Cyclistic Capstone Project (Google Analytics course)

This capstone is a project part of the Google analytics course. The information was provided by Motivate International inc. Cyclistic is a fictional company, therefore the data set will have a different name(Divy-tripdata).

1-The objective

The director of marketing for Cyclistic believes the company's future success depends on maximizing the number of annual memberships. The analytics team needs to understand how casual riders and annual members use Cyclistic bikes differently. Once they have this information, the analytics team will design a new marketing strategy to convert casual riders into annual members.

2-The Prepare phase

The information was downloaded from the actual website (https://ride.divvybikes.com/system-data),then information was uploaded to SQL BigQuery. The information is the record of the year 2022 and it's organized by months. There were some anomalies in the data names, these anomalies were fixed before uploading it to BigQuery.

3-The Process phase

First we need to inspect the information for any inconsistencies. Created a column named "ride_length" to identify how long the rides are. Also added a column named "day_of_week" to identify what days of the week are the busiest. Also created a column for months to identify which are the busiest months.Removed any nulls from the information and cleaned the information.

4- The Analyze phase

During this phase we analyzed the information using SQL queries. To uncover trends that were used for our visualizations.

5- Share phase

A Powerpoint presentation was created to share all the discovered trends found during the Analyze phase. We used Google Sheets for the visualizations.

-6 Act phase

The Powerpoint was presented to the marketing team with suggestions and ideas to meet the goal.

SQL Queries

Up loading info into one table

```
select*
from `cyclistic.cyclistic01`
union all

select*
from `cyclistic.cyclistic02`
union all

select*
from `cyclistic.cyclistic03`
union all

select*
from `cyclistic.cyclistic04`
union all

select*
from `cyclistic.cyclistic05`
union all

select*
from `cyclistic.cyclistic06`
union all

select*
from `cyclistic.cyclistic07`
union all

select*
from `cyclistic.cyclistic08`
union all

select*
from `cyclistic.cyclistic09`
```

```
union_all

select*
from_`cyclistic.cyclistic10`
union_all

select*
from_`cyclistic.cyclistic11`
union_all

select*
from_`cyclistic.cyclistic12`
```

Cleaning

- Removing 0 values

```
delete
from `cyclistic.cyclistic-final-table-raw`
WHERE end_lat = 0.0
and end_lng = 0.0
and end_lat= 0
and end_lng =0
and start_lat = 0
and start_lng = 0
and start_lat = 0.0
and start_lng = 0.0
```

- Removing nulls

```
delete
from `cyclistic.cyclistic-final-table-raw`
WHERE start_station_id is null
and start_station_name is null
and end_station_name is null
and  end_station_id is null
```

- Updating inconsistencies

```
UPDATE `cyclistic.cyclistic-final-table-raw`
SET start_station_name = ""
WHERE start_station_name = ""
```

Gathering information

- Member-casual count

```
select member_casual as type_of_member, count(1) as total_rides
from `cyclistic.cyclistic-final-table-raw`
group by member_casual;
```

- Rideable-type count

```
select rideable_type, count(1) as total_rides
from `cyclistic.cyclistic-final-table-raw`
group by rideable_type;
```

- Bike preference

```
SELECT member_type, COUNT (*) AS rider_type_total,
COUNT(CASE WHEN rideable_type = 'electric_bike' THEN 1 END) AS electric,
COUNT(CASE WHEN rideable_type = 'classic_bike' THEN 1 END) AS classic,
COUNT(CASE WHEN rideable_type = 'docked_bike' THEN 1 END)AS docked
FROM `cyclistic.clean3`
GROUP BY member_type
ORDER by member_type
```

- Annually rider-type count

```
SELECT member_type, COUNT (1) AS rides_per_member
 FROM `abes-2.cyclistic.clean3`
```

```
GROUP BY member_type
```

- Rides per weekday

```
SELECT member_type,
COUNT(CASE WHEN day_of_week = 1 THEN 1 END) AS Sunday,
COUNT(CASE WHEN day_of_week = 2 THEN 1 END) AS Monday,
COUNT(CASE WHEN day_of_week = 3 THEN 1 END) AS Tuesday,
COUNT(CASE WHEN day_of_week = 4 THEN 1 END) AS Wednesday,
COUNT(CASE WHEN day_of_week = 5 THEN 1 END) AS Thursday,
COUNT(CASE WHEN day_of_week = 6 THEN 1 END) AS Friday,
COUNT(CASE WHEN day_of_week = 7 THEN 1 END) AS Saturday,
FROM `cyclistic.clean3`
GROUP BY member_type
ORDER by member_type
```

- Rides per month

```
SELECT month_of_ride, COUNT (*) AS monthly_rides,
COUNT(CASE WHEN member_type = 'member' THEN 1 END) AS monthly_member_rides,
COUNT(CASE WHEN member_type = 'casual' THEN 1 END) AS monthly_casual_rides
FROM `cyclistic.clean3`
GROUP BY month_of_ride
ORDER by month_of_ride
```

- Rides over & under an hour

```
SELECT member_type,
count (case when ride_length_mins >= 60 then 1 end) as riders_1hr_or_longer,
count (case when ride_length_mins < 60 then 1 end) as riders_less_then_hour
FROM
  `abes-2.cyclistic.clean3`
  group by member_type
```

- Top 30 start station Casual riders

```
SELECT DISTINCT start_station_name,
COUNT (CASE WHEN member_type = "casual" THEN 1 END) AS casual_users,
FROM `abes-2.cyclistic.clean3`
 group by start_station_name
 order by 2 desc
 limit 30
```

- Top 30 start station Member riders

```
SELECT DISTINCT start_station_name,
COUNT (CASE WHEN member_type = "member" THEN 1 END) AS member_users,
FROM `abes-2.cyclistic.clean3`
 group by start_station_name
 order by 2 desc
 limit 30
```

- New Column time-length

```
SELECT
 ride_id,
  rideable_type,
  started_at,
  ended_at,
  start_station_name,
  start_station_id,
  end_station_name,
  end_station_id,
  start_lat,
  start_lng,
  end_lat,
  end_lng,
  member_casual AS membership_type,
  EXTRACT(DAYofweek FROM started_at) AS day_of_week,
  TIMESTAMP_DIFF(ended_at, started_at, minute) AS ride_length_mins
FROM `cyclistic.cyclistic-final-table-raw`
```