Google Data Analytics Capstone



Bike-share company analysis

Case Study: Cyclistic Bike-Share

Creating a database for the analysis using BigQuery SQL.

Created a table with the columns that it will be used for the analysis:

```
CREATE TABLE cyclisticdata-22.cyclistic_bike_share.rides_data (
ride_id string,
rideable_type string,
started_at timestamp,
ended_at timestamp,
start_station_id string,
end_station_id string,
member_casual string

);

Query completed.
```

Combined the tables from June 2021 to May 2022 to the new table:

```
SAVE ▼
                           +2 SHARE ▼

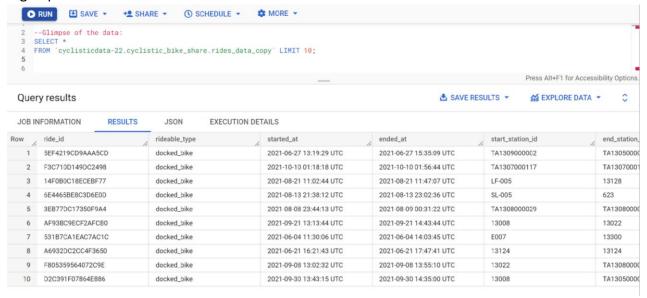
    SCHEDULE ▼

                                                                 MORE -
                                                                                                                               This query will process 409.01 MB when run.
    INSERT INTO `cyclisticdata-22.cyclistic_bike_share.rides_data`(ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id,
    {\tt SELECT} \  \, {\tt ride\_id}, \  \, {\tt rideable\_type}, \  \, {\tt started\_at}, \  \, {\tt ended\_at}, \  \, {\tt start\_station\_id}, \  \, {\tt end\_station\_id}, \  \, {\tt member\_casual}
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>`cyclisticdata-22.cyclistic_bike_share.2021_06`</u>
       UNION ALL
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>`cyclisticdata-22.cyclistic_bike_share.2021_07</u>
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>`cyclisticdata-22.cyclistic_bike_share.2021_08`</u>
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>`cyclisticdata-22.cyclistic_bike_share.2021_09`</u>
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>`cyclisticdata-22.cyclistic_bike_share.2021_10`</u>
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM `cyclisticdata-22.cyclistic_bike_share.2021_11`
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>`cyclisticdata-22.cyclistic_bike_share.2021_12`</u>
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>'cyclisticdata-22.cyclistic_bike_share.2022_01'</u>
UNION ALL
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>'cyclisticdata-22.cyclistic_bike_share.2022_02'</u>
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>'cyclisticdata-22.cyclistic_bike_share.2022_03'</u>
UNION ALL
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
35
36
       FROM <u>'cyclisticdata-22.cyclistic_bike_share.2022_04</u>
UNION ALL
       SELECT ride_id, rideable_type, started_at, ended_at, start_station_id, end_station_id, member_casual
       FROM <u>`cyclisticdata-22.cyclistic_bike_share.2022_05`</u>
```

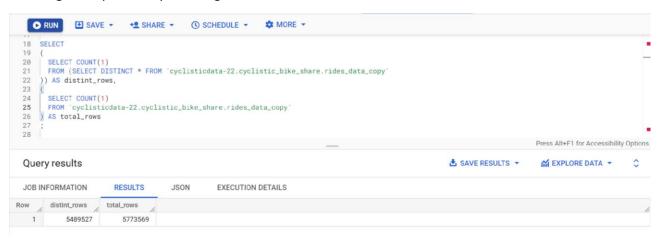
It was created as a duplicate table for backup, in BigQuery there is an option to copy the table.

PROCESS

A glimpse of the data:

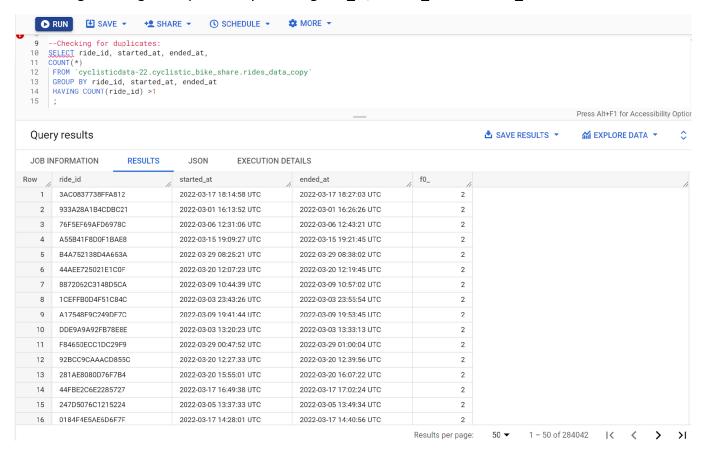


Checking for duplicates by counting distinct rows and total rows:



It can be noticed that there are less distinct rows than total rows.

Continuing checking for duplicates by counting ride id, started at and ended at that are more than once:

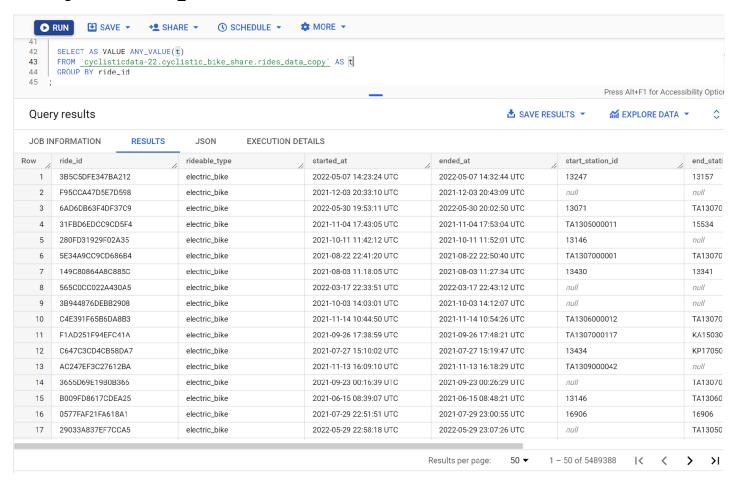


It was found the duplicates have the 'started_at' after the 'ended_at'. As there is a backup table, it can confidently delete the duplicates.

Deleting the 'started at' after the 'ended at':



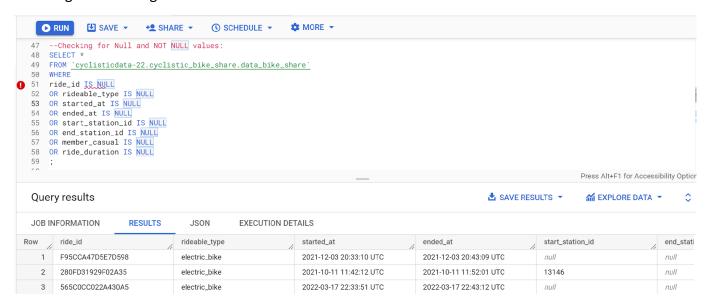
Selecting the distinct ride id:



Checking the number of rows can be observed that after removing the 'started_at' after 'ended_at', the number of rows is less than the distinct number of rows counted earlier.

The result was saved as a new table data_bike_share. Also, it was observed that start_station_id and end station id contain NULL values.

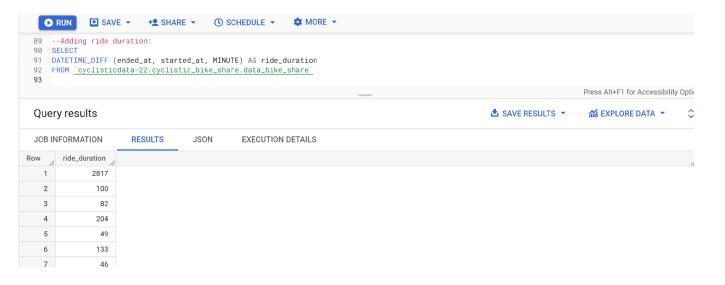
Checking and deleting NULL values:



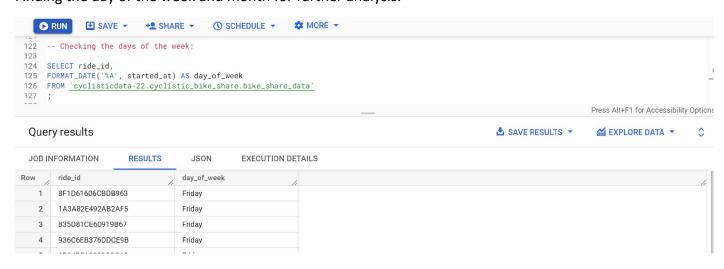
Deleting the NULL values:

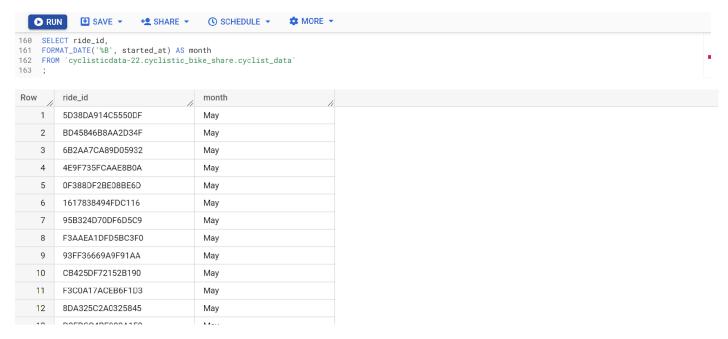


Add a new column with ride duration:

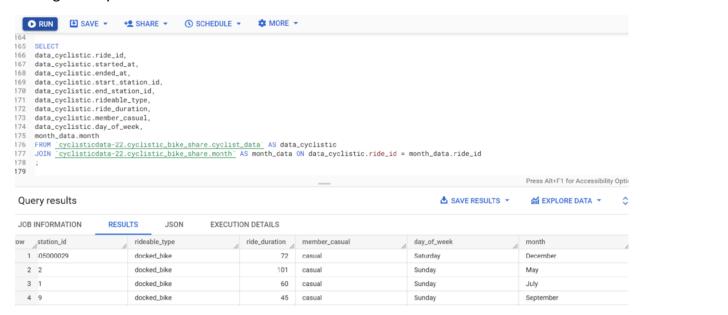


Finding the day of the week and month for further analysis:

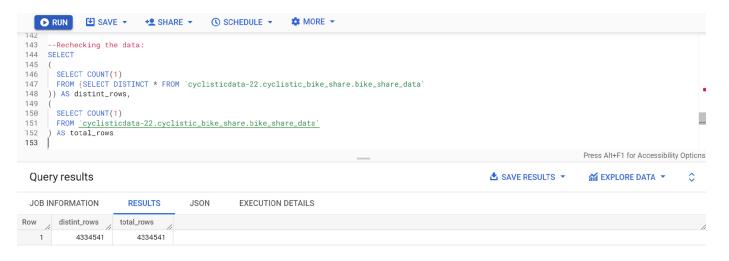




Joining the day of the week and month to the table:



Rechecking the data:



It was observed that the rows are distinct.