SQL Queries for Cyclistic Data Analysis

Detailed Queries and Explanations

Sean Weissman | August 8, 2024

1. Data Cleaning: Remove Duplicates and Standardize Schema

Purpose: This query cleans the data by removing duplicate records and standardizing the schema across different months.

SQL:

```
WITH cleaned_data AS (
  SELECT DISTINCT
    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
  FROM
    `cyclistic-431802.CyclistRides.all_trips`
  WHERE
    ride_id IS NOT NULL
)
SELECT * FROM cleaned_data;
```

2. Unionize Monthly Tables

Purpose: Combines the individual monthly tables into a single comprehensive table for analysis.

SQL:

```
CREATE OR REPLACE TABLE `cyclistic-431802.CyclistRides.all_trips` AS
SELECT * FROM `cyclistic-431802.CyclistRides.trips_2023_july`
UNION ALL
SELECT * FROM `cyclistic-431802.CyclistRides.trips_2023_august`
UNION ALL
SELECT * FROM `cyclistic-431802.CyclistRides.trips_2023_september`
-- Continue adding the remaining months
;
```

3. Analyze Ride Duration

Purpose: Calculates the average ride duration for casual and member riders to identify patterns.

SQL:

```
SELECT
  member_casual,
  AVG(TIMESTAMP_DIFF(ended_at, started_at, SECOND)) AS
avg_ride_duration_seconds,
  COUNT(*) AS total_rides
FROM
  `cyclistic-431802.CyclistRides.all_trips`
GROUP BY
  member_casual;
```

4. Analyze Bike Type Usage

Purpose: Analyzes the preference for bike types (electric vs. traditional) between casual and member riders.

SQL:

```
SELECT
  rideable_type,
  member_casual,
  COUNT(*) AS total_rides
FROM
  `cyclistic-431802.CyclistRides.all_trips`
GROUP BY
  rideable_type, member_casual;
```

5. Start Points Analysis

Purpose: Identifies the most popular start points for both casual and member riders.

SQL:

```
WITH ranked_start_points AS (
  SELECT
    member_casual,
    start_station_name,
    start_lat,
    start_lng,
    COUNT(*) AS total_rides,
    ROW_NUMBER() OVER (PARTITION BY member_casual ORDER BY COUNT(*)
DESC) AS row_num
  FROM
    `cyclistic-431802.CyclistRides.all_trips`
  GROUP BY
    member_casual,
    start_station_name,
    start_lat,
    start_lng
)
SELECT
  member_casual,
  start_station_name,
  start_lat,
  start_lng,
  total_rides
FROM
  ranked_start_points
WHERE
  row_num <= 10
ORDER BY
  member_casual,
  total_rides DESC;
```

6. End Points Analysis

Purpose: Identifies the most popular end points for both casual and member riders.

SQL:

```
WITH ranked_end_points AS (
  SELECT
    member_casual,
    end_station_name,
    end_lat,
    end_lng,
    COUNT(*) AS total_rides,
    ROW_NUMBER() OVER (PARTITION BY member_casual ORDER BY COUNT(*)
DESC) AS row_num
  FROM
    `cyclistic-431802.CyclistRides.all_trips`
  GROUP BY
    member_casual,
    end_station_name,
    end_lat,
    end_lng
)
SELECT
  member_casual,
  end_station_name,
  end_lat,
  end_lng,
  total_rides
FROM
  ranked_end_points
WHERE
  row_num <= 10
ORDER BY
  member_casual,
  total_rides DESC;
```

7. Start and End Points (Routes) Analysis

Purpose: Analyzes the most common routes by combining start and end points for casual and member riders.

SQL:

```
WITH ranked_routes AS (
  SELECT
    member_casual,
    start_station_name,
    end_station_name,
    COUNT(*) AS total_rides,
    ROW_NUMBER() OVER (PARTITION BY member_casual ORDER BY COUNT(*)
DESC) AS row_num
  FROM
    `cyclistic-431802.CyclistRides.all_trips`
  GROUP BY
    member_casual,
    start_station_name,
    end_station_name
)
SELECT
  member_casual,
  start_station_name,
  end_station_name,
  total_rides
FROM
  ranked_routes
WHERE
  row_num <= 10
ORDER BY
  member_casual,
  total_rides DESC;
```

8. Time of Day Analysis

Purpose: Evaluates the distribution of rides across different times of the day for both casual and member riders.

SQL:

```
SELECT
  EXTRACT(HOUR FROM started_at) AS hour_of_day,
  member_casual,
  COUNT(*) AS total_rides
FROM
  `cyclistic-431802.CyclistRides.all_trips`
GROUP BY
  hour_of_day, member_casual
ORDER BY
  hour_of_day, member_casual;
```

9. Weekends vs. Weekdays Analysis

Purpose: Compares ride frequency between weekends and weekdays for casual and member riders.

SQL:

```
SELECT
  CASE
    WHEN EXTRACT(DAYOFWEEK FROM started_at) IN (1, 7) THEN 'Weekend'
    ELSE 'Weekday'
  END AS day_type,
  member_casual,
  COUNT(*) AS total_rides
FROM
  `cyclistic-431802.CyclistRides.all_trips`
GROUP BY
  day_type, member_casual
ORDER BY
  day_type, member_casual;
```

10. Seasonal Trends Analysis

Purpose: Analyzes ride frequency across different months of the year to identify seasonal trends.

SQL:

```
SELECT
  EXTRACT(MONTH FROM started_at) AS month,
  member_casual,
  COUNT(*) AS total_rides
FROM
  `cyclistic-431802.CyclistRides.all_trips`
GROUP BY
  month, member_casual
ORDER BY
  month, member_casual;
```