Milestone 3 Business Insight

Group 3 (Liao Wen, Zhuocai Li, Houming Leng, Jingyi Mao, Kejian Tong)

Ten Question can be answered with our data and corresponding SQL:

1.List the information of all the housing which have no less than 5 NEU students.

```
SELECT T_STUDENTS_NEU.HOUSING_ID, T_STUDENTS_NEU.NEU_STUDENTS_COUNT,
Housing.Zipcode, Housing.Name, Housing.Address, Housing.RentalPrice
FROM
        (SELECT Users.HousingId AS HOUSING_ID, COUNT(*) AS NEU_STUDENTS_COUNT
         FROM Users
         INNER JOIN Students
         ON Users.UserName = Students.UserName
         WHERE Students.school = 'NEU' AND Users.isAuthenticatedResident = TRUE
         GROUP BY Users.HousingId
         HAVING NEU_STUDENTS_COUNT >= 5)
AS T_STUDENTS_NEU
INNER JOIN Housing
ON T_STUDENTS_NEU.HOUSING_ID = housing.HousingId;
```

HOUSING_ID	NEU_STUDENTS_COUNT	Zipcode	Name	Address	RentalPrice
12	6	98107	Commons at Ballard	5621 22nd Ave NW, Seattle	2795
22	5	98109	Union SLU	905 Dexter Ave N, Seattle	1955

2.list top 10 average rated housing then order by price ascending.

```
SELECT * FROM
        (SELECT Housing.Name AS Name, Housing.Address AS Address, Housing.Zipcode AS
Zipcode, Housing.RentalPrice AS RentalPrice, AVG(Reviews.Rating) AS AVG_RATING
        FROM Reviews
        INNER JOIN Housing
        ON Reviews.HousingId = Housing.HousingId
        GROUP BY Reviews.HousingId
        ORDER BY AVG_RATING DESC
        LIMIT 10)
AS TOP_10_RATED
ORDER BY TOP_10_RATED.RentalPrice ASC;
```

Name	Address	Zipcode	RentalPrice	AVG_RATING
Novo	6105 Roosevelt Way NE, Seattle	98115	1250	4.00000
Leilani Apartment Homes	10215 Greenwood Ave N, Seattle	98133	1575	5.00000
Avenue5 Residential	2401 3rd Ave, Seattle	98121	1871	5.00000
The Shoresmith	1170 Republican St, Seattle	98109	1910	4.50000
Union SLU	905 Dexter Ave N, Seattle	98109	1955	4.50000
Cirrus	2030 8th Ave, Seattle	98121	2020	4.50000
The Martin	2105 5th Ave, Seattle	98121	2135	5.00000
Lawrence Lofts	1818 E Madison St, Seattle	98122	2175	4.00000
Neptune	912 Dexter Ave N, Seattle	98109	2240	5.00000
Arrive Seattle	2116 4th Ave, Seattle	98121	2890	5.00000

3. For all the houses which have private schools within one mile, find the top 10 ones that have least crime cases from this year within one mile.

```
SELECT AboutSchool.HousingId, AboutCrime.CaseCount FROM (
        SELECT HousingId,  COUNT(*) AS PrivateSchoolCount FROM Housing
        CROSS JOIN Schools
        WHERE Schools.Longitude != 0 AND Schools.Latitude != 0 AND
                (SQRT(POWER(69.1 * (Schools.Latitude - Housing.Latitude),  2)
                + POWER(69.1 * ( Housing.Longitude  - Schools.Longitude )
                * COS(Schools.Latitude / 57.3), 2))) <= 1
        AND Schools.Type = 'Private'
        GROUP BY (HousingId)
        ORDER BY HousingId
) AS AboutSchool
INNER JOIN (
        SELECT HousingId,  COUNT(*) AS CaseCount FROM Housing
        CROSS JOIN CrimeCases
        WHERE CrimeCases.Longitude != 0 AND CrimeCases.Latitude != 0 AND
                (SQRT(POWER(69.1 * (CrimeCases.Latitude - Housing.Latitude),  2)
                + POWER(69.1 * ( Housing.Longitude  - CrimeCases.Longitude )
                * COS(CrimeCases.Latitude / 57.3), 2))) <= 1
        AND CrimeCases.ReportDate > '2020-12-31'
        GROUP BY (HousingId)
        ORDER BY CaseCount
) AS AboutCrime
ON AboutSchool.HousingId = AboutCrime.HousingId
ORDER BY AboutCrime.CaseCount
LIMIT 10;
```

HousingId	CaseCount
13	1017
10	1044
3	1247
6	1294
2	1477
1	1552
7	1704
15	2531
4	2916
25	3998

4. List housings of which rental price is no more than 1800 and rating no less than 4 with a least 1 public school located within five miles.

```
SELECT Housing.HousingId, AvgReview.AvgRating,  Housing.RentalPrice FROM Housing
INNER JOIN (
        SELECT HousingId, AVG(Rating) AS AvgRating FROM Reviews
        GROUP BY HousingId
) AS AvgReview ON AvgReview.HousingId = Housing.HousingId
INNER JOIN (
        SELECT Housing.HousingId, COUNT(*) AS NumNearbyPublicSchools FROM Schools
        CROSS JOIN Housing
        WHERE (SQRT(POWER(69.1 * ( Schools.Latitude - Housing.Latitude),  2)
        + POWER(69.1 * ( Schools.Longitude  - Housing.Longitude )
        * COS(Schools.Latitude / 57.3), 2)) ) <= 5 AND Schools.Type = 'Public'
        GROUP BY Housing.HousingId
) As SCH ON SCH.HousingId = Housing.HousingId
WHERE SCH.NumNearbyPublicSchools >= 1 AND
AvgReview.AvgRating >= 4 AND Housing.RentalPrice <= 1800;
```

HousingId	AvgRating	RentalPrice
3	4.00000	1395
6	5.00000	1575
10	4.00000	1350
13	4.00000	1250

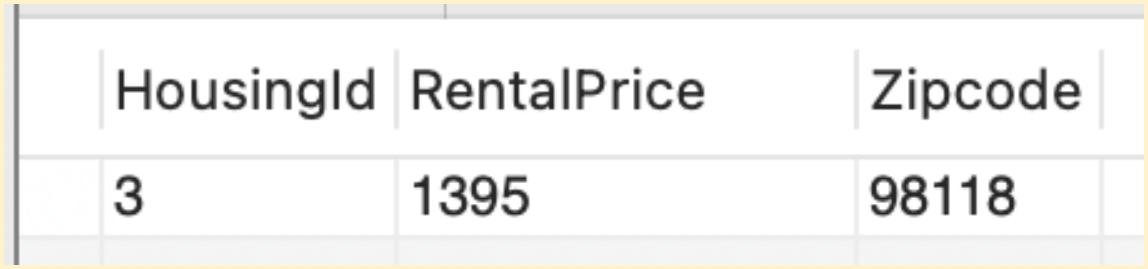
5. List housings with no less than 5 restaurants within 1 miles.

```
SELECT Housing.HousingId, COUNT(*) AS NumNearbyRestaurants FROM Restaurants
CROSS JOIN Housing
WHERE (SQRT(POWER(69.1 * ( Restaurants.Latitude - Housing.Latitude),  2)
        + POWER(69.1 * ( Housing.Longitude  - Restaurants.Longitude )
        * COS(Restaurants.Latitude / 57.3), 2)) ) <= 1
GROUP BY Housing.HousingId
HAVING NumNearbyRestaurants >= 50;
```

HousingId	NumNearbyRestaurants
1	81
4	113
5	289
7	84
8	160
9	211
11	240
14	127
15	80
16	281
17	300
18	242
19	328
20	127
21	171
22	128
23	346
24	179
25	142

6. Filter out the cheapest housing located in the zipcode district with least crime cases.

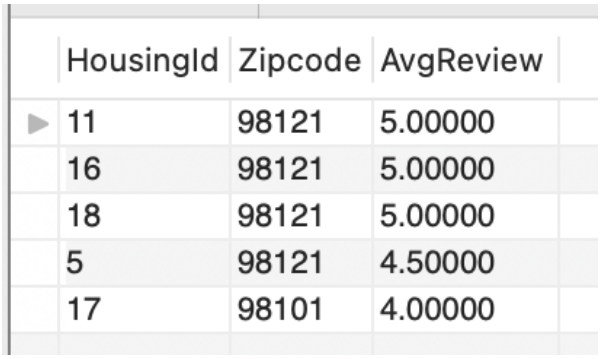
```
SELECT HousingId, RentalPrice, C.Zipcode FROM Housing
INNER JOIN (
        SELECT Zipcode, COUNT(*) AS CNT  from CrimeCases
        WHERE Zipcode != 0
        GROUP BY Zipcode
        ORDER BY CNT
        LIMIT 1
) AS C ON C.Zipcode = Housing.Zipcode
ORDER BY RentalPrice
LIMIT 1;
```



HousingId	RentalPrice	Zipcode
3	1395	98118

7. Filter out the top 5 rated housings within the zipcode from downtown (98121, 98101, 98104)

```
SELECT DTHousing.HousingId, DTHousing.Zipcode, AVG(Reviews.Rating) AS AvgReview
FROM (
        SELECT * FROM Housing
        WHERE Zipcode in (98121, 98101, 98104)
) AS DTHousing
INNER JOIN Reviews ON Reviews.HousingId = DTHousing.HousingId
GROUP BY DTHousing.HousingId
ORDER BY AvgReview DESC
LIMIT 5;
```



HousingId	Zipcode	AvgReview
11	98121	5.00000
16	98121	5.00000
18	98121	5.00000
5	98121	4.50000
17	98101	4.00000

8.Among all the houses that have residents from FB, which one is the cheapest?

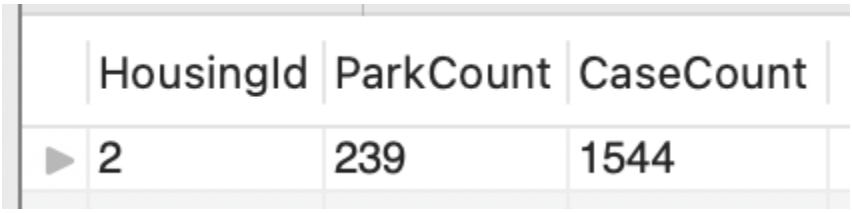
```
SELECT Housing.HousingId, RentalPrice FROM Housing
INNER JOIN (
        SELECT DISTINCT HousingId FROM Professionals
        INNER JOIN User ON Professionals.UserName = Users.UserName
        WHERE Company = 'Facebook'
) AS UsersFromFB
ON Housing.HousingId = UsersFromFB.HousingId
ORDER BY RentalPrice
LIMIT 1;
```

HousingId	RentalPrice
10	1350

9.list the houses that are both in the top 10 in terms of park numbers within 4 miles and in the top 10 in terms of safety within 1 miles from this year.

```
SELECT AboutPark.HousingId, AboutPark.ParkCount, AboutCrime.CaseCount FROM (
        SELECT HousingId,  COUNT(*) AS ParkCount FROM Housing
        CROSS JOIN Parks
        WHERE Parks.Longitude != 0 AND Parks.Latitude != 0 AND
                (SQRT(POWER(69.1 * (Parks.Latitude - Housing.Latitude),  2)
                + POWER(69.1 * ( Housing.Longitude  - Parks.Longitude )
                * COS(Parks.Latitude / 57.3), 2))) < 4
        GROUP BY (HousingId)
        ORDER BY ParkCount DESC
   LIMIT 10
) AS AboutPark
INNER JOIN (
        SELECT HousingId,  COUNT(*) AS CaseCount FROM Housing
        CROSS JOIN CrimeCases
        WHERE CrimeCases.Longitude != 0 AND CrimeCases.Latitude != 0 AND
                (SQRT(POWER(69.1 * (CrimeCases.Latitude - Housing.Latitude),  2)
                + POWER(69.1 * ( Housing.Longitude  - CrimeCases.Longitude )
                * COS(CrimeCases.Latitude / 57.3), 2))) < 1
                AND CrimeCases.ReportDate > '2020-12-21'
        GROUP BY (HousingId)
        ORDER BY CaseCount
   LIMIT 10
```

```
) AS AboutCrime
ON AboutPark.HousingId = AboutCrime.HousingId;
```



HousingId	ParkCount	CaseCount
2	239	1544

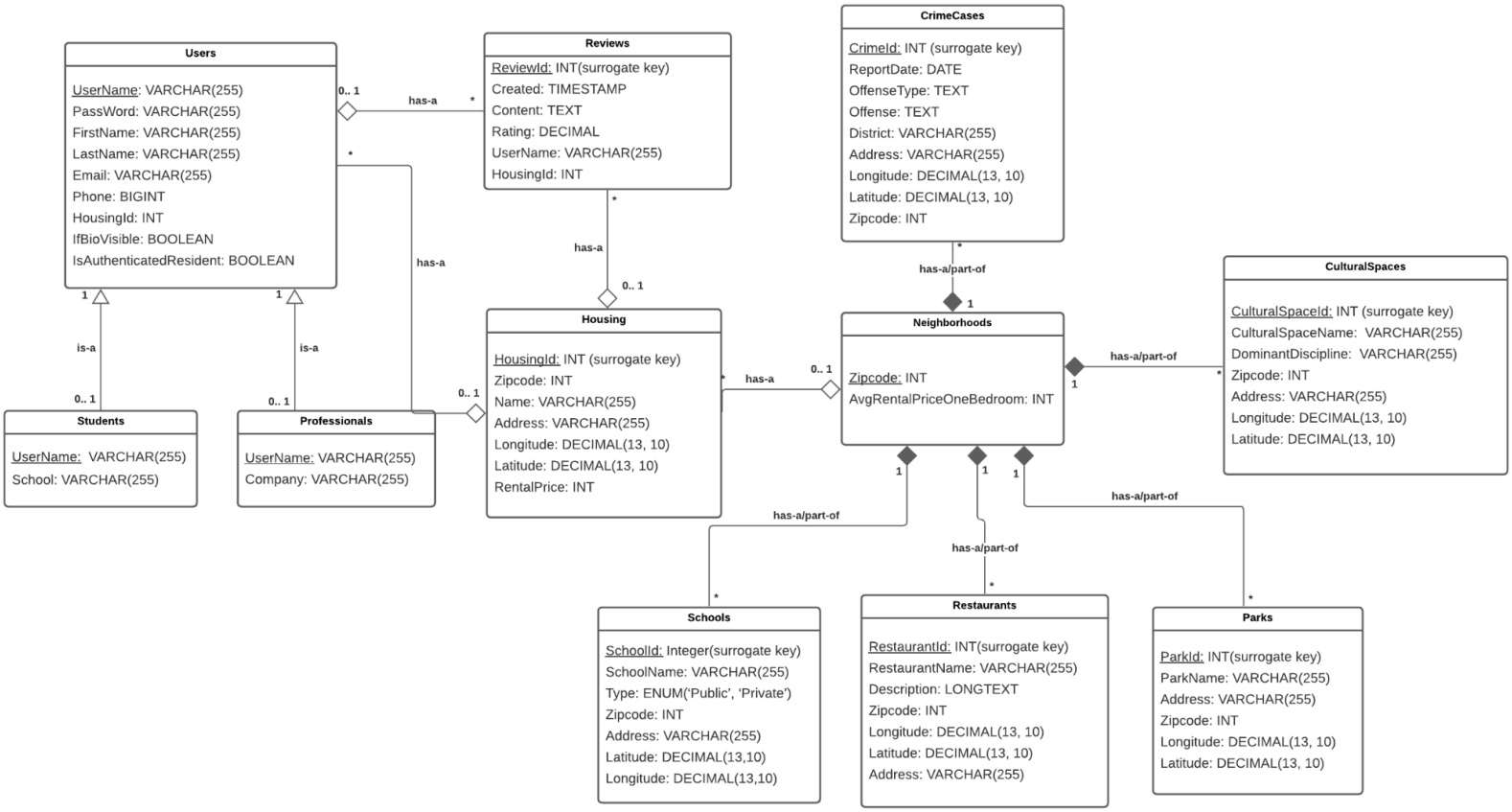
10. list the number of parks and culturespaces that are within 1 miles of each house.

```
SELECT b.HousingId, b.numCultureSpaces, COUNT(*) AS numParks FROM Parks
CROSS JOIN (
	SELECT HousingId, Housing.Latitude, Housing.Longitude, COUNT(*) AS
numCultureSpaces FROM CulturalSpaces
	CROSS JOIN Housing
	WHERE (SQRT(POWER(69.1 * ( CulturalSpaces .Latitude - Housing.Latitude),  2)
		+ POWER(69.1 * ( CulturalSpaces.Longitude  - Housing.Longitude )
		* COS(CulturalSpaces.Latitude / 57.3), 2)) ) <= 1
	GROUP BY HousingId
) AS b
WHERE (SQRT(POWER(69.1 * ( Parks.Latitude - b.Latitude),  2)
	+ POWER(69.1 * ( Parks.Longitude  - b.Longitude )
	* COS(Parks.Latitude / 57.3), 2)) ) <= 1
GROUP BY b.HousingId
ORDER BY b.numCultureSpaces DESC, numParks DESC;
```

	HousingId	numCultureSpac...	numParks
▶	23	315	35
	19	305	36
	16	242	28
	5	230	29
	17	229	32
	11	179	25
	18	179	25
	9	165	29
	21	126	32
	8	122	32
	25	109	34
	24	108	38
	22	100	36
	14	98	34
	20	98	34
	15	70	21
	4	56	31
	1	49	9
	3	35	8
	7	31	12
	6	27	6
	13	18	22
	2	13	31
	10	10	10
	12	1	4

The Housing class in our UML diagram has been changed. We added the information of the housing name and housing address compared with the previous diagram.

Below is the updated UML diagram:



Full creation/insertion sql:

```
CREATE SCHEMA IF NOT EXISTS SeattlerHub;
USE SeattlerHub;

DROP TABLE IF EXISTS CrimeCases;
DROP TABLE IF EXISTS Schools;
DROP TABLE IF EXISTS Restaurants;
DROP TABLE IF EXISTS CulturalSpaces;
DROP TABLE IF EXISTS Parks;
DROP TABLE IF EXISTS Reviews;
DROP TABLE IF EXISTS Students;
DROP TABLE IF EXISTS Professionals;
DROP TABLE IF EXISTS Users;
DROP TABLE IF EXISTS Housing;
DROP TABLE IF EXISTS Neighborhoods;


CREATE TABLE Neighborhoods (
    Zipcode INT,
    AvgRentalPriceOneBedroom INT,
    CONSTRAINT pk_Zipcode PRIMARY KEY(Zipcode)
);

CREATE TABLE Housing (
    HousingId INT AUTO_INCREMENT,
    Zipcode INT,
    Name VARCHAR(255),
    Address VARCHAR(255) NOT NULL,
    Longitude DECIMAL(13,10) NOT NULL,
    Latitude DECIMAL(13,10) NOT NULL,
    RentalPrice INT,
    CONSTRAINT pk_Housing_HousingId PRIMARY KEY (HousingId),
    CONSTRAINT fk_Housing_Zipcode FOREIGN KEY (Zipcode)
        REFERENCES Neighborhoods(Zipcode)
        ON UPDATE CASCADE ON DELETE SET NULL
);

CREATE TABLE Users (
    UserName VARCHAR(255),
    Password VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255) NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL,
    Phone BIGINT,
    HousingId INT DEFAULT NULL,
    IfBioVisible BOOLEAN DEFAULT TRUE,
```

```
    IsAuthenticatedResident BOOLEAN DEFAULT FALSE,
    CONSTRAINT pk_Users_UserName PRIMARY KEY(UserName),
    CONSTRAINT fk_Users_HousingId FOREIGN KEY(HousingId)
        REFERENCES Housing(HousingId)
        ON UPDATE CASCADE ON DELETE SET NULL
);

CREATE TABLE Students (
    UserName VARCHAR(255),
    School VARCHAR(255),
    CONSTRAINT pk_Students_UserName PRIMARY KEY(UserName),
    CONSTRAINT fk_Students_UserName FOREIGN KEY(UserName)
        REFERENCES Users(UserName)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Professionals (
    UserName VARCHAR(255),
    Company VARCHAR(255),
    CONSTRAINT pk_Professionals_UserName PRIMARY KEY(UserName),
    CONSTRAINT fk_Professionals_UserName FOREIGN KEY(UserName)
        REFERENCES Users(UserName)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Reviews (
    ReviewId INT AUTO_INCREMENT,
    Created TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Content TEXT NOT NULL,
    Rating DECIMAL(2,1) NOT NULL,
    UserName VARCHAR(255),
    HousingId INT,
    CONSTRAINT pk_Reviews_ReviewId PRIMARY KEY(ReviewId),
    CONSTRAINT fk_Reviews_UserName FOREIGN KEY(UserName)
        REFERENCES Users(UserName)
        ON UPDATE CASCADE ON DELETE SET NULL,
    CONSTRAINT fk_Reviews_HousingId FOREIGN KEY(HousingId)
        REFERENCES Housing(HousingId)
        ON UPDATE CASCADE ON DELETE SET NULL
);


CREATE TABLE CrimeCases (
    CrimeId INT AUTO_INCREMENT,
    ReportDate DATE NOT NULL,
    OffenseType TEXT,
      Offense TEXT,
```

```
    District VARCHAR(255),
    Address VARCHAR(255),
    Longitude DECIMAL(13, 10) NOT NULL,
    Latitude DECIMAL(13, 10) NOT NULL,
    Zipcode INT,
    CONSTRAINT pk_CrimeId PRIMARY KEY(CrimeId),
    CONSTRAINT fk_Crime_Zipcode FOREIGN KEY (Zipcode)
        REFERENCES Neighborhoods(Zipcode)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Schools (
    SchoolId INT AUTO_INCREMENT,
    SchoolName VARCHAR(255) NOT NULL,
    Type ENUM('Private', 'Public'),
    Zipcode INT,
    Address VARCHAR(255) NOT NULL,
    Latitude DECIMAL(13,10) NOT NULL,
    Longitude DECIMAL(13,10) NOT NULL,
    CONSTRAINT pk_Schools_SchoolId PRIMARY KEY (SchoolId),
    CONSTRAINT fk_Schools_Zipcode FOREIGN KEY (Zipcode)
        REFERENCES Neighborhoods(Zipcode)
        ON UPDATE CASCADE ON DELETE CASCADE
);


CREATE TABLE Restaurants(
    RestaurantId INT AUTO_INCREMENT,
    RestaurantName VARCHAR(255),
    Description LONGTEXT,
    Zipcode INT,
    Longitude DECIMAL(13,10) NOT NULL,
    Latitude DECIMAL(13,10) NOT NULL,
    Address LONGTEXT NOT NULL,
    CONSTRAINT pk_Restaurants_RestaurantId PRIMARY KEY(RestaurantId),
    CONSTRAINT fk_Restaurants_Zipcode FOREIGN KEY(Zipcode)
        REFERENCES Neighborhoods(Zipcode)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE CulturalSpaces (
    CulturalSpaceId INT(11) AUTO_INCREMENT,
    CulturalSpaceName VARCHAR(255) NOT NULL,
    DominantDiscipline VARCHAR(255),
    Zipcode INT,
    Address VARCHAR(255) NOT NULL,
    Longitude DECIMAL(13,10) NOT NULL,
```

```
    Latitude DECIMAL(13,10) NOT NULL,
    CONSTRAINT pk_Cultural_Space_Id PRIMARY KEY (CulturalSpaceId),
    CONSTRAINT fk_Cultural_Space_Zipcode FOREIGN KEY(Zipcode)
        REFERENCES Neighborhoods(Zipcode)
        ON UPDATE CASCADE ON DELETE CASCADE
);


CREATE TABLE Parks (
    ParkId INT AUTO_INCREMENT,
    ParkName VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Zipcode INT,
    Longitude DECIMAL(13,10) NOT NULL,
    Latitude DECIMAL(13,10) NOT NULL,
    CONSTRAINT pk_Park_Id PRIMARY KEY(ParkId),
    CONSTRAINT fk_Park_Zipcode FOREIGN KEY(Zipcode)
        REFERENCES Neighborhoods(Zipcode)
        ON UPDATE CASCADE ON DELETE CASCADE
);


LOAD DATA INFILE 'zipcode.csv' INTO TABLE Neighborhoods
      FIELDS TERMINATED BY ','
      LINES TERMINATED BY '\n'
      IGNORE 1 LINES;

INSERT INTO Housing(Zipcode,Name,Address,Longitude,Latitude,RentalPrice)
     VALUE(98105,'Burke + Union Apartments','4516 Union Bay Pl NE,
Seattle', -122.2933131, 47.6620017, 1975),
            (98102,'2301 Fairview Ave E','2301 Fairview Ave E, Seattle',
-122.3300611, 47.6400825, 3800),
            (98118,'5913 Rainier Ave S','5913 Rainier Ave S, Seattle',
-122.2775154, 47.5492263, 1395),
            (98144,'999 Hiawatha','999 Hiawatha Place South, Seattle',
-122.3083167, 47.5931276, 1549),
            (98121,'Cirrus','2030 8th Ave, Seattle',-122.3374552,
47.6158931, 2020),

            (98133, 'Leilani Apartment Homes', '10215 Greenwood Ave N,
Seattle', -122.35563538650163, 47.70343484420463, 1575),
            (98117, 'Lillehammer Apartments', '7016 15th Ave NW,
Seattle', -122.3766179479987, 47.68014245356699, 1759),
            (98109, 'Mera', '630 Boren Ave N, Seattle',
-122.33562364800076, 47.625533877102335, 3305),
            (98109, 'The Shoresmith', '1170 Republican St, Seattle',
-122.33341721731374, 47.62342989795143, 1910),
```

```
            (98125, 'Riviera Apartments', '11540 Pinehurst Way NE,
Seattle', -122.31312044065241, 47.72643643376515, 1350),

            (98121, 'Arrive Seattle', '2116 4th Ave, Seattle',
-122.3439029, 47.6142715, 2890),
            (98107, 'Commons at Ballard', '5621 22nd Ave NW, Seattle',
-122.4226191, 47.6571719, 2795),
            (98115, 'Novo', '6105 Roosevelt Way NE, Seattle',
-122.2969116, 47.6313662, 1250),
            (98109, 'Neptune', '912 Dexter Ave N, Seattle', -122.3442631,
47.6275439, 2240),
            (98119, 'Harrison Square', '312 2nd Ave W, Seattle',
-122.3612731, 47.6216411, 1595),

            (98121, 'Avenue5 Residential', '2401 3rd Ave, Seattle',
-122.3331735, 47.6023157, 1871),
            (98101, 'Kinects Tower', '1823 Minor Ave, Seattle',
-122.3336853, 47.6171562, 1855),
            (98121, 'The Martin', '2105 5th Ave, Seattle', -122.343267,
47.6144904, 2135),
            (98104, 'Saxton', '520 Terry Ave, Seattle', -122.3254121,
47.6059779, 1980),
            (98106, 'Youngstown Flats', '4040 26th Ave SW, Seattle',
-122.3442681, 47.6275447, 1705),

            (98122, 'Lawrence Lofts', '1818 E Madison St, Seattle',
-122.307985, 47.616970,2175),
            (98109, 'Union SLU', '905 Dexter Ave N, Seattle',
-122.342714, 47.627502, 1955),
            (98104, 'Landes', '901 8th Ave, Seattle', -122.328085,
47.607083, 1841),
            (98122, 'Sylvan Court', '1901 E Fir St, Seattle',
-122.307558, 47.602401, 1595),
            (98122, 'East Union', '2220 E Union St, Seattle',
-122.304258, 47.613445, 2585);


INSERT INTO
Users(UserName,Password,FirstName,LastName,Email,Phone,HousingId,IfBioVisi
ble, IsAuthenticatedResident)

VALUES('wenliao','wen123','Wen','Liao','wen@gmail.com',123456789,null,fals
e,null),

('chen','chen123','Chen','Cheng','chen@gmail.com',987654321,1,true,true),

('liaaa','lia123','Lia','Luo','lia@gmail.com',343234235,3,false,true),
```

```
('kim','kim123','John','Kim','kim@gmail.com',4898765398,5,true,true),

('wendy','wendy123','Wendy','Brown','wendy@gmail.com',88976543,null,true,n
ull),

('jackk','jack123','Jack','Black','jack@outlook.com',14553224,2,true,true)
,

            ('daniel', 'password', 'Daniel', 'Jackson',
'danielj@mail.com', 1234567890, 6, true, true),
             ('james', 'password', 'James', 'Johnson', 'jamesj@mail.com',
2345436754567, 7, true, true),
             ('jason', 'password', 'Jason', 'Jordan', 'jasonj@mail.com',
4568769065, 8, false, true),
             ('bruce', 'password', 'Bruce', 'Lee', 'brucel@mail.com',
90756432345, 9, false, true),
             ('nelson', 'password', 'Nelson', 'Mao', 'nelsonm@mail.com',
8543452789, 10, true, true),
             ('nicole', 'password', 'Nicole', 'Bonde',
'nicoleb@mail.com', 9864326783, 6, true, true),

             ('niiiiiidhi', 'nloveseattle', 'Nidhi', 'Vaishnav',
'nidhigogo123@gmail.com', 2062266253, 11, true, true),
             ('alejandra1', 'alejandra123', 'Alejandra', 'Bordones',
'a.bordones@outlook.com', 2067756213, 12, true, true),
             ('BenBenBen', 'bbb123456', 'Ben', 'Hoover',
'bh1024@gmail.com', 2062386858, 13, true, true),
             ('whereisclefo', 'clefoT129!', 'Clefo', 'Townsel',
'clefoclefo@gmail.com', 2064350275, 14, true, true),
             ('skskskskskk', 'Sk_123@ookk', 'Sean', 'Keeler',
'keelerisaman@hotmail.com', 2061178024, 15, true, true),
             ('richarules', 'rrrmmMathur.123!', 'Richa', 'Mathur',
'richarules2021@gmail.com', 2069702353, 15, true, true),


('tim123','tim123','Tim','Cook','tim123@outlook.com',2061111234, 16,
true,true),

('tom123','tomim123','Tom','Pike','tom123@outlook.com',3061111234, 17,
true,true),

('jim123','jim123','Jim','Ron','jim123@outlook.com',4061111234,
18,true,true),

('sean123','sean123','Sean','Ki','sean123@outlook.com',5061111234,
20,true,true),
```

```
('jude123','jude123','Jude','DD','jude123@outlook.com',6061111234,
20,true,true),

('cici123','cici123','Cici','TT','cici123@outlook.com',7061111234,
19,true,true),

             ('alan', 'alan1', 'Alan', 'Leng', 'alan@mail.com',
2062571111, 21, false, true),
             ('blbn', 'blbn1', 'Blbn', 'lfng', 'blbn@mail.com',
2062572222, 22, true, true),
             ('clcn', 'clcn1', 'Clbn', 'lgng', 'clcn@mail.com',
2062573333, 23, true, true),
             ('dldn', 'dldn1', 'Dlbn', 'lhng', 'dldn@mail.com',
2062574444, 23, false, true),
             ('elen', 'elen1', 'Elbn', 'ling', 'elen@mail.com',
2062575555, 24, true, true),
             ('flfn', 'flfn1', 'Flbn', 'ljng', 'flfn@mail.com',
2062576666, 25, true, true),


             ('yi', 'password', 'Yi', 'Wang', 'yi@mail.com', 2062571112,
22, false, true),
             ('er', 'password', 'Er', 'Zhang', 'er@mail.com', 2062572223,
22, true, true),
             ('san', 'password', 'San', 'Huang', 'san@mail.com',
2062573334, 22, true, true),
             ('Si', 'password', 'Si', 'Li', 'si@mail.com', 2062574445,
22, false, true),
             ('Wu', 'password', 'Wu', 'Bai', 'liu@mail.com', 2062575556,
22, true, true),

             ('liu', 'password', 'Liu', 'Liu', 'liu@mail.com',
2062576668, 12, true, true),
          ('qi', 'password', 'Qi', 'Qi', 'qi@mail.com', 2062571113, 12,
false, true),
             ('ba', 'password', 'Ba', 'Ba', 'ba@mail.com', 2062572224,
12, true, true),
             ('jiu', 'password', 'Jiu', 'Jiu', 'jiu@mail.com',
2062573335, 12, true, true),
             ('shi', 'password', 'Shi', 'Shi', 'shi@mail.com',
2062574446, 12, false, true);

INSERT INTO Students(UserName, School)
     VALUES('wenliao','NEU'),
             ('chen','NEU'),
             ('liaaa','UW'),
```

```
            ('kim','UW'),

            ('daniel', 'NEU'),
            ('james', 'NEU'),
            ('jason', 'UW'),
            ('bruce', 'UW'),
            ('tim123', 'UW'),
            ('tom123', 'UW'),
            ('jim123', 'NEU'),
            ('sean123', 'NEU'),

            ('richarules', 'NEU'),
            ('BenBenBen', 'UW'),
            ('niiiiiidhi', 'UW'),
            ('alejandra1', 'NEU'),

            ('alan', 'NEU'),
            ('blbn', 'UW'),
            ('clcn', 'UW'),
            ('dldn', 'NEU'),


            ('yi', 'NEU'),
            ('er', 'NEU'),
            ('san', 'NEU'),
            ('si', 'NEU'),
        ('wu', 'NEU'),

        ('liu', 'NEU'),
        ('qi', 'NEU'),
            ('ba', 'NEU'),
            ('jiu', 'NEU'),
        ('shi', 'NEU');

INSERT INTO Professionals(UserName, Company)
      VALUES('wendy','Amazon'),
             ('jackk','Amazon'),

             ('nelson', 'Facebook'),
             ('nicole', 'Google'),

             ('whereisclefo', 'Facebook'),
             ('skskskskskk', 'Indeed'),
             ('jude123', 'LinkedIn'),
             ('cici123', 'Apple'),

             ('elen', 'AT&T'),
```

```
                ('flfn', 'Oracle');

INSERT INTO Reviews(Content,Rating,UserName,HousingID)
   VALUES('nice apartment',5.0,'kim',5),
    ('not bad',3.0,'chen',1),
    ('good',4.0,'jackk',2),
    ('the location is perfect',4.0,'wendy',5),
    ('nice apartment',4.0,'liaaa',3),
    ('bad',2.0,'wenliao',1),

    ('Wonderful Place', 5.0, 'daniel', 6),
      ('Don't ever lease this place', 1.0, 'james', 7),
      ('Just a random apartment', 3.5, 'jason', 8),
      ('Super good', 4.5, 'bruce', 9),
      ('Okay place', 4.0, 'nelson', 10),

      ('Wonderful!', 5.0,'niiiiiidhi', 11),
      ('Okay' , 3.0, 'alejandra1', 12),
      ('Great place', 4.0, 'BenBenBen', 13),
      ('Best!', 5.0, 'whereisclefo', 14),
      ('Dont come', 1.0, 'skskskskskk', 15),
      ('Worth the price', 5.0,'richarules', 15),

    ('great', 5.0,'tim123', 16),
    ('good', 4.0,'tom123', 17),
    ('the location is perfect', 5.0,'jim123', 18),
    ('nice apartment', 4.0,'sean123', 19),
    ('bad', 2.0,'jude123',20),


      ('Wonderful Place', 4.0, 'alan', 25),
      ('Don't ever lease this place', 1.0, 'blbn', 24),
      ('Just a random apartment', 3.5, 'clcn', 23),
      ('Super good', 4.5, 'dldn', 22),
      ('Okay place', 4.0, 'elen', 21);


LOAD DATA INFILE 'crime.csv' INTO TABLE CrimeCases
      FIELDS TERMINATED BY ','
      LINES TERMINATED BY '\n'
      IGNORE 1 LINES
      (@dummy, @dummy, @dummy, @dummy,
      @ReportDate,@dummy, @dummy,
      OffenseType, Offense, @dummy, @dummy, @dummy, @dummy,
      District, Address, Longitude, Latitude)
    SET
    ReportDate =  STR_TO_DATE(LEFT(@ReportDate, 10), '%m/%d/%Y'),
```

```
    Zipcode =  (
    select if (lower(District) like "%central%", 98101, (
    select if (lower(District) like "%belltown%", 98121, (
      select if (lower(District) like "%slu%", 98109, (
    select if (lower(District) like "%capitol%", 98102, (
    select if (lower(District) like "%university%", 98105, (
    select if (lower(District) like "%roosevelt%", 98115, (
    select if (lower(District) like "%first hill%", 98122, (
    select if (lower(District) like "%pioneer square%", 98104, (
    select if (lower(District) like "%northgate%", 98125, (
      select if (lower(District) like "%queen anne%", 98119, (
    select if (lower(District) like "%ballard%", 98107, (
    select if (lower(District) like "%chinatown%", 98104, (
    select if (lower(District) like "%brighton%", 98118, 00000)
    ))))))))))))))))))))));


LOAD DATA INFILE 'Private_Schools.csv' INTO TABLE Schools
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
    IGNORE 1 LINES
    (Longitude, Latitude, @dummy, @dummy,
    SchoolName, Address, @dummy, @dummy,
    Zipcode, @dummy, @dummy, @dummy, @dummy, @dummy)
    SET Type = 'Private';

LOAD DATA INFILE 'Public_Schools.csv' INTO TABLE Schools
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
    IGNORE 1 LINES
    (Longitude, Latitude, @dummy, @dummy,
    SchoolName, Address, @dummy, @dummy, @dummy, @dummy,
    Zipcode, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy)
    SET Type = 'Public';

LOAD DATA INFILE 'Restaurants.csv' INTO TABLE Restaurants
    FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
    LINES TERMINATED BY '\n'
    IGNORE 1 LINES
    (Longitude, Latitude, @dummy, @dummy,
    RestaurantName, Address, @dummy, @dummy, @dummy,
    Description, @dummy, @dummy,
    Zipcode);

LOAD DATA INFILE 'culture_space.csv' INTO TABLE CulturalSpaces
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
```

```
    IGNORE 1 LINES
    (CulturalSpaceName, DominantDiscipline, @dummy, Latitude, Longitude,
Address)
    SET Zipcode = (select if (Address like "%981__", RIGHT(Address, 5),
0));

LOAD DATA INFILE 'park.csv' INTO TABLE Parks
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
    IGNORE 1 LINES
    (@dummy,@dummy,ParkName,Address,Zipcode,Longitude,Latitude);
```