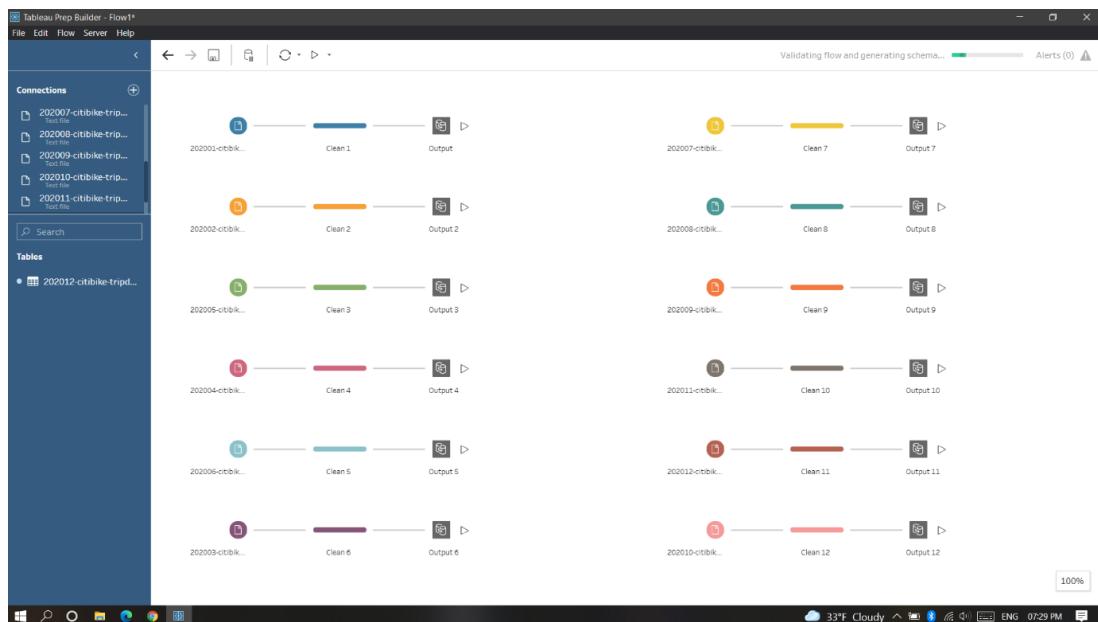
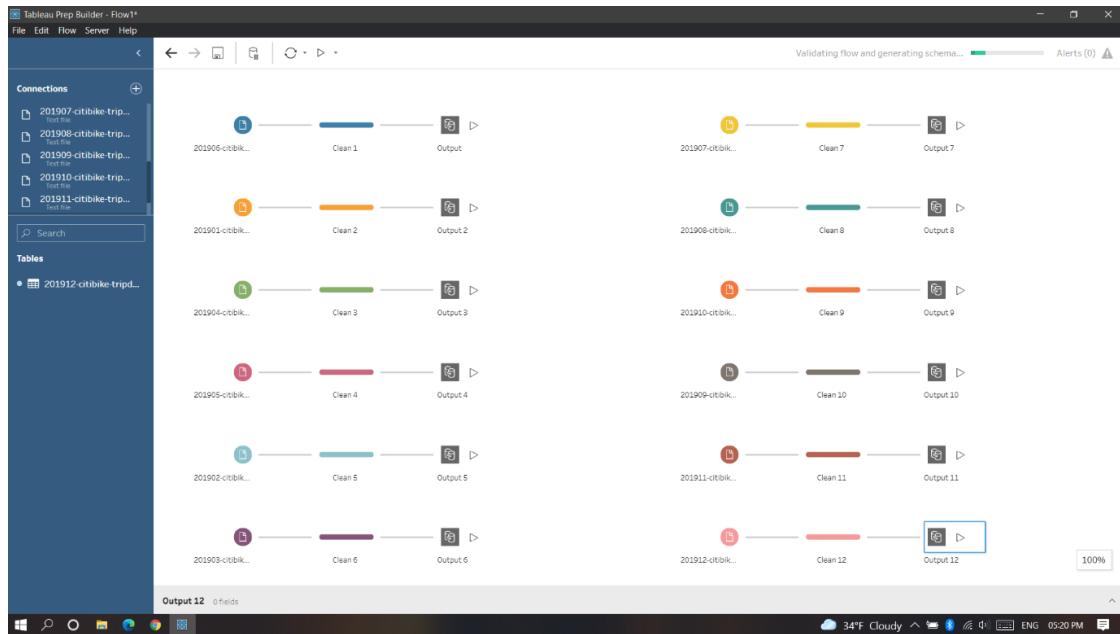


“Project – New York Citi Bike Analysis”

Karan Patil

- **Dataset 1 - Google Big Query Public Dataset (6/1/2013-5/31/2018)**
- **Dataset 2 - Google Big Query Project Folder (1/1/2014-12/31/2020)**
- **Dataset 3 - Oracle Cloud (1/1/2019-12/31/2020)**

- Loading 2019 and 2020 in Google Big Query Project Folder – Dataset 2



- Loading 2019 and 2020 in Oracle Cloud – Dataset 3

The screenshot shows a browser window with the URL http://g826b68bd00535b-nyccitibikes.adb.us-sanjose-1.oraclecloudapps.com/ords/admin/_sdw?nav=worksheet. The page title is "SQL workspace - BigQuery - Project". The main content is a "Database Actions | SQL" worksheet titled "[Worksheet]*". It contains a code editor with the following SQL query:

```
1 select * from NYC_CITIBIKES_2019_2020;
```

The results pane shows a table with 100 rows fetched. The columns are tripduration, starttime, stoptime, start_station_id, start_station_name, start_station_latitude, start_station_longitude, end_station_id, end_station_name, end_station_latitude, end_station_longitude, bikeid, usertype, birth_year, and gender. The first few rows of the table are:

	tripduration	starttime	stoptime	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bikeid	usertype	birth_year	gender
1	1833	2019-01-08T17:54:0	2019-01-08T18:24:0	319	Fulton St & Broadw	40.711	-74.009								
2	1327	2019-01-02T12:49:0	2019-01-02T13:11:0	3137	5 Ave & E 73 St	40.773	-73.967								
3	1336	2019-01-02T12:49:1	2019-01-02T13:11:2	3137	5 Ave & E 73 St	40.773	-73.967								
4	1343	2019-01-02T12:49:1	2019-01-02T13:11:2	3137	5 Ave & E 73 St	40.773	-73.967								
5	1361	2019-01-02T12:48:5	2019-01-02T13:11:2	3137	5 Ave & E 73 St	40.773	-73.967								
6	1886	2019-01-05T15:51:0	2019-01-05T16:23:0	3292	5 Ave & E 93 St	40.786	-73.957								
7	1244	2019-01-11T13:03:0	2019-01-11T13:23:0	491	E 24 St & Park Ave	40.741	-73.986								
8	1149	2019-01-08T18:08:1	2019-01-08T18:27:2	3360	Amsterdam Ave & 1	40.783	-73.979								
9	5539	2019-01-08T16:45:0	2019-01-08T18:17:2	3458	W 55 St & 6 Ave	40.763	-73.978								
10	1317	2019-01-09T05:58:0	2019-01-09T06:20:0	545	E 23 St & 1 Ave	40.737	-73.978								
11	641	2019-01-10T17:08:2	2019-01-10T17:19:2	435	W 21 St & 6 Ave	40.742	-73.994								

Execution time: 0.021 seconds. The status bar at the bottom left says "d4dd44 AM - 100 rows fetched... more to get". The taskbar at the bottom right shows the date as 12/14/2021.

- Total Row Count –

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Project, SQL, MySQL, PostgreSQL, Oracle, Oracle Database Actions, and BigTable. The current tab is "Project - NYC Citi Bike Share". The main area displays a query editor with the following SQL query:

```
1 SELECT count(*) as row_count FROM `bigquery-public-data.new_york_citibike.citibike_trips`;
```

The results pane shows the output of the query:

Row	row_count
1	58937715

The status bar at the bottom right shows the date as 12/14/2021.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'bigrquery-public-data', 'new_york_311', and 'new_york_citibike'. The main area shows a query editor with the following code:

```
1 SELECT count(*) as row_count FROM `bigrquery-public-data.new_york_citibike.citibike_stations`;
```

The 'Processing location: US' section indicates 'Query complete (0.3 sec elapsed, 0 B processed)'. The 'Job information' tab is selected, showing 'Results' (1 row, 1 column), 'JSON', and 'Execution details'. The results table shows:

Row	row_count
1	1584

At the bottom, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'. The taskbar at the bottom shows a file named '201701-citibike-trips.zip'.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share', 'damp_project', and 'bigrquery-public-data'. The main area shows a query editor with the following code:

```
1 SELECT count(*) as total_row_count FROM `project-nyc-citi-bike-share.damp_project.citibike_trips`;
```

The 'Processing location: US' section indicates 'Query complete (0.6 sec elapsed, 0 B processed)'. The 'Job information' tab is selected, showing 'Results' (1 row, 1 column), 'JSON', and 'Execution details'. The results table shows:

Row	total_row_count
1	109857282

At the bottom, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'. The taskbar at the bottom shows a file named '201701-citibike-trips.zip'.

The screenshot shows the Google Cloud Platform BigQuery interface. In the top navigation bar, there are several tabs including 'Project - NYC Citi Bike Share' and 'Google Cloud Platform'. The main area is titled 'EDITOR' with a sub-tab 'UNSANE... 2'. A query is being run:

```
1 SELECT count(*) as total_row_count FROM `project-nyc-citi-bike-share.damg.project.citibike_stations`;
```

The results section shows the following output:

Row	total_row_count
1	1584

Below the results, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

Dataset 3 –

The screenshot shows the Oracle Database Actions interface. The top navigation bar includes tabs for 'SQL workspace - BigQuery - Pro', 'Home | Oracle Cloud Infrastructure', and 'SQL | Oracle Database Actions'. The main area is titled 'ORACLE Database Actions | SQL'.

The left sidebar shows a 'Navigator' with 'ADMIN' selected, and a 'Tables' section listing various columns from the 'NYC_CITIBIKES_2019_2020' table, such as TRIPDURATION, STARTTIME, STOPTIME, START_STATION_ID, START_STATION_NAME, START_STATION_LATITUDE, START_STATION_LONGITUDE, END_STATION_ID, END_STATION_NAME, END_STATION_LATITUDE, END_STATION_LONGITUDE, BIKEID, USERTYPE, BIRTH_YEAR, and GENDER.

The central workspace contains a query editor with the following SQL statement:

```
1 select count(*) from NYC_CITIBIKES_2019_2020;
```

The results section shows the following output:

count(*)	
1	20858413

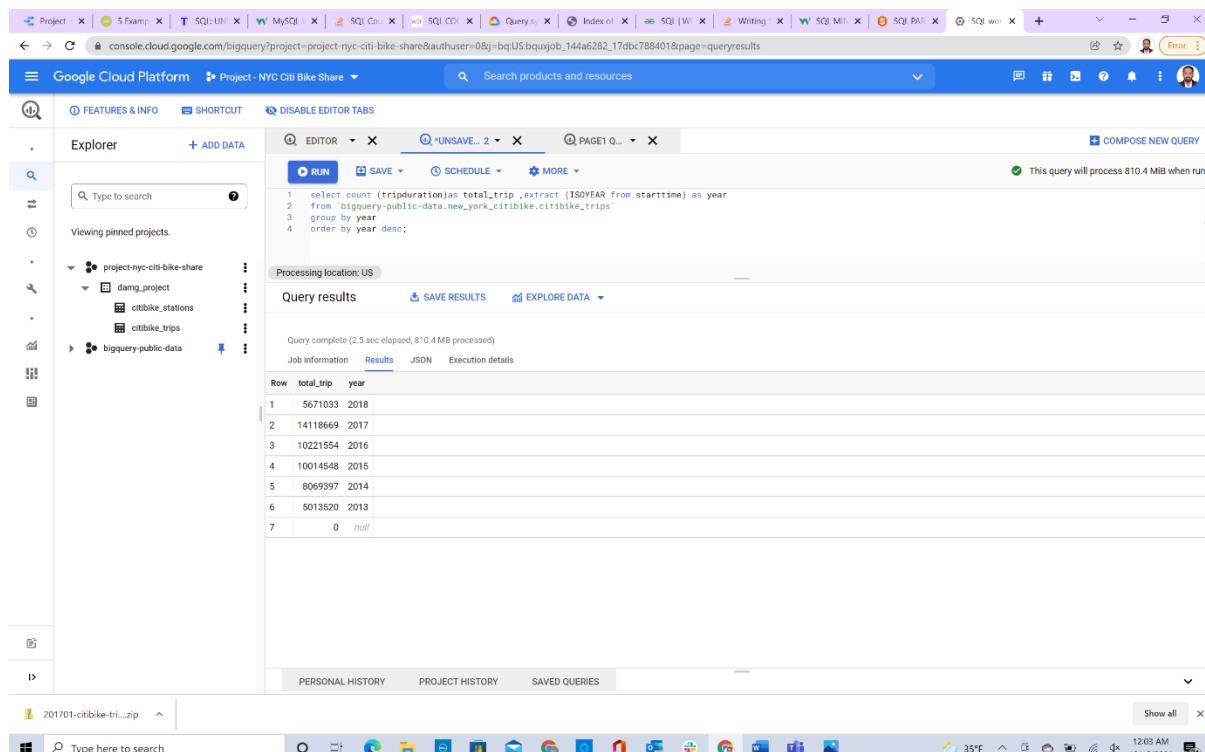
Below the results, there are tabs for 'Query Result', 'Script Output', 'DBMS Output', 'Explain Plan', 'Autotrace', 'SQL History', and 'Data Loading'.

SQL Queries and Result Sets:

The number of bike-sharing trips is a typical example of a time series. It has the following components:

- Trend: overall year-over-year growth

Dataset 1 –



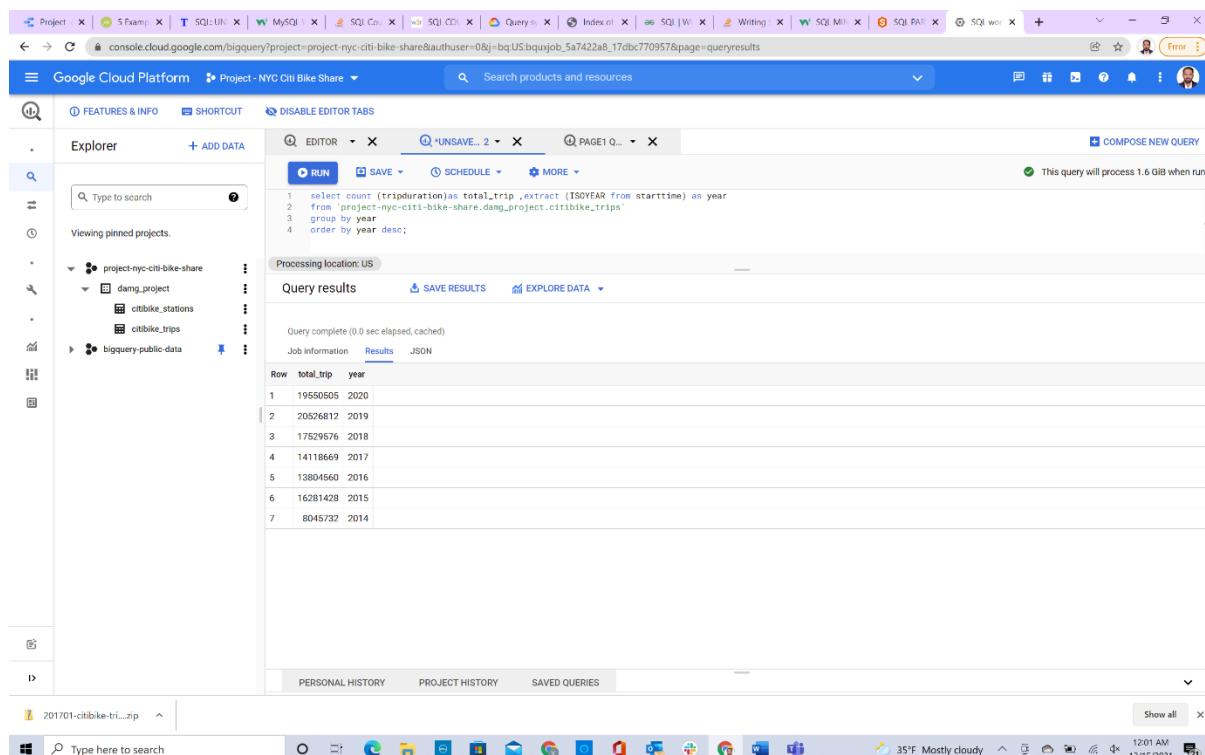
A screenshot of the Google Cloud Platform BigQuery interface. The left sidebar shows a project named "Project - NYC Citi Bike Share" with datasets "damg_project" and "bigquery-public-data". The main area displays a query in the editor:

```
1 select count(tripduration)as total_trip ,extract (ISOYEAR from starttime) as year
2   from `bigquery-public-data.new_york_citibike.citibike_trips`
3   group by year
4   order by year desc;
```

The results table shows the total number of trips per year from 2013 to 2018.

Row	total_trip	year
1	5671033	2018
2	14118669	2017
3	10221554	2016
4	10014548	2015
5	8069397	2014
6	5013520	2013
7	0	null

Dataset 2 –



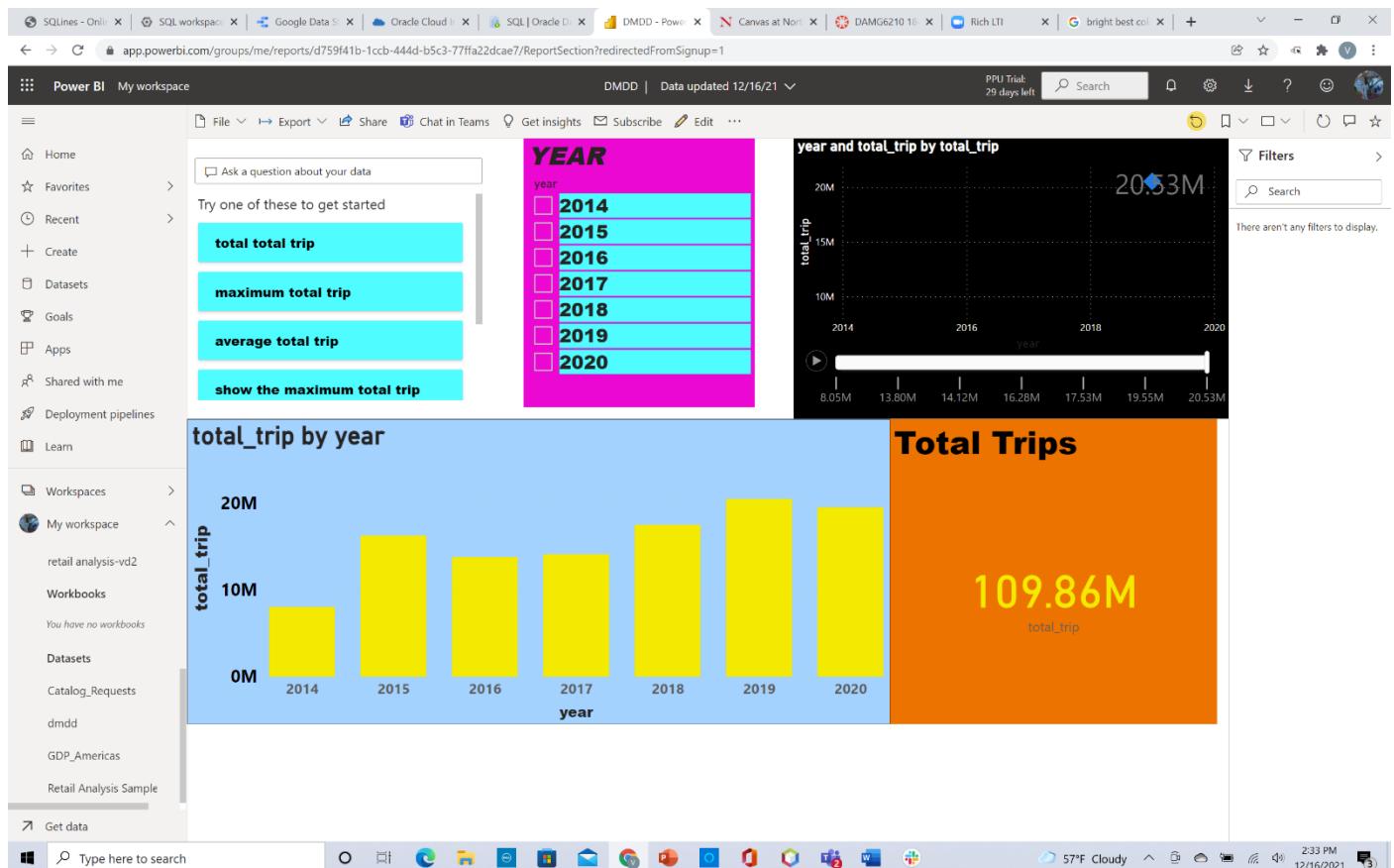
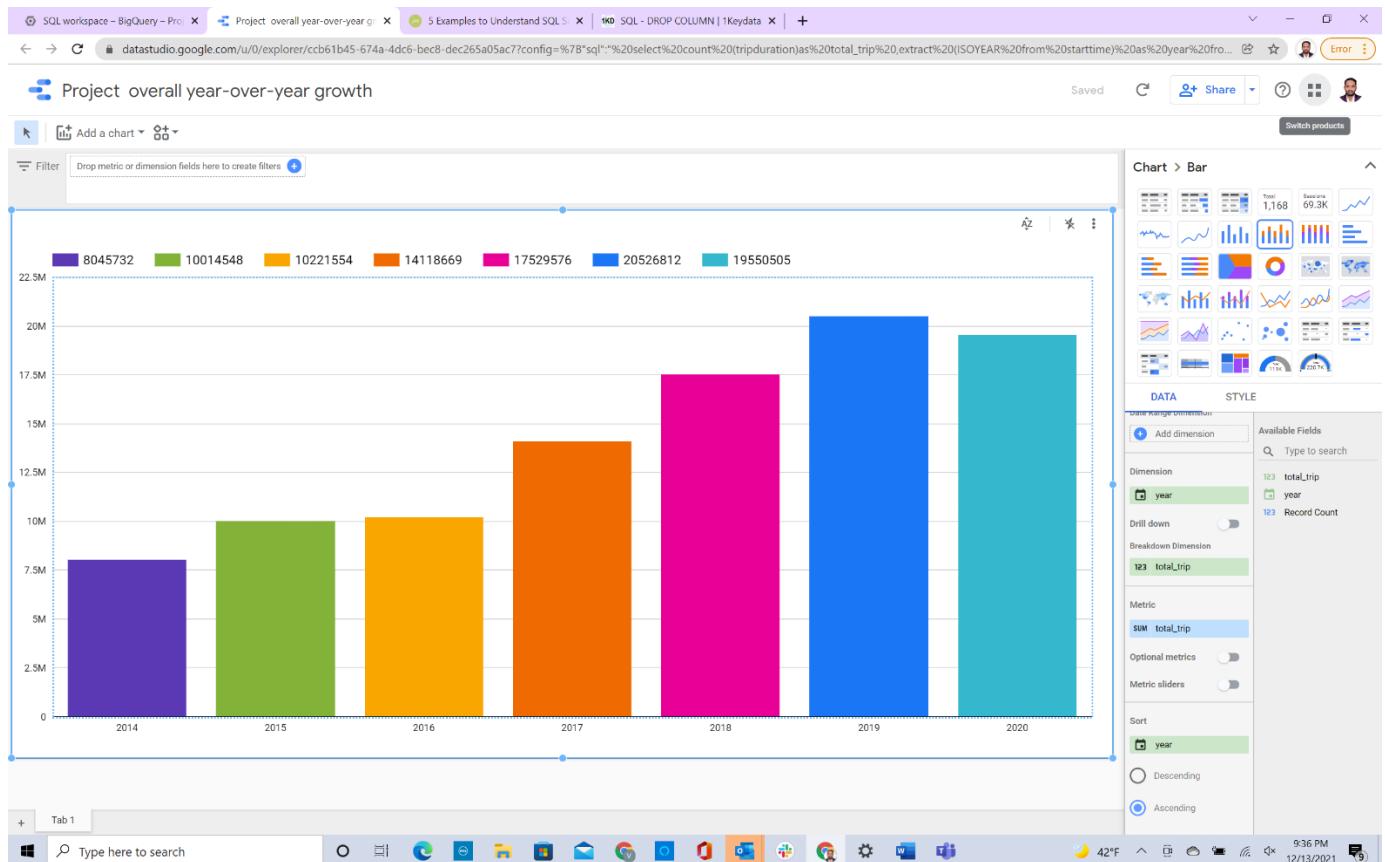
A screenshot of the Google Cloud Platform BigQuery interface, identical to the previous one but with a different dataset selected. The left sidebar shows the same project and datasets. The main area displays a query in the editor:

```
1 select count(tripduration)as total_trip ,extract (ISOYEAR from starttime) as year
2   from `project-nyc-citi-bike-share.damg_project.citibike_trips`
3   group by year
4   order by year desc;
```

The results table shows the total number of trips per year from 2014 to 2020.

Row	total_trip	year
1	19550505	2020
2	20526812	2019
3	17529576	2018
4	14118669	2017
5	13804560	2016
6	16281428	2015
7	8045732	2014

Data Visualization with Google Data Studio -



Dataset 3 –

The screenshot shows an Oracle Database Actions SQL Worksheet. The query is:

```
1 select count(tripduration) as total_trip_count ,extract(YEAR from starttime) as Years
2 from NYC_CITIBIKES_2019_2020
3 group by extract(YEAR from starttime)
4 order by extract(YEAR from starttime) desc;
```

The results table shows two rows:

	total_trip_count	years
1	306716	2020
2	20551697	2019

Below the worksheet, the Windows taskbar shows the query took 3.554 seconds and was run at 10:42 AM on 12/15/2021.

- Seasonality: users make more trips in the summer months and less in the winter.

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query is:

```
1 WITH temporary AS(
2   SELECT (count(tripduration))as total_trips, struct ("summer" as season)
3   from bigquery-public-data.new_york.citibike.citibike_trips
4   where FORMAT_TIMESTAMP("%b",starttime) in("Jun","Jul","Aug")
5   Union All(
6   SELECT (count(tripduration))as total_trips,struct ("winter" as season)
7   from bigquery-public-data.new_york.citibike.citibike_trips
8   where FORMAT_TIMESTAMP("%b",starttime) in("Dec","Jan","Feb"))
9 select * from temporary;
```

The results table shows two rows:

Row	total_trips	f0_season
1	17602306	summer
2	6177205	winter

Below the results, the Windows taskbar shows the query completed in 3.3 seconds and was run at 10:45 AM on 12/15/2021.

Dataset 2 -

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the Explorer sidebar displays pinned projects, including 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

```
1 WITH temporary AS(
2   SELECT (count(tripduration))as total_trips, struct ('summer' as season)
3   | from `project-nyc-citi-bike-share.damg.project.citibike_trips`
4   | where FORMAT_TIMESTAMP("%b",starttime) in('Jun','Jul','Aug')
5   Union All(
6   | SELECT (count(tripduration))as total_trips,struct ('winter' as season)
7   | from `project-nyc-citi-bike-share.damg.project.citibike_trips`
8   | where FORMAT_TIMESTAMP("%b",starttime) in('Dec','Jan','Feb'))
9 select * from temporary;
```

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

Row	total_trips	f0_season
1	14905587	winter
2	36595985	summer

The status bar at the bottom indicates the file '201701-citibike-tri...zip' is open, along with a search bar and system tray.

Data Visualization with Google Data Studio -

The screenshot shows a Google Data Studio visualization titled 'Seasonality'. The chart displays the 'Record Count' and 'total_trips' over time, categorized by season ('summer' and 'winter'). The Y-axis represents the count, ranging from 0 to 40M. The X-axis shows two distinct bars for 'summer' and 'winter'. The chart is styled with a light blue background and white grid lines. The right panel contains the 'Chart > Line' configuration interface, showing the selected dimensions and metrics.

Chart > Line

DATA

- Dimension: f0_season
- Metric: Record Count
- Metric: total_trips

STYLE

- Available Fields: Type to search
- Optional metrics: Record Count
- Metric sliders: Descending

Sort: SUM total_trips

Legend: Record Count (blue line), total_trips (teal bar)

Y-axis: 0, 5M, 10M, 15M, 20M, 25M, 30M, 35M, 40M

X-axis: summer, winter

The status bar at the bottom indicates the file '201701-citibike-tri...zip' is open, along with a search bar and system tray.

Dataset 3 –

The screenshot shows an Oracle Database Actions SQL Worksheet. The code queries the NYC_CITIBIKES_2019_2020 table to count trips by season (Summer, Winter) and displays the results in a table.

```
1 WITH temporary AS(
2   SELECT (count(tripduration))AS total_trips, 'summer' AS season
3   | FROM NYC_CITIBIKES_2019_2020
4   | WHERE TRIM(UPPER(TO_CHAR(starttime,'MON'))) IN('JUN','JUL','AUG')
5   UNION ALL
6   SELECT (count(tripduration))AS total_trips, 'winter' AS season
7   | FROM NYC_CITIBIKES_2019_2020
8   | WHERE TO_CHAR(starttime,'MON') IN('DEC','JAN','FEB'))
9   SELECT * FROM temporary;
```

	total_trips	season
1	6769526	summer
2	2926917	winter

▪ Cycles: weekly cycle where more trips happen on weekdays and less on weekends

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The code queries the bigquery-public-data.new_york_citibike.citibike_trips table to count trips by day of the week (Weekday, Weekend) and displays the results in a table.

```
1 WITH temporary AS(
2   SELECT (count(tripduration))AS total_trips, STRUCT ('Weekday' AS Days)
3   | FROM `bigquery-public-data.new_york_citibike.citibike_trips`
4   | WHERE FORMAT_TIMESTAMP("%A",starttime) IN('Monday','Tuesday','Wednesday','Thursday','Friday')
5   UNION ALL
6   SELECT (count(tripduration))AS total_trips, STRUCT ('weekend' AS Days)
7   | FROM `bigquery-public-data.new_york_citibike.citibike_trips`
8   | WHERE FORMAT_TIMESTAMP("%A",starttime) IN('Saturday','Sunday'))
9   SELECT * FROM temporary;
```

	total_trips	f0_Days
1	23692633	Weekday
2	12783514	weekend

Dataset 2 -

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the sidebar displays 'Explorer' and a list of projects including 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

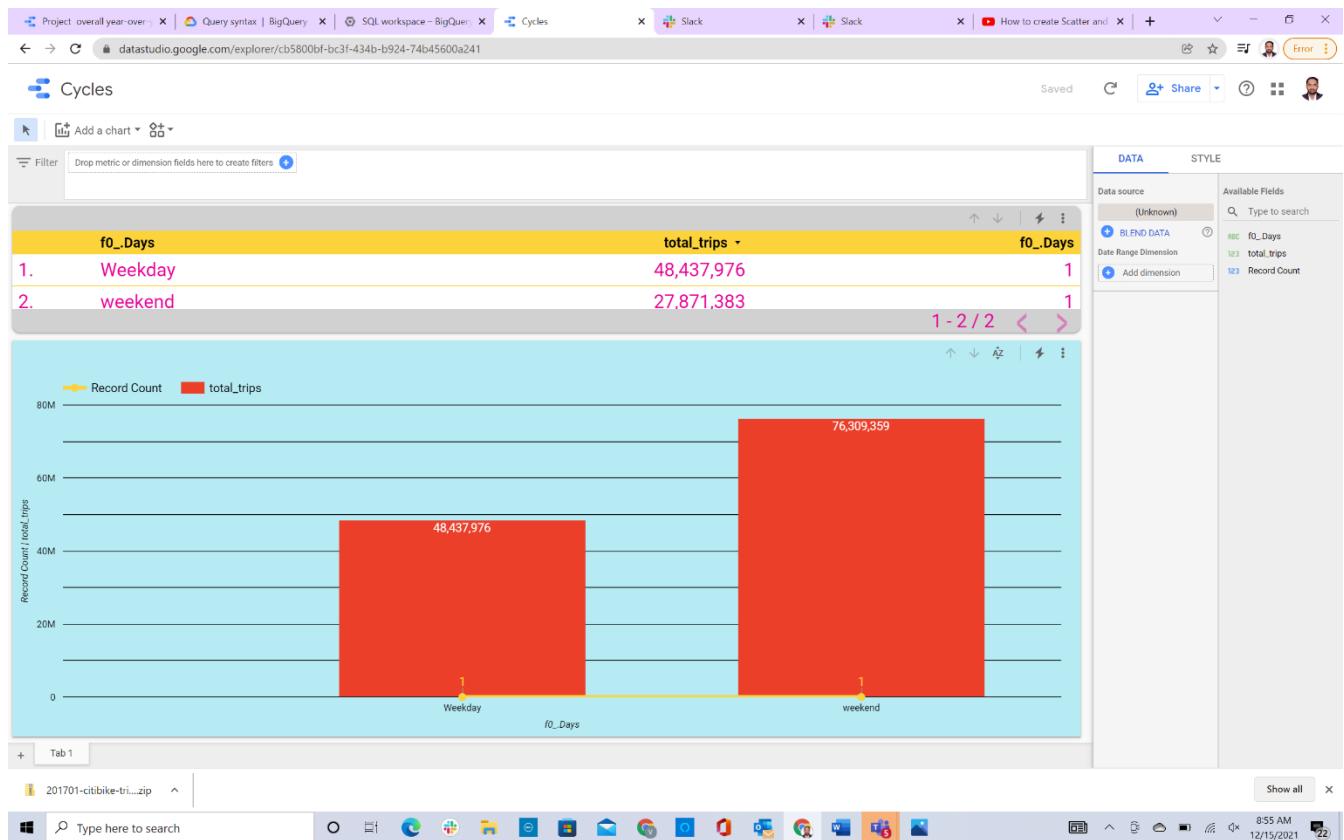
```
1 WITH temporary AS(
2     SELECT (count(tripduration))as total_trips, struct ('Weekday' as Days)
3     | from `project-nyc-citi-bike-share.damg.project.citibike_trips`
4     where FORMAT_TIMESTAMP("%A",starttime) in('Monday','Tuesday','Wednesday','Thursday','Friday')
5     Union All(
6     SELECT (count(tripduration))as total_trips,struct ('Weekend' as Days)
7     | from `project-nyc-citi-bike-share.damg.project.citibike_trips`
8     where FORMAT_TIMESTAMP("%A",starttime) in('Saturday','Sunday') )
9     select * from temporary;
```

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

Row	total_trips	f0_Days
1	27871383	weekend
2	48437976	Weekday

At the bottom, there is a file icon labeled '201701-citibike-trips.zip' and a taskbar with various application icons.

Data Visualization with Google Data Studio -



Dataset 3 –

The screenshot shows the Oracle Database Actions interface. On the left, the Navigator pane lists several worksheets, including pg 24, Pg30, 29 A, pg 27, pg 26, and Pg 25. The main area displays two SQL queries in the SQL worksheet:

```
1 SELECT count (tripduration) as total_trips,
2   'Weekday' as Days
3   from NYC_CITIBIKES_2019_2020
4  where TRIM(upper(to_char(starttime,'Day'))) in ('MONDAY','TUESDAY','WEDNESDAY','THURSDAY','FRIDAY')
5  Union ALL
6 SELECT count (tripduration) as total_trips, 'weekend' as Days from NYC_CITIBIKES_2019_2020
7  where TRIM(UPPER(to_char(starttime,'Day'))) in ('SATURDAY','SUNDAY');
```

The Query Result tab shows the execution time (6.309 seconds) and the resulting data:

	total_trips	days
1	15553640	Weekday
2	5304773	weekend

At the bottom, the taskbar shows a WhatsApp image file and the system status bar indicates 11:10 AM on 12/15/2021.

- How many bike trips and what is the trend over time
 - calendar year (2017)

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The sidebar shows pinned projects: project-nyc-citi-bike-share, danc.project, and bigquery-public-data. The main area displays a query in the Editor tab:

```
1 select count(*) as number_of_trips ,EXTRACT(MONTH from starttime) as months ,extract (ISOYEAR from starttime) as year
2   from `bigquery-public-data.new_york_citibike.citibike_trips`
3  group by months,year
4  having year =2017
5 ;
6
```

The Query results tab shows the execution details and the resulting data:

Row	number_of_trips	months	year
1	889967	12	2017
2	1735599	7	2017
3	1878098	9	2017
4	1523268	5	2017
5	1816498	8	2017
6	1330649	11	2017
7	1731594	6	2017
8	1315404	4	2017
9	1897592	10	2017

At the bottom, the taskbar shows a file named 201701-citibike-tri...zip and the system status bar indicates 12:13 AM on 12/15/2021.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the sidebar displays projects: 'project-nyc-citi-bike-share' (selected), 'dang.project' (with 'citibike_stations' and 'citibike_trips' datasets), and 'bigquery-public-data'. The main area shows a query editor with the following SQL code:

```
1 select count(*) as number_of_trips ,EXTRACT(MONTH from starttime) as months ,extract (ISOYEAR from starttime) as year
2 from `project-nyc-citi-bike-share.dang_project.citibike_trips`
3 group by months,year
4 having year =2017
5 ;
```

The 'Results' tab is selected, showing the following data:

Row	number_of_trips	months	year
1	1731594	6	2017
2	1330649	11	2017
3	1878098	9	2017
4	889967	12	2017
5	1523268	5	2017
6	1315404	4	2017
7	1816498	8	2017
8	1897592	10	2017
9	1735599	7	2017

At the bottom, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'. The system status bar at the bottom right shows '12:12 AM 12/15/2021'.

Dataset 3 – (2019)

The screenshot shows the Oracle Database Actions SQL worksheet interface. On the left, the Navigator pane shows the schema structure for 'NYC_CITIBIKES_2019_2020' table, including columns like TRIPDURATION, STARTTIME, STOPTIME, START_STATION_ID, START_STATION_NAME, START_STATION_LATITUDE, START_STATION_LONGITUDE, END_STATION_ID, END_STATION_NAME, END_STATION_LATITUDE, END_STATION_LONGITUDE, BIKEID, USERTYPE, BIRTH_YEAR, and GENDER.

The main area shows a query editor with the following SQL code:

```
1 select count(*) as number_of_trips,MONTHS,YEAR FROM (SELECT TO_CHAR(starttime,'MON') as months ,TO_CHAR (starttime,'YYYY') as year
2 from NYC_CITIBIKES_2019_2020)
3 group by months,year
4 having year =2019
5 ;
6
7
8
9
```

The 'Query Result' tab is selected, showing the following data:

number_of_trips	months	year
967287	JAN	2019
1924563	MAY	2019
2125370	JUN	2019
2444900	SEP	2019
943744	FEB	2019
1327960	MAR	2019
955210	DEC	2019
2181064	JUL	2019
2344224	AUG	2019
1766094	APR	2019
1478708	NOV	2019
2092573	OCT	2019

At the bottom, there are tabs for 'Script Output', 'DBMS Output', 'Explain Plan', 'Autotrace', 'SQL History', and 'Data Loading'. The system status bar at the bottom right shows '30°F Partly cloudy 5:05 AM 12/15/2021'.

• Calendar month & year (2017-09)

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the sidebar displays the project structure: 'project-nyc-citi-bike-share' contains 'damg_project' which includes 'citibike_stations' and 'citibike_trips', and also has a reference to 'bigquery-public-data'. The main area shows a query editor with the following SQL code:

```

1 select count(*) as number_of_trips ,EXTRACT(Day from starttime) as day,EXTRACT(MONTH from starttime) as months ,extract (ISOYEAR from starttime) as year
2 from `bigquery-public-data.new_york.citibike.citibike_trips`
3 group by day ,months,year
4 having year =2017 and months = 9
5 order by day;

```

The results table shows the following data for September 2017:

Row	number_of_trips	day	months	year
1	54881	1	9	2017
2	37812	2	9	2017
3	30955	3	9	2017
4	55285	4	9	2017
5	65445	5	9	2017
6	42295	6	9	2017
7	69519	7	9	2017
8	69574	8	9	2017
9	60155	9	9	2017
10	60711	10	9	2017
11	71917	11	9	2017
12	74580	12	9	2017

At the bottom, there is a message: 'Syntax error: Expected end of input but got identifier 'bigquery-public-data.new_york.citibike.citibike_trips' at [16:1]'.

Dataset 2 –

This screenshot is identical to the one above, showing the same query and results for September 2017 bike trips. The only difference is a message at the top right of the query editor: 'This query will process 838.1 MiB when run.'

Dataset 3 – (2019-09)

The screenshot shows the Oracle Database Actions interface. On the left, the Navigator pane displays the schema structure for the 'NYC_CITIBIKES_2019_2020' table, including columns like TRIPDURATION, STARTTIME, STOPTIME, START_STATION_ID, START_STATION_NAME, START_STATION_LATITUDE, START_STATION_LONGITUDE, END_STATION_ID, END_STATION_NAME, END_STATION_LATITUDE, END_STATION_LONGITUDE, BIKEID, USERTYPE, BIRTH_YEAR, and GENDER. The main workspace contains a SQL worksheet with the following query:

```

1 select count(*) as number_of_trips,MONTHS,DAY,YEAR FROM (SELECT TO_CHAR(starttime,'Day') as day, TO_CHAR(starttime,'MM') as months ,TO_CHAR(starttime,'YYYY') as year
2 from NYC_CITIBIKES_2019_2020)
3 group by day ,months,year
4 having year =2019 and months = 9
5
6 order by day;
7

```

The results of the query are displayed in a table:

	number_of_trips	months	day	year
1	326663	09	Friday	2019
2	366748	09	Monday	2019
3	334036	09	Saturday	2019
4	366326	09	Sunday	2019
5	334063	09	Thursday	2019
6	357462	09	Tuesday	2019
7	359602	09	Wednesday	2019

At the bottom of the interface, there is a status bar showing '10:12:43 AM - 7 rows total'.

▪ calendar week (2017-36)

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The sidebar on the left shows the project structure under 'Google Cloud Platform' and 'Project - NYC Citi Bike Share'. The main area displays a query editor with the following SQL code:

```

1 select count(*) as number_of_trips ,FORMAT_TIMESTAMP("%A",starttime) AS day, EXTRACT(ISO_WEEK FROM starttime) as week,extract (ISOYEAR from starttime) as year
2 from `bigquery-public-data.new_york.citibike.citibike_trips`
3 group by day ,week,year
4 having year =2017 and week=36
5
6

```

The results of the query are shown in a table:

Row	number_of_trips	day	week	year
1	69574	Friday	36	2017
2	55285	Monday	36	2017
3	60155	Saturday	36	2017
4	60711	Sunday	36	2017
5	69519	Thursday	36	2017
6	65445	Tuesday	36	2017
7	42295	Wednesday	36	2017

The status bar at the bottom indicates '35°F Partly cloudy' and the date '12/15/2021'.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and datasets 'dang.project' containing 'citibike_stations' and 'citibike_trips'. The main area shows a query editor window with a query running. The query is:1 select count(*) as number_of_trips ,FORMAT_TIMESTAMP("%A",starttime) AS day, EXTRACT(ISO_WEEK FROM starttime) as week,extract (ISOYEAR from starttime) as year
2 from `project-nyc-citi-bike-share.dang_project.citibike_trips`
3 group by day ,week,year
4 having year =2017 and week=36
5 order by day;
6

The results table shows the following data:

Row	number_of_trips	day	week	year
1	69574	Friday	36	2017
2	55285	Monday	36	2017
3	60155	Saturday	36	2017
4	60711	Sunday	36	2017
5	69519	Thursday	36	2017
6	65445	Tuesday	36	2017
7	42295	Wednesday	36	2017

Dataset 3 – (2019-36)

The screenshot shows the Oracle Database Actions SQL worksheet interface. The left sidebar displays the 'Navigator' and 'Worksheets' sections, with 'ADMIN' selected. Under 'ADMIN', there is a tree view of tables: 'NYC_CITIBIKES_2019_2020' which includes 'TRIPDURATION', 'STARTTIME', 'STOPTIME', 'START_STATION_ID', 'START_STATION_NAME', 'START_STATION_LATITUDE', 'START_STATION_LONGITUDE', 'END_STATION_ID', 'END_STATION_NAME', 'END_STATION_LATITUDE', 'END_STATION_LONGITUDE', 'BIKEID', 'USERTYPE', 'BIRTH_YEAR', and 'GENDER'. The main workspace shows a query editor with the following SQL:1 select count(*) as number_of_trips,day,week,year from (select TO_CHAR(starttime,'Day') AS day, TO_CHAR(starttime,'WW') as week,TO_CHAR(starttime,'YYYY')as year
2 from NYC_CITIBIKES_2019_2020)
3 group by day ,week,year
4 having year =2019 and week=36
5 order by day;
6

The results table shows the following data:

Row	number_of_trips	day	week	year
1	55227	Friday	36	2019
2	84809	Monday	36	2019
3	78220	Saturday	36	2019
4	72515	Sunday	36	2019
5	89656	Thursday	36	2019
6	83717	Tuesday	36	2019
7	84833	Wednesday	36	2019

• calendar day (2017-09-01)

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays a project structure with pinned projects like 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

```
1 select count(*) as number_of_trip, starttime FROM `bigquery-public-data.new_york_citibike.citibike_trips`  
2 Group by starttime  
3 having date(starttime) = '2017-09-01';
```

The results table shows 35822 rows, each containing a 'number_of_trip' and a 'starttime' timestamp. The first few rows are:

Row	number_of_trip	starttime
1	9	2017-09-01T14:14:56
2	7	2017-09-01T08:57:51
3	7	2017-09-01T17:34:45
4	6	2017-09-01T11:48:20
5	6	2017-09-01T09:43:21
6	6	2017-09-01T15:29:03
7	6	2017-09-01T09:02:46
8	6	2017-09-01T13:39:04
9	7	2017-09-01T08:23:08
10	6	2017-09-01T10:16:48
11	6	2017-09-01T09:37:53
12	6	2017-09-01T08:35:37
13	6	2017-09-01T15:36:02

The interface includes a toolbar with 'RUN', 'SAVE', 'SCHEDULE', and 'MORE' buttons, and a status bar at the bottom showing 'Rows per page: 100' and '1 - 100 of 35822'.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface, similar to the previous one but with a different query. The left sidebar shows the same project structure. The main area shows a query editor with the following SQL code:

```
1 select count(*) as number_of_trip, starttime FROM `project-nyc-citi-bike-share.damg_project.citibike_trips`  
2 Group by starttime  
3 having date(starttime) = '2017-09-01';
```

The results table shows 35822 rows, each containing a 'number_of_trip' and a 'starttime' timestamp. The first few rows are:

Row	number_of_trip	starttime
1	6	2017-09-01 11:32:31 UTC
2	6	2017-09-01 12:18:08 UTC
3	6	2017-09-01 09:01:30 UTC
4	6	2017-09-01 18:06:43 UTC
5	7	2017-09-01 17:34:45 UTC
6	8	2017-09-01 18:35:27 UTC
7	6	2017-09-01 13:42:06 UTC
8	6	2017-09-01 08:15:44 UTC
9	6	2017-09-01 09:43:21 UTC
10	6	2017-09-01 19:12:24 UTC
11	6	2017-09-01 18:34:57 UTC
12	6	2017-09-01 15:16:17 UTC
13	6	2017-09-01 09:10:35 UTC

The interface includes a toolbar with 'RUN', 'SAVE', 'SCHEDULE', and 'MORE' buttons, and a status bar at the bottom showing 'Rows per page: 100' and '1 - 100 of 35822'.

Dataset 3 – (2019-09-01)

The screenshot shows an Oracle Database Actions | SQL worksheet. The query is:

```

1 select count(*) as number_of_trip, starttime FROM NYC_CITIBIKES_2019_2020
2 GROUP BY starttime
3 HAVING TO_CHAR(starttime,'YYYY-MM-DD') = '2019-09-01';
4
5
6
    
```

The results table has two columns: number_of_trip and starttime. The data is as follows:

number_of_trip	starttime
1	2019-09-01T14:00:31Z
2	2019-09-01T19:54Z
3	2019-09-01T12:26:35Z
4	2019-09-01T15:39:32Z
5	2019-09-01T10:26:46Z
6	2019-09-01T16:28:16Z
7	2019-09-01T13:34:56Z
8	2019-09-01T22:37:47Z
9	2019-09-01T21:03:03Z
10	2019-09-01T13:25:32Z
11	2019-09-01T12:12:49Z
12	2019-09-01T11:04:08Z
13	2019-09-01T11:37:31Z
14	2019-09-01T14:15:42Z
15	2019-09-01T11:20:06Z
16	2019-09-01T15:42:51Z
17	2019-09-01T16:07:26Z

Execution time: 0.021 seconds.

- Average number of bike trips and may also want to compare YOY (year over year) changes
 - per day, i.e., Sunday, Monday,...Saturday

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query is:

```

1 select (sum(tripduration)/count(*)) as average, FORMAT_TIMESTAMP("%A", starttime) as day, extract (ISOYEAR from starttime) as year
2 from `bigquery-public-data.new_york_citibike.citibike_trips`
3 group by day, year
4 order by day;
    
```

The results table has three columns: average, day, and year. The data is as follows:

Row	average	day	year
1	null	null	null
2	828.1128299016194	Friday	2014
3	946.3576718497793	Friday	2015
4	965.9011937804454	Friday	2016
5	944.011363483513	Friday	2018
6	1007.5117401993471	Friday	2017
7	865.6874440735363	Friday	2013
8	948.2169244973938	Monday	2018
9	919.329519641554	Monday	2015
10	935.4801387264304	Monday	2016
11	956.7474918193949	Monday	2017
12	828.1055947370571	Monday	2014
13	857.0713957558829	Monday	2013

Rows per page: 100 | 1 - 43 of 43 | First page | Last page

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the Explorer sidebar displays projects like 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

```
1 select (sum(tripduration)/count(*)) as average, FORMAT_TIMESTAMP("%A", starttime) as day, extract (ISOYEAR from starttime) as year
2 from `project-nyc-citi-bike-share.damg.project.citibike_trips`
3 group by day,year
4 order by day;
```

The processing location is set to US. Below the code, the 'Query results' section shows the output:

Row	average	day	year
1	1268.5441522798094	Friday	2020
2	963.4178006840664	Friday	2019
3	937.7438015082	Friday	2016
4	950.508284851324	Friday	2015
5	960.2207012810054	Friday	2018
6	1007.5117401993471	Friday	2017
7	828.1128299016194	Friday	2014
8	922.9213572845984	Monday	2015
9	1232.6284094206733	Monday	2020
10	949.6229150962627	Monday	2018
11	906.1984586622774	Monday	2016
12	935.1516233481327	Monday	2019
13	829.0594889826497	Monday	2014

At the bottom, there are buttons for 'SAVE RESULTS' and 'EXPLORE DATA'. The status bar indicates 'Rows per page: 100' and '1 - 49 of 49'.

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet interface. On the left, the Navigator pane shows the schema 'NYC_CITIBIKES_2019_2020' with various columns listed. The main workspace contains the following SQL query:

```
1 SELECT (SUM(TRIPDURATION)/COUNT(*)) as average,TO_CHAR(starttime,'Day') as day, TO_CHAR(starttime,'YYYY') as year
2 (SELECT TRIPDURATION,TO_CHAR(starttime,'Day') as day, TO_CHAR(starttime,'YYYY') as year
3 from NYC_CITIBIKES_2019_2020)
4 group by day,year
5 order by day;
```

The results table shows the following data:

Row	average	day	year
1	963.417	Friday	2019
2	1478.907	Friday	2020
3	934.401	Monday	2019
4	1378.467	Monday	2020
5	1183.528	Saturday	2019
6	1986.622	Saturday	2020
7	1144.887	Sunday	2019
8	1972.556	Sunday	2020
9	890.975	Thursday	2019
10	1605.564	Thursday	2020
11	893.956	Tuesday	2019
12	1434.484	Tuesday	2020
13	891.220	Wednesday	2019
14	1251.597	Wednesday	2020

The status bar at the bottom indicates '0 rows total'.

• weekday versus weekend

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'damg_project'. The main area contains a query editor window with the following SQL code:

```

1 WITH temporary as(
2   SELECT (sum(tripduration)/count(*))as total_trips_weekdays, extract (year from starttime) as year
3   from `bigquery-public-data.new_york_citibike.citibike_trips`
4   where FORMAT_TIMESTAMP("%A",starttime) in ('Monday','Tuesday','Wednesday','Thursday','Friday')
5   group by year
6   ),
7 temp as(
8   SELECT (sum(tripduration)/count(*))as total_trips_weekends,extract (year from starttime) as years
9   from `bigquery-public-data.new_york_citibike.citibike_trips`
10  where FORMAT_TIMESTAMP("%A",starttime) in ('Saturday','Sunday')
11  group by years )
12 select sum(total_trips_weekdays) as average_weekdays,sum( total_trips_weekends) as average_weekends, extract (year from starttime) as year from temporary,temp,'project-nyc-citi-bike-share'
13 group by year
14 ;
15

```

The 'Processing location: US' section indicates the job is complete. The 'Query results' section shows the following data:

Row	average_weekdays	average_weekends	year
1	4.48847009308921E11	5.59246014933511E11	2016
2	4.576975487521423E11	5.702734449483511E11	2017
3	6.662427840826503E11	8.30112742760723E11	2019
4	5.688002356028696E11	7.088027727435059E11	2018
5	5.2532711597571436E11	6.54537270056819E11	2015
6	6.323712692138864E11	7.87910145177379E11	2020
7	2.6197602303297372E11	3.264119930618092E11	2014

The bottom status bar shows '201701-citibike-tri...zip' and the system clock at '12:30 AM 12/15/2021'.

Dataset 2 –

This screenshot is identical to the one above, showing the same BigQuery interface, query code, and results. The data table is the same:

Row	average_weekdays	average_weekends	year
1	4.48847009308921E11	5.59246014933511E11	2016
2	4.576975487521423E11	5.702734449483511E11	2017
3	6.662427840826503E11	8.30112742760723E11	2019
4	5.688002356028696E11	7.088027727435059E11	2018
5	5.2532711597571436E11	6.54537270056819E11	2015
6	6.323712692138864E11	7.87910145177379E11	2020
7	2.6197602303297372E11	3.264119930618092E11	2014

The bottom status bar shows '201701-citibike-tri...zip' and the system clock at '12:29 AM 12/15/2021'.

Dataset 3 –

The screenshot shows an Oracle Database Actions SQL worksheet. The left sidebar displays a tree view of tables and columns for the 'NYC_CITIBIKES_2019_2020' dataset. The main area contains the following SQL code:

```
1 SELECT count (tripduration) as total_trips,
2       'Weekday' as Days
3   from NYC_CITIBIKES_2019_2020
4  where upper(to_char(starttime,'Day')) in('MONDAY','TUESDAY','WEDNESDAY','THURSDAY','FRIDAY')
5  Union All
6
7 SELECT count (tripduration) as total_trips, 'weekend' as Days from NYC_CITIBIKES_2019_2020
8  where TRIM(UPPER(to_char(starttime,'Day'))) in ('SATURDAY','SUNDAY');
```

The 'Query Result' tab shows the output:

	total_trips	days
1	3189036	Weekday
2	5304773	weekend

Execution time: 0.008 seconds

▪ month

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar shows pinned projects and the current project is 'Project - NYC Citi Bike Share'. The main area contains the following SQL query:

```
2 select sum(tripduration)/count(*) as average, FORMAT_TIMESTAMP("%B", starttime) as month, extract (ISOYEAR from starttime) as year
3   from `bigquery-public-data.new_york_citibike.citibike_trips`
4  group by month,year
5  order by month;
6
7
```

The 'Query results' section shows the output:

average	month	year
null	null	null
1074.3499860336435	April	2016
831.7082724589284	April	2014
929.8844816750717	April	2015
970.9073223595706	April	2018
1085.8632237700356	April	2017
1017.4786674627919	August	2015
884.4017783285539	August	2014
979.5802269156991	August	2013
1156.787824704450	August	2017
980.2857408823346	August	2016
937.4493734583598	December	2015

Rows per page: 100 | 1 - 54 of 54 | First page | Last page

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the sidebar displays the 'Explorer' section with projects like 'project-nyc-citi-bike-share' and datasets 'dang.project' containing 'citibike_stations' and 'citibike_trips'. The main area shows a query editor with the following SQL code:

```
1 select (sum(tripduration)/count(*)) as average, FORMAT_TIMESTAMP("%B", starttime) as month, extract (ISOYEAR from starttime) as year
2 from `project-nyc-citi-bike-share.dang_.project.citibike_trips`
3 group by month,year
4 order by month;
5
```

The 'Query results' tab is selected, displaying the output of the query. The results table has columns 'Row', 'average', 'month', and 'year'. The data shows monthly average trip durations from April 2015 to August 2020.

Row	average	month	year
1	1613.0203013641649	April	2020
2	1059.341500509033	April	2019
3	1074.3499860336435	April	2016
4	929.8844816750717	April	2015
5	1085.863223770035	April	2017
6	831.7082724589284	April	2014
7	970.9073223595706	April	2018
8	1012.0013301793416	August	2018
9	1050.1729045517834	August	2019
10	980.2857408823346	August	2016
11	1017.4786674627919	August	2015
12	1399.6322323025317	August	2020

At the bottom, there are navigation links for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet interface. The left sidebar displays the 'Navigator' and 'Worksheets' sections, with 'ADMIN' selected. Under 'ADMIN', the 'Tables' section lists tables such as 'NYC_CITIBIKES_2019_2020', 'TRIPDURATION', 'STARTTIME', etc. The main area shows a SQL query in the 'Worksheet' tab:

```
1 SELECT (SUM(TRIPDURATION)/COUNT(*)) as average,MONTH,YEAR FROM(SELECT TRIPDURATION,TO_CHAR(starttime,'MON') as MONTH, TO_CHAR(starttime,'YYYY') as year
2 from NYC_CITIBIKES_2019_2020)group by MONTH,year order by MONTH;
```

The 'Query Result' tab is selected, displaying the output of the query. The results table has columns 'average', 'month', and 'year'. The data shows monthly average trip durations from January 2019 to May 2020.

	average	month	year
1	1059.342	APR	2019
2	1803.698	APR	2020
3	1050.173	AUG	2019
4	1712.378	AUG	2020
5	830.840	DEC	2019
6	1387.734	DEC	2020
7	804.773	FEB	2019
8	665.693	FEB	2020
9	775.841	JAN	2019
10	540.898	JAN	2020
11	1035.360	JUL	2019
12	1841.722	JUL	2020
13	1109.359	JUN	2019
14	2296.587	JUN	2020
15	882.658	MAR	2019
16	1384.512	MAR	2020
17	1067.189	MAY	2019

At the bottom, there are navigation links for 'Script Output', 'DBMS Output', 'Explain Plan', 'Autotrace', 'SQL History', and 'Data Loading'.

• calendar week

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. A sidebar on the left lists projects, with 'project-nyc-citi-bike-share' selected. The main area displays a query editor with the following SQL code:

```

2
3 select (sum(tripduration)/count(*)) as average, EXTRACT(WEEK FROM starttime) AS week, extract (ISOYEAR from starttime) as year
4 from `bigquery-public-data.new_york_citibike.citibike_trips`
5 group by week,year
6 order by week desc ;
7
  
```

The results section shows a table with three columns: 'Row', 'average', 'week', and 'year'. The data consists of 12 rows of trip statistics. At the bottom, there are navigation links for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

At the bottom of the screen, a Windows taskbar is visible with various icons and a system tray showing the date and time as 12:28 AM 12/15/2021.

Dataset 2 –

This screenshot shows the same Google Cloud Platform BigQuery interface as the previous one, but with a different query. The SQL code is identical to the one in the first screenshot:

```

2
3 select (sum(tripduration)/count(*)) as average, EXTRACT(WEEK FROM starttime) AS week, extract (ISOYEAR from starttime) as year
4 from `project-nyc-citi-bike-share.damg_project.citibike_trips`
5 group by week,year
6 order by week desc ;
7
  
```

The results table shows 358 rows of data, starting with the first few rows:

Row	average	week	year
1	790.3461714111737	53	2017
2	903.809422222222	52	2015
3	752.3096701397225	52	2017
4	769.0114359932842	52	2014
5	729.4108609271523	52	2013
6	778.3953411208195	51	2017
7	818.782091134381	51	2013
8	1053.024280255654	51	2015
9	818.7592143629395	51	2014
10	753.7389956152039	50	2017
11	707.6031945975262	50	2014
12	767.2004245431858	50	2013

At the bottom, there are navigation links for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

At the bottom of the screen, a Windows taskbar is visible with various icons and a system tray showing the date and time as 12:26 AM 12/15/2021.

Dataset 3 –

The screenshot shows the Oracle Database Actions interface with a SQL worksheet open. The query retrieves average trip duration by week and year from the NYC_CITIBIKES_2019_2020 table.

```
1 SELECT (SUM(TRIPDURATION)/COUNT(*)) as average, WEEK, YEAR FROM (SELECT TRIPDURATION, TO_CHAR(starttime, 'WW') as WEEK,  
2 TO_CHAR(starttime, 'YYYY') as year from NYC_CITIBIKES_2019_2020) group by WEEK, year order by WEEK;  
3  
4  
5  
6
```

The results are displayed in a table:

	average	week	year
1	825.486	01	2019
2	511.056	01	2020
3	781.864	02	2019
4	630.938	02	2020
5	768.424	03	2019
6	472.777	03	2020
7	754.368	04	2019
8	579.287	04	2020
9	797.347	05	2019
10	481.647	05	2020
11	817.432	06	2019
12	918.019	06	2020
13	846.263	07	2019
14	571.555	07	2020
15	774.401	08	2019
16	647.846	08	2020
17	759.404	09	2019

Examining when trips occur during the day

- Breakdown of subscriber (or member) vs customer (non-subscriber) rides

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Project, 5 Examples, SQL, MySQL, SQL Col, Query, Index of, SQL W, Writing, SQL MIN, SQL PAR, and SQL w. The main title is "Google Cloud Platform - Project - NYC Citi Bike Share". A search bar at the top right says "Search products and resources". On the left, there's a sidebar with icons for Explorer, SHORTCUT, and DISABLE EDITOR TABS. The main content area has tabs for EDITOR, UNSAVE..., PAGE 26, GENDER..., and a blue "+ COMPOSE NEW QUERY" button. Below these are buttons for RUN, SAVE, SCHEDULE, and MORE. A status message indicates "This query will process 1,012.3 MiB when run." The query code is as follows:

```
1 with k1 as (
2     select distinct extract(time from starttime) as Subscriber from `bigquery-public-data.new_york_citibike.citibike_trips` where usertype ='Subscriber'
3 ),
4     k2 AS (
5         select distinct extract(time from starttime) as Customer from `bigquery-public-data.new_york_citibike.citibike_trips` where usertype ='Customer'
6     )
7     select k1.Subscriber, k2.Customer from k1,k2 limit 20000;
```

The processing location is set to US. The "Query results" section shows the following data:

Row	Subscriber	Customer
1	13:28:24	14:13:05
2	13:28:24	18:02:53.903000
3	13:28:24	12:39:40
4	13:28:24	13:25:02
5	13:28:24	10:25:25.532000
6	13:28:24	15:27:28

Below the table, there are buttons for SAVE RESULTS and EXPLORE DATA. The status bar at the bottom shows "Rows per page: 100 1 - 100 of 20000 First page < > | Last page". At the very bottom, there are tabs for PERSONAL HISTORY, PROJECT HISTORY, and SAVED QUERIES, along with a "Saved queries" section.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. On the left, the sidebar displays 'Explorer' and a list of datasets: 'project-nyc-citi-bike-share' (selected), 'damg_project', and 'bigquery-public-data'. The main area shows a query editor with the following SQL code:

```
1 with k1 as (
2     select distinct extract(time from starttime) as Subscriber from `project-nyc-citi-bike-share.damg_project.citibike_trips` where usertype = 'Subscriber' limit 10
3 ),
4     k2 AS (
5         select distinct extract(time from starttime) as Customer from `project-nyc-citi-bike-share.damg_project.citibike_trips` where usertype = 'Customer' limit 10
6     )
7     select k1.Subscriber, k2.Customer from k1,k2;
```

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

Row	Subscriber	Customer
1	19:38:00	19:10:47
2	19:38:00	21:05:27
3	19:38:00	16:04:43
4	19:38:00	15:50:06
5	19:38:00	16:46:50
6	19:38:00	17:15:26

At the bottom, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'. The status bar at the bottom right shows '1258 AM 12/15/2021'.

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet. The left sidebar shows the 'Navigator' with 'ADMIN' selected and a list of tables: 'NYC_CITIBIKES_2019_2020', 'TRIPDURATION', 'STARTTIME', 'STOPTIME', 'START_STATION_ID', 'START_STATION_NAME', 'START_STATION_LATITUDE', 'START_STATION_LONGITUDE', 'END_STATION_ID', 'END_STATION_NAME', 'END_STATION_LATITUDE', 'END_STATION_LONGITUDE', 'BIKEID', 'USERTYPE', 'BIRTH_YEAR', and 'GENDER'. The main area contains the following SQL query:

```
1 with temporary as (SELECT DISTINCT to_char(starttime, 'HH24:MI:SS') as subscriber from NYC_CITIBIKES_2019_2020
2 where USERTYPE='Subscriber'),temp as(SELECT DISTINCT to_char(starttime, 'HH24:MI:SS') as customer from NYC_CITIBIKES_2019_2020
3 where USERTYPE='Customer')
4 select * from temporary,temp
```

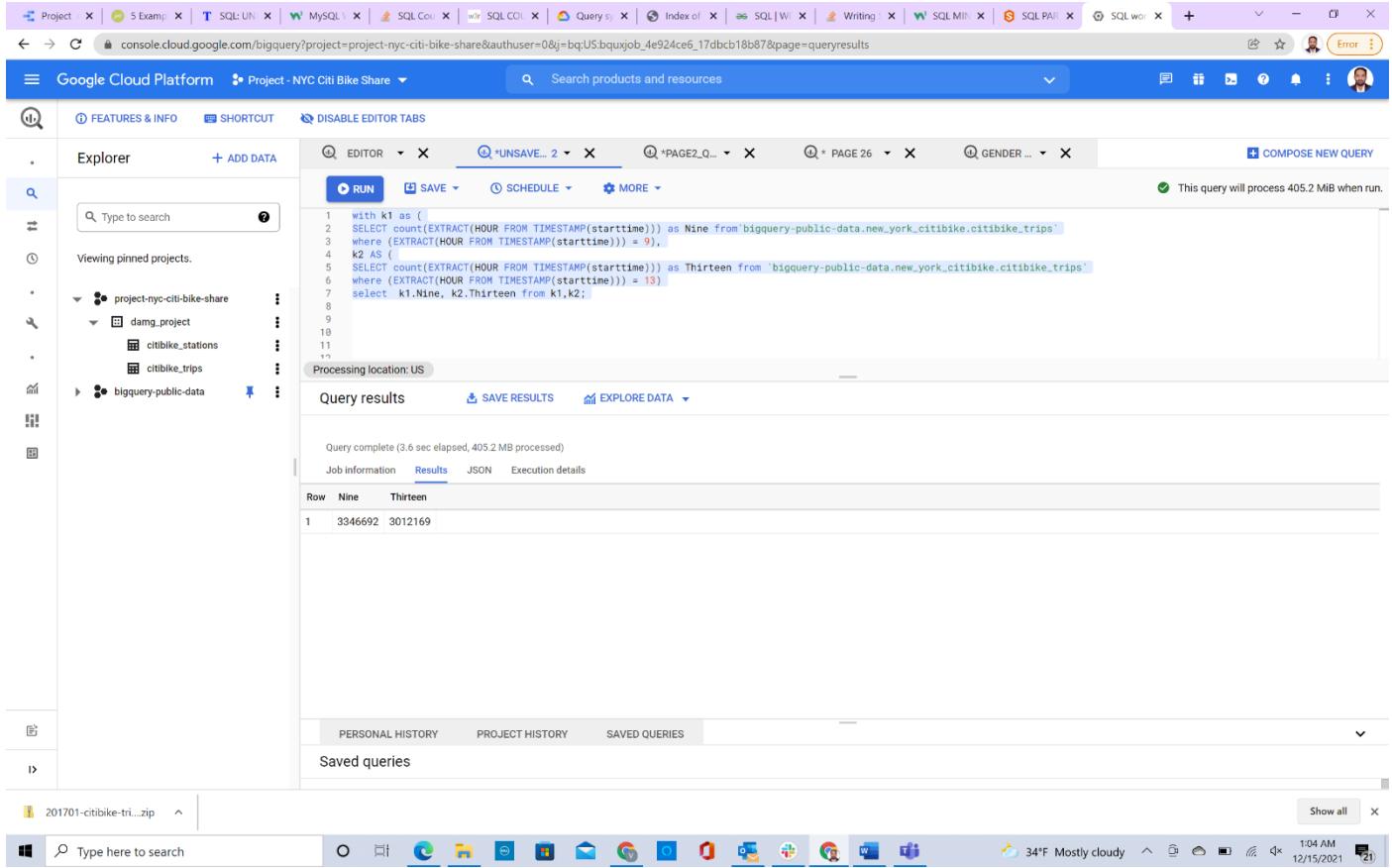
The 'Query Result' tab shows the output of the query:

	subscriber	customer
1	00:00:00	17:37:06
2	00:00:01	17:37:06
3	00:00:02	17:37:06
4	00:00:03	17:37:06
5	00:00:04	17:37:06
6	00:00:05	17:37:06
7	00:00:06	17:37:06
8	00:00:07	17:37:06
9	00:00:08	17:37:06

The status bar at the bottom right shows '10:31 PM 12/15/2021'.

- How many bike trips occur during what periods of the day:
- By hour (0900, 1300)

Dataset 1 –



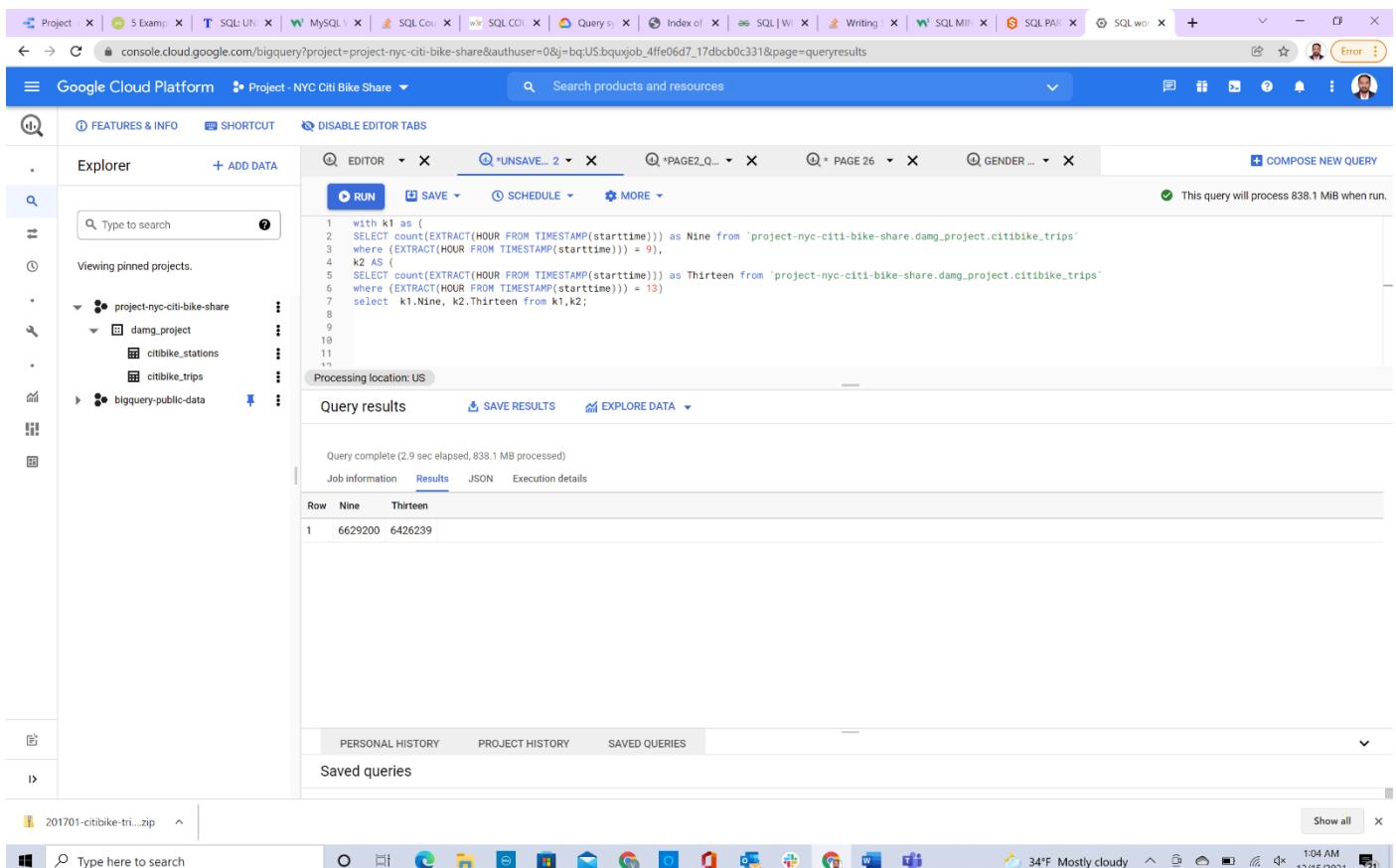
The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays pinned projects, including 'project-nyc-citi-bike-share' which contains 'damg_project' with datasets 'citibike_stations' and 'citibike_trips'. The main area shows a query editor with the following SQL code:

```

1 with k1 as (
2   SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Nine from `bigquery-public-data.new_york_citibike.citibike_trips`
3   where (EXTRACT(HOUR FROM TIMESTAMP(starttime)) = 9),
4   k2 AS (
5     SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Thirteen from `bigquery-public-data.new_york_citibike.citibike_trips`
6     where (EXTRACT(HOUR FROM TIMESTAMP(starttime)) = 13)
7   select k1.Nine, k2.Thirteen from k1,k2;
8
9
10
11
12
  
```

The processing location is set to US. The results show two rows: Row, Nine, and Thirteen. The first row has a value of 3346692 for Nine and 3012169 for Thirteen. The second row has a value of 6629200 for Nine and 6426239 for Thirteen.

Dataset 2 –



The screenshot shows the Google Cloud Platform BigQuery interface, similar to the previous one. The left sidebar shows the same project structure. The main area shows a query editor with the same SQL code as Dataset 1:

```

1 with k1 as (
2   SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Nine from `project-nyc-citi-bike-share.damg_project.citibike_trips`
3   where (EXTRACT(HOUR FROM TIMESTAMP(starttime)) = 9),
4   k2 AS (
5     SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Thirteen from `project-nyc-citi-bike-share.damg_project.citibike_trips`
6     where (EXTRACT(HOUR FROM TIMESTAMP(starttime)) = 13)
7   select k1.Nine, k2.Thirteen from k1,k2;
8
9
10
11
12
  
```

The processing location is set to US. The results show two rows: Row, Nine, and Thirteen. The first row has a value of 6629200 for Nine and 6426239 for Thirteen. The second row has a value of 1 for Nine and 1 for Thirteen.

Dataset 3 –

The screenshot shows the Oracle Database Actions | SQL interface. In the Navigator pane, the schema is set to ADMIN and the table is NYC_CITIBIKES_2019_2020. The table structure includes columns such as TRIPDURATION, STARTTIME, STOPTIME, START_STATION_ID, START_STATION_NAME, START_STATION_LATITUDE, START_STATION_LONGITUDE, END_STATION_ID, END_STATION_NAME, END_STATION_LATITUDE, END_STATION_LONGITUDE, BIKEID, USERTYPE, BIRTH_YEAR, and GENDER.

The main area displays a SQL query:

```

1  WITH k1 AS (
2    SELECT count(EXTRACT(HOUR FROM starttime)) as Nine from nyc_citibikes_2019_2020
3    where EXTRACT(HOUR FROM starttime) = 9,
4    k2 AS (
5      SELECT count(EXTRACT(HOUR FROM starttime)) as Thirteen from nyc_citibikes_2019_2020
6    where EXTRACT(HOUR FROM starttime) = 13)
7    select k1.Nine, k2.Thirteen from k1,k2;
  
```

The Query Result tab shows the execution time: 1.564 seconds. The results are:

	nine	thirteen
1	1321572	1182099

At the bottom, there are file icons and a status bar indicating 10:42:18 AM - 1 rows total.

- During periods during the day i.e., morning 6am-11:59am, afternoon noon-5:59pm, evening 6pm-10pm
- Dataset 1 –**

The screenshot shows the Google Cloud Platform BigQuery interface. The project is named 'Project - NYC Citi Bike Share'. The sidebar shows pinned projects: 'project-nyc-citi-bike-share' containing 'dang_project' (with 'citibike_stations' and 'citibike_trips') and 'bigquery-public-data'. The main area displays a query:

```

1  with k1 as (
2    SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Morning from `bigquery-public-data.new_york_citibike.citibike_trips`
3    where (EXTRACT(HOUR FROM TIMESTAMP(starttime))) in (6,7,8,9,10,11,59),
4    k2 as (
5      SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Afternoon from `bigquery-public-data.new_york_citibike.citibike_trips`
6    where (EXTRACT(HOUR FROM TIMESTAMP(starttime))) in (12,13,14,15,16,17,59),
7    k3 as (
8      SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Evening from `bigquery-public-data.new_york_citibike.citibike_trips`
9    where (EXTRACT(HOUR FROM TIMESTAMP(starttime))) in (18,19,20,21,22)
10   select k1.Morning, k2.Afternoon, k3.Evening from k1,k2,k3;
  
```

The results show the count of trips for Morning, Afternoon, and Evening hours:

Row	Morning	Afternoon	Evening
1	13321397	16043162	13785329

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Project, SQL, MySQL, SQL Col, Query, Index of, SQL Writing, SQL ML, SQL PAI, and SQL workbench. The main window displays a query editor with the following SQL code:

```
1 with k1 as (
2   SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Morning from `project-nyc-citi-bike-share.damg_project.citibike_trips`
3   where (EXTRACT(HOUR FROM TIMESTAMP(starttime)) in (6,7,8,9,10,11,59)),
4   k2 as (
5     SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Afternoon from `project-nyc-citi-bike-share.damg_project.citibike_trips`
6     where (EXTRACT(HOUR FROM TIMESTAMP(starttime)) in (12,13,14,15,16,17,59)),
7     k3 as (
8       SELECT count(EXTRACT(HOUR FROM TIMESTAMP(starttime))) as Evening from `project-nyc-citi-bike-share.damg_project.citibike_trips`
9       where (EXTRACT(HOUR FROM TIMESTAMP(starttime)) in (18,19,20,21,22))
10      select k1.Morning, k2.Afternoon, k3.Evening from k1,k2,k3;
```

The results section shows the following data:

Row	Morning	Afternoon	Evening
1	26732889	34304749	28151028

Dataset 3 –

The screenshot shows the Oracle Database Actions worksheet interface. The top navigation bar includes tabs for Canvas at, Course grader, Thank you, Oracle Cloud, (160) What's new, Final press, Index of, Cloud Sign-in, Timestamp, What is this, (1) Feed, and +. The main window displays a query editor with the following SQL code:

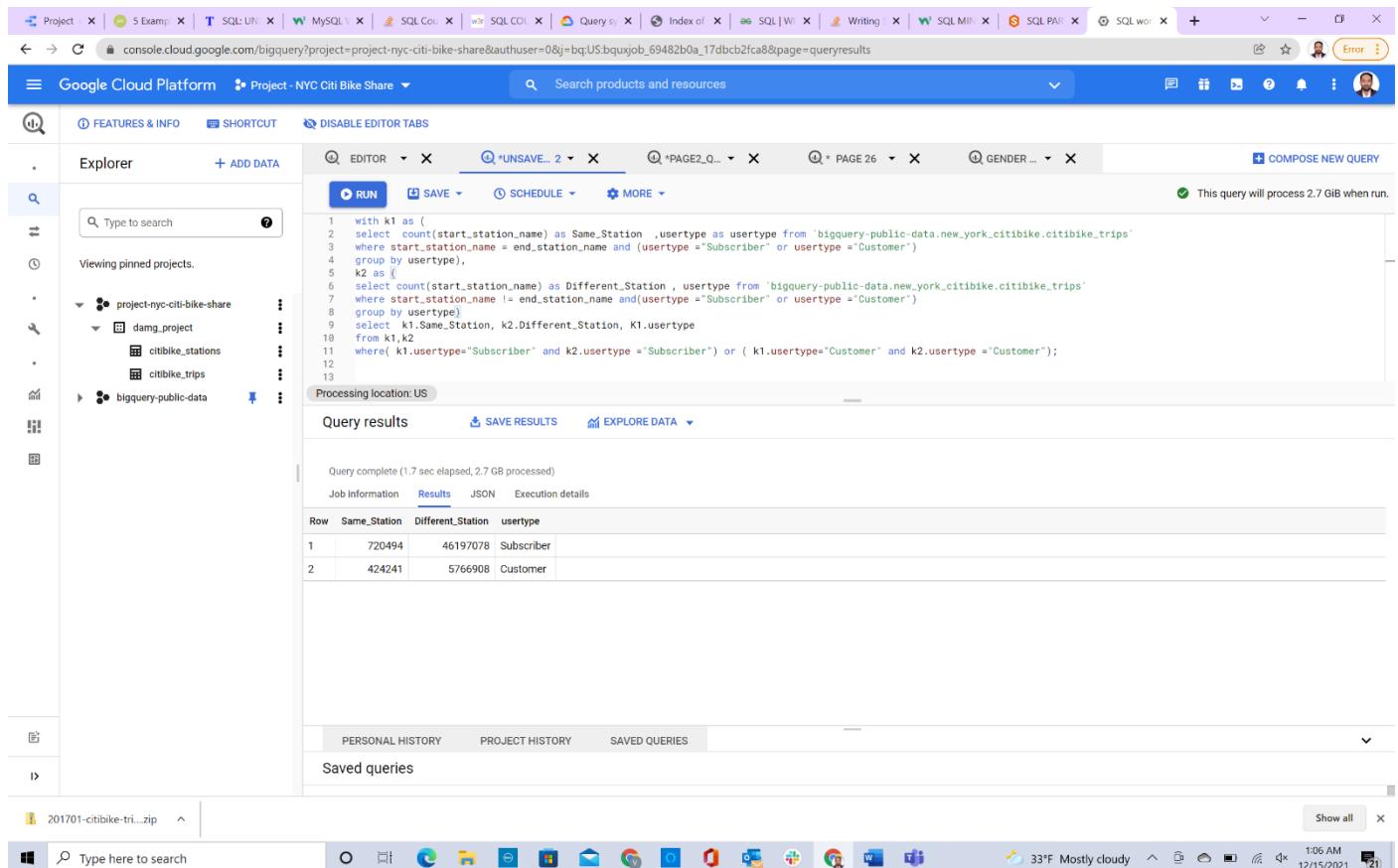
```
1 with k1 as (SELECT count(EXTRACT(HOUR FROM starttime)) as Morning from nyc_citibikes_2019_2020 where EXTRACT(HOUR FROM starttime) in (6,7,8,9,10,11,59)),k2 as
2   (SELECT count(EXTRACT(HOUR FROM starttime)) as Afternoon from nyc_citibikes_2019_2020 where EXTRACT(HOUR FROM starttime) in
3   (12,13,14,15,16,17,59)),k3 as(SELECT count(EXTRACT(HOUR FROM starttime)) as Evening from nyc_citibikes_2019_2020 where (EXTRACT(HOUR FROM starttime))
4   in (18,19,20,21,22))select k1.Morning, k2.Afternoon, k3.Evening from k1,k2,k3;
```

The results section shows the following data:

	morning	afternoon	evening
1	5311204	6317966	5291594

- How many round trips (start and end station are the same) versus non-round trips (different start and end station)
- Further breakdown subscriber vs customer

Dataset 1 –



The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes links for Project, SQL, MySQL, SQL Core, Query, Index of, SQL Writing, SQL ML, SQL PAI, and SQL Workload. Below the navigation bar is a search bar for products and resources.

The main area displays a query editor window with several tabs open. The active tab is "EDITOR". The code in the editor is:

```

1 with k1 as (
2 select count(start_station_name) as Same_Station ,usertype as usertype from `bigquery-public-data.new_york_citibike.citibike_trips`
3 where start_station_name = end_station_name and (usertype = "Subscriber" or usertype = "Customer")
4 group by usertype,
5 k2 as (
6 select count(start_station_name) as Different_Station ,usertype from `bigquery-public-data.new_york_citibike.citibike_trips`
7 where start_station_name != end_station_name and (usertype = "Subscriber" or usertype = "Customer")
8 group by usertype)
9 select k1.Same_Station, k2.Different_Station, k1.usertype
10 from k1,k2
11 where( k1.usertype="Subscriber" and k2.usertype ="Subscriber") or ( k1.usertype="Customer" and k2.usertype ="Customer");
12
13

```

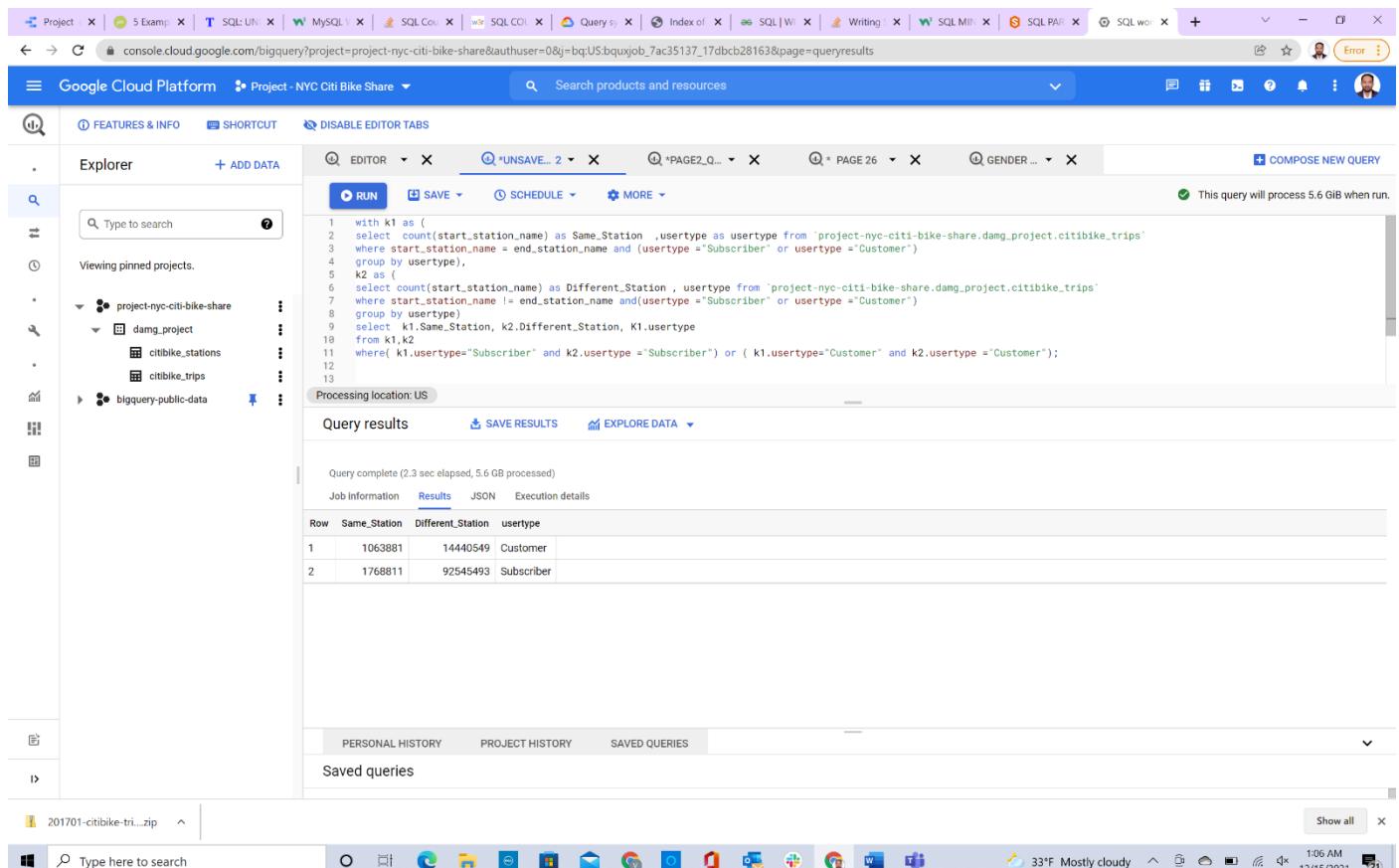
The status bar at the bottom right indicates "This query will process 2.7 GiB when run."

Below the editor is a "Query results" section with tabs for "SAVE RESULTS" and "EXPLORE DATA". The results table has columns: Row, Same_Station, Different_Station, and usertype. The data is:

Row	Same_Station	Different_Station	usertype
1	720494	46197078	Subscriber
2	424241	5766908	Customer

The status bar at the bottom right shows "Query complete (1.7 sec elapsed, 2.7 GB processed)".

Dataset 2 –



The screenshot shows the Google Cloud Platform BigQuery interface, similar to the previous one but for a different dataset. The top navigation bar and search bar are identical.

The main area displays a query editor window with several tabs open. The active tab is "EDITOR". The code in the editor is:

```

1 with k1 as (
2 select count(start_station_name) as Same_Station ,usertype as usertype from `project-nyc-citi-bike-share.damg.project.citibike_trips`
3 where start_station_name = end_station_name and (usertype = "Subscriber" or usertype = "Customer")
4 group by usertype,
5 k2 as (
6 select count(start_station_name) as Different_Station ,usertype from `project-nyc-citi-bike-share.damg_project.citibike_trips`
7 where start_station_name != end_station_name and (usertype = "Subscriber" or usertype = "Customer")
8 group by usertype)
9 select k1.Same_Station, k2.Different_Station, k1.usertype
10 from k1,k2
11 where( k1.usertype="Subscriber" and k2.usertype ="Subscriber") or ( k1.usertype="Customer" and k2.usertype ="Customer");
12
13

```

The status bar at the bottom right indicates "This query will process 5.6 GiB when run."

Below the editor is a "Query results" section with tabs for "SAVE RESULTS" and "EXPLORE DATA". The results table has columns: Row, Same_Station, Different_Station, and usertype. The data is:

Row	Same_Station	Different_Station	usertype
1	1063881	14440549	Customer
2	1768811	92545493	Subscriber

The status bar at the bottom right shows "Query complete (2.3 sec elapsed, 5.6 GB processed)".

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL Worksheet interface. On the left, there's a Navigator pane with 'ADMIN' selected, showing tables like TRIPDURATION, STARTTIME, and START_STATION_ID. The main area contains a SQL query:

```

1 with k1 as (
2 select count(start_station_name) as Same_Station from nyc_citibikes_2019_2020
3 where start_station_name = end_station_name,
4 k2 as (
5 select count(start_station_name) as Different_Station from nyc_citibikes_2019_2020
6 where start_station_name != end_station_name)
7 select k1.Same_Station, k2.Different_Station from k1,k2;
8
9

```

The 'Query Result' tab is selected, showing the output:

	same_station	different_station
1	491630	20366603

At the bottom, it says 'Execution time: 1.487 seconds'.

Average length of trip

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The sidebar shows pinned projects: project-nyc-citi-bike-share, damg_project, and bigquery-public-data. The main area displays a query in the editor:

```

1 with k1 as (
2 select count(start_station_name) as round_trip ,(tripduration) as tripduration.round_trip from `bigquery-public-data.new_york_citibike.citibike_trips`
3 where start_station_name = end_station_name and (usertype ="Subscriber" or usertype ="Customer")
4 group by tripduration,
5 k2 as (
6 select count(start_station_name) as nonround_trip ,(tripduration ) as tripduration.nonround_trip from `bigquery-public-data.new_york_citibike.citibike_trips`
7 where start_station_name != end_station_name and(usertype ="Subscriber" or usertype ="Customer")
8 group by tripduration)
9 select k1.round_trip,avg( k1.tripduration.round_trip ) as average_round_trip ,k2.nonround_trip ,avg( k2.tripduration.nonround_trip) as average_nonroundtrip
10 from k1,k2
11 group by k1.round_trip ,k2.nonround_trip;
12

```

The 'Query results' tab is selected, showing the output:

Row	round_trip	average_round_trip	nonround_trip	average_nonroundtrip
1	496	290.5	11774	1448.0
2	496	290.5	11	11708.750491159139
3	463	339.0	2277	2330.0
4	463	339.0	12354	1418.0
5	463	339.0	92	4701.842105263157
6	303	1388.5882352941178	18280	1123.0
7	303	1388.5882352941164	26	7890.637931034486
8	303	1388.5882352941178	154	3929.5

At the bottom, it says 'Rows per page: 100' and '1 - 100 of 2154600'.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Project, SQL, MySQL, SQL Copy, Query, Index of, SQL Writing, SQL MIN, SQL PAI, and SQL workbench. The main title is "Google Cloud Platform Project - NYC Citi Bike Share". A search bar at the top right says "Search products and resources". The left sidebar has sections for Explorer, Viewing pinned projects, and project-nyc-citi-bike-share (which contains damg_project, citibike_stations, and citibike_trips). Below the sidebar is a query editor with tabs for RUN, SAVE, SCHEDULE, and MORE. The query itself is a complex SQL statement involving multiple CTEs (K1 and K2) to calculate average round-trip durations for subscribers and customers. The results section shows a table with 8 rows of data, including columns for Row, round_trip, average_round_trip, nonround_trip, and average_nonroundtrip. The bottom status bar shows "Rows per page: 100 | 1 - 100 of 4510508 | First page | Last page". The taskbar at the bottom includes icons for File, Home, Recent, Task View, Start, Taskbar settings, and a search bar.

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet. The top navigation bar includes tabs for Canvas, Course, Thank you, Oracle Cloud, SQL, (160) Who, Final press, Index of b, Cloud Sign, Timestamp, What is th, (1) Feed, and +. The main title is "ORACLE Database Actions | SQL". The left sidebar shows the Navigator and Worksheets sections, with ADMIN selected. Under Tables, there is a table named "NYC_CITIBIKES_2019_2020" which contains columns like TRIPDURATION, STARTTIME, STOPTIME, START_STATION_ID, START_STATION_NAME, START_STATION_LATITUDE, START_STATION_LONGITUDE, END_STATION_ID, END_STATION_NAME, END_STATION_LATITUDE, END_STATION_LONGITUDE, BIKEID, USERTYPE, BIRTH_YEAR, and GENDER. The central workspace displays a SQL query and its execution results. The query uses CTEs (K1 and K2) to calculate average trip duration for subscribers and customers. The results table has two columns: subscriber and customer, with one row showing values 11.570 and 20.126 respectively. The bottom status bar shows "1050:18 AM - 1 rows total". The taskbar at the bottom includes icons for File, Home, Recent, Task View, Start, Taskbar settings, and a search bar.

- Length of trips (in time)
- What is the average length (in minutes) of bike trip?
- Further breakdown by subscriber versus customer

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'bigquery-public-data'. The main area contains a query editor window with the following SQL code:

```

1 with k1 as (
2   select AVG((TIMESTAMP_DIFF( stoptime, TIMESTAMP (starttime), MINUTE))) as Subscriber from `bigquery-public-data.new_york_citibike.citibike_trips`
3   where usertype = 'Subscriber',
4     k2 AS (
5   select AVG((TIMESTAMP_DIFF( stoptime, TIMESTAMP (starttime), MINUTE))) as Customer from `bigquery-public-data.new_york_citibike.citibike_trips`
6   where usertype = 'Customer'
7   select k1.Subscriber, k2.Customer from k1,k2;
8
9
10
11
12

```

The 'Processing location: US' is indicated. Below the code, the 'Query results' section shows the output:

Row	Subscriber	Customer
1	12.970576589086951	35.28363119672949

The status bar at the bottom indicates 'Query complete (0.0 sec elapsed; cached)'.

Dataset 2 –

This screenshot is identical to the one above, showing the Google Cloud Platform BigQuery interface. The left sidebar shows the 'Explorer' section with the same pinned projects. The main area contains the same SQL query and output. The status bar at the bottom indicates 'Query complete (1.9 sec elapsed; 2.8 GB processed)'.

Dataset 3 –

The screenshot shows an Oracle Database Actions SQL worksheet. The code is as follows:

```
20 with k1 as (
21 select avg(EXTRACT(minute FROM(stoptime - starttime))) as Subscriber from nyc_citibikes_2019_2020
22 where usertype = 'Subscriber',
23      24
24 select avg(EXTRACT(minute FROM(stoptime - starttime))) as Customer from nyc_citibikes_2019_2020
25 where usertype = 'Customer'
26 select k1.Subscriber, k2.Customer from k1,k2;
```

The results table shows:

subscriber	customer
11.570	20.126

At the bottom, the status bar indicates: 0 0 0 0 | 357/03 AM - 1 rows total.

- How many trips are less than 15 minutes in duration?

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query is:

```
1 select count(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)) as Trip_Count
2 from `bigquery-public-data.new_york.citibike.citibike_trips`
3 where (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)) < 15;
```

The results table shows:

Row	Trip_Count
1	35966066

At the bottom, the status bar indicates: 10:32 AM 12/15/2021.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

```
1 select count(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)) as Trip_Count
2 from `project-nyc-citi-bike-share.damg.project.citibike_trips`
3 where (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)) < 15;
4
5
6
7
8
9
10
```

The results pane shows a single row of data:

Row	Trip_Count
1	72166567

At the bottom, there are tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet. The left sidebar displays the 'Navigator' and 'Worksheets' sections, with 'ADMIN' selected. The 'Worksheets' section shows a list of tables under 'NYC_CITIBIKES_2019_2020'. The main area shows a query editor with the following SQL code:

```
1 select count(EXTRACT(minute FROM(stoptime - starttime))) as Trip_Count
2 from NYC_CITIBIKES_2019_2020
3 HAVING count(EXTRACT(minute FROM(stoptime - starttime))) < 15;
4
5
6
7
8
```

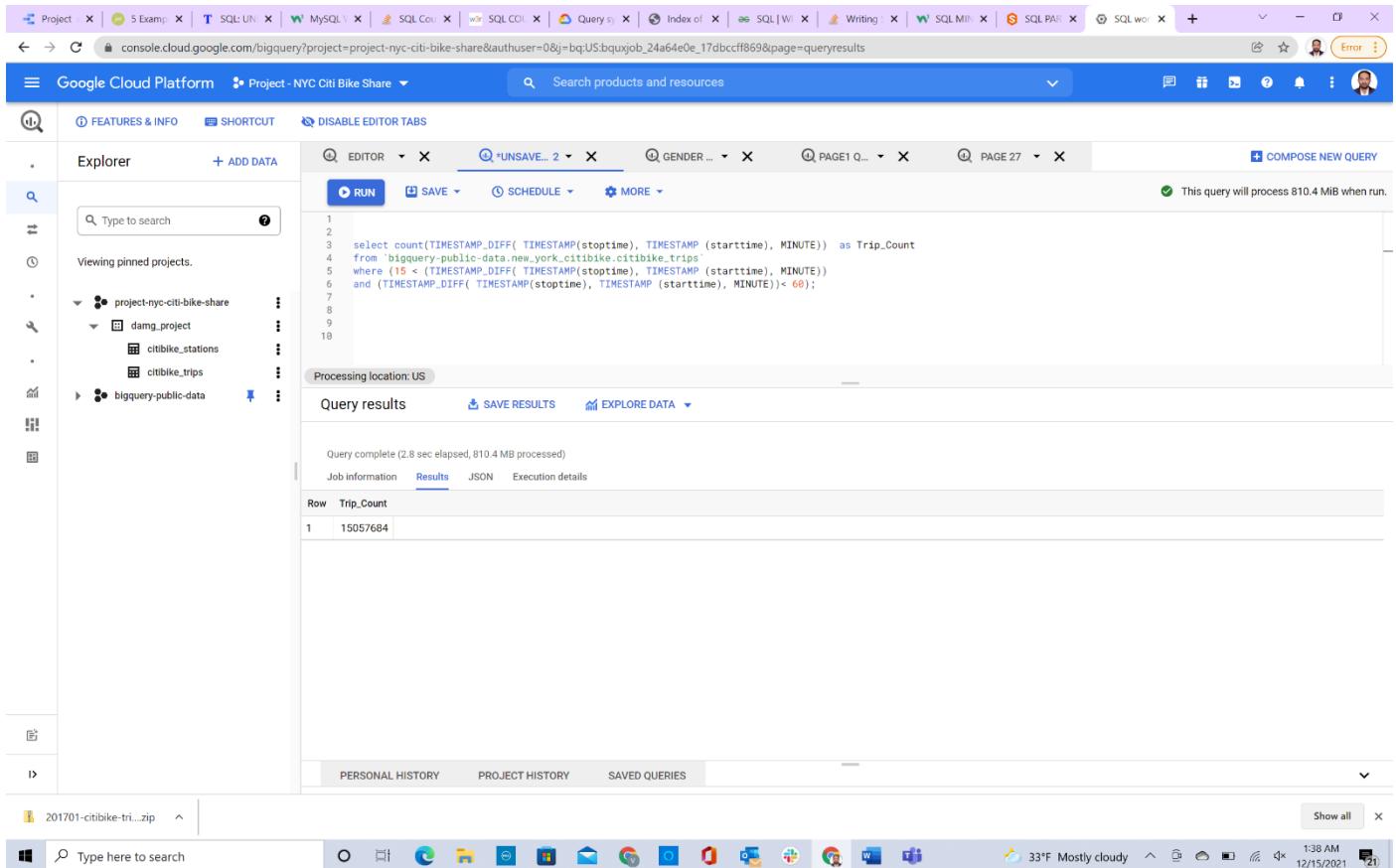
The results pane shows a single row of data:

Consumer Group:	LOW
No items to display.	

At the bottom, there are tabs for 'Query Result', 'Script Output', 'DBMS Output', 'Explain Plan', 'Autotrace', 'SQL History', and 'Data Loading'. The status bar at the bottom indicates '33629 AM 0 rows total'.

- How many trips are greater than 15 minutes but less than 1 hour in duration?

Dataset 1 –



The screenshot shows the Google Cloud Platform BigQuery interface. The query results show a single row with a value of 15057684.

```

1
2
3   select count(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE))  as Trip_Count
4   from `bigquery-public-data.new_york_citibike.citibike_trips`
5   where (15 < (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE))
6   and (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)< 60);
7
8
9
10

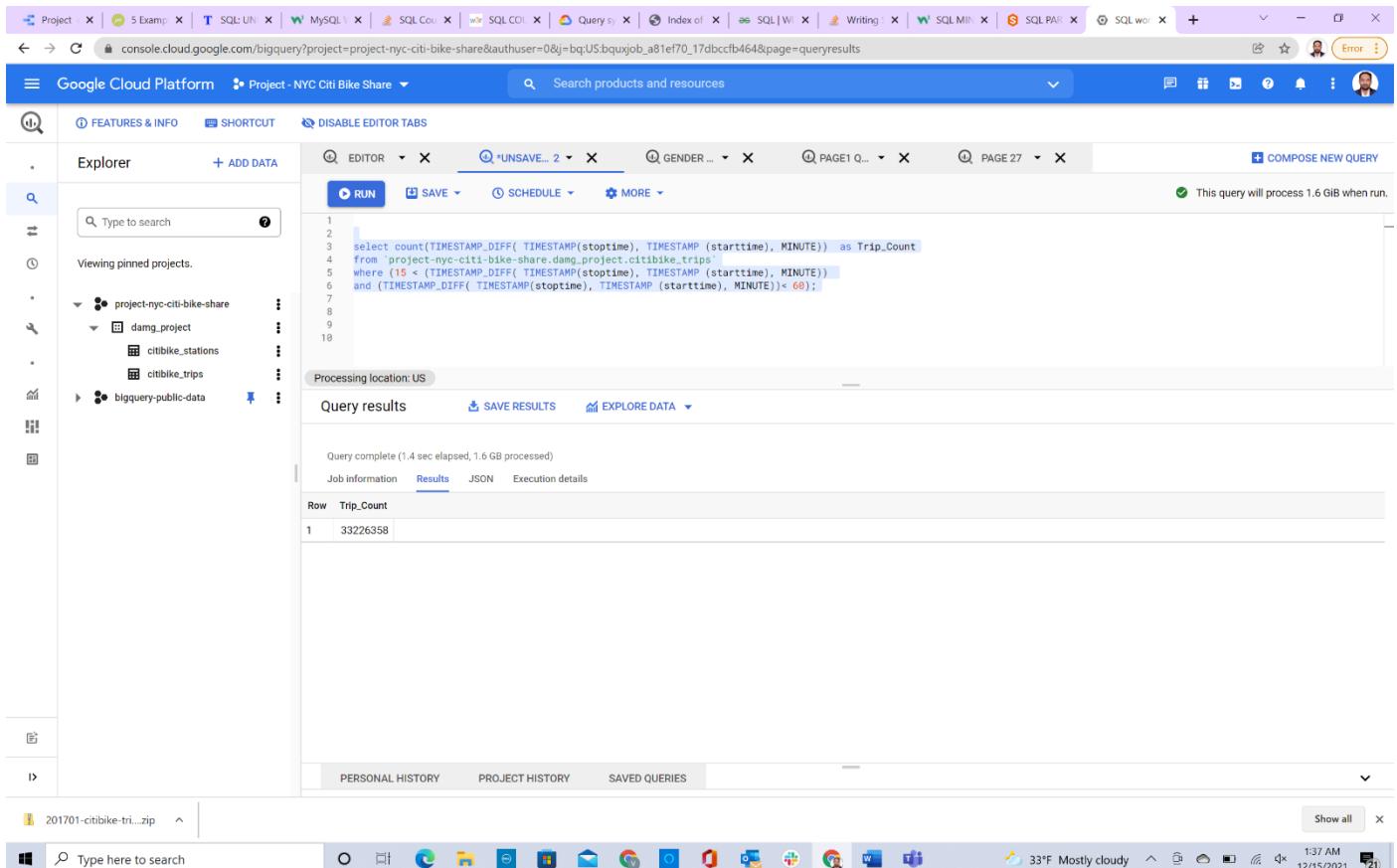
Processing location: US

Query results  SAVE RESULTS  EXPLORE DATA

Query complete (2.8 sec elapsed, 810.4 MB processed)
Job information  Results  JSON  Execution details

Row Trip_Count
1 15057684
  
```

Dataset 2 –



The screenshot shows the Google Cloud Platform BigQuery interface. The query results show a single row with a value of 33226358.

```

1
2
3   select count(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE))  as Trip_Count
4   from `project-nyc-citi-bike-share.damg_project.citibike_trips`
5   where (15 < (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE))
6   and (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)< 60);
7
8
9
10

Processing location: US

Query results  SAVE RESULTS  EXPLORE DATA

Query complete (1.4 sec elapsed, 1.6 GB processed)
Job information  Results  JSON  Execution details

Row Trip_Count
1 33226358
  
```

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL Worksheet interface. The query window contains the following SQL code:

```
1 select EXTRACT(MINUTE from starttime) as Trip_Count
2 from nyc_citibikes_2019_2020
3 where (15 = (EXTRACT(MINUTE from ((stoptime)-(starttime) )))  
4 and (EXTRACT(MINUTE from ((stoptime)-(starttime)))= 60)) group by starttime ;
```

The results table is titled "trip_count" and displays the following data:

trip_count
48
3
45
5
19
29
30
32
35
39
45
58
19
46
59
19
46
??

The status bar at the bottom indicates "347/33 AM - 100 rows fetched, more to get".

- How many trips are greater than 1 hour in duration?

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query window contains the following SQL code:

```
1
2
3 select count(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), HOUR)) as Trip_Count
4 from `bigquery-public-data.new_york.citibike.citibike_trips`
5 where (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), HOUR)) > 1;
6
7
8
```

The results table is titled "Query results" and displays the following data:

Row	Trip_Count
1	198935

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Project, SQL, MySQL, SQL Col, Query, Index of, SQL Writing, SQL MIN, SQL PAI, and SQL work. The main title is "Google Cloud Platform - Project - NYC Citi Bike Share". A search bar at the top right says "Search products and resources". The left sidebar has sections for Explorer, Viewing pinned projects, and a list including "project-nyc-citi-bike-share" (with "damg_project" and "citibike_stations" sub-items), and "bigquery-public-data". The main area shows a query editor with the following code:

```
1
2
3 select count(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP( starttime), HOUR)) as Trip_Count
4 from `project-nyc-citi-bike-share.damg_project.citibike_trips`
5 where (TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP( starttime), HOUR)) > 1;
6
7
8
```

The status bar indicates "Processing location: US". Below the editor is a "Query results" section with a table:

Row	Trip_Count
1	433478

Below the table, it says "Query complete (1.3 sec elapsed, 1.6 GB processed)". The bottom of the screen shows a taskbar with various icons and a system tray indicating "33°F Mostly cloudy" and the date "12/15/2021".

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet interface. The top navigation bar includes tabs for How to Con, Oracle Cloud, SQL | Oracle, SQLLines - O, SQL workspace, ORACLE-BASE, MongoDB, how to extract, GROUP BY, rick sherman, and others. The main title is "ORACLE Database Actions | SQL". The left sidebar has a "Navigator" section with items like pg 24, Pg30, 29 A, pg 27, pg 26, and Pg 25. The main area shows a SQL worksheet with the following query:

```
1 select EXTRACT(Minute from starttime) as Trip_Count
2 from nyc_citibikes_2019_2020
3 where (EXTRACT( hour from (stoptime) - (starttime)) > 1 group by starttime;
```

The results are displayed in a table titled "trip_count":

trip_count
31
47
28
9
1
44
39
26
47
31
38
5
15
14
24
43
32
15
20

At the bottom, it says "Execution time: 4.428 seconds". The bottom of the screen shows a taskbar with various icons and a system tray indicating "43°F Cloudy" and the date "12/15/2021".

- What are the top 10 bike stations based on bike trips?
- For starting a trip
- As a trip destination
- Overall, as either a start and/or destination

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query in the editor is:

```

1 SELECT APPROX_TOP_COUNT(start_station_id, 10) as most_frequent_startStation_ID,
2 APPROX_TOP_COUNT(END_station_id, 10) as most_frequent_eNDStation_ID
3 FROM `bigquery-public-data.new_york_citibike.citibike_trips`
4 WHERE start_station_id is not NULL
5
6
7
8
9 LIMIT 10;
10
11
12
  
```

The results table shows the top 10 most frequent start and end stations with their counts:

	most_frequent_startStation_ID.value	most_frequent_startStation_ID.count	most_frequent_eNDStation_ID.value	most_frequent_eNDStation_ID.count
1	519	551078	519	511019
	497	423334	497	444460
	435	403795	435	407982
	426	384116	426	399033
	293	372255	402	377854
	402	367194	293	372679
	285	344546	285	344033
	490	330378	459	323647
	151	318700	151	319866
	477	311402	477	311402

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query in the editor is identical to the one in Dataset 1:

```

1 SELECT APPROX_TOP_COUNT(start_station_id, 10) as most_frequent_startStation_ID,
2 APPROX_TOP_COUNT(END_station_id, 10) as most_frequent_eNDStation_ID
3 FROM `project-nyc-citi-bike-share.damg_project.citibike_trips`
4 WHERE start_station_id is not NULL
5
6
7
8
9 LIMIT 10;
10
11
12
  
```

A message indicates "Query running (4.4 sec - 0/2 stages)". The results table shows the top 10 most frequent start and end stations with their counts:

	most_frequent_startStation_ID.value	most_frequent_startStation_ID.count	most_frequent_eNDStation_ID.value	most_frequent_eNDStation_ID.count
1	519	934233	519	892534
	497	736678	497	766872
	435	723296	435	726781
	426	686594	426	719711
	402	658639	402	688148
	285	620000	285	622653
	293	596393	293	603564
	490	572585	514	577489
	499	563928	358	564031
	474	562242	474	562242

Dataset 3 –

- Determine the most popular routes (staring to end location)
- Total number of trips

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query being run is:

```
15
16
17
18 select * from (SELECT count(*) as cnt,
19 start_station_name,
20 end_station_name ,
21
22 row_number() over (order by count(*) desc) as rnm
23
24 FROM `bigquery-public-data.new_york_citibike.citibike_trips`
25 where start_station_name=end_station_name
26 group by start_station_name,end_station_name) where rnm=1;
```

The results show a single row:

Row	cnt	start_station_name	end_station_name	rnm
1	18667	12 Ave & W 40 St	West St & Chambers St	1

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query being run is identical to Dataset 1:

```
1 select * from (SELECT count(*) as cnt,
2 start_station_name,
3 end_station_name ,
4
5 row_number() over (order by count(*) desc) as rnm
6
7 FROM `project-nyc-citi-bike-share.damg_project.citibike_trips`
8 where start_station_name=end_station_name
9 group by start_station_name,end_station_name) where rnm=1;
```

The results show a single row:

Row	cnt	start_station_name	end_station_name	rnm
1	34470	12 Ave & W 40 St	West St & Chambers St	1

Below the results, there is a "Saved queries" section containing a complex query:

```
1 -- What is the average length (in minutes) of bike trip? Further breakdown by subscriber versus customer
2
3 with k1 as (
4 select AVG(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)) as Subscriber from `project-nyc-citi-bike-share.damg_project.citibike_trips`
5 where usertype ="Subscriber",
6 k2 AS (
7 select AVG(TIMESTAMP_DIFF( TIMESTAMP(stoptime), TIMESTAMP (starttime), MINUTE)) as Customer from `project-nyc-citi-bike-share.damg_project.citibike_trips`
8 where usertype ='Customer')
```

Dataset 3 –

The screenshot shows the Oracle Database Actions interface. On the left, the Navigator pane displays a tree view of tables and columns under the schema 'ADMIN'. One table, 'NYC_CITIBIKES_2019_2020', is expanded, showing columns like TRIPDURATION, STARTTIME, STOPTIME, START_STATION_ID, START_STATION_NAME, START_STATION_LATITUDE, START_STATION_LONGITUDE, END_STATION_ID, END_STATION_NAME, END_STATION_LATITUDE, END_STATION_LONGITUDE, BIKEID, USERTYPE, BIRTH_YEAR, and GENDER. The main workspace contains a SQL worksheet with the following query:

```

1 select * from (SELECT count(*) as cnt,
2 start_station_name,
3 end_station_name,
4 row_number() over (order by count(*) desc) as rnm
5
6 FROM NYC_CITIBIKES_2019_2020
7 where start_station_name!=end_station_name
8 group by start_station_name,end_station_name) where rnm=1;

```

The results tab shows one row of data:

	cnt	start_station_name	end_station_name	rnm
1	8426	E 7 St & Avenue A	Cooper Square & A	1

The status bar at the bottom indicates the execution time was 6.455 seconds.

- Total number of trips by subscriber and customer

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The sidebar on the left shows pinned projects, including 'project-nyc-citi-bike-share' which contains 'dang_project' with 'citibike_stations' and 'citibike_trips' datasets, and 'bigquery-public-data'. The main workspace displays a query editor with the following SQL code:

```

1 select * from (SELECT count(*) as cnt,
2 usertype,
3 start_station_name,
4 end_station_name,
5 row_number() over (order by count(*) desc) as rnm
6 FROM `bigquery-public-data.new_york.citibike.citibike_trips`
7 where start_station_name != end_station_name and usertype="Subscriber"
8 group by start_station_name,end_station_name,usertype) where rnm=1
9 union all
10 select * from (SELECT count(*) as cnt,usertype,
11 start_station_name,
12 end_station_name,
13
14 row_number() over (order by count(*) desc) as rnm
15
16 FROM `bigquery-public-data.new_york.citibike.citibike_trips` where start_station_name != end_station_name and usertype="Customer"
17
18 group by start_station_name,end_station_name,usertype) where rnm=1;
19
20

```

The status bar at the bottom indicates the processing location is US.

The results tab shows two rows of data:

Row	cnt	usertype	start_station_name	end_station_name	rnm
1	11269	Customer	Centre St & Chambers St	Cadman Plaza E & Tillary St	1
2	17260	Subscriber	W 21 St & 6 Ave	9 Ave & W 22 St	1

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays pinned projects, including 'project-nyc-citi-bike-share' which contains datasets 'damg_project' (with tables 'citibike_stations' and 'citibike_trips') and 'bigquery-public-data'. The main area shows a query editor with the following SQL code:

```
1 select * from (SELECT count(*) as cnt,
2 start_station_name,
3 end_station_name ,
4
5 row_number() over (order by count(*) desc) as rnm
6
7
8
9 FROM `project-nyc-citi-bike-share.damg_project.citibike_trips`
10 where start_station_name!=end_station_name
11 group by start_station_name,end_station_name) where rnm=1;
```

The processing location is set to US. Below the query editor, the 'Query results' section shows the output of the query:

Row	cnt	start_station_name	end_station_name	rnm
1	34470	12 Ave & W 40 St	West St & Chambers St	1

The status bar at the bottom indicates the file '201701-citibike-trips.zip' is open, along with the system tray showing weather (32°F Partly cloudy) and date (4:28 AM 12/15/2021).

Dataset 3 –

The screenshot shows the Oracle Database Actions | SQL interface. The left sidebar displays the 'Navigator' and 'Worksheets' sections, with 'ADMIN' selected. Under 'ADMIN', there is a 'Tables' section containing 'NYC_CITIBIKES_2019_2020' which includes columns like 'TRIPDURATION', 'STARTTIME', 'STOPTIME', 'START_STATION_ID', 'START_STATION_NAME', 'START_STATION_LATITUDE', 'START_STATION_LONGITUDE', 'END_STATION_ID', 'END_STATION_NAME', 'END_STATION_LATITUDE', 'END_STATION_LONGITUDE', 'BIKEID', 'USERTYPE', 'BIRTH_YEAR', and 'GENDER'. The main area shows a query editor with the following SQL code:

```
1 select * from (SELECT count(*) as cnt,
2 start_station_name,
3 end_station_name,
4 userstype,
5
6 row_number() over (order by count(*) desc) as rnm
7
8
9 FROM NYC_CITIBIKES_2019_2020
10 where start_station_name != end_station_name and userstype='Subscriber'
11 group by start_station_name,end_station_name,userstype) where rnm=1;
12 union all
13 select * from (SELECT count(*) as cnt,userstype,
14
15 start_station_name,
16
17 end_station_name,
18
19 row_number() over (order by count(*) desc) as rnm
20
21
22
23
24
25
FROM NYC_CITIBIKES_2019_2020 where start_station_name != end_station_name and userstype='Customer' group by start_station_name,end_station_name,userstype) where rnm=1;
```

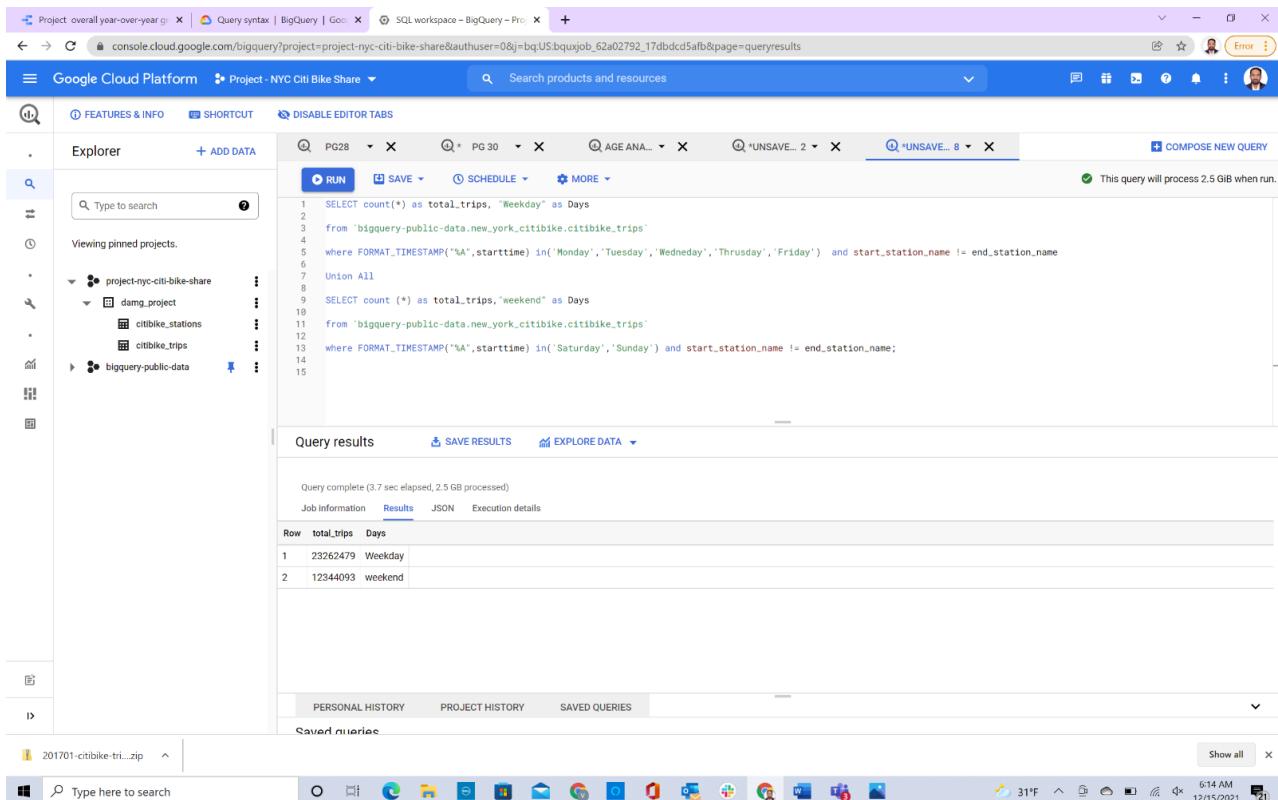
The 'Query Result' tab is selected, showing the output of the query:

cnt	userstype	start_station_name	end_station_name	rnm	
1	8116	Subscriber	E 7 St & Avenue A	Cooper Square & A	1
2	5433	Customer	Central Park S & 6 /	5 Ave & E 88 St	1

The status bar at the bottom indicates the file '201701-citibike-trips.zip' is open, along with the system tray showing weather (55°F Cloudy) and date (12/15/2021).

- Further breakdowns (examples): time of day, weekday vs weekend, month

Dataset 1 –



The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'bigquery-public-data'. The main area contains a query editor window with the following SQL code:

```

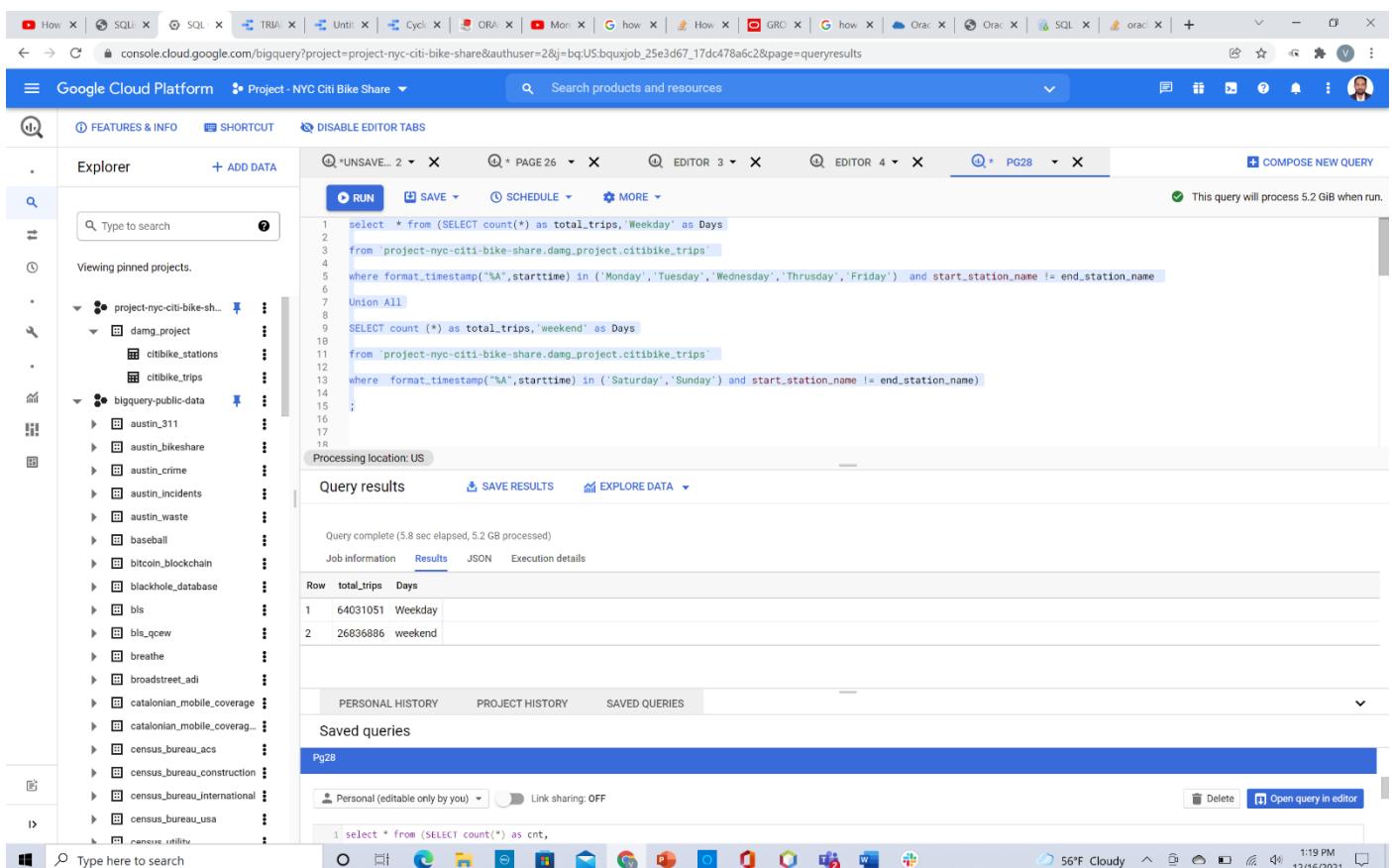
1 SELECT count(*) as total_trips, 'Weekday' as Days
2 from `bigquery-public-data.new_york_citibike.citibike_trips`
3 where FORMAT_TIMESTAMP("%A",starttime) in ('Monday','Tuesday','Wednesday','Thursday','Friday') and start_station_name != end_station_name
4 Union All
5 SELECT count (*) as total_trips,'weekend' as Days
6 from `bigquery-public-data.new_york_citibike.citibike_trips`
7 where FORMAT_TIMESTAMP('%A',starttime) in ('Saturday','Sunday') and start_station_name != end_station_name;
8
9
10
11
12
13
14
15

```

The 'Query results' tab shows the output of the query:

Row	total_trips	Days
1	23262479	Weekday
2	12344093	weekend

Dataset 2 –



The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'bigquery-public-data'. The main area contains a query editor window with the following SQL code:

```

1 select * from (SELECT count(*) as total_trips, Weekday' as Days
2 from `project-nyc-citi-bike-share.damg_project.citibike_trips`
3 where format_timestamp("%A",starttime) in ('Monday','Tuesday','Wednesday','Thursday','Friday') and start_station_name != end_station_name
4 Union All
5 SELECT count (*) as total_trips,'weekend' as Days
6 from `project-nyc-citi-bike-share.damg_project.citibike_trips`
7 where format_timestamp("%A",starttime) in ('Saturday','Sunday') and start_station_name != end_station_name);
8
9
10
11
12
13
14
15
16
17
18

```

The 'Query results' tab shows the output of the query:

Row	total_trips	Days
1	64031051	Weekday
2	26836886	weekend

Dataset 3 –

```

1
2
3
4 SELECT count(tripduration) as total_trips,'Weekday' as Days
5
6 from NYC_CITIBIKES_2019_2020
7
8 where upper( TO_CHAR (starttime,'Day') ) in ('MONDAY','TUESDAY','WEDNESDAY','THURSDAY','FRIDAY') and start_station_name != end_station_name
9
10 Union ALL
11
12 SELECT count (tripduration) as total_trips,'weekend' as Days
13
14 from NYC_CITIBIKES_2019_2020
15
16 where upper( TO_CHAR (starttime,'Day') ) in ('SATURDAY','SUNDAY') and start_station_name != end_station_name;
17
18

```

Query Result

	total_trips	days
1	3129971	Weekday
2	0	weekend

• Gender analysis

- How many trips - female vs male

Dataset 1 –

```

9
10
11 SELECT (count(*))as total_trips_male ,gender  from `bigquery-public-data.new_york_citibike.citibike_trips`
12 where gender in ("1","male","2","female")
13 group by 2;
14
15

```

Query results

Row	total_trips_male	gender
1	35611787	male
2	11376412	female

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for 'Project overall year-over-year', 'Query syntax | BigQuery', 'Google Cloud Platform', 'Project - NYC Citi Bike Share', and 'Search products and resources'. A sidebar on the left titled 'Explorer' shows pinned projects like 'project-nyc-citi-bike-share' and 'bigquery-public-data'. The main area displays a query editor with the following SQL code:

```
1
2
3  SELECT (count(*))as total_trips_male ,gender  from `project-nyc-citi-bike-share.damg_project.citibike_trips`
4  where gender in ('1','male','2','female')
5  group by 2;
```

The status bar indicates 'Processing location: US'. Below the code, the 'Query results' section shows the output:

Row	total_trips_male	gender
1	32276396	male
2	40465673	1
3	15391899	2
4	10341781	female

The bottom of the interface shows a taskbar with various icons and system status.

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet interface. The top navigation bar includes tabs for 'Canvas at', 'Course gr...', 'Thank you...', 'Oracle Clo...', '(160) Who...', 'Final pres...', 'Index of b...', 'Cloud Sig...', 'Timestamp...', 'What Is th...', '(1) Feed...', and 'ADMIN'. The main area shows a SQL worksheet with the following query:

```
1  SELECT count(*) as total_trips_male ,gender from NYC_CITIBIKES_2019_2020
2  where gender in (1,2)
3
4
5  group by gender;
```

The results section shows the output:

total_trips_male	gender
14233935	1
5016236	2

The bottom of the interface shows a taskbar with various icons and system status.

- Average length of trip - female vs male

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

```

12
13
14 SELECT (avg(distinct tripduration)) as average_trips_male ,gender , usertype from `bigquery-public-data.new_york_citibike.citibike_trips`
15 where gender in ("1","male","2","female") and usertype in("Subscriber","Customer")
16 group by 2,3;
  
```

The results table shows the average trip duration for different gender and user type combinations:

Row	average_trips_male	gender	usertype
1	57385.71947251749	male	Subscriber
2	42088.342084741395	female	Subscriber
3	84060.76168600893	male	Customer
4	49068.79949949935	female	Customer

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

```

3
4
5
6 SELECT (avg(distinct tripduration)) as average_trips_male ,gender , usertype from `project-nyc-citi-bike-share.damg_project.citibike_trips`
7 where gender in ("1","male","2","female") and usertype in("Subscriber","Customer")
8 group by 2,3;
  
```

The results table shows the average trip duration for different gender and user type combinations:

Row	average_trips_male	gender	usertype
1	75425.51406212764	1	Customer
2	77206.38830516215	1	Subscriber
3	58251.96550595791	2	Customer
4	53030.056800607	2	Subscriber
5	57821.67618731355	male	Subscriber
6	84060.76168600905	male	Customer
7	42271.90259651674	female	Subscriber
8	49068.7994994995	female	Customer

Dataset 3 –

The screenshot shows an Oracle Database Actions SQL Worksheet interface. The query window contains the following SQL code:

```
1 SELECT (avg(distinct tripduration)) as average_trips_male ,gender , usertype from NYC_CITIBIKES_2019_2020
2 where gender in (1,2) and usertype in('Subscriber','Customer')
3
4 group by gender,usertype;
5
6
7
```

The results table shows the following data:

	average_trips_male	gender	usertype
1	37979.220	1	Customer
2	50802.650	1	Subscriber
3	31432.810	2	Subscriber
4	30114.550	2	Customer

The status bar at the bottom indicates 121357 PM - 4 rows total.

▪ Subscriber versus customer trips - female vs male

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query window contains the following SQL code:

```
1 SELECT (count(*))as total_trips_male ,gender , usertype from
2   `bigquery-public-data.new_york_citibike.citibike_trips`
3 where gender in ("1","male", "2", "female") and usertype in("Subscriber","Customer")
4
5 group by 2,3;
```

The results table shows the following data:

Row	total_trips_male	gender	usertype
1	35308523	male	Subscriber
2	11188711	female	Subscriber
3	303264	male	Customer
4	187701	female	Customer

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'damg_project'. The main area shows a query editor with the following SQL code:

```
1
2
3  SELECT (count(*))as total_trips_male ,gender , usertype from `project-nyc-citi-bike-share.damg_project.citibike_trips`
4  where gender in ("1","male","2","female") and usertype in("Subscriber","Customer")
5  group by 2,3
```

The status bar at the bottom indicates 'Processing location: US'. Below the code, the 'Query results' section shows the output:

Row	total_trips_male	gender	usertype
1	37420052	1	Subscriber
2	13493605	2	Subscriber
3	1885874	2	Customer
4	10154081	female	Subscriber
5	3022527	1	Customer
6	303222	male	Customer
7	31973174	male	Subscriber
8	187700	female	Customer

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL worksheet. The left sidebar has a 'Navigator' section with items like 'Search...', 'pg 27', 'pg 26', and 'Pg 25'. The main area displays the following SQL query:

```
1  SELECT (count(*))as total_trips_male ,gender , usertype from NYC_CITIBIKES_2019_2020
2
3  where gender in (1,2) and usertype in('Subscriber','Customer')
4
5  group by gender,usertype;
```

The results are shown in a table:

	total_trips_male	gender	usertype
1	13158165	1	Subscriber
2	1075770	1	Customer
3	4409073	2	Subscriber
4	607163	2	Customer

- Any trends from 2014 to 2020 regarding above
 - How many trips - female vs male

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a pinned project 'project-nyc-citi-bike-share'. Inside this project, there are two datasets: 'dang_project' containing tables 'citibike_stations' and 'citibike_trips', and 'bigquery-public-data' which is expanded to show the 'new_york_citibike.citibike_trips' table. The main area contains a query editor window with the following SQL code:

```

1 WITH temporary AS(
2   SELECT (count(*))AS total_trips, gender, extract (year from starttime) AS year
3   FROM `bigquery-public-data.new_york_citibike.citibike_trips`
4   WHERE gender='1' OR gender = 'male'
5   GROUP BY gender, year
6 UNION ALL
7   SELECT (count(*))AS total_trips,gender , extract (year from starttime) AS year
8   FROM `bigquery-public-data.new_york_citibike.citibike_trips`
9   WHERE gender='2' OR gender="female"
10  GROUP BY gender, year )
11 select total_trips, gender,year FROM temporary
12 GROUP BY gender,total_trips ,year
13 ;
14
  
```

The status bar at the bottom right indicates 'Query complete (1.7 sec elapsed, 759.4 MB processed)'. Below the editor, the 'Query results' section shows a table with the following data:

Row	total_trips	gender	year
1	2006962	female	2015
2	3236735	female	2017
3	1260893	female	2018
4	1034631	female	2013
5	2186917	female	2016
6	1650274	female	2014

The bottom navigation bar includes tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

Dataset 2 –

This screenshot shows the same Google Cloud Platform BigQuery interface as the previous one, but with a different query. The main area contains the following SQL code:

```

1 WITH temporary AS(
2   SELECT (count(*))AS total_trips, gender, extract (year from starttime) AS year
3   FROM `project-nyc-citi-bike-share.dang_project.citibike_trips`
4   WHERE gender='1' OR gender = "male"
5   GROUP BY gender, year
6 UNION ALL
7   SELECT (count(*))AS total_trips,gender , extract (year from starttime) AS year
8   FROM `project-nyc-citi-bike-share.dang_project.citibike_trips`
9   WHERE gender='2' OR gender="female"
10  GROUP BY gender, year )
11 select total_trips, gender,year FROM temporary
12 GROUP BY gender,total_trips ,year
13 ;
14
  
```

The status bar at the bottom right indicates 'Query complete (2.4 sec elapsed, 1.3 GB processed)'. Below the editor, the 'Query results' section shows a table with the following data:

Row	total_trips	gender	year
1	5635962	male	2014
2	8015451	1	2018
3	9306602	male	2017
4	5551873	2	2020
5	2006962	female	2015
6	2186917	female	2016

The bottom navigation bar includes tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

- Any trends from 2014 to 2020
 - Average length of trip - female vs male

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for 'Project overall year-over-year', 'Query syntax', 'BigQuery', 'Google Cloud', and 'SQL workspace - BigQuery'. Below the navigation is a search bar and a toolbar with icons for 'RUN', 'SAVE', 'SCHEDULE', and 'MORE'.

The main area displays a query editor with the following SQL code:

```

1
2
3   SELECT (count(*))as total_trips_male ,gender ,extract (year from starttime) as year from `bigquery-public-data.new_york_citibike.citibike_trips`
4   where gender in ("1" , "male","2","female")
5   group by 2,3;
6
7
  
```

The status bar indicates 'Processing location: US'.

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

Row	total_trips_male	gender	year
1	3955871	male	2018
2	6769032	male	2016
3	3335391	male	2013
4	2186917	female	2016
5	9306602	male	2017
6	5635962	male	2014
7	1260893	female	2018
8	6608929	male	2015
9	3236735	female	2017
10	1034631	female	2013
11	2066667	female	2015

Below the table are buttons for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'. The status bar at the bottom right shows 'Rows per page: 100', '1 - 12 of 12', 'First page <', '> Last page', and a timestamp '7:02 AM 12/15/2021'.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface, similar to the previous one but with a different query. The top navigation bar, search bar, and toolbar are identical.

The main area displays a query editor with the following SQL code:

```

1
2   SELECT (avg(distinct tripduration)) as average_trips_male ,gender ,usertype,extract (year from starttime) as year from `project-nyc-citi-bike-share.damg.project.citibike_trips`
3   where gender in ("1" , "male","2","female") and usertype in("Subscriber","Customer")
4   group by 2,3,4;
5
  
```

The status bar indicates 'Processing location: US'.

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

Row	average_trips_male	gender	usertype	year
1	50714.09830696985	1	Subscriber	2019
2	37785.78280863109	1	Customer	2019
3	31554.748678746688	2	Subscriber	2019
4	29330.047219558965	2	Customer	2019
5	25470.54671669792	1	Subscriber	2015
6	17915.50383487481	2	Subscriber	2015
7	39053.626099191	1	Customer	2018
8	39439.133072563265	1	Subscriber	2018
9	25038.374926020915	2	Customer	2018
10	22358.520964240844	2	Subscriber	2018
11	42888.514542112665	2	Customer	2020
12	52440.1380716774646	1	Customer	2020

Below the table are buttons for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'. The status bar at the bottom right shows 'Rows per page: 100', '1 - 36 of 36', 'First page <', '> Last page', and a timestamp '6:56 AM 12/15/2021'.

▪ Any trends from 2014 to 2020

Subscriber versus customer trips - female vs male

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a pinned project 'project-nyc-citi-bike-share'. The main area shows a query editor with the following SQL code:

```

1 SELECT (count(*))as total_trips_male ,gender , usertype ,extract (year from starttime) as year from `bigquery-public-data.new_york_citibike.citibike_trips`
2 where gender in ("1","male","2","female") and usertype in("Subscriber","Customer")
3 group by 2,3,4;
4

```

The status bar at the top right indicates an error: 'Invalid value: Table name 'temp' missing dataset while no default dataset is set in the request. at [38:1]'.

The 'Query results' section shows the following data:

Row	total_trips_male	gender	usertype	year
1	3335349	male	Subscriber	2013
2	2006962	female	Subscriber	2015
3	9114616	male	Subscriber	2017
4	3121810	female	Subscriber	2017
5	3864653	male	Subscriber	2018
6	5635912	male	Subscriber	2014
7	1201265	female	Subscriber	2018
8	91218	male	Customer	2018
9	6749064	male	Subscriber	2016
10	1034630	female	Subscriber	2013
11	6608929	male	Subscriber	2015
12	2173792	female	Subscriber	2016

Below the table, there are navigation links for 'Rows per page' (100), '1 - 22 of 22', 'First page', 'Last page', and tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with a pinned project 'project-nyc-citi-bike-share'. The main area shows a query editor with the same SQL code as Dataset 1:

```

1 SELECT (count(*))as total_trips_male ,gender , usertype ,extract (year from starttime) as year from `project-nyc-citi-bike-share.damg.project.citibike_trips`
2 where gender in ("1","male","2","female") and usertype in("Subscriber","Customer")
3 group by 2,3,4;
4

```

A note in the top right says 'This query will process 2.5 GB when run.'

The 'Query results' section shows the following data:

Row	total_trips_male	gender	usertype	year
1	2611383	2	Subscriber	2018
2	1270612	2	Subscriber	2015
3	714	1	Customer	2016
4	5635912	male	Subscriber	2014
5	1650252	female	Subscriber	2014
6	2173792	female	Subscriber	2016
7	1201265	female	Subscriber	2018
8	50	male	Customer	2014
9	786034	2	Subscriber	2016
10	19968	male	Customer	2016
11	22	female	Customer	2014
12	1500746	1	Customer	2000

Below the table, there are navigation links for 'Rows per page' (100), '1 - 36 of 36', 'First page', 'Last page', and tabs for 'PERSONAL HISTORY', 'PROJECT HISTORY', and 'SAVED QUERIES'.

- Age analysis (note: age at the time of the ride i.e., age is year of bike trip vs birth year)
 - What are the number of trips per age

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Project, Examples, SQL, MySQL, SQL Count, Query sync, Index of b, Writing SQL, SQL PARTI, and SQL work. The main title is "Project - NYC Citi Bike Share". A search bar says "Search products and resources". The left sidebar has sections for Explorer, SHORTCUT, and DISABLE EDITOR TABS. The Explorer section shows pinned projects like "project-nyc-citi-bike-share" which contains "damg_project" with "citibike_stations" and "citibike_trips", and "bigquery-public-data". The main area shows a query editor with the following SQL code:

```

1 select distinct (ride_year-birth)as age ,total_ride from (select count(*) as total_ride, birth_year as birth, extract (ISOYEAR from starttime) as ride_year
2 from `bigquery-public-data.new_york_citibike.citibike_trips`
3 group by ride_year,birth )
4 ;

```

The status bar indicates "This query will process 767.8 MiB when run." Below the editor is a "Query results" section with "SAVE RESULTS" and "EXPLORE DATA" buttons. It shows a message "Query complete (1.8 sec elapsed, 767.8 MB processed)". The results table has columns "Row", "age", and "total_ride". The data is as follows:

Row	age	total_ride
1	63	42615
2	21	106926
3	69	9329
4	20	49001
5	20	77841
6	68	12902
7	66	39533
8	64	42255
9	61	22000

At the bottom, there are buttons for "PERSONAL HISTORY", "PROJECT HISTORY", and "SAVED QUERIES". The status bar at the bottom right shows "Rows per page: 100", "1 - 100 of 557", "First page", "Last page", and "Show all". The taskbar at the bottom includes icons for File, Home, Task View, Start, Taskbar settings, and a search bar.

Dataset 2 –

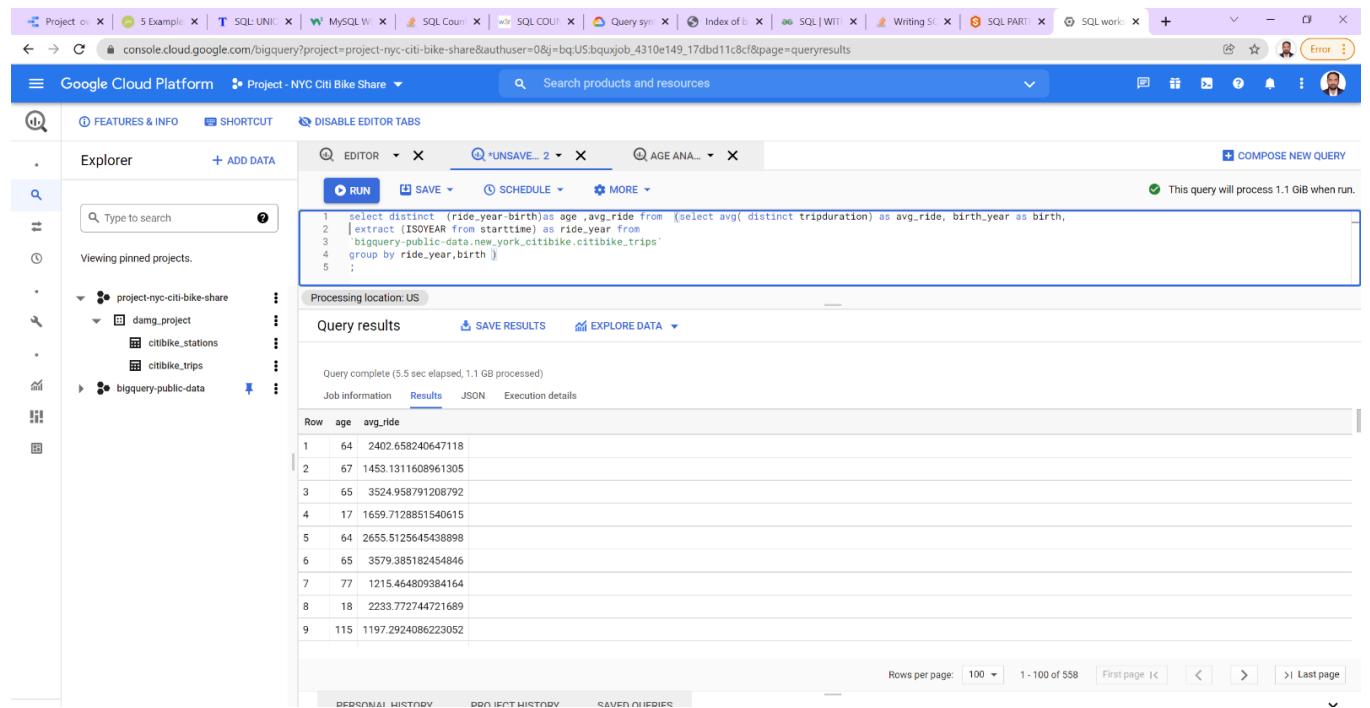
This screenshot is identical to the one above, showing the Google Cloud Platform BigQuery interface for Dataset 2. The setup, query code, and resulting data table are the same. The data table for Dataset 2 is as follows:

Row	age	total_ride
1	29	813694
2	32	745719
3	27	728920
4	38	445911
5	43	296791
6	21	188381
7	50	274713
8	59	156219
9	67	64338

Dataset 3 –

- Average length of trip per age

Dataset 1 –



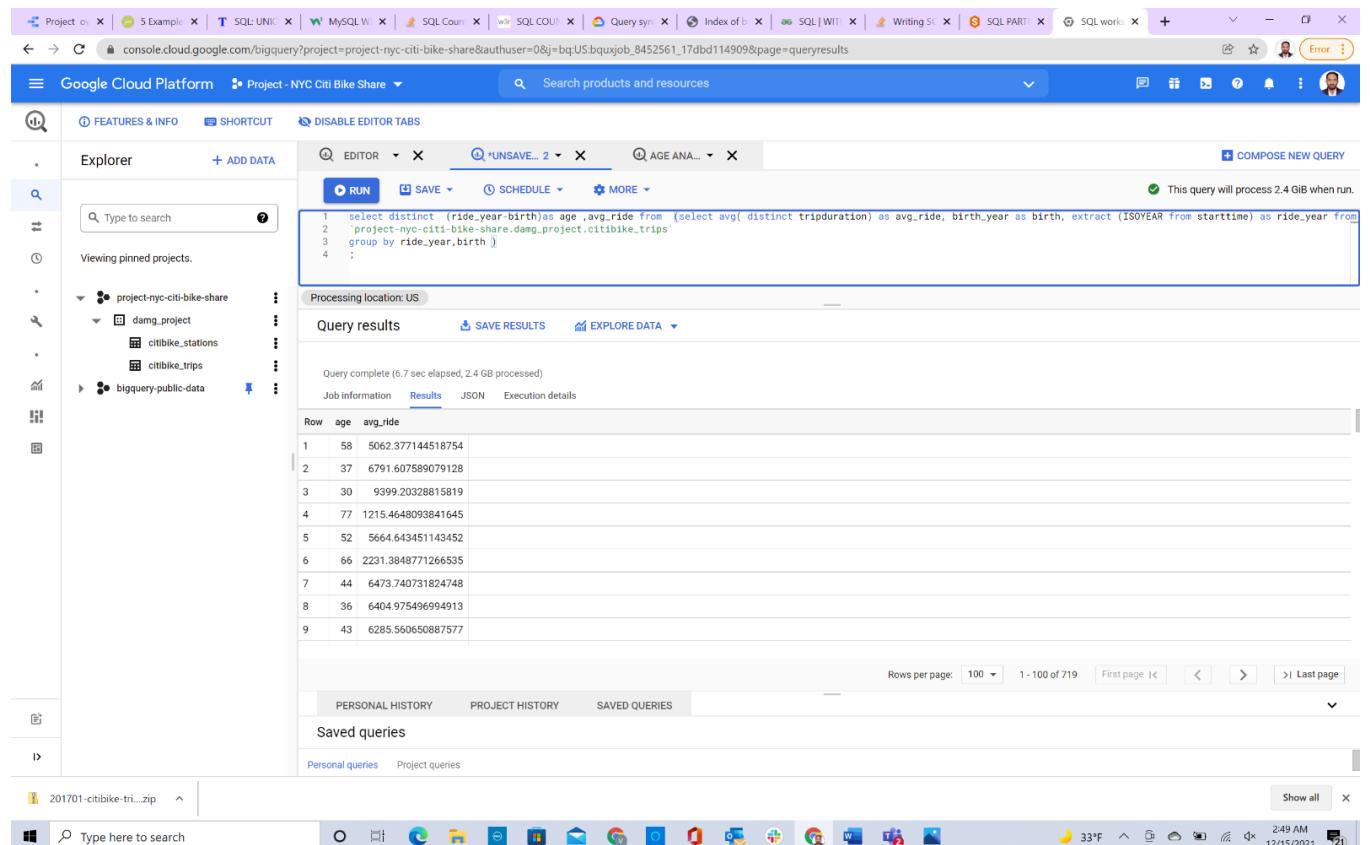
The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'bqpublicdata'. The main area shows a query editor with the following SQL code:

```
1 select distinct (ride_year-birth)as age ,avg_ride from (select avg( distinct tripduration) as avg_ride, birth_year as birth,
2 extract (ISOYEAR from starttime) as ride_year
3 from `bigquery-public-data.new_york_citibike.citibike_trips`
4 group by ride_year,birth )
5 ;
```

The results table shows the average ride duration for different age groups. The data is as follows:

age	avg_ride
64	2402.65824067118
67	1453.1311608961305
65	3524.958791208792
17	1659.7128851540615
64	2655.5125645438898
65	3579.385182454846
77	1215.464809384164
18	2233.772744721689
115	1197.2924086223052

Dataset 2 –



The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'bqpublicdata'. The main area shows a query editor with the same SQL code as Dataset 1:

```
1 select distinct (ride_year-birth)as age ,avg_ride from (select avg( distinct tripduration) as avg_ride, birth_year as birth,
2 extract (ISOYEAR from starttime) as ride_year
3 from `bigquery-public-data.new_york_citibike.citibike_trips`
4 group by ride_year,birth )
5 ;
```

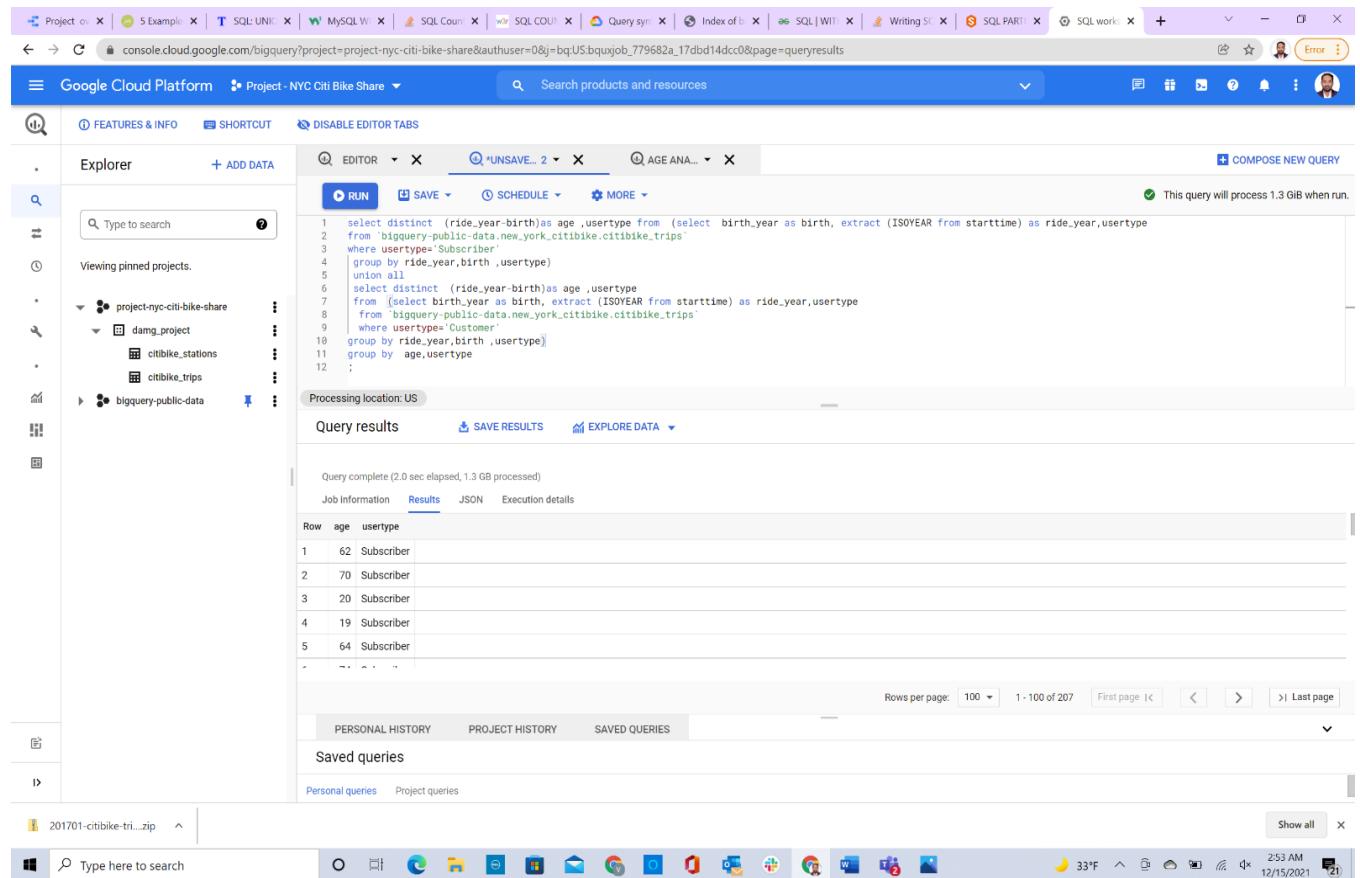
The results table shows the average ride duration for different age groups. The data is as follows:

age	avg_ride
58	5062.377144518754
37	6791.607589079128
30	9399.20328815819
77	1215.4648093841645
52	5664.643451143452
66	2231.3848771266535
77	6473.740731824748
36	6404.975496994913
43	6285.560650887577

Dataset 3 –

- Subscriber versus customer trips – per age

Dataset 1 –



The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays pinned projects, including 'project-nyc-citi-bike-share' which contains 'damp_project' with 'citibike_stations' and 'citibike_trips'. The main area shows a query editor with the following SQL code:

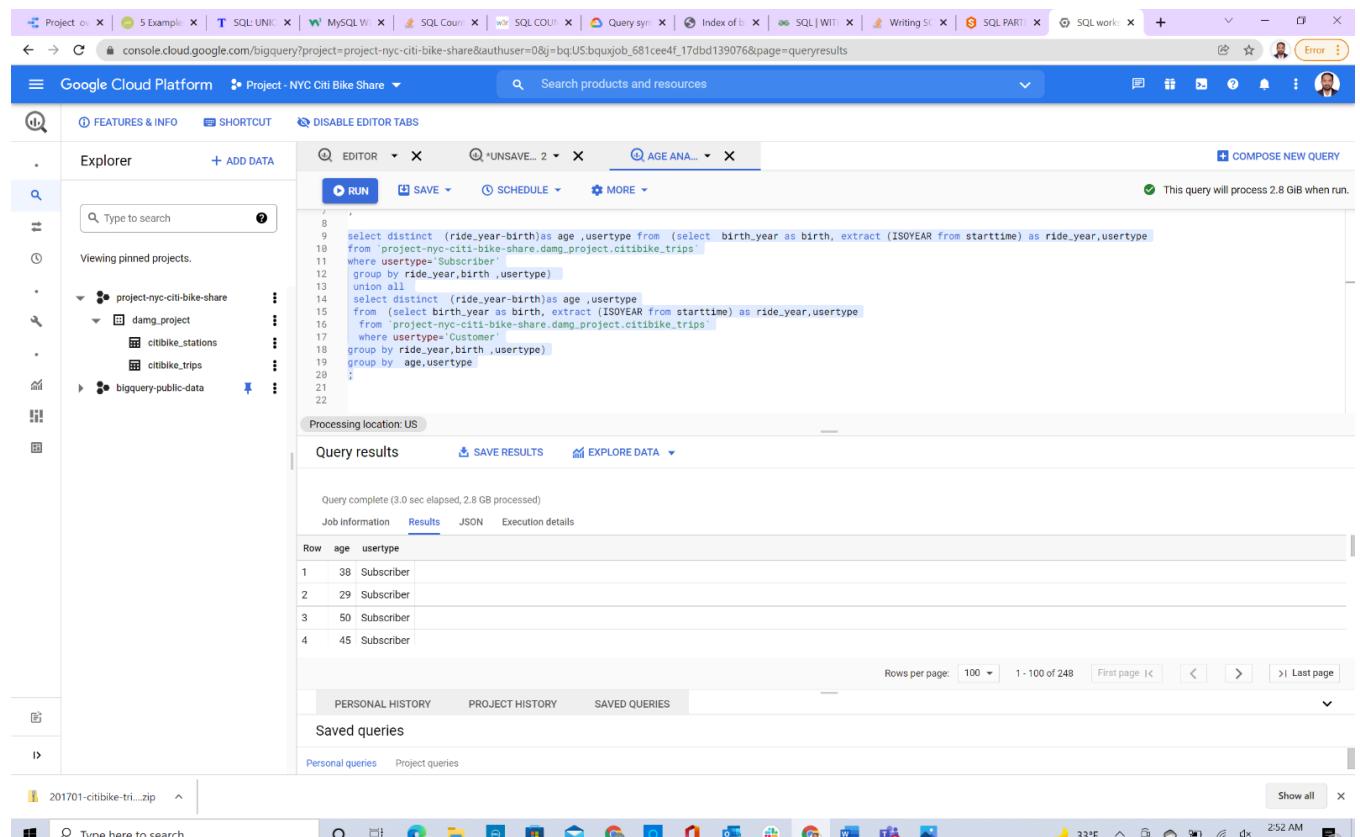
```
1 select distinct (ride_year-birth)as age ,usertype from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
2 from `bigquery-public-data.new_york_citibike.citibike_trips`
3 where usertype='Subscriber'
4 group by ride_year,birth ,usertype)
5 union all
6 select distinct (ride_year-birth)as age ,usertype
7 from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
8 from `bigquery-public-data.new_york_citibike.citibike_trips`
9 where usertype='Customer'
10 group by ride_year,birth ,usertype)
11 group by age,usertype
12 ;
```

The processing location is set to US. The results table shows the following data:

Row	age	usertype
1	62	Subscriber
2	70	Subscriber
3	20	Subscriber
4	19	Subscriber
5	64	Subscriber

There are 207 rows in total. The status bar at the bottom right indicates the time is 2:53 AM on 12/15/2021.

Dataset 2 –



The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays pinned projects, including 'project-nyc-citi-bike-share' which contains 'damp_project' with 'citibike_stations' and 'citibike_trips'. The main area shows a query editor with the following SQL code:

```
1
2
3
4
5
6
7
8
9 select distinct (ride_year-birth)as age ,usertype from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
10 from `project-nyc-citi-bike-share.damp_project.citibike_trips`
11 where usertype='Subscriber'
12 group by ride_year,birth ,usertype)
13 union all
14 select distinct (ride_year-birth)as age ,usertype
15 from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
16 from `project-nyc-citi-bike-share.damp_project.citibike_trips`
17 where usertype='Customer'
18 group by ride_year,birth ,usertype)
19 group by age,usertype
20 ;
21
22
```

The processing location is set to US. The results table shows the following data:

Row	age	usertype
1	38	Subscriber
2	29	Subscriber
3	50	Subscriber
4	45	Subscriber

There are 248 rows in total. The status bar at the bottom right indicates the time is 2:52 AM on 12/15/2021.

- Any trends from 2014 to 2020 regarding above
 - What are the number of trips per age

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for Project, Example, SQL UNION, MySQL View, SQL COUNT, Query syntax, Index of, SQL WITH, Writing SQL, SQL PARTITION, and SQL workbench. The main area displays a query editor with the following SQL code:

```

1 select distinct (ride_year-birth)as age ,total_ride ,ride_year as year from (select count(*) as total_ride, birth.year as birth, extract (ISOYEAR from starttime) as ride_year
2 group by ride_year,birth )
3 group by total_ride,age,year;
4

```

The results section shows a table of data with columns: Row, age, total_ride, and year. The data includes rows such as (90, 36, 359428, 2016), (91, 29, 485878, 2016), (92, 54, 188875, 2016), (93, 25, 388638, 2016), (94, 24, 321820, 2016), (95, 21, 90686, 2016), (96, 76, 5019, 2016), (97, 82, 384, 2016), (98, 78, 3093, 2016), (99, 104, 132, 2016), and (100, 84, 487, 2016).

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface, similar to the first one but with a different query. The top navigation bar includes tabs for Project, Example, SQL UNION, MySQL View, SQL COUNT, Query syntax, Index of, SQL WITH, Writing SQL, SQL PARTITION, and SQL workbench. The main area displays a query editor with the following SQL code:

```

1 select distinct (ride_year-birth)as age ,total_ride ,ride_year as year from (select count(*) as total_ride, birth.year as birth,
2 extract (ISOYEAR from starttime) as ride_year from `bigquery-public-data.new_york_citibike.citibike_trips`
3 group by ride_year,birth )
4 group by total_ride,age,year;

```

A note in the editor says "This query will process 767.8 MB when run." The results section shows a table of data with columns: Row, age, total_ride, and year. The data includes rows such as (1, 22, 46904, 2018), (2, 69, 17417, 2016), (3, 71, 8560, 2015), (4, 70, 20846, 2017), (5, 20, 49001, 2016), and so on.

- Any trends from 2014 to 2020

Average length of trip per age –

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays pinned projects, including 'project-nyc-citi-bike-share' which contains 'dang_project' with datasets 'citibike_stations' and 'citibike_trips', and 'bqquery-public-data'. The main area shows a query editor with the following SQL code:

```

1 select distinct (ride_year-birth)as age ,avg_ride ,ride_year as year from (select avg( distinct tripduration) as avg_ride, birth_year as birth,
2 extract (ISOYEAR from starttime) as ride_year from `bqquery-public-data.new_york_citibike.citibike_trips`
3 group by ride_year,birth )
4 group by avg_ride,age,year;
    
```

The processing location is set to US. Below the code, the 'Query results' section shows the output of the query, which is a table with columns 'Row', 'age', 'avg_ride', and 'year'. The data includes rows for various years and average ride durations, such as 2014, 2015, 2013, 2017, 2016, 2018, 2012, 2014, 2016, 2015, and 2015.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface, similar to the previous one but with a different query. The left sidebar shows the same project structure. The main area shows a query editor with the following SQL code:

```

1 select distinct (ride_year-birth)as age ,avg_ride ,ride_year as year from (select avg( distinct tripduration) as avg_ride, birth_year as birth,
2 extract (ISOYEAR from starttime) as ride_year from `project-nyc-citi-bike-share.dang_project.citibike_trips`
3 group by ride_year,birth )
4 group by avg_ride,age,year;
    
```

A note at the top right of the code area says 'This query will process 2.4 GiB when run.' The processing location is set to US. Below the code, the 'Query results' section shows the output of the query, which is a table with columns 'Row', 'age', 'avg_ride', and 'year'. The data includes rows for various years and average ride durations, such as 2015, 2015, 2015, 2015, 2015, 2015, and 2015.

- Any trends from 2014 to 2020
- Subscriber versus customer trips – per age

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'dang_project'. The main area shows a query editor with the following SQL code:

```

1 select distinct (ride_year-birth)as age ,usertype,ride_year as year from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
2 from `bigquery-public-data.new_york.citibike.citibike_trips`
3 where usertype='Subscriber'
4 group by ride_year,birth ,usertype)
5 union all
6 select distinct (ride_year-birth)as age ,usertype,ride_year as year
7 from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
8 from `bigquery-public-data.new_york.citibike.citibike_trips`
9 where usertype='Customer'
10 group by ride_year,birth ,usertype)
11 group by age,usertype,year
12 ;

```

The results pane shows the following data:

Row	age	usertype	year
1	19	Customer	2017
2	21	Customer	2017
3	21	Customer	2018
4	22	Customer	2018
5	19	Customer	2018
6	65	Customer	2017
7	20	Customer	2017

At the bottom, there is a Windows taskbar with various icons and a system tray showing the date and time.

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar displays the 'Explorer' section with pinned projects like 'project-nyc-citi-bike-share' and 'dang_project'. The main area shows a query editor with the same SQL code as Dataset 1:

```

1 select distinct (ride_year-birth)as age ,usertype,ride_year as year from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
2 from `project-nyc-citi-bike-share.dang_project.citibike_trips`
3 where usertype='Subscriber'
4 group by ride_year,birth ,usertype)
5 union all
6 select distinct (ride_year-birth)as age ,usertype,ride_year as year
7 from (select birth_year as birth, extract (ISOYEAR from starttime) as ride_year,usertype
8 from `project-nyc-citi-bike-share.dang_project.citibike_trips`
9 where usertype='Customer'
10 group by ride_year,birth ,usertype)
11 group by age,usertype,year
12 ;

```

The results pane shows the following data:

Row	age	usertype	year
1	38	Customer	2019
2	35	Customer	2019
3	31	Customer	2019
4	45	Customer	2018
5	42	Customer	2018
6	41	Customer	2018
7	39	Customer	2018

At the bottom, there is a Windows taskbar with various icons and a system tray showing the date and time.

Bike Analysis

- What are the top 5 bikes (bike id) for latest (full) year by
 - Total trips
 - Total time
 - Average trip time

Dataset 1 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query results are displayed in a table:

Row	bikeID	total_trips	total_duration	Avg_tripduration
1	19860	502	553520	1102.62
2	20972	64	51149	799.2
3	32056	755	581716	770.48
4	28692	572	682865	1193.81
5	25249	225	175711	780.93

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The query results are displayed in a table:

Row	bikeID	total_trips	total_duration	Avg_tripduration
1	40979	1440	1536686	1067.14
2	38822	1179	2062500	1749.36
3	34358	817	865457	1059.31
4	32437	823	855601	1039.61
5	41724	1056	1723816	1632.4

Dataset 3 –

The screenshot shows a browser window with multiple tabs open. The active tab is a SQL worksheet titled '[Worksheet]'. The query executed is:

```

1  SELECT bikeID, count(tripduration) AS total_trips, sum(tripduration) AS total_duration, trunc(Avg(tripduration),2) AS Avg_tripduration
2
3  FROM ... ADMIN.NYC_CITIBIKES_2019_2020 group by BIKEID;

```

The results are displayed in a table with columns: bikeid, total_trips, total_duration, and avg_tripduration. The data shows the top 16 bikes based on total trips, with the last row being bikeid 31781.

bikeid	total_trips	total_duration	avg_tripduration
1	27714	1496	1234603
2	35543	810	613774
3	30719	1677	1423193
4	33614	1704	1582381
5	33978	1444	1166217
6	35060	2593	2251248
7	15929	854	702160
8	25370	972	1456382
9	31033	1691	1488697
10	34695	447	431030
11	34857	469	391801
12	33121	1380	1922732
13	20241	1236	1043902
14	18625	1176	1031463
15	20831	1113	1076680
16	31781	1544	1345543

Execution time: 0.012 seconds

- For top bike (by total trips) create a history for the latest (full) year
 - Bike id
 - Route (state and end location)
 - Start time
 - Start location (id, longitude and latitude)
 - Duration

Dataset 1 –

The screenshot shows a browser window with multiple tabs open. The active tab is a BigQuery query editor titled 'Project: overall-year-over-year'.

The query executed is:

```

1
2
3
4  With top AS (SELECT bikeid, count(tripduration), DENSE_RANK() over (order by count(tripduration)desc) as rankn
5   FROM bigquery-public-data.new_york_citibike.citibike_trips
6   WHERE extract (ISOYEAR FROM starttime)>2018 group by 1 qualify rankn=1 order by rankn limit 1000),
7   zip AS (select bikeid,starttime,start_station_name,end_station_name,start_station_id,tripduration,start_station_latitude
8   ,start_station_longitude
9   ,start_station_name
10  FROM bigquery-public-data.new_york_citibike.citibike_trips )
11
12  Select top.bikeid,zip.starttime,zip.start_station_name,zip.end_station_name,zip.tripduration,zip.start_station_id,zip.start_station_latitude,zip.start_station_longitude
13
14  FROM top left join zip on top.bikeid=zip.bikeid;

```

The results are displayed in a table with columns: bikeid, starttime, start_station_name, end_station_name, tripduration, start_station_id, start_station_latitude, and start_station_longitude. The data shows the top 1000 bikes based on total trips, with the last row being bikeid 32221.

bikeid	starttime	start_station_name	end_station_name	tripduration	start_station_id	start_station_latitude	start_station_longitude
1	2017-10-28T14:54:34	Clinton St & 4 Place	Coffey St & Conover St	2789	3396	40.6783563	-74.00014502
2	2018-03-12T14:02:13.619000	Bushwick Ave & Powers St	Berry St & N 8 St	811	3083	40.71247661	-73.94100005
3	2018-02-06T19:31:09.665000	Great Jones St	Bank St & Washington St	742	229	40.72743423	-73.99379025
4	2018-05-08T15:56:49.587000	2 Ave & E 99 St	E 91 St & 2 Ave	312	3338	40.7862586	-73.94552579
5	2018-02-14T13:44:15.298000	Washington Pl & Broadway	St Marks Pl & 1 Ave	262	335	40.72903917	-73.99404649
6	2017-10-14T12:34:26	Pioneer St & Van Brunt St	Atlantic Ave & Furman St	717	3344	40.679043	-74.011169
7	2018-03-11T22:02:34.405000	E 25 St & 1 Ave	1 Ave & E 16 St	244	174	40.7381765	-73.97738662

Query complete (7.3 sec elapsed, 4.5 GB processed)

Dataset 2 –

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for 'Project overall year-over-year g...', 'Query syntax | BigQuery | Go...', and 'SQL workspace - BigQuery - Pro...'. The main window has a blue header with 'Google Cloud Platform' and 'Project - NYC Citi Bike Share'.

The left sidebar shows pinned projects, including 'project-nyc-citi-bike-share' which contains 'damg.project' (with 'citibike_stations' and 'citibike_trips' sub-folders) and 'bigquery-public-data'. The right pane displays a query editor with the following SQL code:

```
1
2
3
4 With top as (SELECT bikeid, count(tripduration), DENSE_RANK() over (order by count(tripduration)desc) as rannk
5 FROM `project-nyc-citi-bike-share.damg.project.citibike_trips`
6 WHERE _extract((ISOYEAR, From starttime))=2019 group by 1 qualify rannk=1 order by rannk Limit 1000,
7 zip as (select bikeid,starttime,start_station_name,end_station_name,start_station_id,tripduration,start_station_latitude
8 ,start_station_longitude
9 FROM `project-nyc-citi-bike-share.damg.project.citibike_trips`)
10
11 Select top.bikeid.zipstarttime,zip.start_station_name,zip.end_station_name,zip.tripduration,zip.start_station_id,zip.start_station_latitude,zip.start_station_longitude
12
13
14 FROM top left join zip on top.bikeid=zip.bikeid;
```

The 'Processing location: US' section indicates the job is complete. The 'Query results' section shows the first 10 rows of data:

Row	bikeid	starttime	start_station_name	end_station_name	tripduration	start_station_id	start_station_latitude	start_station_longitude
1	30657	2019-02-11 08:05:52 UTC	9 Ave & W 22 St	Pershing Square North	944	509	40.7454973	-74.00197139
2	30657	2019-02-28 17:09:04 UTC	Watts St & Greenwich St	Pershing Square North	1291	328	40.72405549	-74.00965965
3	30657	2019-02-06 16:25:23 UTC	Railroad Ave & Kay Ave	Adelphi St & Myrtle Ave	428	2005	40.70531194	-73.97100056
4	30657	2019-02-11 17:05:11 UTC	W 84 St & Broadway	Broadway & E 14 St	1752	3177	40.7867947	-73.977112
5	30657	2019-02-27 00:06:22 UTC	Driggs Ave & N Henry St	Driggs Ave & Lorimer St	273	3106	40.72325	-73.94308
6	30657	2019-02-26 21:36:51 UTC	E 100th St & Nassau Ave	Brinne Ave & N Henry St	965	3648	40.7295674	-73.9492345

The bottom status bar shows the system is at 33°F Partly cloudy, 11:32 AM on 12/15/2021.

Dataset 3 –

The screenshot shows the Oracle Database Actions SQL interface. The top navigation bar includes tabs for 'Nav', 'Team', 'Rich', 'Thar...', 'Ora...', '(160...', 'Final...', 'Inde...', 'Aut...', 'Time...', 'Wha...', 'G tera...', '(6)', 'how...', 'TO...', 'SQL'.

The main window has a blue header with 'ORACLE Database Actions | SQL' and 'ADMIN'.

The left sidebar shows a 'Navigator' with files like pg24, pg30, pg29A, pg27, pg26, pg25. The right pane displays a query editor with the following SQL code:

```
1
2 with temp as (
3 select * from (SELECT bikeid,
4 count(tripduration),
5 row_number() over (order by count(tripduration)desc) as rnm
6 FROM NYC_CITIBIKES_2019_2020
7 WHERE to_char(starttime,'YYYY')=2019
8 group by BIKEID) where rnm=1)
9 select temp.bikeid,ny starttime,ny.start_station_name,ny.end_station_name,ny.start_station_id,ny.tripduration,ny.start_station_latitude,
10 ny.start_station_longitude from temp inner join NYC_CITIBIKES_2019_2020 ny on temp.bikeid=ny.bikeid;
```

The 'Query Result' section shows the first 14 rows of data:

bikeid	starttime	start_station_name	end_station_name	start_station_id	tripduration	start_station_latitude	start_station_longitude	
1	35194	2019-01-05T19:33:2	Lafayette St & Jersey	Grand St & Havemeyer	3427	956	40.724	-73.996
2	35194	2019-01-04T17:32:0	W 33 St & 7 Ave	Broadway & W 122	492	2374	40.750	-73.991
3	35194	2019-01-28T07:06:1	W 53 St & 10 Ave	E 48 St & 5 Ave	480	545	40.767	-73.991
4	35194	2019-01-18T13:46:0	Central Park S & 6 J	E 55 St & 3 Ave	2006	263	40.766	-73.976
5	35194	2019-01-16T18:36:5	Forsyth St & Canal St	Henry St & Grand St	412	540	40.716	-73.994
6	35194	2019-01-18T09:33:4	E 55 St & 3 Ave	Central Park S & 6 J	3223	458	40.759	-73.969
7	35194	2019-01-19T16:19:5	Garfield Pl & 8 Ave	Eastern Pkwy & Franklin	3358	706	40.671	-73.975
8	35194	2019-01-16T13:14:4	Vesey Pl & River Ter	Murray St & West St	327	190	40.715	-74.017
9	35194	2019-01-01T13:26:5	E 31 St & 3 Ave	Duane St & Greenwich	476	1369	40.744	-73.980
10	35194	2019-01-02T19:23:1	St Marks Pl & 1 Ave	E 16 St & 5 Ave	438	774	40.728	-73.986
11	35194	2019-01-04T15:25:1	W 34 St & 11 Ave	12 Ave & W 40 St	525	236	40.756	-74.002
12	35194	2019-01-04T11:37:1	W 33 St & 7 Ave	W 34 St & 11 Ave	492	380	40.750	-73.991
13	35194	2019-01-15T13:39:1	Broadway & W 25 St	Broadway & W 29 St	3641	195	40.743	-73.989
14	35194	2019-01-14T18:11:2	Christopher St & Greenwich	W 34 St & 11 Ave	358	867	40.733	-74.007

The bottom status bar shows the system is at 41°F, 11:32 AM on 12/15/2021.

Anomalies in the Dataset -

1. Here is the screenshot for null value dataset in Bigquery public dataset.

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar shows a project named 'NYC Citi Bike Share' with a folder 'bigquery-public-data' containing datasets like 'new_york_311', 'new_york_citibike', and 'new_york_mv_collisions'. The main editor window contains a query:

```
1 select * from `project-nyc-citi-bike-share.damg_project.citibike_trips`
2 where tripduration is null;
3
```

The results section shows a table with 6 rows of data, all with 'tripduration' values as null. The table has columns: Row, tripduration, starttime, stoptime, start_station_id, start_station_name, start_station_latitude, start_station_longitude, end_station_id, end_station_name, end_station_latitude, end_station_longitude, bikeid, usertype, and bl.

Row	tripduration	starttime	stoptime	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bikeid	usertype	bl
1	null	null	null	null		null	null	null		null	null	null	null	
2	null	null	null	null		null	null	null		null	null	null	null	
3	null	null	null	null		null	null	null		null	null	null	null	
4	null	null	null	null		null	null	null		null	null	null	null	
5	null	null	null	null		null	null	null		null	null	null	null	
6	null	null	null	null		null	null	null		null	null	null	null	

2. Reporting missing data for year 2015 by providing row count for BigQuery Public Dataset vs Final Project data set

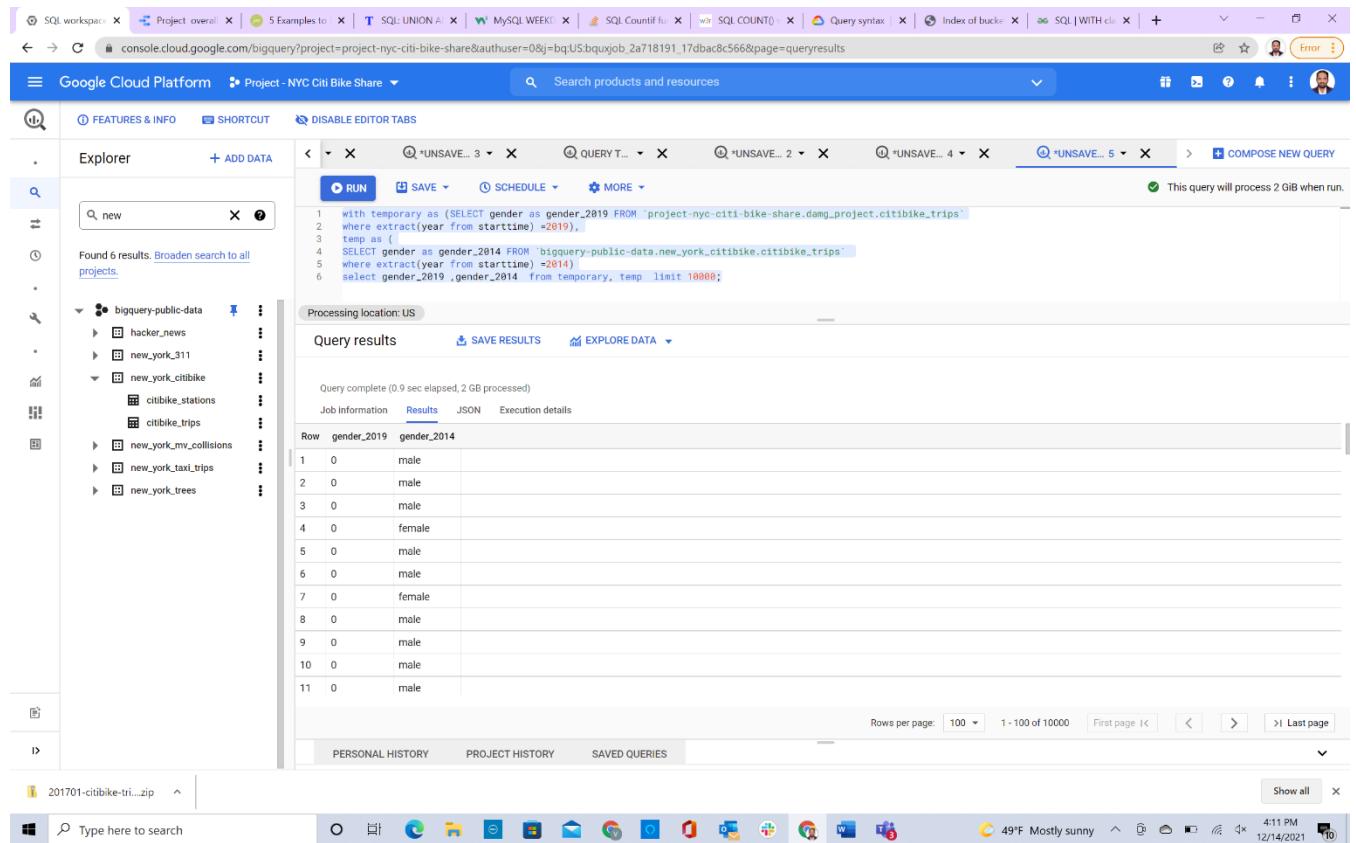
The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar shows a project named 'NYC Citi Bike Share' with a folder 'bigquery-public-data' containing datasets like 'new_york_311', 'new_york_citibike', and 'new_york_mv_collisions'. The main editor window contains a complex query:

```
1 with temporary as (select count(*) as total_row ,extract (ISOYEAR from starttime) as year FROM `bigquery-public-data.new_york_citibike.citibike_trips`
2 group by year
3 having year =2015,
4 temp as (select count(*) as total_updated_row ,extract (ISOYEAR from starttime) as years FROM `project-nyc-citi-bike-share.damg_project.citibike_trips`
5 group by years
6 having years =2015)
7 select total_row,total_updated_row from temporary,temp
8 ;
9
10
```

The results section shows a table with 1 row of data, comparing the total number of trips in 2015 from both datasets. The table has columns: Row, total_row, and total_updated_row.

Row	total_row	total_updated_row
1	10014548	16281428

3. There is inconsistency in gender column in public dataset and the final project data set for different year .Where gender column contains value : "male", "female",0,1,2.



The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes tabs for various SQL-related topics like 'SQL workspace', 'Project overall', '5 Examples to...', 'SQL UNION ALL', 'MySQL WEEK()', 'SQL COUNTif()', 'SQL COUNT()', 'Query syntax', 'Index of books', 'SQL WITH clause', and 'Error'. Below the navigation bar, the main header says 'Google Cloud Platform' and 'Project - NYC Citi Bike Share'. A search bar is present above the query editor.

The query editor contains a code snippet:

```

1 with temporary as (SELECT gender as gender_2019 FROM `project-nyc-citi-bike-share.damg_project.citibike_trips`
2 where extract(year from starttime) >=2019),
3 temp as (
4 SELECT gender as gender_2014 FROM `bigquery-public-data.new_york_citibike.citibike_trips`
5 where extract(year from starttime) =2014)
6 select gender_2019 ,gender_2014 from temporary, temp limit 10000;
  
```

The 'Query results' section shows the output of the query. The table has three columns: Row, gender_2019, and gender_2014. The data is as follows:

Row	gender_2019	gender_2014
1	0	male
2	0	male
3	0	male
4	0	female
5	0	male
6	0	male
7	0	female
8	0	male
9	0	male
10	0	male
11	0	male

Below the table, there are buttons for 'SAVE RESULTS' and 'EXPLORE DATA'. The bottom of the screen shows a taskbar with various icons and a system tray indicating the date and time as 12/14/2021 at 4:11 PM.