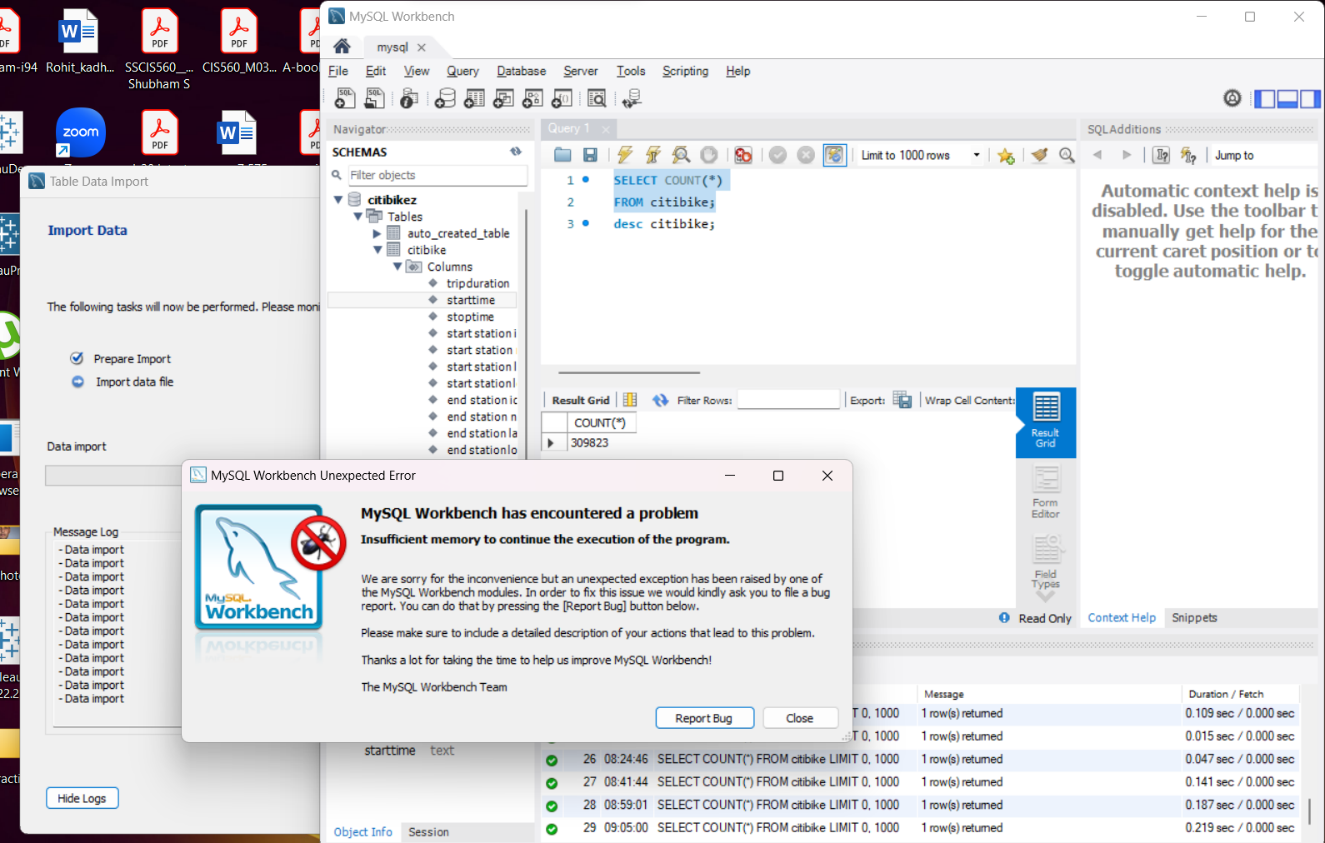
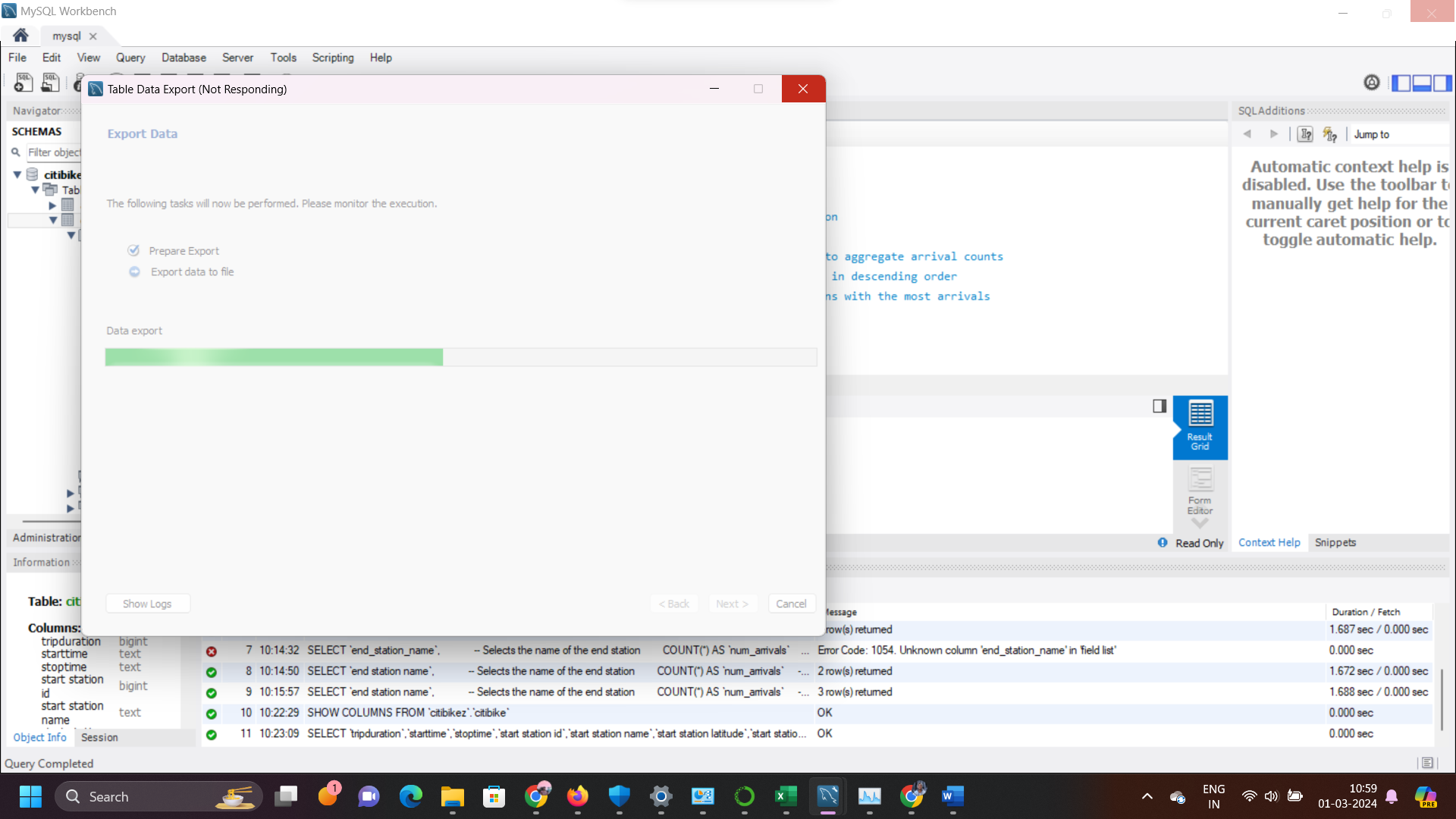
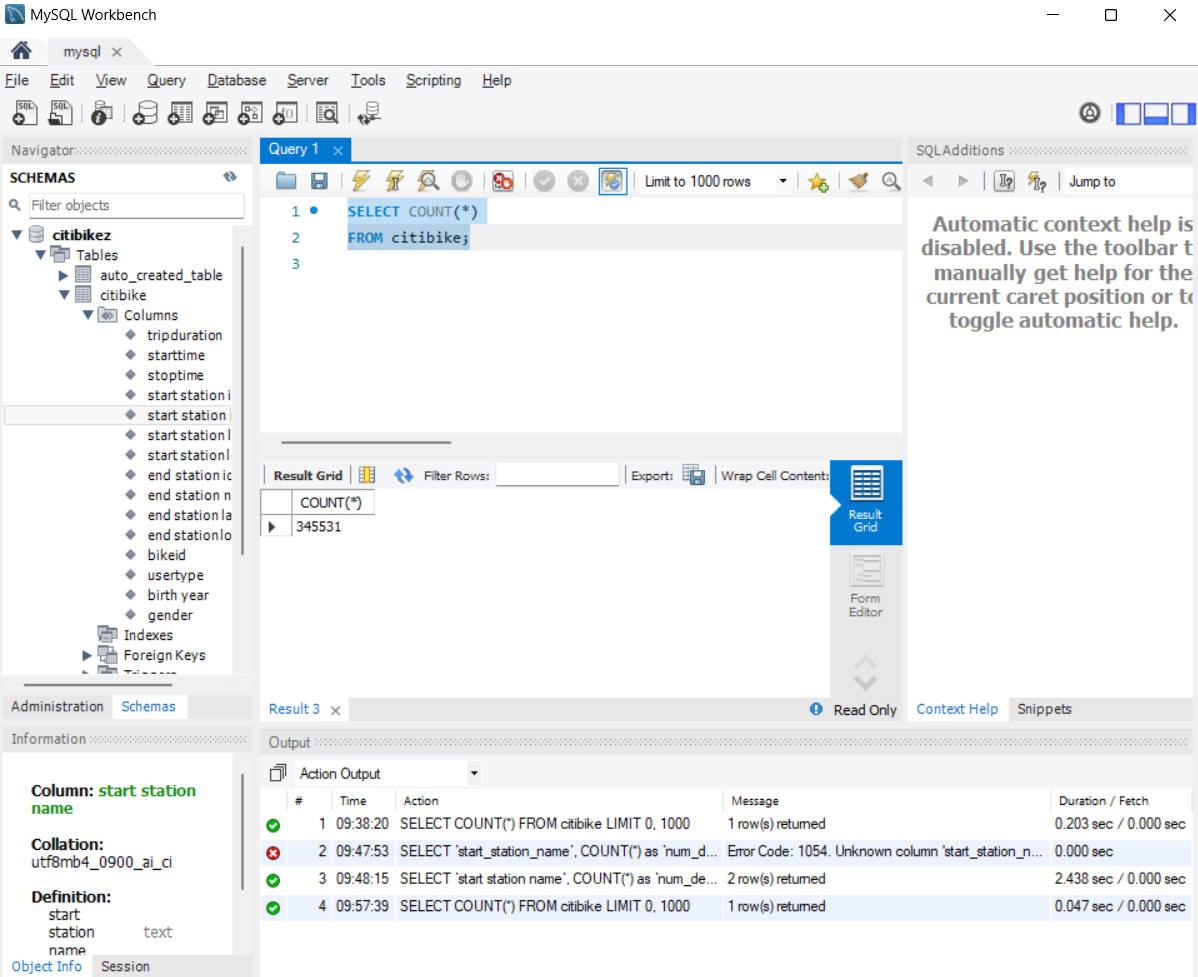
**Running Analysis On MySQl Workbench**

While importing the dataset, only 340k records got imported and then my system ran into some memory problems due to which the import couldn’t complete. I have continued with my analysis with the partial dataset and here are my findings.

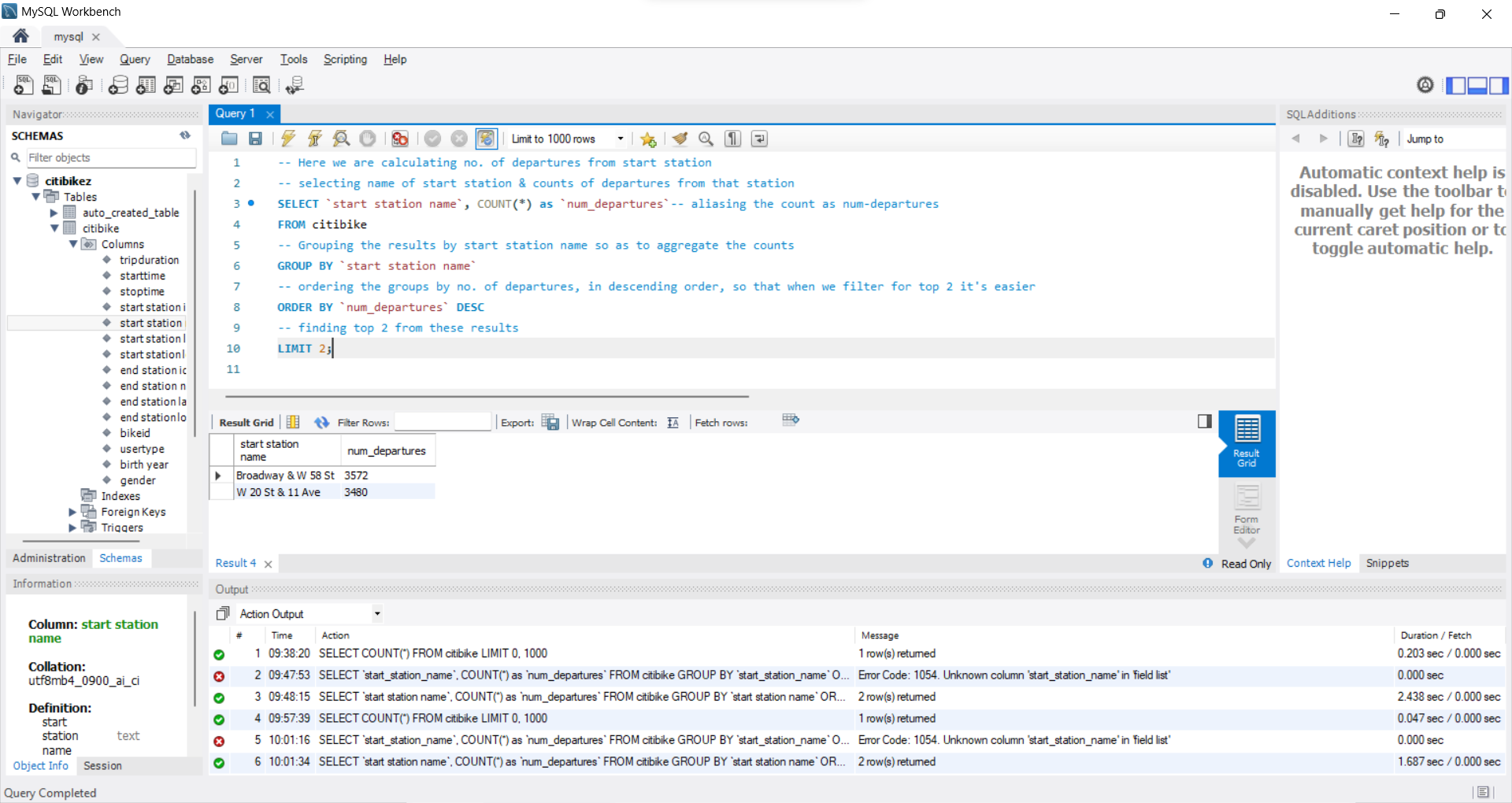






Answering questions based on these records(345531)

**Question 1)** Which 2 Stations are Most Used by riders?

  
Queries

Finding most used start station.

-- Here we are calculating no. of departures from start station

-- selecting name of start station & counts of departures from that station

SELECT `start station name`, COUNT(\*) as `num\_departures`-- aliasing the count as num-departures

FROM citibike

-- Grouping the results by start station name so as to aggregate the counts

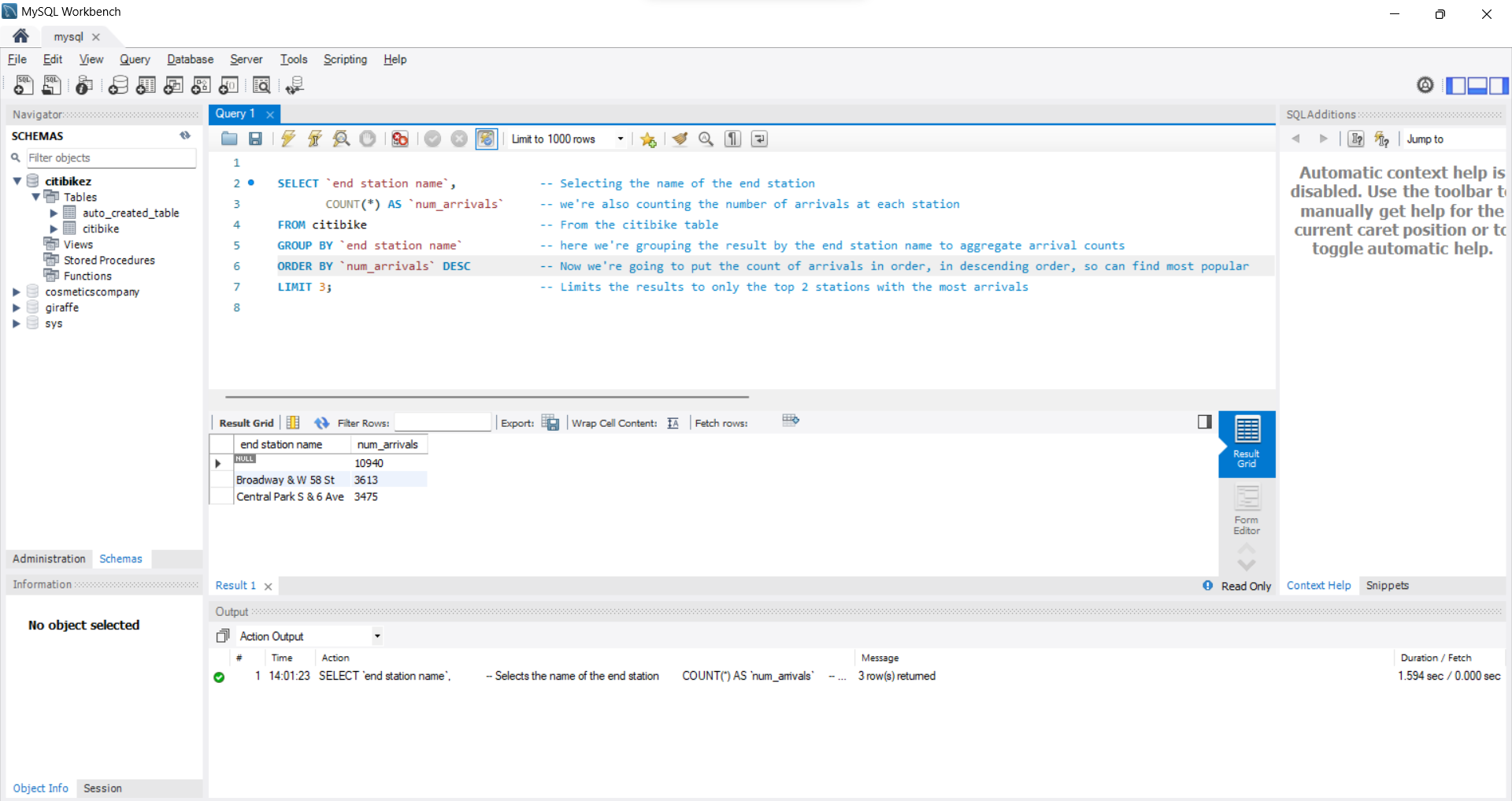
GROUP BY `start station name`

-- ordering the groups by no. of departures, in descending order, so that when we filter for top 2 it's easier

ORDER BY `num\_departures` DESC

-- finding top 2 from these results

LIMIT 2;  
  
Finding most used end station.



Query is below:  
SELECT `end station name`, -- Selecting the name of the end station

COUNT(\*) AS `num\_arrivals` -- we're also counting the number of arrivals at each station

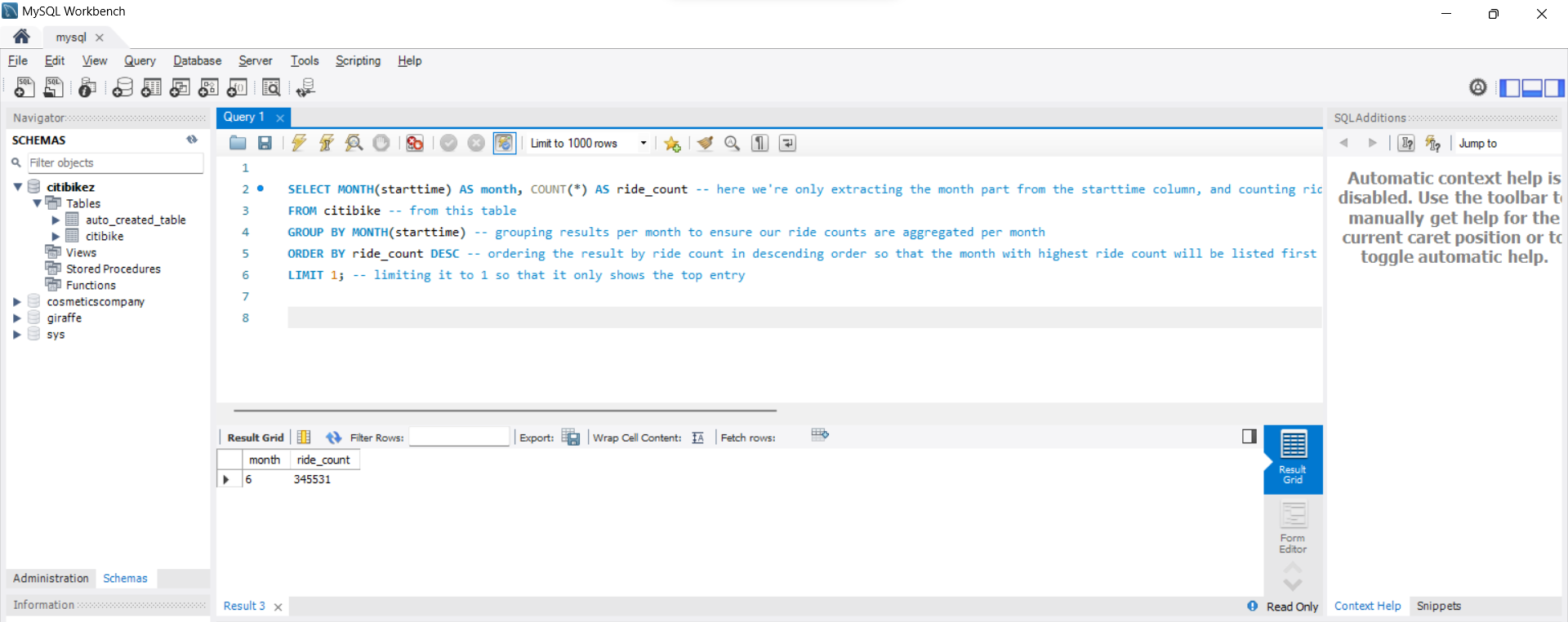
FROM citibike -- From the citibike table

GROUP BY `end station name` -- here we're grouping the result by the end station name to aggregate arrival counts

ORDER BY `num\_arrivals` DESC -- Now we're going to put the count of arrivals in order, in descending order, so can find most popular first in the list

LIMIT 3; -- Limits the results to only the top 2 stations with the most arrivals

**ANSWER**Looking at the outputs from both queries we find that Broadway & W 58 St is the most popular, followed by W 20 St and 11 Ave  
  
**Question 2)** Which month is the busiest time for Citi bike?



Query  
  
SELECT MONTH(starttime) AS month, COUNT(\*) AS ride\_count -- here we're only extracting the month part from the starttime column, and counting rides

FROM citibike -- from this table

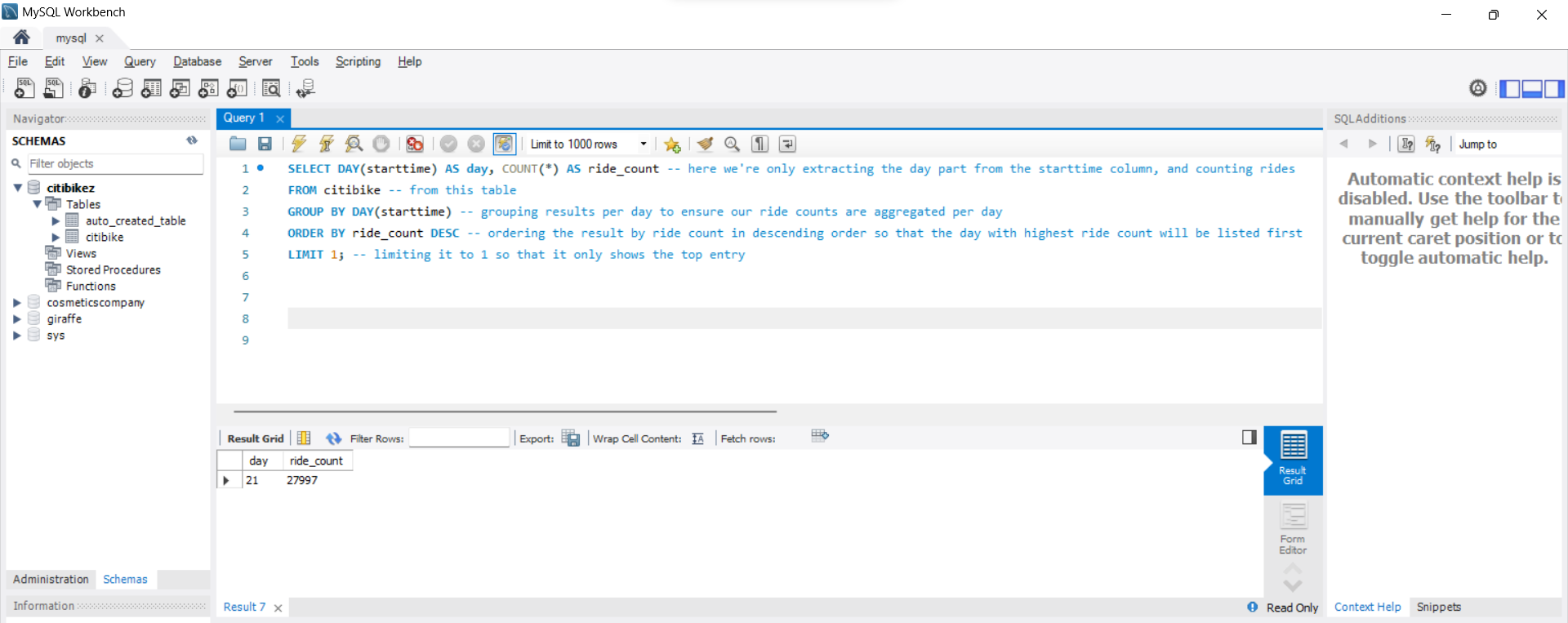
GROUP BY MONTH(starttime) -- grouping results per month to ensure our ride counts are aggregated per month

ORDER BY ride\_count DESC -- ordering the result by ride count in descending order so that the month with highest ride count will be listed first

LIMIT 1; -- limiting it to 1 so that it only shows the top entry

**ANSWER**

It shows that the 6th month is the most popular returning a count of 345531, and then later I realized that the whole dataset has data for only the month of June, hence it’s the most popular month, I then got curious as to which was the most popular day in the month



Query

SELECT DAY(starttime) AS day, COUNT(\*) AS ride\_count -- here we're only extracting the day part from the starttime column, and counting rides

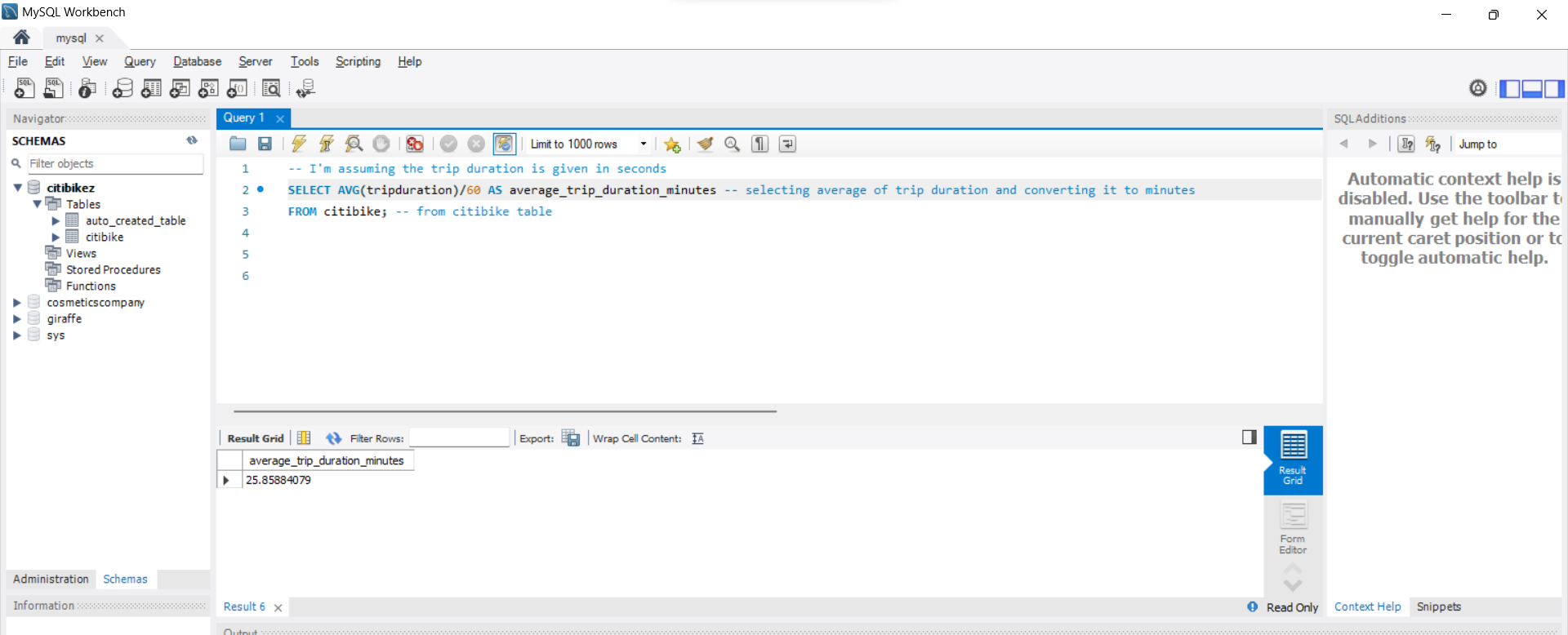
FROM citibike -- from this table

GROUP BY DAY(starttime) -- grouping results per day to ensure our ride counts are aggregated per day

ORDER BY ride\_count DESC -- ordering the result by ride count in descending order so that the day with highest ride count will be listed first

LIMIT 1; -- limiting it to 1 so that it only shows the top entry

Turns out it was 21st of June, with a ride count of almost 28k  
  
**Question 3)** What is the average time of ride?



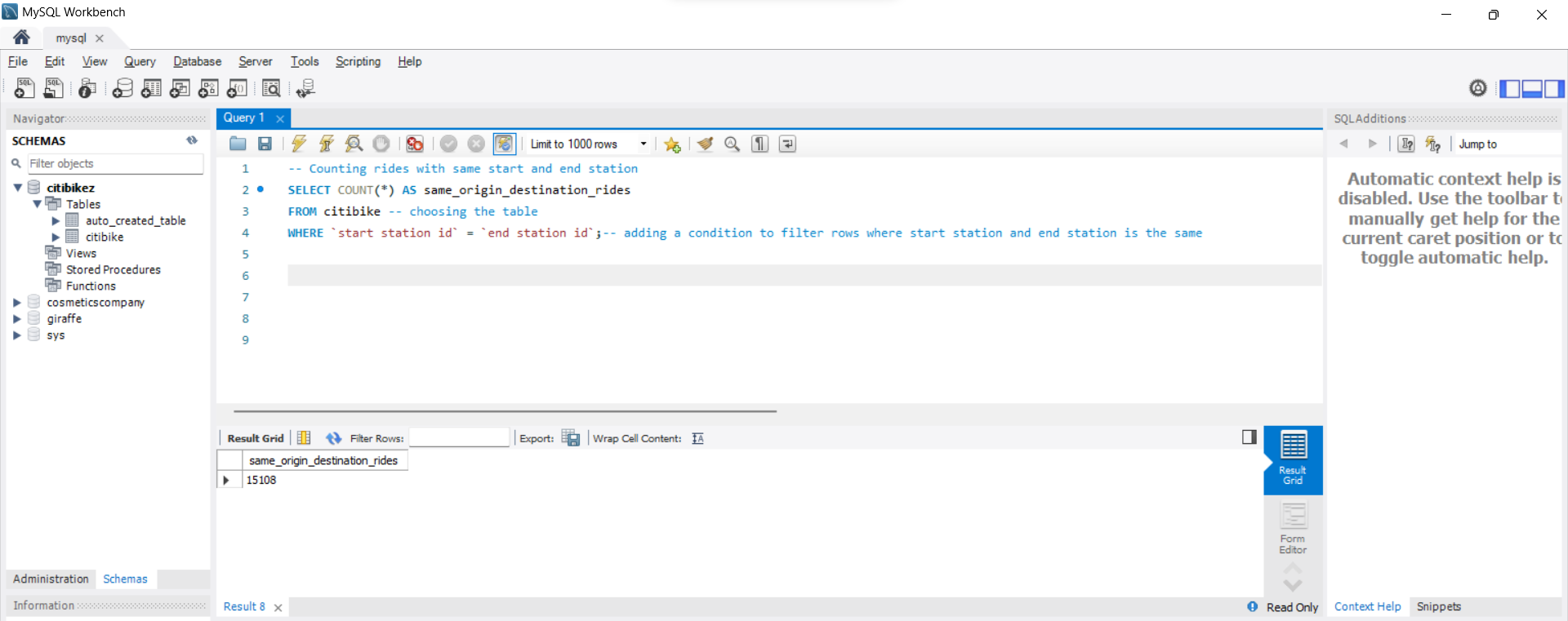
Query

-- I'm assuming the trip duration is given in seconds

SELECT AVG(tripduration)/60 AS average\_trip\_duration\_minutes -- selecting average of trip duration and converting it to minutes

FROM citibike; -- from citibike table

**ANSWER**

25.85 mins  
  
Question 4) How many rides had same origin and destination?  


**Query**

-- Counting rides with same start and end station

SELECT COUNT(\*) AS same\_origin\_destination\_rides

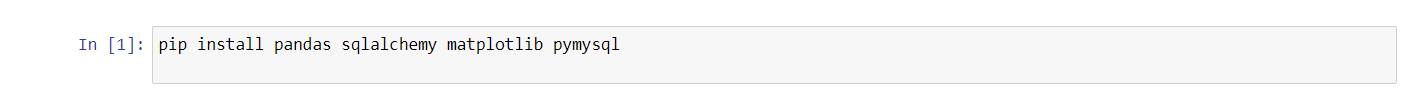
FROM citibike -- choosing the table

WHERE `start station id` = `end station id`;-- adding a condition to filter rows where start station and end station is the same

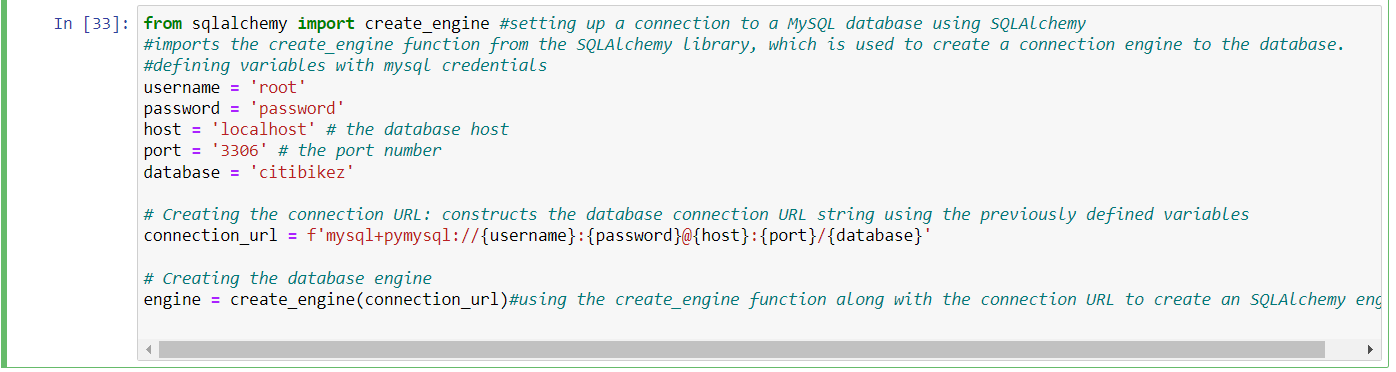
**ANSWER**

15108 rides had the same origin and destination.

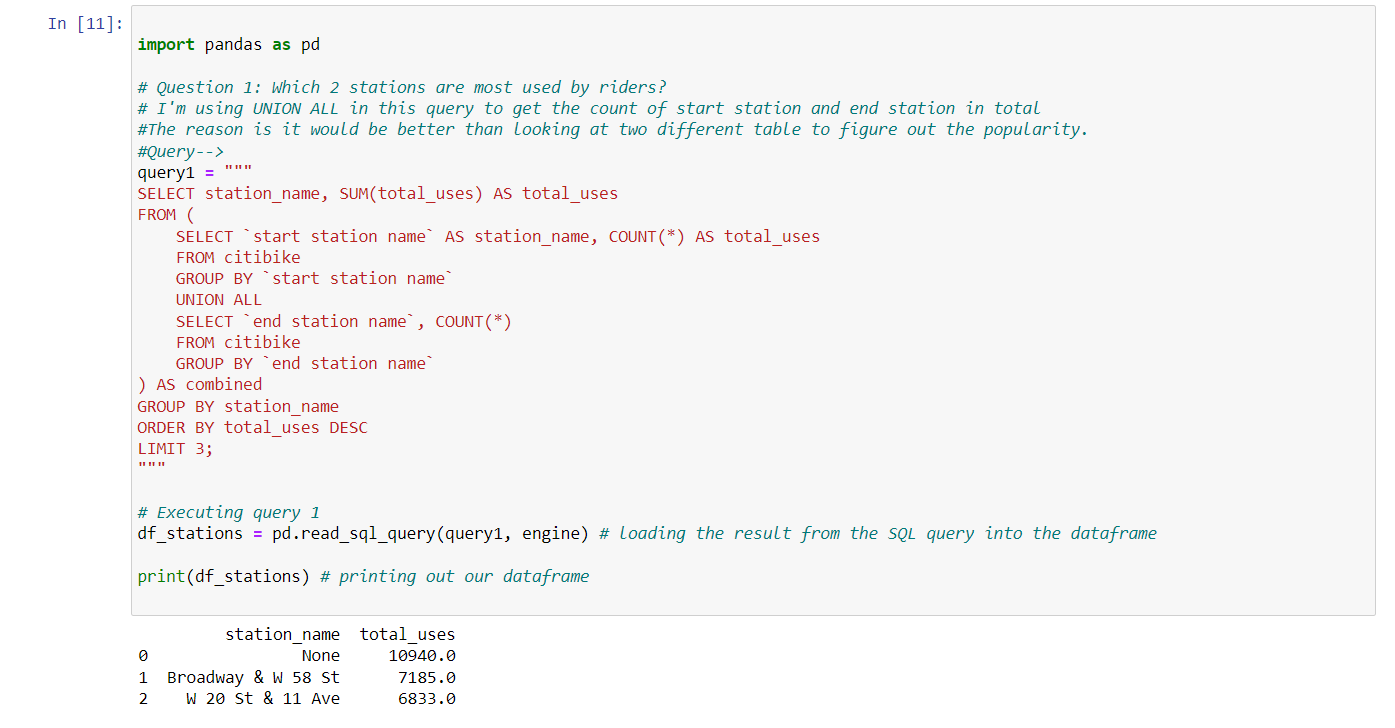
**ANSWERING THE SAME QUESTIONS IN PYTHON**

Importing the packages we need:  


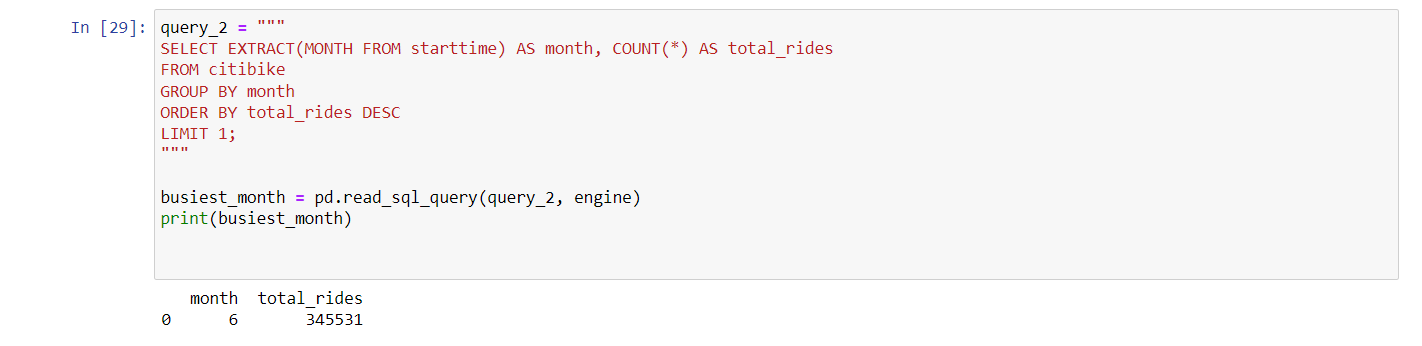
creating an "engine" that will serve as the foundation for communicating with the database.



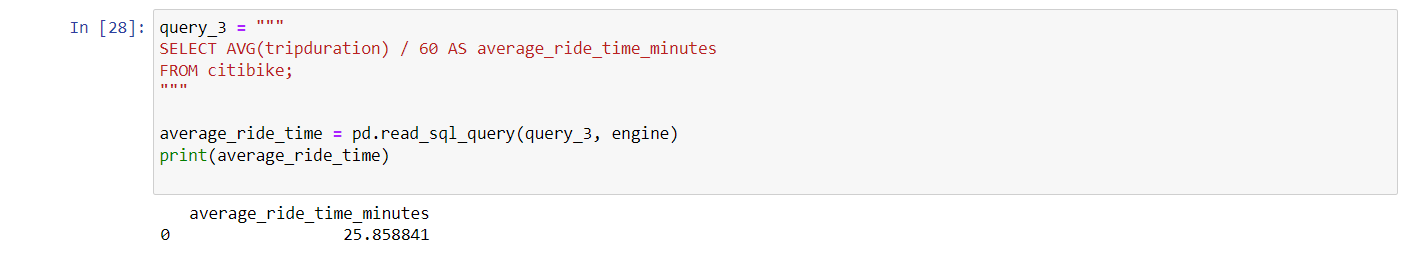
Question 1) Which 2 stations are most used by riders?

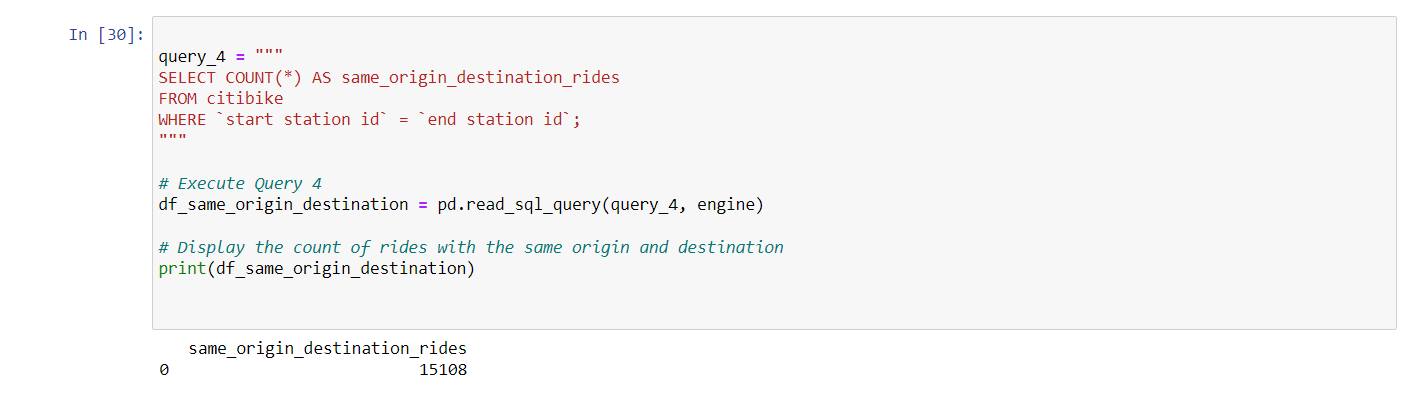


Question 2)Which month is the busiest time for Citi bike?



Question 3) What’s the average time of ride?



Question 4) How many rides had same origin and destination?   


**Pasting the entire Python code below:**

pip install pandas sqlalchemy matplotlib pymysql

from sqlalchemy import create\_engine #setting up a connection to a MySQL database using SQLAlchemy

#imports the create\_engine function from the SQLAlchemy library, which is used to create a connection engine to the database.

#defining variables with mysql credentials

username = 'root'

password = 'password'

host = 'localhost' # the database host

port = '3306' # the port number

database = 'citibikez'

# Creating the connection URL: constructs the database connection URL string using the previously defined variables

connection\_url = f'mysql+pymysql://{username}:{password}@{host}:{port}/{database}'

# Creating the database engine

engine = create\_engine(connection\_url)#using the create\_engine function along with the connection URL to create an SQLAlchemy engine

import pandas as pd

# Question 1: Which 2 stations are most used by riders?

# I'm using UNION ALL in this query to get the count of start station and end station in total

#The reason is it would be better than looking at two different table to figure out the popularity.

#Query-->

query1 = """

SELECT station\_name, SUM(total\_uses) AS total\_uses

FROM (

SELECT `start station name` AS station\_name, COUNT(\*) AS total\_uses

FROM citibike

GROUP BY `start station name`

UNION ALL

SELECT `end station name`, COUNT(\*)

FROM citibike

GROUP BY `end station name`

) AS combined

GROUP BY station\_name

ORDER BY total\_uses DESC

LIMIT 3;

"""

# Executing query 1

df\_stations = pd.read\_sql\_query(query1, engine) # loading the result from the SQL query into the dataframe

print(df\_stations) # printing out our dataframe

query\_2 = """

SELECT EXTRACT(MONTH FROM starttime) AS month, COUNT(\*) AS total\_rides

FROM citibike

GROUP BY month

ORDER BY total\_rides DESC

LIMIT 1;

"""

busiest\_month = pd.read\_sql\_query(query\_2, engine)

print(busiest\_month)

query\_3 = """

SELECT AVG(tripduration) / 60 AS average\_ride\_time\_minutes

FROM citibike;

"""

average\_ride\_time = pd.read\_sql\_query(query\_3, engine)

print(average\_ride\_time)

query\_4 = """

SELECT COUNT(\*) AS same\_origin\_destination\_rides

FROM citibike

WHERE `start station id` = `end station id`;

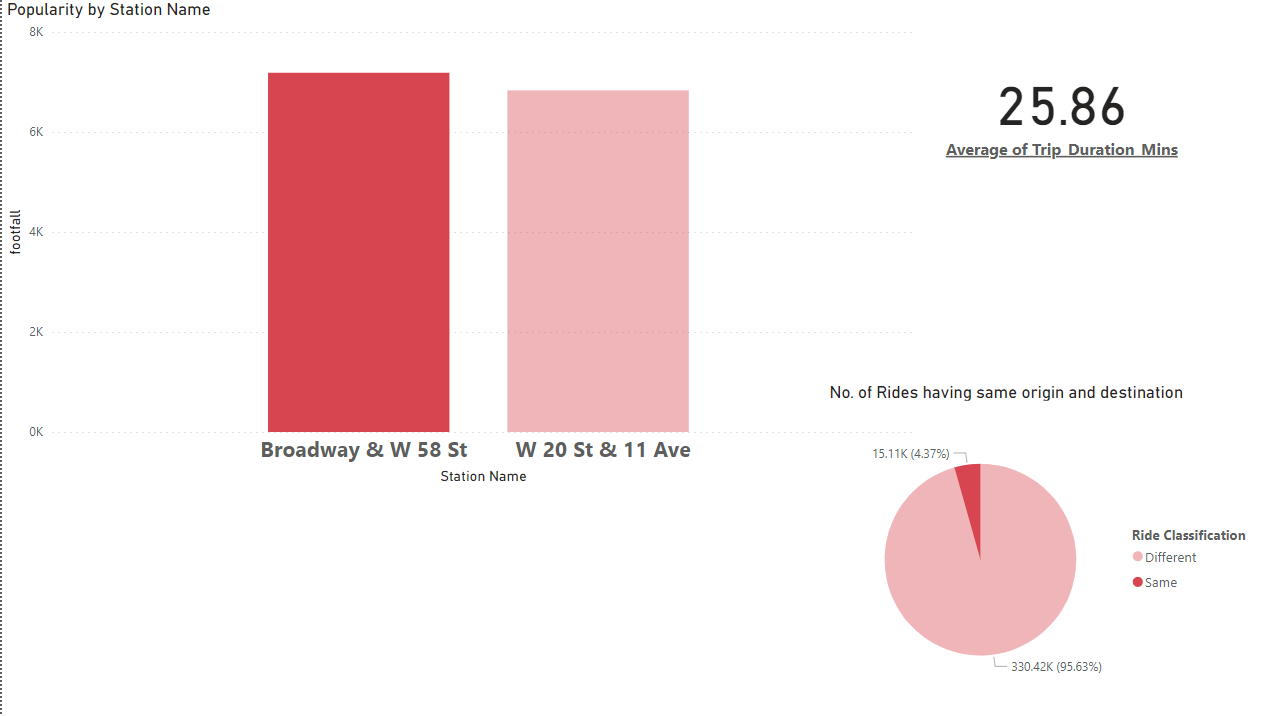
"""

# Execute Query 4

df\_same\_origin\_destination = pd.read\_sql\_query(query\_4, engine)

# Display the count of rides with the same origin and destination

print(df\_same\_origin\_destination)

**POWERBI ANALYSIS**