

SQL Case Study - Bike Sharing Company

Priyanka Zala



Introduction

- You are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago.
- The director of marketing believes the company's future success depends on maximizing the number of annual memberships.
- Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently.
- From these insights, your team will design a new marketing strategy to convert casual riders into annual members.
- But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights.

Business Ask

Analyse the Cyclistic data set for the April'2020 to understand how annual members and casual riders use Cyclistic bikes differently.



Stakeholders



- **Lily Moreno:** The director of marketing. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program.
- **Cyclistic marketing analytics team:** A team of data analysts who are responsible for collecting, analysing, and reporting data that helps guide Cyclistic marketing strategy.
- **Cyclistic executive team:** The executive team will decide whether to approve the recommended marketing programs.

CSV FIDDLE LINK

1. Go to <https://csvfiddle.io/>
2. Upload this data file as csv and give a name table as orders →
https://docs.google.com/spreadsheets/d/1E4KPXX696zcA4nIJRmqt_563LBAMNO0_DNdqo1zM2SM/edit?usp=sharing

Data Dictionary

Column	Data Type	Description
<code>ride_id</code>	String	Unique identifier for each ride
<code>rideable_type</code>	String	Type of rideable (e.g., <code>docked_bike</code>)
<code>started_at</code>	DateTime	an integer value which represents epoch start time
<code>ended_at</code>	DateTime	an integer value which represents epoch end time
<code>start_station_name</code>	String	Name of the starting station
<code>start_station_id</code>	Integer	Unique identifier for the starting station
<code>end_station_name</code>	String	Name of the ending station
<code>end_station_id</code>	Integer	Unique identifier for the ending station
<code>start_lat</code>	Float	Latitude of the starting location
<code>start_lng</code>	Float	Longitude of the starting location
<code>end_lat</code>	Float	Latitude of the ending location
<code>end_lng</code>	Float	Longitude of the ending location
<code>member_casual</code>	String	Type of rider (e.g., member or casual)



Questions for Case Study

1. Write a query to calculate the length of each ride in minutes, the month the ride took place in and a column for the day of the week the ride took place for each ride_id.

Query

```
SELECT
  RIDE_ID,
  EXTRACT(MINUTE FROM (ENDED_AT -
    STARTED_AT)) AS MINUTES,
  EXTRACT(MONTH FROM STARTED_AT) AS
  MONTH,
  DAYOFWEEK(STARTED_AT) AS WEEK
FROM RideSharing
```

	ride_id	minutes	month	WEEK
1	A847FADBBC638E45	26	4	0
2	5405B80E996FF60D	8	4	5
3	5DD24A79A4E006F4	14	4	3
4	2A59BBDF5CDBA725	12	4	2
5	27AD306C119C6158	52	4	6
6	356216E875132F61	5	4	4
7	A2759CB06A81F2BC	5	4	4
8	FC8BC2E2D54F35ED	15	4	2
9	9EC5648678DE06E6	5	4	3
10	A8FFF89140C33017	17	4	6
11	788B1BB8A7491EBD	24	4	6
12	C83C113858BA06DA	4	4	6
13	D2038D92195BDD67	8	4	5
14	C554B4E072B077F8	4	4	6
15	F962D972BC1EF3F0	24	4	1
16	1DDBC1F4D208C2B3	35	4	6

2. Write a query to find the average, minimum and maximum ride lengths for all riders in minutes.

Query

```
SELECT
    ROUND(AVG(EXTRACT(EPOCH FROM
        (ENDED_AT - STARTED_AT)) / 60),2) AS
    AVERAGE_LENGTH,
    ROUND(MIN(EXTRACT(EPOCH FROM
        (ENDED_AT - STARTED_AT)) / 60),2) AS
    MIN_LENGTH,
    ROUND(MAX(EXTRACT(EPOCH FROM
        (ENDED_AT - STARTED_AT)) / 60),2) AS
    MAX_LENGTH
FROM
    RideSharing
```

	AVERAGE_LENGTH	MIN_LENGTH	MAX_LENGTH
1	35.36	-2	58720

3. Did you find any rides where end time was lower than start time? If yes please share the count of those rides.

Query

```
SELECT  
COUNT(*) AS COUNT_OF RIDES  
FROM  
RideSharing  
WHERE  
ENDED_AT < STARTED_AT;
```

	COUNT_OF RIDES
1	51

4. How many rides are longer than 24 hours?

Query

```
SELECT
  COUNT(*) AS COUNT_OF RIDES
FROM
  RideSharing
WHERE
  (ENDED_AT - STARTED_AT) > INTERVAL 24
  HOUR;
```

	COUNT_OF RIDES
1	123

5. Write a query to get the average ride length by type of member_casual.

Query

```
SELECT
  MEMBER_CASUAL,
  ROUND(AVG(EXTRACT(EPOCH FROM
  (ENDED_AT-STARTED_AT)) /60),2) )AS
  AVERAGE_RIDE_LENGTH
FROM
  RideSharing
GROUP BY 1
```

	member_casual	AVERAGE_RIDE_LE...
1	member	20.98
2	casual	72.58

6. Write a query to calculate the average ride length for each day of the week.

Query

```
SELECT  
DAYOFWEEK(STARTED_AT) AS  
DAY_OF_WEEK,  
ROUND(AVG(EXTRACT(EPOCH FROM  
(ENDED_AT-STARTED_AT)) /60),2) AS  
AVERAGE_RIDE_LENGTH  
FROM RideSharing  
GROUP BY 1  
ORDER BY 1
```

	DAY_OF_WEEK	AVERAGE_RIDE_LE...
1	0	37.86
2	1	28.42
3	2	39.12
4	3	25.5
5	4	30.94
6	5	41.09
7	6	40.16

7. Write a query to find Of all the rides taken, how many were by members and how many were by casual riders.

Query

```
SELECT  
member_casual,  
COUNT(*) AS COUNT_OF_RIDE  
FROM RideSharing  
GROUP BY 1
```

	member_casual	COUNT_OF_RIDE
1	member	61148
2	casual	23628

8. Write a query to find How many rides are taken on each day of the week by each group.

Query

```
SELECT  
DAYOFWEEK(started_at) AS DAY_OF_WEEK,  
MEMBER_CASUAL,  
COUNT(*) AS RIDE_COUNT  
FROM RideSharing  
GROUP BY 1,2  
ORDER BY 1
```

	DAY_OF_WEEK	member_casual	RIDE_COUNT
1	0	member	11440
2	0	casual	6475
3	1	member	8064
4	1	casual	2681
5	2	member	9157
6	2	casual	3656
7	3	member	6919
8	3	casual	1799
9	4	member	9264
10	4	casual	2439
11	5	member	7462
12	5	casual	2510
13	6	casual	4068
14	6	member	8842

9. Write a query to find how many rides are taken at each hour of the day, for each day, for members and casual riders.

Query

```
SELECT
    EXTRACT(HOUR FROM started_at) AS HOUR ,
    EXTRACT(DAYOFWEEK FROM started_at) AS
    DAY,
    MEMBER_CASUAL,
    COUNT(*) AS RIDE_COUNT
FROM RIDESHARING
GROUP BY 1,2,3
ORDER BY 1,2,3
```

	hour	day	member_casual	RIDE_COUNT
1	0	0	casual	26
2	0	0	member	43
3	0	1	casual	22
4	0	1	member	27
5	0	2	casual	5
6	0	2	member	30
7	0	3	casual	19
8	0	3	member	32
9	0	4	casual	8
10	0	4	member	18
11	0	5	casual	9
12	0	5	member	23
13	0	6	casual	31
14	0	6	member	36
15	1	0	casual	20
16	1	0	member	29

10 . Write a query to find for each type of rideable_type, how many rides are taken by member and how many are taken by casual.

Query

```
SELECT
  RIDEABLE_TYPE,
  MEMBER_CASUAL,
  COUNT(*) AS RIDE_COUNT
FROM RIDESHARING
GROUP BY 1,2
ORDER BY 1,2
```

	rideable_type	member_casual	RIDE_COUNT
1	docked_bike	casual	23628
2	docked_bike	member	61148

11. What are the top 4 popular locations for members and casual to take rides.

Query

```
SELECT
START_STATION_NAME,
MEMBER_CASUAL,
COUNT(*) AS RIDE_COUNT
FROM RIDESHARING
GROUP BY 1,2
ORDER By 2 DESC
LIMIT 4;
```

	start_station_name	member_casual	RIDE_COUNT
1	Eckhart Park	member	171
2	Drake Ave & Fullerton ...	member	57
3	McClurg Ct & Erie St	member	327
4	California Ave & Divisi...	member	148

Thank you !