/\*

First this code will be used to combine all data from January 1, 2022 to December 31, 2022

into a single combined table labeled combined\_tripdata

\*/

CREATE TABLE Tripdata\_2022.combined\_tripdata 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

);

/\* NOTES:

As shown above the 'SELECT \*' query returned 5,667,717 rows.

As previously calculated the sum of the rows prior to being combined are the same before as they are after.

A UNION ALL was used because it keeps all the rows from the multiple tables and the UNION ALL will remove

all duplicate data within the combined tables.

---------------Analyze all columns from left to right for cleaning----------------------------------------------

#1.ride\_id:

\*/

SELECT LENGTH(ride\_id), count(\*)

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

GROUP BY LENGTH(ride\_id);

SELECT COUNT (DISTINCT ride\_id)

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

/\* NOTES:

Distinct variable was used on ride\_id so no cleaning is necessary

COUNT function was used to determine that ride\_id is 16 characters long

#2: Check for avaialable ride types

\*/

SELECT DISTINCT rideable\_type

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

/\* Notes: As shown above there are 3 available ride types in the data, however, docked\_bike

is another name for classic\_bike. Therefore these two data types should be combined

#3. Do a check on the started\_at and ended\_at data sets to make sure

we are only using data where rides are greater than 1 minute but less

than 24 hours.

\*/

SELECT \*

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

WHERE TIMESTAMP\_DIFF(ended\_at, started\_at, MINUTE) <= 1 OR

TIMESTAMP\_DIFF(ended\_at, started\_at, MINUTE) >= 1440;

/\*

#4. Cleaning up naming errors for start\_station name and ID and end\_station name and ID

\*/

SELECT start\_station\_name, COUNT(\*)

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

GROUP BY start\_station\_name

ORDER BY start\_station\_name;

SELECT end\_station\_name, COUNT(\*)

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

GROUP BY end\_station\_name

ORDER BY end\_station\_name;

SELECT COUNT(DISTINCT(start\_station\_name)) AS startName,

COUNT(DISTINCT(end\_station\_name)) AS endName,

COUNT(DISTINCT(start\_station\_id)) AS startID,

COUNT(DISTINCT(end\_station\_id)) AS endID,

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

/\*

Notes:

-Start and End station names need cleaning

-Remove whitespace from station names

-Some of the stations that are listed are repair stations/maintenance stops, these can be removed

#5. Check for Nulls in station names

\*/

SELECT rideable\_type, count(\*) as num\_of\_rides

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

WHERE start\_station\_name IS NULL AND start\_station\_id IS NULL OR

end\_station\_name IS NULL AND end\_station\_id IS NULL

GROUP BY rideable\_type;

/\*

#6. Check for Nulls in start\_lat/lng and end\_lat/lng

\*/

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

WHERE start\_lat IS NULL OR

start\_lng IS NULL OR

end\_lat IS NULL OR

end\_lng IS NULL;

/\*

Notes:

- These rows will be removed, all rides should have location points in order to be used for analysis

#7. Verify that there are only 2 member types in member\_casual column

\*/

SELECT DISTINCT member\_casual

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

/\*

Notes:

-The only member types that are listed are member and casual

\*/