Our dataset is about Bay Area Bike Share data. The Bay Area Bike Share enables quick, easy, and affordable bike trips around the San Francisco Bay Area. They make data releases, containing information about stations located in this area, bikes and docks available and trip details by customers (who are not part of any plan) and subscribers (who are part of subscription plans).
The goal of this project is to create a database and demonstrate proficiency in extracting data from database files using SQL and the ability to analyze these values in context. This database contains three data tables: station, status, trip.
The Station table contains data about bike station such as id, name, city, dock count etc.
The Trip table contains data about individual bike trips such as start and end time, station, duration etc.
The Status table contains data about the number of bikes and docks available for a given station at a given time.

Business Understanding:

Bike share is a bike rentals corporate company that has gathered the bike share data in the bay area for the purpose of verifying the statistics, our main aim is to study the data and come up with meaningful insights from it to run the company smoothly.
From the gathered data we can mainly identify the most and least preferred bike stations, bike models and routes. We want to determine how different variables can affect the number of bikes rented, along with their duration. Using which we can optimize our resources by allocating more bikes in the most preferred areas, similarly by identifying the most preferred bike models we can optimize the ratio of different bikes present in a bike station.

We are studying this data to gain insightful information about our business and to verify if we're on track with our business goals or if not to take some measures related to the growth of the company.

Business Goals:

- To identify the stations that have been preferred the greatest number of times by customers.
- To identify the stations that have been preferred least number of times by customers.
- Identifying the numbers of subscribers and customers using the service.
- Identifying the most traveled routes.
- The identify the category of customers who took longer trips.
- Trips covered within the cities and between the cities.
- Trips by customers and subscribers over the week.
- Busiest days of the week.

  By identifying the above details, we can propose strategies for the company regarding the bike counts, different offers to customers to convert into subscribers, and increase facilities in the cities at required stations.

Data Understanding:

♦ What information each column of the data contains, data types of each column, values, scale and range of data?

- Please find below the details of the data we are using for the analysis.

	Column Name	Data Type	Information provided	Values
Station	id	int	Unique ID for each station	70 unique values
	name	varchar	Station Name	70 unique names
	latitude	double	Latitude of station location	In degrees from 37.3 to 37.8
	longitude	double	Longitude of station location	In degrees around -122
	dock_count	int	Number of bikes the station can hold	Values ranging from 11 to 27
	city	varchar	City Name	Five Cities
	installation_date	date	Date of station installation	Dates ranging from 2013 & 2014
Status	status_id	int	Unique ID for each status record	Incremental value
	station_id	int	Unique ID for each station	Unique station ID at different times
	bikes_available	int	Number of bikes available at the provided time	Ranging from 0 to 27
	docks_available	int	Number of docks available in the station at the provided time	Ranging from 0 to 27
	time	datetime	Date and Time	Dates ranging from Aug 2013 to Aug 2015
Trip	id	int	Unique ID for each trip	Unique Values
	duration	int	Trip duration in seconds	Time in seconds
	start_date	datetime	Start date and time of Trip	Date and time
	start_station_id	int	Starting station id of Trip	Id's from Station table
	end_date	datetime	End date and time of Trip	Date and Time
	end_station_id	int	Ending station id of Trip	Id's from station table
	bike_id	int	Unique ID for each bike	Unique Values
	subscription_type	varchar	Subscription Type	Subscriber or Customer

♦ Verify the data quality

- In Station table, column name id and name needed to be changed to station_id & station_name respectively. (**Query**: alter table station rename column id to station_id; alter table station rename column name to station_name)

- In table trip, column name id needed to be changed to trip_id. (**Query**: alter table trip rename column id to trip_id)
- In table status, there is no unique value column. So, we have added an additional column with incremental value as primary key for the table. (**Query**: alter table status add status_id int unsigned not null auto_increment, add primary key (status_id))
- In the given data, there are no missing values.

♦ Provide simple statistics of the data and describe what these values mean if you found something interesting

Below are few statistics for numeric attributes

column_data	maximum	minimum	Mean	Std Dev
Docks count	27	11	17.66	3.98
Bikes Available	27	0	8.39	3.99
Docks available	27	0	9.28	4.18
Duration (in mins)	287840	1	18.47	370.92

Below are few of the inferences that can be made from studying the given data.

In the status table, The City column has five values. It contains cities such as San Jose, Redwood City, Mountain View, Palo Alto and San Francisco. 50% of the stations are in San Francisco, 23% of the stations are in San Jose and remaining covers the rest of the cities. (**Query**: Select city, count (*) from station group by city)

The maximum and minimum dock counts for the given station are 27 and 11, respectively. Almost 50% of stations have 15 docks. (**Queries**: Select max(dock_count), min(dock_count) from station; select count (*), dock_count from station group by dock_count order by count (*) desc)

Most of the stations were installed in August 2013. (**Query**: Select count (*), installation_date, city from station group by installation_date order by installation_date)

85% of trips are done by subscribers. (**Query**: Select subscription type, count (*) from trip group by subscription_type)

The average number of bikes and docks available at a station is 8 and 9, respectively.

From all the trips, minimum duration is one-minute, maximum duration is 287840 minutes, and the average duration of trips is 18 minutes. (**Query**: select min(duration/60), max(duration/60), avg(duration/60) from trip)

Database Design:

♦ Schema Design

- Find entities, their attributes, their primary keys, and relationships between them

  We have three entities in our schema, Station, Status and Trip. Please find below their respective attributes.

Station						
station_id	station_name	latitude	longitude	dock_count	city	installation_date

station_id is the primary key of this table.

Status				
status_id	station_id	bikes_available	docks_available	time

status_id is the primary key of this table.

Trip							
trip_id	duration	start_ date	start_station _id	end_ date	end_station _id	bike_id	subscription _type

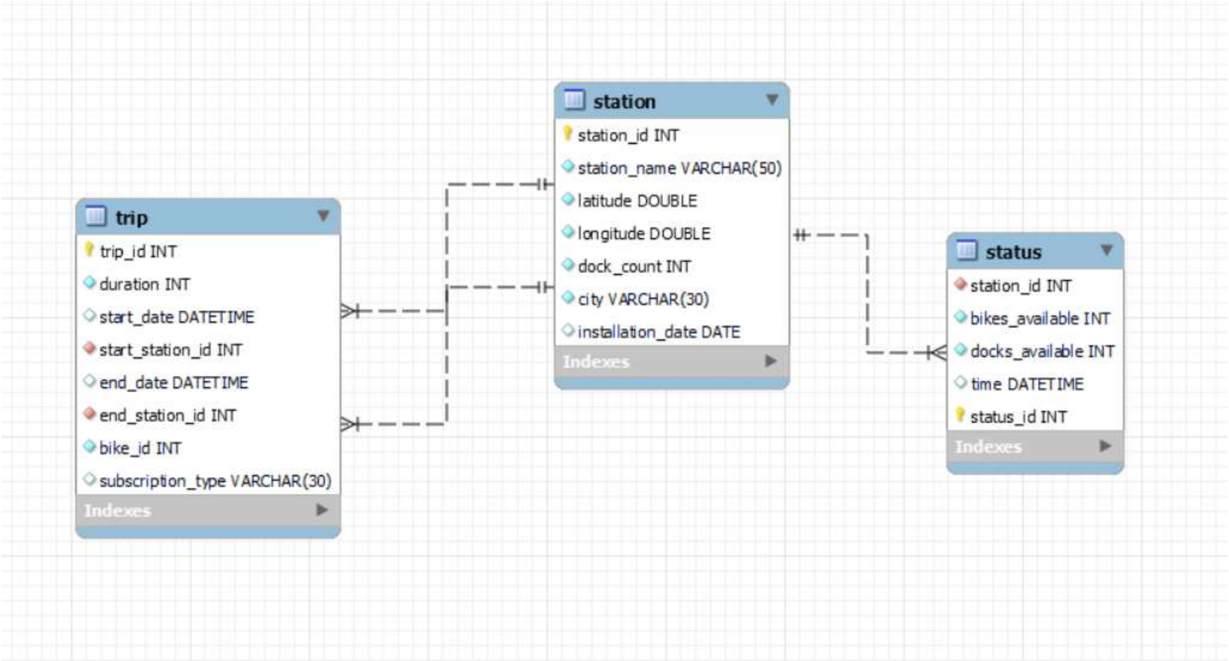
trip_id is the primary key of this table.

Trip table contains start and end station id from Station table.
Status table contains availability of bikes and docks at a station from Station table.
Station table has unique id and name for each station.

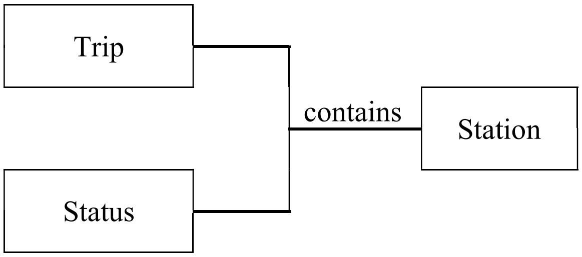
- Model all the constraints you believe should be there in your schema

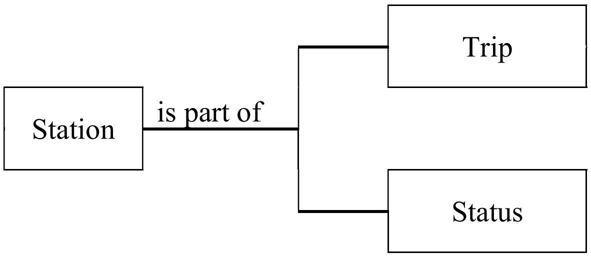
  Trip can have more than one station id as part of start and end station id
  Status can have more than one station id at different time
  Station provides unique id, name and details of each station

- Draw and ER diagram of your dataset



- Translate your ER diagram into relations





♦ Schema Normalization

- Find all the functional dependencies you can from your schema

{station_id} -> {station_name, latitude, longitude, dock_count, city, installation_date} {station_id, station_name} -> {latitude, longitude, dock_count, city, installation_date}

{station_id, longitude, latitude} -> { station_name, dock_count, city, installation_date}

{status_id} -> {time, station_id, bikes_available, docks_available}

{trip_id} -> {start_date, end_date, start_station_id, end_station_id, duration, bike_id, subscription_type}

- Check if the keys you have chosen for your relations are minimal

  For Status table, set of all attributes A = {status_id, time, station_id, bikes_available, docks_available} and functional dependency F = {status_id} -> {time, station_id, bikes_available, docks_available} and X = status_id

  Let's find closure of X. Initialize X+ as X.

  X+ = {status_id}

  Using F, X+ = {status_id, time, station_id, bikes_available, docks_available}

  No more attributes can be added to X+

  So {status_id} is the key.

  Similarly, {station_id} & {trip_id} are keys for their respective tables.

- Check if your schema is in BCNF (Boyce-Codd Normal Form)

  {station_id} -> {station_name, latitude, longitude, dock_count, city, installation_date} – all are in same table, and station_id is the key

  {status_id} -> {time, station_id, bikes_available, docks_available} – all are in same table and status_id is the key

  {trip_id} -> {start_date, end_date, start_station_id, start_station_name, end_station_id, end_station_name, duration, bike_id, subscription_type} – all are in same table and trip_id is the key

  For above FDs, there are no violations for BCNF.

  The schema is in BCNF form, and there is no update in the ER diagram.

♦ Create your database using latest version of schema and import the data.

  Database is created in MySQL using the given schema. Initially tables are created, and csv files are imported using the below query. Errors while importing data were infile restriction errors and date time format errors. To solve this infile access has been provided and date time format has been adjusted according to MySQL preference.

  We have imported the tables using below query:

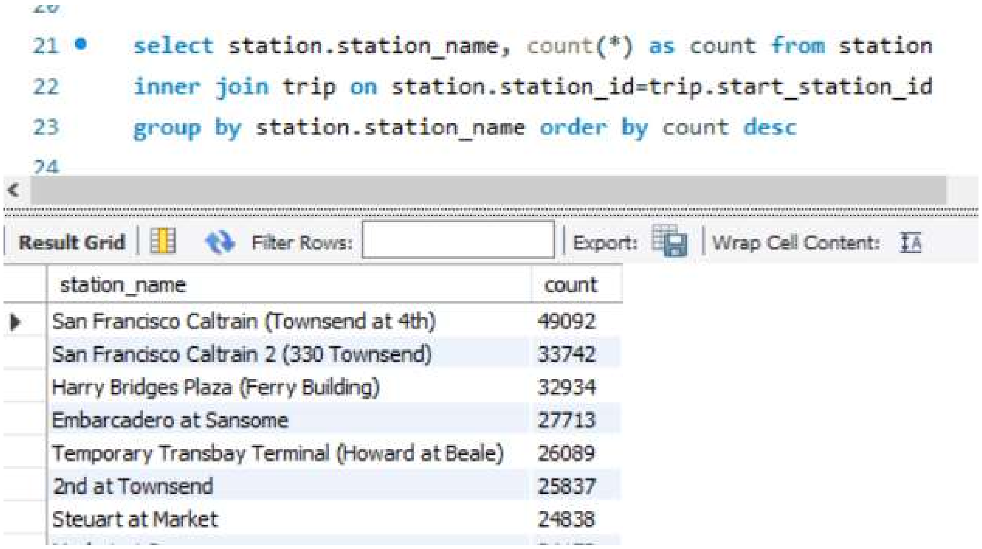
```
load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/status.csv'
into table status
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 rows;
```

Data cleaning and Database Testing:

- For each table in your database, check all the columns and the values they contain – Done in above steps.
- For numeric columns, check for the statistics – Statistics done in above steps

We have found few initial statistics above and below are insights inferred from the data.

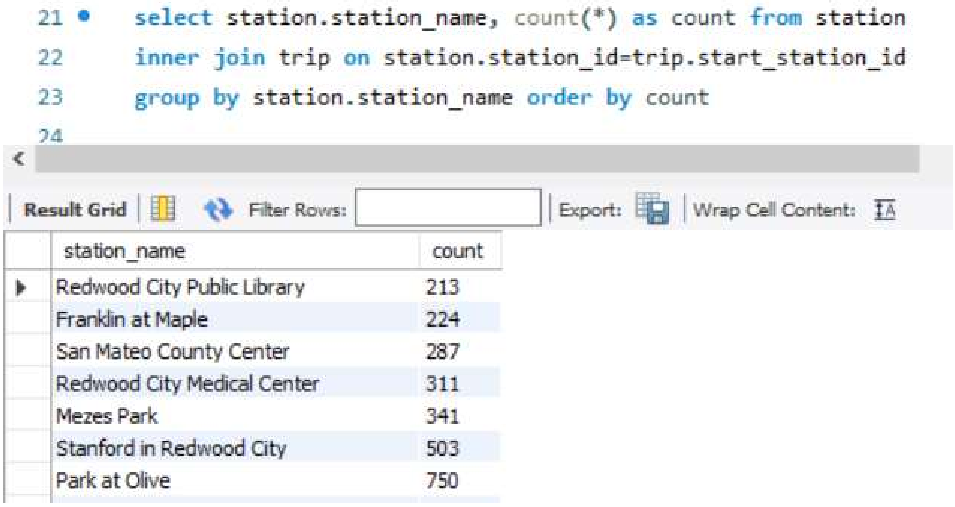
- Identify the stations that have been preferred the greatest number of times by customers:



```
select station.station_name, count(*) as count from station
inner join trip on station.station_id=trip.start_station_id
group by station.station_name order by count desc
```

station_name	count
San Francisco Caltrain (Townsend at 4th)	49092
San Francisco Caltrain 2 (330 Townsend)	33742
Harry Bridges Plaza (Ferry Building)	32934
Embarcadero at Sansome	27713
Temporary Transbay Terminal (Howard at Beale)	26089
2nd at Townsend	25837
Steuart at Market	24838

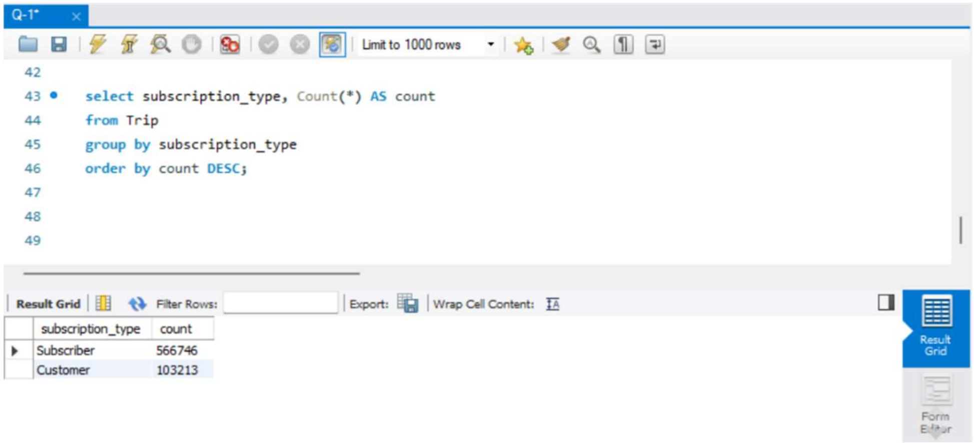
- Identify the stations that have been preferred least number of times by customers:



```
select station.station_name, count(*) as count from station
inner join trip on station.station_id=trip.start_station_id
group by station.station_name order by count
```

station_name	count
Redwood City Public Library	213
Franklin at Maple	224
San Mateo County Center	287
Redwood City Medical Center	311
Mezes Park	341
Stanford in Redwood City	503
Park at Olive	750

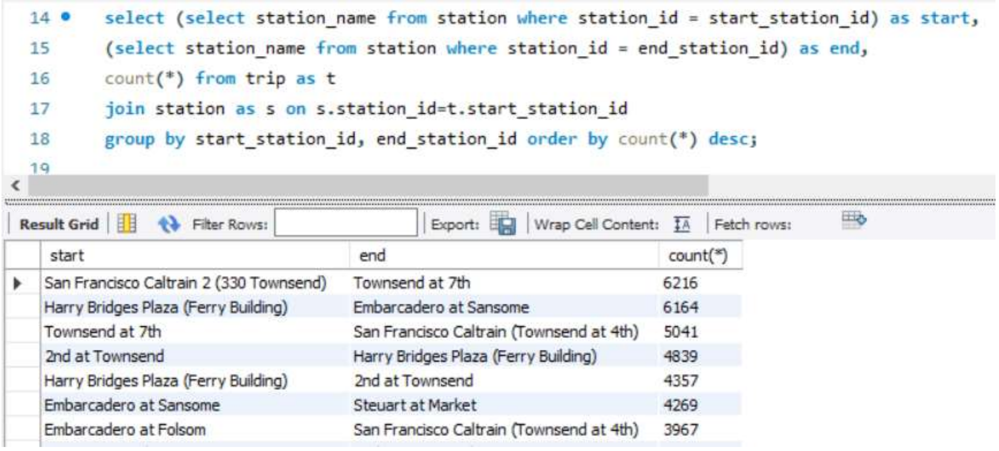
- Getting the numbers of subscribers and customers using the service:



```
select subscription_type, Count(*) AS count
from Trip
group by subscription_type
order by count DESC;
```

subscription_type	count
Subscriber	566746
Customer	103213

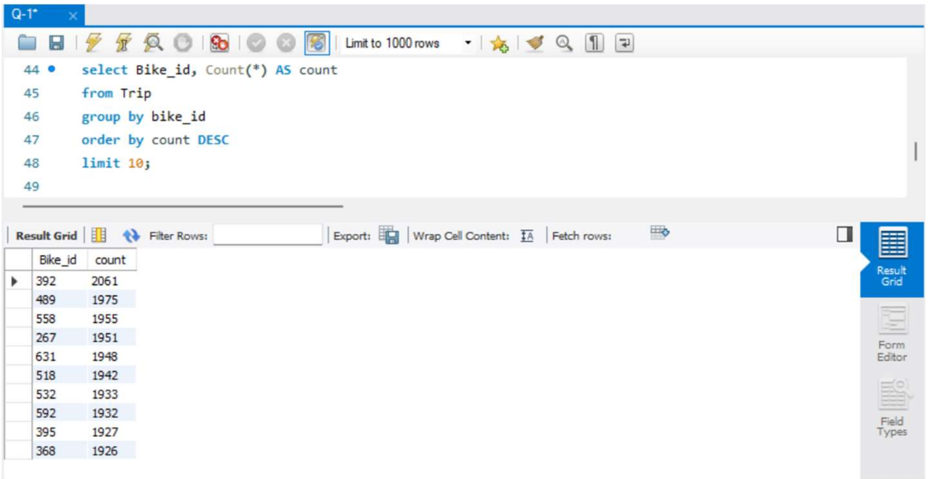
- Identifying the highest travelling routes:



```
select (select station_name from station where station_id = start_station_id) as start,
(select station_name from station where station_id = end_station_id) as end,
count(*) from trip as t
join station as s on s.station_id=t.start_station_id
group by start_station_id, end_station_id order by count(*) desc;
```

start	end	count(*)
San Francisco Caltrain 2 (330 Townsend)	Townsend at 7th	6216
Harry Bridges Plaza (Ferry Building)	Embarcadero at Sansome	6164
Townsend at 7th	San Francisco Caltrain (Townsend at 4th)	5041
2nd at Townsend	Harry Bridges Plaza (Ferry Building)	4839
Harry Bridges Plaza (Ferry Building)	2nd at Townsend	4357
Embarcadero at Sansome	Steuart at Market	4269
Embarcadero at Folsom	San Francisco Caltrain (Townsend at 4th)	3967

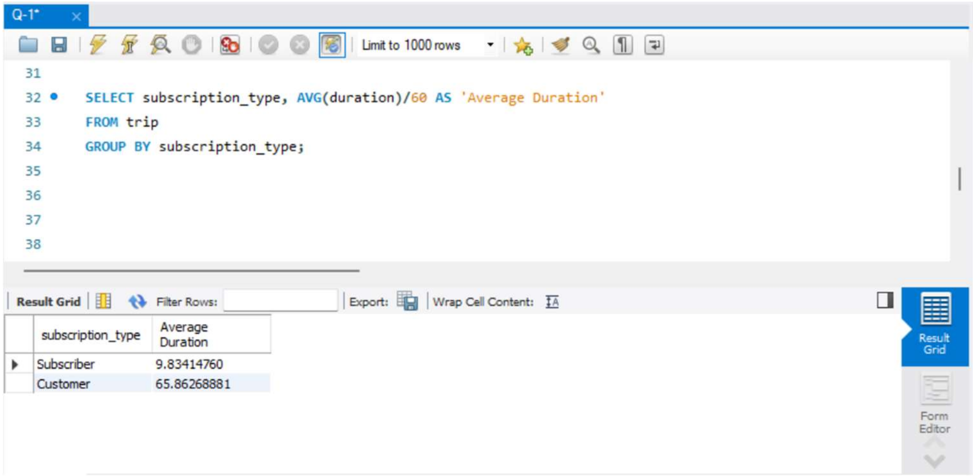
- Bike model that is most preferred by customers:



```
select Bike_id, Count(*) AS count
from Trip
group by bike_id
order by count DESC
limit 10;
```

Bike_id	count
392	2061
489	1975
558	1955
267	1951
631	1948
518	1942
532	1933
592	1932
395	1927
368	1926

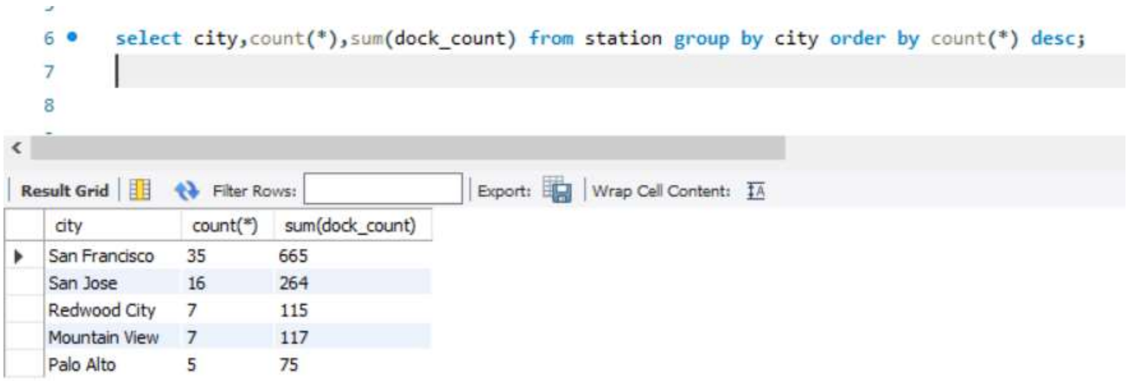
- The category of customers who took longer trips:



```sql
SELECT subscription_type, AVG(duration)/60 AS 'Average Duration'
FROM trip
GROUP BY subscription_type;
```

subscription_type	Average Duration
Subscriber	9.83414760
Customer	65.86268881

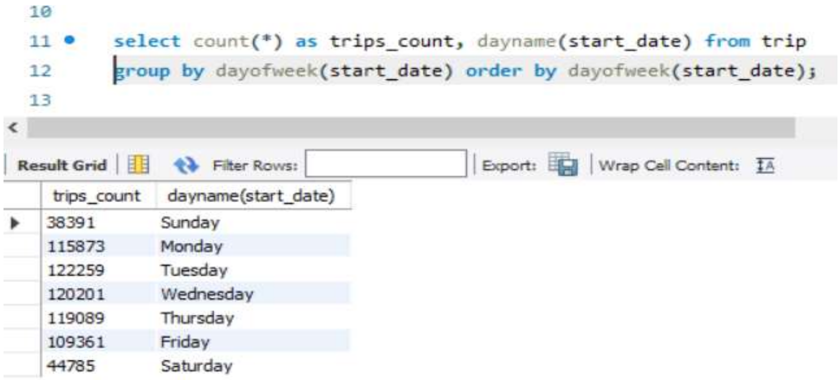
- Number of stations and docks count across various cities:



```sql
select city,count(*),sum(dock_count) from station group by city order by count(*) desc;
```

city	count(*)	sum(dock_count)
San Francisco	35	665
San Jose	16	264
Redwood City	7	115
Mountain View	7	117
Palo Alto	5	75

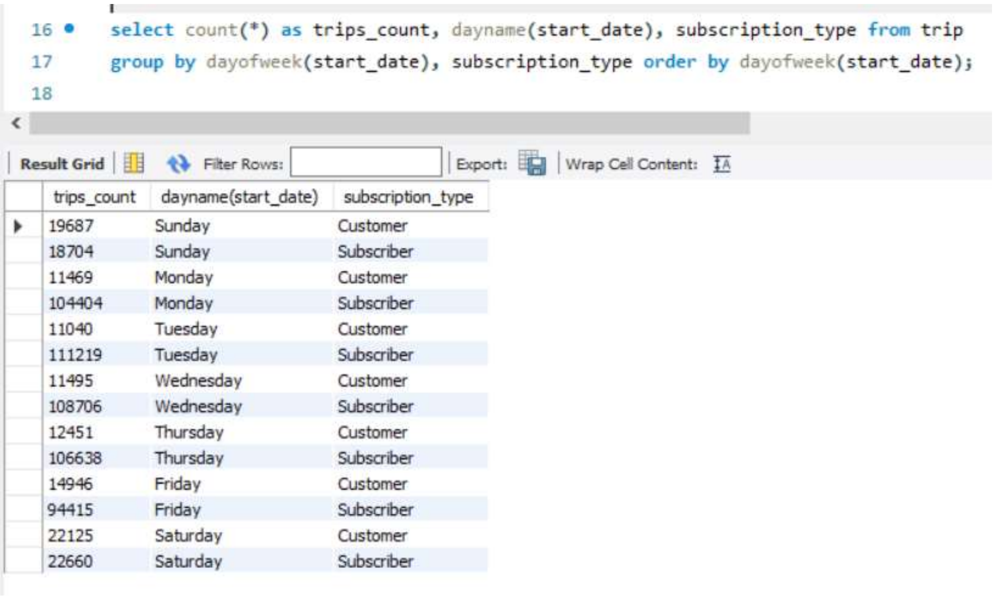
- Busiest days of the week:
  Weekdays are the busiest days of the week, as number of trips are lower on the weekends.



```sql
select count(*) as trips_count, dayname(start_date) from trip
group by dayofweek(start_date) order by dayofweek(start_date);
```

trips_count	dayname(start_date)
38391	Sunday
115873	Monday
122259	Tuesday
120201	Wednesday
119089	Thursday
109361	Friday
44785	Saturday

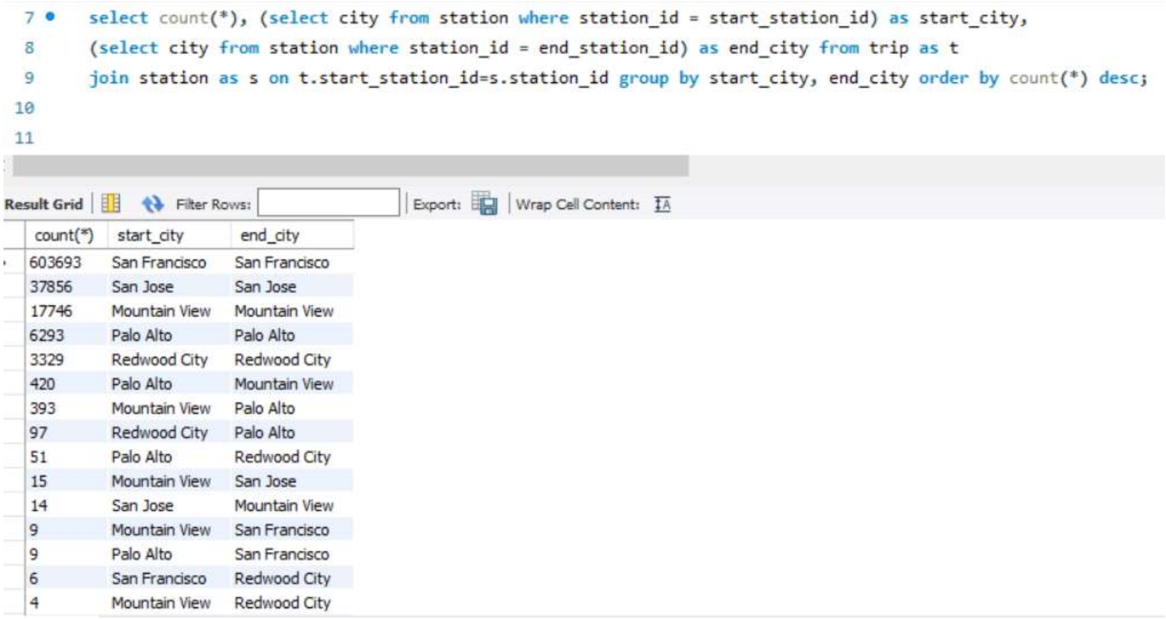
- Trips covered by customers and subscribers over the week:
  On weekdays, many numbers of subscribers use the bikes to travel. And in weekends this number comes down, but the count of trips by customers is increases compared to weekdays. We can infer that most of the subscribers are working on weekdays and have day off on weekends. And on weekends most customers go on trips.



```
select count(*) as trips_count, dayname(start_date), subscription_type from trip
group by dayofweek(start_date), subscription_type order by dayofweek(start_date);
```

trips_count	dayname(start_date)	subscription_type
19687	Sunday	Customer
18704	Sunday	Subscriber
11469	Monday	Customer
104404	Monday	Subscriber
11040	Tuesday	Customer
111219	Tuesday	Subscriber
11495	Wednesday	Customer
108706	Wednesday	Subscriber
12451	Thursday	Customer
106638	Thursday	Subscriber
14946	Friday	Customer
94415	Friday	Subscriber
22125	Saturday	Customer
22660	Saturday	Subscriber

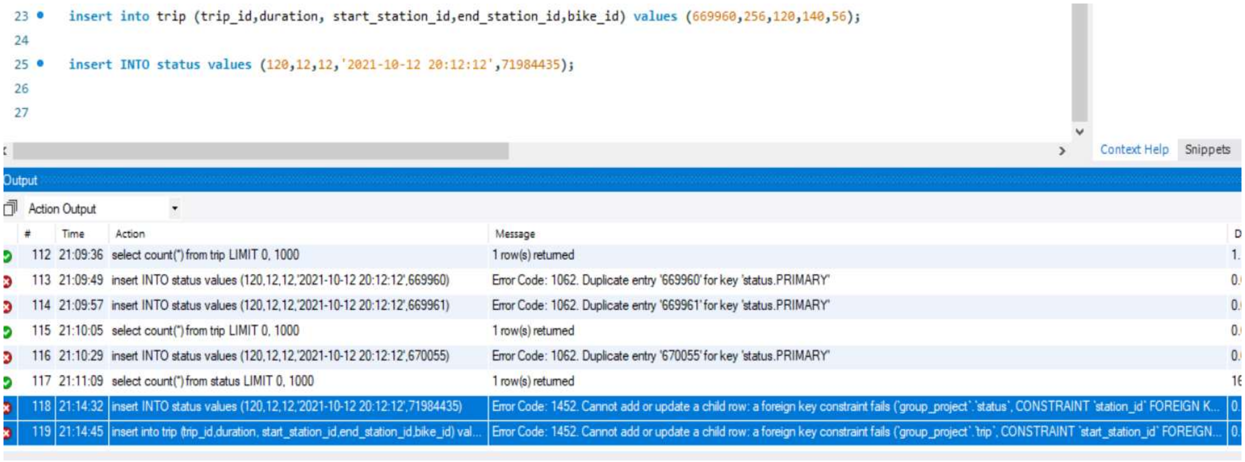
- Trips covered within cities and between cities:
  Highest intracity trips occurred in San Francisco followed San Jose, and highest intercity trips is between Palo Alto & Mountain View.



```
select count(*), (select city from station where station_id = start_station_id) as start_city,
(select city from station where station_id = end_station_id) as end_city from trip as t
join station as s on t.start_station_id=s.station_id group by start_city, end_city order by count(*) desc;
```

count(*)	start_city	end_city
603693	San Francisco	San Francisco
37856	San Jose	San Jose
17746	Mountain View	Mountain View
6293	Palo Alto	Palo Alto
3329	Redwood City	Redwood City
420	Palo Alto	Mountain View
393	Mountain View	Palo Alto
97	Redwood City	Palo Alto
51	Palo Alto	Redwood City
15	Mountain View	San Jose
14	San Jose	Mountain View
9	Mountain View	San Francisco
9	Palo Alto	San Francisco
6	San Francisco	Redwood City
4	Mountain View	Redwood City

- Foreign Key Constraints:
  We have tried to insert values into status and trip table with station id does not present in the station table and we received an error message.

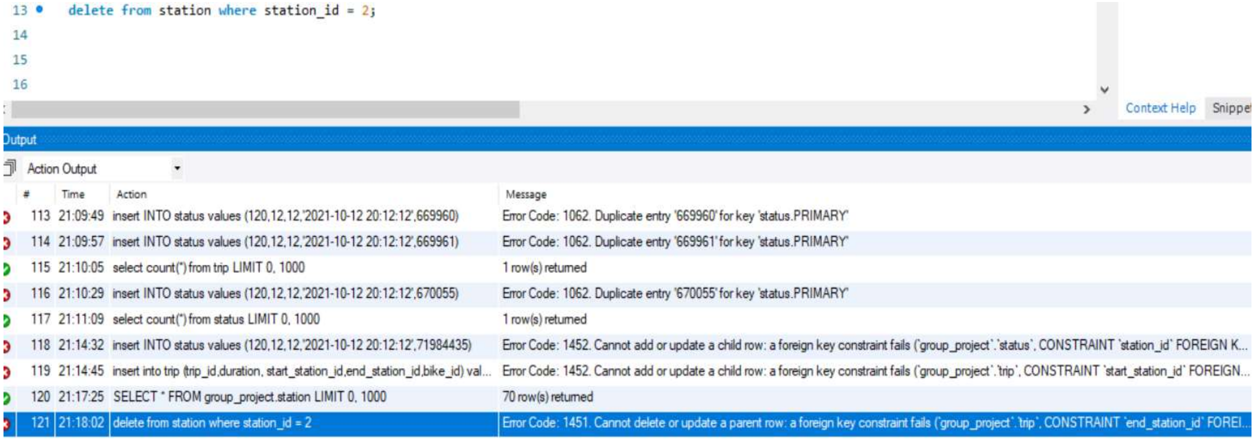


```sql
insert into trip (trip_id,duration, start_station_id,end_station_id,bike_id) values (669960,256,120,140,56);

insert INTO status values (120,12,12,'2021-10-12 20:12:12',71984435);
```

#	Time	Action	Message	D
112	21:09:36	select count(*) from trip LIMIT 0, 1000	1 row(s) returned	1.
113	21:09:49	insert INTO status values (120,12,12,'2021-10-12 20:12:12',669960)	Error Code: 1062. Duplicate entry '669960' for key 'status.PRIMARY'	0.
114	21:09:57	insert INTO status values (120,12,12,'2021-10-12 20:12:12',669961)	Error Code: 1062. Duplicate entry '669961' for key 'status.PRIMARY'	0.
115	21:10:05	select count(*) from trip LIMIT 0, 1000	1 row(s) returned	0.
116	21:10:29	insert INTO status values (120,12,12,'2021-10-12 20:12:12',670055)	Error Code: 1062. Duplicate entry '670055' for key 'status.PRIMARY'	0.
117	21:11:09	select count(*) from status LIMIT 0, 1000	1 row(s) returned	16
118	21:14:32	insert INTO status values (120,12,12,'2021-10-12 20:12:12',71984435)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`group_project`.`status`, CONSTRAINT `station_id` FOREIGN K...	0.
119	21:14:45	insert into trip (trip_id,duration, start_station_id,end_station_id,bike_id) val...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`group_project`.`trip`, CONSTRAINT `start_station_id` FOREIGN...	0.

Similarly, we have tried deleting a row from station table whose station id was present in other two tables and we got an error message.



```sql
delete from station where station_id = 2;
```

#	Time	Action	Message
113	21:09:49	insert INTO status values (120,12,12,'2021-10-12 20:12:12',669960)	Error Code: 1062. Duplicate entry '669960' for key 'status.PRIMARY'
114	21:09:57	insert INTO status values (120,12,12,'2021-10-12 20:12:12',669961)	Error Code: 1062. Duplicate entry '669961' for key 'status.PRIMARY'
115	21:10:05	select count(*) from trip LIMIT 0, 1000	1 row(s) returned
116	21:10:29	insert INTO status values (120,12,12,'2021-10-12 20:12:12',670055)	Error Code: 1062. Duplicate entry '670055' for key 'status.PRIMARY'
117	21:11:09	select count(*) from status LIMIT 0, 1000	1 row(s) returned
118	21:14:32	insert INTO status values (120,12,12,'2021-10-12 20:12:12',71984435)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`group_project`.`status`, CONSTRAINT `station_id` FOREIGN K...
119	21:14:45	insert into trip (trip_id,duration, start_station_id,end_station_id,bike_id) val...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`group_project`.`trip`, CONSTRAINT `start_station_id` FOREIGN...
120	21:17:25	SELECT * FROM group_project.station LIMIT 0, 1000	70 row(s) returned
121	21:18:02	delete from station where station_id = 2	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`group_project`.`trip`, CONSTRAINT `end_station_id` FOREI...