Assessing Data Quality with Dataplex

Overview

This lab demonstrates how to assess data quality in Google Cloud using **the Dataplex Universal Catalog**. The objective is to run automated data quality checks on a Big Query dataset using a YAML-based configuration and review the results to identify issues in the dataset.

Dataplex helps organize, secure, and catalog data at scale. A key feature is the ability to run **data quality tasks**, which is crucial for maintaining clean and reliable datasets for analytics and machine learning pipelines.

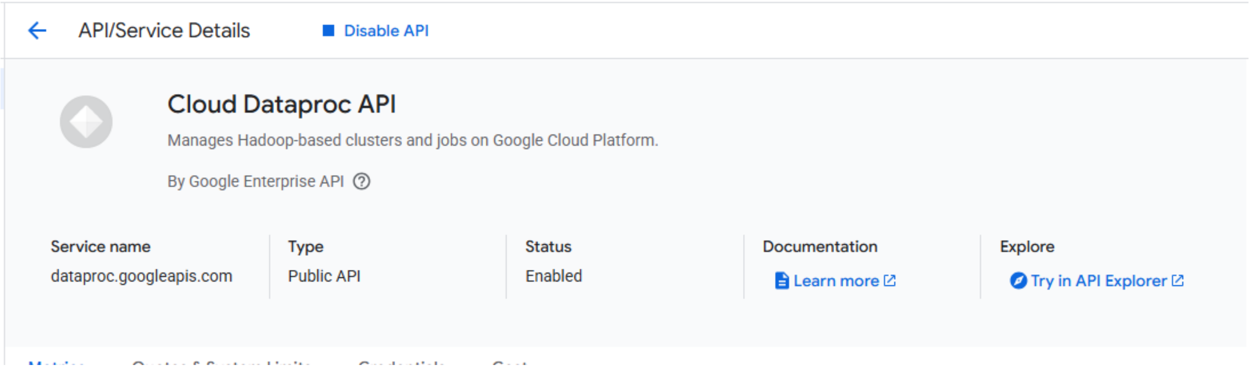
Objectives

- Create Dataplex resources: **Lake**, **Zone**, and **Asset**
- Query Big Query data to identify potential data quality issues
- Define and upload a **CloudDQ YAML specification file**
- Create and run a **data quality task**
- Review and interpret **data quality results**

Tools & Technologies

Tools & Technologies

- **Google Cloud Platform**
  - Dataplex Universal Catalog
  - BigQuery
  - Cloud Storage
  - Cloud Shell
- **CloudDQ YAML specification**
- **gcloud CLI**
- **SQL**



API/Service Details

Cloud Dataproc API

Manages Hadoop-based clusters and jobs on Google Cloud Platform.

By Google Enterprise API

Service name	Type	Status	Documentation	Explore
dataproc.googleapis.com	Public API	Enabled	Learn more	Try in API Explorer

Task 1. Create a lake, zone, and asset in Dataplex

To define and run data quality tasks, you first need to create some Dataplex Universal Catalog resources.

In this task, you create a new Dataplex Universal Catalog lake to store ecommerce customer information, add a raw zone to the lake, and then attach a pre-created BigQuery dataset as a new asset in the zone.

Create a lake

1. In the Google Cloud Console, in the Navigation menu (☰)> View All Products, navigate to Analytics > Dataplex Universal Catalog.

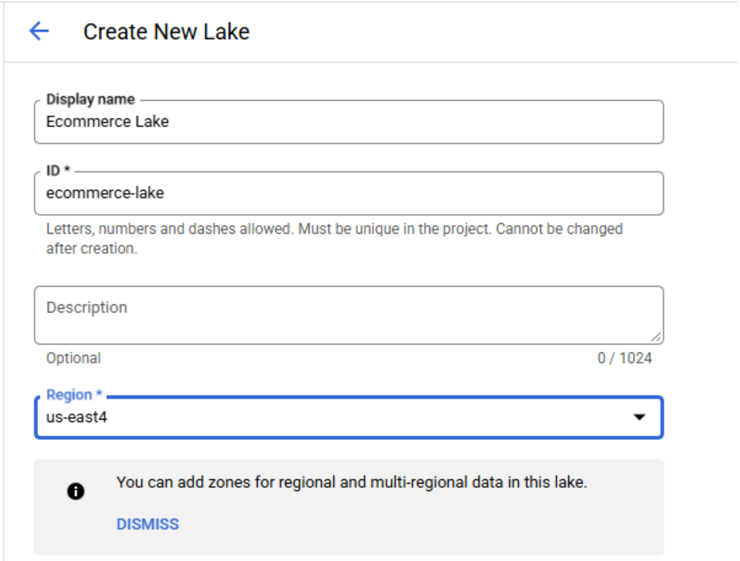
   If prompted Welcome to the new Dataplex Universal Catalog experience, click Close.

2. Under Manage lakes, click Manage.
3. Click Create lake.
4. Enter the required information to create a new lake:

Property	Value
Display Name	Ecommerce Lake
ID	Leave the default value.
Region	_____

   Leave the other default values.

5. Click Create.



Create New Lake

Display name: Ecommerce Lake

ID *: ecommerce-lake

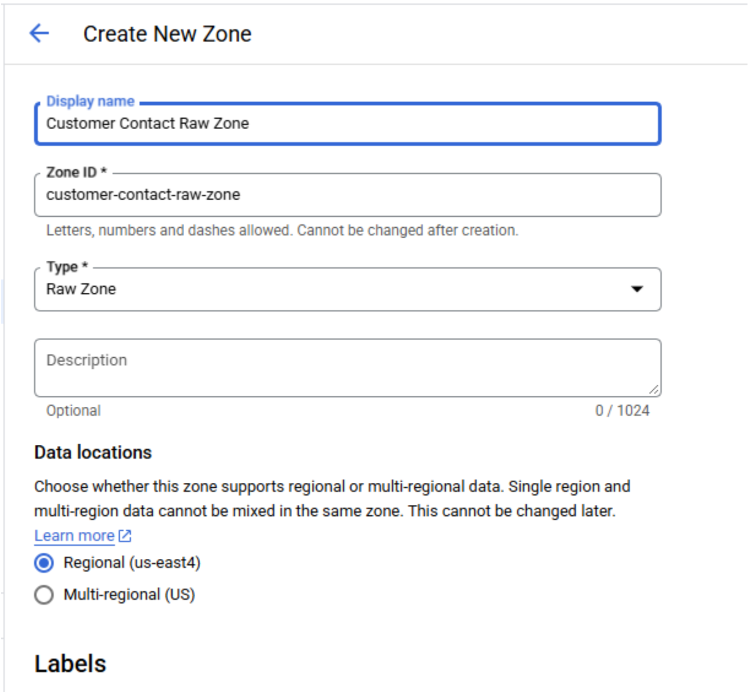
Letters, numbers and dashes allowed. Must be unique in the project. Cannot be changed after creation.

Description

Optional 0 / 1024

Region *: us-east4

You can add zones for regional and multi-regional data in this lake.



Create New Zone

Display name: Customer Contact Raw Zone

Zone ID *: customer-contact-raw-zone

Letters, numbers and dashes allowed. Cannot be changed after creation.

Type *: Raw Zone

Description

Optional 0 / 1024

Data locations

Choose whether this zone supports regional or multi-regional data. Single region and multi-region data cannot be mixed in the same zone. This cannot be changed later.

● Regional (us-east4)

○ Multi-regional (US)

Labels

Add a zone to the lake

1. On the **Manage** tab, click on the name of your lake.
2. Click **+ADD ZONE**.
3. Enter the required information to create a new zone:

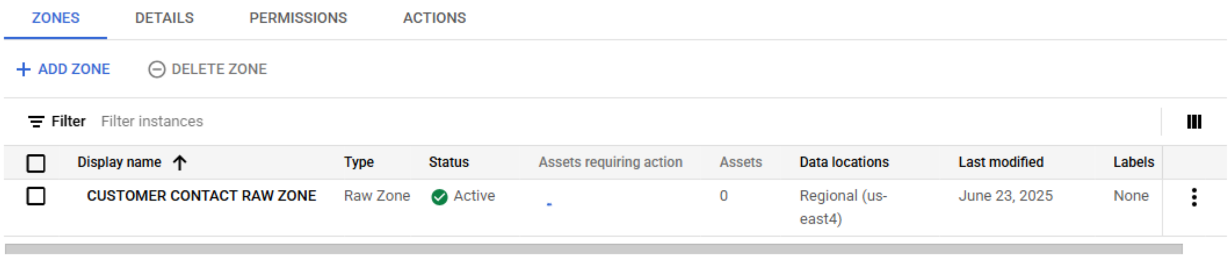
Property	**Value**

Display Name	Customer Contact Raw Zone
ID	Leave the default value.
Type	**Raw zone**
Data locations	**Regional**

Leave the other default values.

For example, the option for **Enable metadata discovery** under **Discovery settings** is enabled by default and allows authorized users to discover the data in the zone.

4. Click **Create**.



□	Display name ↑	Type	Status	Assets requiring action	Assets	Data locations	Last modified	Labels
□	CUSTOMER CONTACT RAW ZONE	Raw Zone	✓ Active	-	0	Regional (us-east4)	June 23, 2025	None

Attach an asset to a zone

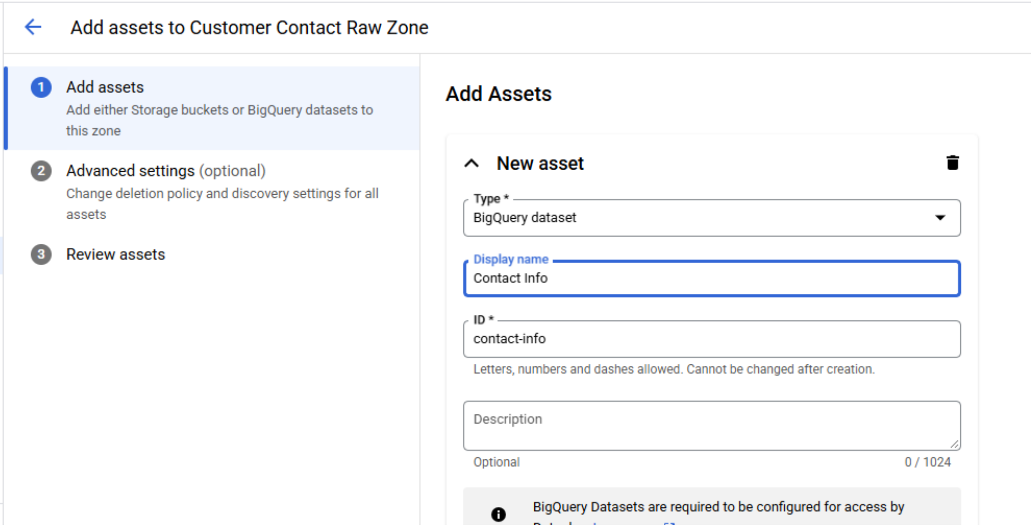
1. On the **Zones** tab, click on the name of your zone.
2. On the **Assets** tab, click **+ADD ASSET**.
3. Click **Add an asset**.
4. Enter the required information to attach a new asset:

Property	**Value**
Type	**BigQuery dataset**
Display Name	Contact Info
ID	Leave the default value.

Dataset	____.customers

Leave the other default values.

5. Click **Done**.
6. Click **Continue**.
7. For **Discovery settings**, select **Inherit** to inherit the Discovery settings from the zone level, and then click **Continue**.
8. Click **Submit**.



Add assets to Customer Contact Raw Zone

1 Add assets
Add either Storage buckets or BigQuery datasets to this zone

2 Advanced settings (optional)
Change deletion policy and discovery settings for all assets

3 Review assets

Add Assets

New asset

Type *: BigQuery dataset

Display name: Contact Info

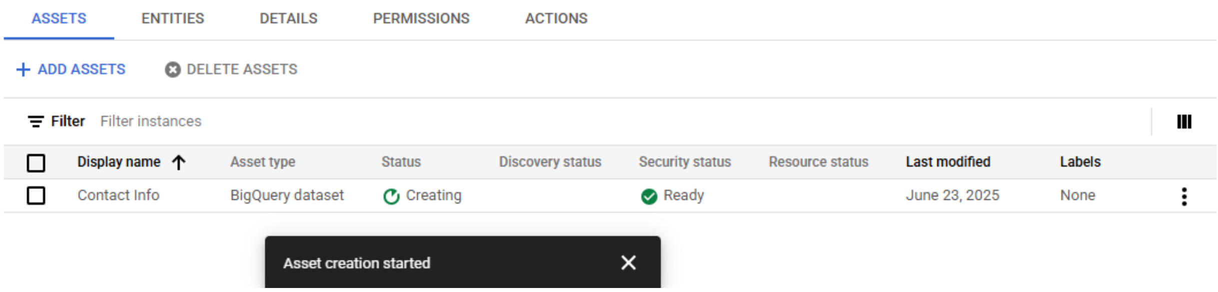
ID *: contact-info

Letters, numbers and dashes allowed. Cannot be changed after creation.

Description

Optional 0 / 1024

BigQuery Datasets are required to be configured for access by



□	Display name ↑	Asset type	Status	Discovery status	Security status	Resource status	Last modified	Labels
□	Contact Info	BigQuery dataset	Creating		Ready		June 23, 2025	None

Asset creation started

Task 2. Query a BigQuery table to review data quality

In the previous task, you created a new Dataplex Universal Catalog asset from a BigQuery dataset named **customers** that has been pre-created for this lab. This dataset contains a table named **contact_info** which contains raw contact information for customers of a fictional ecommerce company.

In this task, you query this table to start identifying some potential data quality issues that you can include as checks in a data quality job. You also identify another precreated dataset that you can use to store data quality job results in a later task.

1. In the Google Cloud Console, in the **Navigation menu** (☰), navigate to **BigQuery**.
2. In the Explorer pane, expand the arrow next to your project ID to list the contents: ____

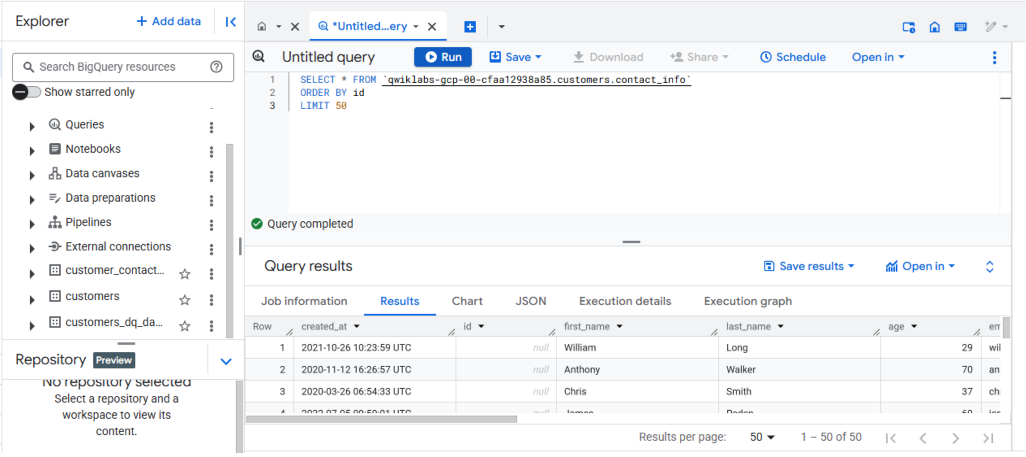
In addition to the **customer_contact_raw_zone** dataset created by Dataplex Universal Catalog to manage that zone, there are two BigQuery datasets that were precreated for this lab:

- customers
- customers_dq_dataset

The dataset named **customers** contains one table named **contact_info**, which contains contact information for customers such as a customer ID, name, email, and more. This is the table that you explore and check for data quality issues throughout this lab.

The dataset named **customers_dq_dataset** does not contain any tables. When you define a data quality job in a later task, you use this dataset as the destination for a new table containing the data quality job results.

3. In the SQL Editor, click on **+ SQL query**. Paste the following query, and then click **Run**:



```
SELECT * FROM `qwiklabs-gcp-00-cfaa12938a85.customers.contact_info`
ORDER BY id
LIMIT 50
```

Row	created_at	id	first_name	last_name	age	em
1	2021-10-26 10:23:59 UTC	null	William	Long	29	wil
2	2020-11-12 16:26:57 UTC	null	Anthony	Walker	70	an
3	2020-03-26 06:54:33 UTC	null	Chris	Smith	37	ch

Task 3. Create and upload a data quality specification file

Dataplex data quality check requirements are defined using CloudDQ YAML specification files. Once created, the YAML specification file is uploaded to a Cloud Storage bucket that is made accessible to the data quality job.

The YAML file has four keys sections:

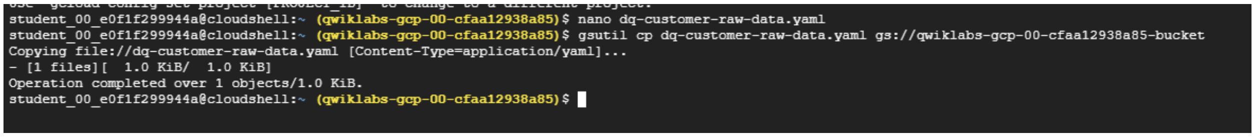
- a list of rules to run (either pre-defined or customized rules)
- row filters to select a subset of data for validation
- rule bindings to apply the defined rules to the table(s)
- optional rule dimensions to specify the types of the rules that the YAML file can contain

In this task, you define a new YAML specification file for data quality checks that identify null customer IDs and emails in the specified BigQuery table. After you define the file, you upload it to a pre-created Cloud Storage bucket for use in a later task to run the data quality job.

Create the data quality specification file

1. In Cloud Shell, run the following command to create a new empty file for the data quality specification:

nano dq-customer-raw-data.yaml

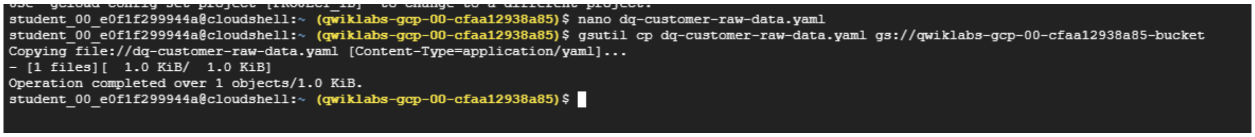


```
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
student_00_e0f1f299944a@cloudshell:~ (qwiklabs-gcp-00-cfaa12938a85)$ nano dq-customer-raw-data.yaml
student_00_e0f1f299944a@cloudshell:~ (qwiklabs-gcp-00-cfaa12938a85)$ gsutil cp dq-customer-raw-data.yaml gs://qwiklabs-gcp-00-cfaa12938a85-bucket
Copying file://dq-customer-raw-data.yaml [Content-Type=application/yaml]...
- [1 files][  1.0 KiB/  1.0 KiB]
Operation completed over 1 objects/1.0 KiB.
student_00_e0f1f299944a@cloudshell:~ (qwiklabs-gcp-00-cfaa12938a85)$ 
```

Upload the file to Cloud Storage

- In Cloud Shell, run the following command to upload the file to a Cloud Storage bucket that has been created for this lab:

gsutil cp dq-customer-raw-data.yaml gs://Project ID-bucket



```
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
student_00_e0f1f299944a@cloudshell:~ (qwiklabs-gcp-00-cfaa12938a85)$ nano dq-customer-raw-data.yaml
student_00_e0f1f299944a@cloudshell:~ (qwiklabs-gcp-00-cfaa12938a85)$ gsutil cp dq-customer-raw-data.yaml gs://qwiklabs-gcp-00-cfaa12938a85-bucket
Copying file://dq-customer-raw-data.yaml [Content-Type=application/yaml]...
- [1 files][  1.0 KiB/  1.0 KiB]
Operation completed over 1 objects/1.0 KiB.
student_00_e0f1f299944a@cloudshell:~ (qwiklabs-gcp-00-cfaa12938a85)$ 
```

Task 4. Define and run a data quality job in Dataplex

The data quality process uses a data quality specification YAML file to run a data quality job and generates data quality metrics that are written to a BigQuery dataset.

In this task, you define and run a data quality job using the data quality specification YAML file uploaded to Cloud Storage in the previous task. When you define the job, you also specify a pre-created BigQuery dataset named **customer_dq_dataset** to store the data quality results.

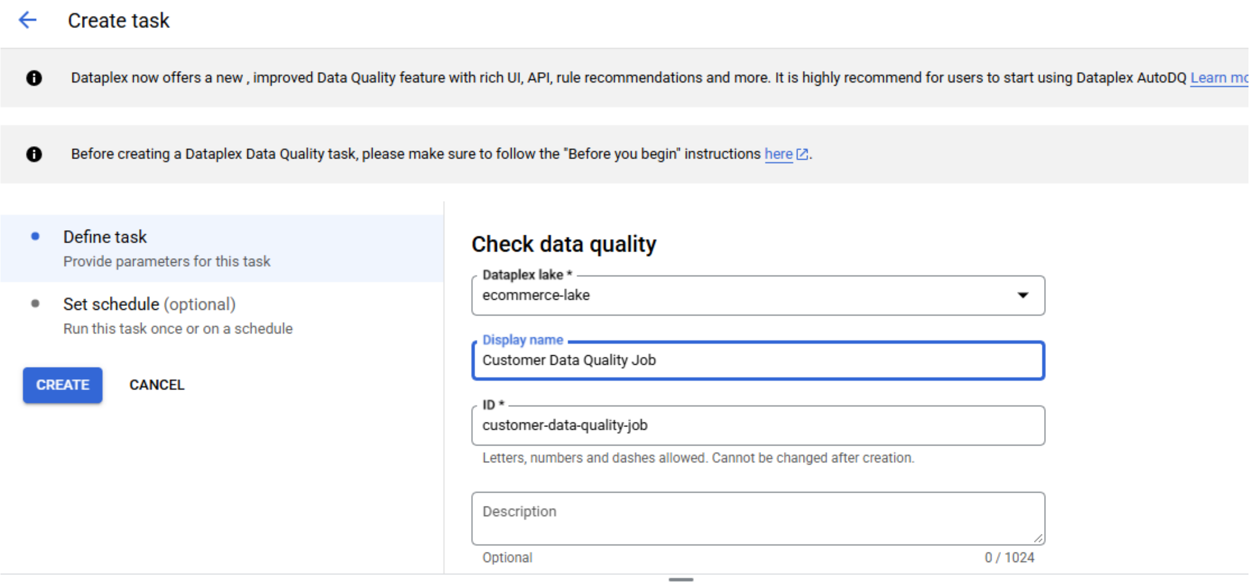
1. In the Google Cloud Console, in the **Navigation menu** (≡)> View All Products, navigate to **Analytics** > **Dataplex Universal Catalog**.
2. Under **Manage lakes**, click **Process**.
3. Click **+CREATE TASK**.
4. Under Check Data Quality, click **Create task**.
5. Enter the required information to create a new data quality job:

Property	**Value**
Dataplex lake	**ecommerce-lake**
Display name	Customer Data Quality Job
ID	Leave the default value.
Select GCS file	____-bucket/dq-customer-raw-data.yaml
Select BigQuery dataset	____.customers_dq_dataset
BigQuery table	dq_results
User service account	**Compute Engine default service account**

Leave the other default values.

Note that the Compute Engine default service account has been preconfigured for this lab to have the appropriate IAM roles and permissions. For more information, review the Dataplex Universal Catalog documentation titled Create a service account.

6. Click **Continue**.
7. For **Start**, select **Immediately**.
8. Click **Create**.



Create task

Dataplex now offers a new , improved Data Quality feature with rich UI, API, rule recommendations and more. It is highly recommend for users to start using Dataplex AutoDQ Learn mo

Before creating a Dataplex Data Quality task, please make sure to follow the "Before you begin" instructions here.

Define task
Provide parameters for this task
Set schedule (optional)
Run this task once or on a schedule
CREATE
CANCEL

Check data quality

Dataplex lake *: ecommerce-lake

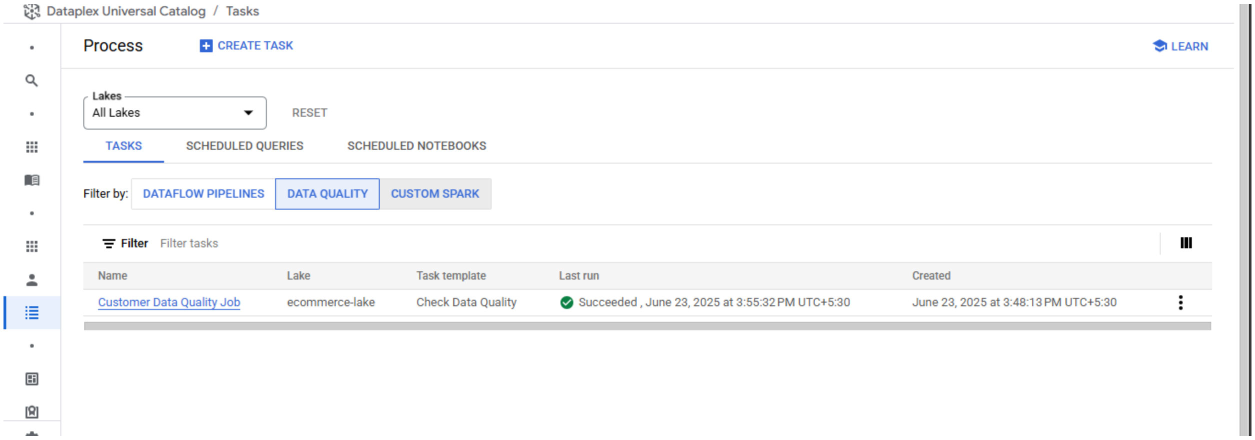
Display name: Customer Data Quality Job

ID *: customer-data-quality-job

Letters, numbers and dashes allowed. Cannot be changed after creation.

Description

Optional 0 / 1024



Dataplex Universal Catalog / Tasks

Lakes: All Lakes

Filter by: DATA QUALITY

Name	Lake	Task template	Last run	Created
Customer Data Quality Job	ecommerce-lake	Check Data Quality	✓ Succeeded , June 23, 2025 at 3:55:32 PM UTC+5:30	June 23, 2025 at 3:48:13 PM UTC+5:30

Task 5. Review data quality results in BigQuery

In this task, you review the tables in the **customers_dq_dataset** to identify records that are missing customer ID values or have an invalid values for emails.

1. In the Google Cloud Console, in the **Navigation menu** (☰), navigate to **BigQuery**.
2. In the Explorer pane, expand the arrow next to your project ID to list the contents: ____
3. Expand the arrow next to the **customer_dq_dataset** dataset.
4. Click on the **dq_summary** table.
5. Click on the **Preview** tab to see the results.

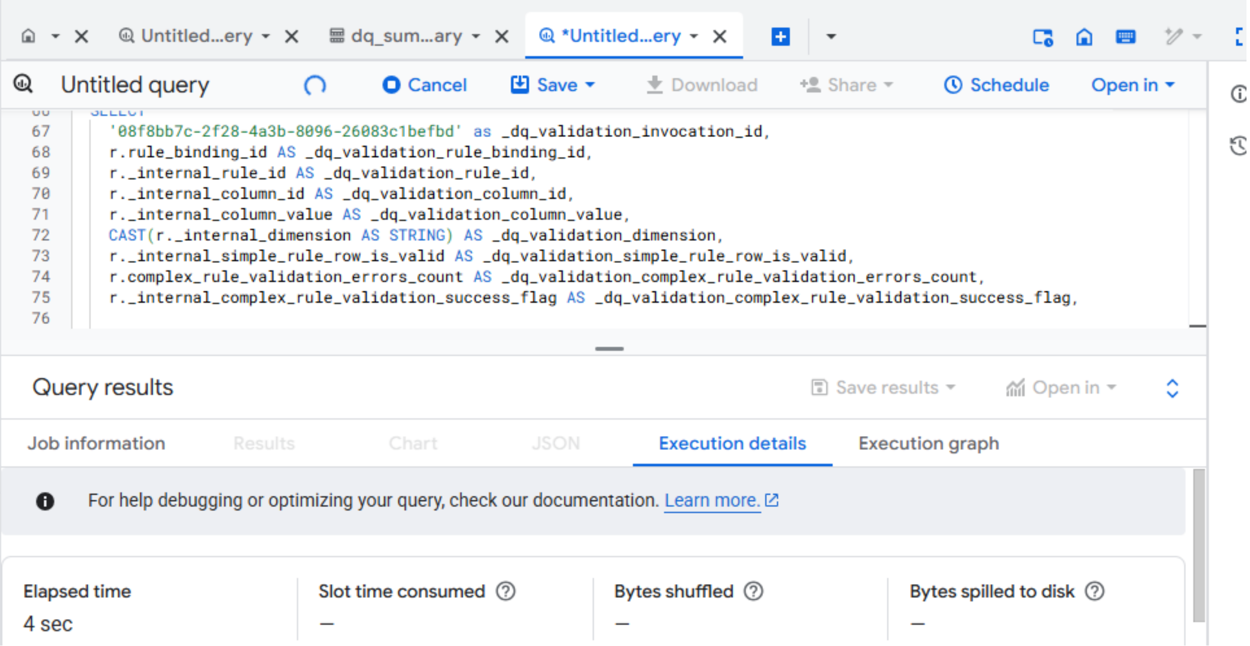
The **dq summary** table provides useful information about the overall data quality including the number of records that were identified to not adhere to the two rules in the data quality specification file.

6. Scroll to the last column named **failed_records_query**.

7. Click on the down arrow in the first row to expand the text and view the entire query for the **VALID_EMAIL** rule results.

   Note that the query is quite long and ends with ORDER BY _dq_validation_rule_id.

8. Click on **+ SQL query**. Copy and paste the query into SQL Editor, and click **Run**.



```
SELECT
  '08f8bb7c-2f28-4a3b-8096-26083c1befbd' as _dq_validation_invocation_id,
  r.rule_binding_id AS _dq_validation_rule_binding_id,
  r._internal_rule_id AS _dq_validation_rule_id,
  r._internal_column_id AS _dq_validation_column_id,
  r._internal_column_value AS _dq_validation_column_value,
  CAST(r._internal_dimension AS STRING) AS _dq_validation_dimension,
  r._internal_simple_rule_row_is_valid AS _dq_validation_simple_rule_row_is_valid,
  r.complex_rule_validation_errors_count AS _dq_validation_complex_rule_validation_errors_count,
  r._internal_complex_rule_validation_success_flag AS _dq_validation_complex_rule_validation_success_flag,
```

For help debugging or optimizing your query, check our documentation. Learn more.

Elapsed time	Slot time consumed	Bytes shuffled	Bytes spilled to disk
4 sec	–	–	–

Learnings & Reflections

- Learned how Dataplex integrates metadata management with data quality.
- Understood how to define **reusable** and **modular** rules via YAML.
- Gained insights into integrating Cloud Storage, BigQuery, and Dataplex for automated data validation pipelines.
- Saw how **rule dimensions** (e.g., completeness, conformance) help categorize data issu

Conclusion

This lab effectively demonstrates the power of **Dataplex** in monitoring and improving data quality within the Google Cloud ecosystem. It enables organizations to create transparent, automated, and scalable validation workflows, reducing the risk of data corruption and increasing trust in data-driven decisions.