Big Query: Qwik Start – Console

1. Objective

The goal of this hands-on lab was to explore the capabilities of **Google BigQuery**, an enterprise data warehouse for fast SQL analytics on large datasets. The lab covered:

- Querying a public dataset.

- Creating a dataset and loading custom data.

- Executing SQL queries on the loaded dataset.

2. Tools & Environment

- **Google Cloud Platform (GCP)**

- **BigQuery Console**

- **Cloud Shell (for CLI access and authentication)**

- **Temporary student credentials** (to avoid personal account charges)

Task 1. Open BigQuery

The BigQuery console provides an interface to query tables, including public datasets offered by BigQuery. The query you will run accesses a table from a public dataset that BigQuery provides. It uses standard query language to search the dataset, and limits the results returned to 10.
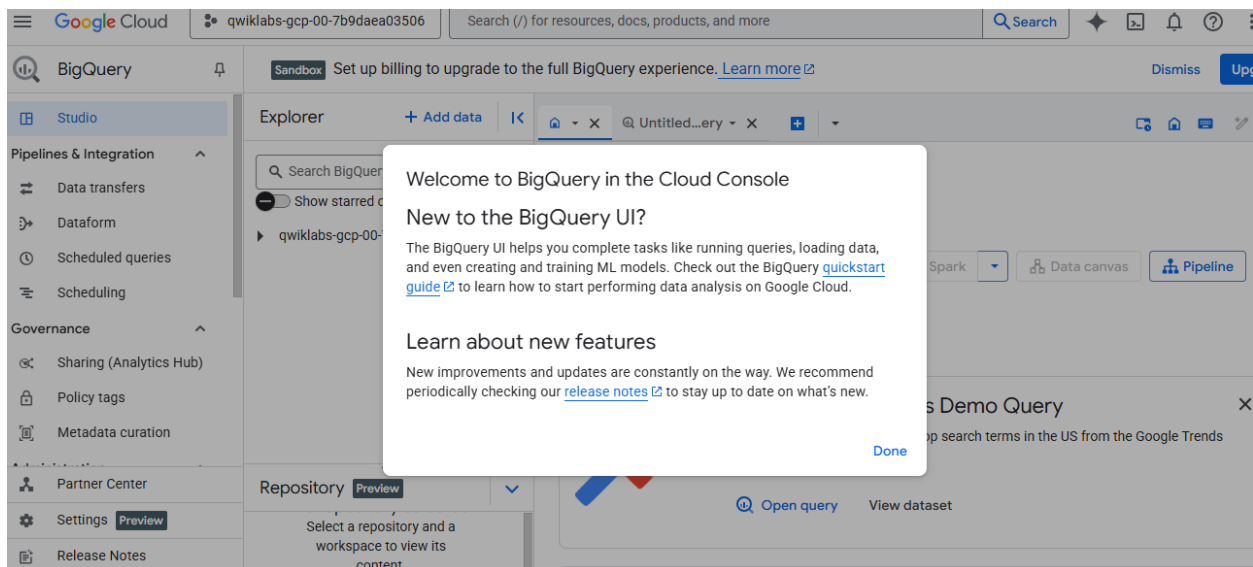
Open the BigQuery console

1. In the Google Cloud Console, select **Navigation menu** > **BigQuery**.

The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and the release notes.
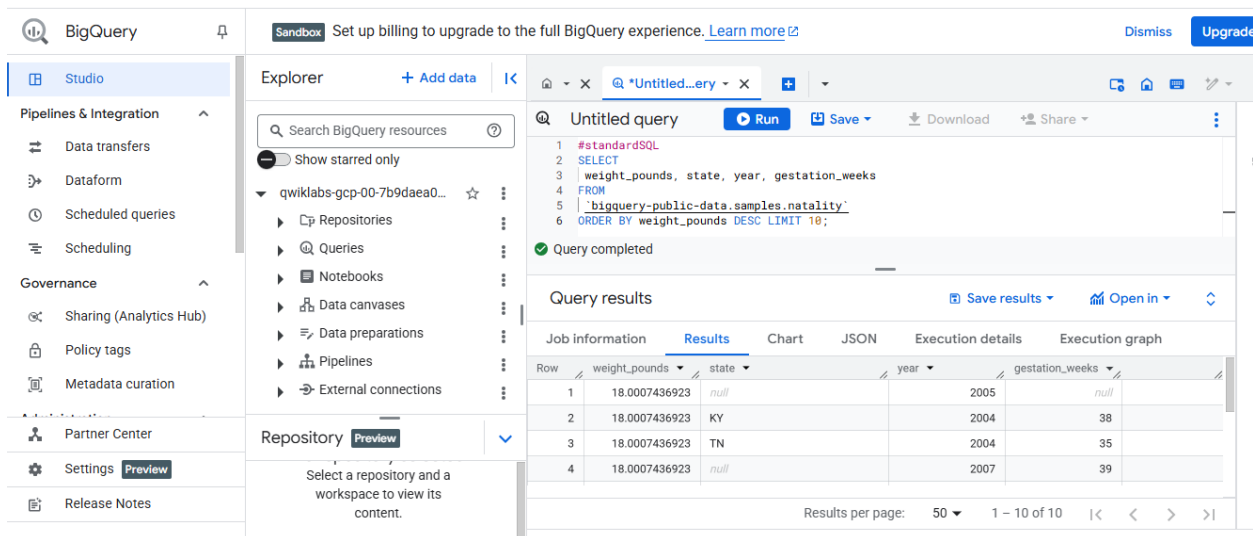
2. Click **Done**.

The BigQuery console opens.

Task 2. Query a public dataset

1. Click **+** (SQL query) to create a new query. Copy and paste the following query into the BigQuery Query editor:

#standardSQL SELECT weight_pounds, state, year, gestation_weeks FROM `bigquery-public-data.samples.natality` ORDER BY weight_pounds DESC LIMIT 10;

This data sample holds information about US natality (birth rates).

A green or red check displays depending on whether the query is valid or invalid. If the query is valid, the validator also describes the amount of data to be processed after you run the query.

This information helps determine the cost to run a query.

2. Click the **Run** button.

Task 3. Create a new dataset

To load custom data into a table, you first need to create a BigQuery dataset.

Datasets help control access to tables and views in a project. This lab uses only one table, but you still need a dataset to hold the table.

1. In the **Explorer** pane, near your project id, click on **View actions** (⋮) then click **Create dataset**.

2. Set **Dataset ID** to **babynames**.

3. Leave all other fields at their default settings. Click **Create dataset**.

Task 4. Load data into a new table

Next you create a table inside the **babynames** dataset, then load the data file from your storage bucket into the new table.

The custom data file you'll use contains approximately 7 MB of data about popular baby names, provided by the US Social Security Administration.

1. In the Cloud Console, select **Navigation menu** > **BigQuery** to return to the BigQuery console.

2. Navigate to the **babynames** dataset, by clicking **View actions** (⋮) near your dataset then click **Create table**.

3. In the Create table dialog, set the following fields, leave all others at the default value:

Field	Value
Create table from	**Google Cloud Storage**
Select file from GCS bucket	spls/gsp072/baby-names/yob2014.txt
File format	**CSV**
Table	names_2014
Schema > **Edit as text**	Slide on, then add the following in the textbox: name:string,gender:string,count:integer

4. Click the **Create table** button.

Create table

Source

Create table from
Google Cloud Storage

Select file from GCS bucket or use a URI pattern ⬀ *
☑ spls/gsp072/baby-names/yob2014.txt   Browse ⑦

File format
CSV

☐ Source Data Partitioning

Destination

Project *
qwiklabs-gcp-00-7b9daea03506   Browse

Dataset *
babynames

Table *
names_2014

Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, and spaces are allowed.

Table type

Create table

babynames

Table *
names_2014

Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, and spaces are allowed.

Table type
Native table   ▼ ⑦

Schema

☐ Auto detect

🔵 Edit as text

Press Alt+F1 for Accessibility Options.

```
1    ame:string,gender:string,count:integer
```

Create table   Cancel

Task 5. Preview the table

Check your table! View the first few rows of the data.

- Click the names_2014 table in the left panel, then click **Preview**.

Task 6. Query a custom dataset

Running a query against custom data is identical to the querying a public dataset that you did earlier, except that now you're querying your own table instead of a public table.

1. In BigQuery, click the **+** (SQL query) icon at the top.

2. Paste or type the following query into the query **Editor**.

```
#standardSQL

SELECT

name, count

FROM

`babynames.names_2014`

WHERE

gender = 'M'

ORDER BY count DESC LIMIT 5;
```

3. Click the **Run** button. The query displays the top 5 boys names for the year of data (2014) you loaded into the table.

⊕ Untitled query   ▶ Run   ⊞ Save ▾   ⬇ Download   +⦿ Share ▾   ⋮

```
1  #standardSQL
2  SELECT
3    name, count
4  FROM
5    `babynames.names_2014`
6  WHERE
7    gender = 'M'
8  ORDER BY count DESC LIMIT 5;
```

✓ Query completed

Query results   ⊡ Save results ▾   ⋔ Open in ▾   ↕

Job information   **Results**   Chart   JSON   Execution details   Execution graph

Row	name ▾	count ▾	
1	Noah	19286	
2	Liam	18451	
3	Mason	17192	

                                    Results per page: 50 ▾   1 – 5 of 5 |< < > >|

⊕ Untitled query   ▶ Run   ⊞ Save ▾   ⬇ Download   +⦿ Share ▾   ⋮

```
1  #standardSQL
```

✓ Query completed

Query results   ⊡ Save results ▾   ⋔ Open in ▾   ↕

Job information   **Results**   Chart   JSON   Execution details   Execution graph

Row	name ▾	count ▾	
1	Noah	19286	
2	Liam	18451	
3	Mason	17192	
4	Jacob	16869	
5	William	16809	

Key Learnings

- BigQuery supports seamless querying of both public and custom datasets.

- Dataset and table creation workflows in BigQuery are user-friendly.

- SQL-based analytics is efficient even on large datasets.

- Integration with Cloud Storage for data ingestion is simple and effective.

Final Thoughts

This lab provided a practical introduction to Google BigQuery's capabilities. Key takeaways include:

- Efficient data querying using SQL on GCP infrastructure.

- Easy UI-based dataset and table management.

- Cloud-native data warehousing solutions reduce infrastructure overhead