Final Project Phase I

data source:

Inside Airbnb

covid total cases in SF by neighborhood

covid cases in last 30 days in SF by neighborhood

sentiment analysis

(1) Who are your team members

Nanxi Ye and Linghao Jin

(2) Target domain

Airbnb listing information in San Francisco, CA combined with SF neighborhood data including covid cases, demographic, income level etc.

(3) List of questions

```
 (1) Show the listings start hosting in 2008 with review score over 95.
(2) Show the name, host_acceptance_time and host_response_rate of the hosts who have more than 3 listings currently.
(3) Show the listings by hosts who is related to UCSF in their host_about.
(4) Show host_response_time and host_response_rate of those who have different host_location and listing location.
(5) Show listings that offers entire home/apartment with more than 3 accommodations and coffee maker in neighborhood Financial Distr
(6) Show the average price of listings in Financial District.
(7) Show the average rating of listings with price less than $100, between $100 and $200, between $200 and $300 and above $300.
(8) Show listings from hosts who were originally not from San Francisco. (from host_about)
(9) Show the average rental price of each neighborhood in San Francisco.
(10) Show the entire houses that are 20 miles (Euclidean distance) away from my current location.
(11) Show the percentage of reviewers have reviewed multiple listings under 50 in Airbnb.
(12) Show the review rating score stats(average, max, min) or listings group by property type, order by average review rating score.
(13) Show the listings that are instant bookable, have over 50 reviews, rated above 90 and have over 30 days available in a year.
(14) Show all the listings that owned by the same superhost who responses within an hour and are verified by government id.
(15) Show the name of reviewers who reviewed most each year.
(16) Show the listings that have good views (name including view) that are rated highest in each neighborhood.
```

(4) Relational data model

```
 DROP TABLE Listing;
CREATE TABLE Listing(
    id INT NOT NULL,
    listing_url VARCHAR(100) NOT NULL,
    name VARCHAR(100) NOT NULL,
    description VARCHAR(2000) NOT NULL,
    neighborhood_overview VARCHAR(2000) NOT NULL,
    picture_url VARCHAR(100) NOT NULL,
    host_id INT NOT NULL,
    neighbourhood VARCHAR(20) NOT NULL,
    neighbourhood_cleansed VARCHAR(20) NOT NULL, -- Neighborhood.id?
    latitude DECIMAL(8, 5) NOT NULL,
    longitude DECIMAL(8, 5) NOT NULL,
    property_type VARCHAR(20) NOT NULL,
    room_type VARCHAR(20) NOT NULL,
    accommodates INT NOT NULL,
    bathrooms_text VARCHAR(20) NOT NULL,
    bedrooms INT NOT NULL,
    beds INT NOT NULL,
    amenities VARCHAR(1000) NOT NULL,
    price DECIMAL NOT NULL,
    minimum_nights INT NOT NULL,
    maximum_nights INT NOT NULL,
    availability_365 INT NOT NULL,
    number_of_reviews INT NOT NULL,
    first_review DATE NOT NULL,
    last_review DATE NOT NULL,
    review_scores_rating INT NOT NULL,
    review_scores_accuracy INT NOT NULL,
    review_scores_cleanliness INT NOT NULL,
    review_scores_checkin INT NOT NULL,
    review_scores_communication INT NOT NULL,
    review_scores_location INT NOT NULL,
    review_scores_value INT NOT NULL,
    instant_bookable BOOLEAN NOT NULL,
    calculated_host_listings_count INT NOT NULL,
    calculated_host_listings_count_entire_homes INT NOT NULL,
    calculated_host_listings_count_private_rooms INT NOT NULL,
    calculated_host_listings_count_shared_rooms INT NOT NULL,
    reviews_per_month DECIMAL(3, 2) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (host_id) REFERENCES Host.host_id
);

DROP TABLE Host;
CREATE TABLE Host(
    host_id INT NOT NULL,
    host_url VARCHAR(100) NOT NULL,
    host_name VARCHAR(20) NOT NULL,
    host_since VARCHAR(10)NOT NULL,
    host_location VARCHAR(20) NOT NULL, -- Neighborhood.id?
    host_about VARCHAR(2000)NOT NULL,
    host_response_time VARCHAR(50),
    host_response_rate INT,
    host_acceptance_rate INT NOT NULL,
    host_is_superhost VARCHAR(1) NOT NULL,
    host_thumbnail_url VARCHAR(100) NOT NULL,
    host_picture_url VARCHAR(100) NOT NULL,
    host_neighbourhood VARCHAR(20) NOT NULL,
    host_listings_count INT NOT NULL,
    host_total_listings_count INT NOT NULL,
    host_verifications VARCHAR(100) NOT NULL,
    host_has_profile_pic VARCHAR(1) NOT NULL,
    host_identity_verified VARCHAR(1) NOT NULL,
    PRIMARY KEY (host_id)
);

DROP TABLE Neighbourhood;
CREATE TABLE Neighbourhood(
    id INT NOT NULL,
    name VARCHAR(20) NOT NULL,
    city VARCHAR(20) NOT NULL,
    state VARCHAR(20) NOT NULL,
    country VARCHAR(20) NOT NULL,
    covid_case INT
    PRIMARY KEy (id)
);

DROP TABLE Review;
CREATE TABLE Review(
    id INT NOT NULL,
    listing_id INT NOT NULL,
    date DATE NOT NULL,
    reviewer_id INT NOT NULL,
    comments VARCHAR(2000),
    sentiment_score INT,
    PRIMARY KEY (id),
    FOREIGN KEY (listing_id) REFERENCES Listing.id,
    FOREIGN KEY (reviewer_id) REFERENCES Reviewer.reviewer_id
);

DROP TABLE Reviewer;
CREATE TABLE Reviewer(
    reviewer_id INT NOT NULL,
    reviewer_name VARCHAR(20) NOT NULL,
    PRIMARY KEY (reviewer_id)
    PRIMARY KEY (reviewer_id)
);
```

(5) SQL statements for representative sample of target queries

```
 /* Show the name, host_acceptance_time and host_response_rate of the hosts who have more than 3 listings currently. */
SELECT host_name, host_acceptance_time, host_response_rate
FROM Hosts AS H
WHERE EXISTS (
    SELECT *
    FROM listings AS L1, Listings AS L2, Listings AS L3
    WHERE L1.host_id = L2.host_id AND
    L2.host_id = L3.host_id AND
    H.host_id = L1.host_id AND
    L1.id <> L2.id AND
    L2.id <> L3.id AND
    L1.id <> L3.id);


/* Show host_response_time and host_response_rate of those who have different host_location and listing location. */
SELECT host_response_time, host_response_rate
FROM Hosts AS H, Neighborhood N1, Listings as L, Neighborhood N2
WHERE H.id = L.host_id AND
    N1.id = H.neighborhood_id AND
    N2.id = L.neighborhood_id AND
    N1.id <> N2.id;


/* Show the average rental price of each neighborhood in San Francisco. */
SELECT N.name, AVG(price)
FROM Listings AS L
JOIN Neighborhood AS N ON L.neighborhood_id = N.id
WHERE N.city = 'San Francisco'
GROUP BY N.id;


/* Show the percentage of reviewers have reviewed over 10 listings in Airbnb. */
SELECT COUNT(distinct reviewer_id) / r2_ids
FROM Reviews, (SELECT COUNT(distinct reviewer_id) AS r2_ids
        FROM Reviews
        GROUP BY reviewer_id
        HAVING COUNT(listing_id) > 10) AS R2;


/* Show the listings that are instant bookable, have over 50 reviews, rated above 90 and have over 30 days available in a year. */
SELECT distinct id
FROM Listings
JOIN Reviews ON Listings.id = Reviews.listing_id
WHERE instant_bookable = 't' AND
    review_scores_rating > 90 AND
    availability_365 > 30
HAVING COUNT(Reviews.id) > 50;


/* Show the name of reviewers who reviewed most each year. */
SELECT reviewer_name
FROM Reviews
GROUP BY reviewer_id
HAVING COUNT(id) = (SELECT COUNT(id)
            FROM Reviews
            GROUP BY reviewer_id);

/* Show the listings that have good views (name including view) that are rated highest in each neighborhood. */
SELECT id, name
FROM Listings
WHERE name LIKE %View% OR
WHERE name LIKE %view% OR
    name LIKE %view% AND
GROUP BY neighborhood_id
ORDER BY review_scores_rating DESC
LIMIT 1;


/* Show the review rating score stats(average, max, min) or listings group by property type, order by average review rating score. *
SELECT property_type, AVG(review_scores_rating), MAX(review_scores_rating), MIN(review_scores_rating)
FROM Listings
GROUP BY property_type,
ORDER BY AVG(review_scores_rating)
```

(6) How to load the database with values

We plan to use sql to bulk load the data from csv files to sql. There will be issues related to datatype conversion (such as format of date, 95% to 95), duplicate tuples, NULL values and data parsing problems ("", escape /, line break in quoted text). An example of how we load data is:

```
 load data local infile 'listings2.csv' into  table listings
fields terminated by ',' optionally enclosed by '"'
(id, name, description...);
```

(7) Result of project

We plan to deploy a web application that has a search feature, which would be useful for users to search for specific listings they look for on Airbnb. The specific implementation is still under discussion. At this stage we have some prelimiary thoughts on such search website. In addition to traditional search on Airbnb listing (location, date, avaliability), we want the search engine to allow user to search for additional information in specific neighborhoods, such as covid cases(total/last 30 days), listing density, demongraphic, living cost, income levels etc. The users could also do a "vague" search. For example, a user can search for "sunset view" which would not be included as part of amentities in the database, but could be mentioned in listing description or reviewers comments. We want the search results beyond the scope of simple queries so that they provide further understanding of the data.

(8) Topics of database design

Some potential topics include data mining, complex data extraction issues from online sources and some fields in natural language interfaces. We will conduct some data mining practice on our dataset to generate useful analysis for users. One example of using natural language related knowledge in data mining is to categorize/sort review comments by their sentiment scores, which is evaluated based on how postive/negative some adjetive words used in the comments. This would be provide an additional way to reflect how positive a user evalute its experience other than old-fashioned numeric rating mechanism.