

Bootcamp Data Science - Caso Práctico S16 - Gerardo Rodríguez.docx

Paso 1: Preparación del Dataset y Entorno de Trabajo

1. Agregar un Dataset y una tabla a BigQuery:

- Ve a la consola de Google Cloud.
- Crea un nuevo proyecto y navega a BigQuery.
- En el panel de recursos, crea un nuevo conjunto de datos.
- Carga un dataset público disponible en BigQuery o un dataset personal subiéndolo a tu espacio de trabajo.

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the project name 'UVM-CasoPractico-S16', a search bar, and user profile icons. The left sidebar contains the 'Explorer' panel with a search bar and a list of resources. The 'sales' dataset is highlighted in the 'Case16' folder. The main panel displays the 'sales' dataset in a table view, showing columns: Row, invoice_and_item_number, date, store_number, store_name, address, and city. The table contains 12 rows of data. The bottom of the console shows the 'Job history' section.

Row	invoice_and_item_number	date	store_number	store_name	address	city
1	INV-20328300048	2019-07-01	5476	ELDRIDGE MART	209 E LECLAIRE RD	ELDRIDGE
2	INV-20944300023	2019-07-30	5440	QUIK-PIK	502 WEST ERIE ST.	MISSOURI VALLEY
3	INV-61865000058	2023-09-01	5327	MEGA SAVER / COUNCIL BLUF...	3540 W BROADWAY	COUNCIL BLUFFS
4	INV-45179500032	2022-03-01	4844	IOWA CITY FAST BREAK	2580, NAPLES AVE	IOWA CITY
5	INV-41371300043	2021-10-26	5669	GUDDI MART / WATERLOO	306 BYRON AVE	WATERLOO
6	INV-30039200025	2020-09-08	6000	PORTERS CONVENIENCE 66 / ...	1512 LINCOLN WAY	CLINTON
7	INV-58818500021	2023-05-26	6137	STAR CONVENIENCE / FORT D...	1601 5TH AVENUE	FORT DODGE
8	INV-42988300002	2021-12-17	6080	CHEEMA GAS AND LIQUORS / ...	202 S MAIN ST	NEW SHARON
9	INV-22028100005	2019-09-20	4369	KUM & GO #28 / NORWALK	2992 SUNSET DR	NORWALK
10	INV-45310200027	2022-07-27	6214	CASEY'S #3934 / ANKENY	1515 SE CORPORATE WOODS ...	ANKENY
11	INV-36255300109	2021-04-30	4291	FAREWAY STORES #597 / CRE...	105 E ADAMS ST	CRESTON
12	INV-39914400030	2021-09-09	6161	BUCKY'S EXPRESS #22 / COU...	3434 NEBRASKA AVE	COUNCIL BLUFFS

```

[1] pip install google-cloud-bigquery
[1] Python
Defaulting to user installation because normal site-packages is not writable
Requirement already satisfied: google-cloud-bigquery in c:\programdata\anaconda3\lib\site-packages (0.23.0)
Requirement already satisfied: google-api-core<2.0.0,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from google-cloud-bigquery) (1.26.0)
Requirement already satisfied: google-auth<3.0.0dev,>=2.14.1 in c:\programdata\anaconda3\lib\site-packages (from google-cloud-bigquery) (2.20.0)
Requirement already satisfied: google-cloud-core<0.6.0dev,>=1.4.0 in c:\programdata\anaconda3\lib\site-packages (from google-cloud-bigquery) (2.4.1)
Requirement already satisfied: google-reusable-media<0.6.0,>=0.6.0 in c:\programdata\anaconda3\lib\site-packages (from google-cloud-bigquery) (2.0.0)
Requirement already satisfied: packaging<20.0 in c:\programdata\anaconda3\lib\site-packages (from google-cloud-bigquery) (21.1)
Requirement already satisfied: python-dateutil<2.9.0dev,>=2.7.2 in c:\programdata\anaconda3\lib\site-packages (from google-cloud-bigquery) (2.8.2)
Requirement already satisfied: requests<3.0.0dev,>=2.21.0 in c:\programdata\anaconda3\lib\site-packages (from google-cloud-bigquery) (2.28.1)
Requirement already satisfied: googleapis-common-protos<0.6.0dev,>=1.56.2 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<2.0.0,>=1.21.1,>=1.20.0) (1.56.2)
Requirement already satisfied: protobuf<3.20.0,>=3.20.1 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<2.0.0,>=1.21.1,>=1.20.0) (3.19.5)
Requirement already satisfied: proto-plus<0.8.0dev,>=1.2.3 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<2.0.0,>=1.21.1,>=1.20.0) (1.2.3)
Requirement already satisfied: grpcio<0.6.0dev,>=1.33.1 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<2.0.0,>=1.21.1,>=1.20.0) (1.36.0)
Requirement already satisfied: grpcio-status<0.6.0dev,>=1.33.2 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<2.0.0,>=1.21.1,>=1.20.0) (1.33.2)
Requirement already satisfied: cachetools<5.0.0dev,>=2.1.1 in c:\programdata\anaconda3\lib\site-packages (from google-auth<3.0.0dev,>=2.14.1) (4.2.2)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\programdata\anaconda3\lib\site-packages (from google-auth<3.0.0dev,>=2.14.1) (4.7.1)
Requirement already satisfied: google-crc32c<0.6.0dev,>=1.0 in c:\programdata\anaconda3\lib\site-packages (from google-reusable-media<0.6.0,>=0.6.0) (1.0.0)
Requirement already satisfied: charset-normalizer<2.0.0,>=1.2 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0.0dev,>=2.21.0) (2.0.4)
Requirement already satisfied: idna<2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0.0dev,>=2.21.0) (3.0)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0.0dev,>=2.21.0) (1.26.1)
Requirement already satisfied: certifi<2017.4.17,>=2017.4.15 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0.0dev,>=2.21.0) (2020.2.2)
Requirement already satisfied: pyasn1<0.5.0dev,>=0.4.6 in c:\programdata\anaconda3\lib\site-packages (from pyasn1<0.5.0dev,>=0.4.6) (0.4.8)
Note: you may need to restart the kernel to use updated packages.

[1] pip install pandas
[1] Python
Defaulting to user installation because normal site-packages is not writable
Requirement already satisfied: pandas-gbq in c:\programdata\anaconda3\lib\site-packages (0.23.0)
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (58.2.2)
Requirement already satisfied: db-dtypes<0.8.0,>=0.4.0 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (1.0.0)
Requirement already satisfied: numpy<1.16.6 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (1.20.4)
Requirement already satisfied: pandas<1.1.4 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (1.1.4)
Requirement already satisfied: google-auth<2.0.0,>=1.0 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (2.20.0)
Requirement already satisfied: pydata-google-auth<1.5.0 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (1.8.2)
Requirement already satisfied: google-api-core<0.6.0dev,>=1.20.2 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (2.19.0)
Requirement already satisfied: google-cloud-core<0.6.0dev,>=1.2.3 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (2.4.1)
Requirement already satisfied: google-auth-oauthlib<0.7.0 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (1.2.0)
Requirement already satisfied: google-cloud-bigquery<0.6.0dev,>=1.3.5 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (3.2.1)
Requirement already satisfied: packaging<20.0 in c:\programdata\anaconda3\lib\site-packages (from pandas-gbq) (21.1)
Requirement already satisfied: proto-plus<0.8.0dev,>=1.2.3 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<0.6.0dev,>=1.20.2) (1.2.3)
Requirement already satisfied: protobuf<3.20.0,>=3.20.1 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<0.6.0dev,>=1.20.2) (3.19.5)
Requirement already satisfied: proto-plus<0.8.0dev,>=1.2.3 in c:\programdata\anaconda3\lib\site-packages (from google-api-core<0.6.0dev,>=1.20.2) (1.2.3)
Requirement already satisfied: requests<3.0.0dev,>=2.18.0 in c:\programdata\anaconda3\lib\site-packages (from google-auth-oauthlib<0.7.0) (2.28.1)
Requirement already satisfied: google-auth<2.0.0,>=1.0 in c:\programdata\anaconda3\lib\site-packages (from google-auth-oauthlib<0.7.0) (2.20.0)
Requirement already satisfied: pyasn1-modules<0.2.1 in c:\programdata\anaconda3\lib\site-packages (from google-auth<2.0.0,>=1.0) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\programdata\anaconda3\lib\site-packages (from google-auth<2.0.0,>=1.0) (4.7.1)
Requirement already satisfied: requests-oauthlib<0.7.0 in c:\programdata\anaconda3\lib\site-packages (from google-auth-oauthlib<0.7.0) (1.0.0)

[1] pip install Matplotlib
[1] Python
Defaulting to user installation because normal site-packages is not writable
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: Matplotlib in c:\programdata\anaconda3\lib\site-packages (3.8.0)
Requirement already satisfied: contourpy<1.0.0,>=1.0 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (1.2.0)
Requirement already satisfied: cycler<0.10,>=0.10 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (0.11.0)
Requirement already satisfied: fonttools<0.42.0 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver<1.0.1 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (1.4.4)
Requirement already satisfied: numpy<2,>=1.21 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (1.26.4)
Requirement already satisfied: packaging<20.0 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (23.1)
Requirement already satisfied: pillow<6.2.0 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (10.0.0)
Requirement already satisfied: pyparsing<2.3.1 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil<2.7 in c:\programdata\anaconda3\lib\site-packages (from Matplotlib) (2.8.2)
Requirement already satisfied: six<1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil<2.7) (1.16.0)

[1] pip install pandas
[1] Python
Defaulting to user installation because normal site-packages is not writable
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (2.1.4)
Requirement already satisfied: numpy<2,>=1.23.2 in c:\programdata\anaconda3\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil<2.8.2 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz<2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2020.3)
Requirement already satisfied: tzdata<2021.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: requests<3.0.0dev,>=2.18.0 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil<2.8.2) (2.28.1)
Note: you may need to restart the kernel to use updated packages.

[1] pip install seaborn
[1] Python
Defaulting to user installation because normal site-packages is not writable
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (0.12.2)
Requirement already satisfied: numpy<1.24.0,>=1.17 in c:\programdata\anaconda3\lib\site-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas<0.25 in c:\programdata\anaconda3\lib\site-packages (from seaborn) (2.1.4)
Requirement already satisfied: matplotlib<3.6.1,>=3.1 in c:\programdata\anaconda3\lib\site-packages (from seaborn) (3.8.0)
Requirement already satisfied: contourpy<0.1.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (1.2.0)
Requirement already satisfied: cycler<0.10,>=0.10 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (0.11.0)
Requirement already satisfied: fonttools<0.42.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (4.25.0)
Requirement already satisfied: kiwisolver<1.0.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (1.4.4)
Requirement already satisfied: packaging<20.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (23.1)
Requirement already satisfied: pillow<6.2.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (10.0.0)
Requirement already satisfied: pyparsing<2.3.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (3.0.9)
Requirement already satisfied: python-dateutil<2.7 in c:\programdata\anaconda3\lib\site-packages (from matplotlib<3.6.1,>=3.1) (2.8.2)
Requirement already satisfied: six<1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil<2.7) (1.16.0)
Requirement already satisfied: tzdata<2021.1 in c:\programdata\anaconda3\lib\site-packages (from pandas<0.25) (2023.3)
Requirement already satisfied: requests<3.0.0dev,>=2.18.0 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil<2.7) (2.28.1)
Note: you may need to restart the kernel to use updated packages.

```

Paso 2: Consultas SQL en BigQuery (Sugeridas: 2)

A continuación, se presentan 4 ejemplos de consultas a ejecutar en BigQuery:

a) Consulta de datos agregados	Transacciones totales por tienda
<pre>SELECT category, COUNT(*) as total_category FROM `project.dataset.table` GROUP BY category ORDER BY total_category DESC</pre>	<pre>SELECT store_name, COUNT(*) as transacciones_totales_por_tienda FROM `uvm-casopractico-s16.Caso16.sales` GROUP BY store_name ORDER BY transacciones_totales_por_tienda DESC LIMIT 10000</pre>

Untitled query

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

```

1 SELECT store_name, COUNT(*) as transacciones_totales_por_tienda
2 FROM `uvm-casopractico-s16.Caso16.sales`
3 GROUP BY store_name
4 ORDER BY transacciones_totales_por_tienda DESC
5 LIMIT 10000

```

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	store_name	transacciones_totale				
1	HY-VEE #3 / BDI / DES MOINES	238201				
2	CENTRAL CITY 2	206094				
3	CENTRAL CITY LIQUOR, INC.	181243				
4	HY-VEE FOOD STORE / CEDAR ...	153205				
5	HY-VEE WINE AND SPIRITS / B...	150875				
6	HY-VEE #7 / CEDAR RAPIDS	142392				
7	HY-VEE WINE AND SPIRITS / I...	141557				

Results per page: 50
 1 – 50 of 3271
 |< < > >|

Se muestran las tiendas mostrando las transacciones que han entregado.

Se identifican 3271 tiendas diferentes.

b) Consulta de Datos con Filtro:	Promedio de ventas de tiendas por Condado
<pre>SELECT name, number FROM `project.dataset.table` WHERE number > 100.0</pre>	<pre>SELECT county, promventas_por_tienda FROM (SELECT county, SUM(sale_dollars) / COUNT(store_name) as promventas_por_tienda FROM `uvm-casopractico-s16.Caso16.sales` GROUP BY county) sub WHERE promventas_por_tienda >= 100 ORDER BY promventas_por_tienda DESC</pre>

Untitled query RUN SAVE DOWNLOAD SHARE SCHEDULE MORE Query completed.

```
1 SELECT county, promventas_por_tienda
2 FROM (
3     SELECT county, SUM(sale_dollars) / COUNT(store_name) as promventas_por_tienda
4     FROM `uvm-casopractico-s16.Caso16.sales`
5     GROUP BY county
6 ) sub
7 WHERE promventas_por_tienda >= 100
8 ORDER BY promventas_por_tienda DESC
9
```

Press Alt+F1 for Accessibility Options

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	county	promventas_por_tien				
1	DALLAS	265.1338954121...				
2	POLK	178.3690968750...				
3	JOHNSON	172.9858387922...				
4	SCOTT	168.5630438626...				
5	CLINTON	162.6155465376...				

Results per page: 50 1 - 50 of 67 REFRESH

Job history

Solo 67 de los 101 condados tienen un promedio de ventas por tienda superior a 100.

c) Consulta de Tendencias a lo Largo del Tiempo:	Sumatoria de las ventas totales por año
SELECT year, number FROM `project.dataset.table` GROUP BY year ORDER BY year	SELECT SUM(sale_dollars) AS ventas_totales, DATE_TRUNC(date, YEAR) AS ano FROM `uvm-casopractico-s16.Caso16.sales` GROUP BY ano ORDER BY ano

Se muestran la sumatoria de las ventas totales por año.

Untitled query

RUNSAVEDOWNLOADSHARESCHEDULEMORE

```
1 SELECT
2 SUM(sale_dollars) AS ventas_totales,
3 DATE_TRUNC(date, YEAR) AS ano
4 FROM `uvm-casopractico-s16.Caso16.sales`
5 GROUP BY ano
6 ORDER BY ano
```

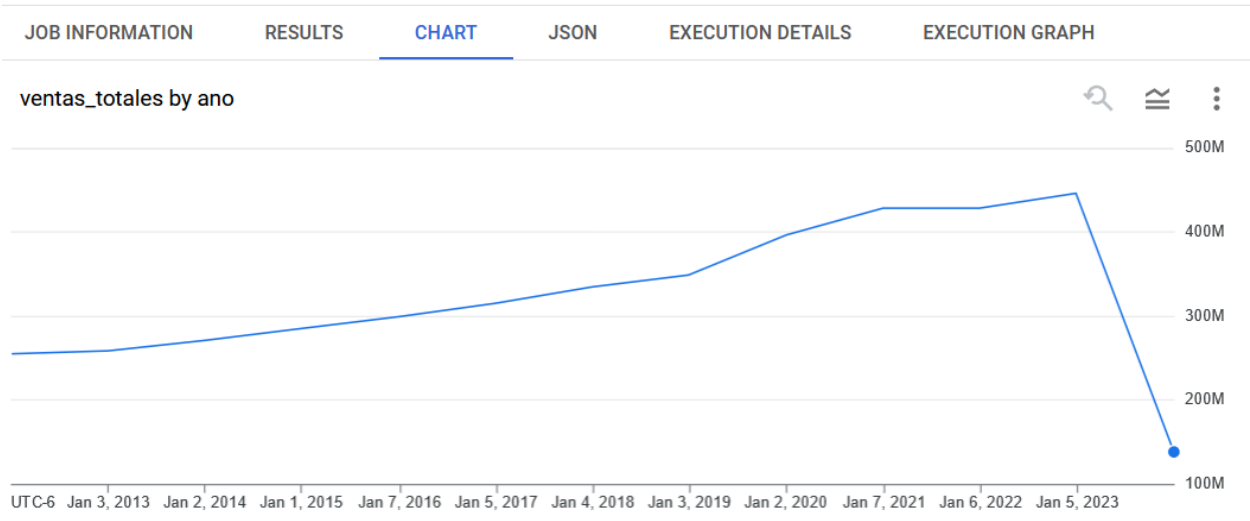
Press Alt+F1 for Accessibility Option

Query results

SARERESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	ventas_totales	ano				
1	255203313.5700...	2012-01-01				
2	258103792.8199...	2013-01-01				
3	270587738.0900...	2014-01-01				
4	284328572.1100...	2015-01-01				
5	299447564.8500...	2016-01-01				
6	314530044.8100...	2017-01-01				
7	334216843.0700...	2018-01-01				
8	349220341.6300...	2019-01-01				
9	396663131.6600...	2020-01-01				
10	428123535.4000...	2021-01-01				

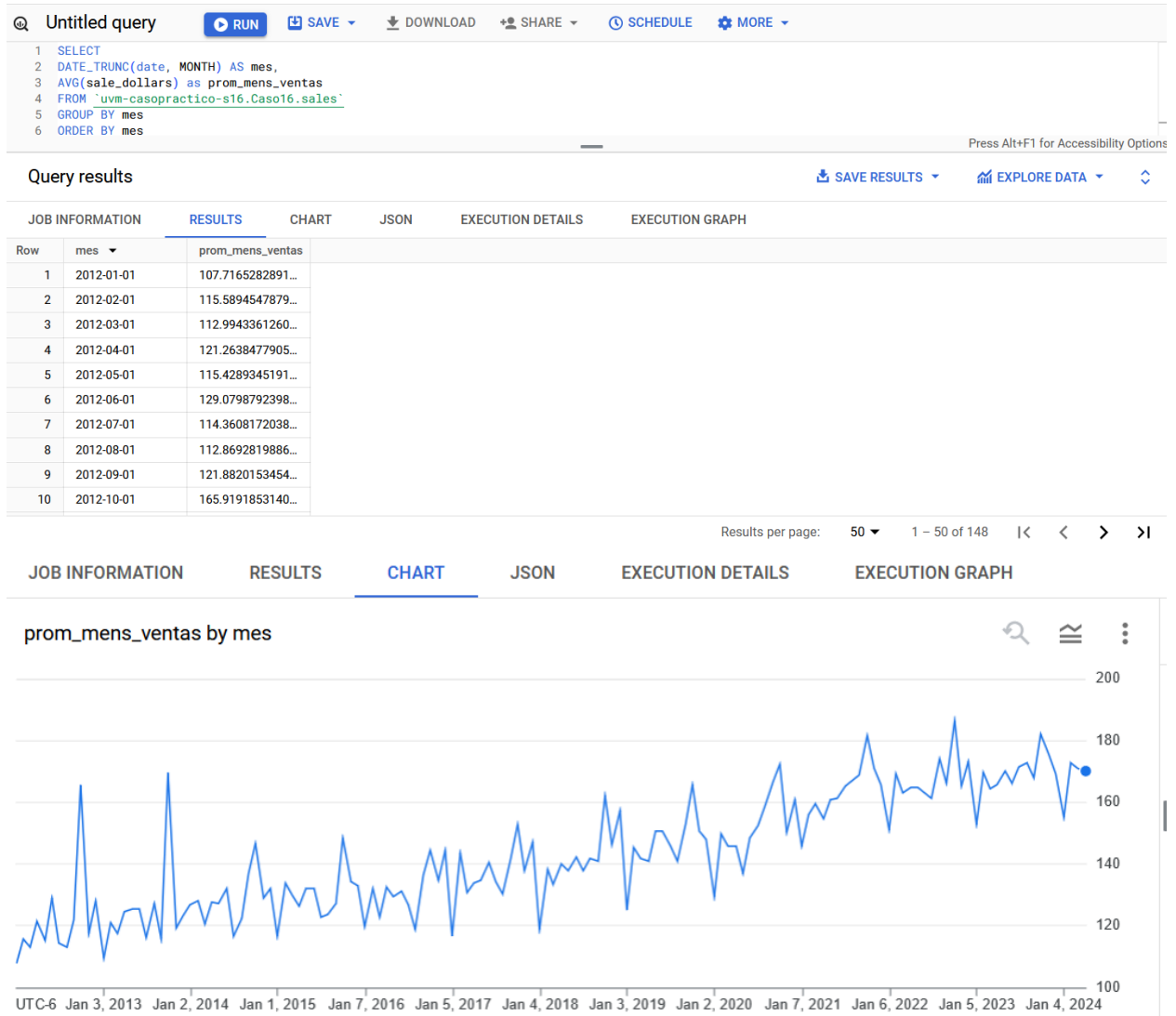
Results per page: 501 – 13 of 13



```
SELECT DATE_TRUNC(date, MONTH) as month,  
AVG(sales) as avg_sales  
FROM `project.dataset.table`  
GROUP BY month  
ORDER BY month
```

```
SELECT  
DATE_TRUNC(date, MONTH) AS mes,  
AVG(sale_dollars) as prom_mens_ventas  
FROM `uvm-casopractico-s16.Caso16.sales`  
GROUP BY mes  
ORDER BY mes
```

Se muestra el promedio de las ventas por mes



Paso 3: Consultas SQL en Python (Sugeridas: 2, pueden ser las del paso anterior)

1. Conexión a BigQuery desde Python:

Crea un archivo Jupyter Notebook en VSCode.

Utiliza la biblioteca google-cloud-bigquery para conectarte a BigQuery y ejecutar tus consultas.

- a) Cargamos localmente nuevo set de datos “Jugadores de Basketball” que consta de 4 tablas

```
import pandas as pd
df1 = pd.read_csv('players.csv')
df2 = pd.read_csv('players_teams.csv')
df3 = pd.read_csv('teams.csv')
df4 = pd.read_csv('awards_players.csv')
```

- b) Creamos nuevo proyecto y cuenta de servicio

Service accounts for project "UVM-CasoPractico-S16"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts.](#)

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies.](#)

Filter Enter property name or value									?	⋮
<input type="checkbox"/>	Email	Status	Name ↑	Description	Key ID	Key creation date	OAuth 2 Client ID	?	Actions	
<input type="checkbox"/>	uvm-casopractico-s16@uvm-casopractico-s16.iam.gserviceaccount.com	Enabled	UVM-CasoPractico-S16	UVM-CasoPractico-S16	7619a96886b9006a469b85a67826381ec6acf96a	May 30, 2024	1034653802437377		⋮	

- c) Creamos conexión y subimos los archivos locales a BigQuery

```
# Objetivo: Vamos a crear la conexión con BigQuery y vamos a leer mediante consulta SQL desde Python.
# Generamos conexión a BQ
from google.oauth2 import service_account # para generar conexión
bq_cred = service_account.Credentials.from_service_account_file('uvm-casopractico-s16-7619a96886b9.json')

# Enviamos el df a BQ - Podemos crear o reemplazar una tabla
import pandas_gbq # para interactuar con BQ
pandas_gbq.to_gbq(df1, 'Caso16b.players', project_id= 'uvm-casopractico-s16', if_exists= 'replace', credentials = bq_cred)

pandas_gbq.to_gbq(df2, 'Caso16b.players_teams', project_id= 'uvm-casopractico-s16', if_exists= 'replace', credentials = bq_cred)

pandas_gbq.to_gbq(df3, 'Caso16b.teams', project_id= 'uvm-casopractico-s16', if_exists= 'replace', credentials = bq_cred)

pandas_gbq.to_gbq(df4, 'Caso16b.awards_players', project_id= 'uvm-casopractico-s16', if_exists= 'replace', credentials = bq_cred)
```

d) Validamos datos migrados en BigQuery

type to search

Viewing resources.
SHOW STARRED ONLY

▼

uvm-casopractico-s16

☆

⋮

▶

🔍

Queries

⋮

▶

📓

Notebooks

⋮

▶

📄

Data canvases

⋮

▶

🔗

External connections

⋮

▶

📁

Caso16

☆

⋮

▶

📁

Caso16b

☆

⋮

▶

📄

awards_players

☆

⋮

▶

📄

players

☆

⋮

▶

📄

players_teams

☆

⋮

▶

📄

teams

☆

⋮

SUMMARY

^

players

QUERY

SHARE

COPY

SNAPSHOT

DELETE

EXPORT

REFRESH

SCHEMA

DETAILS

PREVIEW

LINEAGE

DATA PROFILE

DATA QUALITY

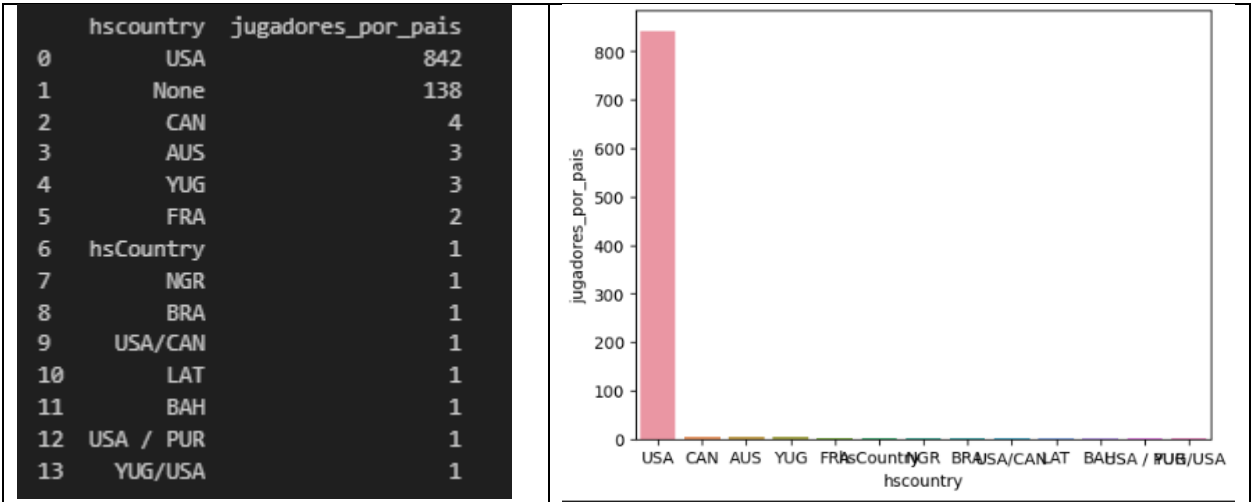
Row	playerID	useFirst	firstName	middleName	lastName	nameGiven
1	anthopa01	null	Paul	null	Anthony	null
2	cablesa01	null	Howard	null	Cable	null
3	chanejo01	John	John	Louie	Chaney	null
4	cottobo01	null	Bob	null	Cotton	null
5	calihbo01	null	Robert	J.	Calihan	null
6	bovento01	Don	Donald	E.	Boven	null
7	darrojo01	Jimmy	James	K.	Darrow	null
8	birchpa01	Paul	Paul	null	Birch	null
9	bryanjo01	Joe	Joseph	Washington	Bryant	null
10	bioID	bioID	firstName	middleName	lastName	nameGiven
11	adubari01	Richie	Richie	null	Adubato	null
12	albecst01	Stan	Stan	null	Albeck	null
13	anderla01	LaDell	LaDell	null	Andersen	null
14	ankerja01	Jack	Jack	null	Ankerson	null
15	ashbi01	Bill	William	null	Ash	null
16	auebrea01	Red	Red	null	Auerbach	null

Results per page: 50 1 - 50 of 1000 |< < > >|

Paso 2: Consultas SQL en BigQuery (Sugeridas: 2)

A continuación, se presentan 4 ejemplos de consultas a ejecutar en BigQuery:

a) Consulta de datos agregados	Contar jugadores únicos por país
<pre>SELECT category, COUNT(*) as total_category FROM `project.dataset.table` GROUP BY category ORDER BY total_category DESC</pre>	<pre>sql = """ SELECT hscountry, COUNT(DISTINCT playerID) as jugadores_por_pais FROM `uvm-casopractico- s16.Caso16b.players` GROUP BY hscountry ORDER BY jugadores_por_pais DESC """ df_bq = pd.read_gbq(sql, project_id='uvm-casopractico-s16', credentials = bq_cred, dialect='standard') #Dialect standard: para usar BigQuery's standard SQL dialect print(df_bq)</pre>



Se muestran los jugadores por país en la Base de datos, identificando que Estados Unidos es el que tienen más jugadores.

b) Consulta de Datos con Filtro:	Jugadores con mas de 2000 puntos																																												
SELECT name, number FROM `project.dataset.table` WHERE number > 100.0	<pre>sql = "" SELECT tmID, playerid, MAX(points) AS MAYOR_PUNTAJE FROM `uvm-casopractico- s16.Caso16b.players_teams` WHERE points >= 2000 GROUP BY tmID, playerID ORDER BY MAYOR_PUNTAJE DESC "" df_bq = pd.read_gbq(sql, project_id='uvm- casopractico-s16', credentials = bq_cred, dialect='standard') #Dialect standard: para usar BigQuery's standard SQL dialect print(df_bq)</pre>																																												
<table><tr><th></th><th>tmID</th><th>playerid</th><th>MAYOR_PUNTAJE</th></tr><tr><td>0</td><td>MIL</td><td>abdulka01</td><td>2822</td></tr><tr><td>1</td><td>KCO</td><td>archina01</td><td>2719</td></tr><tr><td>2</td><td>WAS</td><td>arenagi01</td><td>2346</td></tr><tr><td>3</td><td>DAL</td><td>aguirma01</td><td>2330</td></tr><tr><td>4</td><td>LAL</td><td>abdulka01</td><td>2275</td></tr><tr><td>5</td><td>CIN</td><td>archina01</td><td>2145</td></tr><tr><td>6</td><td>DEN</td><td>anthoca01</td><td>2122</td></tr></table>		tmID	playerid	MAYOR_PUNTAJE	0	MIL	abdulka01	2822	1	KCO	archina01	2719	2	WAS	arenagi01	2346	3	DAL	aguirma01	2330	4	LAL	abdulka01	2275	5	CIN	archina01	2145	6	DEN	anthoca01	2122	<table><tr><th>playerid</th><th>MAYOR_PUNTAJE</th></tr><tr><td>abdulka01</td><td>2822</td></tr><tr><td>archina01</td><td>2719</td></tr><tr><td>arenagi01</td><td>2346</td></tr><tr><td>aguirma01</td><td>2330</td></tr><tr><td>anthoca01</td><td>2122</td></tr></table>	playerid	MAYOR_PUNTAJE	abdulka01	2822	archina01	2719	arenagi01	2346	aguirma01	2330	anthoca01	2122
	tmID	playerid	MAYOR_PUNTAJE																																										
0	MIL	abdulka01	2822																																										
1	KCO	archina01	2719																																										
2	WAS	arenagi01	2346																																										
3	DAL	aguirma01	2330																																										
4	LAL	abdulka01	2275																																										
5	CIN	archina01	2145																																										
6	DEN	anthoca01	2122																																										
playerid	MAYOR_PUNTAJE																																												
abdulka01	2822																																												
archina01	2719																																												
arenagi01	2346																																												
aguirma01	2330																																												
anthoca01	2122																																												

Abdulka01 del equipo MIL es el jugador con mayor puntaje en la Base de datos.

c) Consulta de Tendencias a lo Largo del Tiempo:		Jugadores con más de un premio por año	
<pre>SELECT year, number FROM `project.dataset.table` GROUP BY year ORDER BY year</pre>		<pre>sql = """ SELECT playerID, year, COUNT(playerID) AS MVP FROM `uvm-casopractico- s16.Caso16b.awards_players` GROUP BY playerID, year HAVING COUNT(playerID) > 1 ORDER BY year DESC; """ df_bq = pd.read_gbq(sql, project_id='uvm-casopractico-s16', credentials = bq_cred, dialect='standard') #Dialect standard: para usar BigQuery's standard SQL dialect print(df_bq)</pre>	
<pre> playerID year MVP 0 jamesle01 2011 4 1 irvinky01 2011 2 2 chandty01 2011 3 3 howardw01 2011 2 4 bryanko01 2011 2 171 chambwi01 1965 2 172 lucasje01 1963 2 173 chambwi01 1959 3 174 bayloel01 1958 2 175 cousybo01 1956 2 [176 rows x 3 columns]</pre>			

Mostramos los jugadores con más premios agrupados por año.

Premios por año

```
sql = """
SELECT year, COUNT(playerID) AS MVP
FROM `uvm-casopractico-
s16.Caso16b.awards_players`
GROUP BY year
ORDER BY year DESC;
"""

df_bq = pd.read_gbq(sql,
project_id='uvm-casopractico-s16',
credentials = bq_cred,
dialect='standard') #Dialect standard:
para usar BigQuery's standard SQL
dialect
print(df_bq)
```

```
...      year  MVP
0      2011    30
1      2010    27
2      2009    30
3      2008    27
4      2007    25
..      ...    ...
67     1942     1
68     1941     2
69     1939     1
70     1938     1
71     1937     2

[72 rows x 2 columns]
```

```
# Para una serie temporal
import matplotlib.pyplot as plt
# Asegúrate de que 'year' y 'number' están en el formato correcto
# df['year'] debe ser de tipo datetime, para asegurarlos
df_bq['year'] = pd.to_datetime(df_bq['year'], format='%Y')
# Asegúrate de que los datos están ordenados por 'year'
df_bq = df_bq.sort_values('year')
# Crea la figura y el eje
plt.figure(figsize=(10, 5))
plt.plot(df_bq['year'], df_bq['MVP'], marker='', color='blue', linewidth=2)
# Título y etiquetas
plt.title('Serie de Tiempo de JAMES a través del tiempo')
plt.xlabel('year')
plt.ylabel('MVP')
# Mostrar la gráfica
plt.show()
```



d) Consulta de Datos Relacionales (Join):	Lista de Jugadores por Equipo
SELECT a.name, b.purchase_date, b.amount FROM `project.dataset.customers` a JOIN `project.dataset.purchases` b ON a.customer_id = b.customer_id	<pre> sql = """ SELECT DISTINCT t.name, p.firstName, p.lastName FROM `uvm-casopractico- s16.Caso16b.players` p JOIN `uvm-casopractico- s16.Caso16b.players_teams` pt ON p.playerID = pt.playerID JOIN `uvm-casopractico- s16.Caso16b.teams` t ON pt.lgID = t.lgID ORDER BY t.name,p.lastName ; """ df_bq = pd.read_gbq(sql, project_id='uvm-casopractico-s16', credentials = bq_cred, dialect='standard') #Dialect standard: para usar BigQuery's standard SQL dialect print(df_bq) </pre>
<pre> ... name firstName lastName 0 Akron Firestone Non-Skids John Abramovic 1 Akron Firestone Non-Skids Howard Adams 2 Akron Firestone Non-Skids Raymond Adams 3 Akron Firestone Non-Skids Glenn Adams 4 Akron Firestone Non-Skids Willie Adams ... 11289 Youngstown Bears Carl Baer 11290 Youngstown Bears Frank Baird 11291 Youngstown Bears Bob Baker 11292 Youngstown Bears Art Bakeraitis 11293 Youngstown Bears Herbert Ball [11294 rows x 3 columns] </pre>	

Se muestra la lista de jugadores por equipo.