University of Toronto Scarborough

~ CSCC43 – Introductions to Databases (3rd Year Undergrad Course) ~

2016 Summer Project

Written Report – Explanations, Queries and Ramblings

Due: July 25, 2016

Students: *Evan Chen (cheneva2) and William Li (lizong4)*

Professor for CSCC43: *Professor Nick Koudas*

TA: *Penney*

Please Note:

1. *The things listed in this written report is made alongside the project program – "mybnb_project."*

2. *The project is coded in Java. However, some elements (which will be listed here in this report) use MySQL Workbench and use MySQL queries and update statements.*

3. *There are a few assumptions (listed in this report).*

4. *Input in the program HAS TO BE TYPED IN AS IS if it is shown in quotes in the program's output.*

The Purpose of the Project

This project, *Mybnb*, is designed to help users be able to book places for rent using an easy and simplistic interface (through terminal, currently). This design follows the popular home sharing service AirBnB. Users are able to be either hosts or renters. If the user is a host, they are able to easily host places for renters to rent out. These are represented in the form of listings – each new hosted place will have it's own listing, regardless if the user is the same user hosting more places. Of course, the listing needs to be available in order to be able to be rented by other users. If the user is a renter, then they are able to easily rent places that the hosts have provided through the *Mybnb*. They can book as many places as they can, as long as they are able to afford it and if the dates do not conflict. The database includes a few tables that save data from any of the users when they book or host places. The goal of the project is to promote simplicity and easy usability for all users around the world while also promoting a simple system in which users are able to trust each other when renting and/or hosting places.

~Conceptual Problems and Justified Solutions~

Some of the problems we ran into are time schedules and agreement on key types and key placements (including certain things such as objects that can be easily translated into Java code).

To start off, time schedules were more or less of a struggle during this time as no one had free time during most of the week. The only time we were able to meet was on Monday since we would always have the Lecture + Tutorial combination. However, there may be times when, say, one of us cannot make it because of other things such as other assignments. So, in order to solve this issue, we had to plan everything out for two Mondays before we start coding, both in Java and in MySQL languages. With this planning, we created a solid foundation, something that we are able to rely on nearly most of the time.

To add on, another problem was the agreement on key types and key placements for certain objects in the database. Although this problem was a small issue, we are glad that we took the time to plan out the schema from ground up. With our own schemas, we wanted to come to an agreement – something that we are able to utilize in our Java coding and MySQL database tables. We want it designed the way it is designed so that every piece of information is somehow linked with each other (which is another goal in the project).

Overall, with these problems, planning was the key solution here. Although time played a role in this, planning the project first was tactical since programming the project had its own slew of problems.

To continue, when we were programming the project, we started to realize that we needed more keys – more resources – in order to get more information across. Otherwise, by linking the database tables together, it wouldn't have much meaning in the end of it all. (The rough draft is located near the end of this report.) However, we persisted with each new problem thrown at us and found ways to solve each problem without having to "band-aid" it most of the time.

Some Assumptions to Take Note Of

Of course, most projects will have assumptions for many things – in this case, our assumptions will stem from the users themselves.

We assume that the users are able to read in order to enjoy the software the we have created and that they will always know when to give inputs. Also, we assume that users are able to copy letter-for-letter what is inside the quotation marks when giving inputs that are asking for their commands.

We also assume that the age of all users must be 18 years of age or above. However, that is already covered in a constraint. This constraint will be shown in this report.

***Some Statements to Initialize Database** mybnb*

```
CREATE TABLE `Users` (
            `u_id` INT NOT NULL AUTO_INCREMENT
   ,`u_name` varchar(50) NOT NULL
            ,`u_username` varchar(50) NOT NULL
   ,`u_pwd` varchar(20) NOT NULL
            ,`u_address` varchar(100) NOT NULL
   ,`u_dob` varchar(15) NOT NULL
            ,`u_occupation` varchar(50) NOT NULL
   ,`u_phonenum` varchar (15) NOT NULL
            ,`u_email` varchar(50) NOT NULL
   ,`u_SIN` varchar(100) NOT NULL
            ,`u_creditcard` varchar(100) NOT NULL
            ,`u_age` INT NOT NULL
            ,`u_renter` varchar(3) NOT NULL
            , `u_hoster` varchar(3) NOT NULL
            , `u_longitude` FLOAT(6,4) NOT NULL
            ,`u_lattitude` FLOAT(7,4) NOT NULL
            ,`u_city` varchar(100) NOT NULL
            ,`u_province` varchar(100) NOT NULL
            ,`u_country` varchar(100) NOT NULL
            ,`u_cancellations` INT NULL
            ,PRIMARY KEY (`u_id`)
   ,UNIQUE (`u_id`)
) ENGINE = INNODB;
```

```
CREATE TABLE `Listings` (
         `l_id` INT NOT NULL AUTO_INCREMENT
   ,`l_address` varchar(100) NOT NULL
         ,`l_propertyType` varchar(100) NOT NULL
         ,`l_roomType` varChar(100) NOT NULL
   ,`l_availability` varchar(13) NOT NULL
         ,`l_date` varchar(1000) NOT NULL
         ,`l_price` FLOAT(7,2) NOT NULL
   ,`l_washrooms` INT NOT NULL
         ,`l_bedrooms` INT NOT NULL
   ,`l_postalCode` varchar (15) NOT NULL
         ,`l_rating` INT NOT NULL
   ,`l_longitude` FLOAT(7,4) NOT NULL
         ,`l_lattitude` FLOAT(7,4) NOT NULL
         ,`u_id` INT NOT NULL
   ,`l_amenities` varchar(1000) NOT NULL

         ,PRIMARY KEY (`l_id`)
   ,UNIQUE (`l_id`)
) ENGINE = INNODB;
```

```
CREATE TABLE `CompleteHistory` (
            `log_id` INT NOT NULL AUTO_INCREMENT
   ,`u_id` INT NOT NULL
            ,`log_reason` varchar(1000) NOT NULL

            ,PRIMARY KEY (`log_id`)
   ,UNIQUE (`log_id`)
) ENGINE = INNODB;
```

```
CREATE TABLE `Bookings` (
            `b_id` INT NOT NULL AUTO_INCREMENT
   ,`u_id` INT NOT NULL
   ,`l_id` INT NOT NULL
            ,`b_date` varchar(1000) NOT NULL

            ,PRIMARY KEY (`b_id`)
   ,UNIQUE (`b_id`)
) ENGINE = INNODB;
```

```
CREATE TABLE `BookingsHistory` (
          `bh_id` INT NOT NULL AUTO_INCREMENT
   ,`u_id` INT NOT NULL
   ,`l_id` INT NOT NULL
          ,`b_date` varchar(1000) NOT NULL
          ,`l_address` varchar(1000) NOT NULL

          ,PRIMARY KEY (`bh_id`)
   ,UNIQUE (`bh_id`)
) ENGINE = INNODB;



CREATE TABLE `Comments` (
          `c_id` INT NOT NULL AUTO_INCREMENT
   ,`l_id` INT NOT NULL
   ,`u_renterid` INT NOT NULL
          ,`u_hosterid` INT NOT NULL
          ,`b_date` varchar(1000) NOT NULL
          ,`c_comment` varchar(1000) NOT NULL
   ,`c_commentnouns` varchar(1000) NOT NULL

          ,PRIMARY KEY (`c_id`)
   ,UNIQUE (`c_id`)
) ENGINE = INNODB;
```

```
delimiter |

CREATE TRIGGER usersIns BEFORE INSERT ON Users

FOR EACH ROW BEGIN

DECLARE msg varchar(255);

IF NEW.u_age < 18 THEN

SET msg = 'Constraints violated!';

SIGNAL sqlstate '45000' set message_text = msg;

END IF;

END

|


grant all privileges on mybnb.users to 'gamingturtile'@'localhost'  identified by 'password';

grant all privileges on mybnb.listings to 'gamingturtile'@'localhost'  identified by 'password';

grant all privileges on mybnb.completehistory to 'gamingturtile'@'localhost'  identified by 'password';

grant all privileges on mybnb.bookings to 'gamingturtile'@'localhost'  identified by 'password';

grant all privileges on mybnb.bookingshistory to 'gamingturtile'@'localhost'  identified by 'password';

grant all privileges on mybnb.comments to 'gamingturtile'@'localhost'  identified by 'password’;
```

Some Data that Populates the Database

```
INSERT INTO Users (u_name, u_username, u_pwd, u_address, u_dob, u_occupation, u_phonenum,
u_email, u_SIN, u_creditcard,  u_age, u_renter, u_hoster, u_longitude, u_lattitude, u_city, u_province,
u_country) VALUES ('Evan', 'gamingturtile', 'abc123', '123 ABC Way', '10/31/19XX', 'Student', '647-123-
4567', 'gamingturtile@random.com', '123456789', '1234123412341234', 21, ‘No’, ‘Yes’, 0.0, 0.0,
‘Toronto’, ‘Ontario’, ‘Canada’);
```

```
INSERT INTO Users (u_name, u_username, u_pwd, u_address, u_dob, u_occupation, u_phonenum,
u_email, u_SIN, u_creditcard,  u_age, u_renter, u_hoster, u_longitude, u_lattitude, u_city, u_province,
u_country) VALUES ('William', 'sytios', 'abc123', '1 Apple St.', '05/31/19XX', 'Student', '647-915-1267',
'sytios@random.com', '123456788', '8907890712341234', 21, 'No', 'Yes', 0.0, 0.0, 'Toronto', 'Ontario',
'Canada');
```

```
INSERT INTO Users (u_name, u_username, u_pwd, u_address, u_dob, u_occupation, u_phonenum,
u_email, u_SIN, u_creditcard,  u_age, u_renter, u_hoster, u_longitude, u_lattitude, u_city, u_province,
u_country, u_cancellations) VALUES ('Vivian', 'viv', 'xyzBio', '12 Christian Rd.', '05/27/19XX', 'Student',
'647-290-3377', 'viv@random.com', '123456790', '1231231231231239', 21, 'Yes', 'No', 0.0, 0.0,
'Toronto', 'Ontario', 'Canada', '0');
```

```
INSERT INTO Users (u_name, u_username, u_pwd, u_address, u_dob, u_occupation, u_phonenum,
u_email, u_SIN, u_creditcard,  u_age, u_renter, u_hoster, u_longitude, u_lattitude, u_city, u_province,
u_country, u_cancellations) VALUES ('Daniella', 'ciarrocchi', 'econ123', '1 Italy Ave.', '01/17/19XX',
'Student', '647-757-8229', 'ciarrocchi@random.com', '123456791', '9089786756891234', 30, 'Yes', 'No',
0.0, 0.0, 'Toronto', 'Ontario', 'Canada', '0');
```

```
INSERT INTO Users (u_name, u_username, u_pwd, u_address, u_dob, u_occupation, u_phonenum,
u_email, u_SIN, u_creditcard,  u_age, u_renter, u_hoster, u_longitude, u_lattitude, u_city, u_province,
u_country, u_cancellations) VALUES ('Jackson', 'JacksonJesus', 'toocoolforskool', '1 Awesome Cresent',
'02/28/19XX', 'Student', '647-952-7399', 'jj@random.com', '123456792', '1234567890123456', 21, 'Yes',
'No', 0.0, 0.0, 'Toronto', 'Ontario', 'Canada', '0');
```

INSERT INTO Users (u_name, u_username, u_pwd, u_address, u_dob, u_occupation, u_phonenum, u_email, u_SIN, u_creditcard, u_age, u_renter, u_hoster, u_longitude, u_lattitude, u_city, u_province, u_country) VALUES ('Joy', 'xJayson', 'cutiepie', '10 Cool Ave.', '01/29/19XX', 'Student', '647-123-1234', 'xJayson@random.com', '123456793', '1236173820192385', 21, 'No', 'Yes', 0.0, 0.0, 'Toronto', 'Ontario', 'Canada');

INSERT INTO Listings (l_address, l_propertyType, l_roomType, l_availability, l_date, l_price, l_washrooms, l_bedrooms, l_postalCode, l_rating, l_longitude, l_lattitude, u_id, l_amenities) VALUES ('68 Havey Junction', 'Villa', 'Shared Room', 'AVAILABLE', '(2017-02-06, 2017-02-07, 2017-02-08, 2017-02-09, 2017-02-10, 2017-02-11, 2017-02-12, 2017-02-13, 2017-02-14, 2017-02-15, 2018-03-24) ', '108.00', '8', '5', 'M7G9H4', '3', '-107.90217', '-79.27354', '1', ' (Suitable for Events,Essentials) ');

INSERT INTO Listings (l_address, l_propertyType, l_roomType, l_availability, l_date, l_price, l_washrooms, l_bedrooms, l_postalCode, l_rating, l_longitude, l_lattitude, u_id, l_amenities) VALUES ('68893 Kipling Lane', 'Bungalow ', 'Shared Room ', 'AVAILABLE ', '(2016-11-27, 2016-11-28, 2016-11-29, 2016-11-30, 2016-12-01, 2017-03-29, 2017-03-30, 2017-03-31, 2017-04-01, 2017-04-02, 2017-04-03, 2017-04-22, 2017-04-23, 2017-04-24, 2017-04-25, 2017-04-26, 2017-04-27, 2017-05-30, 2017-05-31, 2017-06-01, 2017-06-02, 2017-06-03, 2017-06-04, 2017-06-11, 2017-06-12, 2017-06-13, 2017-06-14, 2017-06-15, 2017-06-16, 2017-06-17, 2017-11-17, 2017-11-18, 2017-11-19, 2017-11-20, 2017-11-21, 2017-11-22, 2017-12-11, 2017-12-12, 2017-12-13, 2017-12-14, 2017-12-15, 2017-12-16, 2017-12-17, 2017-12-18, 2018-09-24, 2018-09-25, 2018-09-26, 2018-09-27, 2018-09-28, 2018-09-29, 2018-09-30)', '70.00', '5', '13', 'M7G9H4', '2', '73.43425', '-66.37328', '1', '(Iron, Washer, Buzzer/Wireless Intercom, Smoking Allowed, Breakfast, TV, Kitchen, Free Parking, Doorman, Indoor Fireplace, A/C, Gym, Pool, 24hr Check-in, Elevator, Wired Internet, Hair Dryer, Pets Allowed, Heating, Hot Tub, Family/Kid Friendly, Shampoo, WiFi, Wheelchair Accessible, Laptop Friendly Workspace)');

```
INSERT INTO Listings (l_address, l_propertyType, l_roomType, l_availability, l_date, l_price,
l_washrooms, l_bedrooms, l_postalCode, l_rating, l_longitude, l_lattitude, u_id, l_amenities) VALUES
('0269 Homewood Plaza ', 'Bungalow ', 'Shared Room ', 'AVAILABLE ', '( 2017-06-16, 2017-06-17,
2017-06-18, 2017-06-19, 2017-06-20, 2017-06-21, 2017-06-22, 2017-06-23, 2017-06-24, 2017-
06-25, 2017-06-26, 2017-06-27, 2017-06-28, 2017-11-18, 2017-11-19, 2017-11-20, 2017-11-21
)', '109.00', '13', '8', 'M7G9H4', '3', '109.05643', '72.27959', '3', '( TV, Smoking Allowed, Breakfast,
Essentials, Pets Allowed, Cable TV, Washer, Gym, Laptop Friendly Workspace, WiFi, Kitchen ) ');
```

```
INSERT INTO Listings (l_address, l_propertyType, l_roomType, l_availability, l_date, l_price,
l_washrooms, l_bedrooms, l_postalCode, l_rating, l_longitude, l_lattitude, u_id, l_amenities) VALUES
('3048 Rieder Court ', 'Villa ', 'Shared Room ', 'NOT AVAILABLE', '()', '127.00', '14', '6', 'M7G9H4',
'4', '144.39272', '40.44396', '3', '( Shampoo, Hangers, Doorman, Kitchen, Breakfast, Wired
Internet, Wheelchair Accessible, Washer, Pets Allowed, Dryer, Family/Kid Friendly, Smoking
Allowed, Heating, A/C, TV, Essentials, Suitable for Events, Cable TV, Gym, Buzzer/Wireless
Intercom, Indoor Fireplace ) ');
```

```
INSERT INTO Listings (l_address, l_propertyType, l_roomType, l_availability, l_date, l_price,
l_washrooms, l_bedrooms, l_postalCode, l_rating, l_longitude, l_lattitude, u_id, l_amenities) VALUES
('0269 Homewood Plaza ', ' Condominium', 'Entire Home/Apt ', 'NOT AVAILABLE', '()', '142.00',
'5.5', '14', 'M7G9H4', '3', '114.04629', '3.422989', '2', '( Essentials )');
```

```
INSERT INTO Listings (l_address, l_propertyType, l_roomType, l_availability, l_date, l_price,
l_washrooms, l_bedrooms, l_postalCode, l_rating, l_longitude, l_lattitude, u_id, l_amenities) VALUES
('94743 Coolidge Junction ', 'Bed and Breakfast ', 'Entire Home/Apt ', 'AVAILABLE', '(2016-09-30,
2016-10-01, 2016-10-02, 2016-10-03, 2016-10-04, 2016-10-05, 2016-10-06, 2016-10-07, 2016-
10-08, 2016-12-28, 2016-12-29, 2016-12-30, 2016-12-31, 2017-01-01, 2017-01-02, 2017-01-03,
2017-01-04)', '108.00', '4', '8', 'M7G9H4', '3', '-69.54810', '6.51031', '1', '(Hair Dryer, Pets
Allowed, Elevator, Heating, Hangers, Kitchen, Dryer, Wired Internet, Free Parking, Essentials,
Washer, Doorman, Gym, Family/Kid Friendly, Buzzer/Wireless Intercom, Iron, Laptop Friendly
Workspace, 24hr Check-in, Pool, Cable TV, TV, Indoor Fireplace, Wheelchair Accessible, Hot
Tub, Suitable for Events, WiFi, Smoking Allowed, Breakfast )');
```

```
INSERT INTO Listings (l_address, l_propertyType, l_roomType, l_availability, l_date, l_price,
l_washrooms, l_bedrooms, l_postalCode, l_rating, l_longitude, l_lattitude, u_id, l_amenities) VALUES
('40 Grim Drive', ' Townhouse', ' Shared Room', 'AVAILABLE', '( 2016-08-26, 2016-08-27,
2016-08-28, 2016-08-29, 2016-08-30, 2016-08-31, 2016-09-01, 2016-09-02, 2016-09-03, 2016-09-04,
2016-09-05, 2016-09-06, 2016-11-29, 2016-11-30, 2016-12-01, 2016-12-02, 2016-12-03,
2016-12-04, 2016-12-05, 2016-12-06, 2016-12-07, 2017-01-28, 2017-01-29, 2017-01-30, 2017-01-31,
2017-03-26, 2017-03-27, 2017-05-26, 2017-05-27, 2017-05-28, 2017-05-29, 2017-08-03,
2017-08-04, 2017-08-05, 2017-08-06, 2018-07-13, 2018-07-14, 2018-07-15, 2018-07-16, 2018-07-17,
2018-07-18, 2018-07-19, 2018-07-20, 2018-07-21, 2018-07-22, 2018-07-23, 2018-07-24,
2018-07-25 )', '126.00', '10', '9', 'M7G9H4', '3', '135.93776', '86.96837', '1', '( Breakfast, Laptop
Friendly Workspace, Washer, Doorman, Shampoo, Family/Kid Friendly, Suitable for Events,
Hair Dryer, A/C, Wheelchair Accessible, Kitchen, Indoor Fireplace, Buzzer/Wireless Intercom,
WiFi, Smoking Allowed )');
```

User Manual

Please note that when you are given prompts, during your input, you HAVE to match up the strings that are given to you in the question. (These inputs are CASE SENSITIVE, so be warned that you DO need to capitalize letters when you need to.)

When you start the program, you will be prompted to give an input. The question goes:

```
Would you like to "Sign In" or "Add User"?
```

Your input should be either `Sign In` or `Add User`. When you select `Add User`, you will be prompted to give another input. You have to choose between a `Renter` or `Hoster`. Again, you have to be exact and choose between one of the two. When you do give input to add the user, you will come across a sign up form that both users use. Here is what the sign up form might look like.

```
Welcome. Please add your information.
Your name: Test <string>
Your desired username: Test <string>
Your desired password: Test <string>
Your address: 123 Test Ave. <string>
Your date of birth: 01/01/1950 <string, following MM/DD/YYYY format>
Your current occupation: Tester <string>
Your phone number: 647-000-0000 <string>
Your email: test@tester.com <string>
Your Security Insurance Number: 000000000 <string>
Your credit card number: 0000000000000000 <string>
The city you live in currently: Toronto <string>
The province you live in currently: Ontario <string>
The country you live in currently: Canada <string>
Your birth year: 1950 <int>
The longitude you are currently located on: 10.0 <double/float>
The lattitude you are currently located on: 10.0 <double/float>
```

Please note that you can ignore the <> angle brackets in this example. It just shows what the form would look like and which line takes in what type of input.

Once you have entered in your information, assuming that the current year – your birth year is greater than or equals 18 years of age, you will be greeted to this text:

```
Added user! Please go back to login screen!
See you next time!
```

From here, you need to restart the application.

Now, you are able to either `Sign In` or `Add User` again. This time, if you choose `Sign In`, you must enter in your credentials in order to continue.

```
Welcome. Please sign in using your username and password.
Username: Test
Password: Test
```

You must type in your credentials. These credentials MUST match at MOST one user in the database (that means that you cannot create another user with the same username). It is case-sensitive, be careful!

Now, depending on the user that's signing in, you will be greeted with one of the two instances of the given options:

(For the hoster's options, we have)

```
Welcome user.
1) Comment on a renter.
2) Rate a renter.
3) Host a new location.
4) Delete hosted location.
5) Show all your uploaded listings.
6) Update listing information.
7) View all renter accounts sorted by cancellations.
8) Delete account.
9) Leave application.
Your command:
```

(For the renter's options, we have)

```
Welcome user.
1) Comment on place rented.
2) Rent out a place.
3) Cancel booking for rented location.
4) Update billing information.
5) View booking history.
6) View all listings.
7) Search through all listings.
8) View all listings in descending order by price.
9) Delete account.
10) Leave application.
Your command:
```

From here, you will be given the option to choose from the selection of options. Your command MUST BE AN INTEGER HERE.

Now, most of the options here are trivial, but I would like to touch upon the commenting sections and the listing sections.

As a `Hoster`, you are able to make listings. When you do choose 3 to be your command input, you will be able to add a listing to the database. You will be greeted with the following:

```
If you wish to add a listing, please enter the following information.
Address: 779 Grasskamp Lane <string>
Property Type: Townhouse <string>
Room Type: Private Room <string>
Availability (you can only put "AVAILABLE" or "NOT AVAILABLE"): AVAILABLE
<string>
Date of Availability: ( 2016-07-28 ) <string>
Postal Code of Location: M8P2J3 <string>
Amenities Included: ( Free Parking, Hot Tub, Internet, Cable, Swimming Pool )
<string>
Price: 10000.00 <double>
Number of Washrooms: 7 <int>
Number of Bedrooms: 7 <int>
Rating of Location (only numbers 1 to 5): 4 <int>
Longitude of Location: 93.12345 <double>
Lattitude of Location: 10.65409 <double>
Added listing successfully!
```

Again, you can ignore the angle brackets. They are there to tell you what the input types should be.

Once the listing gets approved by the database, it will be added into it. Thus, `Renter` are allowed to book these listings (and cancel AFTER booking it) whenever they want. When these renters try to cancel no bookings, the program is user friendly and does not cancel anything more.

Now, when we give comments, the renters MUST comment first. The renter MUST have at least booked a place first before commenting on that listing. He can also give descriptions for others to see as well. Once the renter has commented, the hoster will be able to comment. Otherwise, if the hoster tries to comment on no renter comments, then the hoster can TRY to comment on something but it will not be added to the database.

Aside from that, users are able to delete their accounts. You simply need to input in your username and password (the ones you used before during the sign in section) in order for this to happen. (This is for authentication purposes.)

With all of that aside, hope you enjoy the program! :)

Some Queries to Test Out

```
SELECT u_username FROM Users WHERE u_occupation='CEO' AND u_hoster=true

SELECT l_id FROM Listings WHERE l_propertyType LIKE "%house%" AND l_price<50

SELECT u_name FROM Users AS a NATURAL JOIN Listings AS b WHERE b.l_address="6 Eastwood Road"

SELECT l_id FROM Listings ORDER BY l_price DESC LIMIT 10

SELECT l_price, l_propertyType FROM Listings WHERE l_date LIKE '%12/25/2016%'

SELECT l_id FROM Listings WHERE l_price BETWEEN 50 AND 100 AND l_amenities LIKE "%internet%" AND l_amenities LIKE "%A/C%" AND l_amenities NOT LIKE "%pet friendly%"

SELECT u_id FROM Users AS a NATURAL JOIN Listings AS b WHERE (b.l_price>=20) HAVING COUNT(l_id)>10

SELECT u_id, u_city FROM Users AS a NATURAL JOIN Listings AS b GROUP BY a.u_city ORDER BY b.l_rating DESC

SELECT u_id FROM Users AS a NATURAL JOIN BookingsHistory AS b WHERE b.l_address="3980 Clyde Gallagher Junction" GROUP BY a.u_id

SELECT u_id FROM Users AS a NATURAL JOIN BookingsHistory AS b WHERE b.l_address="3980 Clyde Gallagher Junction" HAVING COUNT(u_id)>=5
```