/\*

#1. Find classic and docked bike trips that do not have a

beginning or ending docking station

\*/

WITH combined\_data AS

(

SELECT \*

FROM (

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_January

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_February

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_March

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_April

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_May

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_June

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_July

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_August

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_September

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_October

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_November

UNION ALL

SELECT \* FROM cycling-tripdata-2022.Tripdata\_2022.Tripdata\_December

)

),

-----------------------------------------------------------------------------------------------------

----------------------------------------CLEAN THE DATA-----------------------------------------------

-----------------------------------------------------------------------------------------------------

----------#1. Find classic/docked bike trips which do not begin or end at a docking station----------

CREATE TABLE Tripdata\_2022.null\_station\_names AS (SELECT ride\_id AS bad\_ride\_id

FROM (SELECT ride\_id, start\_station\_name, start\_station\_id,end\_station\_name, end\_station\_id

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_tripdata

WHERE rideable\_type = 'docked\_bike' OR rideable\_type = 'classic\_bike')

WHERE start\_station\_name IS NULL AND start\_station\_id IS NULL OR end\_station\_name IS NULL AND end\_station\_id IS NULL);

/\*

Shows all rides where start/end station name is null for classic/docked bikes

#2. Remove null rows from previous query, removing rows that do not have starting and ending

Latitude and Longitude data

\*/

CREATE TABLE Tripdata\_2022.cleaned\_station\_names AS

(

SELECT \*

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_tripdata

LEFT JOIN cycling-tripdata-2022.Tripdata\_2022.null\_station\_names

ON ride\_id = bad\_ride\_id

WHERE bad\_ride\_id IS NULL AND

start\_lat IS NOT NULL AND

start\_lng IS NOT NULL AND

end\_lat IS NOT NULL AND

end\_lng IS NOT NULL

);

/\*

#3. Replace docked\_bike with classic\_bike

\*/

SELECT ride\_id, REPLACE (rideable\_type, 'docked\_bike', 'classic\_bike') AS ride\_type

FROM cycling-tripdata-2022.Tripdata\_2022.cleaned\_station\_names

/\*

#4. For electric bikes with null station names, null will be replaced with Bike\_Locked instead

\*/

SELECT started\_at, ended\_at, IFNULL(NULL, "Bike Locked")

FROM cycling-tripdata-2022.Tripdata\_2022.cleaned\_station\_names;

/\*

#5. Create a new column for the date and ride length

\*/

CREATE TABLE cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned AS

(

SELECT ride\_id,

REPLACE(rideable\_type, 'docked\_bike', 'classic\_bike') AS ride\_type,

started\_at,

ended\_at,

IFNULL(TRIM(REPLACE(start\_station\_name, '(Temp)', '')), 'On Bike Lock') AS starting\_station\_name,

IFNULL(TRIM(REPLACE(end\_station\_name, '(Temp)', '')), 'On Bike Lock') AS ending\_station\_name,

CASE

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 1 THEN 'Sun'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 2 THEN 'Mon'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 3 THEN 'Tues'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 4 THEN 'Wed'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 5 THEN 'Thur'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 6 THEN 'Fri'

ELSE'Sat'

END AS day,

CASE

WHEN EXTRACT(MONTH FROM started\_at) = 1 THEN 'Jan'

WHEN EXTRACT(MONTH FROM started\_at) = 2 THEN 'Feb'

WHEN EXTRACT(MONTH FROM started\_at) = 3 THEN 'Mar'

WHEN EXTRACT(MONTH FROM started\_at) = 4 THEN 'Apr'

WHEN EXTRACT(MONTH FROM started\_at) = 5 THEN 'May'

WHEN EXTRACT(MONTH FROM started\_at) = 6 THEN 'Jun'

WHEN EXTRACT(MONTH FROM started\_at) = 7 THEN 'July'

WHEN EXTRACT(MONTH FROM started\_at) = 8 THEN 'Aug'

WHEN EXTRACT(MONTH FROM started\_at) = 9 THEN 'Sept'

WHEN EXTRACT(MONTH FROM started\_at) = 10 THEN 'Oct'

WHEN EXTRACT(MONTH FROM started\_at) = 11 THEN 'Nov'

ELSE 'Dec'

END AS month,

EXTRACT(DAY FROM started\_at) as day,

EXTRACT(YEAR FROM started\_at) AS year,

TIMESTAMP\_DIFF(ended\_at, started\_at, MINUTE) AS ride\_time\_minutes,

start\_lat,

start\_lng,

end\_lat,

end\_lng,

member\_casual AS member\_type

FROM cycling-tripdata-2022.Tripdata\_2022.cleaned\_station\_names

);

/\*

Ride type and total amount

\*/

SELECT ride\_type, member\_type, count(\*) AS amount\_of\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

GROUP BY ride\_type, member\_type

ORDER BY member\_type, amount\_of\_rides DESC;

/\*

Monthly amount of rides

\*/

SELECT member\_type, month, COUNT(\*) AS monthly\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

GROUP BY member\_type, month;

/\*

Daily amount of rides

\*/

SELECT member\_type, day\_of\_week, COUNT(\*) AS daily\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

GROUP BY member\_type, day\_of\_week;

/\*

Hourly amount of rides

\*/

SELECT member\_type, EXTRACT(HOUR FROM started\_at) AS time\_daily, COUNT(\*) AS hourly\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

GROUP BY member\_type, time\_daily;

/\*

Avg ride length per day

\*/

SELECT member\_type, day\_of\_week,ROUND(AVG(ride\_time\_minutes),0) AS average\_time,

AVG(AVG(ride\_time\_minutes)) OVER(PARTITION BY member\_type) AS combined\_avg\_ride\_time

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

GROUP BY member\_type, day\_of\_week;

/\*

Starting station for casuals

\*/

SELECT starting\_station\_name, AVG(start\_lat) AS start\_lat, AVG(start\_lng) AS start\_lng,

COUNT(\*) AS number\_of\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

WHERE member\_type = 'casual' AND starting\_station\_name != 'On Bike Lock'

GROUP BY starting\_station\_name;

/\*

Starting station for members

\*/

SELECT starting\_station\_name, AVG(start\_lat) AS start\_lat, AVG(start\_lng) AS start\_lng,

COUNT(\*) AS number\_of\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

WHERE member\_type = 'member' AND starting\_station\_name != 'On Bike Lock'

GROUP BY starting\_station\_name;

/\*

Ending station for casuals

\*/

SELECT ending\_station\_name, AVG(end\_lat) AS end\_lat, AVG(end\_lng) AS end\_lng,

COUNT(\*) AS number\_of\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

WHERE member\_type = 'casual' AND starting\_station\_name != 'On Bike Lock'

GROUP BY ending\_station\_name;

/\*

Ending station for members

\*/

SELECT ending\_station\_name, AVG(end\_lat) AS end\_lat, AVG(end\_lng) AS end\_lng,

COUNT(\*) AS number\_of\_rides

FROM cycling-tripdata-2022.Tripdata\_2022.combined\_data\_cleaned

WHERE member\_type = 'member' AND starting\_station\_name != 'On Bike Lock'

GROUP BY ending\_station\_name;