

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

Dear Professor & TA,

As I have not received credit yet, I have done all this activity in the GCP account created using IIT M mail id with my credit card.

Hope that will be fine. I have executed all the instructions and commands in that GCP instance and shared the screenshot for the same. Thanks.

Regards,

Aloy Banerjee

CH22M503

Note: All the results are exactly matches with below information,

1. Number of records: 83434866 instead of 24369201 for the first query
2. Number of records: 26441016 instead of 7,334,890 for the second query
3. Roughly 32% instead of 30% of London bikeshare rides lasted 20 minutes or longer
4. Number of records: 954 instead of 880 for the third query
5. Number of records for london1: 955 rows instead of 881
6. Number of records for london2: 959 rows instead of 883
7. In the final results: We will receive 629 rows as result-set instead of 28

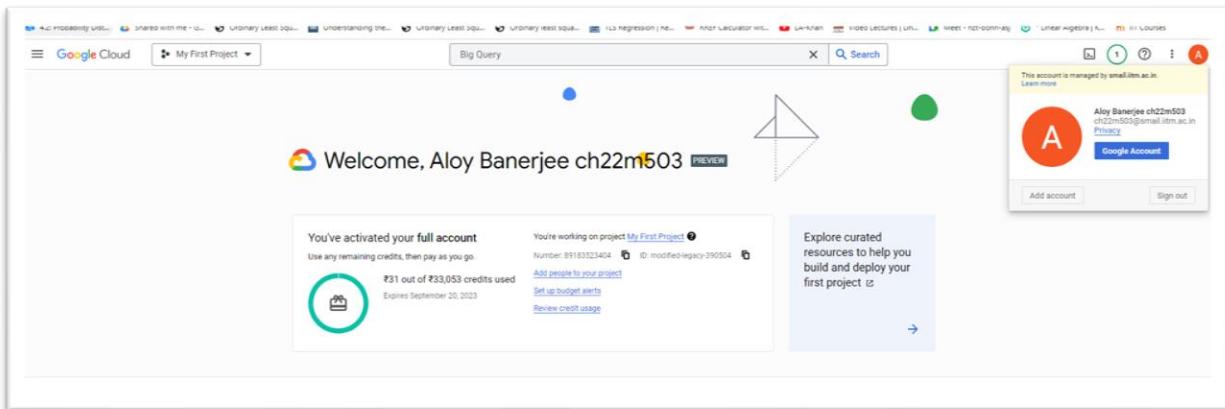
Big Query Soccer Data Analysis: -

Showing the credentials:

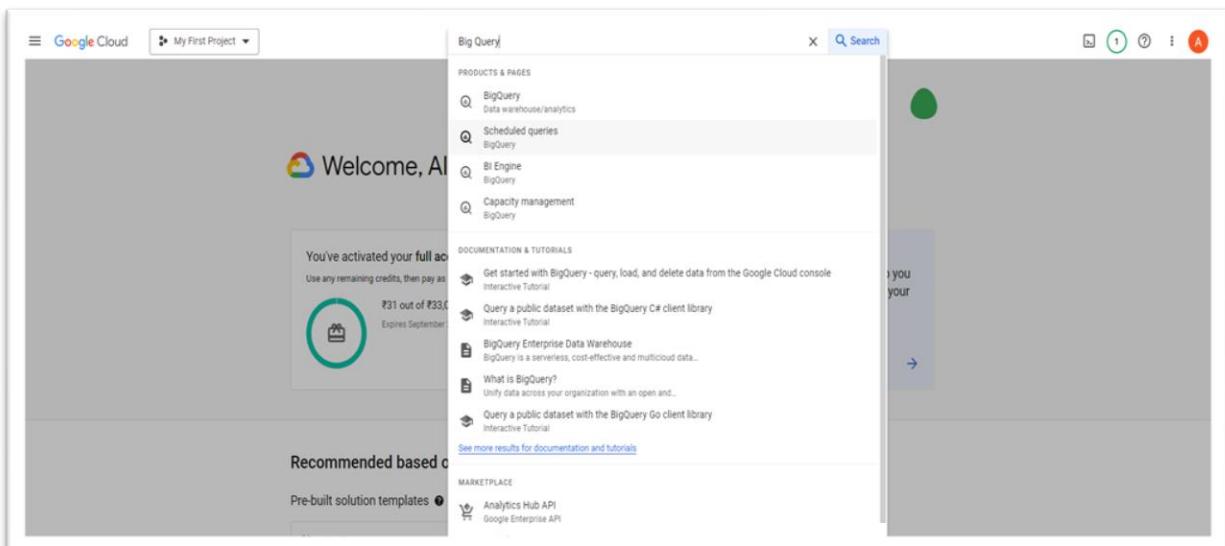
AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



Search for Big Query:

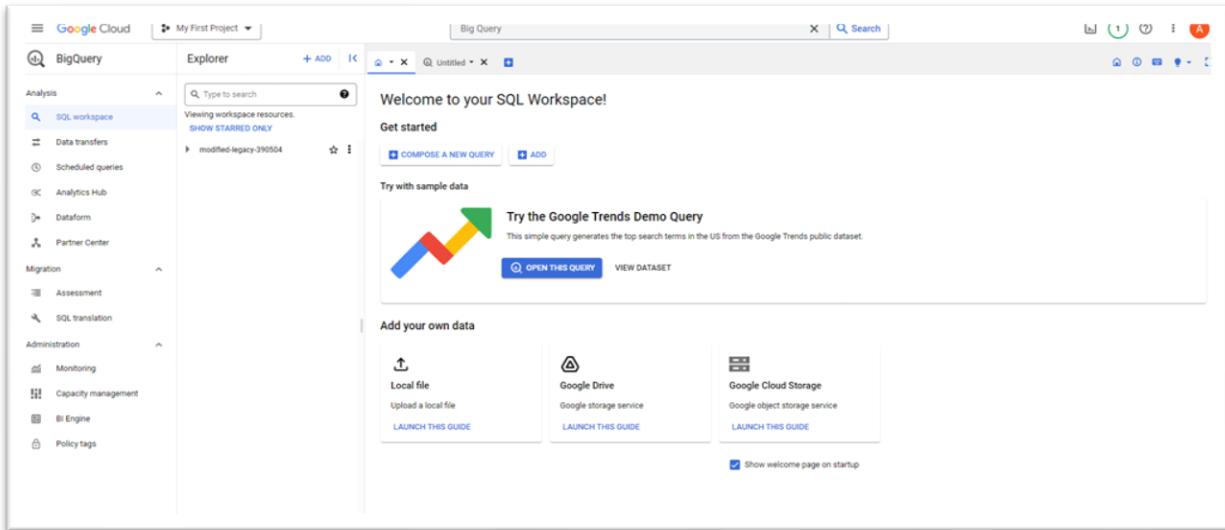


End up in a Big Query console:

AI Lab 2 (Assignment – 2)

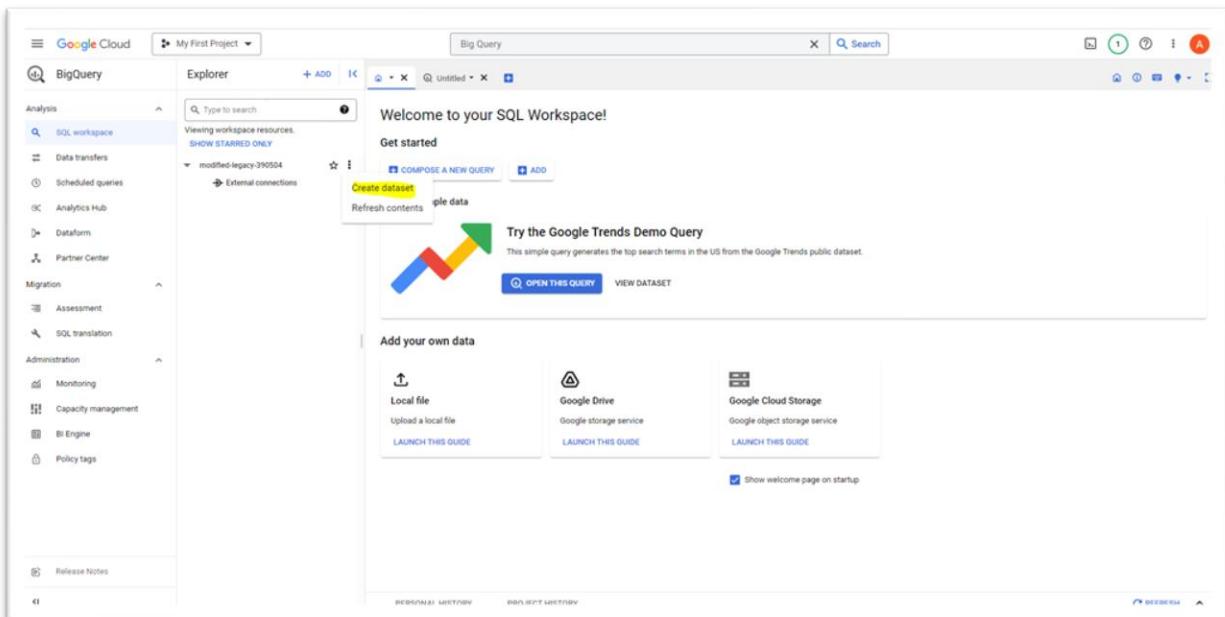
Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



BigQuery Soccer Data Ingestion: As a part of the Big Query Soccer Data Analysis, we must complete the data ingestion pre-requisite.

Create Dataset:



Fill the dataset creation pane:

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google BigQuery interface within the Google Cloud Platform. On the left, the navigation sidebar includes sections like Analysis, Data transfers, Scheduled queries, Analytics Hub, Dataform, Partner Center, Migration, Assessment, SQL translation, Administration, Monitoring, Capacity management, BI Engine, and Policy tags. The main workspace shows a 'Welcome to your SQL Workspace!' message with a 'Try with sample data' section featuring a 'Google Trends Demo Query'. Below this are sections for 'Add your own data' via Local file, Google Drive, or Google Cloud Storage, and 'PERSONAL HISTORY' and 'PROJECT HISTORY' buttons at the bottom.

Create dataset

Project ID: modified-legacy-390504

Dataset ID:

Location type: Multi-region (Allow BigQuery to colocate your datasets with other GCP services.)

Multi-region: US (multiple regions in United States)

Default table expiration: Enable table expiration

Default maximum table age: Days

Advanced options

Encryption: Google-managed encryption key (no configuration required)

Custom-managed encryption key (CMK) Manage in Google Cloud Key Management Service

Case insensitive: Enable case insensitive table names

Default Collation: Enable default collation

Default Collation: Latin1_General_CI_AS

CREATE DATASET CANCEL

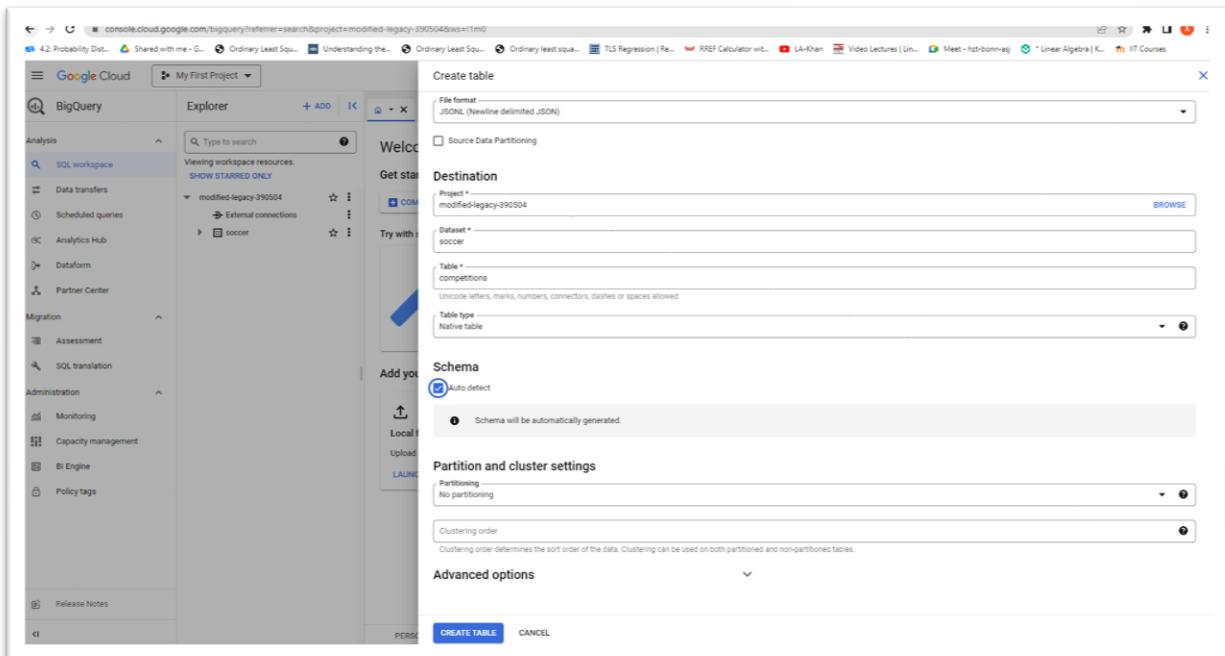
Click on the CREATE DATASET: Dataset will be created.

Click on the Create table button,

The screenshot shows the same Google BigQuery interface. A context menu is open over a dataset named 'soccer'. The menu options include Open, Open in, Refresh contents, Get link, Create table (which is highlighted with a yellow box), Share, Copy ID, and Delete. The main workspace shows the 'Welcome to your SQL Workspace!' message and the 'Try with sample data' section. A notification at the bottom states "'soccer' created." and has a 'GO TO DATASET' button.

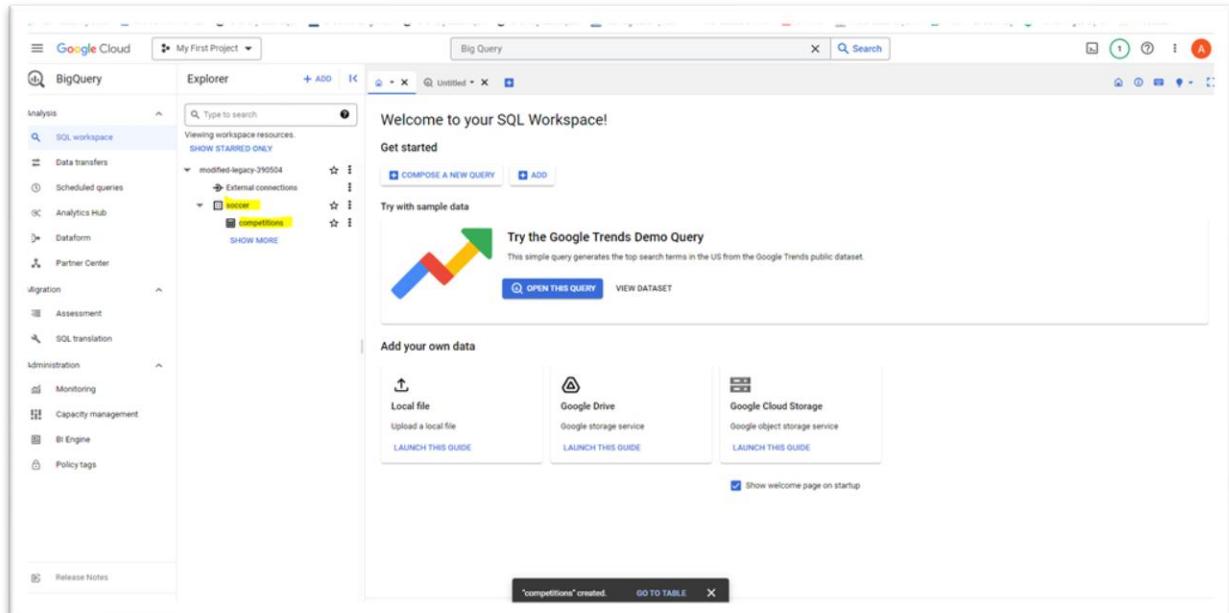
AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

Filling up the right pane information for creating the table,



Click on the CREATE TABLE button to finalize the operation.

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023



Repeating the same data ingestion process for below information,

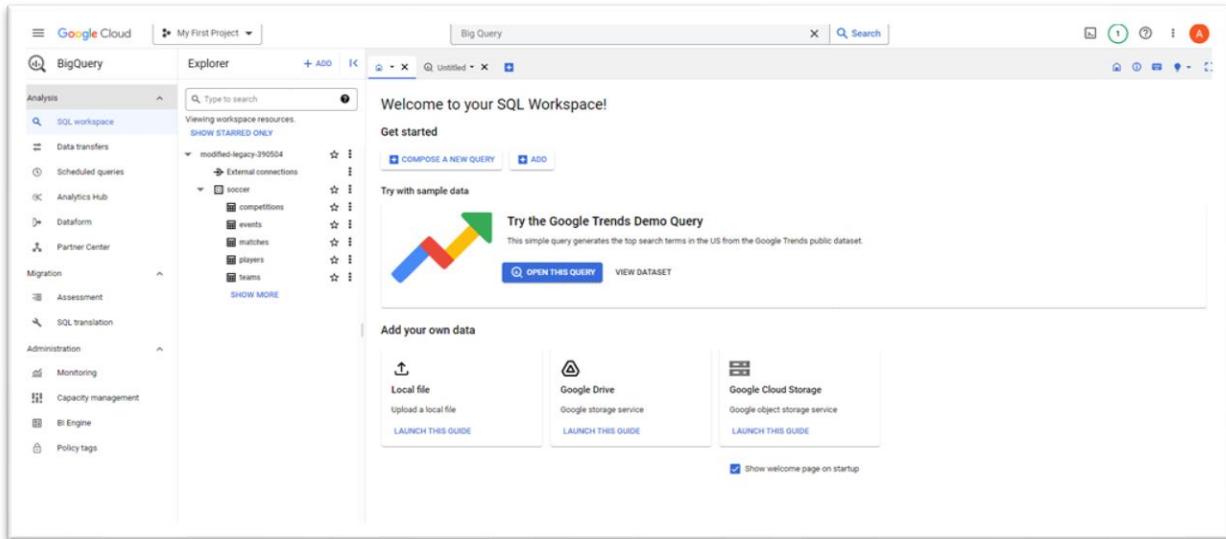
GCS bucket file	Table name
spls/bq-soccer-analytics/matches.json	matches
spls/bq-soccer-analytics/teams.json	teams
spls/bq-soccer-analytics/players.json	players
spls/bq-soccer-analytics/events.json	events

After following the same process, we have created all the above-mentioned tables.

AI Lab 2 (Assignment – 2)

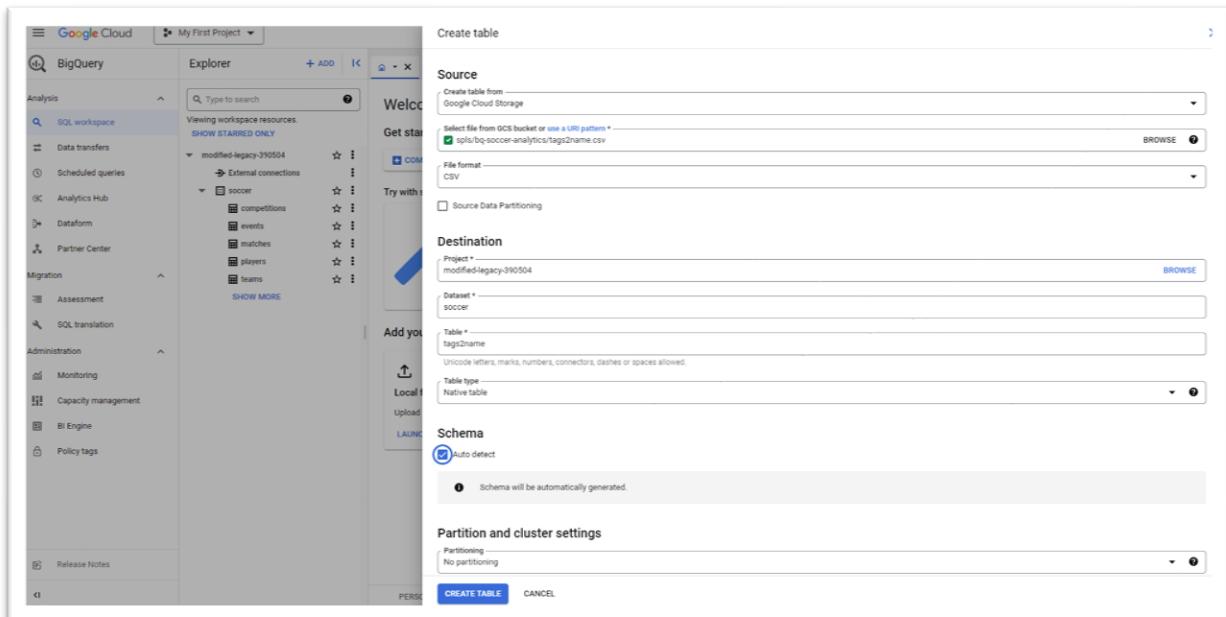
Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



Load the CSV data,

Create a table by clicking on the View actions icon next to your soccer dataset in the Explorer section and select Create table. Fill out the form in right pane,



AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

Previewing the table: In the left pane, select **soccer > competitions** in the navigation panel. In the Details panel, click the **Preview** tab.

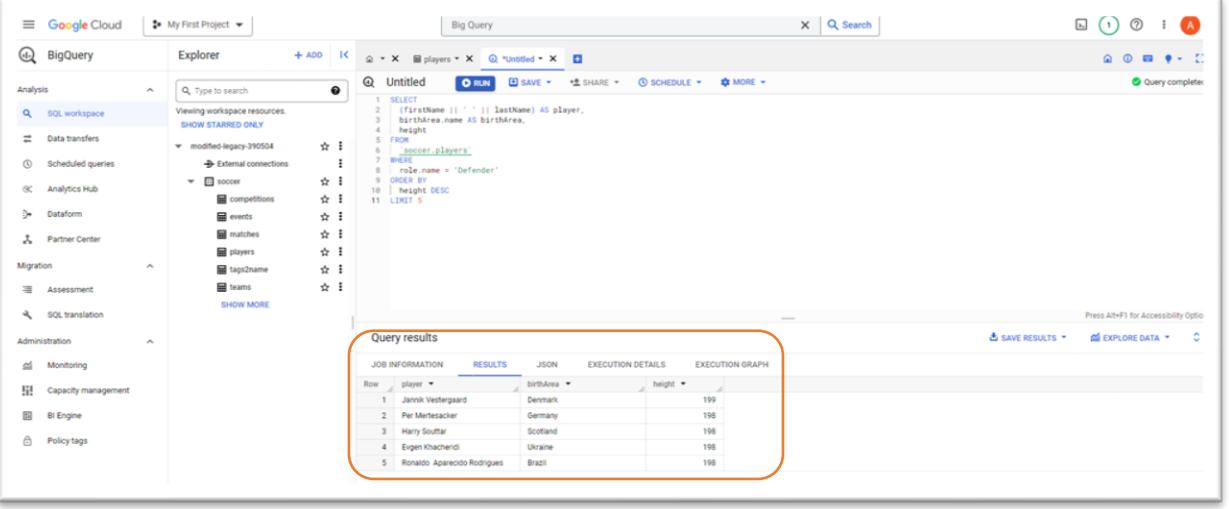
Row	type	format	area_alpha2code	area_alpha3code	area_id	area_name	wylid	name
1	club	Domestic league	IT	ITA	380	Italy	524	Italian first c
2	club	Domestic league		XEN	0	England	364	English first
3	club	Domestic league	ES	ESP	724	Spain	795	Spanish first
4	club	Domestic league	FR	FRA	250	France	412	French first
5	club	Domestic league	DE	DEU	276	Germany	426	German first
6	international	International cup		XEU	0		102	European Cr
7	international	International cup		XWO	0		28	World Cup

Query Player data: Click on the Query and choose In new tab and run the given query,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



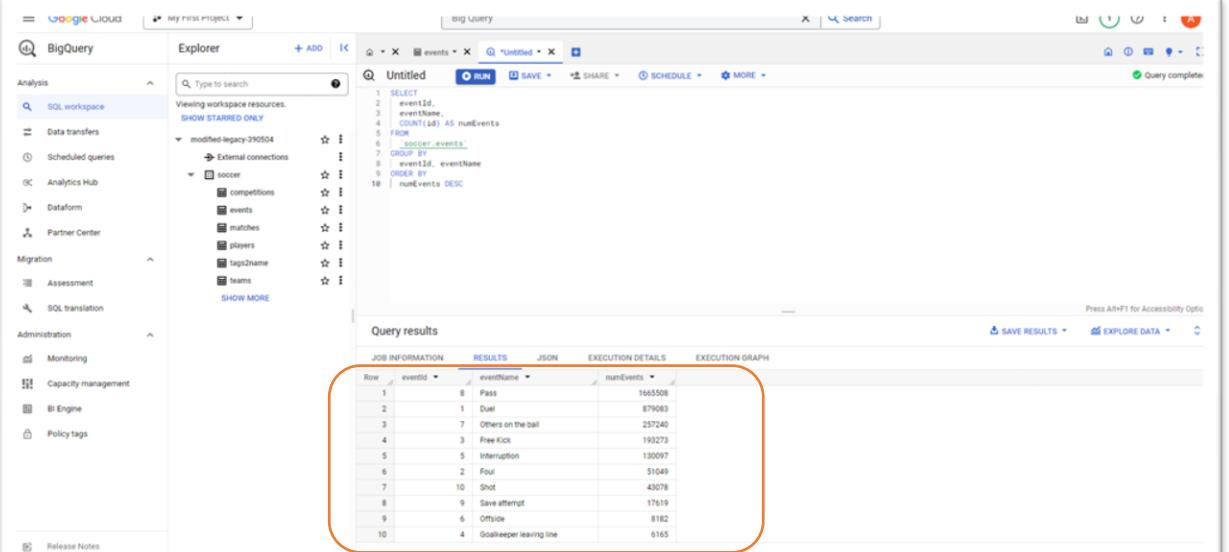
The screenshot shows the Google BigQuery interface within the Google Cloud Platform. The left sidebar displays various services like BigQuery, Analytics Hub, Dataform, and Partner Center. The main area shows a query editor titled 'Untitled' with the following SQL code:

```
1 SELECT
2   (firstName || ' ' || lastName) AS player,
3   birthArea.name AS birthArea,
4   height
5 FROM
6   soccer_players
7 WHERE
8   role_name = 'Defender'
9 ORDER BY
10  height DESC
11 LIMIT 5
```

The 'Query results' section below the code displays a table with five rows of data:

Row	player	birthArea	height
1	Jannik Vestergaard	Denmark	199
2	Per Mertesacker	Germany	198
3	Harry Soutar	Scotland	198
4	Eugen Kachenzi	Ukraine	198
5	Ronaldo Aparecido Rodrigues	Brazil	198

Query events data: Follow the previous steps and execute the different query,



The screenshot shows the Google BigQuery interface within the Google Cloud Platform. The left sidebar displays various services like BigQuery, Analytics Hub, Dataform, and Partner Center. The main area shows a query editor titled 'Untitled' with the following SQL code:

```
1 SELECT
2   eventId,
3   eventName,
4   COUNT(eventId) AS numEvents
5 FROM
6   soccer_events
7 GROUP BY
8   eventId, eventName
9 ORDER BY
10  numEvents DESC
11 LIMIT 10
```

The 'Query results' section below the code displays a table with ten rows of data:

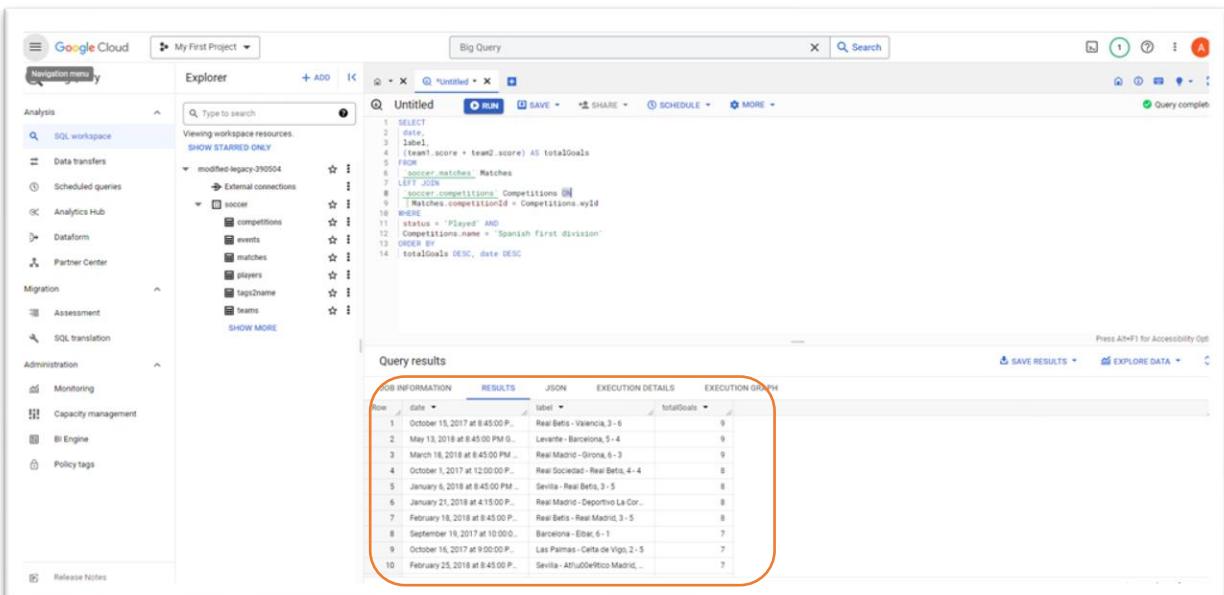
Row	eventId	eventName	numEvents
1	8	Pass	1665508
2	1	Duel	879083
3	7	Others on the ball	257240
4	3	Free Kick	193273
5	5	Interruption	130097
6	2	Foul	51049
7	10	Shot	43078
8	9	Save attempt	17619
9	6	Offside	9182
10	4	Goalkeeper leaving line	6165

Now the data ingestion is completed hence going back to main lab of Soccer Data Analysis.

Matches with the most goals: Opening the query editor run below query,

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

```
SELECT
    date,
    label,
    (team1.score + team2.score) AS totalGoals
FROM
    `soccer.matches` Matches
LEFT JOIN
    `soccer.competitions` Competitions ON
        Matches.competitionId = Competitions.wyId
WHERE
    status = 'Played' AND
    Competitions.name = 'Spanish first division'
ORDER BY
    totalGoals DESC, date DESC
```



The screenshot shows the Google Cloud Big Query interface. On the left, the navigation menu is visible with sections like Analysis, Data transfers, Analytics Hub, Dataform, Partner Center, Migration, Assessment, and SQL translation. The main area shows a query editor with the following code:

```
1 SELECT
2     date,
3     label,
4     (team1.score + team2.score) AS totalGoals
5 FROM
6     `soccer.matches` Matches
7 LEFT JOIN
8     `soccer.competitions` Competitions ON
9         Matches.competitionId = Competitions.wyId
10 WHERE
11     status = 'Played' AND
12     Competitions.name = 'Spanish first division'
13 ORDER BY
14     totalGoals DESC, date DESC
```

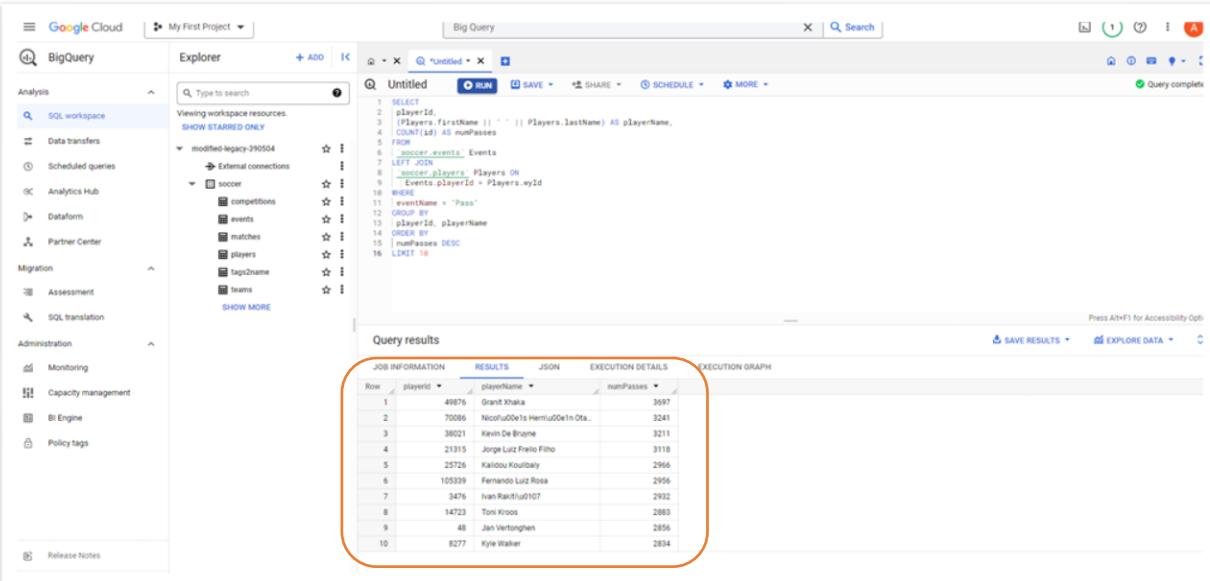
Below the code, the "Query results" section displays a table with 10 rows of data. The table has columns: Row, date, label, and totalGoals. The data is as follows:

Row	date	label	totalGoals
1	October 15, 2017 at 8:45:00 P.M.	Real Betis - Valencia, 3 - 6	9
2	May 13, 2018 at 8:45:00 P.M.	Levante - Barcelona, 5 - 4	9
3	March 18, 2018 at 8:45:00 P.M.	Real Madrid - Girona, 6 - 2	9
4	October 1, 2017 at 12:00:00 P.M.	Real Sociedad - Real Betis, 4 - 4	8
5	January 6, 2018 at 8:45:00 P.M.	Sevilla - Real Betis, 3 - 5	8
6	January 21, 2018 at 4:15:00 P.M.	Real Madrid - Deportivo La Cor...	8
7	February 18, 2018 at 8:45:00 P.M.	Real Betis - Real Madrid, 3 - 5	8
8	September 19, 2017 at 10:00:00 P.M.	Barcelona - Eibar, 6 - 1	7
9	October 16, 2017 at 9:00:00 P.M.	Las Palmas - Celta de Vigo, 2 - 5	7
10	February 25, 2018 at 8:45:00 P.M.	Sevilla - Atlético de Madrid, ...	7

Players with the most passes: Opening the query editor run below query,

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

```
SELECT
    playerId,
    (Players.firstName || ' ' || Players.lastName) AS playerName,
    COUNT(id) AS numPasses
FROM
    `soccer.events` Events
LEFT JOIN
    `soccer.players` Players ON
        Events.playerId = Players.wyId
WHERE
    eventName = 'Pass'
GROUP BY
    playerId, playerName
ORDER BY
    numPasses DESC
LIMIT 10
```



The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar displays various Google Cloud services like Analytics Hub, Dataform, and Partner Center. The main area shows the query editor with the previously written SQL code. Below the code, the 'Query results' section displays a table with 10 rows of data. The table has columns: Row, playerId, playerName, and numPasses. The data is as follows:

Row	playerId	playerName	numPasses
1	49876	Grant Xhaka	3697
2	70086	Nicolo Della Hemidio Ota...	3241
3	38021	Kevin De Bruyne	3211
4	21315	Jorge Luiz Frelo Filho	3118
5	25726	Kalidou Koulibaly	2966
6	105399	Fernando Luiz Rosa	2956
7	3476	Ivan Rakitic107	2932
8	14723	Toni Kroos	2883
9	48	Jan Vertonghen	2856
10	8277	Kyle Walker	2834

Determine penalty kick success rate: Opening the query editor run below query,

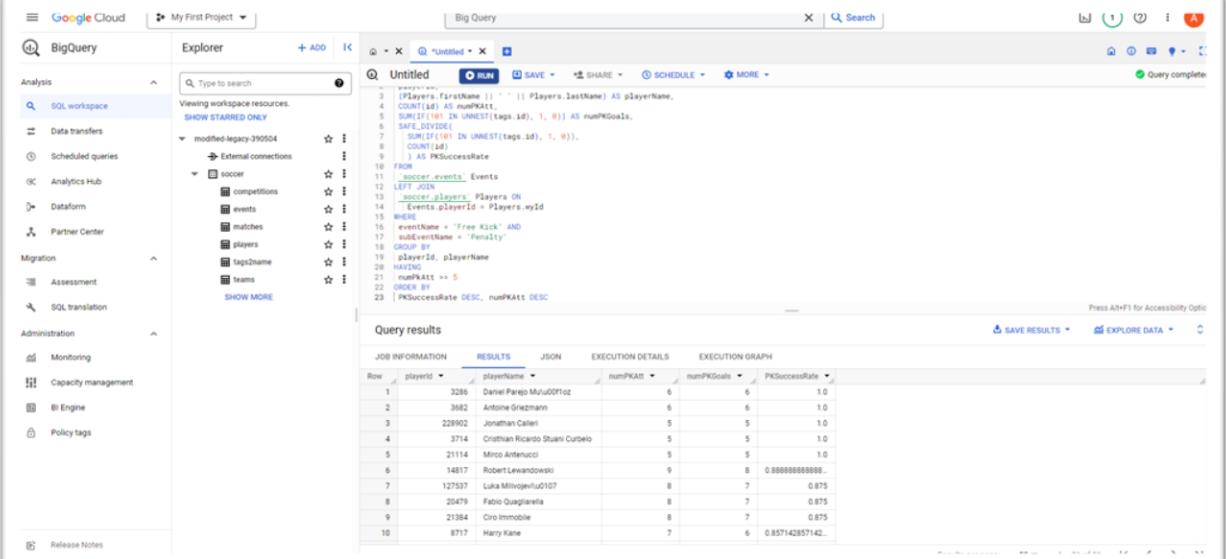
AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

```
SELECT
    playerId,
    (Players.firstName || ' ' || Players.lastName) AS playerName,
    COUNT(id) AS numPKAtt,
    SUM(IF(101 IN UNNEST(tags.id), 1, 0)) AS numPKGoals,
    SAFE_DIVIDE(
        SUM(IF(101 IN UNNEST(tags.id), 1, 0)),
        COUNT(id)
    ) AS PKSuccessRate
FROM
    `soccer.events` Events
LEFT JOIN
    `soccer.players` Players ON
        Events.playerId = Players.wyId
WHERE
    eventName = 'Free Kick' AND
    subEventName = 'Penalty'
GROUP BY
    playerId, playerName
HAVING
    numPkAtt >= 5
ORDER BY
    PKSuccessRate DESC, numPKAtt DESC
```

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar lists various Google Cloud services like Google Analytics Hub, Dataform, and Partner Center. The main area shows a query editor titled "Untitled" with the following SQL code:

```
1 SELECT
2   (Players.firstName || ' ' || Players.lastName) AS playerName,
3   COUNT(id) AS numPKAtt,
4   SUM(IF(id IN UNNEST(tags.id), 1, 0)) AS numPKGoals,
5   SUM(IF(id IN UNNEST(tags.id), 1, 0)) / COUNT(id)
6   ) AS PKSuccessRate
7   FROM
8   `soccer_events`_Events
9   LEFT JOIN
10  `soccer_players`_Players ON
11  Events.playerId = Players.myId
12  WHERE
13  eventName = 'Free Kick' AND
14  addEventName = 'Penalty'
15  GROUP BY
16  playerId, playerName
17  HAVING
18  numPKAtt >= 5
19  ORDER BY
20  PKSuccessRate DESC, numPKAtt DESC
```

The "Query results" section displays a table with the following data:

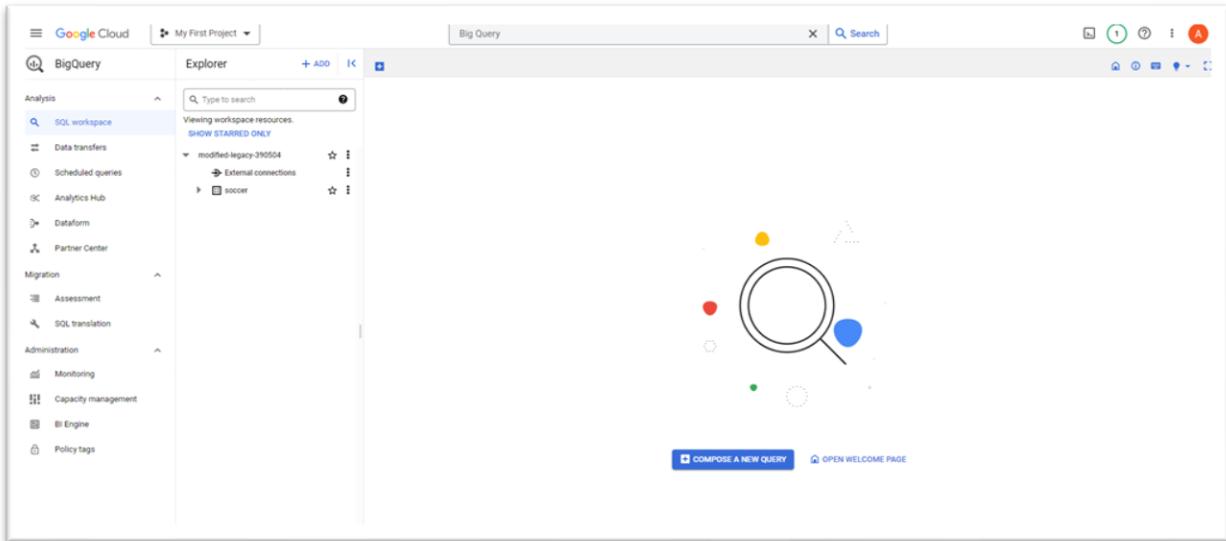
Row	playerId	playerName	numPKAtt	numPKGoals	PKSuccessRate
1	3286	Daniel Parejo Muñoz0f1oz	6	6	1.0
2	3682	Antoine Griezmann	6	6	1.0
3	229902	Jonathan Calleri	5	5	1.0
4	3714	Cristian Ricardo Stuani Curbelo	5	5	1.0
5	21114	Miroslav Antenucci	5	5	1.0
6	14817	Robert Lewandowski	9	8	0.8888888888888888
7	127537	Luka Modrić0u107	8	7	0.875
8	20479	Fabio Quagliarella	8	7	0.875
9	21384	Ciro Immobile	8	7	0.875
10	8717	Harry Kane	7	6	0.8571428571428571

[End of Big Query Soccer Data Analysis hands on](#)

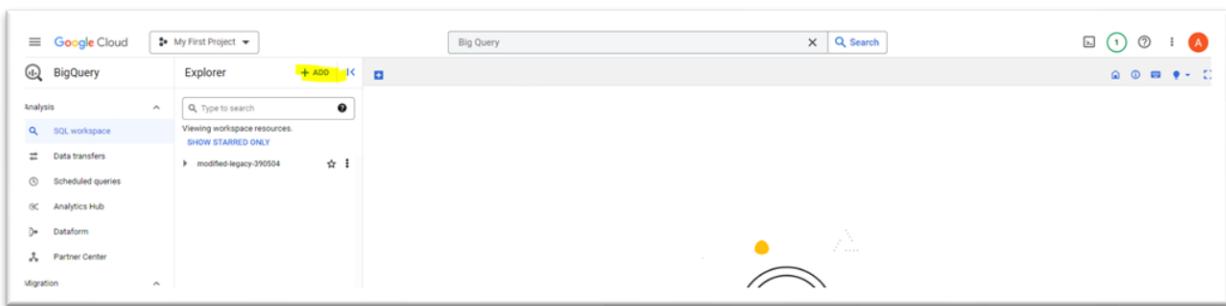
Introduction to SQL for Big Query and Cloud SQL: -

We are now in Big Query console window as below screenshots,

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023



Click on the 'ADD'

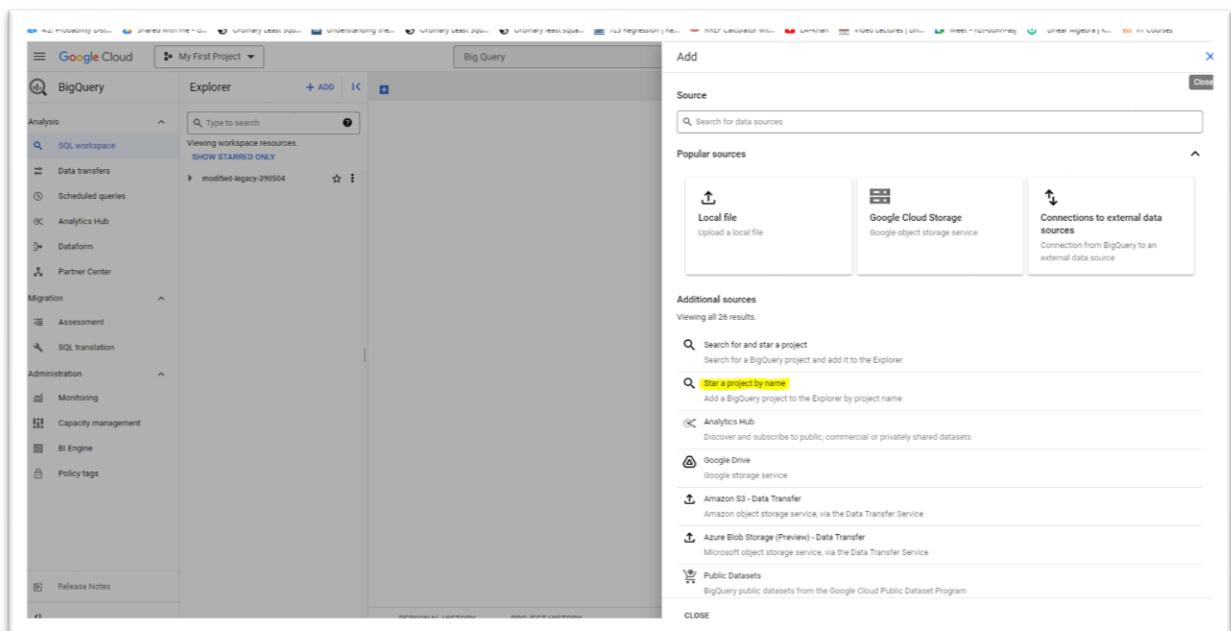


Choose Star a project by name.

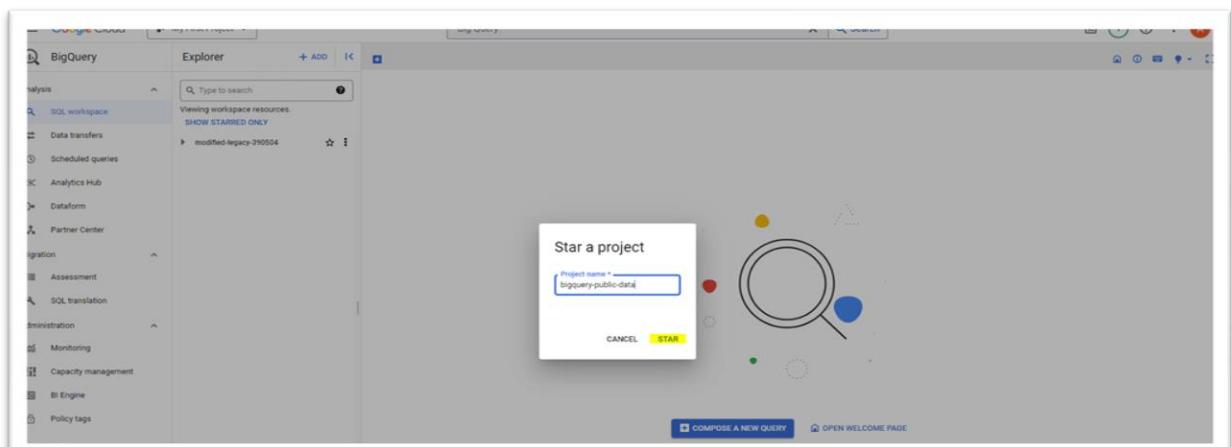
AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

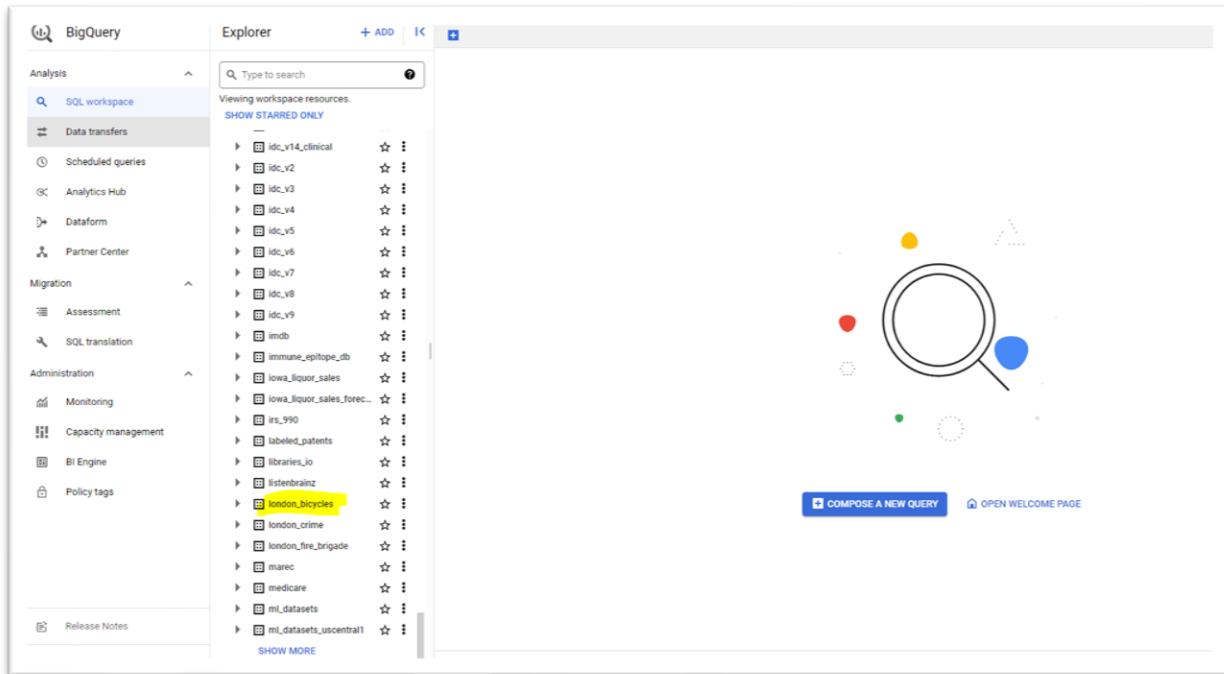


Enter the Project Name and click on the STAR,



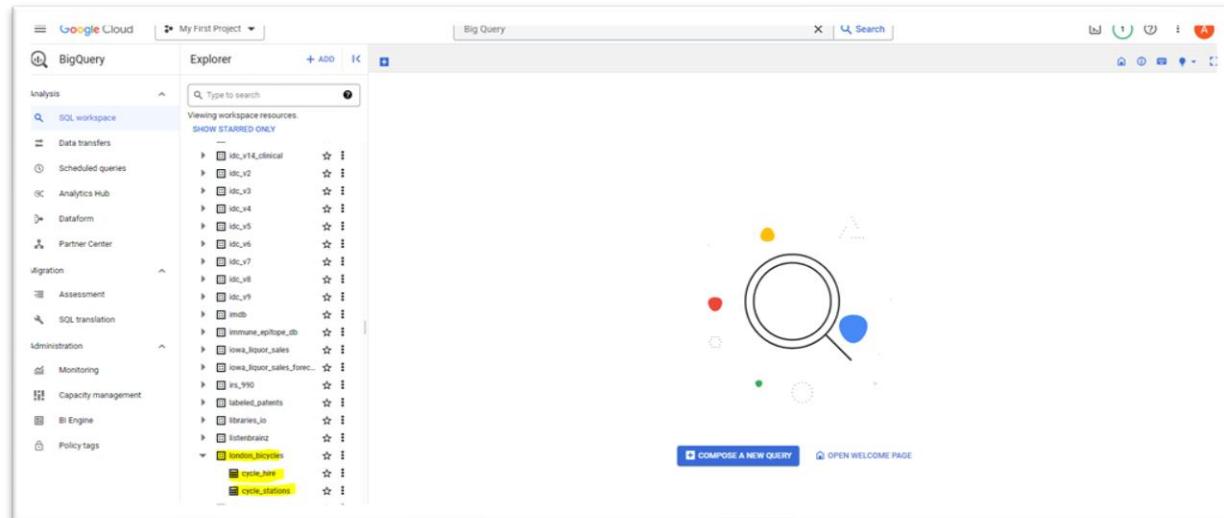
AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

Click on the london_bicycles dataset to reveal the associated tables,



The screenshot shows the Google BigQuery interface. On the left, the sidebar lists various project components: Analysis, Data transfers, Scheduled queries, Analytics Hub, Dataform, Partner Center, Migration, Assessment, SQL translation, Administration, Monitoring, Capacity management, BI Engine, and Policy tags. Under the 'Data transfers' section, the 'london_bicycles' dataset is highlighted with a yellow box. The main pane displays a search bar at the top and a list of datasets below it. The 'london_bicycles' dataset is expanded, revealing three associated tables: 'cycle_hire', 'cycle_stations', and 'cycle_stations'. The interface includes decorative icons of colored circles and a magnifying glass.

Expanded version,



This screenshot is identical to the one above, showing the Google BigQuery interface with the 'london_bicycles' dataset expanded. The three associated tables—'cycle_hire', 'cycle_stations', and 'cycle_stations'—are clearly visible under the expanded dataset entry. The interface layout, sidebar navigation, and decorative elements are consistent with the first screenshot.

Previewing the tables,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud BigQuery interface with the 'cycle_hire' table selected. The table has 74 rows and 13 columns. The columns are: rental_id, duration, station_start, bike_id, bike_model, end_date, end_station_id, end_station_name, start_date. The data includes various rental records with details like duration (e.g., 3840, 3840000), bike models (e.g., 4229, 728, 4375), and locations (e.g., Golden Square, Soho, Embankment (Savoy), Strand, Green Park Station, Mayfair, Park Lane, Hyde Park, Wellington Arch, Hyde Park, St. Martin's Road, North Kensington, Sloane's Gate, Westminster, Black Lion Gate, Kensington Ga., Little Brook Green, Brook Green, The Vale, Chelsea, Flandersburgh Street, Ladbroke Grove, Denry Street, Kensington, Park Lane, Hyde Park, Gottman Street, St. Paul's, Hyde Park Corner, Hyde Park, Arlington Road, Camden Town, Golden Lane, Barbican, Podium, Queen Elizabeth Gym., Hanley Crescent, Camden Town, Finsbury Street, Hoxton, Park Lane, Marylebone, Copper Box Arena, Queen Eliza., Ilchester Gardens, Bayswater, Copper Box Arena, Queen Eliza., Marylebone Lane, Marylebone, Whitehall, Rawson). The interface also shows the schema, preview, and lineage tabs.

The screenshot shows the Google Cloud BigQuery interface with the 'cycle_stations' table selected. The table has 74 rows and 13 columns. The columns are: id, instance, latitude, locked, longitude, name, bikes_count, docks_count, nbEmptyDocks, temp, terminal_name. The data includes various station locations with details like coordinates (e.g., 51.499732, 51.499083, 51.305044, 51.496468, 51.49382705, 51.50204238, 51.499547, 51.5193043, 51.51070161, 51.4805753, 51.51362054, 51.51733427, 51.514274, 51.5162164, 51.50923, 51.494412, 51.48053171, 51.492462, 51.51733427, 51.51932, 51.4918566, 51.485821, 51.50486, 51.5208417, 51.50942848, 51.51071065, 51.512999, 51.494412, 51.4744969, 51.492462, 51.71619959, 51.482704, 51.490024, 51.13001, 51.50486, 0.07340162, 51.14261981). The interface also shows the schema, preview, and lineage tabs.

Executing the queries on the tables,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar lists various projects and datasets. The main area shows a query titled "Untitled" with the following SQL code:

```
1 SELECT end_station_name FROM bigquery-public-data.london_bicycles.cycle_hire;
```

The results table displays 10 rows of data from the "london_bicycles" dataset, specifically from the "cycle_hire" table, showing the names of the end stations. The results are as follows:

end_station_name
Doddington Grove, Kennington
Doddington Grove, Kennington
St. Martins Grove, West End
Graham Street, Angel
Newgate Street, St. Paul's
Abingdon Green, Westminster
Abingdon Green, Westminster

This screenshot shows the same Google Cloud BigQuery interface as the previous one, but with a different query. The query is:

```
1 SELECT * FROM bigquery-public-data.london_bicycles.cycle_hire WHERE duration>1200;
```

The results table displays 10 rows of data, showing bicycle rental details where the duration is greater than 1200 seconds. The results are as follows:

rental_id	duration	duration_ms	bike_id	bike_model	end_date	end_station_id	end_station_name	start_date
72477971	28260	28260000	14778	null	2018-01-09 15:07:00 UTC	598	Parsons Green, Parsons Green	2018-01-09 15:07:00 UTC
98617474	3050	30500000	17719	null	2020-07-20 17:09:00 UTC	194	Hop Exchange, The Borough	2020-07-20 17:09:00 UTC
86652536	3120	31200000	16213	null	2019-05-09 11:21:00 UTC	357	Howland Street, Fitzrovia	2019-05-09 11:21:00 UTC
100728157	3660	36600000	7727	null	2020-08-14 18:54:00 UTC	462	Bonny Street, Camden Town	2020-08-14 18:54:00 UTC
57702407	7660	76800000	1523	null	2016-08-14 21:25:00 UTC	613	Woodstock Grove, Shepherd's Bush	2016-08-14 21:25:00 UTC
97302983	3120	31200000	15374	null	2020-05-20 13:40:00 UTC	682	Crop Road, Hammersmith	2020-05-20 13:40:00 UTC
97016581	3720	37200000	6540	null	2020-05-17 12:49:00 UTC	428	Exhibition Road, Knightsbridge	2020-05-17 12:49:00 UTC
75123547	3120	31200000	9821	null	2018-05-01 22:24:00 UTC	755	The Vale, Chelsea	2018-05-01 22:24:00 UTC
68284788	6000	60000000	1596	null	2017-06-19 00:14:00 UTC	510	Westberry DLR, Limehouse	2017-06-19 00:14:00 UTC
471630448	4670	46700000	4671	null	2015-03-28 13:08:00 UTC	79	Old Street Station, Finsbury Park	2015-03-28 13:08:00 UTC

More SQL Keywords: GROUP BY, COUNT, AS, and ORDER BY: Running different queries on the table and sharing the result screenshots below,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud Big Query interface. On the left is a navigation menu with sections like Analysis, Migration, Administration, and Monitoring. The main area has tabs for 'Explorer' and 'Big Query'. A query editor window titled 'Untitled' contains the following SQL code:

```
1 SELECT start_station_name FROM bigquery-public-data.london\_bicycles.cycle\_hire GROUP BY start_station_name;
```

The 'Query results' section displays a table with 10 rows of data, each representing a start station name. The columns are labeled 'Row' and 'start_station_name'. The data is as follows:

Row	start_station_name
1	Bridge Avenue, Hammersmith
2	Grosvenor Road, Pimlico
3	Great Tower Street, Monument
4	Cardinal Place, Victoria
5	Howick Place, Westminster
6	Clinton Road, Mile End
7	Croner Street, Bloomsbury
8	Mile End Park Leisure Centre, ...
9	Fournier Street, Whitechapel
10	Chesilton Road, Fulham

At the bottom of the results table, it says 'Results per page: 50 1 - 50 of 954'.

This screenshot is identical to the one above, showing the same Google Cloud Big Query interface and results table. The data in the table remains the same, listing 10 start station names from the London bicycle hire dataset.

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar lists various projects and datasets. The main area shows an untitled query window with the following SQL code:

```
1 SELECT start_station_name, COUNT(*) AS num_starts FROM bikery-public-data.london_bicycles.cycle_hire GROUP BY start_station_name;
```

The results table displays the top 10 start stations with their respective counts:

start_station_name	num_starts
Hyde Park Corner, Hyde Park	671688
William IV Street, Strand	210375
Belgrave Road, Victoria	136434
Finsbury Circus, Liverpool Street	239326
Bourne Street, Brixton	74388
London Zoo Car Park, The Reg...	81054
Pott Street, Bethnal Green	129141
Southampton Place, Holborn	112333
Old Street Station, St. Louis	288738
Fins Street, Bethnal Green	71212

This screenshot is identical to the one above, showing the Google Cloud BigQuery interface with the same query results for bicycle hire data. The results table displays the top 10 start stations with their respective counts:

start_station_name	num_starts
Bermondsey Street, Bermondsey	183604
Charles II Street, West End	109423
Great Dover Street, The Borough	98435
St. George's Square, Pimlico	127805
Rodney Road, Walworth	105565
Uxbridge Road, Chigwell Junction	92689
Southwark Station 2, Southwark	122640
Alfred Place, Bloomsbury	88574
Green Park Station, Mayfair	233217
Holy Trinity Brompton, Knights...	103862

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar lists various Google Cloud services like Analytics Hub, Dataform, and Partner Center. The main area shows a query titled "Untitled" running against the dataset "bigquery-public-data.london_bicycles.cycle_hire". The query is:

```
SELECT start_station_name, COUNT(*) AS num FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name ORDER BY start_station_name;
```

The results table has columns "start_station_name" and "num". The data is as follows:

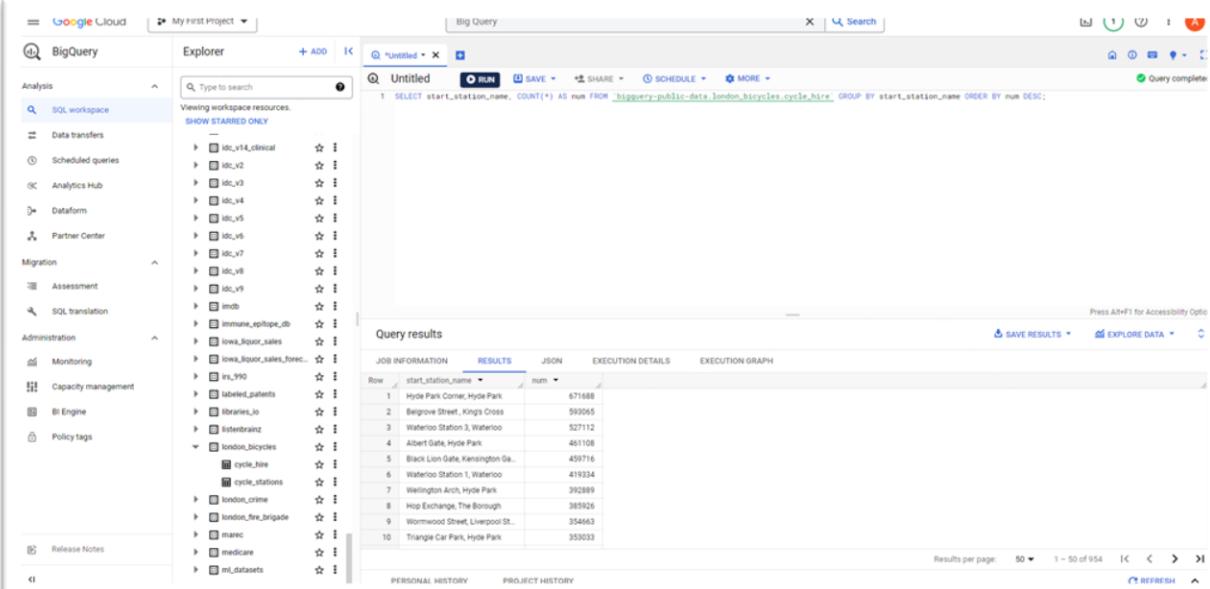
start_station_name	num
Abbey Orchard Street, Westminster	121574
Abbotsbury Road, Holland Park	38191
Aberdeen Place, St. Johns Wood	61413
Aberfeldy Street, Poplar	21158
Abingdon Green, Great College St.	4569
Abingdon Green, Westminster	134157
Abingdon Villas, Kensington	64812
Abingdon Close, Clapham Junction	52771
Ackroyd Drive, Bow	61193
Ada Street, Hackney Central	106514

This screenshot is identical to the one above, showing the same BigQuery interface and results for the same query. The data in the results table is the same, listing the top 10 start stations with their respective counts.

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



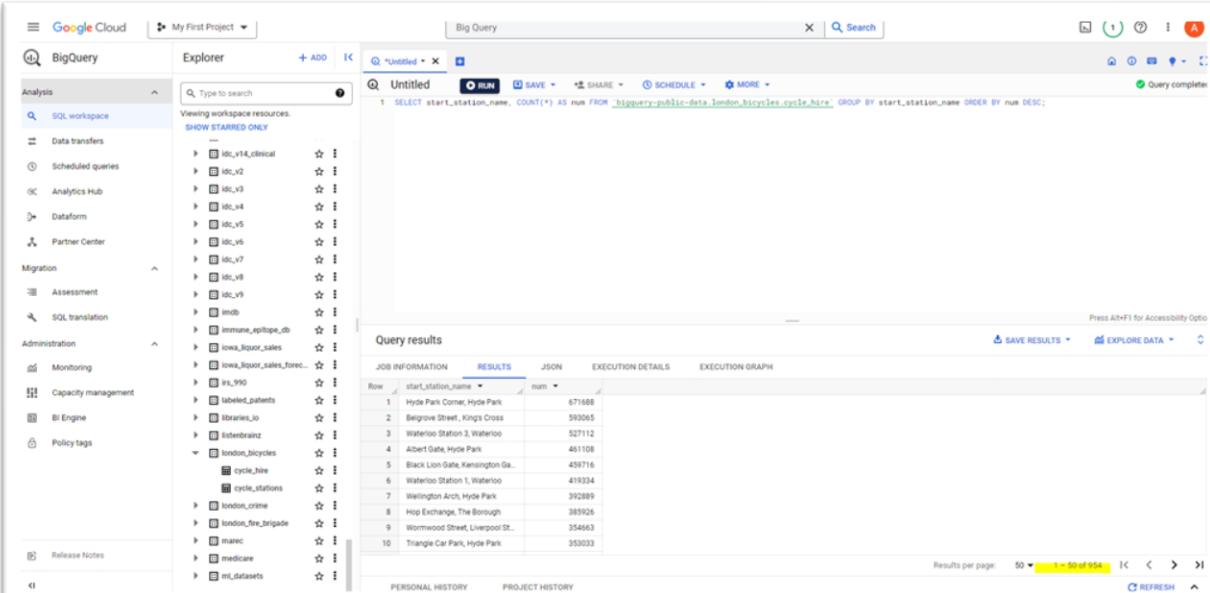
The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar lists various Google Cloud services like Analytics Hub, Dataform, and Partner Center. The main area shows a query editor with an untitled query tab. The query is:

```
1 SELECT start_station_name, COUNT(*) AS num FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name ORDER BY num DESC;
```

The results table has columns: Row, start_station_name, and num. The data is:

Row	start_station_name	num
1	Hyde Park Corner, Hyde Park	671688
2	Belgrave Street, King's Cross	593065
3	Waterloo Station 3, Waterloo	527112
4	Albert Gate, Hyde Park	461108
5	Black Lion Gate, Kensington Ga.	459716
6	Waterloo Station 1, Waterloo	419334
7	Wellington Arch, Hyde Park	392889
8	Hop Exchange, The Borough	385926
9	Wormwood Street, Liverpool St.	354663
10	Triangle Car Park, Hyde Park	353033

Working with Cloud SQL: Running different queries on the table and sharing the result screenshots below,



This screenshot is identical to the one above, showing the same query and results table. The data is the same as follows:

Row	start_station_name	num
1	Hyde Park Corner, Hyde Park	671688
2	Belgrave Street, King's Cross	593065
3	Waterloo Station 3, Waterloo	527112
4	Albert Gate, Hyde Park	461108
5	Black Lion Gate, Kensington Ga.	459716
6	Waterloo Station 1, Waterloo	419334
7	Wellington Arch, Hyde Park	392889
8	Hop Exchange, The Borough	385926
9	Wormwood Street, Liverpool St.	354663
10	Triangle Car Park, Hyde Park	353033

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar navigation includes Analysis, SQL workspace, Data transfers, Scheduled queries, Analytics Hub, Dataform, Partner Center, Migration, Assessment, SQL translation, Administration, Monitoring, Capacity management, BI Engine, Policy tags, and Release Notes. The main area displays a query titled "Untitled" with the following SQL code:

```
1 SELECT end_station_name, COUNT(*) AS num FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY end_station_name ORDER BY num DESC;
```

The "Query results" section shows the output of the query:

end_station_name	num
Hyde Park Corner, Hyde Park	671680
Belgrave Street, Kings Cross	586568
Hop Exchange, The Borough	519033
Waterloo Station 3, Waterloo	508421
Albert Gate, Hyde Park	464713
Black Lion Gate, Kensington Ga...	452412
Waterloo Station 1, Waterloo	403861
Wellington Arch, Hyde Park	388554
Brunel Street, Liverpool St...	380069
Wormwood Street, Liverpool St...	372718

At the bottom right of the results table, there are buttons for "SAVE RESULTS" and "EXPLORE DATA". Below the results table, there are "PERSONAL HISTORY" and "PROJECT HISTORY" links.

Saved the file from the executed query,

Saved as Start

The screenshot shows the Google Cloud BigQuery interface, similar to the previous one but with different results. The main area displays a query titled "Untitled" with the following SQL code:

```
1 SELECT start_station_name, COUNT(*) AS num FROM `bigquery-public-data.london_bicycles.cycle_hire` GROUP BY start_station_name ORDER BY num DESC;
```

The "Query results" section shows the output of the query:

start_station_name	num
Hyde Park Corner, Hyde Park	671688
Belgrave Street, Kings Cross	593065
Waterloo Station 3, Waterloo	527112
Albert Gate, Hyde Park	461108
Black Lion Gate, Kensington Ga...	459716
Waterloo Station 1, Waterloo	419334
Wellington Arch, Hyde Park	392889
Hop Exchange, The Borough	385926
Wormwood Street, Liverpool St...	354663
Triangle Car Park, Hyde Park	353033

To the right of the results table, there are several export options:

- CSV (Google Drive): Save up to 1TB as CSV to Google Drive.
- CSV (local file): Save up to 1TB as CSV locally.
- JSON (local file): Save up to 1TB as JSON locally.
- JSONL (newline delimited): Save up to 1TB as newline delimited JSON to Google Drive.
- BigQuery table: Save results as a BigQuery table.
- Google Sheets: Save up to 1MB to Google Sheets.
- Copy to Clipboard: Copy up to 1MB to the clipboard.

Saved as

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google BigQuery interface. On the left, the sidebar includes sections for Analysis, Data transfers, Analytics Hub, Dataform, Partner Center, Migration, Assessment, SQL translation, Administration, Monitoring, Capacity management, BI Engine, and Policy tags. The main area displays a query titled "Untitled" with the following SQL code:

```
1 SELECT end_station_name, COUNT(*) AS num FROM bikeshare-public-data.london_bicycles.cycle_hire GROUP BY end_station_name ORDER BY num DESC;
```

The "Query results" table has columns: JOB INFORMATION, RESULTS (selected), JSON, EXECUTION DETAILS, and EXECUTION GRAPH. The results show the top 10 end stations with their counts:

end_station_name	num
Hyde Park Corner, Hyde Park	671600
Belgrave Street, King's Cross	586568
Hop Exchange, The Borough	519033
Waterloo Station 3, Waterloo	508421
Albert Gate, Hyde Park	464713
Black Lion Gate, Kensington Ga.	452412
Waterloo Station 1, Waterloo	403961
Wellington Arch, Hyde Park	388354
Brudenell Street, Liverpool St...	380669
Wormwood Street, Liverpool St...	372718

On the right, there are options to save results to CSV (Google Drive, local file), JSON (local file), or BigQuery table, or to export to Google Sheets or copy to clipboard.

Upload CSV files to Cloud Storage: Select **Navigation menu > Cloud Storage > Buckets**, and then click **CREATE BUCKET**

Land into to the Cloud Storage Dashboard,

The screenshot shows the Google Cloud Storage Dashboard. The left sidebar includes Cloud Storage, Monitoring (NEW), and Settings. The main area features a "Try The New Cloud Storage Monitoring Dashboard" section with a "TRY NOW" button. Below it is a "View security recommendations" section with a "VIEW IN TABLE" button. A table lists buckets:

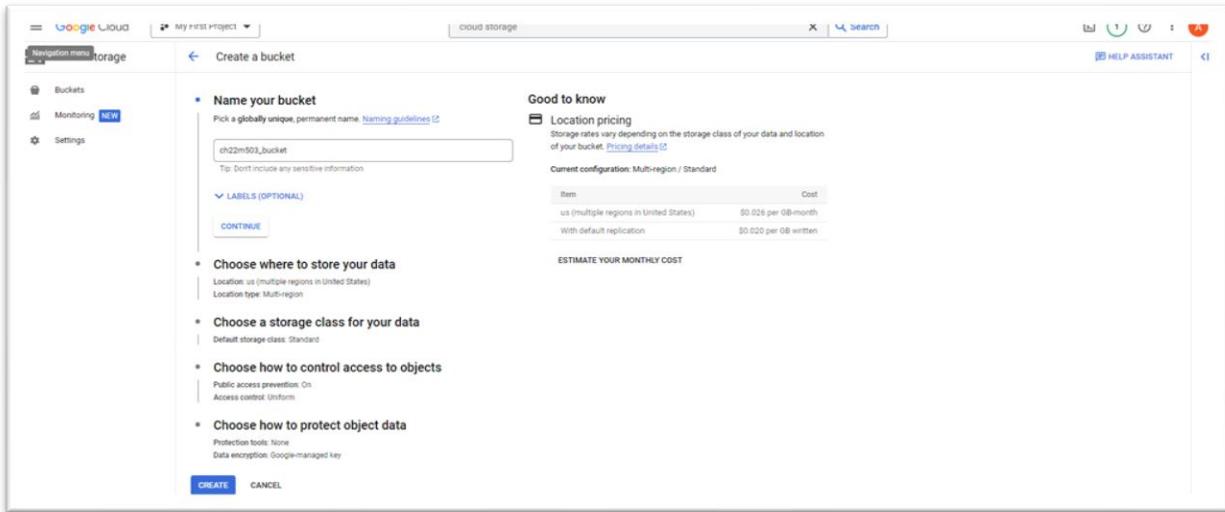
Name	Created	Location type	Location	Default storage class	Last modified	Public access	Action
dataproc-staging-us-central1-89183523	Jun 21, 2023, 10:37:05 AM	Region	us-central1	Standard	Jun 21, 2023, 10:37:05 AM	Subject to object ACLs	Fire+
dataproc-temp-us-central1-891835240	Jun 21, 2023, 10:37:05 AM	Region	us-central1	Standard	Jun 21, 2023, 10:37:05 AM	Subject to object ACLs	Fire+

The right side of the dashboard contains a sidebar with links to "Getting bucket information", "Uploading objects", "Downloading objects", "Use cases for Cloud Storage", "Terraform samples", and "Making data public".

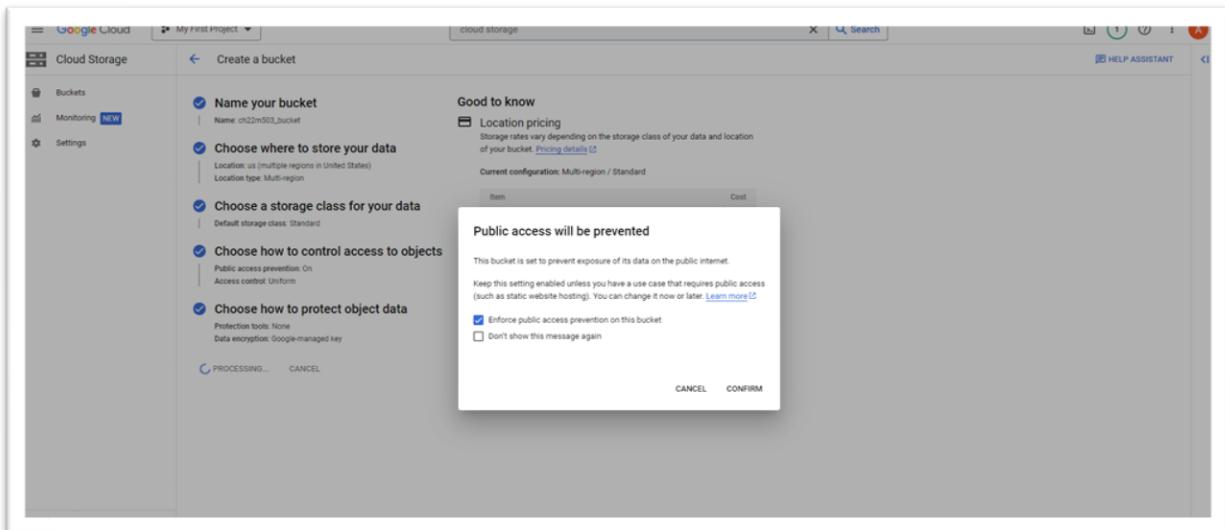
AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



click **Confirm** for Public access will be prevented dialog,



Bucket created,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud Storage interface. The left sidebar has 'My First Project' selected under 'Cloud Storage'. The main area shows a bucket named 'ch22m503_bucket'. The bucket details are as follows:

- Location:** us (multiple regions in United States)
- Storage class:** Standard
- Public access:** Not public
- Protection:** None

The 'OBJECTS' tab is selected. Below it, there are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'TRANSFER DATA', 'MANAGE HOLDS', 'DOWNLOAD', and 'DELETE'. A filter bar allows filtering by name prefix, type, and other metadata. The message 'No rows to display' is shown.

Click on upload files for uploading the CSV files,

This screenshot is identical to the previous one, but the 'UPLOAD FILES' button is highlighted with a yellow box. All other elements, including the bucket details and the 'No rows to display' message, remain the same.

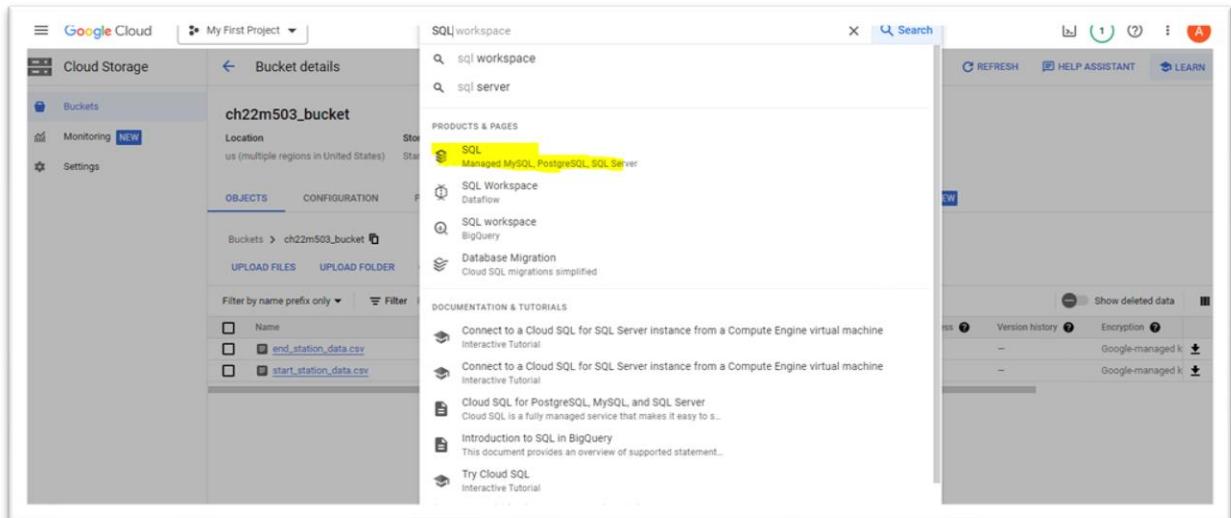
Uploaded files are available in bucket details page,

The screenshot shows the Google Cloud Storage interface with the 'Storage' tab selected in the top navigation bar. The bucket 'ch22m503_bucket' is selected. The 'OBJECTS' tab is selected, showing two uploaded CSV files:

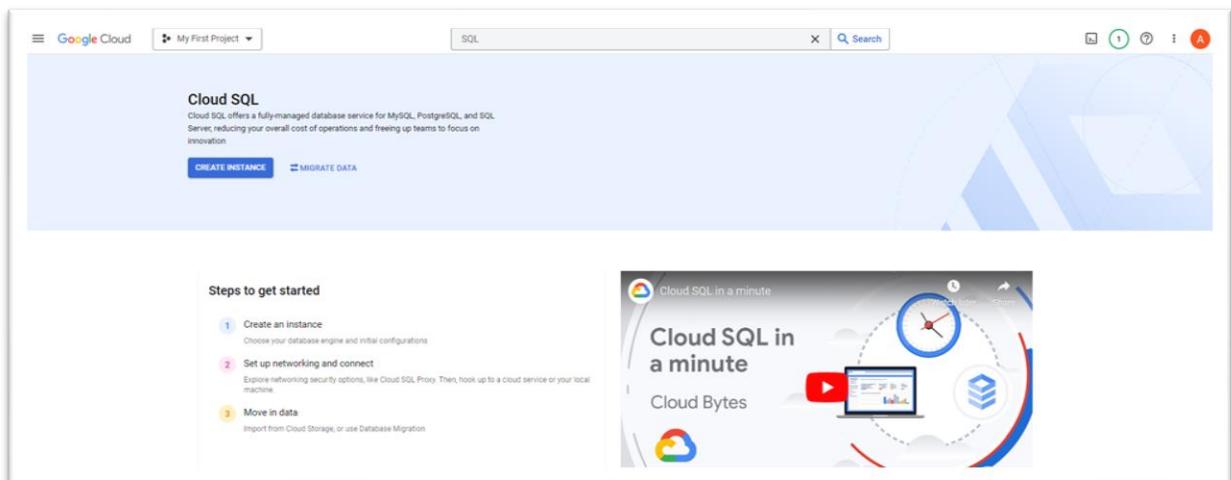
Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption
end_station_name.csv	33.8 KB	text/csv	Jun 22, 2023, 7:43:42 PM	Standard	Jun 22, 2023, 7:43:42 PM	Not public	—	Google-manage
start_station_name.csv	33.7 KB	text/csv	Jun 22, 2023, 7:43:42 PM	Standard	Jun 22, 2023, 7:43:42 PM	Not public	—	Google-manage

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

Create a Cloud SQL instance: Search and select SQL,



Create instance home page,



Follow the below instruction,

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

In the console, select **Navigation menu > SQL**.

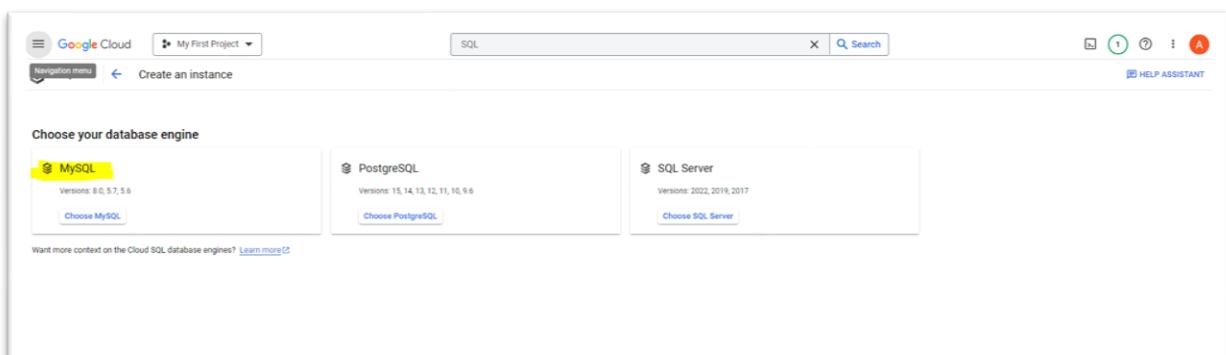
1. Click **CREATE INSTANCE > Choose MySQL**.
2. Enter instance id as **my-demo**.
3. Enter a secure password in the **Password** field (remember it!).
4. Select the database version as **MySQL 5.7**.
5. Set the **Multi zones (Highly available)** field as **<Lab Region>**
6. Click **CREATE INSTANCE**.

It might take a few minutes for the instance to be created. Once it is, you will see a green checkmark next to the instance name.

7. Click on the Cloud SQL instance. The **SQL Overview** page opens.

Screenshot of each step are as follows,

Click on the **CREATE INSTANCE** and choose **Choose MySQL**,



Fill out the form and click on Create,

AI Lab 2 (Assignment – 2)
Aloy Banerjee (CH22M503)
Date of Submission: 23-06-2023

Instance info

Instance ID * Use lowercase letters, numbers, and hyphens. Start with a letter.

Password * GENERATE

No password

PASSWORD POLICY

Database version *

Choose a configuration to start with

These suggested configurations will pre-fill this form as a starting point for creating an instance. You can customize as needed later.

Production
Optimized for the most critical workloads. Highly available, performant, and durable.

Development
Performant but not highly available, while reducing cost by provisioning less compute and storage.

CONFIGURATION DETAILS

Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region

Zonal availability

Single zone
In case of outage, no failover. Not recommended for production.

Multiple zones (Highly available)
Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost.

SPECIFY ZONES

Customize your instance

... (More options like Advanced settings, Firewall rules, etc.)

Summary

Region	us-central1 (Iowa)
DB Version	MySQL 5.7
vCPUs	4 vCPU
Memory	26 GB
Storage	100 GB
Network throughput (MB/s)	1,000 of 2,000
Disk throughput (MB/s)	Read: 48.0 of 240.0 Write: 48.0 of 240.0
IOPS	Read: 3,000 of 15,000 Write: 3,000 of 15,000
Connections	Public IP
Backup	Automated
Availability	Multiple zones (Highly available)
Point-in-time recovery	Enabled

SQL Instance will appear as below,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

The screenshot shows the Google Cloud SQL interface for a project named 'My First Project'. On the left, a navigation menu includes options like Overview, System insights, Query insights, Connections, Users, Databases, Backups, Replicas, and Operations. The main area displays a MySQL instance named 'my-demo' (MySQL 5.7). A message indicates that the instance is being created. Below this, there's a chart for CPU utilization, which shows no data available for the selected time frame. To the right, there's a 'Configuration' section detailing vCPUs (4), Memory (26 GB), and SSD storage (100 GB). A note states that the database version is MySQL 5.7.42 and auto storage increase is enabled.

New queries in Cloud SQL:

Open cloud shell terminal,

The screenshot shows the Google Cloud Platform dashboard for the same project. The dashboard features several cards: 'Project info' (Project name: My First Project, Project number: 891803523404, Project ID: modified-legacy-390504), 'SQL' (Storage used: 105MB), 'Google Cloud Platform status' (All services normal), 'Billing' (Estimated charges: \$0.00), 'Monitoring' (Create my dashboard, Set up alerting policies, Create uptime checks), and 'APIs' (Requests (requests/sec): 1.0). At the bottom, a terminal window titled 'CLOUD SHELL Terminal' is open, showing the command 'gcloud config set project modified-legacy-390504'. A blue banner at the bottom right encourages users to click for details about their Cloud Shell session and usage quota.

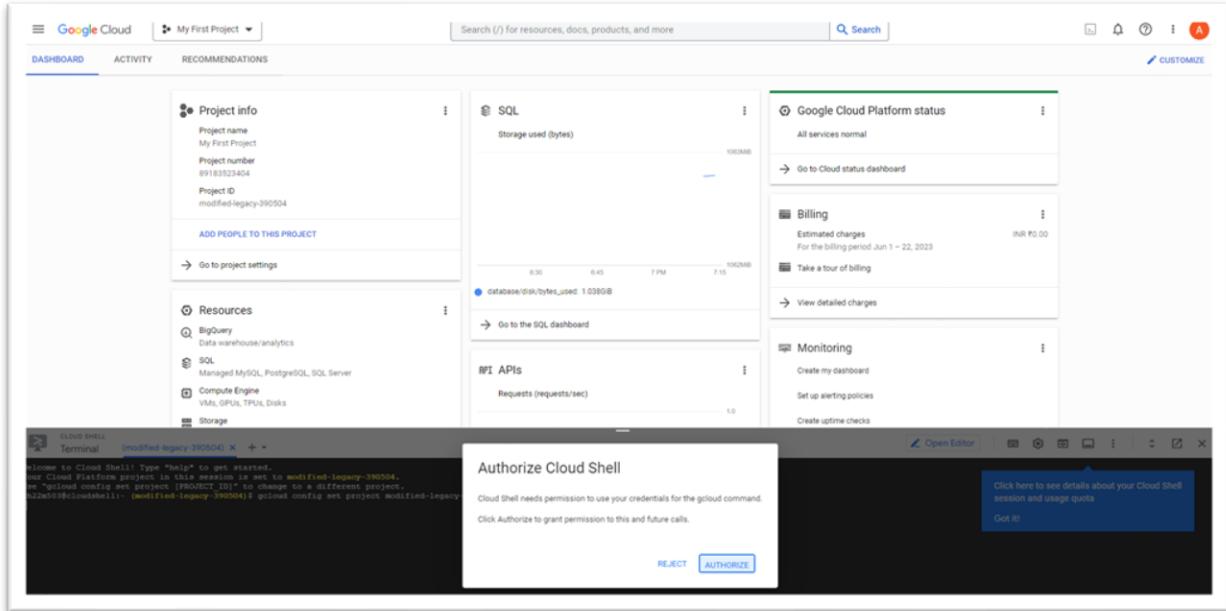
Set your project ID:

- ❖ `gcloud config set project modified-legacy-390504`

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



Click on Authorize button for further processing.

Shell script executed so far,

```
gcloud auth activate-service-account [REDACTED]@[REDACTED].iam.gserviceaccount.com --key-file=[REDACTED]
```

```
[REDACTED]@[REDACTED]:~$ gcloud auth activate-service-account [REDACTED]@[REDACTED].iam.gserviceaccount.com --key-file=[REDACTED]
```

```
[REDACTED]@[REDACTED]:~$ gcloud sql connect my-demo --user=root --quiet
```

```
[REDACTED]@[REDACTED]:~$ mysql -u root -p
```

```
[REDACTED]@[REDACTED]:~$ 
```

Now different MySQL command gets executed,

Database created along with necessary tables,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

Google Cloud My First Project Search (/) for resources, docs, products, and more Search

CLOUD SHELL Terminal (modified-legacy-390504) + +

Do you want to continue (Y/n)? Y
Go to the following link in your browser:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32559940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fcloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fuserinfo.email%2Fhttps%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatform%2Faccounts%2Fjohannawhaleonfire%2Faccounts%2Fregister&state=1fghahwahm0cPflex8kXkm071la&prompt=consent#access_type=offline&code_challenge=V1jQ3Xs7o7c-3w3jX3hSdtkbK1lvqjrs9z-2X-24&code_challenge_method=5256

After authorization code 0/ABaQJGVNCh0r2JhrlndUpXeA_0Bjek0-8pQhduj= was fetched from https://accounts.google.com/o/oauth2/token

You are now logged in as (oh2a503@mail.lttm.edu).
Our current project is (modified-legacy-390504). You can change this setting by running:
gcloud config set project modified-legacy-390504
Updated property [core/project].
oh2a503@cloudshell: ~ (modified-legacy-390504)\$ gcloud config set project modified-legacy-390504
Updated property [core/project].
oh2a503@cloudshell: ~ (modified-legacy-390504)\$ gcloud sql connect my-demo --user=root --quiet
(gcloud.sql.connect) unrecognized arguments: -quiet

To search the help text of gcloud commands, run:
gcloud help -- SEARCH TERMS

oh2a503@cloudshell: ~ (modified-legacy-390504)\$ gcloud sql connect my-demo --user=root --quiet
Cloud SQL Admin API has not been used in project 89183523404 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/sqladmin.googleapis.com/overview?project=89183523404
Cloud SQL Admin API needs to be enabled. If it is enabled for this API recently, wait a few minutes for the action to propagate to our systems and retry.

*Type: type.googleapis.com/google.rpc.Help

Link: https://cloud.google.com/sql/docs/mysql/enable-admin-api

Description: Google developers console API activation

Url: https://console.developers.google.com/apis/api/sqladmin.googleapis.com/overview?project=89183523404

*Type: type.googleapis.com/google.rpc.ErrorInfo

Link: https://cloud.google.com/sql/docs/mysql/error-messages

Metadata:

Consumer project: projects/89183523404

Region: us-central1

Region URL: https://us-central1.googleapis.com

Reason: SERVICE_DISABLED

oh2a503@cloudshell: ~ (modified-legacy-390504)\$ gcloud sql connect my-demo --user=root --quiet
Cloud SQL Admin API has not been used in project 89183523404 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/sqladmin.googleapis.com/overview?project=89183523404
Cloud SQL Admin API needs to be enabled. If it is enabled for this API recently, wait a few minutes for the action to propagate to our systems and retry.
Connecting to database with SQL user [root]. Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 141.
Server version: 5.7.42-google-log (Google)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE bike;

Query OK, 1 row affected (0.21 sec)

mysql> |

Upload data from CSV files to tables,

Uploading the Start Station Name in london1

Google Cloud My First Project Cloud mysql Select a file to import

Import data from Cloud Storage

Make sure all fields are correct to continue GO TO ISSUE(S)

Source

Choose a file to import from. Make sure you have read access first. [Learn more](#)

bucket-name/file-name * BROWSE

Required

File format

SQL A plain text file with a sequence of SQL commands, like the output of mysqldump

CSV If your Cloud Storage file is a CSV file, select CSV. The CSV file should be a plain text file with one line per row and comma-separated fields.

Destination

Choose the database and table in your instance for this file to import into. [Learn more](#)

Database * bike

Table * london1

Enter the name of an existing table in the database to house your CSV file

When you import, a Cloud SQL service account will be granted read access to the selected file and bucket, which will be reflected in your permissions.

IMPORT CANCEL

2 files successfully uploaded

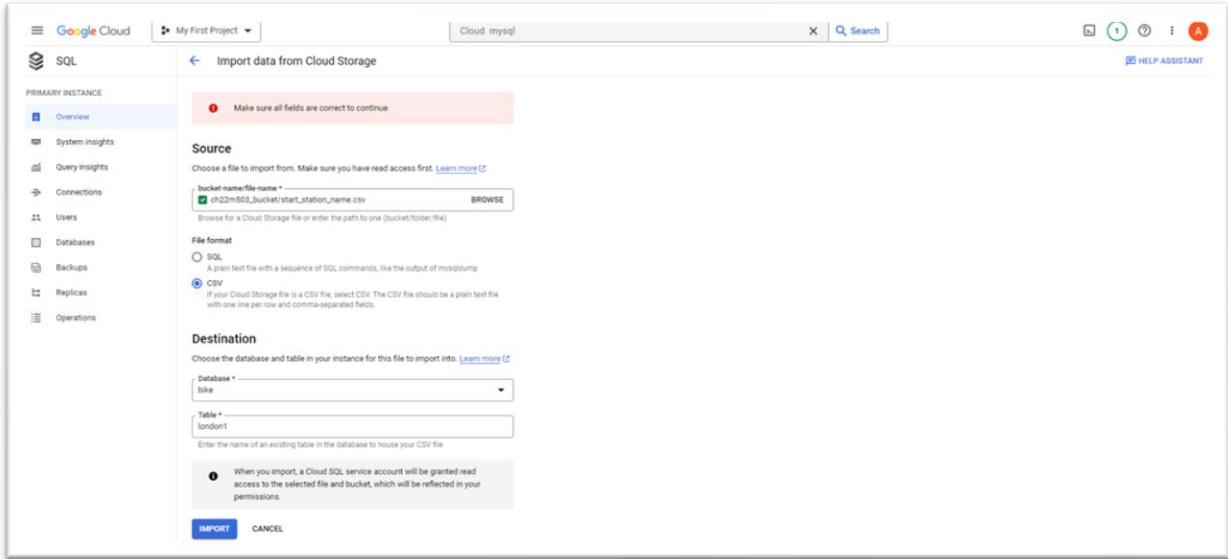
Select a SQL dump or CSV file to import
ch22m503_bucket
 end_station_name.csv
 start_station_name.csv

SELECT CANCEL

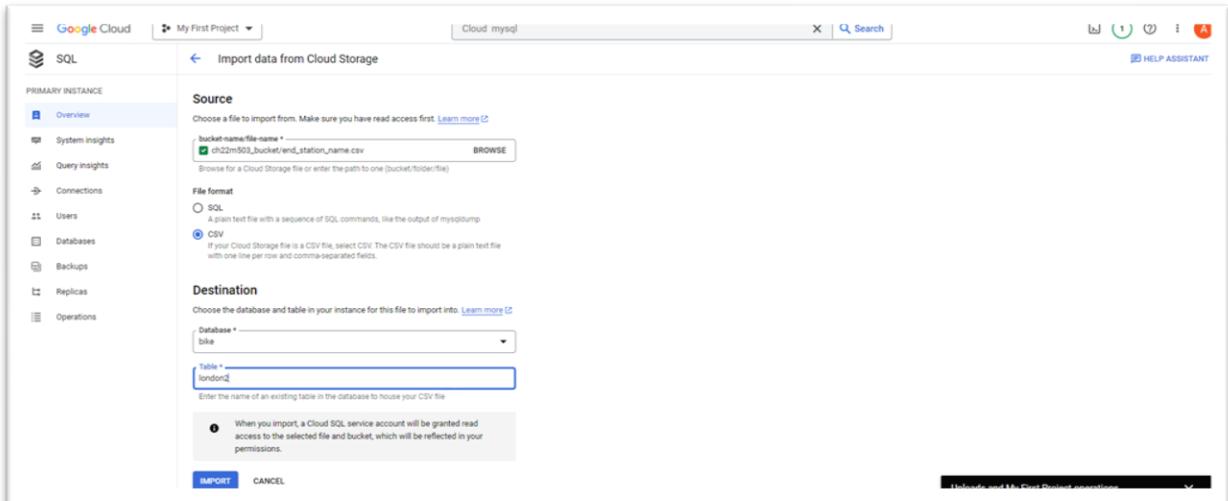
AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



Click on Import, Now again Uploading the End Station Name in london2



Now executing the remaining command,

Sharing the snapshot of the London 1 and London 2 tables data,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

```

Cloud Shell (modified-legacy-390504) + v
Terminal (modified-legacy-390504) + v
Search (/) for resources, docs, products, and more
Q Search
Open Editor
2 A
+ - X
You can turn off this feature to get a quicker startup with -A
atabase changed
mysql> SELECT * FROM london;
+-----+-----+
| start_station_name | num   |
+-----+-----+
| Hyde Park Corner, Hyde Park | 671688 |
| Belgrave Street , King's Cross | 593045 |
| Waterloo Station, Waterloo | 537329 |
| Albert Gate, Hyde Park | 441108 |
| Black Lion Gate, Kensington Gardens | 659716 |
| Grosvenor Square, Mayfair | 443300 |
| Wellington Arch, Hyde Park | 392689 |
| Hop Exchange, The Borough | 385526 |
| Duke of York Street, The Strand | 385523 |
| Triangle Car Park, Hyde Park | 353033 |
| Duke Street Hill, London Bridge | 334976 |
| Belgrave Square, Belgrave Street | 334975 |
| Bethnal Green Road, Hackney | 319191 |
| Aquatic Centre, Queen Elizabeth Olympic Park | 306440 |
| St. James's Gate, Westminster | 293148 |
| St. George's Gate, Westminster | 293148 |
| Park Lane , Hyde Park | 292819 |
| Grosvenor Square, Hyde Park | 292819 |
| Palace Gate, Kensington Gardens | 276525 |
| Tooley Street, Bermondsey | 269750 |
| Holborn Circus, Holborn | 268747 |
| Cheapside, Bank | 266016 |
| Tower Gardens , Tower | 266016 |
| Palmerston Circus, Liverpool Street | 253937 |
| Queen Street , Bank | 241950 |
| Whitehall Place, Strand | 258146 |
| Exhibition Road, Brixton | 255000 |
| Green Park Station, Mayfair | 253317 |
| Crosswall, Tower | 251807 |
| Waterloo Station , Waterloo | 250900 |
| Holborn Circus, Holborn | 244615 |
| Little Argyle Street, Shoreditch | 242851 |
| Little Russell Street, West End | 241043 |
| Vauxhall Bridge, Vauxhall | 241043 |
| Old Street Station, South Kensington | 239200 |
| South Kensington Station, South Kensington | 237045 |
| Grosvenor Square, Belgrave Street | 237045 |
| Newgate Street , St. Paul's | 234077 |
| Great Tower Street, Monument | 232475 |
| Holborn Circus, Holborn | 232475 |
| Green Park Station, Mayfair | 231770 |
| Baywater Road, Hyde Park | 231770 |
| Holborn, Holborn | 231770 |
| Holborn Circus, Holborn | 231770 |
| St. James's Square, St. James's | 221807 |
| Queen's Gate, Kensington Gardens | 221807 |
+-----+-----+
65 rows in set (0.42 sec)

```

```

Cloud Shell (modified-legacy-390504) + v
Terminal (modified-legacy-390504) + v
Search (/) for resources, docs, products, and more
Q Search
Open Editor
2 A
+ - X
CREATE TABLE `london` (
  `start_station_name` varchar(255) NOT NULL,
  `num` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
INSERT INTO `london` VALUES ('Hyde Park Corner, Hyde Park', 671680);
INSERT INTO `london` VALUES ('Belgrave Street , King\'s Cross', 593045);
INSERT INTO `london` VALUES ('Waterloo Station, Waterloo', 537329);
INSERT INTO `london` VALUES ('Albert Gate, Hyde Park', 441108);
INSERT INTO `london` VALUES ('Black Lion Gate, Kensington Gardens', 659716);
INSERT INTO `london` VALUES ('Grosvenor Square, Mayfair', 443300);
INSERT INTO `london` VALUES ('Wellington Arch, Hyde Park', 392689);
INSERT INTO `london` VALUES ('Hop Exchange, The Borough', 385526);
INSERT INTO `london` VALUES ('Duke of York Street, The Strand', 385523);
INSERT INTO `london` VALUES ('Triangle Car Park, Hyde Park', 353033);
INSERT INTO `london` VALUES ('Duke Street Hill, London Bridge', 334976);
INSERT INTO `london` VALUES ('Belgrave Square, Belgrave Street', 334975);
INSERT INTO `london` VALUES ('Bethnal Green Road, Hackney', 319191);
INSERT INTO `london` VALUES ('Aquatic Centre, Queen Elizabeth Olympic Park', 306440);
INSERT INTO `london` VALUES ('St. James\'s Gate, Westminster', 293148);
INSERT INTO `london` VALUES ('St. George\'s Gate, Westminster', 293148);
INSERT INTO `london` VALUES ('Park Lane , Hyde Park', 292819);
INSERT INTO `london` VALUES ('Grosvenor Square, Hyde Park', 292819);
INSERT INTO `london` VALUES ('Palace Gate, Kensington Gardens', 276525);
INSERT INTO `london` VALUES ('Tooley Street, Bermondsey', 269750);
INSERT INTO `london` VALUES ('Holborn Circus, Holborn', 268747);
INSERT INTO `london` VALUES ('Cheapside, Bank', 266016);
INSERT INTO `london` VALUES ('Tower Gardens , Tower', 266016);
INSERT INTO `london` VALUES ('Palmerston Circus, Liverpool Street', 253937);
INSERT INTO `london` VALUES ('Queen Street , Bank', 241950);
INSERT INTO `london` VALUES ('Whitehall Place, Strand', 258146);
INSERT INTO `london` VALUES ('Exhibition Road, Brixton', 255000);
INSERT INTO `london` VALUES ('Green Park Station, Mayfair', 253317);
INSERT INTO `london` VALUES ('Crosswall, Tower', 251807);
INSERT INTO `london` VALUES ('Waterloo Station , Waterloo', 250900);
INSERT INTO `london` VALUES ('Holborn Circus, Holborn', 244615);
INSERT INTO `london` VALUES ('Little Argyle Street, Shoreditch', 242851);
INSERT INTO `london` VALUES ('Little Russell Street, West End', 241043);
INSERT INTO `london` VALUES ('Vauxhall Bridge, Vauxhall', 241043);
INSERT INTO `london` VALUES ('Old Street Station, South Kensington', 239200);
INSERT INTO `london` VALUES ('South Kensington Station, South Kensington', 237045);
INSERT INTO `london` VALUES ('Grosvenor Square, Belgrave Street', 237045);
INSERT INTO `london` VALUES ('Newgate Street , St. Paul\'s', 234077);
INSERT INTO `london` VALUES ('Great Tower Street, Monument', 232475);
INSERT INTO `london` VALUES ('Holborn Circus, Holborn', 232475);
INSERT INTO `london` VALUES ('Green Park Station, Mayfair', 231770);
INSERT INTO `london` VALUES ('Baywater Road, Hyde Park', 231770);
INSERT INTO `london` VALUES ('Holborn, Holborn', 231770);
INSERT INTO `london` VALUES ('Holborn Circus, Holborn', 231770);
INSERT INTO `london` VALUES ('St. James\'s Square, St. James\'s', 221807);
INSERT INTO `london` VALUES ('Queen\'s Gate, Kensington Gardens', 221807);
+-----+-----+
65 rows in set (0.42 sec)

mysql> SELECT * FROM london2;
+-----+-----+
| end_station_name | num   |
+-----+-----+
| Hyde Park Corner, Hyde Park | 671680 |
| Belgrave Street , King's Cross | 584548 |
| Waterloo Station, Waterloo | 535029 |
| Albert Gate, Hyde Park | 447113 |
| Black Lion Gate, Kensington Gardens | 659716 |
| Waterloo Station 1, Waterloo | 403861 |
| Wellington Arch, Hyde Park | 388354 |
| Grosvenor Square, Mayfair | 388354 |
| Wormwood Street, Liverpool Street | 372718 |
| Triangle Car Park, Hyde Park | 349623 |
| Holborn Circus, Holborn | 349623 |
| Bethnal Green Road, Shoreditch | 322209 |
| St. George's Gate, Westminster | 318794 |
| Holborn Circus, Holborn | 314000 |
| Aquatic Centre, Queen Elizabeth Olympic Park | 309576 |
| Holborn Circus, Holborn | 308103 |
| Queen Street , Bank | 307325 |
| Tooley Street, Bermondsey | 300025 |
| St. James's Square, St. James's | 294442 |
| Queen Street , Liverpool Street | 277824 |
| Whitehall Place, Strand | 277824 |
| Cheapside, Bank | 284615 |
| Tower Gardens , Tower | 284615 |
| Park Lane , Hyde Park | 282455 |
| Serpentine Car Park, Hyde Park | 281158 |
| Grosvenor Square, Belgrave Street | 279329 |
| Newgate Street , St. Paul's | 278570 |
| Queen Street 1, Bank | 277824 |
| Holborn Circus, Holborn | 277824 |
| Green Park Station, Mayfair | 270166 |
| Crosswall, Tower | 247956 |
| Little Argyle Street, West End | 244628 |
| William IV Street, Strand | 257283 |
| Great Tower Street, Monument | 252243 |
| Holborn Circus, Holborn | 252243 |
| South Kensington Station, South Kensington | 246288 |
| Shoreditch High Street, Shoreditch | 245305 |
| Vauxhall Bridge, Vauxhall | 243355 |
| Queen Street 2, Holborn | 242900 |
| Kennington Lane Rail Bridge, Vauxhall | 240417 |
| Queen's Gate, Kensington Gardens | 233355 |
+-----+-----+
65 rows in set (0.42 sec)

```

DELETE keyword:

Delete the first row,

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023

```
Google Cloud My First Project Search ( ) for resources, docs, products, and more Search Terminal (modified-legacy-390504) + + + 1044
| Bromley Street, Hackney
| Bromley High Street, Bromley
| Upper Richmond Road, East Putney
| Upper Richmond Road, West Putney
| Harrington Square, Camden Town
| Ledbrooke Grove Central, Notting Hill
| Union May
| Lansdowne Mews, Notting Hill
| St. Mary and St. Michael Church, Stepney
| Colmore Row, Birmingham
| Halford Road, Fulham
| Westfield Eakers Access Road, Shepherd's Bush
| Lansdowne Drive, Hackney Central
| Spanish Road, Walthamstow
| Spanish Road, Wandsworth
| Stewart's Road, Wim Elms
| Lansdowne Drive, Hackney
| Ingrove Street, Battersea
| Lansdowne Drive, Hackney Central
| Lansdowne Drive, Hackney Central
| Lansdowne Drive, Hackney Central
| Chiswick Street, South Bank
| Queenaway, Paddington
| Duxbury Street
| Limburg Road, Clapham Common
| St John's Park, Cabbitt Town
| Lansdowne Drive, Southbury House
| Coborne Street, Mile End
| Exhibition Road, South Kensington
| Lansdowne Drive, Hackney Central
| Crimcoot Street, Bermondsey
| Pop Up Book 3
| Lansdowne Drive, Walthamstow
| Carter Circle, Canary Wharf
| Pop Up Book 2
| Coborne Street, Southbury House
| Allington street, Off Victoria Street, Westminster
| Monier Road, Newham
| Kynock Road
| LSW2
| Blackfriars road, Southwark
| Gt London
| LSW1
| PENTON STREET COMM TEST TERMINAL - CONTACT MATT McNALLY
| York May, Camden
555 rows in set (0.41 sec)

mysql> DELETE FROM london1 WHERE num=0;
Query OK, 1 row affected (0.21 sec)

mysql> DELETE FROM london2 WHERE num=0;
Query OK, 1 row affected (0.21 sec)

mysql>
```

INSERT INTO keyword

```
Google Cloud My First Project Search ( ) for resources, docs, products, and more Search Terminal (modified-legacy-390504) + + + ch22m503@cloudshell: ~ (modified-legacy-390504)$ mysql -u root --password
Allowing your IP for incoming connection for 5 minutes...done.
Connecting to database with SGT user [root].Enter password:
Your MySQL connection id is 495
Server version: 5.7.42-google-log (Google)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help' or 'h' for help. Type '\c' to clear the current input statement.

mysql> USE bike;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with '-A'

Database changed
mysql> INSERT INTO london1 (start_station_name, num) VALUES ('test destination', 1);
Query OK, 1 row affected (0.21 sec)

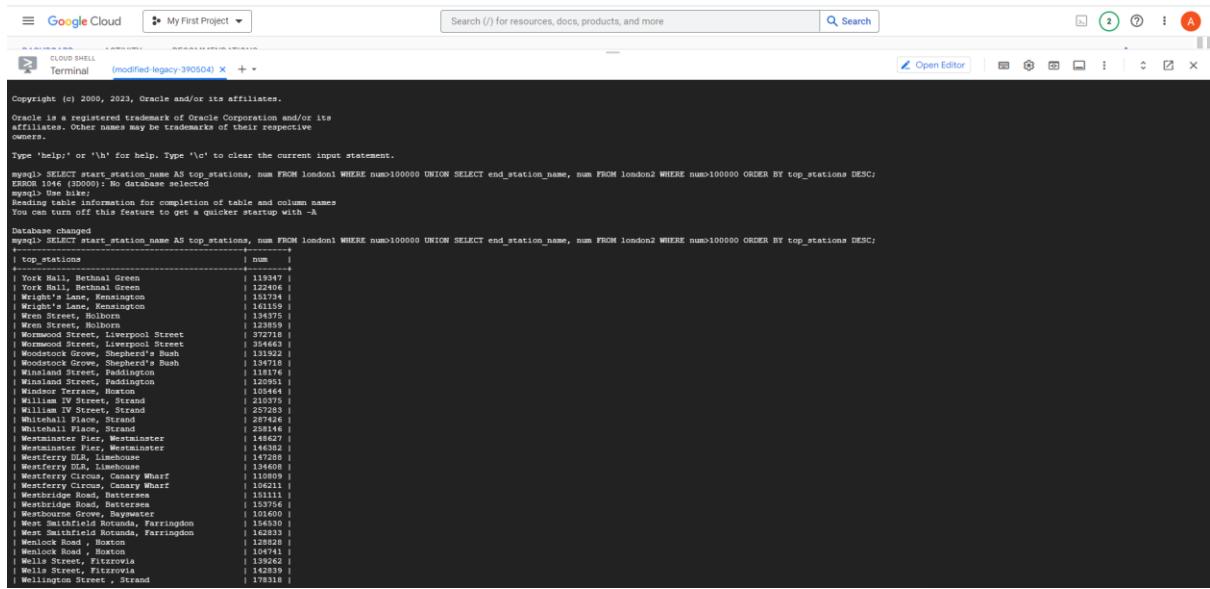
mysql>
```

UNION keyword

AI Lab 2 (Assignment – 2)

Aloy Banerjee (CH22M503)

Date of Submission: 23-06-2023



The screenshot shows a Google Cloud Terminal window titled '(modified-legacy-290504)'. The terminal displays a MySQL query results table. The table has two columns: 'top_stations' and 'num'. The data shows various locations in London with their respective counts.

top_stations	num
York Hall, Bethnal Green	13547
York Hall, Bethnal Green	122406
Wright's Lane, Kensington	151734
Westbourne Grove, Kensington	150559
West Street, Holborn	134975
West Street, Holborn	123899
Westgate Street, Liverpool Street	151748
Woodmood Street, Liverpool Street	314463
Woodstock Grove, Shepherd's Bush	131922
Woodstock Grove, Shepherd's Bush	131920
Miniland Street, Paddington	118176
Miniland Street, Paddington	120951
Whitefriars Street, Strand	11044
William IV Street, Strand	210379
William IV Street, Strand	257283
Whitchapel Road, Shoreditch	258146
Westminster Pier, Westminster	148627
Westferry DLR, Limehouse	147288
Westferry DLR, Limehouse	134608
Westferry Circus, Canary Wharf	13999
Westferry Circus, Canary Wharf	104211
Westbridge Road, Battersea	151111
Westbridge Road, Battersea	10446
West India Quay, Battersea	101600
West Smithfield Rotunda, Farringdon	154530
West Smithfield Rotunda, Farringdon	102893
Wellcock Road , Hoxton	128088
Wellcock Road , Hoxton	104741
Wellie Street, Fitzrovia	103926
Wellington Street, Fitzrovia	104733
Wellington Street , Strand	178318

Count of the record mentioned in each query by TA is matching.

End of Introduction to SQL for Big Query and Cloud SQL hands on