Milestone 3:

Aakash Jajoo, Karan Jaisingh, Kush Pandey, Yathushan Nadanapathan

Credentials:

Instructions: Connect to the Database On MySQL Workbench with the Credentials Below
Hostname: cis550-proj.clcqtdmdkob4.us-east-2.rds.amazonaws.com
Username: guest_user
Password: AJ1kj2KP3yn4

Small Queries:

Query 1 - Avg price of an airbnb in each neighbourhood of a particular city (city can be the input by the user)

```
WITH tab1 AS
        (SELECT name, neighbourhood, price
        FROM Airbnb
        WHERE city='Los Angeles')
SELECT neighbourhood, AVG(price) as avg_price
FROM tab1
GROUP BY neighbourhood
ORDER BY avg_price;
```

Query 2 - Select room based on user preferences on city, price, and min. nights

```
SELECT name, neighbourhood, host_name, room_type, price, city
FROM Airbnb
WHERE city='Los Angeles' AND price < 120 AND minimum_nights  < 2;
```

Query 3 (Party complaints in a particular region of NYC. In this case, I have limited the latitude to Midtown Manhattan using longitude and latitude)

```
SELECT *
FROM Parties
WHERE Borough = 'MANHATTAN' AND
latitude >= 40.74 AND latitude <= 40.75 AND
longitude >=-74 AND longitude <= -73.8;
```

Query 4 (Number of noise complaints in a particular borough of NYC)

```
WITH tab1 AS
        (SELECT COUNT(*) AS num, Borough
        FROM Bars
        GROUP BY Borough),
tab2 AS
        (SELECT COUNT(*) AS num1, Borough
        FROM Parties
        GROUP BY Borough)
SELECT tab1.num AS bar_complaints, tab2.num1 AS party_complaints, tab1.num + tab2.num1
AS total_complaints, tab1.Borough
FROM tab1, tab2
WHERE tab1.Borough = tab2.Borough;
```

Query 5 (Average minimum nights that an Airbnb needs to be booked for in a particular neighbourhood)

```
SELECT AVG(minimum_nights) AS avg_min_nights, neighbourhood, city
FROM Airbnb
WHERE city = 'New York City'
GROUP BY neighbourhood;
```

Query 6 (Look at experienced hosts based on the number of reviews they got on avg per property, as well as the number of listings they have in a particular city)

```
WITH tab1 AS
        (SELECT host_id, host_name, AVG(number_of_reviews) AS avg_reviews
        FROM Airbnb
        WHERE city = 'Denver' AND calculated_host_listings_count > 2
        GROUP BY host_id)
SELECT *
FROM tab1
WHERE avg_reviews > 25;
```

Long Queries:

Query 1 (This query returns the 20 closest Airbnb listings to the location with the most party reports in Manhattan. It is useful if someone wants to stay at an Airbnb surrounded by a partying city.)

```
WITH tab1 AS (
        SELECT Location_type, Incident_ZIP, City, Borough, latitude, longitude, COUNT(*) as NumParties
        FROM Parties
        WHERE Borough = 'MANHATTAN'
        GROUP BY Location_type, Incident_ZIP, City, Borough, latitude, longitude
        ORDER BY NumParties DESC
        LIMIT 1),
tab2 AS
        (SELECT A.name, A.neighbourhood, A.latitude, A.longitude, A.room_type, A.price, A.number_of_reviews
        FROM Airbnb as A),
tab3 AS
        (SELECT A.name, A.neighbourhood, A.latitude, A.longitude, A.room_type, A.price, A.number_of_reviews, B.latitude as Blat, B.longitude as Blong
        FROM tab1 as B,tab2 as A)
SELECT name, neighbourhood, latitude, longitude, room_type, price, number_of_reviews, (latitude - Blat)*(latitude - Blat) + (longitude - Blong)*(longitude - Blong) as distance
FROM tab3
ORDER BY distance ASC
LIMIT 20;
```

Query 2 (This query returns the 250 closest Airbnbs to any of the 20 locations with the least amount of party+bar complaints. It is useful if someone wants to visit an Airbnb that is in a more tranquil part of Manhattan.

```
WITH tab1 AS (
        SELECT Location_type, Incident_ZIP, City, Borough, latitude, longitude, COUNT(*) as NumParties
        FROM Parties
        WHERE Borough = 'MANHATTAN'
        GROUP BY Location_type, Incident_ZIP, City, Borough, latitude, longitude
        ORDER BY NumParties DESC),
tab2 AS (
        SELECT Bars.id, tab1.City, tab1.Borough, tab1.latitude, tab1.longitude, MIN(num_calls)
        as numberCalls, SUM(NumParties) as numParties, MIN(num_calls) + SUM(NumParties)
        as issues
        FROM Bars JOIN tab1 ON Bars.latitude = tab1.latitude AND Bars.longitude =
        tab1.longitude
        GROUP BY Bars.id, tab1.city, tab1.Borough, tab1.latitude, tab1.longitude
        ORDER by issues ASC
        LIMIT 25),
tab3 AS (
        SELECT A.name, A.neighbourhood, A.latitude, A.longitude,  A.room_type, A.price,
        A.number_of_reviews, B.latitude as Blat, B.longitude as Blong, issues
        FROM tab2 as B,Airbnb as A
)
SELECT name, neighbourhood, latitude, longitude, room_type, price, number_of_reviews,
(latitude - Blat)*(latitude - Blat) + (longitude - Blong)*(longitude - Blong) as distance, issues
FROM tab3
ORDER BY distance ASC
LIMIT 250;
```

Query 3 (This query contains a correlated subquery and it returns all Airbnbs in NYC that have 50 Club/Bars/Restaurants within approximately .16 square miles (by roughly converting degrees of latitude and longitude to miles). Can easily adjust the number of establishments and the distance.

```
With tab1 AS
    (SELECT *
    FROM Airbnb
    WHERE city = "New York City")
SELECT *
FROM tab1 as A
WHERE EXISTS (
    SELECT COUNT(*) as count FROM Bars B
    WHERE ABS(A.latitude - B.latitude) < 0.003 AND ABS(A.longitude - B.longitude) <
    0.003
    GROUP BY Location_type
    HAVING COUNT(*) >= 50);
```

Query 4 (This query takes as input the latitude and longitude of a desired bar and returns all Airbnb listings within .16 square miles (same logic from previous query for distance) and the listing must be for those who want a 1 bedroom rental (1 bedroom can be adjusted in the future). The output is sorted by price in ascending order.

```
WITH tab1 AS
        (SELECT * FROM Bars
        WHERE latitude = 40.84711 AND longitude = -73.93818),
tab2 AS
        (SELECT A.name, A.neighbourhood, A.latitude, A.longitude, A.room_type, A.price,
        A.number_of_reviews
         FROM Airbnb A, tab1
        WHERE A.city = "New York City" AND
        ABS(A.latitude - tab1.latitude) < 0.003 AND ABS(A.longitude - tab1.longitude) < 0.003)
SELECT * FROM tab2
WHERE name like '%1 Bedroom%' OR name like '%One Bedroom%'
ORDER BY price;
```