CYCLISTIC BIKESHARE CASE STUDY

This is the capstone case study I worked on as part of the Google Data Analytics Certificate course offered by Coursera .

The data was downloaded via this link -https://divvy-tripdata.s3.amazonaws.com/index.html which has been made available by Motivate International Inc. under this license. Data for the period **May 2020 to May 2021** (730 MB data) was taken up for the study. This data was available as CSV files, which were downloaded and then those individual files were uploaded to **Bigquery-Google Cloud Platform** for data preparation and cleaning process in SQL.

The following codes were used to prepare and clean the data using BigQuery SQL.

1. To consolidate all the data into one table / view making data consistent wherever needed.
2. To get the month and year from the timestamp instead of the day
3. To check whether there are any rides with duration as zero or negative
4. To check for missing values in other column
5. To check missing start station name or end station name
6. To get to the final data having no NULL values
7. To know the number of casual and member riders
8. To know the average ride duration of casual and member riders
9. To know the average ride distance of casual and member riders
10. To know the rides taken by casual and member riders in different months

---

Code 1 : To consolidate all the data into one table / view making data consistent wherever needed.

```
WITH
  temp_2020 AS (     -- Temp table 1- may2020-nov2020
    SELECT
      ride_id,
      rideable_type,
      started_at,
      ended_at,
      start_station_name,
      CAST(start_station_id AS STRING) AS start_id,
      end_station_name,
      CAST(end_station_id AS STRING) AS end_id,        -- to make data consistent
      start_lat,
      start_lng,
      end_lat,
      end_lng,
      member_casual
    FROM (
        SELECT *
        FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202005`
        UNION ALL
        SELECT *
        FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202006`
        UNION ALL
        SELECT *
        FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202007`
        UNION ALL
        SELECT *
        FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202008`
        UNION ALL
        SELECT *
        FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202009`
        UNION ALL
        SELECT *
        FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202010`
        UNION ALL
        SELECT *
        FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202011`)
        ),
  temp_all AS (         -- temp table 2:data from dec 2020,jan-may 2021,and temp table 1
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202012`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202101`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202102`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202103`
    UNION ALL
```

```sql
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202104`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202105`
    UNION ALL
    SELECT *
    FROM temp_2020
    ),
  temp_metrics AS (      --temp table 3 : adding new metrics using data from temp table 2
SELECT
      ride_id,
      TIMESTAMP_DIFF(ended_at, started_at, minute) AS ride_total_minute,
ST_GEOGPOINT(start_lng, start_lat) AS start_point,
      ST_GEOGPOINT(end_lng, end_lat) AS end_point,
      CASE
            WHEN EXTRACT(DAYOFWEEK FROM started_at) = 1 THEN 'Sunday'
            WHEN EXTRACT(DAYOFWEEK FROM started_at) = 2 THEN 'Monday'
            WHEN EXTRACT(DAYOFWEEK FROM started_at) = 3 THEN 'Tuesday'
            WHEN EXTRACT(DAYOFWEEK FROM started_at) = 4 THEN 'Wednesday'
            WHEN EXTRACT(DAYOFWEEK FROM started_at) = 5 THEN 'Thursday'
            WHEN EXTRACT(DAYOFWEEK FROM started_at) = 6 THEN 'Friday'
            WHEN EXTRACT(DAYOFWEEK FROM started_at) = 7 THEN 'Saturday'
            END AS start_day                    --To get the start day names of the rides
    FROM temp_all
    )
SELECT
  a.ride_id,
  a.rideable_type,
  b.start_day,
  a.started_at,
  a.ended_at,
  b.ride_total_minute,
  a.start_station_name,
  a.start_station_id,
  a.end_station_name,
  a.end_station_id,
  ST_DISTANCE(b.start_point, b.end_point) AS ride_distance,
  a.member_casual
FROM
  temp_all AS a
JOIN
  temp_metrics AS b
ON a.ride_id = b.ride_id
```

Code 2 :To get the month and year from the timestamp instead of the day

```
SELECT
      ride_id,
      TIMESTAMP_DIFF(ended_at, started_at, minute) AS ride_total_minute,
      ST_GEOGPOINT(start_lng, start_lat) AS start_point,
      ST_GEOGPOINT(end_lng, end_lat) AS end_point,
      FORMAT_DATETIME("%B,%Y",started_at) as MONTH

    FROM temp_all
    )
SELECT
  a.ride_id,
  a.rideable_type,
  b.start_day,
  a.started_at,
  a.ended_at,
  b.ride_total_minute,
  a.start_station_name,
  a.start_station_id,
  a.end_station_name,
  a.end_station_id,
  ST_DISTANCE(b.start_point, b.end_point) AS ride_distance,  ( compute distance in meters)
  a.member_casual
FROM
  temp_all AS a       (from temp table 2)
JOIN
  temp_metrics AS b        (from temp table 3)
ON a.ride_id = b.ride_id
```

After the above queries were run , the results were saved as a new table in BigQuery itself and named as Merged_data.

Code 3 : To check whether there are any rides with duration as zero or negative

```
SELECT *
FROM `bikeshare-project-336114.Cyclistic_data.Merged_data`
WHERE
  ride_total_seconds <= 0
```

This resulted in the number of rides having negative or zero ride duration which could hamper the analysis and results.

Code 4 :To check for missing values in other column

```
SELECT *
FROM `bikeshare-project-336114.Cyclistic_data.Merged_data`
WHERE
  start_day IS NULL
  OR
  started_at IS NULL
  OR
  ended_at IS NULL
  OR
  member_casual IS NULL
```

There were no results for this query i.e. none of the records in the above mentioned columns were NULL.

Code 5 : To check missing start station name or end station name

```
SELECT *
FROM `bikeshare-project-336114.Cyclistic_data.Merged_data`
WHERE
  start_station_name IS NULL
  OR
  end_station_name IS NULL
```

Resulted in 313753 rows where the values are missing.

Code 6 : To get to the final data having no NULL values

```
WITH
  temp_2020 AS (
    SELECT
      ride_id,
      rideable_type,
      started_at,
      ended_at,
      start_station_name,
      CAST(start_station_id AS STRING) AS start_id,
       end_station_name,
      CAST(end_station_id AS STRING) AS end_id,
      start_lat,
      start_lng,
      end_lat,
      end_lng,
      member_casual
    FROM (
         SELECT *
         FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202005`
         UNION ALL
         SELECT *
         FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202006`
         UNION ALL
         SELECT *
         FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202007`
         UNION ALL
         SELECT *
         FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202008`
         UNION ALL
         SELECT *
         FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202009`
         UNION ALL
         SELECT *
         FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202010`
         UNION ALL
         SELECT *
         FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202011`)
         ),
  temp_all AS (
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202012`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202101`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202102`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202103`
    UNION ALL
    SELECT *
```

```
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202104`
    UNION ALL
    SELECT *
    FROM `bikeshare-project-336114.Cyclistic_data.divvy_tripdata_202105`
    UNION ALL
    SELECT *
    FROM temp_2020
    ),
  temp_metrics AS (
    SELECT
      ride_id,
      TIMESTAMP_DIFF(ended_at, started_at, MINUTE) AS ride_total_minutes,
      ST_GEOGPOINT(start_lng, start_lat) AS start_point,
      ST_GEOGPOINT(end_lng, end_lat) AS end_point
      FORMAT_DATETIME("%B,%Y",started_at) as MONTH
    FROM temp_all
    )
SELECT
  a.ride_id,
  a.rideable_type,
  b.MONTH ,
  a.started_at,
  a.ended_at,
  b.ride_total_minutes,
  a.start_station_name,
  a.start_station_id,
  a.end_station_name,
  a.end_station_id,
  ST_DISTANCE(b.start_point, b.end_point) AS ride_distance,
  a.member_casual
FROM
  temp_all AS a
JOIN
  temp_metrics AS b
ON a.ride_id = b.ride_id
WHERE
   ride_total_minutes > 0
   AND
   start_station_name IS NOT NULL
   AND
   end_station_name IS NOT NULL
```

After the above query was run , the results were saved as a new table in BigQuery itself and named as final_data.

Code 7 :To know the number of casual and member riders

```
select
 count(*) as Total,
 countif(member_casual='member') as Member,
 countif(member_casual='casual') as Casual
from
`bikeshare-project-336114.Cyclistic_data.final_data`;
```

Code 8 :To know the average ride duration of casual and member riders

```
SELECT
member_casual,
avg (ride_total_minutes) as avg_ride_duartion,
from `bikeshare-project-336114.Cyclistic_data.final_data`
group by member_casual
```

Code 9 :To know the average ride distance of casual and member riders

```
SELECT
member_casual,
avg (ride_distance) as avg_ride_distance,
from `bikeshare-project-336114.Cyclistic_data.final_data`
group by member_casual
```

Code 10 :To know the rides taken by casual and member riders in different months

```
SELECT MONTH,
countif(member_casual='member') as Member,
 countif(member_casual='casual') as Casual,
from `bikeshare-project-336114.Cyclistic_data.Final_data_month`
group by MONTH
ORDER BY MONTH
```