Comprehensive Final Deliverable

Data Management & SQL - DAT-5305 - FMSBA3

Andrew Peynado

Deniz Gürcan

Student ID: 4927234

Master of Business Analytics

MSBA 3

HULT International Business School

Table of Contents

Theory **3**

1.1 Data Moat **3**

1.2 Data Team **3**

1.3 Data Privacy **4**

1.4 Where vs. Having **4**

1.5 Entity Relationship Model **5**

Database Design **6**

2.1 Students and Classes **6**

2.2 Customers and Products **7**

2.3 Library Reservation System **9**

Data Analysis with SQL **10**

San Francisco Bikeshare Analysis **10**

Data Visualization **11**

San Francisco Bikeshare Dashboard **11**

Appendix **12**

Appendix 1 – SQL Queries and Output Tables **12**

Theory

1.1 Data Moat

What is a Data Moat? Why is it important to have one?

A data moat is a long-term competitive advantage due to the ownership of unique or proprietary data. This data has to be large unable to synthesize or prawn to be replicated and difficult to acquire. Once a data moat s established it grows exponentially, by enhancing a company's data pool and its expertise for instance through automated decision making or fueling Machine Learning and AI. This creates even higher barriers of entry for competitors. Thus, to compete in times where data is the new oil a moat will ultimately enable and protect a company's ability to operate.

1.2 Data Team

What are the 3 different roles in a modern data team? Which problems do each of them solve? How do they compare with each other?

Three roles in data teams:

- Data Engineer
- Data Analyst
- Data Scientist

One might imagine the relationship between these roles as hierarchical or complementary. A Data Engineer is responsible for laying out the fundament, meaning laying out a data infrastructure that enables others to access and work with the data effectively. A Data Analyst dives deep into the data with the purpose of drawing business insights. The Data Scientist builds on that by utilizing the data to provide guidance for future decisions or to build predictive models. Many companies define these roles differently, thus creating differing responsibilities and synergies between these parties.

1.3 Data Privacy

Share your opinion on current data privacy laws. Are they doing enough to protect consumers? Why or why not?

Data privacy laws are finally in the focus of politics with major regulations such as GDPR and CCPA. However, these laws and politics in general are rather reactive and slow. This implies that policy makers don't understand the subject and more importantly cannot anticipate next steps of innovators such as Google or Facebook. This exposes the general public to legal grey areas which gives big (tech) companies an unjustified amount of control and power over consumer and more importantly politics. These policies cannot be introduced and then stay the same way. They have to be constantly and rapidly improved, as companies won't stop either to evolve.

1.4 Where vs. Having

What is the difference between the WHERE and HAVING clauses?

Both clauses are utilized to exclude specific information within a query. WHERE filters rows and usually comes before a HAVING or GROUP BY clause. HAVING filters groups based on certain specified conditions and functions, not limited to aggregations such as AVG, MAX, SUM etc. Another difference is that WHERE uses indexes, whereas HAVING doesn't.

The main difference is that WHERE is used to filter rows before grouping and HAVING excludes records after grouping, thus the usage of those clauses should be assessed logically to achieve the desired output.

1.5 Entity Relationship Model

How would you define the relationship between employees and offices in the Entity Relationship (ER) model? Please provide an explanation why using real world examples.

The relationship in an Entity-Relationship model between employees and offices can be 1:M. An office needs many employees but an employee can only work in one office, if the data is static and an employee works normal hours and only one job. Looking at amazon for instance, There is one HQ but there are many employees that can work there. It also can be argued that the relationship is M:N, since many Amazon offices need many employees and many employees can work in amazon offices worldwide. However an employee cannot work at different offices at the same time (physically), thus the relationship is 1:M.

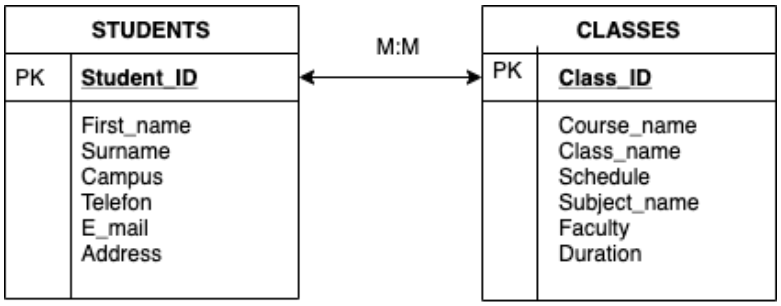
Database Design

2.1 Students and Classes

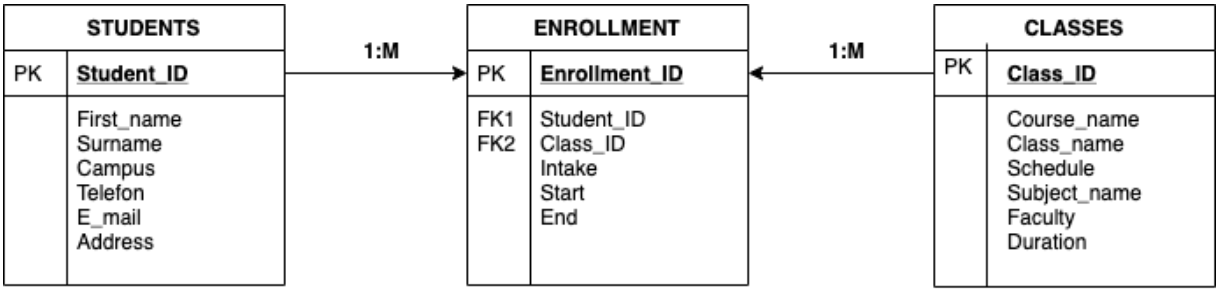
You are asked to model the many to many relationship between students and classes in a relational database.

- *What changes do you need to make to support this relationship?*
- *Please create an ER diagram to show how these entities will relate to each after your changes.*

The entities are STUDENTS and CLASSES. The attributes for Students are First_name, Surname, Enrollment_ID, Campus, Telefon, E_mail and Address. Course_name, Class_name, Schedule, Subject_name, Faculty and Duration build the attributes for Classes.



However, to properly create a M:M relationship we have to split into two 1:M relationships. Therefore, relevant primary keys (PK) and foreign or secondary keys (FK) are defined for each entity. The enrollment entity is a join table which is created to connect each student to a class. This is necessary since every student can attend multiple classes. The entities have unique identifiers and therefore can have many to many relationships.

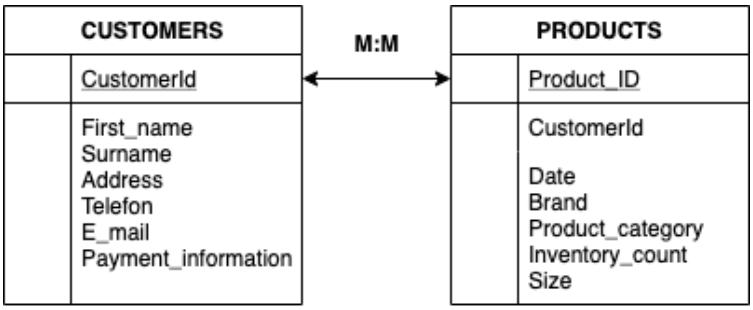


2.2 Customers and Products

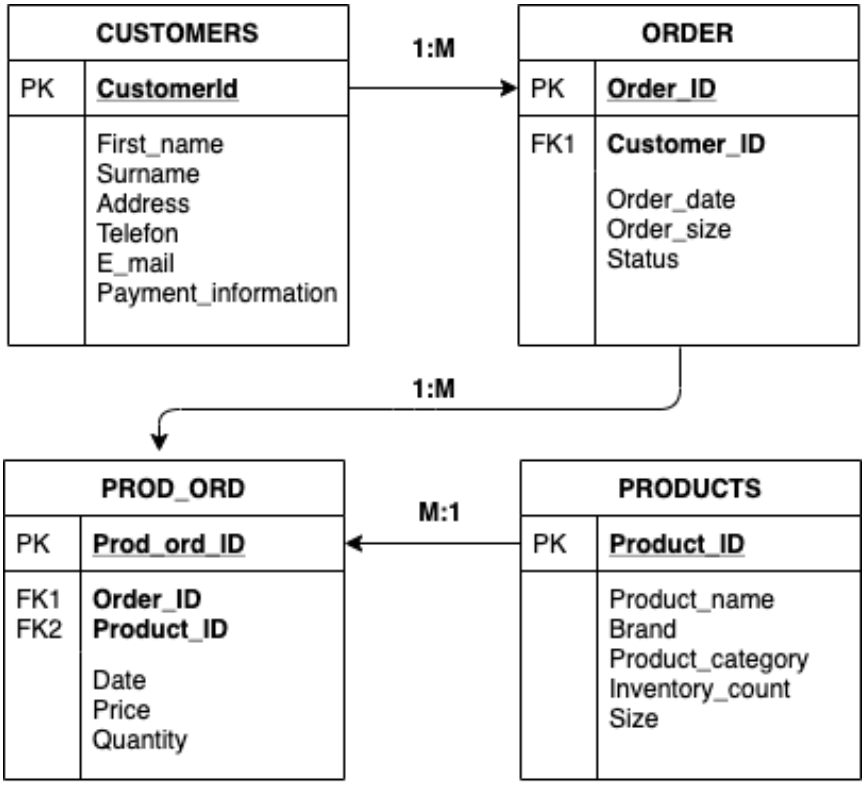
You are asked to model the many to many relationship between customers and products in a relational database.

- *What changes do you need to make to support this relationship?*
- *Please create an ER diagram to show how these entities will relate to each after your changes.*

The entities for this scenario are Customers and Products. An initial draft for the attributes can be found within each table. In order to properly display the relationship, further division into several 1:M relationships has to be conducted.



It's important to realize that every customer can order every product and that very product can be ordered by every customer. Thus, the ER has to be enhanced with two 1:M relationships. This makes it possible to relate an order to a single customer, which then can make multiple orders. Due to the fact that Orders and Products can have a M:M relationship a joined table (Prod_ord) has to be introduced to distinguish between different orders from different products. Specific attributes can be found within the tables and were cleaned to avoid overlapping information.


CUSTOMERS
PK
CustomerId
First_name
Surname
Address
Telefon
E_mail
Payment_information
1:M
ORDER
PK
Order_ID
FK1
Customer_ID
Order_date
Order_size
Status
1:M
PROD_ORD
PK
Prod_ord_ID
FK1
FK2
Order_ID
Product_ID
Date
Price
Quantity
M:1
PRODUCTS
PK
Product_ID
Product_name
Brand
Product_category
Inventory_count
Size

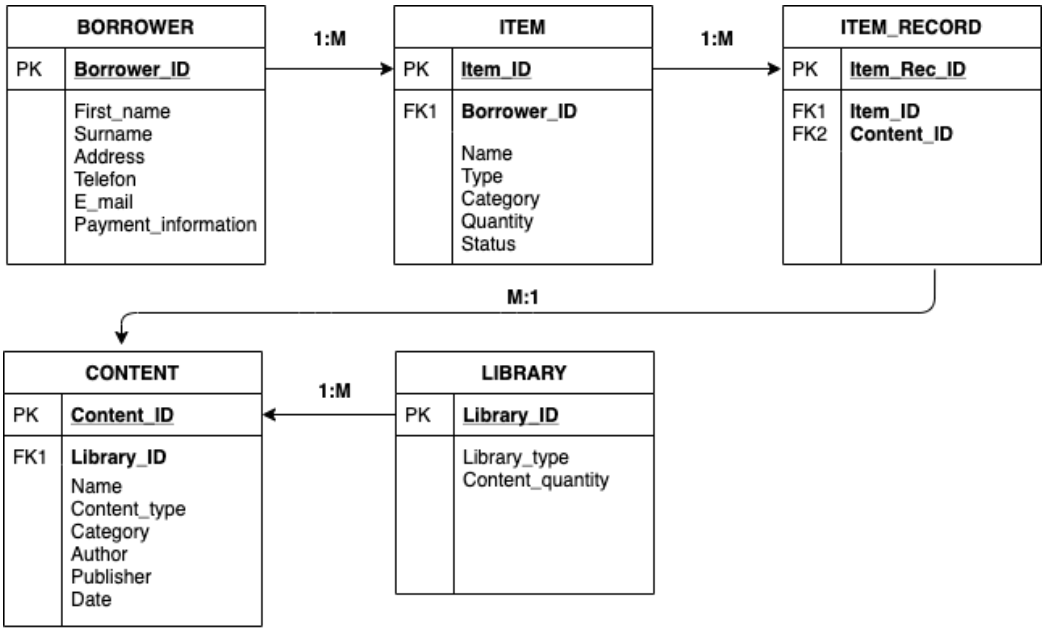
2.3 Library Reservation System

Design an ER diagram for a library reservation system for a family of libraries based on the given characteristics.

- *This system is for multiple libraries*
- *This system is for multiple borrowers*
- *There are multiple types of content that can be borrowed*
- *Borrowers can borrow multiple items at the same time*
- *Borrowers can borrow multiple types of content*

The following ER-Diagram assumes that even within multiple libraries, content is limited to a single specific library. For instance, Dodge Ram in Truck or Ferrari in Sportscar).

In this case many borrowers can borrow many items. Therefore, this M:M relationship has to be split into two 1:M, which are item and Item_rec. Item_rec is furthermore connected to Content and therefore Library. Many libraries can obtain multiple contents but a single content belongs to a unique library. Thus a 1:M relationship was created. If unique content were to be found within several libraries (Dodge Ram in Truck AND SUV), the diagram would be significantly different. This might require further segmentation.



Data Analysis with SQL

San Francisco Bikeshare Analysis

Complaints about empty stations seem very subjective looking at the data. A first exploration reveals that of the stations deployed on the street, there are 3 out of 452 stations that have no bikes available. Only one station reports a total capacity of 0, meaning this station might be dysfunctional. The two other stations range amongst a total of 12 stations that don't allow rental or return of bikes. However, the overall condition and availability of stations and bikes is rather good, indicating the problem must be somewhere else.

Looking at the most popular station is San Francisco Caltrain (Townsend at 4th) accounting for 3.7% of all start points and 4.7% of all end points of the trips. It is notable that none of the identified potential dysfunctional stations are in amongst the top ten stations. Almost all of the top ten start stations and end stations are in San Francisco in the Financial District spreading to the Mission and are located in proximity to train stations.

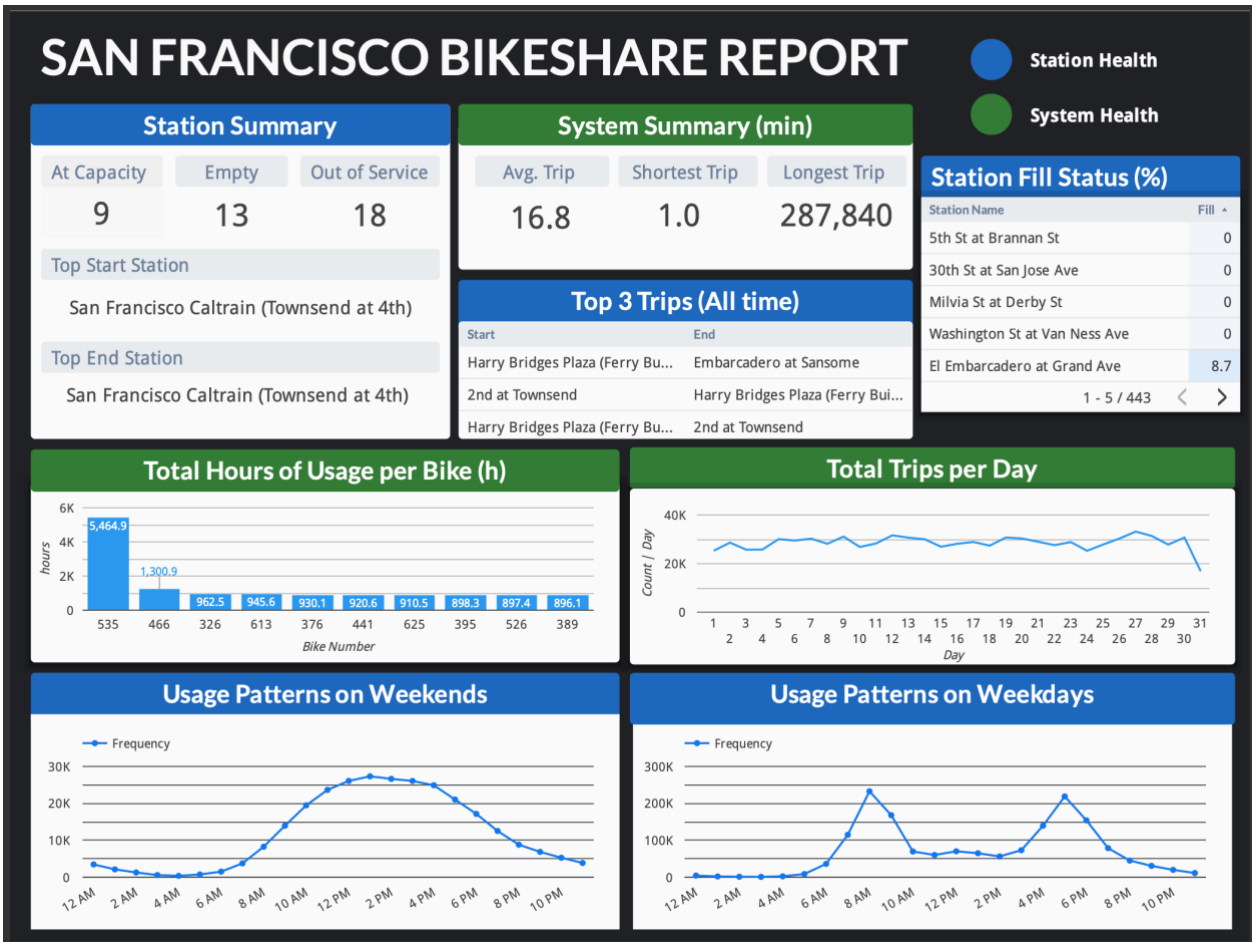
The trip duration peaks on the weekends on Saturday and Sunday, whereas the number of trips is higher during the week. Therefore, it is assumed that roundtrips which account 3.26% of all trips and have an average duration of more than 1.5h are conducted for recreation or tourism purposes. Commuter are likely to drop their bikes at different stations. They account for 96.7% of all trips with an average time less than 0.5h. Regarding the types of subscriptions, tourists account for 13,9% with daily or three-day passes, whereas commuter account for 86% with annual or 30-day subscriptions.

Dysfunctional stations are rather located outside of high intensity areas. They can be found primarily in Oakland Berkley or San Jose. This emphasizes a potential lack of urgency to maintain those stations especially in regards to the last reported status update which is not frequent in most cases. Therefore, there is a necessity to reevaluate the importance of such stations in terms of potential future usage, since it is better to reduce the number of unused stations rather than gaining negative customer reviews and satisfaction. Common sense dictates that if highly used stations are in the areas such as the financial district there will be scarcity of bikes especially during rush hours since commuters make their way home and will not return the bikes for another commuter to use. Almost all trips are conducted for commuting purposes and therefore short but very frequent. That indicates that the number of docks and/or the number of docks per station and thus the number of available bikes have to be increased in such areas in order to meet the demand. This can be done by allocating bikes from low intensity areas to where they are needed, which ultimately increases efficiency, coverage and most importantly customer satisfaction.

The overall essence of this analysis is that in general stations work sufficiently. However, there is a slight misunderstanding of the necessity of low volume areas and high intensity areas. Thus, complaints of customers occur not in regards of the functionality but in regards of the scarcity of bikes. Therefore, the recommendation in order to increase customer satisfaction is to focus on the needs and requirements of the largest customer group: the commuters. It is advisable to implement a real time tracking system which reports the condition of stations. Furthermore, a partnership with a service provider that maintains and allocates bikes to docks and areas where needed will increase the overall coverage. Since those might be short term wins, the number of docks in high intensity areas has to be increased in the long run. This is regarded to increase the perceived availability of bikes and therefore customer satisfaction which will ultimately help to grow the business.

Data Visualization

San Francisco Bikeshare Dashboard



Appendix

Appendix 1 – SQL Queries and Output Tables

-- Identifying empty stations with all relevant columns

```
SELECT info.station_id, info.name as location, info.short_name, capacity,
status.num_bikes_available, status.num_bikes_disabled, status.is_renting, status.is_installed,
status.is_returning, info.station_geom
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` as info
JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status` as status
on info.station_id = status.station_id
where is_installed = True
and num_bikes_available = 0
order by capacity
```

Row	station_id	location	short_name	capacity	num_bikes_available	num_bikes_disabled	is_renting	is_installed	is_returning	station_geom
1	295	William St at 10th St	SJ-O11	0	0	0	true	true	true	POINT(-121.8759263 37.3327938)
2	146	30th St at San Jose Ave	SF-T21	15	0	4	false	true	false	POINT(-122.4231805 37.7423139)
3	242	Milvia St at Derby St	BK-F7	23	0	0	false	true	false	POINT(-122.269384413958 37.8601245991169)

-- Extracting the stations that are not accepting rentals or returns with all relevant columns

```
SELECT info.station_id, info.name as location, info.short_name, capacity,
status.num_bikes_available, status.num_bikes_disabled, status.num_docks_disabled,
num_docks_available, status.is_renting, status.is_installed, status.is_returning,
status.last_reported, info.station_geom
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` as info
JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status` as status
on info.station_id = status.station_id
where is_installed = True AND ( is_returning = FALSE OR is_renting=FALSE)
order by capacity
```

station_id	location	short_name	capacity	num_bikes_available	num_bikes_disabled	num_docks_disabled	num_docks_available	is_renting	is_installed	is_returning	last_reported	station_geom
385	Woolsey St at Sacramento St	BK-I5	14	4	2	0	8	FALSE	TRUE	FALSE	1585374479	POINT(-122.2781754 37.8505777)
146	30th St at San Jose Ave	SF-T21	15	0	4	0	11	FALSE	TRUE	FALSE	1585378324	POINT(-122.4231805 37.7423139)
476	19th St at William St	SJ-N14-2	15	10	0	0	5	FALSE	TRUE	FALSE	1585375921	POINT(-121.866520643234 37.338468014146)
46	San Antonio Park	OK-L12	15	5	0	0	10	FALSE	TRUE	FALSE	1585355150	POINT(-122.242373228073 37.7901398518536)
213	32nd St at Adeline St	OK-H2	15	4	1	0	10	FALSE	TRUE	FALSE	1583945934	POINT(-122.2811926 37.8238474)
226	International Blvd	OK-L15	15	8	0	0	7	FALSE	TRUE	FALSE	1584184904	POINT(-122.2329915 37.781123)
219	Marston Campbell Park	OK-K3	18	14	0	0	4	FALSE	TRUE	FALSE	1583961013	POINT(-122.2801923 37.8098236)
425	Bird Ave at Willow St	SJ-Q5	19	11	0	0	8	FALSE	TRUE	FALSE	1585327912	POINT(-121.896325349808 37.3112839461174)
158	Shattuck Ave at Telegraph Ave	OK-E4	19	13	0	0	6	FALSE	TRUE	FALSE	1585186134	POINT(-122.2634901 37.8332786)
174	Shattuck Ave at 51st St	OK-D3-2	19	15	0	0	4	FALSE	TRUE	FALSE	1583370556	POINT(-122.2640037 37.8368013)
242	Milvia St at Derby St	BK-F7	23	0	0	22	1	FALSE	TRUE	FALSE	1585321454	POINT(-122.269384413958 37.8601245991169)
203	Webster St at 2nd St	OK-N7	27	20	1	0	6	FALSE	TRUE	FALSE	1585361352	POINT(-122.273969650269 37.795194764386)

-- Extracting the most frequent commuter trips

```
SELECT trips.start_station_name, trips.end_station_name, count(*) as Number_of_Trips,
AVG(trips.duration_sec)/60/60 AS Trips_hours
FROM
  `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips` as trips
  JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status` as station
  ON trips.start_station_id = station.station_id
WHERE start_station_name != end_station_name
Group by  start_station_name, end_station_name
ORDER By 3 Desc, Trips_hours Desc
limit 10;
```

Row	start_station_name	end_station_name	Number_of_Trips	Trips_hours
1	Harry Bridges Plaza (Ferry Building)	Embarcadero at Sansome	9150	0.33000106253794825
2	2nd at Townsend	Harry Bridges Plaza (Ferry Building)	7620	0.16159857101195632
3	Harry Bridges Plaza (Ferry Building)	2nd at Townsend	6888	0.17708978577880977
4	Embarcadero at Sansome	Steuart at Market	6874	0.14194694985937373
5	Embarcadero at Folsom	San Francisco Caltrain (Townsend at 4th)	6351	0.19381488479504583
6	San Francisco Caltrain (Townsend at 4th)	Harry Bridges Plaza (Ferry Building)	6215	0.22138955930991328
7	Steuart at Market	2nd at Townsend	6039	0.1597701514231567
8	Steuart at Market	San Francisco Caltrain (Townsend at 4th)	5959	0.20288900076448335
9	Temporary Transbay Terminal (Howard at Beale)	San Francisco Caltrain (Townsend at 4th)	5796	0.18471752549651035
10	San Francisco Ferry Building (Harry Bridges Plaza)	The Embarcadero at Sansome St	5543	0.36332621725098574

-- Extracting the numbers for the trips and avg duration for commuters and tourists and overall

```
SELECT count(trip_id) as roundtrips, AVG(duration_sec)/60/60 AS Trips_hours
FROM `bigquery-public-data.san_francisco.bikeshare_trips`
WHERE start_station_name = end_station_name;

SELECT count(trip_id) as commute, AVG(duration_sec)/60/60 AS Trips_hours
FROM `bigquery-public-data.san_francisco.bikeshare_trips`
WHERE start_station_name != end_station_name;


SELECT count(trip_id) as total_trips, AVG(duration_sec)/60/60 AS Trips_hours
FROM `bigquery-public-data.san_francisco.bikeshare_trips`;
```

-- Identifying the most popular start station

```
SELECT start_station_name, count(*) as start_station_count,
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
group by 1 order by start_station_count desc
limit 10;
```

-- Identifying the most popular end station with numbers of docks

```
SELECT end_station_name, count(*) as end_station_count, max(num_docks_available) as
max_docks, count(status.num_bikes_available) as count_bikes,
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
join `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` as info
on info.name = start_station_name
join `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status` as status
on info.station_id = status.station_id
group by 1 order by end_station_count desc
limit 10
```

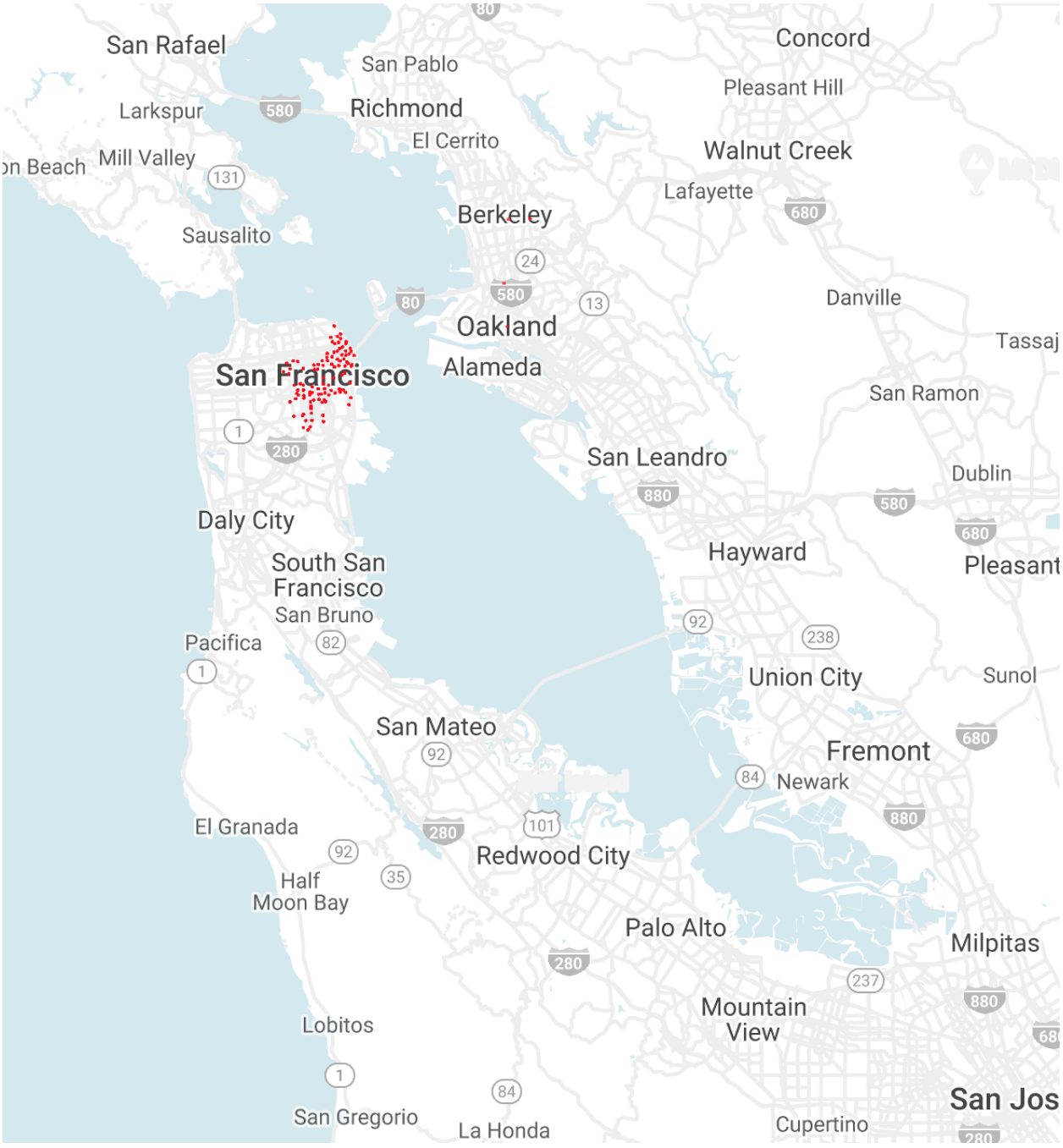
-- Extracting the weekday with the highest number and duration of rides

```
SELECT
  CASE
    WHEN EXTRACT(DAYOFWEEK  FROM start_date) = 1 THEN '2. Sun'
    WHEN EXTRACT(DAYOFWEEK
  FROM
    start_date) = 2 THEN '3. Mon'
    WHEN EXTRACT(DAYOFWEEK  FROM start_date) = 3 THEN '4. Tue'
    WHEN EXTRACT(DAYOFWEEK
  FROM
    start_date) = 4 THEN '5. Wed'
    WHEN EXTRACT(DAYOFWEEK  FROM start_date) = 5 THEN '6. Thu'
    WHEN EXTRACT(DAYOFWEEK
  FROM
    start_date) = 6 THEN '7. Fri'
    WHEN EXTRACT(DAYOFWEEK  FROM start_date) = 7 THEN '1. Sat'
  END AS day_of_week,
  COUNT(*) number_of_rides,
  AVG(duration_sec)/60 AS avg_duration
FROM
  `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
GROUP BY
  day_of_week
ORDER BY
  day_of_week ASC;
```

Row	day_of_week	number_of_rides	avg_duration
1	1. Sat	153147	31.778015131431538
2	2. Sun	132374	31.450000881340962
3	3. Mon	322673	14.15313501077963
4	4. Tue	349190	13.577150357493887
5	5. Wed	345075	13.57261503537879
6	6. Thu	335488	14.120670138226739
7	7. Fri	309472	16.01796716127258

-- Mapping out the most popular start and end stations (Map on next page – note)

```
SELECT start_station_geom, end_station_geom
FROM
  `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
WHERE start_station_name = 'San Francisco Caltrain (Townsend at 4th)'
or start_station_name =  'San Francisco Caltrain 2 (330 Townsend)'
or start_station_name = 'Harry Bridges Plaza (Ferry Building)'
or start_station_name = 'Embarcadero at Sansome'
or start_station_name = '2nd at Townsend'
or start_station_name = 'Temporary Transbay Terminal (Howard at Beale)'
or start_station_name = 'Steuart at Market'
or start_station_name = 'Market at Sansome'
or start_station_name = 'Townsend at 7th'
or start_station_name = 'Market at 10th'
or end_station_name = 'Powell St BART Station (Market St at 5th St)'
or end_station_name = 'Montgomery St BART Station (Market St at 2nd St)'
or end_station_name = 'Powell St BART Station (Market St at 4th St)'
or end_station_name = 'San Francisco Caltrain Station 2  (Townsend St at 4th St)'
or end_station_name = 'Berry St at 4th St'
```


San Rafael
San Pablo
Concord
Pleasant Hill
Larkspur
580
Richmond
El Cerrito
Walnut Creek
Mill Valley
131
Lafayette
680
Berkeley
Sausalito
24
580
80
13
Danville
Oakland
Tassaj
San Francisco
Alameda
San Ramon
1
280
San Leandro
Dublin
880
580
Daly City
680
Hayward
South San
Francisco
Pleasant
San Bruno
92
238
Pacifica
82
1
Union City
Sunol
San Mateo
680
92
Fremont
84
Newark
880
El Granada
280
101
92
Redwood City
Half
Moon Bay
35
Palo Alto
Milpitas
280
237
880
Mountain
View
Lobitos
1
84
San Gregorio
La Honda
Cupertino
San Jos
280

-- Extracting the numbers of subscription types commuters and tourists and overall

```
SELECT count(*) as Subscriber
FROM `bigquery-public-data.san_francisco.bikeshare_trips`
WHERE subscriber_type = "Subscriber";

SELECT count(*) as Customer
FROM `bigquery-public-data.san_francisco.bikeshare_trips`
WHERE subscriber_type = "Customer";

SELECT count(*) as Total
FROM `bigquery-public-data.san_francisco.bikeshare_trips`;
```

-- Identifying hours per weekday

```
SELECT
  EXTRACT(HOUR FROM start_date ) AS hour,
  COUNT(*) as freq_trips,
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
WHERE  EXTRACT(DAYOFWEEK FROM start_date) = 2 or EXTRACT(DAYOFWEEK
FROM start_date) = 3
or EXTRACT(DAYOFWEEK FROM start_date) = 4
or EXTRACT(DAYOFWEEK FROM start_date) = 5
or EXTRACT(DAYOFWEEK FROM start_date) = 6
GROUP BY hour
ORDER BY freq_trips DESC
```

-- Identifying hours per weekend

```
SELECT
  EXTRACT(HOUR FROM start_date ) AS hour,
  COUNT(*) as freq_trips,
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
WHERE  EXTRACT(DAYOFWEEK FROM start_date) = 7 or EXTRACT(DAYOFWEEK
FROM start_date) = 1
GROUP BY hour
ORDER BY freq_trips DESC
```

-- Identifying station metrics: At capacity, Empty Station and Out of Service

```
SELECT (
  select count(*)
  FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status` as status
  join `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` as info
  on info.station_id = status.station_id
  where status.num_bikes_available = info.capacity)  as at_capacity,
  (select count(*)
  FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status`
  where num_bikes_available = 0) as station_empty,
  (select count(*)
  FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status`
  where is_returning = False  and is_renting = False) as out_of_service
```

-- Bike usage per hour

```
SELECT bike_number, ROUND(SUM(duration_sec)/3600,1) as hours_usage
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
GROUP BY bike_number
HAVING bike_number is NOT Null
ORDER BY hours_usage DESC
```

-- Station Count

```
SELECT num_bikes_available, count(*) as station_count FROM `bigquery-public-
data.san_francisco_bikeshare.bikeshare_station_status` group by 1 order by 1
```

-- Start and End Station Count

```
SELECT start_station_name, count(*) as start_station_count,
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
group by 1 order by start_station_count desc
limit 1;
```

```
SELECT end_station_name, count(*) as end_station_count,
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
group by 1 order by end_station_count desc
limit 1;
```

-- Station Fill

```
SELECT name, (num_bikes_available/capacity)*100 as fill_rate,
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_status` as station
join `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` as info
on station.station_id = info.station_id
where capacity > 0
```

-- Min/Max Trip duration

```
SELECT ROUND(MIN(duration_sec)/60,2) as min_trip_duration
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
```

```
SELECT ROUND(MAX(duration_sec)/60,2) as max_trip_duration
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
```

-- Usage per Day Count

```
SELECT EXTRACT(DAY FROM start_date ) AS Day, COUNT(EXTRACT(Day FROM
start_date )) as Count
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_station_info` as info
JOIN `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips` as trips
ON info.name = trips.start_station_name
GROUP BY Day
ORDER BY Day ASC
```

-- Hour weekday

```
select EXTRACT(HOUR FROM start_date ) AS Hour, EXTRACT(WEEKDAY FROM
start_date)
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips`
WHERE  EXTRACT(HOUR FROM start_date ) IS NOT NULL
GROUP BY hour, start_station_name
ORDER BY hour DESC;
```