



MASTER IN ENTREPRENEURSHIP  
INNOVATION MANAGEMENT  
IN COLLABORATION WITH **MIT SLOAN**

IN COLLABORATION WITH  
**MIT MANAGEMENT**  
SLOAN SCHOOL



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**PARTHENOPE**

MASTER MEIM 2021-2022

# Digital Strategies - Clustering use cases

## Google Cloud Platform

Lesson given by prof. Alessio Ferone

[www.meim.uniparthenope.it](http://www.meim.uniparthenope.it)



MASTER IN ENTREPRENEURSHIP  
INNOVATION MANAGEMENT  
IN COLLABORATION WITH MIT SLOAN



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
PARTHENOPE

# Digital Strategies

## Machine Learning with BigQuery ML

# Agenda

- **Google Cloud Platform**
- Google BigQuery
- Google BigQuery SQL
- Google BigQuery ML

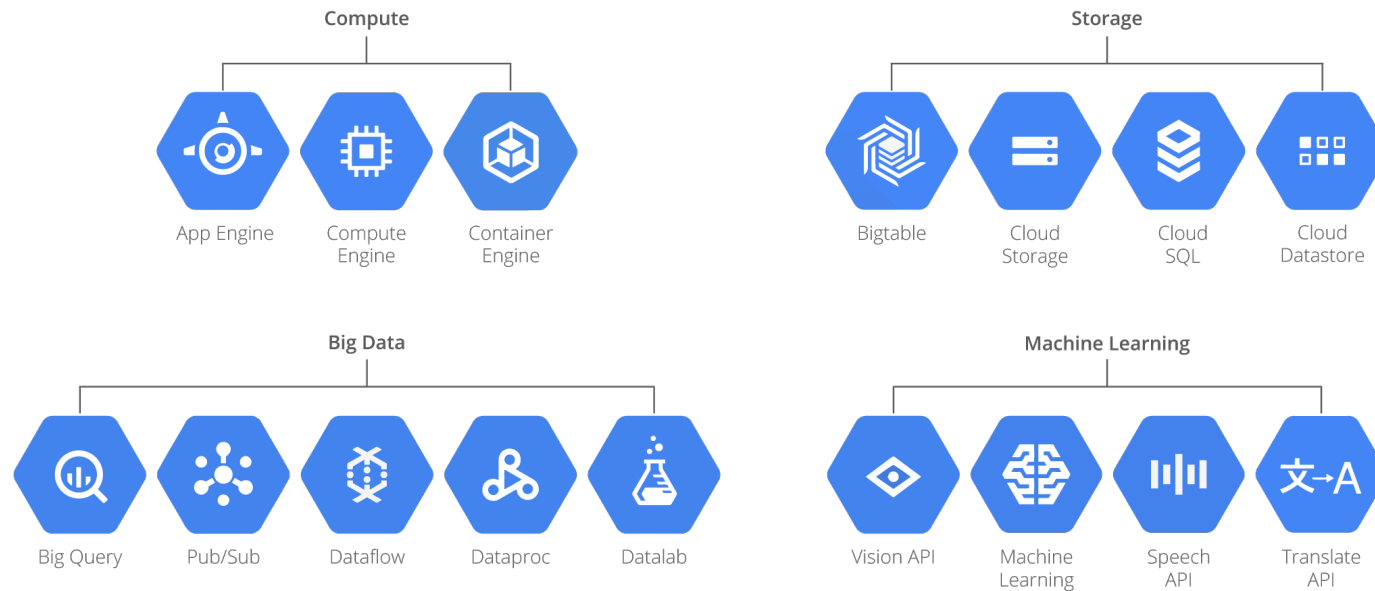
# Introduction

- The adoption of the **public cloud** enables companies and users to access **innovative technologies**
- In particular for **Big Data** and **Artificial Intelligence (AI)**
- To be effective the new **AI capabilities** need to be **shared** between **different roles**
- Most cloud providers are currently addressing the challenge of **democratizing AI**
- In this context, **Google Cloud** provides several services to handle and process large amount of data

# Google Cloud Platform

- Starting from 1998 with Google Search, Google developed one of the largest and powerful infrastructures in the world
- Gmail, YouTube, Maps
- In 2008 the infrastructure is opened to business customers launching **Google Cloud Platform (GCP)** -> <https://console.cloud.google.com>
- Services comprise:
  - **Compute:** virtual machines, containers, app engines
  - **Storage and Database:** Cloud Storage, Cloud SQL
  - **Networking:** Virtual Private Clouds
  - **Big Data:** DataProc, Hadoop, BigQuery
  - **AI and Machine Learning:** TensorFlow, AutoML, BQML
  - **IoT:** IoT Core

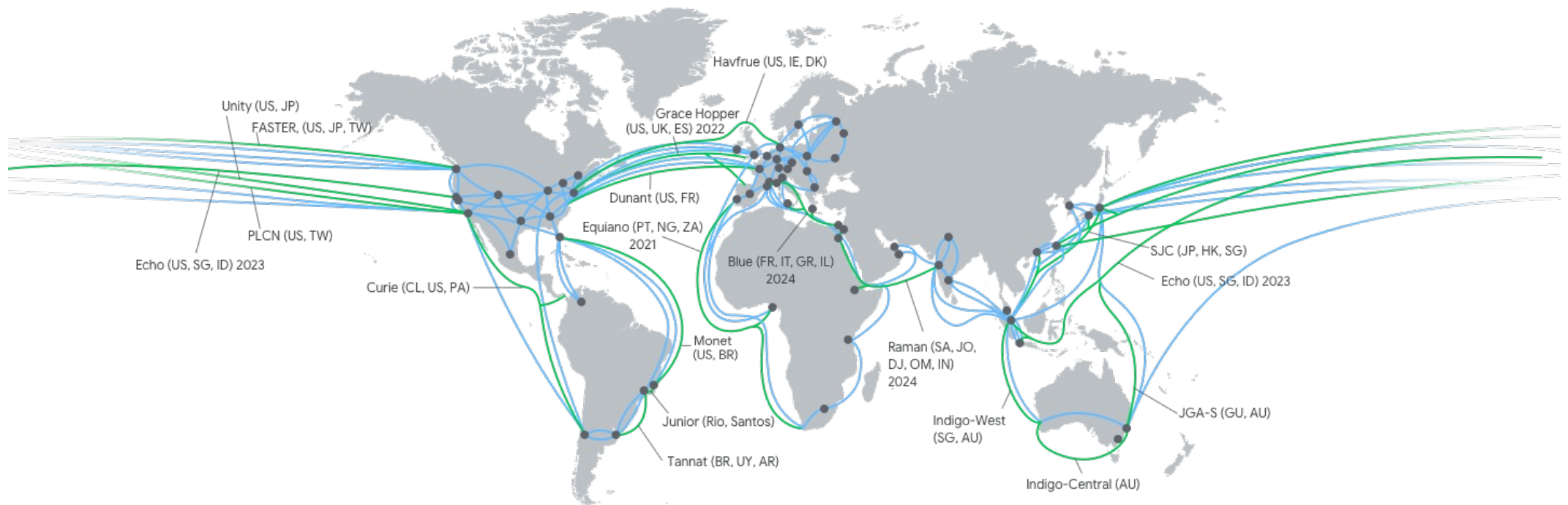
# Google Cloud Platform



# Google Cloud Platform: regions



# Google Cloud Platform: network

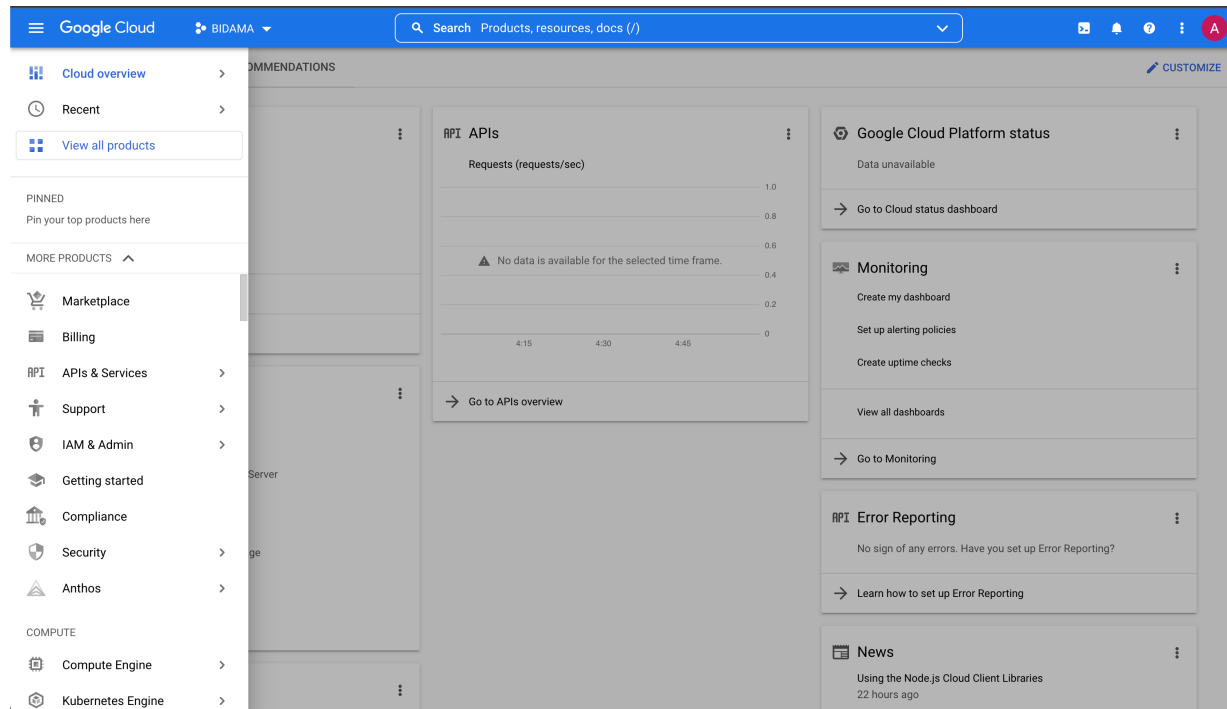




# Google Cloud Platform: advantages

- **Fully managed** and **serverless** services: no maintenance
- Leading research in the AI and ML
- **Google Cloud Console** is a web-based interface accessible from compatible web browser

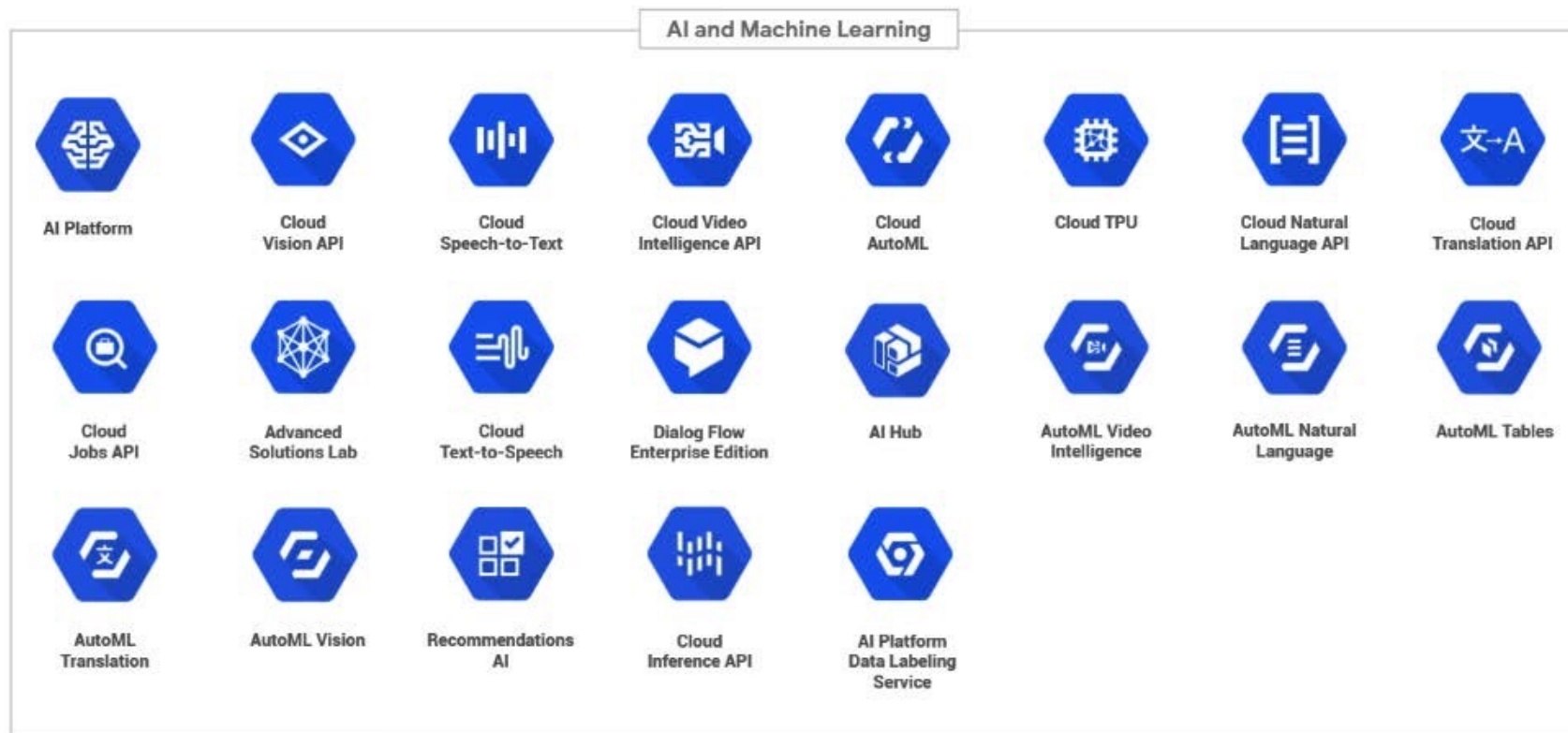
# Google Cloud Platform: advantages



# Google Cloud Platform: AI and ML

- AI and ML features are embedded within many Google product
  - **Google Maps:** predict arrival time
  - **Google Translate**
  - **YouTube:** recommend video to watch
  - **Google Photos:** recognize people, object, places
- GCP provides **services** to address all the steps in a typical **ML model**:
  - **Ingestion** and **preparation** of data
  - **Building** and **training** the model
  - **Evaluation** and **validation**
  - **Deployment** and **maintenance**

# Google Cloud Platform: AI and ML



# Google Cloud Platform: AI and ML

- **AI and ML services** can be divided into **3 categories**
  - **Core platform:** Infrastructure-as-a-Service (IaaS) approach to provide different processing units (CPU, GPU, TPU), Deep Learning VM Image, AI Platform and AI Platform notebooks
  - **AI building blocks:** AutoML, BigQuery ML
  - **Solution:** AI Hub (marketplace for AI components), Document AI (extracting relevant information from different types of documents)

# Agenda

- Google Cloud Platform
- **Google BigQuery**
- Google BigQuery SQL
- Google BigQuery ML

# BigQuery

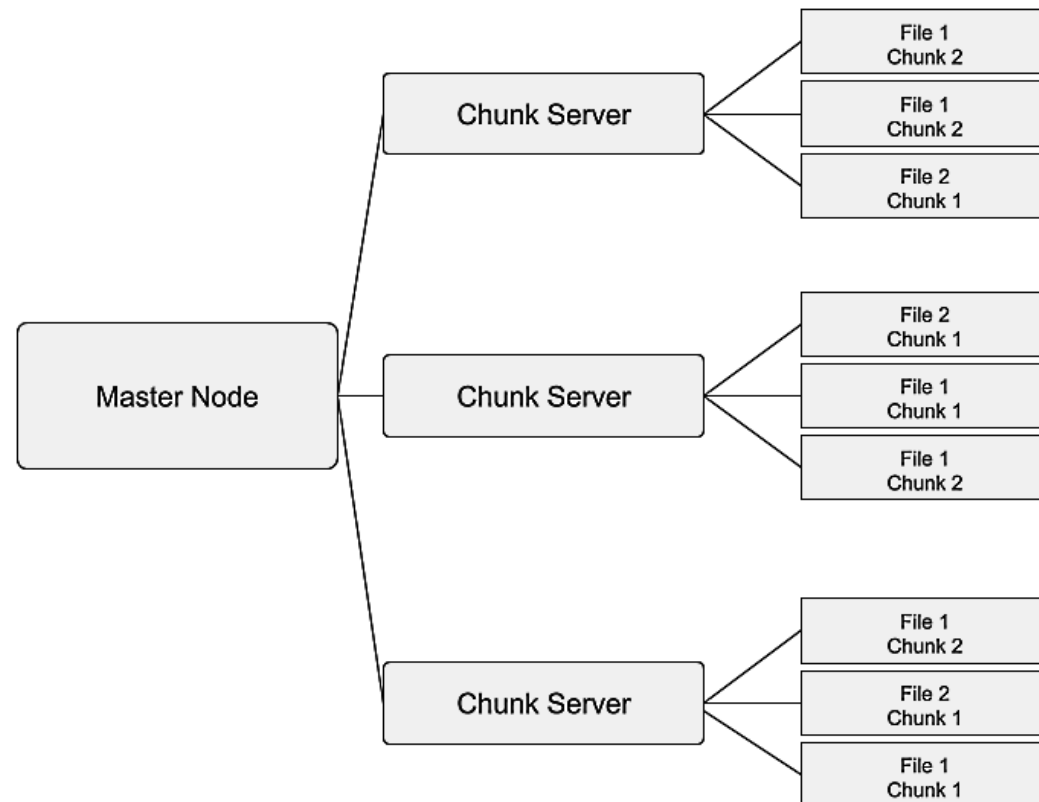
- Google BigQuery is a **highly scalable, serverless, distributed** data warehouse technology
- It can **store** petabytes of **data** and **query** them with high performance
  - Giga:  $10^9$  Tera:  $10^{12}$  Peta:  $10^{15}$
- Being **serverless**, users who store and query data on BigQuery don't have to **manage** the underlying **infrastructure**
- BigQuery has a **distributed architecture** running on thousands of nodes across Google's data centers
  - datasets are chunked and replicated across different regions to guarantee **maximum performance** and **availability**
- The **storage** and **compute** layers are fully **decoupled** in BigQuery

# BigQuery: storage

- BigQuery stores data in columnar format rather than in row format
- **Data** is stored in Google's proprietary distributed filesystem named **Google File System** (codename Colossus)
- Google File System is based on two different server types
  - **Master servers:** Nodes that don't store data but are responsible for managing the metadata of each file, such as the location and available number of replicas of each chunk that compose a file
  - **Chunk servers:** Nodes that actually store the chunks of files that are replicated across different servers



# BigQuery: storage



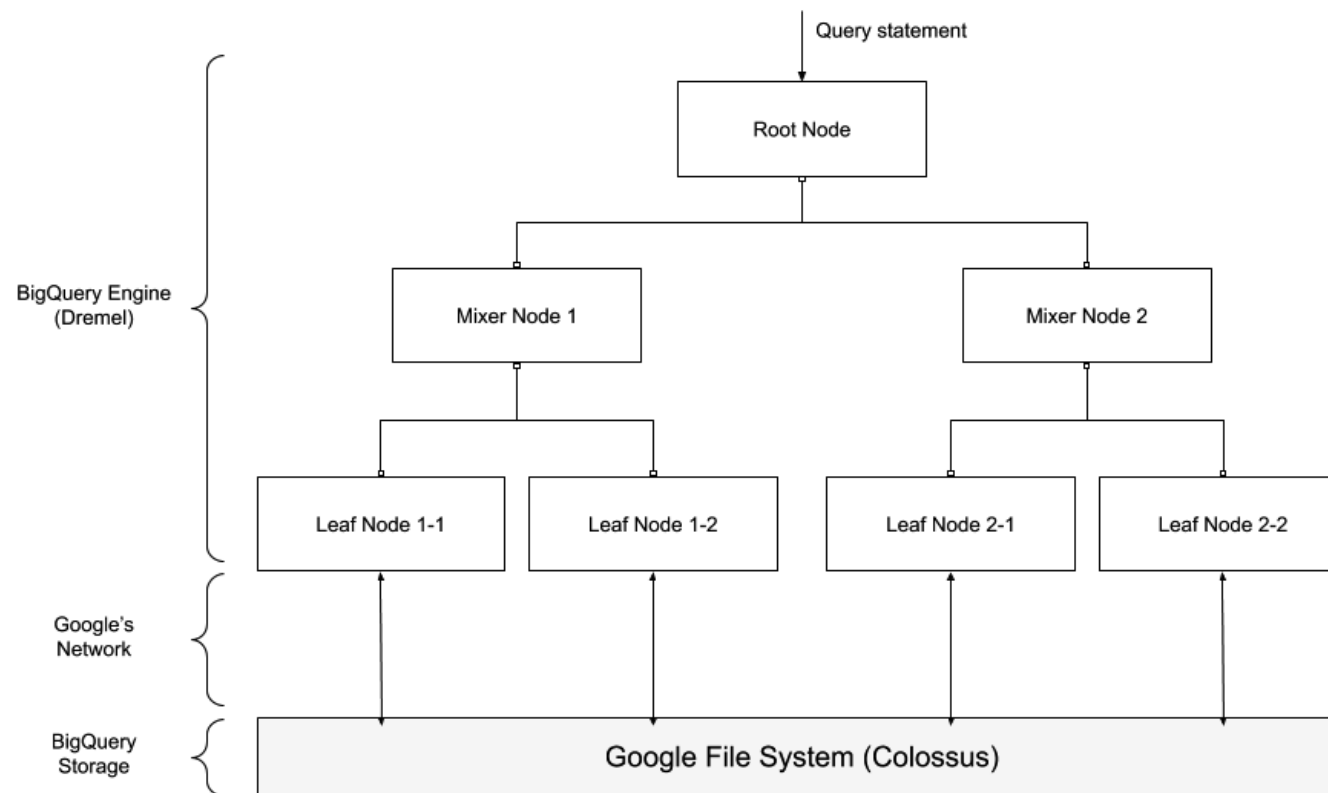
# BigQuery: compute

- The **compute layer** is responsible for **receiving query** statements from BigQuery users and **executing** them in the fastest way
- The **query engine** is based on Dremel, a technology developed by Google, that leverages a **multi-level tree architecture**

# BigQuery: compute

1. The **root node** of the tree **receives** the **query** to execute
2. The root node **splits** and **distributes** the query to **intermediate nodes** (mixers)
3. Mixer nodes rewrite queries before **passing** them to the **leaf nodes** or to other mixer nodes
4. **Leaf nodes** are responsible for parallelizing the **reading** of the chunks of **data** from Google File System
5. Leaf nodes **perform computations** on the data and eventually shuffle them across other leaf nodes
6. At the end of the computation, each leaf node **produces a result** that is returned to the parent node
7. When all the **results** are **returned** to the **root node**, the outcome of the query is sent to the user or application that requested the execution

# BigQuery: compute

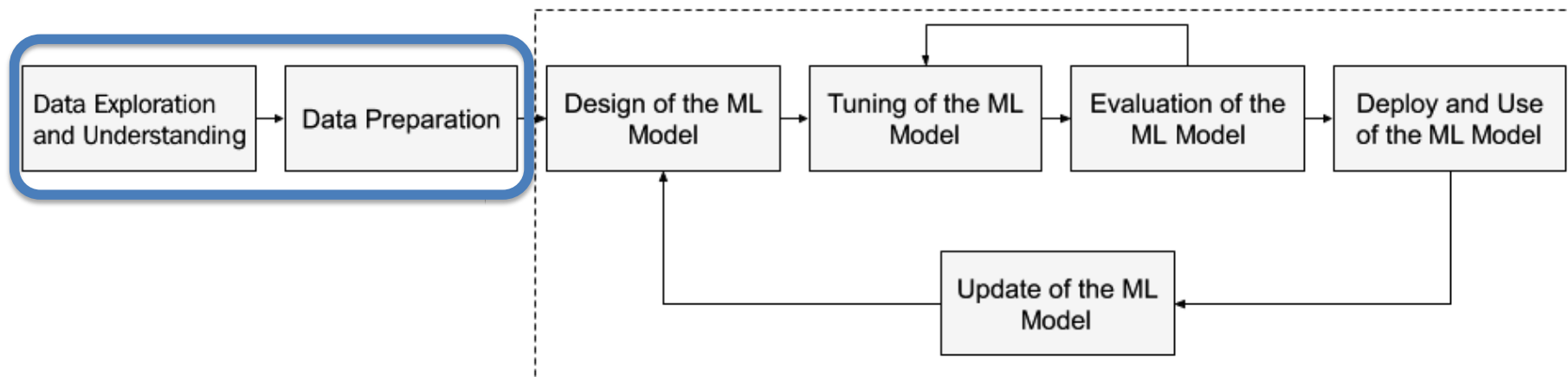


# BigQuery structure

- BigQuery structures (tables, views and ML models) are organized in **datasets**
- Each dataset is a **container** for different **structures** and can be used to control access to underlying data structures
- A **dataset** is directly linked to the following:
  - A **GCP project** that hosts the dataset and is linked to the billing account
  - A geographic **location**
  - A **name** assigned to the dataset that should be unique in the GCP project

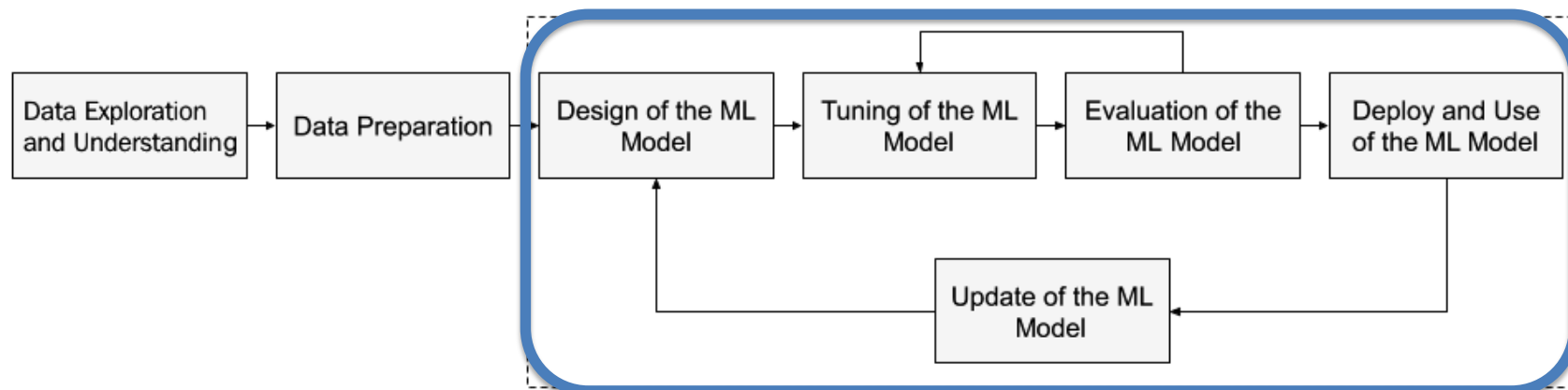
# ML model development

- The first two steps involve preliminary raw data analyses and operations:
  - **Data Exploration and Understanding:** understand the meaning of all the columns in the dataset and select the fields to take into consideration
  - **Data Preparation:** filter, aggregate and clean up the dataset -> ready to use for the training phase



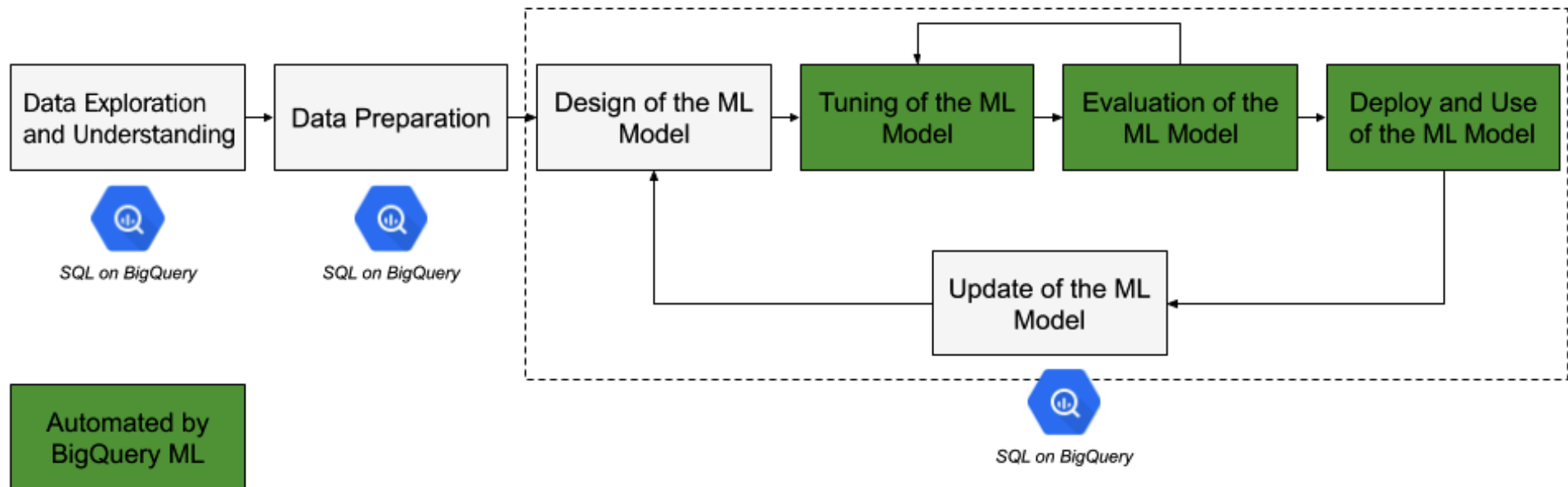
# ML model development

- Developing ML model
  - **Experimenting** with different algorithms on the training dataset
  - Parameter **tuning** to get better performance out of the ML model
  - **Evaluating** the model on the test dataset (different from training)
  - **Deploying** and eventually **updating** the ML model



# BigQuery ML

- BigQuery ML simplifies, accelerates and automates most of the activities involved in the development of a ML model



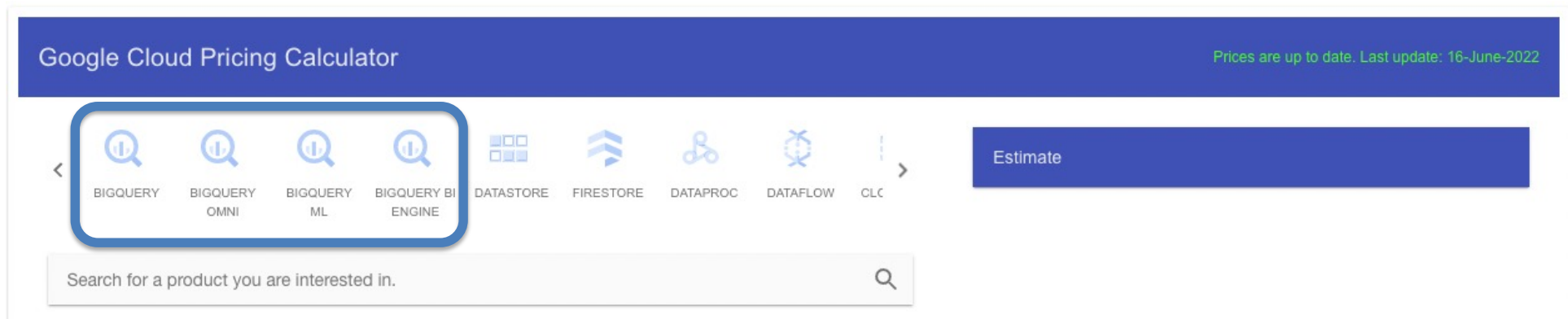


# BigQuery ML: algorithms

- **Linear regression:** forecast numerical values with a linear model
- **Binary logistic regression:** binary classification (Yes or No, 1 or 0, True or False)
- **Multiclass logistic regression:** classification with multiple options
- **Matrix factorization:** for developing recommendation systems
- **Time series:** forecast leveraging timeseries data from the past
- **Boosted tree:** classification and regression
- **AutoML table:** AutoML capabilities from the BigQuery SQL interface
- **Deep Neural Network:** classification or regression
- **K-means clustering:** data segmentation of similar objects

# BigQuery: pricing

- BigQuery operations have a cost
  - Storage
  - Compute
  - Training
  - Evaluation



Google Cloud Pricing Calculator

Prices are up to date. Last update: 16-June-2022

BIGQUERY, BIGQUERY OMNI, BIGQUERY ML, BIGQUERY BI ENGINE, DATASTORE, FIRESTORE, DATAPROC, DATAFLOW, CLC

Estimate

Search for a product you are interested in.

# BEFORE CONTINUING


## CREATE A GOOGLE CLOUD ACCOUNT

<https://console.cloud.google.com>

Redeem Google Cloud Coupon

# New Project

### Select a project

 **NEW PROJECT**

Search projects and folders

RECENT   STARRED   ALL

	Name	ID
✓ ☆ ⚙	BIDAMA ?	bidama
☆ ⚙	BlockchainParthenope ?	blockchainparthenope
☆ ⚙	Moodle ?	gifted-symbol-273720
☆ ⚙	SOD2017 ?	sod2017-166009

CANCEL   OPEN

# New Project

☰ Google Cloud

## New Project

**Project name \***  
BIDAMA ?

Project ID: bidama-356211. It cannot be changed later. [EDIT](#) ←

**Billing account \***  
BIDAMA ▼

Any charges for this project will be billed to the account you select here.

**Location \***  
 No organization [BROWSE](#)

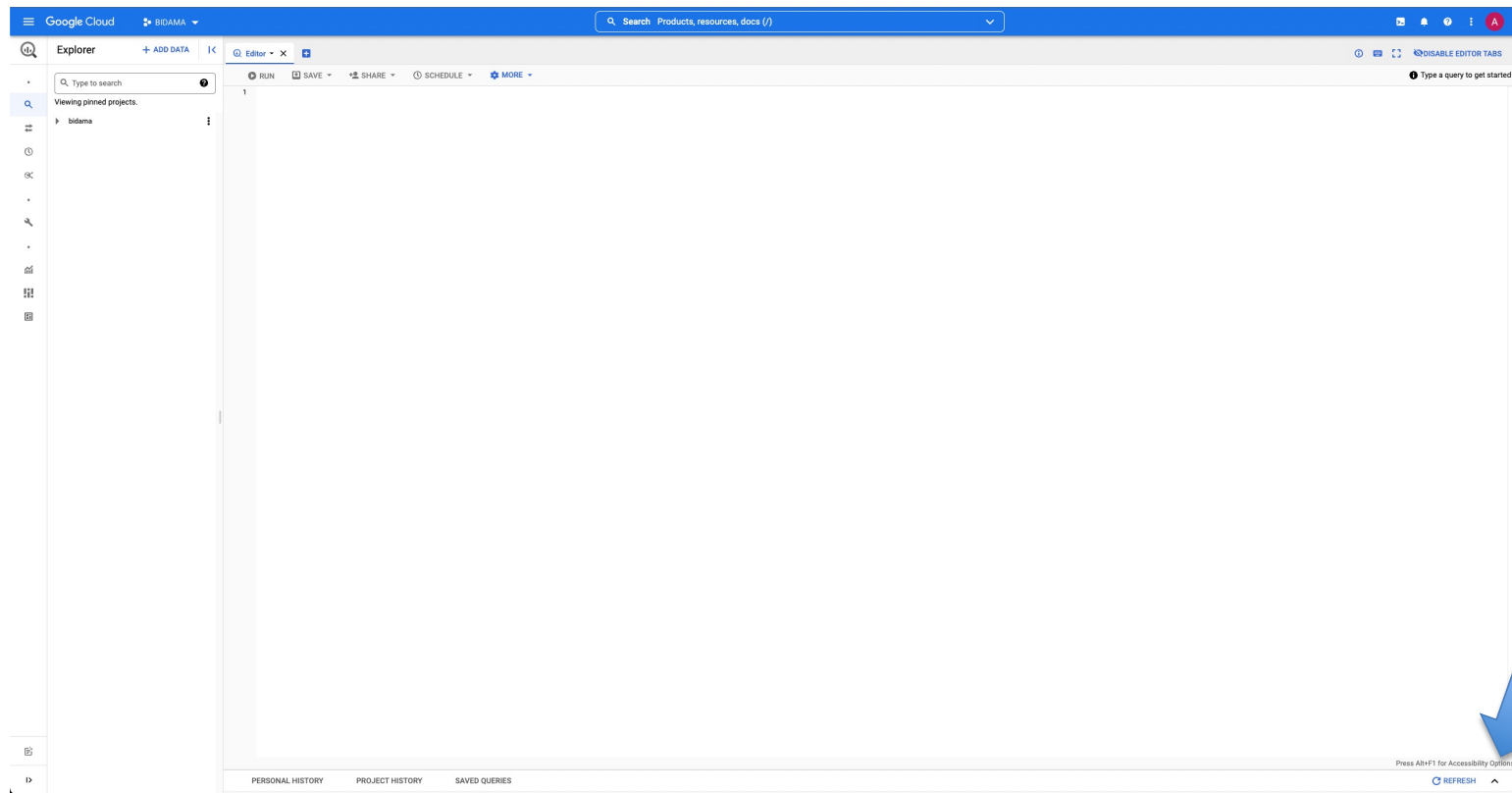
Parent organization or folder

[CREATE](#) [CANCEL](#)

# Interacting with BigQuery

The screenshot shows the Google Cloud console interface for project BIDAMA. The left sidebar is expanded to the 'ANALYTICS' section, where 'BigQuery' is highlighted. The main content area displays project information, resources (including BigQuery, SQL, Compute Engine, Storage, Cloud Functions, and App Engine), and various monitoring and news panels.

# Interacting with BigQuery



# Interacting with BigQuery

The screenshot shows the Google Cloud BigQuery interface. A red box highlights the Explorer on the left, which displays 'Viewing pinned projects' and a list containing 'bidama'. Another red box highlights the top toolbar, specifically the '+ ADD DATA' button and the 'Commands' section. A third red box encompasses the central 'Editor' area, which is currently empty. A fourth red box highlights the 'Output' section at the bottom, which shows a table with columns: Job ID, Creation time, Owner, Type, Summary, Session ID, and Actions. The table is currently empty, displaying 'No rows to display'.



# BigQuery: datasets

- (Big) Data is fundamental to exploit machine learning algorithms
- Collecting data and build large datasets is one of the most time-consuming and tedious task in the data management field
- **Cloud Public Datasets Program** allows to use data already collected and ingested into BigQuery
- The BigQuery public datasets are available in the Datasets section of the Google Cloud Marketplace

# BigQuery: datasets

The screenshot shows the Google Cloud BigQuery interface. The top navigation bar includes 'Google Cloud', 'BIDAMA', and a search bar. Below the navigation bar, the '+ ADD DATA' button is highlighted with a red box and a red arrow. A dropdown menu is open, listing options: 'Pin a project', 'Explore public datasets' (highlighted with a red box and a red arrow), 'Explore Analytics Hub', 'Informatica Data Loader', and 'External data source'. The main workspace contains a query editor and a 'PERSONAL HISTORY' table with columns: Job ID, Creation time, Owner, Type, Summary, Session ID, and Actions. The table currently shows 'No rows to display'.

# BigQuery: datasets

The screenshot shows the Google Cloud Marketplace interface. On the left, the Explorer pane shows a project named 'bidama'. The main area displays the Marketplace 'Datasets' section with 239 results. A filter sidebar on the left lists categories such as Maps (7), Big data (48), Analytics (33), Databases (7), Machine learning (4), Developer tools (24), Advertising (11), Social (5), Economics (37), and Healthcare (46). The 'Type' filter is set to 'Datasets'. The dataset grid includes:

- About COVID-19 Public Datasets**: BigQuery Public Datasets Program. Includes a sub-entry: 'Getting started with COVID-19 Public Datasets'.
- Cymbal**: About Cymbal: Google Cloud's demo brand. Cymbal Group. Includes a sub-entry: 'Synthetic datasets across industries showcasing Google Cloud'.
- AFSC Open Data Portal**: NOAA. Fisheries research data for the Alaska region.
- Aion On-Chain Transaction Data**: cmorq. Easy access to on-chain transaction data.
- Algorand On-Chain**
- American Community Survey**

# BigQuery: datasets

The screenshot shows the Google Cloud Marketplace interface. On the left, the Explorer pane shows a project named 'bidama'. The main Marketplace pane has a search bar with 'taxi' entered. Below the search bar, the breadcrumb path is 'Marketplace > "taxi" > Datasets'. A filter box is present with the text 'Filter Type to filter'. Under the 'Type' section, 'Datasets' is selected, and it shows '2 results'. The first result is 'Chicago Taxi Trips' by the City of Chicago, with a description: 'This dataset includes taxi trips from 2013 to the present, reported to the City of Chicago in its role as : allow for aggregate analyses, the Taxi ID is consistent for any given taxi medallion number but does n suppressed in some cases, and times are rounded to the nearest 15 minutes. Due to the data reportin'. The second result is 'NYC TLC Trips' by the City of New York, with a description: 'This dataset is collected by the NYC Taxi and Limousine Commission (TLC) and includes trip records Green taxis in NYC from 2009 to present, and all trips in for-hire vehicles (FHV) from 2015 to present. and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, pay'.

# BigQuery: datasets

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer pane shows a project named 'bidama'. The main area displays the details for the 'Chicago Taxi Trips' dataset, which is provided by the City of Chicago. A red arrow points to the 'VIEW DATASET' button. Below the button, there are tabs for 'OVERVIEW' and 'SAMPLES'. The 'OVERVIEW' tab is active, showing a description of the dataset and its details.

**Chicago Taxi Trips**  
City of Chicago  
Chicago taxi trips from 2013 to present

[VIEW DATASET](#)

[OVERVIEW](#) [SAMPLES](#)

**Overview**

This dataset includes taxi trips from 2013 to the present, reported to the City of Chicago in its role as a regulatory agency. To protect privacy but allow for aggregate analyses, the Taxi ID is consistent for any given taxi medallion number but does not show the number, Census Tracts are suppressed in some cases, and times are rounded to the nearest 15 minutes. Due to the data reporting process, not all trips are reported but the City believes that most are. For more information about this dataset and how it was created, see [this post](#) on the City of Chicago's blog.

This public dataset is hosted in Google BigQuery and is included in BigQuery's 1TB/mo of free tier processing. This means that each user receives 1TB of free BigQuery processing every month, which can be used to run queries on this public dataset. Watch this short video to learn how to get started quickly using BigQuery to access public datasets. [What is BigQuery](#)

**Additional details**

Type: [Datasets](#)  
Category: [Encyclopedic](#)  
Dataset source: [Chicago Data Portal](#)  
Cloud service: BigQuery  
Expected update frequency: Monthly

# BigQuery: datasets

The screenshot shows the Google Cloud BigQuery interface. The top navigation bar includes 'Google Cloud', 'BIDAMA', and a search bar. The main content area is divided into an Explorer on the left and a Dataset info panel on the right. The Explorer shows a project named 'bidama'. The Dataset info panel displays the following details for the 'chicago\_taxi\_trips' dataset:

Dataset info	
Dataset ID	bigquery-public-data.chicago_taxi_trips
Created	11.04.2017, 3:04:24 PM UTC+2
Default table expiration	Never
Last modified	27.04.2017, 11:56:26 PM UTC+2
Data location	US
Description	
Default collation	[null]

At the bottom of the interface, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES', along with a 'REFRESH' button.

# BigQuery: datasets

The screenshot shows the Google Cloud BigQuery Explorer interface. The search bar at the top left contains 'Chicago Taxi Trips'. Below the search bar, the Explorer pane shows a search result for 'taxi\_trips' under the 'bigquery-public-data' project. The 'taxi\_trips' dataset is selected, and the 'SCHEMA' tab is active. The schema table is displayed below, listing various fields such as 'unique\_key', 'taxi\_id', 'trip\_start\_timestamp', etc.

Field name	Type	Mode	Collation	Policy Tags	Description
unique_key	STRING	REQUIRED			Unique identifier for the trip.
taxi_id	STRING	REQUIRED			A unique identifier for the taxi.
trip_start_timestamp	TIMESTAMP	NULLABLE			When the trip started, rounded to the nearest 15 minutes.
trip_end_timestamp	TIMESTAMP	NULLABLE			When the trip ended, rounded to the nearest 15 minutes.
trip_seconds	INTEGER	NULLABLE			Time of the trip in seconds.
trip_miles	FLOAT	NULLABLE			Distance of the trip in miles.
pickup_census_tract	INTEGER	NULLABLE			The Census Tract where the trip began. For privacy, this Census Tract is not shown for some trips.
dropoff_census_tract	INTEGER	NULLABLE			The Census Tract where the trip ended. For privacy, this Census Tract is not shown for some trips.
pickup_community_area	INTEGER	NULLABLE			The Community Area where the trip began.
dropoff_community_area	INTEGER	NULLABLE			The Community Area where the trip ended.
fare	FLOAT	NULLABLE			The fare for the trip.
tips	FLOAT	NULLABLE			The tip for the trip. Cash tips generally will not be recorded.
tolls	FLOAT	NULLABLE			The tolls for the trip.
extras	FLOAT	NULLABLE			Extra charges for the trip.
trip_total	FLOAT	NULLABLE			Total cost of the trip, the total of the fare, tips, tolls, and extras.
payment_type	STRING	NULLABLE			Type of payment for the trip.

# BigQuery: datasets

The screenshot shows the Google BigQuery interface. On the left, the Explorer pane shows a search for 'Chicago Taxi Trips' with one result: 'taxi\_trips' under the 'bigquery-public-data' > 'chicago\_taxi\_trips' hierarchy. The main pane shows the 'taxi\_trips' table selected, with tabs for 'SCHEMA', 'DETAILS', and 'PREVIEW'. The 'DETAILS' tab is active and highlighted with a red box. Below the tabs is the 'Table info' section, which includes the following details:

Table info	
Table ID	bigquery-public-data.chicago_taxi_trips.taxi_trips
Table size	72.88 GB
Long-term storage size	0 B
Number of rows	201,102,217
Created	12.04.2017, 9:35:35 PM UTC+2
Last modified	09.06.2022, 3:31:30 PM UTC+2
Table expiration	NEVER
Data location	US
Default collation	[null]
Description	

At the bottom of the interface, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES', along with a 'REFRESH' button.



# BigQuery: datasets

The screenshot shows the Google Cloud BigQuery Explorer interface. On the left, the 'Explorer' pane shows the project 'bigquery-public-data' and the dataset 'taxi\_trips'. The main area displays the 'taxi\_trips' dataset with a 'PREVIEW' button highlighted in a red box. Below the 'PREVIEW' button, a table of data is shown with columns 'unique\_key' and 'taxi\_id'. The table contains 23 rows of data.

Row	unique_key	taxi_id
1	6ac248f401e05799558f7a015bf03c5354d622	8ac089148362cd529ebbb3d11fba4ca368d94478c0e83f055f4e6a53bf0fba4ad1935a89916d93fb79acfdce012af7212702c6651cddf3d1d
2	85dfe100e44ed32347972e7ecc0f5fb9f086146	8ac089148362cd529ebbb3d11fba4ca368d94478c0e83f055f4e6a53bf0fba4ad1935a89916d93fb79acfdce012af7212702c6651cddf3d1d
3	cc7852078d7422df7764a9276fd239209976d0	8ac089148362cd529ebbb3d11fba4ca368d94478c0e83f055f4e6a53bf0fba4ad1935a89916d93fb79acfdce012af7212702c6651cddf3d1d
4	9bfaddca9732699f6de5617a3cb0091ba580135	8ac089148362cd529ebbb3d11fba4ca368d94478c0e83f055f4e6a53bf0fba4ad1935a89916d93fb79acfdce012af7212702c6651cddf3d1d
5	3392da9b0dabb69f59832ec82944c6496bbf117	5fa0a0f93d7c8f729c7f3132eed2de1dcf04fd38e73ceec586e20e57f09e782b9918db5c33987206b2df3263cc9d364dc7f3ec1c2ad8200d63c04
6	ecfbeb98b66d09b8818fe32f118715f561f6c84d	5fa0a0f93d7c8f729c7f3132eed2de1dcf04fd38e73ceec586e20e57f09e782b9918db5c33987206b2df3263cc9d364dc7f3ec1c2ad8200d63c04
7	11511f72eb07c30fdcc2d795e4b3593b9240543	616cb0ddec6abf371a7e856398d2913659aad0c9b3fcc066aeea8b9b8e7eefcf02b94ad0452c13c024d7248d6b4b6eb9e9daf88fb2777ef80f4
8	d4185579f29e17bd0e9d36192d4b8fd932d07dff	616cb0ddec6abf371a7e856398d2913659aad0c9b3fcc066aeea8b9b8e7eefcf02b94ad0452c13c024d7248d6b4b6eb9e9daf88fb2777ef80f4
9	9d08be2c413410f71ee03345c247814636e58607	616cb0ddec6abf371a7e856398d2913659aad0c9b3fcc066aeea8b9b8e7eefcf02b94ad0452c13c024d7248d6b4b6eb9e9daf88fb2777ef80f4
10	2c9712c0f56cfa82cde4878fbcc09fa56a8e	616cb0ddec6abf371a7e856398d2913659aad0c9b3fcc066aeea8b9b8e7eefcf02b94ad0452c13c024d7248d6b4b6eb9e9daf88fb2777ef80f4
11	01ce741026ba46fe34d9865cf322caf116fdee2	7800175596c88be3bc368d279f7e6999f35424ba54de242478ec320f9a18c38127b975d9fd174f6095c88835b9dbfddc1dd4bec3a9baadc6aa34
12	716174f9d6e69dca97f8f8d145061e4ccc4725c	3fc5b45d2e3b125ffcc0c767afcc8934579505e28a42b56472bdf50060258ea70a031372fd0efef98eb2166f2e0f70494ed9f0c23472fb76501c36f
13	74ff1000fd92bd233f3d4faf3bfb6ebcdbeeb6fe	3fc5b45d2e3b125ffcc0c767afcc8934579505e28a42b56472bdf50060258ea70a031372fd0efef98eb2166f2e0f70494ed9f0c23472fb76501c36f
14	55a0cf1220134950f359138a67255edebbaa723e	3fc5b45d2e3b125ffcc0c767afcc8934579505e28a42b56472bdf50060258ea70a031372fd0efef98eb2166f2e0f70494ed9f0c23472fb76501c36f
15	c96f1691bd01ced6f2871d15b5f2443e55c0a400	3fc5b45d2e3b125ffcc0c767afcc8934579505e28a42b56472bdf50060258ea70a031372fd0efef98eb2166f2e0f70494ed9f0c23472fb76501c36f
16	cabae109bcbf9649e2bd99e9e9019e9bfa50e309	3fc5b45d2e3b125ffcc0c767afcc8934579505e28a42b56472bdf50060258ea70a031372fd0efef98eb2166f2e0f70494ed9f0c23472fb76501c36f
17	ce6db78db322a722f5a41b7f8593085f1b734f2	3fc5b45d2e3b125ffcc0c767afcc8934579505e28a42b56472bdf50060258ea70a031372fd0efef98eb2166f2e0f70494ed9f0c23472fb76501c36f
18	a806f84e4d31c017b851637161a67169a3aeb71	bd29edce02bda40440d48e6f090bd8a20e2e6557cfe3b8f2d9cf84d69beea2792e553693e826a0cb46a0a09aada5b10f3b25e51ea4796eebf
19	e99a41308cca0da5467e2730eea6d70f87fea3d	bd29edce02bda40440d48e6f090bd8a20e2e6557cfe3b8f2d9cf84d69beea2792e553693e826a0cb46a0a09aada5b10f3b25e51ea4796eebf
20	5124f87400f1bcdead5281bcd0dd9be78d2f2cc0	bd29edce02bda40440d48e6f090bd8a20e2e6557cfe3b8f2d9cf84d69beea2792e553693e826a0cb46a0a09aada5b10f3b25e51ea4796eebf
21	4e1b2c02de504322c7de790dd89a3dcbdcfc8c6	bd29edce02bda40440d48e6f090bd8a20e2e6557cfe3b8f2d9cf84d69beea2792e553693e826a0cb46a0a09aada5b10f3b25e51ea4796eebf
22	8b73d025f4dc1b4596f2f83609a9927042670b	a3cc3825247ea5ea76671d4b88eb0005442bd7c8b5cbe570a8ad4cc087f1bef5f347173bcd2667392e7e499bc159ddd707f13412df92a8aea9a
23	ee683a0ac18ba369872255dc0eebbf56284853	a3cc3825247ea5ea76671d4b88eb0005442bd7c8b5cbe570a8ad4cc087f1bef5f347173bcd2667392e7e499bc159ddd707f13412df92a8aea9a

# Agenda

- Google Cloud Platform
- Google BigQuery
- **Google BigQuery SQL**
- Google BigQuery ML

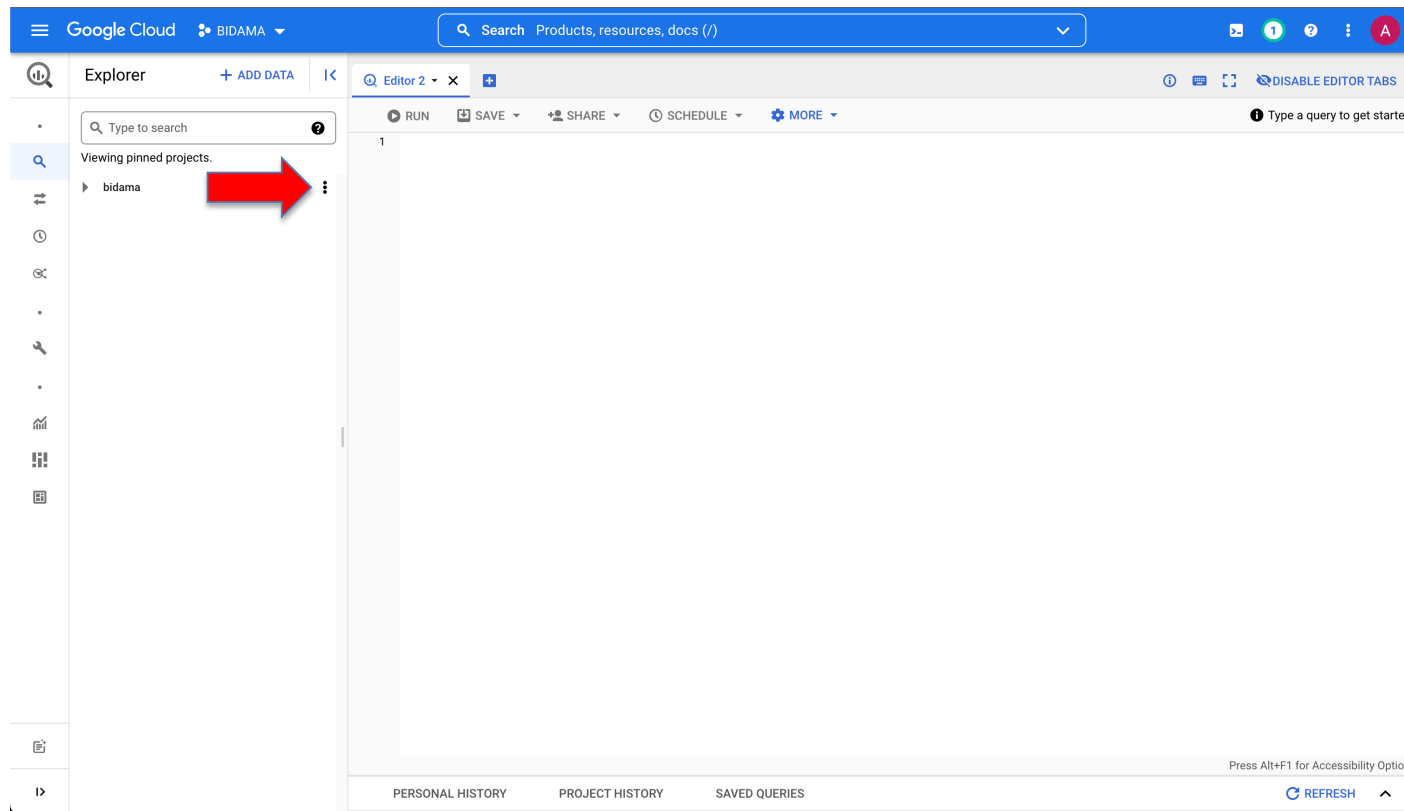
# BigQuery: syntax

- **BigQuery** is based on **SQL** with extensions which allow to use Machine Learning features
- In particular the process will go through
  - BigQuery dataset
  - BigQuery SQL
  - BigQuery ML

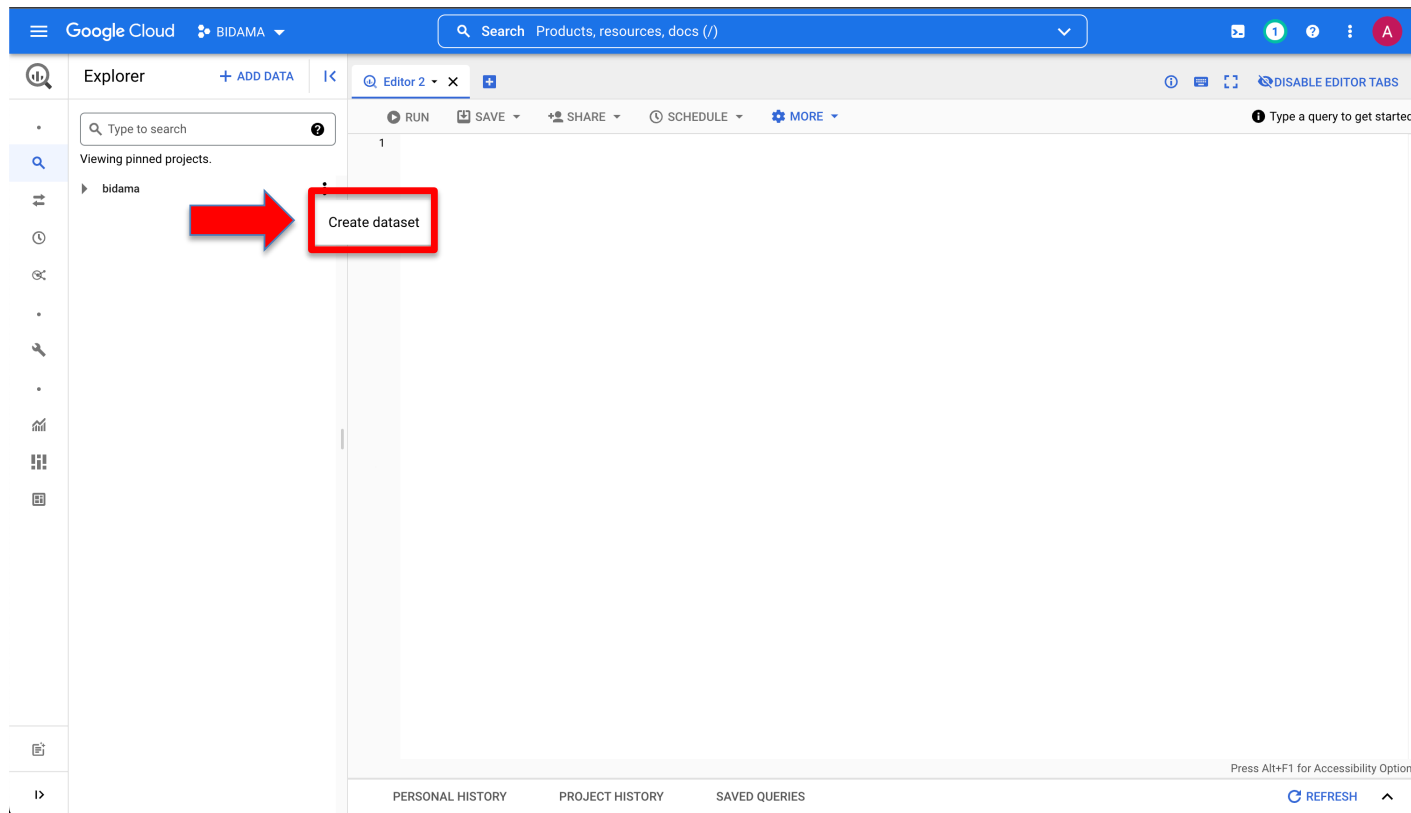
# Creating BigQuery dataset

- First of all it is necessary to create a BigQuery dataset
- Select **BigQuery service** in GCP
- **Create Dataset**

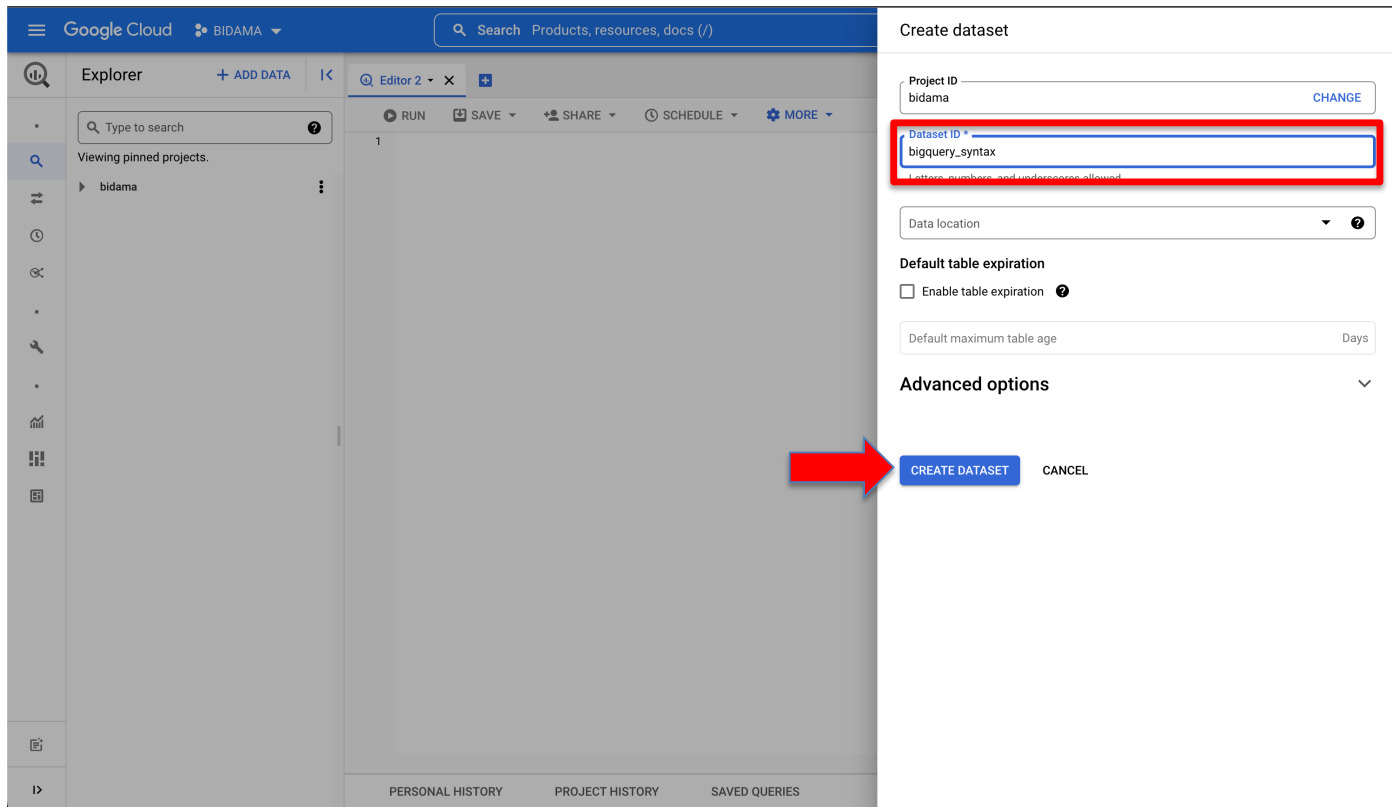
# Creating BigQuery dataset



# Creating BigQuery dataset



# Creating BigQuery dataset

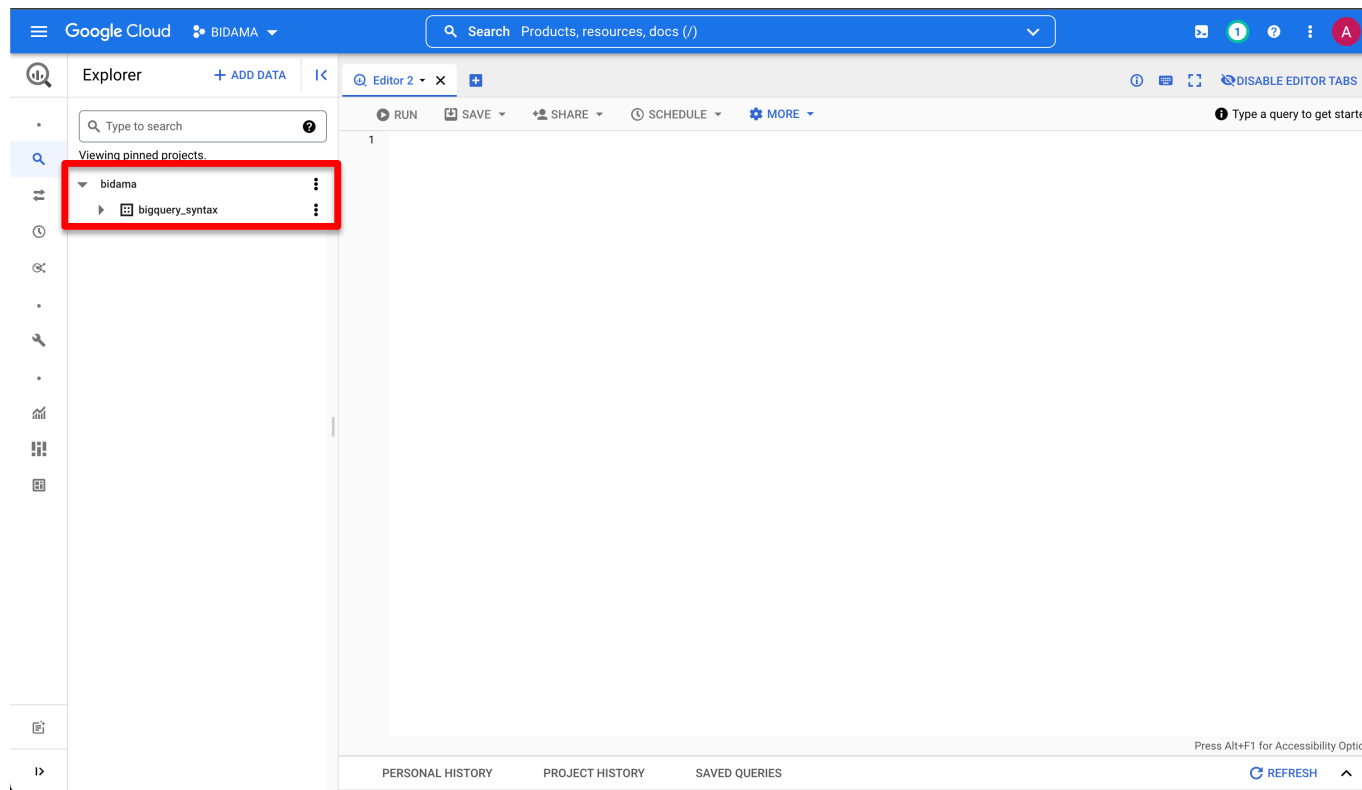


The screenshot displays the Google Cloud console interface for creating a BigQuery dataset. The 'Create dataset' dialog is open, showing the following fields and options:

- Project ID:** bidama (with a CHANGE link)
- Dataset ID:** bigquery\_syntax (highlighted with a red box and a red arrow pointing to the 'CREATE DATASET' button)
- Data location:** (dropdown menu)
- Default table expiration:**
  - Enable table expiration
  - Default maximum table age: (input field) Days
- Advanced options:** (dropdown menu)

At the bottom of the dialog, there are two buttons: **CREATE DATASET** (highlighted with a red arrow) and **CANCEL**.

# Creating BigQuery dataset





# BigQuery SQL

The screenshot shows the Google Cloud BigQuery SQL Editor interface. The top navigation bar includes 'Google Cloud', 'BIDAMA', and a search bar. The left sidebar shows the 'Explorer' view with a search bar and a tree view containing 'bidama' and 'bigquery\_syntax'. The main editor area is titled 'Editor 2' and contains a 'MORE' menu. The 'MORE' menu is open, displaying the following options:

- Format query
- Query settings
- Enable SQL translation  
Offered by BigQuery Migration Service
- Translation settings

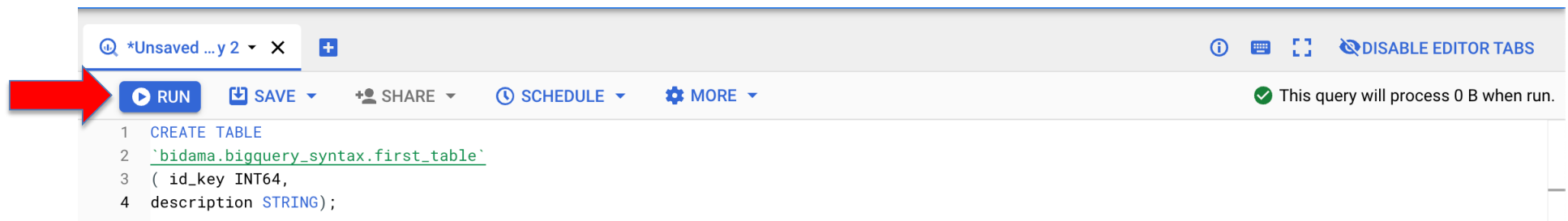
A red arrow points to the 'Query settings' option. At the bottom of the editor, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES', along with a 'REFRESH' button and a 'Type a query to get started' prompt.

# BigQuery SQL: CRUD

- **CRUD** stands for
  - Create
  - Read
  - Update
  - Delete
- That are the same basic operations for handling BigQuery objects

# BigQuery SQL: CREATE

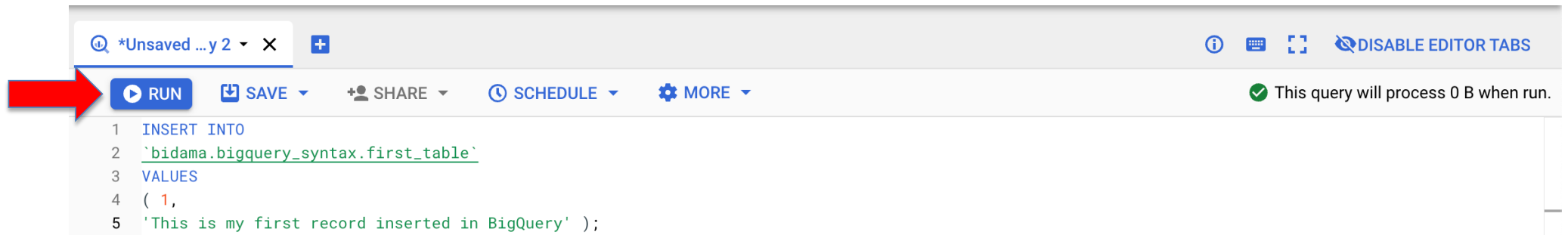
- The **CREATE** statement is used to create objects or to insert new items into an existing table
- The first two words of the query statement, **CREATE TABLE**, are used to start the creation of a new table
- Next, the id of the object is composed by the **project name**, the **dataset name** and the **table name** concatenated with the . symbol and enclosed by the backtick character `
- Last, the list of fields with their data type separated by the comma character



```
1 CREATE TABLE
2 `bidama.bigquery_syntax.first_table`
3 ( id_key INT64,
4  description STRING);
```

# BigQuery SQL: INSERT

- The **INSERT** statement is used to insert values into the table



```
1 INSERT INTO
2 `bidama.bigquery_syntax.first_table`
3 VALUES
4 ( 1,
5 'This is my first record inserted in BigQuery' );
```

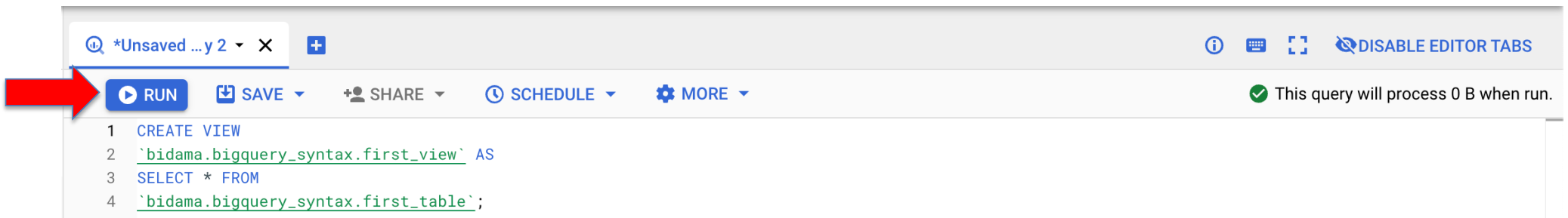
ⓘ ⓘ ⓘ ⓘ DISABLE EDITOR TABS

▶ RUN ⌵ SAVE ⌵ 👤 SHARE ⌵ ⌚ SCHEDULE ⌵ ⚙ MORE ⌵

✔ This query will process 0 B when run.

# BigQuery SQL: VIEW

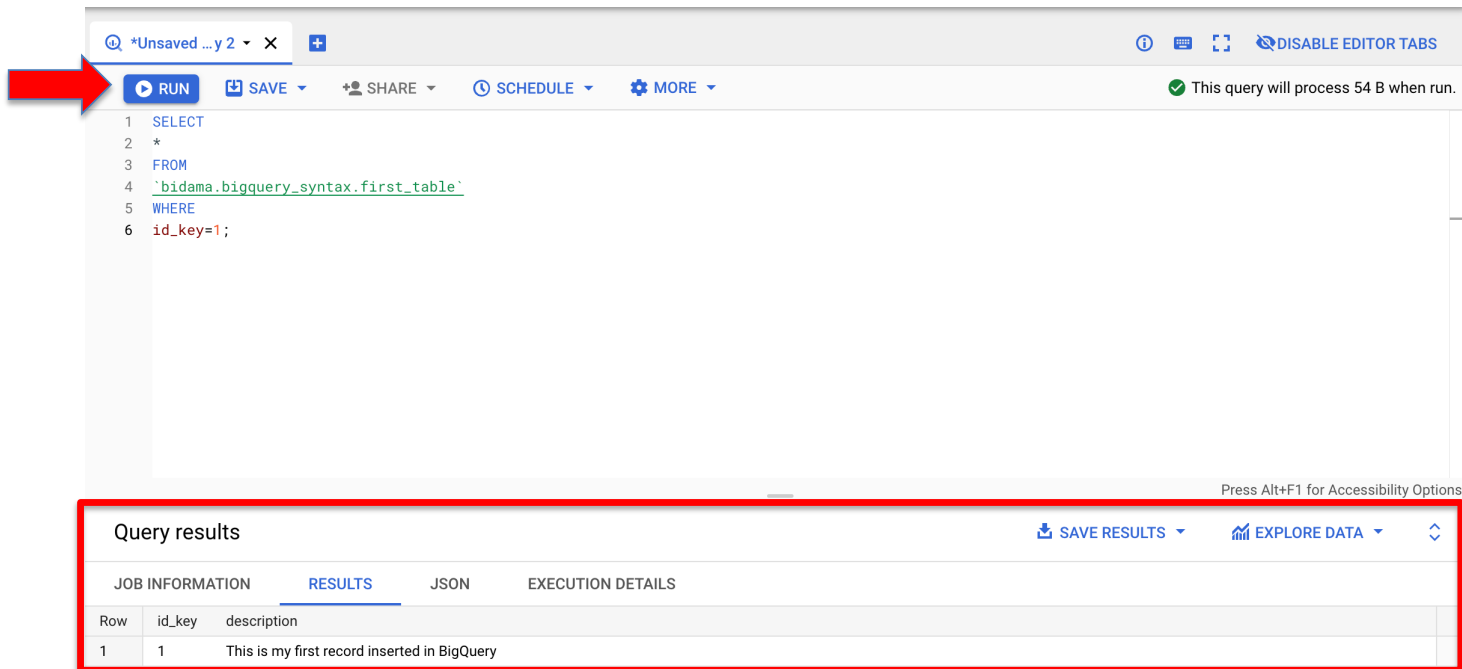
- The **CREATE VIEW** statement is used to create a **view**, i.e. to access records of an underlying table



```
1 CREATE VIEW
2 `bidama.bigquery_syntax.first_view` AS
3 SELECT * FROM
4 `bidama.bigquery_syntax.first_table`;
```

# BigQuery SQL: READ

- **Read** operations are mainly based on **SELECT** statements and can be applied to different database objects such as tables and views



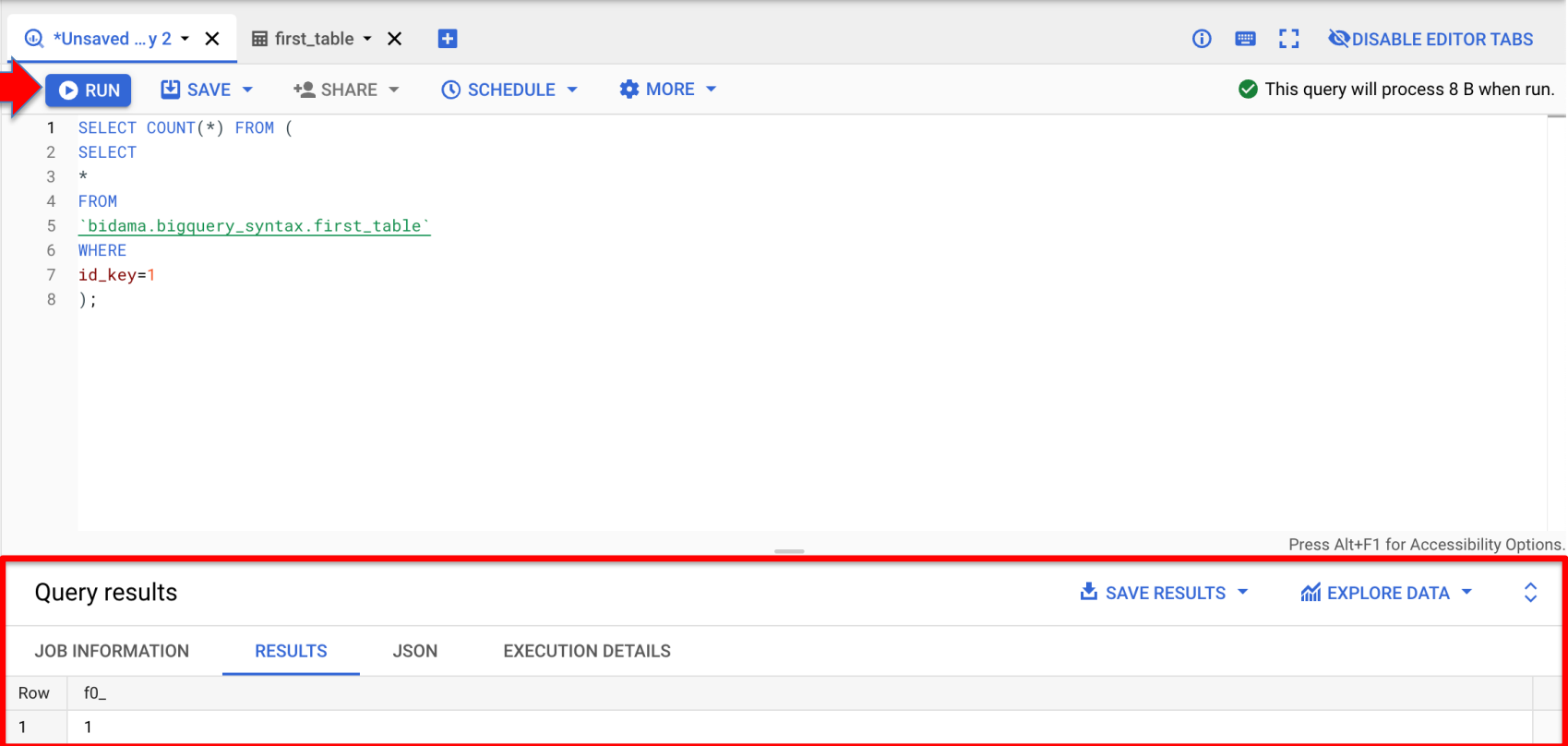
The screenshot displays the BigQuery SQL editor interface. A red arrow points to the **RUN** button. The query editor contains the following SQL code:

```
1 SELECT
2 *
3 FROM
4 `bidama.bigquery_syntax.first_table`
5 WHERE
6 id_key=1;
```

Below the query editor, the **Query results** section is highlighted with a red box. It shows the following table:

Row	id_key	description
1	1	This is my first record inserted in BigQuery

# BigQuery SQL: READ



The screenshot shows the BigQuery console interface. At the top, there are tabs for the query editor and a table named 'first\_table'. The query editor contains the following SQL code:

```
1 SELECT COUNT(*) FROM (  
2 SELECT  
3 *  
4 FROM  
5 `bidama.bigquery_syntax.first_table`  
6 WHERE  
7 id_key=1  
8 );
```

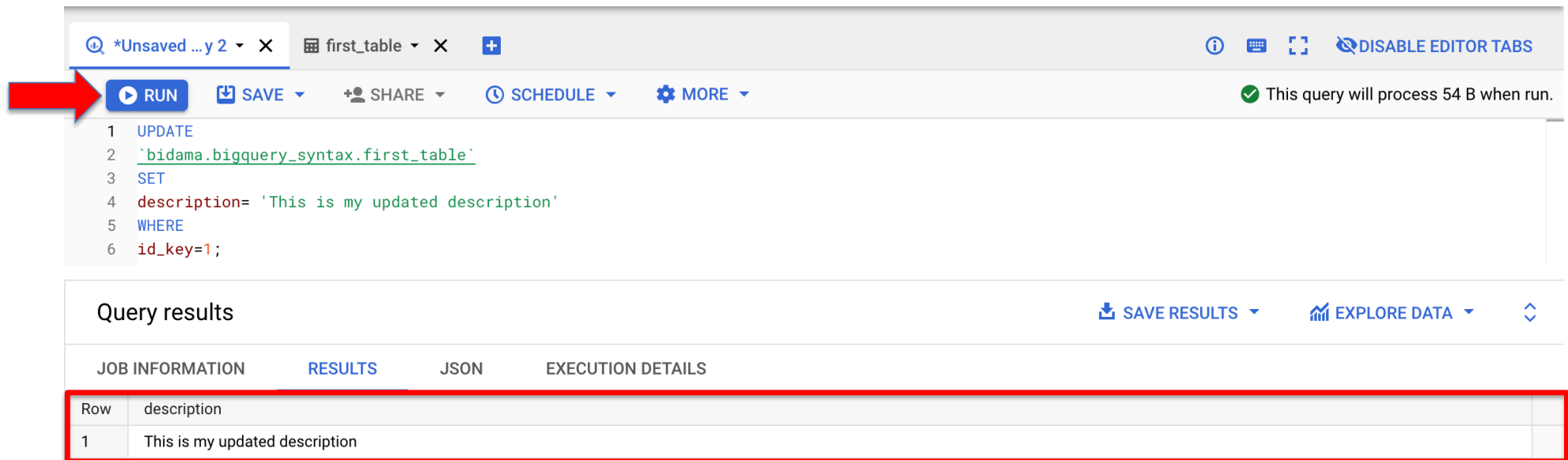
A red arrow points to the 'RUN' button in the toolbar. To the right of the toolbar, a status message indicates: "This query will process 8 B when run." Below the query editor, the 'Query results' section is highlighted with a red border. It shows the following table:

Row	f0_
1	1

At the bottom of the 'Query results' section, there are tabs for 'JOB INFORMATION', 'RESULTS', 'JSON', and 'EXECUTION DETAILS'. The 'RESULTS' tab is currently selected. To the right of the results, there are buttons for 'SAVE RESULTS' and 'EXPLORE DATA'.

# BigQuery SQL: UPDATE

- Update operations such as **UPDATE** and **MERGE** are supported and can be used to change the value of a record or a set of records



The screenshot displays the BigQuery SQL editor interface. At the top, there are tabs for the query, including '\*Unsaved ...y 2' and 'first\_table'. A red arrow points to the 'RUN' button. The SQL query is as follows:

```
1 UPDATE
2 `bidama.bigquery_syntax.first_table`
3 SET
4 description= 'This is my updated description'
5 WHERE
6 id_key=1;
```

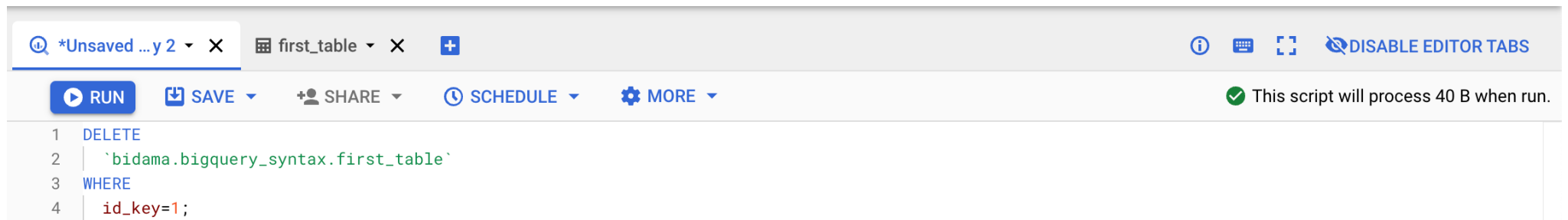
Below the query, the 'Query results' section is visible, showing a table with one row. The table has two columns: 'Row' and 'description'. The first row contains the value '1' in the 'Row' column and 'This is my updated description' in the 'description' column. The 'RESULTS' tab is selected, and the table is highlighted with a red border.

Row	description
1	This is my updated description



# BigQuery SQL: DELETE

- DELETE a record

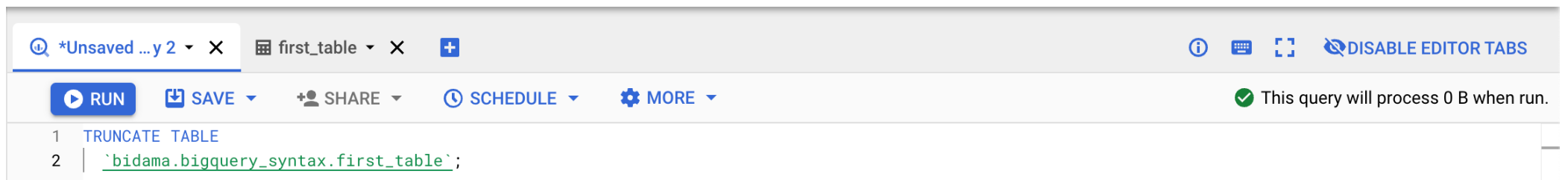


The screenshot shows a BigQuery SQL editor interface. At the top, there are tabs for '\*Unsaved ...y 2' and 'first\_table'. Below the tabs is a toolbar with buttons for 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. A status bar on the right indicates 'This script will process 40 B when run.' The main editor area contains the following SQL code:

```
1 DELETE
2   `bidama.bigquery_syntax.first_table`
3 WHERE
4   id_key=1;
```

# BigQuery SQL: DELETE

- **DELETE** all record from a table

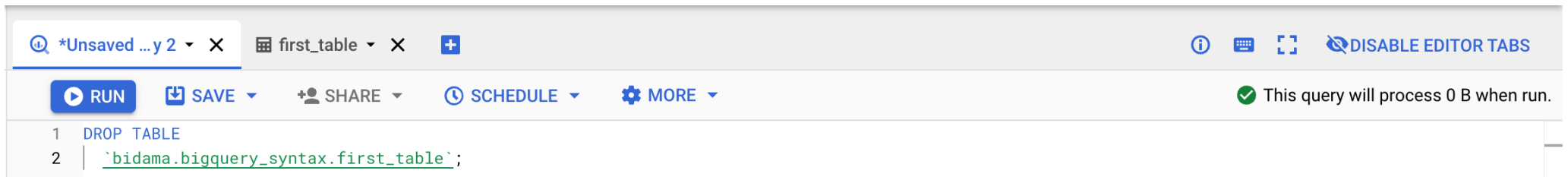


The screenshot shows the BigQuery SQL editor interface. The top bar includes a search icon, a tab labeled '\*Unsaved ...y 2', and another tab labeled 'first\_table'. On the right side of the top bar, there are icons for help, keyboard shortcuts, and a 'DISABLE EDITOR TABS' button. Below the top bar, there is a row of action buttons: 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. To the right of these buttons, a green checkmark icon is followed by the text 'This query will process 0 B when run.' The main editor area contains two lines of SQL code: '1 TRUNCATE TABLE' and '2 `bidama.bigquery\_syntax.first\_table`;'.

```
1 TRUNCATE TABLE
2 `bidama.bigquery_syntax.first_table`;
```

# BigQuery SQL: DELETE

- DELETE a table

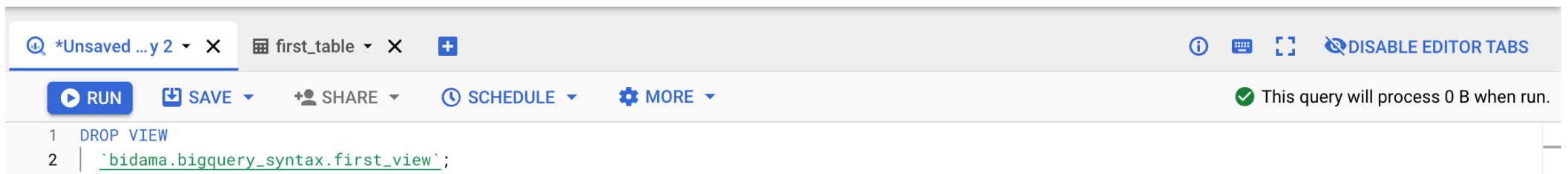


The screenshot shows the BigQuery SQL editor interface. At the top, there are tabs for '\*Unsaved ...y 2' and 'first\_table'. Below the tabs is a toolbar with buttons for 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. To the right of the toolbar, there is a status message: 'This query will process 0 B when run.' The main editor area contains the following SQL code:

```
1 DROP TABLE
2 | `bidama.bigquery_syntax.first_table`;
```

# BigQuery SQL: DELETE

- **DELETE** a view (only metadata are deleted)



The screenshot shows the BigQuery SQL editor interface. At the top, there are tabs for '\*Unsaved ...y 2' and 'first\_table'. Below the tabs, there are action buttons: 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. On the right side, there is a status message: 'This query will process 0 B when run.' The main editor area contains the following SQL code:

```
1 DROP VIEW
2 | `bidama.bigquery_syntax.first_view`;
```