# Task 1: Get to Know Your Company

**Q1.** What are the total numbers of:

**Q1.1** Bike Stations

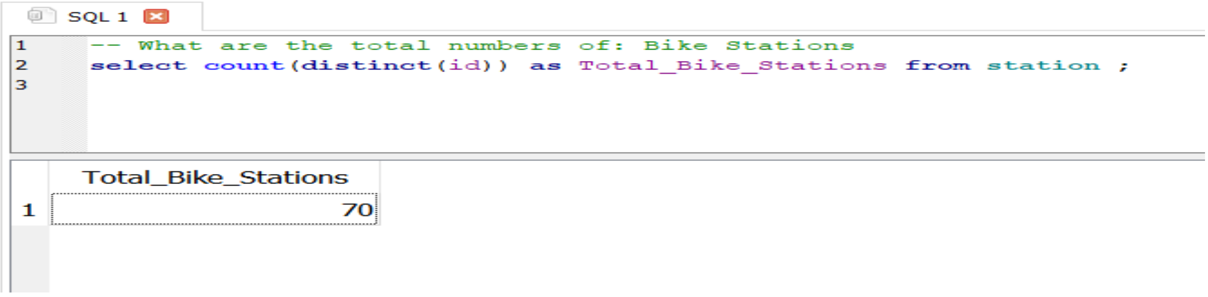
**SQl_Query --** select count(distinct(id)) as Total_Bike_Stations from station ;

**Q1.2** Bikes

**SQl_Query --** select count(distinct(bike_id)) as Total_Bikes from trip ;

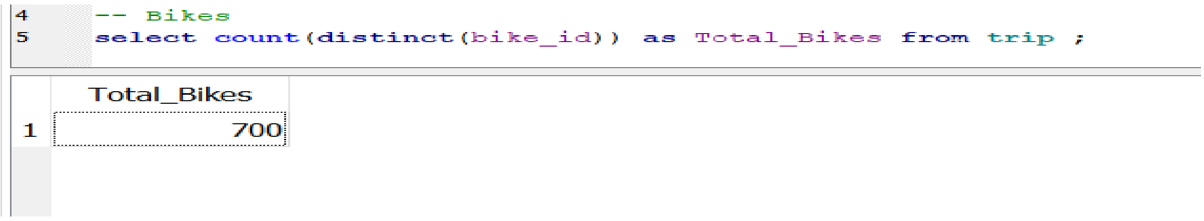
**Q1.3** Trips

**SQl_Query --** select count(distinct(id)) as Total_Trips from trip ;



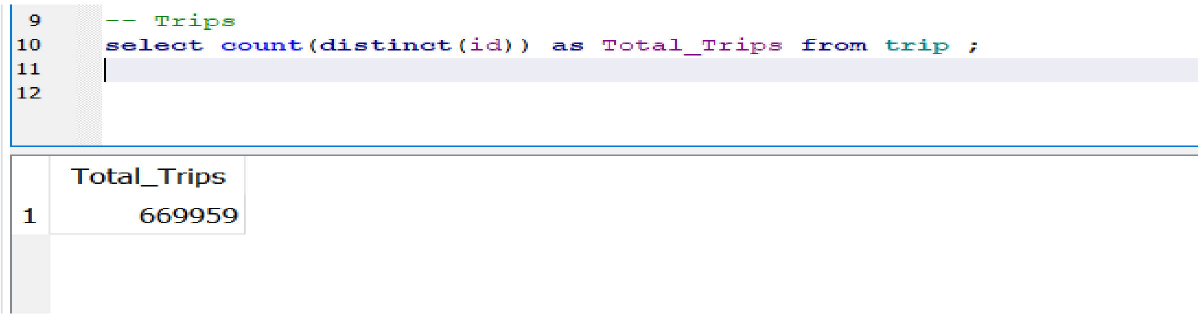
```sql
-- What are the total numbers of: Bike Stations
select count(distinct(id)) as Total_Bike_Stations from station ;
```

|   | Total_Bike_Stations |
|---|---|
| 1 | 70 |



```sql
-- Bikes
select count(distinct(bike_id)) as Total_Bikes from trip ;
```

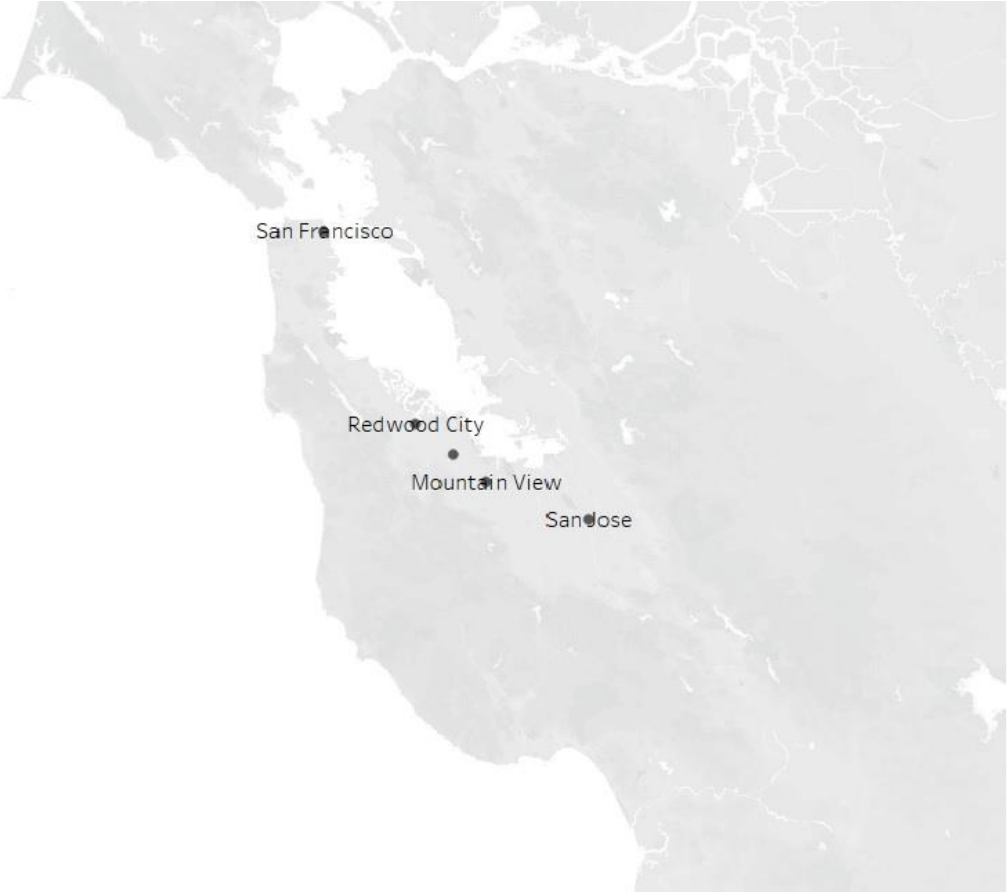
|   | Total_Bikes |
|---|---|
| 1 | 700 |



```sql
-- Trips
select count(distinct(id)) as Total_Trips from trip ;
```

|   | Total_Trips |
|---|---|
| 1 | 669959 |

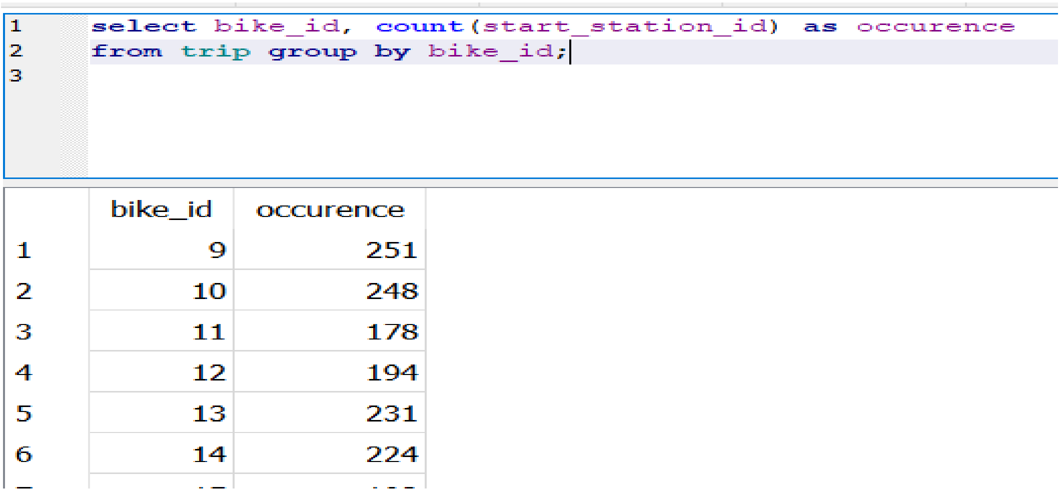
**Q 2. Construct a geographical plot to show the location of each bike station using the latitude and longitude provided under the Station table.**



**Q 3. What is the relationship between the following columns (one to one, many to one, many to many)?**

**Q. 3.1.** bike_id (Trip table) and start_station_id (Trip table)

- Many to many relation has been observed



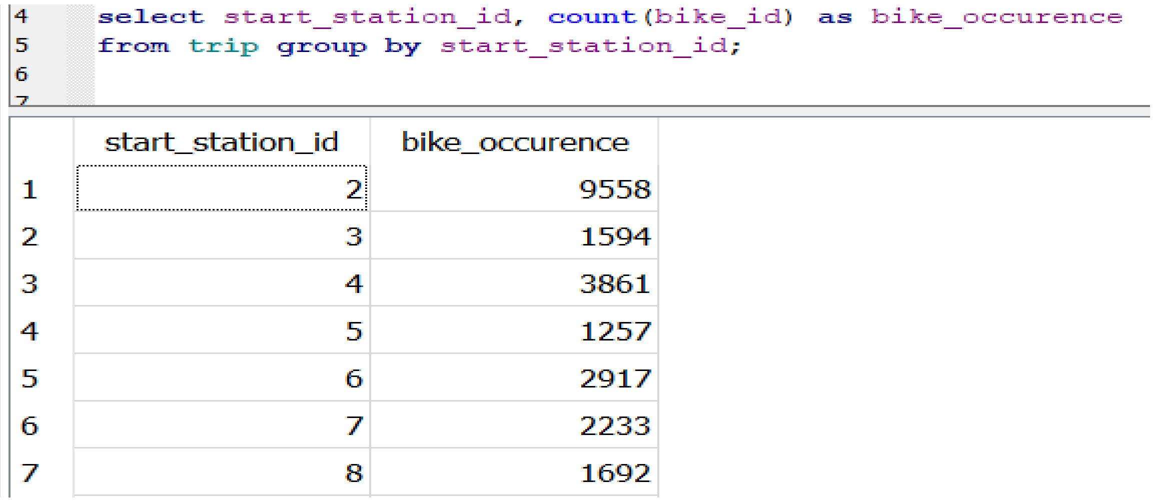
```
select bike_id, count(start_station_id) as occurence
from trip group by bike_id;
```

```
     bike_id   occurence
1          9         251
2         10         248
3         11         178
4         12         194
5         13         231
6         14         224
```

**Q 3.2.** pincode (Weather table) and station location (latitude and longitude in Station table)

- No relation has been found between pincode and station location

** No common key found between Weather and Station table

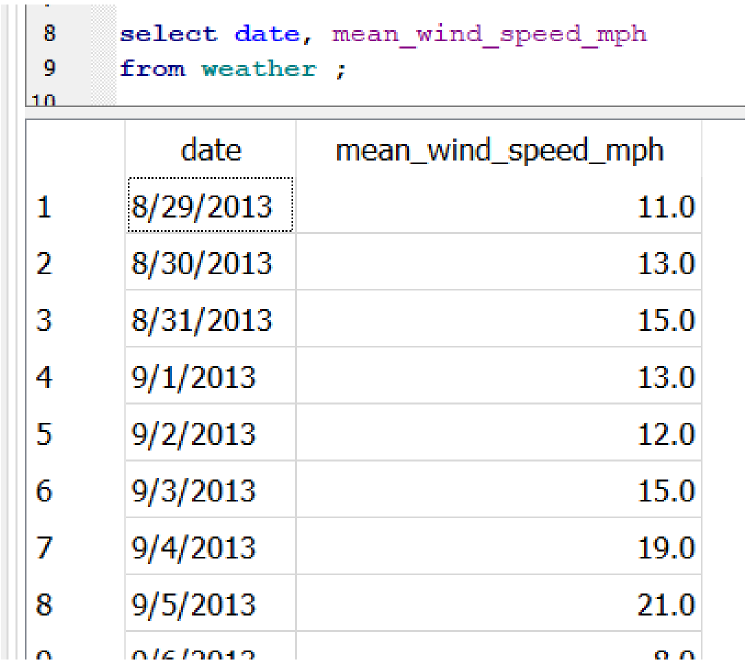


```
select start_station_id, count(bike_id) as bike_occurence
from trip group by start_station_id;
```

| | start_station_id | bike_occurence |
|---|---|---|
| 1 | 2 | 9558 |
| 2 | 3 | 1594 |
| 3 | 4 | 3861 |
| 4 | 5 | 1257 |
| 5 | 6 | 2917 |
| 6 | 7 | 2233 |
| 7 | 8 | 1692 |

**Q 3.3** 8/29/2013 (date column in Weather table) and mean wind speed (Weather table)

- One to many relation has been observed



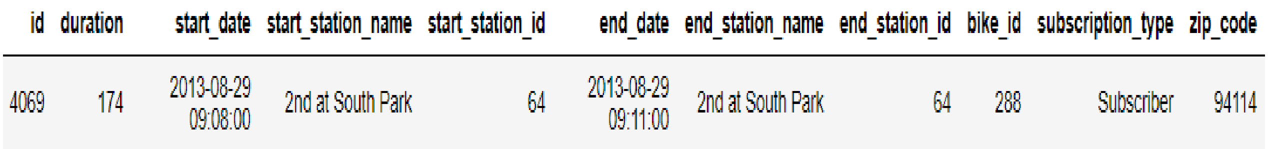
```
select date, mean_wind_speed_mph
from weather ;
```

| | date | mean_wind_speed_mph |
|---|---|---|
| 1 | 8/29/2013 | 11.0 |
| 2 | 8/30/2013 | 13.0 |
| 3 | 8/31/2013 | 15.0 |
| 4 | 9/1/2013 | 13.0 |
| 5 | 9/2/2013 | 12.0 |
| 6 | 9/3/2013 | 15.0 |
| 7 | 9/4/2013 | 19.0 |
| 8 | 9/5/2013 | 21.0 |

## Q 4. Find the first and the last trip in the data.

- First trip

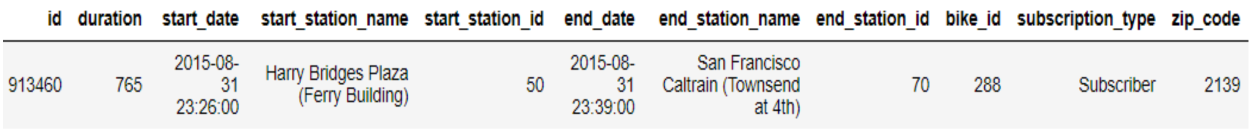
**SQL_Query:** select * from trip where start_date = (select min(start_date) from trip) ;



| id | duration | start_date | start_station_name | start_station_id | end_date | end_station_name | end_station_id | bike_id | subscription_type | zip_code |
|---|---|---|---|---|---|---|---|---|---|---|
| 4069 | 174 | 2013-08-29 09:08:00 | 2nd at South Park | 64 | 2013-08-29 09:11:00 | 2nd at South Park | 64 | 288 | Subscriber | 94114 |

- Last trip

**SQL_Query:** select * from trip where end_date = (select max(end_date) from trip) ;

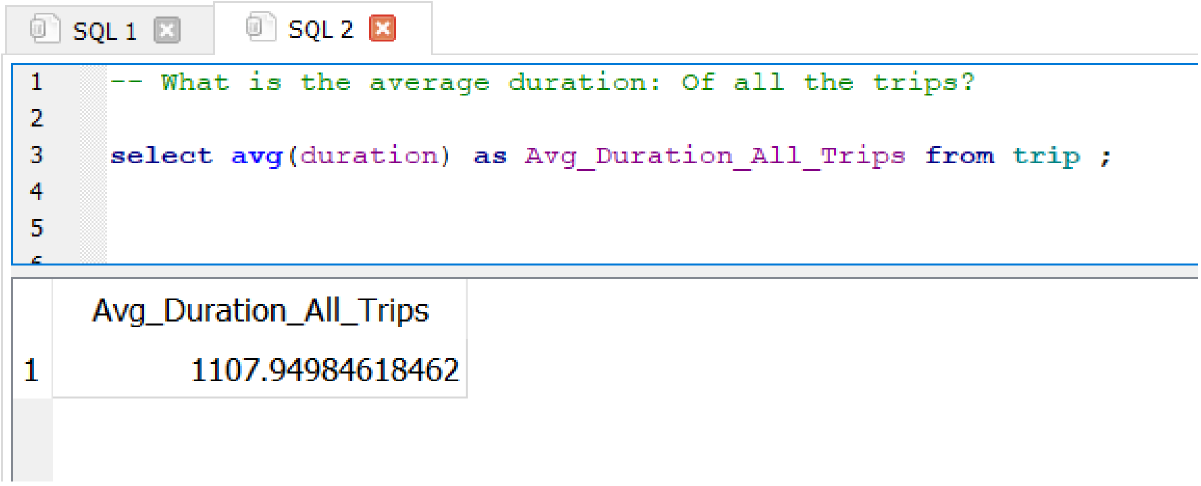


| id | duration | start_date | start_station_name | start_station_id | end_date | end_station_name | end_station_id | bike_id | subscription_type | zip_code |
|---|---|---|---|---|---|---|---|---|---|---|
| 913460 | 765 | 2015-08-31 23:26:00 | Harry Bridges Plaza (Ferry Building) | 50 | 2015-08-31 23:39:00 | San Francisco Caltrain (Townsend at 4th) | 70 | 288 | Subscriber | 2139 |

## Q 5. What is the average duration

### Q. 5.1 Of all the trips?

**SQL_Query:** select avg(duration) as Avg_Duration_All_Trips from trip ;



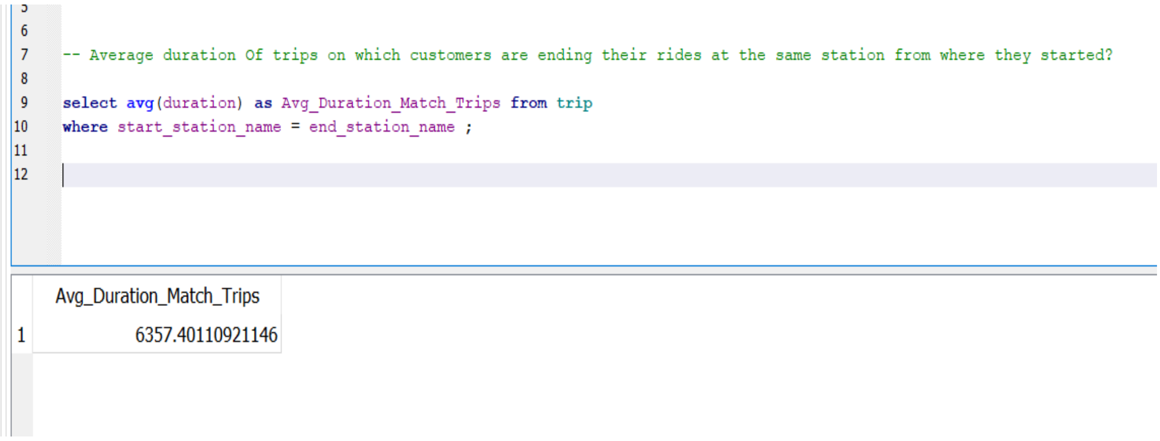
```
-- What is the average duration: Of all the trips?

select avg(duration) as Avg_Duration_All_Trips from trip ;
```

| | Avg_Duration_All_Trips |
|---|---|
| 1 | 1107.94984618462 |

### Q. 5.2 Average duration Of trips on which customers are ending their rides at the same station from where they started?

**SQL_Query:** select avg(duration) as Avg_Duration_Match_Trips from trip

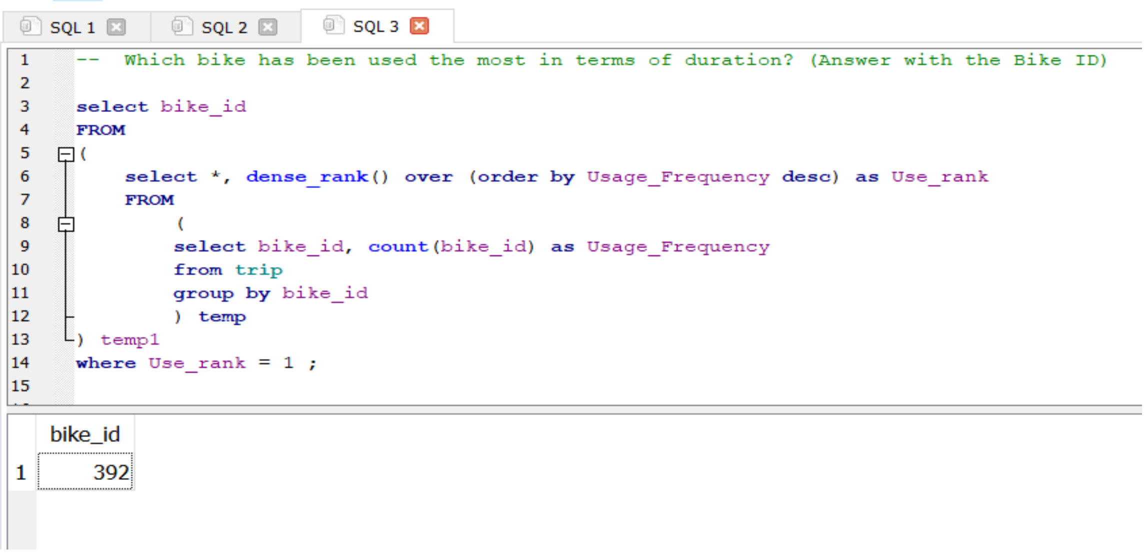
where start_station_name = end_station_name ;



```
-- Average duration Of trips on which customers are ending their rides at the same station from where they started?

select avg(duration) as Avg_Duration_Match_Trips from trip
where start_station_name = end_station_name ;
```

| | Avg_Duration_Match_Trips |
|---|---|
| 1 | 6357.40110921146 |

## Q 6. Which bike has been used the most in terms of duration? (Answer with the Bike ID)



```
-- Which bike has been used the most in terms of duration? (Answer with the Bike ID)

select bike_id
FROM
(
    select *, dense_rank() over (order by Usage_Frequency desc) as Use_rank
    FROM
        (
        select bike_id, count(bike_id) as Usage_Frequency
        from trip
        group by bike_id
        ) temp
) temp1
where Use_rank = 1 ;
```

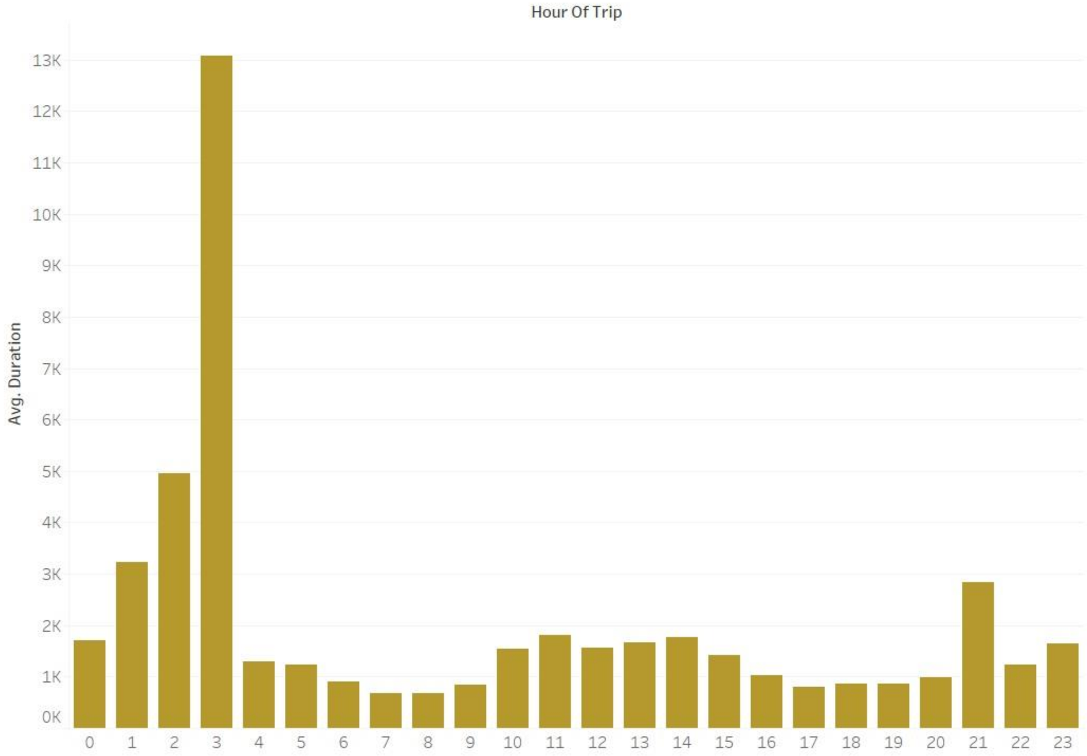
| | bike_id |
|---|---|
| 1 | 392 |

### SQL_Query:

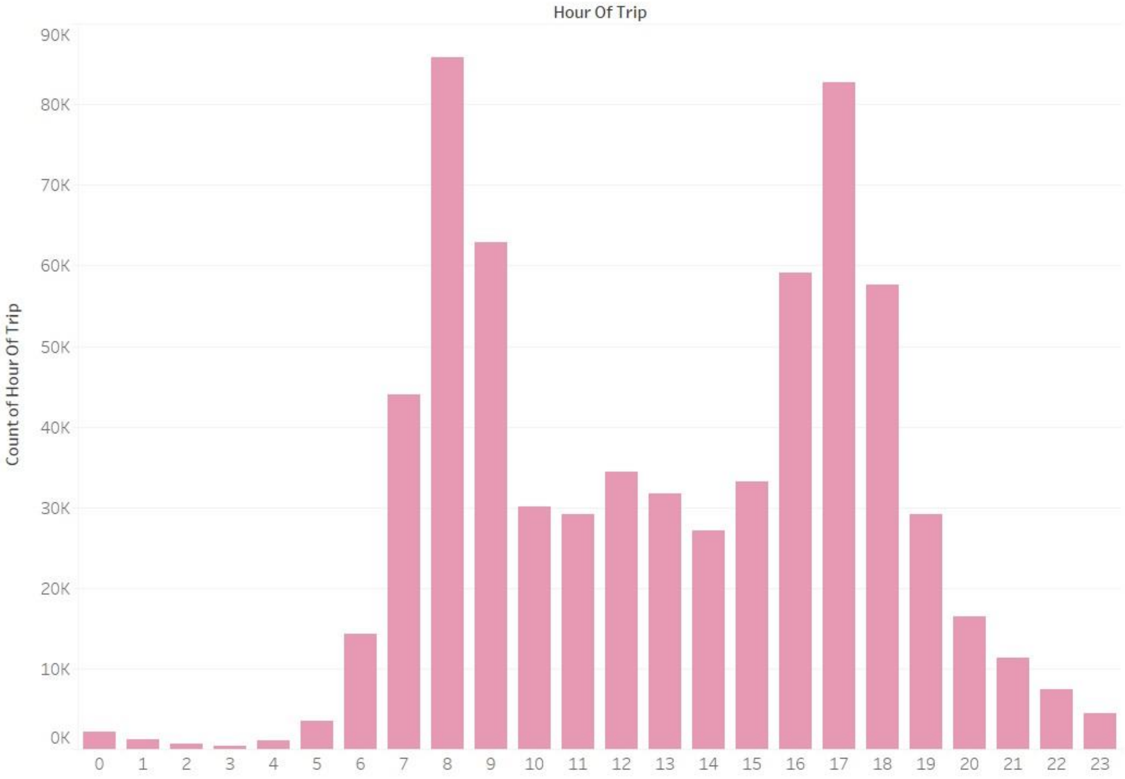
```
select bike_id

FROM

(          select *, dense_rank() over (order by Usage_Frequency desc) as Use_rank

           FROM

                      (

                      select bike_id, count(bike_id) as Usage_Frequency

                      from trip

                      group by bike_id

                      ) temp

) temp1

where Use_rank = 1 ;
```

**Q. 7** Plot the most suitable graph for the followings:

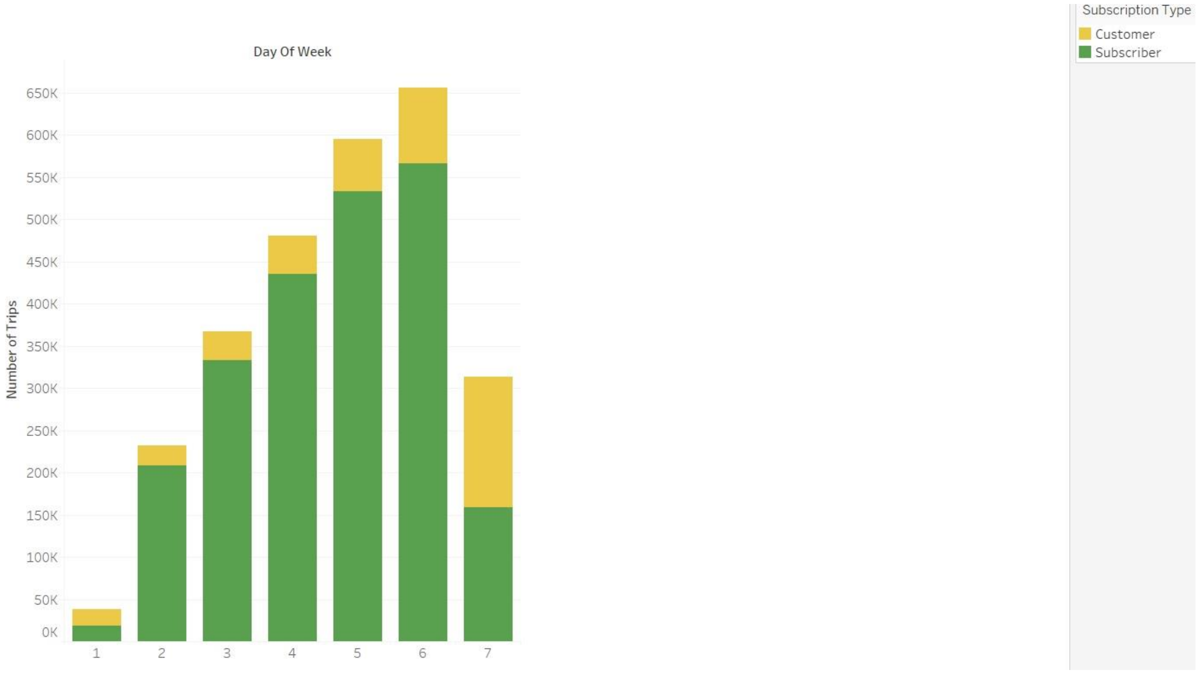
**Q. 7.1** Average duration of a trip versus Number of trips



**Q. 7.2** Hour of start time versus No. of trips.

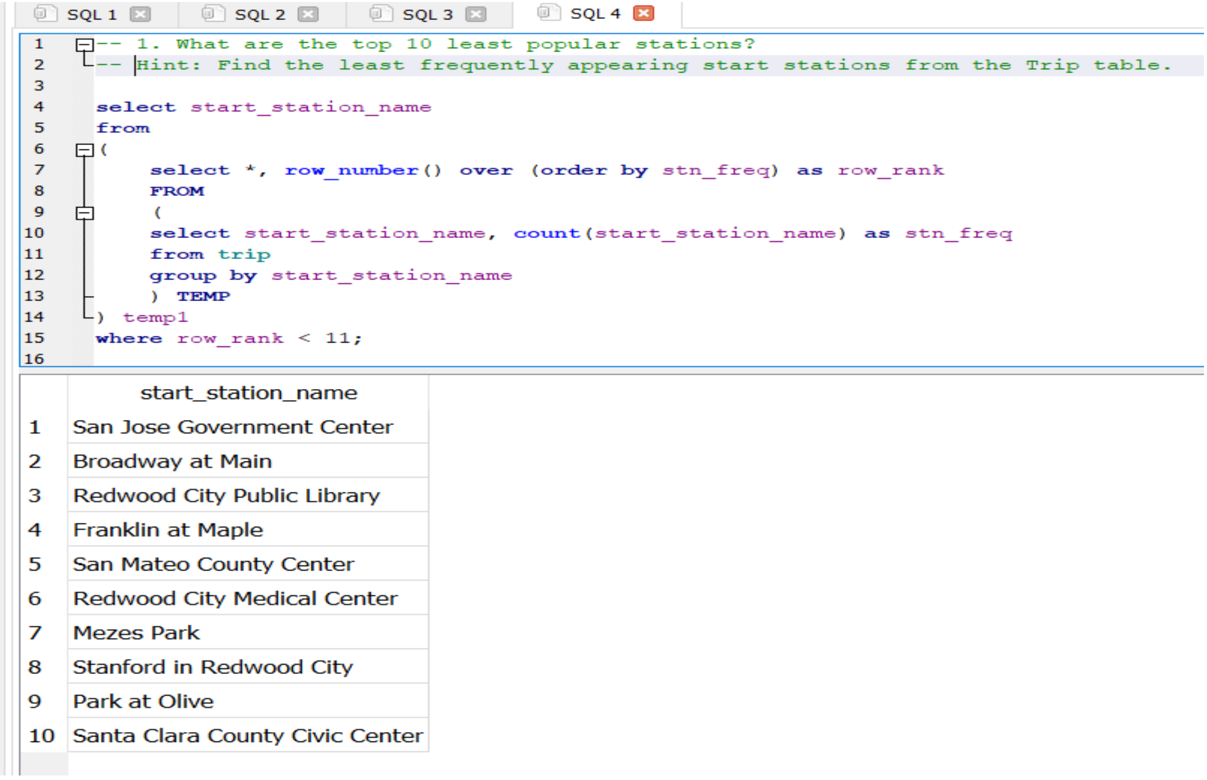


Q. 7.3 Day of the week versus No. of trips also denote subscribers and customers with different colours.



## Task2 (Demand Prediction)

**Q 1 What are the top 10 least popular stations? Hint: Find the least frequently appearing start stations from the Trip table.**



```
-- 1. What are the top 10 least popular stations?
-- Hint: Find the least frequently appearing start stations from the Trip table.

select start_station_name
from
(
     select *, row_number() over (order by stn_freq) as row_rank
     FROM
     (
     select start_station_name, count(start_station_name) as stn_freq
     from trip
     group by start_station_name
     ) TEMP
) temp1
where row_rank < 11;

```

| | start_station_name |
|---|---|
| 1 | San Jose Government Center |
| 2 | Broadway at Main |
| 3 | Redwood City Public Library |
| 4 | Franklin at Maple |
| 5 | San Mateo County Center |
| 6 | Redwood City Medical Center |
| 7 | Mezes Park |
| 8 | Stanford in Redwood City |
| 9 | Park at Olive |
| 10 | Santa Clara County Civic Center |

**SQL_Query:**

```
select start_station_name
from
(
        select *, row_number() over (order by stn_freq) as row_rank
        FROM
        (
        select start_station_name, count(start_station_name) as stn_freq
        from trip
        group by start_station_name
        ) TEMP
) temp1
where row_rank < 11;
```

**Q 2. -- Idle time for station 2**

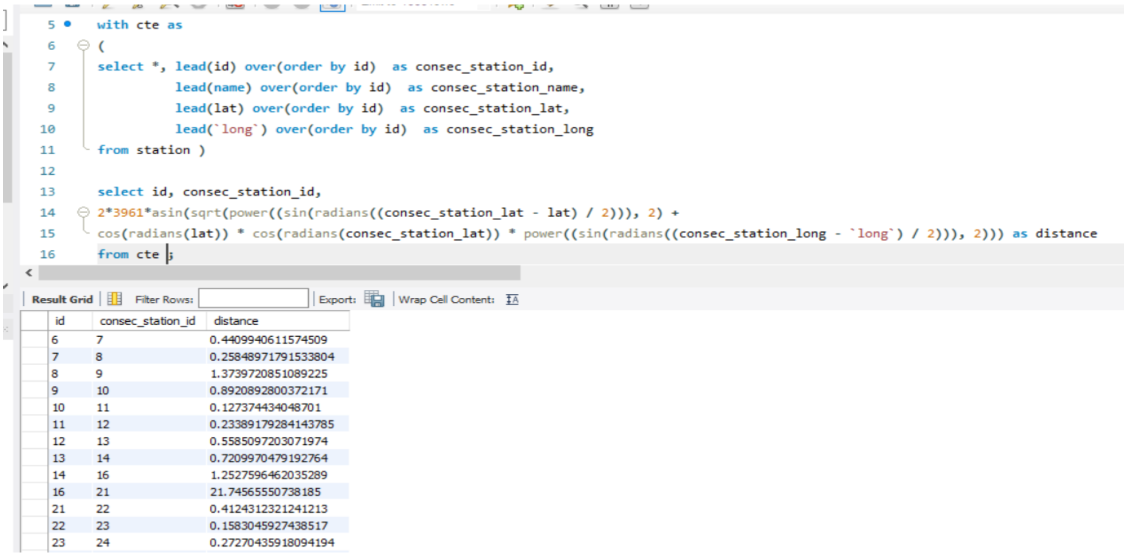
-- Idle time is the duration for which a station remains inactive.

-- You can consider this as the time for which a station has more than 3 bikes available.

**SQL_Query:**

```
select count(station_id) from status
where station_id = 2 & bikes_available > 3 ;
```

**Q2.3. Find distance between 2 consecutive stations**



```
with cte as
(
select *, lead(id) over(order by id)  as consec_station_id,
          lead(name) over(order by id)  as consec_station_name,
          lead(lat) over(order by id)  as consec_station_lat,
          lead(`long`) over(order by id)  as consec_station_long
from station )

select id, consec_station_id,
2*3961*asin(sqrt(power((sin(radians((consec_station_lat - lat) / 2))), 2) +
cos(radians(lat)) * cos(radians(consec_station_lat)) * power((sin(radians((consec_station_long - `long`) / 2))), 2))) as distance
from cte ;
```

| id | consec_station_id | distance |
|---|---|---|
| 6 | 7 | 0.4409940611574509 |
| 7 | 8 | 0.25848971791533804 |
| 8 | 9 | 1.3739720851089225 |
| 9 | 10 | 0.8920892800372171 |
| 10 | 11 | 0.127374434048701 |
| 11 | 12 | 0.23389179284143785 |
| 12 | 13 | 0.5585097203071974 |
| 13 | 14 | 0.7209970479192764 |
| 14 | 16 | 1.2527596462035289 |
| 16 | 21 | 21.74565550738185 |
| 21 | 22 | 0.4124312321241213 |
| 22 | 23 | 0.1583045927438517 |
| 23 | 24 | 0.27270435918094194 |

**SQL_Query:**

```
with cte as

(

select *, lead(id) over(order by id)  as consec_station_id,

                         lead(name) over(order by id)  as consec_station_name,

                         lead(lat) over(order by id)  as consec_station_lat,

                         lead(`long`) over(order by id)  as consec_station_long

from station )

select id, consec_station_id,

2*3961*asin(sqrt(power((sin(radians((consec_station_lat - lat) / 2))), 2) +

cos(radians(lat)) * cos(radians(consec_station_lat)) *
power((sin(radians((consec_station_long - `long`) / 2))), 2))) as distance

from cte ;
```

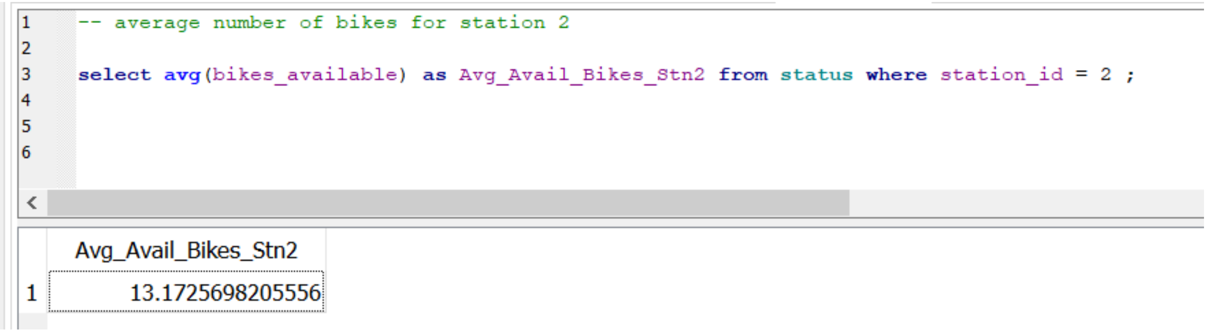
## Task 3: Optimizing Operations

**Q.1. Calculate the average number of bikes and docks available for Station 2 and Station 3 (Hint: Use the Status table.)**

-- average number of bikes for station 2

**SQL_Query:**

```
select avg(bikes_available) as Avg_Avail_Bikes_Stn2 from status where station_id = 2 ;
```



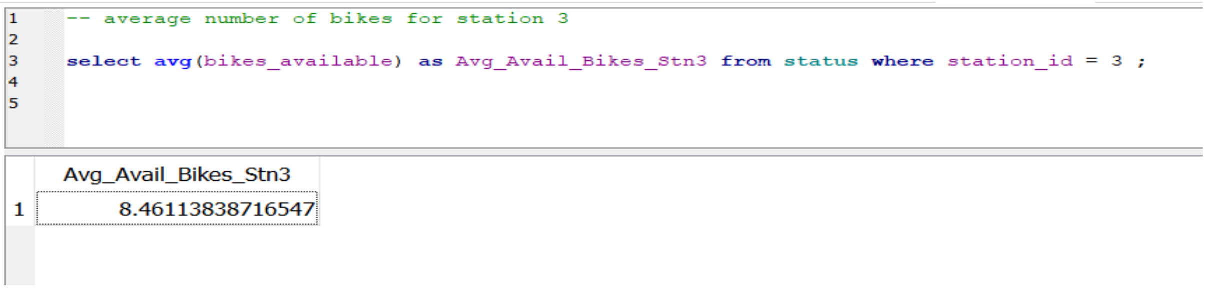
```
-- average number of bikes for station 2

select avg(bikes_available) as Avg_Avail_Bikes_Stn2 from status where station_id = 2 ;
```

```
     Avg_Avail_Bikes_Stn2
1        13.1725698205556
```

-- average number of bikes for station 3

**SQL_Query:**

```
select avg(bikes_available) as Avg_Avail_Bikes_Stn3 from status where station_id = 3 ;
```



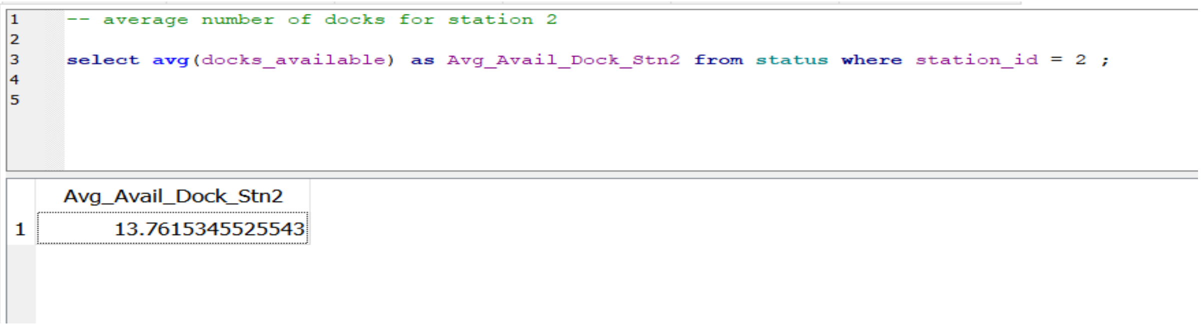
```sql
-- average number of bikes for station 3

select avg(bikes_available) as Avg_Avail_Bikes_Stn3 from status where station_id = 3 ;
```

| | Avg_Avail_Bikes_Stn3 |
|---|---|
| 1 | 8.46113838716547 |

-- average number of docks for station 2

**SQL_Query:**

select avg(docks_available) as Avg_Avail_Dock_Stn2 from status where station_id = 2 ;



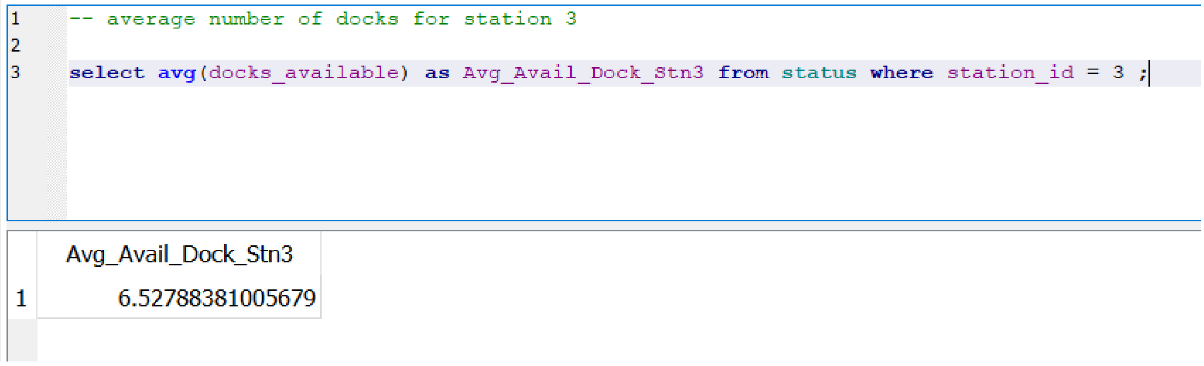
```sql
-- average number of docks for station 2

select avg(docks_available) as Avg_Avail_Dock_Stn2 from status where station_id = 2 ;
```

| | Avg_Avail_Dock_Stn2 |
|---|---|
| 1 | 13.7615345525543 |

-- average number of docks for station 3

**SQL_Query:**

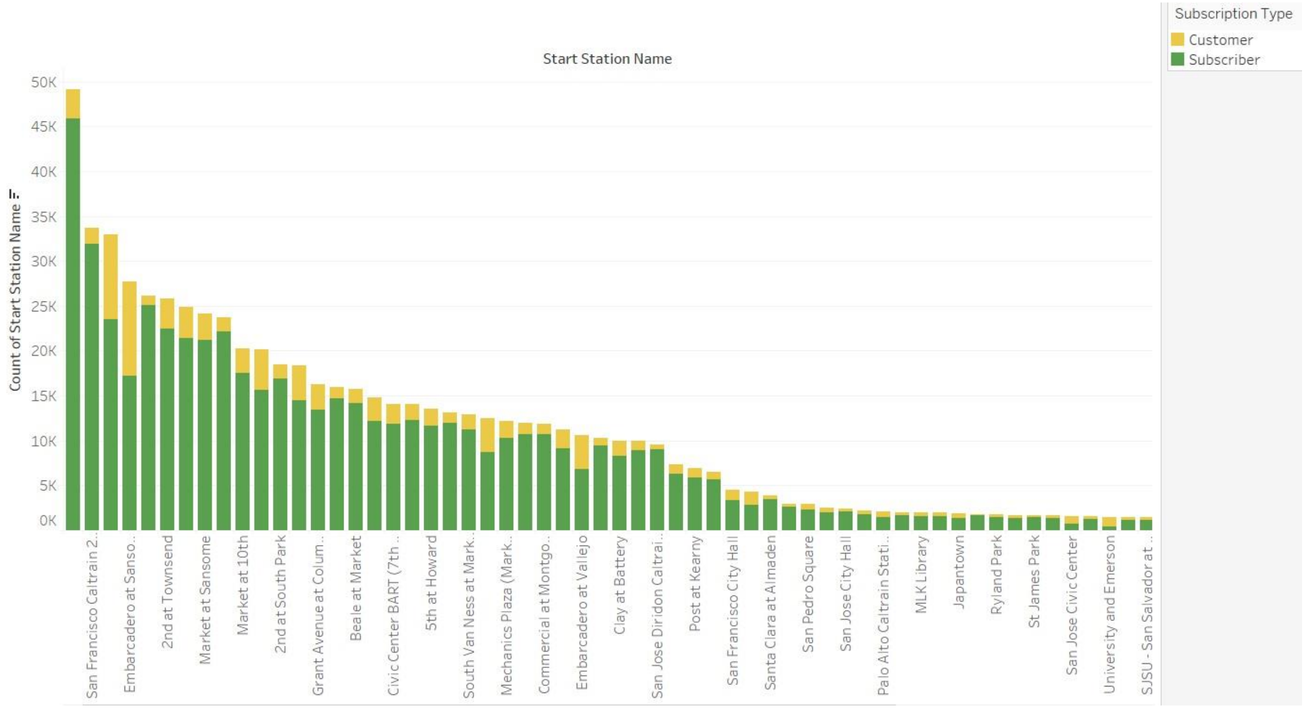
select avg(docks_available) as Avg_Avail_Dock_Stn3 from status where station_id = 3 ;



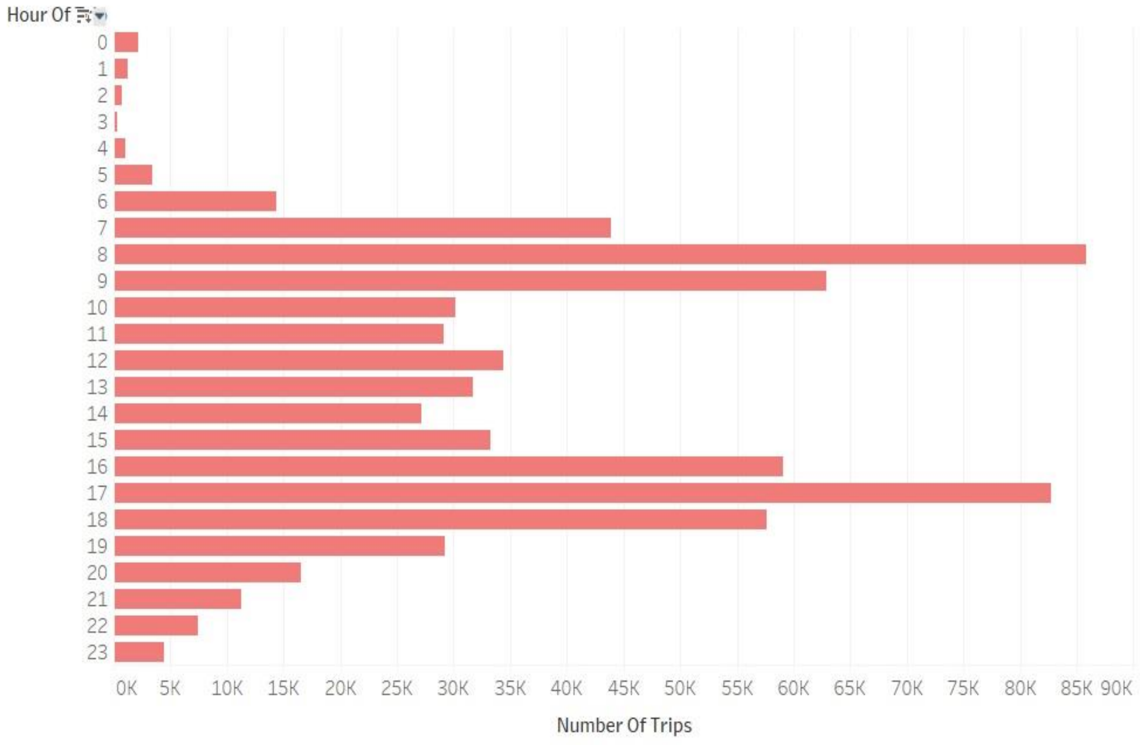
```sql
-- average number of docks for station 3

select avg(docks_available) as Avg_Avail_Dock_Stn3 from status where station_id = 3 ;
```

| | Avg_Avail_Dock_Stn3 |
|---|---|
| 1 | 6.52788381005679 |

**Q 2.** Plot the Popularity of each station on a map for subscribers and customers.



**Q.3** Plot the number of trips per hour for all the data provided in the Trip table.



# Task 4

**Q.1** Zulip has decided to start a new product line called Couple Bikes. So for that what are some factors you will have to consider while validating the idea of couple bikes?

**Solution:**

a) Company should start the Couple Bikes between the stations where no. of users are ending their rides at the same station from where they have started.
b) Should consider the Weather conditions, where weather is good.
c) Popularity of the station
d) Availability of bikes and docks for a particular station