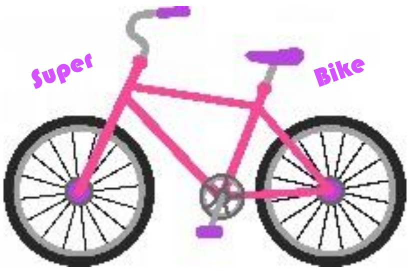
SuperBikes Project

SQL Queries



Company: SuperBikes Inc.

Version: Final Report

Author: Rafael Vazquez

SuperBikes Project - SQL Queries

For this project MySQL Workbench was used for the evaluation, manipulation, and transformation of the data. After the dataset was cleaned, an analysis was performed, saving the information in a file with the proper format for visualization. The visualization phase will be done in Microsoft PowerBI software, which offers a better way of seeing the data analysis, trends, and patterns for a more accurate recommendation based on the results.

Procedure:

- First, we load the original data (csv files) into MySQL. Then, we evaluate those files for missing values, errors, inconsistencies, formatting, duplicates, etc and fix those problems.

- As an example of the procedure in this project with SQL, we will use the month of February.

- The first field must be renamed due to spelling errors and inconsistencies in the name:

```
ALTER TABLE February
RENAME COLUMN ï»¿ride_id TO ride_id;
```

- Check for missing values:

```
SELECT * FROM February
WHERE ride_id IS NULL;
```

- Check for duplicates:

```
SELECT DISTINCT COUNT(ride_id)
FROM February          -- count how many unique records there is in the field ‘ride_id’

SELECT COUNT(ride_id)
FROM February           -- count all the records in the field ‘ride_id'. We compared with the
                       above query
```

- We create a new table ‘feb’ (as the month of February) with the clean data ready for analysis only with the necessary fields; the data comes from our processes of modification and transformation queries to clean the data.

```
--Change data from text format to a datetime format, separate the date from the time in different columns
```

```
    -- Calculate the day and time of the start and end rides, separate in different columns

    -- Calculate the difference between start and end for day and time

    -- Calculate the day of the week for each ride

    CREATE TABLE feb AS
    SELECT *,
            DATE(STR_TO_DATE(started_at, '%m/%d/%Y %T')) AS start_day,
            TIME(STR_TO_DATE(started_at, '%m/%d/%Y %T')) AS start_time,
            DATE(STR_TO_DATE(ended_at, '%m/%d/%Y %T')) AS end_day,
            TIME(STR_TO_DATE(ended_at, '%m/%d/%Y %T')) AS end_time,
            TIMESTAMPDIFF(DAY, DATE(STR_TO_DATE(started_at, '%m/%d/%Y %T')),
            DATE(STR_TO_DATE(ended_at, '%m/%d/%Y %T'))) AS diff_day,
            TIMESTAMPDIFF(MINUTE, TIME(STR_TO_DATE(started_at, '%m/%d/%Y %T')),
            TIME(STR_TO_DATE(ended_at, '%m/%d/%Y %T'))) AS ride_length_min,
            DAYNAME(DATE(STR_TO_DATE(started_at, '%m/%d/%Y %T'))) AS day_of_week
    FROM february;
```

- Drop some columns that we don't need from the table 'feb':

```
    ALTER TABLE feb
    DROP COLUMN started_at,
    DROP COLUMN ended_at,
    DROP COLUMN end_day,
    DROP COLUMN diff_day;
```

- Get values to populate the tables for the month of February:
Tables:

```
    avg_trip_rides:         SELECT AVG(ride_length_min)             -- to calculate the average
                            FROM feb
                            WHERE member_casual = 'member';

                            INSERT INTO avg_trip_rides(id, rider_type, Avg_trip_min, month)
                            VALUES (3, 'casual', 0, 'Feb'), (4, 'member', 9.20, 'Feb');
                            -- populate the table with the results of the average query

    mode_ride_min:          SELECT ride_length_min AS mode_ride_min
                            FROM(SELECT ride_length_min, cnt,
                                    DENSE_RANK() OVER(
                                    ORDER BY cnt DESC
                                    ) as rnk
                                    FROM(SELECT ride_length_min, COUNT(*) as cnt
                                            FROM feb
                                            WHERE member_casual = 'member'
                                            GROUP By ride_length_min
                                            ) x
                                     ) y
                            WHERE rnk = 1
                            -- to calculate the mode for casual and members users in February

    num_ride_day_week:      SELECT COUNT(ride_id)
                            FROM feb
```

```
                    WHERE day_of_week = 'Sunday';   -- to calculate the number of riders per day of
                                                    the week

                    SELECT COUNT(ride_id)
                    FROM feb
                    WHERE member_casual = 'member' AND day_of_week = 'Sunday';

                    INSERT INTO num_ride_day_week(id, rider_type, Sunday, Monday, Tuesday,
                    Wednesday, Thursday, Friday, Saturday, total, month)
                    VALUES (3, 'causal', 0, 0, 0, 0, 0, 0, 0, 0, 'Feb'), (4, 'member', 2, 5, 2, 1, 0, 2, 3, 15,
                    'Feb');

num_rides:          SELECT COUNT(*)                --total of number of rides in February
                    FROM bike_share.feb;

                    SELECT COUNT(ride_id)          --number of casual riders in February
                    FROM feb
                    WHERE member_casual = 'casual';

perc_num_rides:     SELECT (COUNT(ride_id)*100)/15      --calculate the % of rides per type
                    FROM bike_share.feb
                    WHERE member_casual = 'casual';

                    UPDATE perc_num_rides
                    SET Feb = 100
                    WHERE id = 4;            --update the values in the table

                    UPDATE perc_num_rides         --changing the existing value for another.
                    SET id = 1                      we change the existing value of ‘id’ of 3
                    WHERE rider_type = 'casual';    for the value 1.
```