Cyclistic Bike Rides Analysis

Introduction

Here we are going to analyse the Cyclistic Bike data using MS SQL server.

Data Source: https://divvy-tripdata.s3.amazonaws.com/index.html

We are going to use January to July data for our analysis.

MSSQL server

(Note- we have already cleaned our excel csv files)

First, we create a database “Cyclistic_bikerides” and add Excel CSV files

Tables name in database are like Jandata, febdata,and mardata etc.

After adding CSV files now, we start our analysis.

Step1

Adding our different table in one place using Union all and store it in CTE

-Query1-

```
With bikedata AS (
select* from jandata
UNION all
select* from febdata
UNION all
select* from Mardata
UNION all
select* from Aprdata
UNION all
select* from Maydata
UNION all
select* from Jundata
UNION all
select* from Juldata)
```

Step 2

Finding Maximum ride length in minute

-Query2-

```
Select Top 1 *, datediff(minute,started_at, ended_at) as Ride_length_in_min
from bikedata
order by Ride_length_in_min desc
```

(Important Note : Every query that we are going to write after query1 should be run with query1.

Ex. Query1 and query 2 should be run together otherwise SQL give an error because we are going to use bikedata table in query1 as our main table.

So every query will run without error with main table in query 1)

Cyclistic Bike Rides Analysis

Step3

Finding total riders and average ride length

-Query3-

```
Select Count(ride_id) as Total_no_of_rides,
AVG(datediff(minute,started_at, ended_at)) as Average_ride_length
from bikedata
```

Step4

Finding Total trips by customer Type

-Query4

```
SELECT rideable_type,COUNT(ride_id) AS total_riders
FROM bikedata
GROUP BY rideable_type
ORDER BY 2 DESC
```

Step5

Finding average ride length of customers

-Query5-

```
SELECT member_casual,count(ride_id) AS total_riders,
AVG(datediff(minute,started_at, ended_at)) AS Average_Ride_length_in_min
FROM bikedata
GROUP BY member_casual
ORDER BY 2 DESC
```

Step6

Finding Bike demand by hour

-Query6-

```
SELECT member_casual,
CAST(started_at as time) AS time,
COUNT(ride_id) AS total_rides
FROM bikedata
GROUP BY member_casual, CAST(started_at as time)
ORDER BY 2 DESC
```

Step7

Finding total rides by weekdays

-Query7-

Cyclistic Bike Rides Analysis

```
SELECT member_casual, DATEPART(Weekday,started_at) AS weekdays,
COUNT(ride_id) AS total_rides
FROM bikedata
GROUP BY member_casual,DATEPART(Weekday,started_at)
ORDER BY 1
```

Step8

Finding Total rides by month

-Query8-

```
SELECT member_casual, DATEPART(month,started_at) AS Month,
COUNT(ride_id) AS total_rides
FROM bikedata
GROUP BY member_casual, DATEPART(MONTH,started_at)
ORDER BY 1
```

Step9

Finding total rides by bike type

-Query9-

```
SELECT member_casual, rideable_type,COUNT(ride_id) AS total_rides
FROM bikedata
GROUP BY member_casual, rideable_type
ORDER BY 1
```