**Google data Analytics Capstone Project**

**Part 2 (a): *2019 Q1 and 2020 Q1(Using SQL and Tableau)***

For this analyzation, SQLite was used for cleaning and transforming two data sets: Divvy\_Trips\_2019\_Q1 and Divvy\_Trips\_2020\_Q1. The programming started from simple SELECT, FROM and WHERE steps in SQL to calculate total number of males, females, subscribers, and customers in 2019 data set. This progressed towards more defined codes for modifying columns in 2020 data set to match those in 2019 data set, and finally combining the data sets for visualizations in Tableau. of trip duration (length of a single trip) and day of week has been made for both 2019 and 2020.

Tableau Viz: [Trips Trend Analysis](https://public.tableau.com/app/profile/bhavna1106/viz/TripsTrendAnalysis/Dashboard1?publish=yes)

**Using SQl to Clean data**

1. **Calculating total number of males from 2019 data set.**

A screenshot of a computer

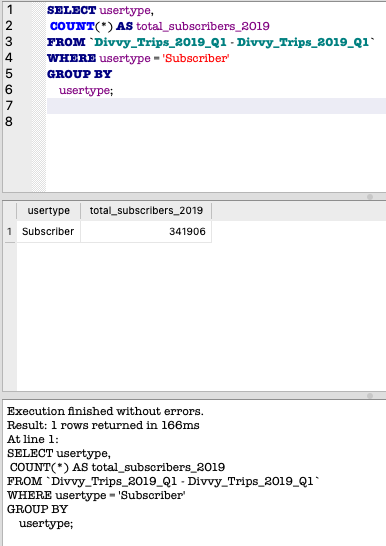
Description automatically generated

1. **Calculating total number of females from 2019 data set.**

A screenshot of a computer

Description automatically generated

1. **Calculating total number of Subscribers from 2019 data set.**



1. **Calculating total number of Customers from 2019 data set.**

A screenshot of a computer

Description automatically generated

1. **Calculating number of trips from each station, from 2019 data-set and creating a new table named stations\_2019.**

CREATE TABLE IF NOT EXISTS stations\_2019 (

from\_station\_name TEXT,

number\_of\_trips INTEGER

);

INSERT INTO stations\_2019 (from\_station\_name, number\_of\_trips)

SELECT

from\_station\_name,

COUNT(\*) AS number\_of\_trips

FROM `Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1`

GROUP BY from\_station\_name;

1. **Creating a summary table of 2019**

CREATE TABLE IF NOT EXISTS summary\_2019 (

from\_station\_name TEXT,

station\_frequency INTEGER,

total\_males INTEGER,

total\_females INTEGER,

male\_subscribers INTEGER,

female\_subscribers INTEGER,

male\_customers INTEGER,

female\_customers INTEGER,

total\_subscribers INTEGER,

total\_customers INTEGER

);

INSERT INTO summary\_2019 (from\_station\_name, station\_frequency, total\_males, total\_females, male\_subscribers, female\_subscribers, male\_customers, female\_customers, total\_subscribers, total\_customers)

SELECT

from\_station\_name,

COUNT(\*) AS station\_frequency,

SUM(CASE WHEN gender = 'Male' THEN 1 ELSE 0 END) AS total\_males,

SUM(CASE WHEN gender = 'Female' THEN 1 ELSE 0 END) AS total\_females,

SUM(CASE WHEN usertype = 'Subscriber' AND gender = 'Male' THEN 1 ELSE 0 END) AS male\_subscribers,

SUM(CASE WHEN usertype = 'Subscriber' AND gender = 'Female' THEN 1 ELSE 0 END) AS female\_subscribers,

SUM(CASE WHEN usertype = 'Customer' AND gender = 'Male' THEN 1 ELSE 0 END) AS male\_customers,

SUM(CASE WHEN usertype = 'Customer' AND gender = 'Female' THEN 1 ELSE 0 END) AS female\_customers,

SUM(CASE WHEN usertype = 'Subscriber' THEN 1 ELSE 0 END) AS total\_subscribers,

SUM(CASE WHEN usertype = 'Customer' THEN 1 ELSE 0 END) AS total\_customers

FROM `Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1`

GROUP BY from\_station\_name

ORDER BY station\_frequency DESC;

1. **Calculating total number of Members from 2020 data set.**

A screenshot of a computer

Description automatically generated

1. **Calculating total number of Casual riders from 2020 data set.**

A screenshot of a computer

Description automatically generated

1. **Calculating number of trips from each station, from 2020 data-set and creating a new table named stations\_2020.**

CREATE TABLE IF NOT EXISTS stations\_2020 (

start\_station\_nameTEXT,

number\_of\_trips\_2020 INTEGER

);

INSERT INTO stations\_2020 (start\_station\_name, number\_of\_trips\_2020)

SELECT

start\_station\_name,

COUNT(\*) AS number\_of\_trips\_2020

FROM `Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1`

GROUP BY start\_station\_name;

1. **Joining two tables: stations\_2019 and stations\_2020 and creating a new table combined\_data**

CREATE TABLE IF NOT EXISTS combined\_data AS

SELECT

start\_time,

end\_time,

from\_station\_name,

to\_station\_name AS end\_station\_name,

usertype

FROM "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

UNION ALL

SELECT

start\_time,

end\_time,

from\_station\_name,

to\_station\_name,

usertype

FROM "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1";

1. **Separating time from start\_time in Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1**

-- Add a new column to the existing table

ALTER TABLE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

ADD COLUMN s\_time TEXT;

-- Update the new column with the desired values

UPDATE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

SET s\_time = SUBSTR(start\_time, 12, 8);

1. **Separating time from end\_time in Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1**

-- Add a new column to the existing table

ALTER TABLE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

ADD COLUMN e\_time TEXT;

-- Update the new column with the desired values

UPDATE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

SET e\_time = SUBSTR(end\_time, 12, 8);

1. **Create a new column “length\_of\_trip” by subtracting s\_time from e\_time**

--Add a new column to store the length of the trip

ALTER TABLE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

ADD COLUMN length\_of\_trip TEXT;

-- Update the new column with the calculated difference

UPDATE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

SET length\_of\_trip = TIME(

julianday(e\_time) \* 86400 - julianday(s\_time) \* 86400,

'unixepoch'

);

1. **Separating day from start\_time**

-- Add a new column to store the day of the week

ALTER TABLE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

ADD COLUMN day\_of\_week TEXT;

-- Update the new column with the day of the week

UPDATE "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

SET day\_of\_week = CASE

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 0 THEN 'Sunday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 1 THEN 'Monday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 2 THEN 'Tuesday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 3 THEN 'Wednesday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 4 THEN 'Thursday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 5 THEN 'Friday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 6 THEN 'Saturday'

END;

1. **Separating day from start\_time in Divvy\_Trips\_2020\_Q1 – Divvy\_Trips\_2020\_Q1**

-- Add a new column to store the day of the week

ALTER TABLE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

ADD COLUMN day\_of\_week TEXT;

-- Update the new column with the day of the week

UPDATE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

SET day\_of\_week = CASE

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 0 THEN 'Sunday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 1 THEN 'Monday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 2 THEN 'Tuesday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 3 THEN 'Wednesday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 4 THEN 'Thursday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 5 THEN 'Friday'

WHEN CAST(strftime('%w', start\_time) AS INTEGER) = 6 THEN 'Saturday'

END;

1. **Separating s\_time from start\_time in Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1**

-- Add a new column to the existing table

ALTER TABLE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

ADD COLUMN s\_time TEXT;

-- Update the new column with the desired values

UPDATE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

SET s\_time = SUBSTR(start\_time, 12, 8);

1. **Separating e\_time from end\_time in Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1**

-- Add a new column to the existing table

ALTER TABLE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

ADD COLUMN e\_time TEXT;

-- Update the new column with the desired values

UPDATE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

SET e\_time = SUBSTR(end\_time, 12, 8);

1. **Create a new column “length\_of\_trip” by subtracting s\_time from e\_time**

Add a new column to store the length of the trip

ALTER TABLE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

ADD COLUMN length\_of\_trip TEXT;

-- Update the new column with the calculated difference

UPDATE "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1"

SET length\_of\_trip = TIME(

julianday(e\_time) \* 86400 - julianday(s\_time) \* 86400,

'unixepoch'

);

1. **Combining data from both tables:**

-- Create a new table "summary\_data"

CREATE TABLE summary\_data AS

SELECT from\_station\_name AS station\_name, NULL AS end\_station\_name, length\_of\_trip, day\_of\_week

FROM "Divvy\_Trips\_2019\_Q1 - Divvy\_Trips\_2019\_Q1"

UNION ALL

SELECT from\_station\_name AS station\_name, to\_station\_name AS end\_station\_name, length\_of\_trip, day\_of\_week

FROM "Divvy\_Trips\_2020\_Q1 - Divvy\_Trips\_2020\_Q1";