copy to Clipboard
Copy to Editor
Submit to Collibra AI

No.	Component	Description
1	Collibra AI options	Click the dropdown menu and select an option. • **Generate Rule** creates SQL suggestions based on your plain-text input in the request prompt. • **Troubleshoot Rule** provides suggestions to fix and rewrite 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.
2	Edit and reset prompt	Optionally click ✎ to manually edit the default request and request prompt, or click ↻ to refresh your request prompt.
3	Default request	The dataset metadata that instructs [[[Undefined variable DataQuality.SQL assistant]]] on what to include in the query. This is typically all of the details before the `WHERE` clause in a SQL query.
4	Request prompt	An input field where you can enter a prompt for [[[Undefined variable DataQuality.SQL assistant]]] in plain-text format.
5	Results	The results that [[[Undefined variable DataQuality.SQL assistant]]] returns based on the content of your prompt.
6	Action buttons	• 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 on the Dataset Overview modal. • 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.

Prerequisites

To use [[[Undefined variable DataQuality.SQL assistant]]] in Collibra DQ, you first need to:

- Ensure that the AI_TENANT application configuration is set to **TRUE** in the Admin Console ▸ Configuration Settings.
- Safelist network egress for **oauth2.googleapis.com** and **us-central1-aiplatform.googleapis.com** over port 443.
- Have one of the following Collibra DQ roles assigned to your user account:
  - ROLE_ADMIN
  - ROLE_GENAI_USER

> Note Collibra DQ administrators can add or update roles from the Admin Console ▸ Role Management ▸ Roles.

Using [[[Undefined variable DataQuality.SQL assistant]]]

[[[Undefined variable DataQuality.SQL assistant]]] uses Collibra AI to automatically write and troubleshoot SQL queries. When working on a Pullup dataset, the type of SQL that [[[Undefined variable DataQuality.SQL assistant]]] generates depends on the rule type. For instance, if your rule type is Freeform SQL, the SQL generated by [[[Undefined variable DataQuality.SQL assistant]]] 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 following steps show you how to use [[[Undefined variable DataQuality.SQL assistant]]] and from where you can access it.

From Dataset Overview

1. From the Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » The Dataset Overview modal appears.

2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] 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.

Examples

- "high is more than 50"
- "create a complex rule that is 25 lines long and returns only 1 record"
- "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**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] to automatically troubleshoot and fix the error.

1. From the Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Troubleshoot Rule**.

   » The [[[Undefined variable DataQuality.SQL assistant]]] 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**.

   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Categorical**.
4. In the query input field, enter the name of the column in your dataset to include in the SQL query.
5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:

   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Dupe**.
4. In the query input field, enter the name of the column in your dataset to include in the SQL query.
5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder

   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Record**.

4. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

> Important You must specify one column with numeric data and one date column.

5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 Metadata Bar at the top of any of the following dataset-level pages, click the to open Dataset Overview:
   a. Findings
   b. Profile
   c. Dataset Rules
   d. Alert Builder
   » The Dataset Overview modal appears.
2. In the upper right corner of the SQL editor, click **Collibra AI**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
3. From the Prompt dropdown menu, select **Pattern**.
4. In the query input field, enter the names of the columns in your dataset to include in the SQL query.

5. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

From the Rule Workbench

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 [[[Undefined variable DataQuality.SQL assistant]]] 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.

Examples

- "high is more than 50"
- "create a complex rule that is 25 lines long and returns only 1 record"
- "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**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
6. From the Prompt dropdown menu, select **Troubleshoot Rule**.
   » The [[[Undefined variable DataQuality.SQL assistant]]] 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**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL rule 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Categorical**.
5. In the query input field, enter the name of the column in your dataset to include in the SQL query.
6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Dupe**.
5. In the query input field, enter the name of the column in your dataset to include in the SQL query.

6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Record**.
5. In the query input field, enter the names of the columns in your dataset to include in the SQL query.
6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool appears.
4. From the Prompt dropdown menu, select **Pattern**.
5. In the query input field, enter the names of the columns in your dataset to include in the SQL query.
6. Click **Submit to Collibra AI**.
   » [[[Undefined variable DataQuality.SQL assistant]]] 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 [[[Undefined variable DataQuality.SQL assistant]]] tool without copying the suggested query.
   d. Rewrite your prompt and click **Submit to Collibra AI** to generate a new SQL suggestion.

FAQ

- What models are used?
  The generative AI features use foundational models from Google Model Garden on Vertex AI. The 2 models that we use are PaLM 2 for Text (text-bison) and

Embeddings for text (textembedding-gecko).

- Are the models fine-tuned or retrained in any way?
  No, we do not fine-tune or retrain them in the beta implementation.
- Where, regionally, are the models located?
  The models are located in the US Central region.
- Where, regionally, are the other components located?
  The US Central region.
- How is data secured when it is sent from the proxy to the Vertex Models?
  All data is encrypted in transit using HTTPS with TLS protocols.
- What data is sent to Vertex AI?
  When a user submits a request to Collibra AI, the following info is sent to Vertex AI:
  - Generate Rule
    - Table name
    - Column names
    - User-generated plain language query description
  - Troubleshoot Rule
    - Table name
    - Column names
    - Current query
    - Contents of error message from SQL engine
- What data is stored in Vertex AI?
  The only data stored in Vertex AI is metadata regarding user queries, which does not currently include any user query or model output. This includes metadata such as total billable characters, character count, and latency metrics. Additionally, Google has made it clear that, in our current setup, no prompts are used to improve their foundational models. We do foresee a scenario where user queries might be stored in the future, but we will engage the Legal team and obtain approval prior to any changes to this.

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.

```
email_address is not null and email_address != ''
```

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, CTE's and window statements.

```
select * from @DATASET_NAME where
email_address is not null and email_address != ''
```

> Note All built-in spark functions are available to use. (https://spark.apache.org/docs/2.3.0/api/sql/) for simple and freeform sql rules.

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 Stat Rules section for more details.

Data Type

- **Empty check**
  - Rule type: EMPTYCHECK
  - Description: Checking whether the target column has empty values or not
- **Null check**
  - Rule type: NULLCHECK
  - Description: Checking whether the target column has *NULL* values or not
- **Date check**
  - Rule type: DATECHECK
  - Description: Checking whether the target column has only DATE values or not
- **Integer check**
  - Rule type: INTCHECK
  - Description: Checking whether the target column has only INTEGER values or not
- **Double check**
  - Rule type: DOUBLECHECK
  - Description: Checking whether the target column has only DOUBLE values or not
- **String check**
  - Rule type: STRINGCHECK
  - Description: Checking whether the target column has only STRING values or not.
- **Mixed datatype check**
  - Rule type: DATATYPECHECK
  - **Description:** ---

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. (https://spark.apache.org/docs/2.3.0/api/sql/) for simple and freeform sql rules.

Freeform

Fully defined SQL for more detailed checks.

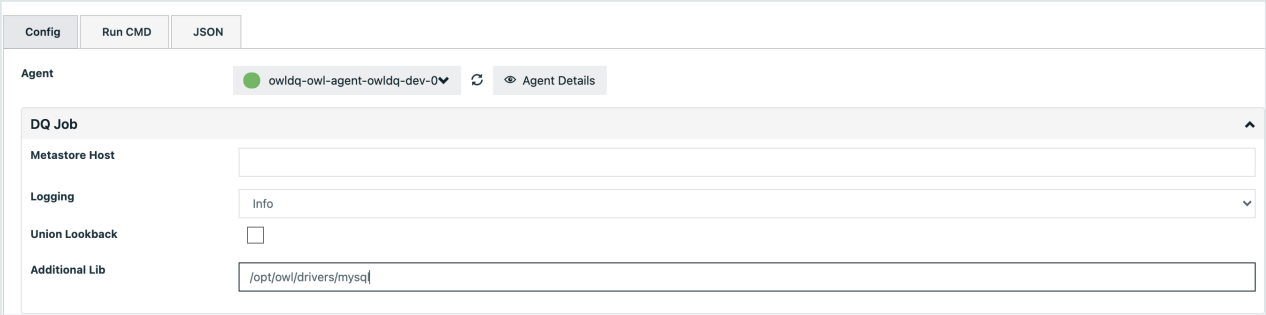
Where 'Simple' rules just use the condition, 'Freeform' rules use the complete SQL statement. When more complex SQL is required, you can express more with Freeform including joins, CTE's and window statements.

```
select * from @DATASET_NAME where
email_address is not null and email_address != ''
```

> Warning Freeform rules should include a semi-colon at the end of the rule value.

> Note All built-in Spark functions for non-Pushdown connections are available for use. See (https://spark.apache.org/docs/2.3.0/api/sql/) for Simple and Freeform SQL rules.

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.

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
```

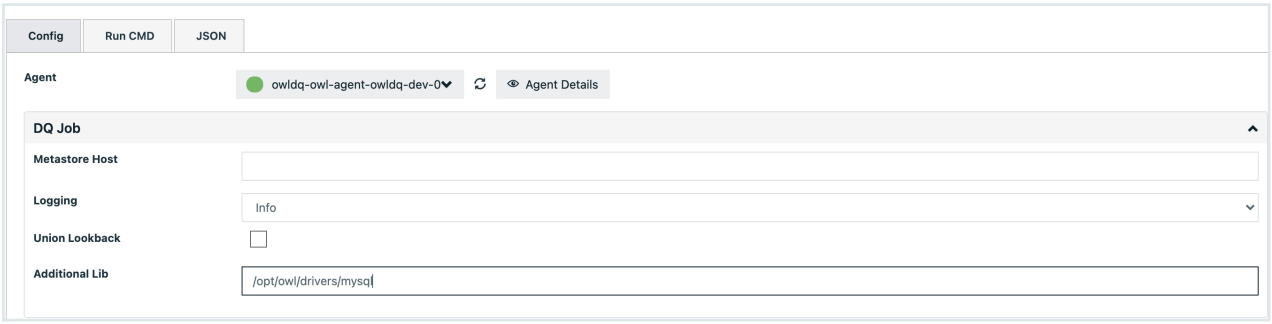
A possible workaround to this limitation is to use aliasing instead.

Cross-Data Set Rules

Warning Best practice for cross-table joins is to define a view and scan the view. Only in circumstances when a view cannot be created should you define cross-table joins with 2 separate data sets (DQ Jobs) and express the join in the rule.

If you're doing multiple lookups, this will improve long-term performance and consolidate maintenance.

Note Cross-dataset rules require -libsrc (prior to 2021.11) or -addlib (post 2022.01)



In-Clause (Single Column)

```
select * from @table1 where id
not in ( select id from @table2 )
```

Except (Multi-Column)

```
select id, app_id, email, guid_num from @table1
EXCEPT
select id, app_id, email, guid_num from @table2
```

Referencing secondary data sets

When you create a Freeform SQL rule with secondary data sets, there are three important points to consider 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. >> If the column name uses CAPS, then your query must also use CAPS.
- Ensure that any data set referenced in your rule maintains a score of greater than or equal to (>=) a passing score of 75.

> Note These three points apply to all rules, not just Freeform Rules.

> Note 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.

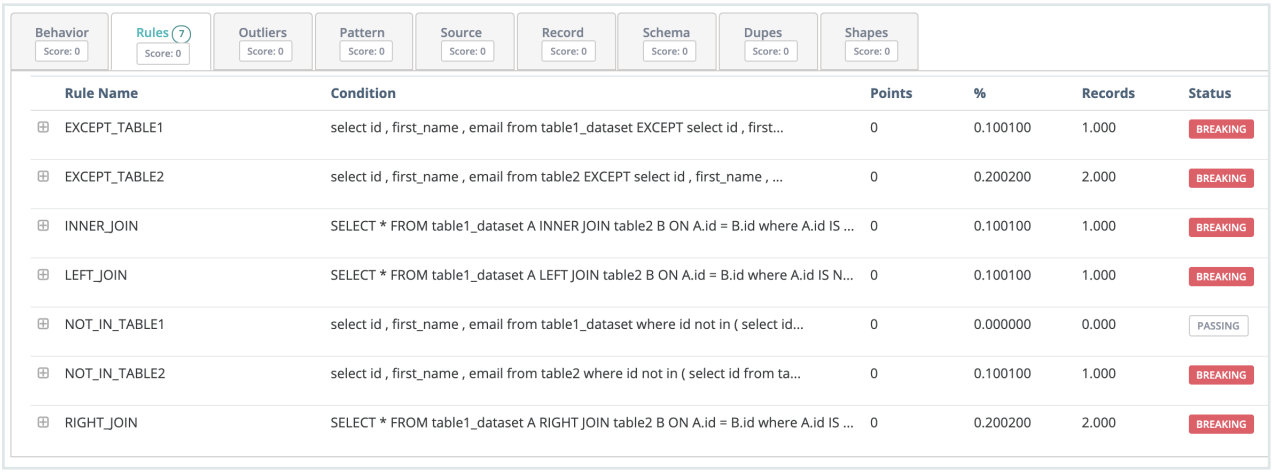
Joins

A **join** lets you reference two or more data sets in one rule. Joins are useful when you want to compare the values of a data set from a previous, or when you want to verify that all values are valid across your data sets.

Join example

```
SELECT
    *
FROM
    @table1 A
    LEFT JOIN @table2 B ON A.id = B.id
where
    B.id is null OR (A.email != B.email)
```

Sample Results



Behavior Score: 0 | Rules 7 Score: 0 | Outliers Score: 0 | Pattern Score: 0 | Source Score: 0 | Record Score: 0 | Schema Score: 0 | Dupes Score: 0 | Shapes Score: 0

Rule Name	Condition	Points	%	Records	Status
EXCEPT_TABLE1	select id , first_name , email from table1_dataset EXCEPT select id , first...	0	0.100100	1.000	BREAKING
EXCEPT_TABLE2	select id , first_name , email from table2 EXCEPT select id , first_name , ...	0	0.200200	2.000	BREAKING
INNER_JOIN	SELECT * FROM table1_dataset A INNER JOIN table2 B ON A.id = B.id where A.id IS ...	0	0.100100	1.000	BREAKING
LEFT_JOIN	SELECT * FROM table1_dataset A LEFT JOIN table2 B ON A.id = B.id where A.id IS N...	0	0.100100	1.000	BREAKING
NOT_IN_TABLE1	select id , first_name , email from table1_dataset where id not in (select id...	0	0.000000	0.000	PASSING
NOT_IN_TABLE2	select id , first_name , email from table2 where id not in (select id from ta...	0	0.100100	1.000	BREAKING
RIGHT_JOIN	SELECT * FROM table1_dataset A RIGHT JOIN table2 B ON A.id = B.id where A.id IS ...	0	0.200200	2.000	BREAKING

Cross-Table (Guided). Use our wizard to do ad-hoc analysis and visual setup.

Join Example Example (vs. cross-table guided seen above).

Multi-part condition rules with the rule builder. Combines profiling metrics & builder in one screen.

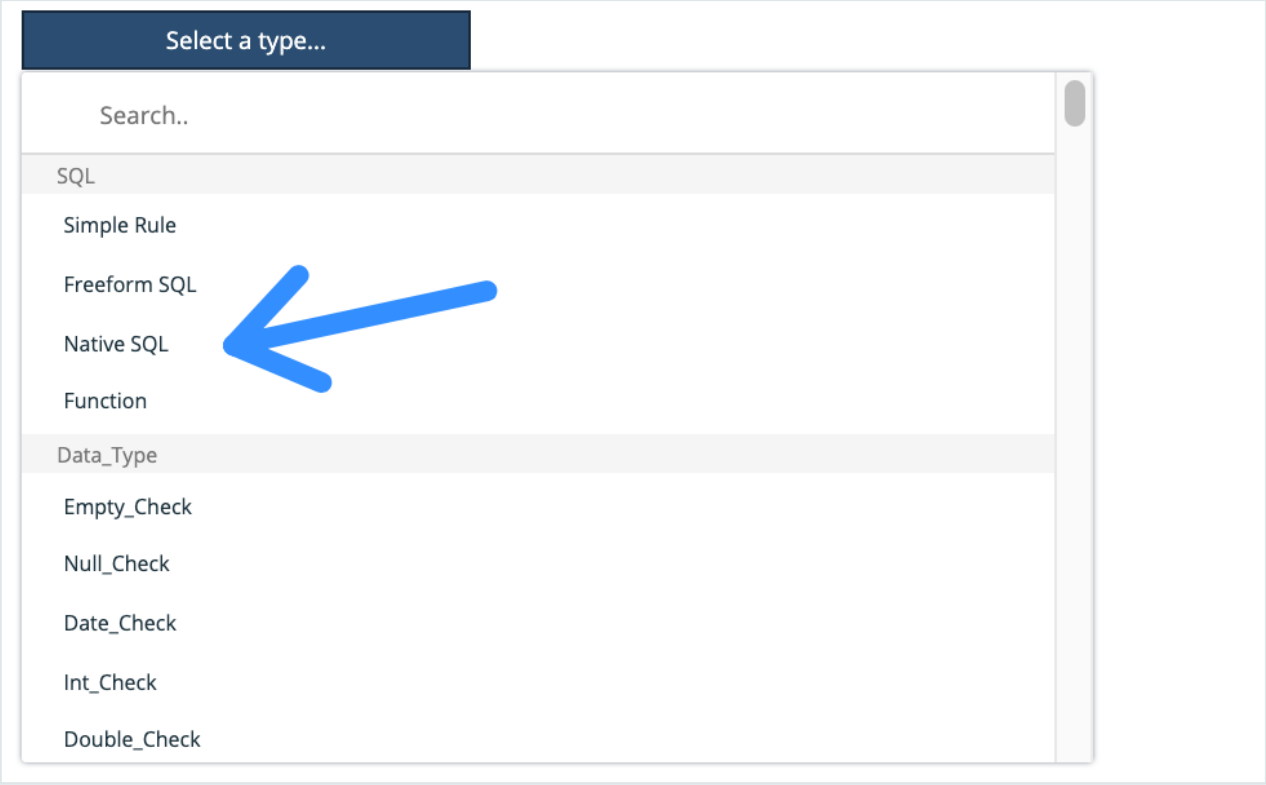
Known Limitations

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.

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.



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.

> 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

A powerful technique is to access profile statistics in your rules. 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. These are simplified below, for example `fname.$type != 'String'`.

```
select * from @dataset where
fname.$type != 'String' AND $rowCount < 800
```

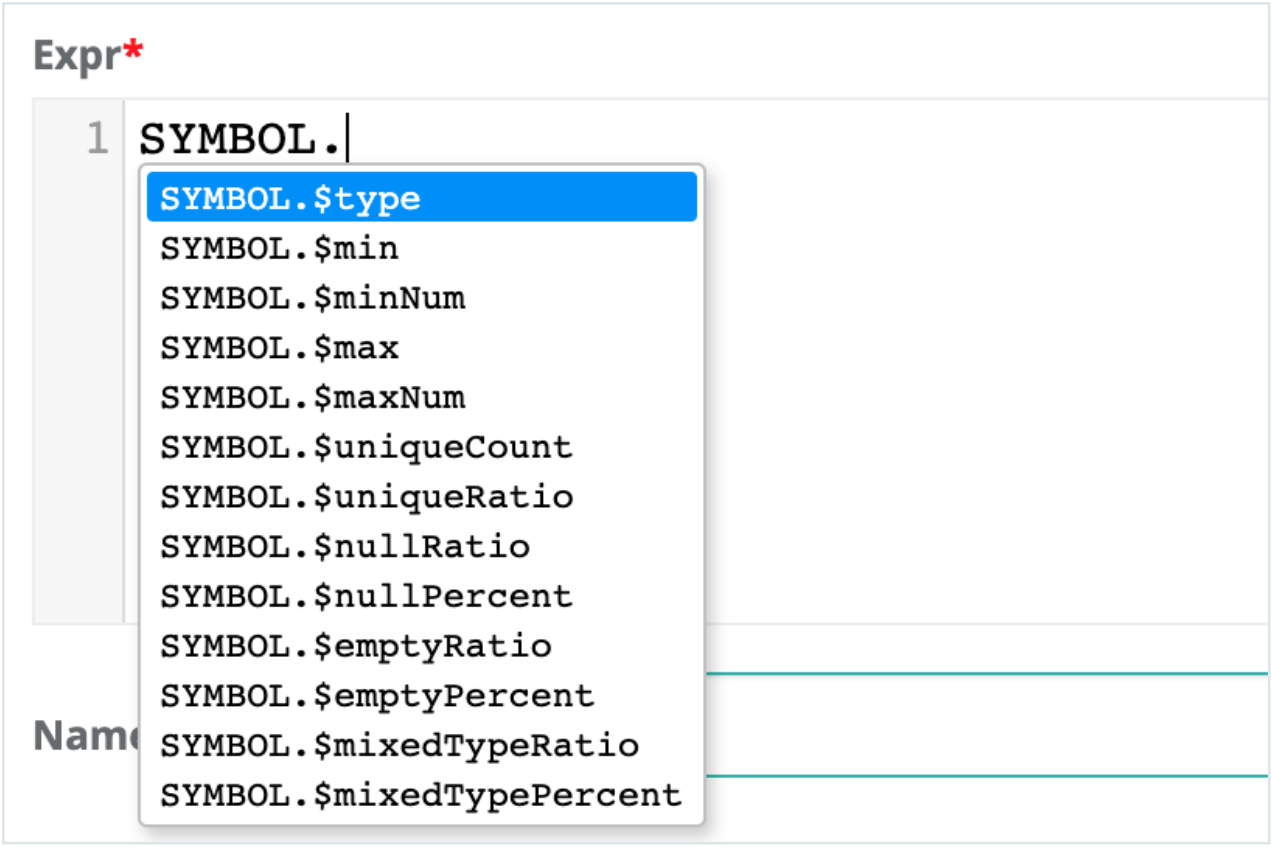
Dataset-level stat rules

Dataset-level Stat Rule	Rule Example	Description
$totalTimeInSeconds	$totalTimeInSeconds > 25	Alert when DQ job runs longer than 25 seconds.
$totalTimeInMinutes	$totalTimeInMinutes > 5	Alert when DQ job runs longer than 5 minutes.
$totalTimeInHours	$totalTimeInHours > 1	Alert when DQ job runs longer than 1 hour.
$rowCount	$rowCount < 9000	Alert when row count less than 9,000.
$runDate	$runDate='2020-01-24'	Use the ${rd} variable in rules.
$daysWithoutData	$daysWithoutData > 4	Alert when a dataset has missing or no rows for 5 days.
$runsWithoutData	$runsWithoutData > 4	Alert when a dataset runs but has missing or no rows for 5 days.
$daysSinceLastRun	$daysSinceLastRun > 4	Alert when a dataset has not run for 5 days.

Column-level stat rules

Column-level Stat Rule	Rule Example	Description
.$type	fname.$type != 'String'	Alert when fname is not a string.
.$min	fname.$min > 'apple'	Lexicographical sort works for strings and numbers.
.$minNum	age.$minNum > 13	Type casted to a numeric for simple number checks.
.$mean	row_id.$mean > '4.500'	Alert when the mean is greater than a given value.
.$max	fname.$max > 'apple'	Alert when the max is greater than a given value.

Column-level Stat Rule	Rule Example	Description
.$maxNum	age.$maxNum > 13	Alert when the numeric value falls outside an acceptable range.
.$uniqueCount	id.$uniqueCount != $rowCount	Alert when the uniqueCount of a field doesn't match the rowCount.
.$uniqueRatio	gender.$uniqueRatio between .4 and .6	Alert when the ratio of uniqueCounts of a given field doesn't match the rowCount.
.$nullRatio	lname.$nullRatio not between .4 and .6	Alert when the ratio of nulls no longer falls within acceptable range.
.$nullPercent	lname.$nullPercent not between 40 and 60	Alert when the percent of nulls no longer falls within acceptable range
.$nullCount	lname.$nullCount >= 1	Test for a single null.
.$emptyRatio	nc.$emptyRatio > 0.2	Alert when the ratio of empties no longer falls within acceptable range.
.$emptyPercent	nc.$emptyPercent > 20	Alert when the percent of empties no longer falls within an acceptable range.
.$emptyCount		Alert when the emptyCounts of a field no longer fall within an acceptable range.
.$mixedTypeRatio	nc.$mixedTypeRatio > 0.2	Alert when the ratio of mixed data types no longer falls within an acceptable range. For example, Strings and Ints in the same field.
.$mixedTypePercent	nc.$mixedTypeRatio > 20	Alert when the percent of mixed data types no longer falls within an acceptable range. For example, Strings and Ints in the same field.
.$mixedTypeCount	id.$mixedTypeCount >= 1	Alerts when the mixed data typeCount no longer falls within an acceptable range. For example, Strings and Ints in the same field.

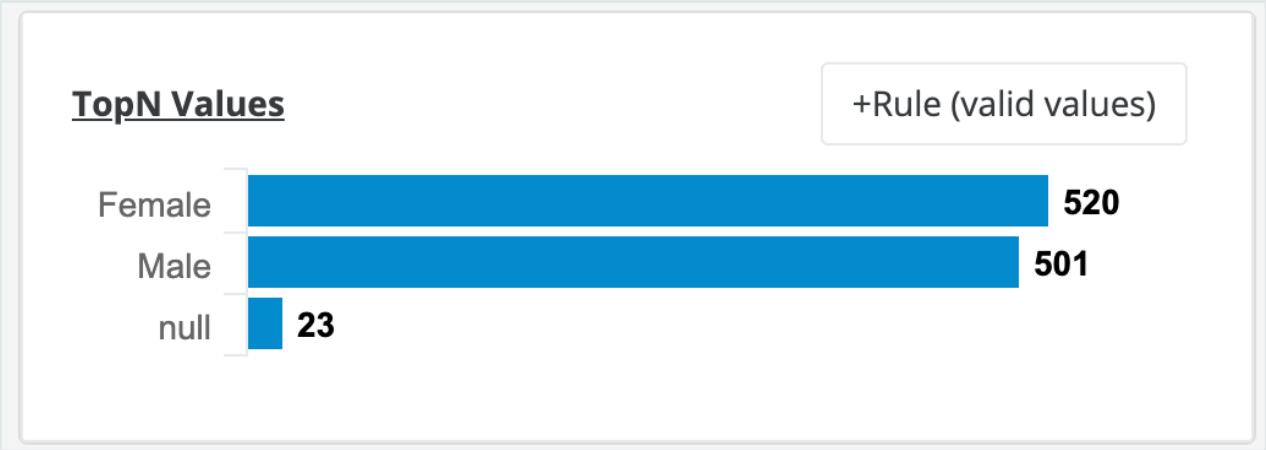


Distribution Rule

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

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.

Rule	Rule type	Description
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.
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**:

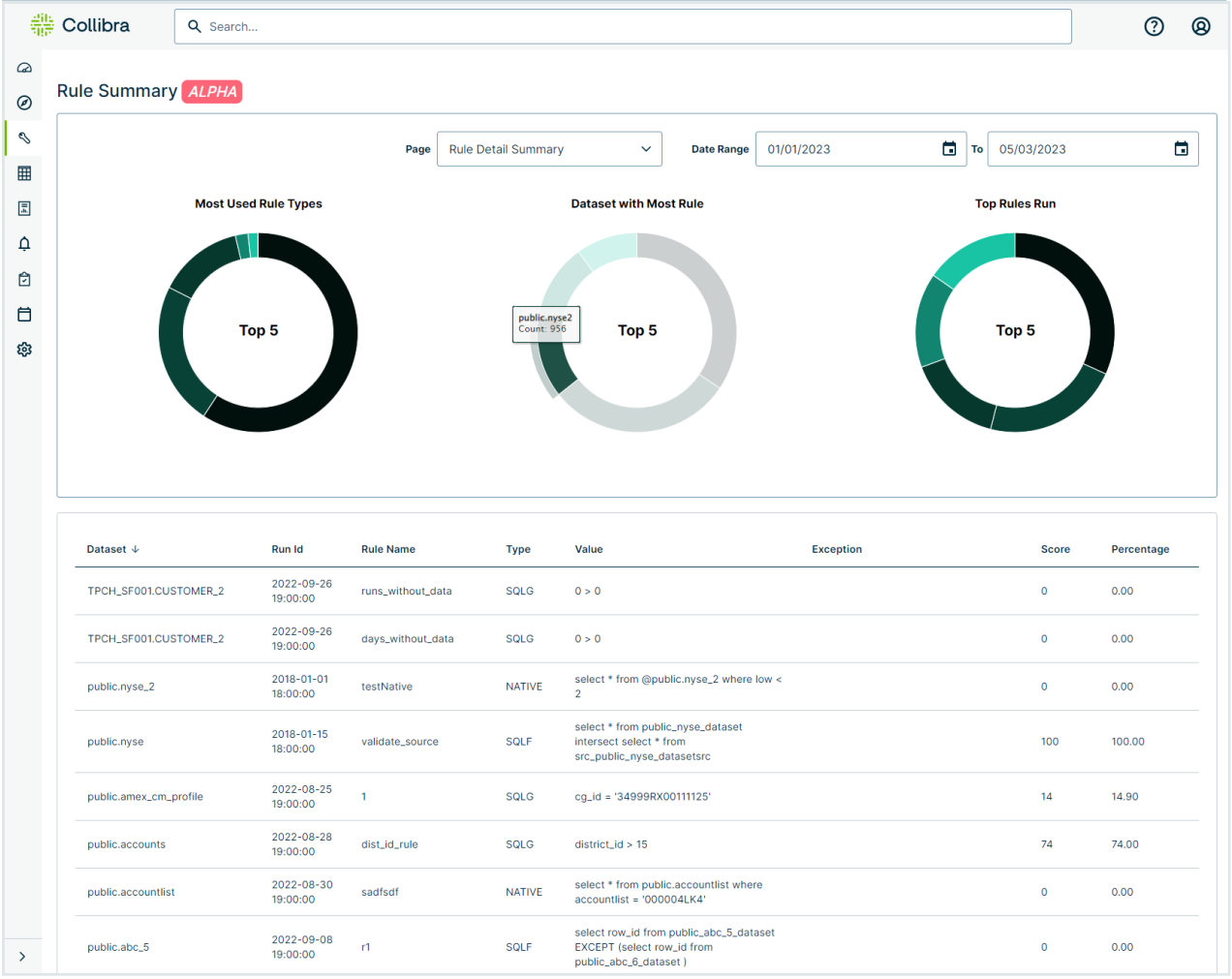
- Rule Detail Summary
- Rule Breaks
- Rule Dimension Summary

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.

Chart	Description
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.

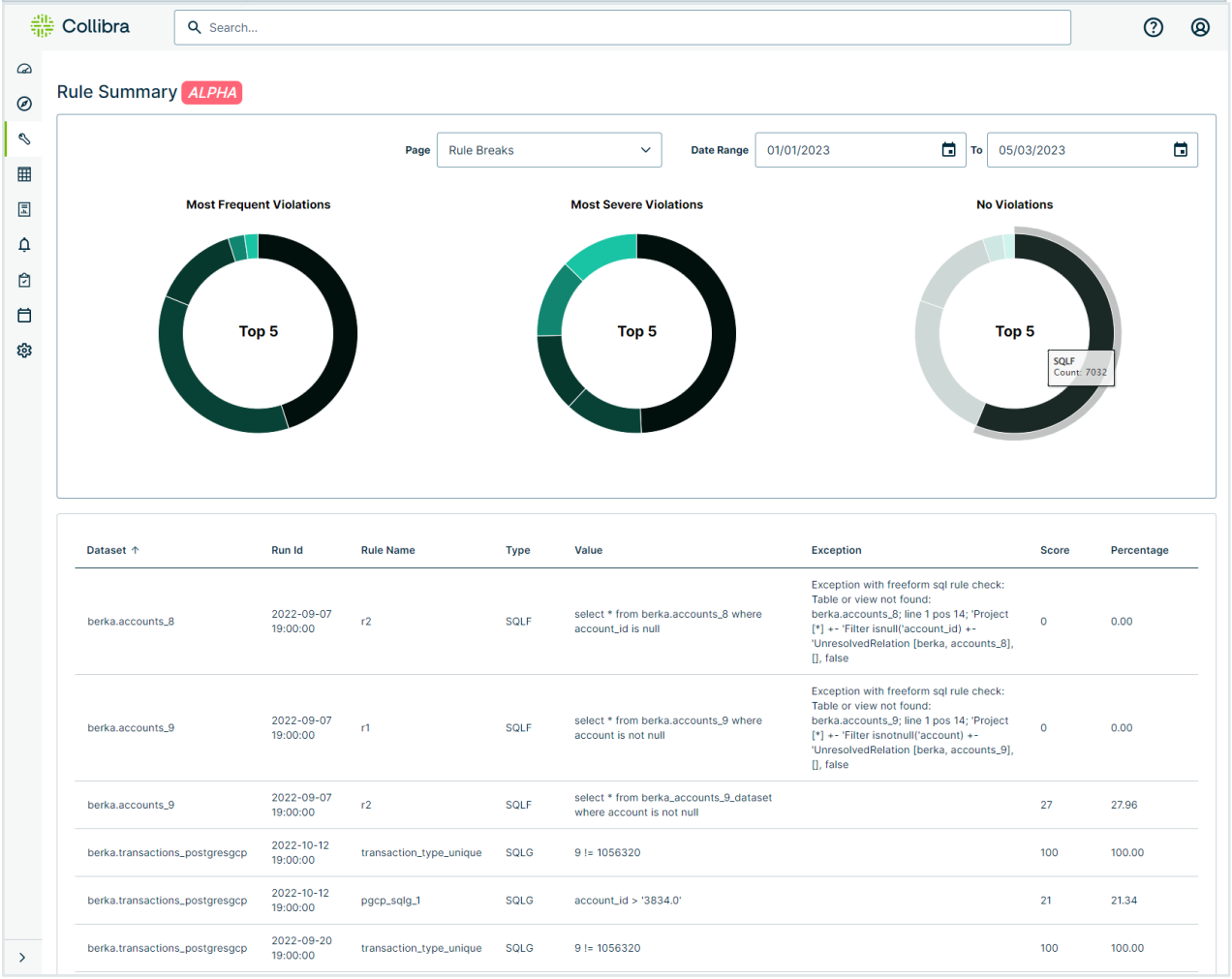


Dataset	Run Id	Rule Name	Type	Value	Exception	Score	Percentage
TPCH_SF001.CUSTOMER_2	2022-09-26 19:00:00	runs_without_data	SQLG	0 > 0		0	0.00
TPCH_SF001.CUSTOMER_2	2022-09-26 19:00:00	days_without_data	SQLG	0 > 0		0	0.00
public.nyse_2	2018-01-01 18:00:00	testNative	NATIVE	select * from @public.nyse_2 where low < 2		0	0.00
public.nyse	2018-01-15 18:00:00	validate_source	SQLF	select * from public_nyse_dataset intersect select * from src_public_nyse_datasetsrc		100	100.00
public.amex_cm_profile	2022-08-25 19:00:00	1	SQLG	cg_id = '34999RX00111125'		14	14.90
public.accounts	2022-08-28 19:00:00	dist_id_rule	SQLG	district_id > 15		74	74.00
public.accountlist	2022-08-30 19:00:00	sadfsdf	NATIVE	select * from public.accountlist where accountlist = '000004LK4'		0	0.00
public.abc_5	2022-09-08 19:00:00	r1	SQLF	select row_id from public_abc_5_dataset EXCEPT (select row_id from public_abc_6_dataset)		0	0.00

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.

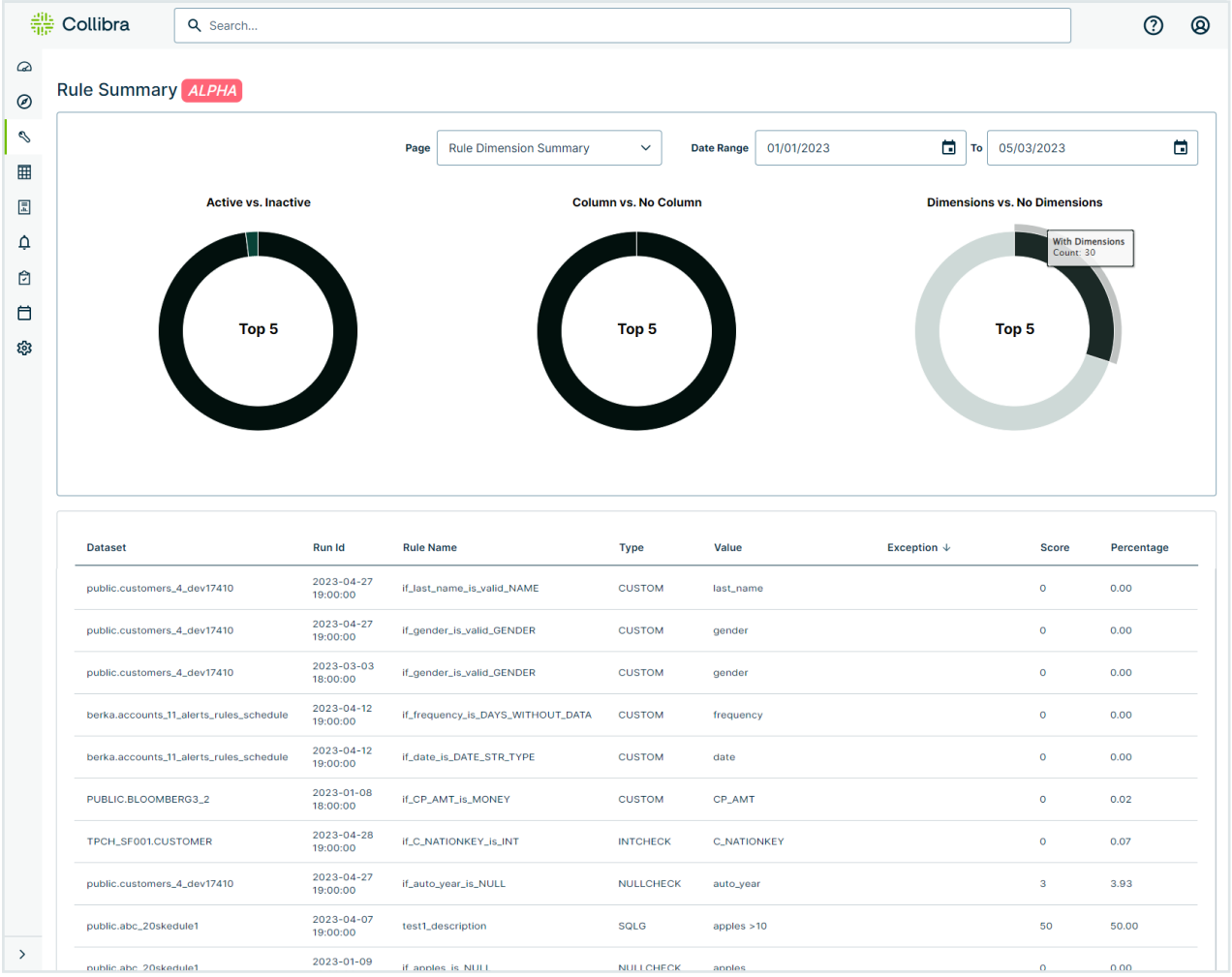


Dataset	Run Id	Rule Name	Type	Value	Exception	Score	Percentage
berka.accounts_8	2022-09-07 19:00:00	r2	SQLF	select * from berka.accounts_8 where account_id is null	Exception with freeform sql rule check: Table or view not found: berka.accounts_8; line 1 pos 14; 'Project [*] +- 'Filter isnull('account_id) +- 'UnresolvedRelation [berka, accounts_8], [], false	0	0.00
berka.accounts_9	2022-09-07 19:00:00	r1	SQLF	select * from berka.accounts_9 where account is not null	Exception with freeform sql rule check: Table or view not found: berka.accounts_9; line 1 pos 14; 'Project [*] +- 'Filter isnotnull('account) +- 'UnresolvedRelation [berka, accounts_9], [], false	0	0.00
berka.accounts_9	2022-09-07 19:00:00	r2	SQLF	select * from berka_accounts_9_dataset where account is not null		27	27.96
berka.transactions_postgresgcp	2022-10-12 19:00:00	transaction_type_unique	SQLG	9 != 1056320		100	100.00
berka.transactions_postgresgcp	2022-10-12 19:00:00	pgcp_sqlg_1	SQLG	account_id > '3834.0'		21	21.34
berka.transactions_postgresgcp	2022-09-20 19:00:00	transaction_type_unique	SQLG	9 != 1056320		100	100.00

Rule Dimension Summary

The Rule Dimension Summary shows summaries of rule activity, rules with columns, and rules with dimensions in three charts:

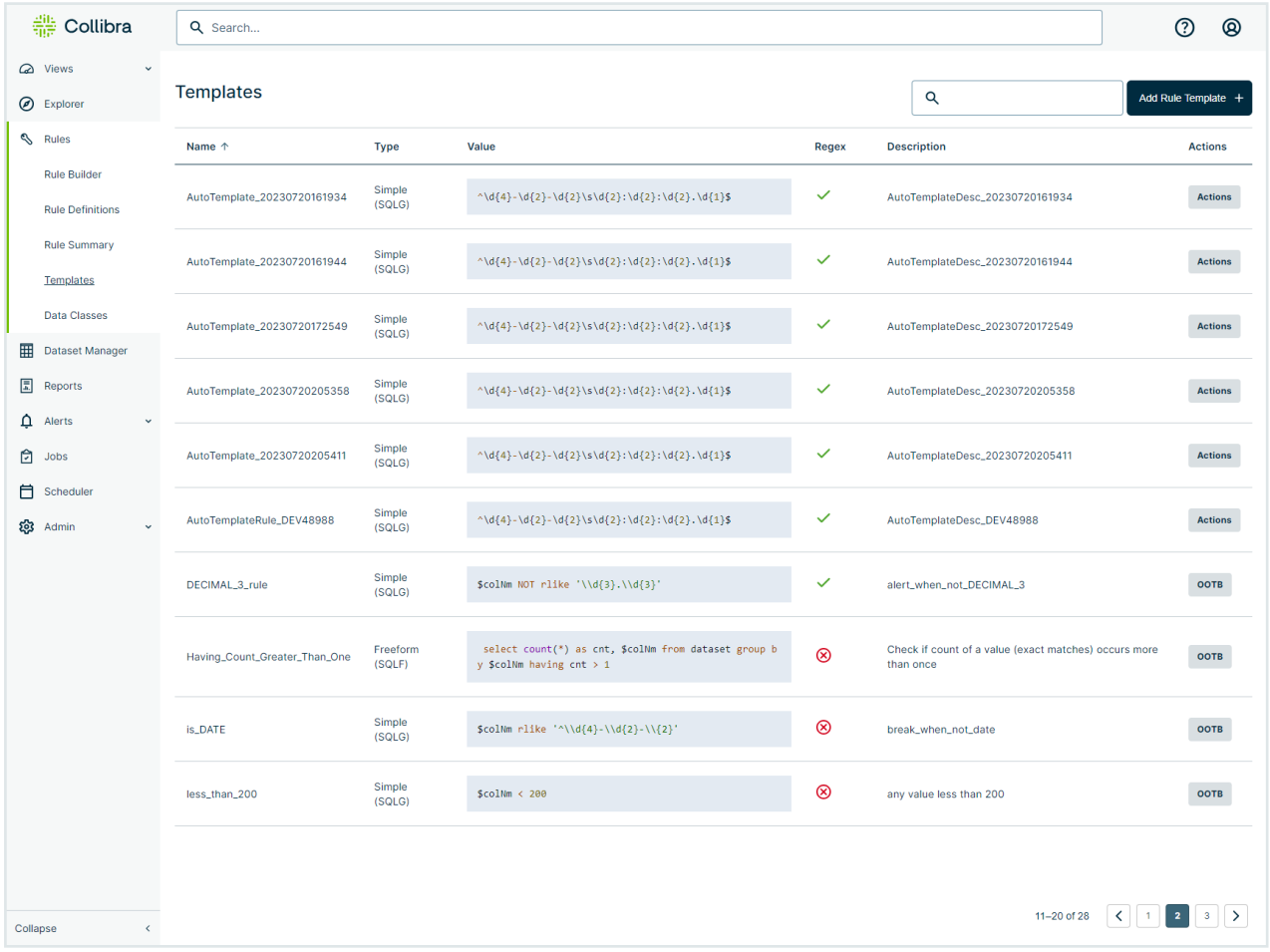
Chart	Description
Active vs. Inactive	Shows the number of active and inactive rules on all of your datasets over a period of time. Hover your cursor over the donut chart to view the details.
Column vs No Column	Shows the number of rules with columns. Hover your cursor over the donut chart to view the 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.



Dataset	Run Id	Rule Name	Type	Value	Exception ↓	Score	Percentage
public.customers_4_dev17410	2023-04-27 19:00:00	if_last_name_is_valid_NAME	CUSTOM	last_name		0	0.00
public.customers_4_dev17410	2023-04-27 19:00:00	if_gender_is_valid_GENDER	CUSTOM	gender		0	0.00
public.customers_4_dev17410	2023-03-03 18:00:00	if_gender_is_valid_GENDER	CUSTOM	gender		0	0.00
berka.accounts_11_alerts_rules_schedule	2023-04-12 19:00:00	if_frequency_is_DAYS_WITHOUT_DATA	CUSTOM	frequency		0	0.00
berka.accounts_11_alerts_rules_schedule	2023-04-12 19:00:00	if_date_is_DATE_STR_TYPE	CUSTOM	date		0	0.00
PUBLIC.BLOOMBERG3_2	2023-01-08 18:00:00	if_CP_AMT_is_MONEY	CUSTOM	CP_AMT		0	0.02
TPCH_SF001.CUSTOMER	2023-04-28 19:00:00	if_C_NATIONKEY_is_INT	INTCHECK	C_NATIONKEY		0	0.07
public.customers_4_dev17410	2023-04-27 19:00:00	if_auto_year_is_NULL	NULLCHECK	auto_year		3	3.93
public.abc_20skedule1	2023-04-07 19:00:00	test1_description	SQLG	apples >10		50	50.00
public.abc_20skedule1	2023-01-09	if_apples_is_NULL	NULLCHECK	apples		0	0.00

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.



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.

Adding a Template Rule

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 to easily identify your template rule.	✓ 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. 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 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.

Name	Value	Description
is_DATE	`$colNm rlike '^\\d{4}-\\d{2}-\\{2}'`	The rule breaks when a column value is not in YYYY-MM-DD format.
less_than_200	`$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.
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.

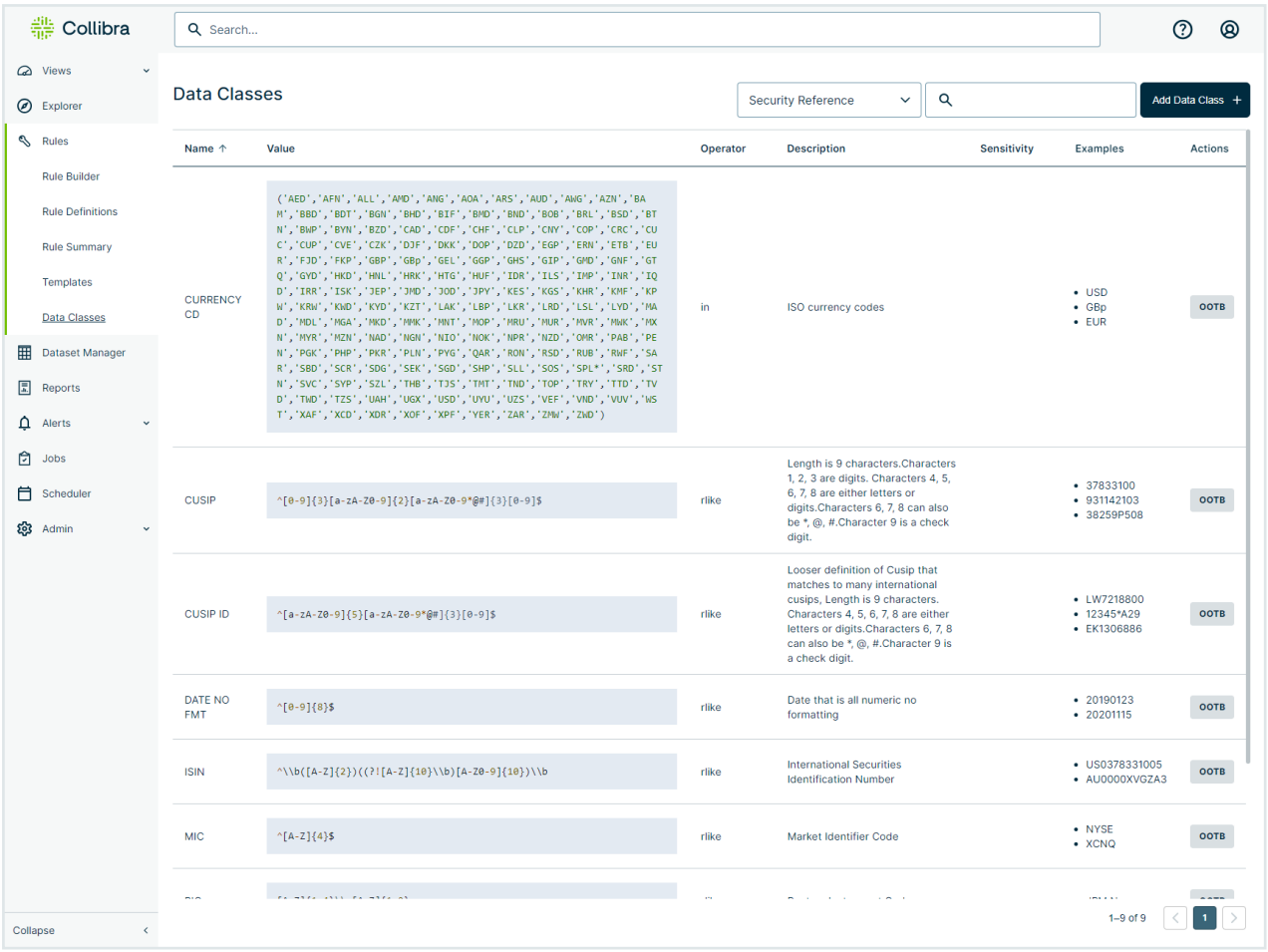
Name	Value	Description
Percent_Move_20	`select i.$colNm, s.averg from dataset i , (select avg (cast($colNm as double ) ) as averg from dataset ) s where i.$colNm < (s.averg * 0.80 ) or i.$colNm > (s.averg * 1.20)`	Checks if there are column values present that are 20% greater or less than the average value in that column.
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.

Name	Value	Description
Two_Decimal_Places	`select d[1] as rs from ( select SPLIT ($colNm, '\\.') as d from dataset ) where length(d[1]) > 2`	Checks if the column decimal place precision exceeds a length of 2.
USTAX_rule	`$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
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

Option	Description	Required
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

Collibra Data Quality & Observability 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 #. Charac

Name	Value	Description
		ter 9 is a check 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.

Name	Value	Description
PASSWORD MEDIUM	`^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%^&+=])(?=\\S+$).{8,20}$`	Checks password values containing 1 number, 1 letter, and 1 uppercase letter.
PERCENT	`^\\d+(?:\\.\\d+)?%$`	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.

Name	Value	Description
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.
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.

Name	Value	Description
SEDOL	`^(?<Sedol>[B-Db-dF-Hf-hJ-Nj-nP-Tp-tV-Xv-xYyZz\\d]{6}\\d)$`	Validates that the alphanumeric value is a valid Stock Exchange Daily Official List (SEDOL) code, such as B12345-6.
SHIRT SIZE	`('L','M','S','XS','XL','XXL','2XL','3XL','l','m','s','xs','xl','xxl','Small','Medium','Large','X-Large')`	Validates that a value contains a valid shirt size.
SSN	`^(?!(000|666|9))\\d{3}-(?!00)\\d{2}-(?!0000)\\d{4}$`	3 digits dash 2 digits dash 4 digits.

Name	Value	Description
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.
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.

Name	Value	Description
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.
SUFFIX	`('Jr','Jr.', 'Sr', 'Sr.', 'II', 'III', 'IV')`	Validates that a value contains a valid suffix after surnames, such as Jr., Sr., and II.

Name	Value	Description
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.
TIME HHMMSS	`^(\\d\\d:\\d\\d:\\d\\d)`	Time in a HHMMSS format, such as 23:59:59 or 08:30:00.

Name	Value	Description
UK POSTAL CD	`^[A-Z]{1,2}[0-9R][0-9A-Z]? [0-9][ABD-HJLNP-UW-Z]{2}$`	Checks that a string value is in a valid UK postal code format.
URL	`(http|https|www.|ftp)(www.)?[a-zA-Z0-9@:%._\\+~#?&//=]{2,256}\\.[a-z]{2,6}\\b([-a-zA-Z0-9@:%._\\+~#?&//=]*)`	Checks that a string value is in a valid website URL format.
VIN	`^(?<wmi>[A-HJ-NPR-Z\\d]{3})(?<vds>[A-HJ-NPR-Z\\d]{5})(?<check>[\\dX])(?<vis>(?<year>[A-HJ-NPR-Z\\d])(?<plant>[A-HJ-NPR-Z\\d])(?<seq>[A-HJ-NPR-Z\\d]{6}))$`	Checks if a value does not match a US Vehicle Identification Number format.

Name	Value	Description
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.

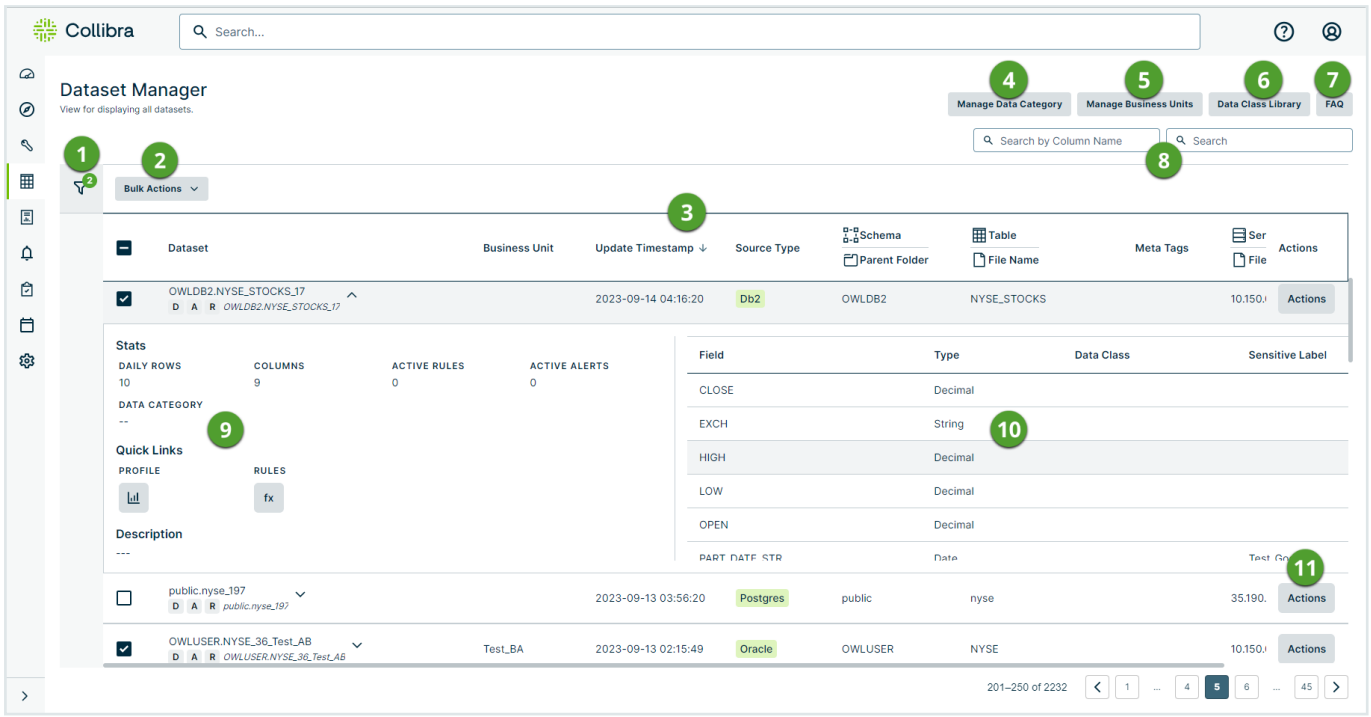
Dataset Directory

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.

About Dataset Manager



No.	Component	Description
1	Filter	Apply one or many filters to refine the list of datasets based on sensitivity labels, run mode, connection type (Pushdown or Pullup), data source type, data class type, business units, data categories, row count, column count, integration status, and number of DQ scans.
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. These options include Bulk Manage Business Units, Bulk Manage Data Categories, and Bulk Delete.

<table>
<tr><th>No.</th><th>Component</th><th>Description</th></tr>
<tr><td>3</td><td>Dataset table</td><td>This table lists all datasets in your Collibra DQ environment and provides the high-level data points described in the following table.
<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>☐</td><td>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.</td></tr>
<tr><td>Dataset</td><td>The name of your dataset. You can also see the following in this column:
When a dataset is in draft status, a D displays beneath the dataset name.
When a dataset is in published status, a P displays beneath the dataset name.
When a dataset does not have alerts configured, a A displays beneath the dataset name.
When a dataset has alerts configured, a A displays beneath the dataset name.
When a dataset does not have rules configured, a R displays beneath the dataset name.
When a dataset has rules configured, a R displays beneath the dataset name.
Click the V to the right of the dataset name to drill down into the dataset.</td></tr>
<tr><td>Business Unit</td><td>The business unit with which the dataset is associated.</td></tr>
<tr><td>Update Timestamp</td><td>The timestamp of the last run of the dataset in YYYY-MM-DD hh-mm-ss format.</td></tr>
<tr><td>Source Type</td><td>The data source of the dataset. For example, Postgres or BigQuery.</td></tr>
<tr><td>Schema/Parent Folder</td><td>The schema or parent folder from which the dataset originates.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Component</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Table/File Name</td><td>The table or file name from which the dataset was created.</td></tr>
<tr><td>Meta Tags</td><td>The meta tags with which the dataset is associated. Optionally click Actions ▸ Edit, then enter meta tags in the Meta Tags input fields.</td></tr>
<tr><td>Server/File Path</td><td>The server or file path associated with the origin of your dataset.</td></tr>
<tr><td>Actions</td><td>Click Actions to take a variety of actions on your dataset. Available actions include:
Edit 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 Data Intelligence Cloud.
Delete your dataset.</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>Manage Data Category</td><td>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.</td></tr>
<tr><td>5</td><td>Manage Business Units</td><td>Opens the Business Units page in the Admin Console where you can create and manage business units.

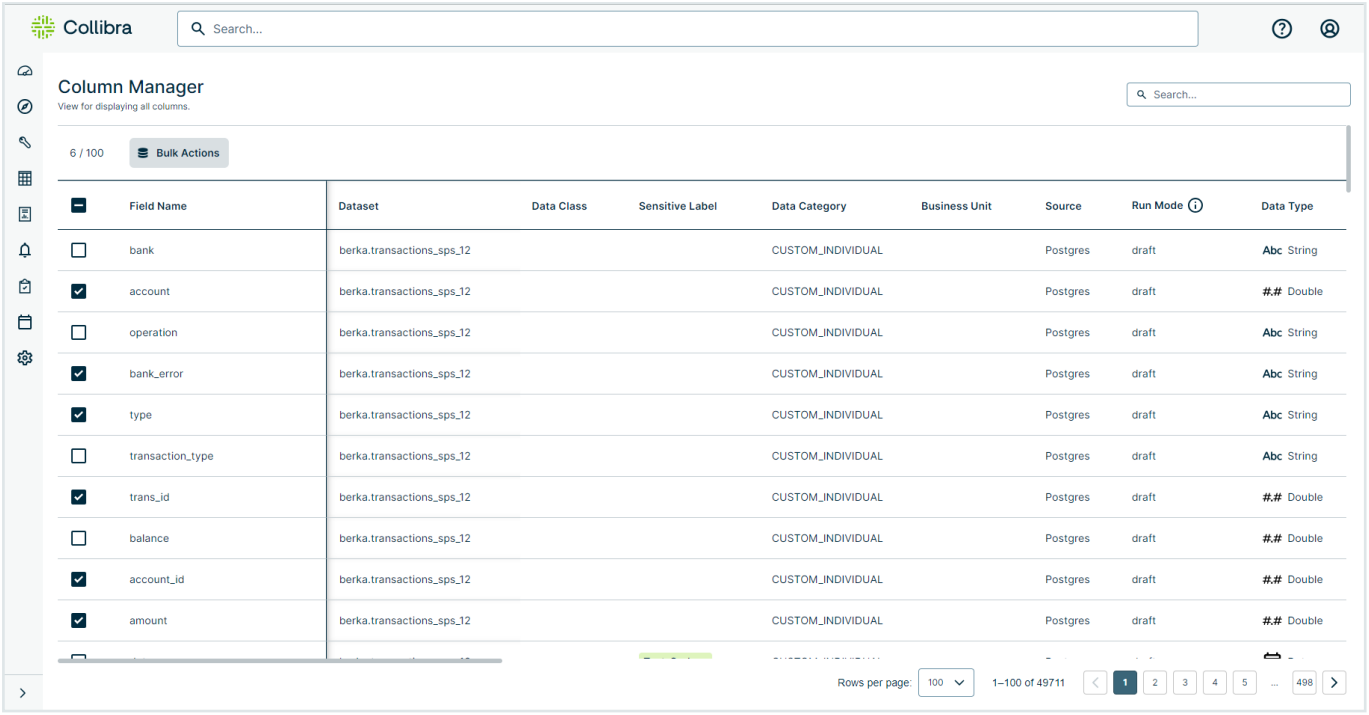
Note ROLE_ADMIN or ROLE_DATA_GOVERNANCE_MANAGER is required to access the Admin Console.</td></tr>
<tr><td>6</td><td>Data Class Library</td><td>Opens the Data Class Library modal where you can view and search the data classes in your Collibra DQ environment.</td></tr>
</table>

<table>
<tr><th>No.</th><th>Component</th><th>Description</th></tr>
<tr><td>7</td><td>FAQ</td><td>Opens the Top Questions and Answers modal where a sample of common dataset management questions appear.</td></tr>
<tr><td>8</td><td>Search</td><td>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.</td></tr>
<tr><td>9</td><td>Dataset drill down</td><td>Click the V to the right of the dataset name to drill down into the dataset and reveal the following high-level data points.
<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td colspan="2">Stats</td></tr>
<tr><td>Daily Rows</td><td>The number of rows scanned in the last run.</td></tr>
<tr><td>Columns</td><td>The number of columns in the dataset.</td></tr>
<tr><td>Active Rules</td><td>The number of active rules included in the last run.</td></tr>
<tr><td>Active Alerts</td><td>The number of alerts configured to notify specified users when their conditions were met in the last run.</td></tr>
<tr><td>Data Category</td><td>The data category assigned to your dataset on the Rule Workbench or Metadata Bar.</td></tr>
<tr><td colspan="2">Quick Links</td></tr>
<tr><td>Profile</td><td>Click the to open the Profile page of your dataset.</td></tr>
<tr><td>Rules</td><td>Click the to open the Dataset Rules page of your dataset.</td></tr>
<tr><td>Description</td><td>The description of your dataset given to it on the Explorer ▸ Review page.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>No.</th><th>Component</th><th>Description</th></tr>
<tr><td>10</td><td>Column overview</td><td>This section of the dataset drill down shows the following column-level details of all columns in your dataset.<table><tr><th>Column</th><th>Description</th></tr><tr><td>Field</td><td>The name of the column.</td></tr><tr><td>Type</td><td>The data type of the column.</td></tr><tr><td>Data Class</td><td>The data class labels of a column, when applied.</td></tr><tr><td>Sensitive Label</td><td>The sensitive label to obscure sensitive information, when applied.</td></tr></table></td></tr>
<tr><td>11</td><td>Actions</td><td>Click Actions to take a variety of actions on your dataset. Available actions include:
Edit 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 Data Intelligence Cloud.
Delete your dataset.</td></tr>
</table>

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.

Column	Description
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.
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.

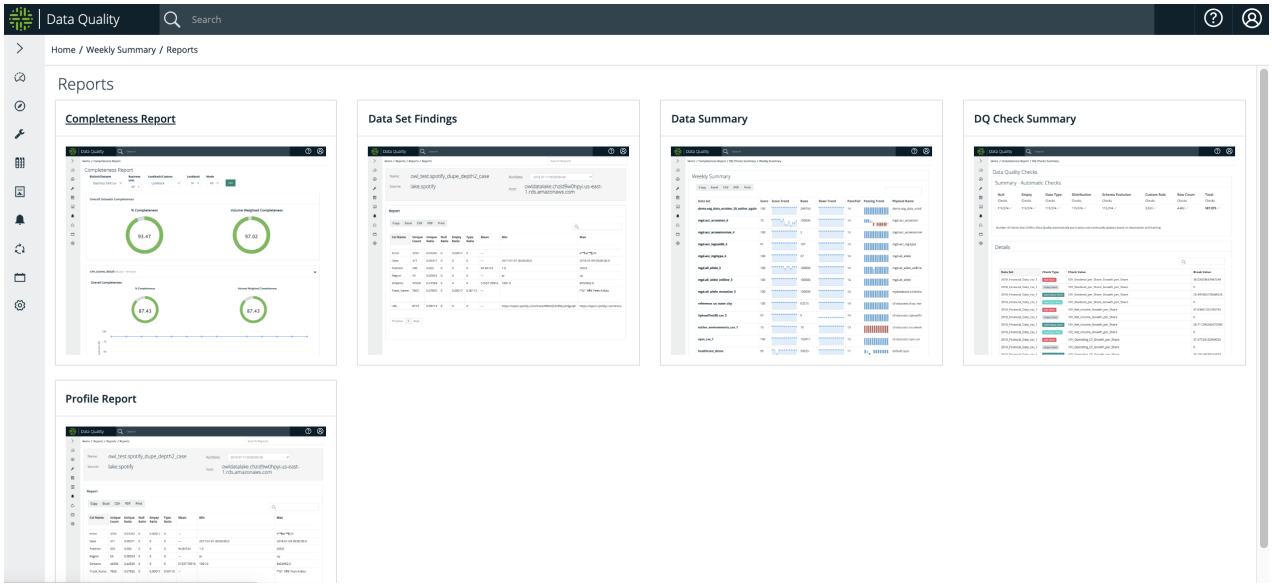
4. Click **Apply**.
   » The column with the sensitive label will be is flagged as containing sensitive information and a badge displays in that column's row on Column Manager.

Chapter 8

Reports

Built-In

Collibra DQ provides various out of the box reports that allow you to access insights quickly.



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.

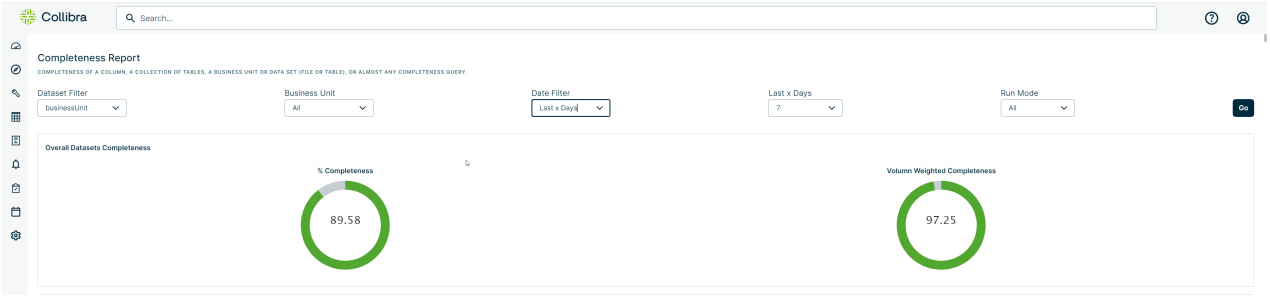
- **Custom Range**: the custom range you set for Run Date/Update Time.
    i. From the Date Column dropdown list, select one of the following options:
        - **Run Date**: the date/time the data represents.
        - **Update Time**: time the DQ job ran.
    ii. In the Date Range fields, select a 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 data sets 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

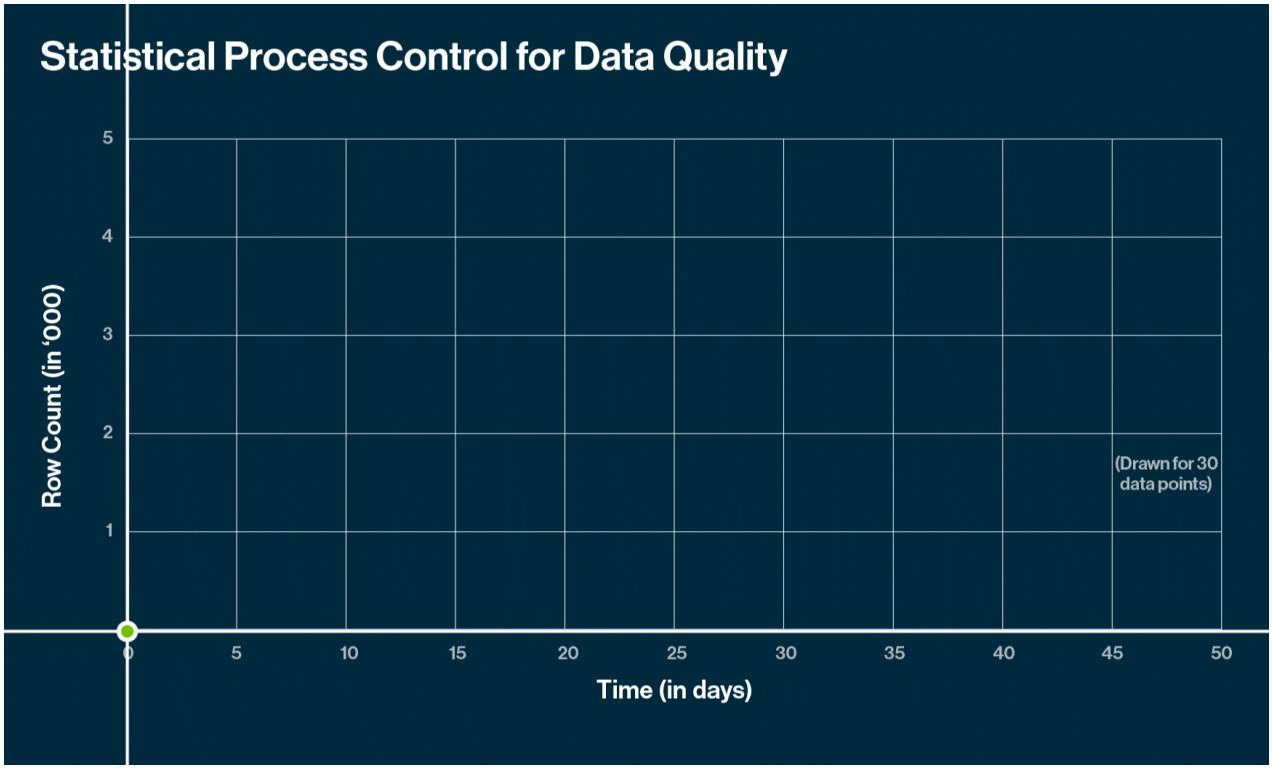
The column view shows the completeness of specific data sets, which makes it easy to see the columns that are least complete and, therefore, of possible concern. The columns range from 0% to 100% complete.

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

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 Collibra DQ Job Back Run and Profile (automatic) features.



Coverage Report

What is a Data Quality Coverage Report

The Coverage Report provides a view that shows DQ coverage across all your technical data sets (for example, schemas, files, tables, and items in the database) via a donut chart, as well as time-series bar charts that show the DQ run level information aggregated by month.

You can generate a Coverage 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. In Connections, select one of the databases, for which you want to generate a report and click **Generate Report**. The Coverage Report displays. The report includes a

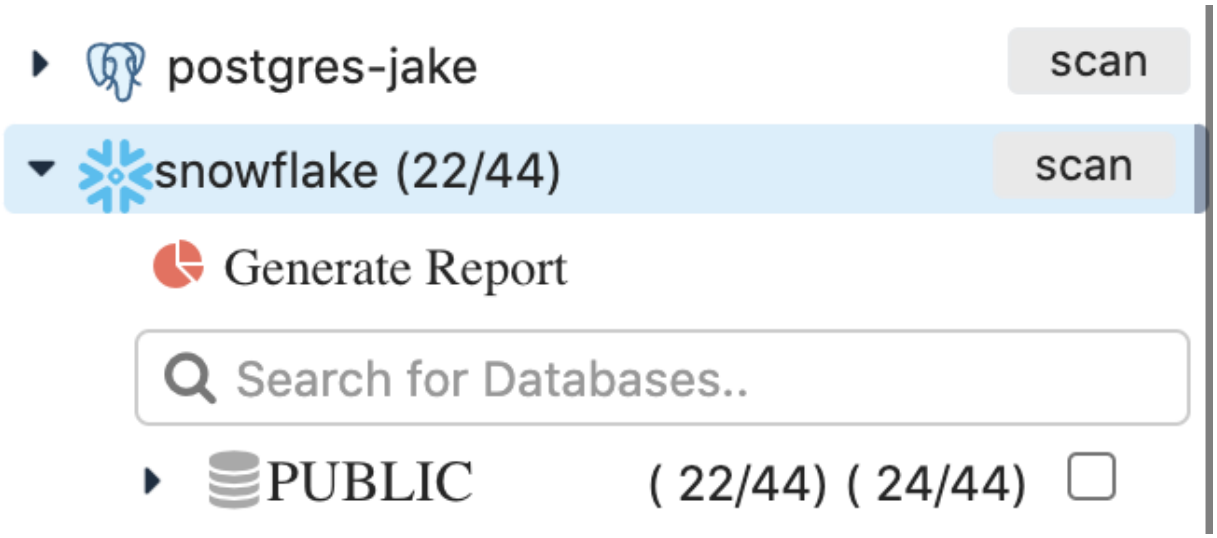
donut chart showing the percentage of all schemas in the database that are running data quality on them and interactive bar charts showing which data sets are running data quality jobs on a monthly basis.

3. To see how many new data quality jobs have been added, click one of the following bar charts (the green color represents new DQ jobs and the gray are the existing jobs):

View	Description
1m	Shows new and existing jobs after one month.
3m	Shows new and existing jobs after three months.
6m	Shows new and existing jobs after six months.
YTD	Shows new and existing jobs year-to-date.
1y	Shows new and existing jobs after one year.
All	Shows new and existing jobs for the entire range of months.

4. To generate a Coverage Report for a specific schema, expand the schema and click **Generate Report**. The Coverage Report displays for that schema. You can click into the charts to see specific information for DQ coverage for this schema.

   In the following screenshot, `(22/44)` next to the PUBLIC schema represents 22 out of 44 tables that have DQ jobs running on them and `(24/44)` represents 24 out of 44 jobs that have been cataloged, meaning they are registered and have metadata.



Dataset Findings

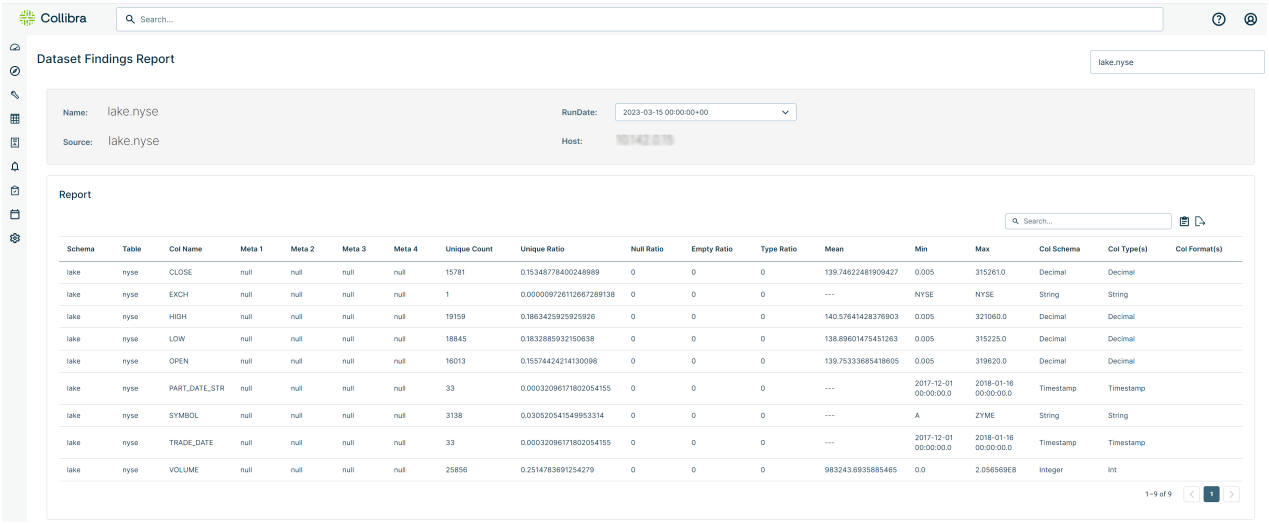
What is the Dataset Findings Report?

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

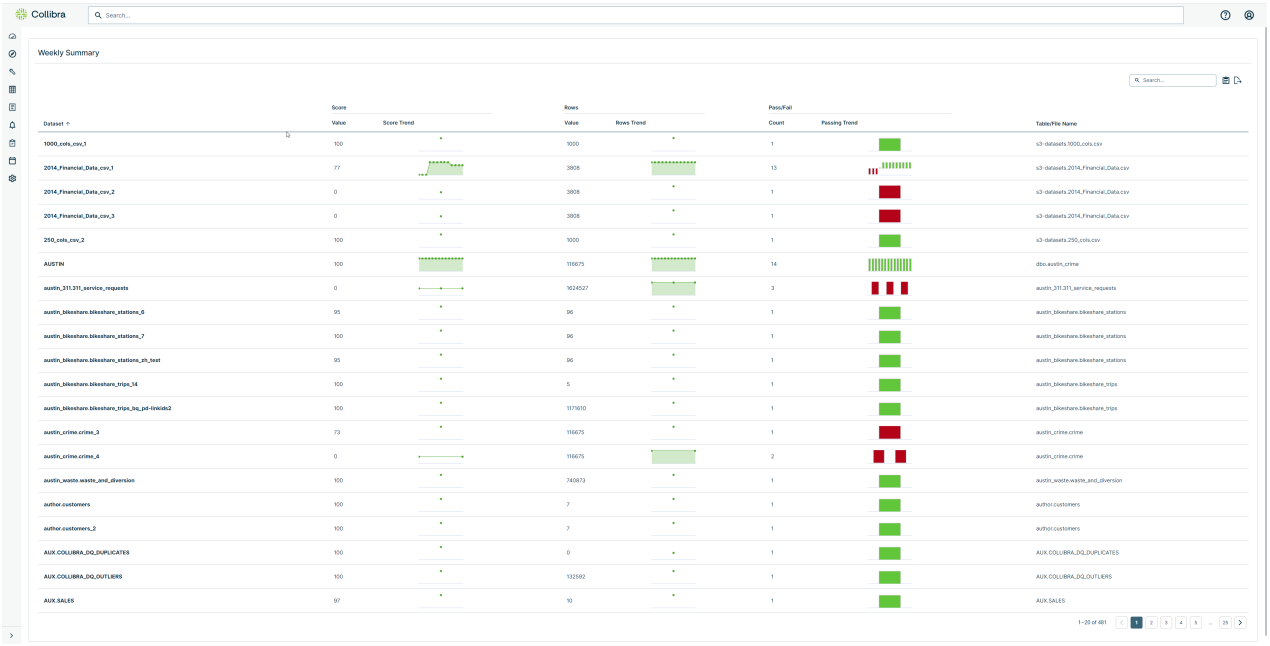
Collibra Data Quality Summary reports provide information about your datasets rolled up in high level summaries. These reports include the What is the Weekly Summary Report? and What is the Data Quality Checks Report?.

> 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.

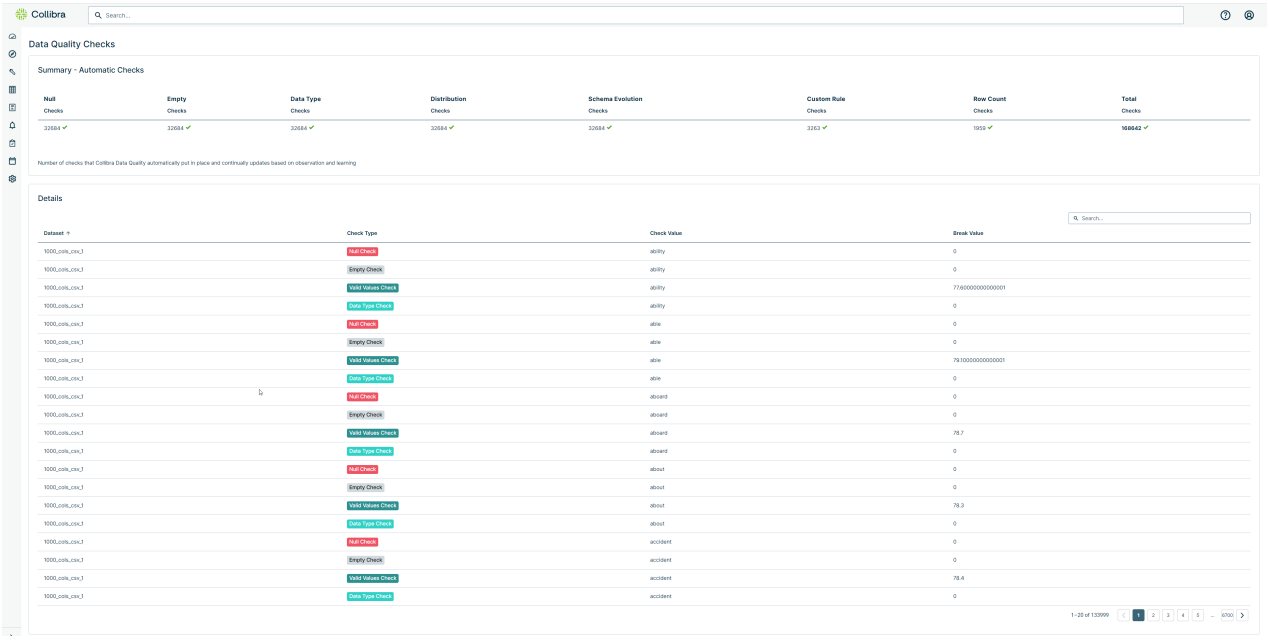
Column	Description
Score Trend	A line graph representation of data quality scores over one week. Hover over the blue dots to see the score trend.
Rows	The total number of rows of a particular dataset scanned over one week.
Rows Trend	A line graph representation of row count over one week. Hover over 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.

Profile Report

What is the Profile Report

The Profile Report allows you to search for a particular data set and generate, view, copy, print, and export the report to an Excel or CSV file format.

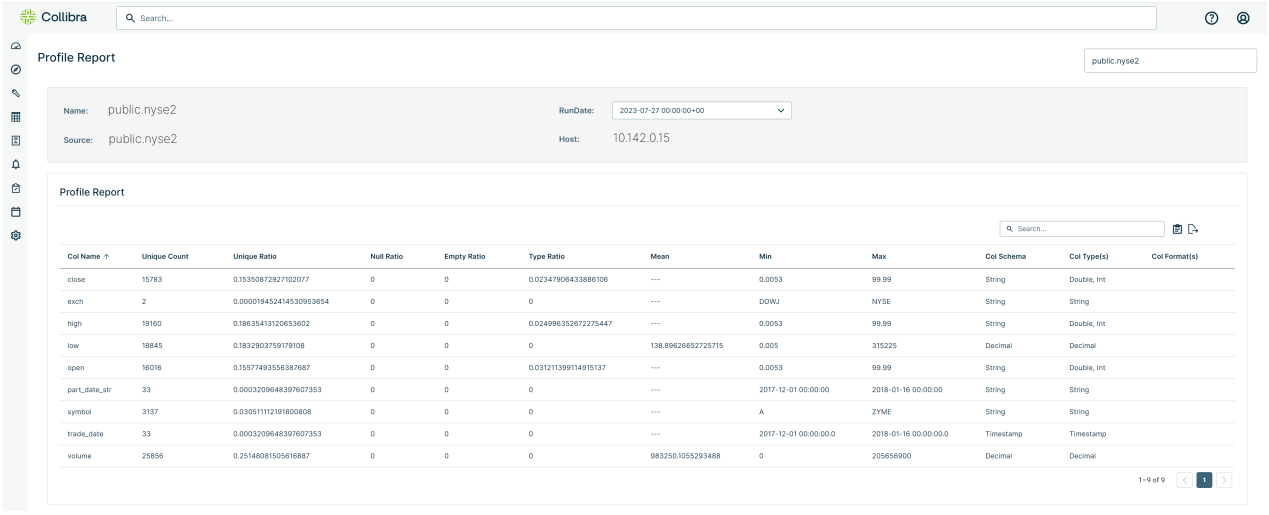
> Note As of the 2022.08 release, PDF is no longer a supported file format for exporting and printing reports. These functions are now restricted to the CSV file format only.

Steps

To generate a Profile 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 **Profile** link.
3. In the search bar in the upper right corner, enter the name of your target dataset.
   A list of available data sets populates in the dropdown list.
4. Select the **dataset** for which you want to view and export results.
   The Profile Report page displays for your dataset.
   The report includes the name, source, RunDate, 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 Profile Report.



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 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 (beta)

What is the Column Dimension Dashboard?

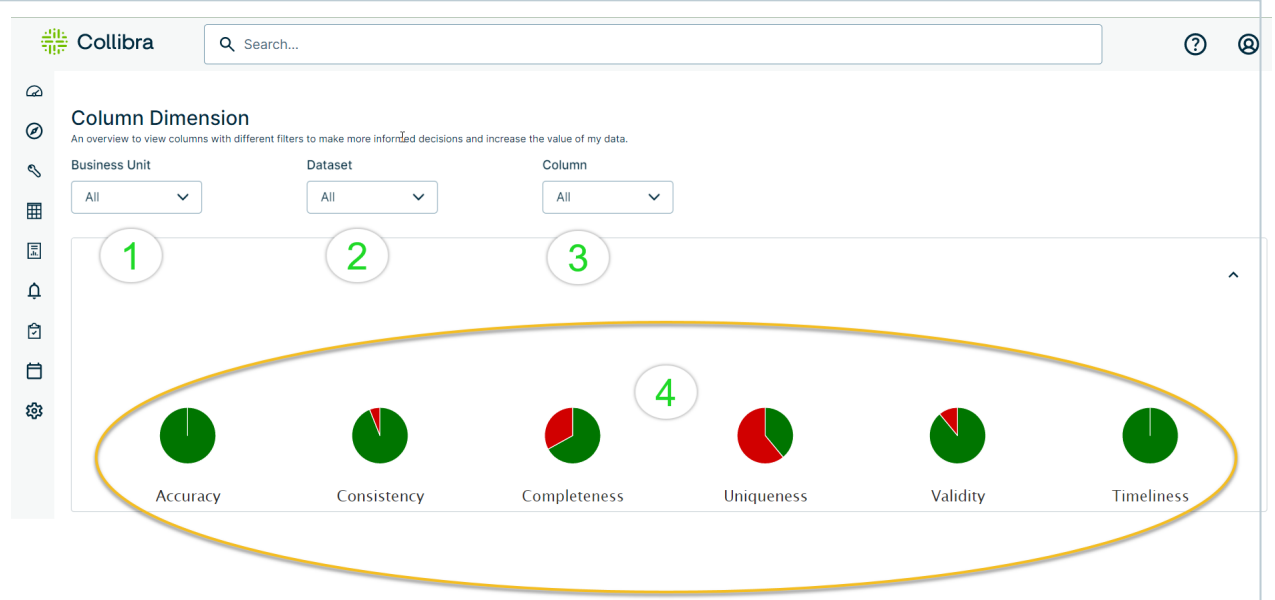
The Column Dimension Dashboard provides a way to view and filter Collibra DQ dimension scores by business units, datasets, and column. This dashboard is essentially an overview of all business units, datasets, and column in the Collibra DQ application, which provides insights about the total current DQ Dimension scores, Dimension scores over time, and DQ scores (as well as other metadata) for each column.

> Note To access the Column Dimension dashboard, contact the Collibra DQ Product team or your CSM.

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.

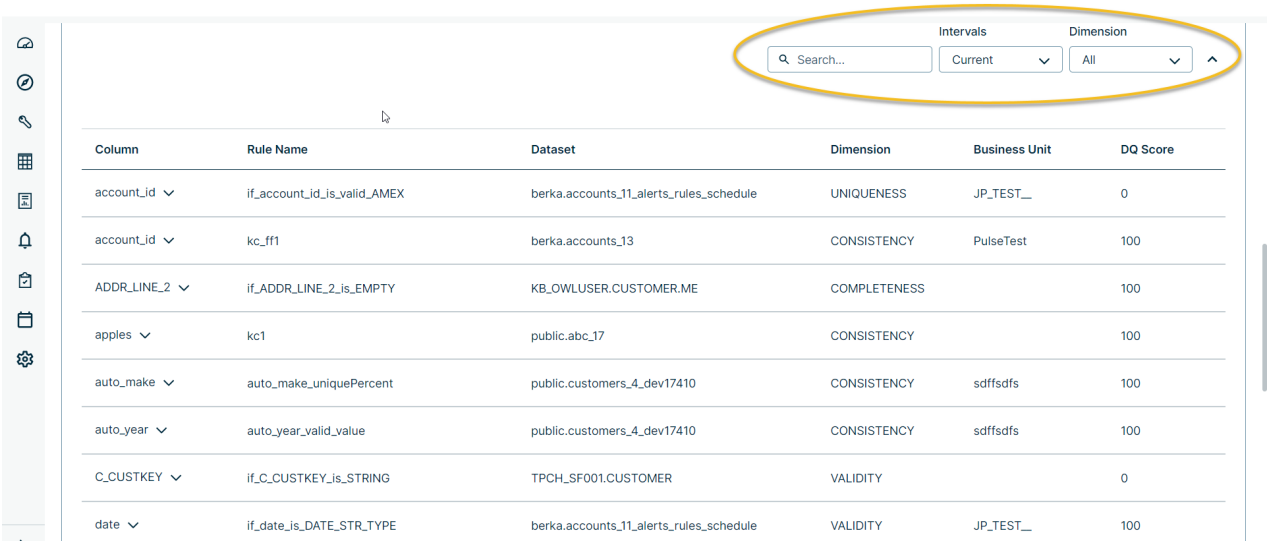
4	Dimension results pie charts.	Charts that display dimensions based on the criteria you select from the dropdown lists. These charts measure the accuracy, consistency, completeness, uniqueness, validity, and timeliness of your data. Tip You can hover over a chart to view the percentages for that specific 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 (beta)

What is the Dataset Dimension Report?

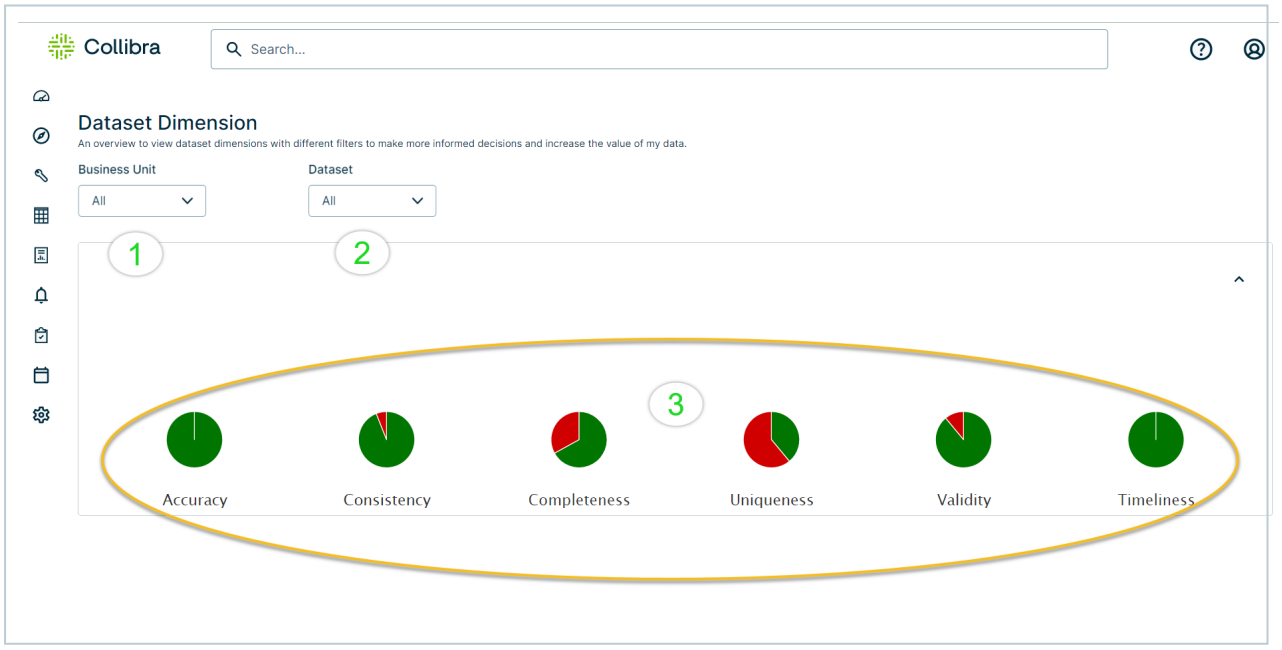
The Dataset Dimension Report provides a way to view and filter Collibra DQ dimension scores by datasets and business units. This report is essentially an overview, which provides insights about averaged or aggregated scores across datasets and business units in the Collibra DQ application.

> Note To access the Dataset Dimension report, contact the Collibra DQ Product team or your CSM.

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 ⌃.

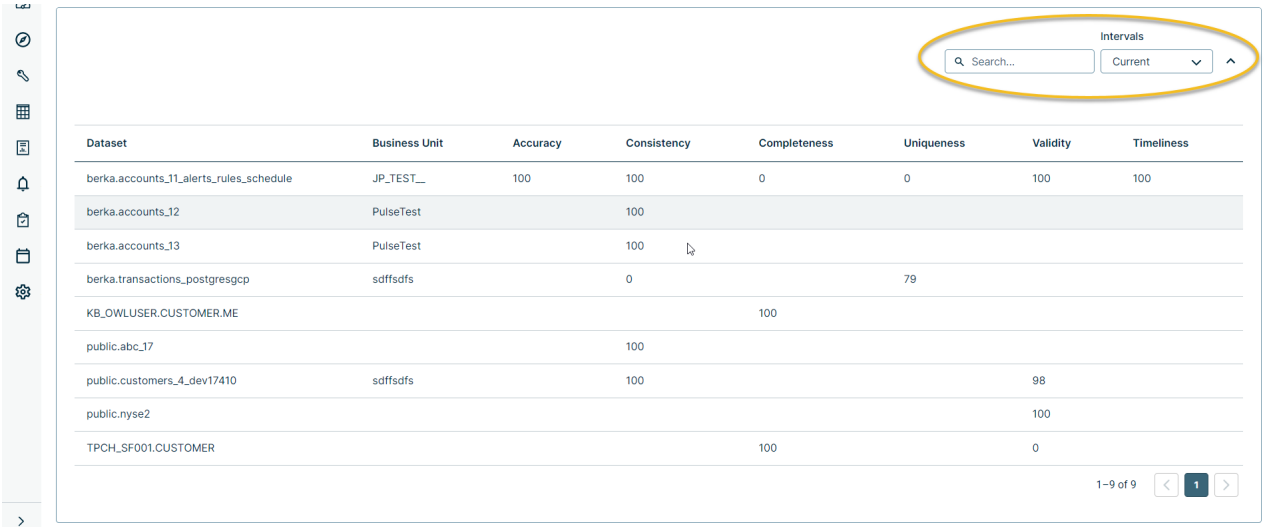


1	Dimension results by business unit.	Select a business unit from the **Business Unit** dropdown.
2	Dimension results by dataset.	Select the dataset from the **Dataset** dropdown.
3	Dimension results pie charts.	Charts that display dimensions based on the criteria you select from the dropdown lists. 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

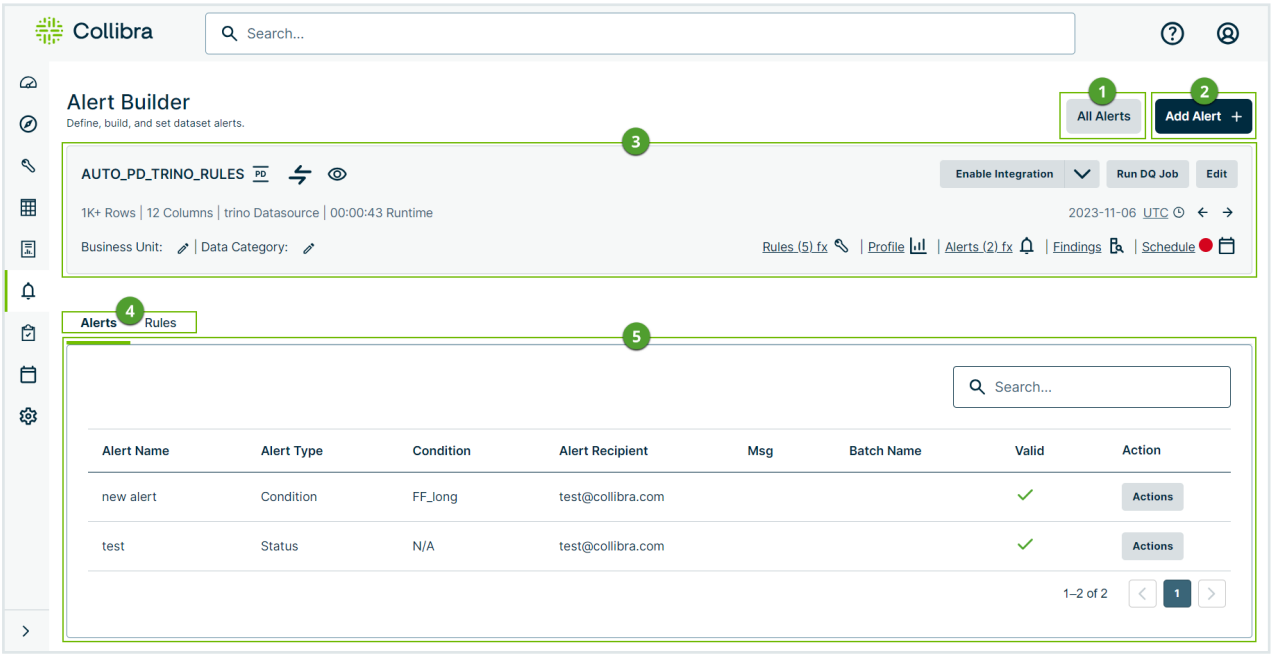
```
SELECT table_name FROM information_schema.tables WHERE table_schema
= 'public' ORDER BY table_name
```

Alerts

This section provides information about how to configure alerts in the Collibra DQ application.

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 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.
Valid	The validity of the alert. If the alert is valid, a check mark displays in this column.
Action	Click **Actions** and then select one of the following options from the dropdown menu: **Edit** Modify the values of the selected alert. **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.
Dataset	The name of the selected dataset.

Column	Description
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 section shows you how to configure dataset alerts.

Steps

1. Sign into your Collibra DQ instance.
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, either the Condition Alert or Status Alert modal appears.
6. Enter the required information.

   A dataset run 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.
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
Batch Name	Enter a batch name to send the alert to multiple recipients.

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".
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.
Batch Name	Enter a batch name to send the alert to multiple recipients.

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".
Custom Msg	A custom message relevant to the alert.

7. Click **Save**.

Configuring Alert Emails

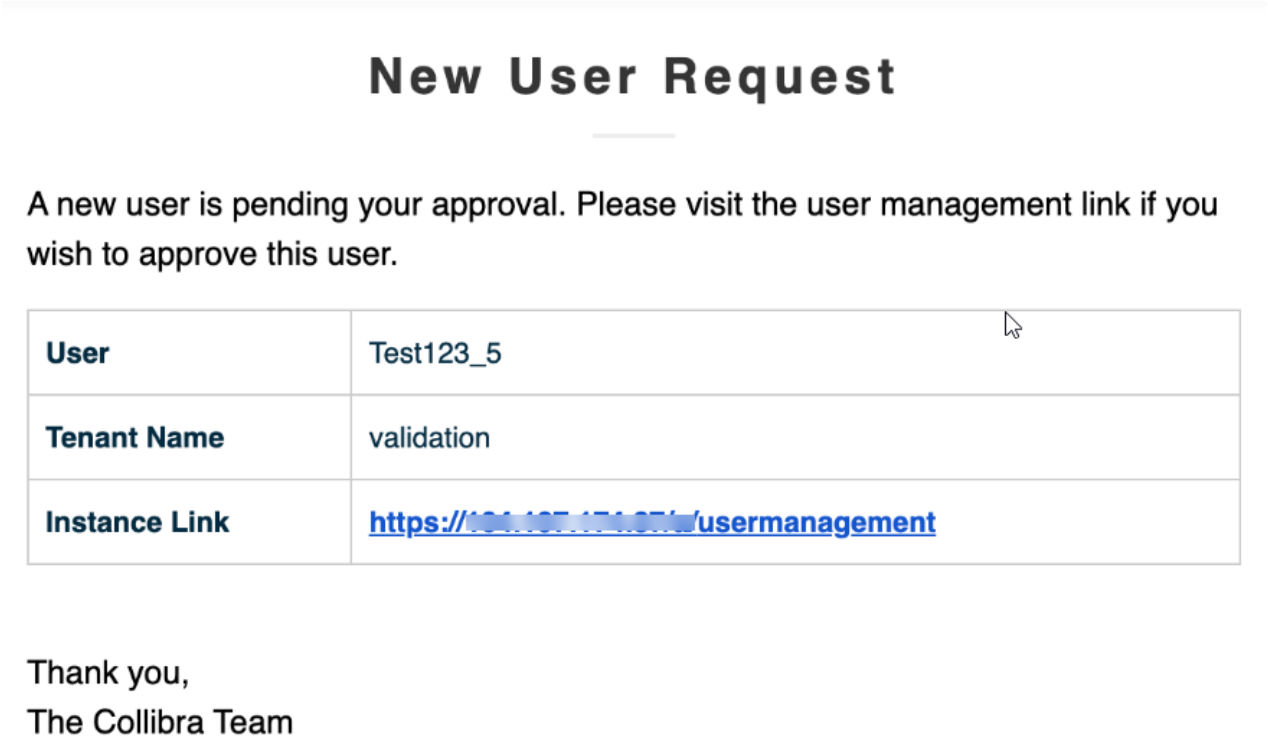
Email alerts let you send emails to specified recipients when an alert condition is met for a given dataset.

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

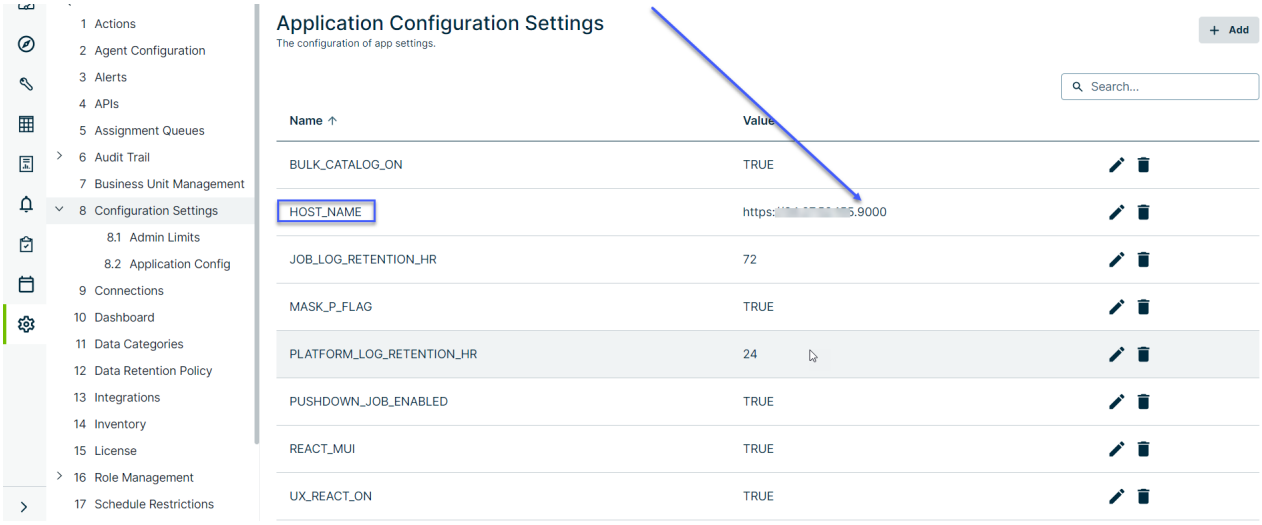
A new user is pending your approval. Please visit the user management link if you wish to approve this user.

User	Test123_5
Tenant Name	validation
Instance Link	https://[redacted]/usermanagement

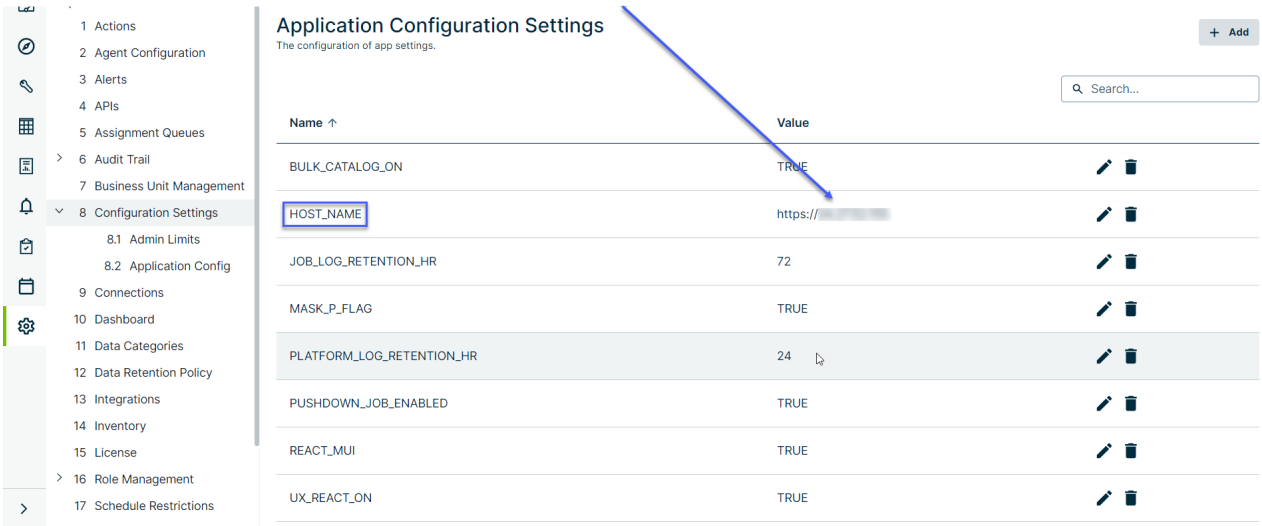
Thank you,
The Collibra Team

> 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 in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Click **Settings**.
   The Admin Limits page opens.
3. Select **+ Add** in the upper right corner.

> Note All fields in the Add a Custom Configuration page are required.

4. Enter a property in the **Name** field.
5. Click **Description** to enter a description for this setting.
6. Click **Value** to enter a value.
7. Click **Tenant** to enter a tenant.

Property	Default Value	Description
mail.smtp.auth	True	When set to `TRUE`, the server attempts to authenticate the user using the `AUTH` command. When set to `FALSE`, username and password authentication are turned off.
mail.smtp.starttls.enable	True	When 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. When TLS is not supported by your mail server, this property must be set to `FALSE`.
mail.smtp.ssl.trust	*	If set, and a socket factory hasn't been specified, this property enables use of a MailSSLSocketFactory. If the value is set to "*", all hosts are trusted. If, however, 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 properties are preset to their default values. For example, mail.stmp.auth is preset to `TRUE`.

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 10

Scheduler

Running a Full Scan ..... mxx

Running a Partial Scan ..... mxxi

Scanning with SQL Query ..... mxxii

Schedule a job

After you successfully run a job, you can schedule it to run automatically. If your monthly or quarterly jobs typically load after the month or quarter has ended, you can schedule the job for the day when the data lands, but set the offset to the proper run date required for charting or reporting.

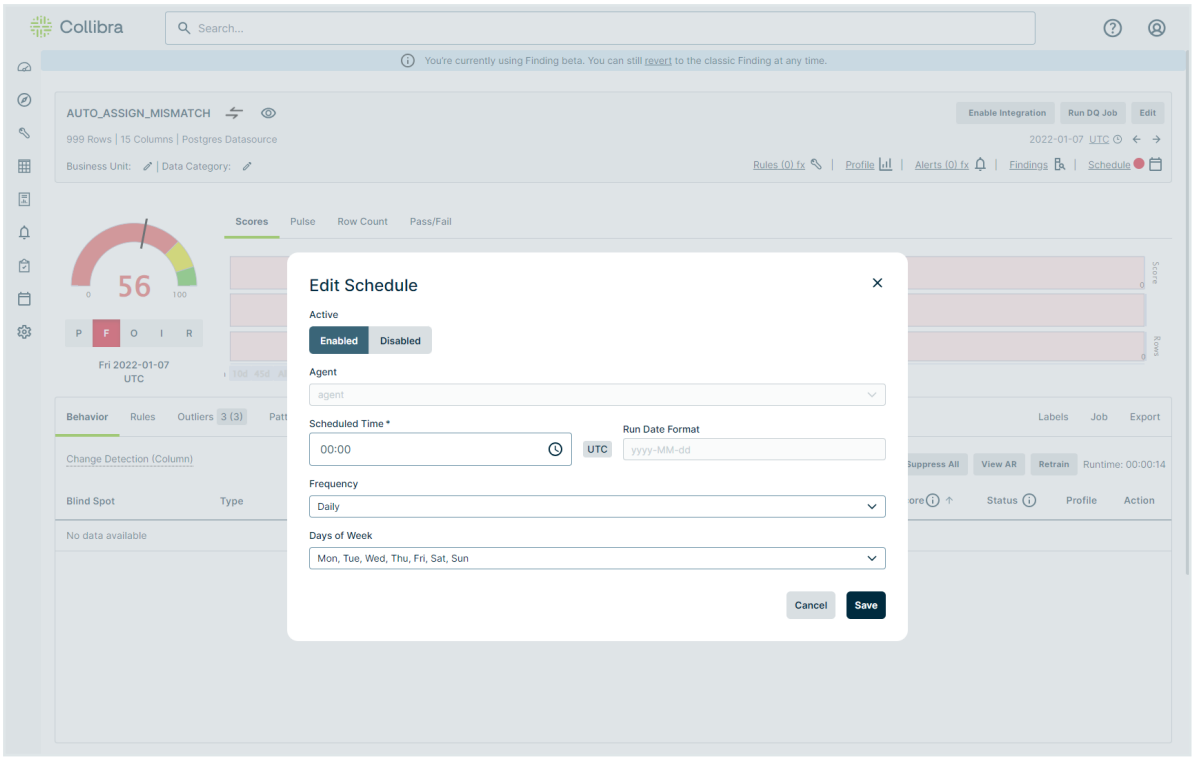
Steps

1. On the right side of the metabar of a dataset, click **Schedule**.
   » The Edit Schedule modal appears.
2. Enter or select from the options.

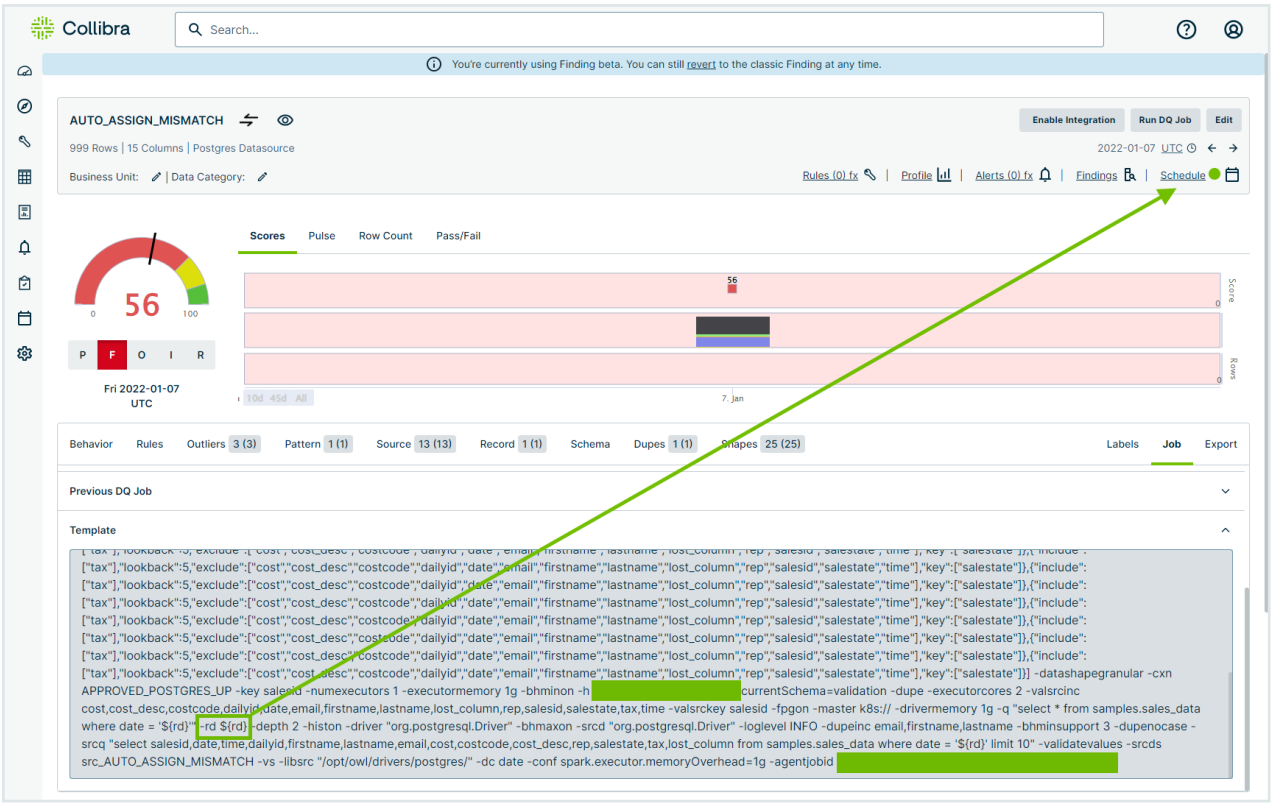
Option	Description
Active	Enables or disables the scheduled runs of a job. Click **Enabled** to enable jobs to run according to the schedule, or **Disabled** to disable them.
Agent	The agent used to submit your job for processing. This field is not editable.
Schedule Time	Enter a time in 24-hour clock format, or use the clock icon to click and drag a time.
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 job should run.
Daily	Select this option from the Frequency dropdown menu to run your 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 job should run. Select or deselect all that apply.
Monthly	Select this option from the Frequency dropdown menu to run your job every month at a specified time and date.
Days of Month	The day of the month when your scheduled job runs.

Option	Description
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 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 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 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 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 job, as shown in the following screenshot.



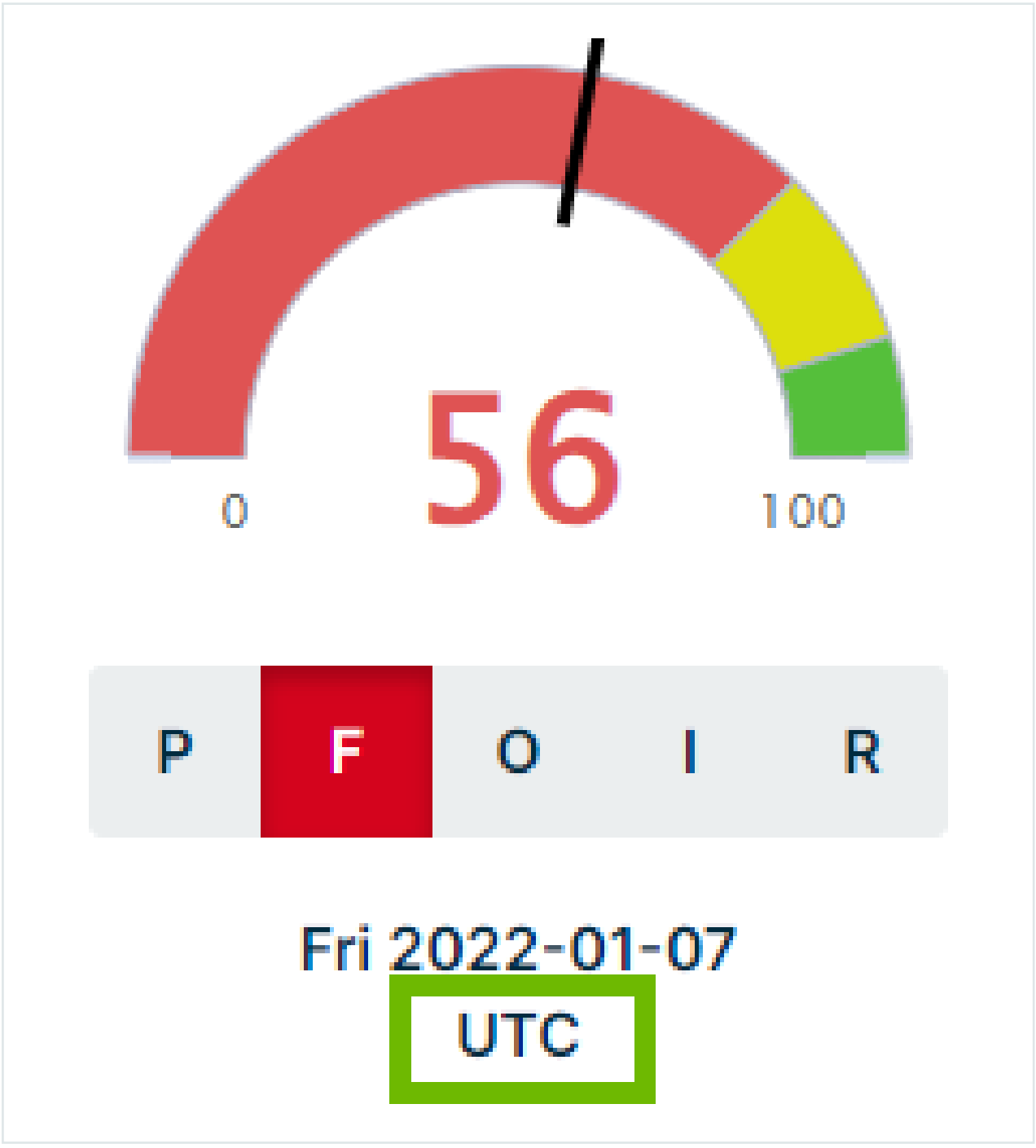
Updating the timezone

While scheduled jobs run according to the default UTC timezone, you may find you want to configure them to run according to the runId of a specific timezone.

Steps

1. Below the score meter on the Findings page of a dataset or on the right side of the metabar, click the timezone.
   See screenshot...[1]

1



> Note The default timezone is UTC, but if an alternative timezone was previously set, the one that displays may be different.

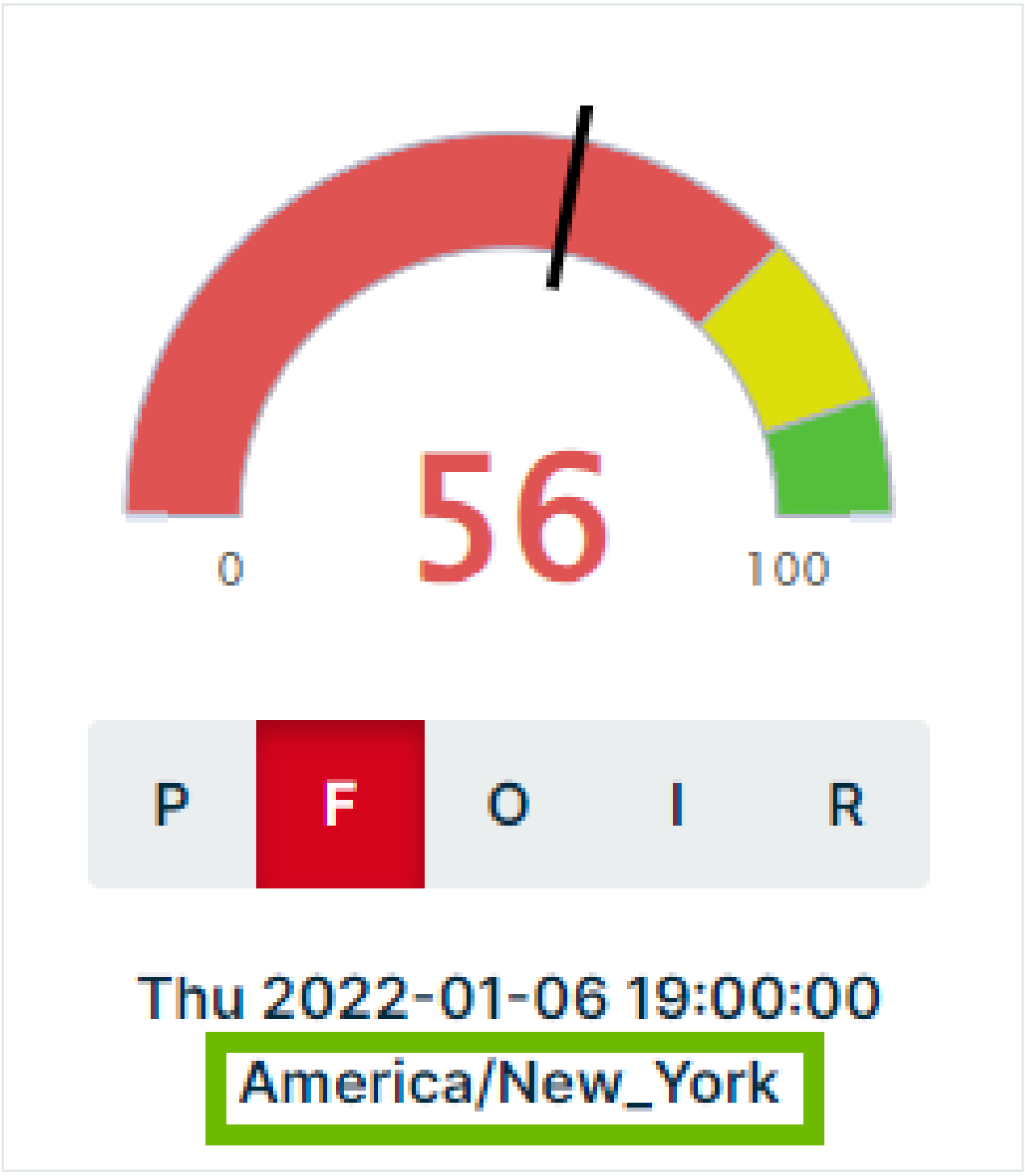
» The Update Timezone modal appears.

2. Select an option from the dropdown menu.
3. Click **Save**.

> Note Scheduled jobs now use the runId of your updated timezone.

See screenshot...[1]

1



Known limitations

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.

Schedule Restrictions

Admin users can manage schedule restrictions from the Admin Console ▸ Schedule Restrictions.

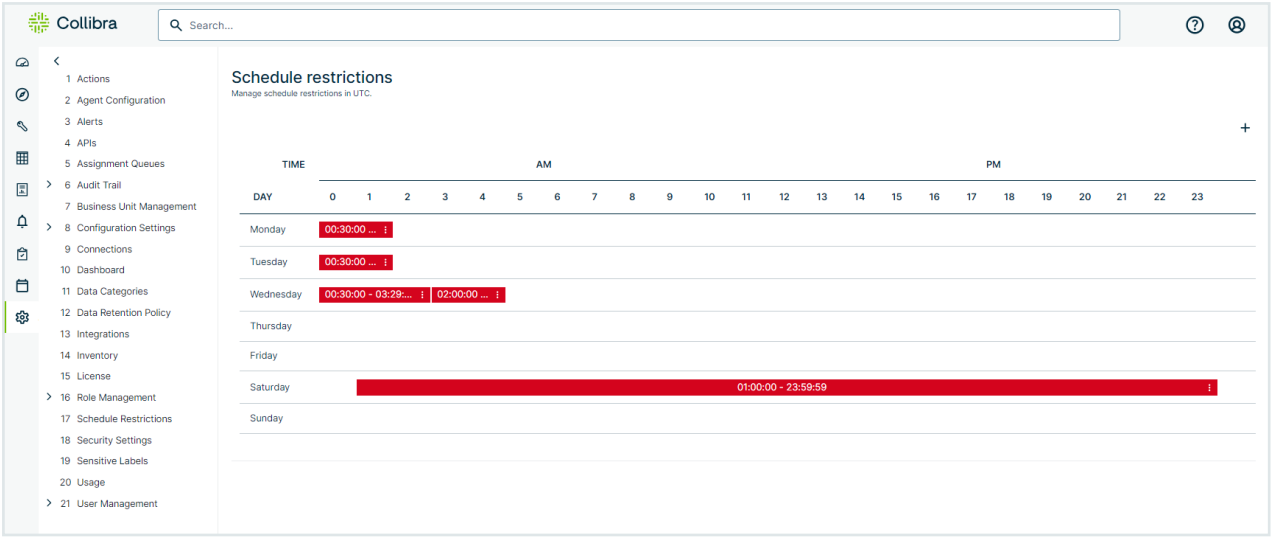
> Note All times on the Schedule Restrictions page reflect the UTC timezone.

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

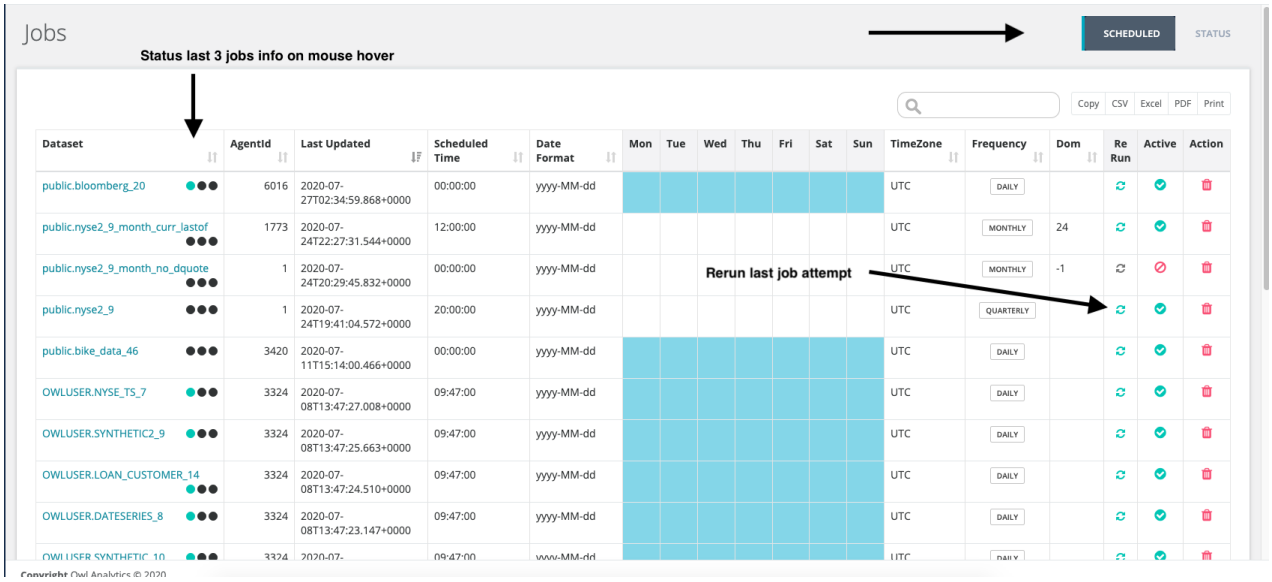
When you don't want automated jobs to run during business hours or during a particular day/time, you can set off-peak times to prevent authorized users from selecting them when scheduling a job.



1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
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.

View/Re-Run Scheduled Jobs

You can view your schedule jobs from the scheduled tab on the jobs page.



DQ Views

Running a Full Scan ..... mxx

Running a Partial Scan ..... mxxi

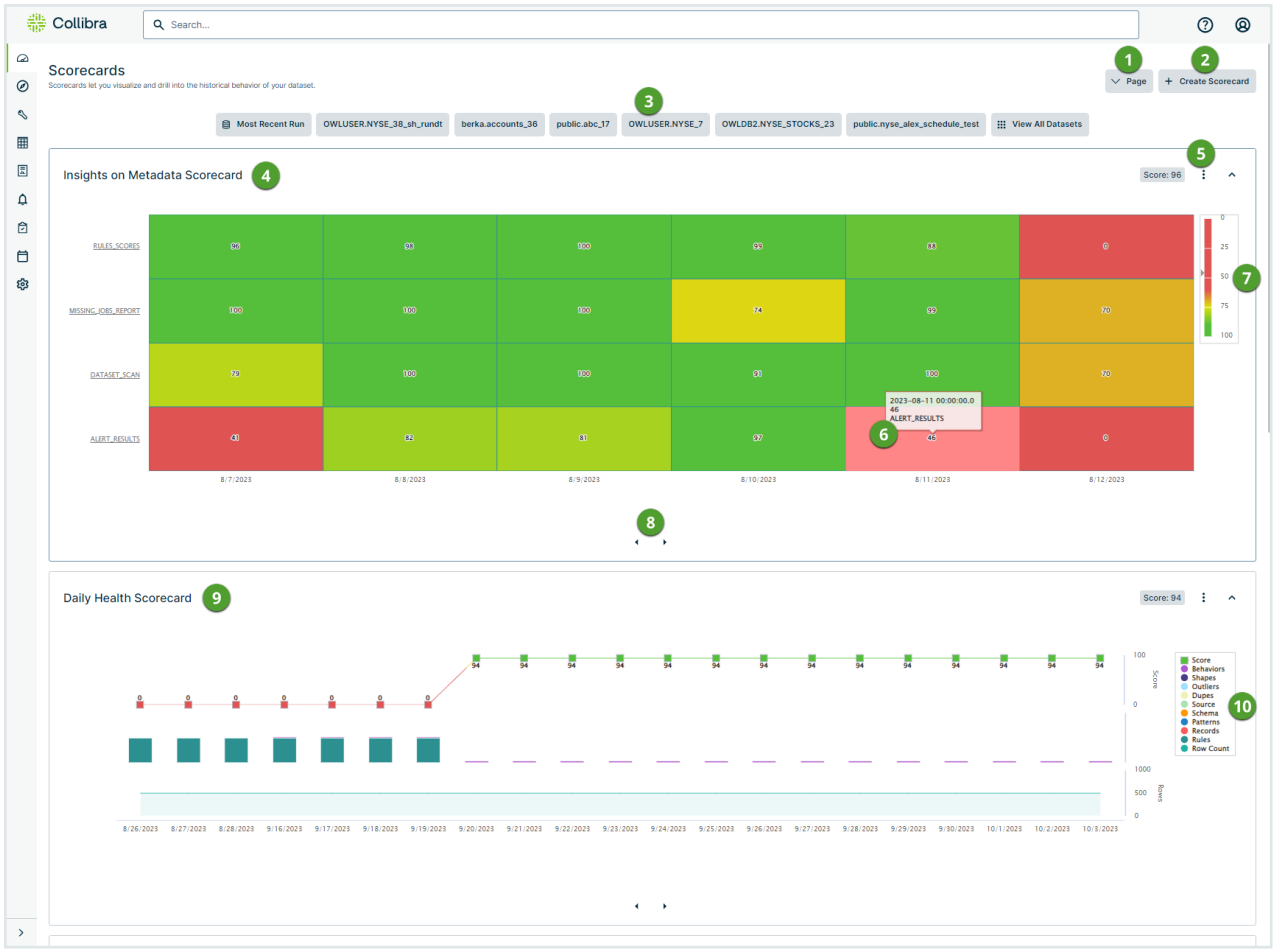
Scanning with SQL Query ..... mxxii

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 select any available scorecard in your environment.
2	Click **+ Create Scorecard** to create a new scorecard based on one or more datasets in your environment.
3	The dataset navigation buttons provide quick access to the most recent dataset run, frequently run datasets, and Dataset Manager.
4	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.

Number	Description
5	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.
6	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.
7	When you hover your cursor over a score in the scorecard, this meter shows the score for the corresponding run.
8	Click the arrows to open available date ranges.
9	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.
10	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.

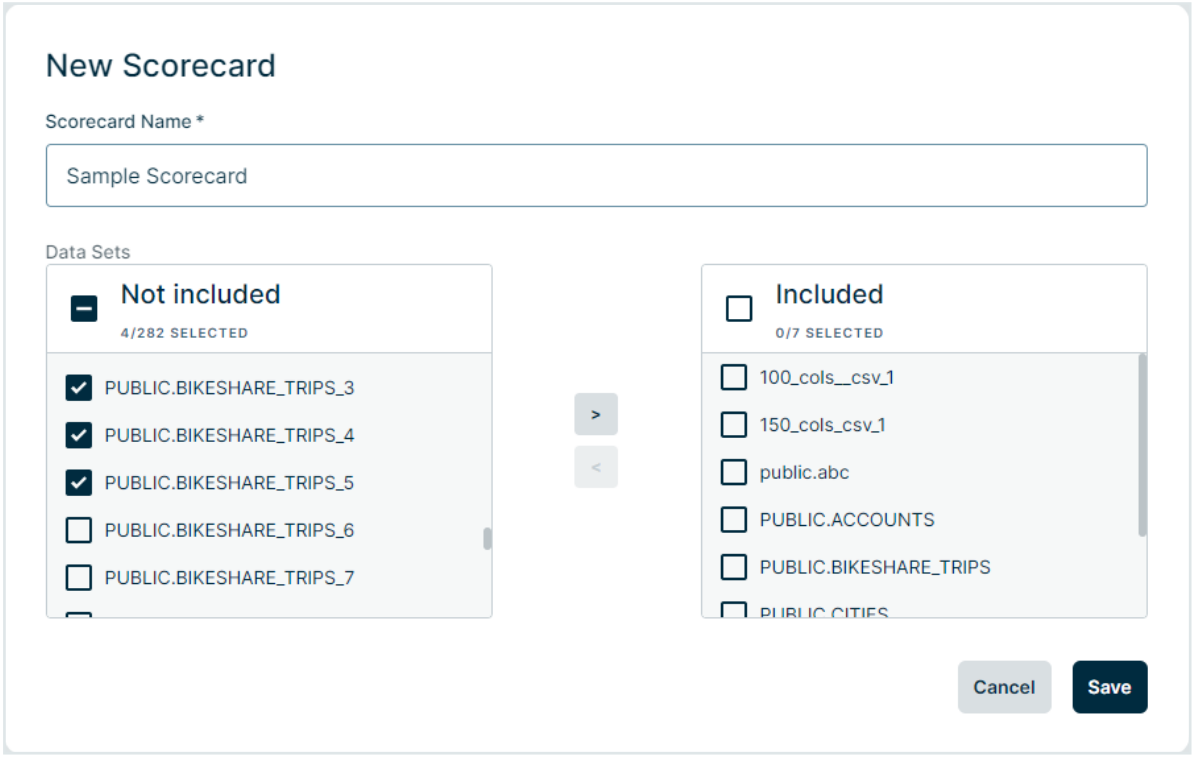
Create a scorecard

If you want to view the performance and health of specific datasets, you can create a scorecard.

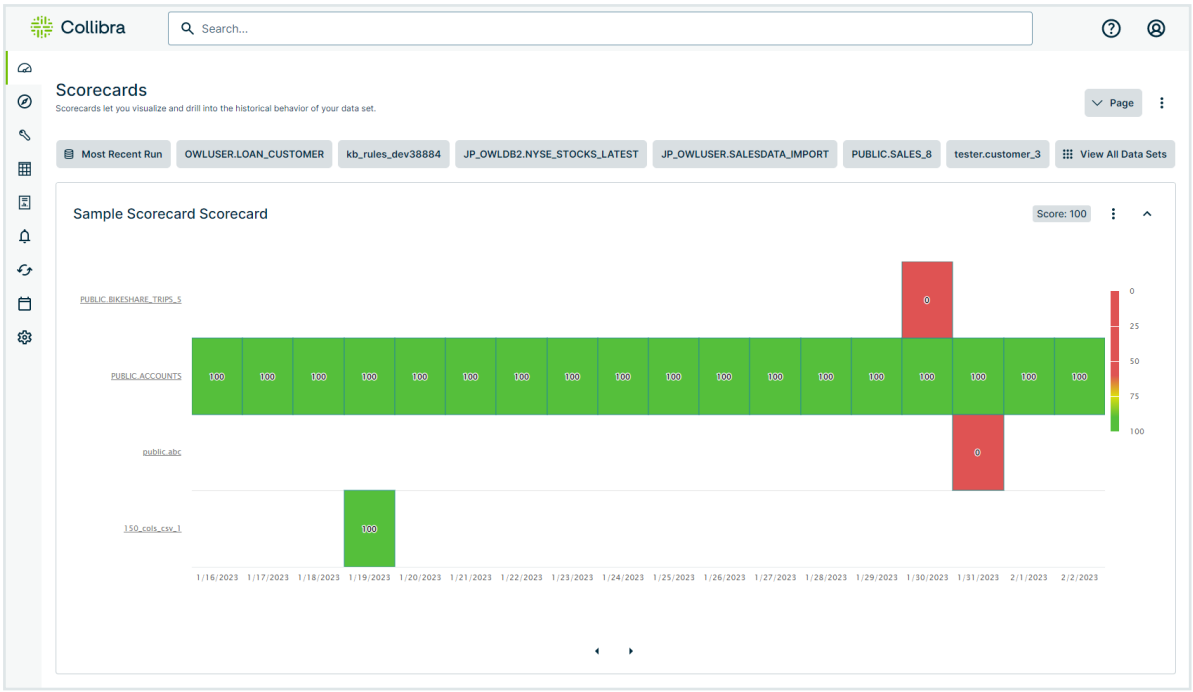
Steps

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.

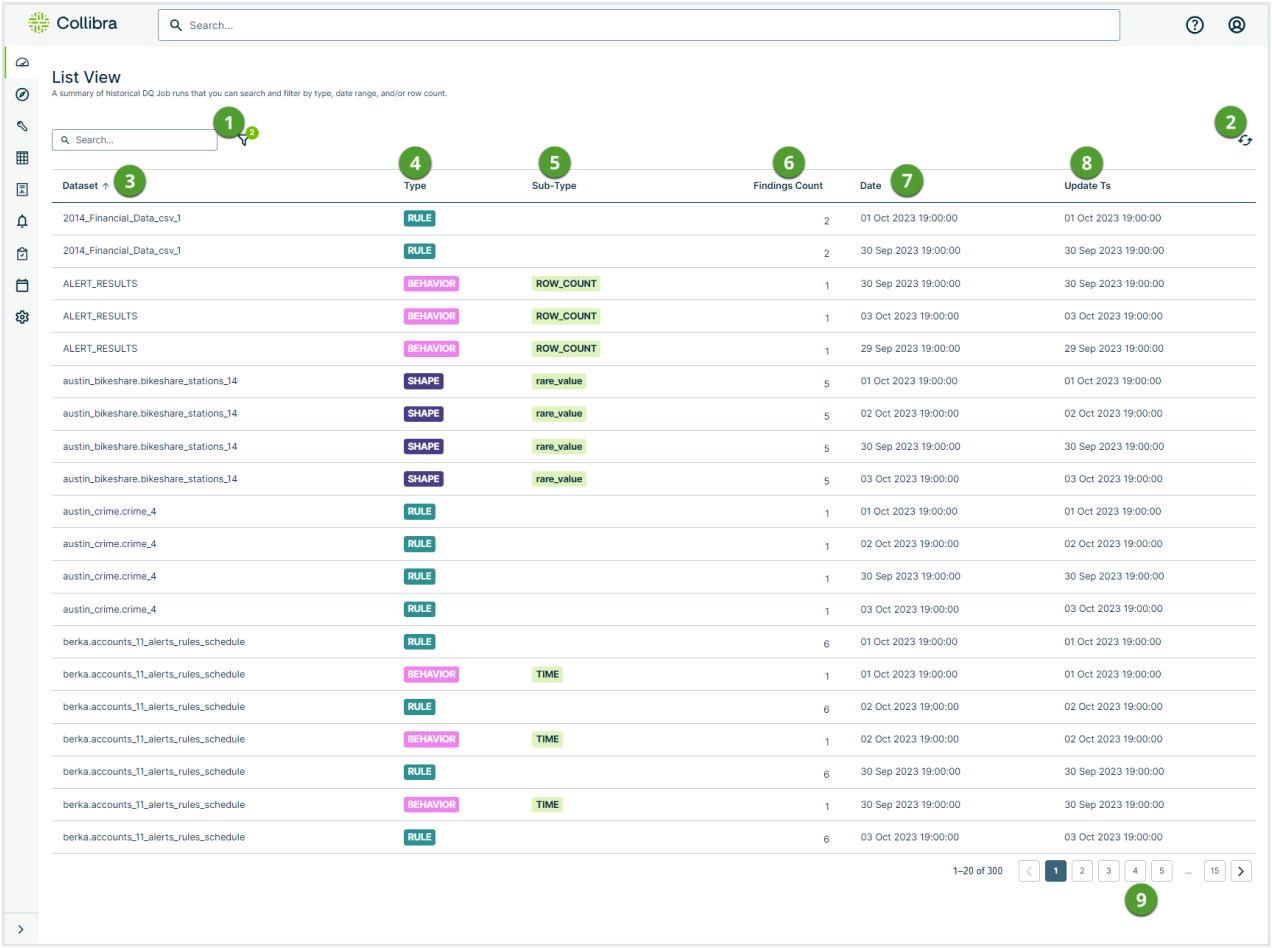


> Tip To remove a dataset from your scorecard, select it from the Included section and then click <.

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 filter icon 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.

Number	Element	Description
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).

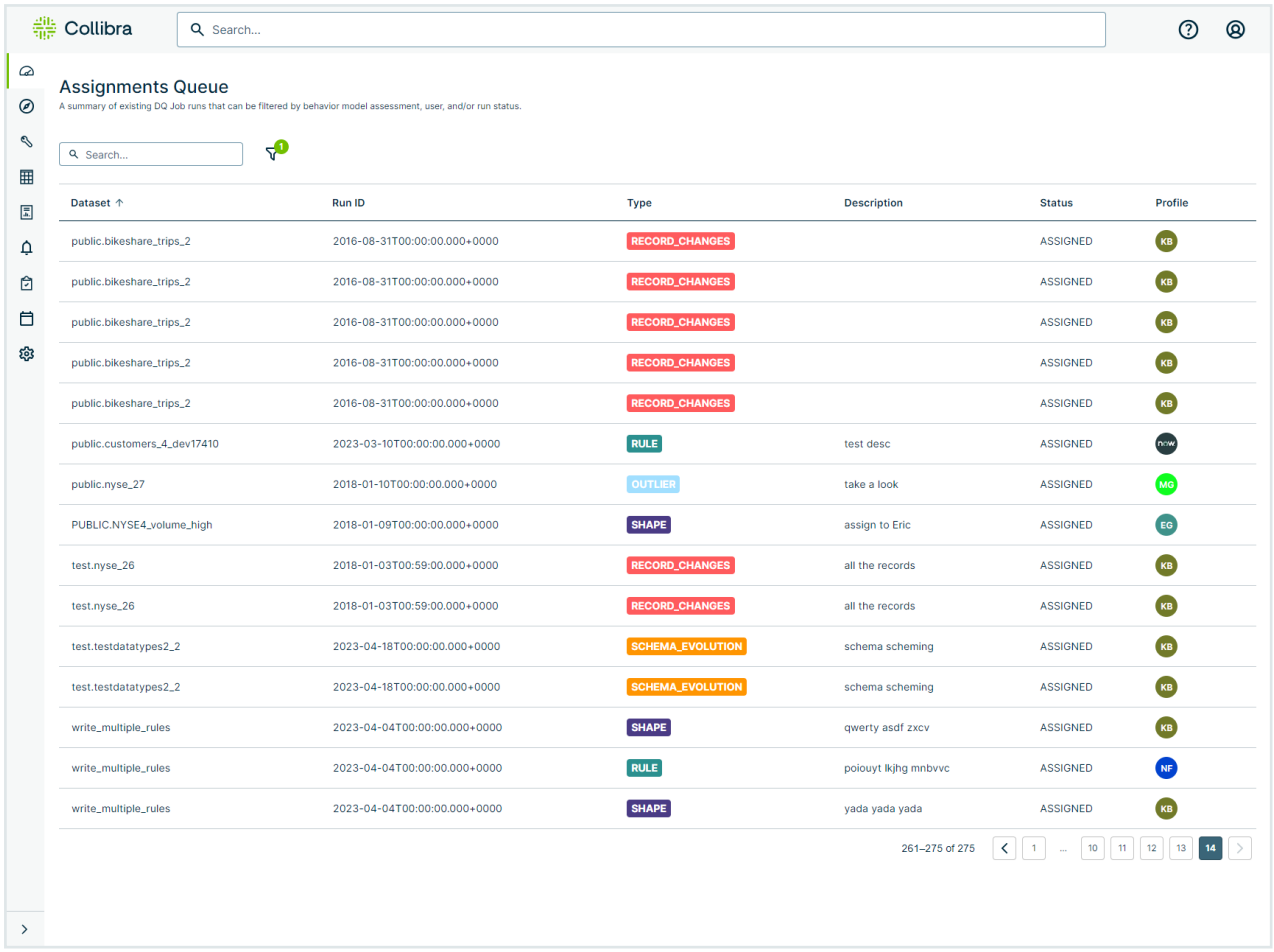
Tip Click the name of the dataset or the timestamp in the Date column to open the Findings page.

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, 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).

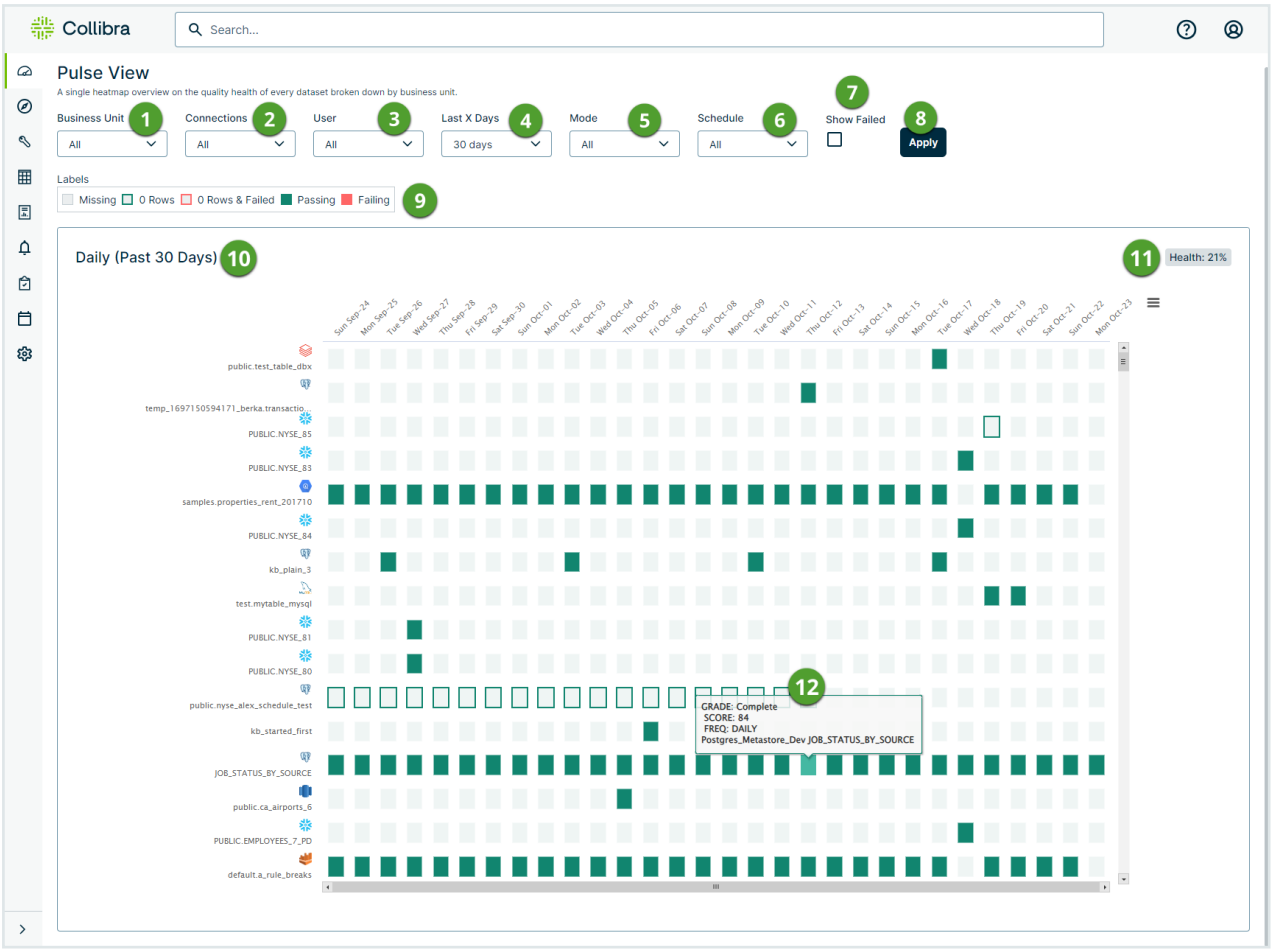
The Assignments Queue also allows you to filter all findings without an assignee, which can help to ensure that all datasets in your environment are actively managed. For instance, if a scheduled job ordinarily runs without any day-over-day issues but then detects an unusual schema evolution on the latest run, it may be beneficial to assign someone to validate it.



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	<table><tr><th>Label</th><th>Label icon</th><th>Description</th></tr><tr><td>Missing</td><td></td><td>A run that is missing either failed at run time or was never created, started, or picked up.</td></tr><tr><td>0 Rows</td><td> </td><td>A run that is passing or complete and has resulted in 0 rows of data.</td></tr><tr><td>0 Rows & Failing</td><td> </td><td>A run that is failing or incomplete and has resulted in 0 rows of data.</td></tr><tr><td>Passing</td><td> </td><td>A run that is passing or complete and has resulted in rows of data.</td></tr><tr><td>Failing</td><td> </td><td>A run that is failing or incomplete and has resulted in rows of data.</td></tr></table>
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.
12	Run inform-ation	Hover your cursor over a specific run on the heat map to show its grade, score, scheduled frequency, and the name of the dataset.

More

Collibra DQ advanced features.

When specific DQ challenges require specific DQ detection techniques, Collibra DQ 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. Read more to understand if specific dimensions can be applied to your data.

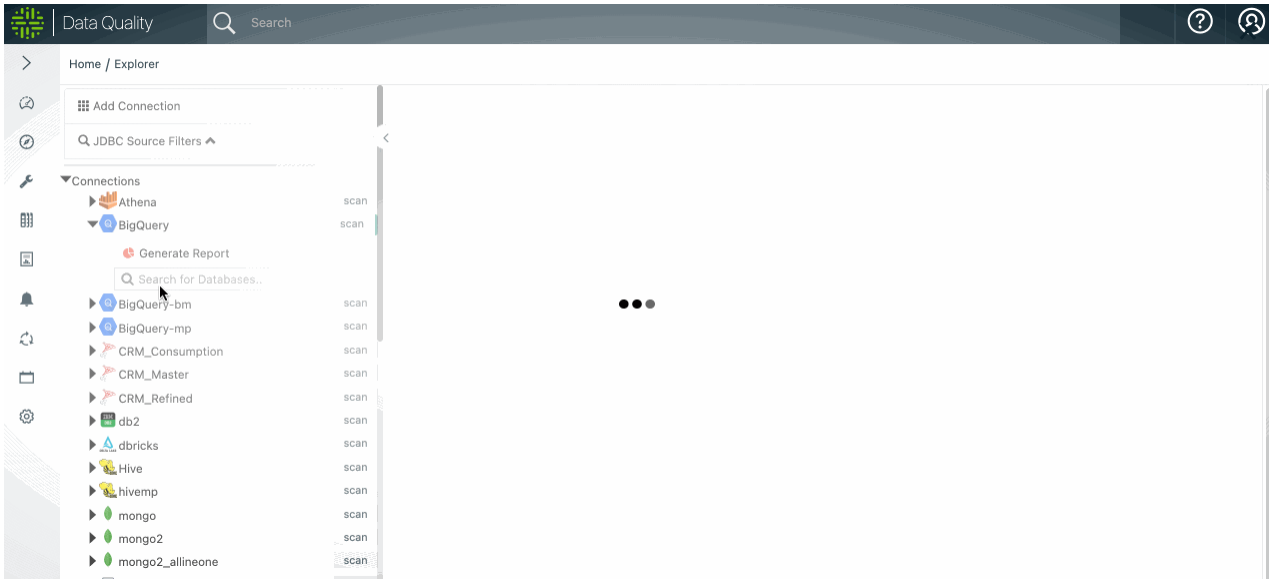
Running a Full Scan ..... mxx

Running a Partial Scan ..... mxxi

Scanning with SQL Query ..... mxxii

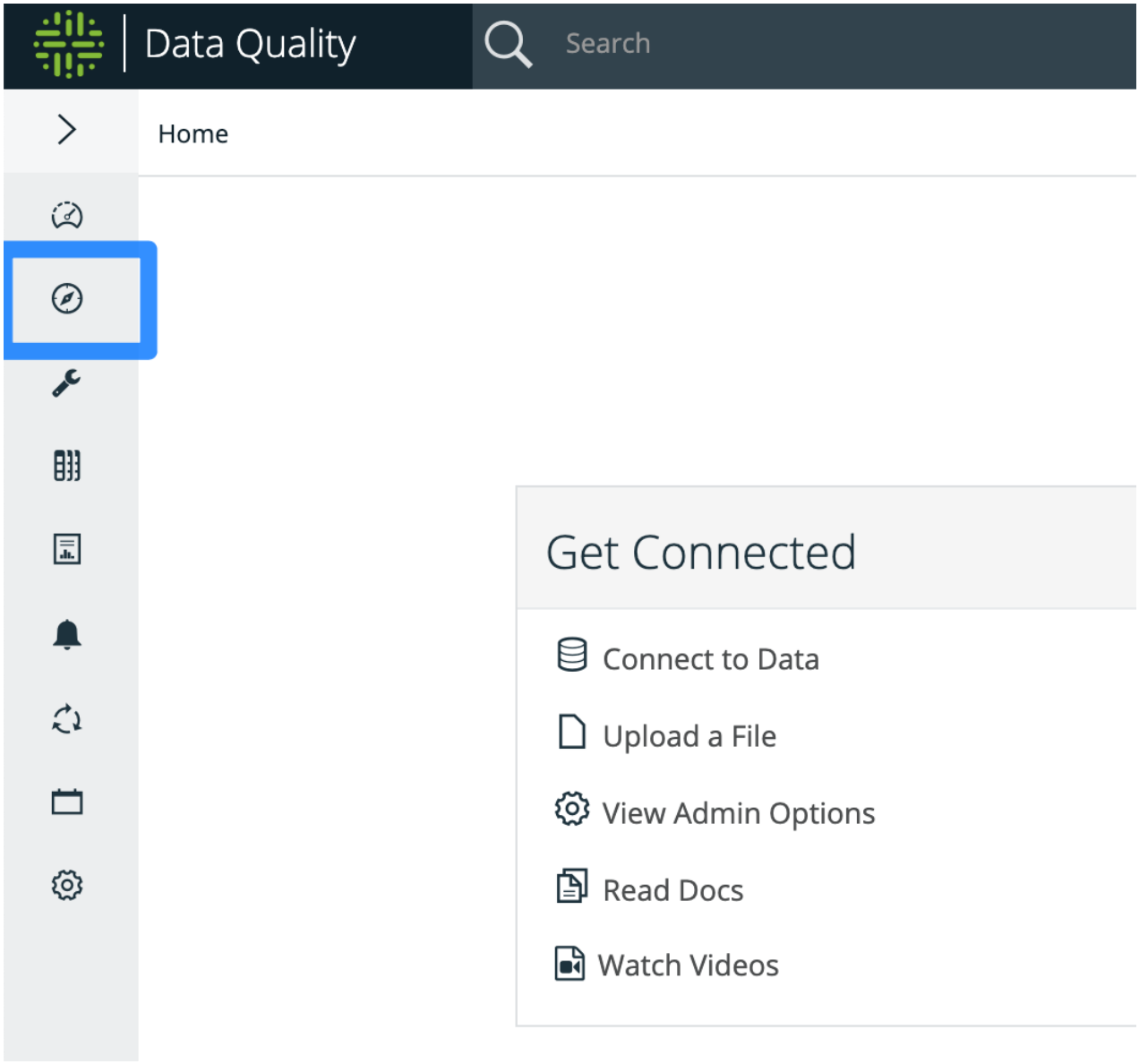
Explorer (no-code)

A no-code option to get started quickly and onboard a dataset.



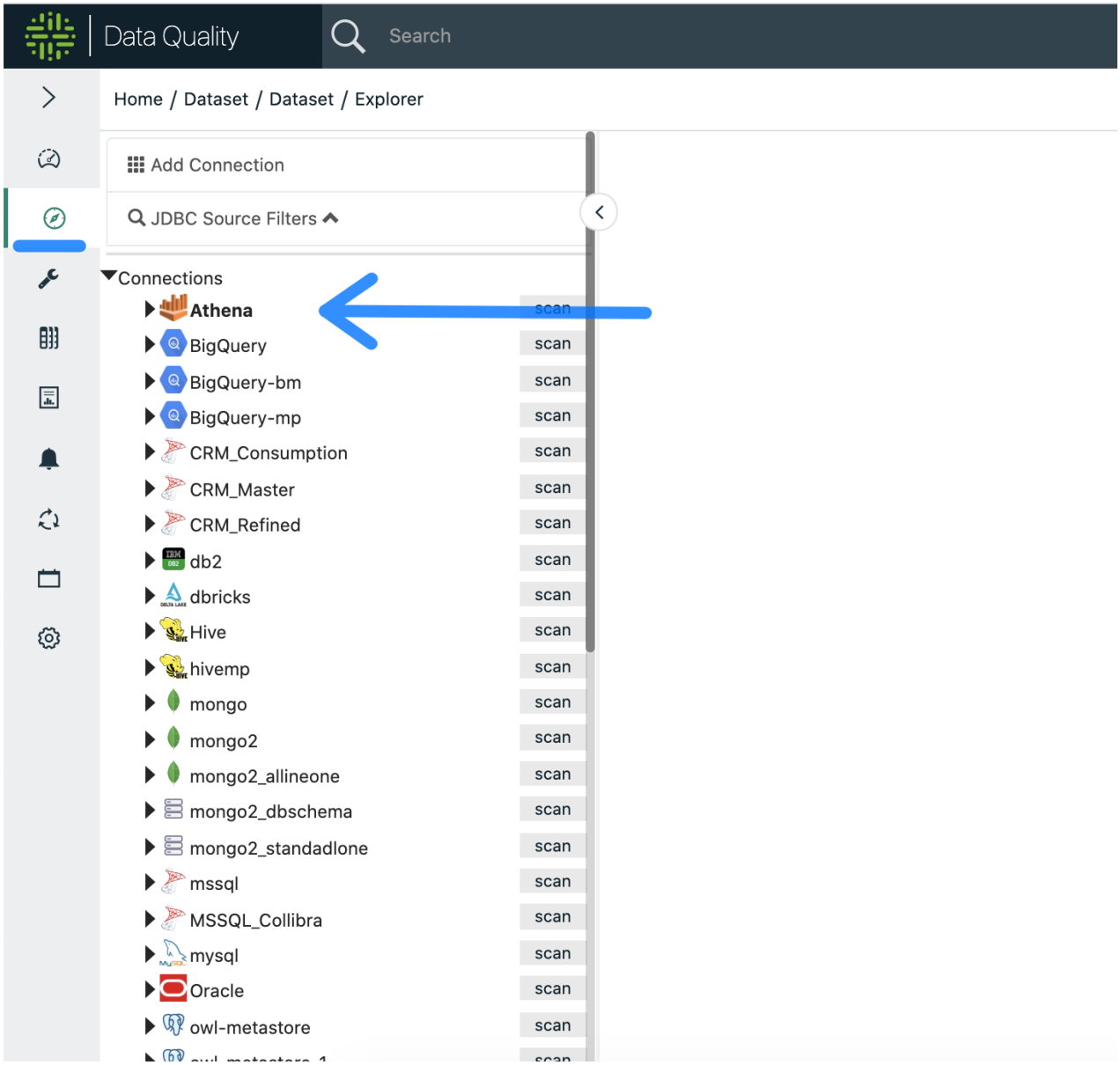
Getting Started

This page can be accessed by clicked the Explorer option (the compass icon).

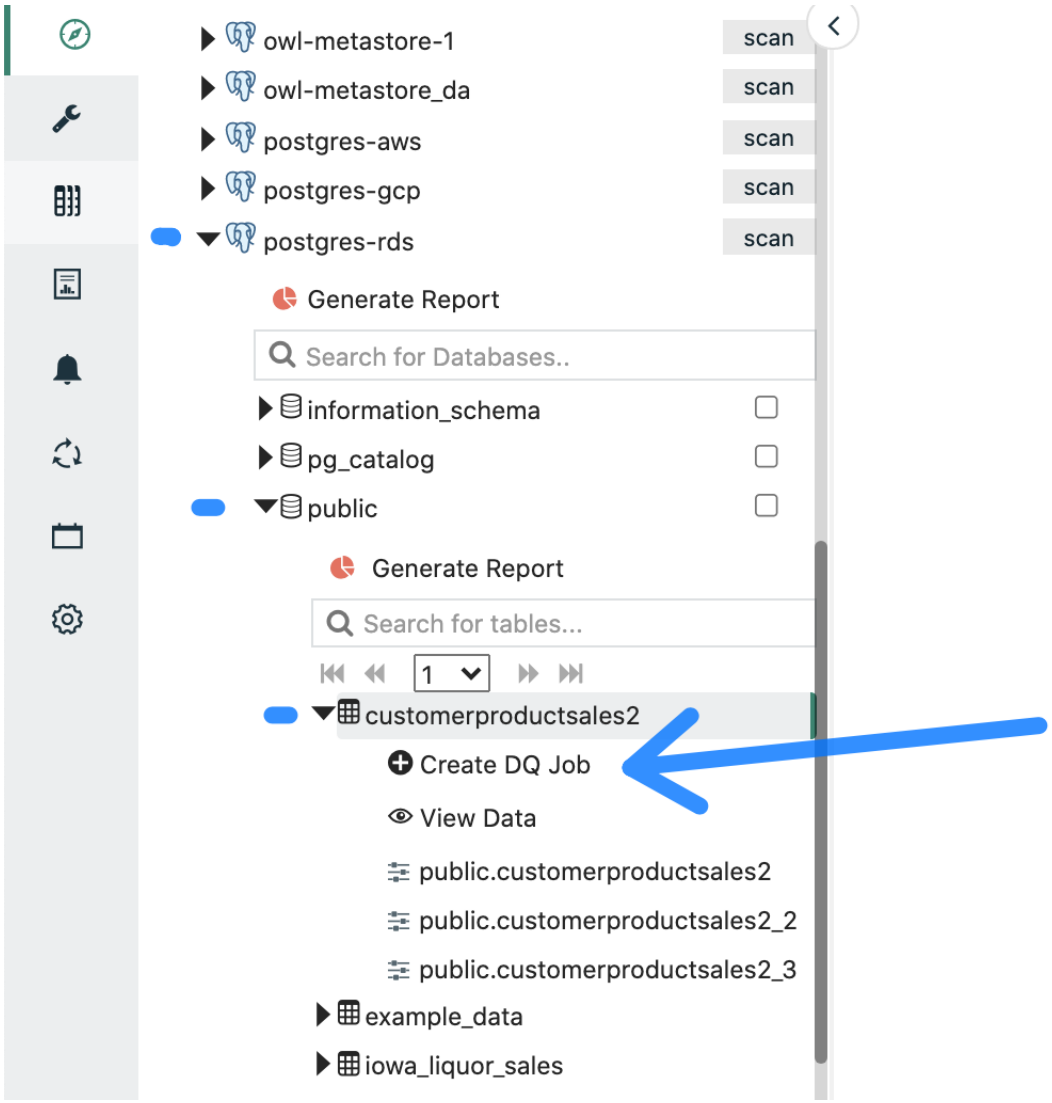


Note All UI functionality has corresponding API endpoints to define, run, and get results programmatically.

Select Your Data Source



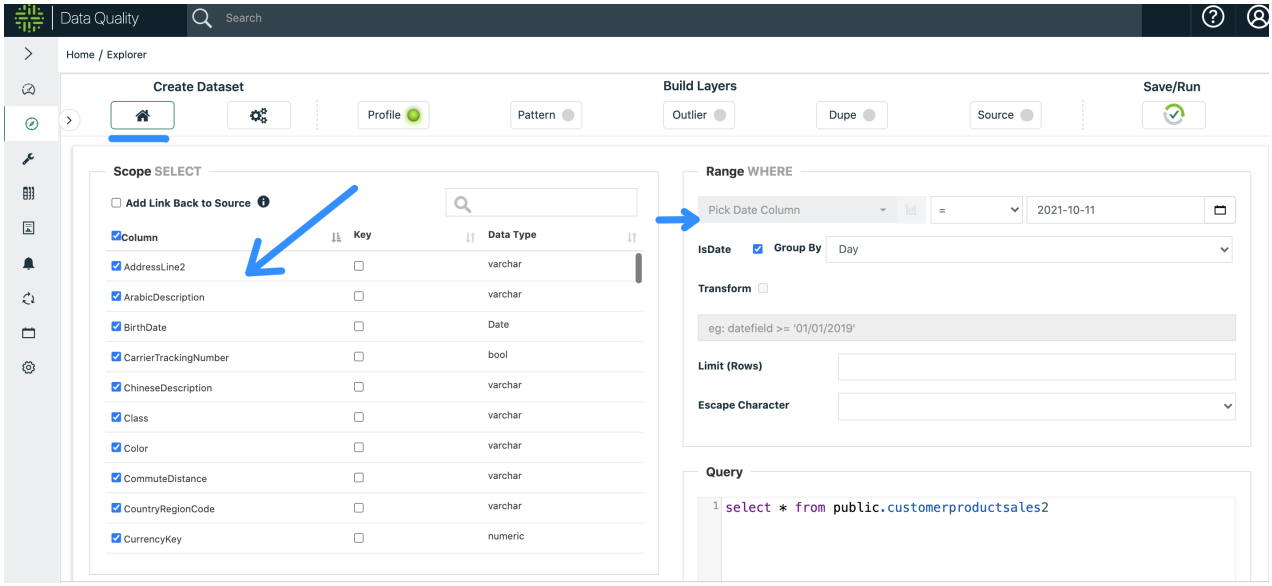
Create a new DQ Job by clicking +Create DQ Job



View Data is an interactive option to run queries and explore the data

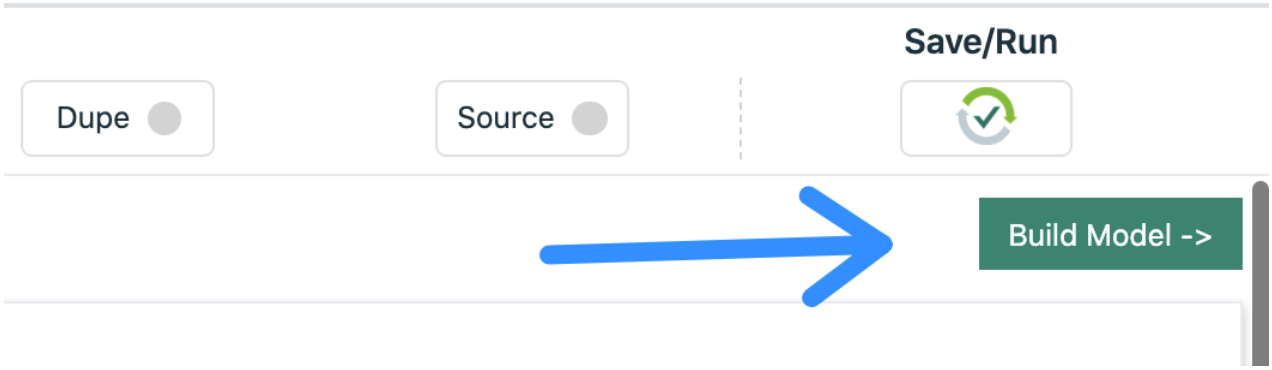
The bar chart icon will take you to a profile page of the dataset created prior to Explorer 2

Select The Scope and Define a Query



Pick Date Column if your dataset contains an appropriate time filter

Click Build Model -> to Save and Continue

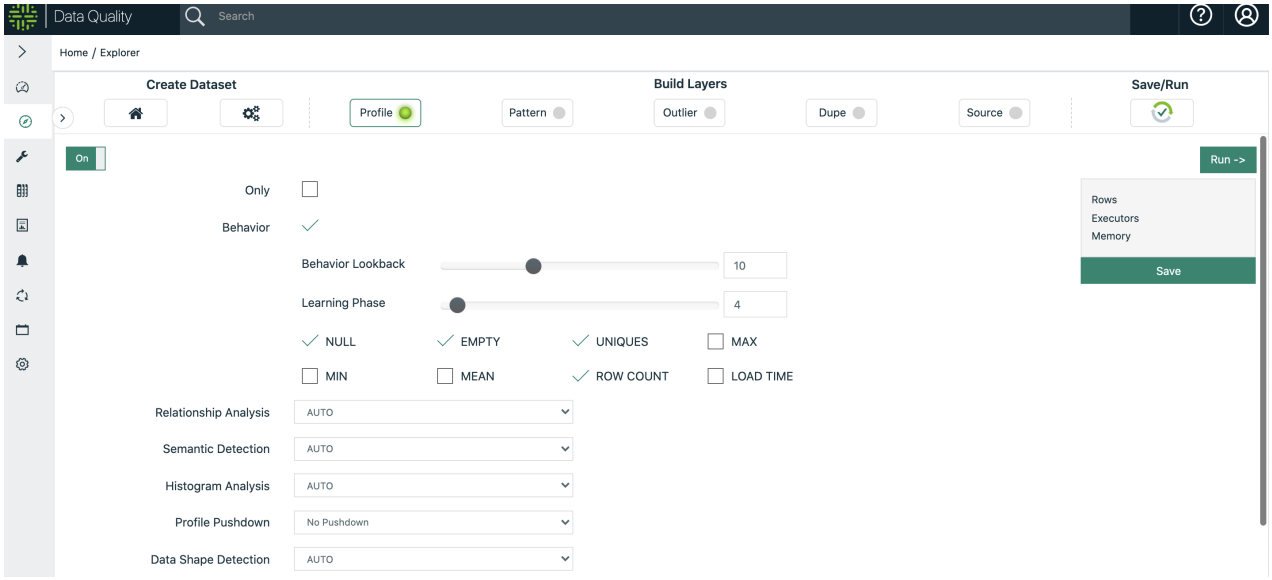


Transform Tab (advanced / optional)

Transform

Click Build Model -> to Save and Continue

Profile



Use the drop-downs to enable different analysis. Best practice is to leave the defaults.

Pattern (advanced / optional)

Toggle on Pattern to enable this layer.

Click +Add to define a group and series of columns.

Patterns (advanced)

Click Save to and Click Outlier to Continue

Outlier (advanced / optional)

Outliers (advanced)

Click Save to and Click Dupe to Continue

Dupe (advanced / optional)

Dupes (advanced)

Click Save to and Click Source to Continue

Source (advanced / optional)

Navigate to the source dataset.

Click Preview to interlace the columns.

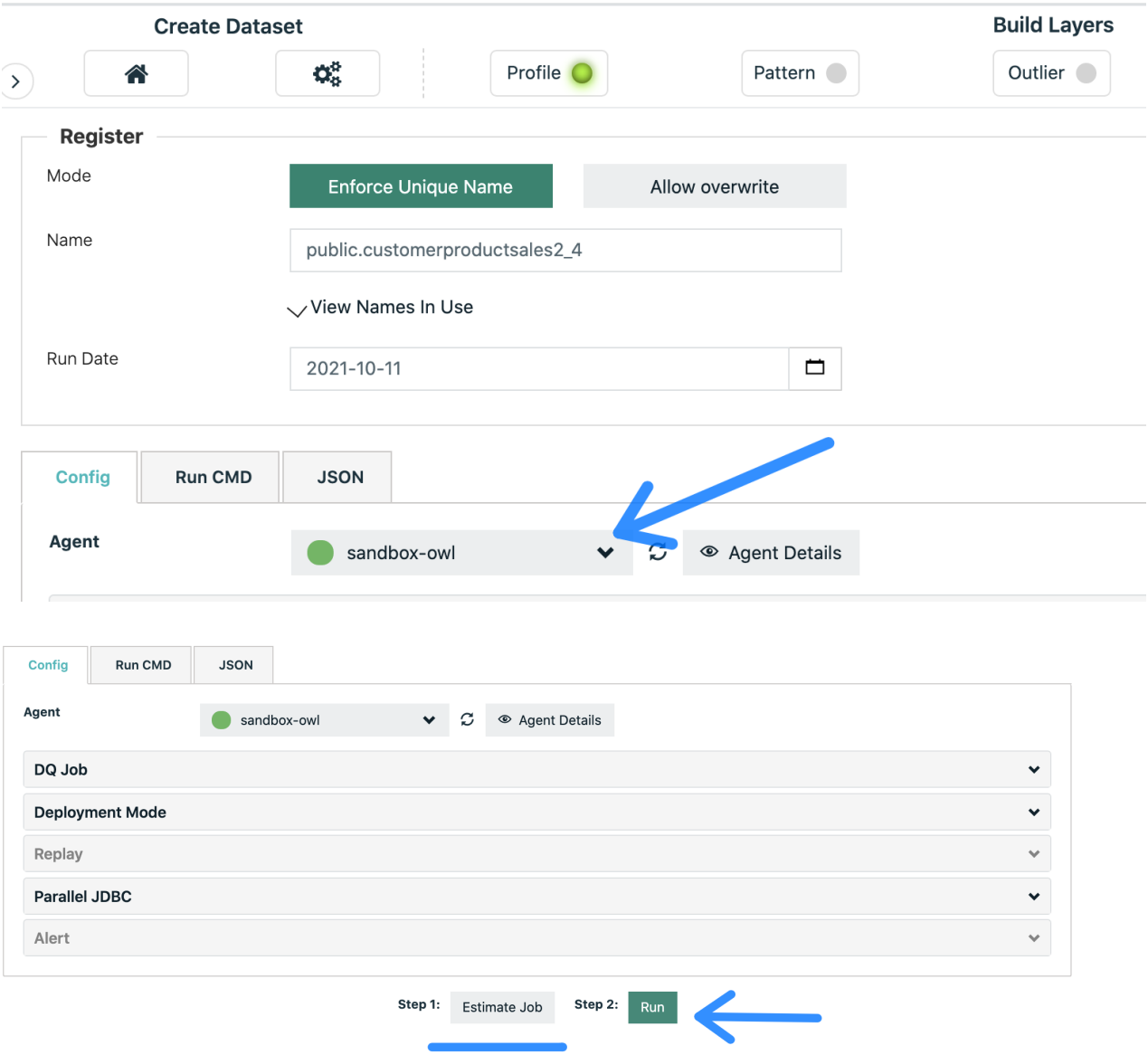
Manually map the columns by dragging left to right or deselect columns.

Source (advanced)

Click Save to and Click Save/Run to Continue

Run

1. Select an agent.
2. Click Estimate Job.
3. Click Run to start the job.



> Note *If you do not see your agent, please verify the agent has been assigned to your connection via Adding a connection to a DQ Agent.

Terminating a DQ Job

If a DQ Job is in progress, incorrectly submitted, or stuck in Staged status, you can terminate the job by clicking the Actions drop-down list and selecting **Terminate job**. If you

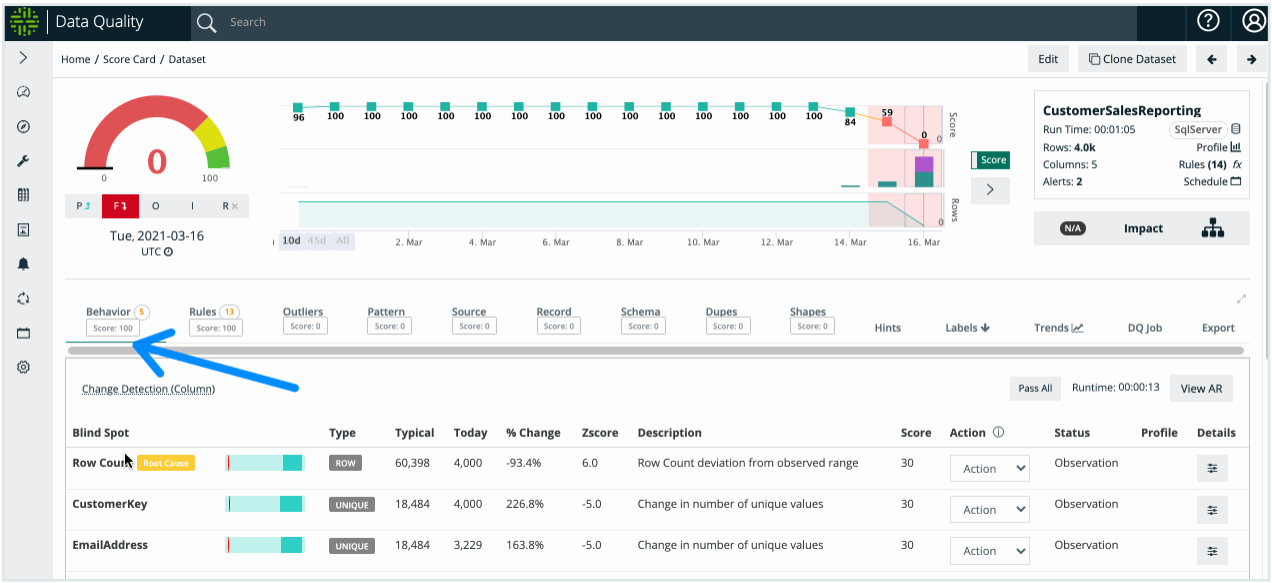
have an email address configured for alerts to be sent, two alerts are sent when a job is terminated.

> Note Jobs in the Spark UI display a Finished status when terminated, even though they are terminated from the Collibra DQ UI.

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.

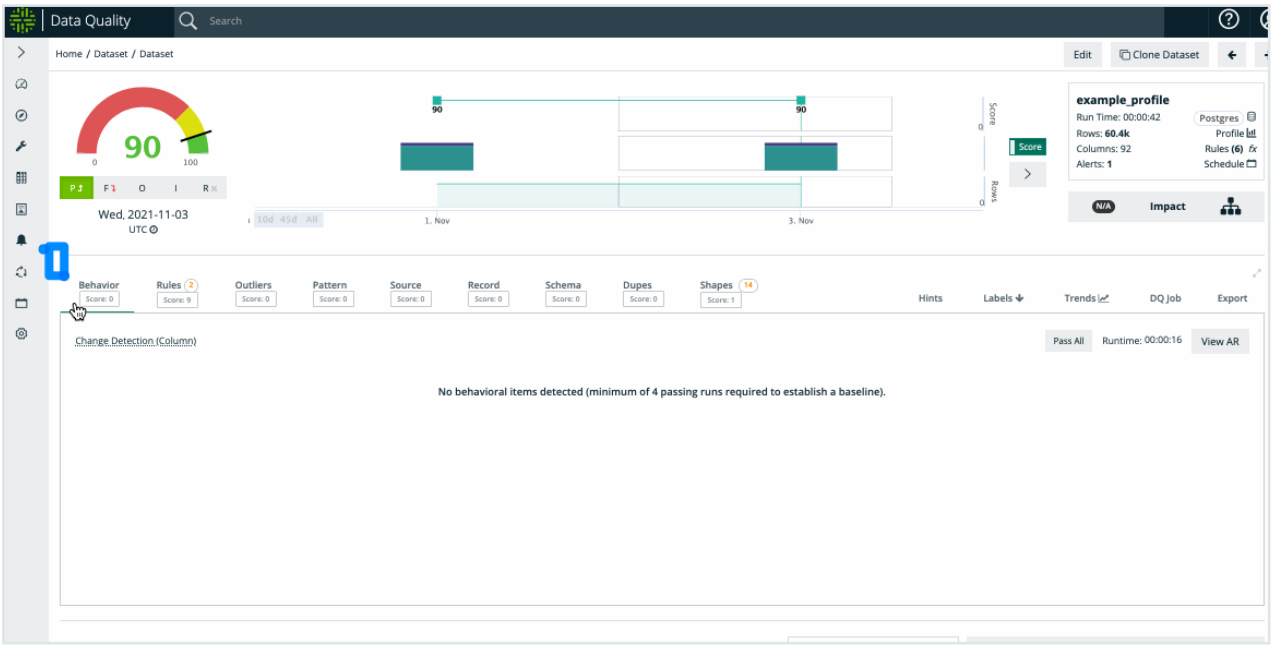


Evolution of Rule-based Data Quality

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.



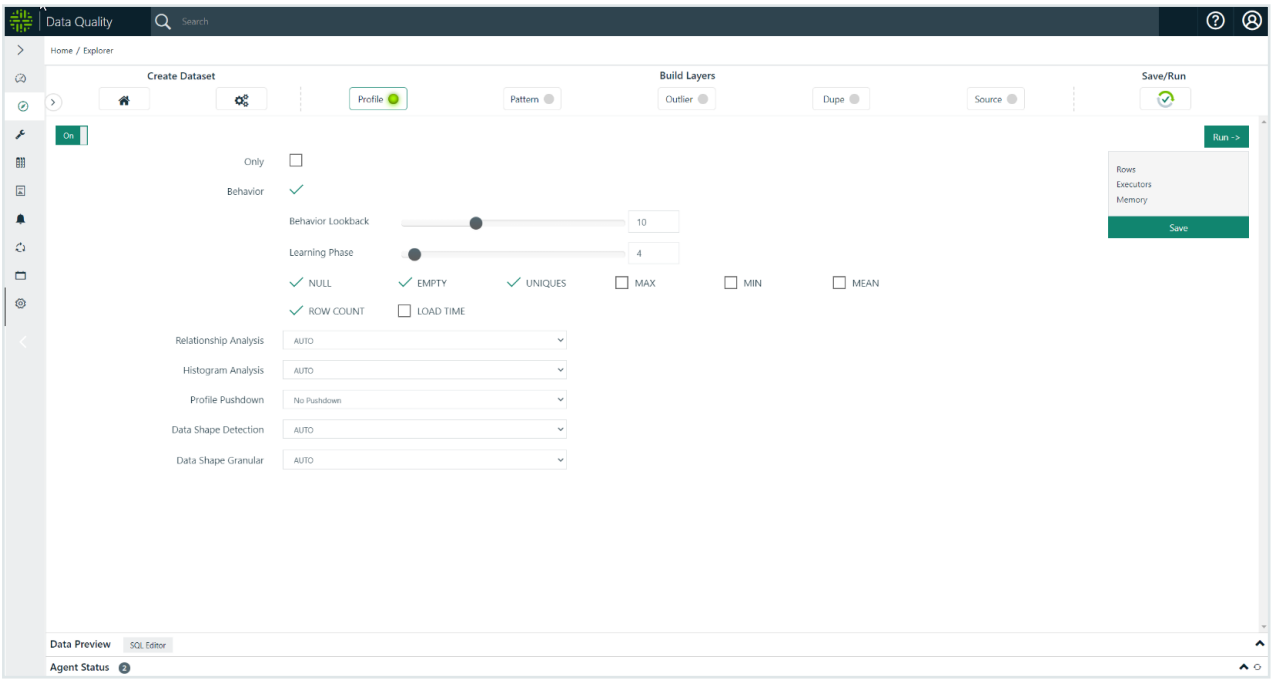
Using Behavioral Analytics (Change Detection)

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

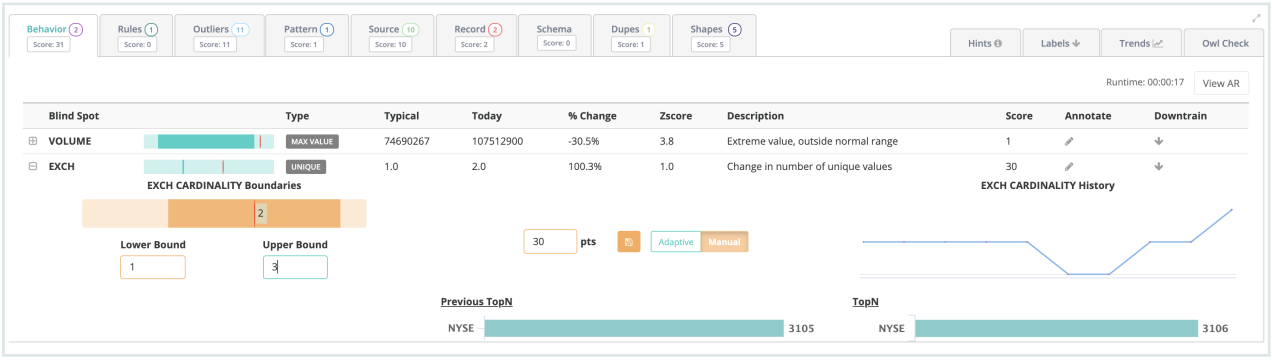
Over time, the definition of normal for any given column within the dataset can change. The data may legitimately become more sparse or decrease in volume. Collibra Data Quality continues to learn and adjust the model throughout the life of the dataset. However, if there is a drastic (but legitimate) 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.

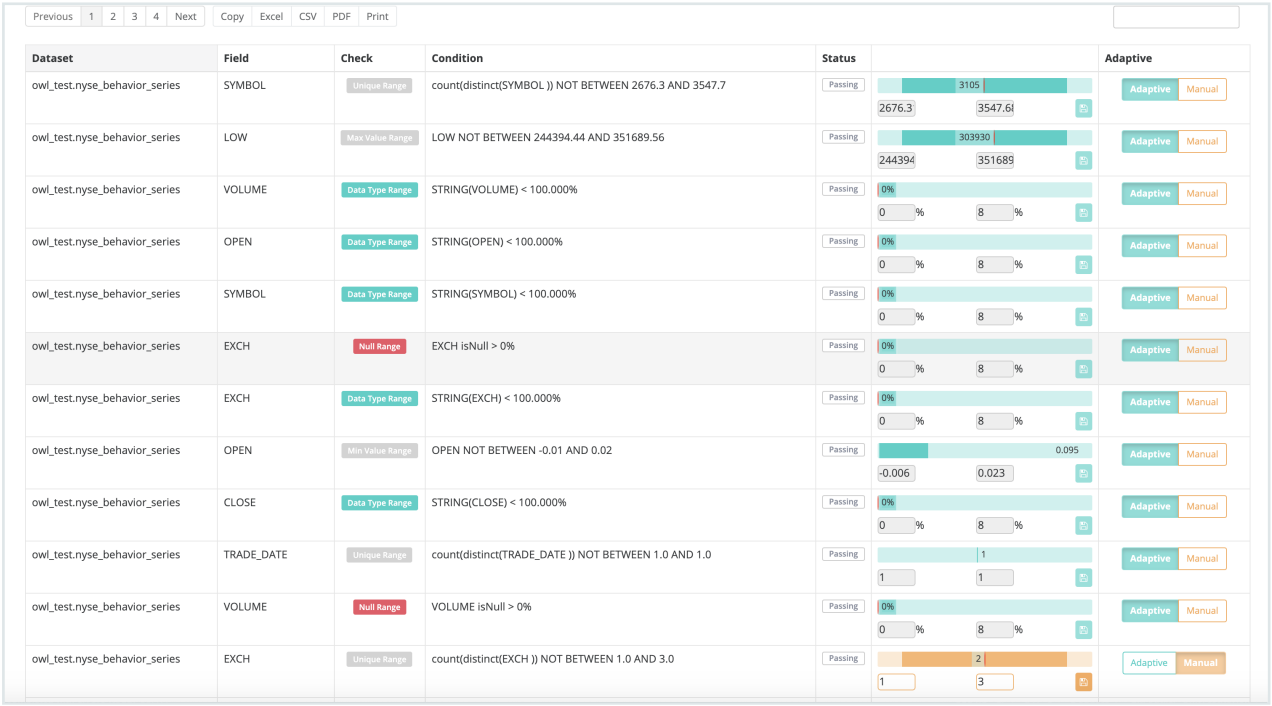
Item	Description
Blind Spot	The name of the column with a possible change detected during a DQ check.
Type	The type of DQ check performed on a given column. For example, Unique (Range) is the number of unique values that fall outside a given range.
Baseline	The baseline value is the mean of the preceding number of scans determined by the value selected for Behavior Lookback in the Profile section of the Explorer page.
Today	
% 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.

Item	Description
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.

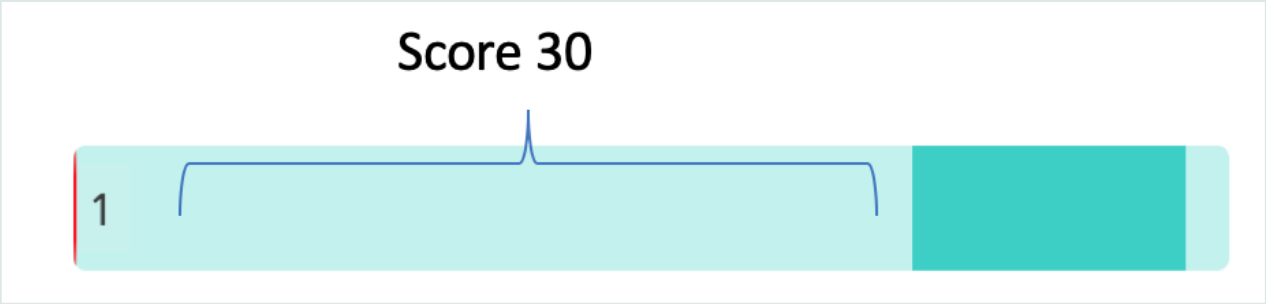


The Adaptive Rules also provide you with the ability to adjust ranges derived from the behavior model. You can manually adjust the tolerance range and score of any Adaptive

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.

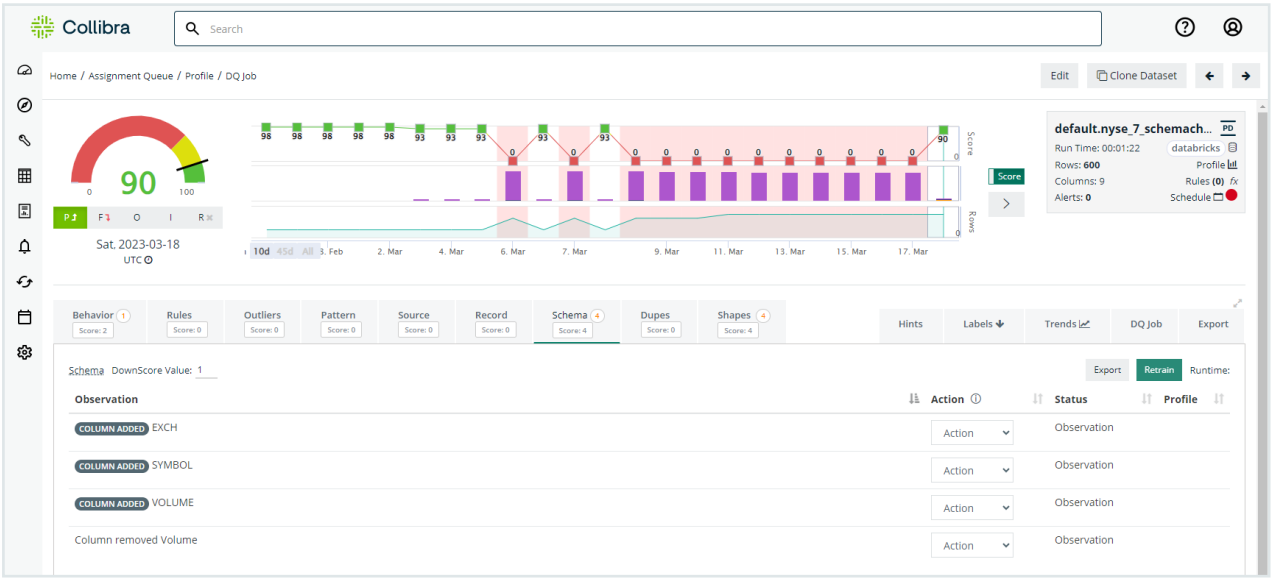
Schema (automatic)

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.

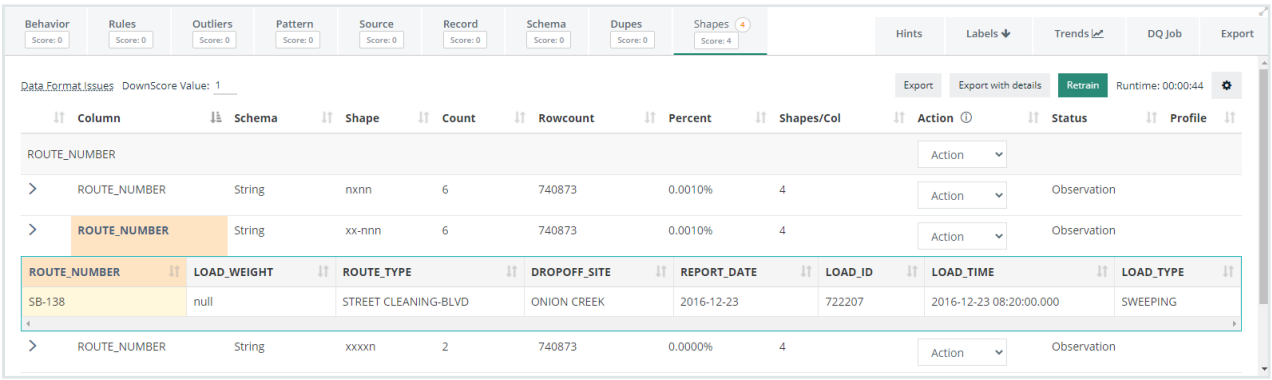
Shapes (automatic)

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

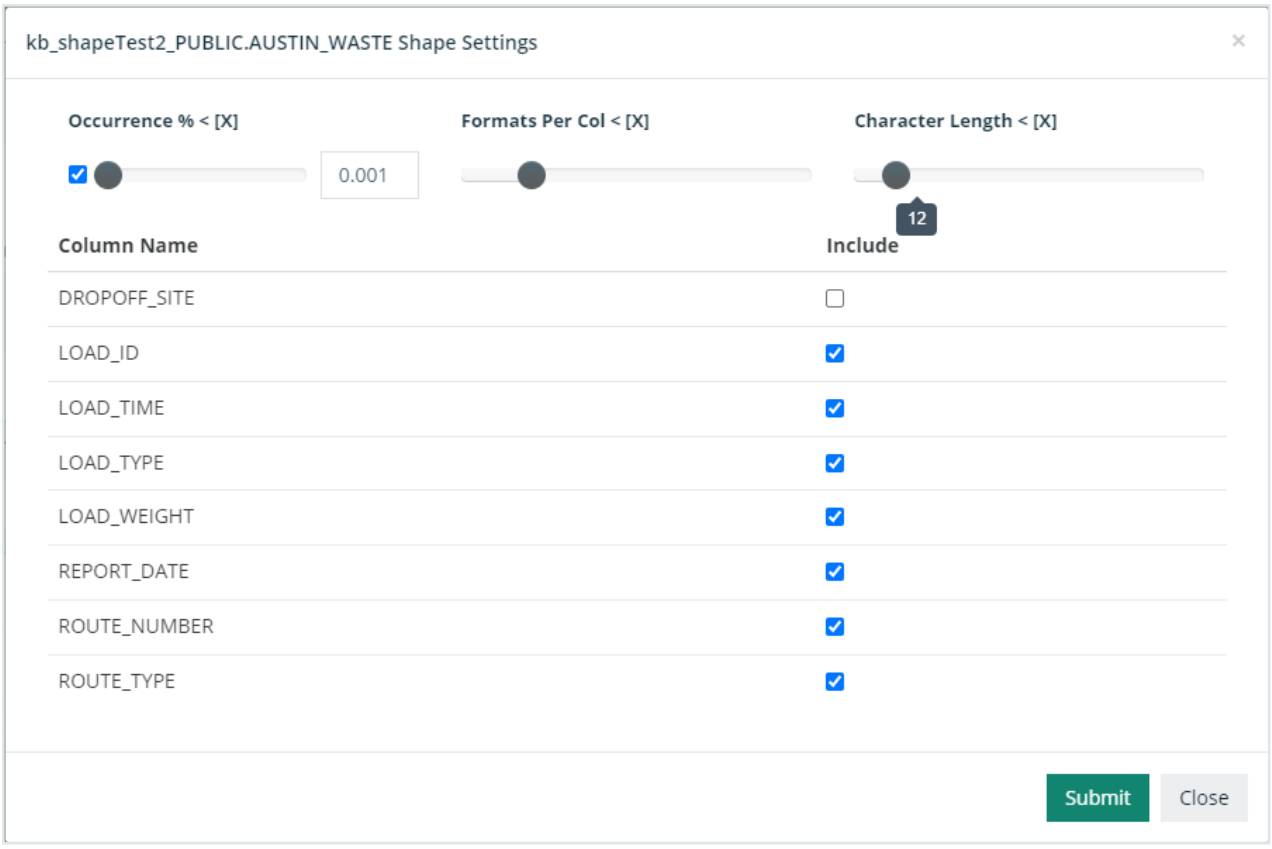
Drill-in to any Shape anomaly and see a visual example

See an itemized list view of the most infrequent or odd shapes in your datasets.



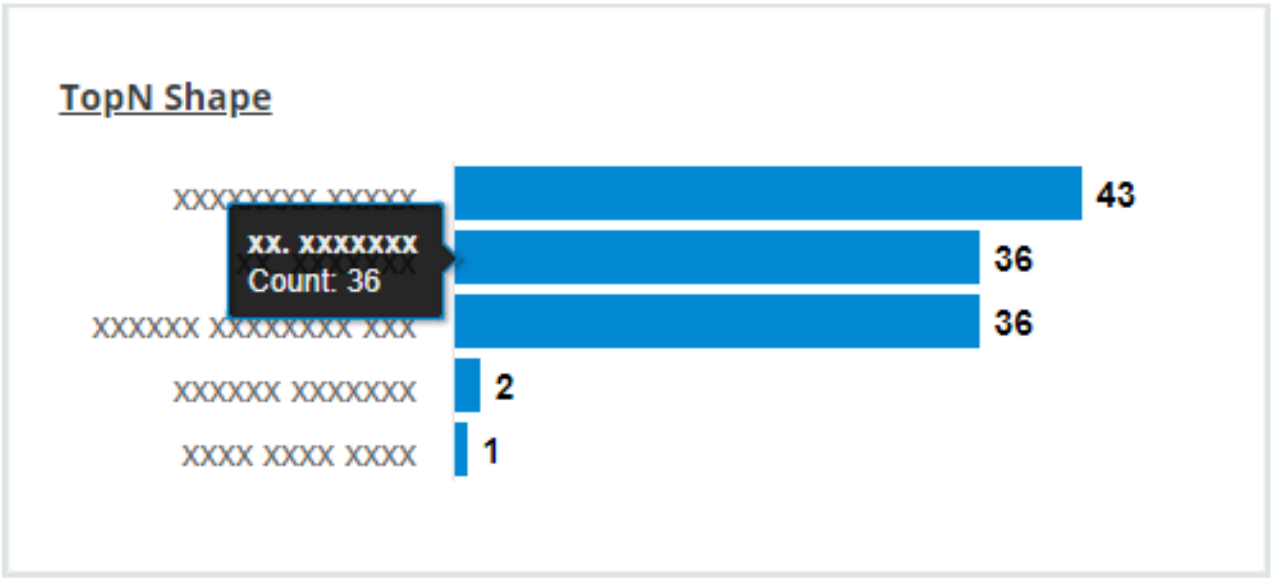
Shape Tuning

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.

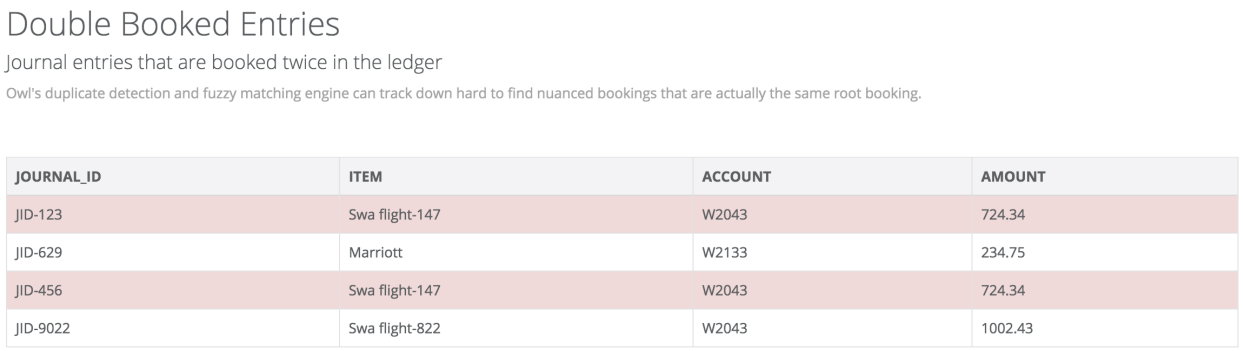
Exact matches detect entries that may match despite formatting issues. For example, Collibra DQ flags `John Doe`, `JOHN DOE`, and `john doe` as exact match dupes of each other. Exact match groups the same dupes and displays their counts in the **occurs** column on the Findings page.

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

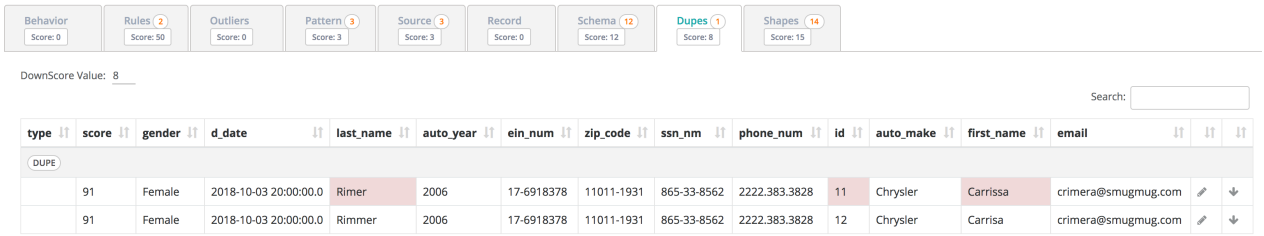
Owl's duplicate detection and fuzzy matching engine can track down hard to find nuanced bookings that are actually the same root booking.

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

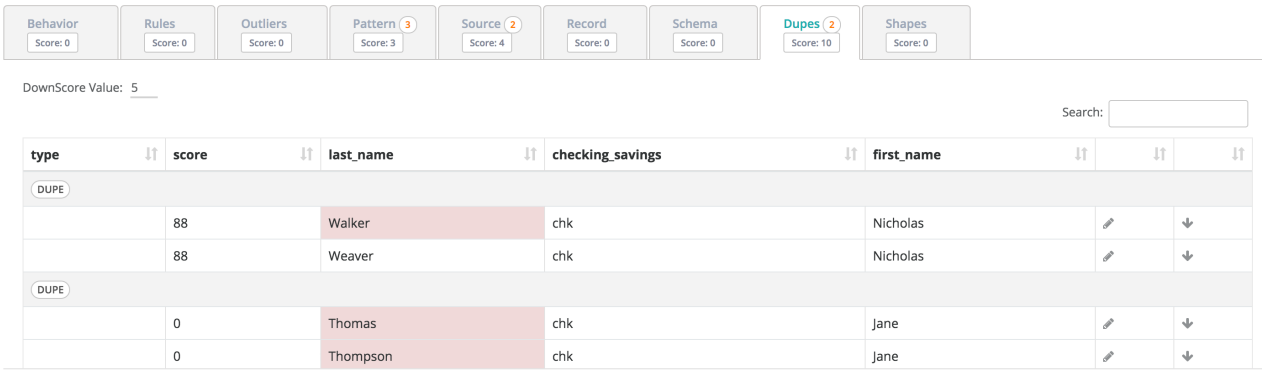
Carrisa Rimmer vs Carrissa Rimer



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

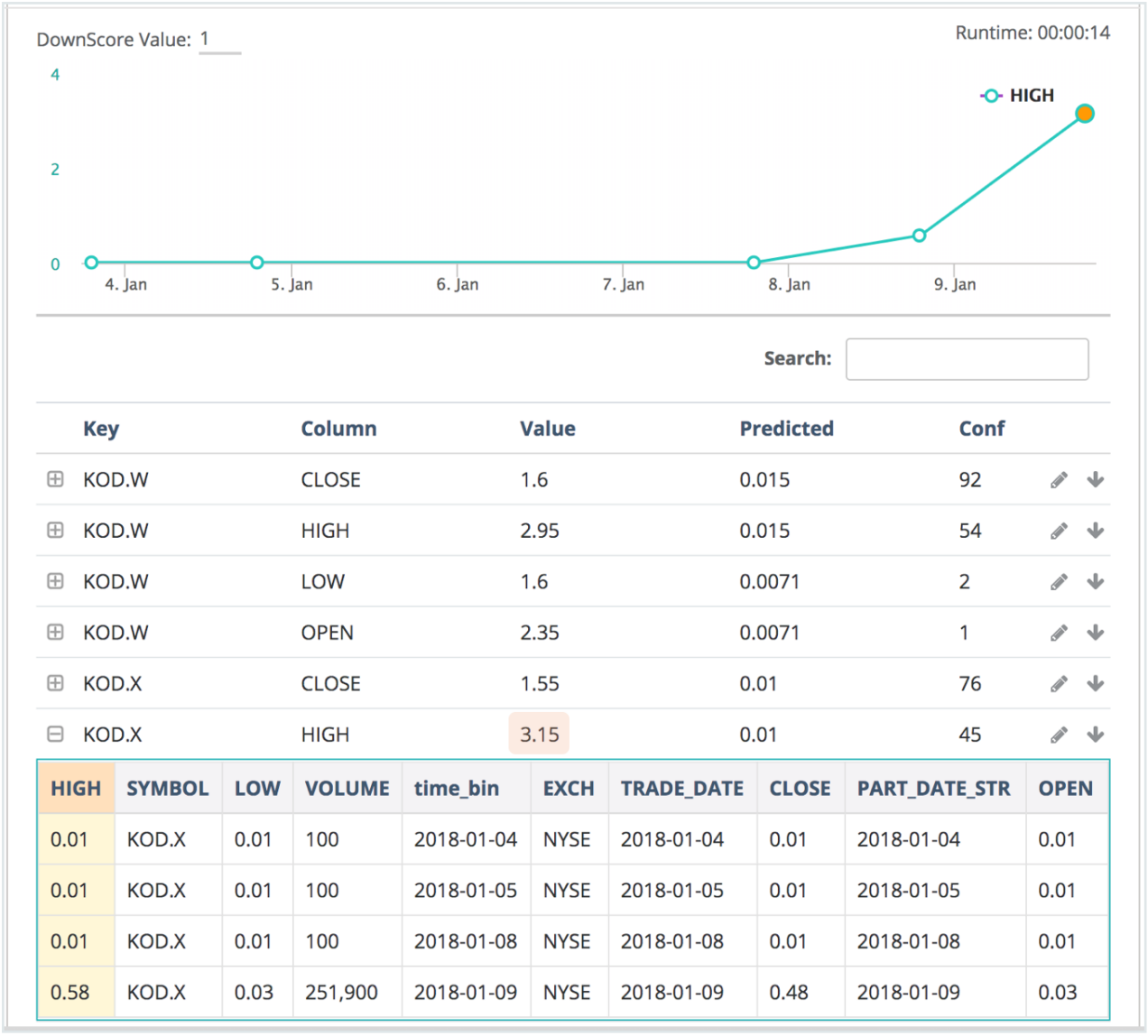
There is a known limitation where you cannot perform a dupes check on all columns or more than 10 columns at once.

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.

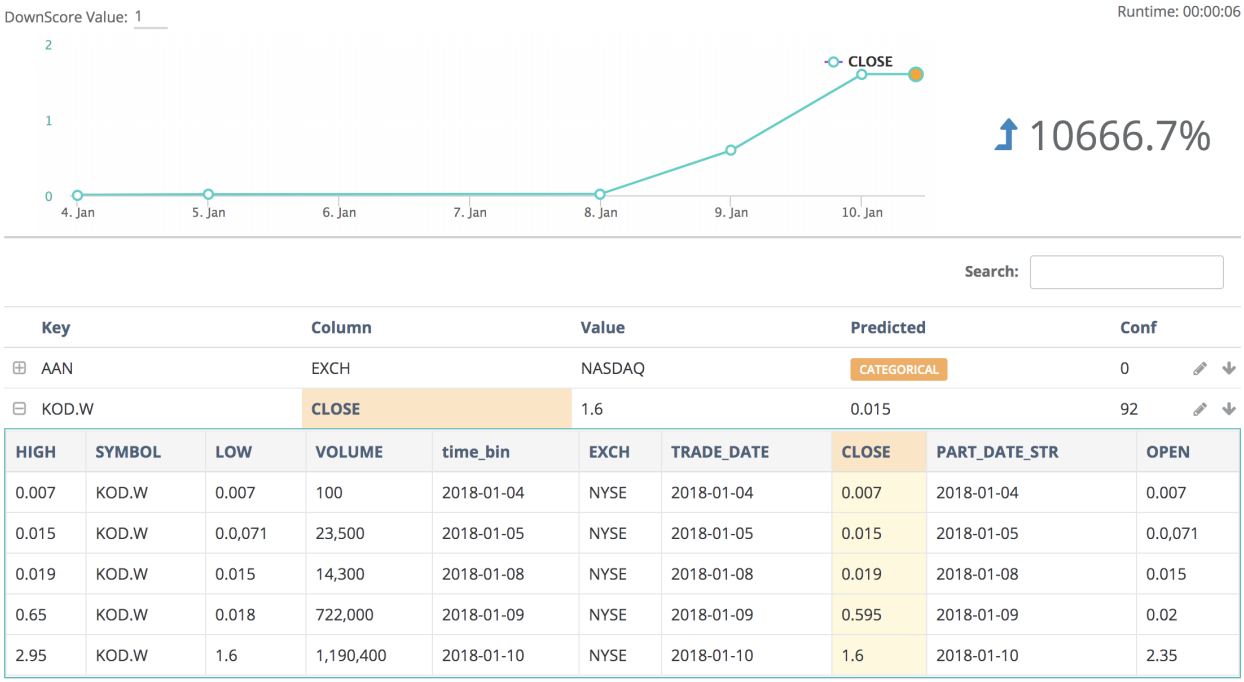


Key	Column	Value	Predicted	Conf
KOD.W	CLOSE	1.6	0.015	92
KOD.W	HIGH	2.95	0.015	54
KOD.W	LOW	1.6	0.0071	2
KOD.W	OPEN	2.35	0.0071	1
KOD.X	CLOSE	1.55	0.01	76
KOD.X	HIGH	3.15	0.01	45

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.



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.

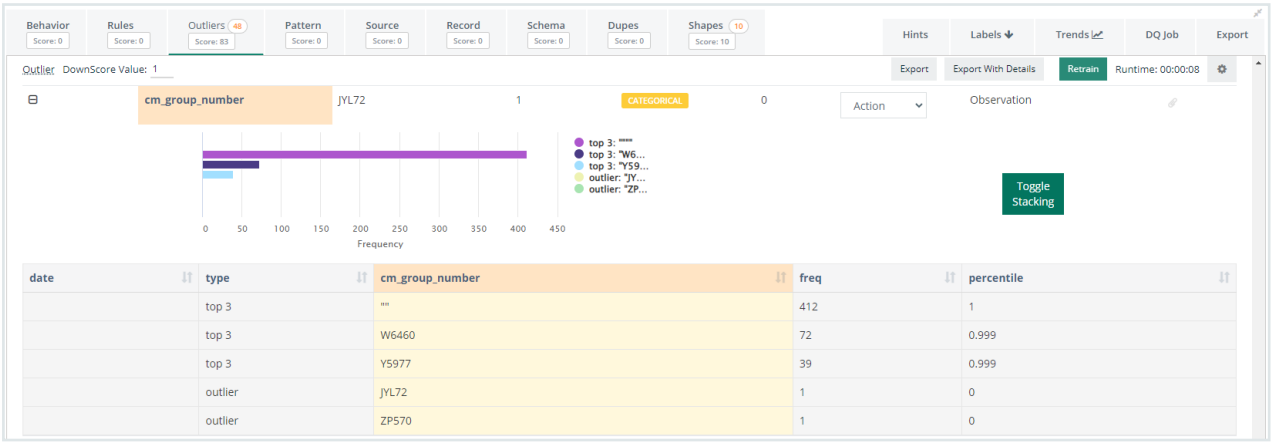
Categorical Outliers

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

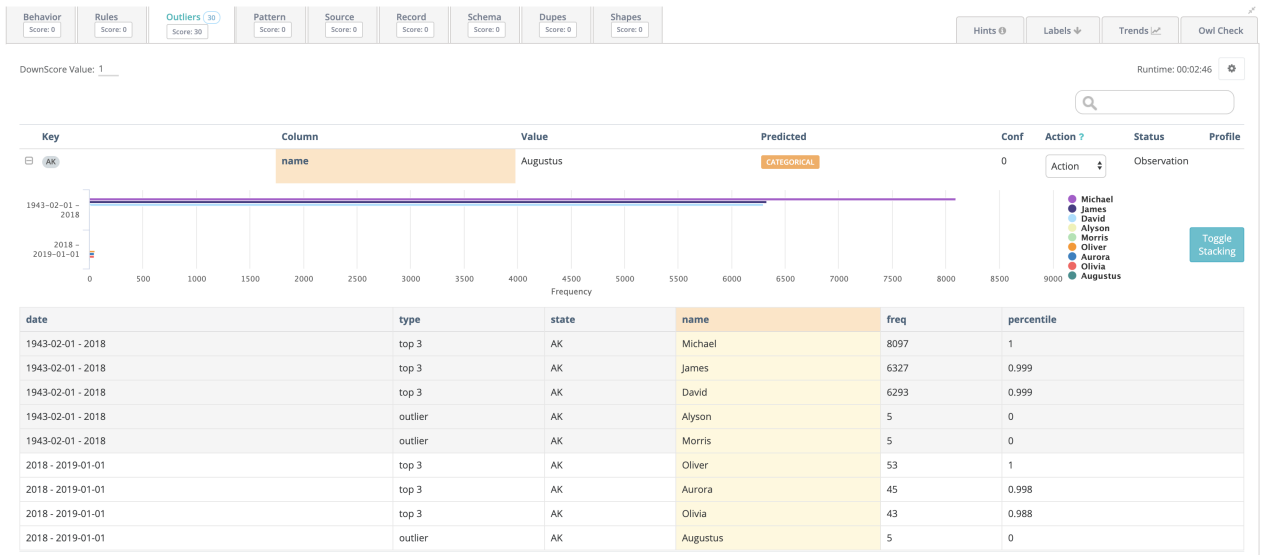
Collibra DQ will automatically learn the normal behavior of your String and Categorical attributes such as STOCK,OPTION,FUTURE or state codes such as MD,NC,D.C. When a strange pattern occurs (e.g NYC instead of NY), Collibra DQ will show this as a categorical outlier.

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).



date	type	cm_group_number	freq	percentile
	top 3	""	412	1
	top 3	W6460	72	0.999
	top 3	Y5977	39	0.999
	outlier	JYL72	1	0
	outlier	ZP570	1	0

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.

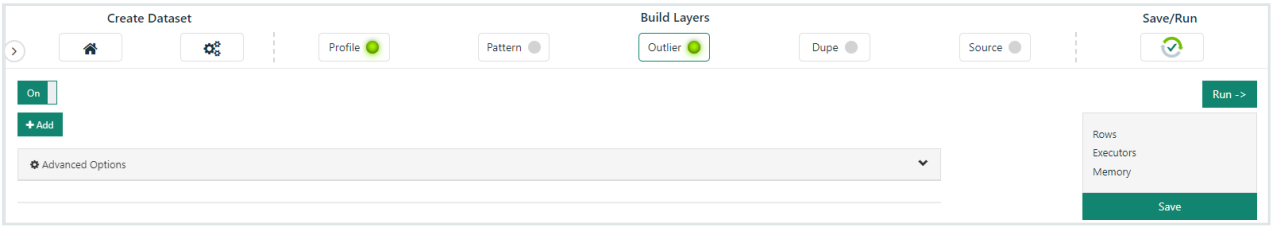


date	type	state	name	freq	percentile
1943-02-01 - 2018	top 3	AK	Michael	8097	1
1943-02-01 - 2018	top 3	AK	James	6327	0.999
1943-02-01 - 2018	top 3	AK	David	6293	0.999
1943-02-01 - 2018	outlier	AK	Alyson	5	0
1943-02-01 - 2018	outlier	AK	Morris	5	0
2018 - 2019-01-01	top 3	AK	Oliver	53	1
2018 - 2019-01-01	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

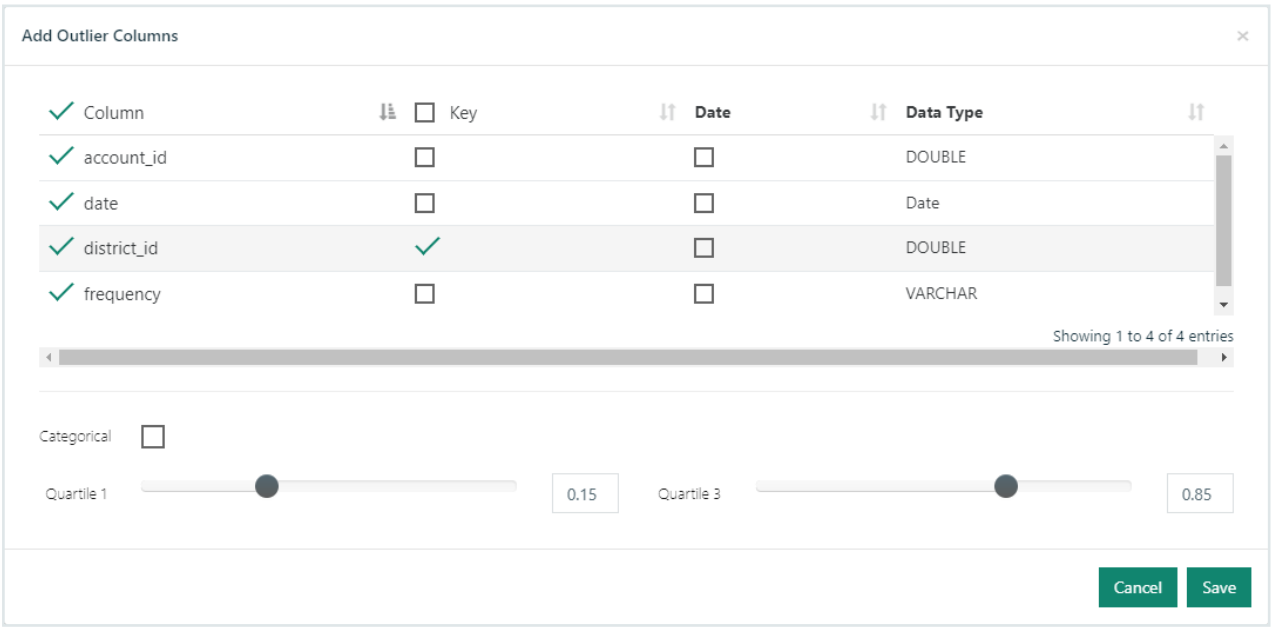
Training Outlier Detection Model

Although Collibra DQ uses different techniques to detect Numerical and Categorical Outliers, the training process is very similar.

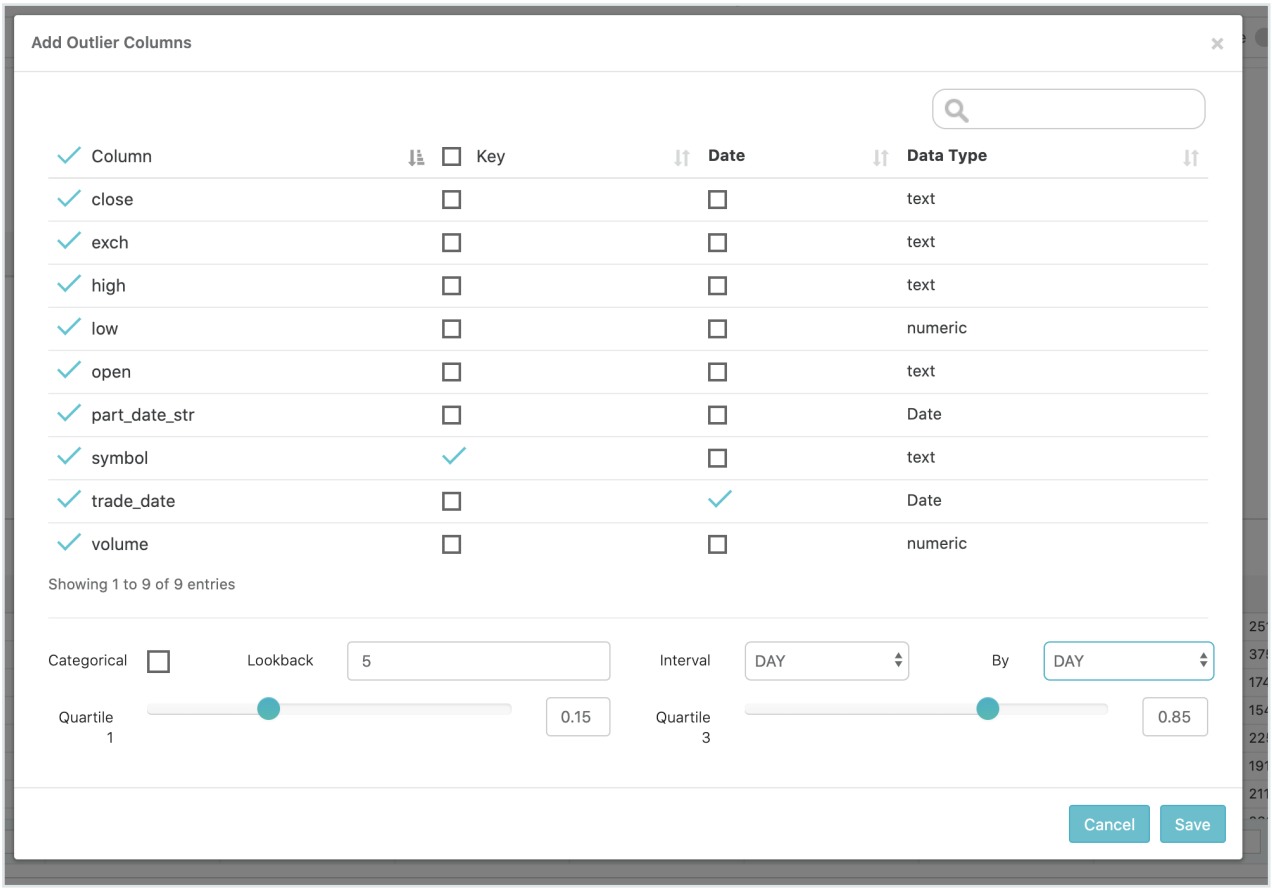
At a minimum, Collibra DQ requires historical data that can be used as the training dataset. If no other input is provided, Collibra DQ will calculate the normal range for each selected column and look for numerical and categorical outliers within the training dataset without any further context. The output will essentially consist of infrequent values that fall outside the normal range of each column.



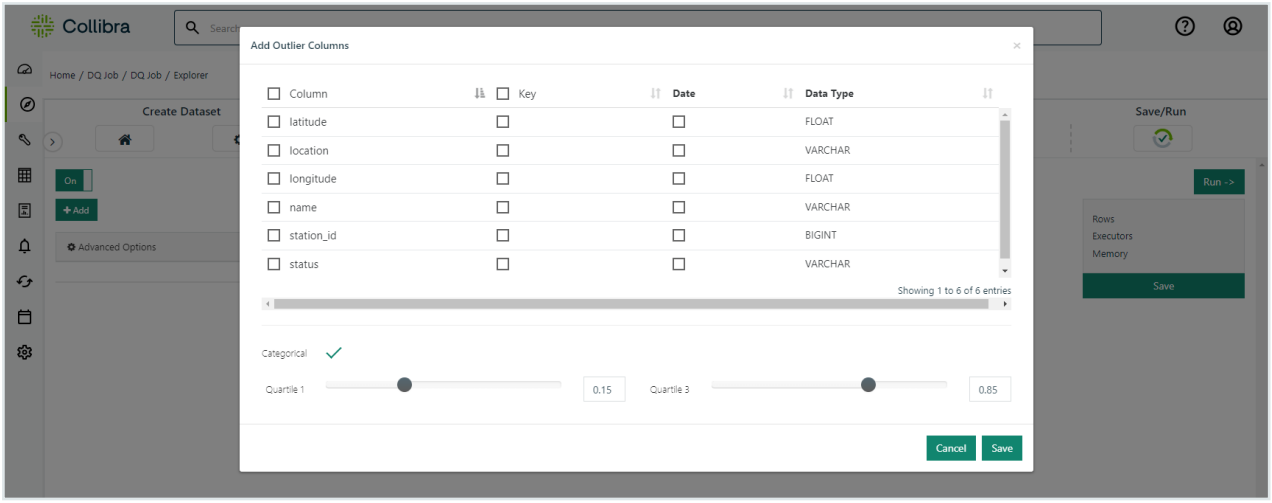
To obtain more targeted results, the Collibra DQ requires a "key" column. This column will be used to provide context by grouping each column by the key column. Defining a good key column tends to provide results that are a better indicators of actual data quality issues instead of simply infrequent values.



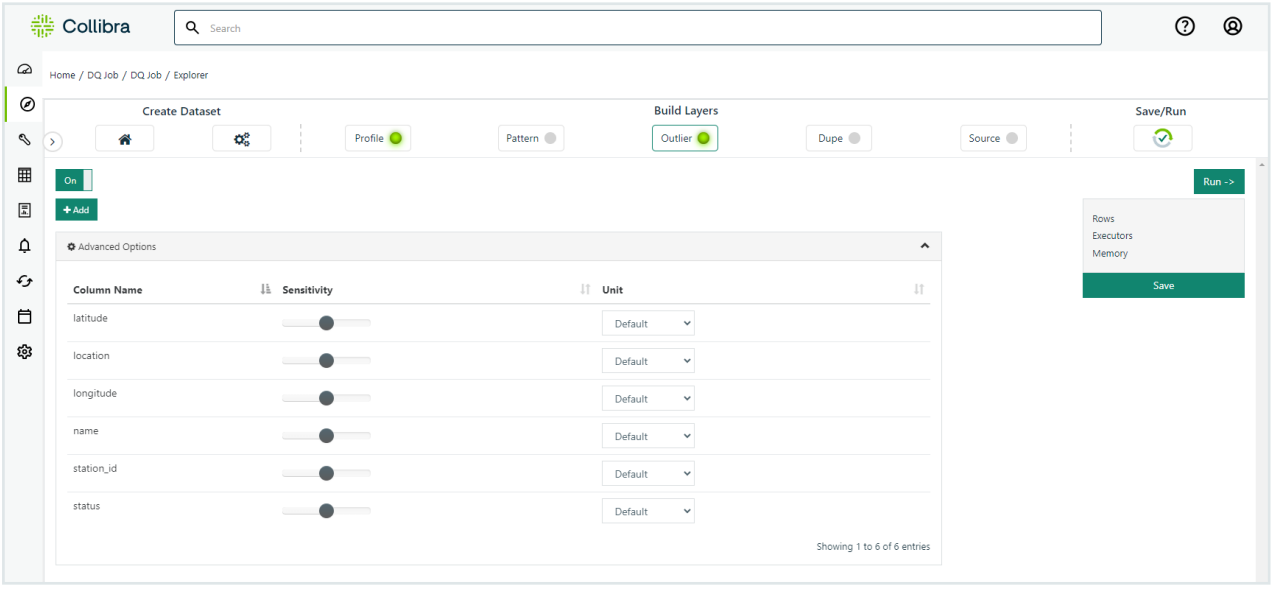
Another input that can make outlier detection more precise is a date/time column and a look back period. This enables a more precise calculation of the normal range for a column and in the case of numerical outliers, makes it possible for Collibra DQ to establish a trend. Given a time column and key column, Collibra DQ will not only identify numerical outliers, it will plot the historical trend of the column value trailing the outlier.



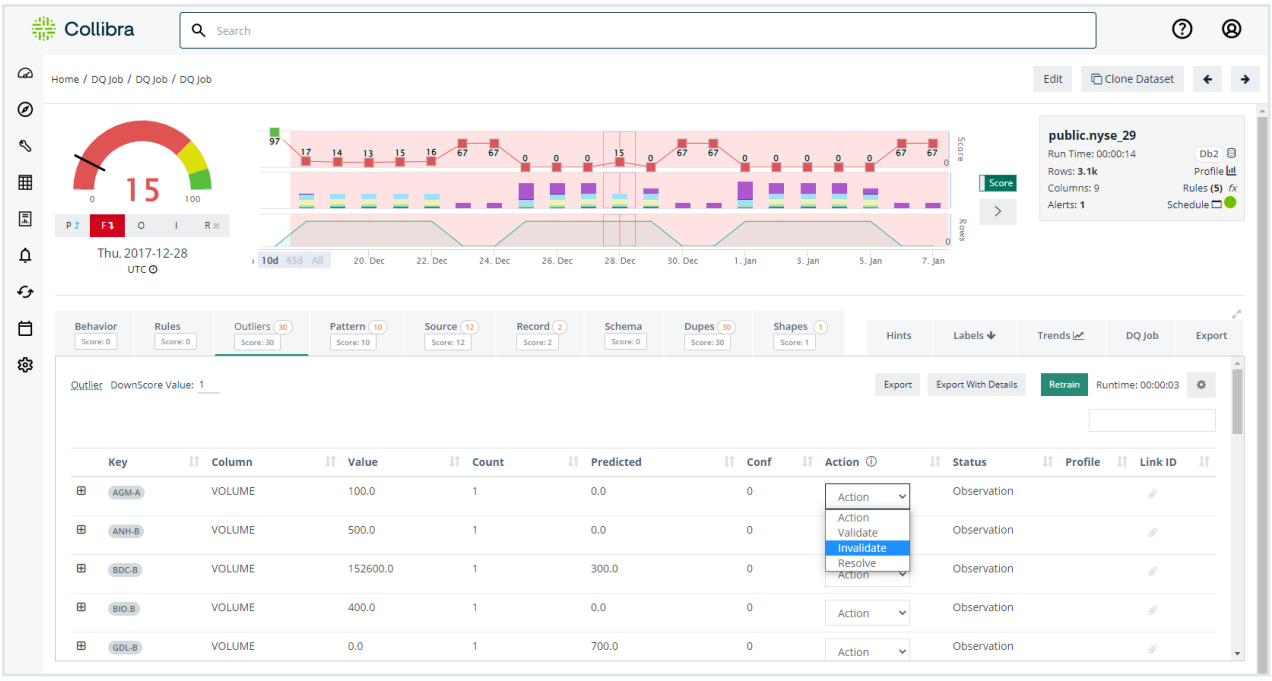
Collibra DQ also allows further refinement of the time dimension by defining time bins and processing intervals. By default, when given a time column, Collibra DQ will bin the data into days and process the data in daily interval. However, if the data is high frequency, day bins and day intervals might 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

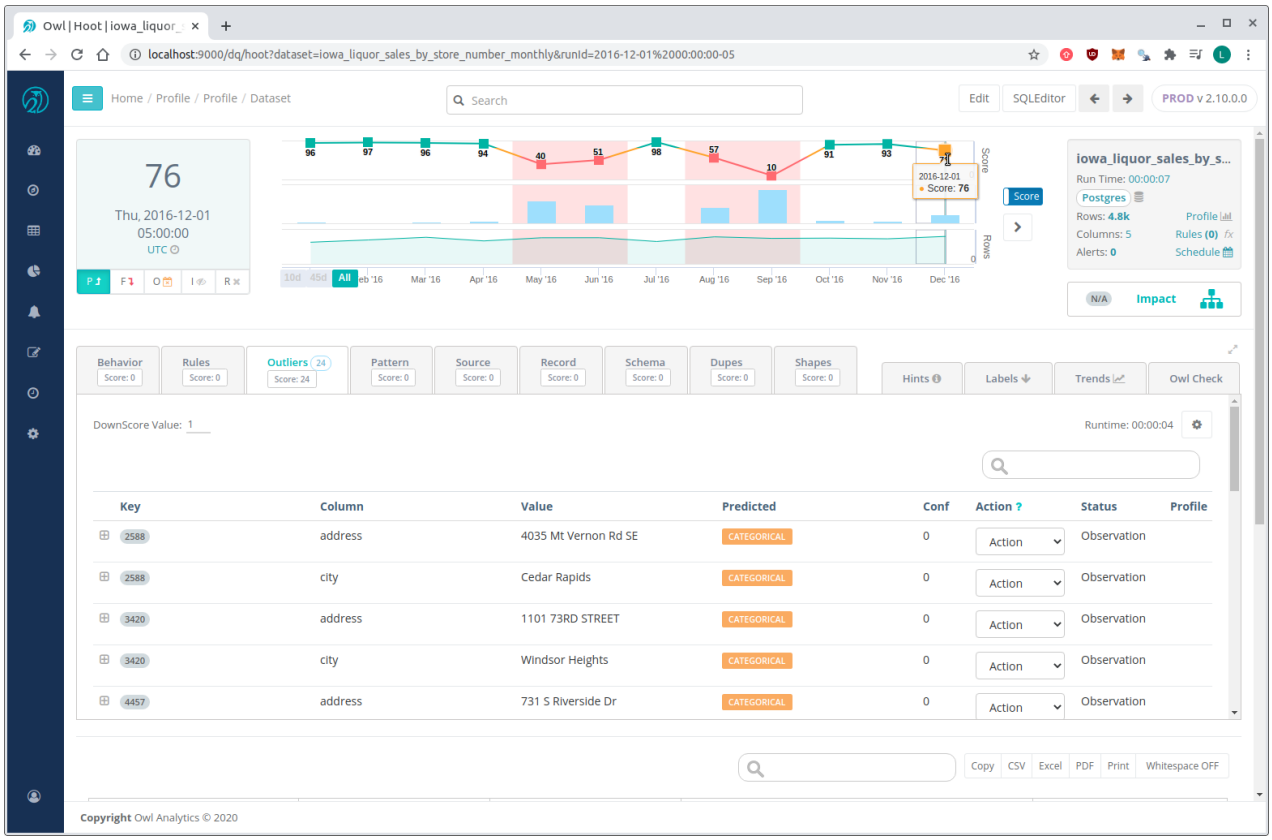
one by one. With over 2,000 unique stores, 19 million rows, and 8 years of data, you need an automated way to detect meaningful categorical outliers.

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`.

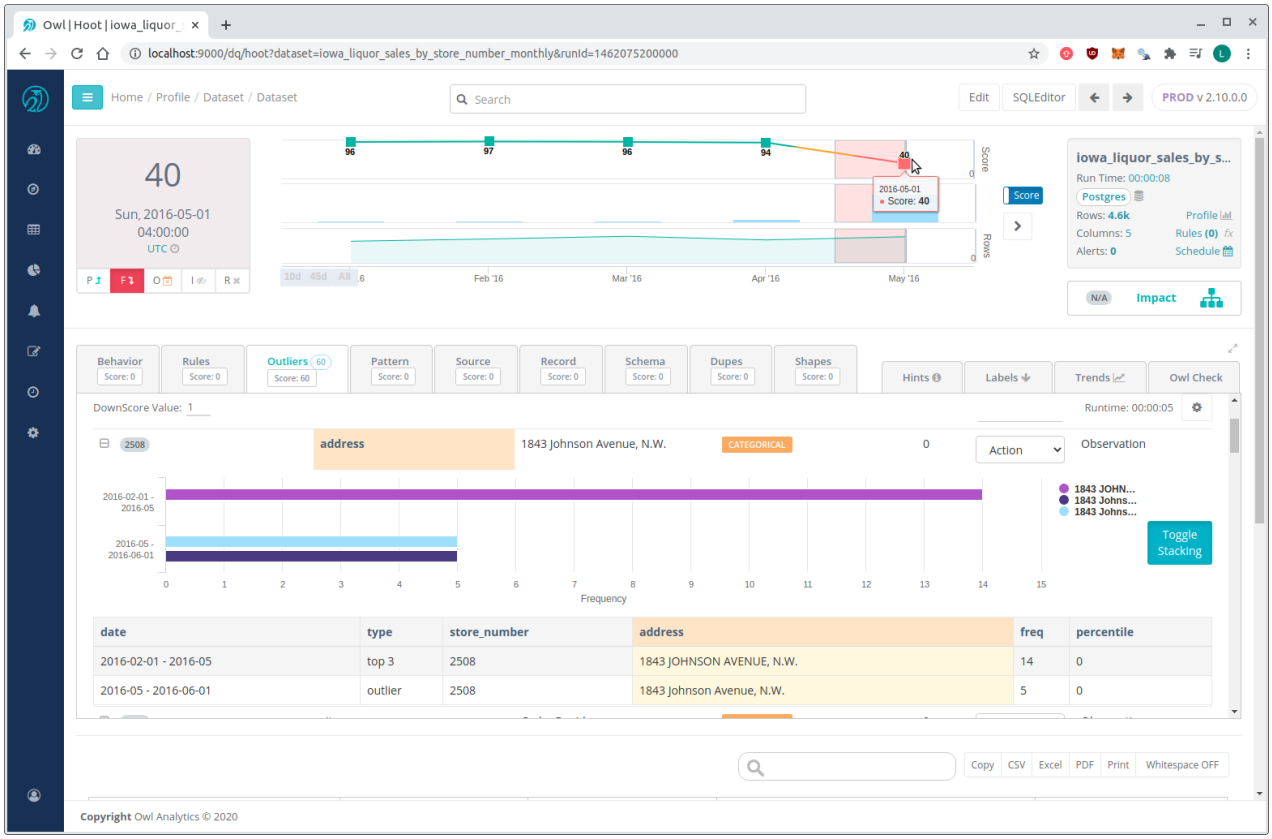
```
/opt/owl/bin/owlcheck
  # connection information to data
  -lib "/opt/owl/drivers/postgres/" -driver "org.-
postgresql.Driver"
  -c, "jdbc:postgresql://localhost:5432/postgres"
  -u, "postgres", "-p", "password"
  # Specify dataset name
  -ds "iowa_liquor_sales_by_store_number_monthly"
  # Specify 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).

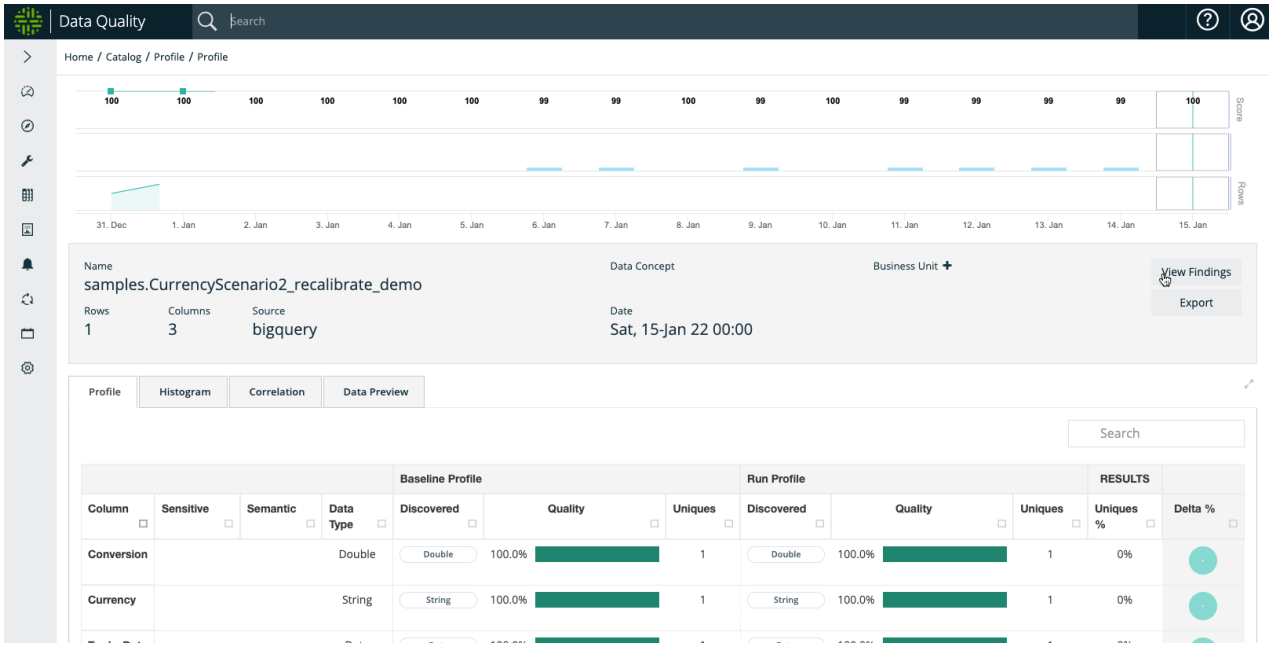
Tech Preview [TP] Outlier Calibration

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.

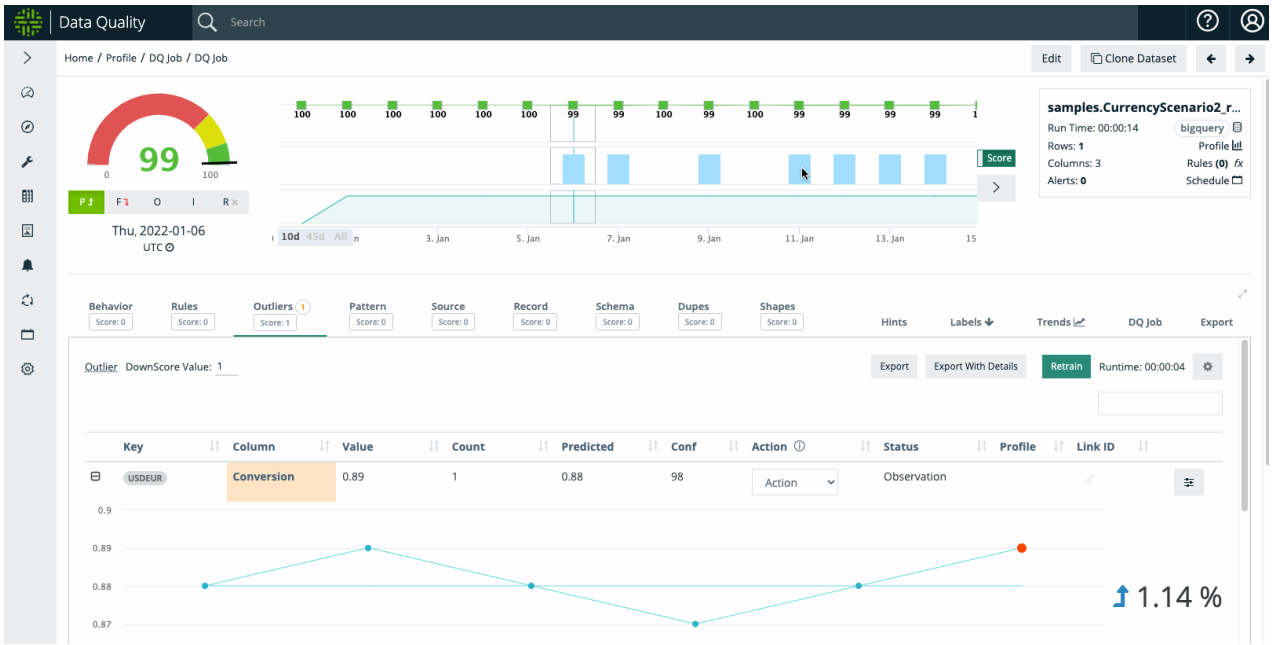
Example Step #1: No Action Necessary

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.



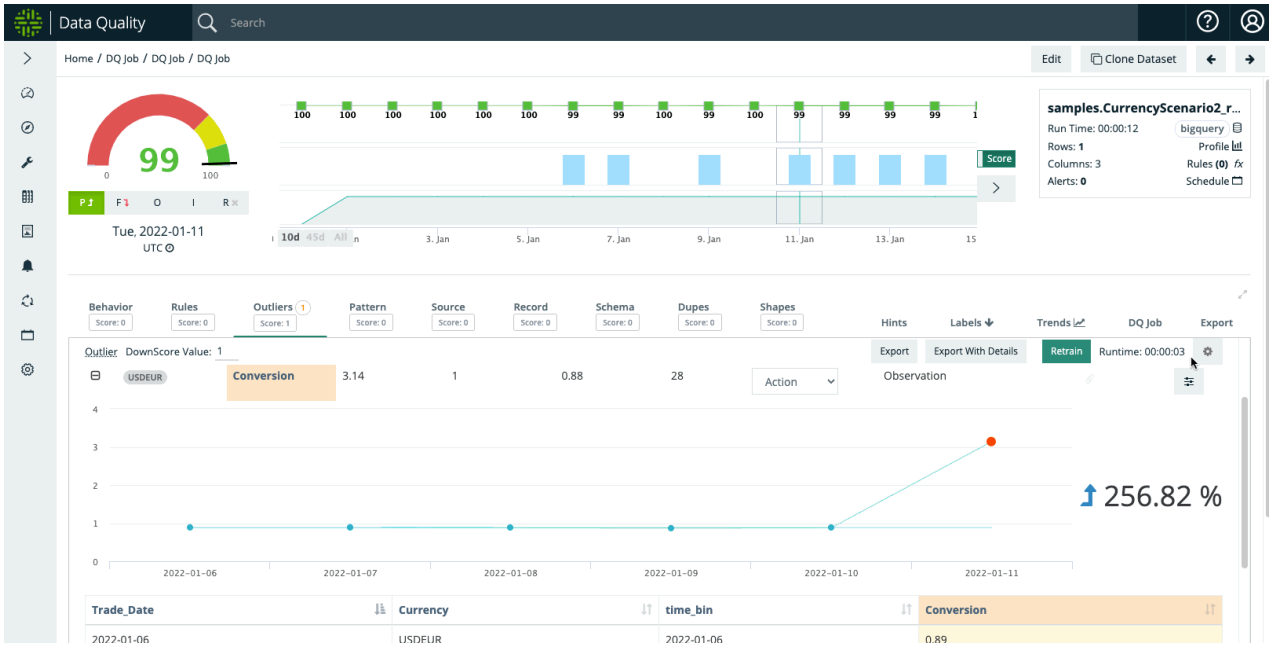
Example Step #2: Macro Event That User Understands e.g. Currency Devaluation or Stock Split

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

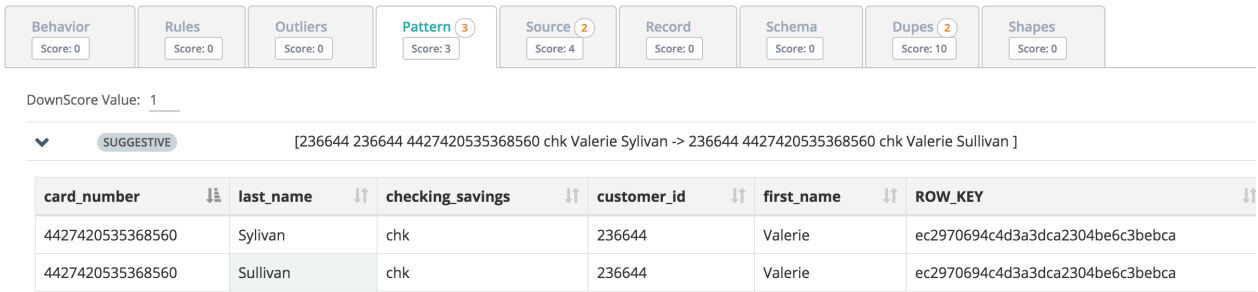
Once Outlier Calibration is enabled, a user can select the boundaries and duration of the 'suppression' period. And once the DQ Job is re-run for the selected date(s), the outliers will not trigger an anomaly / downscore.



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?

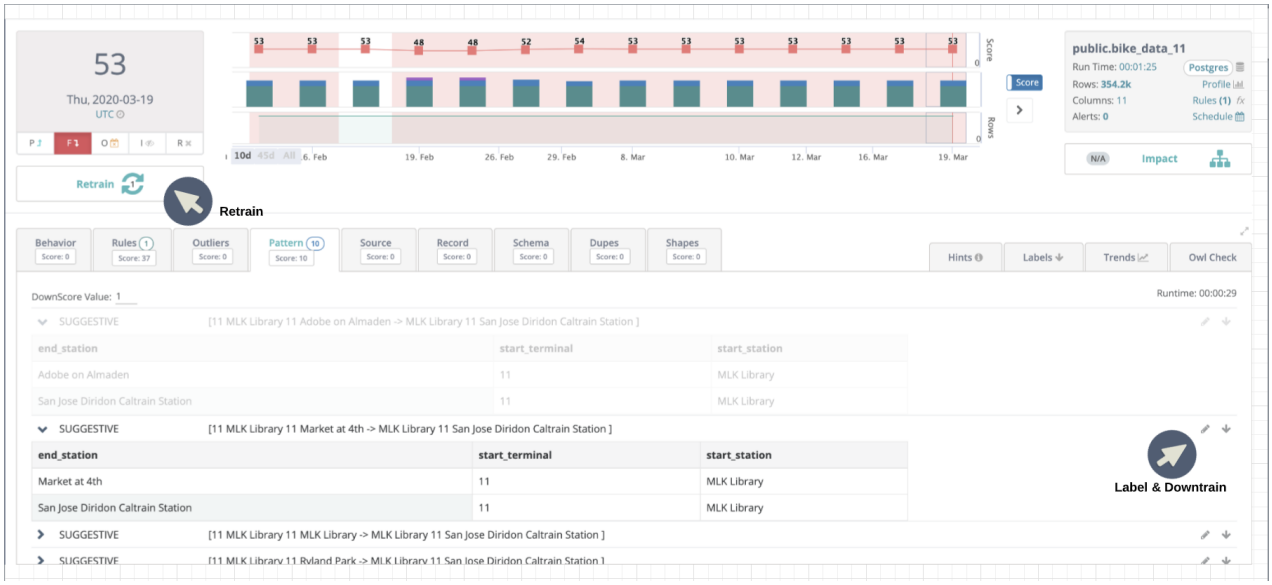


Training Anti-Pattern Detection Model

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?

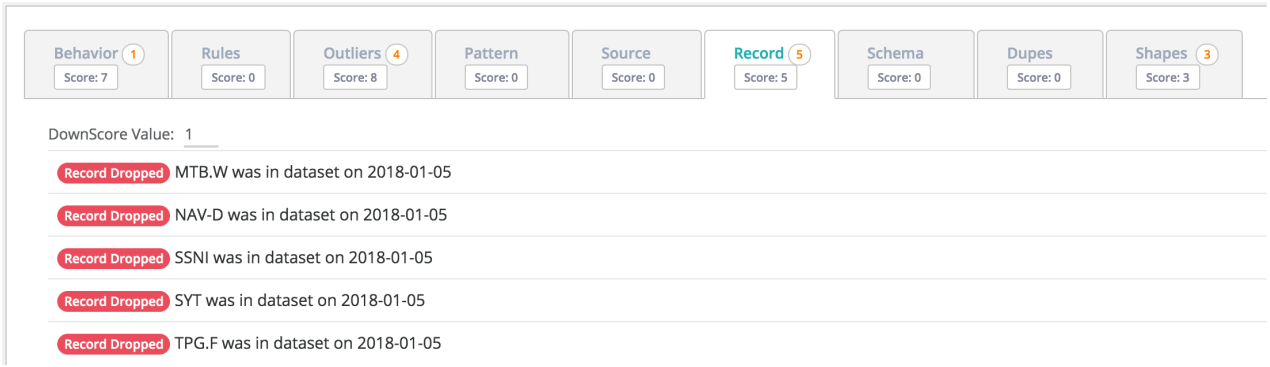
Think about a scenario where a dataset has a SSN column along with FNAME, LNAME and many others. What if your traditional rules engine passes because one of the rows has a valid SSN and a valid Name but the SSN doesn't belong to that person (his or her name and address, etc.)? This is where data mining can derive more sophisticated insights than a rules based approach.

Records (advanced)

> Note This is an advanced opt-in feature.

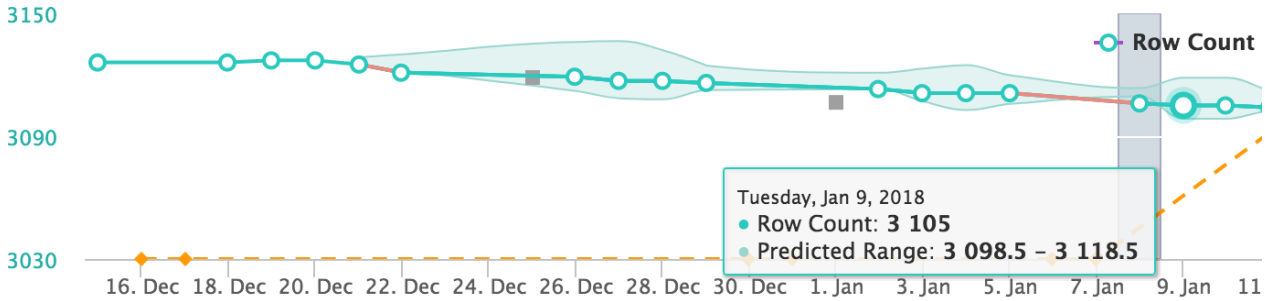
Where did my rows go?

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.

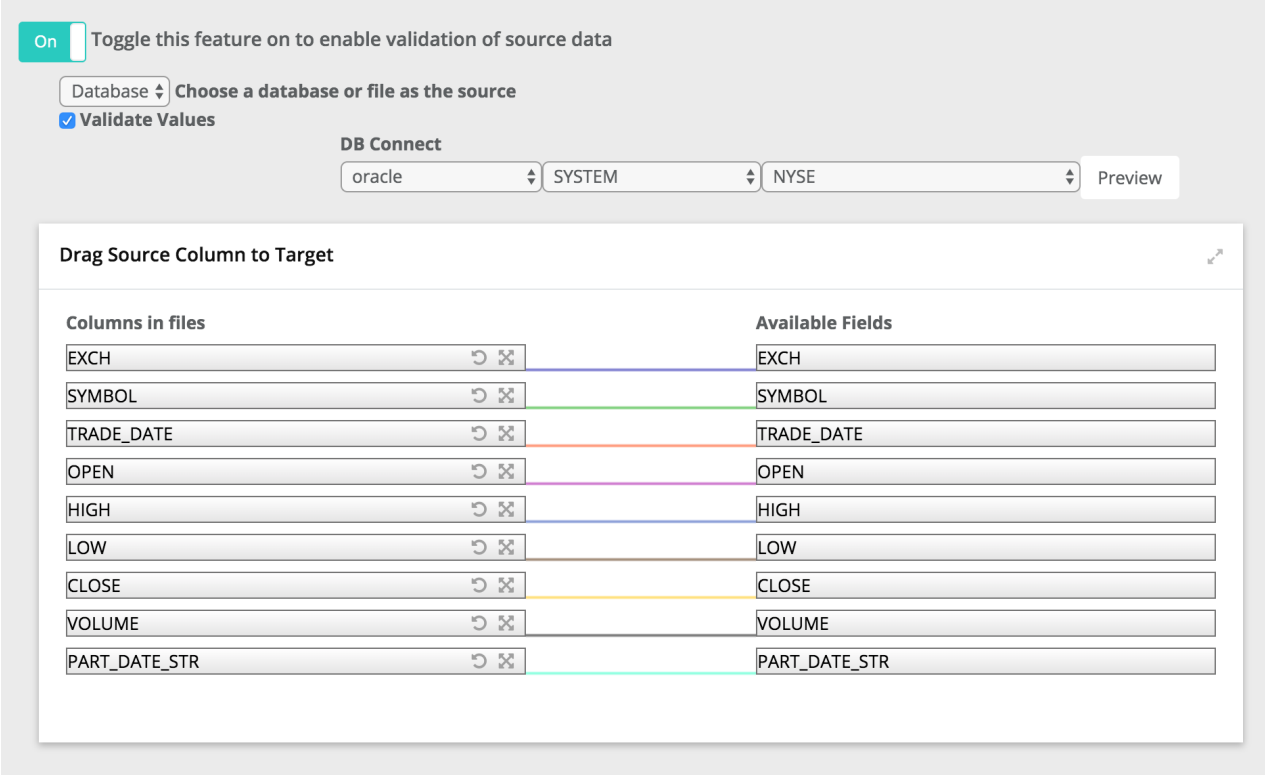
Does your data-lake reconcile with your upstream system?

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.



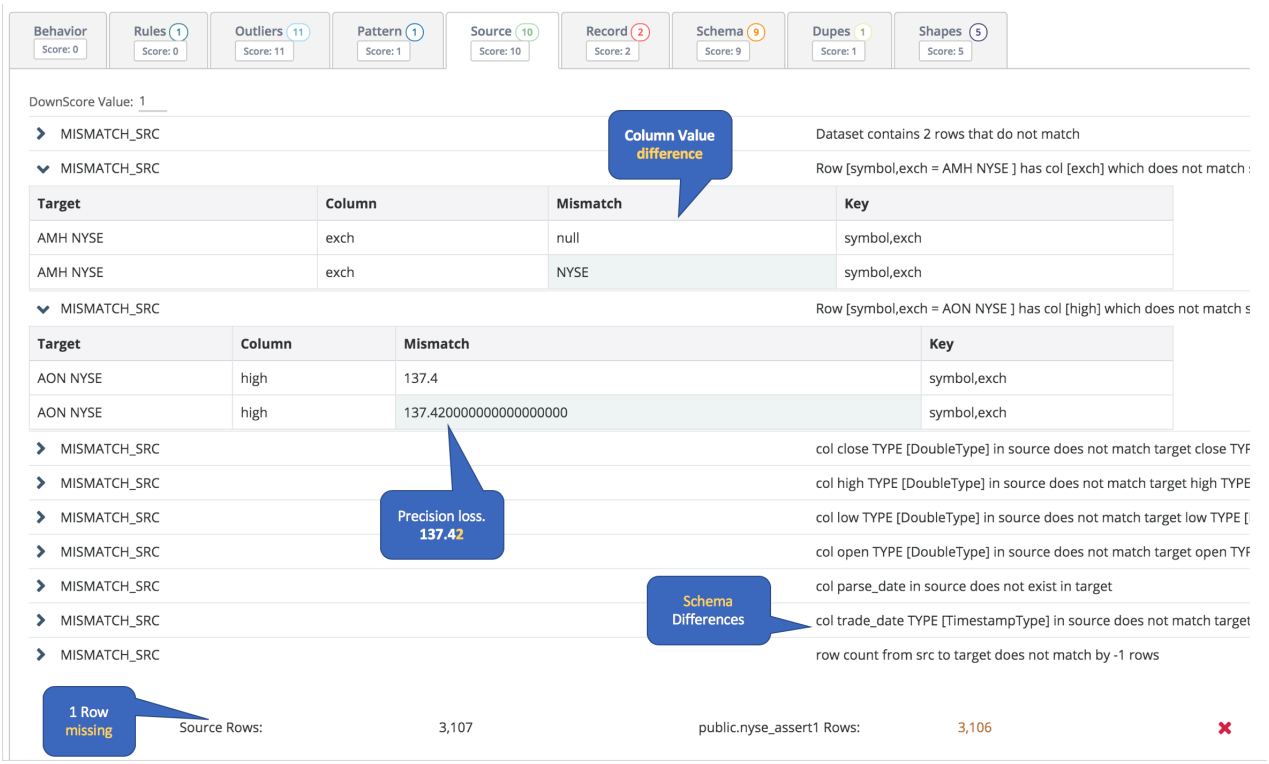
DQ Check Created from Wizard

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.

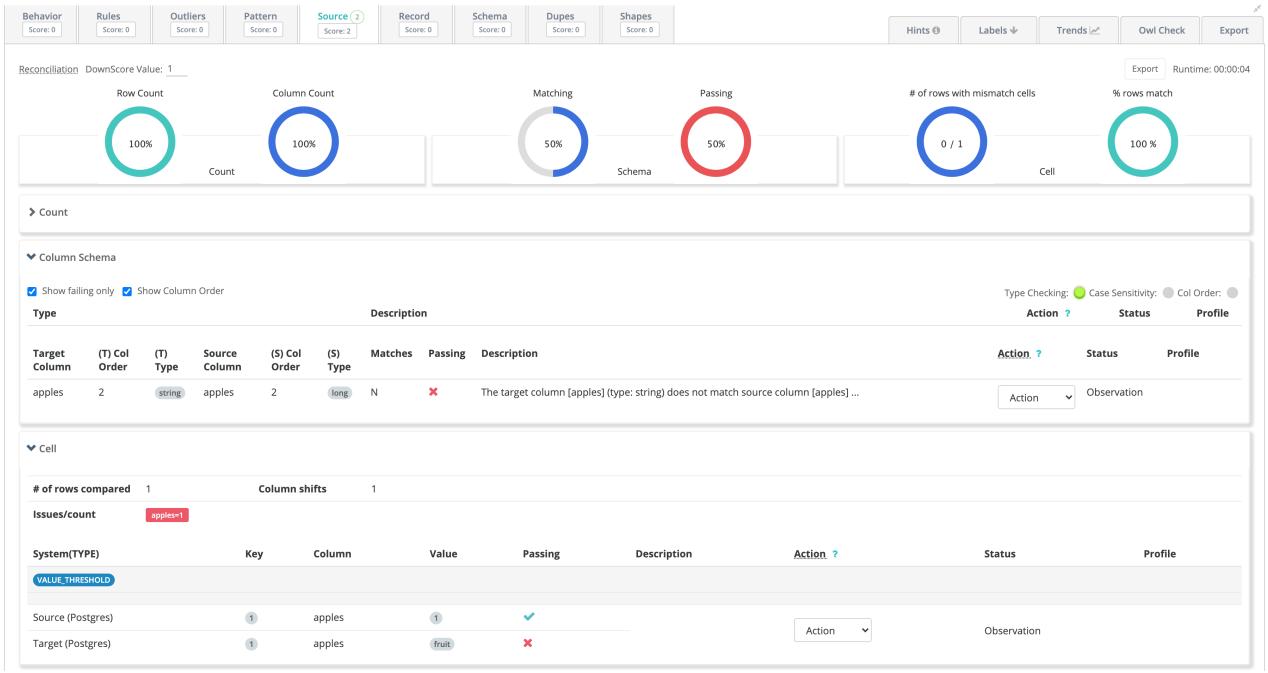
```
-lib /home/ec2-user/owl/drivers/valdrivers \
-driver org.postgresql.Driver \
-u user -p password \
-c "jdbc:postgresql://ec2-34-227-151-67.compute-1.amazon-
aws.com:5432/postgres" \
-q "select * from public.dateseries4" \
-ds psql_dateseries2 -rd 2018-11-07 \
-srcq select dz, sym as symz, high as 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.

Pushdown

> Important We're excited to offer a new Pushdown experience! The version of Pushdown described on this page was released as a beta feature for some data sources and had limited processing capabilities for both the Full Profile and Count options. Check out the new documentation page to learn more about the enhanced Pushdown experience.

Some of the Collibra DQ features support pushdown to avoid transferring large dataset from data source (Database, Cloud storage, file systems etc.) into Spark. When pushdown is enabled and supported, the DQ Job will generate SQL queries to offload the compute to the data source, reducing the amount of data transfer and Spark computation of the DQ Job. Not all features support pushdown nor pushdown completely eliminate data transfer.

Why use pushdown

To analyze a dataset for DQ findings as part of a DQ Job, CDQ uses Spark as the compute engine to power the analysis. This requires the dataset to be loaded into Spark as a Spark DataFrame, and the DQ Job performance (job completion speed) is limited by Spark resources available and the complexity of the job. This data transfer from data source to Spark is dependent on two of the following factors:

1. Bandwidth limitation at the data source
   If the DQ Job requires pulling 100 million rows from a SQL Database, then the Input/Output limit of transferring data out of SQL Database into the Spark cluster will greatly increase the overall speed of the DQ Job.

2. Compute limitation at the data source for complex queries
   If the DQ Job requires complex queries to create the dataset (via `-q`), then this computation is done at the SQL Database level. Some examples of "complex" queries are:
   `-q "SELECT * FROM public.very_long_table_of_transactions WHERE date = ${rd} and department = 'finance'"`
   This query can take long time at the database level due to `WHERE` clause filtering a very long table on columns `date` and `department` that may not be indexed properly, leading to a full table scan without a `LIMIT` clause specified.
   `-q "SELECT * FROM public.very_long_table_of_transactions transactions INNER JOIN public.departments departments ON transactions.department_id = departiments.id WHERE transactions.date = ${rd} and departments.name = 'finance'"`
   This query can take long time at the database level due to a join between tables.
3. Resource limitation at the data source
   The data source may not have enough hardware resource available to efficiently fulfill the query and/or handle multiple DQ Jobs & other non-CDQ applications requesting data.

The bottleneck due to data transfer can be viewed in Jobs page > Job Logs > LOAD stage. This data loading step is the first step for all DQ Jobs. However, pushdown feature can be used to *reduce* (NOT eliminate) the data transfer & compute to Spark from the data source, if the DQ Job specified does not require loading all the data into Spark.

In summary, speed of loading the data from data source to Spark is a `-q query compute time at data source` + `network transfer of -q result between data source and Spark`. Pushdown can reduce both elements, but the efficiency gained is dependent on the complexity of `-q` query and how big the dataset from `-q`result would have been without pushdown.

How pushdown is efficient

Some of the Spark compute performed by the DQ Job can be translated into SQL queries that most relational databases support natively. In such a case, the DQ Job does not need to load all the rows of the dataset. Instead, the DQ Job can query the data source for the results of those SQL queries and reduce the amount of data transferring out of the data

source. The results of these SQL queries are almost always lead to smaller amount of data compared to the full dataset defined by `-q`. Only some of the DQ Job features require the full dataset to be loaded into Spark. Therefore, pushdown can be a useful tool to speed up the overall DQ Job speed -- **provided that the speed of executing these SQL queries are faster than the speed of transferring the data out of the data source into Spark**. In most use cases, pushdown leads to faster DQ Job execution for large datasets. If the `-q` query is sufficiently complex, then the speed reduced by transferring less data into Spark can be cancelled out by the multiple frequent SQL queries made to the data source by the Pushdown process (because each query may have have redundant compute due to the complexity of `-q`).

How pushdown works in CDQ

Using pushdown only *reduces* the amount of data transferred out of the data source. It does NOT skip the LOAD stage in DQ Job. Every DQ Job requires a small sample of rows (10-20) of the dataset defined by `-q` in order to generate Data Preview and analyze schema information for the dataset run. This means the `-q` query may be fully computed at data source before the sampling can occur (depending on the complexity of the `-q`). In such a case, sampling 10-20 rows of data is not a quick and immediate LOAD stage and only efficiency gain comes from lack of transferring data between data source and Spark.

Therefore, pushdown feature would be most efficient if `-q` is a simple select query with simple where filtering. The benefit comes from the fact that if your dataset defined by `-q` results in 100 million rows, only 10-20 rows of the dataset defined by `-q` will be loaded into Spark.

Profile with pushdown will then generate a series of SQL queries and query the data source again for aggregate metric data. Depending on the dataset, these multiple SQL queries can be more efficient than loading all the data into Spark and computing these aggregate metric in Spark. The results between *Profile* with *pushdown* and *Profile* without *pushdown* are (practically) identical.

Profile: pushdown vs no pushdown

Here is the summary of Profile activity with details regarding pushdown support

Feature	Supports pushdown	Description
Row Count	Yes	Computes row count of the data-set.
Distinct Count	Yes	The number of distinct values in a column.
Mean	Yes	The average of all the values in the column. Supports numeric columns only.
Min / Max	Yes	The minimum and maximum values of the column. Supports numeric and boolean columns only.
NULL Count	Yes	The number of null values in the column.
EMPTY Count	Yes	The number of empty values in the column. Supports string columns only.
TYPE Count	Yes	The number of different types inferred in a column (if any).
TopN / BottomN	No	Computes the top 5 most frequent (TopN) and least frequent (BottomN) values. Supports all types. This result is displayed as a frequency bar chart in Profile page. If pushdown is enabled, then TopN and BottomN values are not displayed. Related features like Stat Rules (Distribution) are also disabled.
Data Shape Detection	No	Detects shapes of the values based on Shape parameters provided (automatic or manual).

Feature	Supports pushdown	Description
Histogram	No	Creates histogram of the values in the column.
Correlation Matrix	No	Creates correlation matrix. Supports only numeric columns.

Snowflake Pushdown

> Important This page has moved. Please visit Pushdown processing for general information about Pushdown or Setting up Pushdown for a Snowflake connection.

Features and Limitations of Snowflake Pushdown

> Important This page has moved. Please visit Pushdown processing for general information about Pushdown or Setting up Pushdown for a Snowflake connection.

Running a Snowflake Pushdown job

> Important This page has moved. Please visit Pushdown processing for general information about Pushdown or Setting up Pushdown for a Snowflake connection.

Running a Full Scan

> Important This page has moved. Please visit Pushdown processing for general information about Pushdown or Setting up Pushdown for a Snowflake connection.

Running a Partial Scan

Important This page has moved. Please visit Pushdown processing for general information about Pushdown or Setting up Pushdown for a Snowflake connection.

Scanning with SQL Query

Important This page has moved. Please visit Pushdown processing for general information about Pushdown or Setting up Pushdown for a Snowflake connection.

Chapter 11

Collibra DQ Workflows

Manage Collibra DQ Assignments

Collibra DQ allows you to assign DQ Job findings (observations) to other users to review those findings and make determinations based on whether the findings are legitimate.

Why Would I Assign DQ Job Findings to Users?

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 Collibra DQ (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.

> Note The default configuration for assignments is CDQ, however, you can also configure external ServiceNow assignments in the Validate Finding window.

Assign Job Findings to Users

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).

> Note The default configuration for assignments is CDQ, however, you can also configure external ServiceNow assignments in the Validate Finding window.

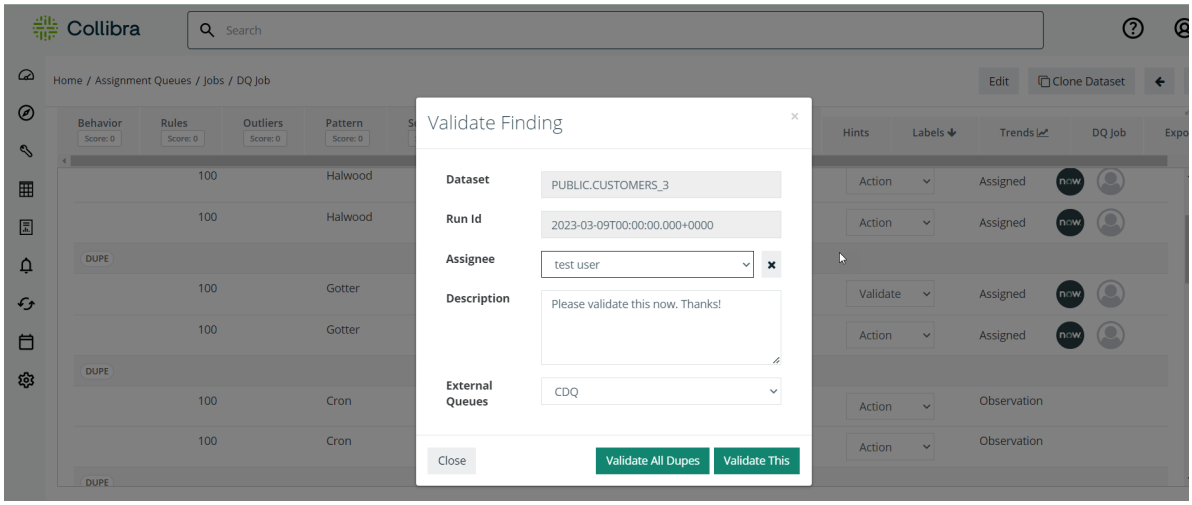
> Warning To configure a queue, you must have ROLE_ADMIN or ROLE_ CONNECTION_MANAGER.

Steps

1. Login to the Collibra DQ instance.
2. Select one of the findings of a previously run DQ Job.
3. From the **Actions** dropdown, select one of the following options.

Option	Description
Validate	Value is expected and represents the truth. Selecting this option means that you want the finding to be reported in the next job run.
Invalidate	Value is not what is expected and must be modified and fixed. You can set to Off-Peak, Ignore, or Remove so that the finding is removed from that day's run.
Resolve	Resolve the job finding. Resolving a finding retrains your Job's quality score if any points were deducted.

4. To validate a finding and assign it to an internal Collibra DQ user, select **Validate** from the **Actions** dropdown.
   The Validate Finding window displays.



5. Complete the following fields.

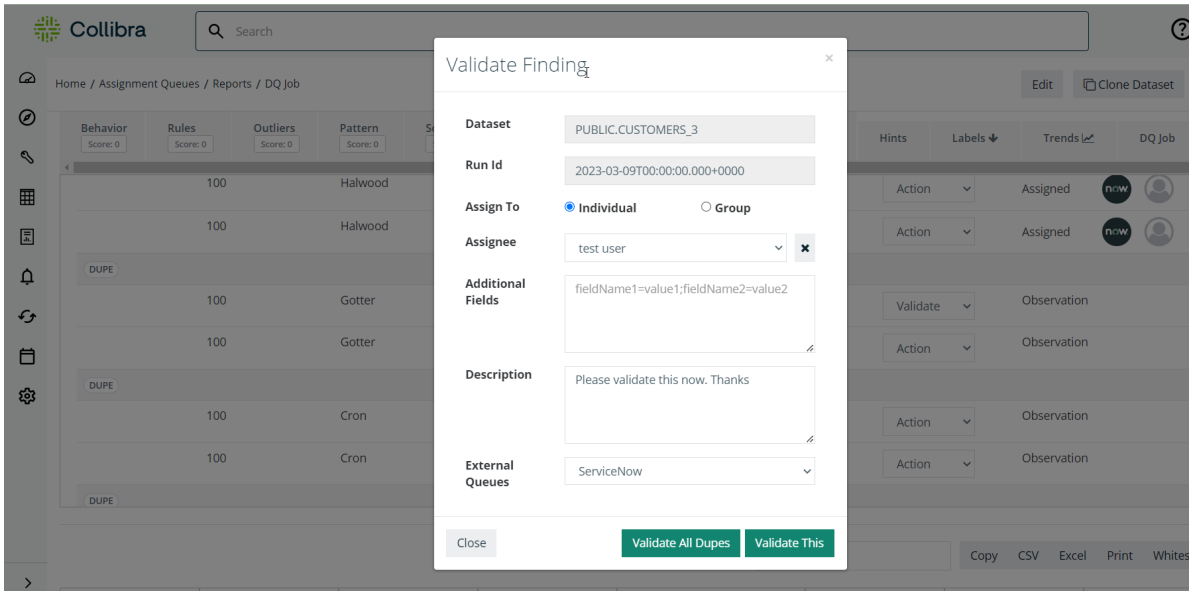
Field	Description
Dataset	Dataset of the finding. This field is already populated.
Run Id	ID of the DQ job run. This field is already populated.
Assignee	Collibra DQ user. Select the user from this dropdown.
Description	Enter a message for the user, such as "Please validate now".
External Queues	Select the internal/external user from this dropdown. Note CDQ is the default value.

6. To validate this finding, click **Validate This**.

> Note You can validate all the behaviors at once by selecting **Validate All <Behaviors>**. To close the window without saving, click **Close**.

7. To validate a finding and assign it to an external user (with a ServiceNow account), select **Validate** from the **Actions** dropdown.
   The Validate Finding window displays.
8. Select **ServiceNow** from the **External Queues** dropdown and complete the following fields.

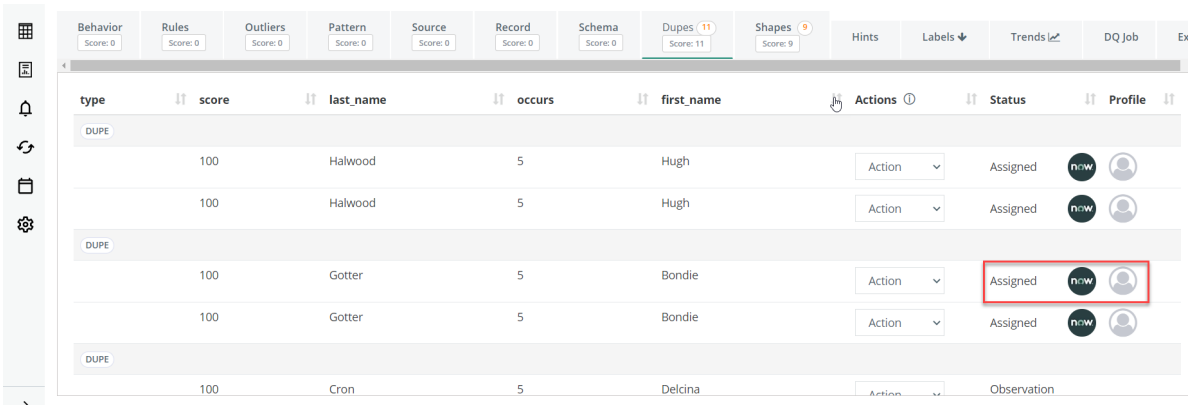
Field	Description
Dataset	Dataset of the finding. This field is already populated.
Run Id	ID of the DQ job run. This field is already populated.
Assign To	Select one of the following options: ◦ **Individual** - Assign to one user ◦ **Group** - Assign to a group of users
Assignee	Collibra DQ user. Select 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** from this dropdown. Note CDQ is the default value.



9. To validate this finding, click **Validate This**.

> Note You can validate all the behaviors at once by selecting **Validate All <Behaviors>**. To close the window without saving, click **Close**.

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.



10. To invalidate a finding, select **Invalidate** from the **Action** dropdown and add a note in the Annotation* field.

> Note The Annotation* field is a required field.

11. Click **Save**, **Save & Retrain**, or **Close** to close without saving.

Known limitations

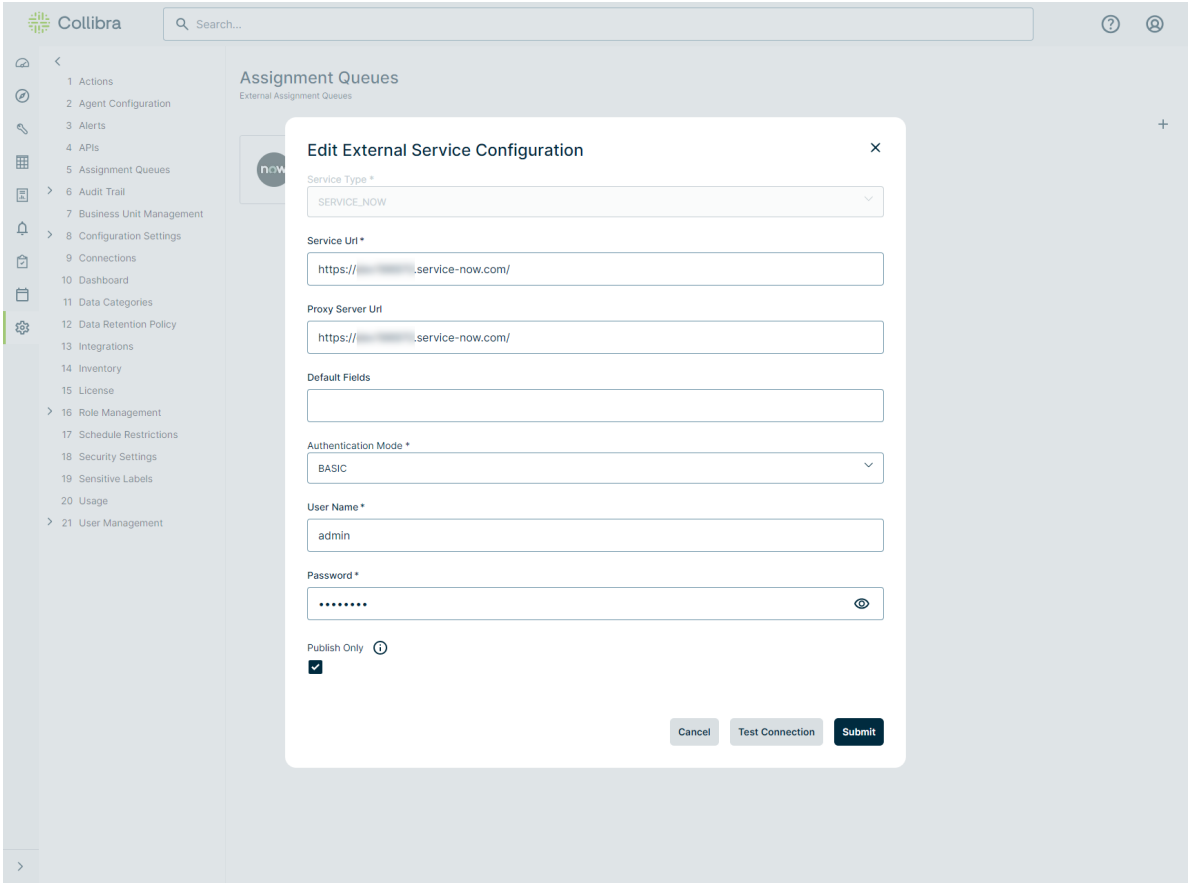
- The ServiceNow integration does not support SSO users currently.

Configure an External Assignment Connection

Add or edit an external service configuration connection in the Admin Console.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
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.
> You can also click **Close** to close the window without saving.

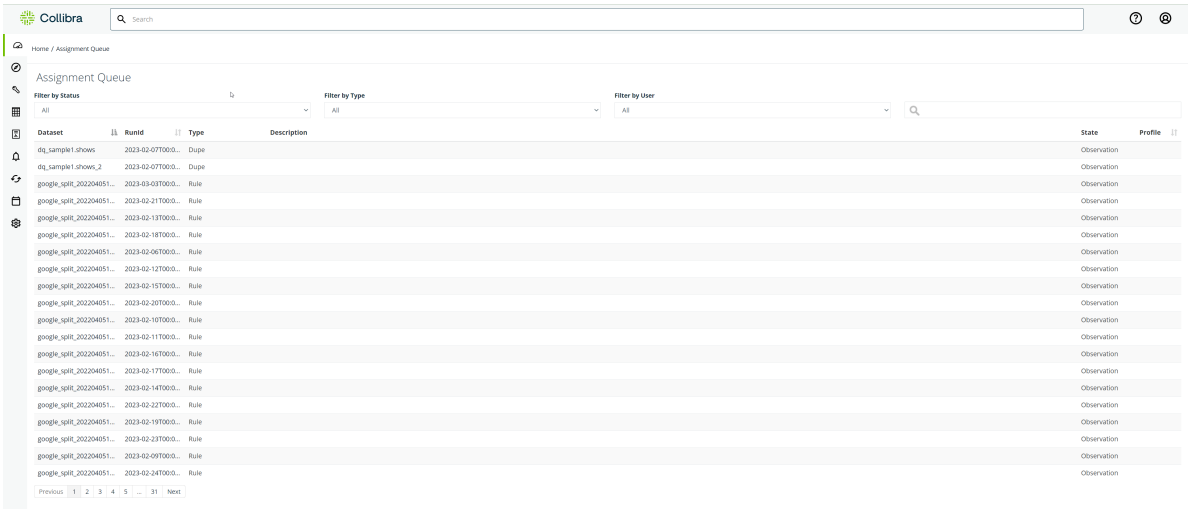
View the Assignment Queue

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.

> Note You can view the Assignments Queue via the icon or via the Workflow option on the Collibra Data Quality & Observability Home page.

Steps

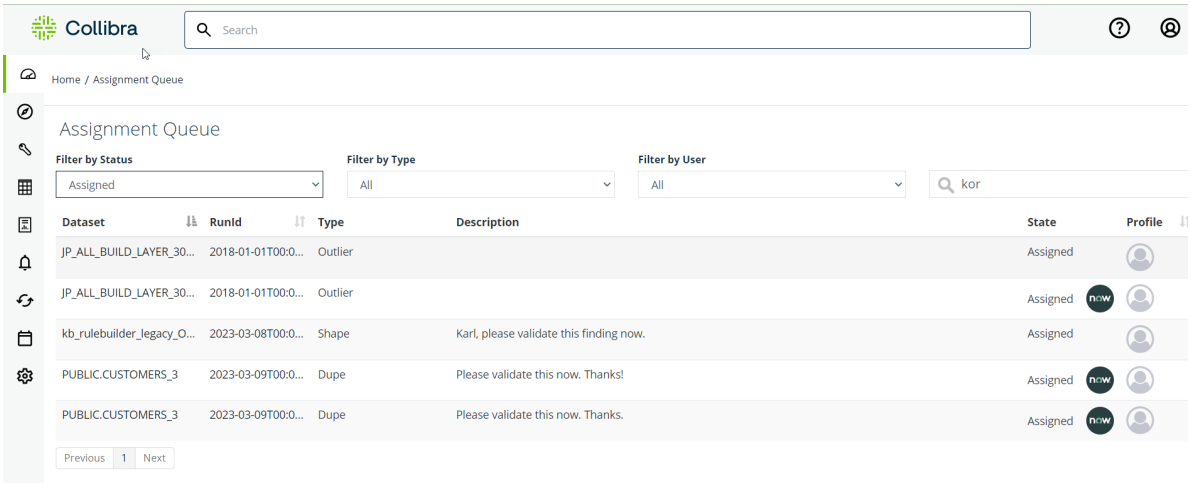
1. Login to the Collibra DQ instance and select the dashboard icon.
2. Select **Assignments**.
   The Assignment Queue displays.



3. You can filter the Assignment Queue by selecting from the following criteria:
   - **Status** - Filter by status by selecting one of the following options in the **Filter by Status** dropdown:
     - **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.
    - **Filter by Type** - Filter by type of behavior model assessment by selecting one of the following options in the **Filter by Type** dropdown:
        - **Outlier**
        - **Shape**
        - **Rule**
        - **Behavior**
        - **Dupe**
        - **Pattern**
        - **Record Changes**
        - **Schema Evolution**
        - **Mismatch Source**
    - **Filter by User** - Filter by user by selecting a user in the **Filter by User** dropdown.

You can also filter by entering your criteria in the Search field.



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.

FAQ

Can anyone assign an item?

Anyone with access to the dataset and TRAIN role (if enabled) can assign an item. Users who have been assigned an item can resolve without the TRAIN role.

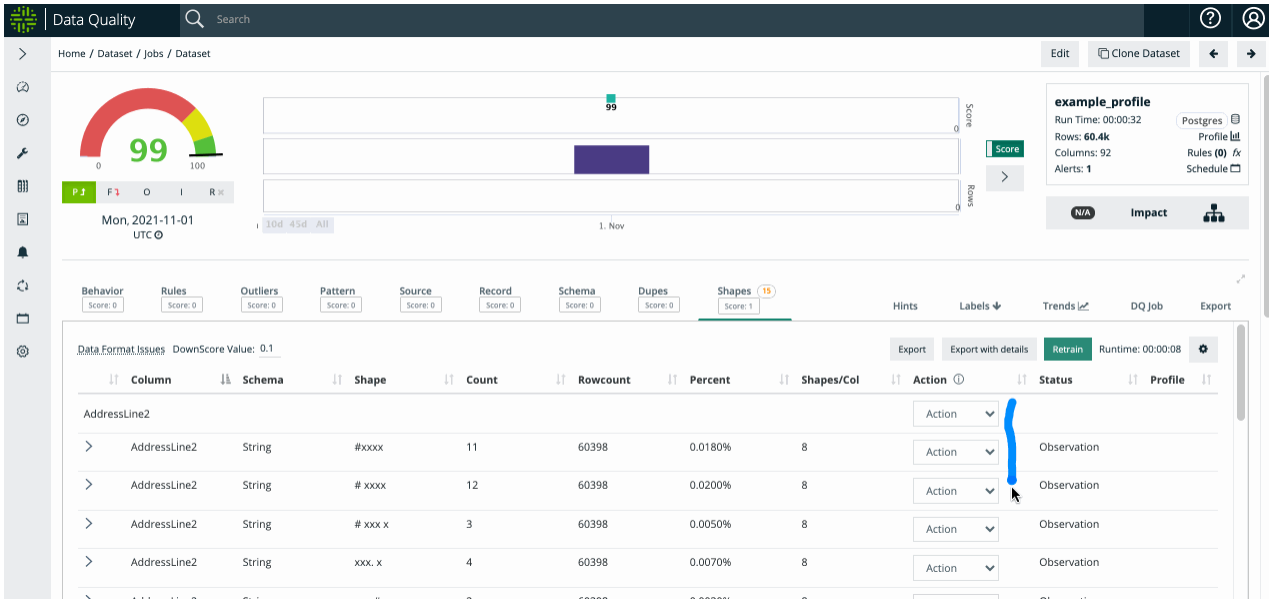
How do I add people to the assignment list

Once a person logs into the Collibra DQ Application they will get a profile and become eligible for assignment.

Labeling / Training

Item Labeling

Quickly click findings to trigger retraining.



Action labeling options

The following action labels instruct Collibra on how to handle a finding:

Action	Description
Validate	Instructs Collibra 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. **Note:** Validating a finding does not improve your score.
Invalidate	Instructs Collibra to ignore a finding and allow the value to pass. There are two invalidation options: Save and Save & Retrain. **Save:** Allows you to mark a finding as invalidated. **Save & Retrain:** Allows you to invalidate a finding and any previously saved invalidated findings (if any). **Note:** 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 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.

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
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.

Validating a finding

When you apply a validate label to a finding, you can assign it to another DQ user to review. This marks the finding for future runs and sends it to the internal Assignments Queue. You can also configure DQ to send assignments to an external queue, such as the ServiceNow tool. For more information, go to Manage Collibra DQ Assignments.

Invalidating a finding

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. The invalidate label allows you to do that. After you add a descriptive annotation of your action, you can then select either Save or Save & Retrain.

Save

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, select Save for all of the findings

you would like to bulk invalidate. On your last finding, select Save & Retrain. All previously saved invalidated findings are removed and DQ retrains your dataset.

Save & Retrain

When you Save & Retrain your dataset, any previously deducted points from a flagged finding are restored and reflected in your overall data quality score. If you do not have many findings to invalidate, you can Save & Retrain individually instead of in bulk.

Resolving a finding

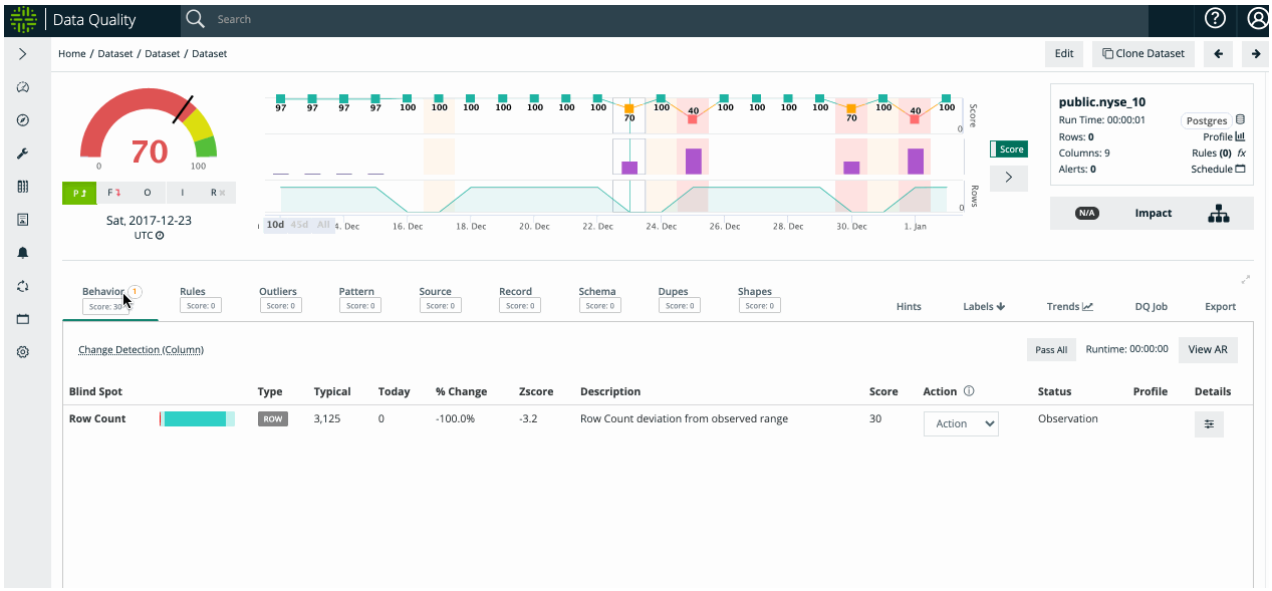
Some features, such as Behavior and Rules, only permit Validate and Resolve actions. When you cannot Validate a finding but you want to apply a label, select Resolve. The Resolve label prevents a finding from appearing in future runs of your dataset, and does not immediately affect your data quality score when applied.

Recalling labeled findings

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, unlabeled. From there you can again choose to Validate, Invalidate, or Resolve it.

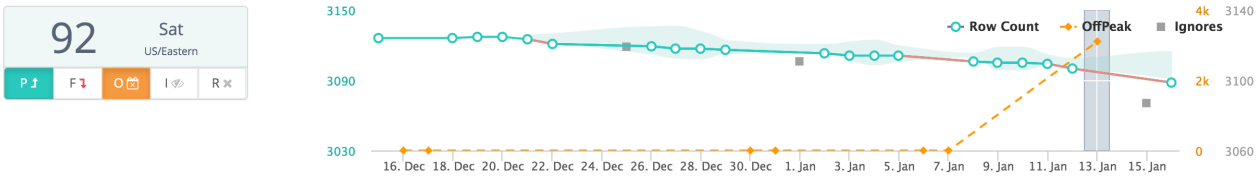
To closely analyze when a finding has received a label, who has applied it, and more, see also the Dataset Audit Trail.

Peak vs Off Peak



But my Weekend runs are not the Same

A common scenario that can fool behavioral analytics and machine learning is when you have a few different but normal patterns. Collibra has a rich labeling system that allows a user to fork the training model to learn these cycles individually without confusing the model.

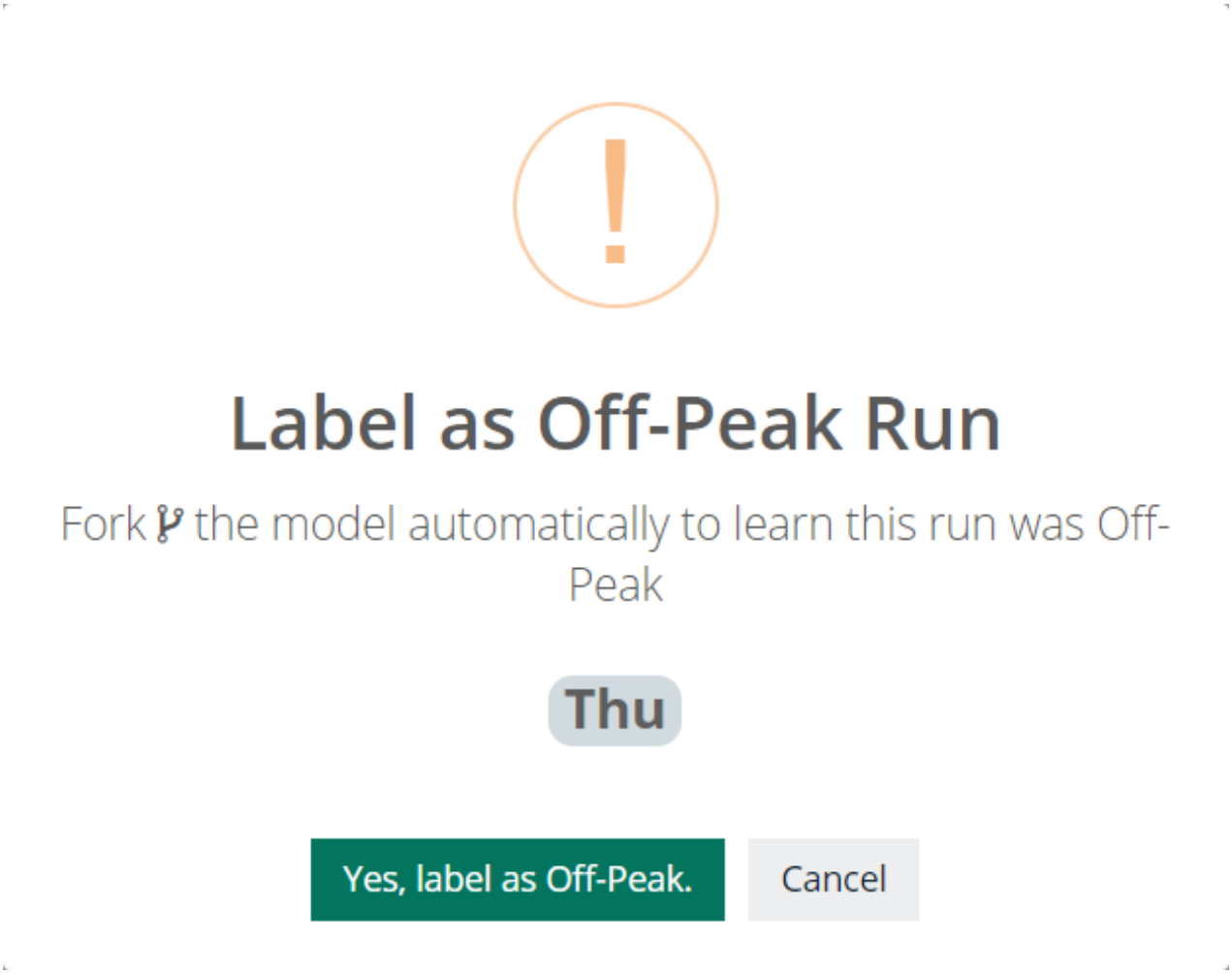


> Note The off-peak line chart displays values less than 100 when a score is failing, because failing scores override days labeled as off-peak.

Click the Green Button

By clicking the green button, you can label the day of the week as peak vs off peak. You can also chose your time zone - this will help determine the day of the week accurately.

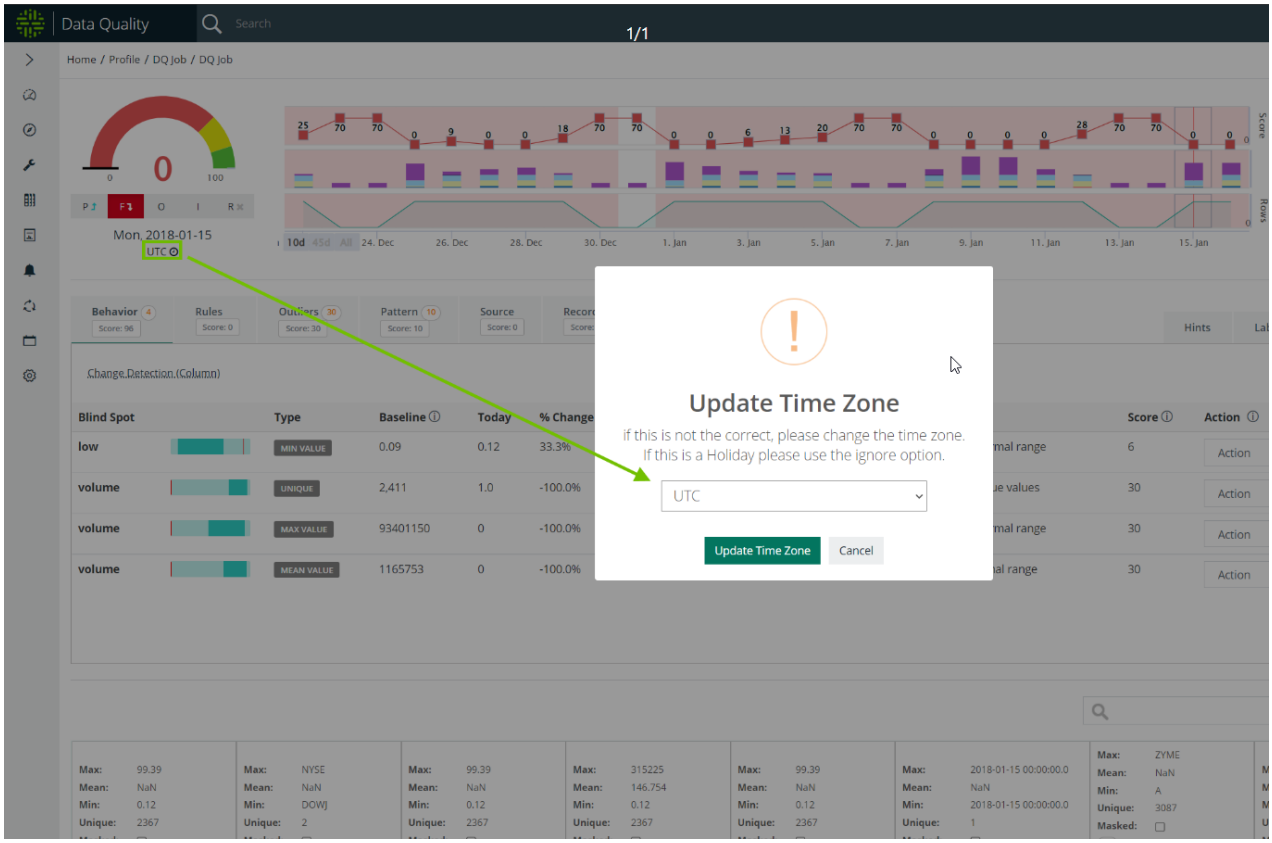
You only need to click the peak scheduler once and the model will learn and forecast this understanding for every run in the future. This feature commonly prompts for a re-train.



Time Zones

Updating time zones

By default, Collibra Data Quality's time zone associated with the RunID is located in Coordinated Universal Time (UTC). To update the server time zone, select the Update Time Zone link from the findings page. An Update Time Zone dialog displays with the option to select your time zone from the dropdown menu. Click the Update Time Zone button to confirm your selection.



Note Since the server time zone can differ from the configurable RunID time zone on the findings page, datasets in List View may have different dates than the date listed on the findings page of the same dataset. For example, a dataset with a RunID in the default UTC time zone may appear as 2022-01-15 00:00:00 on the findings page, but because the server is located in US/Central time, the date appears as 2022-01-14 19:00:00.

Collibra DQ DIC Integration

Configuring integrations

This section provides information about how to integrate Collibra Data Quality & Observability into Collibra Data Intelligence Cloud.

About Collibra DQ integrations

Collibra DQ Integrations allow DQ admin users to synchronize metadata from Collibra Data Quality & Observability into Collibra Data Intelligence Cloud. You can use the Integration Configuration wizard for a guided mapping of objects between Collibra DQ and Collibra Data Intelligence Cloud in only a few clicks. When you integrate Collibra DQ metadata into Collibra Data Intelligence Cloud, 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:

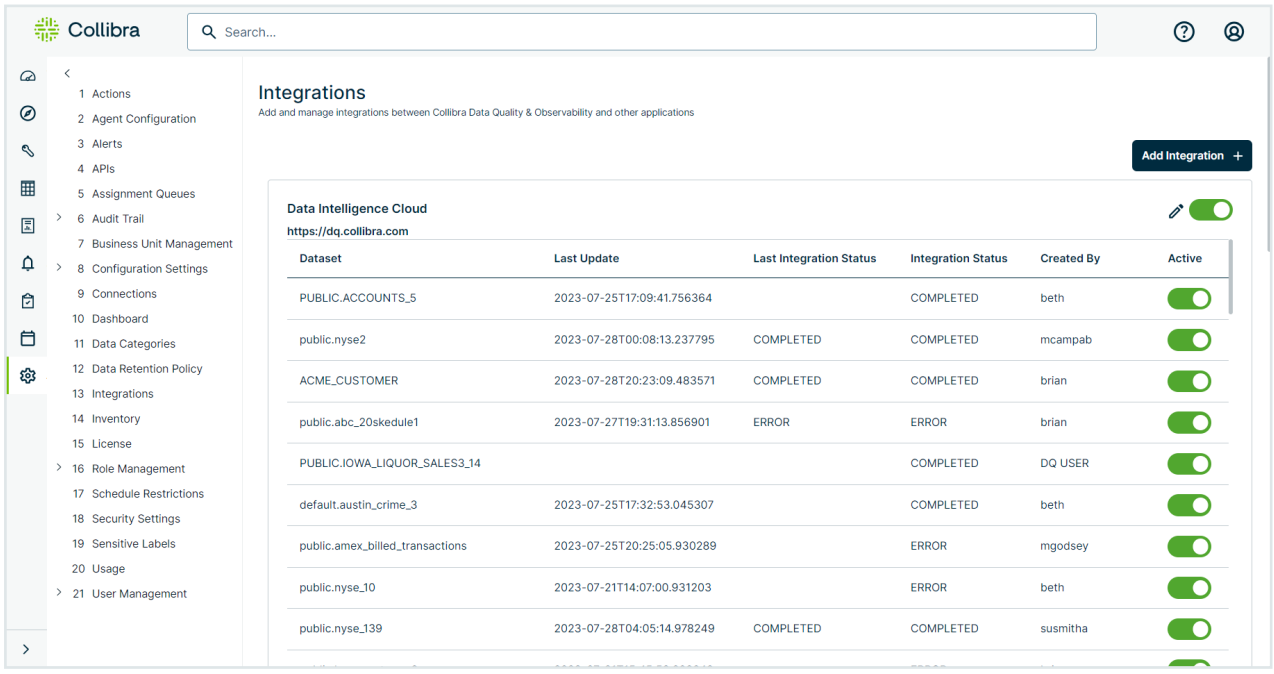
- Broader data sharing between platforms to include not only user defined rules but also adaptive rules, shapes, schema, source, record, pattern, outlier, and duplicate findings along with their scores.
- Automated synchronization between integrated platforms based on job execution and job scheduling.
- Simplified configuration and maintenance of the integration.

Prerequisites

- Your Collibra DQ version is 2023.07 or later and Collibra Data Intelligence Cloud version is 2023.05 or later.
- You have an Admin role in Collibra Data Quality & Observability to use the configuration wizard.
- You have an Admin or Dataset Manager role in Collibra Data Quality & Observability to enable or disable individual datasets for integration.
- Your Collibra Data Intelligence Cloud 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 Data Intelligence Cloud ▸ Settings ▸ Operating model ▸ Data quality rules.
- React MUI is turned on.

Integrations

From the Integrations page in the Admin Console, you can add and manage integrations between Collibra Data Quality & Observability and Collibra Data Intelligence Cloud. You can conveniently switch integrations between Collibra DQ and Collibra Data Intelligence Cloud instances on and off, and find an overview of the status of all datasets with an attempted or completed integration.



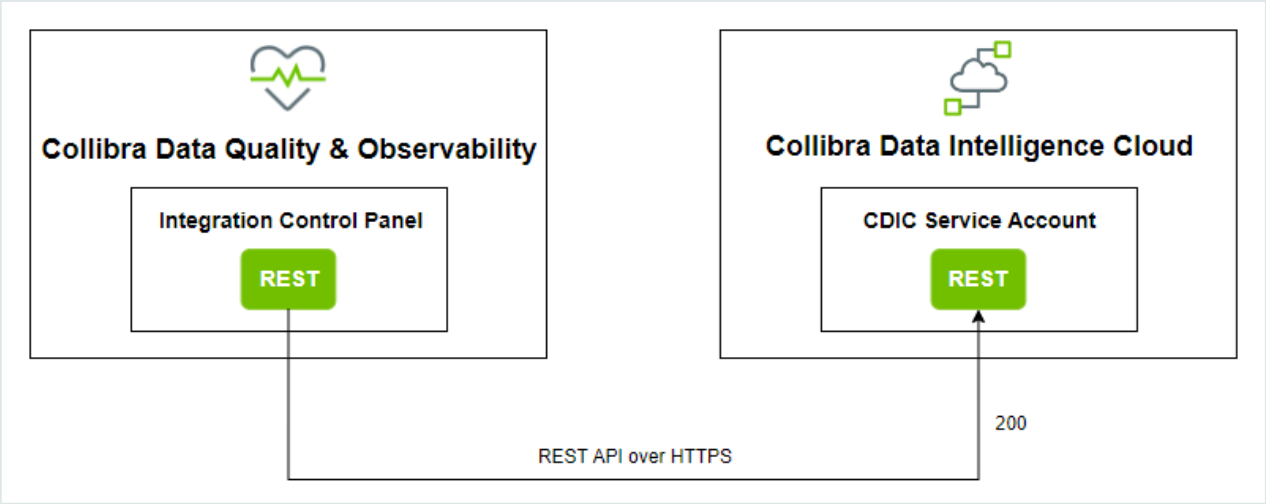
The following table describes each column of the dataset integration status table on the Integrations page.

Column	Description
Dataset	The name of the Collibra DQ dataset whose metadata integrated into Collibra Data Intelligence Cloud.
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.
Last Integration Status	The status of the last attempt to send metadata to Collibra Data Intelligence Cloud.
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 dataset can actively integrate into Collibra Data Intelligence Cloud.

Integration architecture

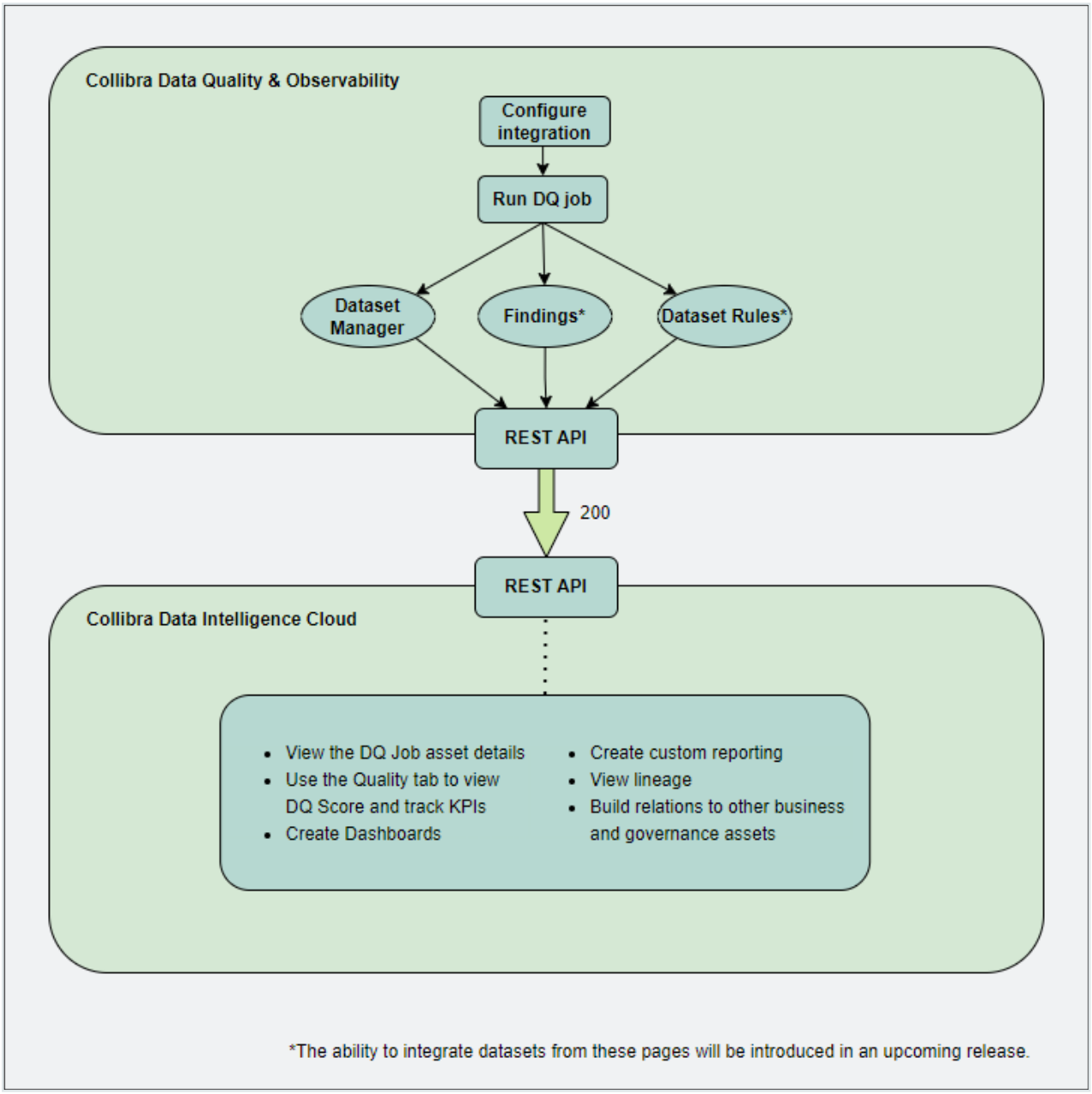
This section shows the architecture of an integration of Collibra Data Quality & Observability into Collibra Data Intelligence Cloud.

Integration Control Panel

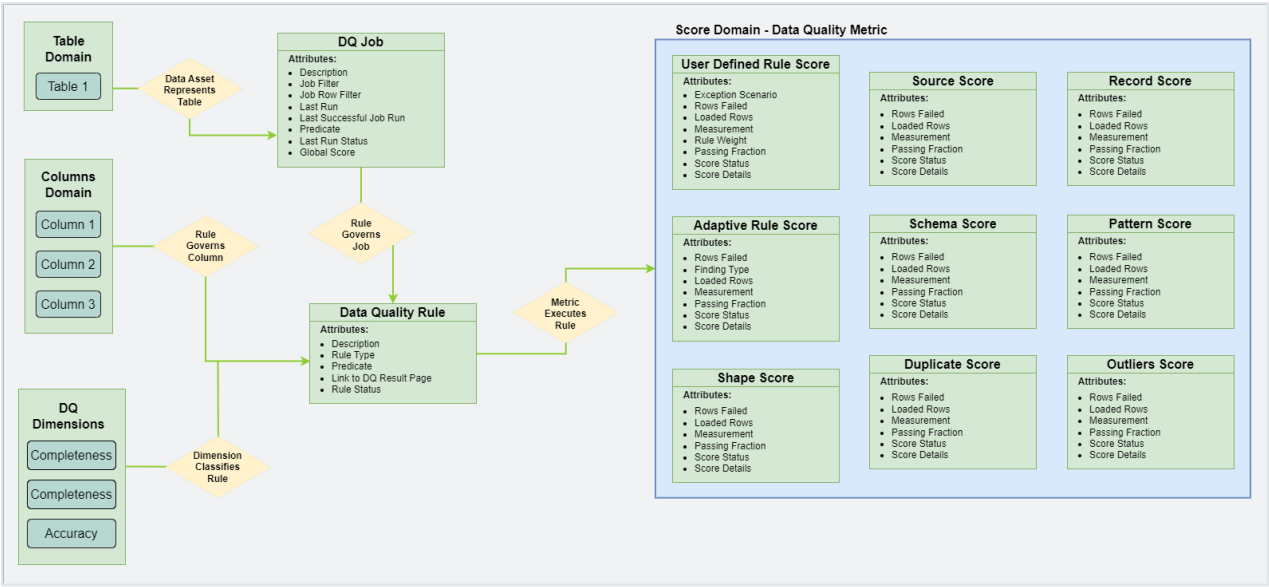


This diagram shows the flow of events upon the initial configuration of a new integration between Collibra Data Quality & Observability and Collibra Data Intelligence Cloud. This uses the Integration API to communicate between Collibra DQ and Collibra Data Intelligence Cloud.

Integrating a dataset into Collibra Data Intelligence Cloud



Operating model



Configuring a new integration

This section shows you how to use the Integration configuration wizard to set up a new integration of Collibra Data Quality & Observability metadata from profile findings, rules, and alerts into Collibra Data Intelligence Cloud.

> 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 Collibra DQ version is 2023.07 or later and Collibra Data Intelligence Cloud version is 2023.05 or later.

- You have an Admin role in Collibra Data Quality & Observability to use the configuration wizard.
- You have an Admin or Dataset Manager role in Collibra Data Quality & Observability to enable or disable individual datasets for integration.
- Your Collibra Data Intelligence Cloud 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 Data Intelligence Cloud ▸ Settings ▸ Operating model ▸ Data quality rules.
- React MUI is turned on.

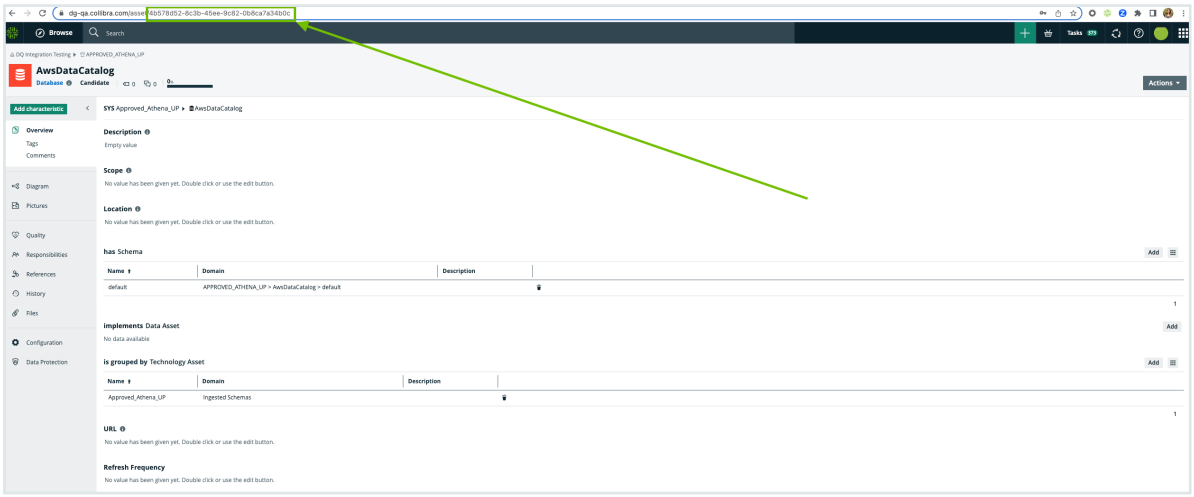
Steps

1. Hover your cursor over the ⚙ icon and click **Integrations**.
   » The Integrations page opens.
2. In the upper right corner, click **Add Integration**.
   » The Add New Integration wizard appears.
3. On the **Credentials** tab, enter the required information.

Option	Description	Required
Service Type	The application with which to integrate Collibra Data Quality & Observability metadata. This value is always **Data Intelligence Cloud**.	✓ Yes
Service URL	The URL of the Collibra Data Intelligence Cloud 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 Data Intelligence Cloud. This value is always **basic**.	✓ Yes
Service Account	The username of your Collibra Data Intelligence Cloud environment. For example, "integration.user" or "Admin".	✓ Yes
Password	The password of your Collibra Data Intelligence Cloud environment.	✓ Yes

4. Click **Next**.
   » The Connections tab opens.

5. On the **Connections** tab, map your Collibra DQ connections to the appropriate database, schema, table, and column assets in Collibra Data Intelligence Cloud.
    a. From the **Select Connection** dropdown menu, select a Collibra DQ database to integrate into Collibra Data Intelligence Cloud.
    b. In yourCollibra Data Intelligence Cloud 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.



    c. From the Connections tab in Collibra DQ, paste the copied UUID from Collibra Data Intelligence Cloud, or enter the UUID manually.
    d. Click **Run** to create an automatic mapping between the schemas within your selected Collibra DQ database and the schemas within Collibra Data Intelligence Cloud.

> Important
> Only schemas ingested in Collibra Data Intelligence Cloud return when you click Run.

    e. 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 dropdown menu and select a different schema ( ) to update the schema mapping.

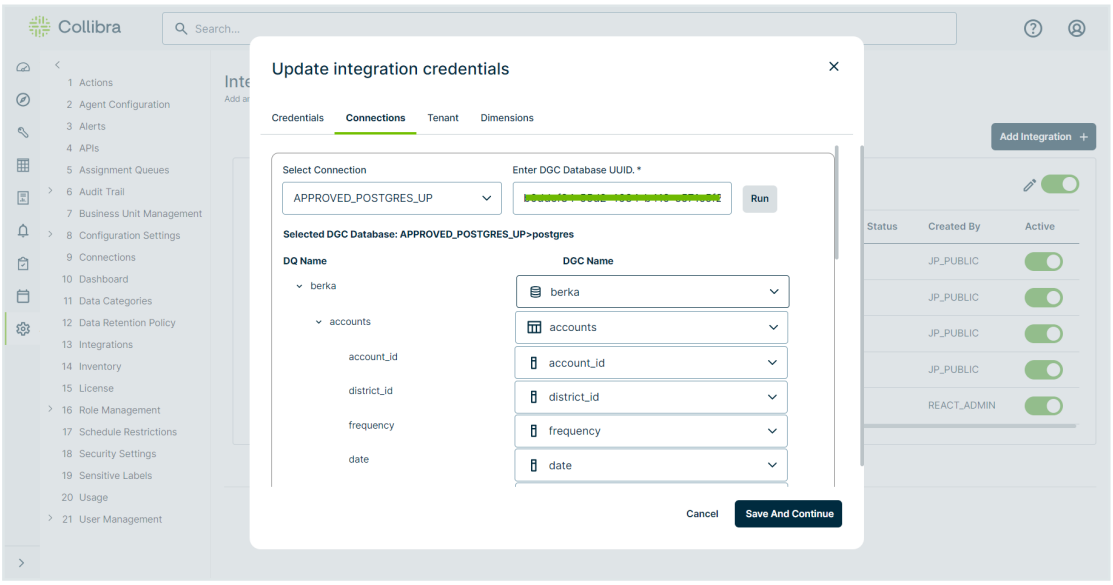
> Note When a name match cannot be made, a "Not Configured" message displays.

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 dropdown menu 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.

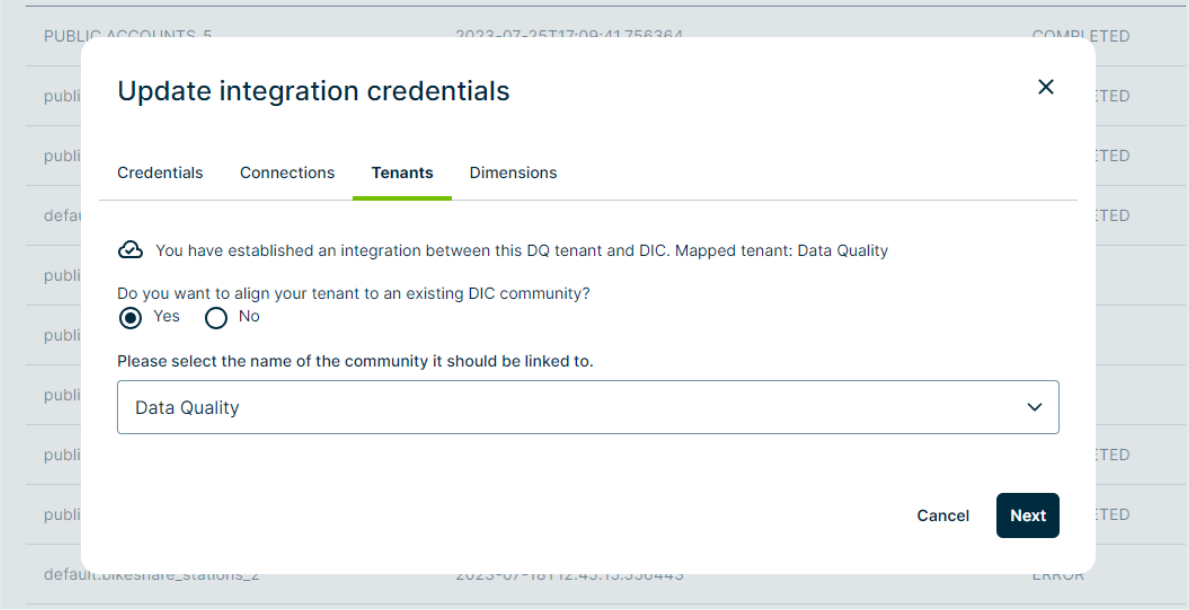
v. In the **DGC Name** column, click the dropdown menu and select a different column to update the column mapping, if necessary.



f. After you finish reviewing the mappings of your connection, click **Save and Continue**.

g. Optionally click **Add New** to map another connection and then repeat this process.

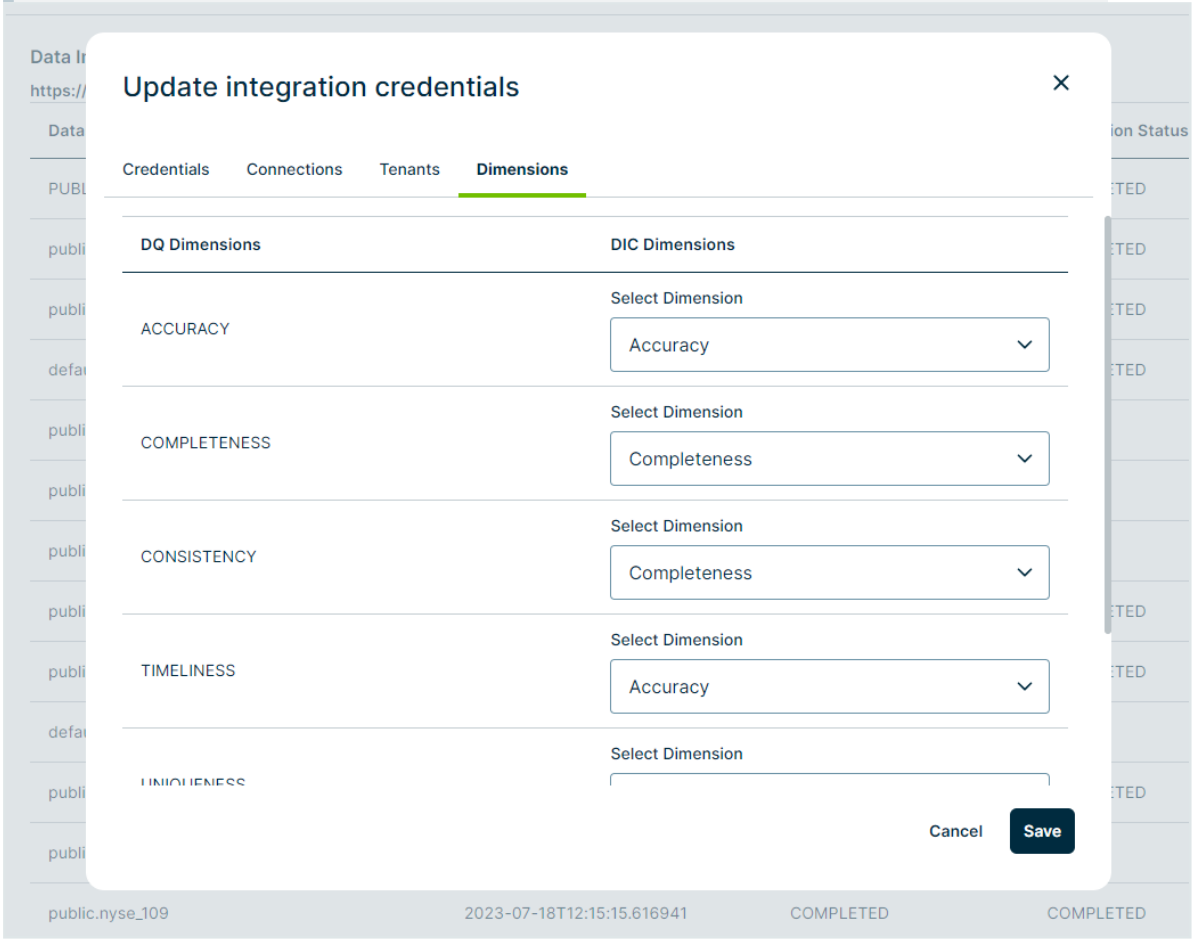
6. Click **Next**.
   » The Tenants tab opens.
7. On the **Tenants** tab, enter the required information.
   a. Optionally select whether to align your Collibra DQ tenant with an existing Collibra Data Intelligence Cloud community.
      i. If Yes,
         ◦ Select a Collibra Data Intelligence Cloud community from the dropdown menu with which to align your tenant.
      ii. If No,
         ◦ 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 Data Intelligence Cloud. If a dataset is not assigned to a Business Unit, it is added to a community called *Unassigned DQ Job* by default.

8. Click **Next**.
   » The Dimensions tab opens.
9. On the **Dimensions** tab, enter the required information.
   a. Select a dimension from the dropdown menu of the DIC Dimension column that corresponds with the DQ Dimension you want to identify as a match.

> Note All DQ Dimensions, from your Collibra Data Intelligence Cloud environment are available to map to DQ Dimensions in Collibra DQ, including custom dimensions. However, DQ Dimensions from Collibra DQ are not customizable at this time.



10. Click **Save**.

What's next?

Enable a dataset for integration and start ingesting metadata from Collibra DQ into Collibra Data Intelligence Cloud.

Enabling dataset integrations

This section shows you how to integrate Collibra Data Quality & Observability metadata from datasets into Collibra Data Intelligence Cloud.

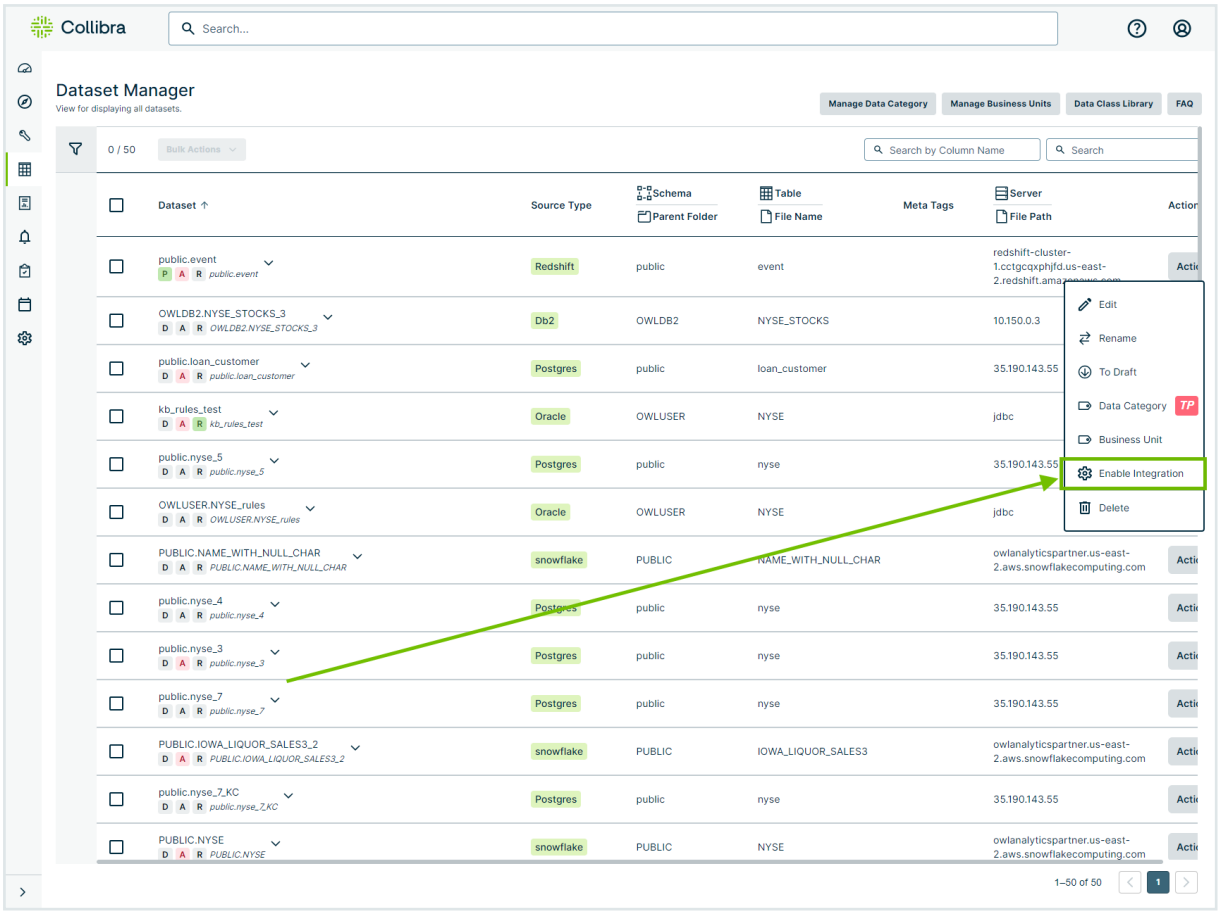
Prerequisites.

- You have set up an integration between Collibra DQ and Collibra Data Intelligence Cloud.
- 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 Data Intelligence Cloud ingests the assets that pertain to your dataset in Collibra DQ.

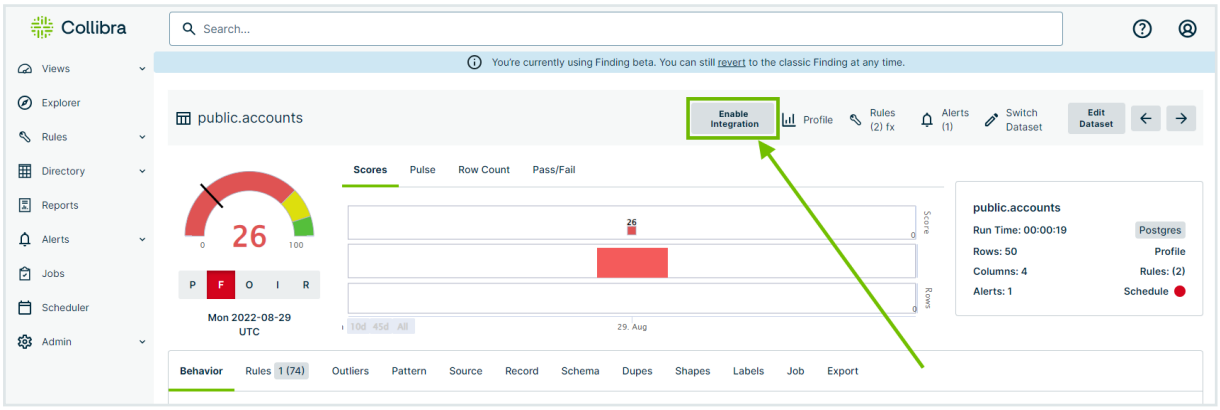


Important To enable an integration from the Findings page, React must be on.

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 Data Intelligence Cloud ingests the assets that pertain to your dataset in Collibra DQ.



What's next?

You may find that you do not want Collibra Data Intelligence Cloud 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

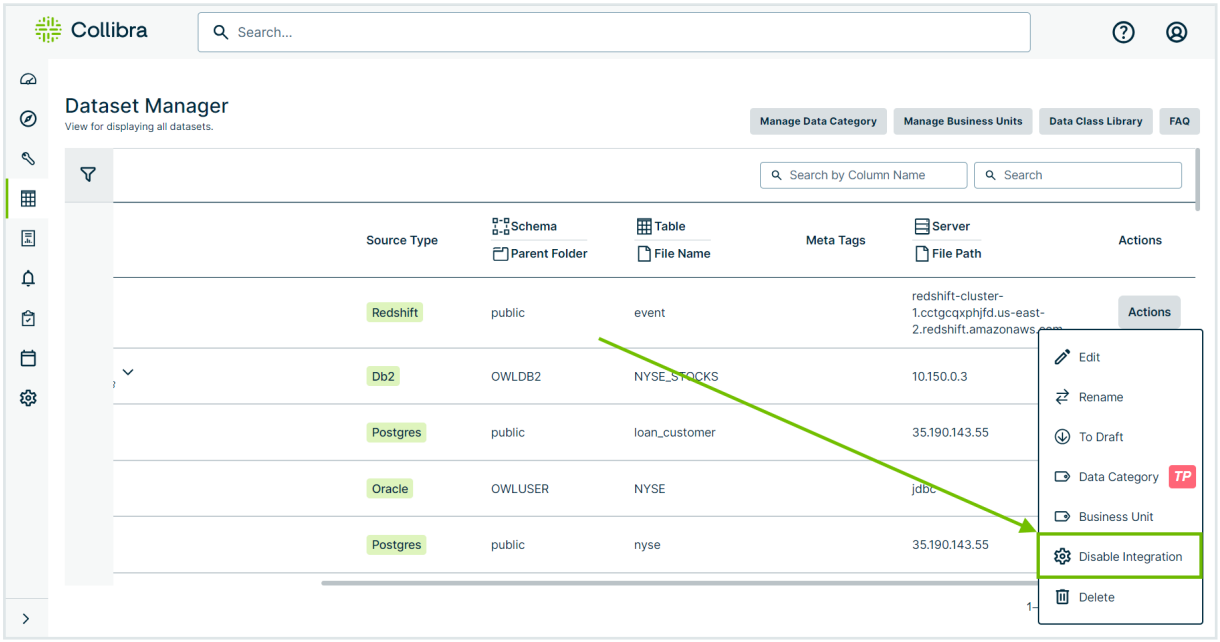
This section shows you how to disable automatic integrations of metadata from datasets in Collibra DQ into Collibra Data Intelligence Cloud.

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 Data Intelligence Cloud 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 Data Intelligence Cloud.



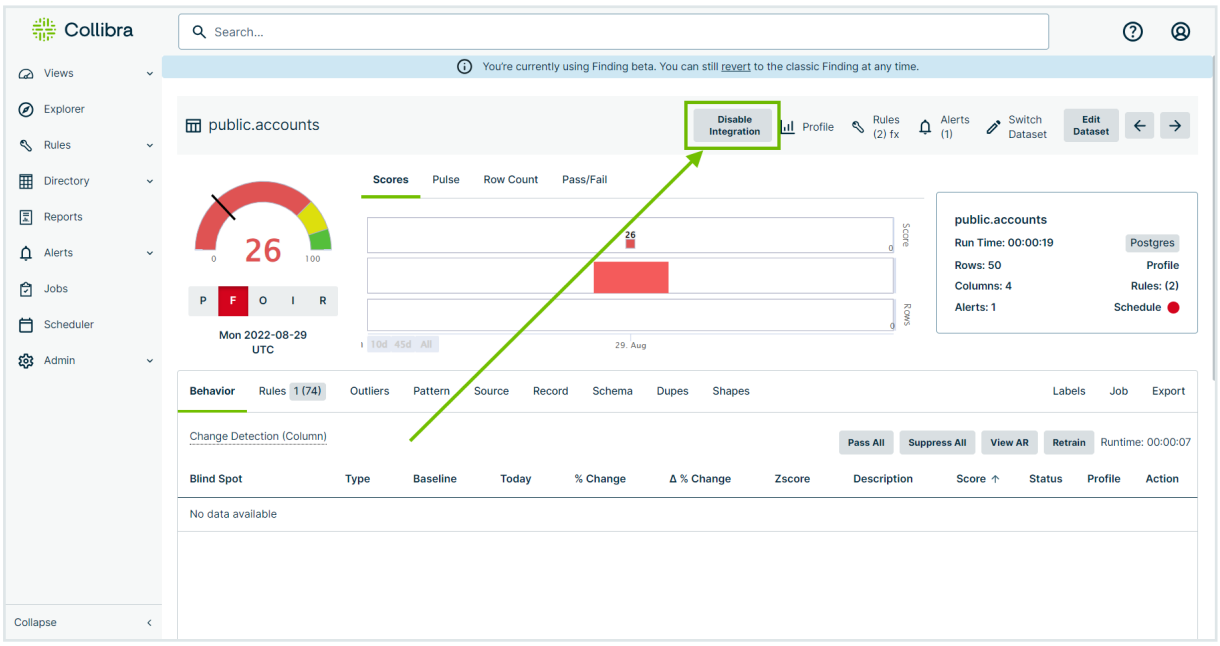
> Important To enable an integration from the Findings page, React must be on.

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 Data Intelligence Cloud 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 Data Intelligence Cloud.



Modifying an existing integration

This section shows you how to use the Integration Configuration Wizard to modify an existing integration of Collibra Data Quality & Observability with Collibra Data Intelligence

Cloud.

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 Collibra Data Quality & Observability Integration.

Question	Answer
What roles are required in Collibra DQ and Collibra Data Intelligence Cloud?	In **Collibra DQ**: • `ROLE_ADMIN` is required to complete the configuration wizard steps, which includes adding Collibra Data Intelligence Cloud 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 Data Intelligence Cloud**: • The credentials shared in the configuration wizard in Collibra DQ must have `Sysadmin` and `DataSteward` permissions in Collibra Data Intelligence Cloud. • Any user can view DQ assets in Collibra Data Intelligence Cloud.

Question	Answer

What assets are created in Collibra Data Intelligence Cloud 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 Data Intelligence Cloud be reflected in Collibra DQ?	No. In its current implementation, metadata only flows from Collibra DQ to Collibra Data Intelligence Cloud. Changes you make to DQ rules, jobs, or metric assets in Collibra Data Intelligence Cloud will not be reflected in Collibra DQ.
How are data quality scores calculated in Collibra Data Intelligence Cloud? 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 Data Intelligence Cloud. 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 Data Intelligence Cloud? Is this configurable?	Collibra DQ shares all DQ dimensions with Collibra Data Intelligence Cloud, 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 Data Intelligence Cloud. 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 Data Intelligence Cloud does not automatically delete. However, DQ rules deleted in Collibra DQ update to "Suppressed" in Collibra Data Intelligence Cloud 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 Data Intelligence Cloud, you need to perform a manual clean-up of the assets in Collibra Data Intelligence Cloud 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 data-set but the Quality tab/dashboard in Collibra Data Intel-ligence Cloud 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 Data Intelligence Cloud, 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.

DQ Connector

> Important
> This capability has been archived. For the latest option to integrate your Collibra DQ metadata with Collibra Data Intelligence Cloud, check out our enhanced integration documentation.

DQ Workflows

Benefits

The DQ Workflows package listed on Collibra Marketplace allows you to 1) create and manage Data Quality Issues, 2) receive Notifications on Rule Metrics, and 3) request Rule Creation and Modification within **Collibra Data Intelligence Cloud**. Data stewards will be able to organize and prioritize all requests within DIC before they take any action within **Collibra Data Quality**.

Once deployed, the workflows will facilitate quicker data issue remediation by involving business analysts and other personas who can now participate in your data quality workstreams.

Please note: DQ Workflows are listed on Collibra Marketplace and are templates to get customers started. Collibra-provided Marketplace listings are not subject to the same SLA obligations (https://marketplace.collibra.com/marketplace-terms/) In addition, they can only be leveraged within Collibra Data Intelligence Cloud. In the future, we will work towards releasing bi-directional workflows.

Step 0: Prerequisites

Resource	Notes
Collibra Edge Site	DQ Connector is a capability of Edge
Collibra Data Intelligence Cloud	2021.07 Release (or newer)

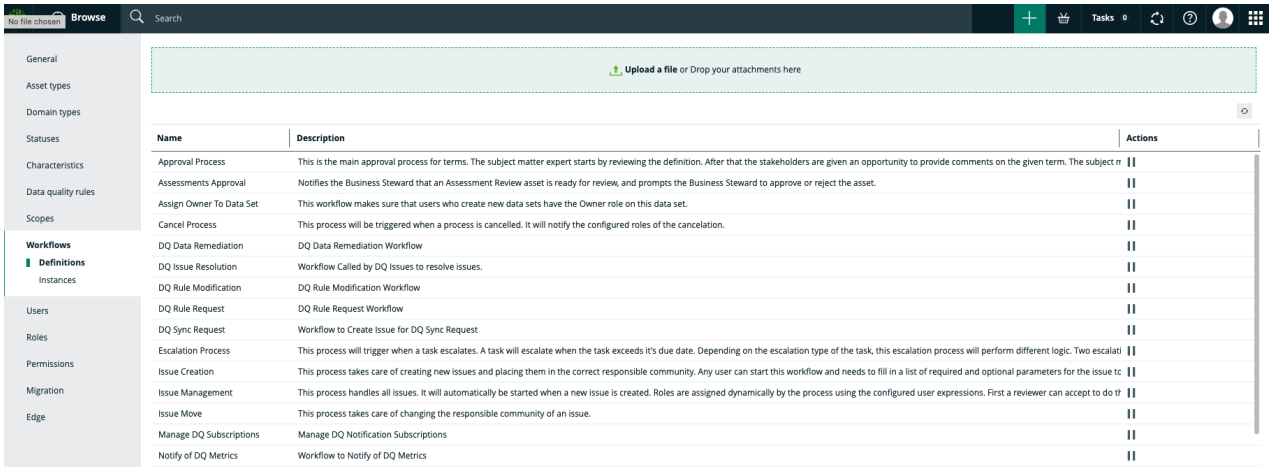
Resource	Notes
Collibra Data Quality	2.15 (or newer)
Collibra DQ Connector	Synchronized Rules from Data Quality to Catalog

> Note After gathering all the prerequisites, you can now proceed to the next step.

Step 1: Download, Deploy and Start DQ Workflows

1A. Download Package from Collibra Marketplace and Unzip Files

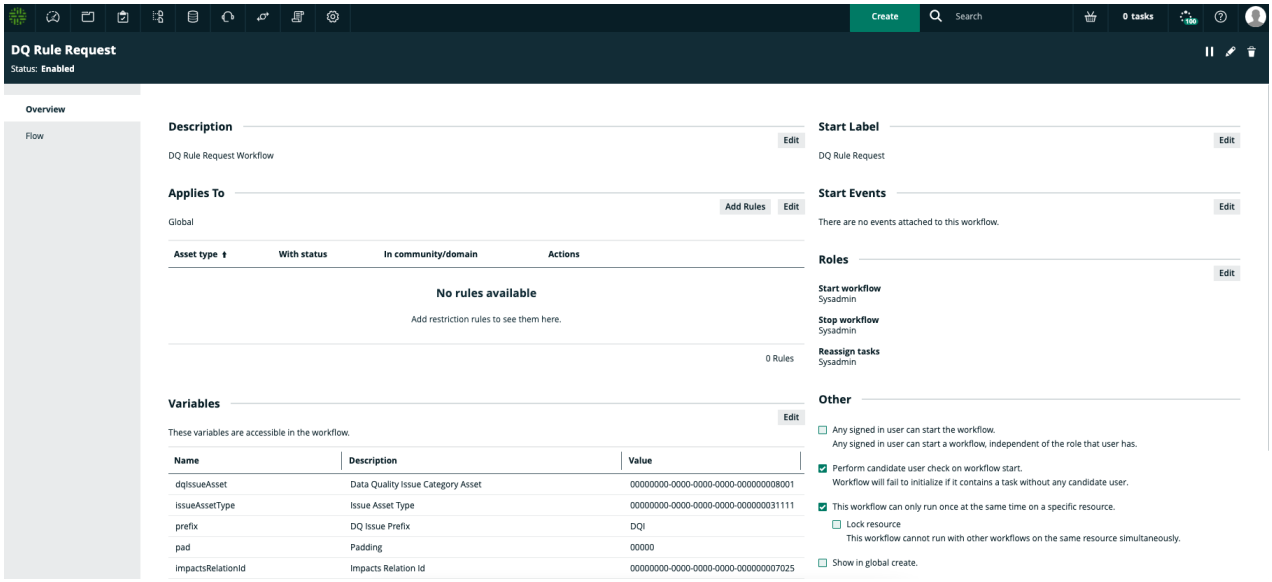
**1B. Deploy Workflows **

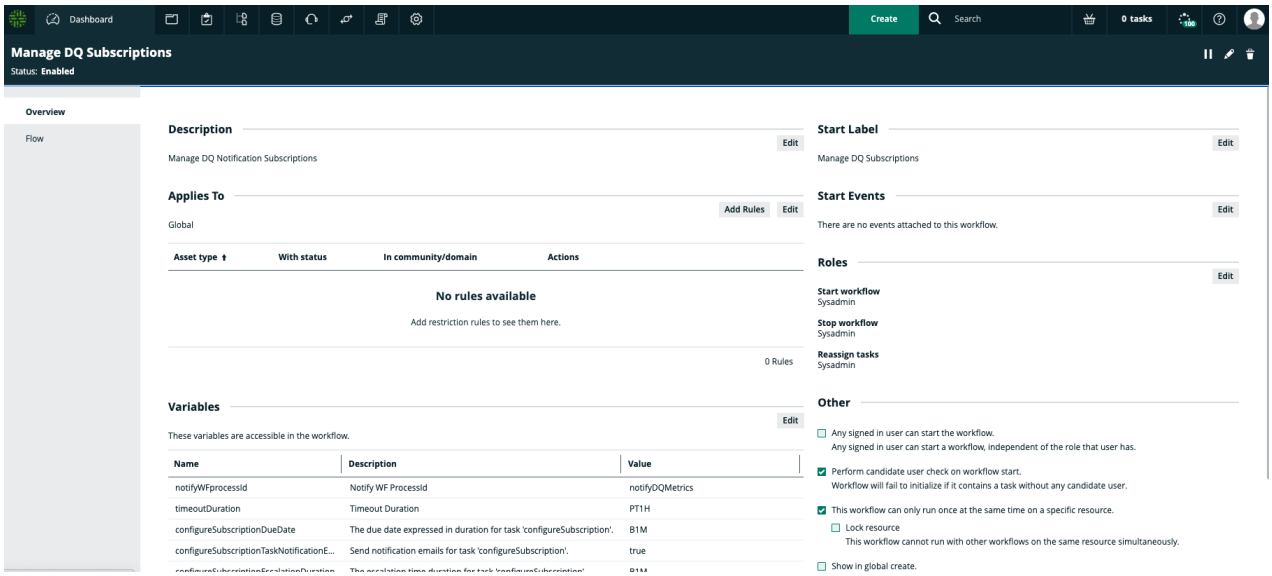


1C. Adjust Workflow Settings (One-Time Setup)

Workflow Configuration Setting	DQ Rule / DQ Sync Request	DQ Rule Modification	DQ Data Remediation	DQ Issue Resolution	Manage DQ Subscriptions	Notify of DQ Metrics
Applies To	Asset	Asset	Asset	Asset	Global	Global
Applies To Asset Type	Column, Table	Column, Table, Data Quality Rule	Column, Table	Issue		

Workflow Configuration Setting	DQ Rule / DQ Sync Request	DQ Rule Modification	DQ Data Remediation	DQ Issue Resolution	Manage DQ Subscriptions	Notify of DQ Metrics
Other: Any Signed In User Can Start Workflow	Y	Y			Y	Y
Other: Perform Candidate User Check on Workflow Start	Y	Y	Y		Y	Y
Other: This Workflow Can Only Run Once At Same Time on Specific Resource					Y	Y
Other: Show In Global Create					Y	Y
Roles: Start Workflow					Sysadmin	Sysadmin
Roles: Stop Workflow					Sysadmin	Sysadmin
Roles: Reassign Tasks					Sysadmin	Sysadmin

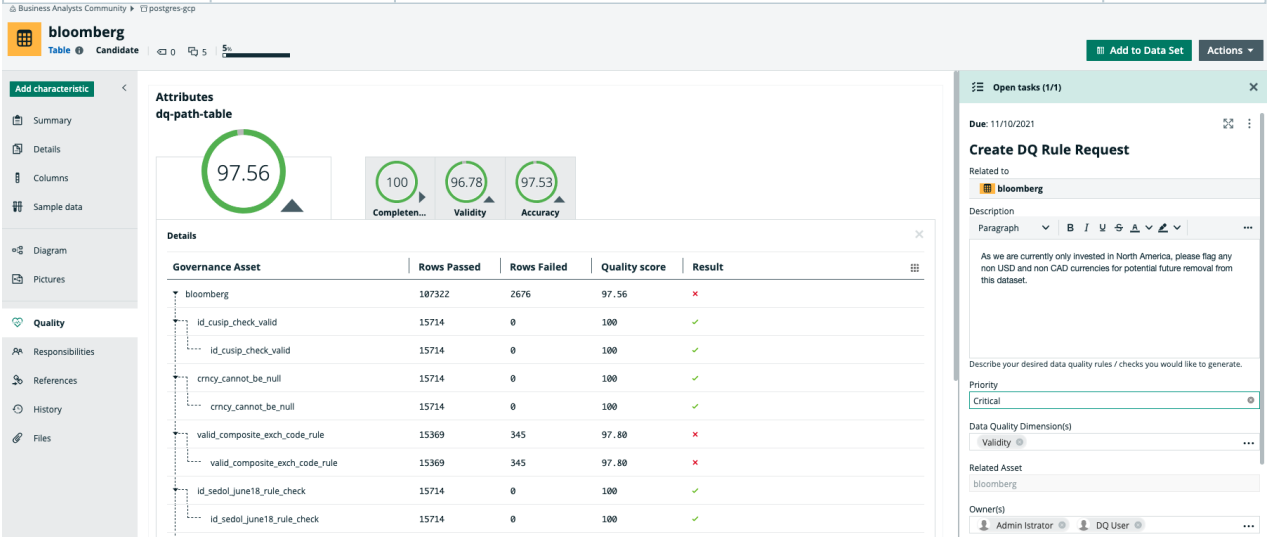

Create
Search
0 tasks
DQ Rule Request
Status: Enabled
Overview
Flow
Description
Edit
DQ Rule Request Workflow
Applies To
Add Rules
Edit
Global
Asset type
With status
In community/domain
Actions
No rules available
Add restriction rules to see them here.
0 Rules
Variables
Edit
These variables are accessible in the workflow.
Name
Description
Value
dqIssueAsset
Data Quality Issue Category Asset
00000000-0000-0000-0000-000000008001
issueAssetType
Issue Asset Type
00000000-0000-0000-0000-000000031111
prefix
DQ Issue Prefix
DQI
pad
Padding
00000
impactsRelationId
Impacts Relation Id
00000000-0000-0000-0000-000000007025
Start Label
Edit
DQ Rule Request
Start Events
Edit
There are no events attached to this workflow.
Roles
Edit
Start workflow
Sysadmin
Stop workflow
Sysadmin
Reassign tasks
Sysadmin
Other
Any signed in user can start the workflow.
Any signed in user can start a workflow, independent of the role that user has.
Perform candidate user check on workflow start.
Workflow will fail to initialize if it contains a task without any candidate user.
This workflow can only run once at the same time on a specific resource.
Lock resource
This workflow cannot run with other workflows on the same resource simultaneously.
Show in global create.

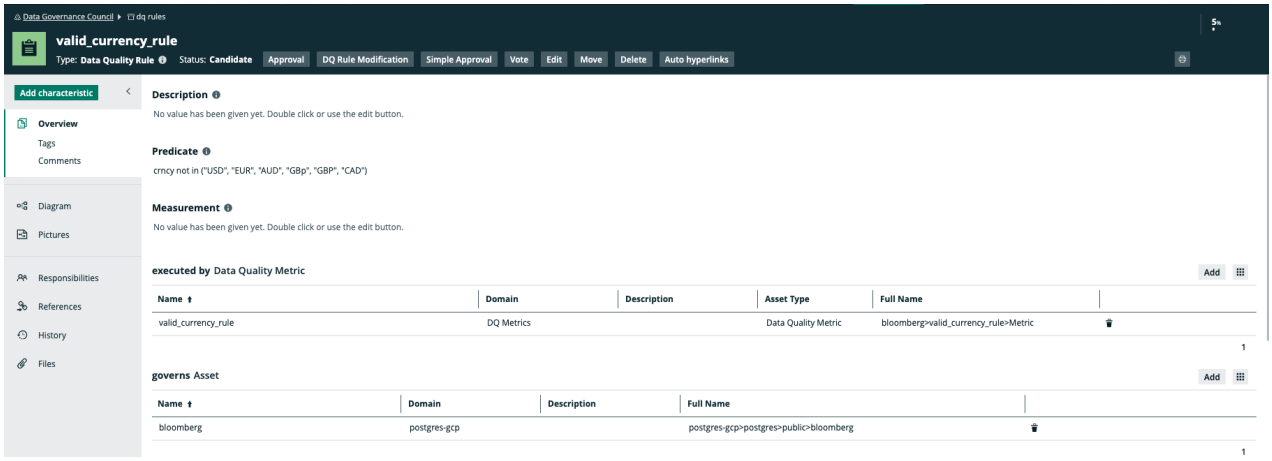

Dashboard
Create
Search
0 tasks
Manage DQ Subscriptions
Status: Enabled
Overview
Flow
Description
Edit
Manage DQ Notification Subscriptions
Applies To
Add Rules
Edit
Global
Asset type
With status
In community/domain
Actions
No rules available
Add restriction rules to see them here.
0 Rules
Variables
Edit
These variables are accessible in the workflow.
Name
Description
Value
notifyWFprocessId
Notify WF ProcessId
notifyDQMetrics
timeoutDuration
Timeout Duration
PT1H
configureSubscriptionDueDate
The due date expressed in duration for task 'configureSubscription'.
B1M
configureSubscriptionTaskNotificationE...
Send notification emails for task 'configureSubscription'.
true
Start Label
Edit
Manage DQ Subscriptions
Start Events
Edit
There are no events attached to this workflow.
Roles
Edit
Start workflow
Sysadmin
Stop workflow
Sysadmin
Reassign tasks
Sysadmin
Other
Any signed in user can start the workflow.
Any signed in user can start a workflow, independent of the role that user has.
Perform candidate user check on workflow start.
Workflow will fail to initialize if it contains a task without any candidate user.
This workflow can only run once at the same time on a specific resource.
Lock resource
This workflow cannot run with other workflows on the same resource simultaneously.
Show in global create.

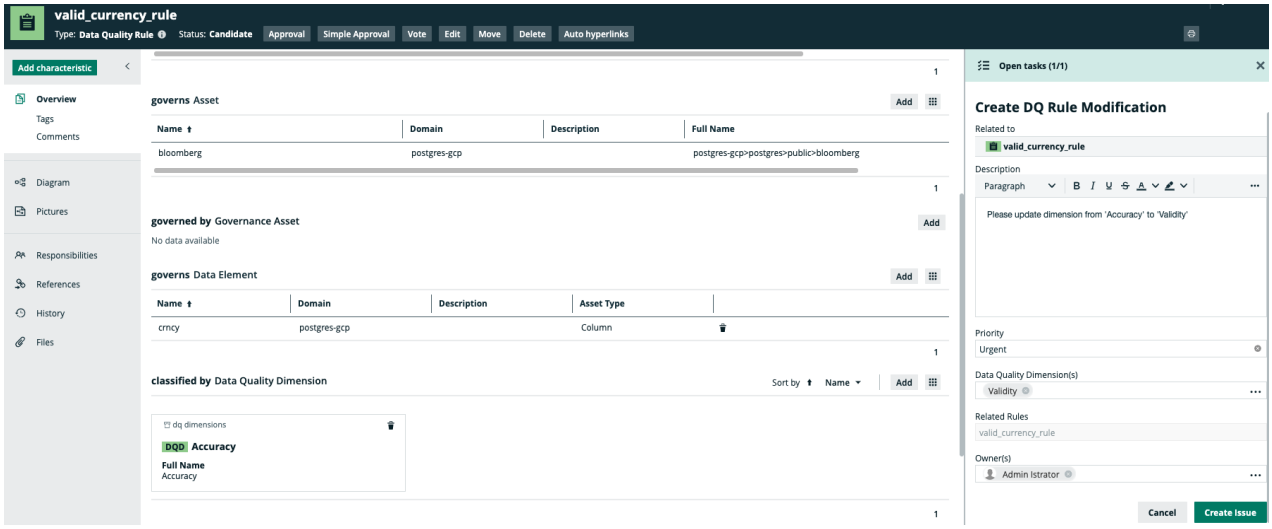
Step 2: Create Data Quality Requests / Issues

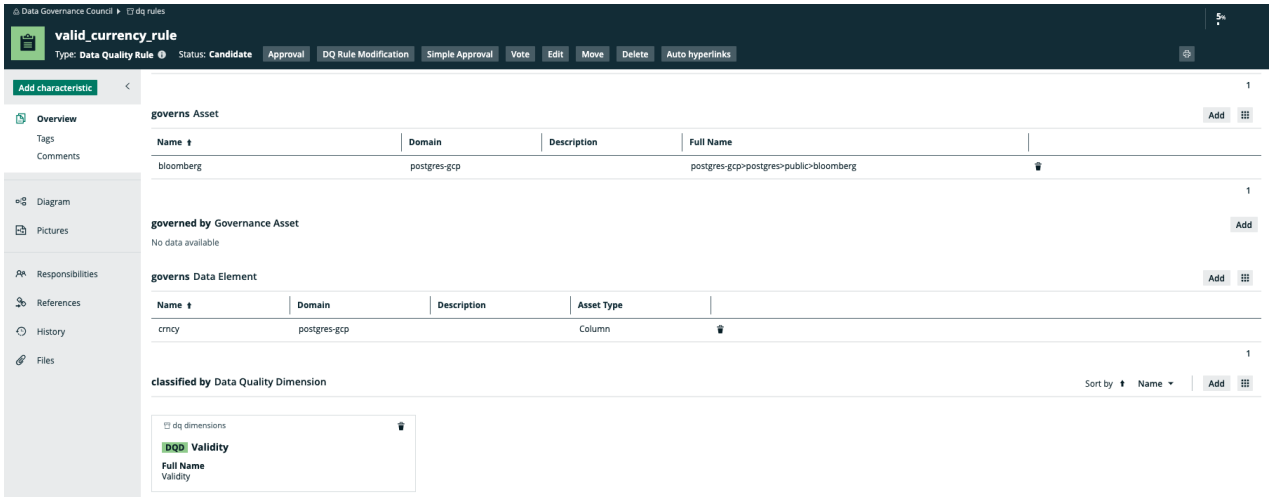
2A. Create Data Quality Issues

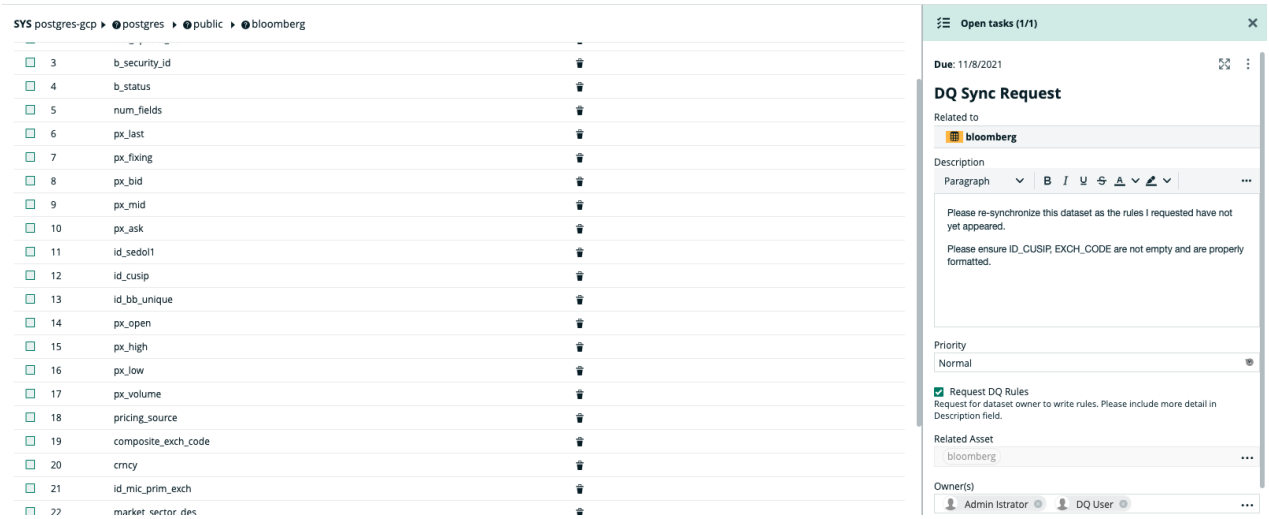
Workflow	Main Requestor Persona	Description	Steward Taking Action
DQ Data Remediation	Data Steward Business Analyst	Tracking / management for confirmed data issues which may require underlying data remediation	Data Lake Admin, ETL Engineer
DQ Rule Request	Business Analyst	Proposing data quality rules in plain language e.g. "flag any German phone numbers in this dataset" or "identify customers with churn risk based on engagement time with our platform"	Data Steward
DQ Rule Modification	Business Analyst	Proposing adjustments to existing rules e.g. values, dimensions, passing thresholds	Data Steward
DQ Sync Request	Business Analyst	Request for synchronization of the DQ Connector to synchronize and/or onboarding a new dataset with pre-populated rules	Data Steward




Data Governance Council
dq rules
valid_currency_rule
Type: Data Quality Rule
Status: Candidate
Approval
DQ Rule Modification
Simple Approval
Vote
Edit
Move
Delete
Auto hyperlinks
Add characteristic
Overview
Tags
Comments
Diagram
Pictures
Responsibilities
References
History
Files
Description
No value has been given yet. Double click or use the edit button.
Predicate
crncy not in ("USD", "EUR", "AUD", "GBp", "GBP", "CAD")
Measurement
No value has been given yet. Double click or use the edit button.
executed by Data Quality Metric
Add
Name
Domain
Description
Asset Type
Full Name
valid_currency_rule
DQ Metrics
Data Quality Metric
bloomberg>valid_currency_rule>Metric
1
governs Asset
Add
Name
Domain
Description
Full Name
bloomberg
postgres-gcp
postgres-gcp>postgres>public>bloomberg
1


valid_currency_rule
Type: Data Quality Rule
Status: Candidate
Approval
Simple Approval
Vote
Edit
Move
Delete
Auto hyperlinks
Add characteristic
Overview
Tags
Comments
Diagram
Pictures
Responsibilities
References
History
Files
governs Asset
Add
Name
Domain
Description
Full Name
bloomberg
postgres-gcp
postgres-gcp>postgres>public>bloomberg
governed by Governance Asset
Add
No data available
governs Data Element
Add
Name
Domain
Description
Asset Type
crncy
postgres-gcp
Column
classified by Data Quality Dimension
Sort by
Name
Add
dq dimensions
DQD Accuracy
Full Name
Accuracy
Open tasks (1/1)
Create DQ Rule Modification
Related to
valid_currency_rule
Description
Paragraph
Please update dimension from 'Accuracy' to 'Validity'
Priority
Urgent
Data Quality Dimension(s)
Validity
Related Rules
valid_currency_rule
Owner(s)
Admin Istrator
Cancel
Create Issue


Data Governance Council
dq rules
valid_currency_rule
Type: Data Quality Rule
Status: Candidate
Approval
DQ Rule Modification
Simple Approval
Vote
Edit
Move
Delete
Auto hyperlinks
Add characteristic
Overview
Tags
Comments
Diagram
Pictures
Responsibilities
References
History
Files
governs Asset
Add
Name
Domain
Description
Full Name
bloomberg
postgres-gcp
postgres-gcp>postgres>public>bloomberg
governed by Governance Asset
Add
No data available
governs Data Element
Add
Name
Domain
Description
Asset Type
crncy
postgres-gcp
Column
classified by Data Quality Dimension
Sort by
Name
Add
dq dimensions
DQD Validity
Full Name
Validity



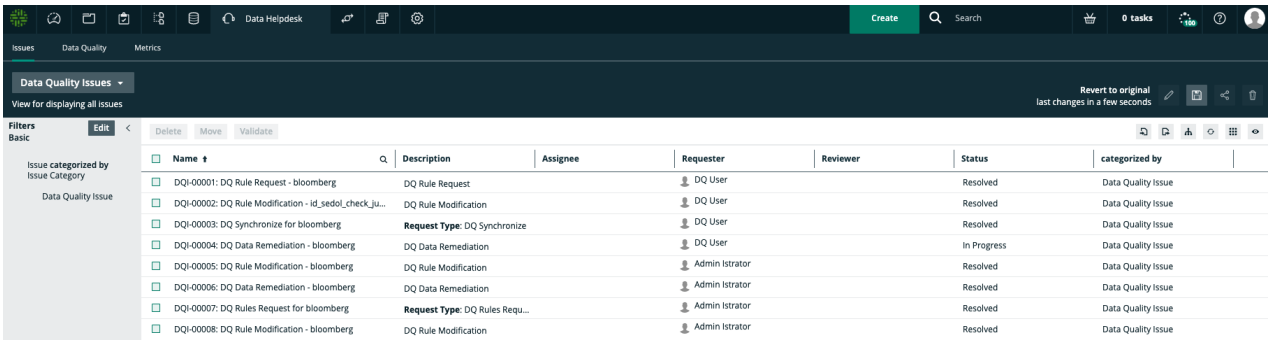
Step 3: Manage Data Quality Issues

3A. Setup Data Helpdesk Filter

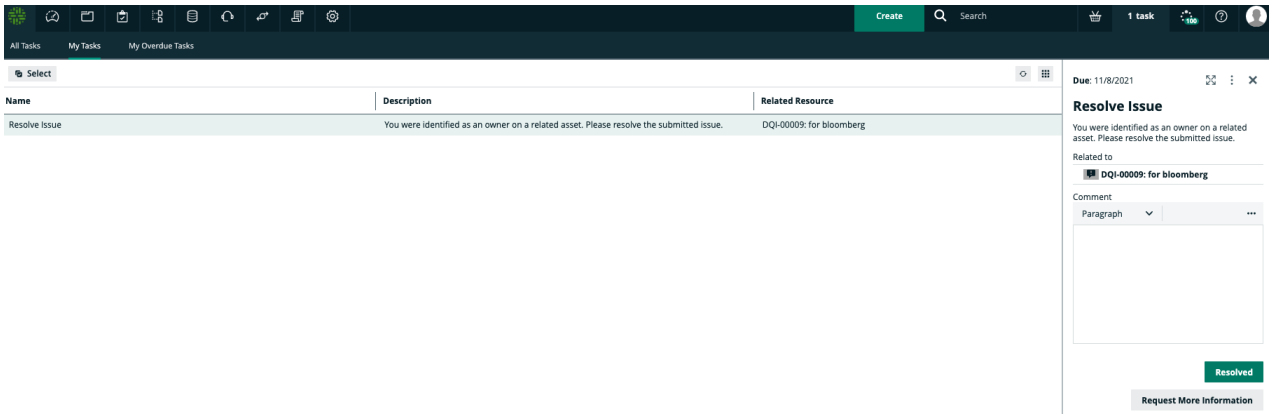
Data Helpdesk

- Select Issues
- Navigate to 'Filters'
- Properties > Attributes > Relations > Issue **categorized by **Issue Category > Input 'Data Quality Issue' > Apply
- Save button > Save View as > **'Data Quality Issues'**
- Optional settings for View: Can pin, promote, make public, make default

3B. Manage Issues From Data Helpdesk View



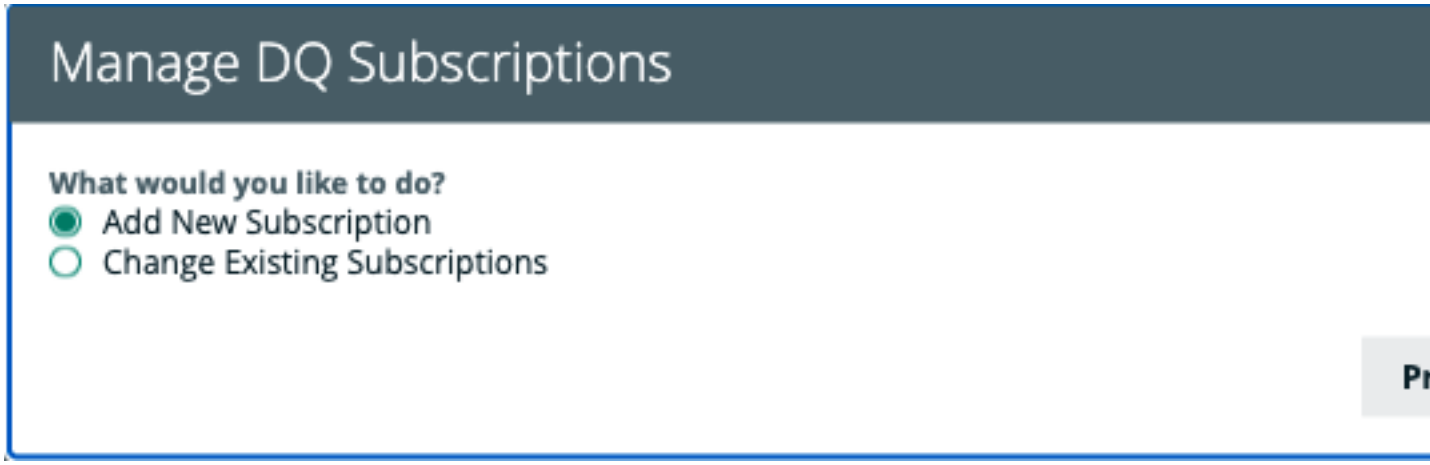
3C. Alternate: Manage Issues From Tasks



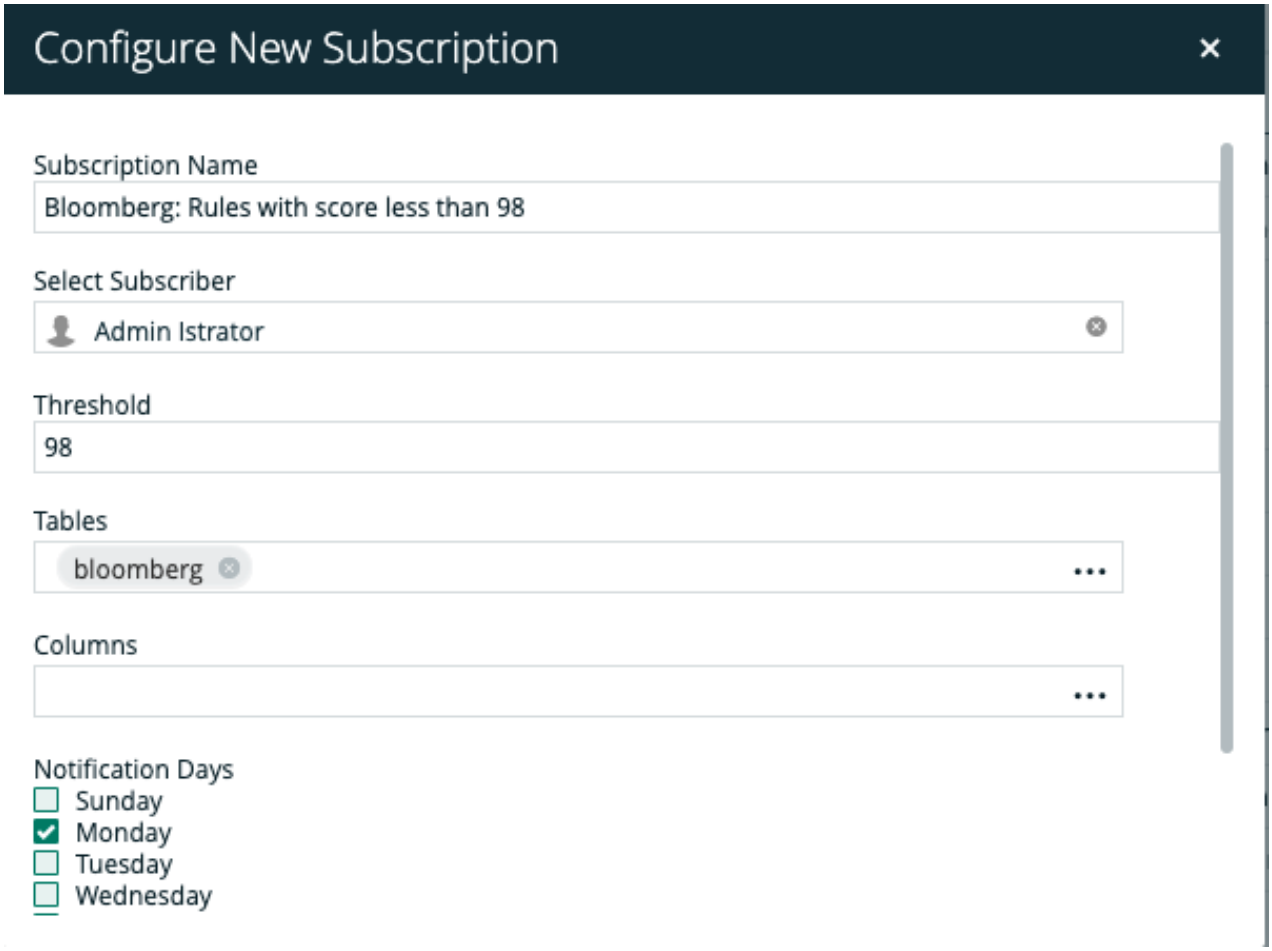
Step 4: Receive Notifications Of DQ Issues And Metrics

4A. Set Up DQ Metric Subscription

Who? Anyone can set up a DQ subscription, for yourself or for your teammates.



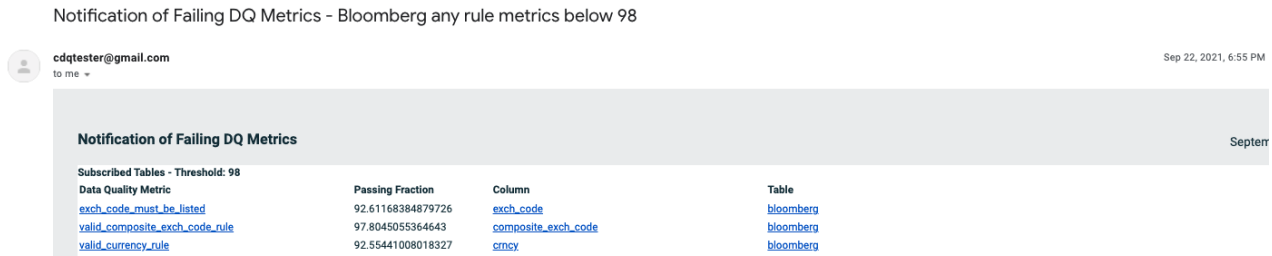
Alerts will be sent based on reviewing rules and metrics associated with **Tables** or **Columns** that violate the specified **Threshold**.



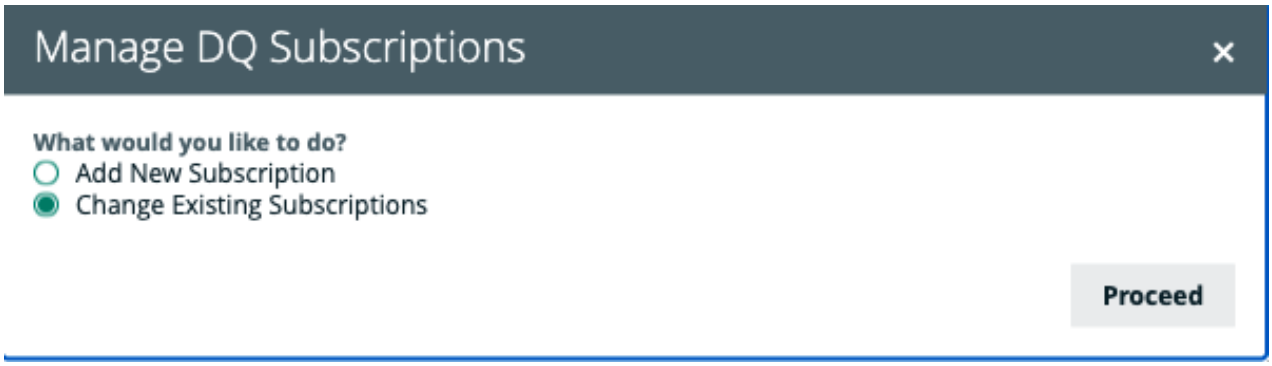
Assuming an e-mail is associated with the Subscriber within Collibra, the Subscriber will receive e-mail notifications by default at **12pm local server time**. This, along with other settings within the provided workflow, can be **adjusted** in **Eclipse**, Collibra's recommended workflow editor.

4B. Review DQ Metric Alerts

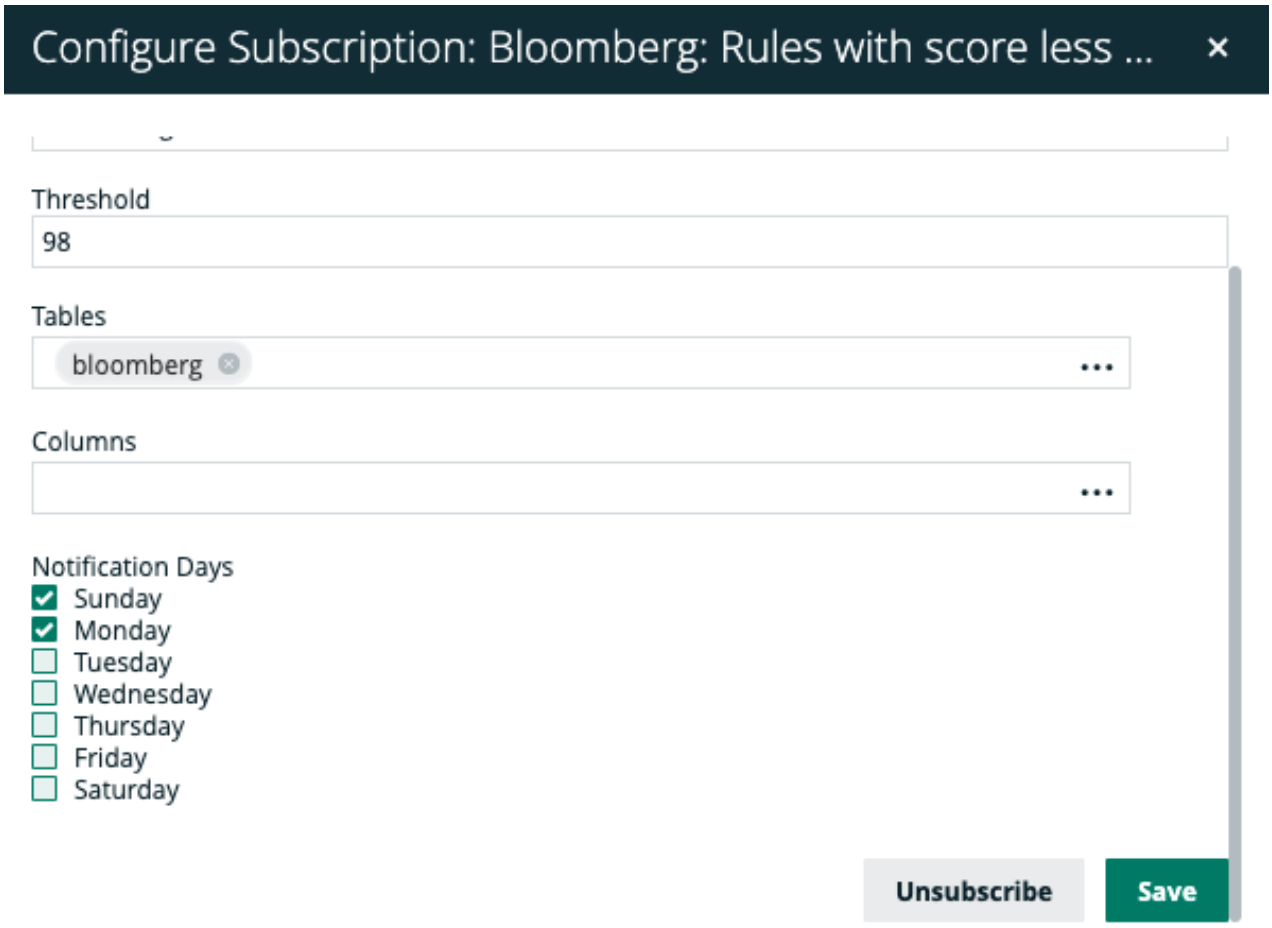
Ensure that the DQ alerts set for you are providing helpful details.



4C. Update Subscription Settings



For every subscription set up for a Subscriber, the Manage DQ Subscriptions modal will cycle through for your review. You can update **Threshold**, add or delete **Notification Days**, add or delete **Tables** or **Columns**, rename the **Subscription title**, **Save the new settings**, or simply **Unsubscribe**.



Chapter 13

Collibra DQ Solutions

Use Cases

Data Projects

Builds a Better DQ Dashboard

- **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

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.

Makes your Data Lake better.

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+ owl 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

- **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.

Assists Data Aggregation

- **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.

Creating a Data Quality Pipeline

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

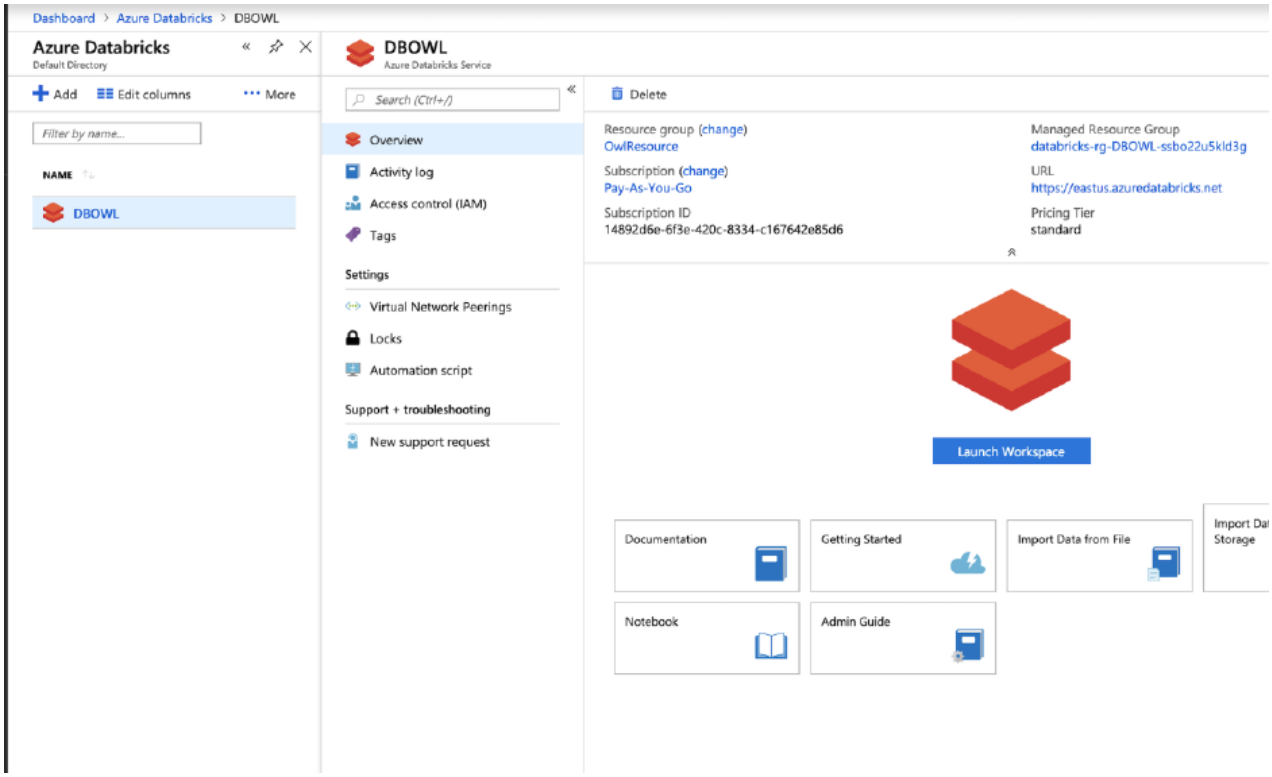
time to value for business insights as Data Scientists get time back, not cleaning / prepping the data, and helping the model maintain a longer life.

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.

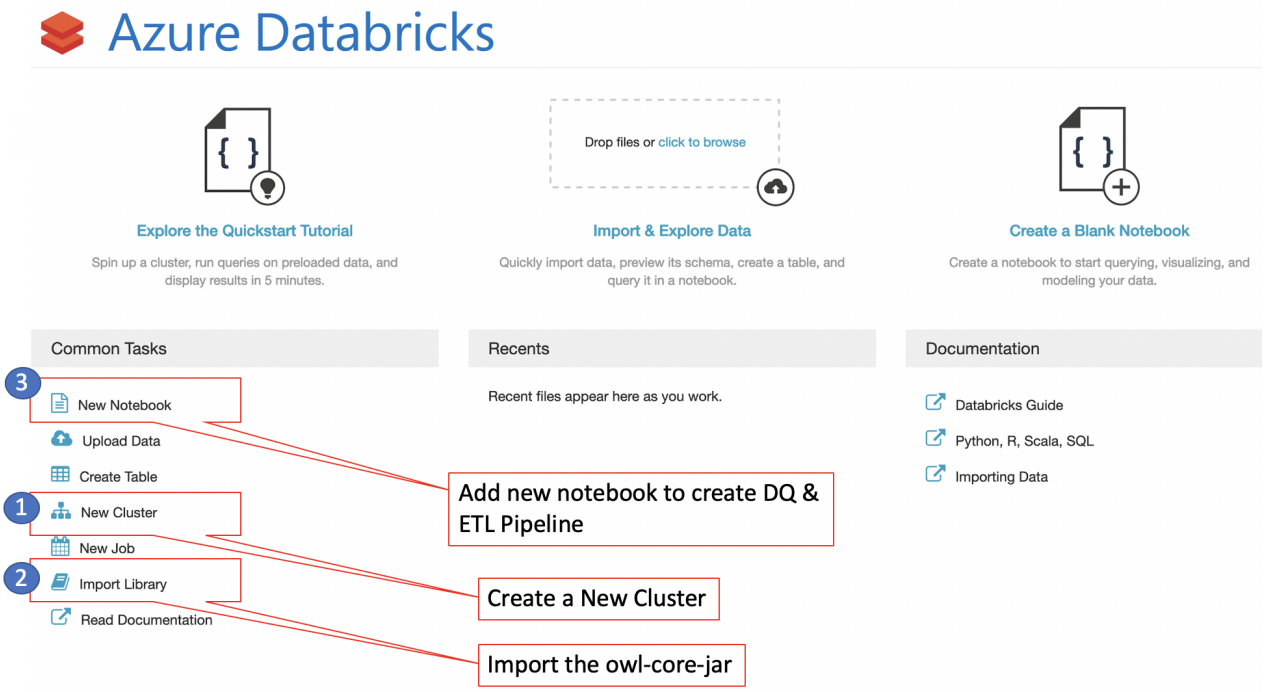


Step 2: Create a Cluster, add the DQ jar and create a Notebook.

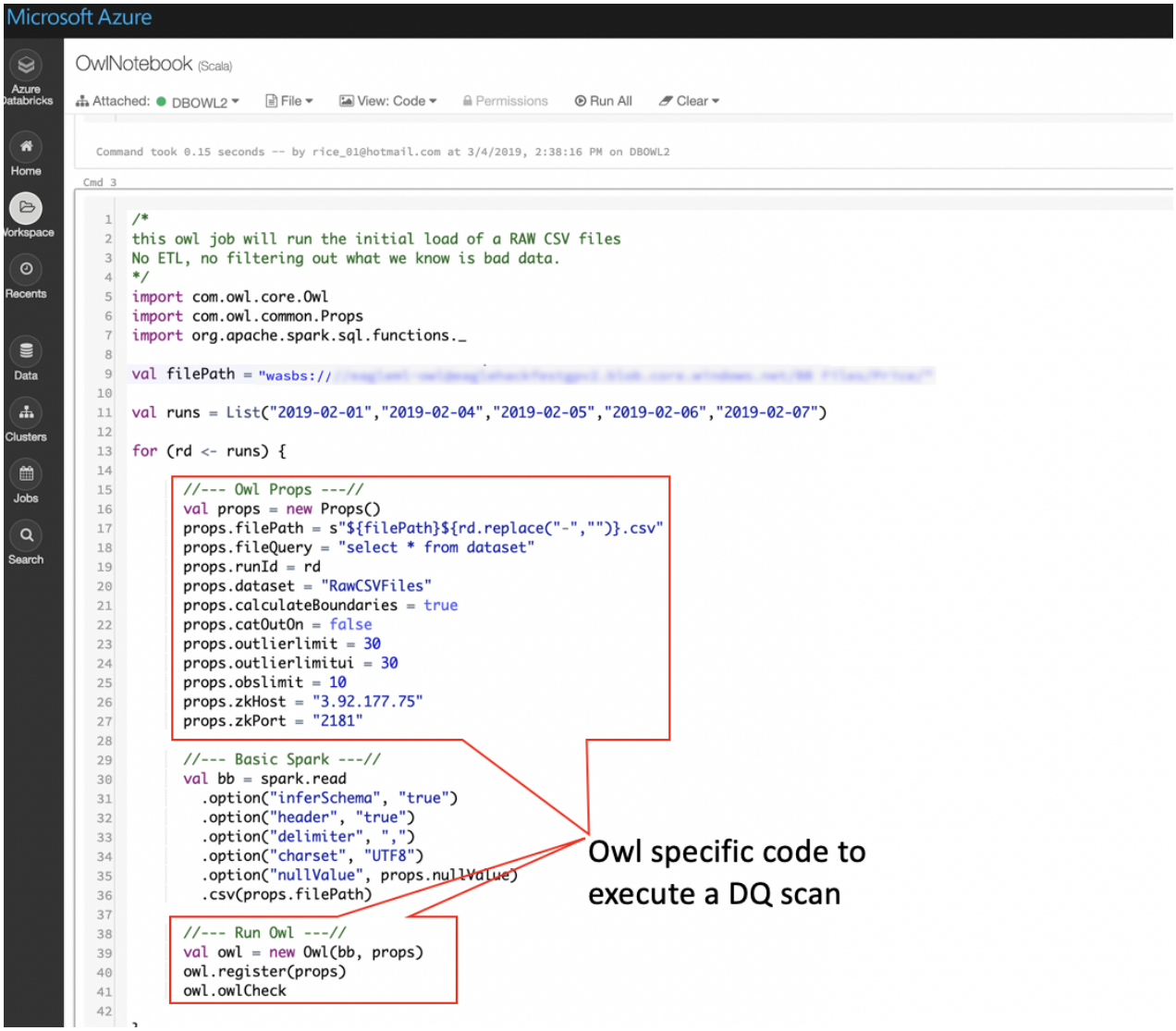
1.) Inside the Azure Databricks UI, create a cluster, provide a cluster name (in this example we will be using DBOWL2 as the cluster name) and select the Databricks Runtime Version that DQ currently supports (as of this blog post), which is Runtime: 5.2 (Scala 2.11, Spark 2.4.0).

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.

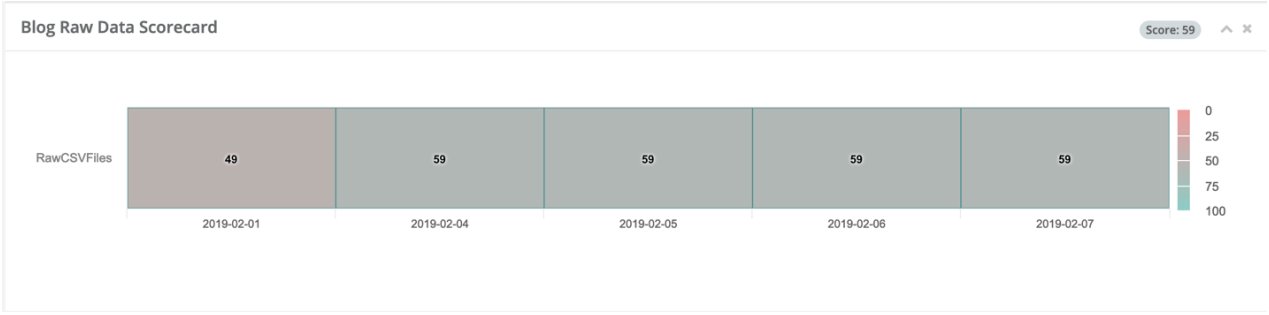
3.) Now that the Jar file has been added, create the Notebook and attach it to the cluster.



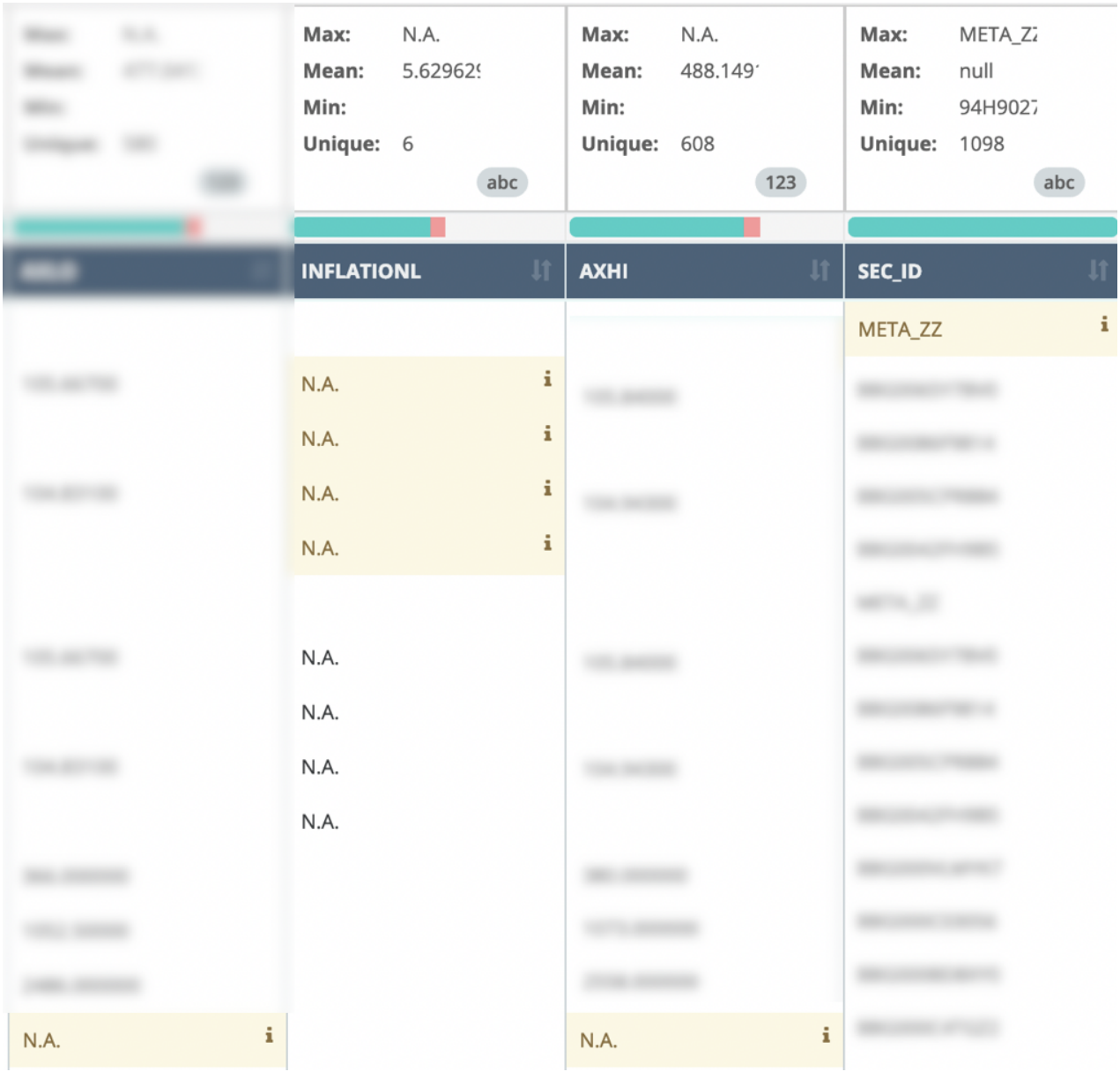
Now the cluster is running with the DQ jar loaded on the cluster. Open the Jupyter notebook attached to the cluster and begin looking at a data set as a Data Engineer would, prepping the data for use by a Data Scientist by leveraging a DQ Pipeline as shown in the below screen shot.



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, as shown in Figure 4 below.



The reason for a score of 49 on the raw data (as shown below in Figure 5) 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.



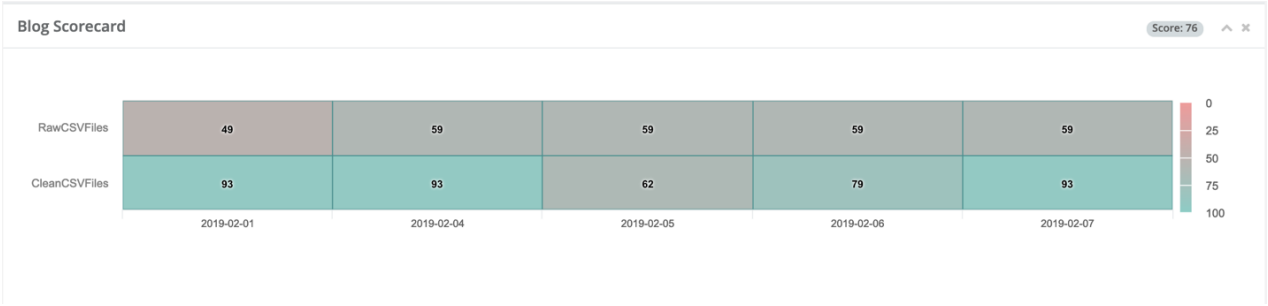
You should now have an understanding of the raw file and how you need to conform it before analysts can start to glean business value from the contents itself. First, ETL or

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 data set did not have a date that conforms easily. After the ETL process cleanses the data, you then have DQ’s Data Quality engine scan the newly processed Spark DF, storing the results into a data set called CleanCSVFiles (as shown in Figure 7 below).



Notice the composite scores in the boxes are substantially better for the CleanCSVFiles data set 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 data set when run over several days and how DQ surfaces different patterns, behaviors, trends and more in the data itself.

Bank Loans

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
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

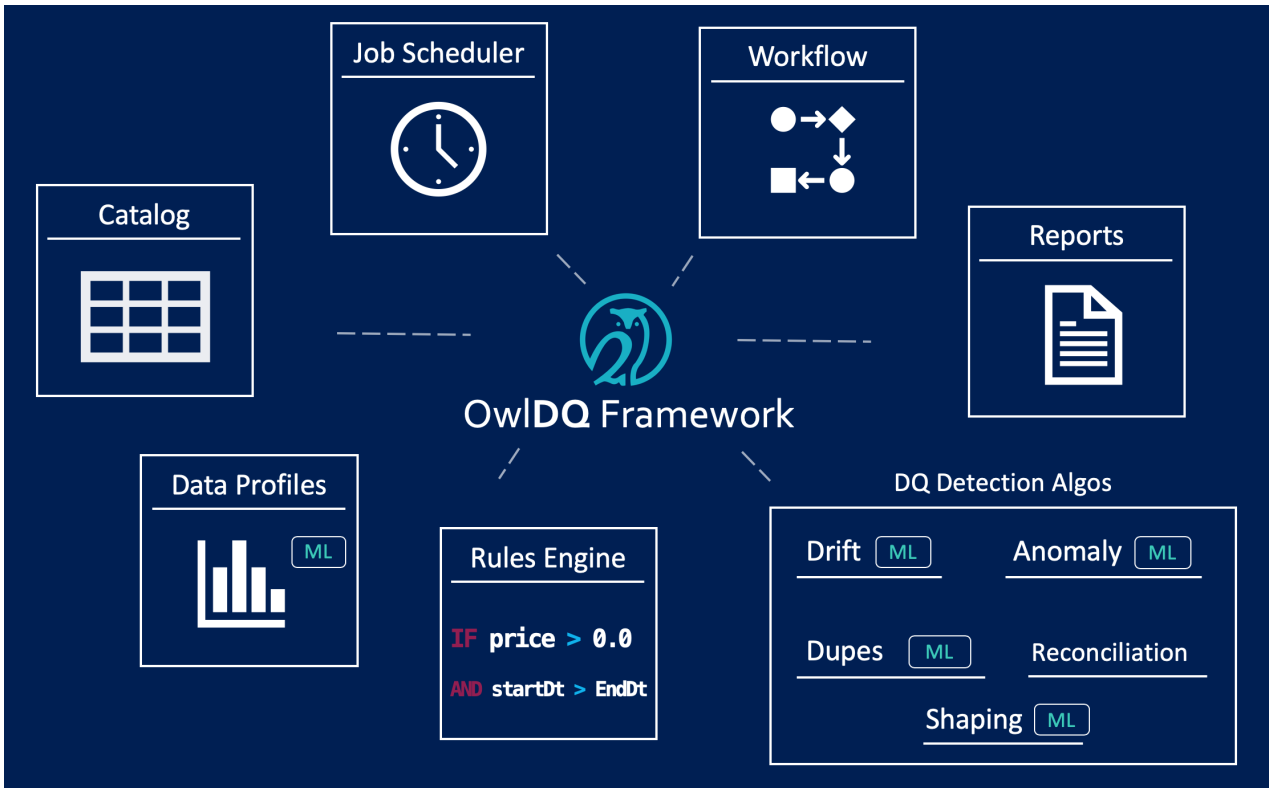
Business Check	Collibra DQ Feature	Manual vs Auto
Loan amount within lendable range	OUTLIER	AUTO
Loan amount within lendable range. Only lend money between 50K and 3M.	RULE	loan_amount between 50000 and 3000000

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.



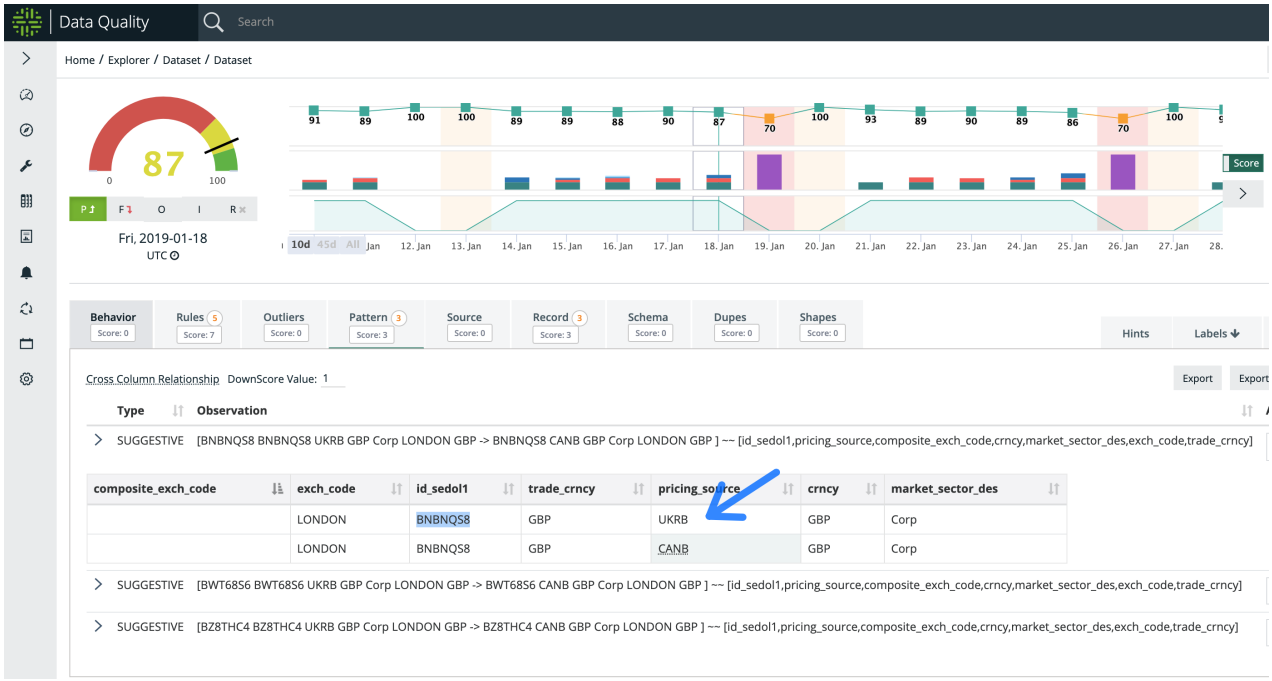
Files that can be used to replicate this example

interest_rates.csv
Binary

Owl Dataset (2).csv
Binary

Bloomberg Data

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.



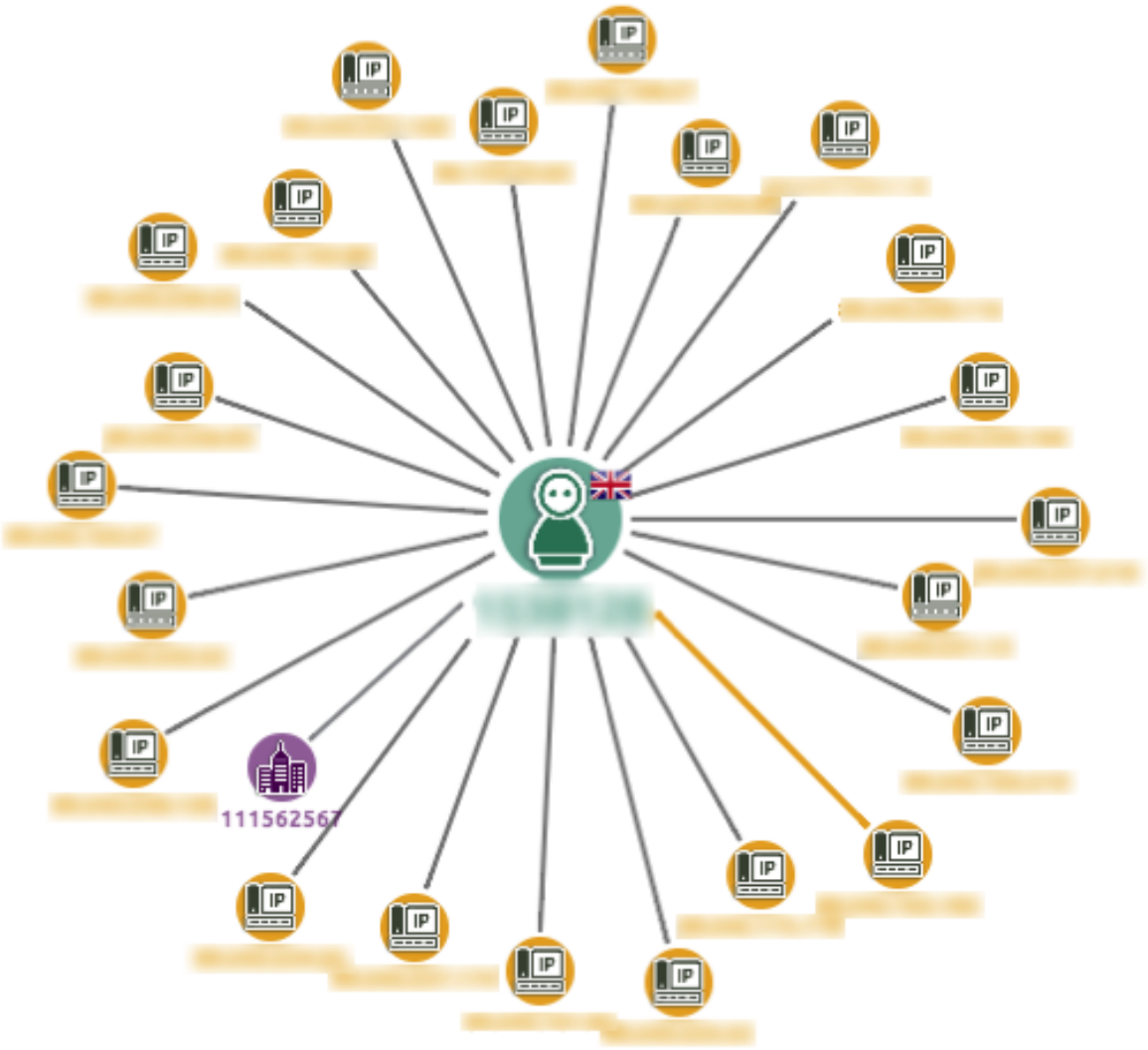
Cyber Anomalies in Real-Time

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

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

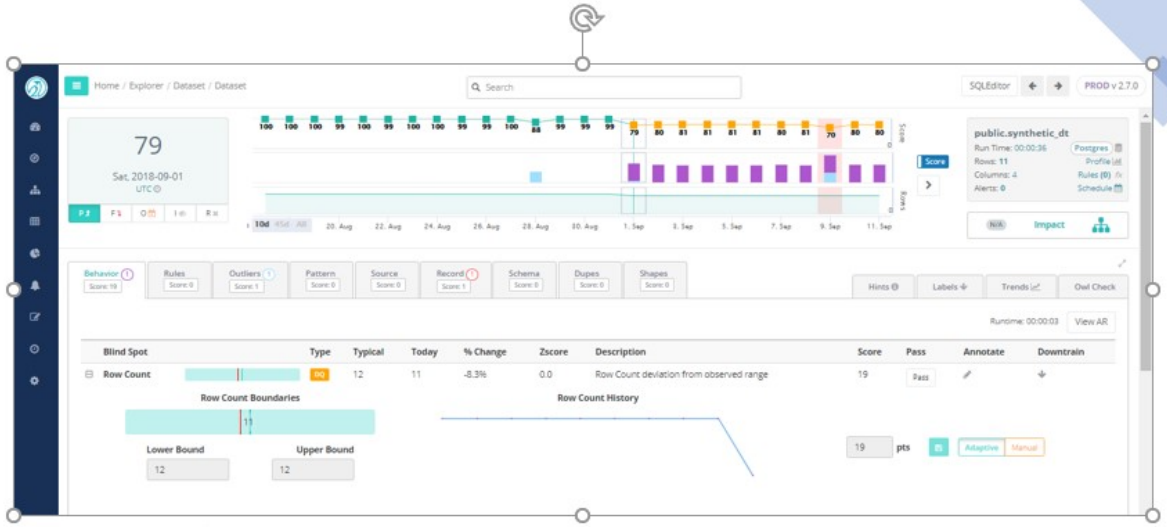
Healthcare Data Quality

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

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.

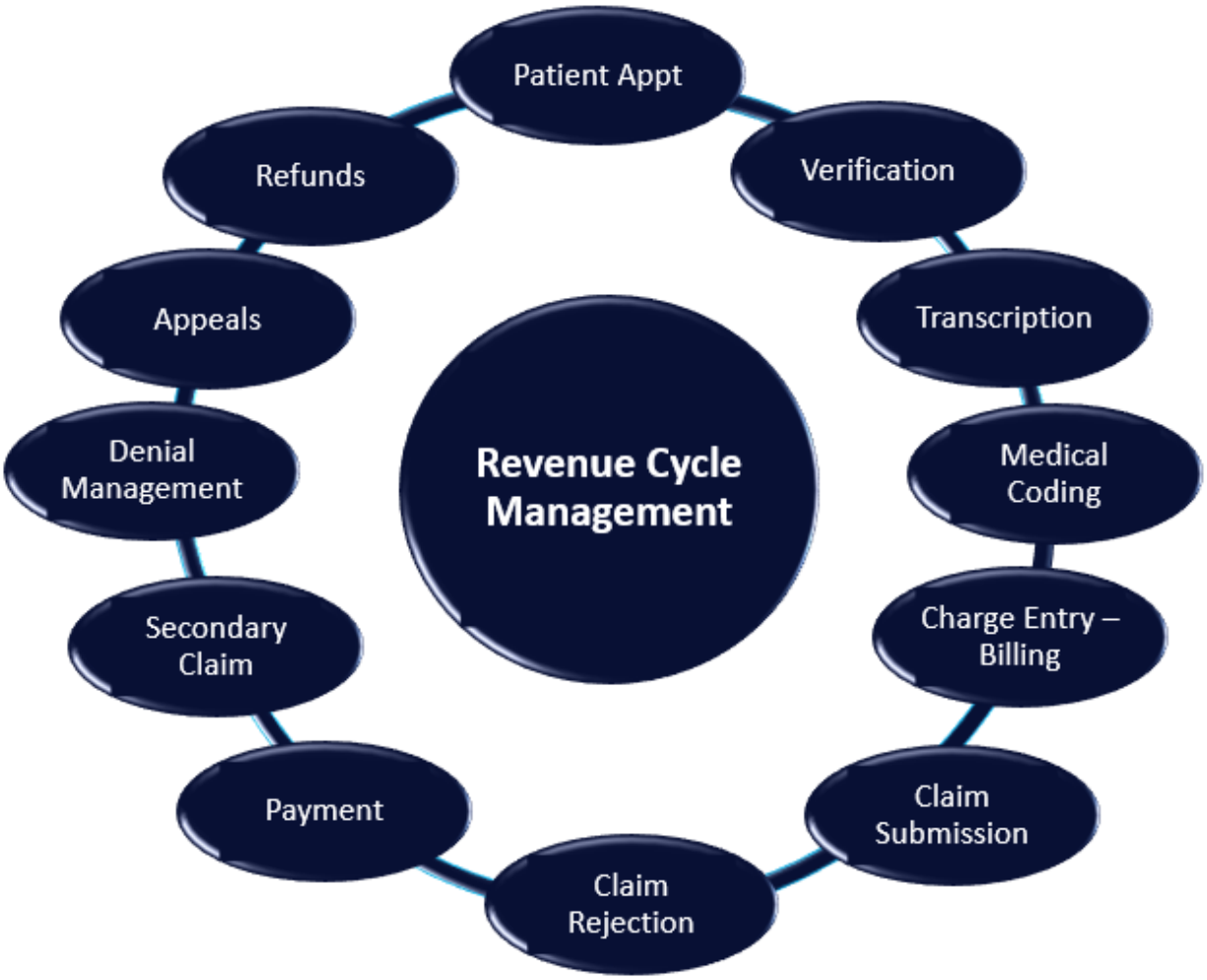
Inaccurate claims data, is the primary cause for diagnostic providers receiving incorrect payments for their services. Most providers struggle with the quality of the data that they receive, and without direct access to the patients, it can be an expensive, laborious process to correct incomplete, or 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

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).

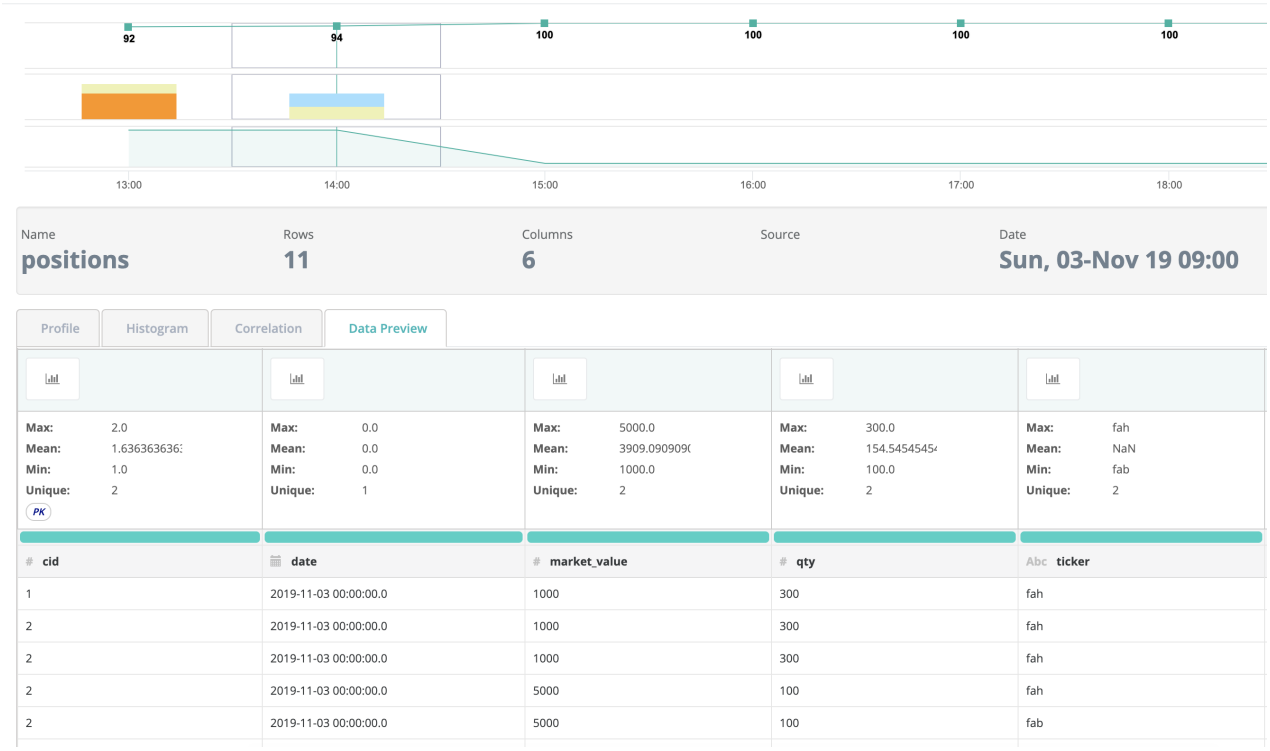
Chapter 13

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 Web



DQ Coverage for Position data

- Schema evolution
- Profiling
- Correlation analysis
- Segmentation
- Outlier detection
- Duplicate detection
- Pattern mining

Security Reference Data

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.

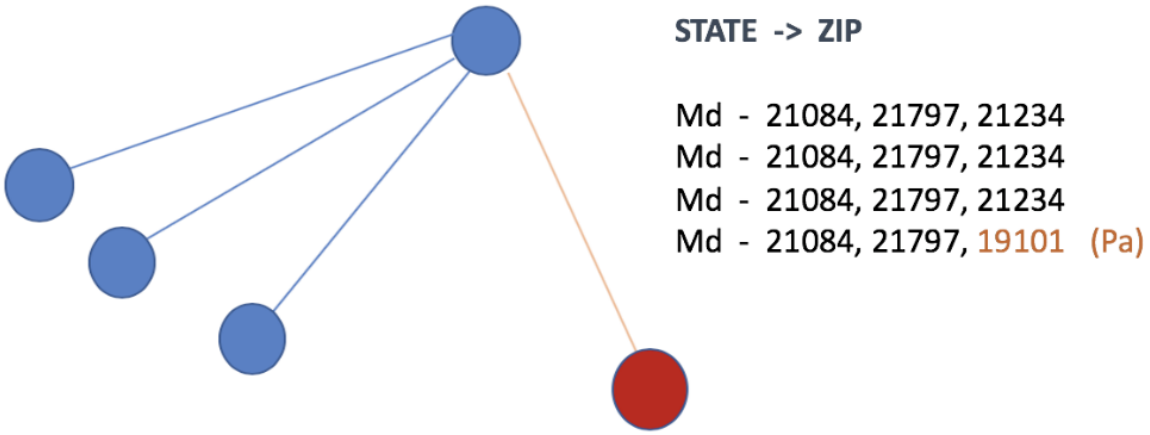
Financial firms of all shapes and sizes ingest financial data for a variety of reasons. A few vendors include Bloomberg, Thomson Reuters, ICE Data Services or SIX Financial Information.

> 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.

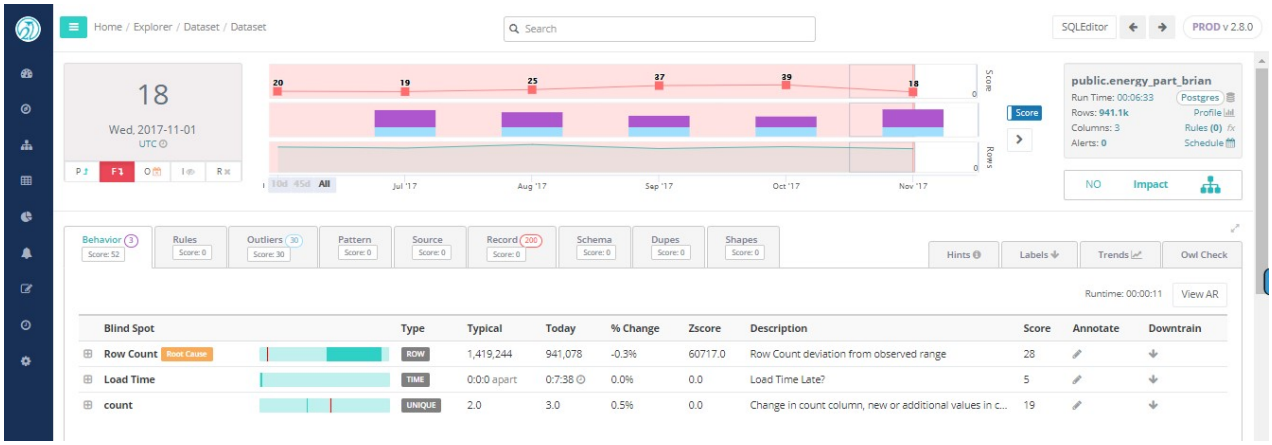
Smart Meter Data

Colliba DQ uncovered $10 million dollars in unbilled revenue for a leading U.S. energy company.

DQ's Smart Meter Data Analytics provides accurate and predictable data quality to companies often inundated with massive amounts of data and aging enterprise systems.

The U.S. smart meter data management market forecast is projected to reach $556.94 million by 2026. In 2018, U.S. electric utilities had about 86.8 million advanced (smart) metering installations. DQ provides an automated process to manage the mountain of data collected and glean critical business insights. By applying our ML algorithms during the normal data ingestion cycles, DQ uncovered $10 million dollars in unbilled revenue for a leading U.S. energy company.

In the example below, DQ detected 200 records missing from the previous run.



Consider the following opportunities that smart meter data analytics provide:

- Generate new customer insights
- Manage and prevent outages
- Improve maintenance techniques
- Build predictive models for program planning
- Develop new services and rate plans based on customer requirements

Identification of unbilled revenue: Meter events and usage information helps illustrate a picture of the customer's energy usage over time. This helps detect energy theft, meter tampering, and equipment issues that may be affecting service levels.

Outage event analysis and prevention: Today, some utilities are still unable to verify an outage unless personnel physically visit the suspected problem area to confirm. With outage event analysis, however, the utility knows the exact piece of equipment causing the problem, along with the customers directly impacted by it.

Meter quality assurance: Focusing on meter reading performance enables utilities to ensure reliability. When meter readings are expected but not delivered, the system provides an alert, and calculates overall data score from previous runs . Utilities are notified to potential data quality issues they never would have identified in the past.

It is meter data analytics that will enable utilities to tackle the problems of the future.

Validating Data Movement

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.

Get Started Today

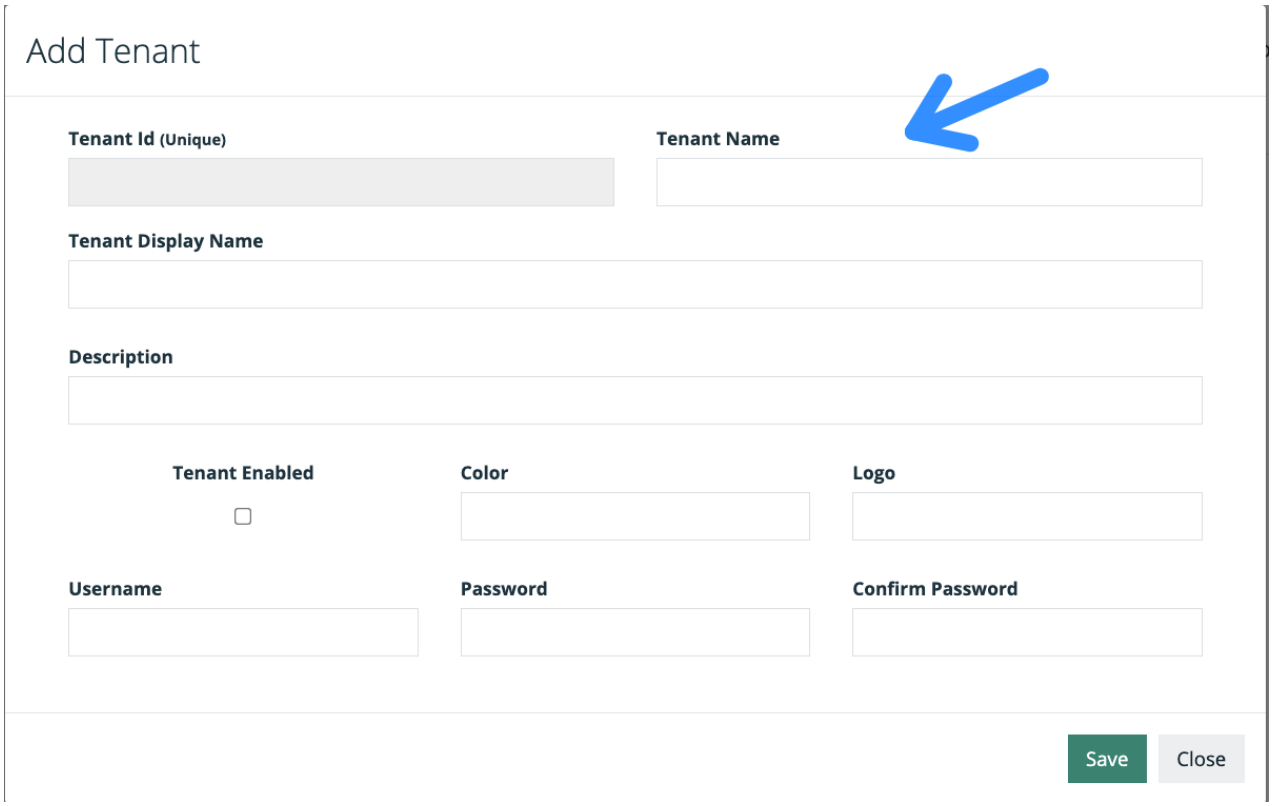
Collibra DQ source validation allows you to focus on things that actually move your project forward. For more information, contact

For more information, please contact info@collibra.com or schedule a demo at https://www.collibra.com/.

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

Logging

When you encounter an issue in Collibra Data Quality & Observability, 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.

Log	Description
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.
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.

Log	Description
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 Hadoop Install.

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.

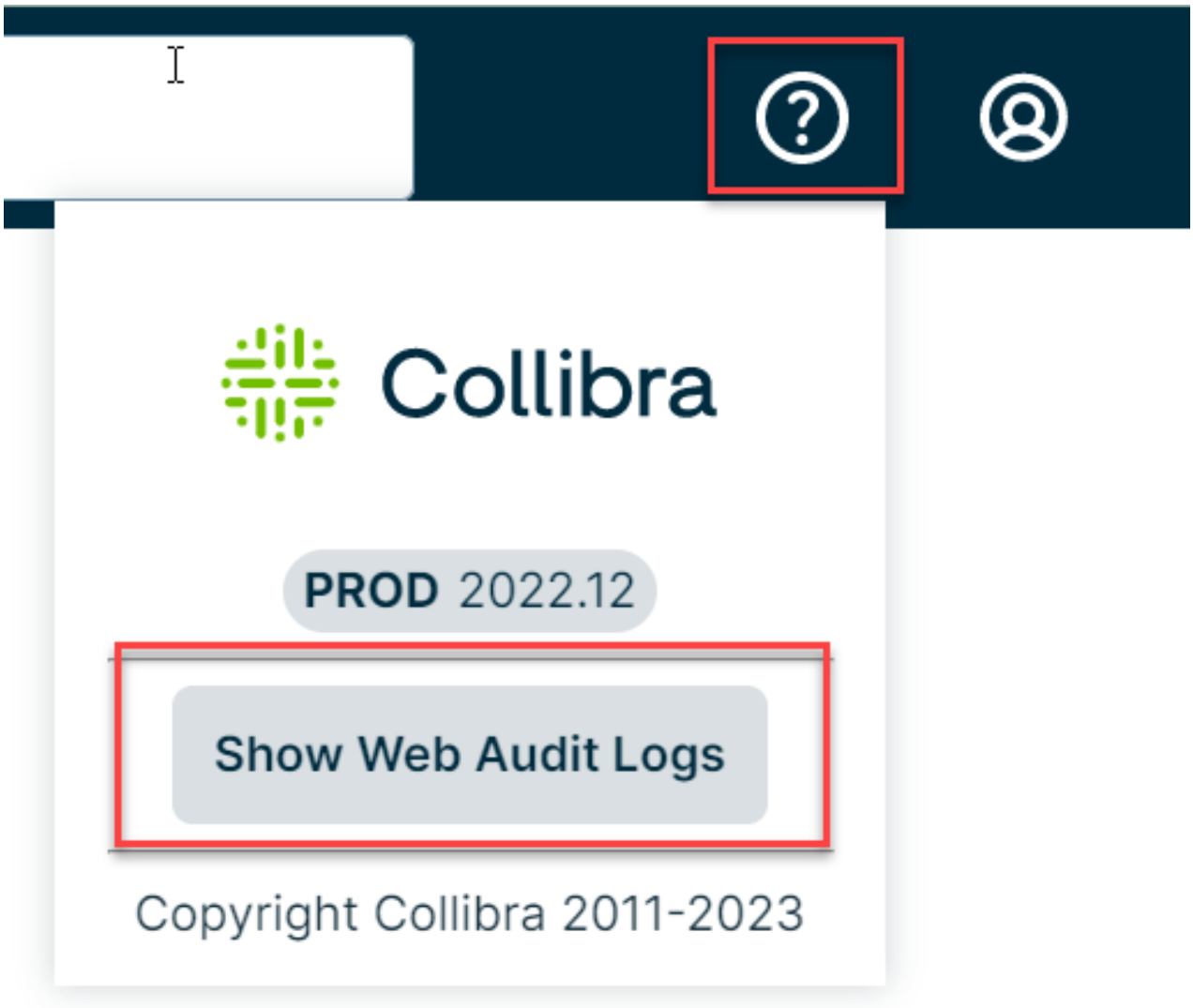
Rules Web Audit Logs

For Rules, the web app only logs the SQL produced to get the DQ Job estimate.

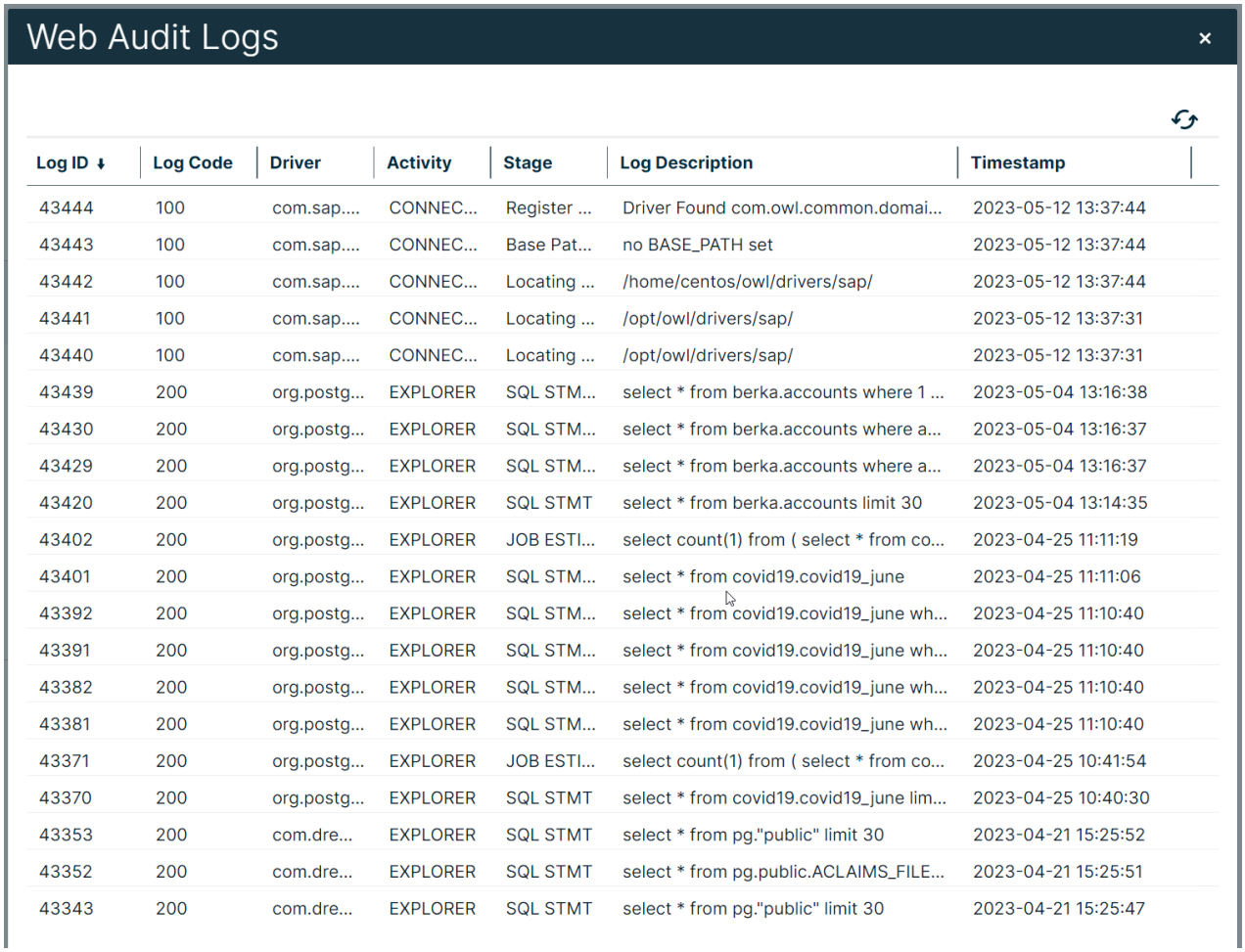
Steps

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.



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.

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.

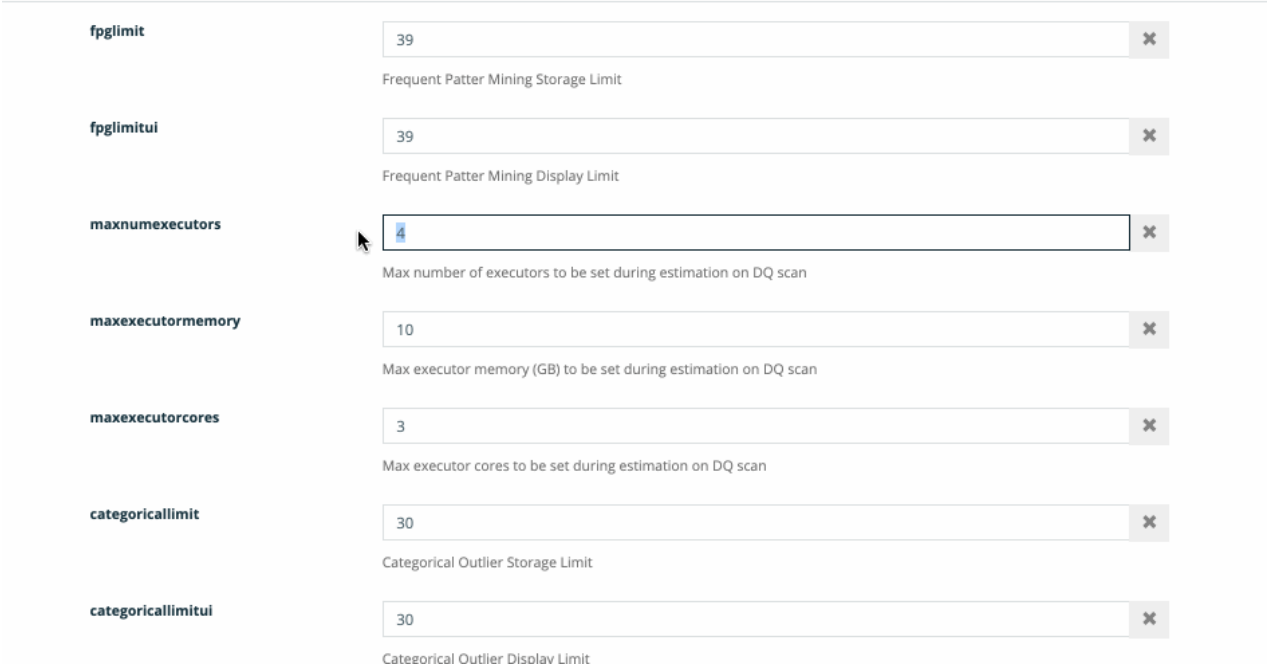
Chapter 14

Collibra DQ Benchmarks

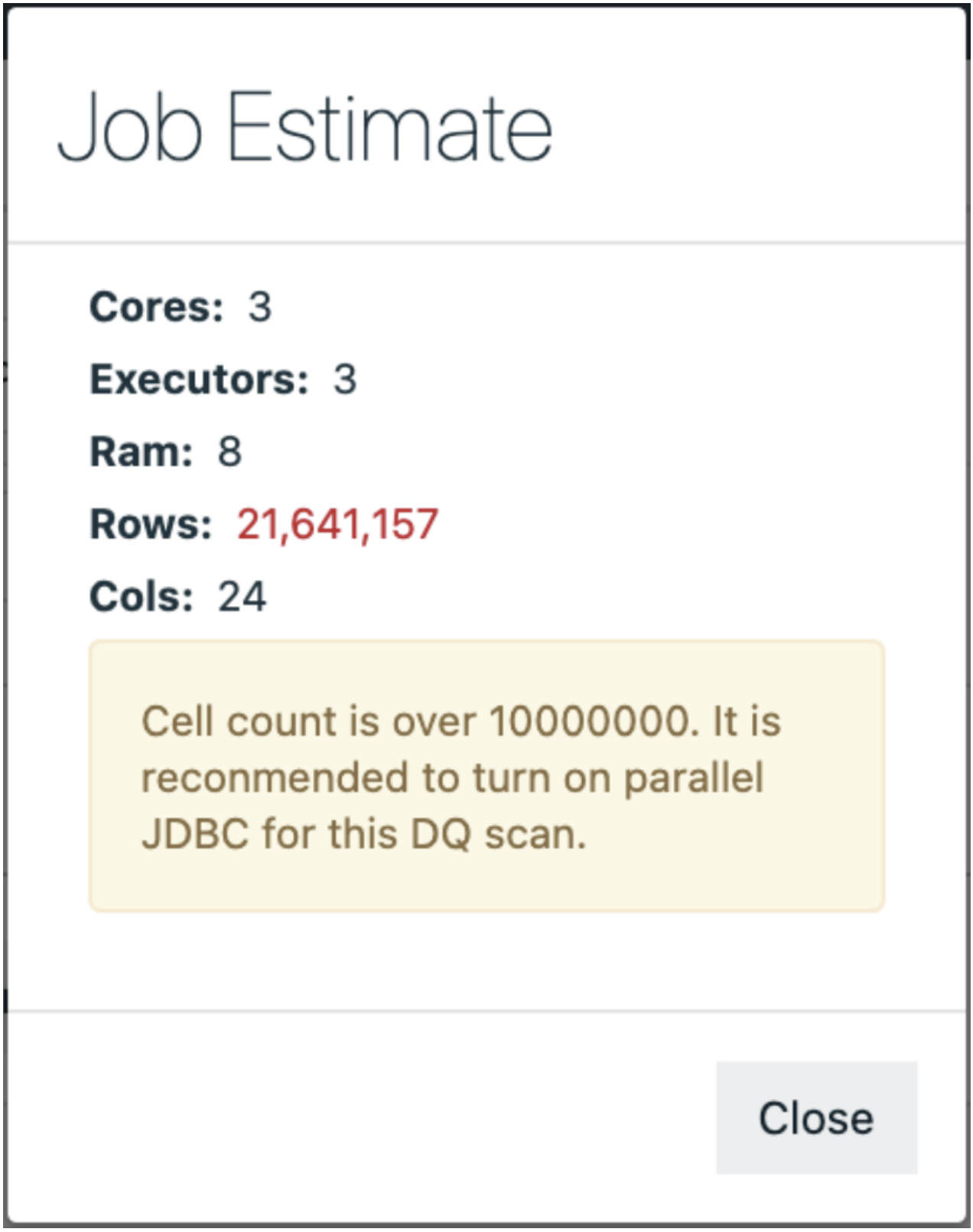
Performance Settings

Job Limits

Limits can be set to limit resources that are requested. There are options for cores, memory, executors, and cells (maxexecutorcores, maxexecutormemory, maxnumexecutors and maxcellcountforparalleljdbc).

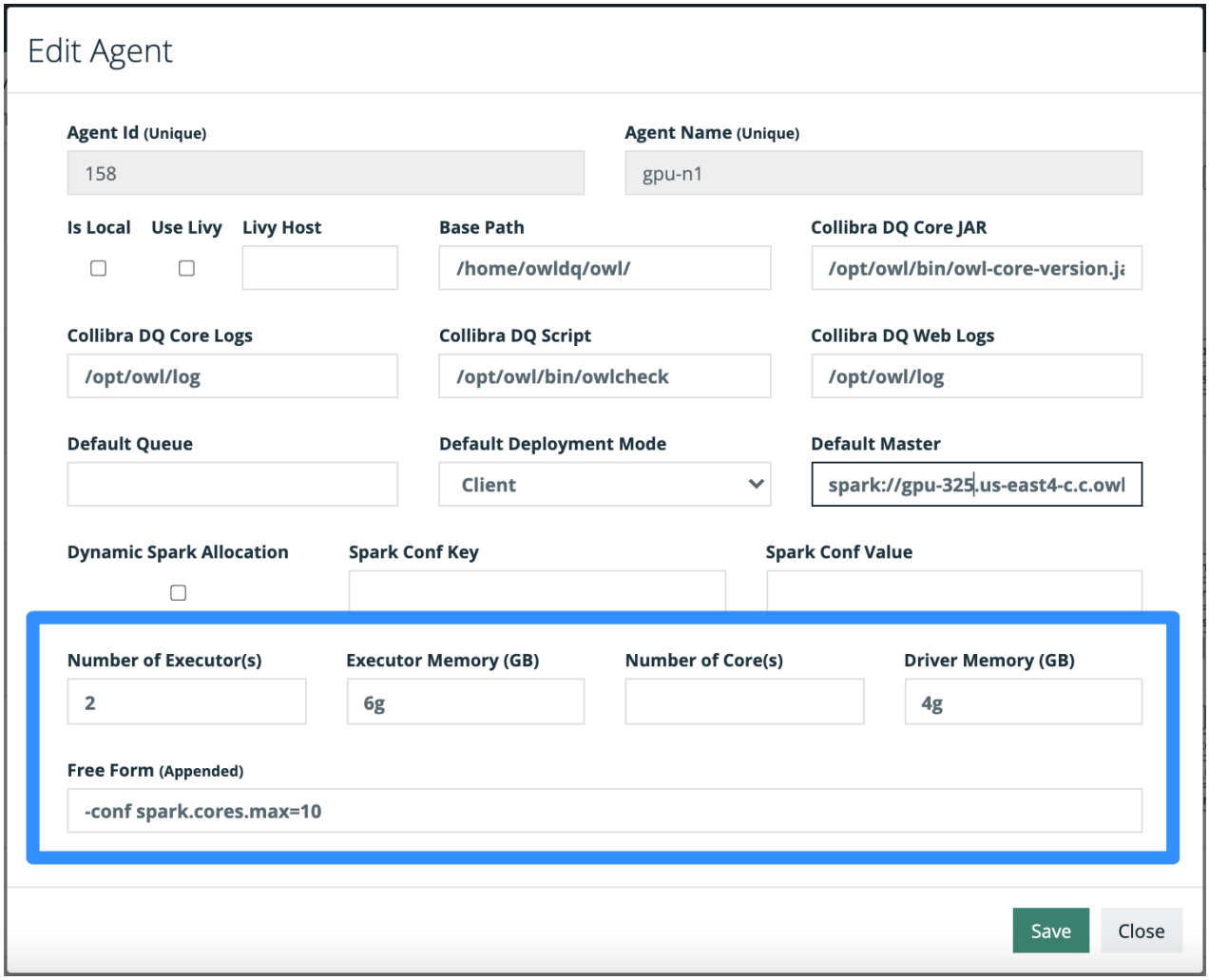


If you request more cells than the limit, you should see a warning message before hitting run.


Job Estimate
Cores: 3
Executors: 3
Ram: 8
Rows: 21,641,157
Cols: 24
Cell count is over 10000000. It is reconmended to turn on parallel JDBC for this DQ scan.
Close

Agent Defaults

Set defaults at the agent level. These should be right-sized to your environment and be used as defaults for jobs with when estimate is not available (primarily local files and remote files).



Performance Tests

Cells Per Second Performance Theory (9.5M CPS)

Rows	Cols	Cells	Seconds	Minutes	Hours	
1,000,000	1	1,000,000	0.11	0.00	0.000	
1,000,000	10	10,000,000	1.05	0.02	0.000	
1,000,000	100	100,000,000	10.53	0.18	0.003	
10,000,000	100	1,000,000,000	105.26	1.75	0.029	
100,000,000	100	10,000,000,000	1,052.63	17.54	0.292	
1,000,000,000	10	10,000,000,000	1,052.63	17.54	0.292	defining critical data elements
1,000,000,000	100	100,000,000,000	10,526.32	175.44	2.92	

Load and Profile

Dataset Name	GBs in Memory	Rows	Cols	Cells	Num Execs	Num Cores	Exec Memory	Network Time	Total Time
NYSE	0.1G	103K	9	816K	1	1	1G	00:00:15	00:00:48
AUM	14G	9M	48	432M	5	1	4G	00:01:20	00:03:50
ENERGY	5G	43M	6	258M	8	3	3G	00:00:00	00:04:35
INVEST_ DATA	20G	3.8M	158	590M	3	2	3G	00:00:40	00:03:32

NYSE

Postgres database call, no concurrent processing, simple case, small data.

```
-bhtimeoff -numexecutors 1
-lib "/opt/owl/drivers/postgres"
-executormemory 1g
-h metastore01.us-east1-b.c.owl-hadoop-
cdh.internal:5432/dev?currentSchema=public
-drivermemory 1g -master k8s:// -ds public.nyse_128 -deploymode
cluster
-q "select * from public.nyse" -bhlb 10 -rd "2020-10-26"
-driver "org.postgresql.Driver" -bhminoff
-loglevel INFO -cxn postgres-gcp -bhmaxoff
```

AUM

Postgres database call uses parallel JDBC, split on aum_id serial id.

```
-owluser kirk
-lib "/opt/owl/drivers/postgres" -datashapeoff
-numpartitions 6 -ds public.aum_dt2_50
-deploymode cluster -bhlb 10 -bhminoff
-cxn postgres-gcp -bhmaxoff -bhtimeoff
-numexecutors 6
-executormemory 4g -semanticoff
-h metastore01.us-east1-b.c.owl-hadoop-
cdh.internal:5432/dev?currentSchema=public
-columnname aum_id -corroff -drivermemory 4g -master k8s://
-q "select * from public.aum_dt2" -histoff -rd "2020-10-27"
-driver "org.postgresql.Driver" -loglevel INFO -agentjobid 7664
```

ENERGY

HDFS file with 43 million rows, converting a string date to date type, deploy mode client.

```
-f "hdfs:///demo/owl_usage_all.csv" \
-rd "2019-02-02" \
-ds energy_file \
-loglevel DEBUG -readonly \
-d "," -df dd-MMM-yy \
-master yarn \
-deploymode client  \
-numexecutors 3 \
-executormemory 10g
```

Load Profile Outliers

NYSE - 1:10 total runtime. 20 seconds for outliers

```
-bhtimeoff -owluser kirk -numexecutors 1
-lib "/opt/owl/drivers/postgres" -executormemory 1g
-dl -h metastore01.us-east1-b.c.owl-hadoop-
cdh.internal:5432/dev?currentSchema=public
-drivermemory 1g -master k8s:// -ds public.nyse_128 -deploymode
cluster
-q "select * from public.nyse" -bhlb 10
-rd "2020-10-27" -driver "org.postgresql.Driver"
-bhminoff -loglevel INFO -cxn postgres-gcp -bhmaxoff -agentjobid
7721
```

Performance Tuning

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
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.

> 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 you 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.

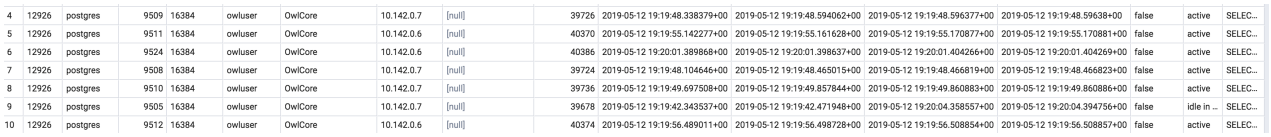
JDBC

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 19:19:48.46681+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",
  }
```

Performance Considerations

Performance Recommendations

Performance is a function of available hardware.

When running Collibra DQ Scans on a Hadoop distribution:

- Check YARN resource manager.
- Check limits on queue size.
- Contact Platform Administration team on any limitations.

When running DQ Scans on Spark standalone (single node):

- Check Spark endpoint (example: http://<IP>:<PORT>).
- Suggested maximum size on 16 core x 64GB machine: 100 million rows * 200 columns = 2 billion cells.
- If exceeding 2 billion cells, limit the width by selecting certain columns or limit depth with a WHERE clause or a FILTER condition.

When running DQ Scans on EKS:

- Check your compute pool for available pods.
- Check your worker configuration and your Spark operator configuration.
- Check minimum and maximum of allowed workers.

How-To

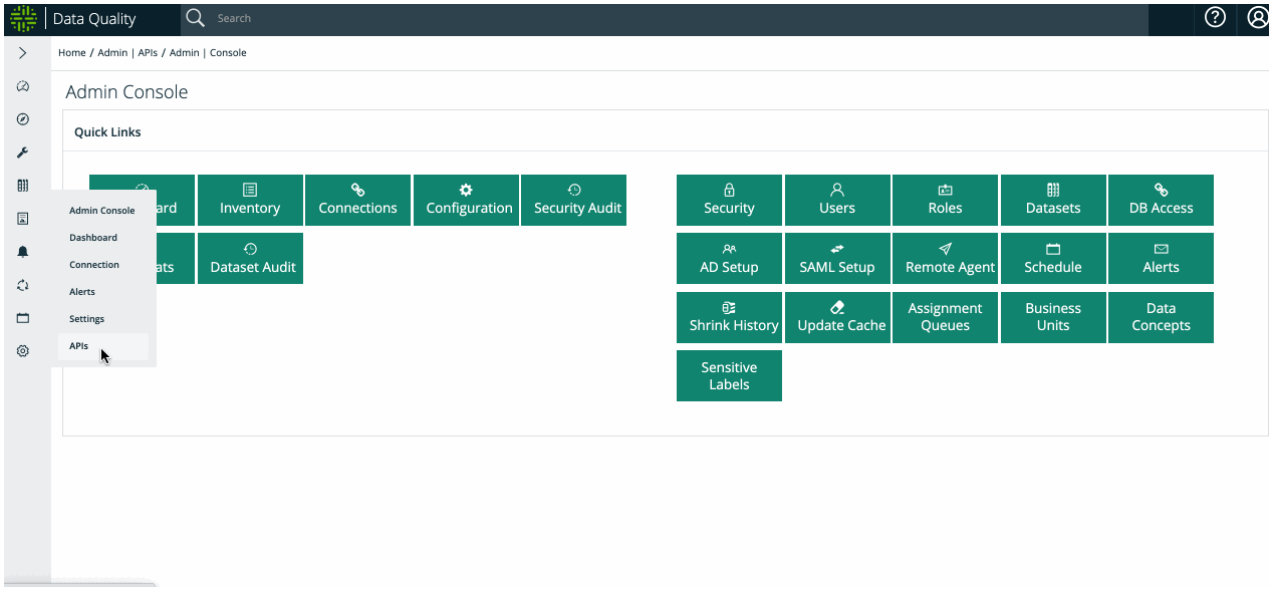
Increment DQ Scans with gradually increasing limits. Starting with a low level allows you to confirm whether database has proper indexing, skip scanning, or partitioning. Incrementing also allows you to validate security and connectivity quickly.

- Test 1: Limit 1k rows.
- Test 2: Limit 1mm rows.
- Test 3: Limit 10mm rows.

Chapter 15

Collibra DQ APIs

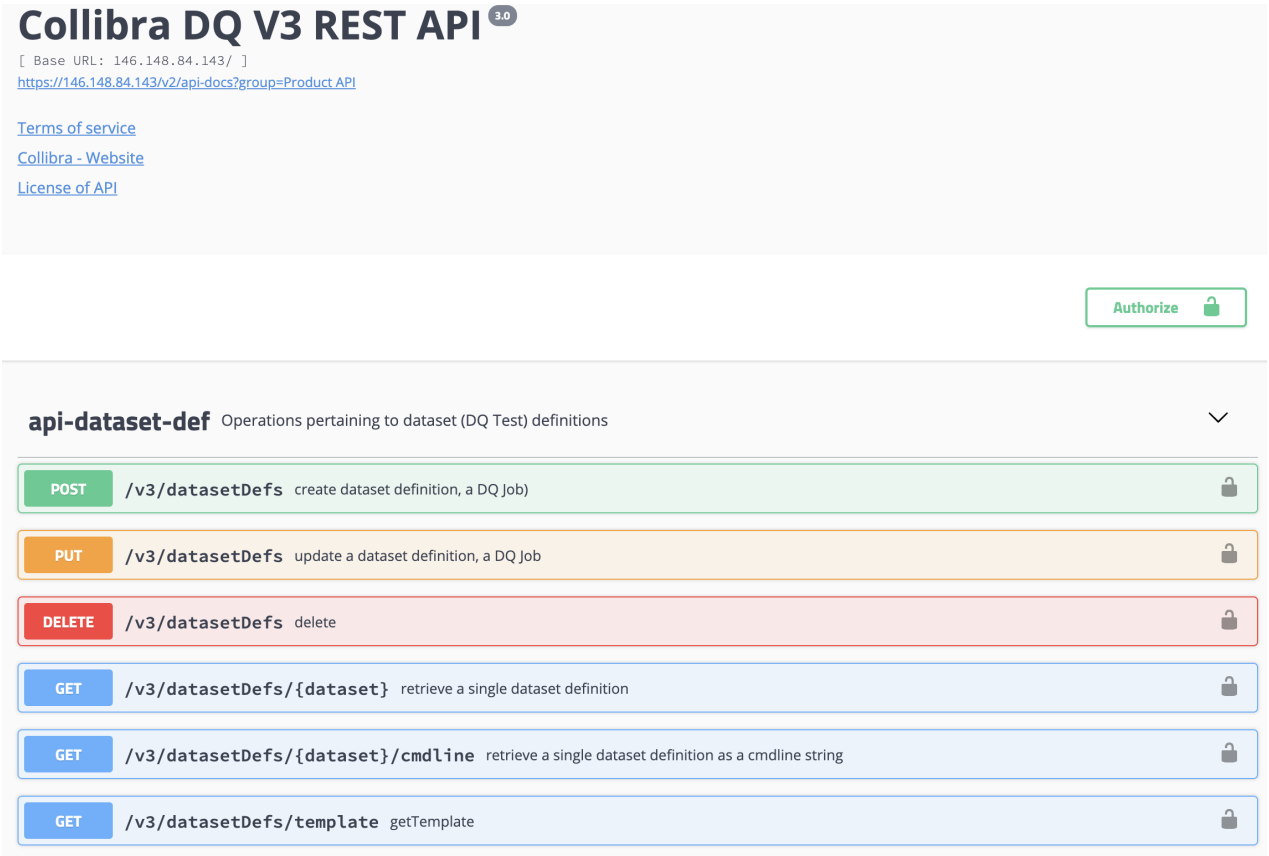
Rest



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.

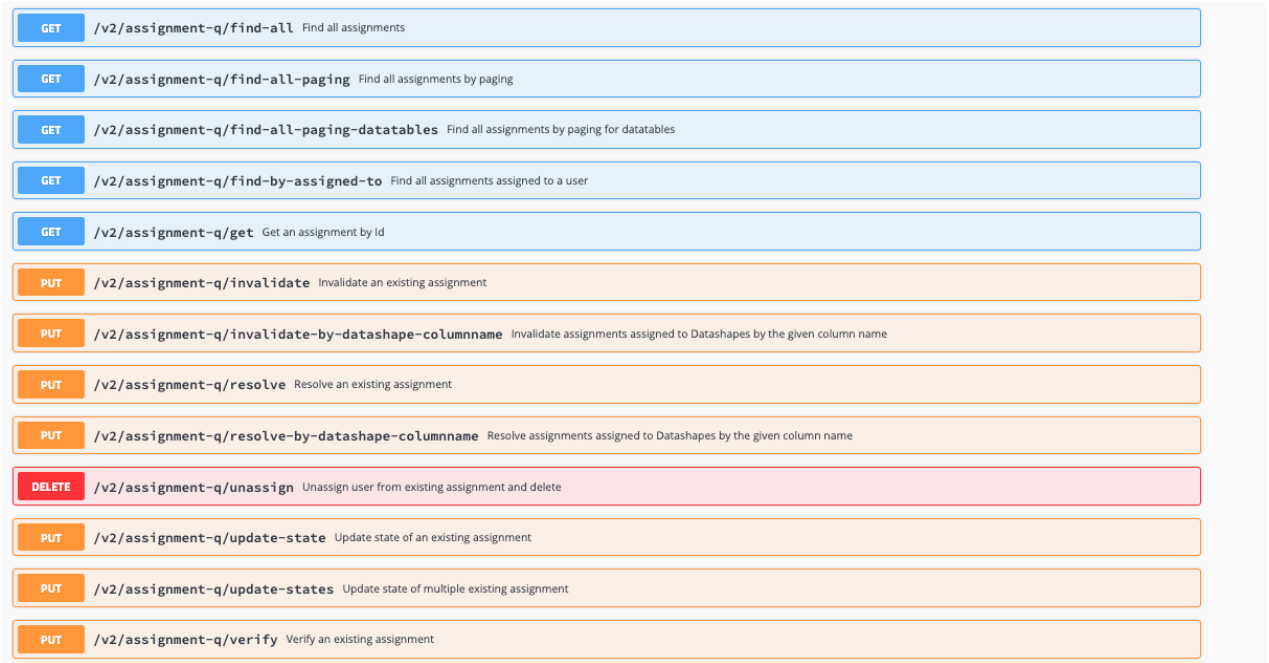
```
response = requests.get(
    url = owl + '/v3/jobs/'+jobId,
    headers=owl_header
)
```

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": {
```

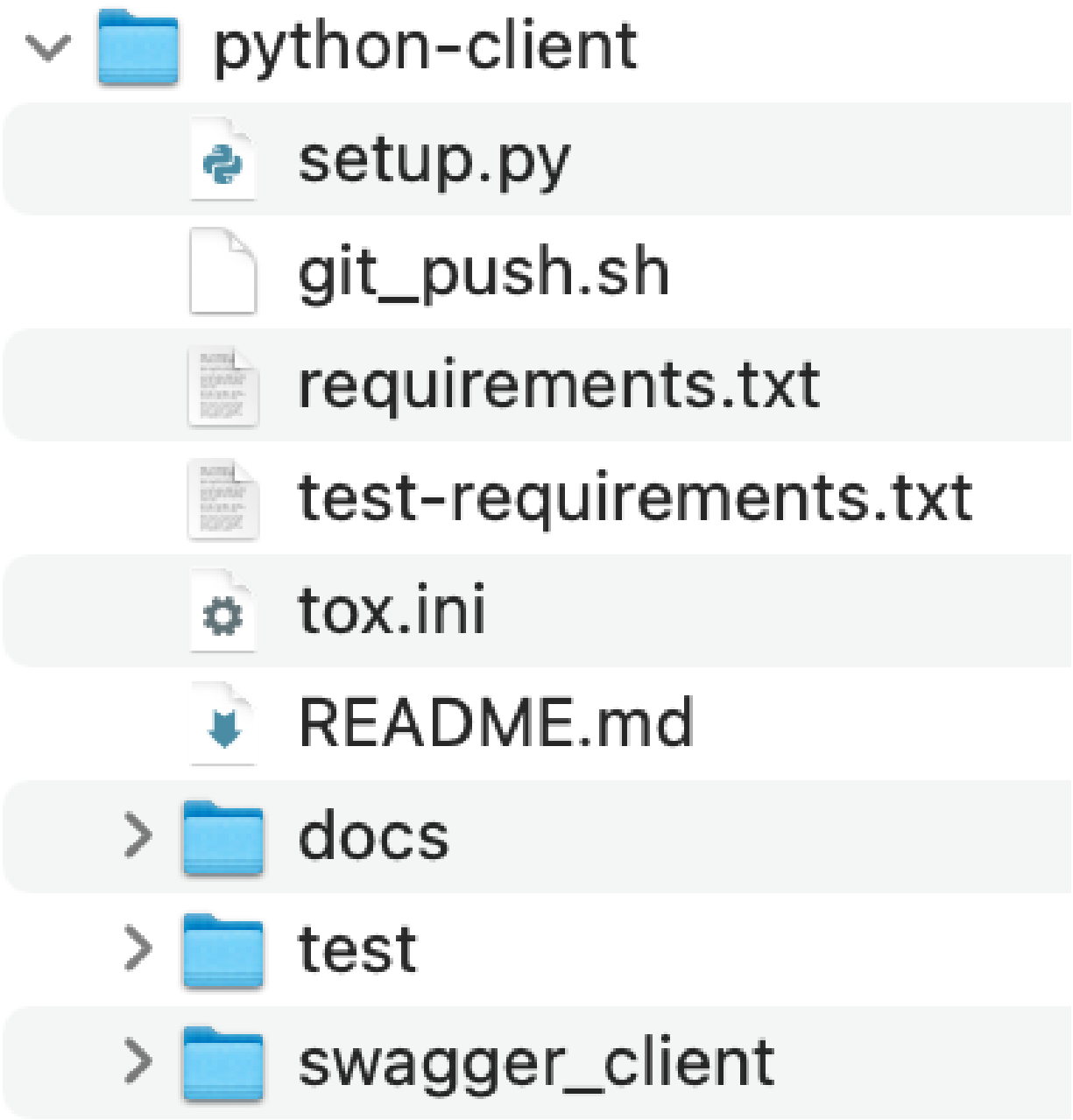
```
      "on": true,
      "only": false,
      "include": null,
      "exclude": null,
      "shape": true,
      "correlation": null,
      "histogram": null,
      "semantic": null,
      "limit": 300,
      "histogramLimit": 0,
      "score": 1,
      "shapeTotalScore": 0,
      "shapeSensitivity": 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)
```

Swagger

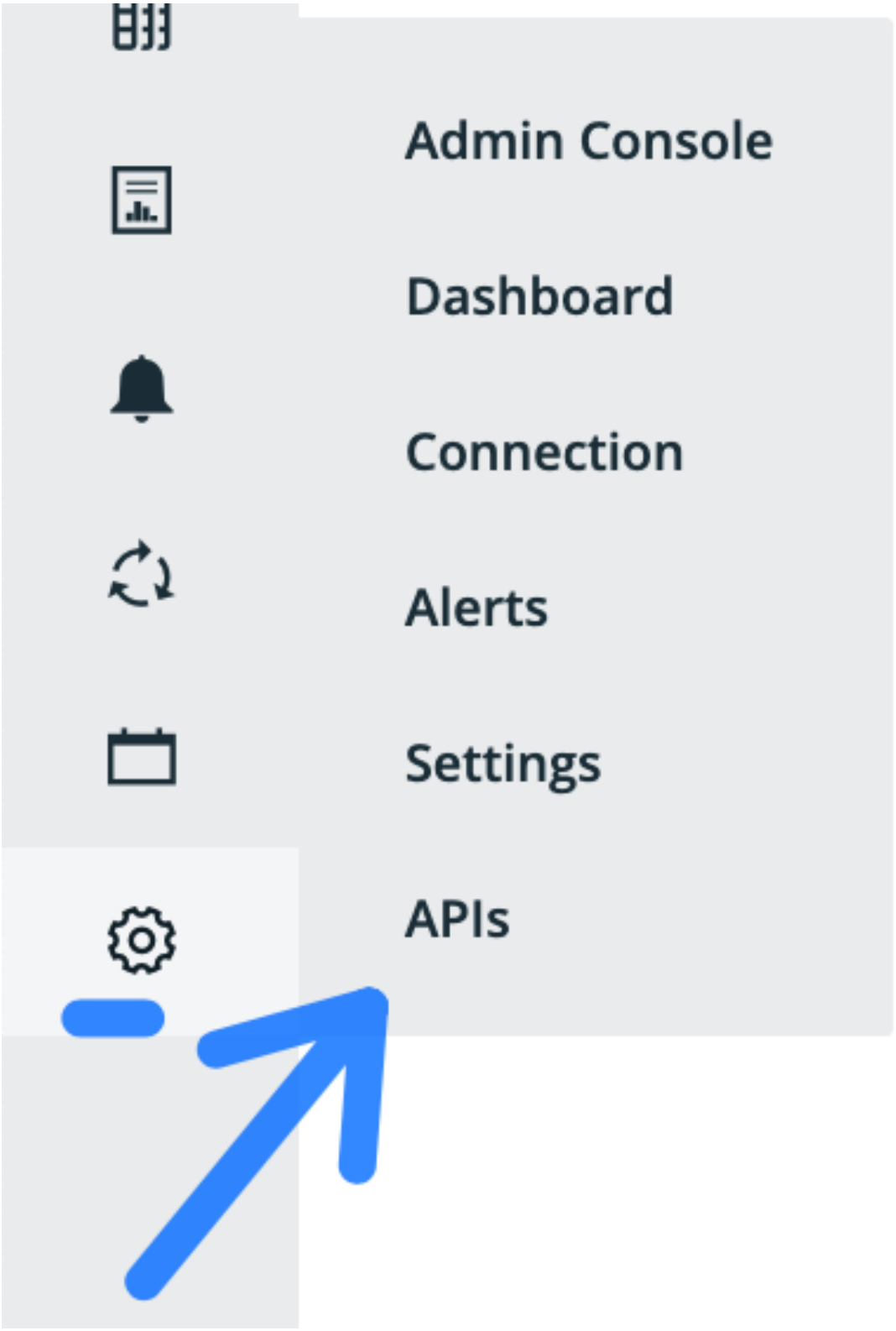
You can find Swagger in the Collibra DQ application.

Docs built into the application

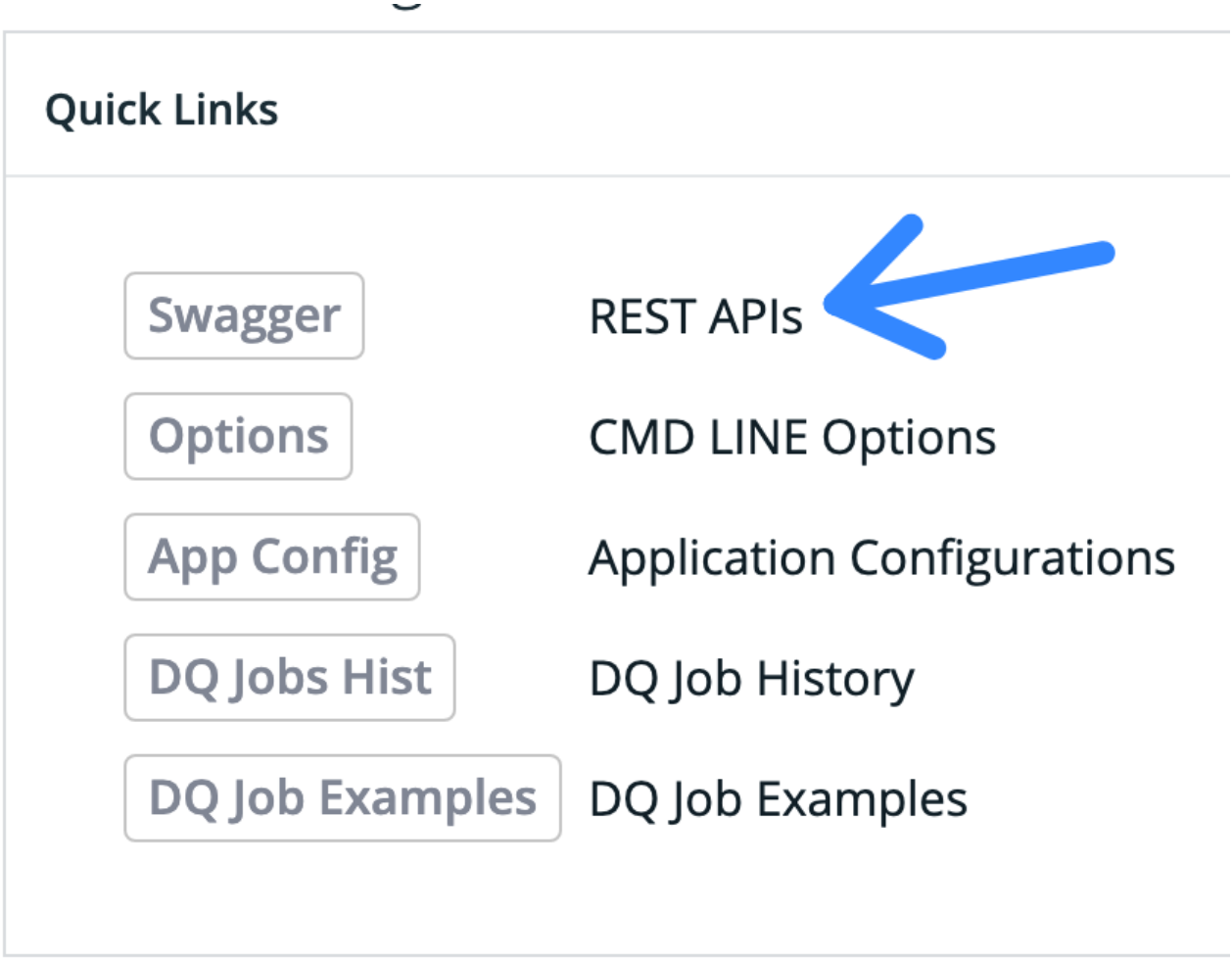
Collibra DQ comes with full swagger support out of the box.

```
http://<YOUR_IP_ADDRESS>/swagger-ui.html
```

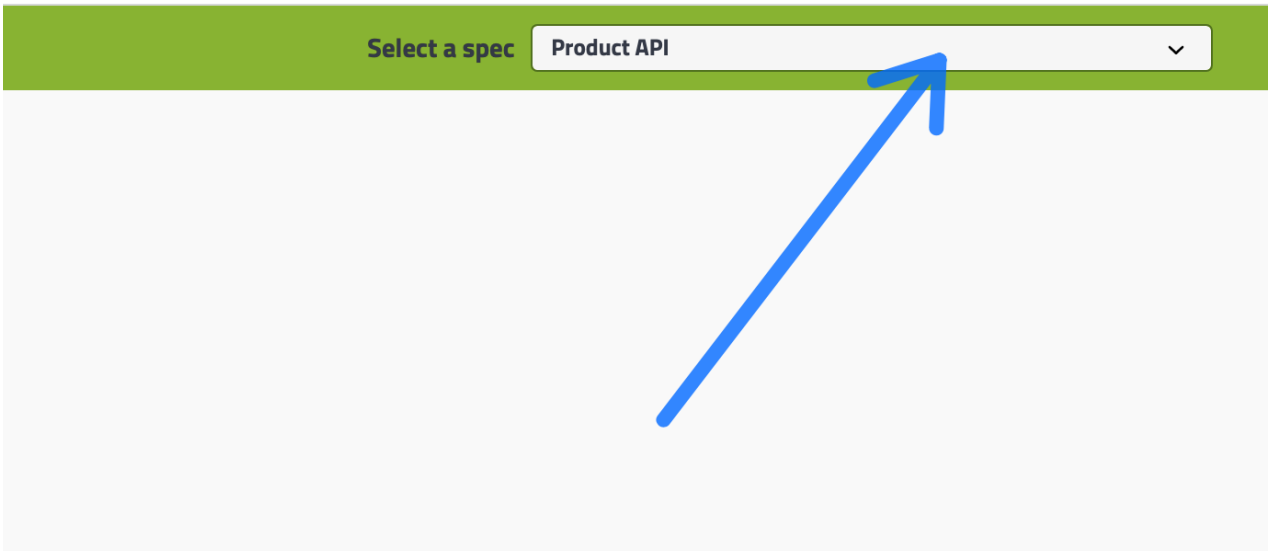
Swagger can be found in the application under the Admin section labeled APIs.



You will find a direct link to the Swagger page



Toggle between Product API and Internal API.



For example swagger API please visit - http://<YOUR_IP>:9000/v2/api-docs?group=UI Internal.

swagger

Api Documentation 1.0

[Base URL: 18.232.221.162/]
http://18.232.221.162/v2/api-docs

Api Documentation

Terms of service

Apache 2.0

basic-error-controller Basic Error Controller

controller Controller

controller-agent Controller Agent

controller-alert Controller Alert

controller-graph Controller Graph

controller-health Controller Health

controller-label Controller Label

controller-owl-check Controller Owl Check

controller-rule Controller Rule

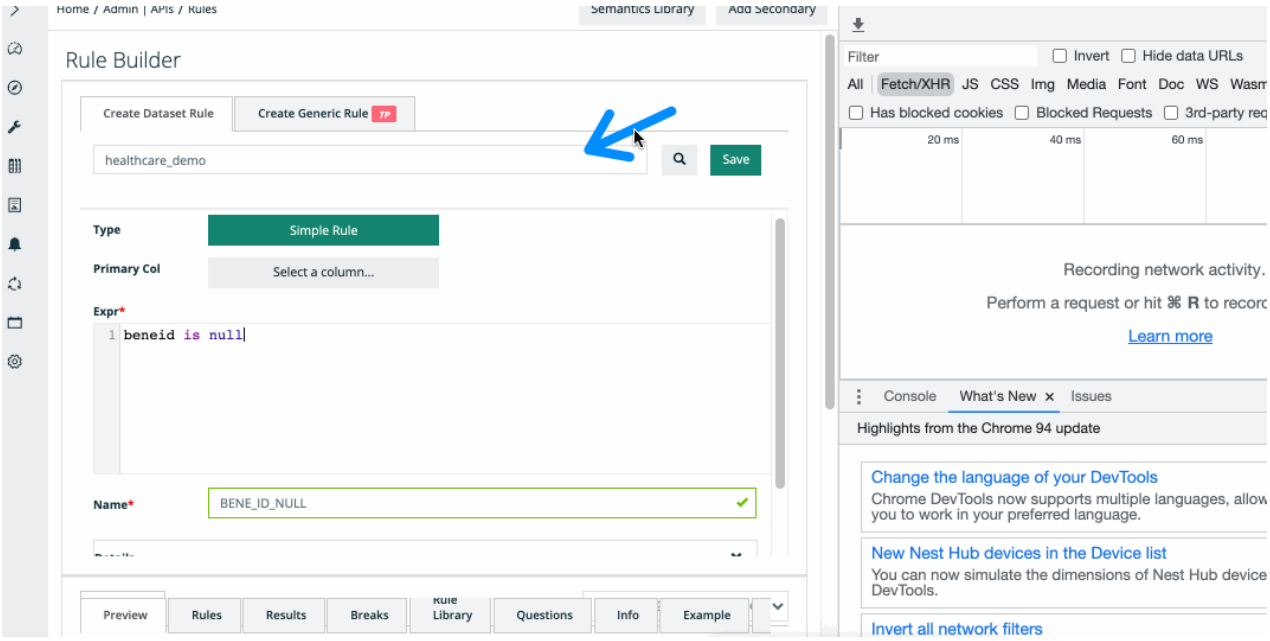
Find the endpoint

Find an API Endpoint

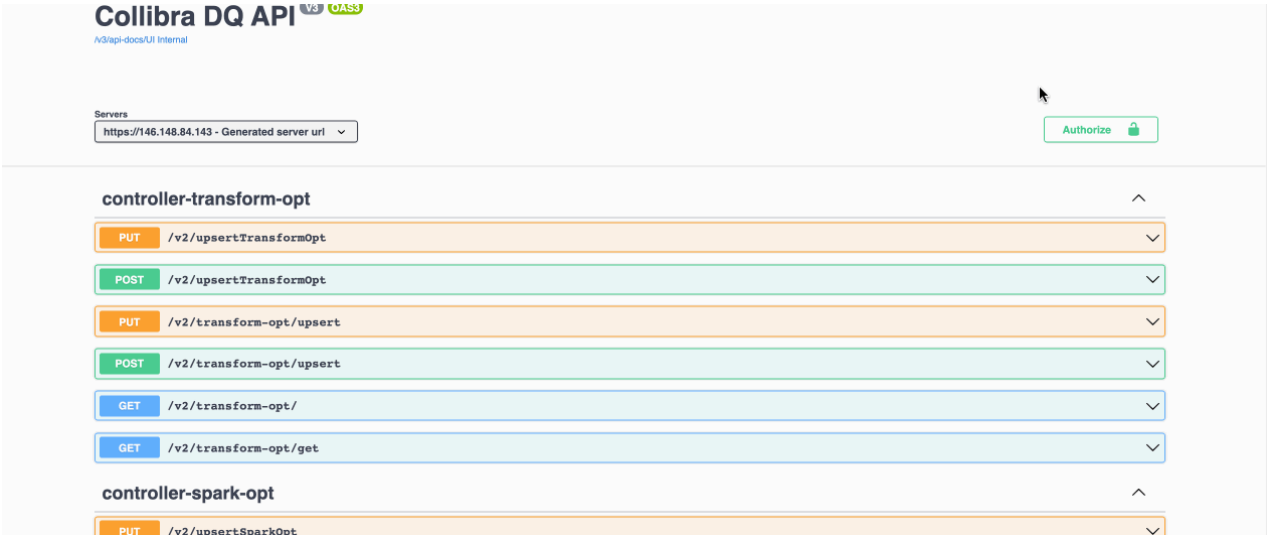
Any front-end action uses the API. You can find the corresponding endpoint using developer tools.

In this example, we will look at the api call for `/create rule`.

Locate the call in Developer Tools



Locate the API in Swagger



All UI endpoints are the API and can be located in swagger. You can script against this externally as well.

Export and Import API

Promoting and moving datasets across environments.

Pre-requirements

> Warning The database needs the stored procedure (function) defined in order to use the Export/Import API.

V2 - Stable - available from 2022.02 release

Step 1a - Export content from source schema

GET https://<collibra-dq-url>/v2/**db-export**

Exports tables from database

Parameters

Query

`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**
Responses		
● **200: OK**		List of SQL - INSERT statements as JSON list.

```
{
    // Response
}
```

● **400: Bad Request** Any error happened with error message

```
{
    // Response
}
```

Step 1b - Import content

POST https://<collibra-dq-url>/v2/**db-import**

Import content into the target tenant

The target schema/tenant name will be part of the input SQL INSERT statements.

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.

Parameters

Body

*	List of SQL INSERT, which will be imported into the target Collibra DQ [[[Undefined variable DataQuality.DQ Metastore]]]. Format: JSON string list
Responses	
200: OK	When the import was successful.

```
{
    // Response
}
```

400: Bad Request Any error happened with error message

```
{
    // Response
}
```

We suggest using db-export, but we will not remove get-exports. We do expect to consolidate the newer logic behind the method.

Step 1c - Get-Exports

You can pass in several dataset names and several tables at once. This endpoint will create a JSON payload

> Note Exports and Imports are currently limited to the 3 tables listed below.

These are the three most common tables. These are the supported tables for re-promotion (running the export multiple times). The most common use case is to copy jobs and rules from environment A to environment B. Running the export/import

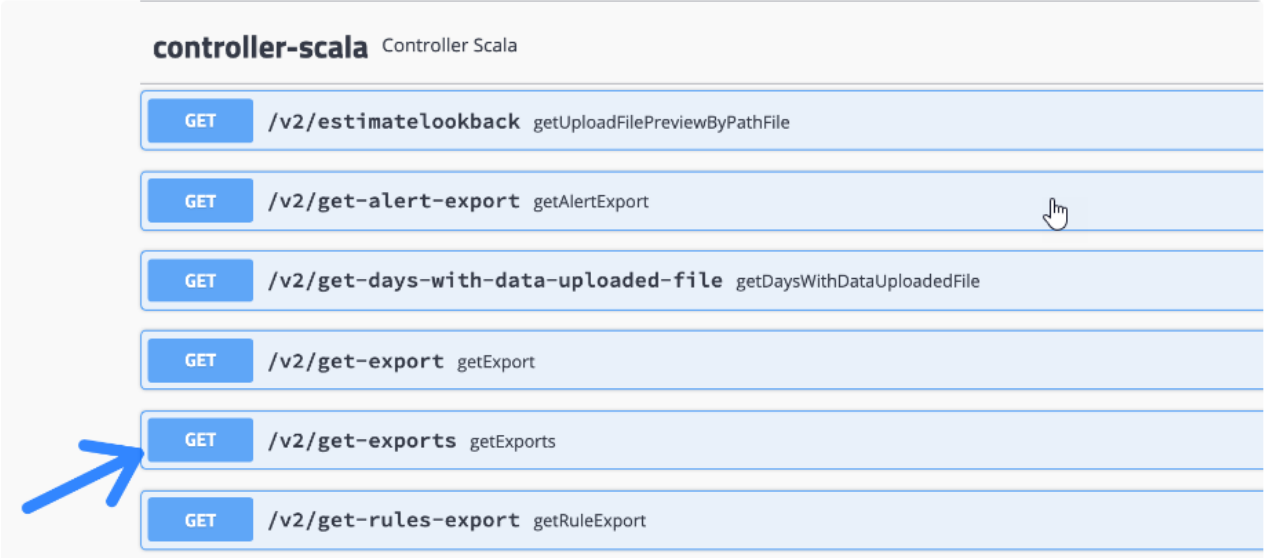
sequence on the same environment likely result in a key constraint conflict, unless in-between edits are made to the insert payload.

- owl_rule
- job_schedule
- owl_check_repo

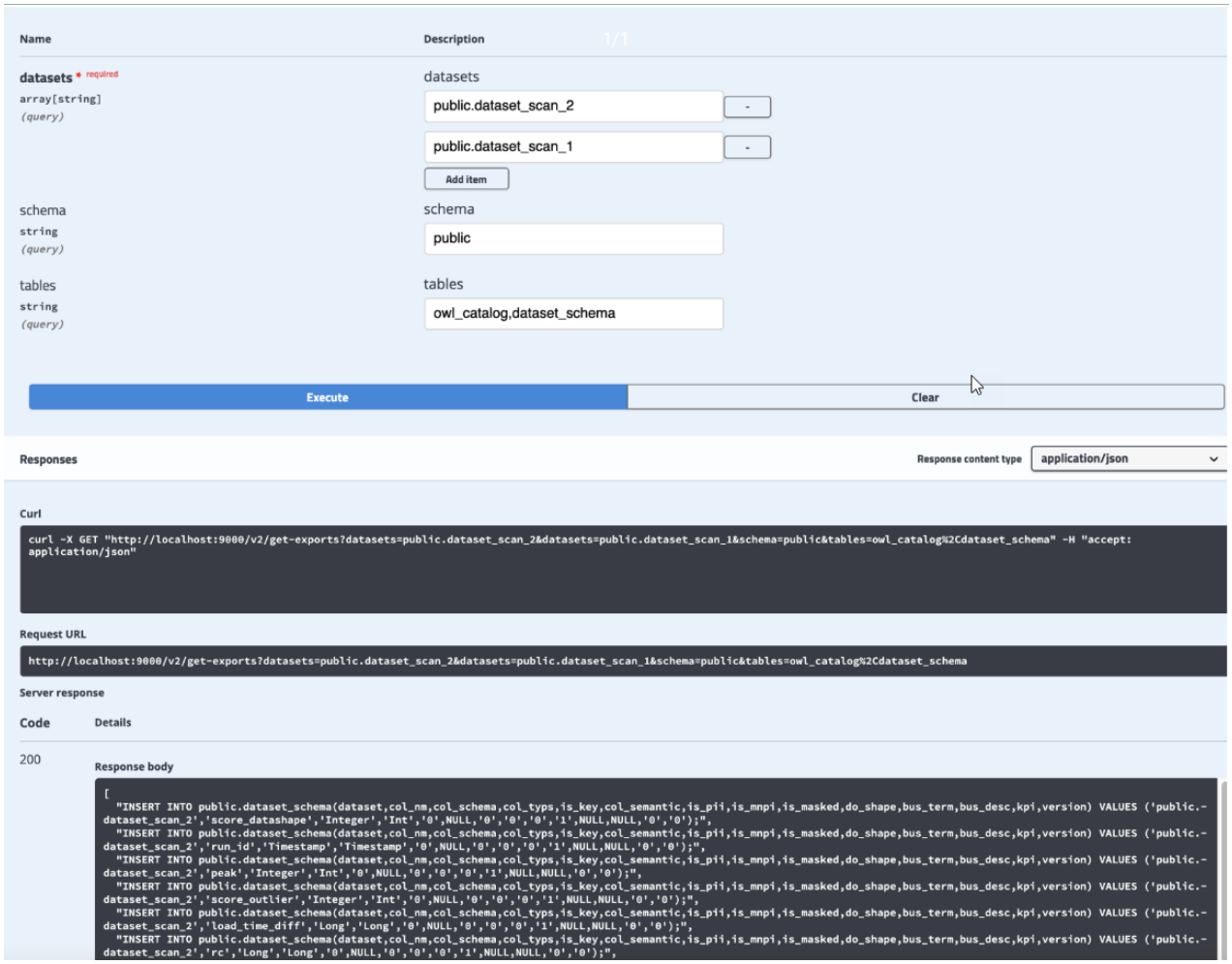
```
http://<url>/v2/get-exports?dataset=public.dataset_scan_2,pub-
lic.dataset_scan_1&schema=public&tables=owl_rule,job_sched-
ule,owl_check_repo
```

Use Swagger to build this for you

This is located under controller-scala (internal API)



Click Try it out to input the details



Step 2 - Run-Import

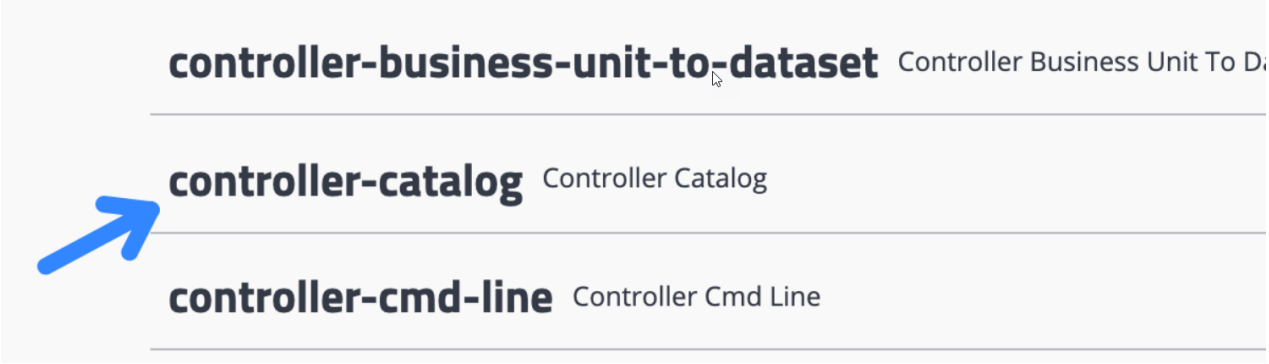
> Note You will want to perform a find/replace on the import payload to check for differences in connections, agents, spark and environment configurations. Migrating to different environments typically requires the payload to be modified.

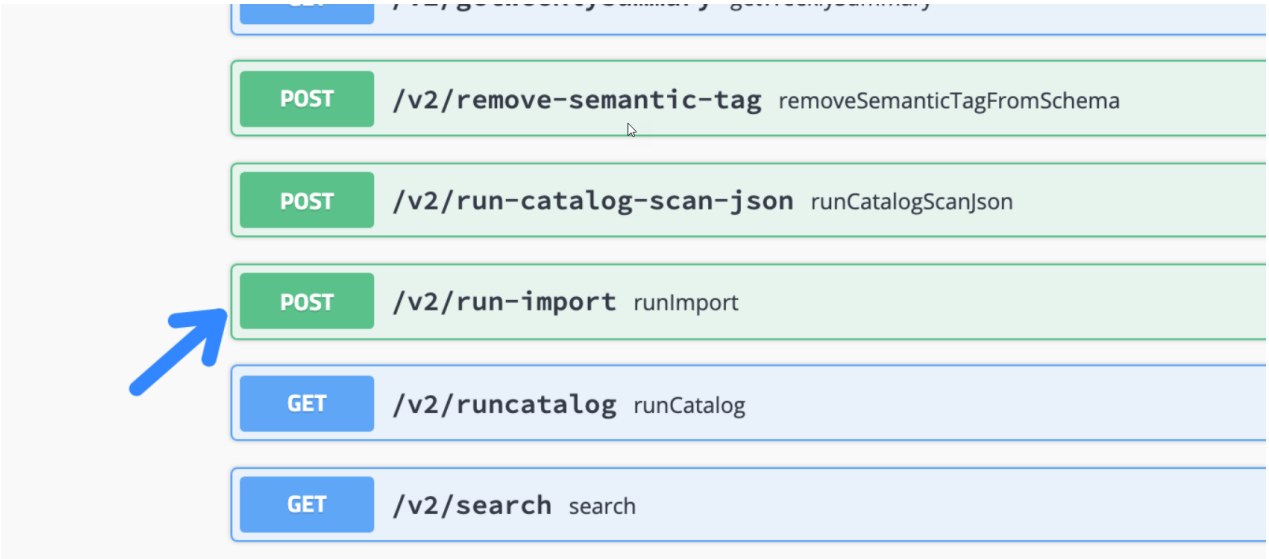
Run import on the desired environment, passing the output of the previous statement to the body of the request.

```
http://<url>/v2/run-import
```

Use Swagger to try it out

This is under controller-catalog.





This would be the body of the POST.

```
JSON print
2   "INSERT INTO public.owl_catalog(dataset,alias,host,source,db_nm,table_nm
      ,mon,tue,wed,thu,fri,sat,sun,time_zone,t1,t2,t3,t4,semantic,runs
      ,source_name,run_mode,data_concept_id) VALUES ('public.dataset_scan_2'
      ,'public.dataset_scan_2','localhost','Postgres','public'
      ,'dataset_scan','1','1','1','1','1','1','1','UTC',NULL,NULL,NULL,NULL
      ,'1','4',NULL,'draft',NULL);",
3   "INSERT INTO public.dataset_scan(dataset,run_id,rc,pass_fail,peak,score
      ,score_rule,score_datashape,score_outlier,score_pattern,score_source
      ,score_behavior,score_dupe,score_record,score_schema,load_time_diff
      ,updt_ts,user_label,rc_src,pass_fail_rc,push_down_options) VALUES
      ('public.dataset_scan_2','2021-04-25 00:00:00-04','11','1','1','100'
      ,'0','0','0','0','0','0','0','0','0',NULL,'2021-04-25 21:03:20.024384'
      ,NULL,NULL,'1',NULL);",
4   "INSERT INTO public.owl_check_repo(dataset,cmd_line_t,cmd_line_v,rd,q,f
      ,v0,v1) VALUES ('public.dataset_scan_2','-bhtimeoff -owluser admin
      -lib \"/opt/owl/drivers/postgres/\" -h localhost:5432/postgres
      -drivermemory 2g -master local[*] -ds public.dataset_scan_2
      -deploymode client -q \"select * from public.dataset_scan where
      updt_ts >= ''${rd}'' AND updt_ts < ''${rdEnd}''\" -bhlb 10 -rd ${rd}
      -driver \"org.postgresql.Driver\" -bhminoff -loglevel INFO -cxn
      localhost_owl -bhmaxoff -rdEnd ${rdEnd} ',NULL,'2021-04-25',NULL,NULL
      ,'~~~~',NULL);"
5   ]
```

Requirement - Stored Procedure

The following function needs to be created in the Collibra DQ [[[Undefined variable DataQuality.DQ Metastore]]], before this can run.

```
CREATE OR REPLACE FUNCTION public.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
 $function$
;
```

This assignment needs to be added.

```
alter function dump(text, text, text) owner to <ownername>;
```

Export and Import Example

> Note Best practice is to use get-exports and the owl_rule table post 2021.09 release. Please refer to the Export and Import API page for more details.

Steps

1. Find your dataset.
2. Pass your table to the following api call - http://<url>/v2/get-rules-export?data-set=public.transit_6.
3. Run import on the desired environment, passing the output of the previous statement to the body of the request - http://<url>/v2/run-import.

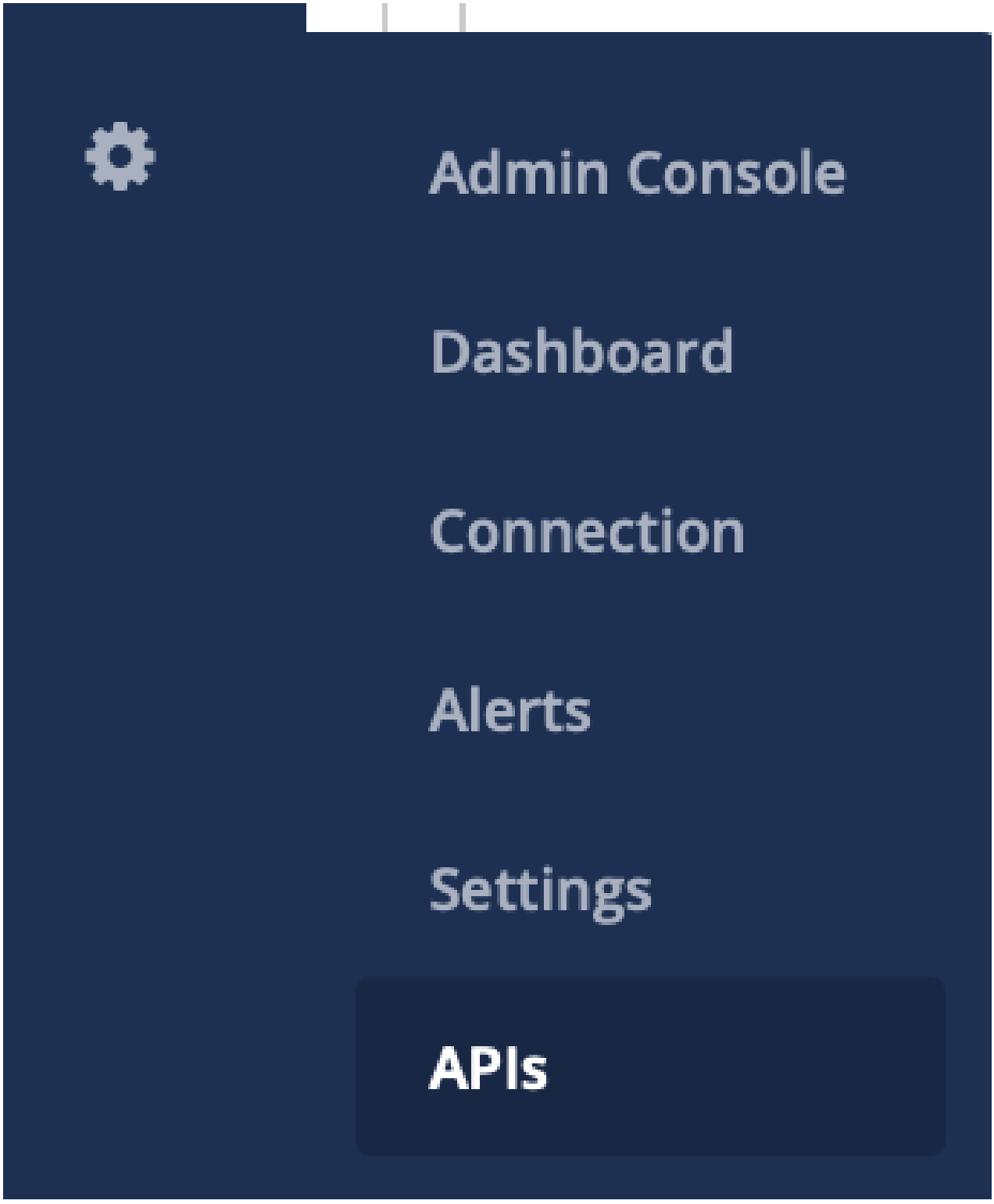
The following function needs to be declared in the postgres metastore before this can run.

```
CREATE OR REPLACE FUNCTION public.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 --
     -- ------ --

     -- ----------- --
```

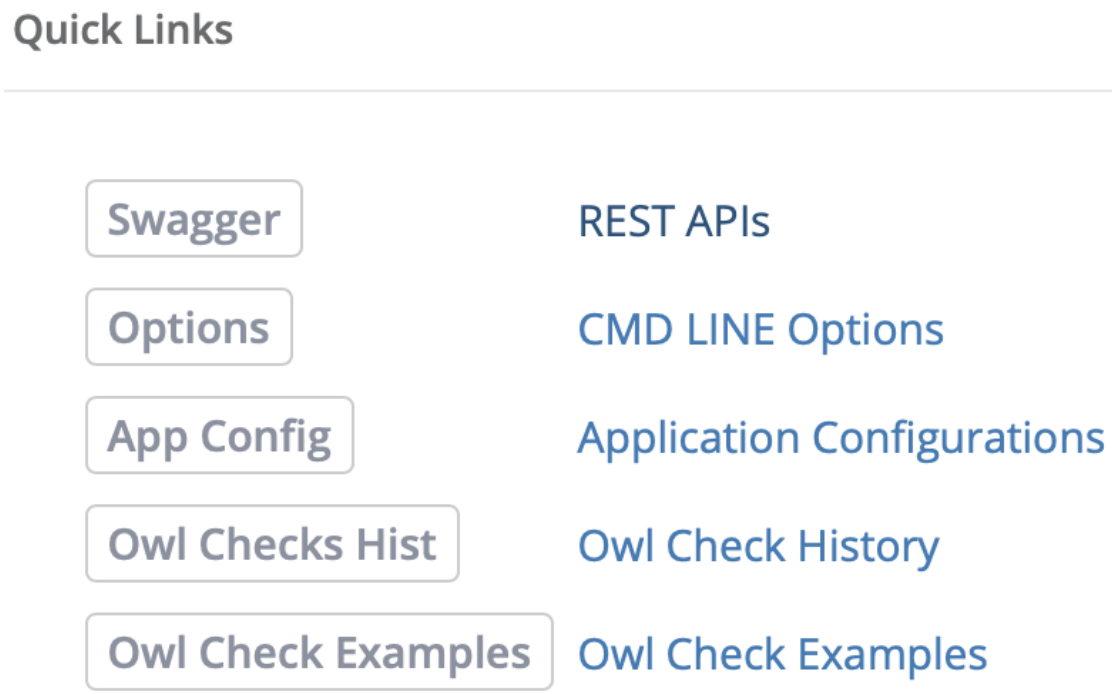
```
      -- SELECT LOOP --
      --   execute SELECT built and loop on each row
      <<label1>>
      FOR valrec IN  EXECUTE  selquery
      LOOP
          dumpquery_1 := '';
          IF not found THEN
              EXIT ;
          END IF;

          -- ----------- --
          -- 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
 $function$
;
```

From Swagger

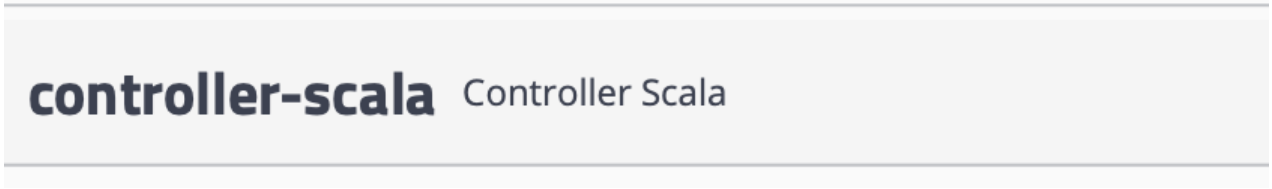
Navigate to the API page.



Find the Rest APIs link.

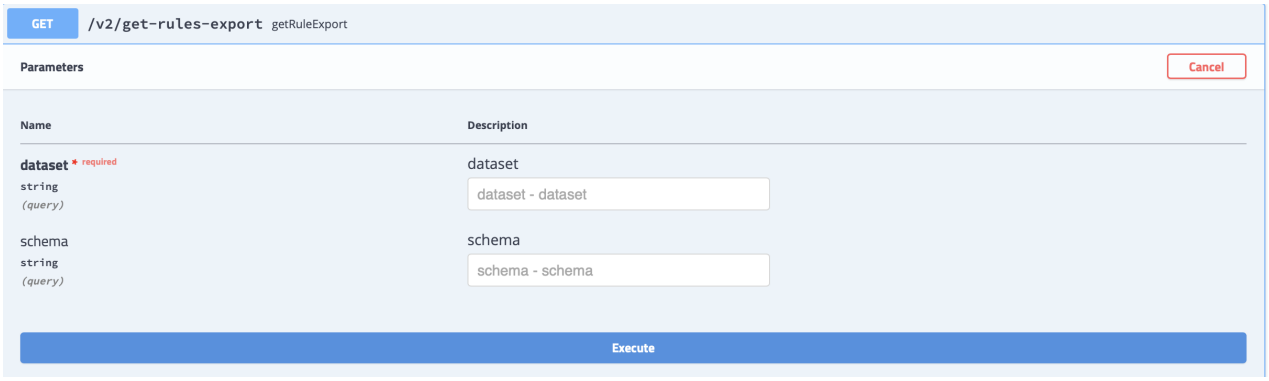


Drill-in to the controller-scala section.

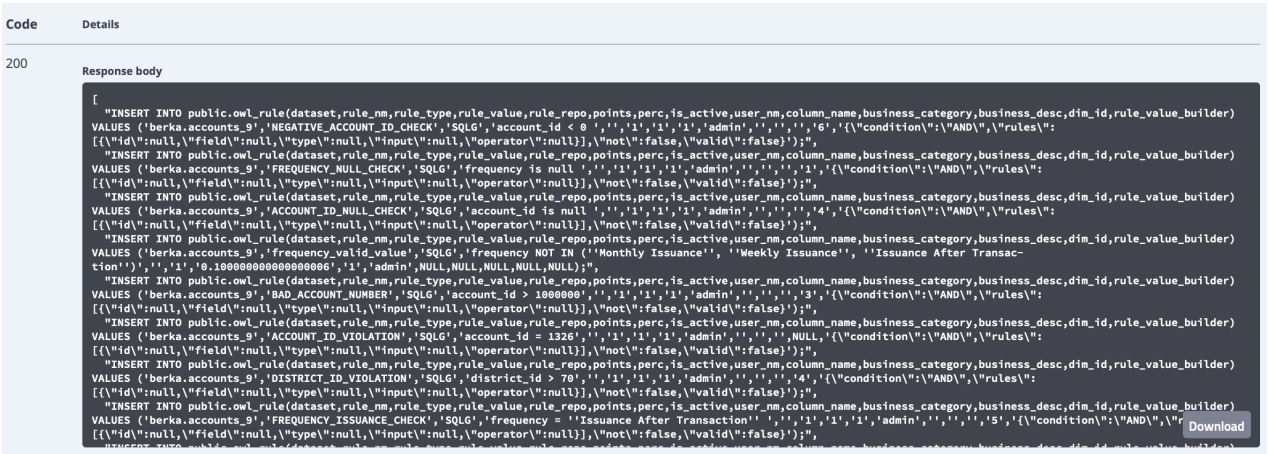


Find the get-rules-export call.

Click Try it out and enter a data set name, Execute to run the call.

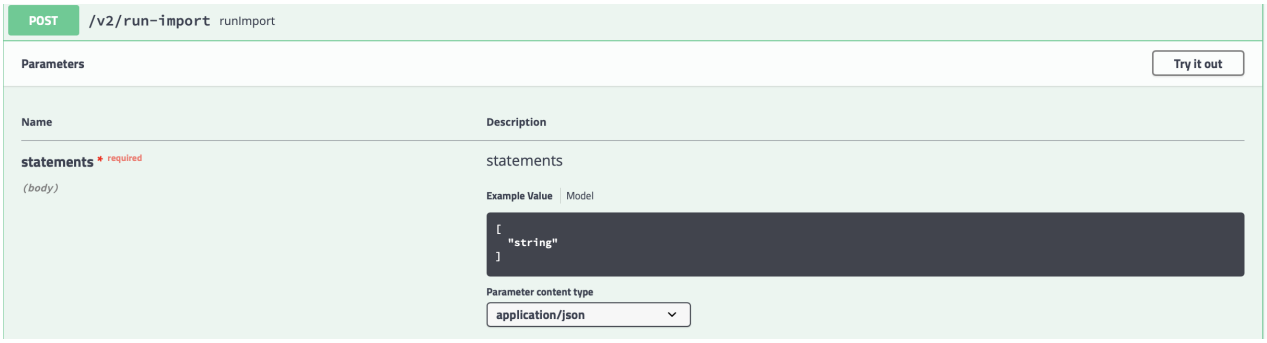


Copy the response body.

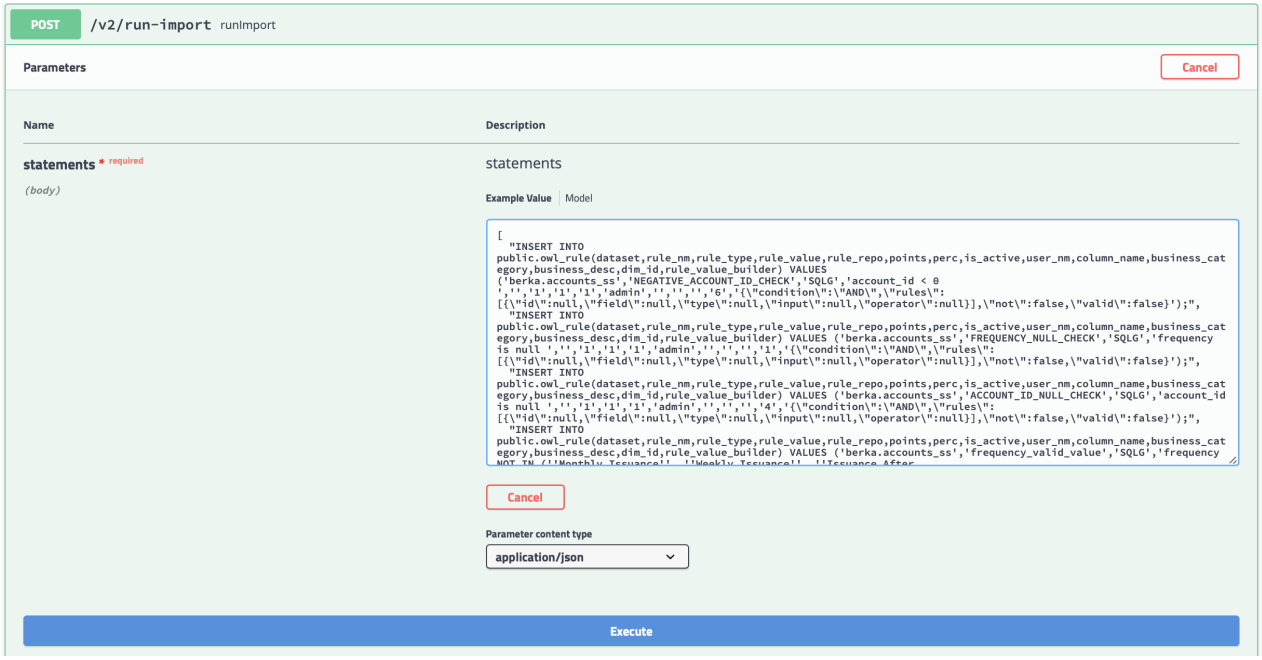


Navigate to the controller-catalog section.

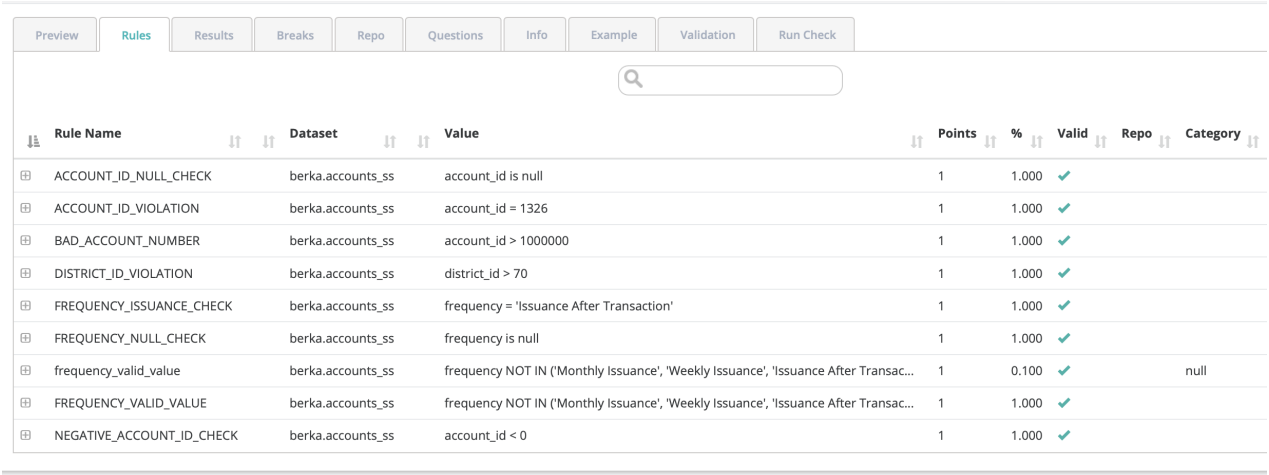
Find run-import and Try it out.



Make any edits and paste in the response body from the previous step.



Visually validate the rules were transferred to another dataset successfully.



Rule Name	Dataset	Value	Points	%	Valid	Repo	Category
ACCOUNT_ID_NULL_CHECK	berka.accounts_ss	account_id is null	1	1.000	✔		
ACCOUNT_ID_VIOLATION	berka.accounts_ss	account_id = 1326	1	1.000	✔		
BAD_ACCOUNT_NUMBER	berka.accounts_ss	account_id > 1000000	1	1.000	✔		
DISTRICT_ID_VIOLATION	berka.accounts_ss	district_id > 70	1	1.000	✔		
FREQUENCY_ISSUANCE_CHECK	berka.accounts_ss	frequency = 'Issuance After Transaction'	1	1.000	✔		
FREQUENCY_NULL_CHECK	berka.accounts_ss	frequency is null	1	1.000	✔		
frequency_valid_value	berka.accounts_ss	frequency NOT IN ('Monthly Issuance', 'Weekly Issuance', 'Issuance After Transac...	1	0.100	✔		null
FREQUENCY_VALID_VALUE	berka.accounts_ss	frequency NOT IN ('Monthly Issuance', 'Weekly Issuance', 'Issuance After Transac...	1	1.000	✔		
NEGATIVE_ACCOUNT_ID_CHECK	berka.accounts_ss	account_id < 0	1	1.000	✔		

Integration API

The Integration API lets you send Collibra DQ rules, ML layer findings, and associated data quality scores to Collibra Data Intelligence Cloud without a dependency on Edge. When you configure the integration, your data quality metrics and scores update in real-time within Collibra Data Intelligence Cloud, empowering you with a robust view of the health of your organization's data.

Prerequisites

- Your Collibra Data Intelligence Cloud profile must have Sysadmin and DataSteward global roles to manage all assets and objects of the integration.
- Both Collibra Data Intelligence Cloud and Collibra Data Quality & Observability must be version 2023.05+.

Add connection credentials

POST /v3/dgc/integrations/connections/credentials

Creates a new integration of your Collibra Data Intelligence Cloud instance with Collibra Data Quality & Observability.

Request

Path parameters

Enter the parameters of your Collibra Data Intelligence Cloud 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 Data Intelligence Cloud instance. For example, `https://dg-example.collibra.com/`	✓ Yes
authMode	string	Is always `basic`	✓ Yes
serviceAccount	string	The account name of your Collibra Data Intelligence Cloud instance. For example, `integration.user`	✓ Yes
password	string	The password of your Collibra Data Intelligence Cloud 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 Data Intelligence Cloud instance.

Request

Path parameters

There are no parameters for this request.

Click **Execute** to return your Collibra Data Intelligence Cloud 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 Data Intelligence Cloud instance.
serviceType	string	Data Intelligence Cloud
serviceUrl	string	The URL of your Collibra Data Intelligence Cloud instance.
defaultFields	string	""
authenticationMode	string	The authentication method.
username	string	The username of your Collibra Data Intelligence Cloud 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 Data Intelligence Cloud

`POST` `/v3/dgc/integrations/import/dgcjson`

Adds a new integration of a given Collibra Data Quality & Observability dataset into Collibra Data Intelligence Cloud.

Request

Path parameters

Enter the name of the dataset to integrate with Collibra Data Intelligence Cloud 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 Data Intelligence Cloud instance credentials on and off. If your credentials are set to false, no data flows between Collibra Data Intelligence Cloud and Collibra Data Quality & Observability.

Request

Path parameters

Enter your Collibra Data Intelligence Cloud 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 Data Intelligence Cloud instance Id with which to integrate Collibra Data Quality & Observability.	✓ Yes

Response examples

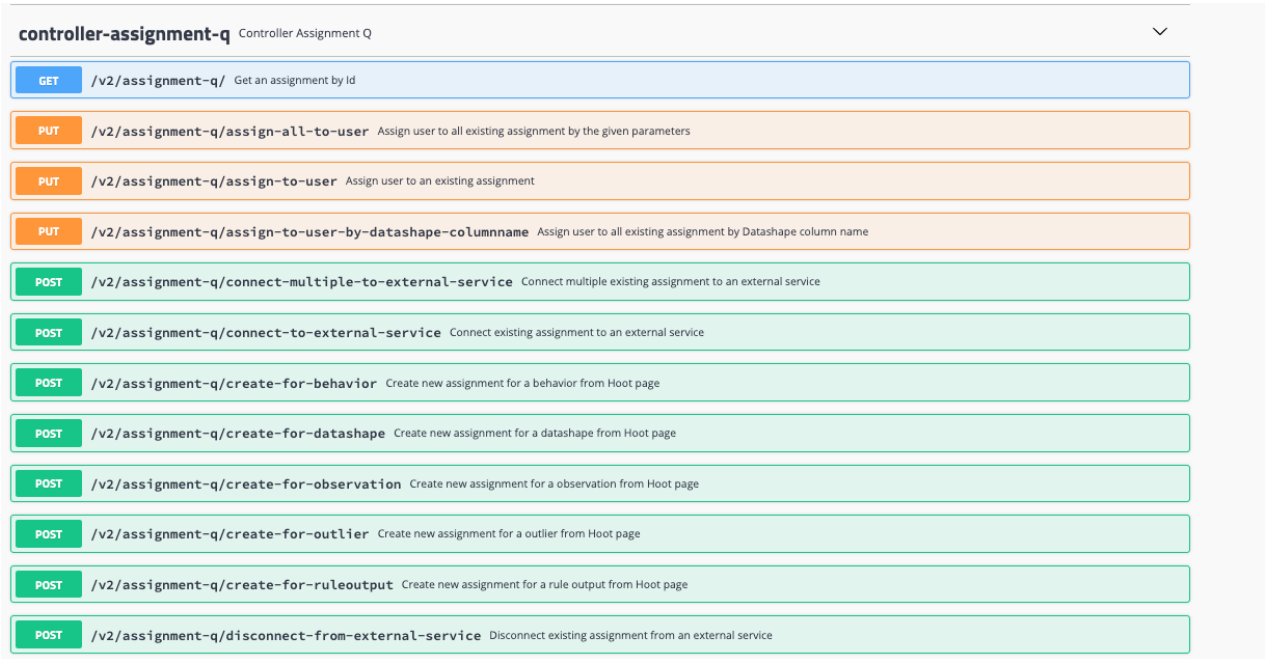
> Note A successful request only returns a response header.

```
 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"
}
```

Assignment API



Time Zone API

UTC + global DateTime standard

Background on common time issues

Controlling dates and times has always been a troublesome topic for global systems. Server clock vs server code such as new Date() which may create a date in the local timezone of the server vs the Browser or clients timezone. Moving to the cloud only makes the problem worse when you need to consider the timezone the server might be in and inherit from it's system clock.

Collibra Data Quality's Solution - Keep it Simple

If everyone worked off of a globally understood format that is not subject to misinterpretation things would be more simple. Take 03/02/2019, for example. Is this March 2nd or February 3rd? That depends on which country you live in. Collibra DQ only accepts this format: YYYY-MM-DD. Extending this to time would mean YYYY-MM-DD 00:00:00.

CmdLine Examples

./owlcheck -ds trades -rd "2019-04-01" or -rd "2019-04-01 00:00:00"

Simple Example

A user running an DQ Check in New York and a user running an DQ Check three hours later in California.

CmdLine Arg	User Location	TimeZone	Stored in Collibra DQ (UTC)
-rd 2019-04-01	New York	implied EST	2019-04-01 04:00:00
-rd 2019-04-01	California	implied PST	2019-04-01 07:00:00

These jobs run three hours apart, even though they appear to run first thing in the morning to each user. Collibra DQ stores all dates in a common **UTC** format for global consistency.

On the Jobs page, the Start Time and Update Time columns are always based on the server time zone of the DQ Web App, and appear in the YYYY-MM-DD 00:00:00 format.

Web API or URL Example

http://localhost:9000/dq/hoot?dataset=atmCustomers&runId=2019-04-01 04:00:00
http://localhost:9000/dq/hoot?dataset=atmCustomers&runId=2019-04-01 07:00:00

For Convenience, if a user prefers seeing and interacting with dates in their local time zone:

http://localhost:9000/dq/hoot?dataset=atmCustomers&runId=2019-04-01 00:00:00&tz=EST http://localhost:9000/dq/hoot?dataset=atmCustomers&runId=2019-04-01 00:00:00&tz=PST

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
```

Livy

What is Livy?

At its core, Apache Livy is an optional component that changes how the **Collibra DQ Web** app caches remote files. From **Explorer**, Livy allows for interaction with Spark clusters

over REST APIs. This is especially useful with larger data or Spark clusters, because with Livy, you can drill into those bigger files.

In a future release, we plan to make Livy a standard component as part of our data visualization, but as of 2022.04 it remains optional. An orange icon shows cached remote files.

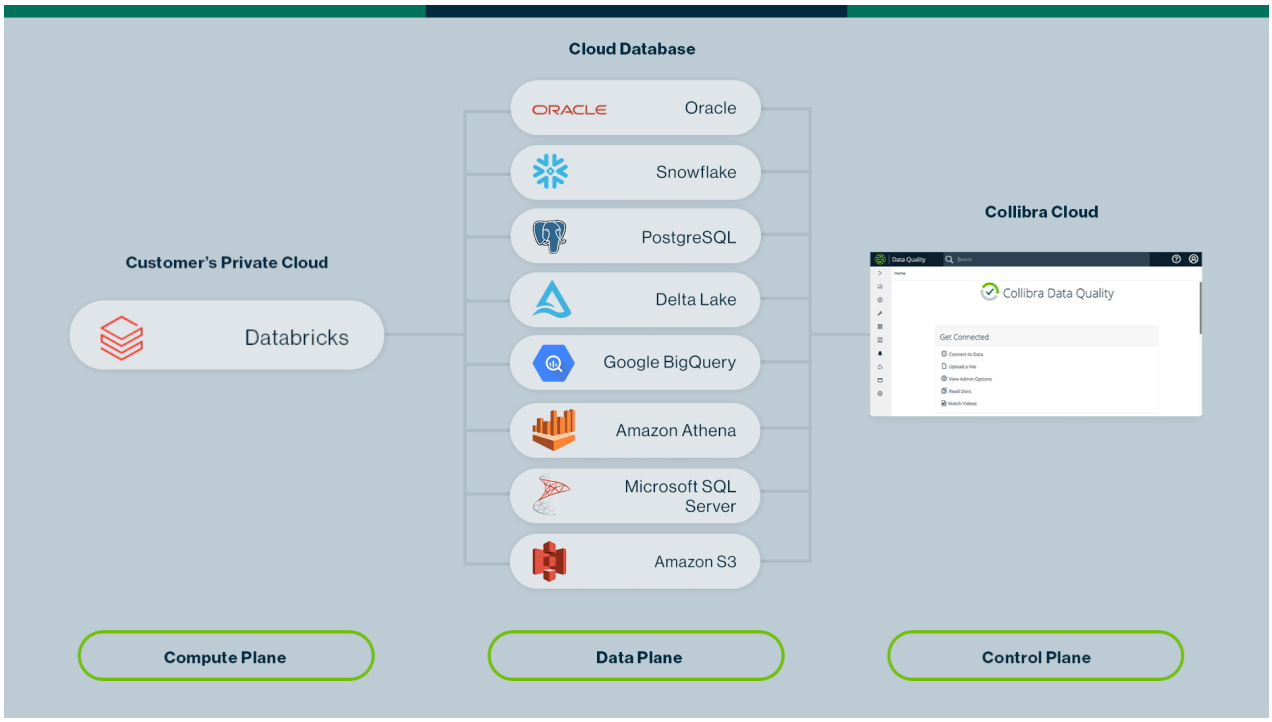
Notebook

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.

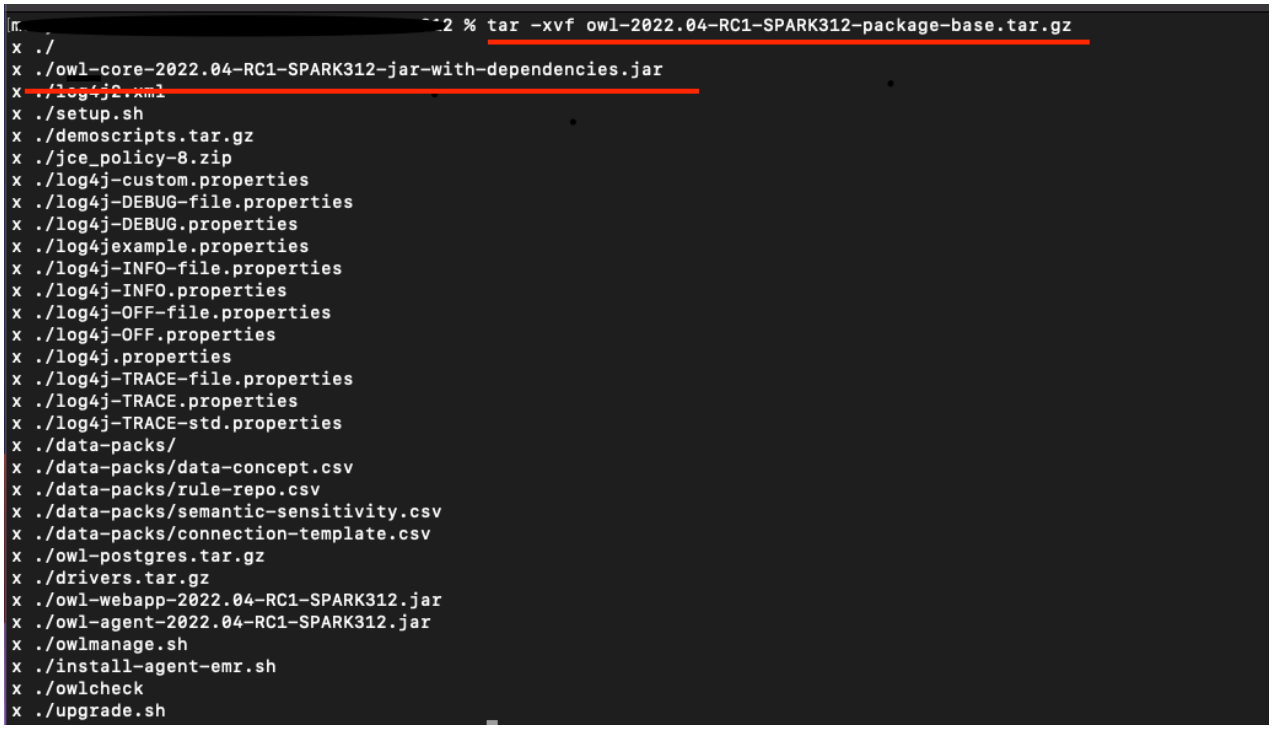
Step 1: Extract the Collibra DQ core jar from Collibra DQ package zipped file.

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`

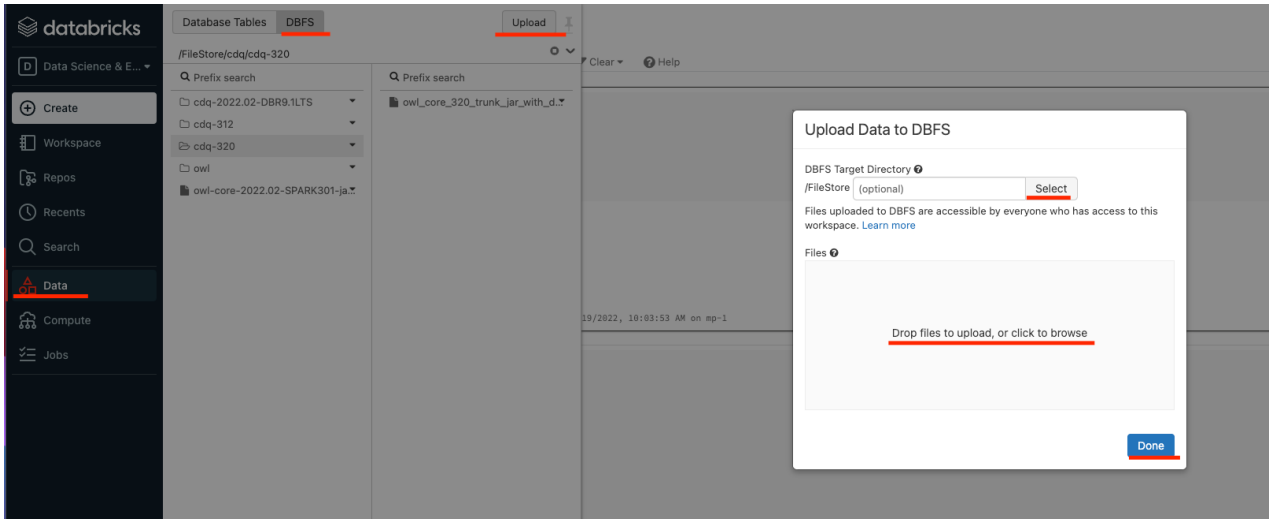
Running this command instructs tar to extract the files from the zipped file. From the list of the files, you need to upload the owl-core-xxxx-jar-with-dependancies.jar to our Databricks file system which will be explained in the next section.



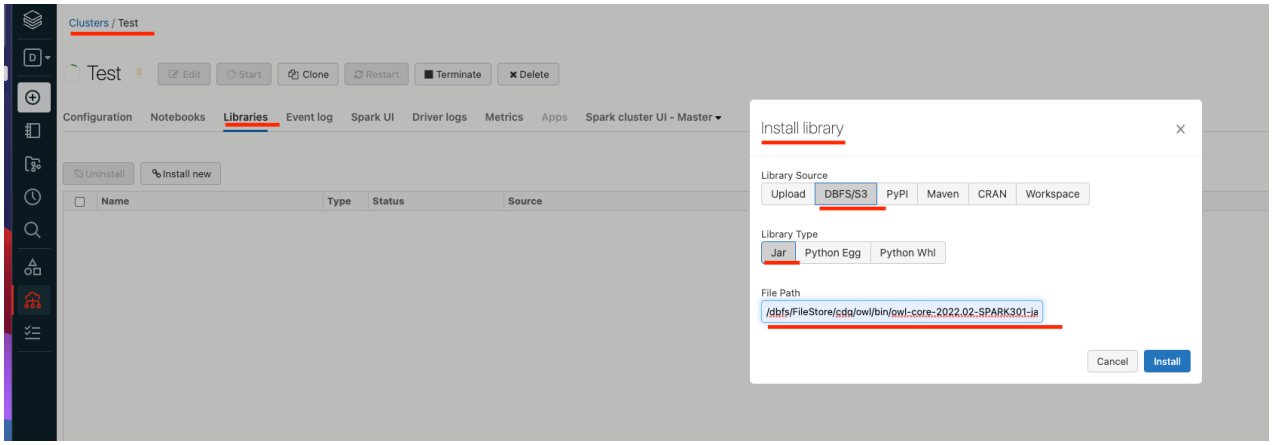
Step 2: Upload the Collibra DQ core jar file to Databricks file system using UI

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

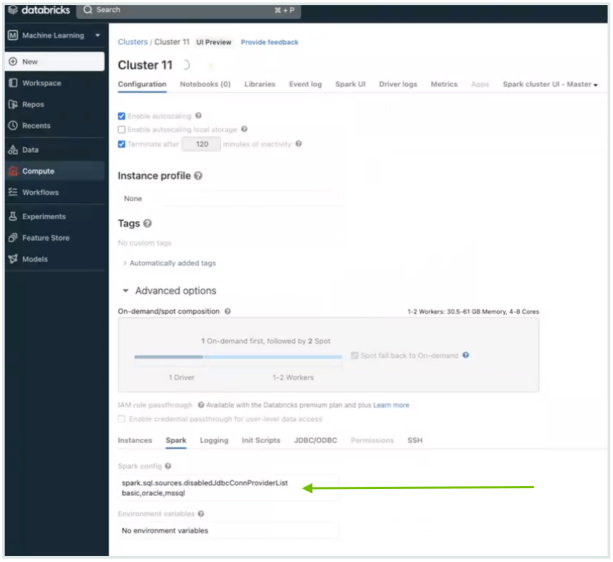
```
spark.sql.sources.disabledJdbcConnProviderList='basic,oracle,mssql'
```

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")
```

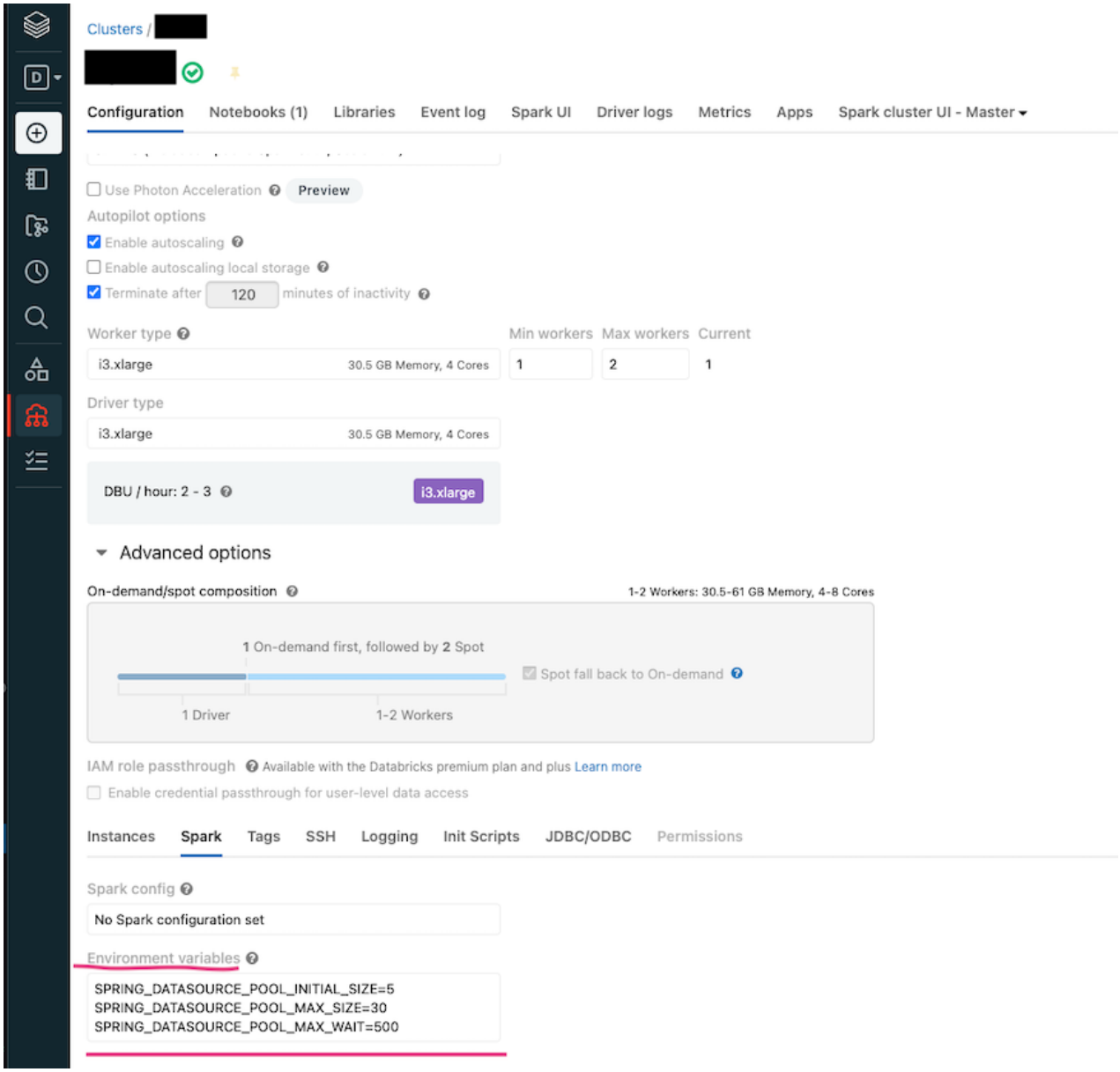
Step 5 (Optional): Update datasource pool size

> 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

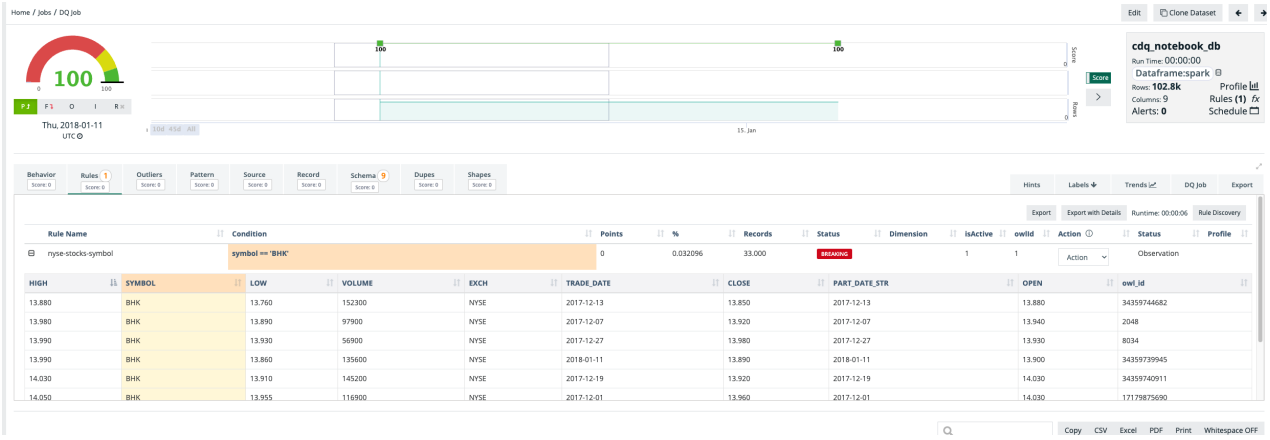
```
// Remove the file if it exists
dbutils.fs.rm("/tmp/databricks-df-example.parquet", true)
breaks.write.parquet("/tmp/databricks-df-example.parquet")
```

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)
 }
```

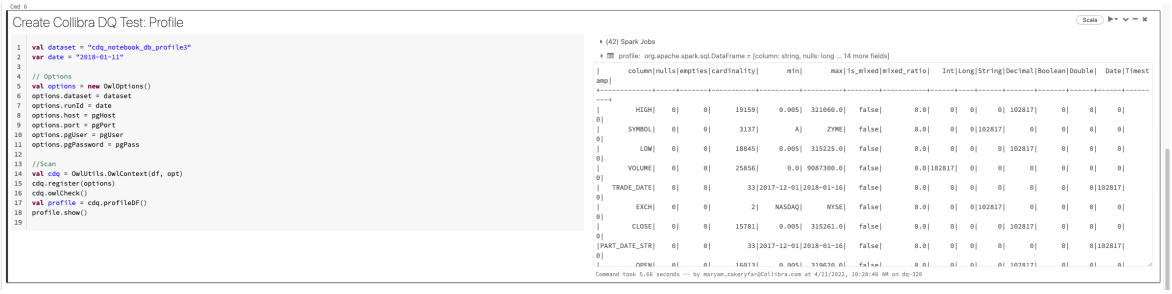
Steps to reassign the rules of one dataset to another via the API.

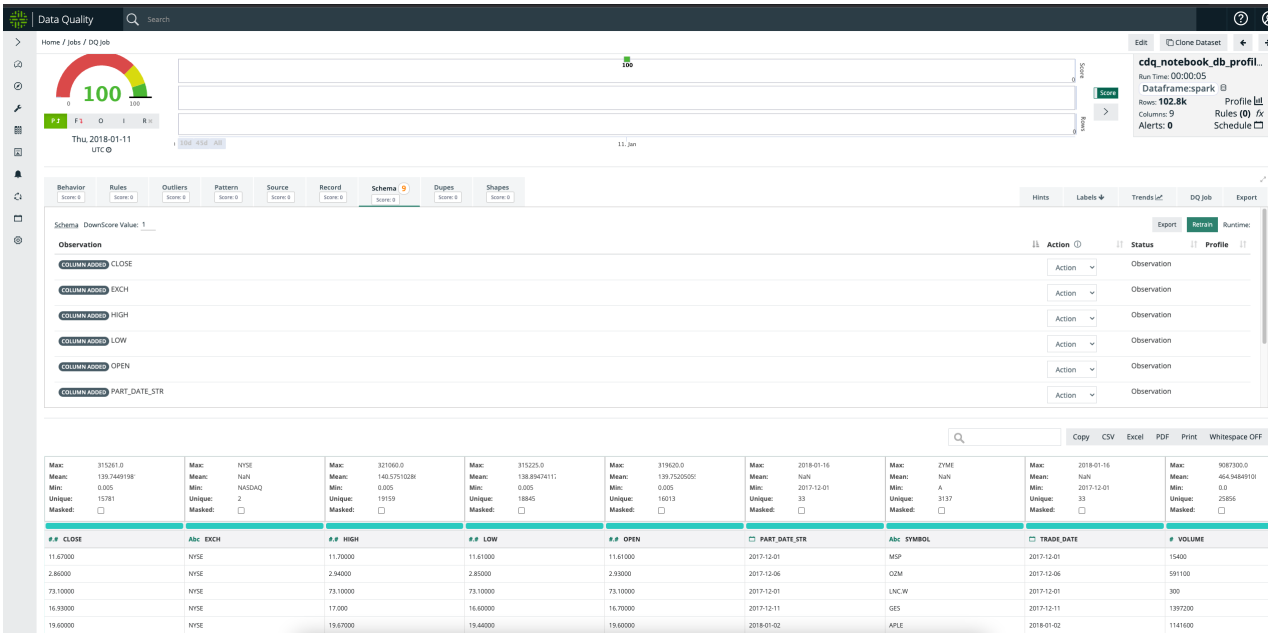
The breaks and the rules can also be viewed in Collibra DQ web.



Create a Collibra DQ Test (Profile)

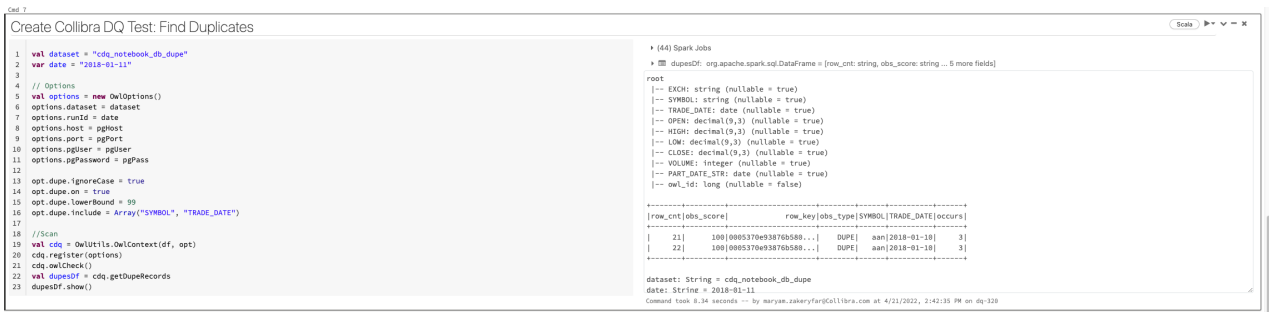
```
val dataset = "cdq_notebook_nyse_profile"

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()
}
```

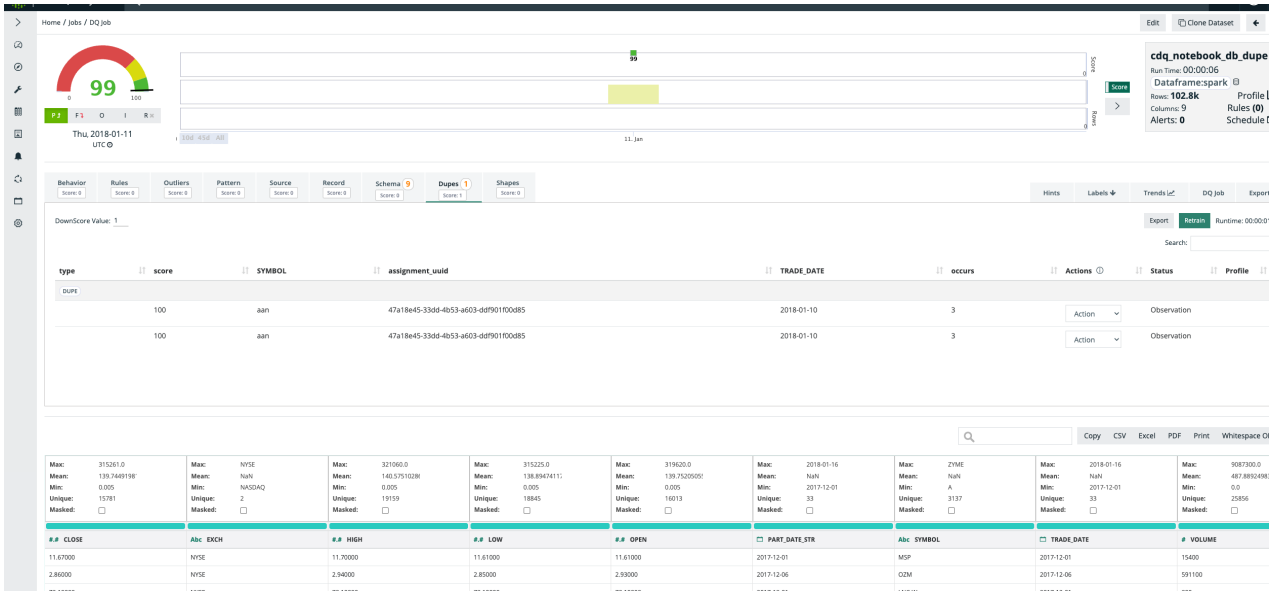

Create Collibra DQ Test: Profile
Scala
val dataset = "cdq_notebook_db_profile3"
var date = "2018-01-11"
// Options
val options = new DwlOptions()
options.dataset = dataset
options.runId = date
options.host = pgHost
options.port = pgPort
options.pgUser = pgUser
options.pgPassword = pgPass
//Scan
val cdq = DwlUtils.DwlContext(df, opt)
cdq.register(options)
cdq.dwlCheck()
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

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.

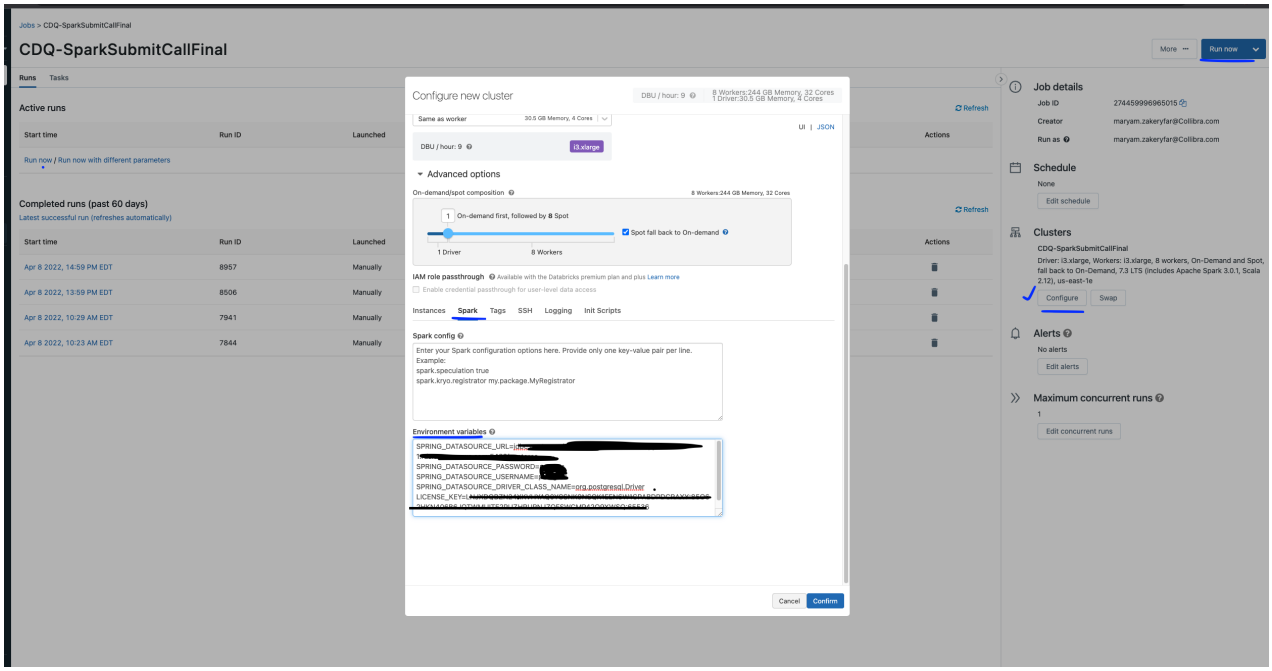
Upload DQ's jars in DBFS

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

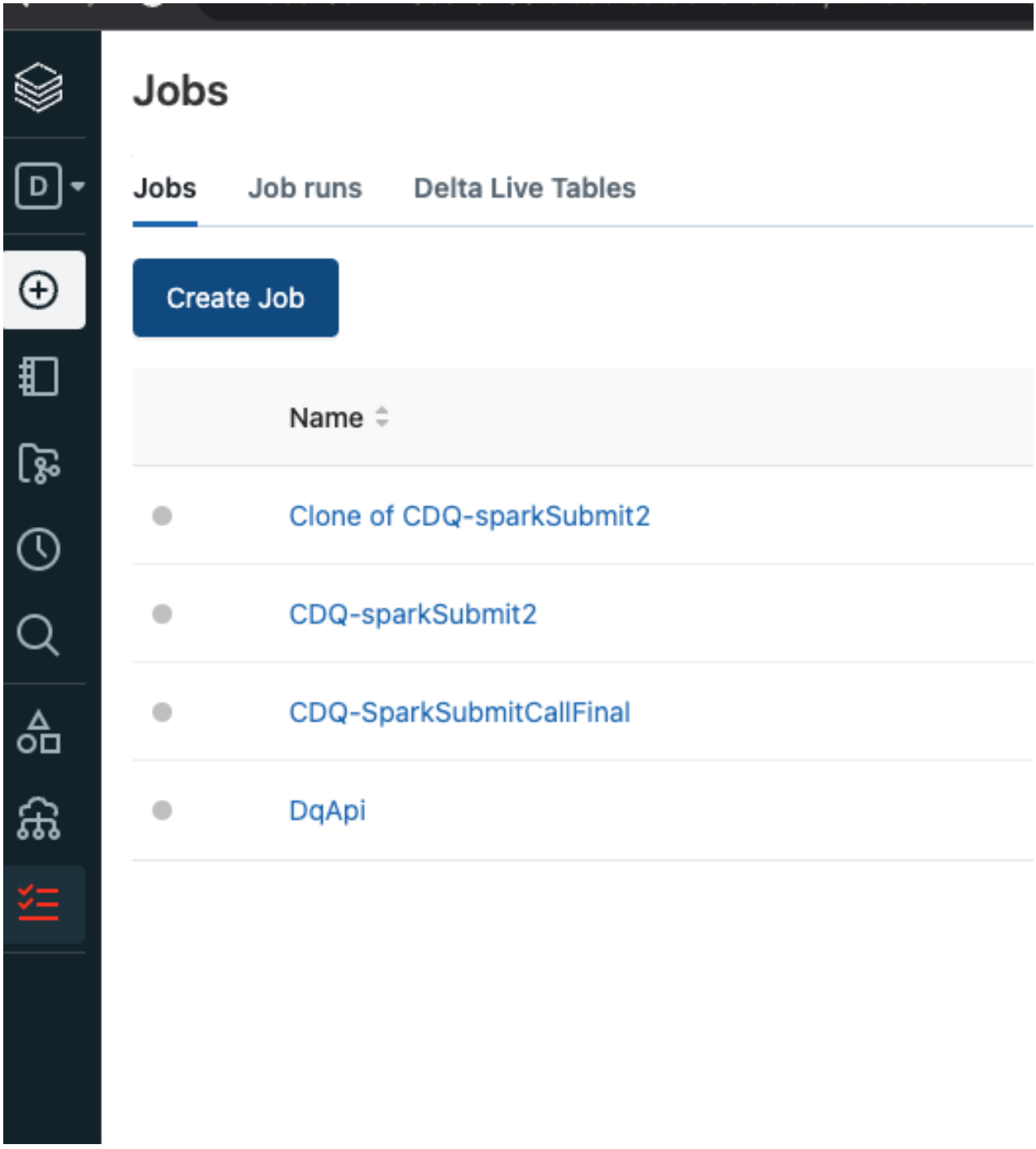
Once the above steps are completed, you can submit a spark submit job with DQ's parameters. Payload parameters can be from DQ's web Run command. You can copy and paste the parameters to prepare a JSON payload. Here is one sample:

```
                    "--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
      }
    ]
}
```

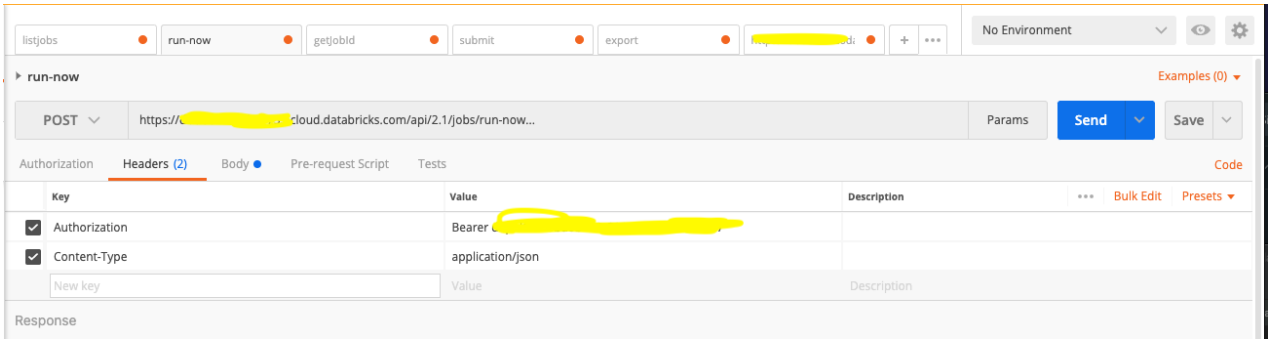
Values to be updated in above payload are:

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. \`\`

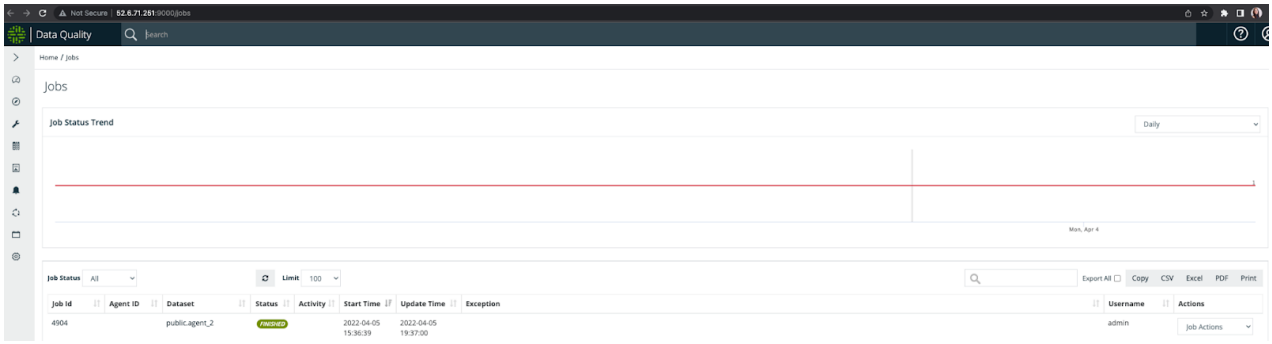
Authenticate the Databricks REST API

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
```

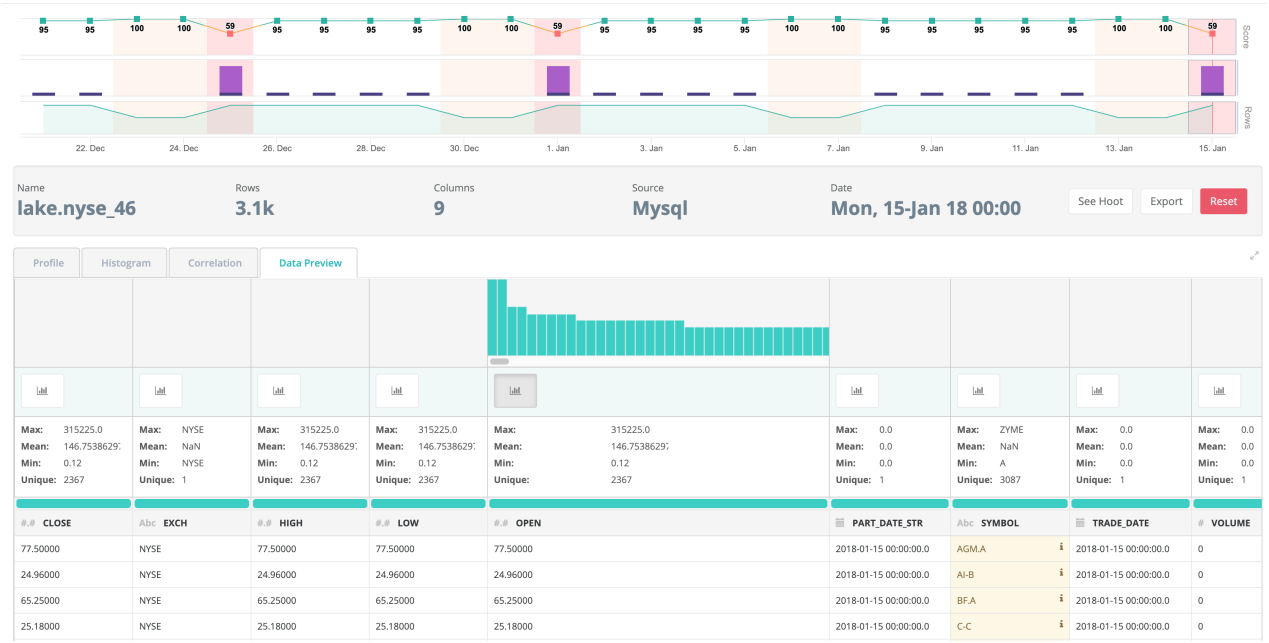
Notice that DQ returns results as Dataframes. This is a fantastic abstraction that allows you to ignore all domain objects and custom types and interact with a scaleable generic result set using common protocols like "where" or "filter" or "save" or "write" all with parallel operations.

```
+--------------+-----+-------+-----------+---+---+--------+-----
------+------+----+------+-------+-------+------+----+---------+
|        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.

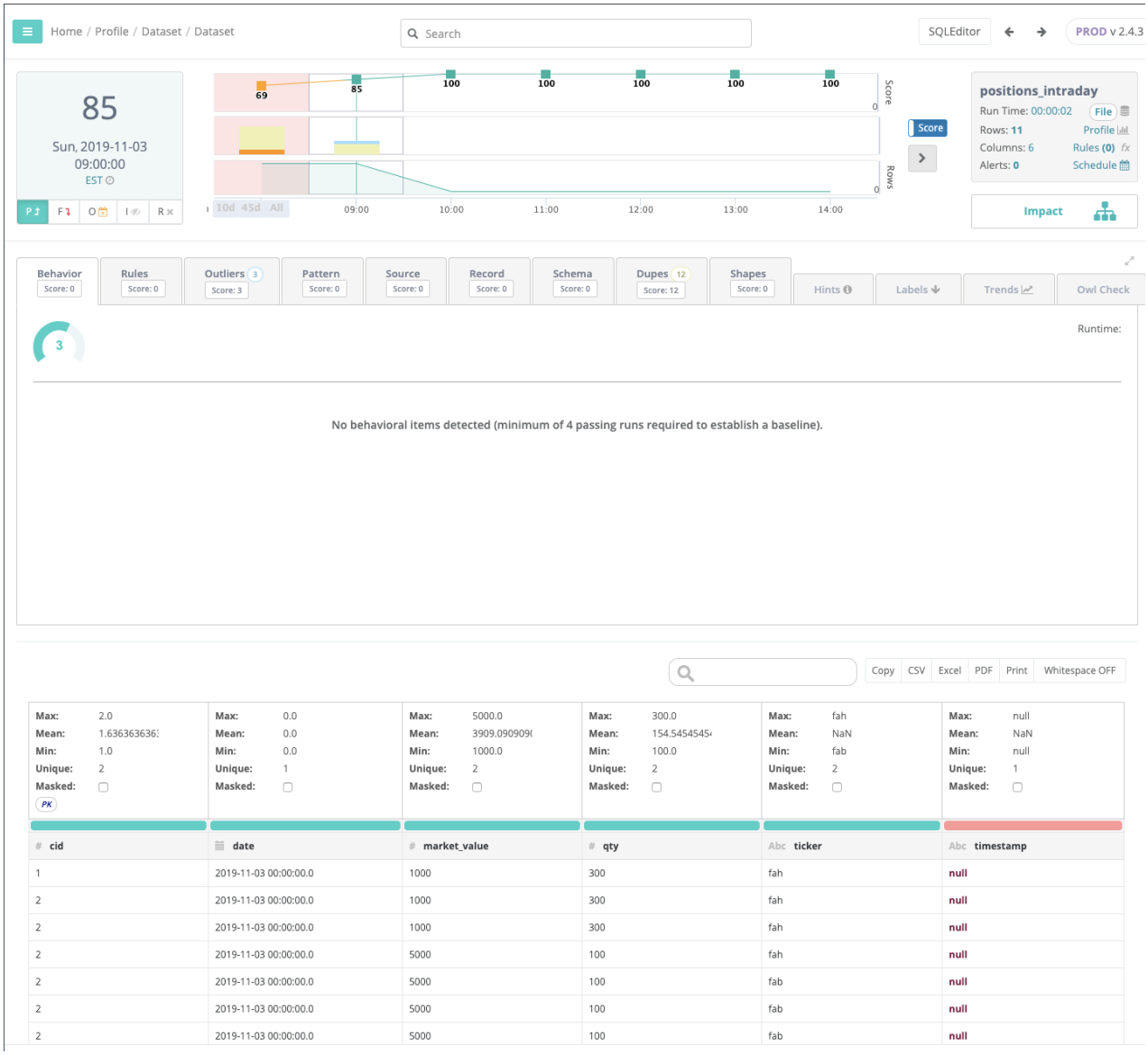
```
val df_source = (spark.read
.format ("csv").option("header", true).option("delimiter", ",")
.load("atmCustFile.csv")
)
val dataset = "atm_customer"
var date = "2018-01-16"

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"
```

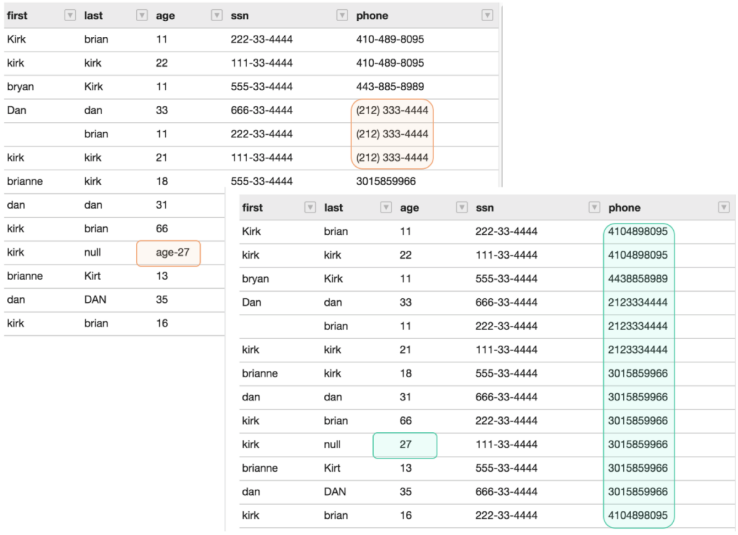
README.md

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

Simple rules would be applied to filter a condition on a single column in a single table.

Example #1

In this example you can see how to create a simple SQL rule, with name **simple_sql_rule**.

Code	Description
rule.setRuleNm("**simple_sql_rule**")	Adding the name of the given rule
rule.setRuleValue("**startDate < '2011-11-01'"**)	Setting the simple SQL expression. No **JOIN** allowed between tables!
rule.setRuleType("**SQLG**")	Setting 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
  }
```

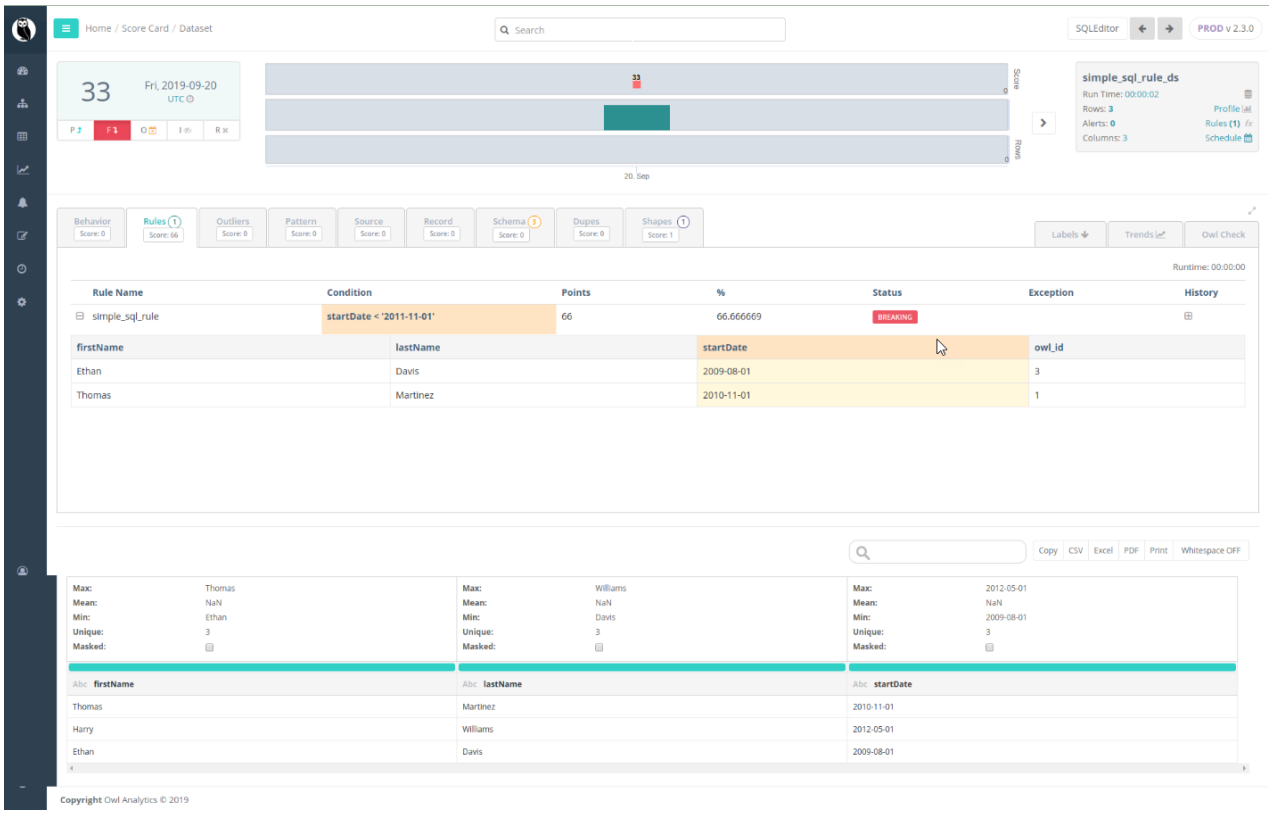
```
  val rule = new Rule {
    setDataset(opt.dataset)
    setRuleNm("simple_sql_rule")
    setRuleValue("startDate < '2011-11-01'")
    setRuleType("SQLG")
    setPerc(1.0)
    setPoints(1)
    setIsActive(1)
    setUserNm("admin")
  }

  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()
```

Result

via Code

You can do multiple assertion on the result of the DQ Check process.
Using **owl.hoot** parameter will provide access to the execution results, in this case for the rule.

via UI



Example #2

In this example you can see how to create a simple SQL with rule with **templates**, with name **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")
```

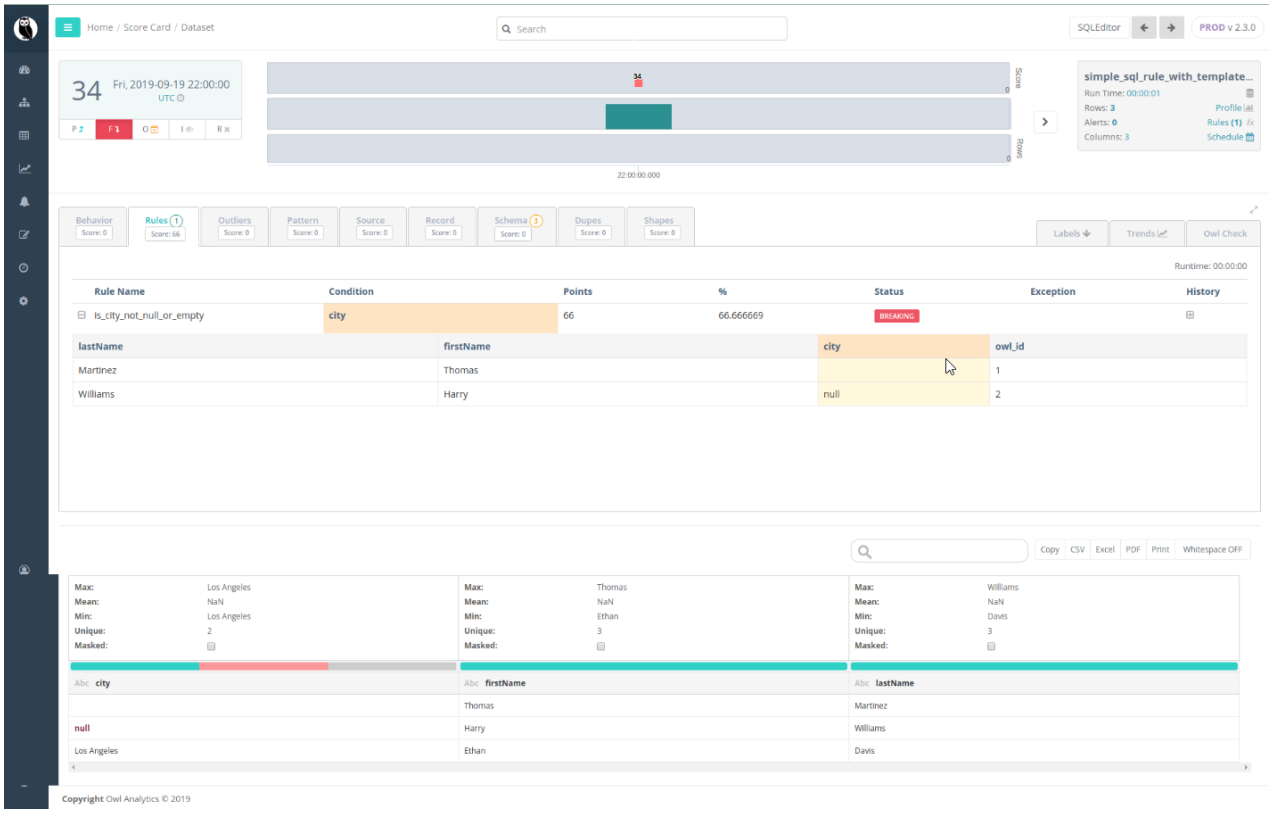
Code

```
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()
```

Result

via UI



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.

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

- 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

- 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.

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")
```

Data type based rules

Simple check for individual columns.

Rule types

Empty check

Rule type: EMPTYCHECK
Description: Checking whether the target column has empty values or not.

Null check

Rule type: NULLCHECK
Description: Checking whether the target column has *NULL* values or not.

Date check

Rule type: DATECHECK
Description: Checking whether the target column has **only** DATE values or not.

Integer check

Rule type: INTCHECK
Description: Checking whether the target column has **only** INTEGER values or not

Double check

Rule type: DOUBLECHECK
Description: Checking whether the target column has **only** DOUBLE values or not.

String check

Rule type: STRINGCHECK
Description: Checking whether the target column has **only** STRING values or not.

Mixed datatype check

Rule type: DATATYPECHECK
Description: ---

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

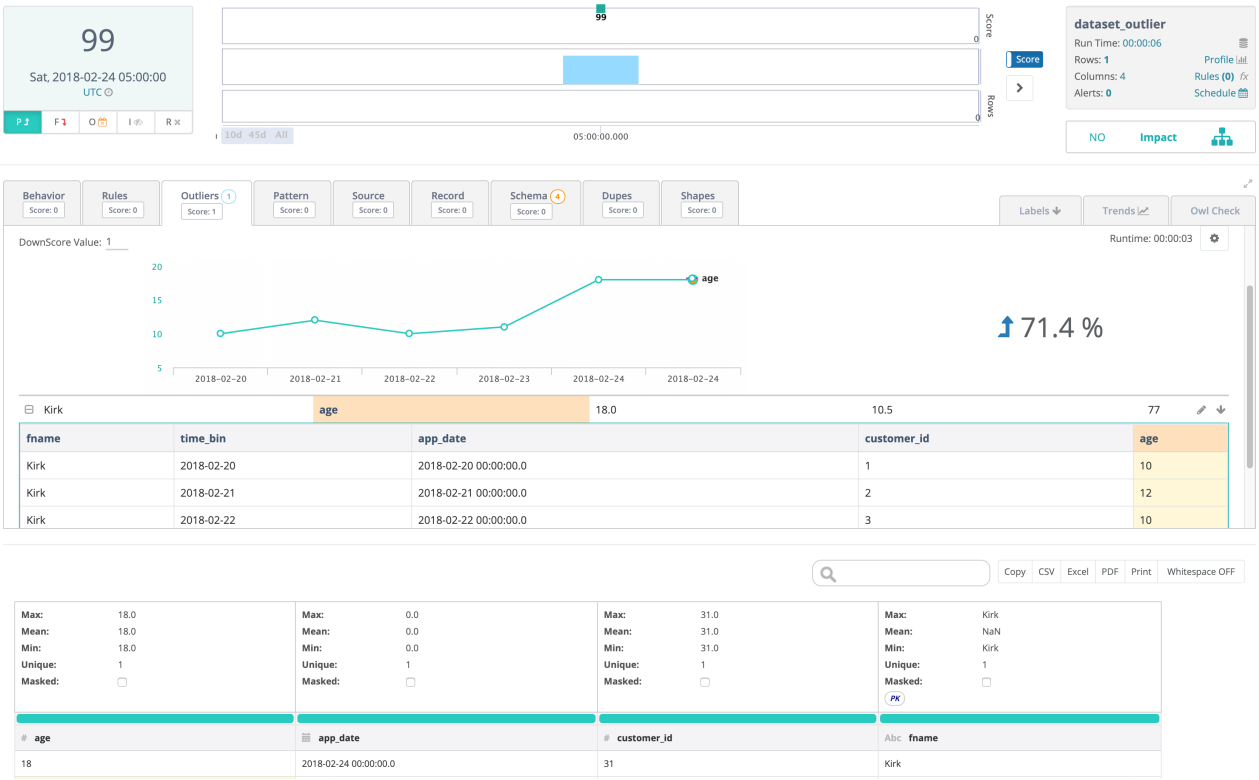
```
fname,app_date,age,customer_id
Kirk,2018-02-24,18,31
Kirk,2018-02-23,11,4
Kirk,2018-02-22,10,3
Kirk,2018-02-21,12,2
Kirk,2018-02-20,10,1
```

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		
sparkkeytab		

Field name	CLI prop	Description
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		
dbLookBack		
connectionURL		

Field name	CLI prop	Description
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.
categorical		
by		
limit		

Field name	CLI prop	Description
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
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	

Field name	CLI prop	Description
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		

Field name	CLI prop	Description
excludes		
dataShapeOn		
statsOn		
correlationOn		
histogramOn		
cardinalityOn		
dataShapeColsInc		
dataShapeColsExc		

Source Options

Field name	CLI prop	Description
on		
includes		
excludes		
key		
fileQuery		
map		
score		
datasetSrc		
driverSrc		
userNameSrc		
passwordSrc		
connectionURLSrc		

Field name	CLI prop	Description
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		

Field name	CLI prop	Description
master		
jars		
libs		
driver		

Misc Options

Field name	CLI prop	Description
obslimit		
nullValue		

Classes

Load

```java
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 OwlOp-
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

Scale + Data Science

Where Scale meets Data Science. Scale linearly with your data by adding executors and/or memory.

```
-f "file:///Users/home/salary_data.csv" \
-d "," \
-rd "2018-01-08" \
-ds "salary_data"
-numexecutors 2 \
-executormemory 2g
```

Yarn Master

If CollibraDQ 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

DQ also runs using spark master by using the -master input and passing in spark:url.

Spark Standalone

DQ runs in standalone most but naturally will not distribute the processing beyond the hardware it was activated on.

Options	Description
deploymode	spark deploymode option
drivermemory	driver memory example 3G for local space
executorcores	spark executor cores
executormemory	spark executor memory option example 3G
master	overrides local[*], i.e. spark://myhost:7077, yarn-client, yarn-cluster
sparkprinc	kerberos principal name ex: 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/
```

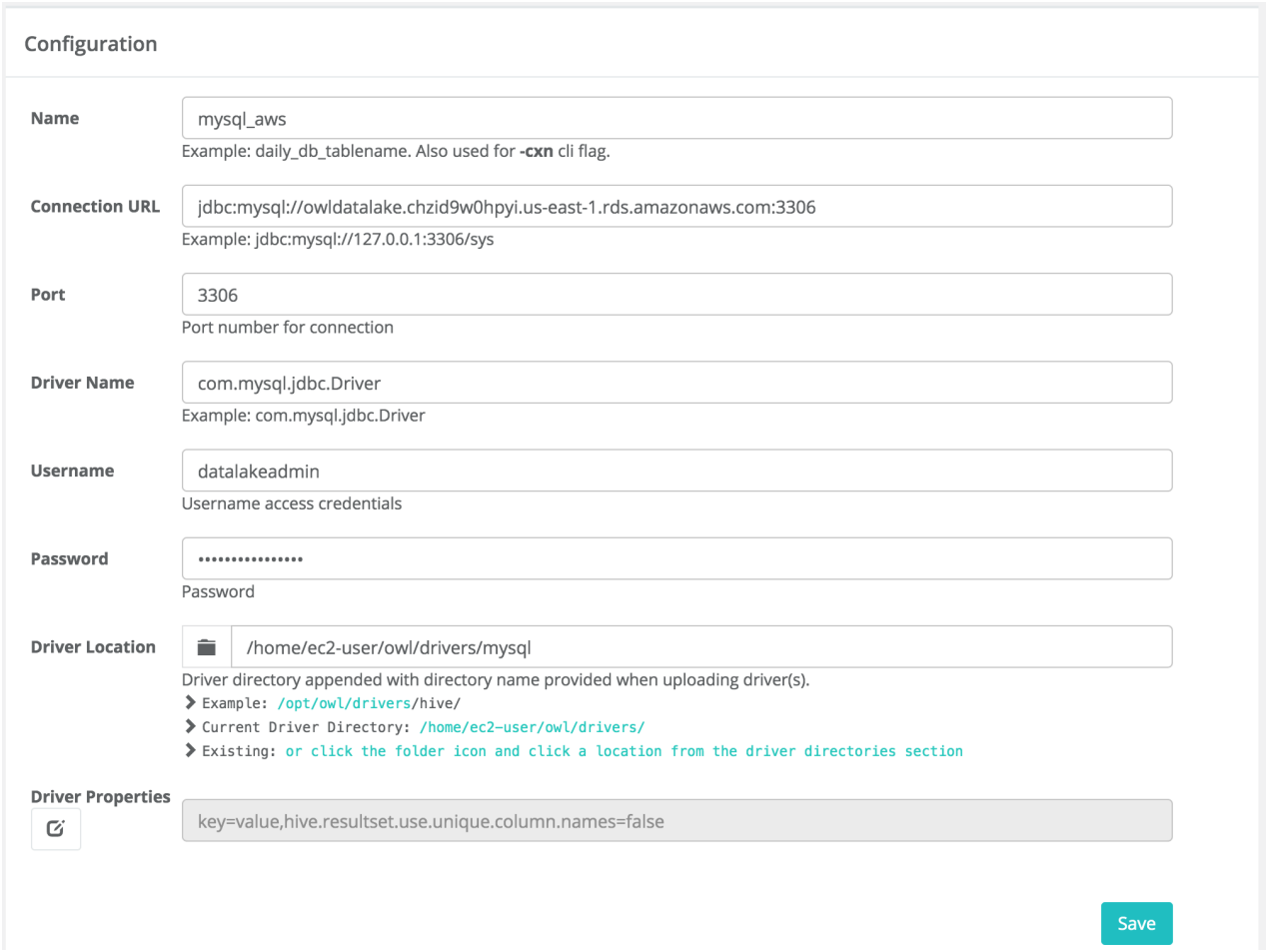
Parallel JDBC Spark-Submit

```
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
```

DQ Job JDBC

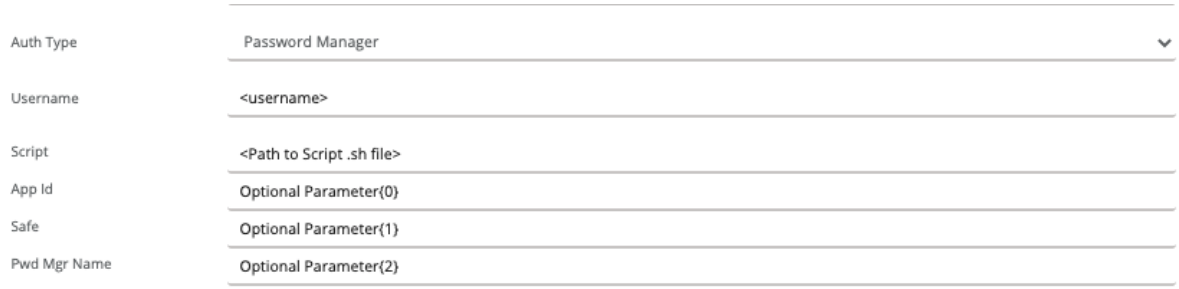
Connect to any database via JDBC.

```
-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

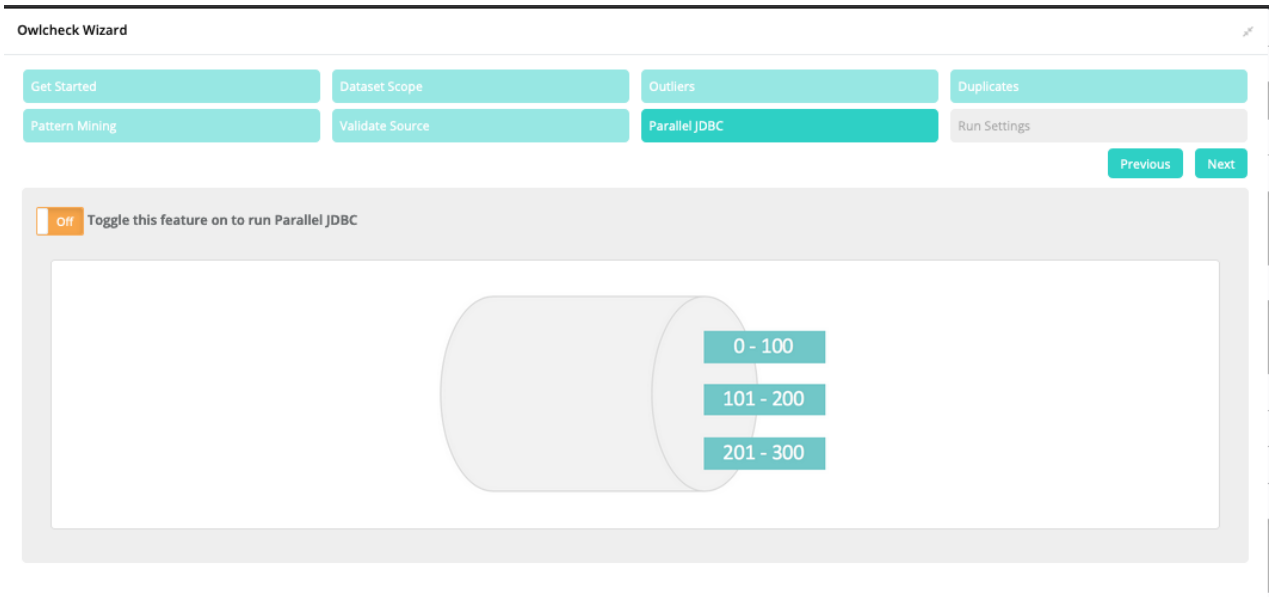
```
props.connectionProps.put("fetchsize", "3000")
```

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

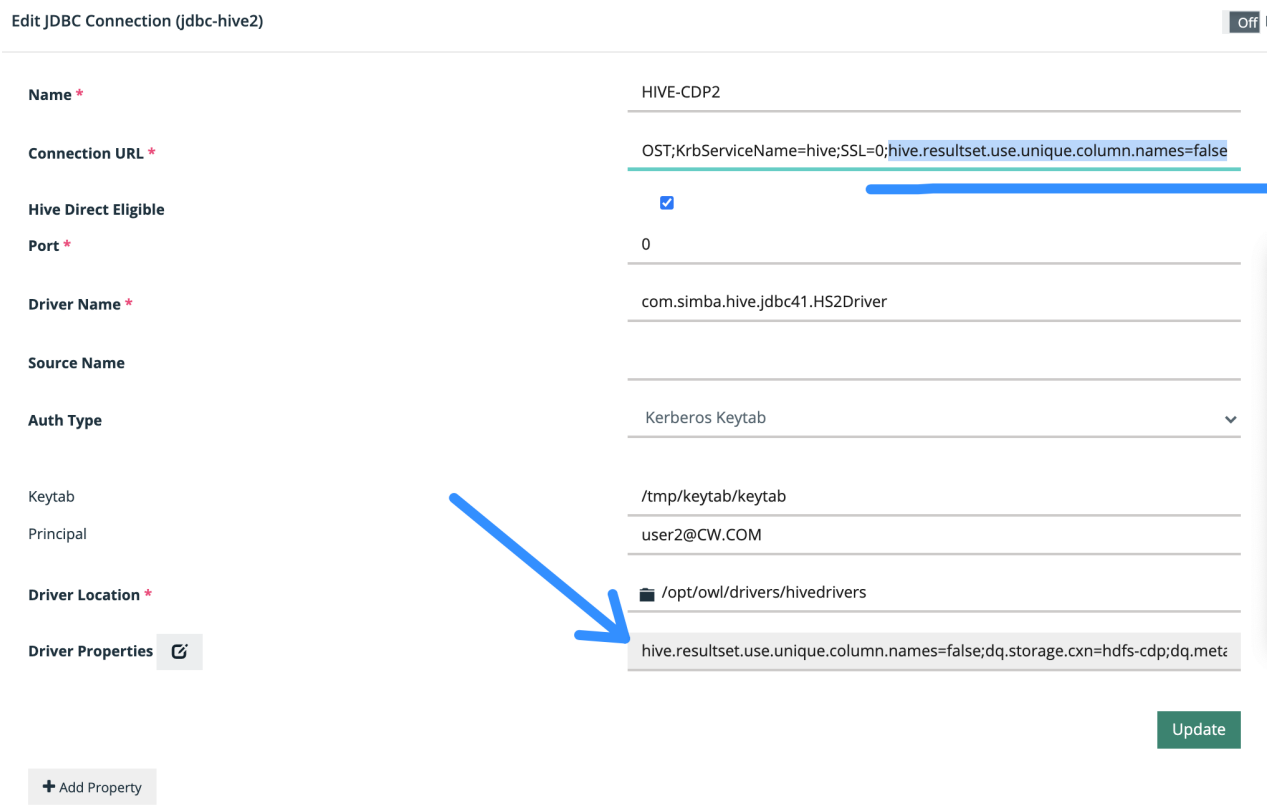
CDH Driver - com.cloudera.hive.jdbc41.Datasource

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.

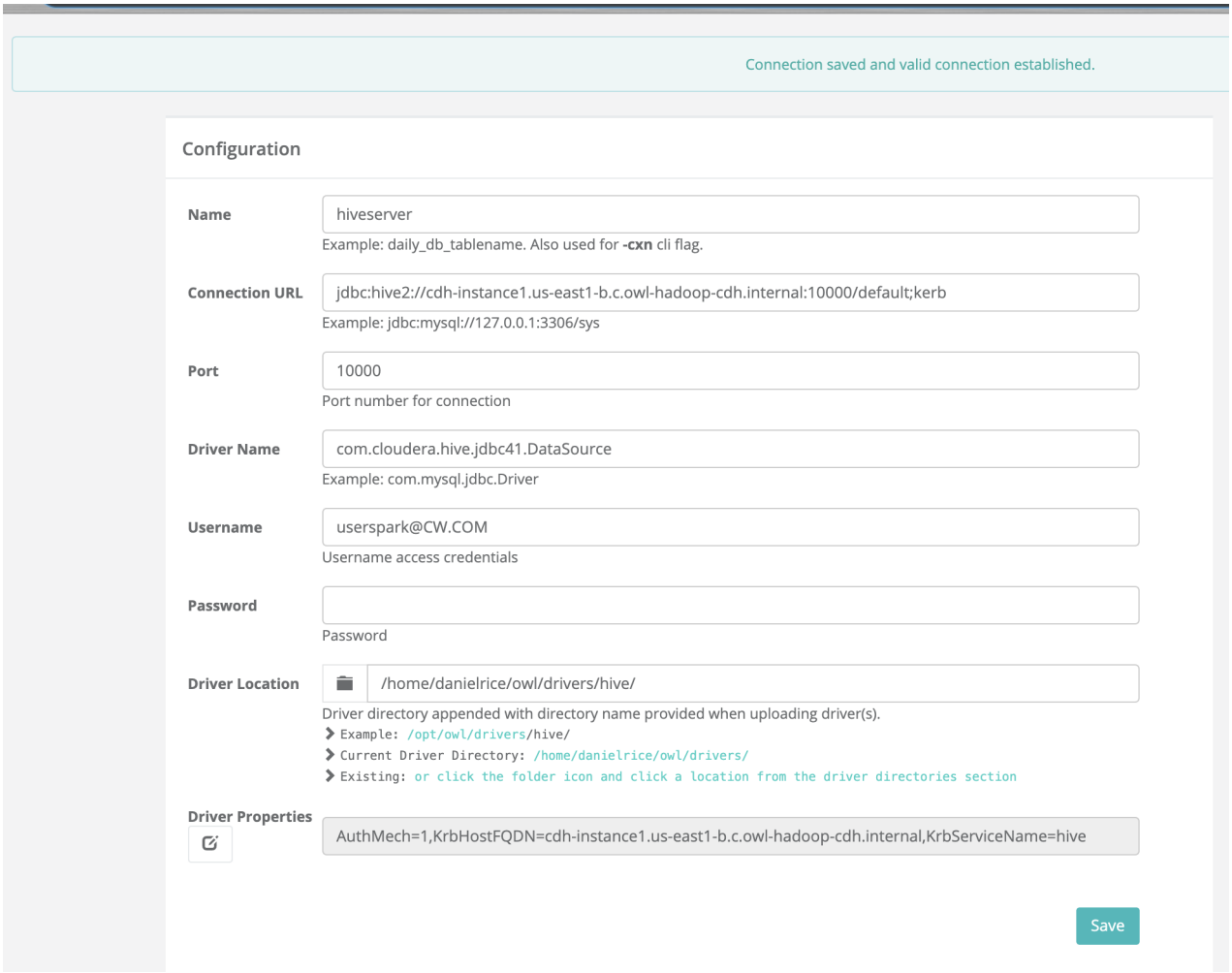
```
beeline -u 'jdb-
c:hive2://<HOST>:10000/default;principal=hive/cdh-instance1.us-
east1-b.c.company-hadoop-cdh.internal@CW.COM;useSSL=true' -d
org.apache.hive.jdbc.HiveDriver
```

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.

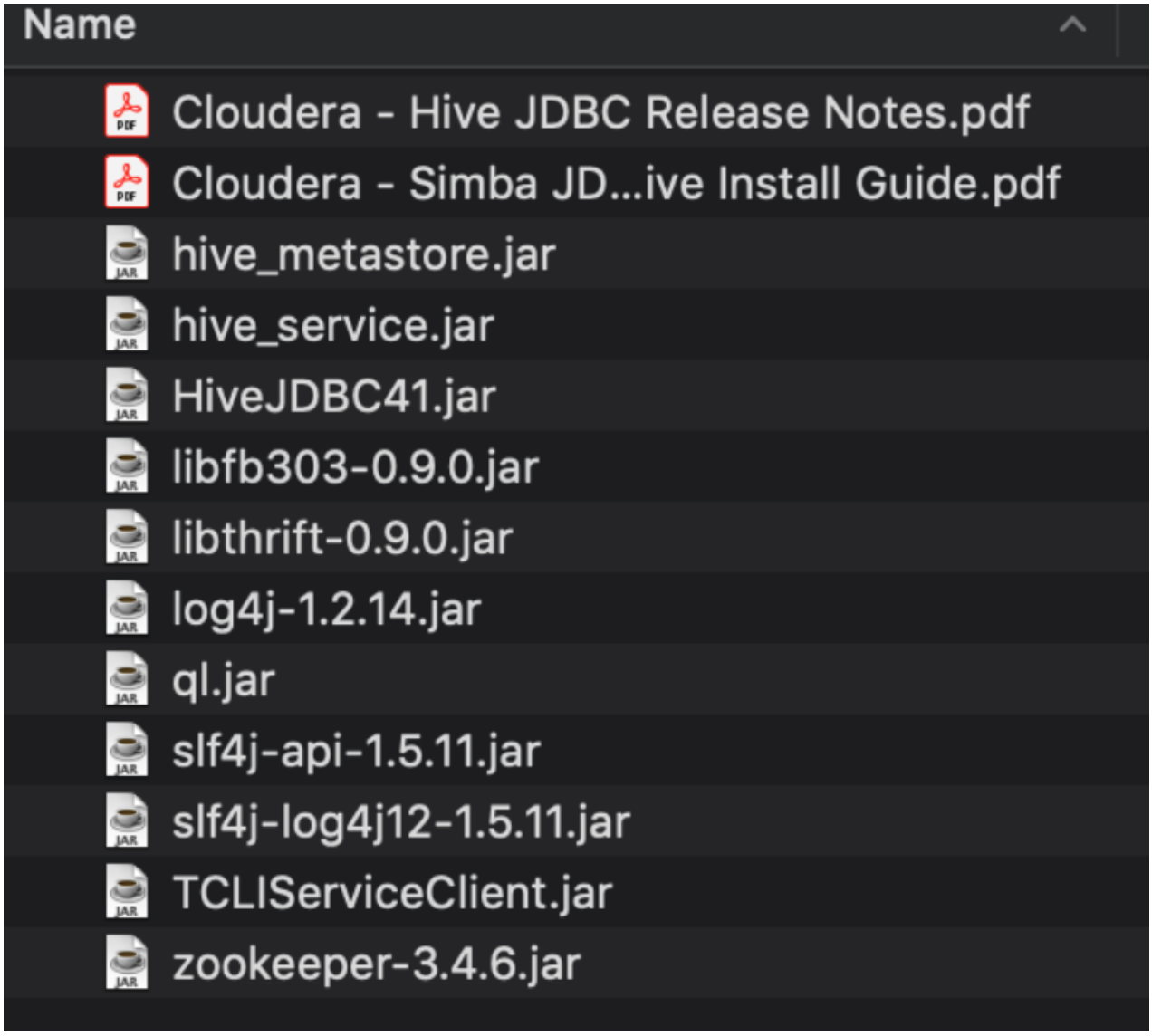
```
$JAVA_HOME/bin/java -Dload-
er.path=lib,/home/danielrice/owl/drivers/hive/ \
-DowlAppender=owlRollingFile \
-DowlLogFile=owl-web -Dlog4j.configurationFile=file://$INSTALL_
PATH/config/log4j2.xml \
$HBASE_KERBEROS -jar $owlweb $ZKHOST_KER \
--logging.level.org.springframework=INFO $TIMEOUT \
--server.session.timeout=$TIMEOUT \
--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.

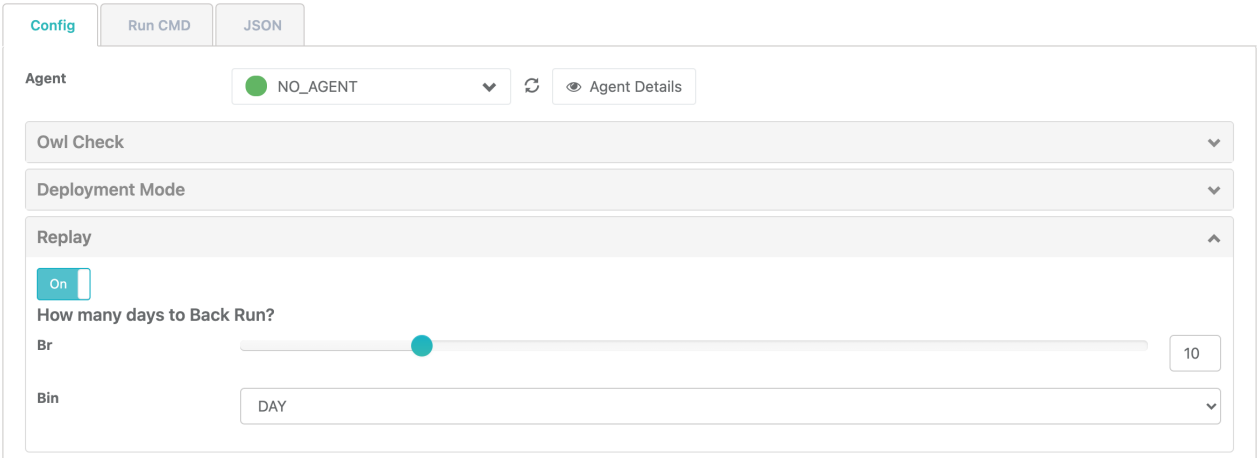
```
#1 authenticate
curl -sb -X POST -d username={user} -d password={password}
http://$OWL_HOST/login -c cookies.txt

#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
```

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
```

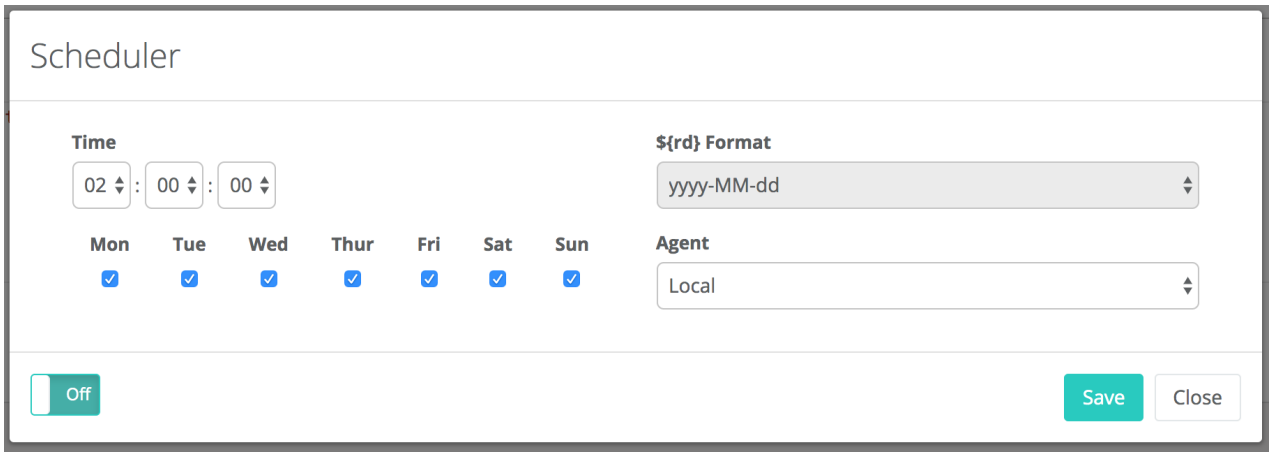
Template

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

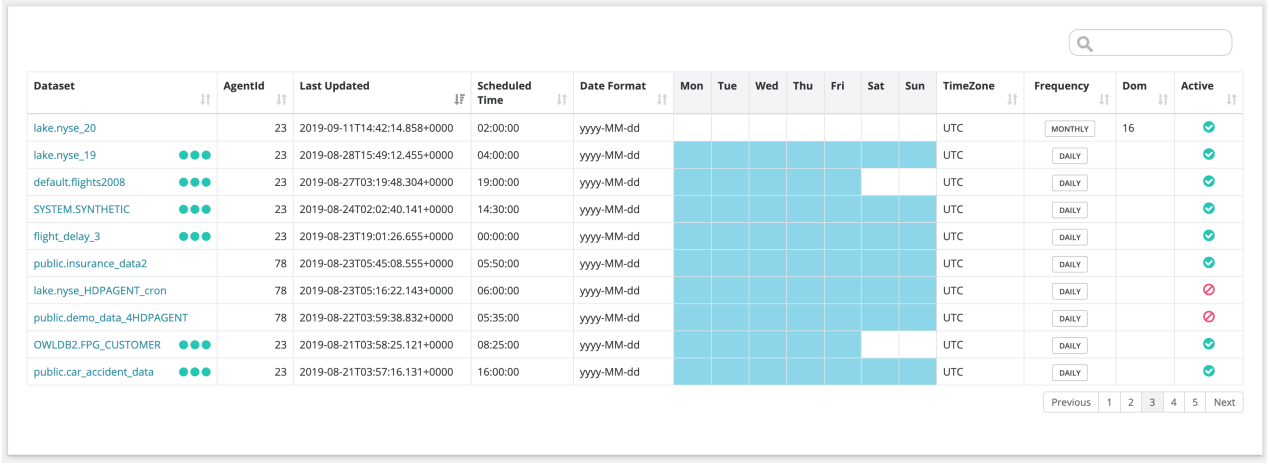
A quick option is to use DQ's built in scheduler. DQ automatically substitutes the runtime variables like $ into the job. This also gives you control to edit the DQCheck.



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 LinkID

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.

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.

Providing the link ID

There are two ways to provide the link ID:

1. From the command line using `-linkid`.
2. In a notebook via opt.linkId.

Combining link ID and Run Discovery

To combine the features of link ID and Run Discovery, first enable link ID and then use Rule Discovery on the Rules tab of the Findings page. This lets you apply sensitivity labels to data classes and trigger breaks for all the records that do not match your link ID.

Link ID and the [[[Undefined variable DataQuality.DQ Metastore]]]

No personal data is stored in the [[[Undefined variable DataQuality.DQ Metastore]]] when using link ID. The [[[Undefined variable DataQuality.DQ Metastore]]] only stores:

- The rule that is 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.

Viewing break records

To view rule break records, navigate to the Breaks tab on the Rule Builder page. Rules with break records have associated link IDs that link back to the original dataset. All remediation for data quality issues is performed outside of Collibra DQ

Steps

1. In Explorer, select a table and create a DQ Job. >> The DQ Job page opens.
2. In the Scope section, select the **Add Link Back to Source** checkbox. >> A new column, Link ID, appears.
3. Select the columns that represent the **primary key(s)** of your dataset. >> By selecting multiple checkboxes, you can create a composite key.
4. Run your DQ Job.
    a. Click **Build Model**.
    b. Click **Save/Run**.
    c. Verify the information on the **Register** page.
    d. Click **Estimate Job** and then click **Run**.
        » Your DQ Job is sent to the Jobs page.
5. Open your DQ Job.
    a. Open the **Jobs** page.
    b. Select your **DQ Job** from the list.
        » Your DQ Job opens.

6. Apply a rule.
    a. In the metadata box, click **Rules**.
        » The Rule Builder opens.
    b. Select a **rule type**.

> Note All Simple and Freeform rules are eligible for Link ID.

7. Re-run your DQ Job.
    a. On the Findings page, click the **DQ Job** tab.
    b. Verify the **run command**.
    c. Click **Run DQ Job**.
        » Your DQ Job is submitted to the Jobs queue.
8. Open your DQ Job.
    a. Open the **Jobs** page.
    b. Select your **DQ Job** from the list.
        » Your DQ Job opens.
9. View your rule breaks.
    a. In the metadata box, click **Rules**.
        » The Rule Builder opens.
    b. Click the **Breaks** tab.
        » A table displays a row for each record in violation of the rule you set. The Link ID column lets you identify broken records within your dataset and mark them for remediation outside the DQ application.
    c. Export the break records. Supported file formats include Excel and CSV.

Notebook

```
val opt = new OwlOptions()
opt.runId = "2018-02-24"
opt.dataset = "orders"
opt.linkId = Array("transaction_id", "trans_time")
```

Command Line

```
./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.

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 on categorical.
DUPE	YES	Each duplicate or fuzzy match will have a link back to the record for remediation.
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 not 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 when using Freeform SQL the linkID columns should be part of the select statement. LinkID columns should be unique identifiers.
BEHAVIOR	NO	This class of data change is when a a section of your data is drifting from its normal tendency there is no 1 record to link.

Activity	Supported	Description
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. This item is still under performance review.

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()

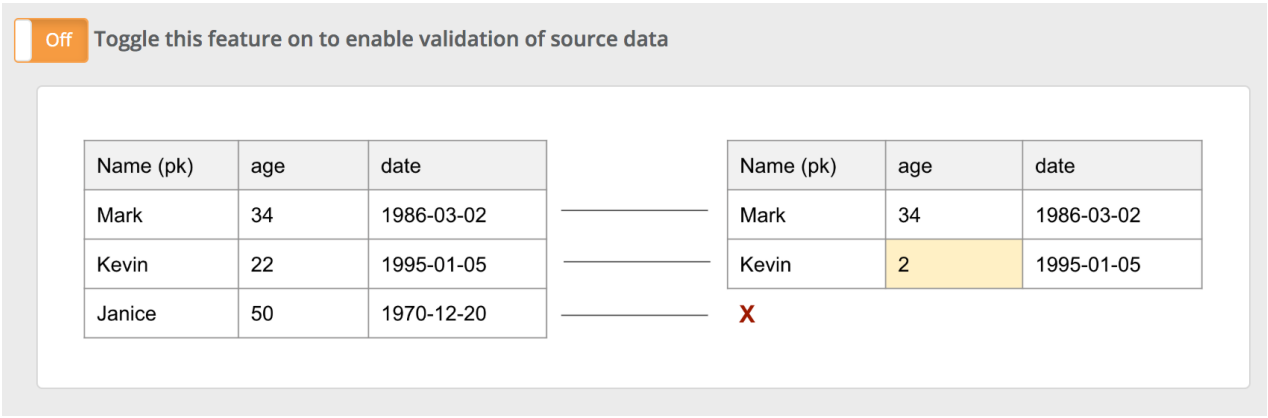
The getRuleBreaks endpoint retrieves all broken records within your dataset. There is no size limit to this API.

```
----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 \
```

DQ Job 43M rows

Collibra DQ commonly benchmarks on large daily datasets. In this case, a 43 million row table with 12 columns completes in under 6 mins (5:30). The best balance for this dataset was 3 executors each with 10G of ram.

```
./owlcheck \
-u user -p password \
-c jdbc:mysql://owldatalake.chzid9w0hpyi.us-east-1.rd-
s.amazonaws.com:3306 \
-q "select * from silo.account_large where acc_upd_ts > '2018-
02-01 05:0:00'" \
-rd 2019-02-02 \
-ds account_large \
-dc acc_upd_ts \
-corroff \
-histoff \
-driver com.mysql.cj.jdbc.Driver \
-lib "/home/ec2-user/owl/drivers/mysql/" \
-master yarn \
-deploymode client \
-numexecutors 3 \
-executormemory 10g \
-histoff -corroff -loglevel DEBUG -readonly
```

> Note *Not all DQ features were turned on during this run. On large datasets it is worth it to consider limiting the columns, DQ-features, or lookbacks if they are not of interest.*

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.

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

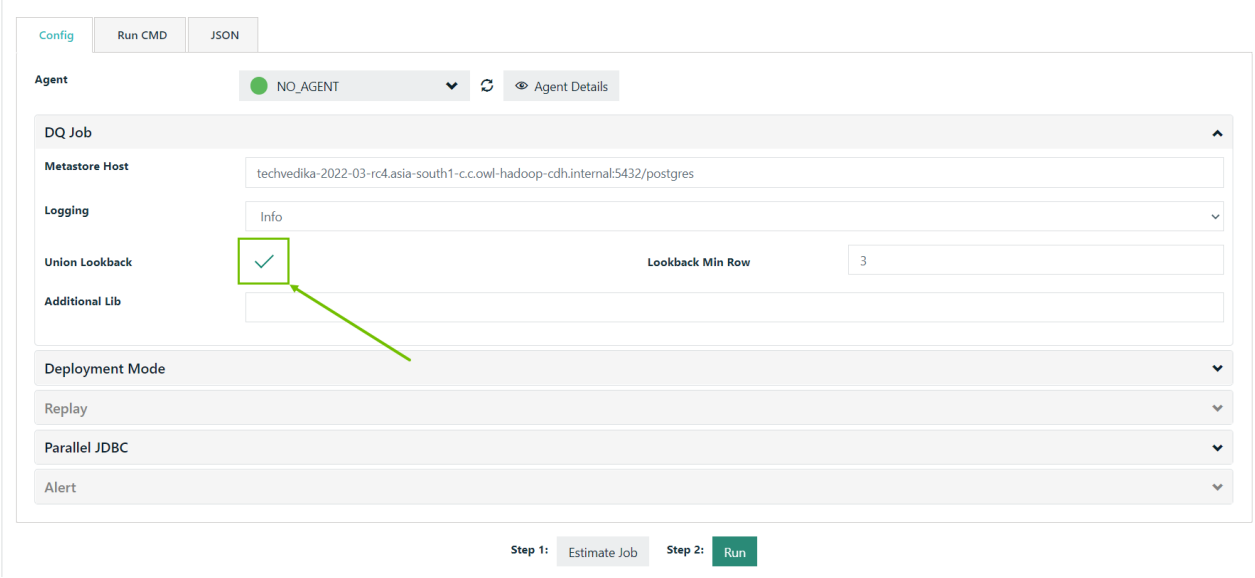
yarn.Client: Application report for application_1557720962505_0085 (state: ACCEPTED)
yarn.Client: Application report for application_1557720962505_0085 (state: ACCEPTED).

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 or 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".

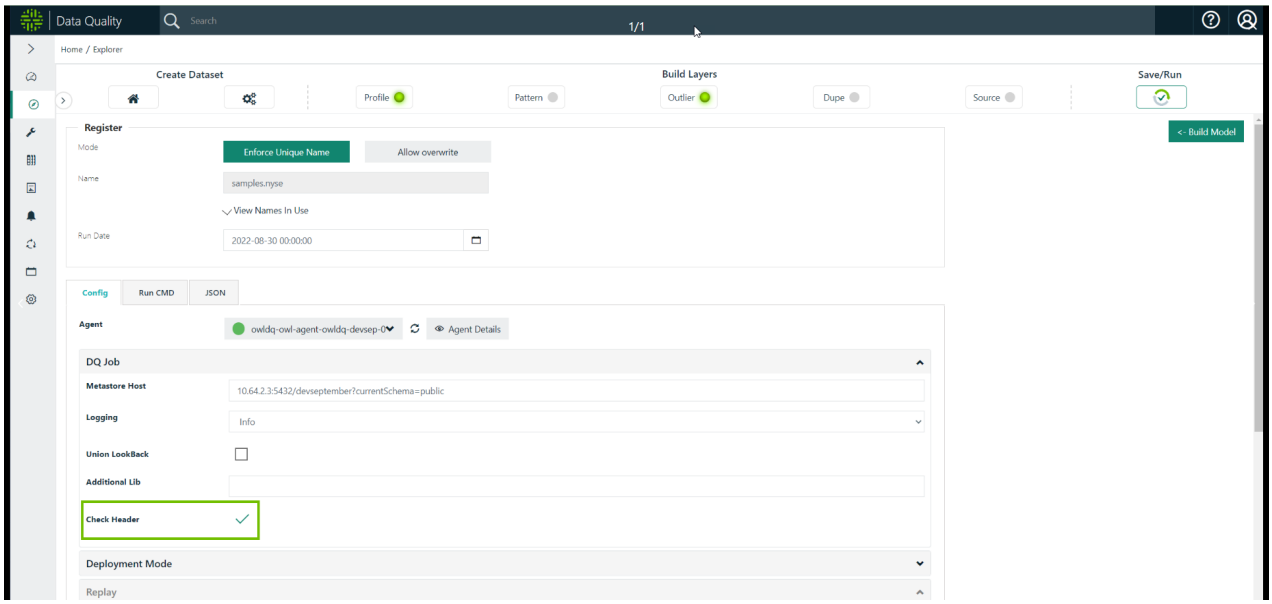
Header Check

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

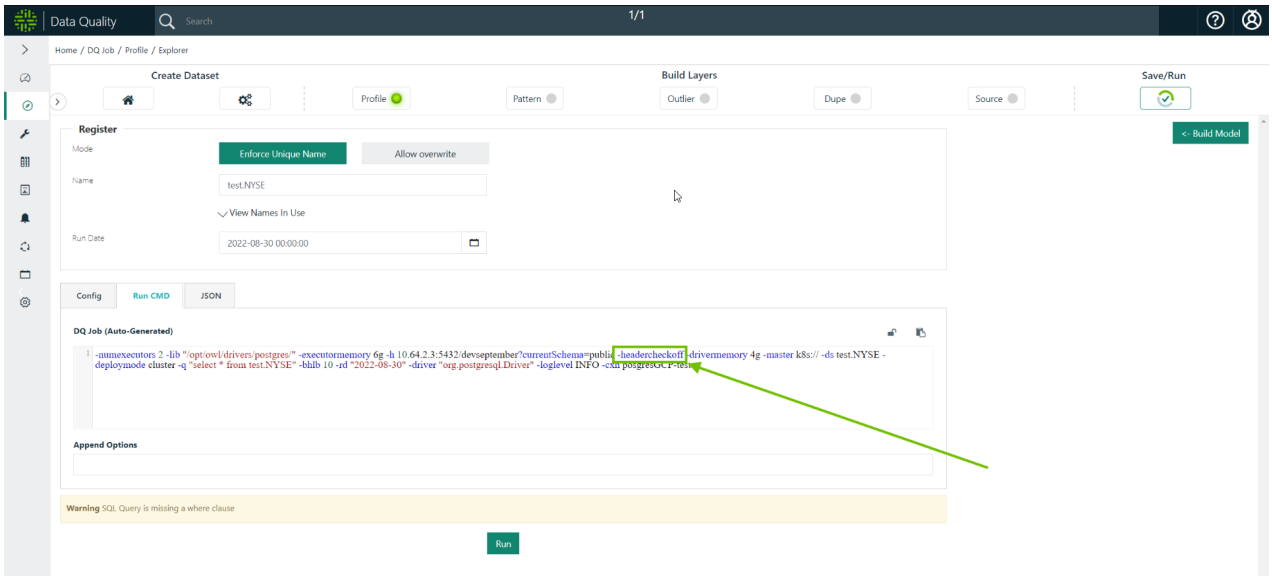
The Check Header checkbox is checked by default. When it is checked, schema findings do not display when detected.

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.

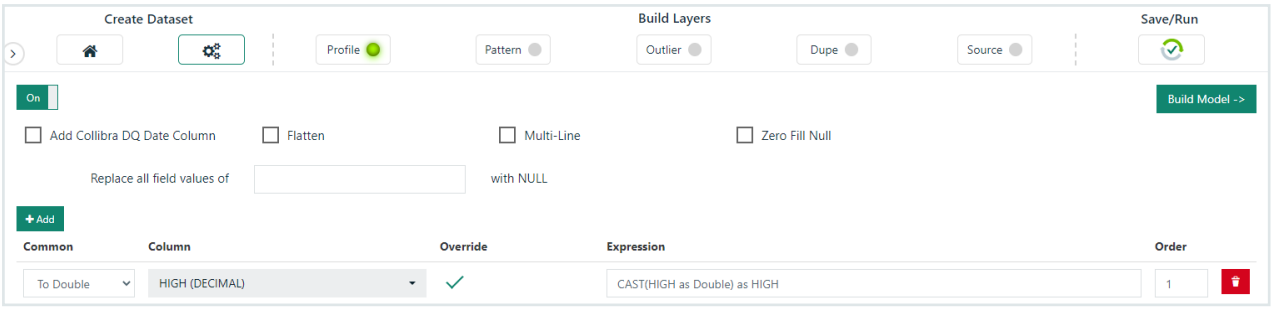
```
val conn =`` ``connProps`` ``+ ("dbtable" -> s"($query)``
``$dataset")val df =`` ``spark.read.format("jdbc").options
(conn).load

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

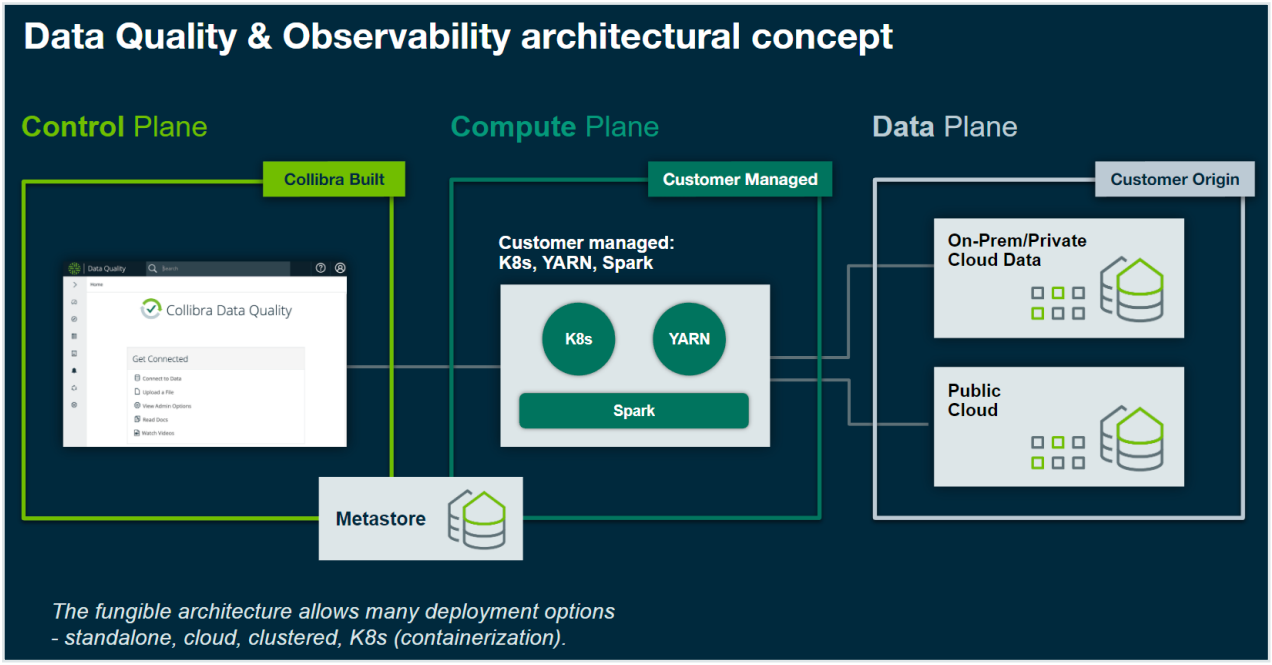
- The UI does not run multi-line. A workaround is to supply these options manually in the command line.
- Explorer can flatten nested JSON array, but DQ Core can only flatten nested JSON with JSON object. Because of this limitation, you may encounter a scenario where the results in Explorer are flattened, but not in DQ Core.

Chapter 16

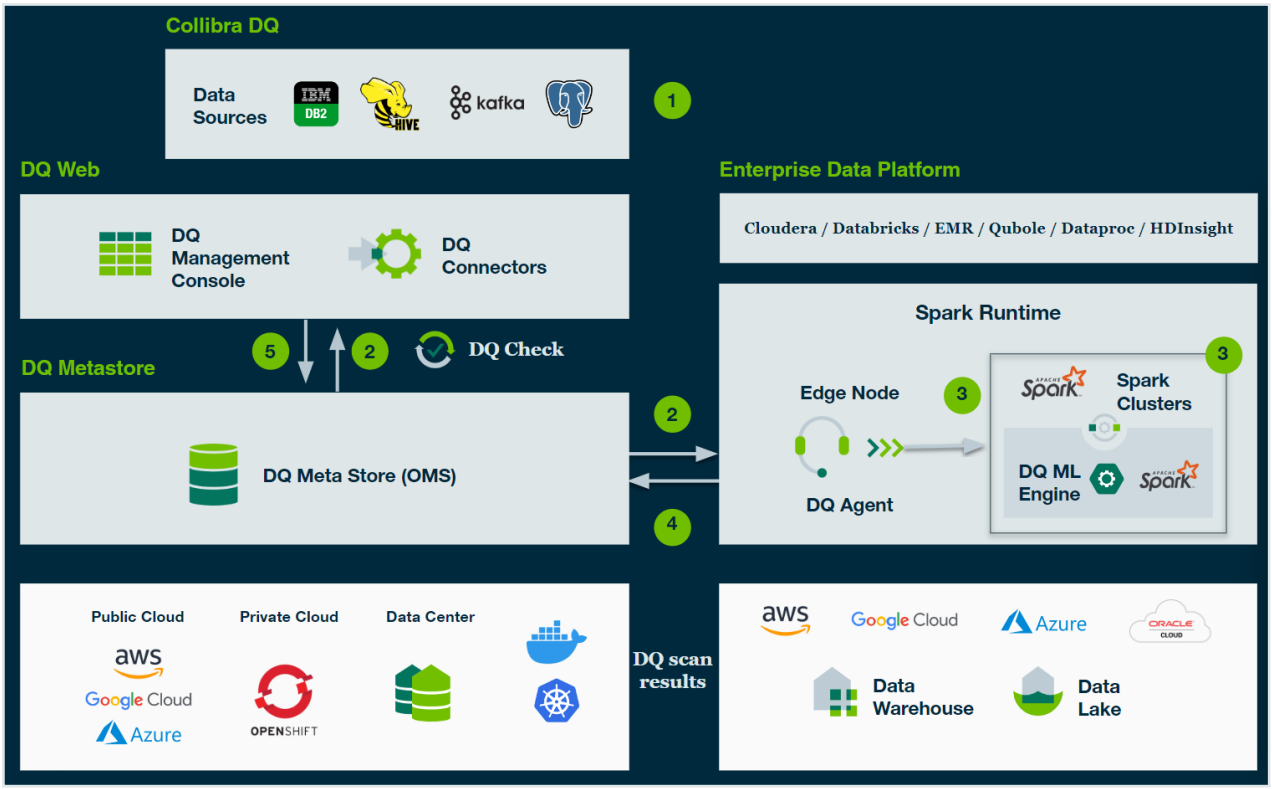
Collibra DQ Architecture

Architecture Diagram

Collibra DQ Architecture

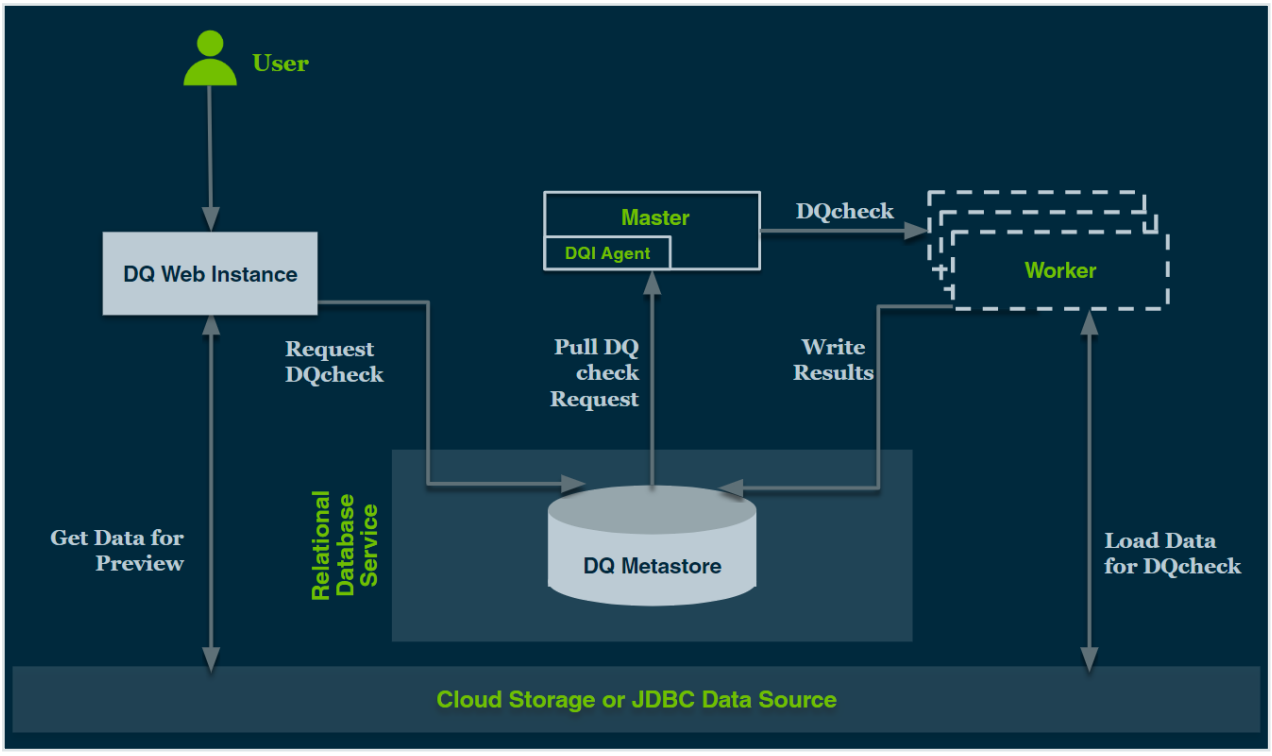


High-Level Diagram

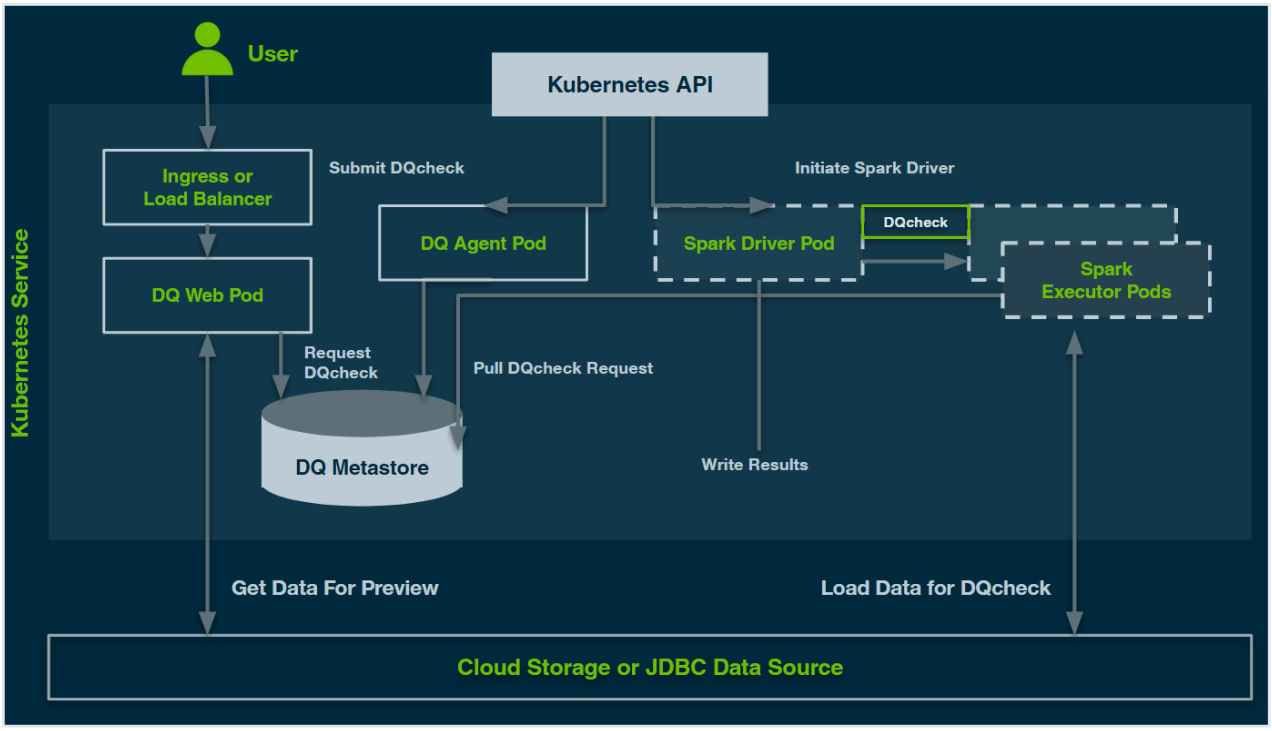


1. Connect to data sources.
2. Build the DQ scan algorithm and submit the job.
3. Execute the Spark job.
4. Write the DQ results in the Metastore.
5. Browse the results of the DQ Scan with the management console.

Collibra DQ Hadoop Deployment Diagram

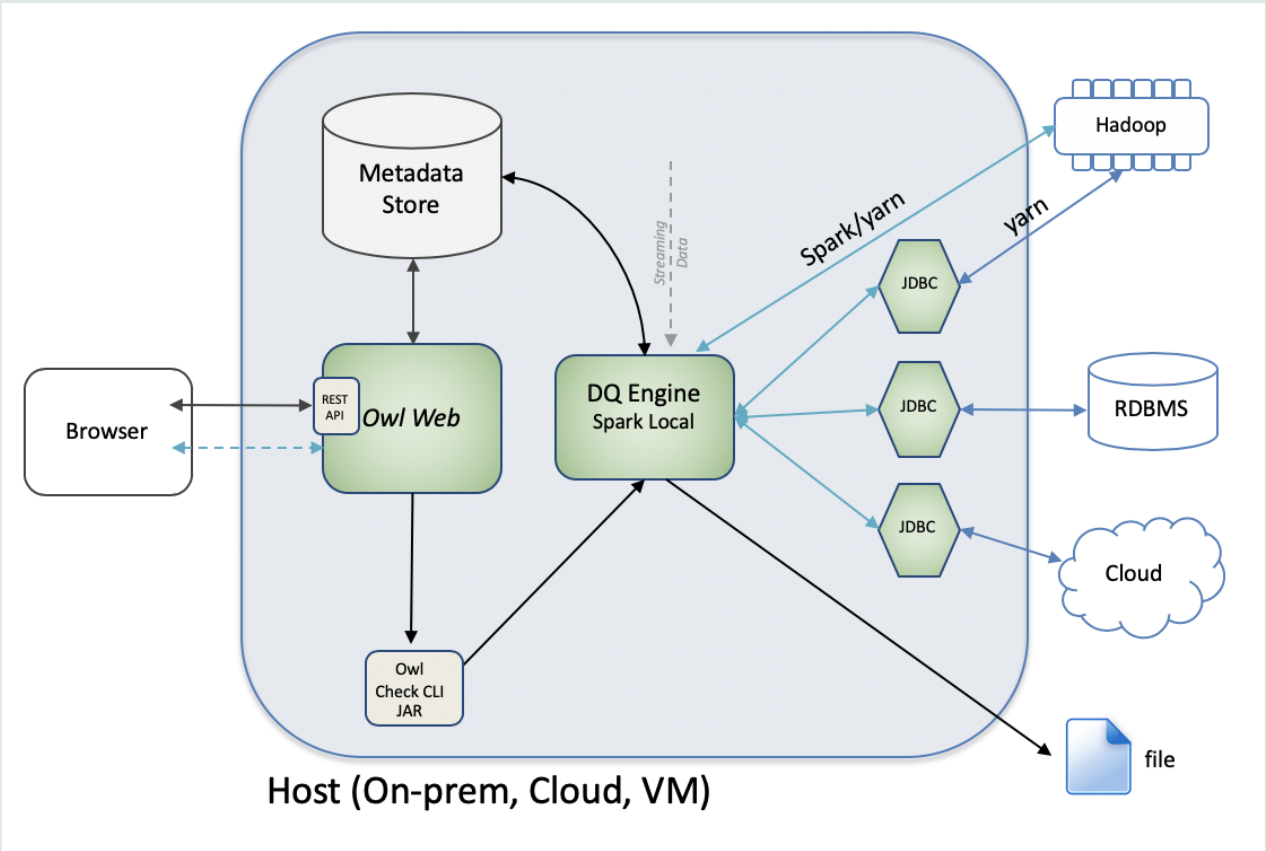


Collibra DQ Kubernetes Deployment Diagram



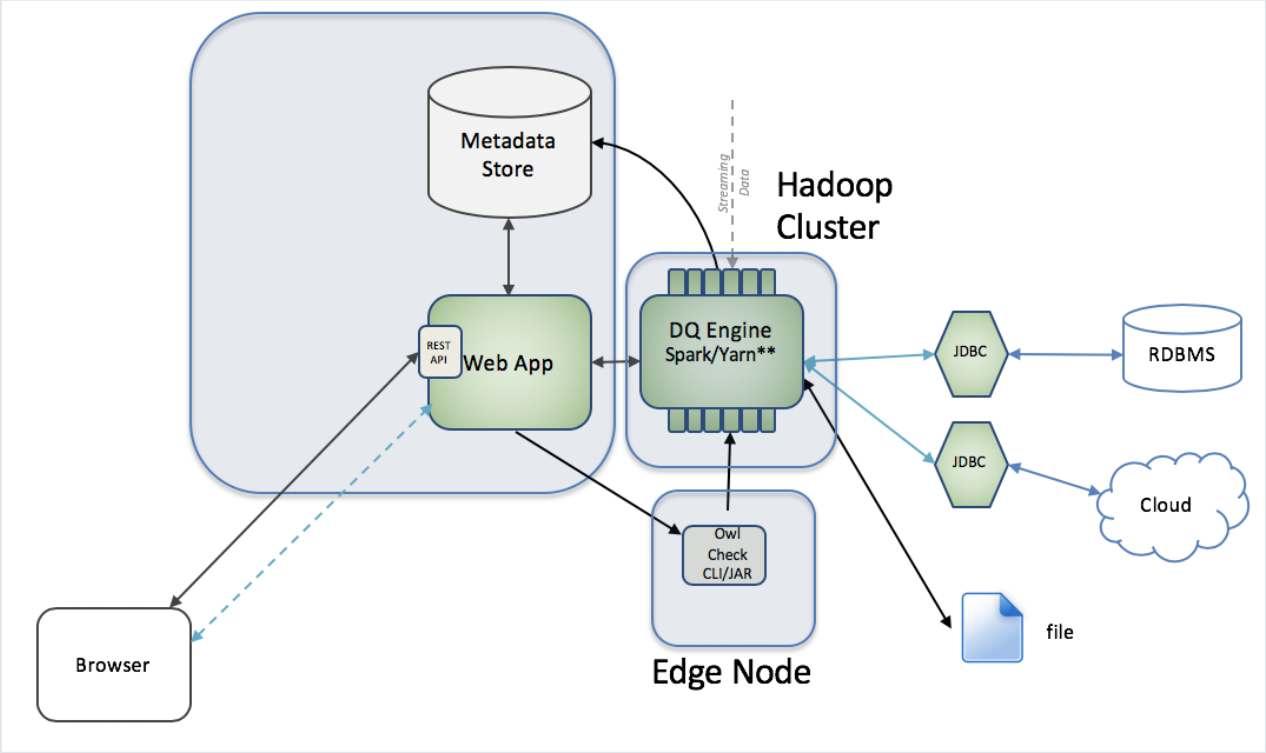
> Note For Kubernetes deployments of Collibra DQ should use Auto Scaling and Spot instances to further increase efficiency and reduce cost.

Collibra DQ Standalone



The image above depicts owl-web, owl-core, Postgres and orient all deployed on the same server. This can be an edge node of a Hadoop cluster or a server that has access to run Spark-submit jobs to the Hadoop cluster. This server could also have JDBC access to other DB engines interested in being quality scanned by Collibra Data Quality & Observability. Looking at this depiction from left to right the client uses their browser to connect to Collibra DQ's Web Application running on the default port 9000. The Collibra DQ Web Application communicates with the metastore.. The Web Application can run a local DQ check, or the Data script can be launched from the CLI natively. The DQ check launches a job using Collibra DQ's built in Spark Local DQ Engine. Depending on the options supplied to the DQ check command, the Collibra DQ can scan a file or database with JDBC connectivity.

Collibra DQ Distributed



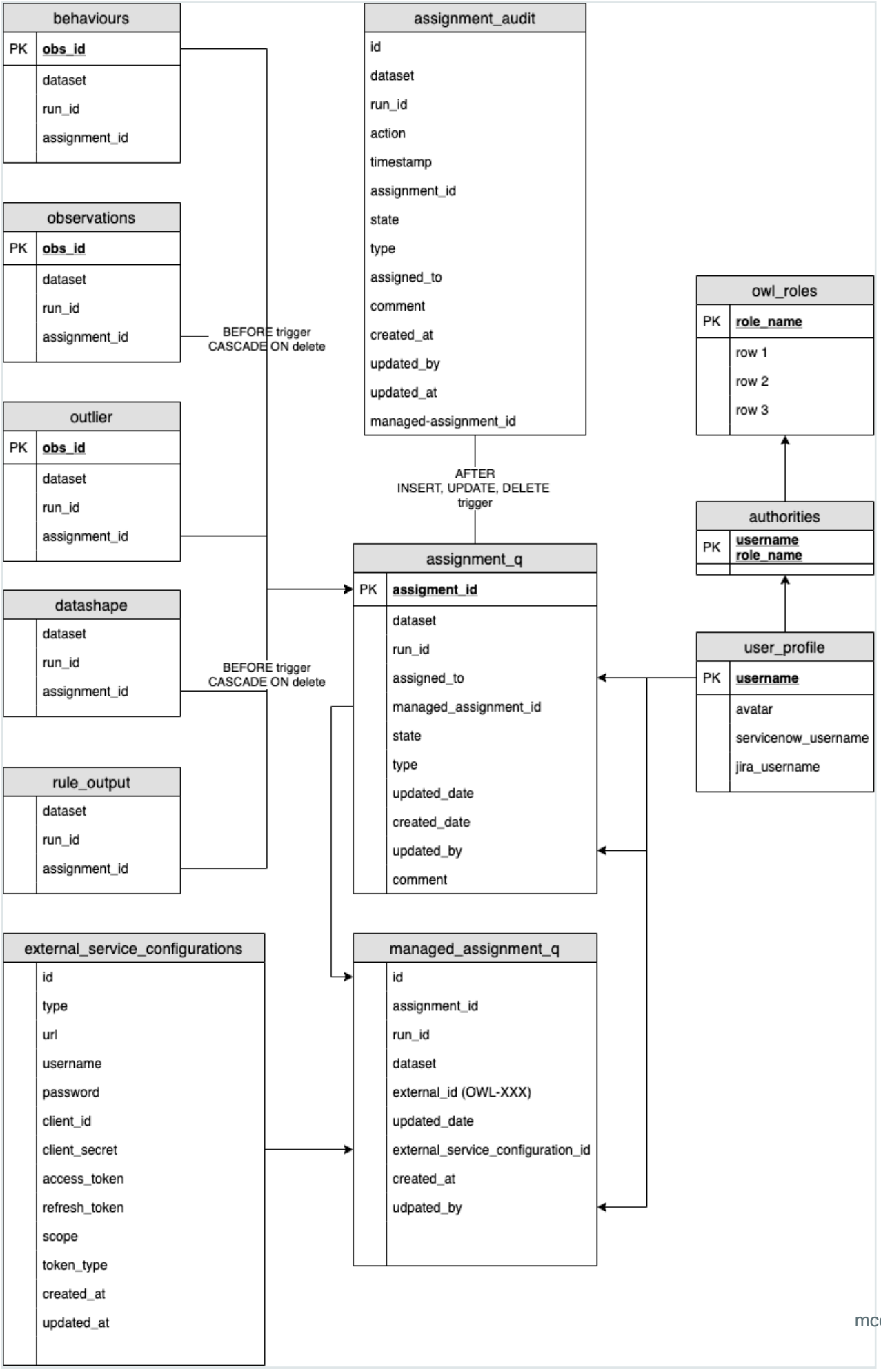
The image above depicts owl-web and owl-core deployed on different servers. In this example Owl-web is NOT deployed on the edge node. Owl-core is installed on the edge node and writes DQ check results back to the metastore that the DQ Web App points to.

> Note In this scenario, the metastore and the web-app run on the same host.

The other change is that the DQ check distributes the work on top of a Hadoop cluster to leverage Spark and use the parallel processing that comes with the Hadoop engine.

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

System Requirements

The following pages include the system requirements for installing Collibra Data Quality & Observability.

Supported Operating Systems

> Note Only 64-bit Linux operating systems are supported.

Standalone operating systems

- Red Hat Enterprise Linux 7.x
- Red Hat Enterprise Linux 8.x
- CentOS 7.x
- CentOS 8.x

Container operating system

- Red Hat Universal Base Image 8 Micro (ubi8-micro)

Hardware Sizing

Hardware Sizing (Standalone Install)

Small Tier - 16 Core, 128G RAM (r5.4xlarge / E16s v3)

Component	RAM	Cores
Web	2g	2
Postgres	2g	2
Spark	100g	10

Component	RAM	Cores
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 Performance Tuning for more information.

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.

Minimum System Requirements

Hardware based on role

Standalone, distributed, or fully-distributed.

- Please see Hardware Sizing for minimum resources for each component.

Java version

- Oracle JDK version 11.0.x
- Oracle JDK version 1.8.0_152.x
- Open JDK version 1.8.0.x

Encryption

- Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction

PostgreSQL Version

- Collibra Data Quality & Observability comes prepackaged with version 11.4 of PostgreSQL.
- Version 9.6.5 and above is supported if wanting to use an external metastore.

User Privileges

- Installation is completed via tarball.
- You must be able to create directories, launch scripts, and start processes (Java processes).
- SUDO is not required.
- ULIMIT settings of 4096 or higher
    - All owl services typically consume about 428 threads.
    - During each DQ Job, about 400 additional threads are consumed.
    - Thus 4096 threads can allow for about 9 concurrent DQ Jobs (on a standalone install). If more are needed, plan accordingly.

Planning the installation

It is always best to consult the Collibra DQ team for more information about what options make the most sense for your environment.

1. Determine how you plan to install Collibra DQ (standalone or distributed), as shown in the Architecture Diagram.
2. Plan your infrastructure, scale, and HA.
3. Validate your system prerequisites.
4. Obtain the packages from Collibra DQ.

Installation Packages/Files (BOM)

- 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

Default Ports used by Collibra DQ

- 5432 – PostgreSQL
- 9000 – DQ Web
- 9101 – Exposes the Health Check API to check that the DQ Agent is running and stable.

Supported Web Browsers

Browser	Version
Google Chrome (recommended)	70.0.3538.102 or newer

Browser	Version
Mozilla Firefox	52.8.0 or newer
Safari	12.0.1 or newer

Build Versions

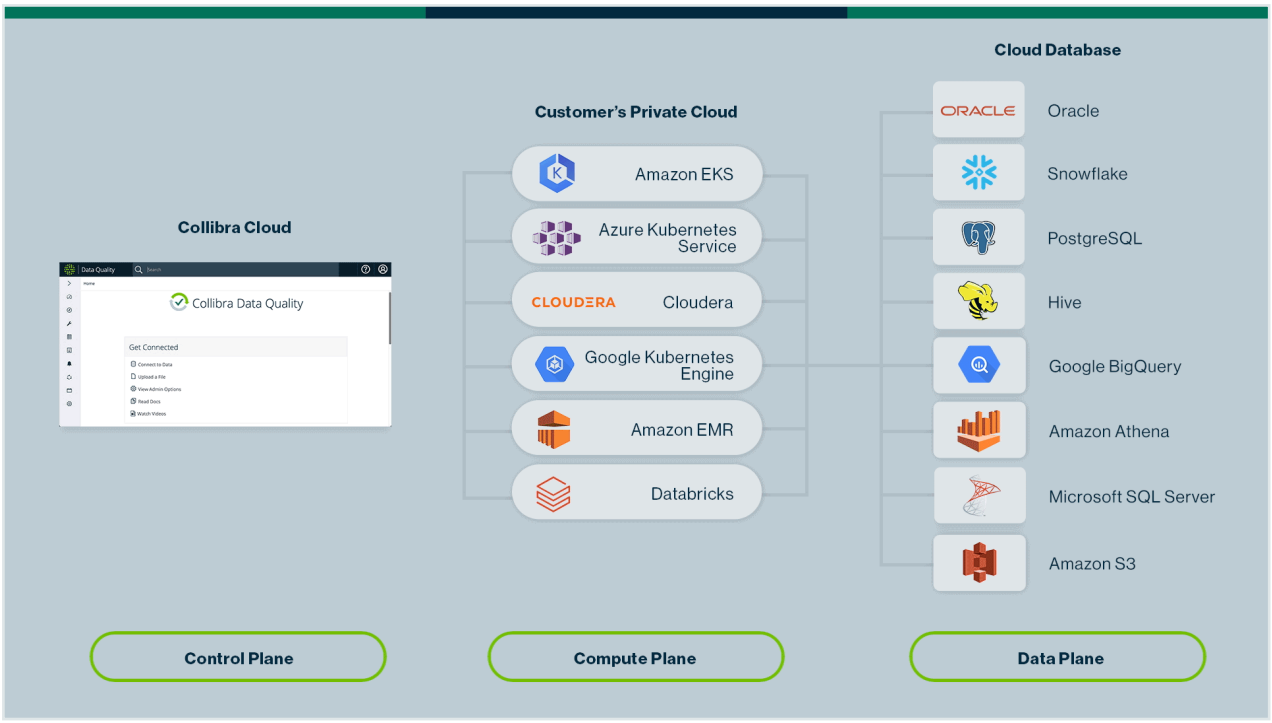
Generally available build versions

- Spark 3.4.1 (default)
- Spark 3.2.2
- Spark 3.1.2
- Spark 3.0.1
- Spark 2.3.0
- HDP 3
- CDH5
- CDH6
- CDH6-NOLOG
- K8s

Diagrams

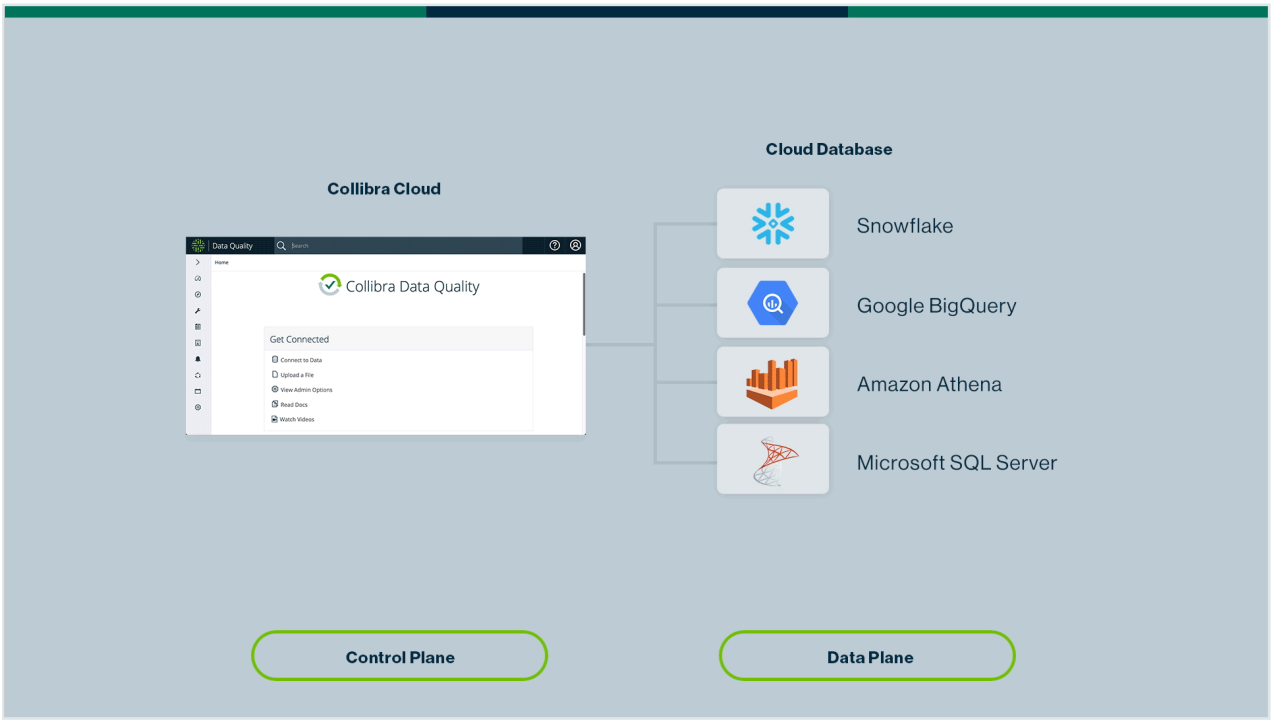
DQ Control to Compute to Data

As of March 2022, DQ Cloud is in private beta for select customers. General availability is expected in Q4 2022. Available compute plane options will start with Collibra Edge submitting jobs to Rancher K3s and Hadoop platforms (Cloudera, Dataproc, and EMR). Additional compute plane options will be added over time.



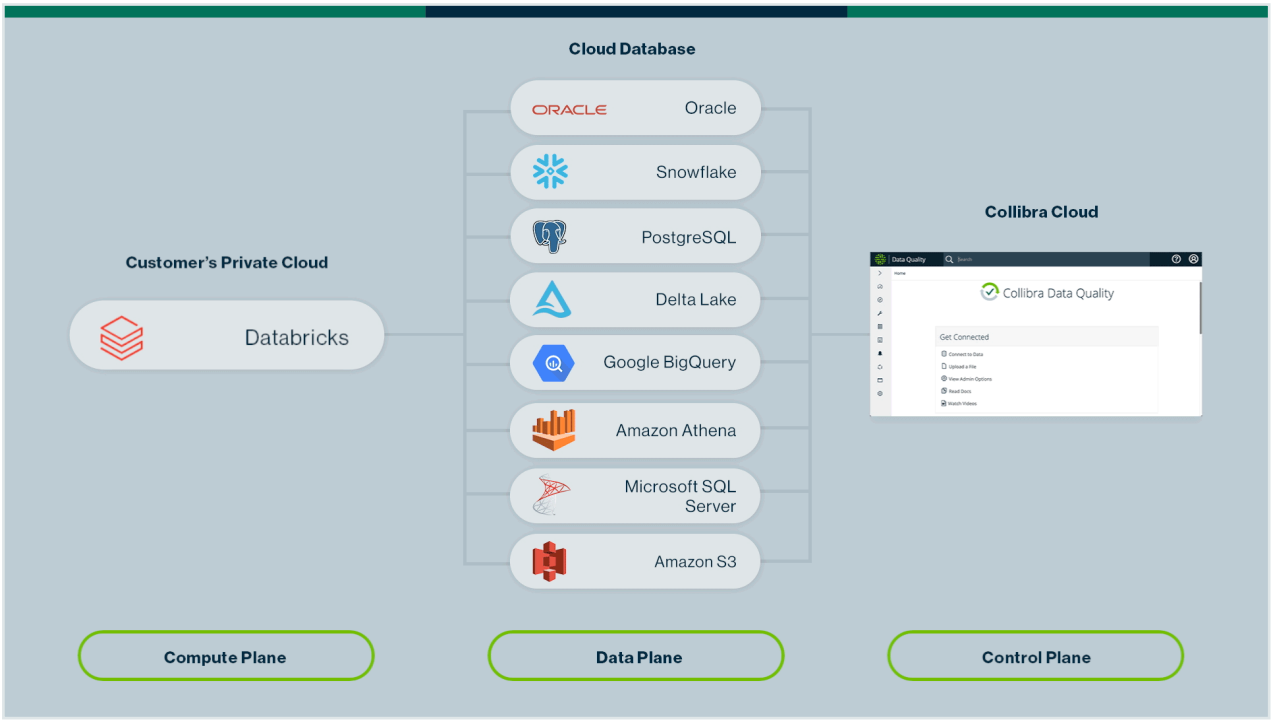
Pushdown DQ Control to Data

As of Q3 2022, Running a Snowflake Pushdown job is in private beta for select customers. Some features may be limited.



Databricks to Data to DQ Control

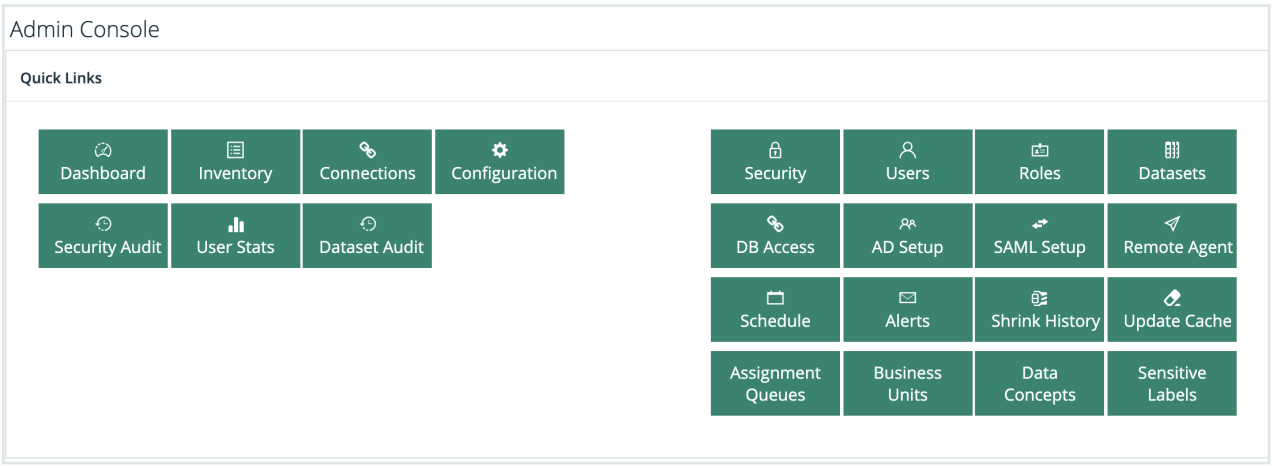
Running DQ jobs from Scala and Pyspark notebooks is generally available.



Chapter 17

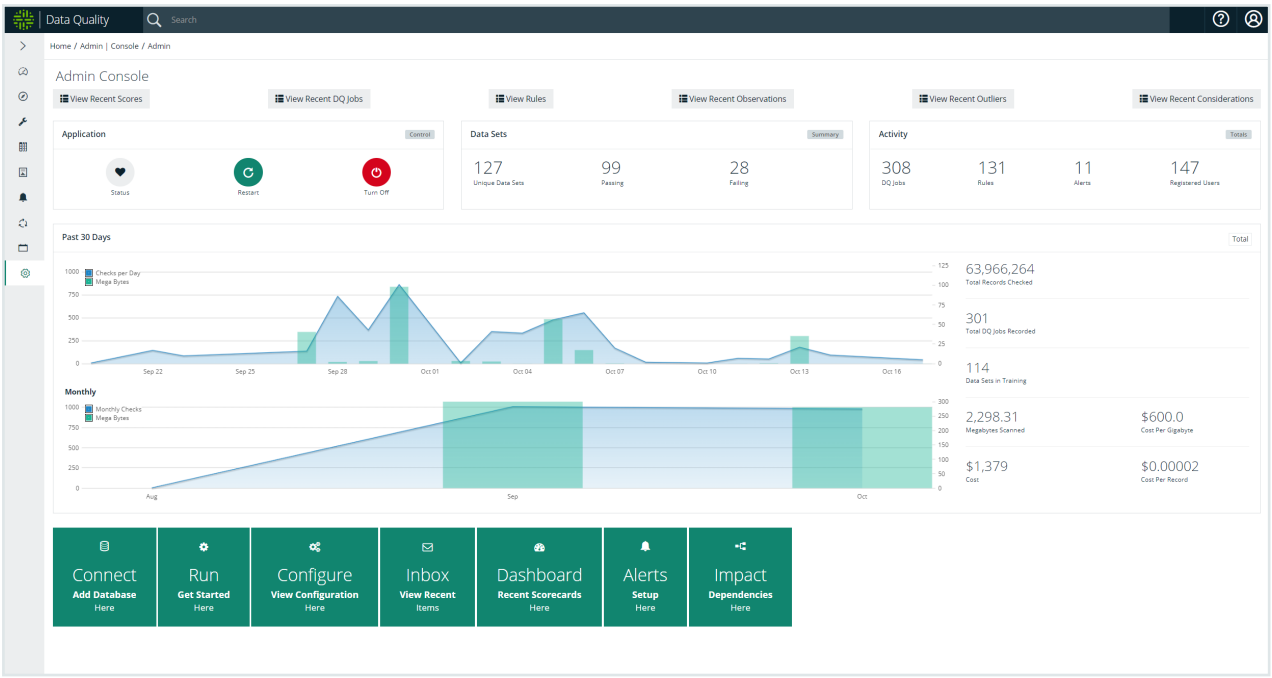
Collibra DQ Admin

Overview



Dashboard, Inventory, Connections, and Configuration are shortcut links to different operations within Collibra Data Quality & Observability.

- Dashboard
  - Shows an overall number of DQ Checks scanned the MB's, total jobs, total records scanned.
  - Total number of data sets, including passing and failed jobs (quality fails), number of alerts, and number of rules.
  - A list of messages on specific DQ Checks and what was found.



- Inventory
  - All DQ Jobs ever executed including Run Date, Data Set, Command Line, Type, Query, and Connection.
- Connections
  - Connect to a data source via the Admin Console.
- Configuration
  - Apply administrative override limits. For example, you can limit the amount of DataShapes recorded on a specific data set with the `datashapelimit` field.
  - If a duplicate record is found have a 1 point (negative) score per record. Allows administrators to increase or decrease the impact of different DQ issues found.
  - Display limits in order to improve performance of the UI.
- Audit Trail
  - List all security related changes by user, action, description and timestamp in a searchable, sortable table.

Configuration

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
```

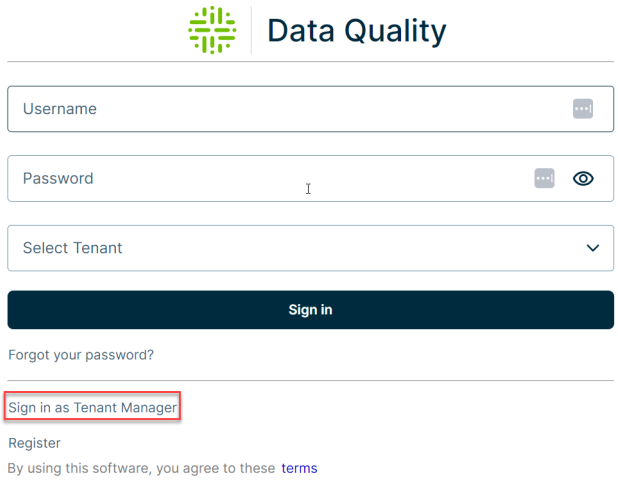
For more information about using Helm charts, go to **Collibra DQ Helm chart** in Cloud native.

Adding, editing, and deleting a tenant

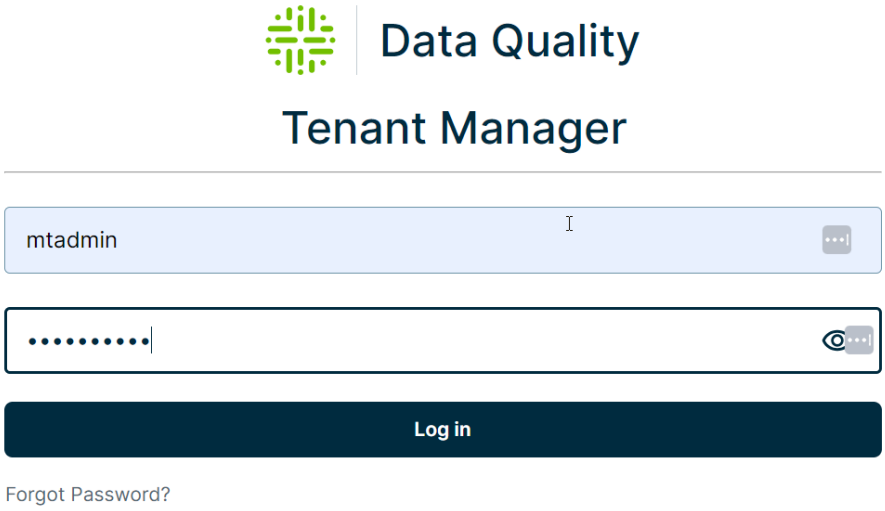
Regardless of the need for isolated environments, Collibra DQ supports dynamically created tenants, as part of the DQ web application. You can add, edit, or delete a tenant on the Tenant Management page.

Steps

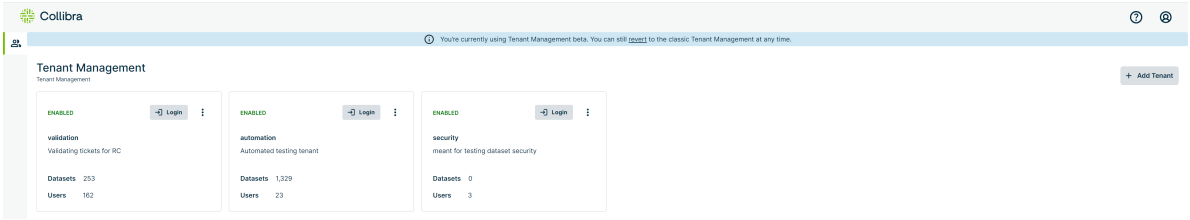
1. Sign into Collibra DQ.
2. Click **Sign in as Tenant Manager**, which appears below the **Forgot your password?** link.



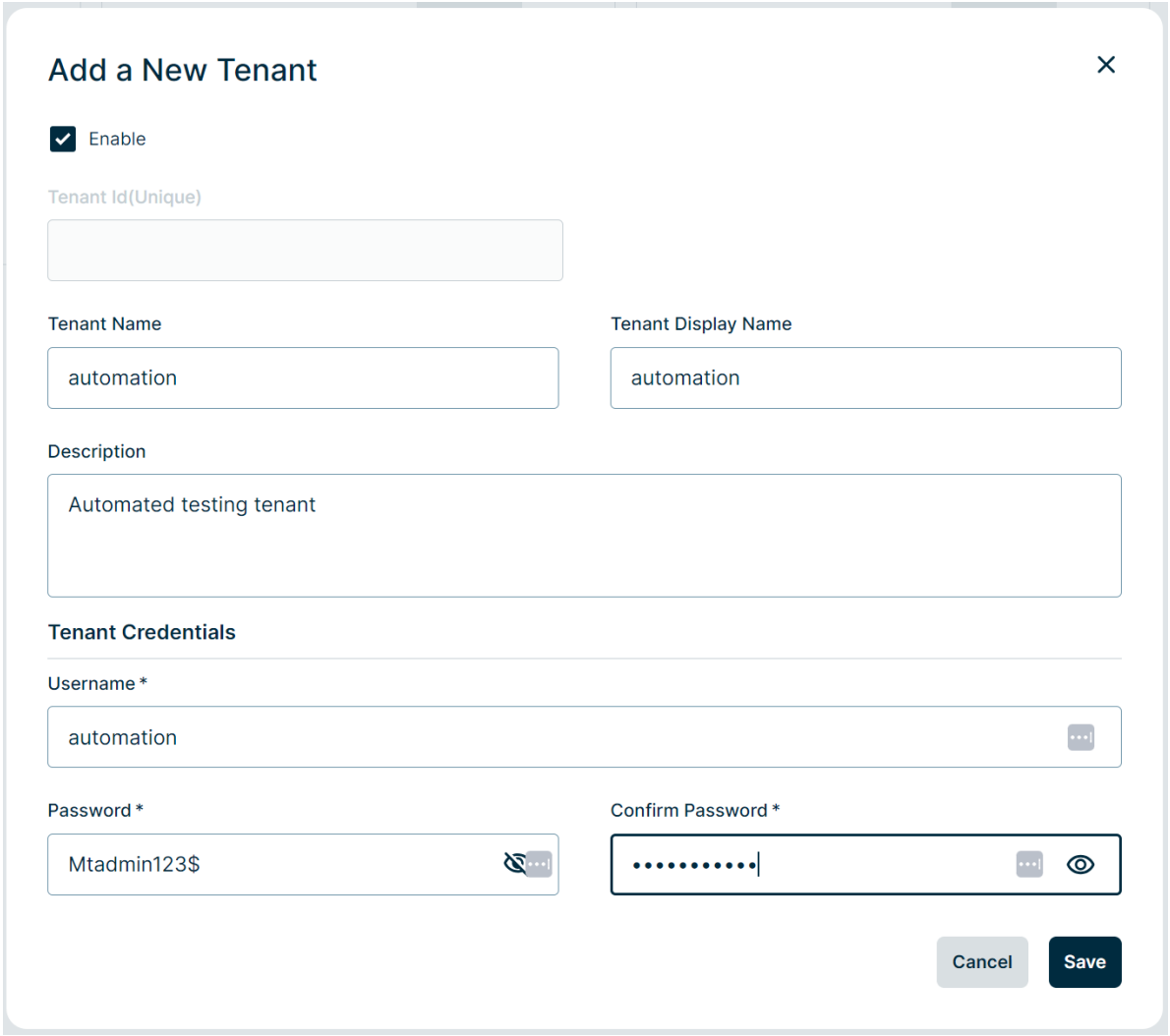
3. On the Tenant Manager login page, enter the following credentials:
   - Username: **mtadmin**
   - Password: **mtadmin123**



4. Click **Log in**.
   » The Tenant Management page opens, where you can add, edit, and delete your tenants.



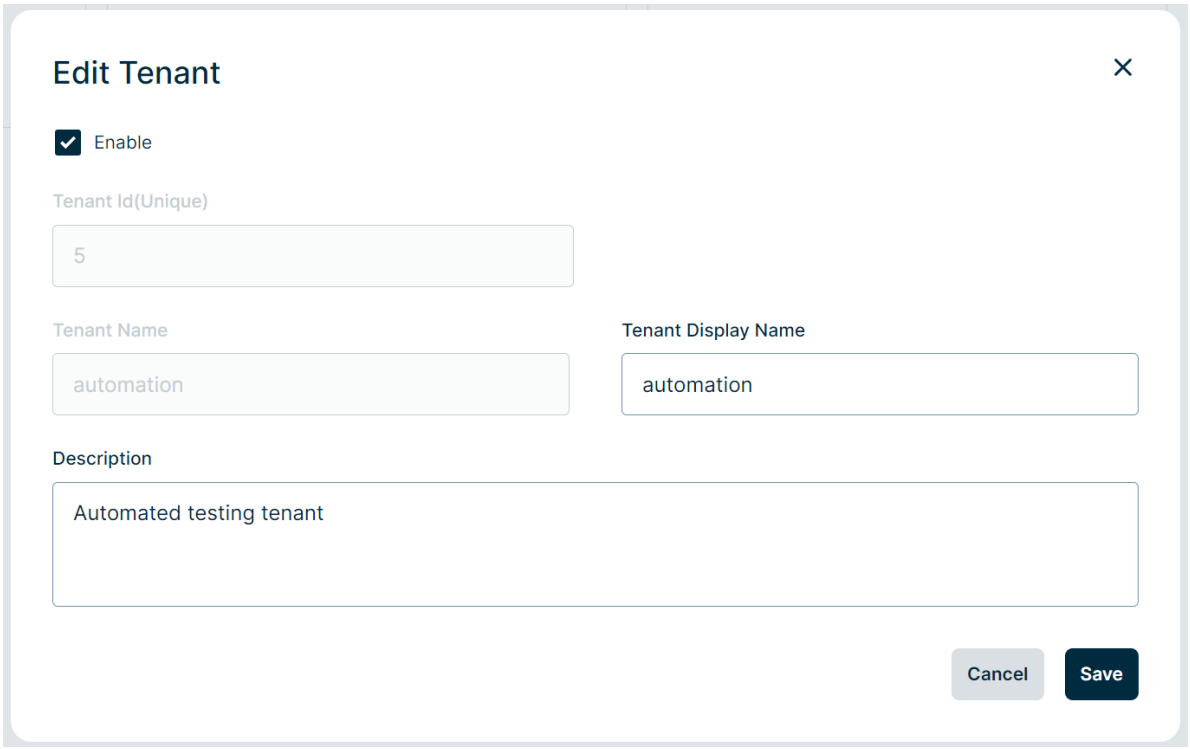
5. To add a new tenant, click the **+ Add Tenant** button in the top-right corner of the Tenant Management window.
   » The Add a New Tenant modal appears.



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**.

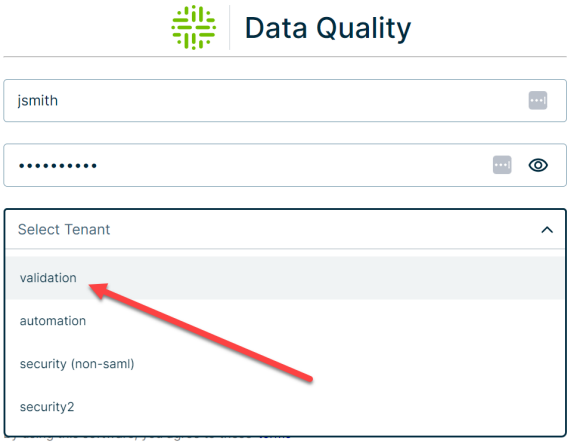
Signing into a tenant

You can select a tenant from a dropdown list on the main Collibra DQ login page.

> Note You can also login to a specific tenant from the Tenant Management page by clicking the **Login** button for that tenant. 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.

Time-Based Data Retention

Setting up Retention Based Data Purge

Retention based purge of data can be turned on to allow data to automatically be cleaned based on an organization's data retention policy.

Benefit

Once enabled, what type of data is removed?

- data_preview *(Drill-in records for rules, outliers, shapes, etc.)*
- dataset_field *(profiling stats)*
- rule_breaks *(Rule Exception records)*
- dataset_scan *(Job Ledger)*

Setup

In order to set up retention based data purge, three (3) environment variables need to be set up in the owl-env.sh configuration script. Note: a restart of the webapp is required for this configuration to take place.

- **cleaner_retention_enabled**
  - TRUE or FALSE on whether this feature is enabled
- **cleaner_retention_days**
  - Number of days to retain data
- **cleaner_retention_field**
  - Controls which field to use to select eligible data set runs
  - Potential values
    - updt_ts: consider the last time a data set run was updated
    - run_id: consider the run id field of the data set

Configuration

Example configuration in owl-env.sh

Organization wants to purge data where the updt_ts is more than 1 year old

In owl-env.sh, add the following lines

```
export cleaner_retention_enabled=TRUE
export cleaner_retention_days=365
export cleaner_retention_field="updt_ts"
```

Config Map

```
autoClean: "false"
cleaner_retention_days: "180"
cleaner_retention_field: updt_ts
cleaner_retention_enabled: "true"
```

Defaults for Auto Clean Process

> Note This is a separate rolling purge that is distinct **time-based retention**. This is **on by default** and uses the predefined limits below. You will see audit records for this clean-up process in Audit History of the Admin Console.

Separate from the time-based retention there is also a default auto clean mechanism that actively purges your old records. This is enabled by default and can be modified by use of the autoClean (AUTOCLEAN) boolean parameter.

```
AUTOCLEAN=false or autoClean="false"

### Depending whether this is part of owl-env.sh
### or the configMap of the web pod
```

These are the defaults. The row count threshold is the global limit when this is triggered. This is based on the records in the data_preview table. The runs threshold and the dataset per row threshold are data set-level limits that require a data set to have at least 4 scans and at least 1000 rows.

This is an example using the owl-env.sh file to control these settings.

```
export AUTOCLEAN=true
export DATASETS_PER_ROW=1000
export RUNS_THRESHOLD=4
export ROW_COUNT_THRESHOLD=200000
```

For example (using the settings above):

When data_preivew table has 200k rows
Look for data sets with 1000+ rows in data_prevew table
And have at least 4 scans
Then delete the oldest scan for those data sets

Auto clean and time-based retention run on a routine thread that triggers while the web application is running. It looks for clean-up candidates every few minutes when AUTOCLEAN=true or cleaner_retention_enabled=TRUE.

Set up SMTP

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.

Steps

1. Sign into Collibra DQ.
2. Select **Alerts**.

   > Note You can also access the Alerts menu via the Admin Console.

3. Configure the SMTP server for alerts in the following input fields. All fields are

required.

Option	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 (check box)	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 Configure Application Settings. Important Be aware that some email servers, such as Microsoft Exchange, require the **SMTP username** and **Reply email** fields to be the same.

4. Click **Save**.

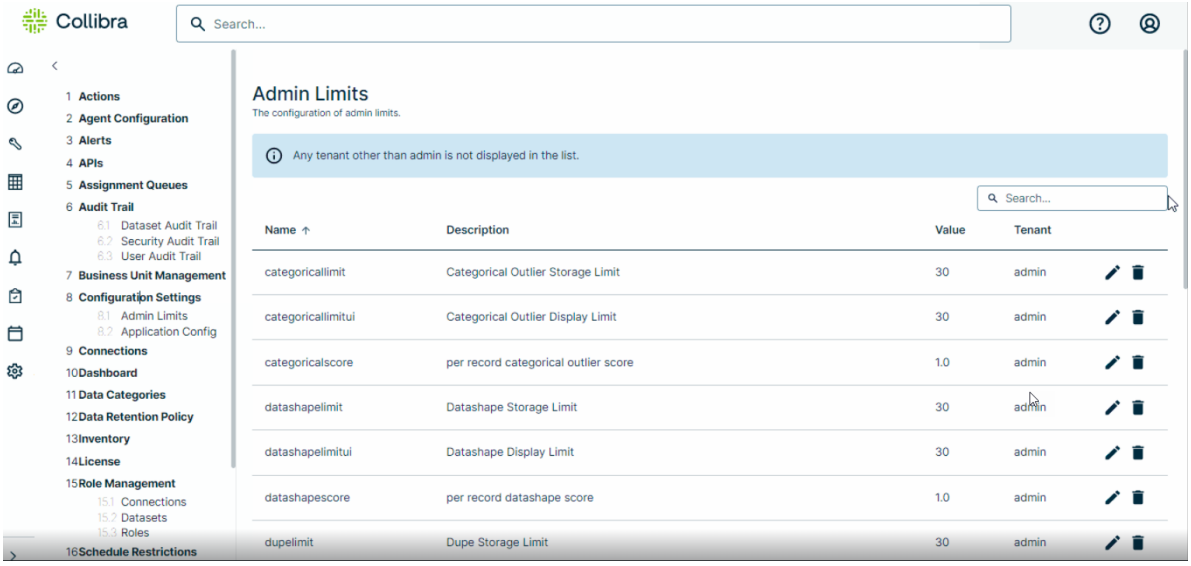
For more information about DQ alerts, go to Working with the Alert Builder and Configuring Alert Emails.

Configure Admin Limit Settings

You can configure Admin Limit settings values in the Admin Console.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Click **Admin Console**.
3. Select **Configuration Settings** in the left navigation panel.
4. Select **Admin Limits**.



> 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.

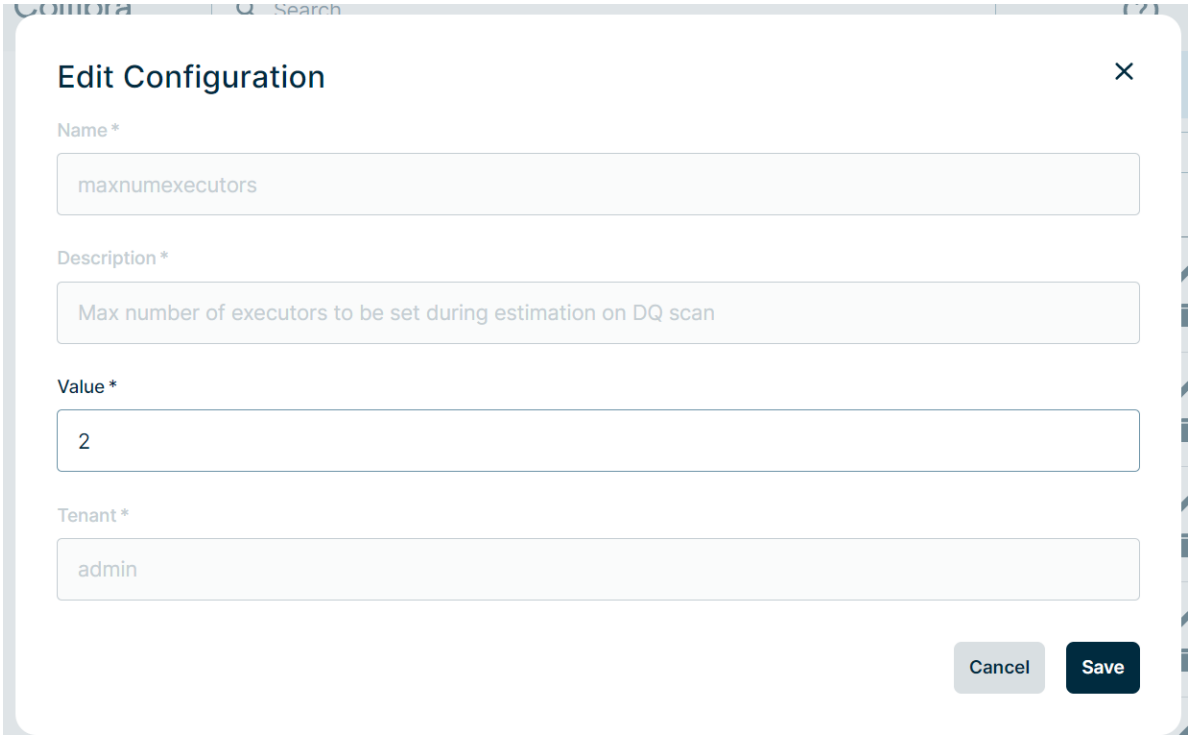
Setting	Description
categoricallimit	Limit of categorical outlier storage findings that get stored in the database.
categoricallimitui	Limit of categorical outlier findings that display in the UI.
categoricalscore	Default categorical outlier score, per record.
datashapelimit	Limit of datashape storage 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.
maxcellcountforparalleljdbc	Maximum number of cells for a DQ scan before it is recommended to use parallel JDBC, which gets stored in the database.

Setting	Description
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.
maxexecutormemory	Maximum number of executor memory (GB) to be set during estimation on a DQ scan, which gets stored in the database.
maxnumexecutors	Maximum number of executors to be set during estimation on a DQ scan, which gets stored in the database.
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
schemascore	Default schema score, per record.
scorecardsize	Limit of scorecard size display findings that get stored in the database.
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.

5. To modify values for the admin limits settings, click the icon, next to the setting for which you want to change the value.

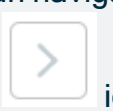
> Important If you click the icon, the application setting deletes immediately and no warning prompt displays.



> Note Settings with "score" in the setting name represent the default score for that setting's type of findings.

> Warning When you increase the UI display limits, the performance of the web application may decrease.

6. Click **Save**.

> Tip You can navigate the pages of the admin limits settings by clicking the and  icons, located bottom-right of the table.

To configure the application settings, click **Application Config** from Configuration Settings in the left navigation panel. For information about how to configure application settings, go to Configure Application Settings.

Add Custom Configuration Settings

This page shows how to add custom Admin Limit and App Config settings in the Admin Console.

- Add Custom Admin Limit Settings
- Add Custom Application Configuration Settings

Add Custom Admin Limit Settings

You can add custom Admin Limit settings from the Limit Settings page, in the Admin Console.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Click **Admin Console**.
3. Click **Configuration** on the Quick Links page.

   > Note You can also access the application configuration settings by clicking the ⚙ icon and selecting **Settings**.

4. Click **+Add Custom** at the top-right of the page and enter values in the input fields described in the following table.

Input Field	Description
name	Create a name for the custom admin limit configuration setting.
value	Enter a value for the custom Admin Limit setting.
admin	Enter a role.
value to limit invalid rows	Enter a value that limits the number of invalid rows.

5. Click **Add**.
   The custom Admin Limit configuration setting you created displays on the Admin Limits page.

Add Custom Application Configuration Settings

You can add custom application configuration settings from the Limit Settings page, in the Admin Console.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Click **Admin Console**.
3. Click **Configuration** on the Quick Links page.

   > Note You can also access the application configuration settings by clicking the ⚙ icon and selecting **Settings**.

4. Click ⚙ **App Config** at the top-right of the page.
5. From the App Config page, click **+Add Custom** at the top-right of the page and enter values in the input fields described in the following table.

Input Field	Description
name	Create a name for the custom App Config setting.
value	Enter a value for the App Config setting.

6. Click **Add**.
   The custom App Config setting you created displays on the App Config page.

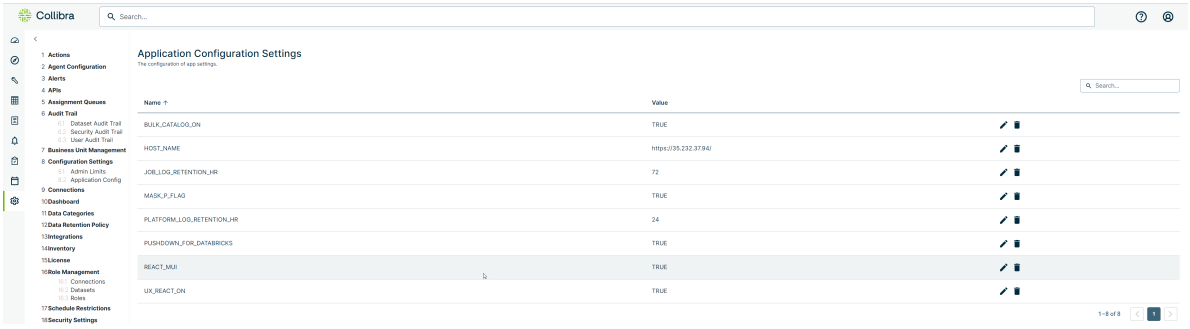
Configure Application Settings

From the Admin Console, you can configure the application settings values.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Select **Admin Console**.
3. Select **Configuration Settings** in the left navigation panel.
4. Select **Application Config**.
5. To modify values for the admin limits settings, click the ✎ icon, next to the setting for which you want to change the value.

   > Important If you click the 🗑 icon, the application setting deletes immediately and no warning prompt displays.



6. The following table shows the Collibra DQ app configuration settings that have been configured.

App Setting	Description	Value	Notes
BULK_ CATALOG_ON	When set to `TRUE`, allows/disallows using the catalog feature from the Wizard in the UI.	`TRUE/FALSE` Default: `TRUE`	

App Setting	Description	Value	Notes
HOST_NAME	Hostname of your URL.	Enter the host-name 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	When set to `TRUE`, enables the number of hours that the logs retain Collibra DQ Jobs.	Enter the num-ber of hours in the input field.	
MASK_P_FLAG	When set to `TRUE`, sets/unsets `-p/-srcp` masking in the UI.	`TRUE/FALSE` Default: `TRUE`	
PLATFORM_ LOG_ RETENTION_ HR	When set to `TRUE`, enables the number of hours platform logging is retained.	`TRUE/FALSE` Default: `TRUE`	
PUSHDOWN_ FOR_ DATABRICKS	When set to `TRUE`, you can run Dat-abricks Pushdown Jobs.	`TRUE/FALSE` Default: `TRUE`	You must have a valid Dat-abricks connection with the Pushdown option, selected in the Connection template.
REACT_MUI	When set to `TRUE`, enables React MUI.	`TRUE/FALSE` Default: `TRUE`	As of the 2023.02 version, React is still in the process of full implementation across the Data Quality Collibra DQ app. As such, some pages are not yet available in the React layout.

App Setting	Description	Value	Notes
UX_REACT_ON	When set to `TRUE`, turns React MUI on. When `FALSE`, the Collibra DQ web application displays the legacy UI.	`TRUE/FALSE` Default: `FALSE`	

7. Click **Save**.

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

Batch Size	Agent Pod Memory
10	2 GB
20	4 GB
40	8 GB
80	16 GB
160	32 GB
320	64 GB

> 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 [[[Undefined variable DataQuality.DQEnv]]] 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 [[[Undefined variable DataQuality.DQEnv]]] file with the following command:

```
vi /home/owldq/owl/config/owl-env.sh
export DEFAULT_BATCH_ID=<your-b64enc-batch-id>
```

3. Use a base64 encoder, such as the free encoder from base64encode.org, to encode your batch ID and enter the encoded value after `DEFAULT_BATCH_ID`.

> Tip Your base64 encoded string should look similar to the following example:
>
> ```
> export DEFAULT_BATCH_
> ID=QUdFbVlYWnNkRDRpWVpiZXZGTnBlbVU5Tkw9PQ==
> ```

4. Enter the following command:

```
export DEFAULT_BATCH_SIZE=<your-batch-size>
```

> Tip Enter a value in `<your-batch-size>` that meets 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-b64enc-batch-id>"
--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-new-batch-size>
```

> Tip Your base64 encoded string should look similar to the following example:
>
> ```
> export DEFAULT_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?

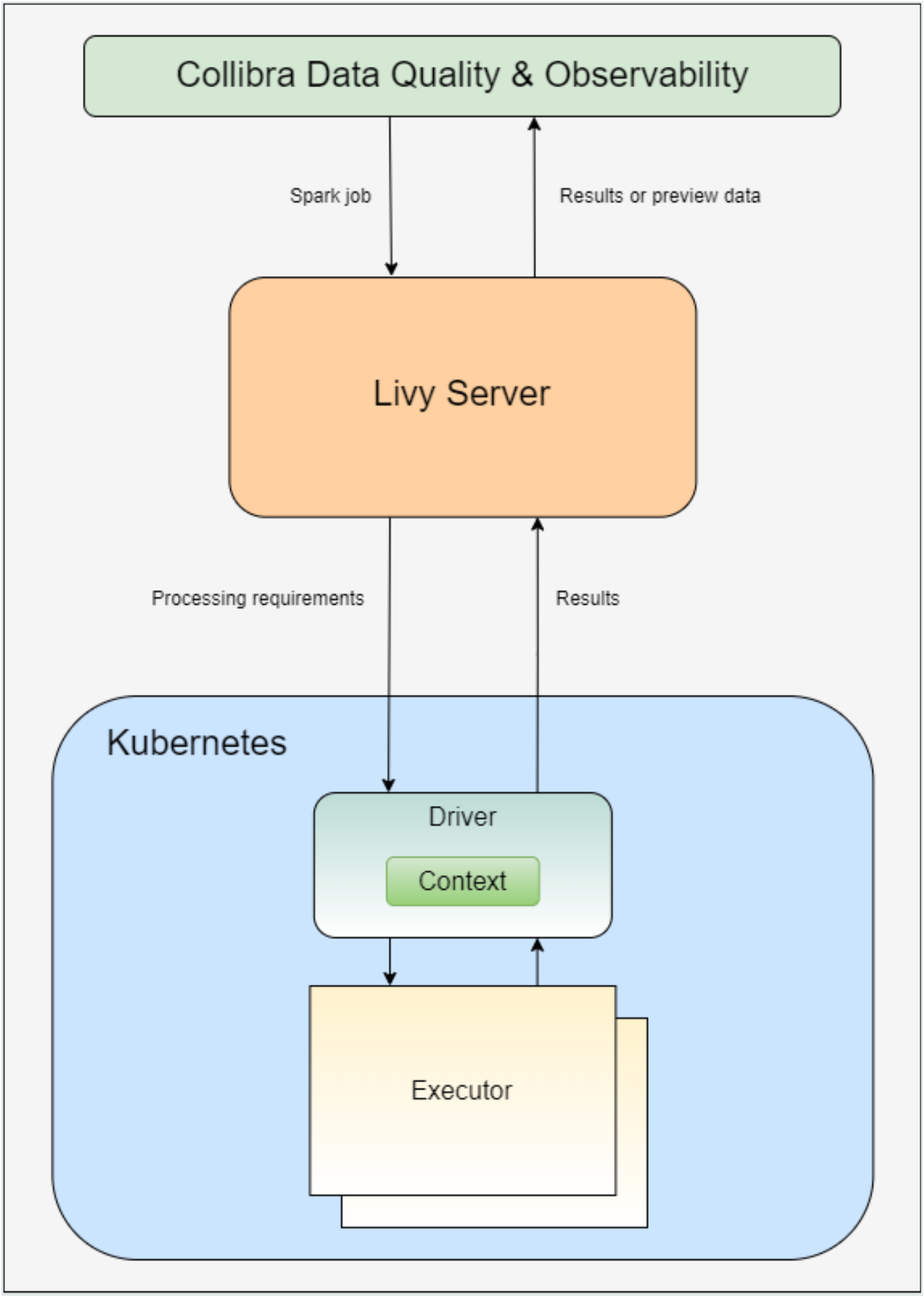
Apache Livy is a RESTful web service that serves as a session manager for Spark jobs on Kubernetes and Hadoop deployments of Collibra Data Quality & Observability. Livy creates temporary Spark pod sessions (Spark Contexts) to efficiently run jobs in the Spark cluster. Collibra Data Quality & Observability 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 Customer Success Manager 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. Collibra Data Quality & Observability 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 Collibra Data Quality & Observability.

Supported deployment types

Livy is currently supported on Kubernetes deployments of Collibra Data Quality & Observability.

> 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

Connection	Supported	Estimate Job
Network File Storage		
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**.

Advanced

Allow/Disallow Using the catalog feature from Wizard: BULK_CATALOG_ON = **TRUE**/FALSE

Preset Wizard Config for showing views: DB_VIEWS_ON = **TRUE**/FALSE

Preset Wizard Config for showing stats: DB_STATS_ON = **TRUE**/FALSE

Set File Search Path From Wizard: UPLOAD_PATH = <Path on Web Server>

Set/Unset -p/-srcp masking in UI: MASK_P_FLAG = **TRUE**/FALSE

Allow/Disallow Zeppelin Notebooks: ZEPPELIN_ENABLED = **TRUE**/FALSE

Allow/Disallow Using the orient DB: ORIENT_ENABLED = **TRUE**/FALSE

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>
```

> 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>
```

> Note For more information, please review Collibra's Subprocessors page.

Usage

> Note You must have admin privileges to access usage metrics.

The Usage page lets you analyze your monthly usage statistics from the Admin Console. Key monthly metrics tracked on the Usage page include:

- Total number of users.
- Total number of DQ Jobs run.
- Total number of rules applied.
- Total number of data sets.
- Total number of columns.

datasets	jobs_passing	users	distinct_columns	jobs_submitted	jobs_unknown	license_key	rules_active
293	47	250	7799	2	3	[redacted]	445
6979	0	139	165519	0	0		6457
298	588	9	4650	20	115		0
1477	527	40	58491	86	12		
233	206	24	6029	170	18	[redacted]	317
1478	556	40	58500	88	12	[redacted]	1820

Viewing the Usage page

To view the Usage page

1. Click the gear icon and then click Admin Console.
   >> The Admin Console opens.
2. Click the Usage tile.
   >> The Usage page opens.
3. Analyze your usage statistics if they are available.

Collibra DQ License Management

Enable the Collibra DQ License Option

To enable the License option in the Collibra DQ web application, you must switch to the React MUI. To do this, you must set the UX_REACT_ON parameter to `TRUE`. Keep in mind, however, that once you have updated your Collibra DQ license, you must set the UX_REACT_ON parameter back to `FALSE`, to ensure proper functionality of Collibra DQ.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Select **Admin Console**.
3. From the Quick Links page, select the **Configuration** tile.

   > Note You can also access the application configuration settings by clicking the ⚙ icon and selecting **Settings**.

4. Select ⚙ **App Config** at the top-right of the window.

5. Set the UX_REACT_ON parameter to `TRUE`.

> Warning The UX_REACT_ON parameter must be set to `FALSE` for the Collibra DQ web app to functional properly. Therefore, when you have completed your license management, you must ensure this parameter has been set back to `FALSE`.

6. Click **Save**.
7. Manage your license accordingly. Go to Manage Your Collibra DQ License to see the License options.
8. To set the UX_REACT_ON parameter back to `FALSE`, from the Admin panel, select **Application Config** in **Configuration Settings**.
9. Click the edit icon and set UX_REACT_ON to `FALSE`.
10. Click the save icon.
11. In the App Config settings, ensure that UX_REACT_ON is set to `FALSE`.

Manage Your Collibra DQ License

This section shows you how to manage your license information provisioned by Collibra.

Prerequisites

You have:

- Admin permissions.
- 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.

> Note The License option displays only when the UX_REACT_ON parameter is set to `TRUE`. Go to React for more information.

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.

React

Switching React MUI on or off

React MUI is on by default. You can switch React MUI on or off.

> Note As of the 2023.02 version, React is still in the process of full implementation across the Collibra DQ app. As such, some pages are not yet available in the React layout.

Steps

1. Hover your cursor over the ⚙ gear icon and select **Settings**.
   » The Limit Settings opens.
2. In the upper-right corner of the Settings page, select **App Config**.
   » The App Config opens.
3. Enter the required information.

Option	Description
REACT_MUI	This flag controls whether React is switched on or off in the Collibra DQ app. You can: ◦ Enter `TRUE` to switch the React MUI on for all available React pages. ◦ Enter `FALSE` to switch the React MUI off.
UX_REACT_ON	This flag must be set to **FALSE**. Warning UX_REACT_ON must be set to `FALSE` to ensure proper functionality of Collibra DQ. If you want to view pages in the React layout, set the REACT_MUI option to `TRUE`. To enable the **License** option in the Collibra DQ web application, you must set this parameter to `TRUE`. Keep in mind, however, that to ensure proper functionality of Collibra DQ, you must set the parameter back to `FALSE`, after you have updated your DQ license. Go to Collibra DQ License Management for more information.

4. Click **Save**.

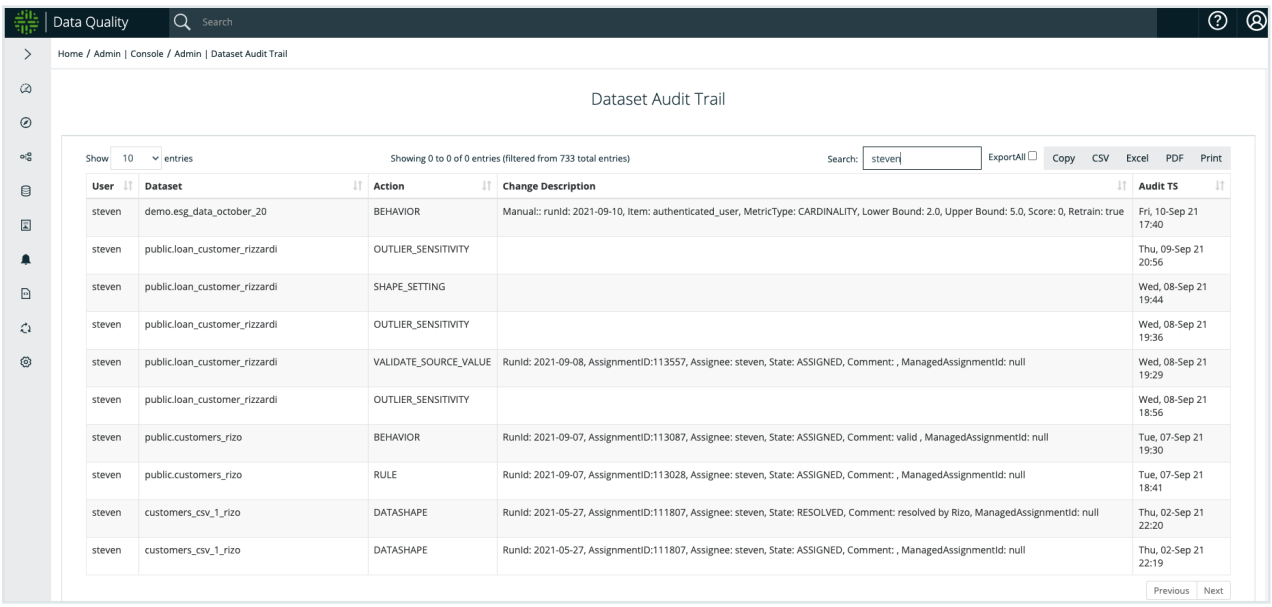
Audit

Built-in auditing allows you to track usage and modifications across data set, security, and user levels.

Dataset Audit Trail

Available Data

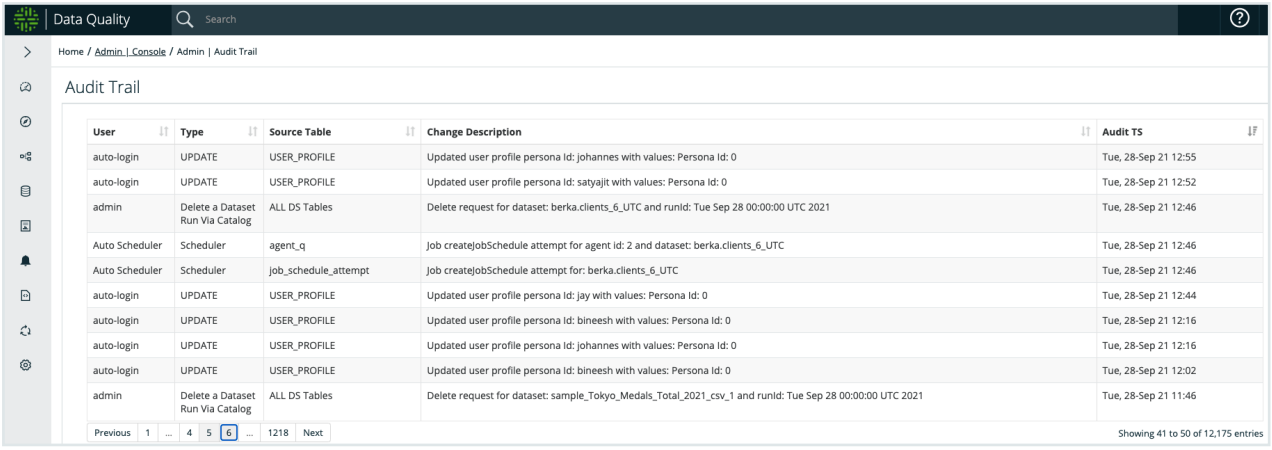
- User
- Data Set
- Selected Features
- runID
- Assignments
- Comments
- Timestamp



Security Audit Trail

Available Data

- User profile updates
- Role updates
- Reference table from metastore
- Data set deletion requests
- Job schedule attempts



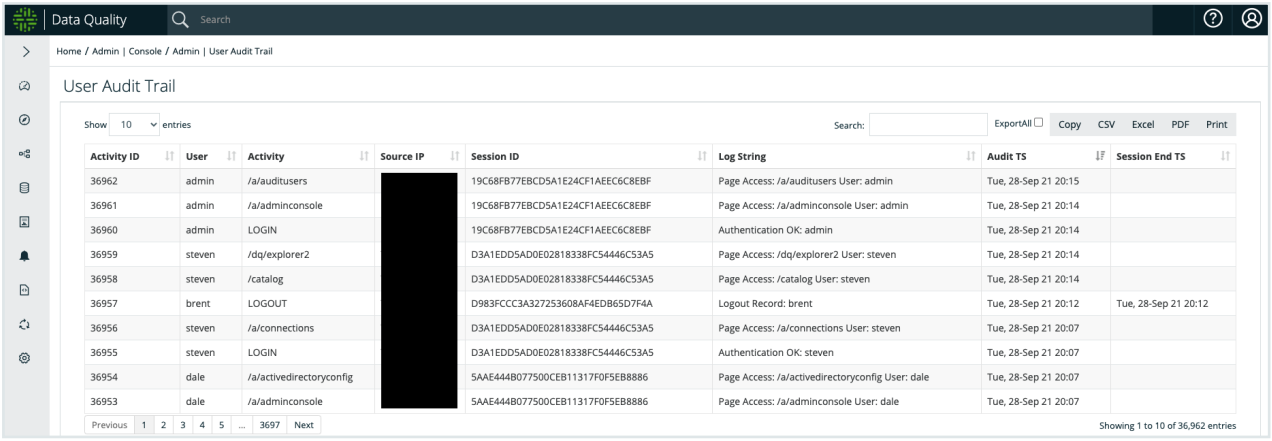
When administrators modify roles mapped to data sets or data sets mapped to roles, changes are documented automatically in the Audit Trail. The information in the entry log includes

- New and original data sets 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 modification.
- Username by which the modifications are made.

User Audit Trail

Available Data

- Logins: Successful / Failed
- Privileged User Access
- User Actions / Activities
- Account Actions
- Source IP
- Timestamps



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.

Column	Description
Page	The page in Collibra DQ from which the request was sent. **/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.
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.

Chapter 18

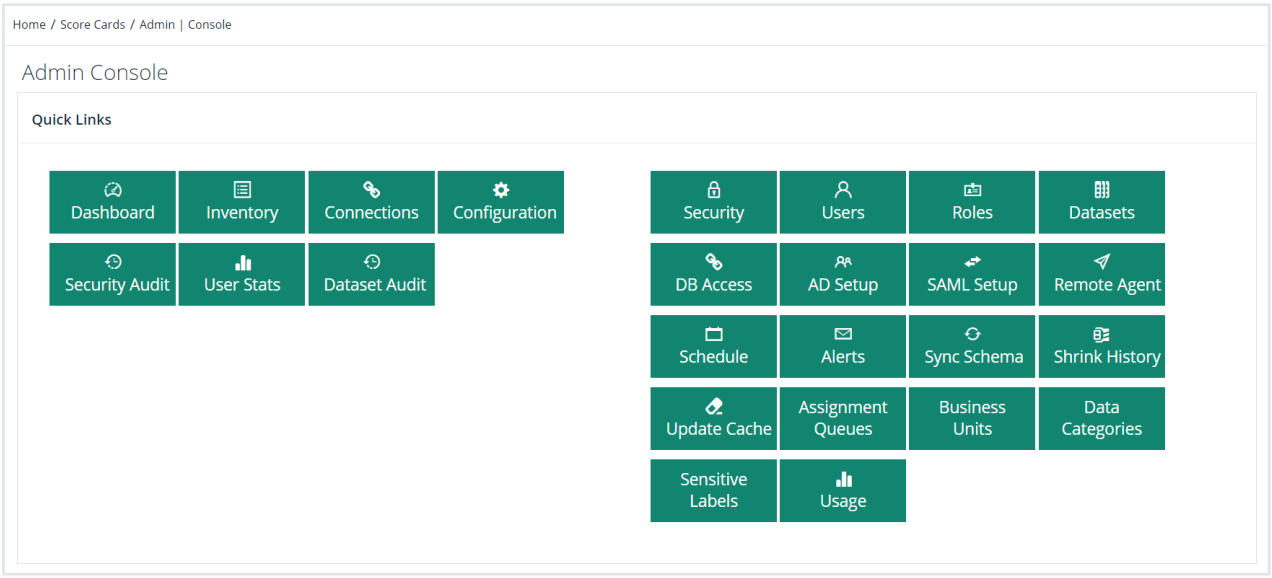
Collibra DQ Security Configuration

Overview

Collibra Data Quality 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), Postgres, and Application layers depending on your security requirements. Encryption is available for data in-transit or at-rest.



Configuration

Configuring Security Settings (AD/LDAP)

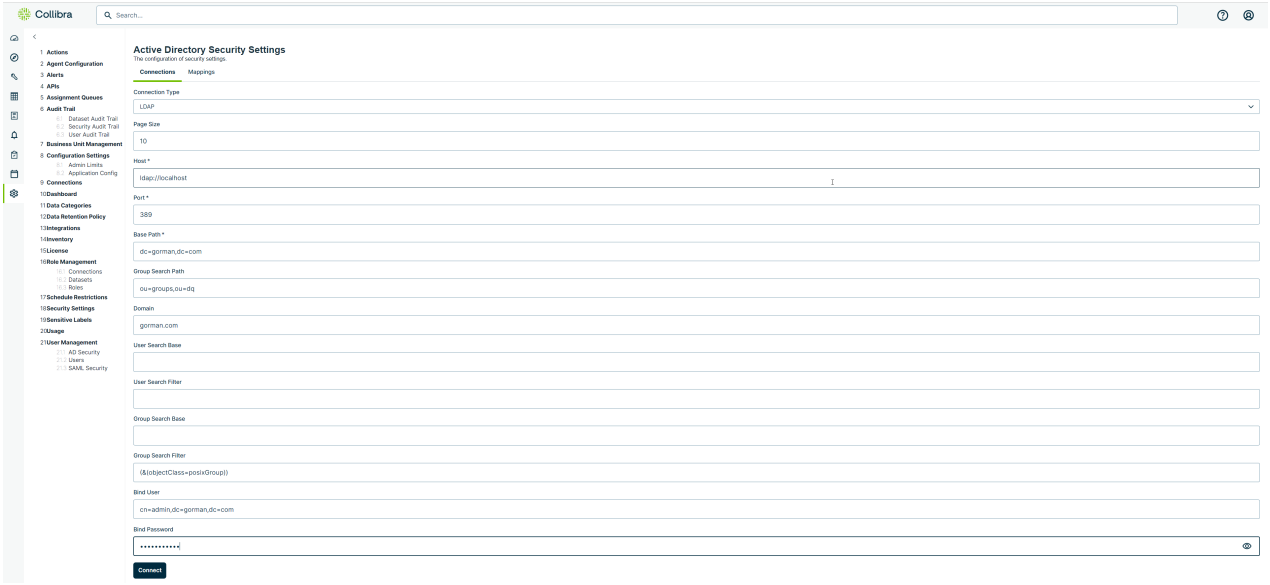
You can configure security settings from User Management in the left navigation panel.

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.

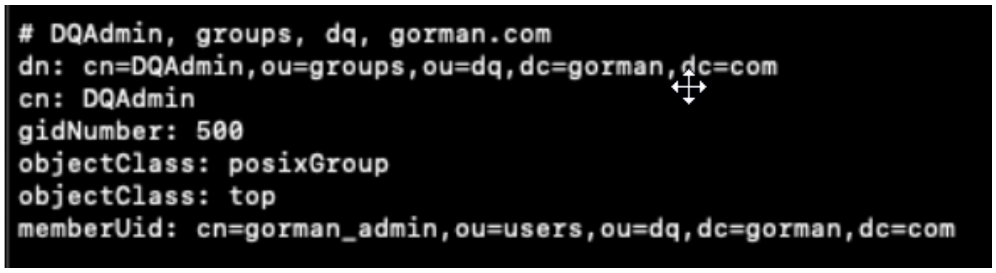
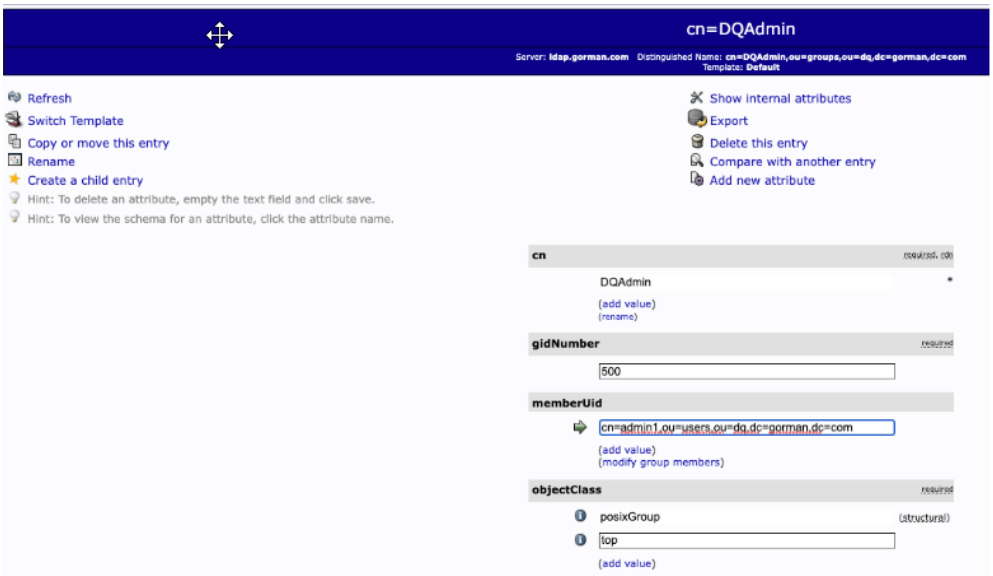
Note An asterisk (*) next to a setting represents a required input field. A restart of the Collibra DQ web application may be needed for values to take effect.

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.

[[[Undefined variable DataQuality.DQEnv]]] 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**.

[[[Undefined variable DataQuality.DQEnv]]] 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.

[[[Undefined variable DataQuality.DQEnv]]] 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

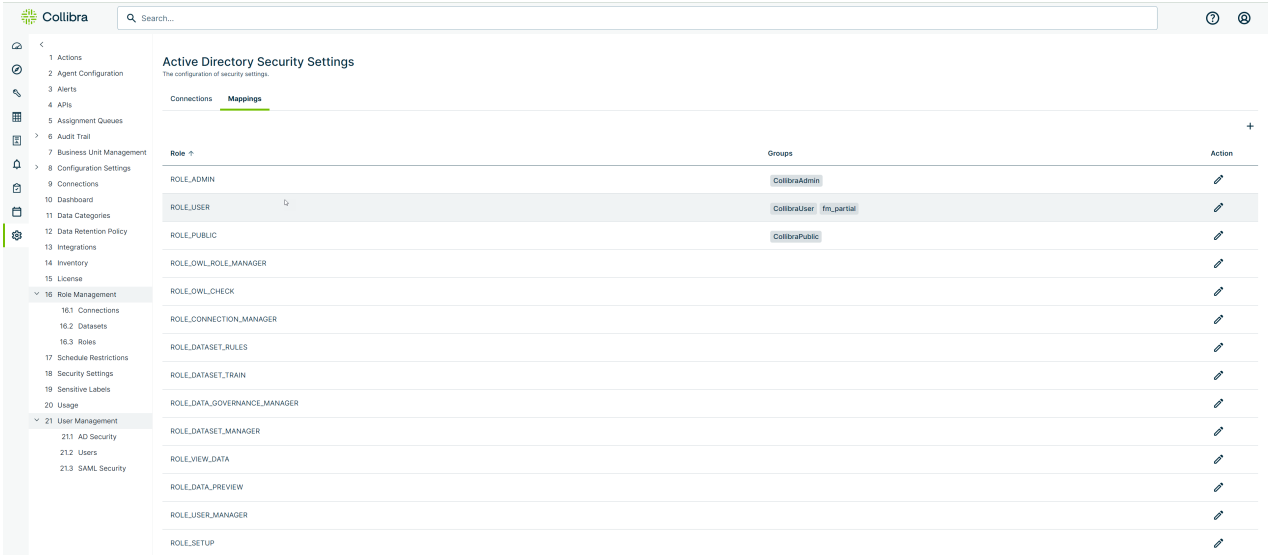
When you map an AD Group to a Collibra DQ role, you grant role-based access to all users from the selected AD Group to the selected Collibra DQ role outlined in the steps

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

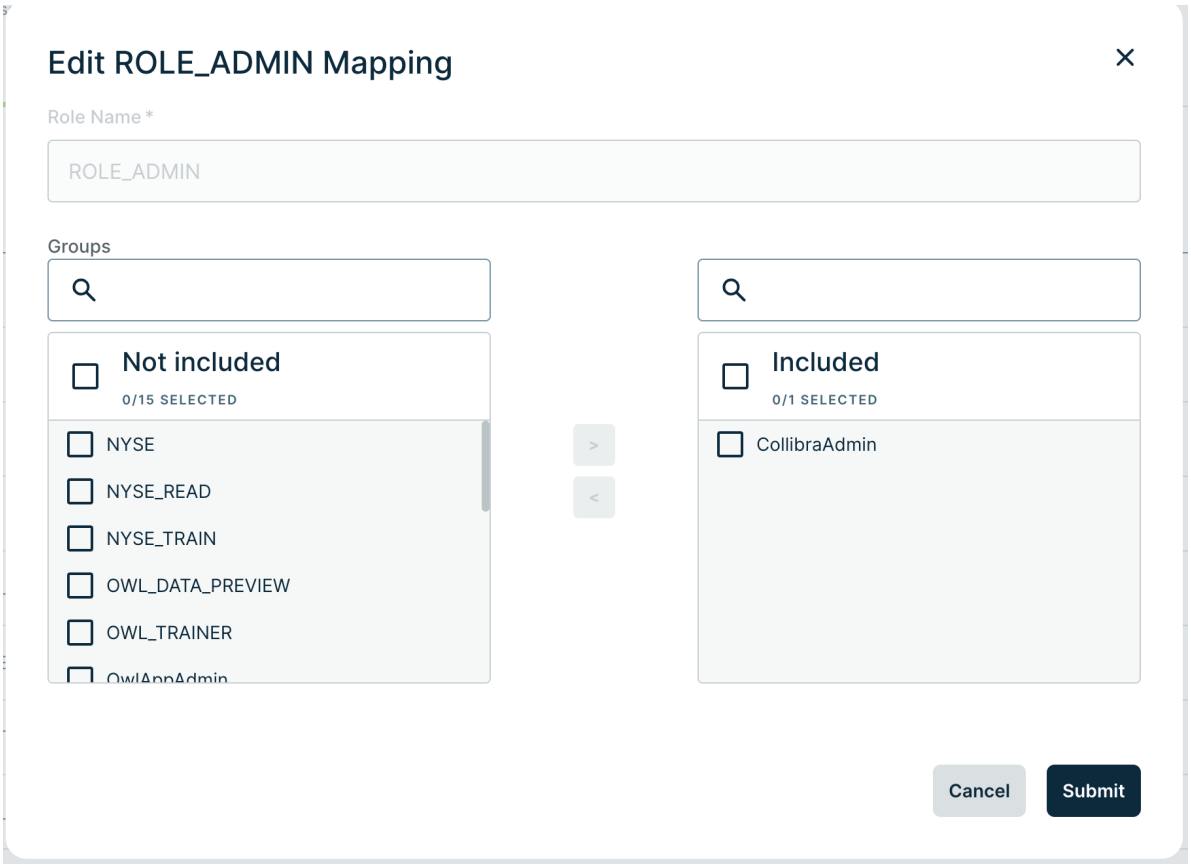
1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Click **Admin Console**.
3. Click **User Management** and then click **AD Security**.
4. Ensure the connection type you want is selected in the **Connections** tab.

   > 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.

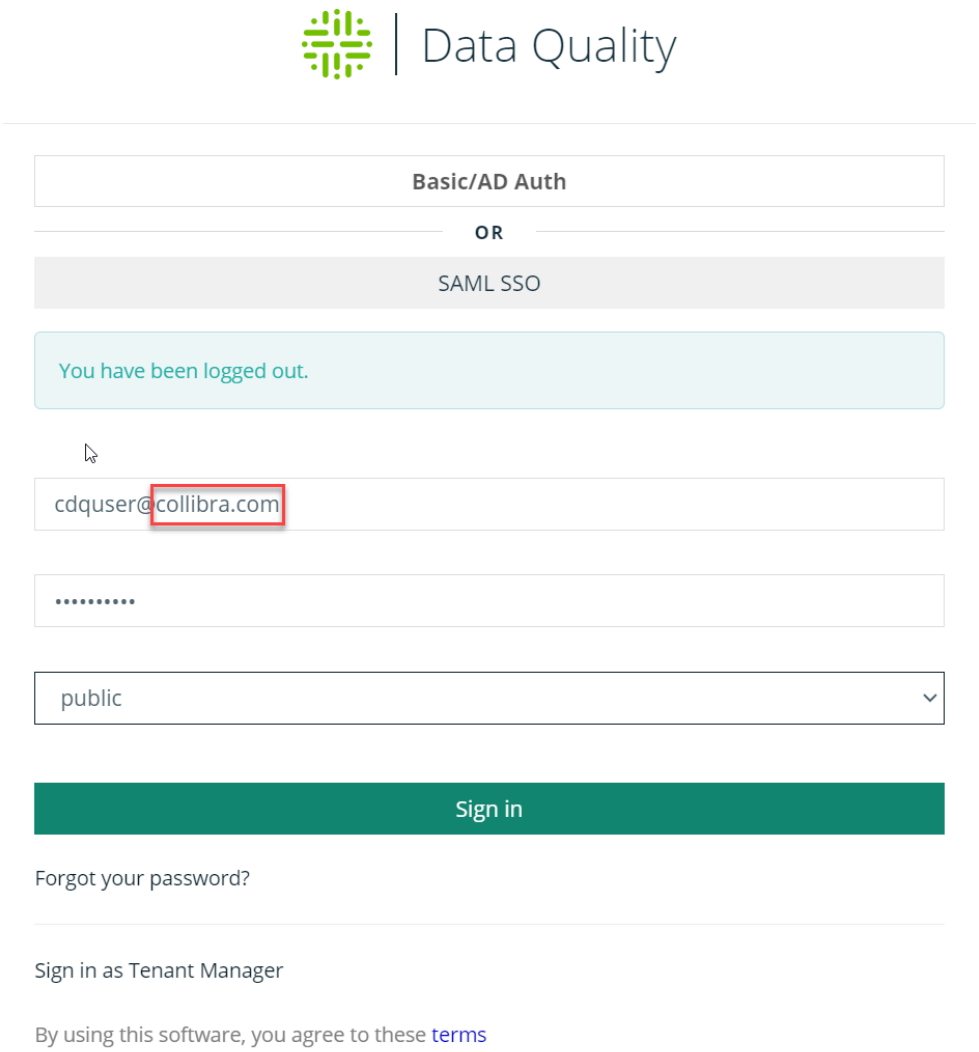


7. Click **Submit**.

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` for Standalone or restart the Web Pod for Cloud Native.

When logging into the Collibra DQ web application, append the domain to the end of the username.



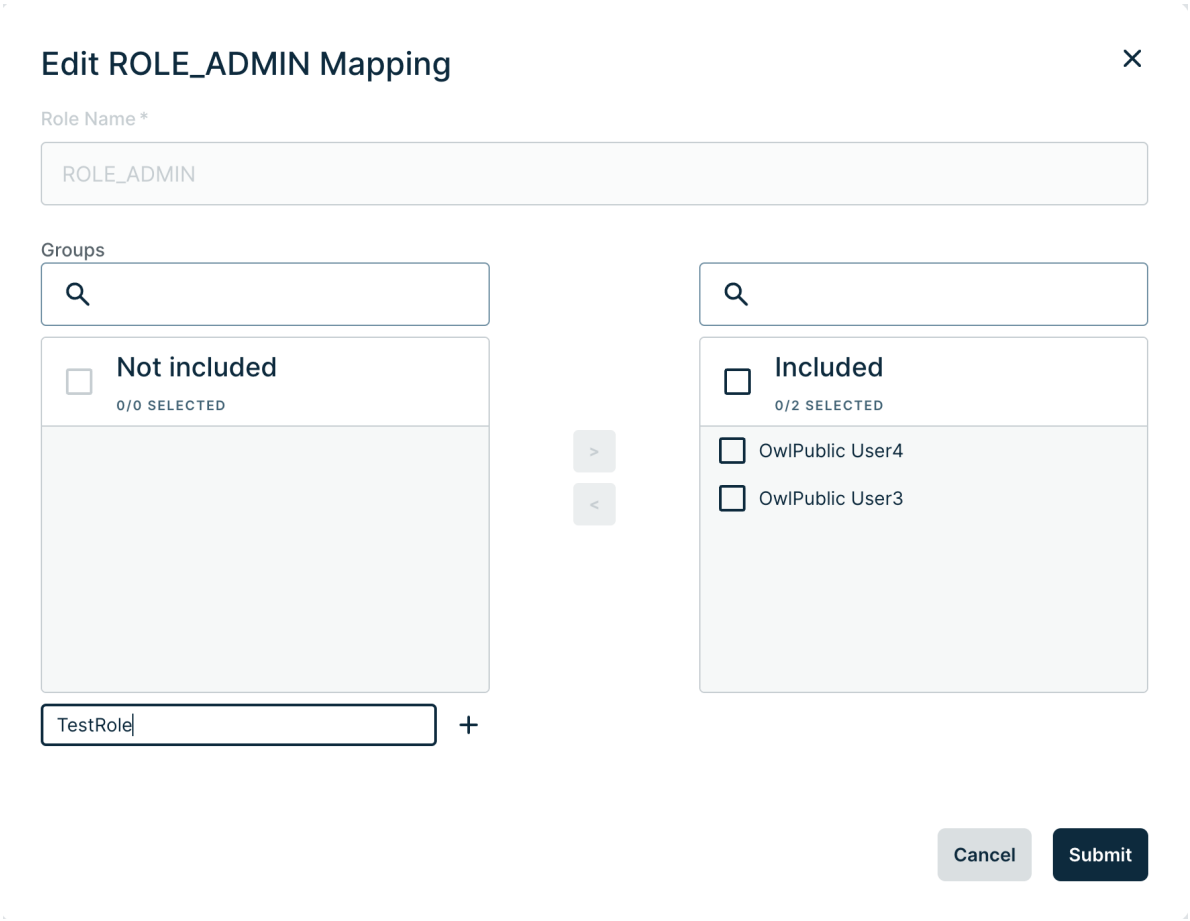
Group to Role Mapping Without AD

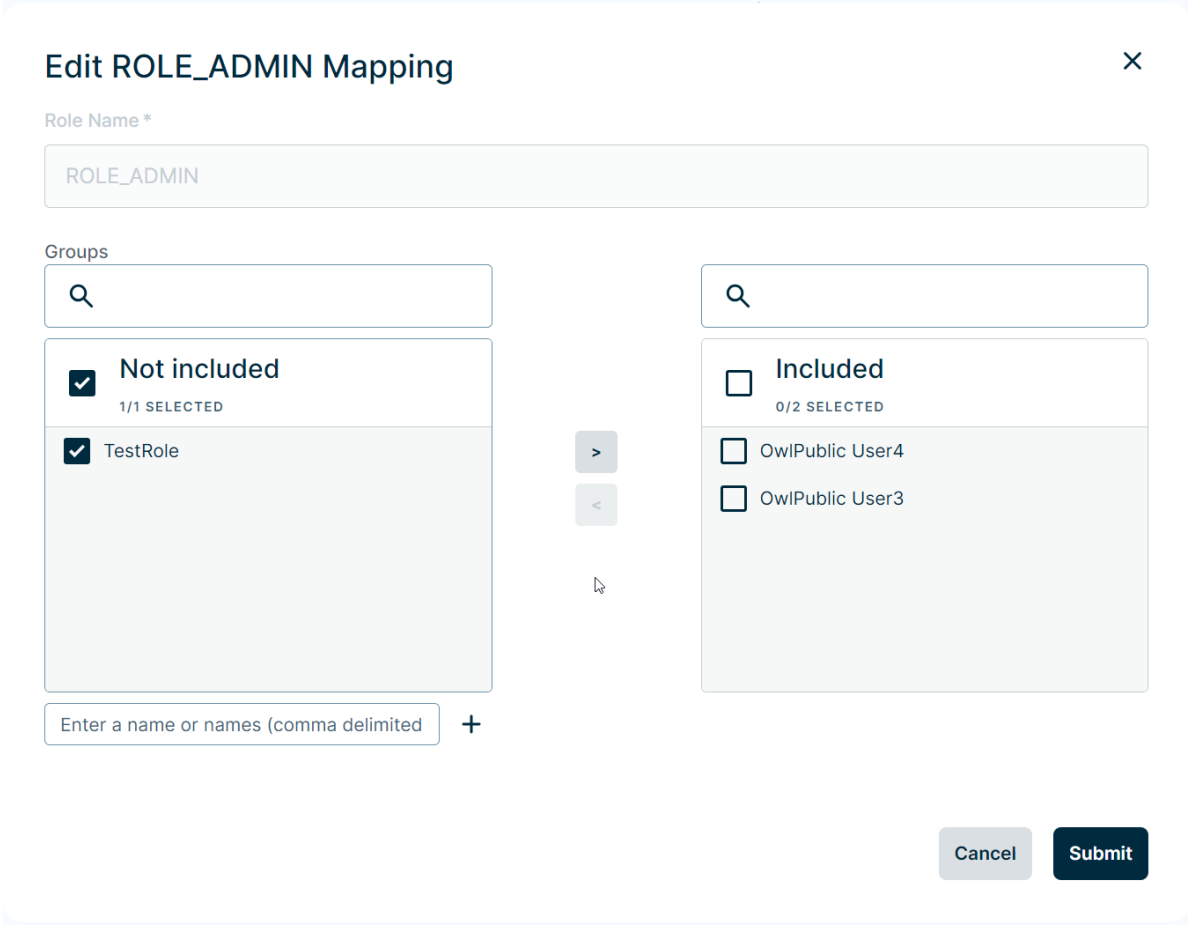
If you are configuring SSO with SAML, you can map your groups to Collibra DQ roles without Active Directory configuration.

> Note To enable SAML, go to SAML Authentication.

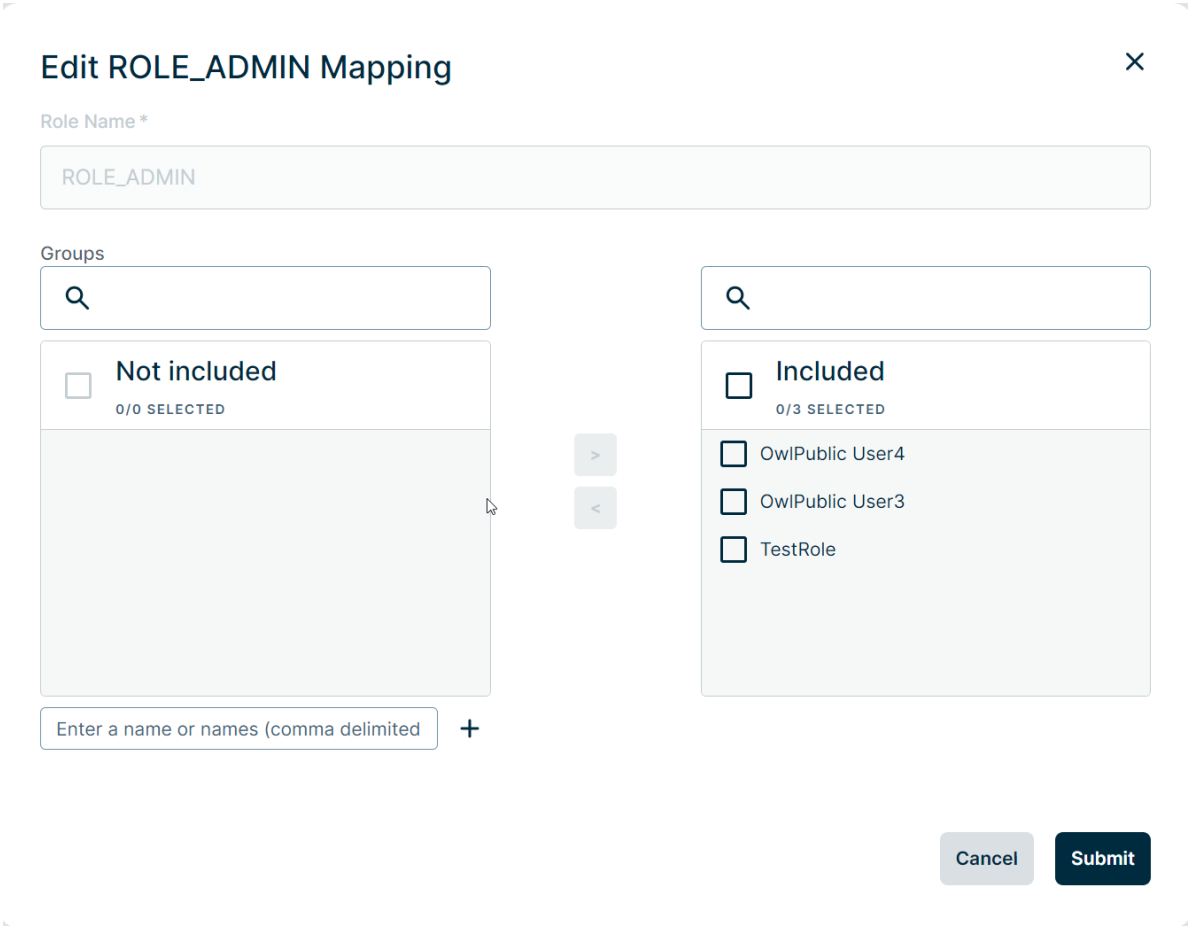
Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
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
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 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.

Setting Up User Search Configuration

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 Collibra DQ and click the ⚙ in the left navigation panel.
2. Select **AD Security**, from User Management in the left navigation panel.
3. From the Connections tab, select **LDAP** from the Connection Type dropdown.
4. 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).

5. 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).

6. 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.

Troubleshooting Common Issues When Configuring LDAP

You can use the web logs to troubleshoot issues LDAP configuration. Each step has logging to see what is being sent to the LDAP provider.

Examples

Issue: User is found but no groups are matching

DEBUG com.owl.web.security2.WebSecurityConfig - checking for roles by: cn

DEBUG com.owl.web.security2.WebSecurityConfig - Trying ldap search with Eq Filter: dc=gorman,dc=com, search filter: distinguishedname=cn=admin123456,ou=users,ou=dq,dc=gorman,dc=com, custom prop: cn

DEBUG com.owl.web.security2WebSecurityConfig - Filter: (distinguishedname=cn=admin123456,ou=users,ou=dq,dc=gorman,dc=com)

DEBUG com.owl.web.security2.WebSecurityConfig - Checking roles with attribute size: 0

Problem: In this case, you can see the filter being sent to LDAP. The filter is looking for a group object that has that user as the property distinguishedname.

Solution: Edit the LDAP_ROLES_PROP_NAME and LDAP_USER_SEARCH_ATTRIBUTE to the right value. After setting the values, this is the successful login.

DEBUG com.owl.web.security2.WebSecurityConfig - Trying ldap search with Eq Filter: dc=gorman,dc=com, search filter: memberUid=cn=admin123456,ou=users,ou=dc,dc=gorman,dc=com, custom prop: cn

DEBUG com.owl.web.security2.WebSecurityConfig - Filter: (memberUid=cn=admin123456,ou=users,ou=dq,dc=gorman,dc=com)

DEBUG com.owl.web.security2WebSecurityConfig - Found attribute! DQAdmin

DEBUG com.owl.web.security2.WebSecurityConfig - Checking roles with attribute size: 1

DEBUG com.owl.web.security2.WebSecurityConfig - checking first item as string: DQAdmin

DEBUG com.owl.web.security2.WebSecurityConfig - checking for mappings for : DQAdmin

Issue: Trying to map groups, nothing is loaded.

Successfully finding groups:

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - using group object class filter = (&(objectClass=posixGroup))

INFO com.owl.web.security2.SpringLDAPSecurityPg - attributes

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - gidNumber-gidNumber: 500

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - memberUid-memberUid: cn=gorman_ admin,ou=users,ou=dq,dc=gorman,dc=com,cn=admin123456,ou=users,ou=dq,dc=gorman,dc=com

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - objectClass-objectClass: posixGroup,top

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - cn-cn: DQAdmin

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - Getting Group Response for name attribute: CNN

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - Error calling map from attributes nested class: null

Problem: Either the groups are not found, or the properties we are looking at for LDAP_GROUP_RESULT_DN_ATTRIBUTE, LDAP_GROUP_RESULT_NAME_ATTRIBUTE do not exist.

Solution: If you know your group search path, base path, and group search filter are correct, you most likely need to set the properties for LDAP_GROUP_XXX to the right values.

Successful logging:

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - using group object class filter = (&(objectClass=posixGroup))

INFO com.owl.web.security2.SpringLDAPSecurityPg - attributes

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - gidNumber-gidNumber: 500

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - memberUid-memberUid: cn=gorman_ admin,ou=users,ou=dq,dc=gorman,dc=com,cn=admin123456,ou=users,ou=dq,dc=gorman,dc=com

DEBUG com.owl.web.security2.SpringLDAPSecurity

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - objectClass objectClass: posixGroup, top

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - cn-cn: DQAdmin

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - Getting Group Response for name attribute: CN

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - Getting Group Response for fully qualified name attribute: distinguishedname

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - using baseGroup from properties

DEBUG com.owl.web.security2.SpringLDAPSecurityPg - LDAPGroup {dn='cn=DQAdmin,', group='DQAdmin'}

Connection Security

You can select and map any role to your connection on the Connection Manager page, so that any user with those mapped roles will see the connection in the UI.

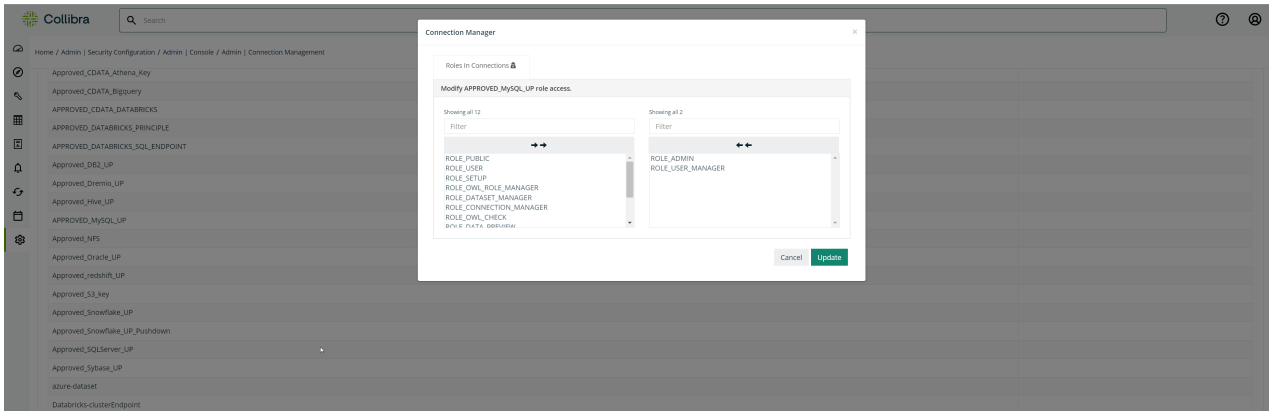
Steps

1. Log in to the Collibra DQ instance and click the gear icon in the left navigation pane.
2. Click **Admin Console**.
3. In Quick Links, click the **Security** tile.
4. Click the **DB Connection Security**check box.
5. Click **Save**.
6. Click **Admin Console** again.
7. Click **DB Access**.
8. From the Connection Name column, select your connection.
   The Connection Manager displays the available roles.
9. Click the role(s) you want to map to your connection from the left column, which moves the role(s) to the right column.

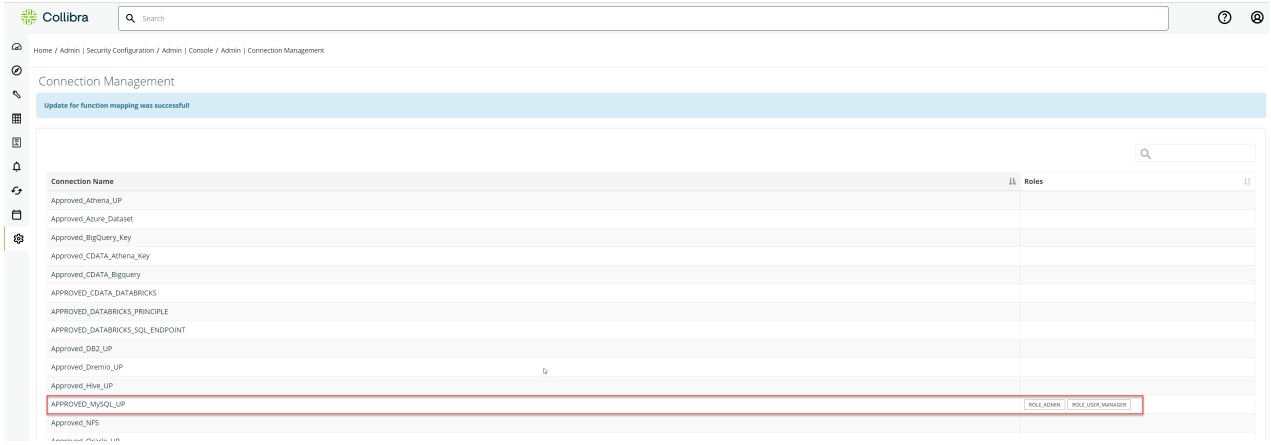
> Note You can move roles from the left column to the right column and back

> by clicking the roles in the Connection Manager.

10. When you have completed your role selections, click **Update**.



The role(s) you selected for the connection displays in the Roles column.



Any user with those mapped roles can now see the connection in the UI.

Configuring Security Settings

Configure general and dataset security settings on the Security Settings page.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Select **Admin Console**.

3. Select **Security Settings**, from the left navigation panel.
4. Enable/disable the options on the Security Settings page.

General Security Settings

Setting	Description
Local User Store Enabled (requires web-app restart)	When this setting is selected, you can create and administer users stored in Collibra DQ's internal database user store. For more information, go to Local User Store Authentication.
Auto Approve Users	When this setting is selected, local users are approved automatically upon login.
DB Connection Security	When this setting is checked, you can select and map roles to your connection. Note Any user with those mapped roles can see the connection in the UI.
DQ Job Security	When this setting is checked, the OWL CHECK role is required. Note The OWL CHECK role is the only role that can run DQ scans when DQ Job Security is enabled.
Allow Temp File Upload for DQ Job	When this setting is checked, users can upload temp files for DQ Jobs.

Dataset Security Settings

When Dataset Security is enabled, dataset access is required to edit, map, or retrieve datasets or business units.

The system fully restricts access to the Findings page for admin users when the following security settings are configured:

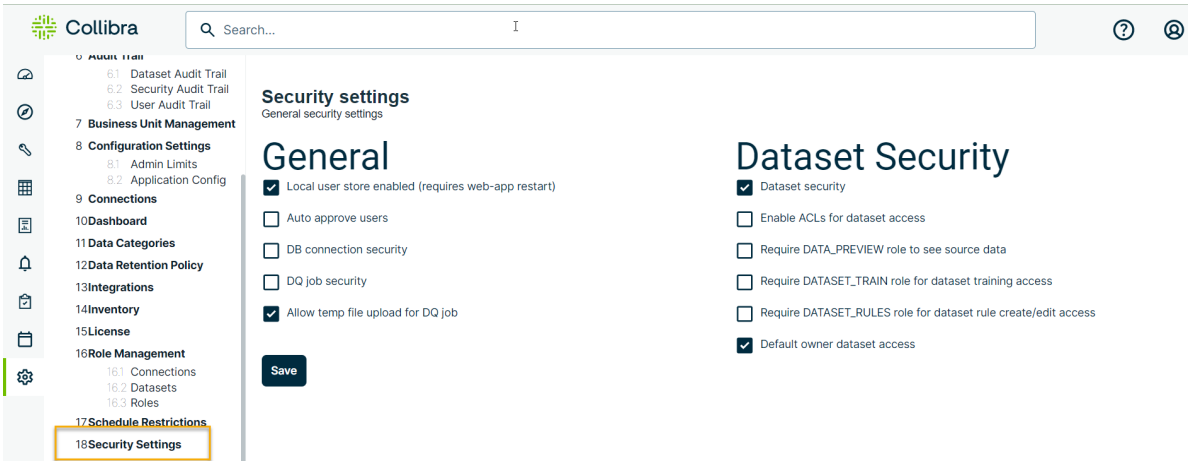
- The Dataset Security setting is enabled.
- The Default Owner Dataset Access setting is unchecked.
- The dataset does not belong to any roles, or any roles to which the user has access.

ACL Security

When ACL is enabled/disabled, an administrator can configure the following options in the Dataset Security pane to limit usage/permissions on datasets:

- Data_Preview role limit (role that can view source data)
- Dataset_train role limit (role that can train data sets)
- Dataset_rules role limit (role that can add / edit / delete rules)

> Note This configuration is tied to data sets and not connections or jobs.



Dataset Masking

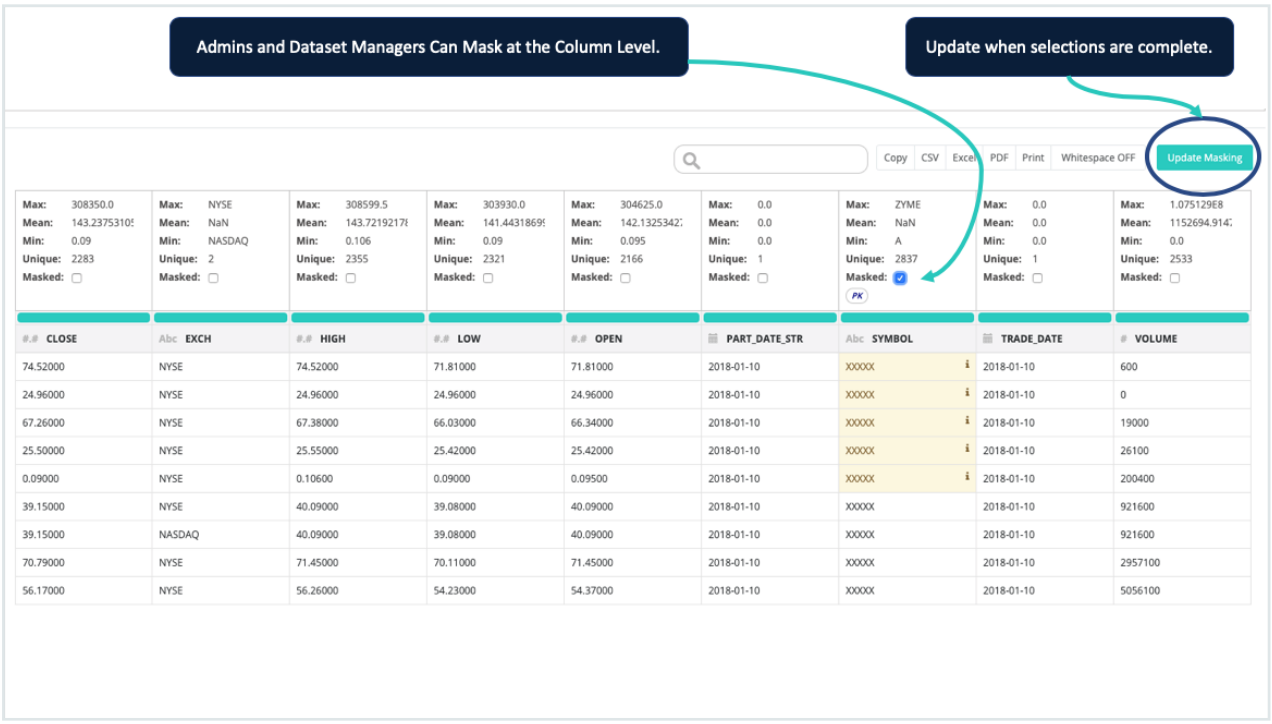
Add column level masking for sensitive data in a dataset.

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.

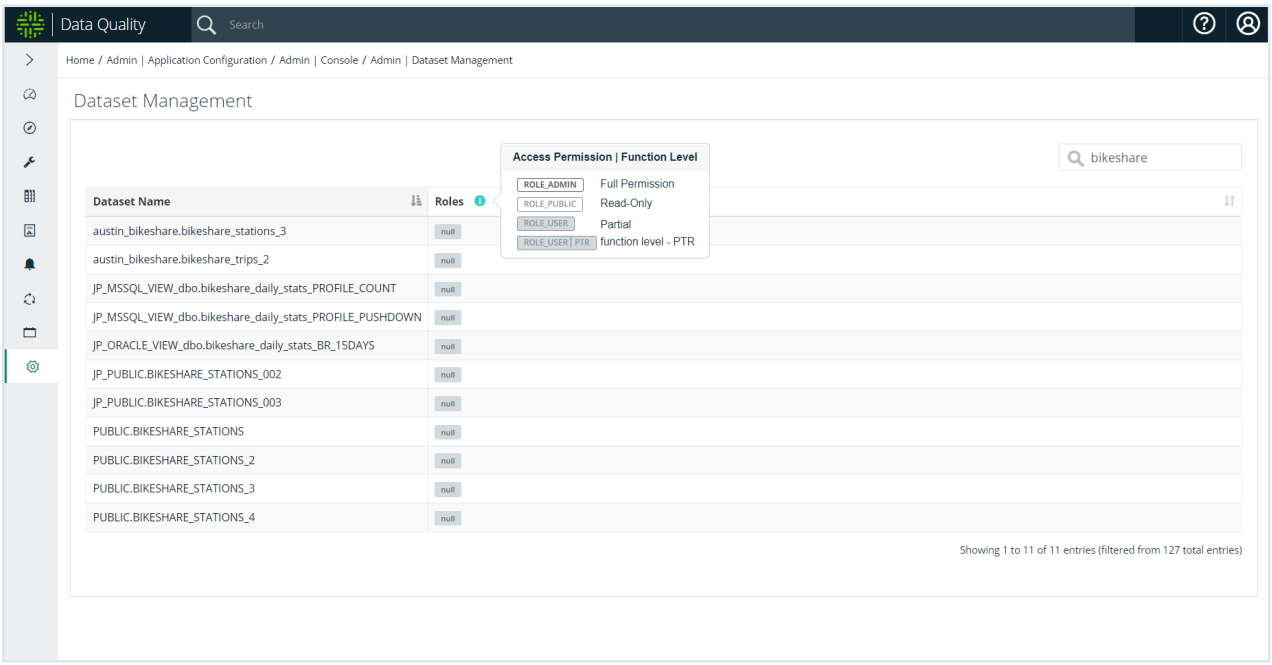


Managing Datasets and Their Roles

You can manage datasets and the roles assigned to them on the Dataset Management page.

Steps

1. Sign in to Collibra DQ and click the ⚙ in the left navigation panel.
2. Select **Admin Console** from the left navigation panel.
3. Select **Security** on the Quick Links page and select **Dataset Security**.
4. Click **Save**.
5. Click the ⚙ gear icon again.
6. Click **Admin Console**.
7. Click **Datasets** on the Quick Links page.



> Note When Dataset Security is enabled, dataset access is required to edit, map, or retrieve datasets or business units.

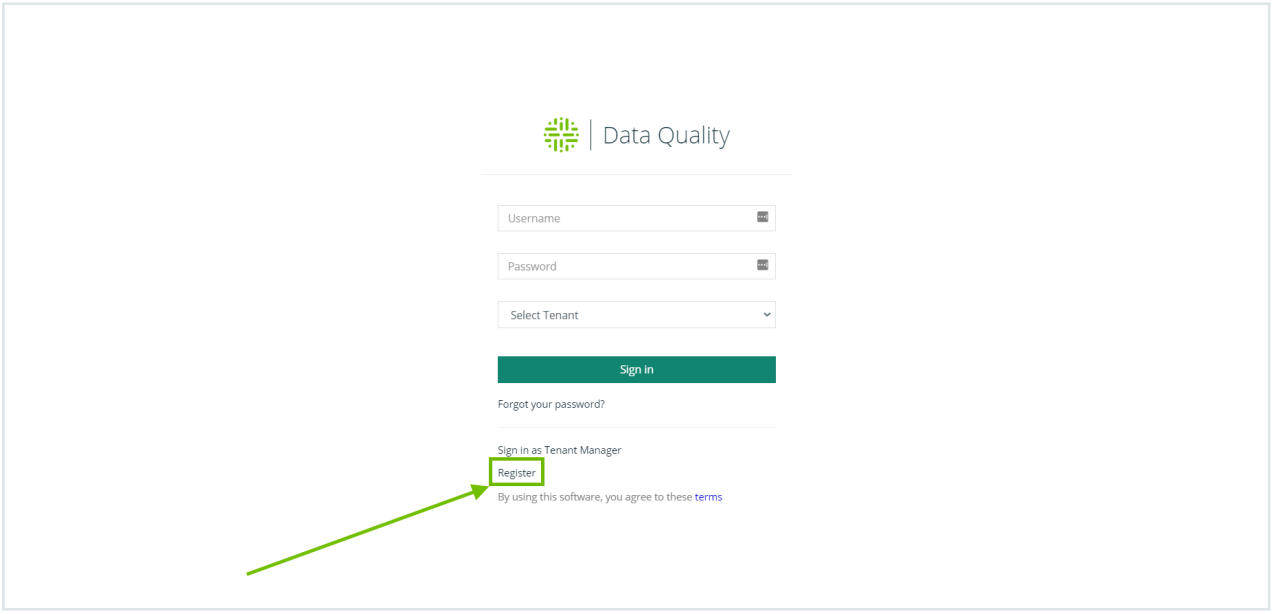
For a better understanding about Collibra DQ roles, go to Role Based Access Control (RBAC).

Local User Store Authentication

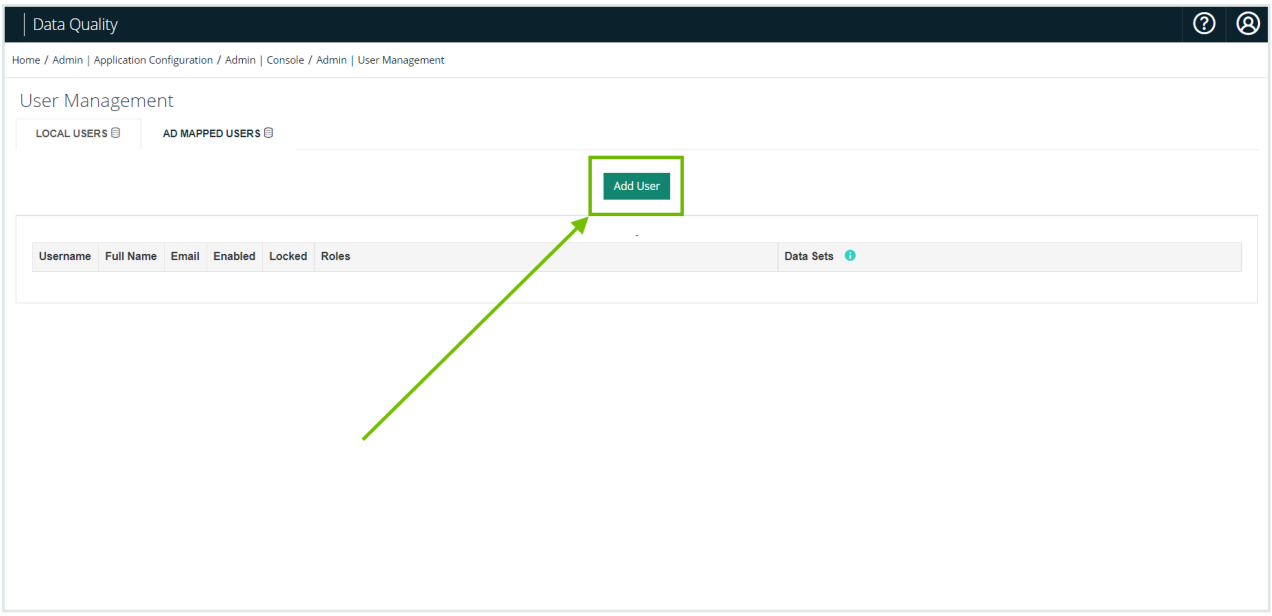
Local user store authentication is enabled by default, and the Collibra Data Quality & Observability ships with a default user with admin privileges. If you should need to enable/disable this mechanism, you can find the setting by navigating to the Admin Console and clicking on the **Security** icon. 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.



Application administrators can also create user accounts by navigating to the Admin Console and clicking the **Users** tile.



In either case, simply fill out the account form and a user record gets created. The user account at this stage does not have access to the application until an application administrator grants that user access to a Collibra DQ role or multiple roles.

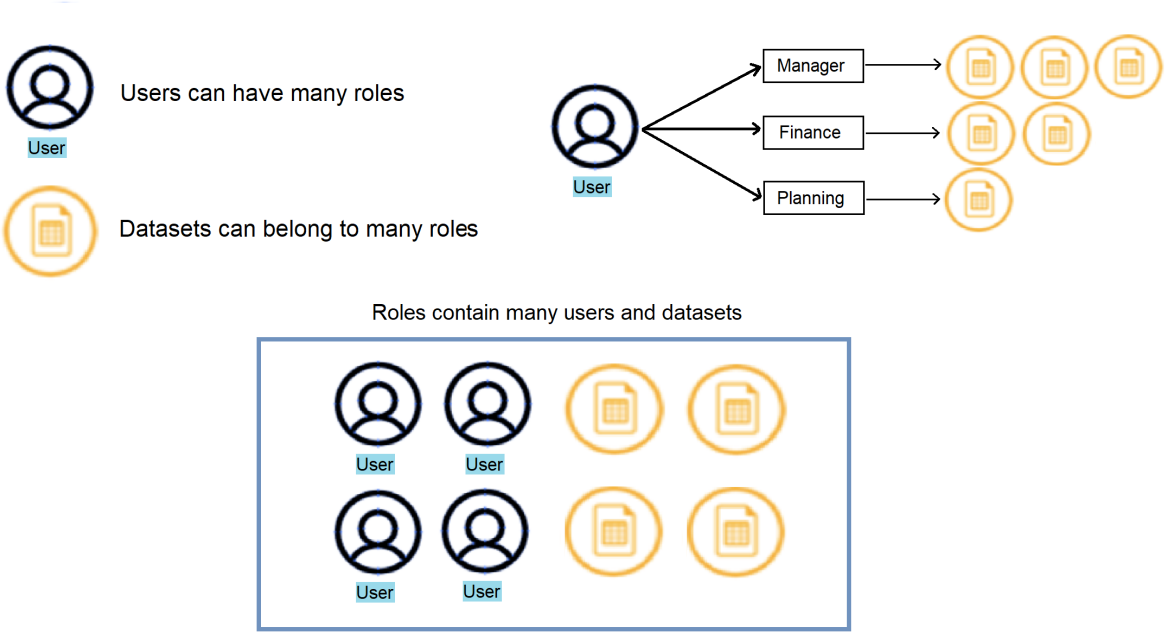
> Note If LDAP is enabled and a local user password or username is invalid and fails to authenticate, Collibra DQ attempts to authenticate against the AD/LDAP server.

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. To create custom application roles, go to Role Based Access Control (RBAC).

Role Based Access Control (RBAC)

The image below depicts Collibra Data Quality's security architecture.



Whether leveraging a Local User Store, Active Directory, or using the out of the box user accounts that come with Collibra Data Quality & Observability via LDIF, security stays the same. An admin can create many ROLEs. A user, whether local user, LDIF user, or AD user can be part of one or many roles. And a ROLE maps to a data set within Collibra Data Quality.

When dataset security is enabled and you want to access a dataset, or want to see, add, or remove an existing business unit for a dataset, you must have a role that is attached to that dataset.

For datasets, when dataset security and default dataset owner access is enabled, a user with a role attached to a dataset **or** the dataset owner can:

- **Add** - User with no dataset access (with no role attached to any existing dataset) can still create a dataset. After creating it, this user (who is the default dataset owner) can see the dataset, profile, and business units, and add and remove business units to their (owned) dataset.
- **Retrieve/See** - User can retrieve/see datasets, based on dataset access.

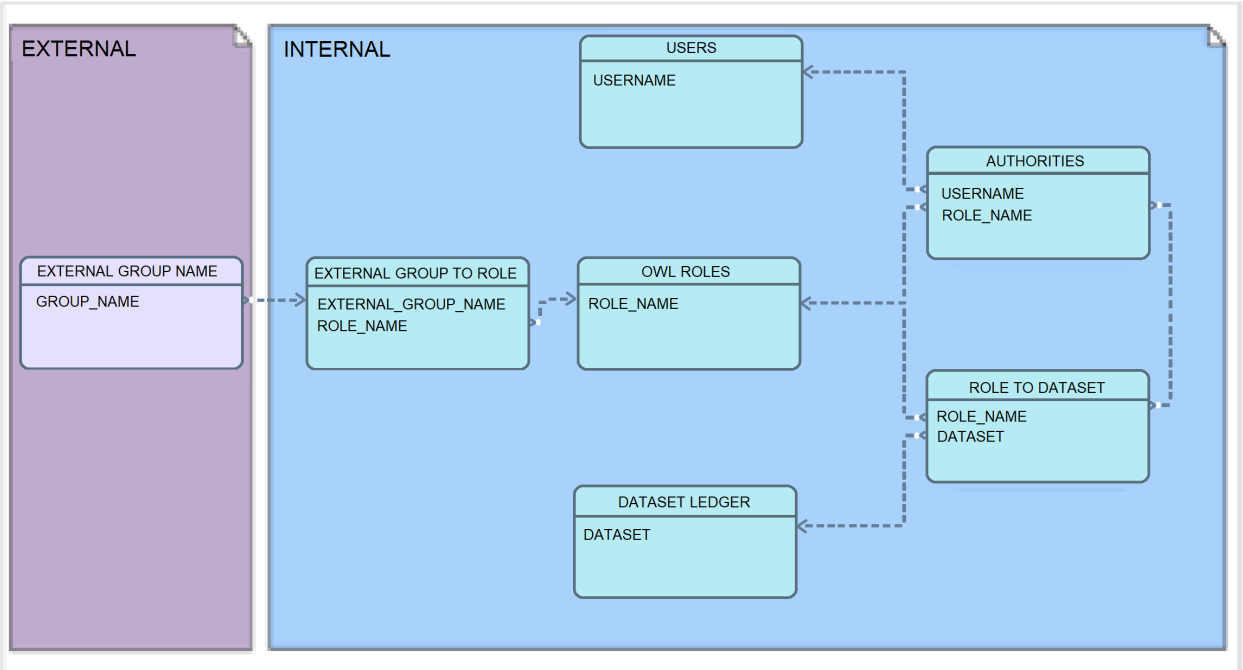
- **Edit** - User can edit datasets, based on dataset access.
- **Remove** - User can remove datasets, based on dataset access.

For business units, when dataset security and default dataset owner access is enabled, a user can:

- **Retrieve/See** - User can retrieve/see business units, based on dataset access.
- **Edit** - User can edit business units, based on dataset access.
- **Remove** - User can remove business units, based on dataset access.

> Note You must be an admin to create a business unit, which can then be added to a dataset.

A unique feature within Collibra DQ is the fact that we do not store information about external user accounts. This avoids the need to sync external users from an external user store such as AD to Collibra DQ. Instead, Collibra DQ will map the external group to an internal role. From here the ROLE can be mapped to the different functionality within Collibra DQ whether they are Admins / Users / and have access to different datasets and future functionality. The other benefit is that if a specific userid within the external user store is terminated, when the user is purged from the external user store such as AD they will immediately not have access to Collibra DQ's web application. This is because when the user logs into Collibra DQ's web application that is backed by AD their login will interrogate AD to authenticate the user account. See logical flow below for how the group to role mappings work.



RBAC Usages

Collibra DQ supports RBAC configuration with both core roles and custom roles. The following table shows the core roles of Collibra DQ's RBAC configuration:

Role	Access Description
ROLE ADMIN	Allows you to modify any access, config settings, connections, and role delegation.
ROLE DATA GOVERNANCE MANAGER	Allows you to manage (create / update / delete) Business Units and Data Concepts.
ROLE USER MANAGER	Allows you to create or modify users and add users to roles.
ROLE OWL ROLE MANAGER	Allows you to create roles and edit role mappings to users, AD groups, and datasets.
ROLE CONNECTION MANAGER	Allows you to add, edit, and delete connections.
ROLE DATASET MANAGER	Allows you to create or modify datasets to roles and mask dataset columns.

Role	Access Description
ROLE OWL CHECK	This is the only role that can run DQ scans when DQ Job Security is enabled.
ROLE DATA PREVIEW	This is the only role that can view source data if Data_Preview security is enabled.
ROLE DATASET TRAIN	This is the only role that can train datasets if Dataset_Train security is enabled.
ROLE DATASET RULES	This is the only role that can add / edit / delete rules if Dataset_ Rules security is enabled.
ROLE VIEW DATA	Controls which users can access the DQ SQL editor to run the SQL against the database.
ROLE PUBLIC	Public: Access to scorecards, no dataset access when dataset security is enabled.
ROLE USER	Do not use.
ROLE SETUP	Do not use.

Custom roles can be added via the Role Management page by navigating to the Admin Console and clicking on the Roles Icon. Custom roles can also be added 'on the fly' during the Active Directory Role Mapping step.

It is these custom roles that will determine the users that have access to data sets (including profile/rules/data preview/scoring), and database connections

Additional information regarding setting up Dataset and Connection security can be found in those documents respectively.

SAML Authentication

This page shows you how to integrate Collibra DQ with an existing SAML solution and have your application act as a service provider. Once you have set up the environment variables, you can then access and configure SAML security settings as an administrator in the **SAML Setup** section 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`.

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td>SAML_
KEYSTORE_ALIAS</td><td>The alias of the keypair (private and public) in the keystore used for SSL verification.</td></tr>
<tr><td>LOCAL_
REGISTRATION_
ENABLED</td><td>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:
<table><tr><th>Property</th><th>Description</th></tr><tr><td>export LOCAL_
REGISTRATION_
ENABLED=true</td><td>Displays the registration link on the sign-in page for local users. This is set to <code>true</code> by default.</td></tr><tr><td>export LOCAL_
REGISTRATION_
ENABLED</td><td>Hides the registration link on the sign-in page for local users.</td></tr></table>To display or hide the registration link from a K8s config map, select from the following properties:
<table><tr><th>Property</th><th>Description</th></tr><tr><td>LOCAL_
REGISTRATION_
ENABLED="true"</td><td>Displays the registration link on the sign-in page for local users. This is set to <code>true</code> by default.</td></tr><tr><td>LOCAL_
REGISTRATION_
ENABLED="false"</td><td>Hides the registration link on the sign-in page for local users.</td></tr></table><blockquote>Note This property must be set globally on deployment.</blockquote></td></tr>
<tr><td>SAML_MAX_
AUTH_AGE</td><td>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 <code>14400 seconds</code> (4 hours).</td></tr>
</table>

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 DQ, or if you have 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 existing administrator with a username and password to the tenant you want to configure.
2. In the **Admin Console**, click **SAML Security**, which is located under User Management in the left navigation panel.
   The SAML Security Settings page displays.
3. From the SAML Enabled drop-down list, select **Enabled**.
4. Click the + icon to add Meta-Data configuration details.
5. Enter the following 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.

6. Click **Save**.

> Tip Once you complete this setup, restart your application 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.

Multi-tenancy support through SAML RelayState

As of Collibra DQ version 2021.11, in a multi-tenant environment, 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`, the app 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 in the in the **SAML Setup** section of the **Admin Console**.

For more information, go to SAML Authentication.

Configuring SAML SSO for Azure

For SAML SSO, if the provider is Azure, some additional configuration is required to read the groups and to do role mapping.

When groups are pulled from Azure Active Directory SSO, if more than five groups are assigned to a user, the group claims are returned 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**.

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).

Generate a Keystore

Before starting the SSL setup, you need to generate a keystore.

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/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.

Use something similar to the following command to create the keystore.

```
keytool -genkey -alias <your alias> -keyalg RSA -keysize 2048 -
keystore <file name>
```

Generate a Certificate Signing Request (CSR) from the Keystore

You must create a CSR from the Keystore to share with the Certificate Authority (CA). This allows you to obtain a valid SSL certificate, which is signed by the CA, to secure your site.

Use a command that is similar to the following to create the CSR.

```
keytool -certreq -alias <your alias> ssl -keystore <file name>
```

> Note When prompted, enter the alias name and password.

You can now submit the CSR to a CA for signing. The CA will send you a CA-signed certificate 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 key store>
```

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 owl 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
```

Collibra DQ Legal

Agreements

This section contains legal agreements specific to Collibra.

Collibra Evaluation Agreement

For the full Collibra Evaluation Agreement document, follow the link below.

Collibra Click-Through Evaluation Agreement

BY ACCEPTING THESE TERMS OR OTHERWISE USING OR ACCESSING THE EVALUATION OFFERINGS, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS AND CONDITIONS IN THIS COLLIBRA EVALUATION AGREEMENT ("EVALUATION AGREEMENT"). COLLIBRA GRANTS YOU ACCESS TO AND USE OF THE EVALUATION OFFERINGS ONLY IF YOU ACCEPT THE TERMS AND CONDITIONS OF THIS EVALUATION AGREEMENT. IF YOU ARE ENTERING INTO THIS EVALUATION AGREEMENT ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THIS EVALUATION AGREEMENT, IN WHICH CASE THE TERMS "YOU," OR "YOUR" SHALL REFER TO SUCH ENTITY. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT AGREE WITH THESE TERMS AND CONDITIONS, YOU MAY NOT ACCESS OR USE THE EVALUATION OFFERING.

This Evaluation Agreement is entered into between you (the individual and any entity for which you are acting, and your permitted successors and assigns) and Collibra, meaning Collibra Inc., if you are located in the United States, Mexico or Canada; or Collibra UK Limited, if you are located in a country other than the United States, Mexico or Canada (in each case, referred to herein as "**Collibra**").

1. Grant and Use Rights for Evaluation

1. **Evaluation Grant.** Subject to your compliance with this Evaluation Agreement, Collibra grants you, at no cost, a non-exclusive, non-transferable, non-sublicensable, temporary and limited (i) right to access and use Collibra's software-as-a-service cloud offering including any relevant accompanying software

components and documentation ("**Service**") and/or (ii) license to install, copy and use the object code form of Collibra's proprietary installed software product including any relevant accompanying documentation ("**Software**"), as applicable to your evaluation ("**Evaluation Offering**"), solely on an evaluation basis for non-production, internal, test and demonstration purposes (the "**Purpose**") during the Evaluation Term. There is no commitment to purchase the Evaluation Offering at the end of the Evaluation Term. Collibra may provision the Evaluation Offering with the assistance of its affiliates and suppliers, and during the Evaluation Term, Collibra may update the Evaluation Offering and/or limit certain features and functionality of the Evaluation Offering at its discretion.

2. **Evaluation Term.** You may use the Evaluation Offering subject to this Evaluation Agreement only for a period of twenty (20) days ("**Evaluation Term**"). Any extension of the Evaluation Term must be agreed to by both parties, in writing.

2. Restrictions

You will not (and will not permit anyone else to) do any of the following: (a) use the Evaluation Offering for anything other than the Purpose; (b) use the Evaluation Offering under this Evaluation Agreement to avoid incurring fees or exceeding any limitations agreed to in an order form entered into by and between you and Collibra; (c) provide access to, distribute or sublicense the Evaluation Offering to a third-party, (d) use the Evaluation Offering on behalf of, or to provide any product or service to, third parties, (e) use or reference the Evaluation Offering to develop a similar or competing product or service, (f) reverse engineer, decompile, disassemble, or seek to access the source code, algorithms, programming interfaces, or non-public APIs to the Evaluation Offering, except to the extent expressly permitted by applicable law (and then only with prior notice to Collibra), (g) circumvent any established usage limits, including restrictions on number of authorized users, whether through the use of APIs or other means (h) modify or create derivative works of the Evaluation Offering, or copy any element of the Service (other than authorized copies of the software components), (i) remove or obscure any product identification or proprietary notices in the Evaluation Offering, (j) publish benchmarks or performance information about the Evaluation Offering, (k) interfere with the Service's operation, circumvent its access restrictions or conduct any security or vulnerability test of the Service, or (l) transmit any viruses or other harmful materials to the Service.

3. Ownership

You will retain all right, title and interest to, and all intellectual property rights in your Content (as defined below) sent to the Evaluation Offering. Except for the limited right to use the Evaluation Offering under this Evaluation Agreement, this Evaluation Agreement does not give you any title, interest or rights, including intellectual property rights, in any component of the Evaluation Offering. The Evaluation Offering and all copyright, patent, and other proprietary rights therein are and shall remain the sole property of Collibra. This includes any information that Collibra collects and analyzes in connection with the Evaluation Offering, such as Usage Data (defined below), user feedback and other information to improve and evolve Collibra's products and services offerings.

4. Your Content

You are solely responsible for any and all applications, files, information, materials, data, software, or other content uploaded to, stored, published or displayed by you through the Service ("**Content**"). You acknowledge that the Service is provisioned for the Purpose only, and therefore you will not use any production or regulated data (including personal information) as part of your Content in the use of the Service. You are responsible for protecting the security of your Content, including any access you might provide to your Content by your employees, customers or other third parties. You will take and maintain appropriate security, protection and backup for your Content. You are responsible for complying with any laws or regulations applicable to your Content. You are solely responsible for any consequences if your Content is inadvertently exposed or lost, whether or not you have encrypted, backed up or otherwise taken steps required by the relevant laws or regulations to protect your Content. To the extent Collibra process Personal Data, as defined in the DPA, each party agrees to comply with the terms of the data processing addendum, which can be found here: www.collibra.com/data-processing-addendum ("**DPA**"), and which is hereby incorporated by reference.

5. Usage Data

Collibra collects information related to your use of the Services so that Collibra can analyze usage, support the Service, and maintain and improve its products and services, and for other lawful business purposes ("**Usage Data**"). Collibra may also use and store

this Usage Data in an aggregated, anonymized form for Collibra's own purposes. You hereby expressly agree that Collibra may collect Usage Data during the Evaluation Term.

6. Support Services

Collibra does not provide support services or service level commitments of any kind in connection with the Evaluation Offering.

7. Term and Termination; Suspension

1. **Term.** This Evaluation Agreement and your use to the Evaluation Offering will commence when you click to accept it as part of the sign-up process or, if earlier, when you use any of the Evaluation Offering (the "**Effective Date**"), and will be effective through the Evaluation Term, unless terminated earlier as permitted under this Section 7.
2. **Termination for Convenience.** This Evaluation Agreement may be terminated effective immediately by Collibra or you for any or no reason and at any time by providing notice of termination to the other party.
3. **Suspension.** Collibra may temporarily suspend your use of the Evaluation Offering if Collibra determines in its sole discretion: (i) you or your use of the Evaluation Offering is in breach of this Evaluation Agreement; (ii) Your use of the Evaluation Offering poses a security risk to the Evaluation Offering, or interferes with, disrupts, damages, or accesses in an unauthorized manner the servers, networks, or other properties or services of any other party; or (iii) suspension is required pursuant to Collibra's receipt of a subpoena, court order or other request by a law enforcement agency.
4. **Effect of Termination or Suspension.** Upon the termination or suspension of this Evaluation Agreement for any reason: (i) all rights and licenses granted to you under this Evaluation Agreement, including your ability to access any of your Content, will immediately terminate; (ii) you must promptly discontinue all access or use of the Evaluation Offering and delete or destroy any of Collibra's Confidential Information; and (iii) Collibra will have no obligation to retain your Content or configurations generated during the Evaluation Term by your use of the Evaluation Offering.

8. Disclaimer of Warranty

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, COLLIBRA PROVIDES THE EVALUATION OFFERING AS-IS WITHOUT ANY WARRANTIES OF ANY KIND, EXPRESS, IMPLIED, STATUTORY, OR IN ANY OTHER PROVISION OF THIS EVALUATION AGREEMENT OR COMMUNICATION WITH YOU, AND COLLIBRA SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, TITLE, AND ANY WARRANTIES ARISING FROM THE COURSE OF DEALING OR COURSE OF PERFORMANCE REGARDING OR RELATING TO THE EVALUATION OFFERING, THE DOCUMENTATION, OR ANY MATERIALS FURNISHED OR PROVIDED TO YOU UNDER THIS EVALUATION AGREEMENT. COLLIBRA DOES NOT WARRANT THAT THE EVALUATION OFFERING WILL OPERATE UNINTERRUPTED, OR THAT IT WILL BE FREE FROM DEFECTS OR THAT THE EVALUATION OFFERING WILL MEET (OR IS DESIGNED TO MEET) YOUR REQUIREMENTS.

9. Limitation of Liability

TO THE MAXIMUM EXTENT MANDATED BY LAW, IN NO EVENT SHALL COLLIBRA BE LIABLE FOR ANY LOST PROFITS OR BUSINESS OPPORTUNITIES, LOSS OF USE, LOSS OF REVENUE, LOSS OF GOODWILL, BUSINESS INTERRUPTION, LOSS OF DATA, INABILITY TO USE THE EVALUATION OFFERING, INCLUDING AS A RESULT OF ANY TERMINATION, DISCONTINUATION, MODIFICATION OR SUSPENSION OF THIS EVALUATION AGREEMENT OR THE EVALUATION OFFERING, OR ANY OTHER INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES UNDER ANY THEORY OF LIABILITY, WHETHER BASED IN CONTRACT, TORT, NEGLIGENCE, PRODUCT LIABILITY OR OTHERWISE. COLLIBRA'S LIABILITY UNDER THIS EVALUATION AGREEMENT SHALL NOT, IN ANY EVENT, EXCEED USD $100. THE FOREGOING LIMITATIONS SHALL APPLY REGARDLESS OF WHETHER COLLIBRA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF WHETHER ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE. YOU MAY NOT BRING A CLAIM UNDER THIS EVALUATION AGREEMENT MORE THAN EIGHTEEN (18) MONTHS AFTER (i) THE END OF THE

EVALUATION TERM, OR (ii) THE DATE THE CLAIM FIRST ARISES, WHICHEVER IS EARLIEST.

10. Confidentiality

1. **Confidential Information**. Each party acknowledges that it or its employees may, in the course of performing its responsibilities under this Evaluation Agreement, be exposed to or acquire information which is proprietary or confidential to the other party. "**Confidential Information**" means information disclosed under this Evaluation Agreement that is designated by the disclosing party as proprietary or confidential or that should be reasonably understood to be proprietary or confidential due to its nature and the circumstances of its disclosure. Collibra's Confidential Information includes any technical or performance information about the Evaluation Offering. Confidential Information excludes information that the receiving party can document (a) is or becomes public knowledge through no fault of the receiving party, (b) it rightfully knew or possessed prior to receipt under this Evaluation Agreement, (c) it rightfully received from a third-party without breach of confidentiality obligations or (d) it independently developed without using the disclosing party's Confidential Information. As receiving party, each party will (a) hold in confidence and not disclose Confidential Information to third parties except as permitted in this Evaluation Agreement, and (b) only use Confidential Information to fulfill its obligations and exercise its rights in this Evaluation Agreement. Each party will use reasonable care to protect the Confidential Information in the same manner as it would protect its own Confidential Information of a similar nature, but in no event with less than reasonable care. The receiving party may disclose Confidential Information to its employees, agents, contractors and other representatives having a legitimate need to know, provided it remains responsible for their compliance with this Section 10 and they are bound to confidentiality obligations no less protective than this Section 10. Unauthorized use or disclosure of Confidential Information may cause substantial harm for which damages alone are an insufficient remedy. Each party may seek appropriate equitable relief, in addition to other available remedies, for breach or threatened breach of this Section 10.
2. **Required Disclosures**. Nothing in this Evaluation Agreement prohibits either party from making disclosures, including Content and other Confidential Information, if required by applicable law, subpoena or court order, provided (if permitted by

applicable law) it notifies the other party in advance and reasonably cooperates in any effort to obtain confidential treatment at disclosing party's expense.

11. General

1. **Governing Law, Jurisdiction and Venue**. This Evaluation Agreement is governed by the laws of the State of New York without regard to conflicts of laws provisions and without regard to the United Nations Convention on the International Sale of Goods, and the jurisdiction and venue for actions related to this Evaluation Agreement will be the state and United States federal courts located in New York, New York, and both parties submit to the personal jurisdiction of those courts.
2. **Compliance with Laws**. You will comply with all applicable laws and regulations in the use of the Evaluation Offering under this Evaluation Agreement.
3. **Assignment**. You may not assign this Evaluation Agreement without Collibra's prior written consent, and any such action in violation of this provision, is null and void, and a breach of this Evaluation Agreement. Collibra may assign or transfer this Evaluation Agreement to an affiliate or in connection with a merger, reorganization, acquisition or other transfer of all or substantially all its assets or voting securities. This Evaluation Agreement will bind and inure to the benefit of the party's permitted successors and assigns.
4. **Independent Contractors**. The parties are independent contractors, not agents, partners or joint venturers.
5. **Third-Party Beneficiaries**. Except as expressly provided in this Evaluation Agreement, this Evaluation Agreement does not create or establish any rights or beneficiaries for any person or entity that is not a party to this Evaluation Agreement.
6. **Export**. You acknowledge that the Evaluation Offering is subject to export control and economic sanctions restrictions imposed by the U.S. government and import restrictions by certain foreign governments (collectively "**Trade Laws**"). In using or accessing the Evaluation Offering, you will not and will not allow any third party to use the Evaluation Offering in violation if any Trade Laws or remove or export from the U.S. or allow the export or re-export of any part of the Evaluation Offering or any direct product thereof to any location, party or end-use which the U.S. government or

any agency thereof requires an export license or other governmental approval at the time of export or re-export without first obtaining such license or approval. You represent and warrant that you and any of your users: (i) are not listed on any U.S. government list of prohibited or restricted parties, including the U.S. Treasury Department list of Specially Designated Nationals and Blocked Persons, or the U.S. Commerce Department Denied Persons List or Entity List; (ii) are not an entity or person who is organized under the laws of, ordinarily resident in, or controlled by the government of, any country or region (1) that is subject to a U.S. government embargo or comprehensive sanction, (2) to which the U.S. has prohibited export transactions or (3) that has been designated by the U.S. government as a "terrorist supporting" country; (iii) will not use the Evaluation Offering for the manufacture, design or development of nuclear, chemical or biological weapons or missile technology, or for terrorist activity; and (iv) will not submit to the Evaluation Offering any information controlled under the U.S. International Traffic in Arms Regulations or listed on the Commerce Control List unless approved in writing by Collibra. You will notify Collibra promptly if you or your users become subject to any order or restriction listed in this Section.

7. **Government End-Users**. Elements of the Evaluation Offering are commercial computer software. If the user or licensee of the Evaluation Offering is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Evaluation Offering or any related documentation of any kind, including technical data and manuals, is restricted by the terms of this Evaluation Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Evaluation Offering was developed fully at private expense. All other use is prohibited.

8. **Amendments**. Except as otherwise provided herein, any amendments, modifications or supplements to this Evaluation Agreement must be in writing and signed by each party's authorized representatives or, as appropriate, agreed through electronic means provided by Collibra.

9. **Waivers and Severability**. Waivers must be signed by the waiving party's authorized representative and cannot be implied from conduct. If any provision of this Evaluation Agreement is held invalid, illegal or unenforceable, it will be limited to the

minimum extent necessary, so the rest of this Evaluation Agreement remains in effect.

10. **Entire Agreement**. This Evaluation Agreement is the parties' entire agreement regarding its subject matter and supersedes any prior or contemporaneous agreements or communications regarding its subject matter, whether written or oral. Notwithstanding the foregoing, to the extent a separate written agreement has been signed between you and Collibra in connection with the subject matter hereof, such separate written agreement will supersede this Evaluation Agreement. In this Evaluation Agreement, headings are for convenience only and "including" and similar terms are to be construed without limitation.

Collibra DQ Documentation Archive

> Important
> This documentation has been retired and is unlikely to be updated. The features contained within this section are no longer supported in the Collibra DQ application.

DQ Cloud

- Cloud

DQ Connector

- DQ Connector

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.

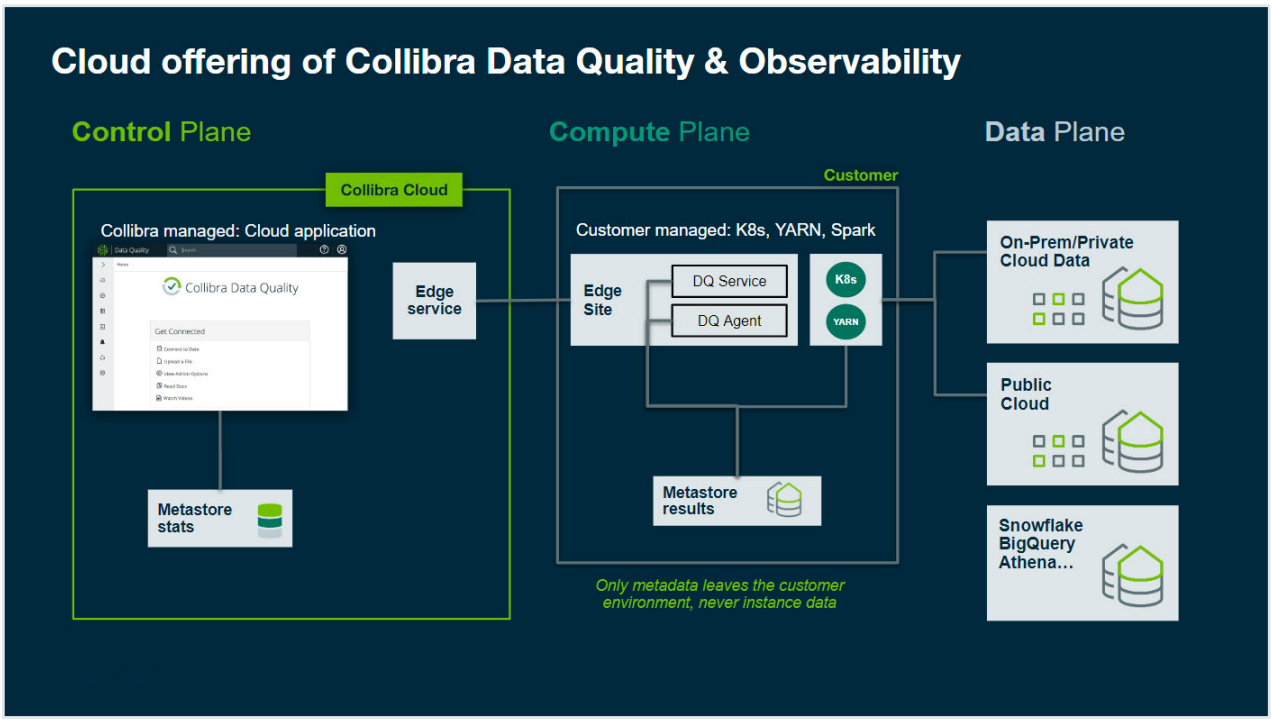
Requirements

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

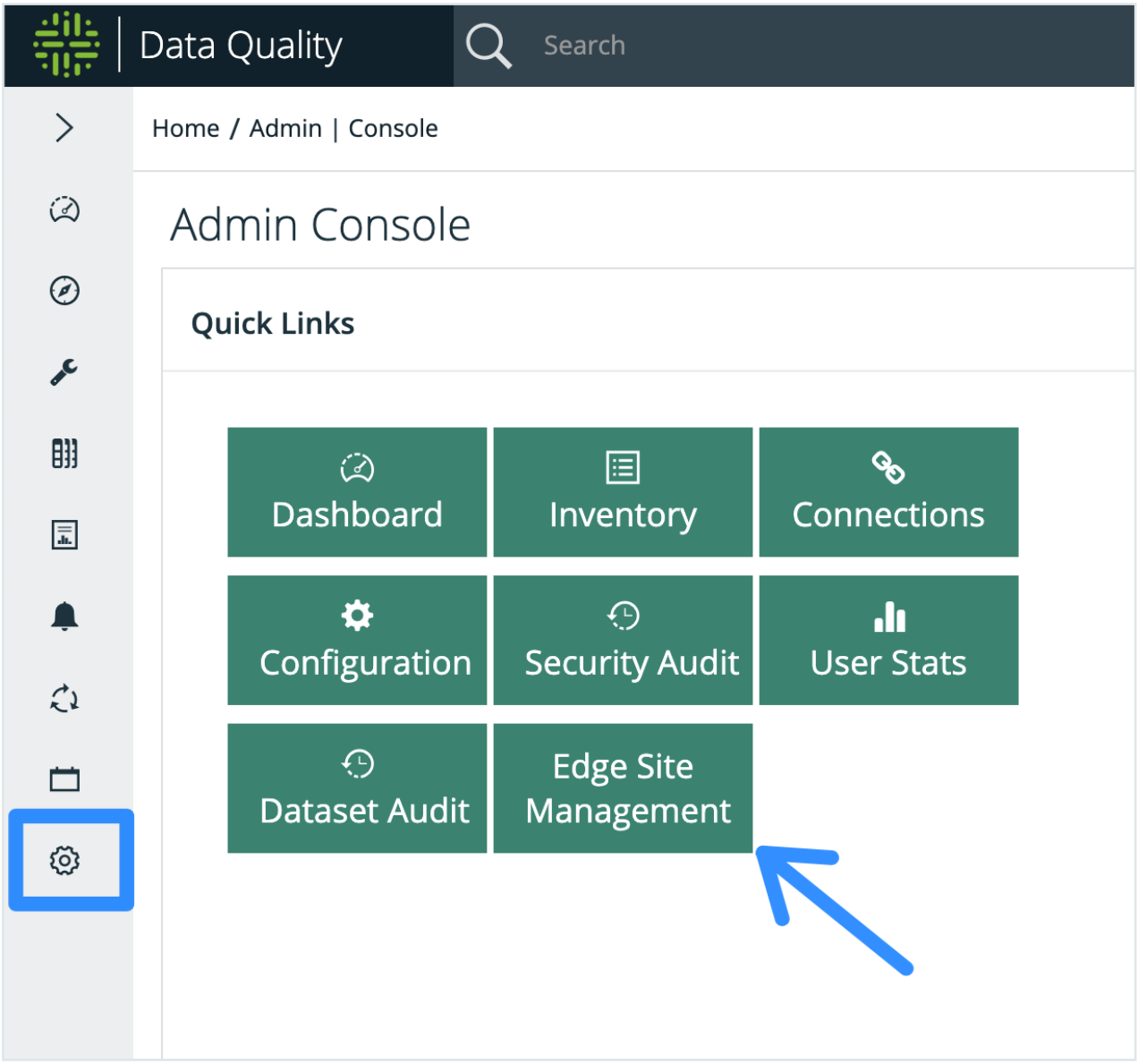
This is provisioned by the customer. There are several ways to install Postgres. You should follow your existing company process to provision a Postgres instance, such as RDS, Azure SQL, Cloud SQL, or standard install with a package manager. Ensure your version is 11 or newer.

> Important Remember your Postgres IP and login credentials. This is required when deploying the Edge site.

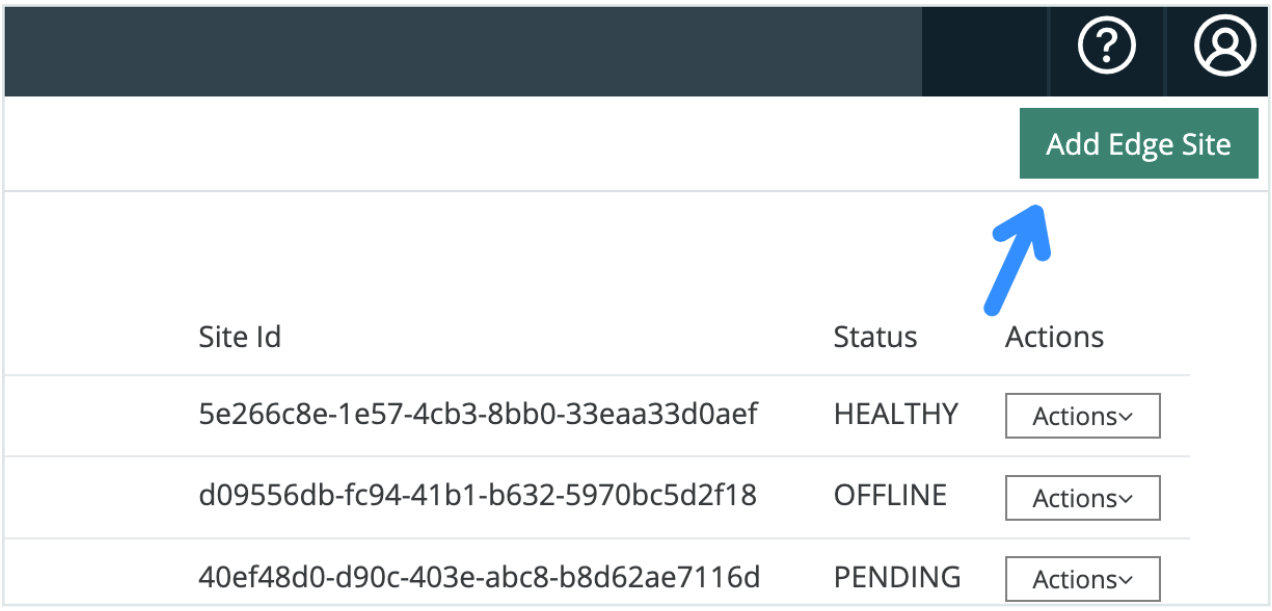
3. Install Edge

Refer to Edge documentation for system requirements.

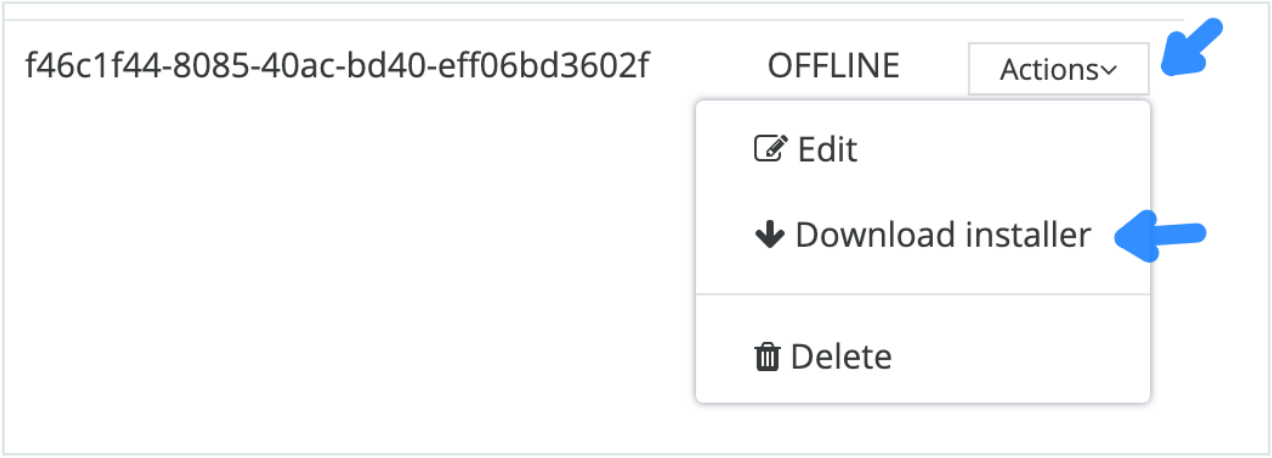
Navigate to the Edge Site Management panel in the Admin Console



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
```

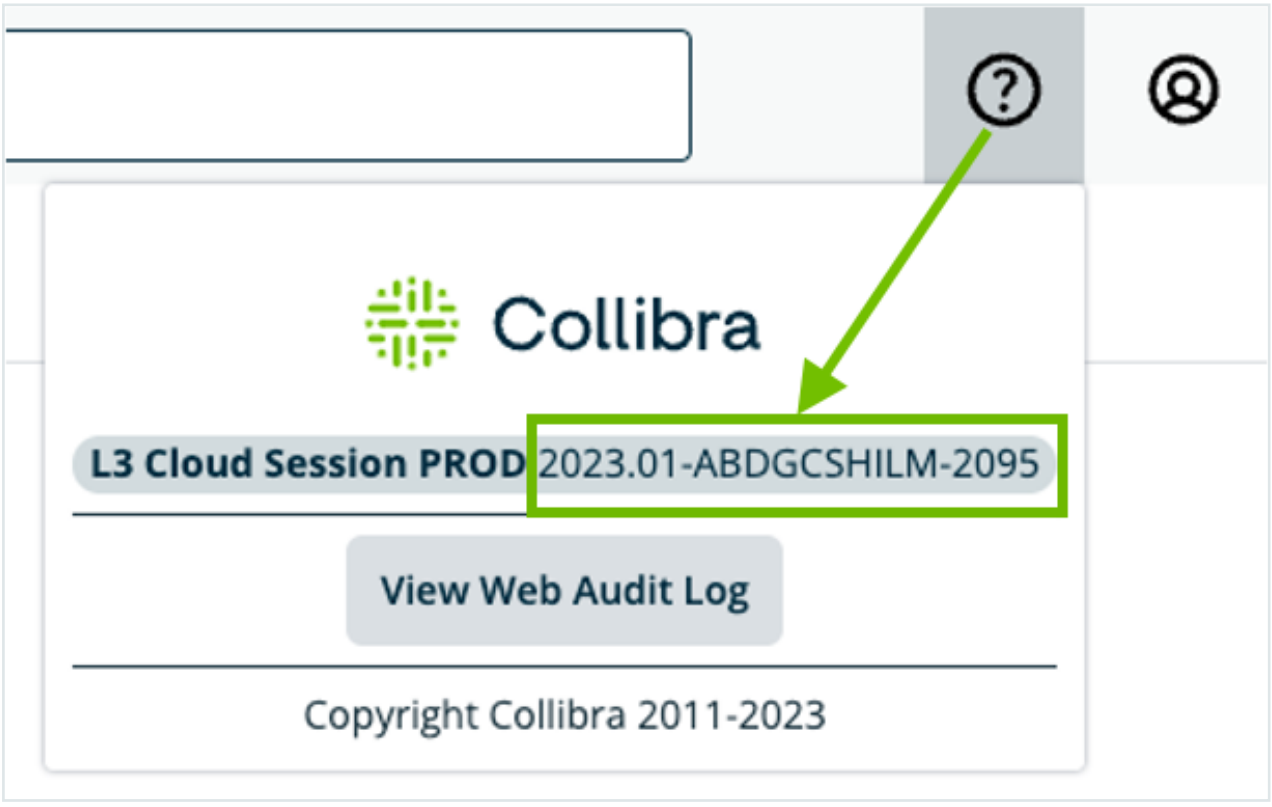
SSH to your VM after uploading the installer package. Untar the .tgz

```
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.

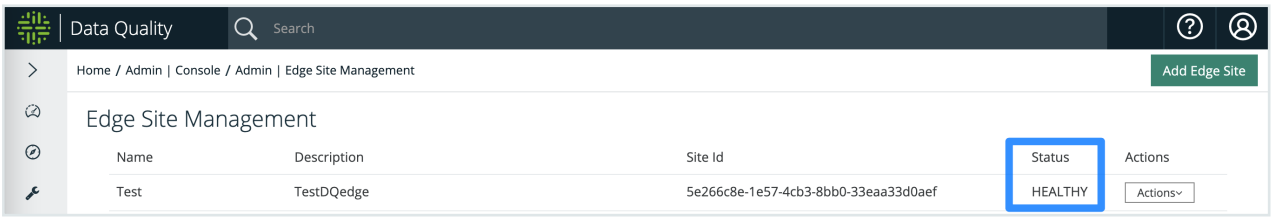
> Note Because Collibra DQ automatically derives its version from the Collibra Data Intelligence Cloud version, you do not need to specify the version of Collibra DQ for successful automatic upgrades.

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.

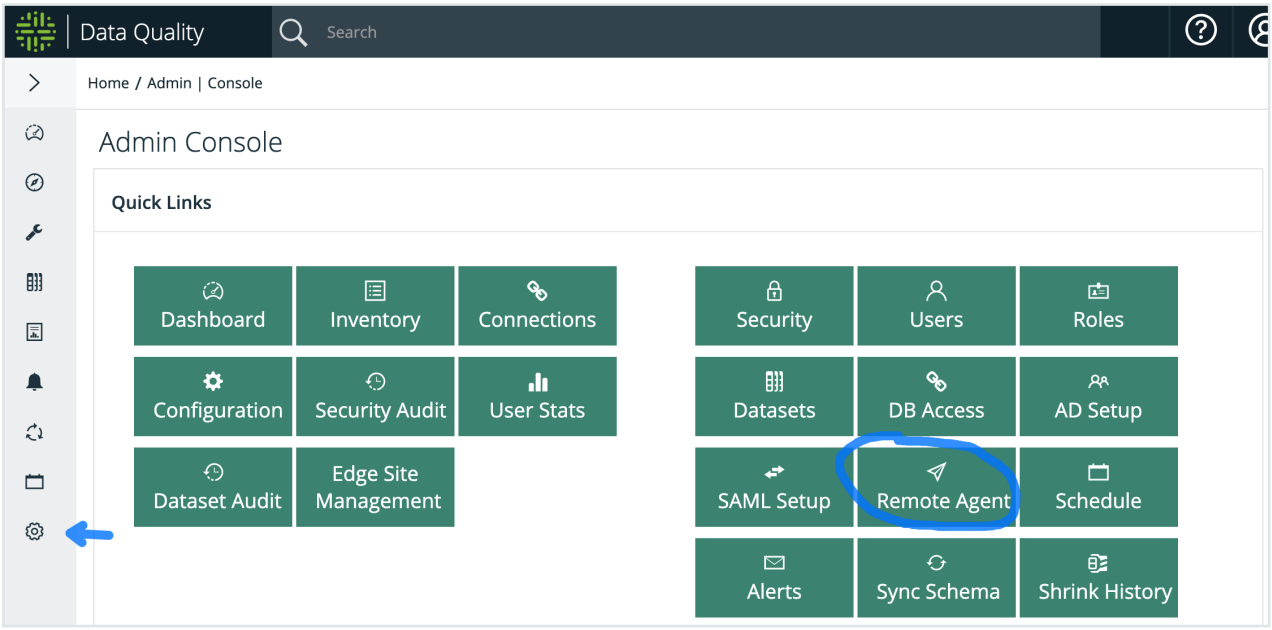
Reinstall the prerequisites if you perform the uninstall

```
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
```

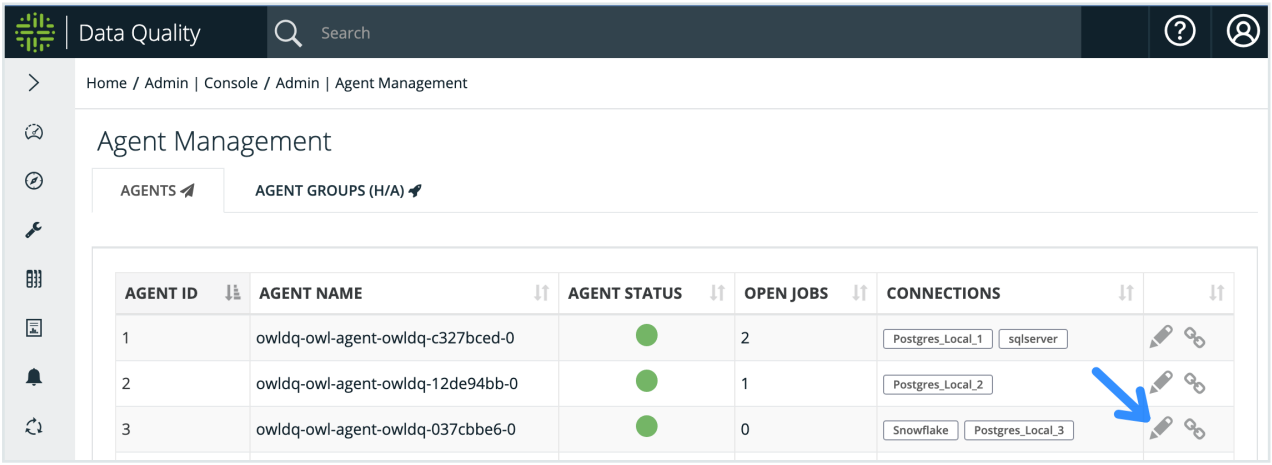
> Warning **To avoid orphaned records, do not delete an Edge using the UI.**

4. Configure an Agent

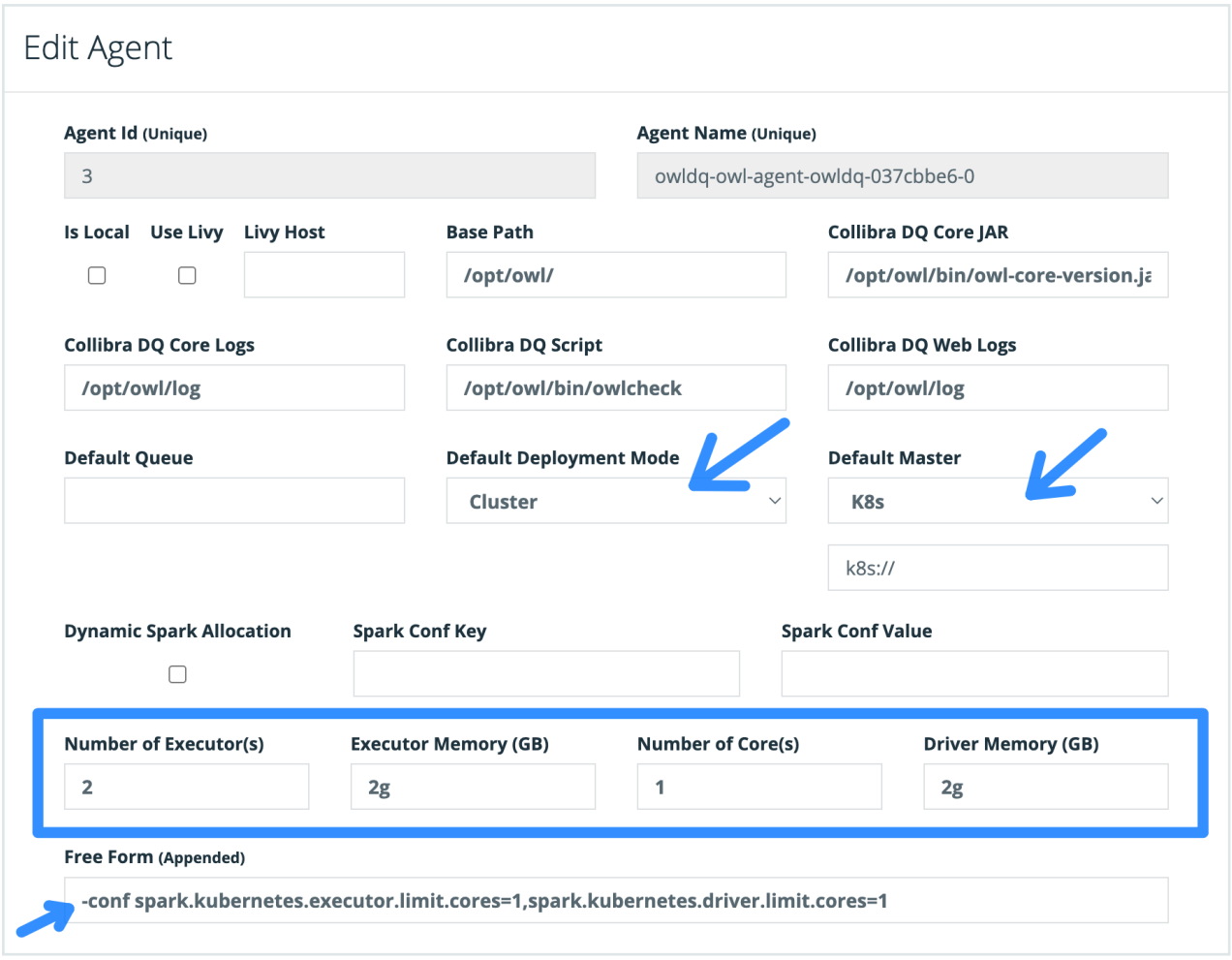
Go to the Remote Agent panel in the admin console



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 beta. 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

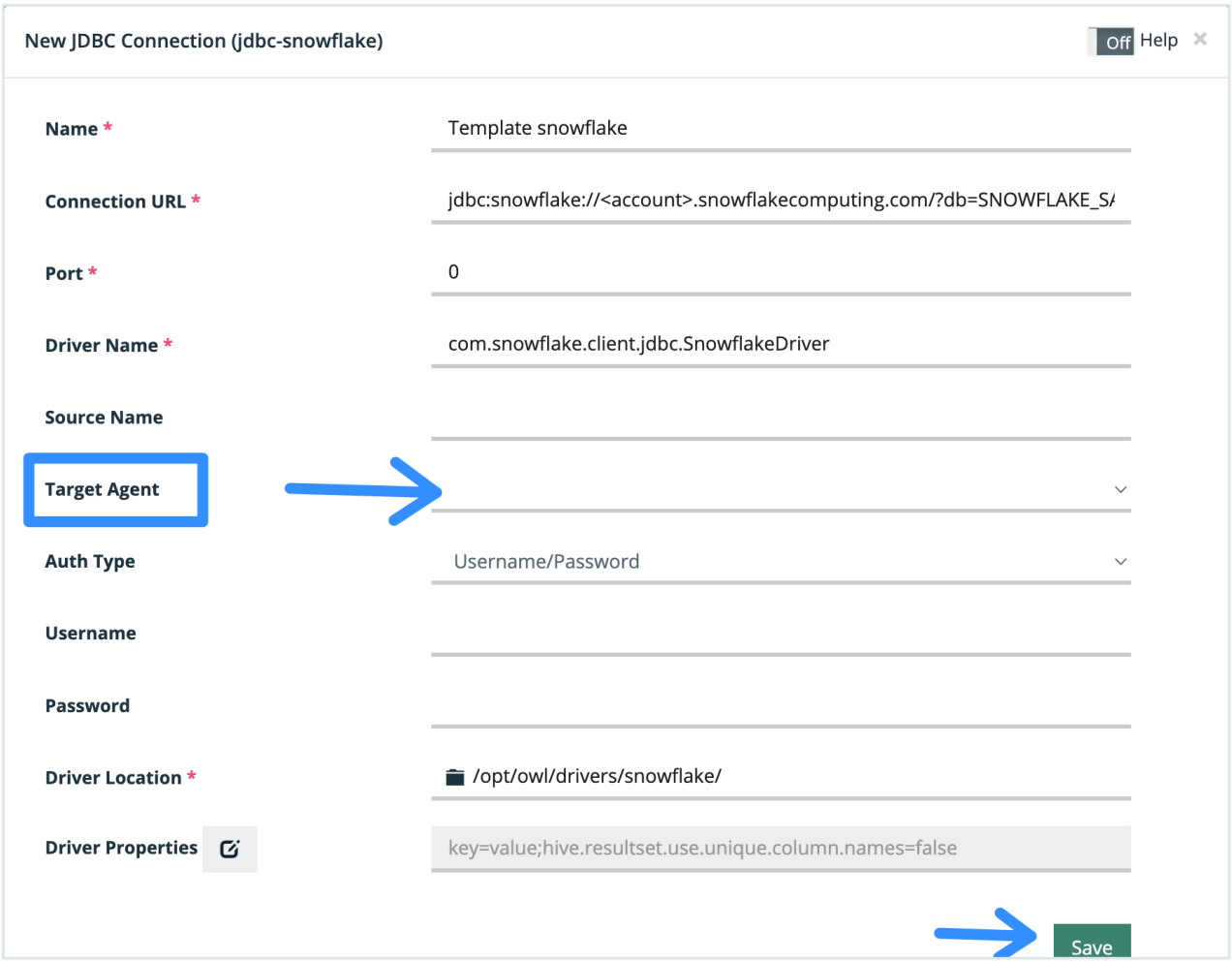
Set max cores to 1 in the job limit settings.

Refer to this link for configuring job limits.

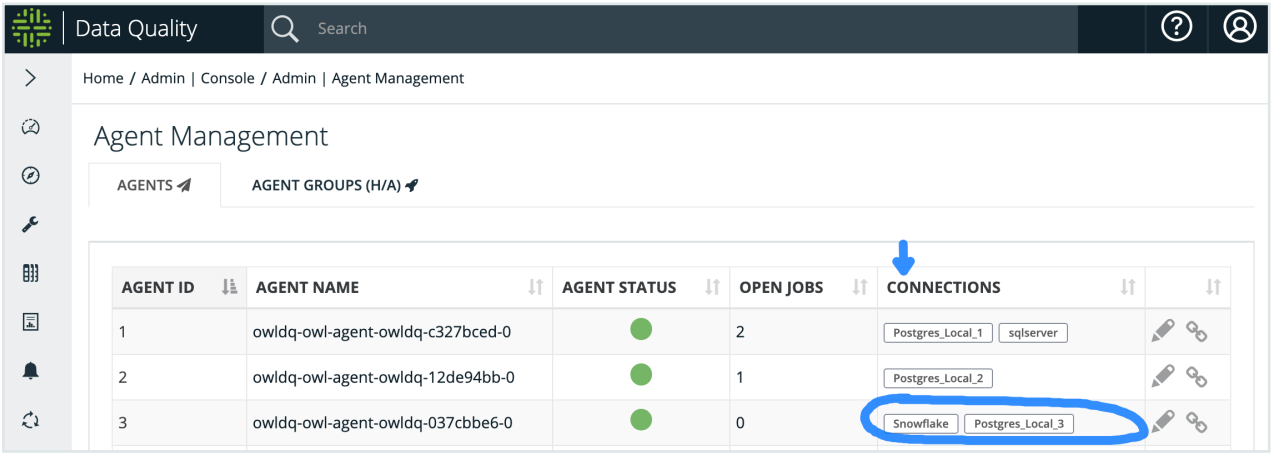
6. Add a Connection

This is the same process of adding a connection found Adding Connectionswith 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 beta 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 beta 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 beta, 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 Edge
site
 --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 Edge
site
 --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.`

DQ Connector

> Important
> This capability has been archived. For the latest option to integrate your Collibra DQ metadata with Collibra Data Intelligence Cloud, check out our enhanced integration documentation.

The Native DQ Connector brings intelligence from **Collibra Data Quality & Observability** into **Collibra Data Intelligence Cloud**. Once this integration is established, you will be able to bring in your Data Quality user-defined rules, metrics, and dimensions into **Collibra Data Catalog**.

> Note Only data sources ingested by both Collibra Data Catalog and Collibra Data Quality & Observability can synchronize data quality assets.

Prerequisites

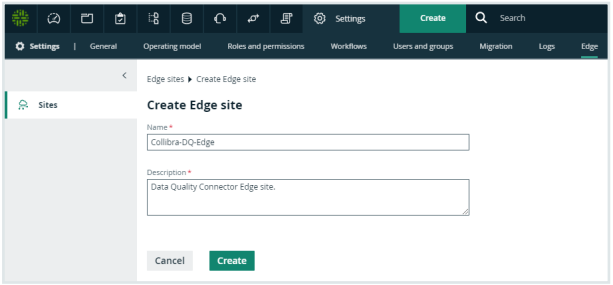
Resource	Notes
Collibra Edge Site	DQ Connector is a capability of Edge
Collibra Data Intelligence Cloud	2021.07 Release (or newer)
Collibra Data Quality	2.15 (or newer)
Databases and Drivers	Proper Access and Credentials (Username / Password)

Because the DQ Connector is an Edge capability, you must be able to ingest data via Edge. For information about enabling and configuring Edge, see the Edge Configuration guide.

Create a Collibra Data Quality & Observability Edge site

Create an Edge site with the following properties:

Field	Description
Name	The name of the Edge site, for example **Collibra-DQ-Edge**. Do not use spaces or special characters. This field is mandatory and the name must be globally unique.
Description	The description of the Edge site. We recommend to put at least basic location information of the Edge site. This field is mandatory.



Install the Collibra Data Quality & Observability Edge site

Follow the instructions for your environment to Install an Edge site.

> Note This process automatically creates an Edge user, which you use later in the setup process.

Connect to your Collibra Data Quality & Observability source

Create a connection for each Collibra Data Quality & Observability data source you want to synchronize. The following table shows the available properties and their descriptions as they appear on the :

Section	Property	Description
Connection settings		
	Name	The same name as the Collibra Data Quality & Observability connection name. Ensure that your connection name does not contain any white spaces, as they are not supported in Collibra DQ. Warning The connection name in Collibra Data Intelligence Cloud must be an exact match to the connection name used in Collibra DQ.
	Description	The description of the JDBC connection. This field is also visible when you register content.
	Connection provider	The connection provider, which determines the available connection parameters. Same as Collibra Data Quality & Observability.
Connection parameters		Example for **Username / Password JDBC driver**
	Username	The same username as the Collibra DQ connection username.

Section	Property	Description
	Password	The same password as the Collibra DQ connection password.
	Driver class name	In most cases, this is the same driver name as the Collibra DQ connection driver name. If you select a different driver in Collibra Data Intelligence Cloud, the driver class name can be different from the Collibra DQ driver class name.
	Driver Jar	In most cases, this is the same driver JAR file as from Collibra DQ. If you select a different driver in Collibra Data Intelligence Cloud, the driver jar can be different from the Collibra DQ driver jar. Ensure that the driver is supported in both Collibra Data Intelligence Cloud and Collibra DQ. Note Some CDATA drivers that are supported in Collibra Data Intelligence Cloud are not supported in Collibra DQ. It is best practice to use a CDATA driver in Collibra Data Intelligence Cloud, but you can use a different driver in Collibra DQ.
	Connection string	In most cases, this is the same URL as the Collibra DQ connection URL. If you select a different driver in Collibra Data Intelligence Cloud, the connection URL can be different from the Collibra DQ connection URL.

Add ingestion capabilities to your Collibra Data Quality & Observability connection

You must add a Catalog JDBC ingestion Edge capability template for each connection you have created to extract and process data for your data source.

Field	Description	Required
Capability	This section contains general information about the capability.	
Name	The name of the Edge capability.	✓ Yes
Description	The description of the Edge capability.	✗ No

Field	Description	Required
Capability template	The capability template. The value that you select in this field determines which sections appear on the page. Select the following Edge capability: `Catalog JDBC ingestion`	✓ Yes
Connection	This section contains information to connect to the data source.	
JDBC connection	The connection to the data source.	✓ Yes
JDBC data source type (Deprecated)	Deprecated field. The field was used to indicate the type of the data source. You no longer need to change this field. The required value is automatically identified. Note The automatically identified value is not shown in this page.	✓ Yes
Supports schemas	A text field where you have to enter *True* to enable database registration of data sources that have no schema. If the data source has schemas, you can ignore this field. Tip If the data source does not have a schema, Data Catalog creates a Schema asset with the same name as the full name of the database.	✗ No

Field	Description	Required
Others	This section can contain additional capability properties. Warning Adding additional properties can have a significant impact on your Edge site. Only add or update them together with Collibra Support.	✕ No

Field	Description	Required

Field	Description	Required
General	This section contains general information about logging.	
Debug	An option to automatically send Edge infrastructure log files to Collibra Data Intelligence Cloud. By default, this option is set to *false*. Note We highly recommend to only send Edge infrastructure log files to Collibra Data Intelligence Cloud when you have issues with Edge. If you set it to *true*, it will automatically revert to *false* after 24h.	✕ No
Log level	An option to determine the verbosity level of Catalog connector log files. By default, this option is set to *No logging*.	✕ No

Configure destinations for Collibra Data Quality & Observability assets

Collibra Data Quality & Observability rules, metrics and dimensions require their own domains in Data Catalog. If you don't have existing domains for data quality or wish to use new ones for the quality extraction purpose, create a domain for each type of data quality asset:

- Rules: **Rulebook Domain**
- Metrics: **Rulebook Domain**
- Dimensions: **Governance Asset Domain**

Assign permissions for Collibra Data Quality & Observability domains

Edge must have the correct resource permissions to manage assets inside the dedicated Collibra Data Quality & Observability domains. For each dedicated domain, assign the **Technical Steward** role to the Edge user.

> Note The Edge user is automatically created when you install the Edge site.

Add Collibra Data Quality & Observability characteristics to assets

To show Collibra Data Quality & Observability statistics for your data source, assign the following characteristic types to the **Table** and **Column** asset types:

Asset type	Characteristic type
Table	governed by Governance Asset
Column	is governed by Data Quality Rule

Add a DQ Connector capability

The DQ Connector facilitates the communication with Collibra Data Quality & Observability. Add a DQ Connector capability to your Collibra Data Quality & Observability Edge site:

Field	Description	Required
Capability	This section contains general information about the capability.	
Name	The name of the Edge capability.	✓ Yes
Description	The description of the Edge capability.	✗ No
Capability template	The capability template. The value that you select in this field determines which sections appear on the page. Select the following capability template to ingest Collibra Data Quality & Observability user-defined rules, metrics, and dimensions into Collibra Data Catalog: `DQ Connector`	✓ Yes
DQ	This section contains information about the Collibra Data Quality & Observability connection.	
Base URL	Your Collibra Data Quality & Observability URL	✓ Yes
Username	The Collibra Data Quality & Observability username for this connection.	✓ Yes

Field	Description	Required
Password	The Collibra Data Quality & Observability password for this connection.	✓ Yes
Encryption options	Select the type of encryption to use. Default: *To be encrypted by Edge management server*.	
Issuer of the JWT	If you have selected *Encrypted with public key*, enter your JWT issuer.	✗ No
Collibra metadata model	This section contains information about where to ingest Collibra Data Quality & Observability assets.	
DQ Rules domain id	The UUID of the **Rulebook Domain** for the ingested Collibra Data Quality & Observability rules.	✓ Yes
DQ Metrics domain id	The UUID of the **Rulebook Domain** for the ingested Collibra Data Quality & Observability metrics.	✓ Yes
DQ Dimensions domain id	The UUID of the **Governance Asset Domain** for the ingested Collibra Data Quality & Observability dimensions.	✓ Yes
Default DQ Dimension name	The default **Data Quality Dimension**, for example *Accuracy*, *Completeness*, *Consistency* and so on. Default: *Completeness*.	✓ Yes
DQ Metric classified by DQ Dimension relation type id	The UUID of the **Data Quality Metric classified by / classifies Data Quality Dimension** relation. If left unspecified, this relation will not be added.	✗ No
Assets are imported in batches of this size	The batch size of the ingestion. Default: *5000*.	✓ Yes

To make the Collibra Data Quality & Observability metadata available in Collibra Data Catalog, you must register the data source for each Collibra Data Quality & Observability data source you want to synchronize.

Create a Data Catalog System Asset

As a prerequisite to registering a data source in Data Catalog, you must create a System asset for each connected data source with the following properties:

Field	Value
Type	**System**
Domain	The domain to which the new assets will belong. You can only create a asset type in any domain of a domain type that is assigned to a selected asset type.
Name	The same name as the Collibra Data Quality & Observability connection name.

Register the Collibra Data Quality & Observability data source in Data Catalog

To make the Collibra Data Quality & Observability metadata available in Collibra Data Catalog, you must register the data source for each Collibra Data Quality & Observability data source you want to synchronize.

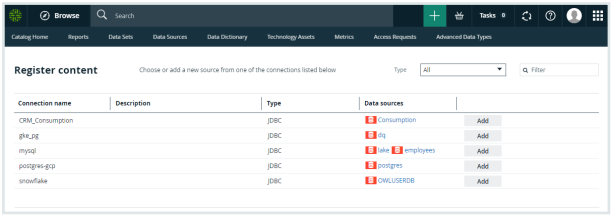
Create a Data Catalog System Asset

As a prerequisite to registering a data source in Data Catalog, you must create a System asset for each connected data source with the following properties:

Field	Value
Type	**System**
Domain	The domain to which the new assets will belong. You can only create a asset type in any domain of a domain type that is assigned to a selected asset type.
Name	The same name as the Collibra Data Quality & Observability connection name. Warning Connection name must be an exact match in both Collibra DQ and Collibra Data Intelligence Cloud. For example, if your connection name is **postgres-gcp** in Collibra DQ, it should also be **postgres-gcp** in Collibra Data Intelligence Cloud.

Register the Collibra Data Quality & Observability data source in Data Catalog

Register each Collibra Data Quality & Observability source in Data Catalog.



Extract Data Quality metadata

After you completed the DQ Connector configuration, you can start ingesting Collibra Data Quality & Observability metadata.

Prerequisites

- You have configured the metadata synchronization properties for the data source.

Steps

1. Open a Database asset page.
2. In the tab pane, click **Configuration**.
3. In the **Quality extraction** section, do one of the following:
    - To select schemas for data quality synchronization:
        i. Click **Edit**.
            » The **Data quality** column becomes editable.
        ii. Select whether to synchronize the available schemas.
        iii. Click **Save**.
    - To synchronize the selected schemas:
        i. Select the schema name to see its configuration.
        ii. Click **Synchronize**.
            » The synchronization job is started for the selected schemas.

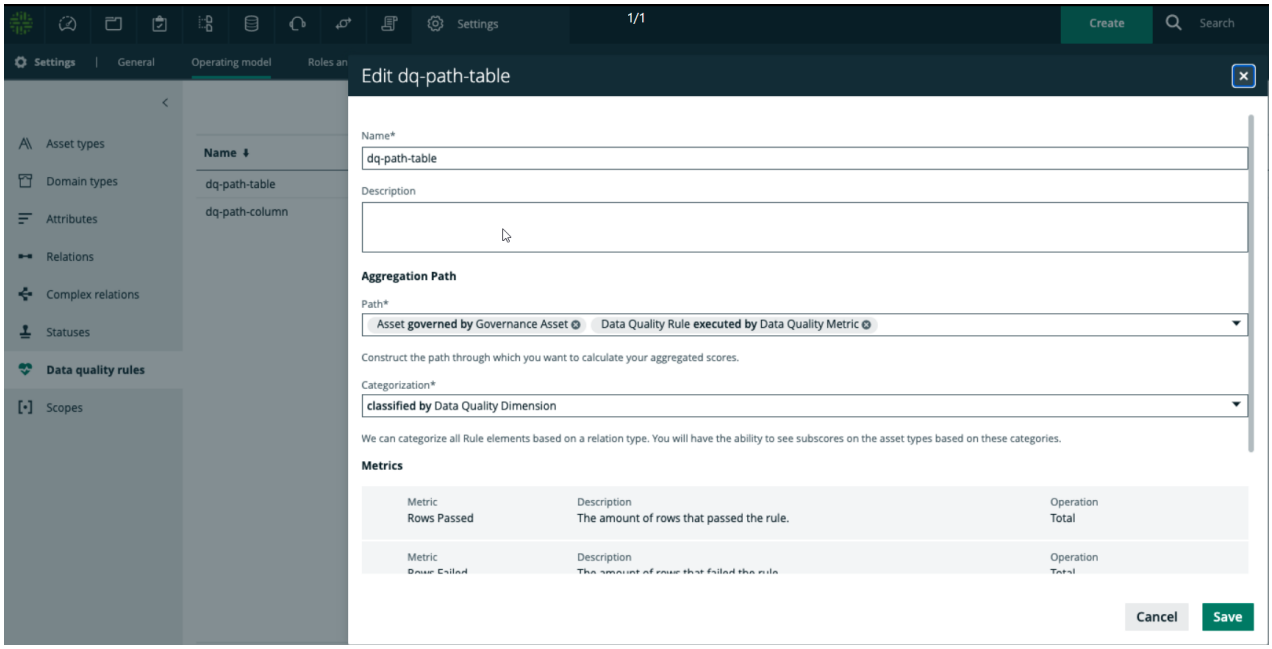
Known Limitations

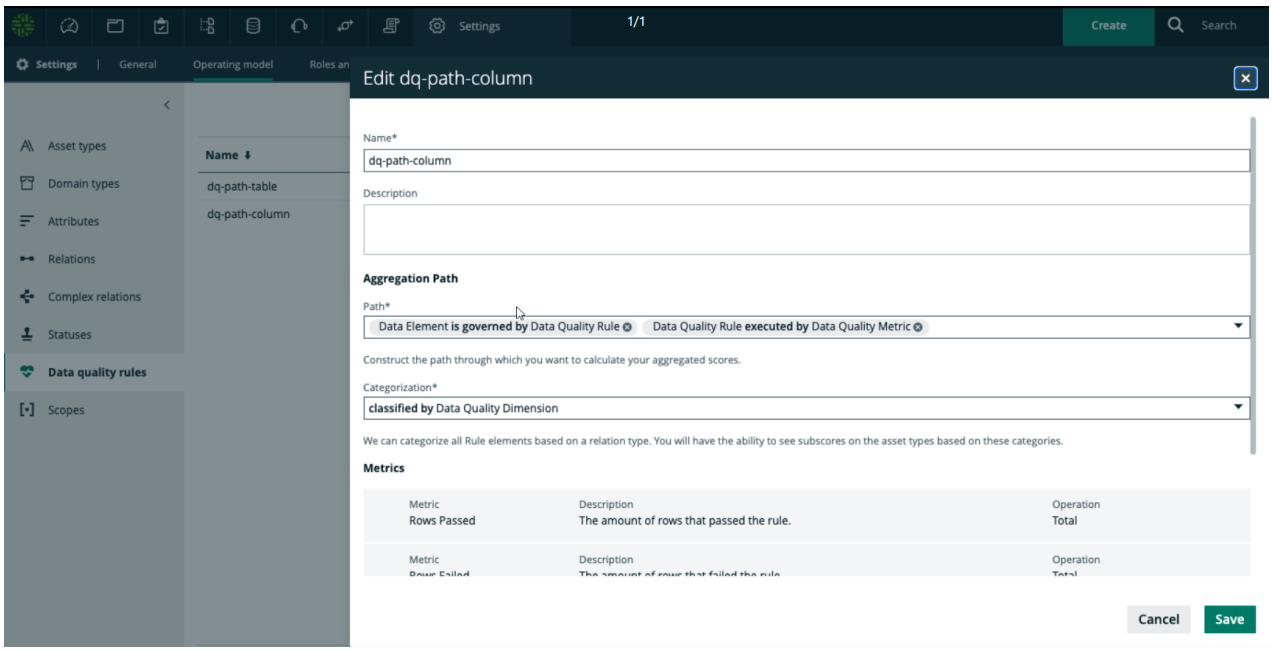
- Only 1 source tenant from Collibra DQ can be specified.
- On-demand ingestion (vs. scheduled).
- Can only specify 1 domain destination for each of Rules, Metrics, and Dimensions.
- Only JDBC sources supported (no file sources).
- When you integrate a dataset based on joins from multiple tables, the columns derived from the secondary dataset in Collibra DQ do not appear on the table ingested by Collibra Data Intelligence Cloud, and thus, the DQ Connector fails due to a missing asset. Because of the limitation where there must be a direct match between table and column names across Collibra DQ and Collibra Data Intelligence Cloud, the DQ Connector does not support aliases.
- 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 Data Intelligence Cloud.
    - A possible workaround until a fix is available is to update `export CSRF_TOKEN_ENABLED=false` in the [[[Undefined variable DataQuality.DQEnv]]] file for Standalone deployments or update `dq.security.csrf.token.enabled=${CSRF_TOKEN_ENABLED:false}` in the Web ConfigMap for Kubernetes deployments.

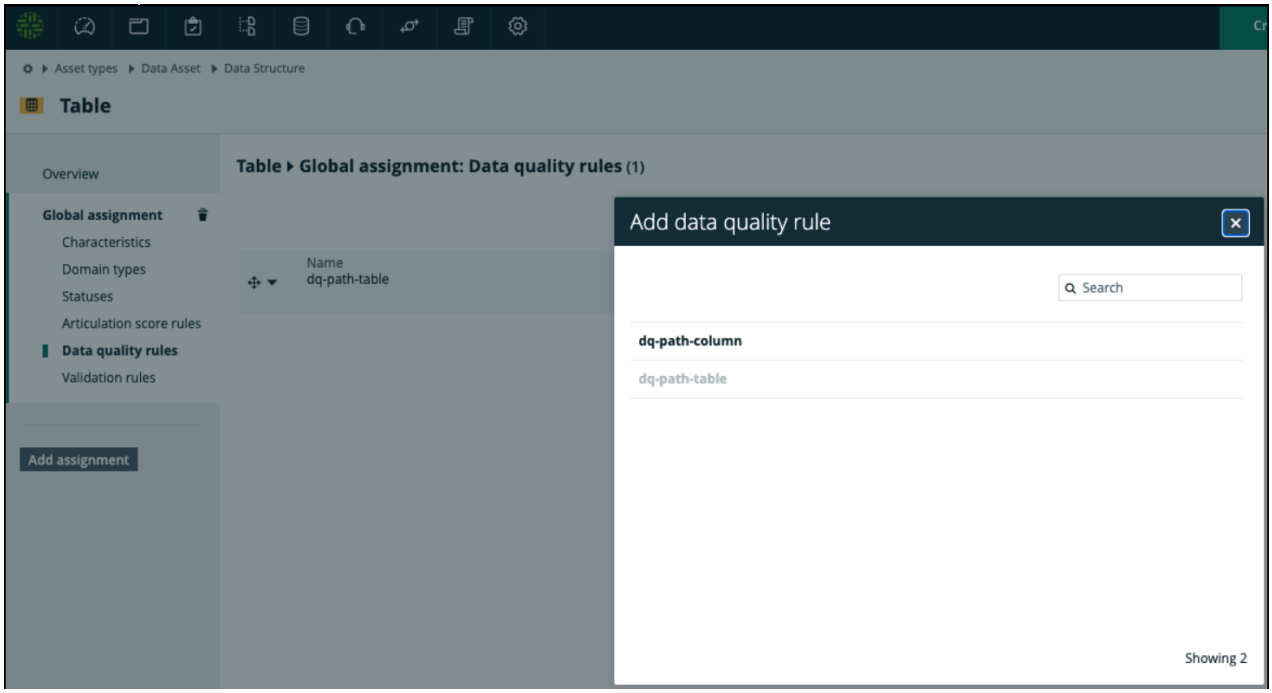
FAQ

Q: DQ Dashboard In DGC: I can verify the DQ Connector is synchronizing Data Quality Rules and Data Quality Metrics, but why don't Data Quality Dashboard Charts display?

A: Ensure correct **Aggregation Paths** and **Global Assignments** (or create, if none exist) for **Table** and **Column** below.

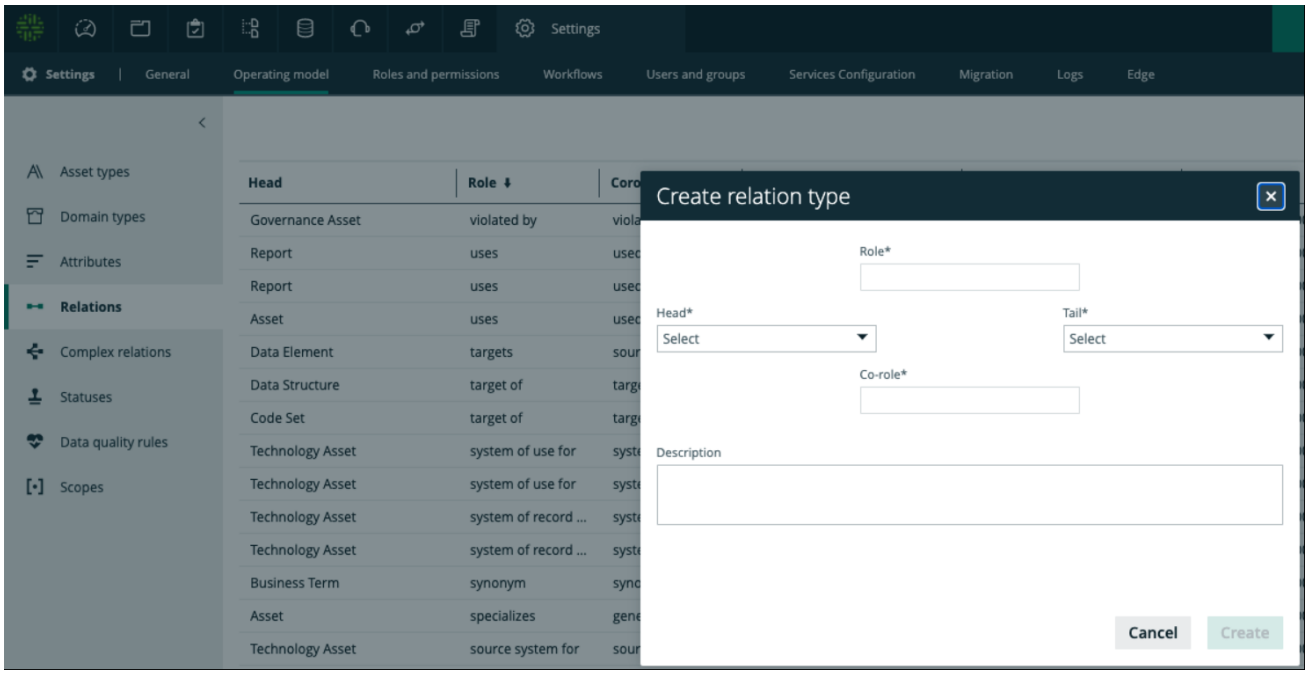

Settings
1/1
Create
Search
Settings | General | Operating model | Roles an
Edit dq-path-table
Asset types
Domain types
Attributes
Relations
Complex relations
Statuses
Data quality rules
Scopes
Name
dq-path-table
dq-path-column
Name*
dq-path-table
Description
Aggregation Path
Path*
Asset governed by Governance Asset
Data Quality Rule executed by Data Quality Metric
Construct the path through which you want to calculate your aggregated scores.
Categorization*
classified by Data Quality Dimension
We can categorize all Rule elements based on a relation type. You will have the ability to see subscores on the asset types based on these categories.
Metrics
Metric
Description
Operation
Rows Passed
The amount of rows that passed the rule.
Total
Cancel
Save

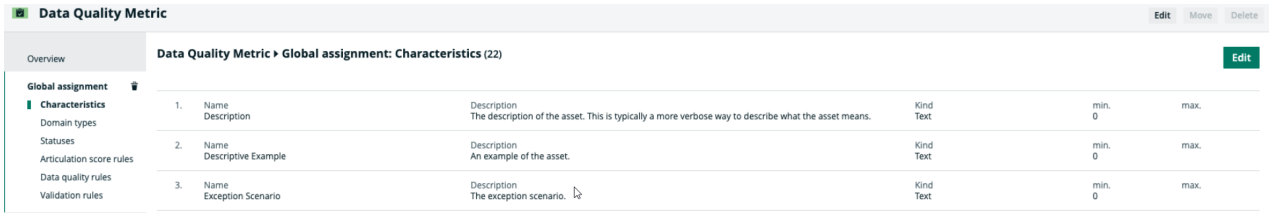

Settings
1/1
Create
Search
Settings | General | Operating model | Roles an
Edit dq-path-column
Asset types
Domain types
Attributes
Relations
Complex relations
Statuses
Data quality rules
Scopes
Name
dq-path-table
dq-path-column
Name*
dq-path-column
Description
Aggregation Path
Path*
Data Element is governed by Data Quality Rule
Data Quality Rule executed by Data Quality Metric
Construct the path through which you want to calculate your aggregated scores.
Categorization*
classified by Data Quality Dimension
We can categorize all Rule elements based on a relation type. You will have the ability to see subscores on the asset types based on these categories.
Metrics
Metric
Description
Operation
Rows Passed
The amount of rows that passed the rule.
Total
Cancel
Save

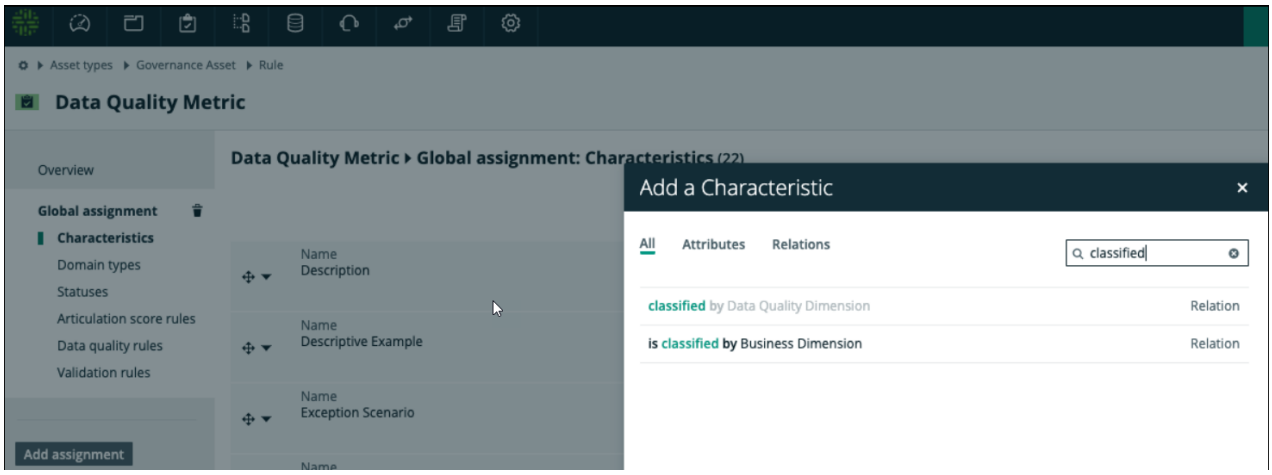


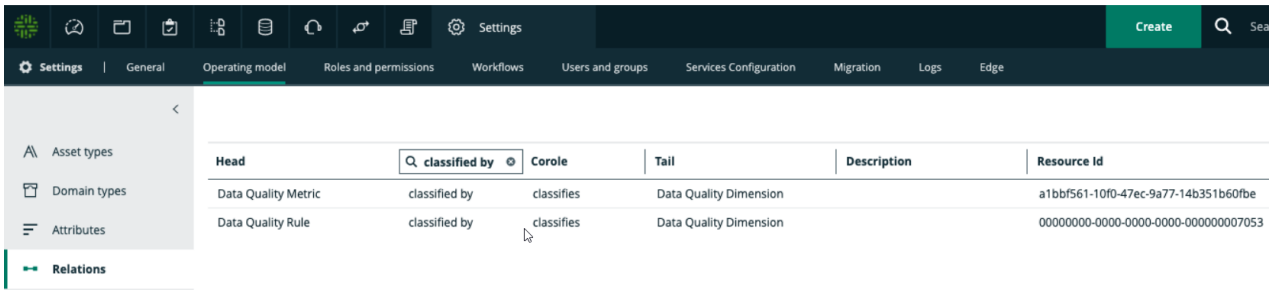
Q: DQ Dashboard In DGC: Why won't my DQ Dimension charts display in my Dashboard?

A: Please 1) add a new custom **Relation** 'Data Quality Metric classified by Data Quality Dimension', 2) **Global Assignment** for 'Data Quality Metric', 3) UUID of the new **Relation** into the **DQ Connector** setup in **Step 1G**, 4).









Q: I've connected and configured data sources correctly, why aren't DQ Rules and DQ Metrics being synchronized?

A: Please ensure Connection / System Names between Collibra DQ, Collibra, and Edge exactly match.

A: Please ensure Edge user has admin permissions to write the assets into Data Catalog.

A: Please ensure correct URL specified within the DQ Connector capability e.g. http://cdq.customer.com:9000/.

Q: Is DQ Connector unidirectional?

A: Yes, from Collibra DQ to Data Catalog in Collibra Data Intelligence Cloud.

Q: How many DQ Connectors can I run simultaneously?

A: Currently, one.

Q: Does the DQ Connector work with On-Prem Collibra DGC?

A: No, any work with on-prem Collibra DGC would be custom API development via Collibra Professional Services or a partner SI.

Q: If I delete a rule from Collibra DQ that I have already synchronized into Data Catalog, will it be deleted from Catalog in the next synchronization?

A: No, the DQ Connector only upserts into Data Catalog. If a rule is deleted from Collibra DQ, it will not be automatically deleted in Data Catalog.

Q: Why are my scores different in Collibra DQ and Data Catalog?

A: Currently, the DQ Connector pulls in the most recent user-defined rules from Collibra DQ. Other components that affect score such as Behaviors, Outliers, Patterns, Dupes, Source are not yet included.

Q: Getting errors when trying to delete both domain that Edge created for DB and the Connection?

A: Please delete Edge created domain via API.

Q: I've hit the synchronize button, how can I tell if my job is complete?

A: Check the Activities circle (button on top right of menu) for the status of your DQ Synchronization.

Q: Why did the rule with joins between two views that I created in Collibra DQ fail to import into Collibra Data Intelligence Cloud?

A: Because the column from the secondary view is flagged as primary, the rule maps the secondary column to the primary view. This causes the rule to import incorrectly, as the primary column does not exist as the primary view. A known workaround for this is to not select a primary column for this rule, and instead write the rule expression, including the columns required from both the primary and secondary views.