

**Module:** Big Data and NoSql DataBases

**Major :** Computer Engineering

**The guide to implementing the project using Google BigQuery cloud service**

**Topic:**

**« Rhythmic Revelations: Analyzing Spotify's Songs Patterns »**

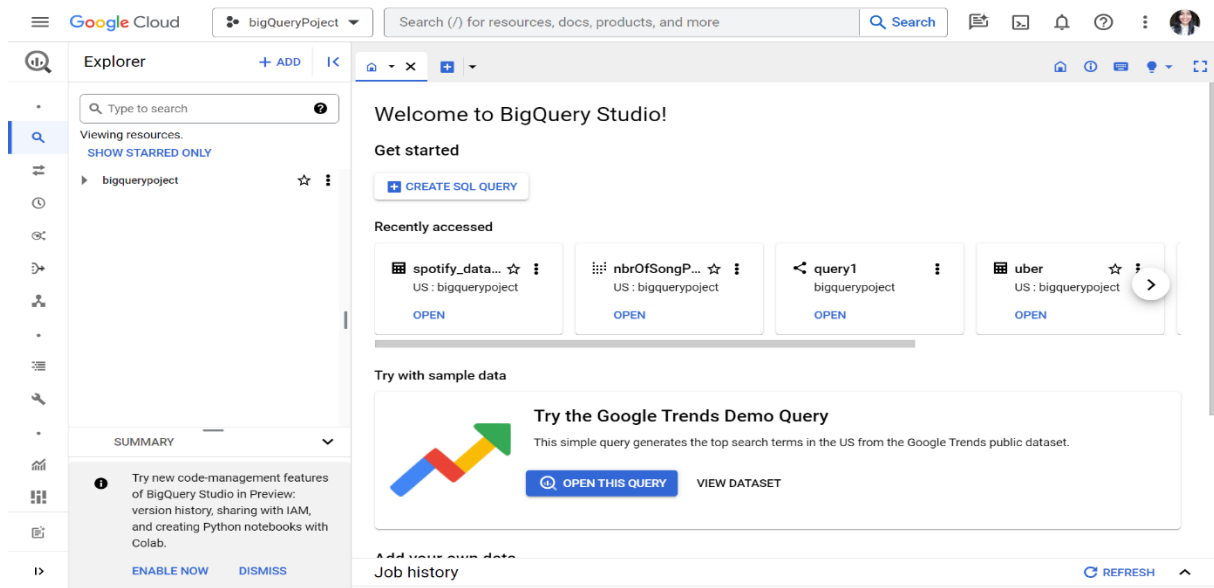
➤ **Realized by:**

AIT ABDIELMOMIN Hajar

➤ **Supervised by:**

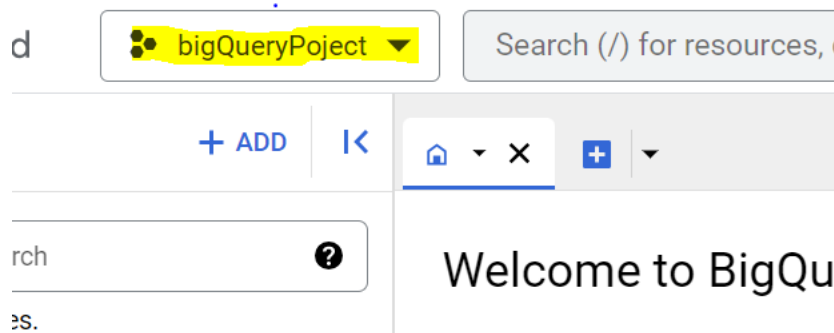
Mrs. KARIM Lamia

# I. Overview of the Google BigQuery web interface

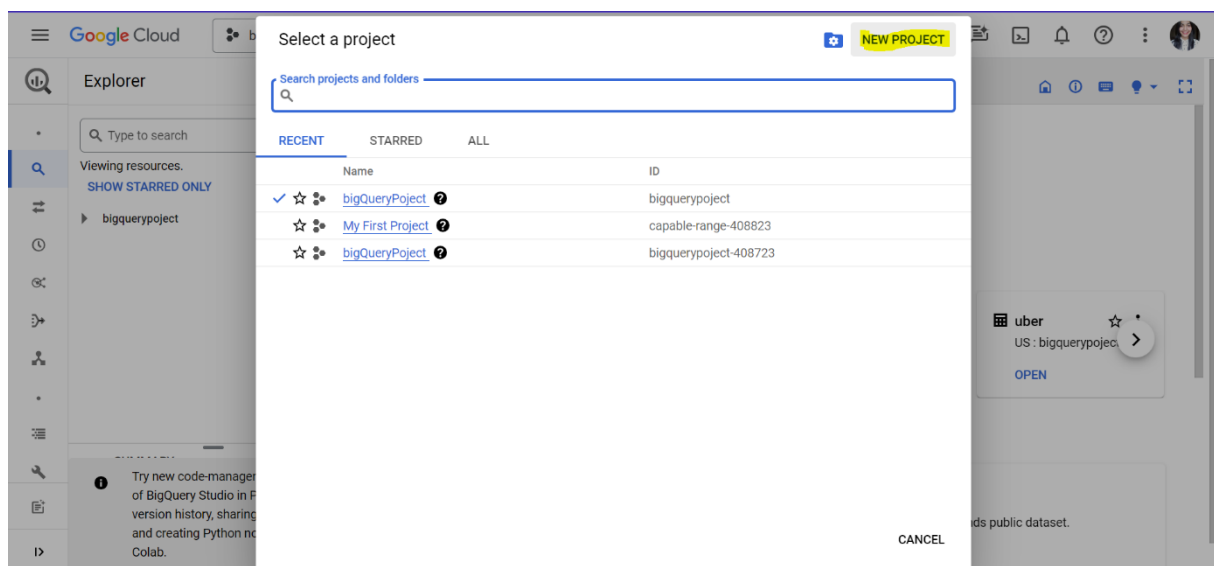


## 1- Creating a BigQuery project

### 1.1 Click in here:



The window below will appear, it contains all the bigQuery that you have created, then click on **'NEW PROJECT'**



Now give your project a name: (you can leave the **organization** field empty):

The screenshot shows the 'New Project' form in the Google Cloud console. At the top, there's a Google Cloud logo and a search bar. Below that, a message states: 'You have 13 projects remaining in your quota. Request an increase or delete projects. [Learn more](#) [MANAGE QUOTAS](#)'. The form has two main input fields: 'Project name \*' with the value 'bigDataProject' and 'Location \*' with the value 'No organization'. Below these, it says 'Project ID: bigdataproject-409418. It cannot be changed later. [EDIT](#)'. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

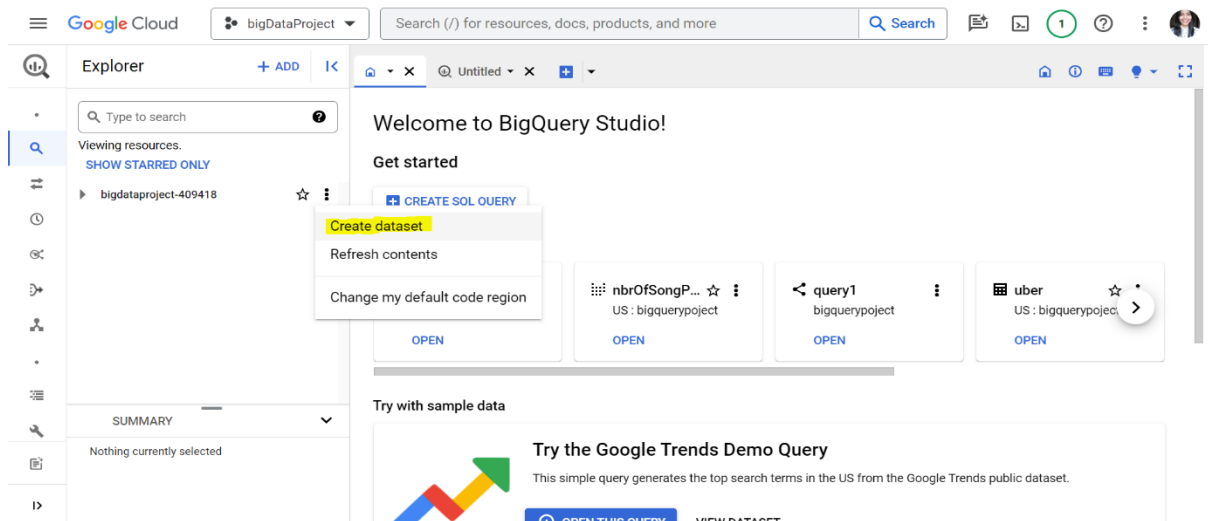
As you can see the project is created successfully:

The screenshot shows the Google Cloud console for the project 'bigDataProject'. The main content area is titled 'Quotas for project "bigDataProject"' and includes a 'Set up quota alerts' section. A notifications overlay is visible on the right, listing several actions: 'Create Project: bigDataProject' (Just now), 'Update firewall rule "mage-access"', 'Create firewall rule "mage-access"', 'Create VM instance "uber-data-engineering" and its boot disk "uber-data-engineering"', 'Enable service: compute.googleapis.com', and 'Create Project: bigQueryProject'. A small error message at the bottom states: 'Now viewing project "bigDataProject" in organization "No organization"'. The top navigation bar shows the Google Cloud logo, the project name 'bigDataProject', and a search bar.

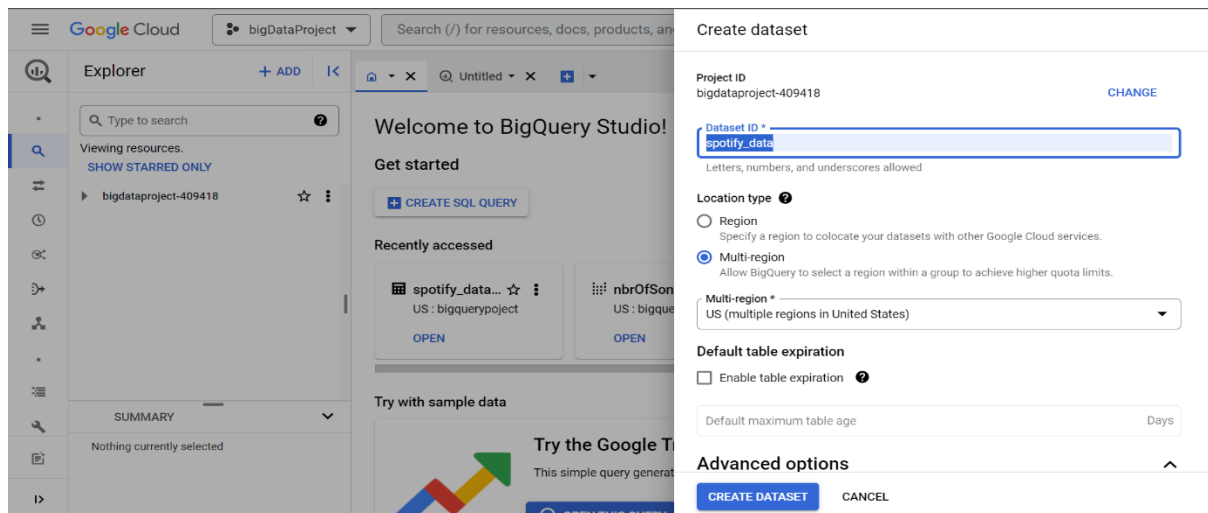
Now click on **'SELECT PROJECT'**:

The screenshot shows the Google Cloud console dashboard for the project 'bigDataProject'. The dashboard is divided into several sections: 'Project info' (Project name: bigDataProject, Project number: 607026668650, Project ID: bigdataproject-409418), 'Resources' (BigQuery, SQL), 'APIs' (Requests (requests/sec)), 'Google Cloud Platform status' (All services normal), and 'Monitoring' (Create my dashboard, Set up alerting policies, Create uptime checks). A small error message at the bottom states: 'Now viewing project "bigDataProject" in organization "No organization"'. The top navigation bar shows the Google Cloud logo, the project name 'bigDataProject', and a search bar.

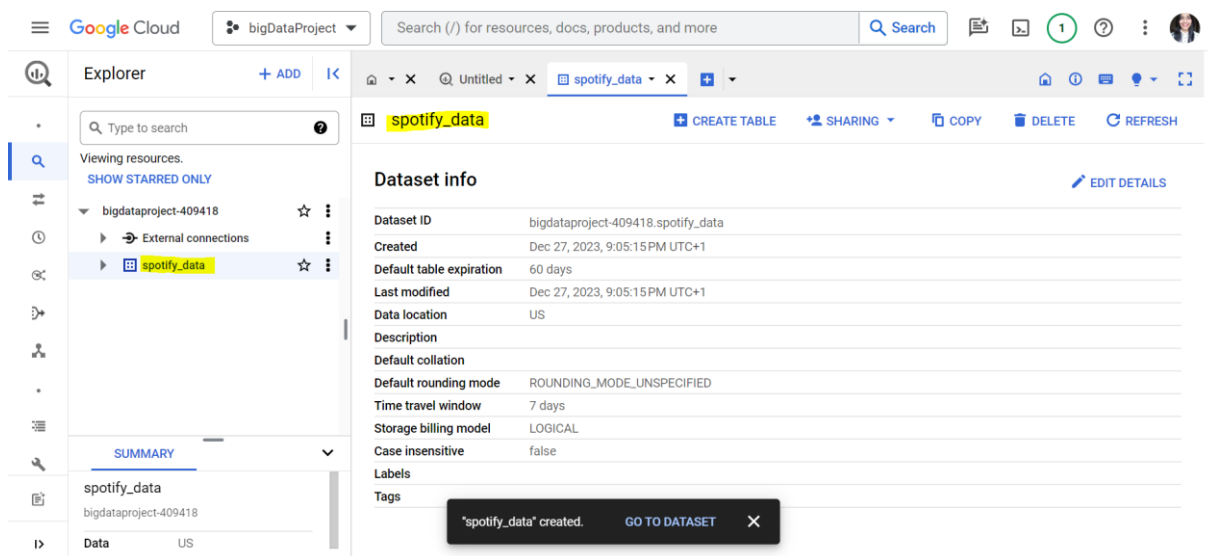
## 1.2 Creating a DataSet, we will name it 'Spotify\_data'



Give a name to the DataSet and you can leave the other parameters with their default values:



And now our DataSet is successfully:



### 1.3 Creating a Table inside our DataSet to store our data

Note that there several ways to load your data in our case we will choose upload, we're going to upload it in our local machine

Create table

Source

Create table from

- Empty table
- Google Cloud Storage
- Upload
- Drive
- Google Bigtable
- Amazon S3
- Azure Blob Storage

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

Table type

Native table

Schema

CREATE TABLE CANCEL

#### - Source – section

In this section of creating our table we uploaded our **spotify.csv** file data

Source

Create table from

Upload

Select file \*

spotify.csv

BROWSE

File format

CSV

#### - Destination – section

In this section we associate a name to our table **data**

Destination

Project \*

bigdataproject-409418

BROWSE

Dataset \*

spotify\_data

Table \*

data

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

Table type

Native table

#### - Schema – section

Check the **Auto detect** so the schema will be automatically generated

Schema

☒ Auto detect

Schema will be automatically generated.

Our table with loaded data has been created successfully:

The screenshot shows the Google Cloud BigQuery Explorer interface. On the left, the 'Explorer' pane shows the project 'bigdatapoint-409418' and the table 'data'. The main pane displays the 'SCHEMA' tab for the 'data' table. The schema table lists the following fields:

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
int64_field_0	INTEGER	NULLABLE	-	-	-	-	-
song_name	STRING	NULLABLE	-	-	-	-	-
song_popularity	STRING	NULLABLE	-	-	-	-	-
song_duration_ms	FLOAT	NULLABLE	-	-	-	-	-
acousticness	STRING	NULLABLE	-	-	-	-	-
danceability	STRING	NULLABLE	-	-	-	-	-
energy	STRING	NULLABLE	-	-	-	-	-
instrumentalness	STRING	NULLABLE	-	-	-	-	-

A notification at the bottom says '"data" created.' with a 'GO TO TABLE' button.

Click on **Preview** the data:

The screenshot shows the Google Cloud BigQuery Explorer interface with the 'Preview' tab selected. The preview table displays the following data:

Row	int64_field_0	song_name	song_popularity	song_duration_ms	acousticness	danceability	energy
1	1757	Party In The U.S.A.	nao_sei	123556.0	0.519mol/L	0.36	0
2	11171	Aura	50	102536.0	0.0774kg	0.0mol/L	0.56
3	7119	Gina Rodriguez - Latinos Trendi...	0	12000.0	0.7kg	0.0mol/L	0.493
4	4873	Balance	54	282727.0	0.907kg	0.2mol/L	0.106
5	558	Poison	60	269600.0	0.0515kg	0.3mol/L	0.894
6	687	Poison	60	269600.0	0.0515kg	0.3mol/L	0.894
7	3319	Mykonos	38	275306.0	0.395kg	0.3mol/L	0.452
8	16266	Mykonos	38	275306.0	0.395kg	0.3mol/L	0.452
9	14139	Broken Bones	33	181453.0	0.000259kg	0.3mol/L	0.98
10	7811	Heavenly Day	53	225400.0	0.7979999999999999kg	0.3mol/L	0.364
11	8436	Red Cold River	7	200907.0	0.000196kg	0.3mol/L	0.961
12	1567	Lover Of The Light	60	314893.0	0.00725kg	0.3mol/L	0.707
13	5528	T-R-O-U-B-L-E - Remastered	48	181333.0	0.0121kg	0.3mol/L	0.965

Results per page: 50. 1 - 50 of 18835.

Now we are going to create SQL query in a split tab

→ Query -> In split tab

The screenshot shows the Google Cloud BigQuery Explorer interface with a split tab. The left pane shows the 'data' table preview. The right pane, titled 'Untitled 2', contains the following SQL query:

```
SELECT * FROM `bigdatapoint-409418.spotify_data.data` LIMIT 1000
```

A notification at the bottom right says 'This query will process 2.35 MB when run.' and 'Press Alt+F1 for Accessibility Options.'

## II. The Principle of this demo:

In this guide, I aim to showcase the process of creating a comprehensive BigQuery project focused on exploring and analyzing Spotify's music dataset. The queries I'll execute serve various purposes: some aim to extract meaningful insights by aggregating data, such as calculating song popularity trends over time, while others involve data cleansing, transforming string values to numeric types for accurate analysis. Additionally, I'll perform exploratory queries to identify duplicated records based on song names and handle inconsistencies, thereby ensuring data accuracy. These queries encompass diverse functionalities, including JOIN operations to merge audio features and song information, struct usage to organize nested data, and conversions from string to numeric types for proper analysis. Overall, these queries serve as a foundational framework to navigate and derive insights from the rich Spotify dataset, encompassing analytical, transformational, and data integrity aspects.

### 1. Execution of queries in BigQuery:

- Let's start with the simple default query:

``bigdataproject-409418.spotify_data.data`` to get data from the source table, `bigdataproject-409418` represents the project's name, `spotify_data` represents the dataset's name and `data` is the table that contains our loaded data

The screenshot shows the Google Cloud BigQuery console interface. At the top, there's a search bar and navigation icons. Below that, a query editor shows a simple query: `SELECT * FROM `bigdataproject-409418.spotify_data.data` LIMIT 1000`. The query is executed, and the results are displayed in a table. The table has columns for `int64_field_0`, `song_name`, `song_popularity`, `song_duration_ms`, `acousticness`, `danceability`, and `energy`. The results show 9 rows of data, including songs like "Party In The U.S.A.", "Aura", "Gina Rodriguez - Latinos Trendi...", "Balance", "Poison", "Mykonos", and "Broken Bones".

Row	int64_field_0	song_name	song_popularity	song_duration_ms	acousticness	danceability	energy
1	1757	Party In The U.S.A.	nao_sei	123556.0	0.519mol/L	0.36	0
2	11171	Aura	50	102536.0	0.0774kg	0.0mol/L	0.5
3	7119	Gina Rodriguez - Latinos Trendi...	0	12000.0	0.7kg	0.0mol/L	0.4
4	4873	Balance	54	282727.0	0.907kg	0.2mol/L	0.1
5	558	Poison	60	269600.0	0.0515kg	0.3mol/L	0.8
6	687	Poison	60	269600.0	0.0515kg	0.3mol/L	0.8
7	3319	Mykonos	38	275306.0	0.395kg	0.3mol/L	0.4
8	16266	Mykonos	38	275306.0	0.395kg	0.3mol/L	0.4
9	14139	Broken Bones	33	181453.0	0.000259kg	0.3mol/L	0.9

- We will segment the queries to 5 categories:

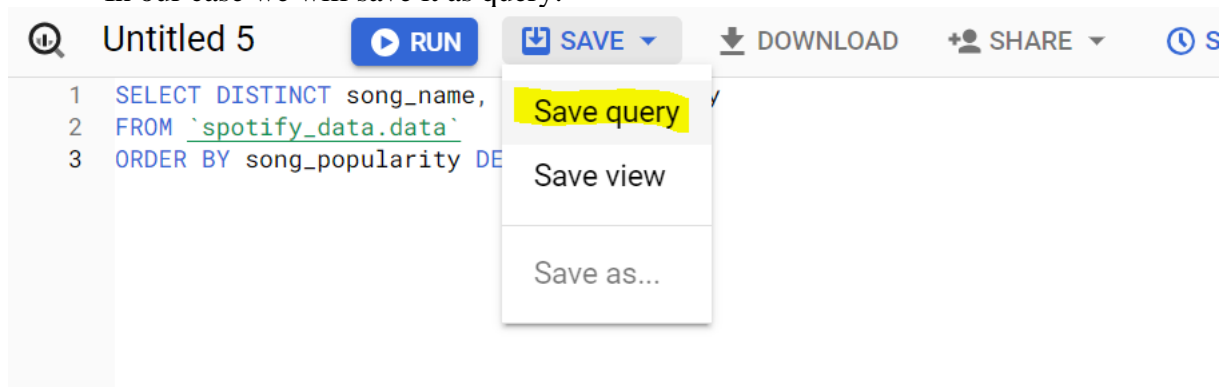
### 1- Select Queries

#### 1.1- Top Tracks by Popularity:

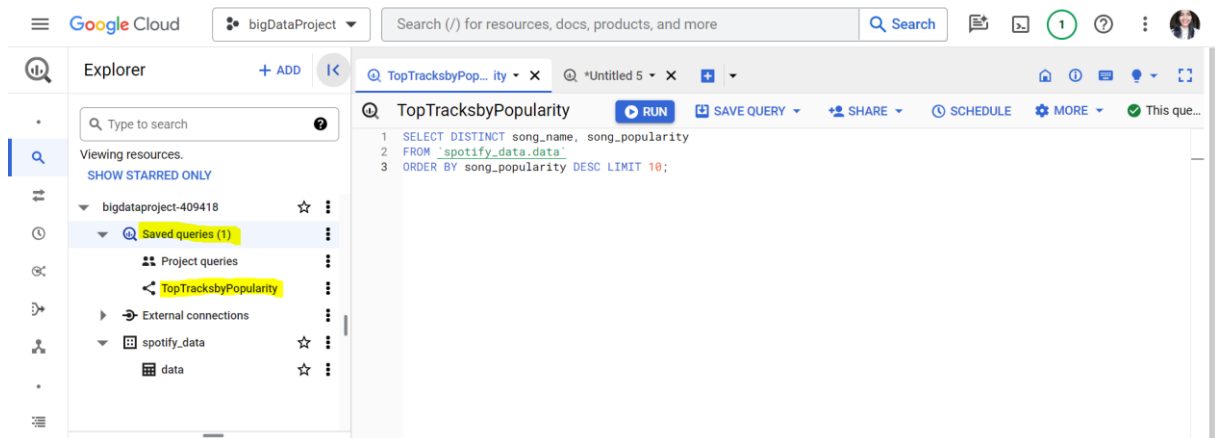
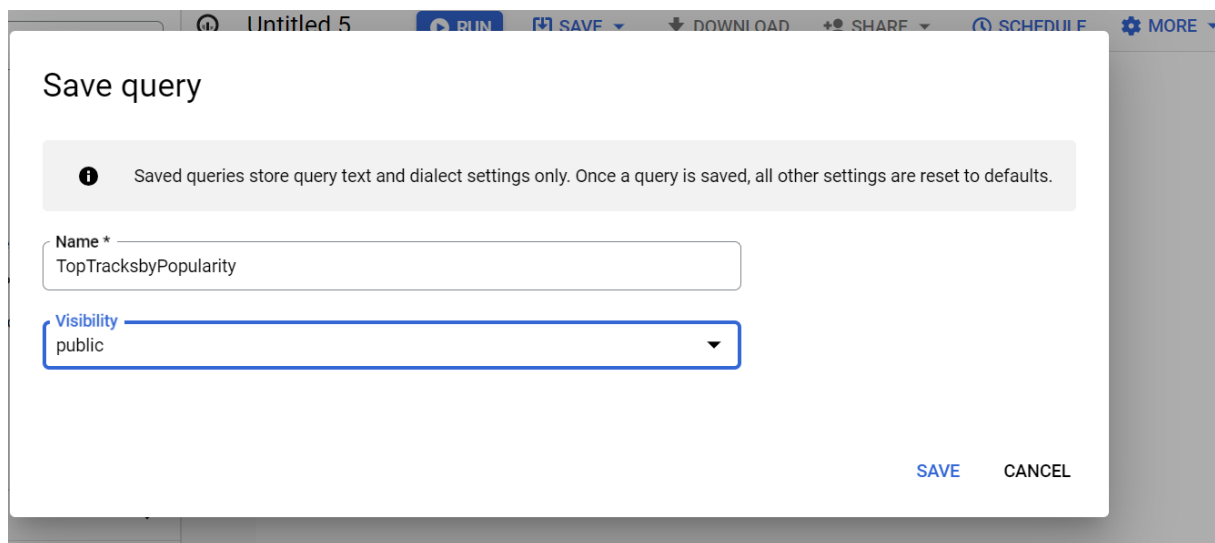
The screenshot shows the Google Cloud BigQuery console interface. The query editor displays a query to select the top 10 tracks by popularity: `#`spotify_data.data`  
SELECT DISTINCT song_name, song_popularity  
FROM `spotify_data.data`  
ORDER BY song_popularity DESC LIMIT 10;`. The query is executed, and the results are displayed in a table. The table has columns for `song_name` and `song_popularity`. The results show 9 rows of data, including songs like "Party In The U.S.A.", "I Love It (& Lil Pump)", "In My Feelings", "Eastside (with Halsey & Khalid)", "Taki Taki (with Selena Gomez, ...)", "Promises (with Sam Smith)", "Falling Down", "In My Mind", and "SICKO MODE".

Row	song_name	song_popularity
1	Party In The U.S.A.	nao_sei
2	I Love It (& Lil Pump)	99
3	In My Feelings	98
4	Eastside (with Halsey & Khalid)	98
5	Taki Taki (with Selena Gomez, ...)	98
6	Promises (with Sam Smith)	98
7	Falling Down	97
8	In My Mind	97
9	SICKO MODE	97

- Save the Query as a view or query and give it a name:  
In our case we will save it as query:



We will give it the following name:





## 1.2- Audio Features Analysis:

The screenshot shows the Google Cloud BigQuery console. At the top, there's a search bar and a dropdown for 'bigDataProject'. Below that, the 'Explorer' panel is open, showing a query named 'Untitled 5'. The query is:

```
1 SELECT song_name, energy, instrumentalness, tempo, time_signature, audio_valence
2 FROM `bigqueryproject.spotify_data.Table`
3 WHERE song_name = 'Poison';
```

The query results are displayed in a table with 7 columns: Row, song\_name, energy, instrumentalness, tempo, time\_signature, and audio\_valence. The results show 5 rows of data for the song 'Poison'.

Row	song_name	energy	instrumentalness	tempo	time_signature	audio_valence
1	Poison	0.894	0.000333	118.631	4	0.
2	Poison	0.894	0.000333	118.631	4	0.
3	Poison	0.575	0.00872	111.892	4	0.
4	Poison	0.394	0.000905	117.489	3	0.
5	Poison	0.573	0.00432	111.848	4	0.8029999999

- We will try now to save this query as a view:

The screenshot shows the Google Cloud BigQuery console with the 'SAVE' dropdown menu open. The menu options are: 'Save query', 'Save view' (highlighted in yellow), and 'Save as...'. The query text is visible in the background:

```
1 SELECT song_name, energy, i
2 FROM `bigqueryproject.spotify_data.Table`
3 WHERE song_name = 'Poison';
```

The screenshot shows the 'Save view' dialog box. It contains the following information:

- Project \***: bigdataproyect-409418 (with a 'BROWSE' button)
- Dataset \***: spotify\_data
- Table \***: AudioFeaturesAnalysisOfTheSong-Poison

Below the table name, there is a note: 'Unicode letters, marks, numbers, connectors, dashes or spaces allowed. The job will create the specified destination table if needed.' At the bottom right, there are 'SAVE' and 'CANCEL' buttons.

Our view is saved:

The screenshot shows the Google Cloud BigQuery interface. On the left, the 'Explorer' pane displays a project named 'bigdatapoint-409418' with a folder 'spotify\_data' containing a view 'AudioFeaturesAnalysis...'. The main pane shows the 'Schema' tab for this view, displaying a table with 8 columns: 'song\_name', 'energy', 'instrumentalness', 'tempo', 'time\_signature', 'audio\_valence', and two unnamed columns. The schema details are as follows:

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
song_name	STRING	NULLABLE	-	-	-	-	-
energy	STRING	NULLABLE	-	-	-	-	-
instrumentalness	STRING	NULLABLE	-	-	-	-	-
tempo	STRING	NULLABLE	-	-	-	-	-
time_signature	STRING	NULLABLE	-	-	-	-	-
audio_valence	FLOAT	NULLABLE	-	-	-	-	-

## 2- Aggregation Queries:

### 2.1- Average Duration of Songs:

The screenshot shows the Google Cloud BigQuery interface with a query titled 'Untitled 5'. The query is:

```
1 SELECT AVG(song_duration_ms) AS avg_duration, song_name AS name
2 FROM `spotify_data.data`
3 GROUP BY song_name;
```

The query results are displayed in a table with 9 rows:

Row	avg_duration	name
1	175896.0	Party In The U.S.A.
2	102536.0	Aura
3	12000.0	Gina Rodriguez - Latinos Trendl...
4	221019.66666666666	Balance
5	255182.2	Poison
6	275306.0	Mykonos
7	181453.0	Broken Bones
8	225400.0	Heavenly Day
9	200907.0	Red Cold River

### 2.2- Distribution of Audio Modes:

The screenshot shows the Google Cloud BigQuery interface with a query titled 'Untitled 5'. The query is:

```
1 SELECT audio_mode, COUNT(*) AS mode_count
2 FROM `spotify_data.data`
3 GROUP BY audio_mode;
```

The query results are displayed in a table with 4 rows:

Row	audio_mode	mode_count
1	0.105	1
2	1	11830
3	0	7003
4	nao_sei	1

### 3- Conditional Queries:

#### 3.1- High Energy and High Danceability Tracks:

Q ALL\_QUERIES x Q \*Untitled 5 x

Untitled 5 **RUN** **SAVE** **DOWNLOAD** **SHARE** **SCHEDULE** **MORE** ✓ This query will process 617.62 KB when run.

```
1 SELECT song_name, energy, audio_valence
2 FROM `spotify_data.data`
3 WHERE CAST(energy AS float64) > 0.8 AND audio_valence > 0.6 LIMIT 9;
```

Press Alt+F1 for Accessibility Options.

Query results **SAVE RESULTS** **EXPLORE DATA**

JOB INFORMATION	RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	song_name	energy	audio_valence			
1	T-R-O-U-B-L-E - Remastered	0.965	0.793			
2	Find Me	0.946	0.611			
3	Danza Kuduro	0.915	0.878			
4	Danza Kuduro	0.915	0.878			
5	Rebellion (Lies)	0.934	0.731			
6	Murder Was The Case (Death After Visualizing Eternity) (feat. Dat Nigga Daz)	0.841	0.62			
7	The Passenger	0.846	0.738			
8	The Passenger	0.846	0.738			
9	You - Ha Ha Ha	0.902	0.619			

#### 3.2- Tracks with High Speechiness and Low Instrumentalness:

Q \*ALL\_QUERIES x Q \*Untitled 5 x

Untitled 5 **RUN** **SAVE** **DOWNLOAD** **SHARE** **SCHEDULE** **MORE** ✓ Query completed.

```
1 SELECT song_name, speechiness, instrumentalness
2 FROM `spotify_data.data`
3 WHERE CAST(speechiness AS float64) > 0.2 AND CAST(instrumentalness AS float64) < 0.2 limit 20
```

Press Alt+F1 for Accessibility Options.

Query results **SAVE RESULTS** **EXPLORE DATA**

JOB INFORMATION	RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	song_name	speechiness	instrumentalness			
1	Danza Kuduro	0.28	0			
2	Danza Kuduro	0.28	0			
3	Flava In Ya Ear (feat. The Notorious B.I.G., LL Cool J, Busta Rhymes & Rampage) - 2016 Remastered	0.467	0			
4	Gold	0.344	0.0025			
5	TONYA	0.299	9.24E-06			
6	I Lied	0.293	0			

#### 3.3- Songs with Lengthy Duration and High Popularity:

Q ALL\_QUERIES x Q \*Untitled 5 x

Untitled 5 **RUN** **SAVE** **DOWNLOAD** **SHARE** **SCHEDULE** **MORE** ✓ Query completed.

```
1 SELECT song_name, song_duration_ms, song_popularity
2 FROM `spotify_data.data`
3 WHERE CAST(song_duration_ms AS int64) > 300000 AND CAST(song_popularity AS float64) > 80 LIMIT 10;
```

Press Alt+F1 for Accessibility Options.

Query results **SAVE RESULTS** **EXPLORE DATA**

JOB INFORMATION	RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	song_name	song_duration_ms	song_popularity			
1	Self Care	345133.0	87			
2	Self Care	345133.0	87			
3	Self Care	345133.0	87			
4	Self Care	345133.0	87			
5	The Scientist	309600.0	86			
6	The Scientist	309600.0	86			
7	The Scientist	309600.0	86			
8	New Patek	343099.0	90			
9	New Patek	343099.0	90			
10	New Patek	343099.0	90			

#### 4- CTE (Common Table Expression)

The CTE enables us to create temporary result sets.

- low Energy Tracks

The screenshot shows the Google Cloud BigQuery console interface. At the top, there's a tab for 'Untitled 5'. Below the query editor, the query is displayed:

```
1 WITH HighEnergyTracks AS (  
2   SELECT song_name, energy  
3   FROM `spotify_data.data`  
4   WHERE CAST(energy AS float64) > 0.4  
5   limit 3  
6 )  
7 SELECT * FROM HighEnergyTracks;
```

The 'Query results' section is visible, showing a table with 3 rows and 2 columns: 'song\_name' and 'energy'.

Row	song_name	energy
1	Aura	0.56
2	Gina Rodriguez - Latinos Trendi...	0.493
3	Poison	0.894

BigQuery supports the use of INSERT, DELETE, and UPDATE statements, but with certain limitations and considerations due to its nature as a fully managed, serverless data warehouse designed primarily for analytical workloads.

Note that there are some queries such as Insert queries won't work if you are using **SandBox**.  
**You have to activate the billing account**

- **Insert statement:**

The screenshot shows the Google Cloud BigQuery console interface. At the top, there's a tab for 'Untitled 3'. Below the query editor, the query is displayed:

```
1 INSERT INTO `spotify_data.data` (song_name)  
2 VALUES ( 'DTF-ME GUSTA'),  
3 | | | ('SYKO-BLOOD');
```

The 'Query results' section is visible, showing a message: 'This statement added 2 rows to data.' and a 'GO TO TABLE' button.

Untitled 3

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Query completed

```
1 select * from `spotify_data.data` where
2 song_name = 'DTF-ME GUSTA' OR song_name = 'SYKO-BLOOD'
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATIONRESULTSCHARTPREVIEWJSONEXECUTION DETAILSEXECUTION GRAPH

Row	int64_field_0	song_name	song_popularity	song_duration_ms	acousticness
1	null	DTF-ME GUSTA	null	null	null
2	null	SYKO-BLOOD	null	null	null

## - Delete statement:

ALL\_QUERIESUntitled 3

Untitled 3

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Query completed

```
1 DELETE FROM `spotify_data.data`
2 WHERE song_name = 'SYKO-BLOOD';
3
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATIONRESULTSEXECUTION DETAILSEXECUTION GRAPH

This statement removed 1 row from data.

GO TO TABLE

Query updated

## - Select query with STRUCT

ALL\_QUERIESUntitled 3

Untitled 3

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Query completed

```
1 SELECT
2   song_name AS Song,
3   STRUCT(song_popularity,song_duration_ms,acousticness,energy,instrumentalness,liveness,loudness,audio_mode ,
4   speechiness,tempo,time_signature,audio_valence
5   ) AS audio_attributes
6 FROM `spotify_data.data`
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATIONRESULTSCHARTPREVIEWJSONEXECUTION DETAILSEXECUTION GRAPH

Row	Song	audio_attr...	song_popularity	song_duration_ms	audio_attribu...	acousticness	audio_attributes.
1	DTF-ME GUSTA	null		null	null		null
2	Party In The U.S.A.	nao_sei		123556.0	0.519mol/L		0
3	Aura	50		102536.0	0.0774kg		0.56
4	Gina Rodriguez - Latinos Trendi...	0		12000.0	0.7kg		0.493
5	Balance	54		282727.0	0.907kg		0.106
6	Poison	60		269600.0	0.0515kg		0.894
7	Poison	60		269600.0	0.0515kg		0.894

## - Query to Count Duplicated Lines Based on Song Name:

ALL\_QUERIES x \*Untitled 3 x +

Untitled 3 RUN SAVE DOWNLOAD SHARE SCHEDULE MORE Query completed

```
1 SELECT song_name, COUNT(*) as nbrName FROM `spotify_data.data`
2 GROUP BY song_name
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS CHART PREVIEW JSON EXECUTION DETAILS EXECUTION GRAPH

Row	song_name	nbrName
1	Party In The U.S.A.	3
2	Aura	1
3	Gina Rodriguez - Latinos Trendi...	1
4	Balance	3
5	Poison	5
6	Mykonos	3
7	Broken Bones	1
8	Heavenly Day	1
9	Red Cold River	1
10	Lover Of The Light	1

Results per page: 50 1 - 50 of 13071

## You can give access to people to access your project ;

Google Cloud IAM

- Cloud overview
- Products & solutions
- PINNED PRODUCTS
- APIs & Services
- Billing
- IAM & Admin**
- Marketplace
- Compute Engine
- Kubernetes Engine
- Cloud Storage
- BigQuery

- Identity & Organization
- Policy Troubleshooter
- Policy Analyzer
- Organization Policies
- Service Accounts
- Workload Identity Federation
- Workforce Identity Federation
- Labels
- Tags
- Settings
- Privacy & Security
- Identity-Aware Proxy
- Roles
- Audit Logs
- Manage Resources

for resources, docs, products, and more Search

Untitled 2 x ALL\_QUERIES x +

ALL\_QUERIES RUN SAVE QUERY SHARE SCHEDULE MORE This script will proce...

```
SELECT DISTINCT song_name, song_popularity
FROM `spotify_data.data`
ORDER BY song_popularity DESC LIMIT 10;

SELECT song_name, energy, instrumentalness, tempo, time_signature, audio_valence
FROM `spotify_data.data`
WHERE song_name = 'Poison';

SELECT AVG(song_duration_ms) AS avg_duration, song_name AS name
FROM `spotify_data.data`
GROUP BY song_name;

SELECT audio_mode, COUNT(*) AS mode_count
FROM `spotify_data.data`
GROUP BY audio_mode;

SELECT song_name, energy, audio_valence
FROM `spotify_data.data`
WHERE (energy AS float64) > 0.8 AND audio_valence > 0.6 LIMIT 9;

SELECT song_name, speechiness, instrumentalness
FROM `spotify_data.data`
WHERE CAST(speechiness AS float64) > 0.2 AND CAST(instrumentalness AS float64) < 0.2 limit 20;
```

Press Alt+F1 for Accessibility Options.

https://console.cloud.google.com/iam-admin/iam?project=bigdataproject-4...

Google Cloud

bigDataProject

Se

IAM

PERMISSIONS

RECOMMENDATIONS HISTORY

Permissions for project "bigDataProject"

These permissions affect this project and all of its resource

VIEW BY PRINCIPALS

VIEW BY ROLES

GRANT ACCESS

REMOVE ACCESS

Filter

Enter property name or value

Type

Principal

Name

aithajar55@gmail.com

HAJAR AIT

Grant access to "bigDataProject"

specific criteria is met. [Learn more about IAM conditions](#)

Resource

bigDataProject

Add principals

Principals are users, groups, domains, or service accounts. [Learn more about principals in IAM](#)

New principals \*

aitabdielmomin.ensa@uhp.ac.ma

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role \*

Owner

IAM condition (optional)

+ ADD IAM CONDITION

Full access to most Google Cloud resources. See the list of included permissions.

SAVE

CANCEL

- **The invited user:**  
He has to accept the invitation so he can access the project

HAJAR AIT ABDIELMOMIN <noreply-cloud@google.com>

02:17 (il y a 0 minute)

☆

↩

⋮

À moi ▾

Hello, [aitabdielmomin.ensa@uhp.ac.ma](mailto:aitabdielmomin.ensa@uhp.ac.ma),

I invite you to join the Google Cloud project "bigDataProject" (id: bigdataproject-409418). Please click this link to accept my invitation:

<https://console.cloud.google.com/invitation?project=bigdataproject-409418&memberEmail=aitabdielmomin.ensa@uhp.ac.ma&account=aitabdielmomin.ensa@uhp.ac.ma>

Thanks,  
[aithajar55@gmail.com](mailto:aithajar55@gmail.com)

Google Cloud Platform

Build your apps on Google's infrastructure.

Take advantage of the power and performance of Google APIs like Maps, Social, Translate, Android, and much more. [Learn more](#)

In addition, Google Cloud lets you build, deploy, and scale applications, websites, and services on the same infrastructure that runs Google. [Learn more](#)

© 2023 Google LLC 1600 Amphitheatre Parkway, Mountain View, CA 94043

You have reached this message's end. To update you about important changes to