Cyclistic Case Study — SQL Queries
Google Data Analytics Capstone Project
Prepared by: Archana
Date:05-07-2025

This document contains the SQL queries used for data cleaning, transformation, and analysis in the Cyclistic Case Study. The queries were run using Google BigQuery to process and analyze 5 months of bike-sharing data. These SQL queries supported the creation of summary statistics and visualizations in Tableau, Google Sheets, and R.

Contents:


```
-- 1.Remove invalid rides: rows with missing start/end station or negative ride length
SELECT *
FROM `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_riders`
```

```
WHERE start_station_name IS NOT NULL
  AND end_station_name IS NOT NULL
  AND TIMESTAMP_DIFF(ended_at, started_at, MINUTE) >
0;

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

-- 2. Create cleaned view with calculated ride_length,
day_of_week, year_month, hour_of_day
SELECT *,
  TIMESTAMP_DIFF(ended_at, started_at, MINUTE) AS
ride_length,
  FORMAT_DATE('%A', DATE(started_at)) AS day_of_week,
  FORMAT_TIMESTAMP('%Y-%m', started_at) AS
year_month,
  EXTRACT(HOUR FROM started_at) AS hour_of_day
FROM  `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_riders`

WHERE start_station_name IS NOT NULL
  AND end_station_name IS NOT NULL
  AND TIMESTAMP_DIFF(ended_at, started_at, MINUTE) >
0;

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

-- 3. Total rides by member type
SELECT member_casual, COUNT(*) AS total_rides
FROM `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_data`
```

```
GROUP BY member_casual;

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

--4. Average ride length by member type
SELECT member_casual, AVG(ride_length) AS
avg_ride_length
FROM `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_data`;
GROUP BY member_casual;

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

-- 5. Rides by day of week
SELECT member_casual, day_of_week, COUNT(*) AS rides
FROM `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_data`
GROUP BY member_casual, day_of_week
ORDER BY day_of_week;

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

-- 6. Rides by hour of day
SELECT member_casual, hour_of_day, COUNT(*) AS rides
FROM `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_data`
GROUP BY member_casual, hour_of_day
ORDER BY hour_of_day;

-------------------------------------------------------
```

```
-- 7. Rides by month
SELECT member_casual, year_month, COUNT(*) AS rides
FROM `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_data`
GROUP BY member_casual, year_month
ORDER BY year_month;

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

-- 8. Total rides by member type and rideable type
SELECT member_casual, rideable_type, COUNT(*) AS
total_rides
FROM `neural-tangent-464717-a3.cyclistic_bike_data.cleaned_data`
GROUP BY member_casual, rideable_type;

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

-- 9. Example WITH clause combining metrics (if you used
WITH + UNION)
with cleaned_data as(
  select *,
  timestamp_diff(ended_at,started_at,minute) as ride_length,
  format_date('%A',started_at) as day_of_week,
  format_timestamp('%y-%m',started_at) as year_month,
  extract(hour from started_at) as hour_of_day
  from
neural-tangent-464717-a3.cyclistic_bike_data.cleaned_data
),
#total rides by member type
```

```
total_rides as(
select 'Total member' as mrtric, member_casual,count(*) as
total_ride
from cleaned_data
group by member_casual
),

avg_ride as(
  select 'Average ride length' as metric ,
 Member_casual,
 avg(ride_length) as avg_ride_length
  from cleaned_data
  group by member_casual
),


rides_by_weekend as (
  Select 'Rides by weekend' as metric,
member_casual,day_of_week,count(*) as
total_rides_weekdays
  from cleaned_data
  group by member_casual,cleaned_data.day_of_week
),

rides_by_hour as (
  select 'Rides by hours of day' as metric,
member_casual,hour_of_day, count(*) as total_ride_hour
  from cleaned_data
```

```
  group by member_casual,hour_of_day
),

rides_by_month as (
  select 'Rides by the month' as
metric,member_casual,year_month,count(*) as
total_ride_month
  from cleaned_data
  group by member_casual,year_month
)
select * from total_rides
union
select * from avg_ride
#select * from rides_by_weekend
#select * from rides_by_hour
#select * from rides_by_month;
```

Note:
These queries were used in BigQuery for data preparation and analysis as part of the Google Data Analytics Capstone project.