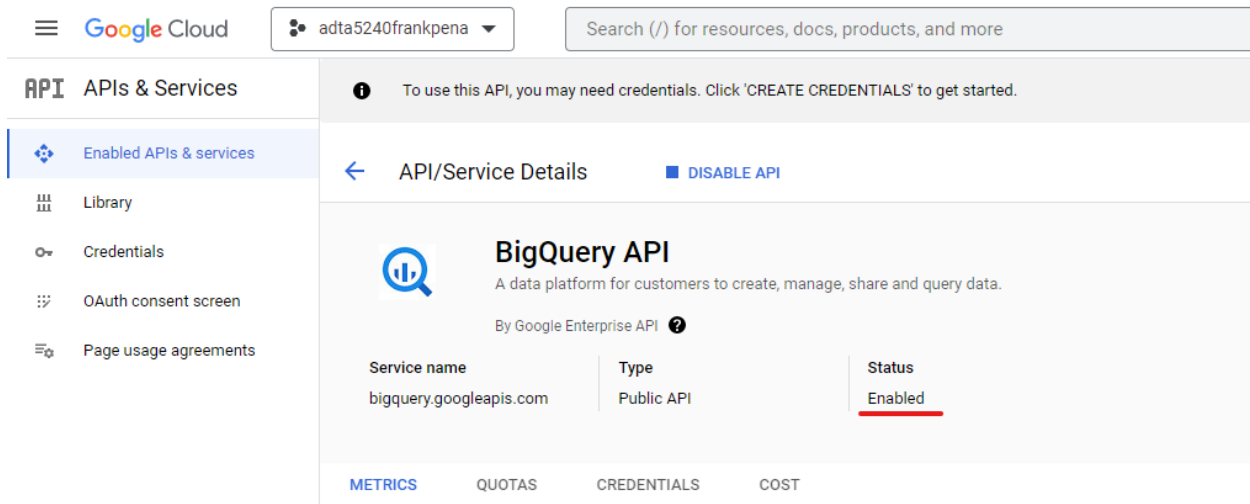
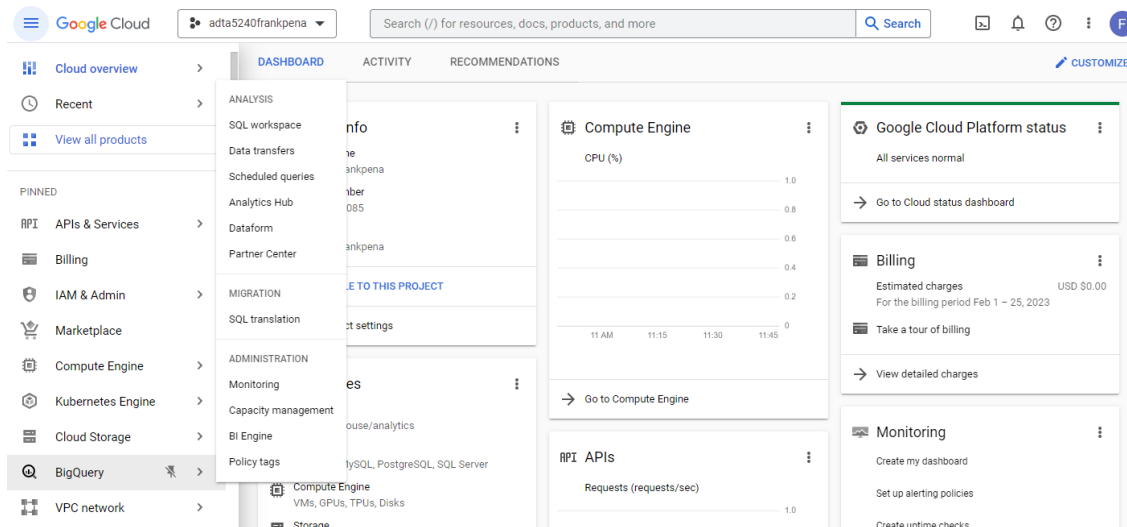


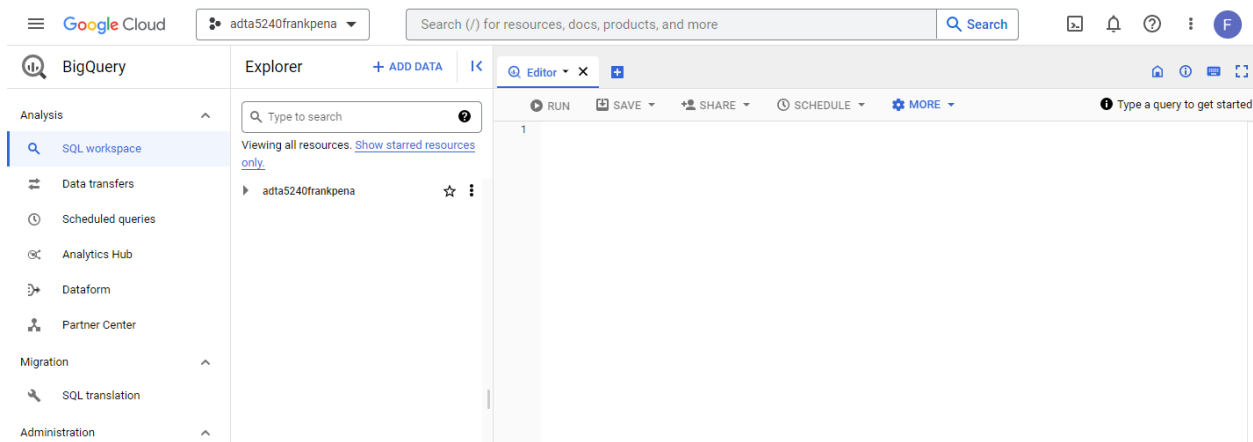
- Go to your GCP console and go to BigQuery.
 - From the GCP console, navigated to BigQuery API and verified the API was enabled.



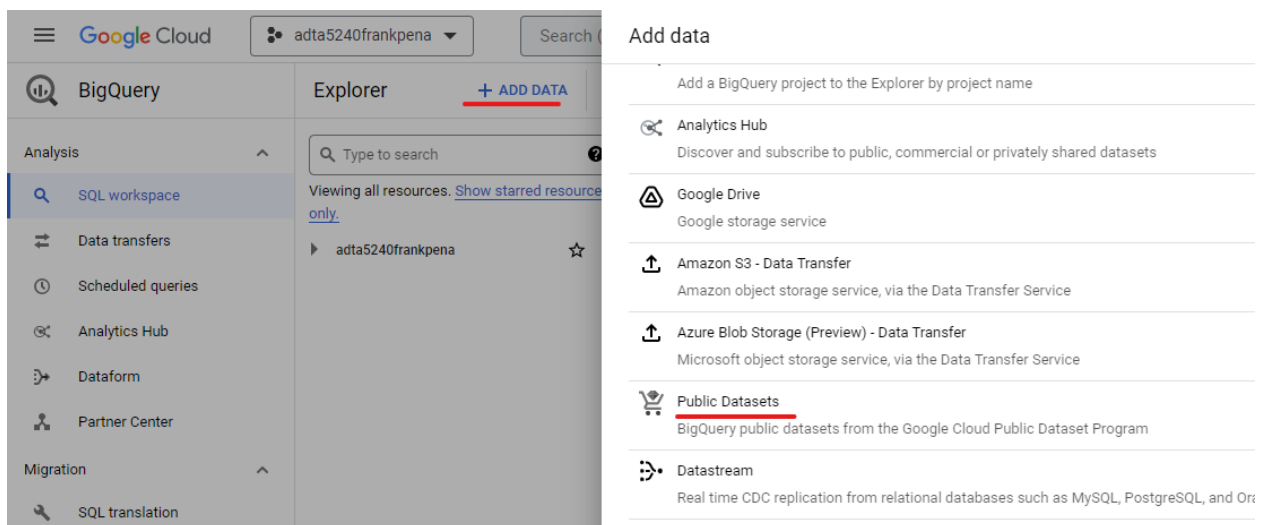
- Returned to the GCP console and navigated to BigQuery in the panel on the left side of the screen then clicked on BigQuery.
-



- The link took me to BigQuery where the data will reside for the NCAA Basketball dataset.



- You will use the NCAA Basketball Dataset. You have already worked with the public datasets in a previous assignment. If you forget how to get access, please revisit the previous homework. Go to the mbb_pbp_sr dataset to see the schema of this predefined table.
 - Clicked on 'Add Data' then a panel opened from the right side of the screen. Clicked on Public Datasets.



- Typed NCAA Basketball in the search box on the top of the screen and there was 1 result shown for NCAA Basketball.

Marketplace > "ncaa basketball" > Datasets

Filter Type to filter

1 result

Category



Analytics

(1)

Encyclopedic

(1)



NCAA Basketball

NCAA

This dataset contains data about NCAA Basketball games, teams, and players. Game data covers play-by-play and box scores back to 2009, as well as final scores back to 1996. Additional data about wins and losses goes back to the 1894-1895 season for some teams. All data runs through the end of the 2017-2018 season.

Type

- Clicked on the result that was displayed then clicked on the 'View Dataset' button.

[←](#) Product details

NCAA Basketball

[NCAA](#)

Statistics for NCAA men's and women's basketball from 1894-2018

[VIEW DATASET](#) [OVERVIEW](#)[SAMPLES](#)

Overview

This dataset contains data about NCAA Basketball games, teams, and players. Game data covers play-by-play and box scores back to 2009, as well as final scores back to 1996. Additional data about wins and losses goes back to the 1894-1895 season for some teams. All data runs through the end of the 2017-2018 season.

Additional details

Type: [Datasets](#)Category: [Analytics](#), [Encyclopedic](#)Dataset source: [NCAA](#) , [Sportradar](#)

Cloud service: BigQuery

Expected update frequency: Historical

- The next window opened displaying the list of datasets. Clicked on the triangle pointing to the right and it opened a list of tables within the dataset. The 'mbb_php_sr' table is the table to be used.

Dataset info

Dataset ID: bigquery-public-data.ncaa_basketball

Created: Jan 23, 2018, 2:25:09 PM UTC-6

Default table expiration: Never

Last modified: Sep 20, 2022, 2:44:16 AM UTC-5

Data location: US

Description: This dataset contains data about NCAA Basketball games, teams, and players. Game data covers play-by-play and box scores back to 2009, as well as final scores back to 1996. Additional data about wins and losses goes back to the 1994-5 season in some teams' cases.

Sportradar: Copyright Sportradar LLC. Access to data is intended solely for internal research and testing purposes, and is not to be used for any business or commercial purpose. Data are not to be exploited in any manner without express approval from Sportradar.

NCAA®: Copyright National Collegiate Athletic Association. Access to data is provided solely for internal research and testing purposes, and may not be used for any business or commercial purpose. Data are not to be exploited in any manner without express approval from the National Collegiate Athletic Association.

Default collation: Labels: Tags: ⚠️

○ dfvdfdvdfv

- Open the mbb_pbp_sr dataset to see the schema of this predefined table. “This table has play-by-play information of all men’s basketball games in the 2013– 2014 season, and each row in the table represents a single event in a game.” Click on preview to see more information about the dataset. How many rows are in the dataset?

○ Clicked on the mbb_pbp_sr table and the schema of the table appeared.

mbb_pbp_sr

SCHEMA DETAILS PREVIEW LINEAGE PREVIEW

Filter: Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/> game_id	STRING	NULLABLE				Unique identifier for the game
<input type="checkbox"/> load_timestamp	TIMESTAMP	NULLABLE				Time at which the data was loaded into the table
<input type="checkbox"/> season	INTEGER	NULLABLE				Season the game was played in
<input type="checkbox"/> status	STRING	NULLABLE				
<input type="checkbox"/> scheduled_date	TIMESTAMP	NULLABLE				Date the game was played
<input type="checkbox"/> venue_id	STRING	NULLABLE				Unique identifier for the venue where the game was played
<input type="checkbox"/> venue_name	STRING	NULLABLE				Name of the venue where the game was played
<input type="checkbox"/> venue_city	STRING	NULLABLE				City where the game was played
<input type="checkbox"/> venue_state	STRING	NULLABLE				State where the game was played
<input type="checkbox"/> venue_address	STRING	NULLABLE				Address of the venue where the game was played
<input type="checkbox"/> venue_zip	STRING	NULLABLE				Zip code of the venue where the game was played
<input type="checkbox"/> venue_country	STRING	NULLABLE				Country where the game was played
<input type="checkbox"/> venue_capacity	INTEGER	NULLABLE				Current capacity of the venue where the

EDIT SCHEMA VIEW ROW ACCESS POLICIES

- Clicked on the PREVIEW button then the first 50 records of 4160393 total records in the dataset were displayed.

Row	game_id	load_timestamp	season	status	scheduled_date	venue_id	venue_name	venue_city
1	3d90529c-12ec-40e0-8731-d5e...	2018-02-19 06:29:20.589864 U...	2017	closed	2018-02-18 21:00:00 UTC	null	null	null
2	5c74a296-4631-4646-bc66-97...	2018-02-19 06:29:28.153849 U...	2017	closed	2018-02-18 20:30:00 UTC	null	null	null
3	ceef5306-1a42-4c28-a09e-885...	2018-02-19 06:28:43.866317 U...	2017	closed	2018-02-18 23:00:00 UTC	null	null	null
4	d4213cb8-f351-4864-a3e0-71b...	2018-02-19 06:28:24.037998 U...	2017	closed	2018-02-19 01:00:00 UTC	null	null	null
5	d4213cb8-f351-4864-a3e0-71b...	2018-02-19 06:28:24.037998 U...	2017	closed	2018-02-19 01:00:00 UTC	null	null	null
6	cc581cf9-11ff-48bb-8e76-618d...	2018-02-19 06:28:08.586975 U...	2017	closed	2018-02-18 20:00:00 UTC	null	null	null
7	3ea4e171-afee-4964-9412-306...	2018-02-19 06:28:05.996438 U...	2017	closed	2018-02-18 20:00:00 UTC	null	null	null
8	act32ca-4689-4eea-a511-e7ab...	2018-02-19 06:29:43.741550 U...	2017	closed	2018-02-18 18:00:00 UTC	null	null	null
9	d4213cb8-f351-4864-a3e0-71b...	2018-02-19 06:28:24.037998 U...	2017	closed	2018-02-19 01:00:00 UTC	null	null	null

- You will now use SQL to query the dataset. Selected the following columns from the table: game_clock, points_scored, team_name, event_description, and timestamp.
 - To create a query, clicked on the dropdown for Query and clicked on 'In new tab'.

Row	game_id	load_timestamp	season	status	scheduled_date	venue_id	venue_name	venue_city
1	3d90529c-12ec-40e0-8731-d5e...	2018-02-19 06:29:20.589864 U...	2017	closed	2018-02-18 21:00:00 UTC	null	null	null

- In the new tab, typed the following query- 'SELECT game_clock, points_scored, team_name, event_description

FROM `bigquery-public-data.ncaa_basketball.mbb_pbp_sr`

WHERE season = 2014

AND home_name = 'Wildcats'

AND away_name = 'Fighting Irish'

AND points_scored IS NOT NULL

ORDER BY timestamp DESC

LIMIT 10;' The query should return the time on the game clock, how many points were scored, the name of the team that scored, and the description of how the point was scored. The other criteria for the game was where the home team was called Wildcats, the away team was called Fighting Irish, there was a point scored, and it will return only 10 records.

The screenshot shows the Google Cloud BigQuery interface. The Explorer on the left lists various datasets. The main editor displays a SQL query:


```
1 SELECT game_clock, points_scored, team_name, event_description FROM `bigquery-public-data.ncaa_basketball.mbb_pbp_sr`
2 WHERE season = 2014
3 AND home_name = 'Wildcats'
4 AND away_name = 'Fighting Irish'
5 AND points_scored IS NOT NULL
6 ORDER BY timestamp DESC
7 LIMIT 10;
```

 The 'RUN' button is highlighted in blue. A status message at the top right indicates: 'This query will process 391.07 MB when run.'

- Clicked 'Run' then the results were displayed from the query.

The screenshot shows the Google Cloud BigQuery interface with the query results displayed. The 'Query results' tab is active, showing a table with 10 rows. The 'RUN' button is now greyed out, and a status message at the top right indicates: 'Query completed.'

Row	game_clock	points_scored	team_name	event_description
1	00:06	1.0	Wildcats	Andrew Harrison makes free th...
2	00:06	1.0	Wildcats	Andrew Harrison makes free th...
3	1:12	2.0	Wildcats	Karl-Anthony Towns makes tw...
4	2:34	3.0	Fighting Irish	Jerian Grant makes three point jump shot (Pat Connaughton assists)...
5	3:15	3.0	Wildcats	Aaron Harrison makes three point jump shot (Tyler Uilis assists)...
6	3:45	1.0	Fighting Irish	Pat Connaughton makes free t...
7	4:08	1.0	Wildcats	Karl-Anthony Towns makes fre...
8	4:08	2.0	Wildcats	Karl-Anthony Towns makes tw...
9	4:28	1.0	Fighting Irish	Jerian Grant makes free throw ...
10	5:05	2.0	Wildcats	Aaron Harrison makes two poi...

- Let's change the SQL code so the query includes a cumulative sum of scores for each team throughout the game. This can be done using analytic (window) functions. Analytic functions computes the aggregates for each row over a group of rows defined by a window whereas aggregate functions compute a single aggregate value over a group of rows.

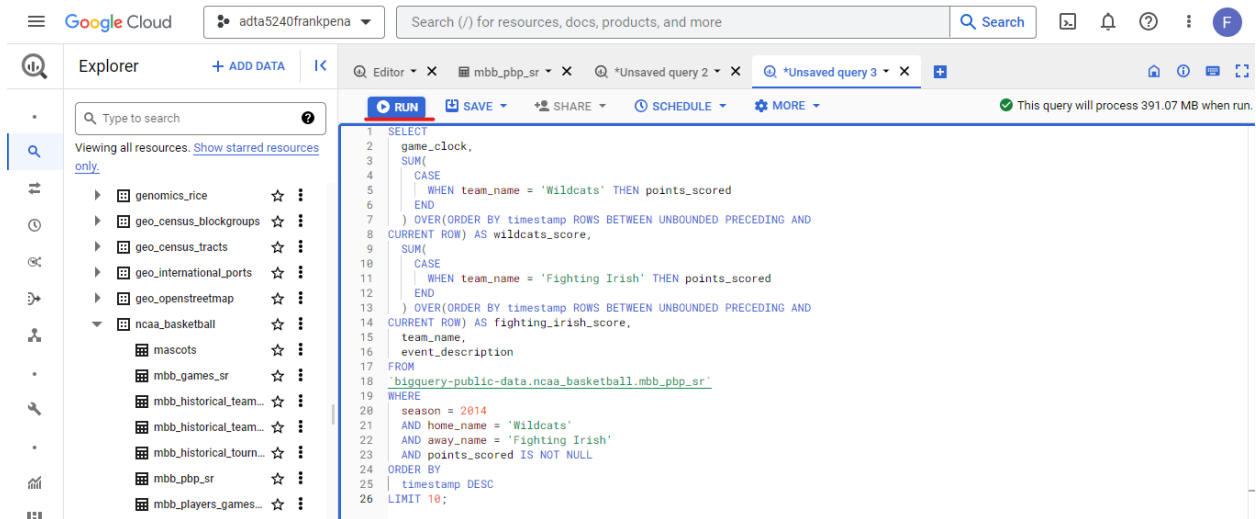
- To create a new query in another tab, clicked on the 'blue +'.

The screenshot shows the Google Cloud BigQuery interface with a new query tab being created. The 'Compose new query' button is highlighted in red. The main editor displays the same SQL query as before. The 'Query results' tab is active, showing the same table of results.

- Typed the query below in the editor.

```
SELECT
  game_clock,
  SUM(
    CASE
      WHEN team_name = 'Wildcats' THEN points_scored
    END
  ) OVER(ORDER BY timestamp ROWS BETWEEN UNBOUNDED PRECEDING AND
CURRENT ROW) AS wildcats_score,
  SUM(
    CASE
      WHEN team_name = 'Fighting Irish' THEN points_scored
    END
  ) OVER(ORDER BY timestamp ROWS BETWEEN UNBOUNDED PRECEDING AND
CURRENT ROW) AS fighting_irish_score,
  team_name,
  event_description
FROM
`bigquery-public-data.ncaa_basketball.mbb_pbp_sr`
WHERE
  season = 2014
  AND home_name = 'Wildcats'
  AND away_name = 'Fighting Irish'
  AND points_scored IS NOT NULL
ORDER BY
  timestamp DESC
LIMIT 10;
```

- Clicked on the RUN button.



- The results of the query was displayed and the data showed the time left on the clock, the score of the wildcats, the score of the Fighting Irish, team name of the team that scored, and the event description which displays the name of the person that scored and how they scored.

Google Cloud

adta5240frankpena

Search (/) for resources, docs, products, and more

Search

Explorer

+ ADD DATA

K

Type to search

Viewing all resources. [Show starred resources only.](#)

genomics_rice

geo_census_blockgroups

geo_census_tracts

geo_international_ports

geo_openstreetmap

ncaa_basketball

mbb_games_sr

mbb_historical_team...

mbb_historical_tourn...

mbb_players_games...

mbb_teams

mbb_teams_games_sr

team_colors

SHOW MORE

Editor

mbb_pbp_sr

*Unsaved query 2

*Unsaved query 3

Query completed.

3

4

5

6

7

8

9

10

11

```
SELECT SUM(
  CASE
    WHEN team_name = 'Wildcats' THEN points_scored
  END
) OVER(ORDER BY timestamp ROWS BETWEEN UNBOUNDED PRECEDING AND
CURRENT ROW) AS wildcats_score,
SUM(
  CASE
    WHEN team_name = 'Fighting Irish' THEN points_scored
```

Query results

SAVE RESULTS

EXPLORE DATA

Press Alt+F1 for Accessibility Options.

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

EXECUTION GRAPH

PREVIEW

Row	game_clock	wildcats_score	fighting_irish_sc	team_name	event_description
1	00:06	68.0	66.0	Wildcats	Andrew Harrison makes free throw 2 of 2
2	00:06	67.0	66.0	Wildcats	Andrew Harrison makes free throw 1 of 2
3	1:12	66.0	66.0	Wildcats	Karl-Anthony Towns makes two point jump shot
4	2:34	64.0	66.0	Fighting Irish	Jerian Grant makes three point jump shot (Pat Connaughton assists)...
5	3:15	64.0	63.0	Wildcats	Aaron Harrison makes three point jump shot (Tyler Ullis assists)...
6	3:45	61.0	63.0	Fighting Irish	Pat Connaughton makes free throw 1 of 2
7	4:08	61.0	62.0	Wildcats	Karl-Anthony Towns makes free throw 1 of 1
8	4:08	60.0	62.0	Wildcats	Karl-Anthony Towns makes two point jump shot
9	4:28	58.0	62.0	Fighting Irish	Jerian Grant makes free throw 1 of 2
10	5:05	58.0	61.0	Wildcats	Aaron Harrison makes two point dunk