

Bixi Project - SQL

Goal

Gain a high level understanding of how people use Bixi bikes, what factors influence the volume of usage, popular stations, and overall business growth.

Data

The data is a cleaned up version of data downloaded from the open data portal at Bixi Montreal:
<https://www.bixi.com/en/open-data>

Usage Volume Overview

The total number of trips for the year of 2016.

- Reviewed description of the columns in the “trips” table: DESC trips
- It seems like “start_date” and “end_date” columns have datetime data type.
- Retrieved total number of trips that started and ended in 2016:

```
SELECT COUNT(*)  
FROM trips  
WHERE start_date >= '2016-01-01 00:00:00' AND end_date < '2017-01-01 00:00:00';
```

There were a total of 3,917,401 trips that started and ended in 2016.

- Checked if there were any trips that started in 2016 and ended in 2017:

```
SELECT COUNT(*)  
FROM trips  
WHERE start_date >= '2016-01-01 00:00:00' AND start_date < '2017-01-01 00:00:00' AND  
end_date >= '2017-01-01 00:00:00';
```

Since there was 0 trips that started in 2016 and ended in 2017, we can use 3,917,401 as the total number of trips for 2016.

The total number of trips for the year of 2017.

- Retrieved total number of trips that started and ended in 2017:

```
SELECT COUNT(*)  
FROM trips  
WHERE start_date >= '2017-01-01 00:00:00' AND end_date < '2018-01-01 00:00:00';
```

There were a total of 4,666,765 trips that started and ended in 2017.

- Checked if there were any trips that started in 2017 and ended in 2018:

```
SELECT COUNT(*)  
FROM trips  
WHERE start_date >= '2017-01-01 00:00:00' AND start_date < '2018-01-01 00:00:00' AND  
end_date >= '2018-01-01 00:00:00';
```

Since there was 0 trips that started in 2017 and ended in 2018, we can use 4,666,765 as the total number of trips for 2017.

The total number of trips for the year of 2016 broken-down by month.

```
SELECT MONTHNAME(start_date) AS Month, COUNT(*) AS Trips_per_Month
FROM trips
WHERE start_date >= '2016-01-01 00:00:00' AND end_date < '2017-01-01 00:00:00'
GROUP BY Month;
```

Output:

April	189923
August	672778
July	699248
June	631503
May	561077
November	150129
October	392480
September	620263

The total number of trips for the year of 2017 broken-down by month.

```
SELECT MONTHNAME(start_date) AS Month, COUNT(*) AS Trips_per_Month
FROM trips
WHERE start_date >= '2017-01-01 00:00:00' AND end_date < '2018-01-01 00:00:00'
GROUP BY Month;
```

Output:

April	195662
August	839938
July	860732
June	741835
May	587447
November	149794
October	559506
September	731851

The average number of trips a day for each year-month combination in the dataset.

- Checked what year-month combinations are there in the dataset:

```
SELECT YEAR(start_date), MONTHNAME(start_date)
FROM trips
GROUP BY YEAR(start_date), MONTHNAME(start_date);
```

Output:

2016	April
2016	August
2016	July
2016	June
2016	May
2016	November

2016	October
2016	September
2017	April
2017	August
2017	July
2017	June
2017	May
2017	November
2017	October
2017	September

April 2016 – November 2016 and April – November 2017.

- Retrieved number of trips per day for April 2016:

```
SELECT DAY(start_date), COUNT(*) AS Num_Trips_per_Day
FROM trips
WHERE start_date >= '2016-04-01 00:00:00' AND start_date < '2016-05-01 00:00:00'
GROUP BY DAY(start_date);
```

Output:

15	9282
16	10661
17	13236
18	5793
19	12261
20	13114
21	15477
22	9365
23	12835
24	10209
25	12058
26	12253
27	11818
28	12700
29	14218
30	14643

- Retrieved average number of trips per day for April 2016:

```
SELECT AVG(Num_Trips_per_Day)
FROM
(
SELECT DAY(start_date), COUNT(*) AS Num_Trips_per_Day
FROM trips
WHERE start_date >= '2016-04-01 00:00:00' AND start_date < '2016-05-01 00:00:00'
GROUP BY DAY(start_date)
)
AS Avg_Trips_per_Day;
```

Output: 11870.1875

- Re-ran query to get average number of trips per day for remaining months. Modified start_date range in the WHERE statement every time).

April 2016: 11870.1875
 May 2016: 18099.2581
 June 2016: 21050.1000
 July 2016: 22556.3871
 August 2016: 21702.5161
 September 2016: 20675.4333
 October 2016: 12660.6452
 November 2016: 10008.6000
 April 2017: 12228.8750
 May 2017: 18949.9032
 June 2017: 24727.8333
 July 2017: 27765.5484
 August 2017: 27094.7742
 September 2017: 24395.0333
 October 2017: 18048.5806
 November 2017: 9986.2667

The total number of trips in the year 2017 broken-down by membership status (member/non-member).

```

SELECT is_member, COUNT(*)
FROM trips
WHERE start_date >= '2017-01-01 00:00:00' AND end_date < '2018-01-01 00:00:00'
GROUP BY is_member;
  
```

Output:

0	882083
1	3784682

non-member: 882,083 trips
 member: 3,784,682

The fraction of total trips that were done by members for the year of 2017 broken-down by month.

```

SELECT YEAR(start_date), MONTHNAME(start_date),
SUM(is_member)/COUNT(is_member) AS Fraction
FROM trips
WHERE start_date >= '2017-01-01 00:00:00' AND end_date < '2018-01-01 00:00:00'
GROUP BY YEAR(start_date), MONTHNAME(start_date);
  
```

Output:

2017	April	0.8352
2017	August	0.7811
2017	July	0.7643
2017	June	0.8081
2017	May	0.8197

2017	November	0.9246
2017	October	0.8641
2017	September	0.8258

Conclusion:

- The demand for Bixi bikes is at its peak in summer months, especially in July.
- If Bixi were to offer non-members a special promotion in an attempt to convert them to members, it would probably make sense to do it in months when the demand is high and a high percentage of trips are done by non-members. It seems like July would be the best month to offer a special promotion to non-members as July has the highest demand and the lowest % of trips done by members. June and August would be good picks as well.

Trip Characteristics

Average trip time across the entire dataset.

```
SELECT AVG(duration_sec)
FROM trips;
```

Output: 824.4291

Membership status.

```
SELECT is_member, AVG(duration_sec)
FROM trips
GROUP BY is_member;
```

Output:

0	1221.2917
1	731.7721

Month.

```
SELECT MONTHNAME(start_date), AVG(duration_sec)
FROM trips
GROUP BY MONTHNAME(start_date);
```

Output:

April	801.7489
August	855.7830
July	879.7803
June	844.4313
May	839.2297
November	654.0278
October	730.4162
September	803.0599

Day of the week.

```
SELECT DAYNAME(start_date), AVG(duration_sec)
FROM trips
```

```
GROUP BY DAYNAME(start_date);
```

Output:

Friday	798.8752
Monday	798.6486
Saturday	908.9840
Sunday	914.1739
Thursday	790.7546
Tuesday	794.6180
Wednesday	792.4604

Station name.

A) Which station has the longest trips on average?

- Will be look at stations where trips started, not ended to identify stations for trips.

```
SELECT s.name, AVG(duration_sec)
FROM trips AS t JOIN stations AS s ON t.start_station_code = s.code
GROUP BY s.name
ORDER BY AVG(duration_sec) DESC
LIMIT 1;
```

Output:

Métro Jean-Drapeau	1899.1624
--------------------	-----------

B) Which station has the shortest trips on average?

- Will be looking at stations where trips started, not ended to identify stations for trips.

```
SELECT s.name, AVG(duration_sec)
FROM trips AS t JOIN stations AS s ON t.start_station_code = s.code
GROUP BY s.name
ORDER BY AVG(duration_sec) ASC
LIMIT 1;
```

Output:

Métro Georges-Vanier (St-Antoine / Canning)	498.5515
---------------------------------------------	----------

C) I will exclude trips with durations above (using 50 seconds) and below certain thresholds (using 3,000 seconds) to avoid extremely long / short trips skewing my results.

- Station with the highest average trips duration:

```
SELECT s.name, AVG(duration_sec)
FROM trips AS t JOIN stations AS s ON t.start_station_code = s.code
WHERE t.duration_sec > 50 AND t.duration_sec < 3000
GROUP BY s.name
ORDER BY AVG(duration_sec) DESC
LIMIT 1;
```

Output:

Métro Jean-Drapeau	1542.0199
--------------------	-----------

Same station as before excluding extreme values, but the average trip duration has changed.

- Station with the lowest average trips duration:

```
SELECT s.name, AVG(duration_sec)
FROM trips AS t JOIN stations AS s ON t.start_station_code = s.code
WHERE t.duration_sec > 50 AND t.duration_sec < 3000
GROUP BY s.name
ORDER BY AVG(duration_sec) ASC
LIMIT 1;
```

Output:

Métro Georges-Vanier (St-Antoine / Canning) 469.6345

Same station as before excluding extreme values, but the average trip duration has changed.

Fraction of round trips by membership status.

A) Fraction of round trips that were done by non-members:

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE start_station_code = end_station_code AND is_member = 0
)/COUNT(*)
FROM trips
WHERE is_member = 0;
```

Output: 0.0488

B) Fraction of round trips that were done by members:

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE start_station_code = end_station_code AND is_member = 1
)/COUNT(*)
FROM trips
WHERE is_member = 1;
```

Output: 0.0142

Day of the week.

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Monday' AND start_station_code = end_station_code) /
COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Monday';
```

Output: 0.0194

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Tuesday' AND start_station_code = end_station_code) /
COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Tuesday';
```

Output: 0.0165

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Wednesday' AND start_station_code = end_station_code) /
COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Wednesday';
```

Output: 0.0160

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Thursday' AND start_station_code = end_station_code) /
COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Thursday';
```

Output: 0.0161

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Friday' AND start_station_code = end_station_code) /
COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Friday';
```

Output: 0.0181

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Saturday' AND start_station_code = end_station_code) /
COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Saturday';
```

Output: 0.0287

```
SELECT (SELECT COUNT(*)
FROM trips
WHERE DAYNAME(start_date) = 'Sunday' AND start_station_code = end_station_code) /
COUNT(*)
FROM trips
```

```
WHERE DAYNAME(start_date) = 'Sunday';
```

Output: 0.0343

Discussion on observed differences.

- A) **Average trip duration by membership status.** Average trip duration is significantly higher for non-members (~ 1,221 seconds for non-members vs. ~ 732 seconds for members). This could be explained by a significant portion of non-members using bikes for recreational purposes.
- B) **Average trip duration in summer months vs. the rest of the year.** It seems like average duration of trips is higher in summer months. Clients could be more likely to use a rented bike to get round as opposed to public transit in warmer months due to more favourable weather conditions.
- C) **Average trip duration by day of the week.** Higher trip duration on weekends.
- D) **Station with highest average trip duration:** Métro Jean-Drapeau (~ 1,899 seconds).
- E) **Station with lowest average trip duration:** Métro Georges-Vanier (~ 499 seconds).
- F) **Round trips.**
 - **Members vs. non-members.** Fraction of round trips for non-members (0.0488 or ~ 4.9%) is significantly higher than the fraction of round trips for members (0.0142 or ~ 1.4%). One possible explanation is that higher proportion of non-members are using rented bikes for casual rides around the city/recreational purposes (sightseeing, etc.) and then return to the starting point as opposed to members who are more likely to be using rented bikes to get from point A to point B (e.g. from home to work) and return back to point A later the same day (this would be counted as a separate trip). People who don't reside in Montreal permanently are more likely to use bikes for recreational purposes and are less likely to be members (especially if the company operates stations in Montreal only).
 - **Weekends vs. weekdays.** Fraction of round trips is much higher for Saturdays and Sundays compared to other days of the week. This can be explained by higher number of clients (who may or may not reside in Montreal permanently) using bikes for recreational purposes on weekends as opposed to weekdays. For example, clients who use rented bikes to get to work and back would rent a bike in the morning, get to work, return it to the station and then do another trip to return home in the evening.

Popular Stations

Five most popular starting stations.

```
SELECT s.name, COUNT(t.start_station_code)
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code
GROUP BY s.name
ORDER BY COUNT(t.start_station_code) DESC
LIMIT 5;
```

Output:

Mackay / de Maisonneuve	97150
Métro Mont-Royal (Rivard / du Mont-Royal)	81279
Métro Place-des-Arts (de Maisonneuve / de Bleury)	78848
Métro Laurier (Rivard / Laurier)	76813
Métro Peel (de Maisonneuve / Stanley)	72298

Five most popular ending stations.

```
SELECT s.name, COUNT(t.end_station_code)
FROM stations AS s JOIN trips AS t ON s.code = t.end_station_code
GROUP BY s.name
ORDER BY COUNT(t.end_station_code) DESC
LIMIT 5;
```

Output:

Berri / de Maisonneuve	103720
Mackay / de Maisonneuve	99128
Métro Place-des-Arts (de Maisonneuve / de Bleury)	95343
Métro St-Laurent (de Maisonneuve / St-Laurent)	86886
Métro Peel (de Maisonneuve / Stanley)	76551

How is the number of starts and ends distributed for the station Mackay / de Maisonneuve throughout the day?

- Created view for all trips that started at this station.

```
CREATE VIEW time_of_day_start AS
SELECT s.name,
CASE
    WHEN HOUR(start_date) BETWEEN 7 AND 11 THEN "morning"
    WHEN HOUR(start_date) BETWEEN 12 AND 16 THEN "afternoon"
    WHEN HOUR(start_date) BETWEEN 17 AND 21 THEN "evening"
    ELSE "night"
END AS "time_of_day"
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code
WHERE s.name = "Mackay / de Maisonneuve";
```

- Created another view for all trips that ended at this station.

```
CREATE VIEW time_of_day_end AS
SELECT s.name,
CASE
    WHEN HOUR(end_date) BETWEEN 7 AND 11 THEN "morning"
    WHEN HOUR(end_date) BETWEEN 12 AND 16 THEN "afternoon"
    WHEN HOUR(end_date) BETWEEN 17 AND 21 THEN "evening"
    ELSE "night"
END AS "time_of_day"
FROM stations AS s JOIN trips AS t ON s.code = t.end_station_code
WHERE s.name = "Mackay / de Maisonneuve";
```

- Calculated the number of trips that started at Mackay / de Maisonneuve by time of the day.

```
SELECT time_of_day, COUNT(*)
FROM time_of_day_start
GROUP BY time_of_day;
```

Output:

afternoon	30718 (~ 31.6% of total)
evening	36781 (~ 37.9% of total)
morning	17384 (~ 17.9% of total)
night	12267 (~ 12.6% of total)

- Calculated the number of trips that ended at Mackay / de Maisonneuve by time of the day.

```
SELECT time_of_day, COUNT(*)  
FROM time_of_day_end  
GROUP BY time_of_day;
```

Output:

afternoon	30429 (~ 30.7% of total)
evening	31983 (~ 32.3% of total)
morning	26390 (~ 26.6% of total)
night	10326 (~ 10.4% of total)

Discussion on observed differences.

Percentage of daily tips that were done in the morning and ended at the Mackay / de Maisonneuve station (26.6%) is significantly higher than the percentage of trips that were done in the morning and started at this station (17.9%). Also, the percentage of evening trips that started at this station (37.9%) is higher than the percentage of evening trips that ended at this station (32.3%). It seems like significant portion of clients take bikes from other stations to get to Mackay / de Maisonneuve station in the morning and then take them from this station to return back to other stations, including stations that they took them from in the morning. One possible explanation is that the station is located in the area with high number of office buildings (could be downtown).

Station that has proportionally the least number of member trips.

```
SELECT s.name, SUM(is_member)/COUNT(is_member) AS Fraction  
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code  
GROUP BY s.name  
HAVING COUNT(start_station_code) >= 10 AND COUNT(end_station_code) >= 10  
ORDER BY Fraction  
LIMIT 1;
```

Output:

Métro Jean-Drapeau 0.2391

- Métro Jean-Drapeau station has the lowest fraction of member trips that started at this station.

```
SELECT s.name, SUM(is_member)/COUNT(is_member) AS Fraction  
FROM stations AS s JOIN trips AS t ON s.code = t.end_station_code  
GROUP BY s.name
```

```
HAVING COUNT(start_station_code) >= 10 AND COUNT(end_station_code) >= 10
ORDER BY Fraction
LIMIT 1;
```

Output:

Métro Jean-Drapeau 0.2303

- Métro Jean-Drapeau station has the lowest fraction of member trips that ended at this station as well.

Station that has proportionally the most number of member trips.

```
SELECT s.name, SUM(is_member)/COUNT(is_member) AS Fraction
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code
GROUP BY s.name
HAVING COUNT(start_station_code) >= 10 AND COUNT(end_station_code) >= 10
ORDER BY Fraction DESC
LIMIT 1;
```

Output:

St-Charles / Sauvé 0.9238

- St-Charles / Sauvé station has the highest fraction of member trips that started at this station.

```
SELECT s.name, SUM(is_member)/COUNT(is_member) AS Fraction
FROM stations AS s JOIN trips AS t ON s.code = t.end_station_code
GROUP BY s.name
HAVING COUNT(start_station_code) >= 10 AND COUNT(end_station_code) >= 10
ORDER BY Fraction DESC
LIMIT 1;
```

Output:

du Mont-Royal / Augustin-Frigon 0.9205

- du Mont-Royal / Augustin-Frigon station has the highest fraction of member trips that ended at this station.

Query that counts the number of starting trips per station.

```
SELECT s.name, COUNT(start_station_code)
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code
GROUP BY s.name;
```

Output: too many rows to paste into this document.

Query that counts the number of round trips for each station.

- Counting number of return trips and grouping by starting stations.

```
SELECT s.name, COUNT(*)
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code
WHERE start_station_code = end_station_code
GROUP BY s.name;
```

Output: too many rows to paste into this document.

Combining the above queries and calculating the fraction of round trips to the total number of starting trips for each station.

```
SELECT s.name, COUNT(CASE WHEN start_station_code = end_station_code THEN 1 END) /
COUNT(start_station_code)
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code
GROUP BY s.name;
```

Output: too many rows to paste into this document.

Filtering down to stations with at least 50 trips originating from them.

```
SELECT s.name, COUNT(CASE WHEN start_station_code = end_station_code THEN 1
END)/COUNT(start_station_code)
FROM stations AS s JOIN trips AS t ON s.code = t.start_station_code
GROUP BY s.name
HAVING COUNT(start_station_code) >= 50;
```

Output: too many rows to paste into this document.

Location of stations with a high fraction of round trips.

I would expect to find stations with a high fraction of return trips in areas with high concentration of tourists and areas where people permanently residing in Montreal spend time on weekends, such as historical district(s), areas with parks, etc.