Data Processing of sample Cyclistic data using SQL By [Aniekan Ekanem](mailto:aniekancharlesekanem@gmail.com)

**Processing Stage**

While using Big Query to process the dataset, out of the 12 data that was downloaded, I was only able to work with 6 data since Big Query can only allow data size of less than 100MB. Using appropriate file naming conventions, the following are the date files studied:

* bike\_sharing\_2022\_nov
* bike\_sharing\_2022\_dec
* bike\_sharing\_2023\_jan
* bike\_sharing\_2023\_feb
* bike\_sharing\_2023\_mar
* bike\_sharing\_2023\_apr

The tool I will use for processing the data is SQL because it is handy for working with larger datasets and provides more functionalities. Another tool I could use is R but I will stick with SQL for now.

After cross checking data in accordance with the credibility check, I have been able to confirm the data's integrity.

1. **Steps taken to ensure data is clean using SQL**

i. Checking for duplicates:

SELECT COUNT(\*) AS duplicate\_rows\_count

FROM (

SELECT \*, COUNT(\*) AS count

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

GROUP BY ride\_id,

rideable\_type,

started\_at,

ended\_at,

start\_station\_name,

start\_station\_id,

end\_station\_name,

end\_station\_id,

start\_lat,

start\_lng,

end\_lat,

end\_lng,

member\_casual,

HAVING COUNT(\*) > 1

)

AS duplicates;

The above code was carried out for the remaining 5 data

Result:

| Row | **duplicate\_rows\_count** |  |
| --- | --- | --- |
| 1 | 0 |  |

ii. Checking for the presence of “NA” string in the tables submitted:

SELECT \*

FROM `motivate\_int\_inc.bike\_sharing\_2023\_mar`

WHERE CONCAT(

ride\_id,

rideable\_type,

started\_at,

ended\_at,

start\_station\_name,

start\_station\_id,

end\_station\_name,

end\_station\_id, start\_lat,

start\_lng,

end\_lat,

end\_lng,

Member\_casual

)

LIKE '%NA%'

Result:

There is no data to display.

1. **Calculating the length of each ride as ‘ride\_length\_hms’.**

**SELECT \*,**

**FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND))) AS ride\_length\_hms**

**FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`**

1. **Calculating the days of the week to know which days each bike-sharing took place as well as the corresponding days based on the day value**

SELECT \*,

EXTRACT(DAYOFWEEK FROM started\_at) AS day\_of\_week,

CASE

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 1 THEN 'sunday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 2 THEN 'monday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 3 THEN 'tuesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 4 THEN 'wednesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 5 THEN 'thursday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 6 THEN 'friday'

ELSE 'saturday'

END AS starting\_day

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`;

**4. Analysis**

1. **Analysis using SQL**
   1. Getting the number of Cyclistic members and casual riders by month period

## For Cyclistic members

SELECT member\_casual, COUNT(\*) AS count

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

WHERE member\_casual = 'member'

GROUP BY member\_casual;

## For Casual riders

SELECT member\_casual, COUNT(\*) AS count

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

WHERE member\_casual = casual

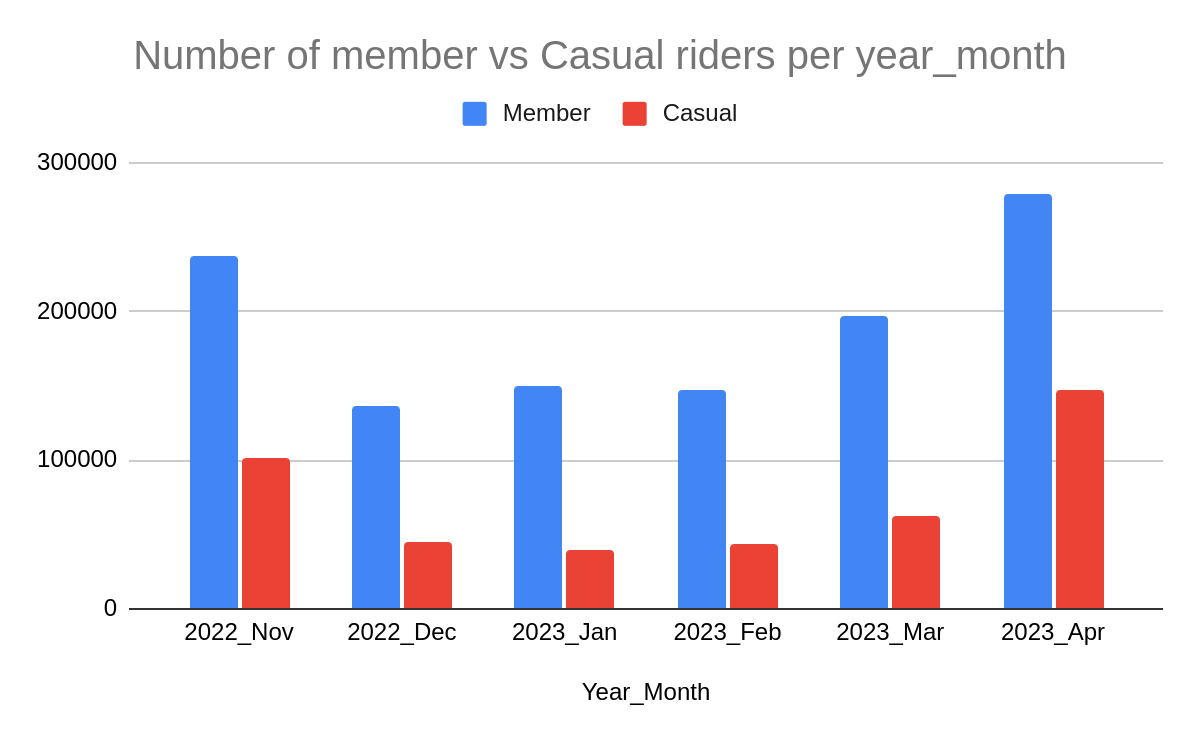
GROUP BY member\_casual;

The above code was also used to get the number of Cyclistic members and Casual riders for other month period.

Below is the table that shows the relationship between the number of Cyclistic members and casual riders according to each month period.

| **month** | **Member** | **Casual** | **Overall Total** | **Member-Casual Difference** |
| --- | --- | --- | --- | --- |
| 2022\_Nov | 236963 | 100772 | 337735 | 136191 |
| 2022\_Dec | 136912 | 44894 | 181806 | 92018 |
| 2023\_Jan | 150293 | 40008 | 190301 | 110285 |
| 2023\_Feb | 147429 | 43016 | 190445 | 104413 |
| 2023\_Mar | 196477 | 62201 | 258678 | 134276 |
| 2023\_Apr | 279305 | 147285 | 426590 | 132020 |

**Table 1: Number of members to casual riders relationship ratio**



**Chart 1: Bar chart of number of member to casual riders**

* 1. Average ride\_length for members and casual riders using SQL

## for Cyclistic members

SELECT

FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(CAST(AVG(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND)) AS INT64))) AS avg\_ride\_length\_hms

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

WHERE member\_casual = 'member';

## for casual riders

SELECT

FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(CAST(AVG(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND)) AS INT64))) AS avg\_ride\_length\_hms

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

WHERE member\_casual = casual;

Below is the summary result

| **month** | **Avg\_ride\_length\_Member** | **Avg\_ride\_length\_Casual** | **Overall\_Total\_Avg\_ride\_length** |
| --- | --- | --- | --- |
| 2022\_Nov | 00:11:08 | 00:21:17 | 00:32:25 |
| 2022\_Dec | 00:10:37 | 00:22:17 | 00:32:54 |
| 2023\_Jan | 00:10:22 | 00:22:55 | 00:33:17 |
| 2023\_Feb | 00:10:43 | 00:23:12 | 00:33:55 |
| 2023\_Mar | 00:10:27 | 00:21:25 | 00:31:52 |
| 2023\_Apr | 00:11:42 | 00:27:40 | 00:39:22 |
| **Average** | **00:10:50** | **00:23:08** | **00:33:58** |

**Table 2: Members to Casual riders relationship ratio on average ride length**

* 1. Calculating the maximum ride length using SQL

SELECT

FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(MAX(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND)))) AS max\_ride\_length\_hms

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`;

Result:

| Row | **max\_ride\_length\_hms** |  |
| --- | --- | --- |
| 1 | 14:52:36 |  |

* 1. Getting the mode day of the week

WITH daily\_counts AS (

SELECT EXTRACT(DAYOFWEEK FROM started\_at) AS day\_of\_week,

COUNT(\*) AS count

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

GROUP BY day\_of\_week

),

max\_count AS (

SELECT MAX(count) AS max\_count

FROM daily\_counts

)

SELECT day\_of\_week, starting\_day

FROM (

SELECT EXTRACT(DAYOFWEEK FROM started\_at) AS day\_of\_week,

CASE

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 1 THEN 'sunday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 2 THEN 'monday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 3 THEN 'tuesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 4 THEN 'wednesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 5 THEN 'thursday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 6 THEN 'friday'

ELSE 'saturday'

END AS starting\_day,

ROW\_NUMBER() OVER (ORDER BY COUNT(\*) DESC) AS rn

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

GROUP BY day\_of\_week, starting\_day

) subquery

WHERE rn = 1

The result is displayed on the table below

| **month** | **Modal\_day of the week** |
| --- | --- |
| 2022\_Nov | Wednesday |
| 2022\_Dec | Thursday |
| 2023\_Jan | Tuesday |
| 2023\_Feb | Tuesday |
| 2023\_Apr | Saturday |

**Table 3: Modal day of the week per month**

* 1. Calculating the number of rides for users based on the days of the week by counting the number of ride\_ids (combine the 6 tables for this)

SELECT starting\_day, COUNT(ride\_id) AS ride\_count

FROM (

SELECT ride\_id,

EXTRACT(DAYOFWEEK FROM started\_at) AS day\_of\_week,

CASE

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 1 THEN 'sunday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 2 THEN 'monday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 3 THEN 'tuesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 4 THEN 'wednesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 5 THEN 'thursday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 6 THEN 'friday'

ELSE 'saturday'

END AS starting\_day

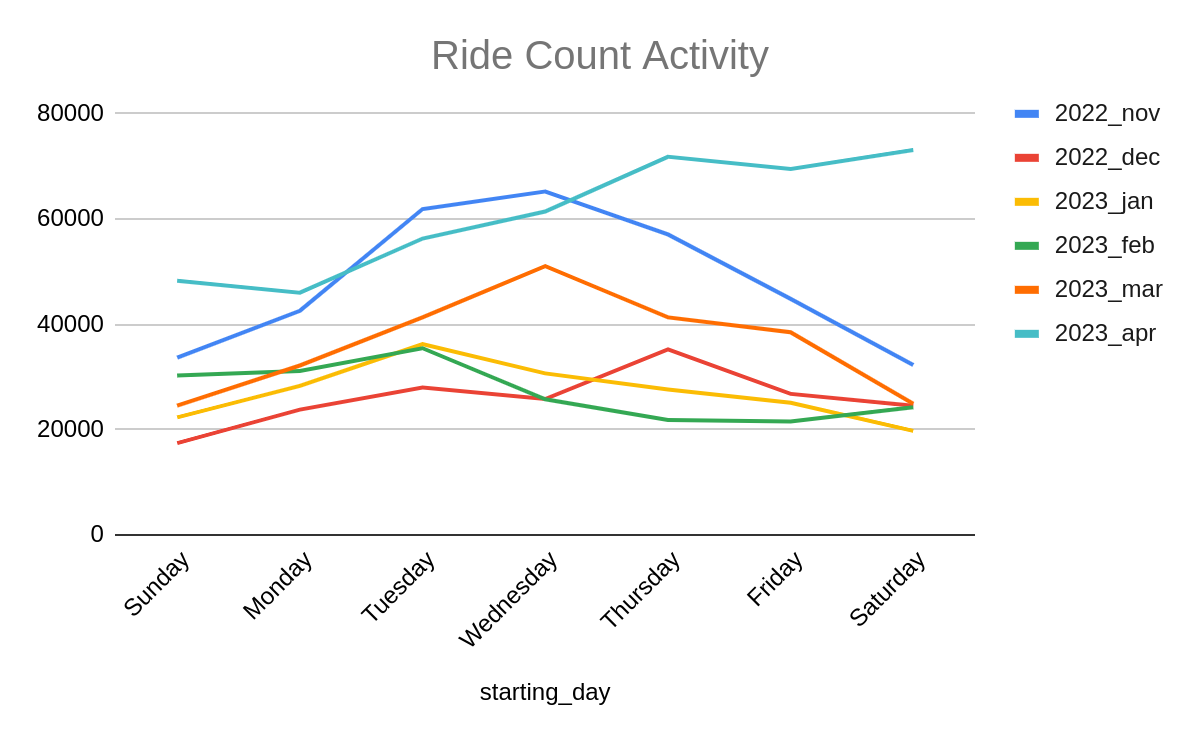
FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

) subquery

GROUP BY starting\_day

| **starting\_day** | **ride-count** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| 2022\_nov | 2022\_dec | 2023\_jan | 2023\_feb | 2023\_mar | 2023\_apr |
| Sunday | 33704 | 17460 | 22366 | 30292 | 24575 | 48314 |
| Monday | 42605 | 23825 | 28347 | 31179 | 32208 | 46043 |
| Tuesday | 61922 | 28025 | 36281 | 35484 | 41347 | 56318 |
| Wednesday | 65227 | 25840 | 30721 | 25792 | 51056 | 61446 |
| Thursday | 57106 | 35272 | 27667 | 21852 | 41347 | 71825 |
| Friday | 44862 | 26818 | 25121 | 21561 | 38517 | 69504 |
| Saturday | 32309 | 24566 | 19798 | 24285 | 24948 | 73140 |

**Table 4: Ride count based on day of week per month**

****

**Chart 2: Line chart of ride count activity**

1. **Analysis from the combined tables**

I combined the 6 tables together to form one table of data using SQL. Find below the code:

CREATE TABLE `motivate\_int\_inc.divvy\_tripdata\_combined2` AS

SELECT \* FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

UNION ALL

SELECT \* FROM `motivate\_int\_inc.bike\_sharing\_2022\_dec`

UNION ALL

SELECT \* FROM `motivate\_int\_inc.bike\_sharing\_2023\_jan`

UNION ALL

SELECT \* FROM `motivate\_int\_inc.bike\_sharing\_2023\_feb`

UNION ALL

SELECT \* FROM `motivate\_int\_inc.bike\_sharing\_2023\_mar`

UNION ALL

SELECT \* FROM `motivate\_int\_inc.bike\_sharing\_2023\_apr`;

* 1. **Calculating the length of each ride as ‘ride\_length\_hms’.**

**SELECT \*,**

**FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND))) AS ride\_length\_hms**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

* 1. **Calculating the days of the week to know which days each bike-sharing took place as well as the corresponding days based on the day value**

SELECT \*,

EXTRACT(DAYOFWEEK FROM started\_at) AS day\_of\_week,

CASE

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 1 THEN 'sunday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 2 THEN 'monday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 3 THEN 'tuesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 4 THEN 'wednesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 5 THEN 'thursday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 6 THEN 'friday'

ELSE 'saturday'

END AS starting\_day

FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`;

* 1. **Getting the number of Cyclistic members and casual riders**

## For Cyclistic members

SELECT member\_casual, COUNT(\*) AS count

FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`

WHERE member\_casual = 'member'

GROUP BY member\_casual;

## For Casual riders

SELECT member\_casual, COUNT(\*) AS count

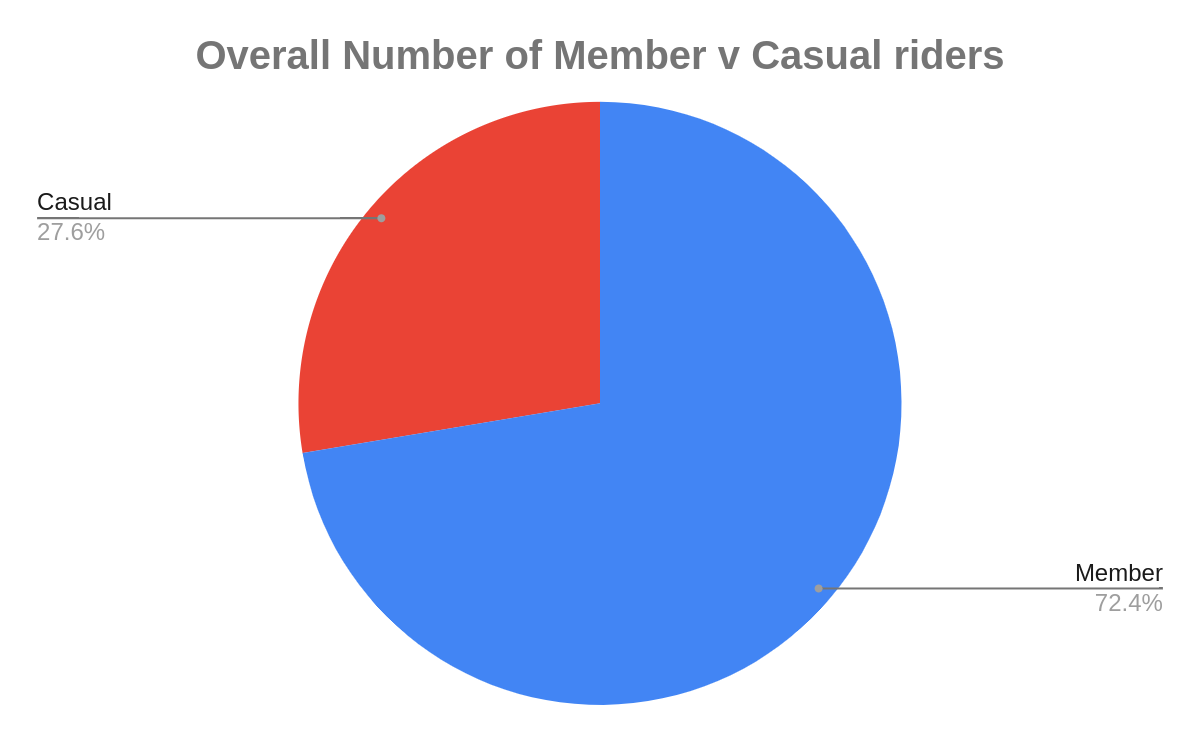
FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`

WHERE member\_casual = 'casual'

GROUP BY member\_casual;

| **Member** | **Casual** | **Overall Total** | **Member-Casual Difference** |
| --- | --- | --- | --- |
| 1147379 | 438176 | 1585555 | 709203 |

**Table 5: Number of members to casual riders relationship ratio using combined tables**

****

**Chart3: Pie chart showing the number of Members to casual riders**

* 1. **Calculating total rides by user type**

**## for member**

**SELECT COUNT(ride\_id)**

**FROM `my-data-project-24723.motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE member\_casual = 'member'**

**## for casual**

**SELECT COUNT(ride\_id)**

**FROM `my-data-project-24723.motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE member\_casual = casual**

The result of this finding is seen in the table below:

| **Member** | 1147379 |
| --- | --- |
| **Casual** | 438176 |

* 1. **Calculating the total rides by bike type**

**## for classic\_bike**

**SELECT COUNT(rideable\_type)**

**FROM `my-data-project-24723.motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE rideable\_type = 'classic\_bike'**

**## for electric\_bike**

**SELECT COUNT(rideable\_type)**

**FROM `my-data-project-24723.motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE rideable\_type = 'electric\_bike'**

This leads to the result as seen in the table below:

| **Classic\_bike** | 674954 |
| --- | --- |
| **Electric\_bike** | 886950 |

* 1. **Calculating the average ride length by user type**

**## for members**

**SELECT**

**FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(CAST(AVG(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND)) AS INT64))) AS avg\_ride\_length\_hms**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE member\_casual = 'member';**

**## for casual riders**

**SELECT**

**FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(CAST(AVG(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND)) AS INT64))) AS avg\_ride\_length\_hms**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE member\_casual = casual;**

This leads to the result as seen in the table below:

| **Members** | 00:10:56 |
| --- | --- |
| **Casual riders** | 00:23:53 |

* 1. **Getting the ride count amount by starting days**

SELECT starting\_day, COUNT(ride\_id) AS ride\_count

FROM (

SELECT ride\_id,

EXTRACT(DAYOFWEEK FROM started\_at) AS day\_of\_week,

CASE

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 1 THEN 'sunday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 2 THEN 'monday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 3 THEN 'tuesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 4 THEN 'wednesday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 5 THEN 'thursday'

WHEN EXTRACT(DAYOFWEEK FROM started\_at) = 6 THEN 'friday'

ELSE 'saturday'

END AS starting\_day

FROM `motivate\_int\_inc.bike\_sharing\_2022\_nov`

) subquery

GROUP BY starting\_day

Below is the result

| **starting\_day** | **ride\_count** |
| --- | --- |
| friday | 226383 |
| tuesday | 259377 |
| saturday | 199046 |
| thursday | 259749 |
| sunday | 176711 |
| monday | 204207 |
| wednesday | 260082 |

* 1. **Calculating the total rides by months**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(MONTH FROM started\_at) = 1;**

***(1 for Jan, 2 for feb, 3 for mar and so on)***

**The result is displayed in the table below**

| **Month** | **ride\_count** |
| --- | --- |
| November | 181806 |
| December | 259377 |
| January | 190301 |
| February | 190445 |
| March | 258678 |
| April | 426590 |

* 1. **Calculating the total riders:**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**Result = 1585555**

* 1. **Calculating average ride length:**

**SELECT**

**FORMAT\_TIMESTAMP('%H:%M:%S', TIMESTAMP\_SECONDS(CAST(AVG(TIMESTAMP\_DIFF(ended\_at, started\_at, SECOND)) AS INT64))) AS avg\_ride\_length\_hms**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**Result: 00:14:31**

* 1. **Getting the busiest time:**

**## for morning hours**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(HOUR FROM started\_at) >= 6 AND EXTRACT(HOUR FROM started\_at) < 12;**

**Result: 434286**

**## for afternoon hours**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(HOUR FROM started\_at) >= 12 AND EXTRACT(HOUR FROM started\_at) < 18;**

**Result; 714686**

**## for evening hours**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(HOUR FROM started\_at) >= 18 AND EXTRACT(HOUR FROM started\_at) < 6;**

**Result; 0**

* 1. **Getting the busiest weekday:**

**## for sundays**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(DAYOFWEEK FROM ended\_at) = 1**

**Result: 177405**

**—----------------------------------------------------------------------------------------------**

**## for mondays**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(DAYOFWEEK FROM ended\_at) = 2**

**Result: 204365**

**—----------------------------------------------------------------------------------------------**

**## for tuesdays**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(DAYOFWEEK FROM ended\_at) = 3**

**Result: 259289**

**—----------------------------------------------------------------------------------------------**

**## for tuesdays**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(DAYOFWEEK FROM ended\_at) = 4**

**Result: 259989**

**—----------------------------------------------------------------------------------------------**

**## for tuesdays**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(DAYOFWEEK FROM ended\_at) = 5**

**Result: 259643**

**—----------------------------------------------------------------------------------------------**

**## for tuesdays**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(DAYOFWEEK FROM ended\_at) = 6**

**Result: 226074**

**—----------------------------------------------------------------------------------------------**

**## for tuesdays**

**SELECT COUNT(\*) AS count**

**FROM `motivate\_int\_inc.divvy\_tripdata\_combined2`**

**WHERE EXTRACT(DAYOFWEEK FROM ended\_at) = 7**

**Result: 198790**

**—----------------------------------------------------------------------------------------------**

* 1. **Getting the busiest season:**  There are four seasons which are winter, summer, spring and fall.

Winter does occur during the months of December, January, and February in the Northern Hemisphere, and also occurs during June, July, and August in the Southern Hemisphere. In this context, Cyclistic is based in Chicago, and Chicago is located in the Northern Hemisphere, hence there will be winter in the months of December, January, and February. For other seasons, the following applies:

* Spring: March, April, May
* Summer: June, July, August
* Fall: September, October, November

Based on these, the busiest season will be gotten from the busiest months which has been deduced from the code in viii above. April being the busiest falls under Spring. This means the busiest season is Spring.