Bike Sharing Assignment

In [36]:
```python
import pandas as pd
from IPython.display import HTML
from IPython.display import Image
import math
```

Reading all CSV files with Pandas

In [4]:
```python
Trip = pd.read_csv("trip.csv")
Trip.head()
```

Out[4]:

	Unnamed: 0	id	duration	start_date	start_station_name	start_station_id	end_date	end_station_name	end_station_id	bike_id
0	0	4576	63	8/29/2013 14:13	South Van Ness at Market	66	8/29/2013 14:14	South Van Ness at Market	66	520
1	1	4607	70	8/29/2013 14:42	San Jose City Hall	10	8/29/2013 14:43	San Jose City Hall	10	661
2	2	4130	71	8/29/2013 10:16	Mountain View City Hall	27	8/29/2013 10:17	Mountain View City Hall	27	48
3	3	4251	77	8/29/2013 11:29	San Jose City Hall	10	8/29/2013 11:30	San Jose City Hall	10	26
4	4	4299	83	8/29/2013 12:02	South Van Ness at Market	66	8/29/2013 12:04	Market at 10th	67	319

In [5]:
```python
Weather = pd.read_csv("weather.csv")
Weather.head()
```

Out[5]:

	Unnamed: 0.1	Unnamed: 0	Date	Temperature	Humidity	Dew Point	mean_wind_speed_mph	Pincode
0	0	0	8/29/2013	68.0	75.0	58.0	11.0	94107
1	1	1	8/30/2013	69.0	70.0	58.0	13.0	94107
2	2	2	8/31/2013	64.0	75.0	56.0	15.0	94107
3	3	3	9/1/2013	66.0	68.0	56.0	13.0	94107
4	4	4	9/2/2013	69.0	77.0	60.0	12.0	94107

In [6]:
```python
Station = pd.read_csv("station.csv")
Station.head()
```

Out[6]:

	id	name	lat	long	dock_count	city	installation_date
0	2	San Jose Diridon Caltrain Station	37.329732	-121.901782	27	San Jose	8/6/2013
1	3	San Jose Civic Center	37.330698	-121.888979	15	San Jose	8/5/2013
2	4	Santa Clara at Almaden	37.333988	-121.894902	11	San Jose	8/6/2013
3	5	Adobe on Almaden	37.331415	-121.893200	19	San Jose	8/5/2013
4	6	San Pedro Square	37.336721	-121.894074	15	San Jose	8/7/2013

In [7]:
```python
Status = pd.read_csv("status.csv")
Status.head()
```

Out[7]:

	station_id	bikes_available	docks_available	time
0	2	2	25	2013/08/29 13:00:02
1	2	2	25	2013/08/29 14:00:01
2	2	2	25	2013/08/29 15:00:02
3	2	2	25	2013/08/29 16:00:01
4	2	2	25	2013/08/29 17:00:01

Creating DataBase & Tables

In [8]:
```python
import sqlite3
```

In [9]:
```python
conn = sqlite3.connect("Yulu.db")
cur = conn.cursor()
```

In [10]:
```python
# Create Trip table
Trip.to_sql("Trip", conn, if_exists="replace", index=False)
```

Out[10]: 669959

In [11]:
```python
# Create Weather table
Weather.to_sql("Weather", conn, if_exists="replace", index=False)
```

Out[11]: 3665

In [12]:
```python
# Create Station table
Station.to_sql("Station", conn, if_exists="replace", index=False)
```

Out[12]: 70

In [13]:
```python
# Create Status table
Status.to_sql("Status", conn, if_exists="replace", index=False)
```

Out[13]: 1000

Task 1: Get to Know Your Company

Q1: What are the total numbers of:

Q1.1: Bike stations?

In [14]:
```python
Total_stations = pd.read_sql_query("SELECT COUNT(id) as total_stations FROM Station",conn)
HTML(Total_stations.to_html(index=False))
```

Out[14]:

total_stations
70

Q1.2: Bikes?

In [15]:
```python
Total_bikes = pd.read_sql_query("SELECT COUNT(distinct(bike_id)) as total_bikes FROM Trip",conn)
HTML(Total_bikes.to_html(index=False))
```

Out[15]:

total_bikes
700

Q1.3: Trips?

In [16]:
```python
Total_trips = pd.read_sql_query("SELECT COUNT(*) as total_bikes FROM Trip",conn)
HTML(Total_trips.to_html(index=False))
```

Out[16]:

total_bikes
669959

Q2: Construct a geographical plot to show the location of each bike station using the latitude and longitude provided under the Station table.

Please refer Tableau file shared

In [40]:
```python
Image(url="screenshots/1.2.png", width=700, height=800)
```

Out[40]:

Q3: What is the relationship between the following columns (one to one, many to one, many to many)?

Q3.1: bike_id (Trip table) and start_station_id (Trip table)

Answer: Many to many relationship i.e as seen in the below results multiple bike_id's are associated to multiple start_station_id

In [17]:
```python
Totalstart_station_id = pd.read_sql_query("SELECT bike_id ,count(DISTINCT(start_station_id)) as Totalstart_stat
HTML(Totalstart_station_id.to_html(index=False))
```

Out[17]:

bike_id	Totalstart_station_id
876	5
323	10
697	10

In [18]:
```python
Totalbikeid = pd.read_sql_query("SELECT start_station_id ,COUNT(DISTINCT(bike_id)) as Totalbikeid FROM Trip gro
HTML(Totalbikeid.to_html(index=False))
```

Out[18]:

start_station_id	Totalbikeid
24	109
23	116
21	119

Q3.2: pincode (Weather table) and station location (latitude and longitude in Station table)

Answer: There is no Relationship between Weather table & (latitude and longitude in Station table) since there are no common columns between them (No foreign key relationship)

Q3.3: 8/29/2013 (date column in Weather table) and mean wind speed (Weather table)

Answer: 8/29/2013 in date column has 5 mean wind speed (as seen below), hence one date (8/29/2013) has many mean wind speed resulting in one to many relationship. But note that even one mean speed has many dates, hence overall the relationship between Date column and mean wind column is many to many but if we pick one specific date such as (8/29/2013) this is one to many (mean wind speed) relationship

In [19]:
```python
mean_speed = pd.read_sql_query("SELECT date ,mean_wind_speed_mph as meanspeed FROM Weather where date = '8/29/2
HTML(mean_speed.to_html(index=False))
```

Out[19]:

Date	meanspeed
8/29/2013	11.0
8/29/2013	6.0
8/29/2013	8.0
8/29/2013	5.0
8/29/2013	7.0

Q4: Find the first and the last trip in the data

In [20]:
```python
#First Trip
first_trip = pd.read_sql_query("SELECT * FROM Trip order by id limit 1",conn)
HTML(first_trip.to_html(index=False))
```

Out[20]:

Unnamed: 0	id	duration	start_date	start_station_name	start_station_id	end_date	end_station_name	end_station_id	bike_id	su
32	4069	174	8/29/2013 9:08	2nd at South Park	64	8/29/2013 9:11	2nd at South Park	64	288	

In [21]:
```python
#last Trip
last_trip = pd.read_sql_query("SELECT * FROM Trip order by id desc limit 1",conn)
HTML(last_trip.to_html(index=False))
```

Out[21]:

Unnamed: 0	id	duration	start_date	start_station_name	start_station_id	end_date	end_station_name	end_station_id	bike_id	s
315807	913460	765	8/31/2015 23:26	Harry Bridges Plaza (Ferry Building)	50	8/31/2015 23:39	San Francisco Caltrain (Townsend at 4th)	70	288	

Q5: What is the average duration

Q5.1: Of all the trips?

In [22]:
```python
# Average Duration of all trips
Average_DurationAlltrip = pd.read_sql_query("SELECT round(avg(duration)) as Average_Duration_of_allTrips FROM T
HTML(Average_DurationAlltrip.to_html(index=False))
```

Out[22]:

Average_Duration_of_allTrips
1108.0

Q5.2: Of trips on which customers are ending their rides at the same station from where they started?

In [23]:
```python
# Average Duration of all trips with same start and end location
AvgDuration_Location = pd.read_sql_query("SELECT round(avg(duration)) as AvgDuration_sametoFroLocation FROM Tri
HTML(AvgDuration_Location.to_html(index=False))
```

Out[23]:

AvgDuration_sametoFroLocation
6357.0

Q6: Which bike has been used the most in terms of duration? (Answer with the Bike ID)

In [24]:
```python
# Will find sum of duration for each bike (group by bike_id) and then order by sum duration desc
most_usedBike = pd.read_sql_query("SELECT bike_id ,sum(duration) as Total_duration FROM Trip group by bike_id o
HTML(most_usedBike.to_html(index=False))
```

Out[24]:

bike_id	Total_duration
535	18611693

Q7: Plot the most suitable graph for the following:

Q7.1 The average duration of a trip versus Number of trips

Please refer Tableau file shared

In [44]:
```python
Image(url="screenshots/1.7.1.png", width=700, height=800)
```

Out[44]:

Q7.2 Hour of start time versus No. of trips

Please refer Tableau file shared

In [45]:
```python
Image(url="screenshots/1.7.2.png", width=700, height=800)
```

Out[45]:

Q7.3 Day of the week versus No. of trips also denote subscribers and customers with different colors.

Please refer Tableau file shared

In [46]:
```python
Image(url="screenshots/1.7.3.png", width=700, height=800)
```

Out[46]:

Task 2: Demand Prediction

Zulip is running under a loss and has decided to shut operations for three of its stations. You have to use the data provided to help Zulip decide which three stations should be shut.

Q1: What are the top 10 least popular stations? Hint: Find the least frequently appearing start stations from the Trip table

In [25]:
```python
Ten_leastpopular = pd.read_sql_query("SELECT start_station_name ,Count(start_station_name) as leastUsed_start_s
HTML(Ten_leastpopular.to_html(index=False))
```

Out[25]:

start_station_name	leastUsed_start_station_name
San Jose Government Center	23
Broadway at Main	67
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	436
Park at Olive	750
Santa Clara County Civic Center	840

Q2: 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.

Q2.1: Find the idle time for Station 2 on the date '2013/08/29'

In [32]:
```python
# Below shows the top 10 idle time for station 2
idleTime_Station2 = pd.read_sql_query("SELECT time as idle_time , bikes_available from Status where station_id
HTML(idleTime_Station2.to_html(index=False))
```

Out[32]:

idle_time	bikes_available

Q3 : In case two stations are nearby, it might be possible to shut one down. Find the distance between consecutive stations (between Stations 1 and 2, Stations 2 and 3, and so on) using Haversine formula

2arcsin(√sin^2((ϕ2−ϕ1)/2)+(1−sin^2((ϕ2−ϕ1)/2)−sin^2((ϕ2+ϕ1)/2).sin^2((λ2−λ1)/The Haversine formula 2))

The Haversine formula is used to find the distance between two points on a sphere given their longitude and latitude. (ϕ1,λ1) is the latitude–longitude pair for the first station, and (ϕ2,λ2)is the latitude–longitude pair for the second station.

Answer for this question alone please refer the .sql file shared along withe the assignment. since sqllite3 does not support trignometric function had to go with mysql workbench

Following is the approach we used to solve this problem,

Created a table using cross join and then applied the formula to check minimum distance(miles)

Below are quries(you will find the same in .sql file)

```sql
create table cross_j (
SELECT a.station_name station_a,
b.station_name station_b, a.latitude station_a_lat, b.latitude station_b_lat,a.longitude
station_a_long,b.longitude station_b_long
FROM station AS a CROSS JOIN station AS b where a.station_name!=b.station_name
);
SELECT station_a,station_b,2 * 3961 * asin(sqrt(power((sin(radians((station_b_lat - station_a_lat)
/ 2))), 2) + cos(radians(station_a_lat)) * cos(radians(station_b_lat)) *
power((sin(radians((station_b_long - station_a_long) / 2))), 2))) as distance
from cross_j order by distance;
```

In [39]:
```python
Image(url="screenshots/Havensine formula.png", width=700, height=800)
```

Out[39]:

Q4: Use the findings above to recommend three stations that can be shut. (open ended) For example, if the Japantown and Ryland stations are nearby, and the Japantown is not as popular as the Ryland station, then it can be recommended to shut.

From the above results of most closest and Question-1 top 10 least popular stations

we see that

'Redwood City Public Library' and **'Franklin at Maple'** are very close as well as not so popular

'Franklin at Maple' and **'San Mateo County Center'** are very close as well as not so popular

'Redwood City Public Library' and **'San Mateo County Center'** are very close as well as not so popular

Therefore, the we stations we recommend to be shutdown are,

1. **Redwood City Public Library**
2. **Franklin at Maple**
3. **San Mateo County Center**

Task 3: Optimizing Operations

Q1: Calculate the average number of bikes and docks available for Station 2 and Station 3 (Hint: Use the Status table.)

In [27]:
```python
avg_num = pd.read_sql_query("SELECT station_id ,round(avg(bikes_available)) as avg_bikes_available, round(avg(d
HTML(avg_num.to_html(index=False))
```

Out[27]:

station_id	avg_bikes_available	avg_docks_available
2	11.0	16.0

Q2: Plot the popularity of each station on a map for subscribers and customers. (Hint: Popular stations appear most frequently under the column start_station_name in the Trip table)

Please refer Tableau file shared

In [47]:
```python
Image(url="screenshots/3.2.png", width=700, height=800)
```

Out[47]:

Q3: Plot the number of trips per hour for all the data provided in the Trip table.

Please refer Tableau file shared

In [48]:
```python
Image(url="screenshots/3.3.png", width=700, height=800)
```

Out[48]:

Q4: Use the findings above to provide insights on how to optimize operations

Answer: From the above finding we could see which are the most popular areas and also through tableau map for customer & subscriber, we could see how popularity varies. To optimize and more profits we could increase the avg available bikes at most popular areas one such example is id = 70 (San Francisco Caltrain (Townsend at 4th)) in station table, this station is one of the most popular amoung subscribers, at this station we can increase avg available bikes

In [49]:
```python
Image(url="screenshots/3.4.png", width=700, height=800)
```

Out[49]:

Task 4: Couple Bikes? (Bonus)

Zulip has decided to start a new product line called Couple Bikes. This will enable two persons to travel from one station to another at the same time. What are some of the factors that you will have to consider while validating the idea of couple bikes?

Factors to consider for validating the possibilty of couple bike are,

1. If Start Station Name and End Station Name are same
2. Target the most popular areas as these areas will mostly be from where working professionals, students etc will travel.
3. Cost when 2 person travelling in single bike must be less than the cost that 2 individual bikes would have cost
4. Least Most popular areas where we must avoid couple bike implementation

Lets discussion all 3 Factors one by one.....................

Factor-1 : If Start Station Name and End Station Name are same

Traget above areas which had same start and end location these area had total 9,68,09,91,556 trips out of 3,08,43,70,70,970 trips in 3 years that is around 3 % of trips with same start and end location

In [33]:
```python
sameLocation = pd.read_sql_query("SELECT start_station_name FROM Trip where start_station_name = end_station_na
HTML(sameLocation.to_html(index=False))
```

Out[33]:

start_station_name
2nd at Folsom
2nd at South Park
2nd at Townsend
5th at Howard
Adobe on Almaden
Arena Green / SAP Center
Beale at Market
Broadway St at Battery St
Broadway at Main
California Ave Caltrain Station
Castro Street and El Camino Real
Civic Center BART (7th at Market)
Clay at Battery
Commercial at Montgomery
Cowper at University
Davis at Jackson
Embarcadero at Bryant
Embarcadero at Folsom
Embarcadero at Sansome
Embarcadero at Vallejo
Evelyn Park and Ride
Franklin at Maple
Golden Gate at Polk
Grant Avenue at Columbus Avenue
Harry Bridges Plaza (Ferry Building)
Howard at 2nd
Japantown
MLK Library
Market at 10th
Market at 4th
Market at Sansome
Mechanics Plaza (Market at Battery)
Mezes Park
Mountain View Caltrain Station
Mountain View City Hall
Palo Alto Caltrain Station
Park at Olive
Paseo de San Antonio
Post at Kearney
Post at Kearny
Powell Street BART
Powell at Post (Union Square)
Redwood City Caltrain Station
Redwood City Medical Center
Redwood City Public Library
Rengstorff Avenue / California Street
Ryland Park
SJSU - San Salvador at 9th
SJSU 4th at San Carlos
San Antonio Caltrain Station
San Antonio Shopping Center
San Francisco Caltrain (Townsend at 4th)
San Francisco Caltrain 2 (330 Townsend)
San Francisco City Hall
San Jose City Hall
San Jose Civic Center
San Jose Diridon Caltrain Station
San Mateo County Center
San Pedro Square
San Salvador at 1st
Santa Clara County Civic Center
Santa Clara at Almaden
South Van Ness at Market
Spear at Folsom
St James Park
Stanford in Redwood City
Steuart at Market
Temporary Transbay Terminal (Howard at Beale)
Townsend at 7th
University and Emerson
Washington at Kearney
Washington at Kearny
Yerba Buena Center of the Arts (3rd @ Howard)

In [29]:
```python
# All trips with same start and end location
AvgDuration_Location = pd.read_sql_query("SELECT sum(id) as total_trips FROM Trip where start_station_name = en
HTML(AvgDuration_Location.to_html(index=False))
```

Out[29]:

total_trips
9680991556

Factor-2 : Target the most popular area as these area will mostly be from where working professionals, students etc will travel

Traget Areas such as,

1. **College/University from nearest metro stations** These are the areas which will have same start and end between college/university to metro vice versa.
2. **IT Tech parks / Hospitals / Major Bank Branches to nearest Metro Stations** These area are also those can be busy 24 hrs as they function 24hrs due to there work culture and since crowed travelling to & fro from metro stations to these areas is more frequent, its one of the best location to tackle
3. **Lastly Grocery/hypermart/Super market such as walmart to the nearest hosuing society/flats/apartment residential areas** These are very good location which will target all group of people, since it involve day to day basic shopping

Factor-3 : Cost when 2 person travelling in single bike must be less than the cost that 2 individual bikes would have cost

1. Cost for transportation is one of the main factors that every individual co least mpares while opting the best options available in public transport
2. At the very least, an individual cost while in couple travel must be less than 25%. Beacuse this % shall give individual 25 % and Bike company will get 50%.
3. Lets say P1 and P2 are two individual who are travelling same destination, in normal situation it will cost them X amount each respectively i.e. 2X in Total and lets also assume travel cost of bike for bike comapny also as 2X.
4. Now if travelled in couple bike with 25% discount P1 and P2 Total cost would be 1.5X (0.75X each) and lets say travel cost of bike for bike comapny as X (assuming same amount from above for single bike), then we clearly see that profit for bike comapny also is 50%.

Note: Factor 2 & 3 cannot be completely accurated/validated since these are just formualted on based market needs that we all see day to day. Also the above data does not contain any such column through which we can check Cost or specific locations related to metro stations/universities/Markets

Factor-4: Least most Popular area

We must avoide those regions/stations which are least used for example below are the 10 most least popular areas based on amount of trips that take place

In [30]:
```python
Ten_leastpopular = pd.read_sql_query("SELECT start_station_name ,Count(id) as total_trips from Trip group by st
HTML(Ten_leastpopular.to_html(index=False))
```

Out[30]:

start_station_name	total_trips
San Jose Government Center	23
Broadway at Main	67
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	436
Park at Olive	750
Santa Clara County Civic Center	840

Conclusion:

Two main parameters must be considered while deciding couple bikes idea,

1. **Location:** 1.1 Best fit stations - Most popular stations 1.2 Worst fit stations - Least popular stations(to avoid) 1.3 Stations near Universities/Markets/Metro Stations/IT Tech Parks/Hospitals
2. **Cost/Price** point for customers such that it is a win-win situation for all