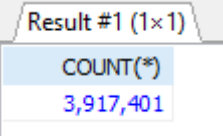
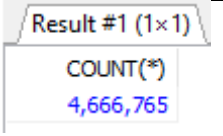


1. Usage Volume Overview

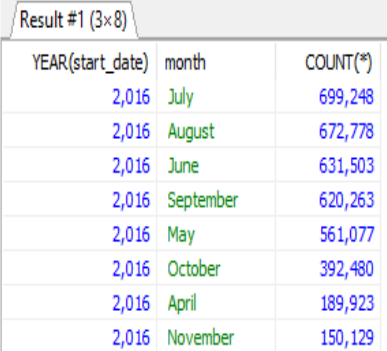
1.1 The total number of trips for the years of 2016.

Result	Explanation	Query
	The function 'Year' was used to pull out all trips from 2016 and a count was taken using COUNT function for all the trips.	<pre>SELECT COUNT(*) FROM trips WHERE YEAR(start_date)=2016 AND YEAR(end_date)=2016</pre>

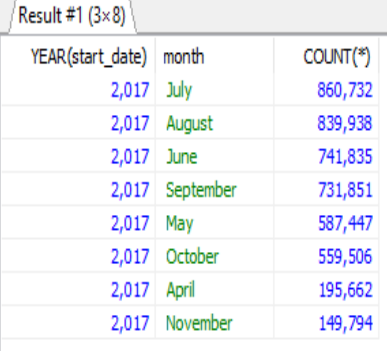
1.2 The total number of trips for the years of 2017.

Result	Explanation	Query
	The function 'Year' was used to pull out all trips from 2017 and a count was taken using 'COUNT' function for all the trips.	<pre>SELECT COUNT(*) FROM trips WHERE YEAR(start_date)=2017 AND YEAR(end_date)=2017</pre>

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

Result	Explanation	Query
	Along with the previous query, the function 'MONTHNAME' was used to extract month from each trip. It was then grouped together by month and a count was taken to get the trips split by month from 2016.	<pre>SELECT YEAR(start_date), MONTHNAME(start_date) AS month, COUNT(*) FROM trips WHERE YEAR(start_date)=2016 AND YEAR(end_date)=2016 GROUP BY month ORDER BY COUNT(*) DESC</pre>

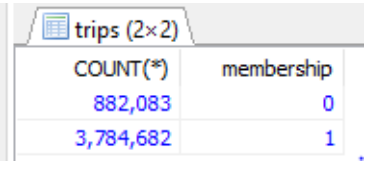
1.4 The total number of trips for the years of 2017 broken-down by month.

Result	Explanation	Query
	Along with the previous query, the function 'MONTHNAME' was used to extract month from each trip. It was then grouped together by month and a count was taken to get the trips split by month from 2017.	<pre>SELECT YEAR(start_date), MONTHNAME(start_date) AS month, COUNT(*) FROM trips WHERE YEAR(start_date)=2017 AND YEAR(end_date)=2017 GROUP BY month ORDER BY COUNT(*) DESC</pre>

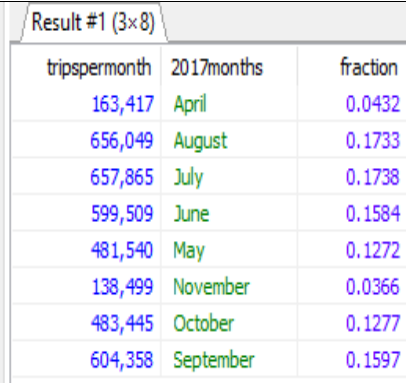
1.5 The average number of trips a day for each year-month combination in the dataset.

Result	Explanation	Query																																				
<div><div>2016_cal (4x8)</div><table><tr><th>Y</th><th>2016months</th><th>numofdays</th><th>avgtripsperday</th></tr><tr><td>2,016</td><td>April</td><td>30</td><td>6,330.7667</td></tr><tr><td>2,016</td><td>May</td><td>31</td><td>18,099.2581</td></tr><tr><td>2,016</td><td>June</td><td>30</td><td>21,050.1000</td></tr><tr><td>2,016</td><td>July</td><td>31</td><td>22,556.3871</td></tr><tr><td>2,016</td><td>August</td><td>31</td><td>21,702.5161</td></tr><tr><td>2,016</td><td>September</td><td>30</td><td>20,675.4333</td></tr><tr><td>2,016</td><td>October</td><td>31</td><td>12,660.6452</td></tr><tr><td>2,016</td><td>November</td><td>30</td><td>5,004.3000</td></tr></table></div>	Y	2016months	numofdays	avgtripsperday	2,016	April	30	6,330.7667	2,016	May	31	18,099.2581	2,016	June	30	21,050.1000	2,016	July	31	22,556.3871	2,016	August	31	21,702.5161	2,016	September	30	20,675.4333	2,016	October	31	12,660.6452	2,016	November	30	5,004.3000	<p>1. 2 tables were created with months in 2016 & months in 2017 months along with the number of days.</p> <p>2. The tables created above were merged with tables which broke down average trips per month using months as the key.</p> <p>Note: Average/day was obtained by dividing average per month by number of days in a month.</p>	<pre>#CREATE TABLE 2016_cal (2016months CHAR(20),numofdays INT) #CREATE TABLE 2017_cal (2017months CHAR(20),numofdays INT) #INSERT INTO 2016_cal (2016months,numofdays) VALUES ('January',31),('February',29),('March',31),('April',30),('May',31),('June',30),('July',31),('August',31),('September',30),('October',31),('November',30),('December',31) #INSERT INTO 2017_cal (2017months,numofdays) VALUES ('January',31),('February',28),('March',31),('April',30),('May',31),('June',30),('July',31),('August',31),('September',30),('October',31),('November',30),('December',31) SELECT Y,partone.2016months,numofdays,trip spermonth/numofdays AS avgtripsperday FROM (SELECT YEAR(start_date) AS Y,MONTHNAME(start_date) AS 2016months,COUNT(*) AS tripspermonth FROM trips WHERE YEAR (start_date)=2016 AND YEAR(end_date)=2016 GROUP BY 2016months) AS partone JOIN 2016_cal ON partone.2016months=2016_cal.2016mon ths SELECT Y,partone.2017months,numofdays,trip spermonth/numofdays AS avgperday FROM (SELECT YEAR(start_date) AS Y,MONTHNAME(start_date) AS 2017months,COUNT(*) AS tripspermonth FROM trips WHERE YEAR (start_date)=2017 AND YEAR(end_date)=2017 GROUP BY 2017months) AS partone JOIN 2017_cal ON partone.2017months=2017_cal.2017mon ths</pre>
Y	2016months	numofdays	avgtripsperday																																			
2,016	April	30	6,330.7667																																			
2,016	May	31	18,099.2581																																			
2,016	June	30	21,050.1000																																			
2,016	July	31	22,556.3871																																			
2,016	August	31	21,702.5161																																			
2,016	September	30	20,675.4333																																			
2,016	October	31	12,660.6452																																			
2,016	November	30	5,004.3000																																			

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

Result	Explanation	Query
	The function 'Year' was used to pull out all trips from 2017 and a COUNT (*) was taken after grouping together using membership status	<pre>SELECT COUNT(*), is_member AS membership FROM trips WHERE YEAR(start_date)=2017 AND YEAR(end_date)=2017 GROUP BY membership</pre>

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

Result	Explanation	Query
	<ol style="list-style-type: none"> 1. The function 'MONTHNAME' was used to extract month from each trip. 2. Filter applied here is 'is_member=1' and 'year=2017'. 3. Grouped together by month and a count was taken to get the total trips per month by members in 2017. 4. Fraction=trips per month/3784682(from previous query) <p>Assumption: The total trips mentioned here are the total trips in 2017 by the members.</p>	<pre>SELECT COUNT(*) AS tripspermonth, MONTHNAME(start_date) AS 2017months, COUNT(*)/3784682 AS fraction FROM trips WHERE (YEAR(start_date)=2017 AND YEAR(end_date)=2017) AND is_member=1 GROUP BY MONTHNAME(start_date)</pre>

3.1 The demand for Bixi bikes is at it's peak during summer time especially in August and July.

3.2 The demand for Bixi bikes is at it's least during April and November. Hence, a special promotion during these months can help convert non-member


to members.

2. Trip Characteristics

1. Calculate the average trip time across the entire dataset.

Result	Explanation	Query		
<div><div>Result #1 (1x1)</div><table><thead><tr><th>averagetime</th></tr></thead><tbody><tr><td>824.4291</td></tr></tbody></table></div>	averagetime	824.4291	The function AVG was used to get average trip time	<pre>SELECT AVG(duration_sec) AS averagetime FROM trips</pre>
averagetime				
824.4291				

2.1 Calculate the average trip time broken-down by: Membership status

Result	Explanation	Query						
<div> trips (2x2)</div> <table><thead><tr><th>membership</th><th>averagetime</th></tr></thead><tbody><tr><td>0</td><td>1,221.2917</td></tr><tr><td>1</td><td>731.7721</td></tr></tbody></table>	membership	averagetime	0	1,221.2917	1	731.7721	The function AVG was used to get average trip time after grouping using membership status	<pre>SELECT is_member AS membership, AVG(duration_sec) AS averagetime FROM trips GROUP BY membership</pre>
membership	averagetime							
0	1,221.2917							
1	731.7721							

2.2 Calculate the average trip time broken-down by: Month

Result	Explanation	Query																		
<div>Result #1 (2x8)</div> <table><tr><th>month</th><th>averagetime</th></tr><tr><td>April</td><td>801.7489</td></tr><tr><td>August</td><td>855.7830</td></tr><tr><td>July</td><td>879.7803</td></tr><tr><td>June</td><td>844.4313</td></tr><tr><td>May</td><td>839.2297</td></tr><tr><td>November</td><td>654.0278</td></tr><tr><td>October</td><td>730.4162</td></tr><tr><td>September</td><td>803.0599</td></tr></table>	month	averagetime	April	801.7489	August	855.7830	July	879.7803	June	844.4313	May	839.2297	November	654.0278	October	730.4162	September	803.0599	<p>1. The function MONTHNAME was used to extract month and then it was grouped together by month.</p> <p>2. Then AVG function was used to break down average trip time by month.</p>	<pre>SELECT MONTHNAME(start_date) AS month, AVG(duration_sec) AS averagetime FROM trips GROUP BY month</pre>
month	averagetime																			
April	801.7489																			
August	855.7830																			
July	879.7803																			
June	844.4313																			
May	839.2297																			
November	654.0278																			
October	730.4162																			
September	803.0599																			

2.3 Calculate the average trip time broken-down by: Day of the week

Result	Explanation	Query
--------	-------------	-------

Result #1 (3×7)			<p>1. The 'WEEKDAY' function was used to extract the day of the week. The function returns values from 0-6 which corresponds to Monday to Sunday respectively.</p> <p>2. Utilised the 'Case – When' clause to display numbers in week day.</p>	<pre>SELECT WEEKDAY(start_date) AS d, AVG(duration_sec) AS averagetime, (CASE WHEN WEEKDAY(start_date)=0 THEN 'Monday' WHEN WEEKDAY(start_date)=1 THEN 'Tuesday' WHEN WEEKDAY(start_date)=2 THEN 'Wednesday' WHEN WEEKDAY(start_date)=3 THEN 'Thursday' WHEN WEEKDAY(start_date)=4 THEN 'Friday' WHEN WEEKDAY(start_date)=5 THEN 'Saturday' WHEN WEEKDAY(start_date)=6 THEN 'Sunday' ELSE 'NODAY' END) AS wkd FROM trips GROUP BY d</pre>
d	averagetime	wkd		
0	798.6486	Monday		
1	794.6180	Tuesday		
2	792.4604	Wednesday		
3	790.7546	Thursday		
4	798.8752	Friday		
5	908.9840	Saturday		
6	914.1739	Sunday		

2.4 Calculate the average trip time broken-down by: station name

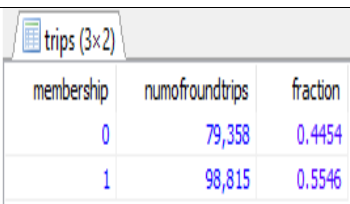
Result	Explanation	Query																
<div><div>stations (2×540)</div><table><tr><th>name</th><th>averagetime</th></tr><tr><td>Méto Jean-Drapeau</td><td>1,899.1624</td></tr><tr><td>LaSalle / 4e avenue</td><td>1,834.6195</td></tr><tr><td>Basile-Routhier / G...</td><td>1,750.3876</td></tr></table></div> <div><div>stations (2×540)</div><table><tr><th>name</th><th>averagetime ▲</th></tr><tr><td>Méto Georges-Vanier (St-Antoine / Canning)</td><td>498.5515</td></tr><tr><td>Méto Lucien-L'Allier (Lucien l'Allier / Argyle)</td><td>544.8251</td></tr><tr><td>Pontiac / Gilford</td><td>578.8532</td></tr></table></div>	name	averagetime	Méto Jean-Drapeau	1,899.1624	LaSalle / 4e avenue	1,834.6195	Basile-Routhier / G...	1,750.3876	name	averagetime ▲	Méto Georges-Vanier (St-Antoine / Canning)	498.5515	Méto Lucien-L'Allier (Lucien l'Allier / Argyle)	544.8251	Pontiac / Gilford	578.8532	Here 2 sets of queries were carried out by grouping start station codes and end station codes. Both resulted in same answer:	<pre>SELECT stations.name, a.averagetime FROM (SELECT start_station_code AS code, AVG(duration_sec) AS averagetime FROM trips GROUP BY code) AS a JOIN stations ON a.code=stations.code ORDER BY averagetime DESC</pre> <hr/> <pre>SELECT stations.name, a.averagetime FROM (SELECT end_station_code AS code, AVG(duration_sec) AS averagetime FROM trips GROUP BY code) AS a JOIN stations ON a.code=stations.code ORDER BY averagetime DESC</pre>
name	averagetime																	
Méto Jean-Drapeau	1,899.1624																	
LaSalle / 4e avenue	1,834.6195																	
Basile-Routhier / G...	1,750.3876																	
name	averagetime ▲																	
Méto Georges-Vanier (St-Antoine / Canning)	498.5515																	
Méto Lucien-L'Allier (Lucien l'Allier / Argyle)	544.8251																	
Pontiac / Gilford	578.8532																	
2.4.1. Station with longest average trip= Metro Jean -Drapeau 2. 4.1. Station with least average trip=Metro Georges-Vanier																		

2.4.3 Skewing of results can be avoided by filtering out trips which are extremely short. Short trips can be due to a variety of reasons

ranging from discomfort in riding bikes to people not enjoying the bike rides. These data would not be valid for calculating mean of the trip duration. Extremely short trips would shift the mean of average duration from right to left. This would not result in accurate representation of averages. This can be achieved by considering trip durations above 5minutes or even 10 minutes.

Eg. WHERE duration_sec >600

3.1 Let's call trips that start and end in the same station "round trips". Calculate the fraction of trips that were round trips and break it down by: Membership status

Result	Explanation	Query									
 <table border="1"> <thead> <tr> <th>membership</th><th>numofroundtrips</th><th>fraction</th></tr> </thead> <tbody> <tr> <td>0</td><td>79,358</td><td>0.4454</td></tr> <tr> <td>1</td><td>98,815</td><td>0.5546</td></tr> </tbody> </table>	membership	numofroundtrips	fraction	0	79,358	0.4454	1	98,815	0.5546	<p>1. Round trips were identified as trips where 'start station code' is the same as 'end station code'.</p> <p>2. An initial query was used to take the total count of round trips.</p> <p>3. Grouping was done using membership status and a count was taken to split round trips based on membership.</p> <p>4. Round trip count based on membership was then divided by total round trips to get fraction of round trips based on membership status.</p> <p>Assumption: the total trips here are the total round trips by members and non-members which was found to be 178173.</p>	<pre>SELECT COUNT(*) FROM trips WHERE start_station_code=end_station_code #178173</pre> <hr/> <pre>SELECT is_member AS membership, COUNT(*) AS numofroundtrips, COUNT(*)/178173 AS fraction FROM trips WHERE start_station_code=end_station_code GROUP BY membership</pre>
membership	numofroundtrips	fraction									
0	79,358	0.4454									
1	98,815	0.5546									

3.2 Let's call trips that start and end in the same station "round trips".

Calculate the fraction of trips that were round trips and break it down by: Day of week

Result	Explanation	Query																																
<table><tr><th>wd</th><th>numofroundtrips</th><th>fraction</th><th>wkd</th></tr><tr><td>0</td><td>22,098</td><td>0.1240</td><td>Monday</td></tr><tr><td>1</td><td>21,785</td><td>0.1223</td><td>Tuesday</td></tr><tr><td>2</td><td>21,930</td><td>0.1231</td><td>Wednesday</td></tr><tr><td>3</td><td>21,354</td><td>0.1198</td><td>Thursday</td></tr><tr><td>4</td><td>22,778</td><td>0.1278</td><td>Friday</td></tr><tr><td>5</td><td>32,547</td><td>0.1827</td><td>Saturday</td></tr><tr><td>6</td><td>35,681</td><td>0.2003</td><td>Sunday</td></tr></table>	wd	numofroundtrips	fraction	wkd	0	22,098	0.1240	Monday	1	21,785	0.1223	Tuesday	2	21,930	0.1231	Wednesday	3	21,354	0.1198	Thursday	4	22,778	0.1278	Friday	5	32,547	0.1827	Saturday	6	35,681	0.2003	Sunday	<p>1. The 'WEEKDAY' function was used to extract the day of the week. The function returns values from 0-6 which corresponds to Monday to Sunday respectively.</p> <p>2. Grouping was done by weekdays and a count was taken.</p> <p>3. Fraction= number of round trips on each day /178173(total number of round trips by members and non –members)</p> <p>Note:The 'Case' clause can be used to display the day of the week like question 2.3</p>	<pre>SELECT WEEKDAY(start_date) as wd, COUNT(*) AS numofroundtrips, COUNT(*)/178173 AS fraction, (CASE WHEN WEEKDAY(start_date)=0 THEN 'Monday' WHEN WEEKDAY(start_date)=1 THEN 'Tuesday' WHEN WEEKDAY(start_date)=2 THEN 'Wednesday' WHEN WEEKDAY(start_date)=3 THEN 'Thursday' WHEN WEEKDAY(start_date)=4 THEN 'Friday' WHEN WEEKDAY(start_date)=5 THEN 'Saturday' WHEN WEEKDAY(start_date)=6 THEN 'Sunday' ELSE 'NODAY' END) AS wkd FROM trips WHERE start_station_code=end_station_code GROUP BY wd</pre>
wd	numofroundtrips	fraction	wkd																															
0	22,098	0.1240	Monday																															
1	21,785	0.1223	Tuesday																															
2	21,930	0.1231	Wednesday																															
3	21,354	0.1198	Thursday																															
4	22,778	0.1278	Friday																															
5	32,547	0.1827	Saturday																															
6	35,681	0.2003	Sunday																															

4. Clearly, 55% of the total round trips were taken by members while Non-members constitute to 45% of the total round trips. Majority of the round trips are taken during the weekends which correlates with the fact that weekends are generally non-working days.

3. Popular Stations

1. What are the names of the 5 most popular starting stations?

Result	Explanation	Query
--------	-------------	-------

stations (2x5)		A count of number of trips was taken by grouping together using start station code. It was then arranged in descending order and limited to 1 st 5 records.	<pre>SELECT name, a.numoftrips FROM (SELECT start_station_code AS code, COUNT(*) AS numoftrips FROM trips GROUP BY code) AS a JOIN stations ON a.code=stations.code ORDER BY a.numoftrips DESC LIMIT 5</pre>
name	numoftrips		
Mackay / de Maisonneuve	97,150		
Métro Mont-Royal (Rivard / du Mont-Royal)	81,279		
Métro Place-des-Arts (de Maisonneuve / de Bleury)	78,848		
Métro Laurier (Rivard / Laurier)	76,813		
Métro Peel (de Maisonneuve / Stanley)	72,298		

2. What are the names of the 5 most popular ending stations?

Result	Explanation	Query												
<table><tr><th>name</th><th>numoftrips</th></tr><tr><td>Berri / de Maisonneuve</td><td>103,720</td></tr><tr><td>Mackay / de Maisonneuve</td><td>99,128</td></tr><tr><td>Métro Place-des-Arts (de Maisonneuve / de Bleury)</td><td>95,343</td></tr><tr><td>Métro St-Laurent (de Maisonneuve / St-Laurent)</td><td>86,886</td></tr><tr><td>Métro Peel (de Maisonneuve / Stanley)</td><td>76,551</td></tr></table>	name	numoftrips	Berri / de Maisonneuve	103,720	Mackay / de Maisonneuve	99,128	Métro Place-des-Arts (de Maisonneuve / de Bleury)	95,343	Métro St-Laurent (de Maisonneuve / St-Laurent)	86,886	Métro Peel (de Maisonneuve / Stanley)	76,551	A count of number of trips was taken by grouping together using end station code. It was then arranged in descending order and limited to 1 st 5 records.	<pre>SELECT name, a.numoftrips FROM (SELECT end_station_code AS code, COUNT(*) AS numoftrips FROM trips GROUP BY code) AS a JOIN stations ON a.code=stations.code ORDER BY a.numoftrips DESC LIMIT 5</pre>
name	numoftrips													
Berri / de Maisonneuve	103,720													
Mackay / de Maisonneuve	99,128													
Métro Place-des-Arts (de Maisonneuve / de Bleury)	95,343													
Métro St-Laurent (de Maisonneuve / St-Laurent)	86,886													
Métro Peel (de Maisonneuve / Stanley)	76,551													

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

Result	Explanation	Query										
<table><tr><th>start_trips</th><th>time_of_day</th></tr><tr><td>30,718</td><td>afternoon</td></tr><tr><td>36,781</td><td>evening</td></tr><tr><td>17,384</td><td>morning</td></tr><tr><td>12,267</td><td>night</td></tr></table>	start_trips	time_of_day	30,718	afternoon	36,781	evening	17,384	morning	12,267	night	<p>1. The station code associated with <i>Mackay / de Maisonneuve</i> was found.</p> <p>2. The station code was then used as a key to find all starting trips associated with <i>Mackay / de Maisonneuve</i> using the 'JOIN' statements.</p> <p>3. The resulting table from step 2 was grouped together using 'time of day' and a count was taken to split starting trips to different time of the day.</p>	<pre>SELECT COUNT(*) AS start_trips, 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 (SELECT trips.start_date, trips.id FROM (SELECT code, name FROM stations WHERE name='Mackay / de Maisonneuve') AS a JOIN trips ON a.code=trips.start_station_code) AS b GROUP BY time_of_day</pre>
start_trips	time_of_day											
30,718	afternoon											
36,781	evening											
17,384	morning											
12,267	night											

	<p>4. The same procedure was carried out to split end trips to the time of the day.</p>	<pre> SELECT COUNT(*) AS end_trips, 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 (SELECT trips.end_date, trips.id FROM (SELECT code, name FROM stations WHERE name='Mackay / de Maisonneuve') AS a JOIN trips ON a.code=trips.end_station_code) AS b </pre>
--	---	--

3.2. The Mackay street is located at the heart of downtown Quebec which hosts the Concordia University, museums, parks and stadiums hence, a hotspot among tourists and biking enthusiasts. The trips split into the time of the day indicates that there is a distinguishable difference between the number of trips during the day. The higher number of trips starting and ending in this station during afternoon and evening indicates most bikers prefer afternoon/evening time for biking around Mackay. Low number of trips at night indicates that it is a less favourable time among biking enthusiasts which could be due to poor visibility and lower temperatures.

4. Which station has proportionally the least number of member trips? How about the most? To damper variance, consider only stations for which there were at least 10 trips starting and ending from it.

Result	Explanation	Query
	<p>1. Created a view which gives the number of member trips from start station codes.</p>	<pre> CREATE VIEW A AS SELECT COUNT(*) AS totalstarttrips, start_station_code, is_member AS membership FROM trips WHERE is_member=1 GROUP BY start_station_code </pre>

<p>Result #1 (3x5)</p> <table border="1"> <thead> <tr> <th>name</th><th>s_code</th><th>total</th></tr> </thead> <tbody> <tr> <td>Mackay / de Maisonneuve</td><td>6,100</td><td>161,984</td></tr> <tr> <td>Métro Place-des-Arts (de Maisonneuve / de Bleury)</td><td>6,078</td><td>142,811</td></tr> <tr> <td>Berri / de Maisonneuve</td><td>6,015</td><td>133,500</td></tr> <tr> <td>Métro Mont-Royal (Rivard / du Mont-Royal)</td><td>6,184</td><td>131,473</td></tr> <tr> <td>Métro Laurier (Rivard / Laurier)</td><td>6,136</td><td>122,359</td></tr> </tbody> </table>	name	s_code	total	Mackay / de Maisonneuve	6,100	161,984	Métro Place-des-Arts (de Maisonneuve / de Bleury)	6,078	142,811	Berri / de Maisonneuve	6,015	133,500	Métro Mont-Royal (Rivard / du Mont-Royal)	6,184	131,473	Métro Laurier (Rivard / Laurier)	6,136	122,359	<p>2. Created a view which gives the number of member trips from end station codes.</p>	<pre>CREATE VIEW B AS SELECT COUNT(*) AS totalendtrips, end_station_code, is_member AS membership FROM trips WHERE is_member=1 GROUP BY end_station_code</pre>
name	s_code	total																		
Mackay / de Maisonneuve	6,100	161,984																		
Métro Place-des-Arts (de Maisonneuve / de Bleury)	6,078	142,811																		
Berri / de Maisonneuve	6,015	133,500																		
Métro Mont-Royal (Rivard / du Mont-Royal)	6,184	131,473																		
Métro Laurier (Rivard / Laurier)	6,136	122,359																		
<p>Result #1 (3x5)</p> <table border="1"> <thead> <tr> <th>name</th><th>s_code</th><th>total</th></tr> </thead> <tbody> <tr> <td>CHSLD Éloria-Lepage (de la Pépinière / de Marseille)</td><td>7,075</td><td>956</td></tr> <tr> <td>Place Longueuil</td><td>5,003</td><td>961</td></tr> <tr> <td>CHSLD Benjamin-Victor-Rousselot (Dickson / Sherbrooke)</td><td>7,009</td><td>964</td></tr> <tr> <td>St-Charles / Montarville</td><td>5,002</td><td>1,412</td></tr> <tr> <td>St-Charles / Charlotte</td><td>5,004</td><td>1,471</td></tr> </tbody> </table> <ul style="list-style-type: none"> Most member trips- Mackay(161984) Least member trips- CHSLD Eloria Lepage(956) 	name	s_code	total	CHSLD Éloria-Lepage (de la Pépinière / de Marseille)	7,075	956	Place Longueuil	5,003	961	CHSLD Benjamin-Victor-Rousselot (Dickson / Sherbrooke)	7,009	964	St-Charles / Montarville	5,002	1,412	St-Charles / Charlotte	5,004	1,471	<p>3. The above 2 views were merged together using the station codes as the key using 'Join' statement. A filter was used were all the stations below 10 starting trips and 10 ending trips were neglected.</p> <p>4. Then the total number of trips (starting and ending) were calculated per station using the '+' operator</p>	<pre>SELECT name, c.s_code, c.total FROM (SELECT A.start_station_code AS s_code, B.end_station_code, A.totalstart trips, B.totalendtrips, A.totalst arttrips+B.totalendtrips AS total FROM A JOIN B ON A.start_station_code=B.end_stat ion_code WHERE (A.totalstarttrips>=10) AND (B.totalendtrips>=10)) AS C JOIN stations ON C.s_code=stations.code ORDER BY c.total DESC</pre>
name	s_code	total																		
CHSLD Éloria-Lepage (de la Pépinière / de Marseille)	7,075	956																		
Place Longueuil	5,003	961																		
CHSLD Benjamin-Victor-Rousselot (Dickson / Sherbrooke)	7,009	964																		
St-Charles / Montarville	5,002	1,412																		
St-Charles / Charlotte	5,004	1,471																		

5. List all stations for which at least 10% of trips are round trips. Recall round trips are those that start and end in the same station. This time we will only consider stations with at least 50 starting trips.

Result	Explanation	Query
	<p>1. Created a view ('total_trips') which gives the number of starting trips per station codes.</p> <p>2. Created a view ('total_round_trips') which gives the number of round trips per station.</p>	<pre>CREATE VIEW total_trips AS SELECT start_station_code, COUNT(*) FROM trips GROUP BY start_station_code CREATE VIEW</pre>

name	fractionoftrips
Métro Jean-Drapeau	0.3020
Métro Angrignon	0.2331
Berlioz / de l'Île des Soeurs	0.2043
LaSalle / 4e avenue	0.2006
Basile-Routhier / Gouin	0.1932
Parc Plage	0.1846
Gare Canora	0.1792
LaSalle / Sénégal	0.1473
Casino de Montréal	0.1437
Quai de la navette fluviale	0.1376
CHSLD Éloria-Lepage (de la Pépinière / de Marseille)	0.1263
de la Commune / Place Jacques-Cartier	0.1106
Jacques-Le Ber / de la Pointe Nord	0.1103
Place du Commerce	0.1082
Collège Édouard-Montpetit	0.1001

3. The above 2 views were merged together using the station codes as the key through 'Join' statement. A filter was used where all the stations below 50 starting trips were ignored. The number of round trips was divided by total trips to get the fraction. Finally a view 'fraction' was created.

4. The stations table was merged with fraction table and all stations with less than 10% round trips were filtered out.

```
total_round_trips AS
SELECT
start_station_code,
COUNT(*)
FROM trips
WHERE
start_station_code=end_
station_code
GROUP BY
start_station_code
```

```
CREATE VIEW fraction AS
SELECT
total_trips.start_stati
on_code AS code,
total_round_trips.total
roundtrips/total_trips.
totaltrips AS
fractionoftrips
FROM total_trips JOIN
total_round_trips ON
total_trips.start_stati
on_code=total_round_tri
ps.start_station_code
WHERE
total_trips.totaltrips
>=50
```

```
SELECT
stations.name, fraction.
fractionoftrips
FROM fraction JOIN
stations ON
fraction.code=stations.
code
WHERE
fraction.fractionoftrip
s >=0.1
ORDER BY
fraction.fractionoftrip
s DESC
```

5.5 From previous queries it is evident that majority of round trips are taken by members during weekends. A higher fraction of round trips are observed in 'Jean Drapeau' and 'Angrignon' stations as these are location which hosts museums, casinos, zoo and racing circuits. Hence, these are prime location of interest among tourists.