

## Database design project (MyBnB)

In this project you will design a database that supports the operations of a platform such as the popular home sharing service AirBnB ([www.airbnb.com](http://www.airbnb.com)). Your database should record the following information.

### Information to record

Listings (such as apartments, houses, rooms) are placed for rent on the service. A listing is associated with information such as the following. The type of the listing (full house, apartment, room), location of the listing (a latitude and longitude) that precisely records the location of the listing on geographical coordinates, the address of the listing (including postal code, city and country), the characteristics of the listing such as the amenities it offers. You should use the same set of amenities and types of listings as those provided in the official AirBnB system.

A listing is associated with a calendar of its availability. This records the days in a year that the listing is available and unavailable for rent along of course with the associated rental price. Your system should make it easy for a host to insert availability and price and also conduct changes for price and availability of a listing.

Users can create an account on the system in order to either start renting listings or becoming hosts. For every user we should record information such as name, address, date of birth (you should be of legal age to participate, say at least 18 years of age), occupation, social insurance number. For renters the system should record payment information such as credit card that will be used to pay for the reservation. For every renter we should keep a complete history of the listings the user has rented with all the future bookings as well. For hosts we should maintain the list of listings they have on the system along with the rental history of each listing.

Users have the capability to insert comments (in a form of free form text) to record their experience. They also have the ability to rate on a scale of one to five their experience with the listing as well as the host. Similarly hosts can rate and comment on renters profiles reporting of their experience. A host can comment on a renter that has rented the listed place and the renter can comment only on a host from who he/she has rented.

## Operations to support

Users (renters or hosts) should be able to create profiles on the system, operations such as create and delete user (renter or host) should be fully supported. Each time a new user is created all required information should be collected and associated with the created record. It's completely up to you and your design to determine how you like to represent users in your system but your decision should be well justified.

A user should be able to book one or more listings, create a new listing and associate it with their account, cancel a booking or remove a listing. Naturally only people that created the listing or the booking are able to delete or cancel it. When a creation, booking or removal of a listing takes place all suitable information should be updated (including of course the calendar that marks the listing as available or unavailable the specific dates). Several obvious constraints should be satisfied. For example if a listing is rented for a day in the future (we assume that the transaction is final), the host cannot change the price. The host however can cancel a booking anytime and similarly the renter can cancel as well.

When a renter books a listing, obviously the listing has to be available in the desired date range and the calendar of availability of the listing should be updated. Otherwise the booking cannot take place.

In addition if a host or renter has canceled a booking, this information needs to be recorded for future reference.

A host should be able to update the price of a listing, but only if the listing is available for rent in the specific date range the change is to be made. Otherwise if the listing is booked the change cannot take place and the host should be informed.

A host cannot change the availability of a listing on a date it is booked; that can only happen through a cancellation (the host has to cancel the booking that day). The host can change the availability of a listing on a date it is available (and make it unavailable for rent that date).

Users should be able to insert comments for renters and hosts respectively. Naturally various constraints should be enforced. For example you cannot

comment on a listing if you haven't rent it recently. Similarly a host cannot comment and rate a renter if the renter hasn't completed a stay recently.

### **Queries to support**

The system at a minimum should be able to search for specific listings in the vicinity of a specific location. Namely if a user specifies a latitude and longitude we should return all listings with a specific distance (the user should have a choice of the distance along with a default provided). The listings returned should be ranked by the distance to the specific search location. You have to decide what distance you will use. Similarly the search should provide option to rank the listings by price (ascending or descending).

Other search options should be possible for example a search by postal code which should return all listings in the same and adjacent postal codes.

The system should also support exact search queries, by address. The search will accept an address in the input and return the listing in that address if one exists.

Another mode of search should refine the above searches with a temporal filter, meaning that we should also provide a date range that we are interested in and the system should return listings which are available for booking in the date range specified.

The system should support filters for the search fully. For example searching by postal code for listings with a set of amenities and time window of availability and a price range should be fully supported.

### **Reports to support**

As the data in the collection increases we would like to run certain reports periodically to understand our data.

We would like to run a report and provide the total number of bookings in a specific date range by city. We would also wish to run the same report by zip code within a city.

We would like to run a report and provide the total number of listings per country, per country and city as well as per country, city and postal code.

We would like to rank the hosts by the total number of listings they have overall per country, but also be able to refine this reporting for the hosts based on the number of listings they have by city.

For every city and country a report should provide the hosts that have a number of listings that is more than 10% of the number of listings in that city and country. This is a query that identifies the possible commercial hosts, something that the system should flag and prohibit.

We would also like to rank the renters by the number of bookings in a specific time periodic as well as rank them by number of bookings in a specific time period per city. For the later report, we are only interested to rank those renters that have made at least two bookings in the year.

We also wish to report the hosts and renters with the largest number of cancelations within a year.

Since renters comment on the listings, the listings accumulate comments in text form. We would like to run a report that presents for each listing the set of most popular noun phrases associated with the listing. That can form the basis of creating a word cloud for each listing that represents what renters say. You do not have to create any visualization as part of this project, only report the noun phrases.

## Host toolkit

MyBnB wishes to be very useful to renters but also to hosts. You are asked to create a simple function that could help a host position their listing in the MyBnB service. For example when you create a listing for the first time, can you come up with a strategy to price the listing (suggest a price for this listing)? Also when a host creates a listing can you suggest what are good amenities to have? Some of these amenities obviously exist in the listing already, but some other amenities the listing may not have, so the host may think to add them. Extra credit will be provided if your algorithm can suggest for each amenity you add (which the listing does not have) what is the expected revenue increase to be anticipated if the amenity is added in the

listing. You should have a working solution on the above. Any solution would be fine as long as it is documented and clearly justified.

## **Rules of the Game**

The project is to be done in groups of one or maximum two students. The groups are 'self policing' (you are responsible for division of labor etc). However if asked, I would expect that you can give me an accounting of what each member did and what was the contribution from each member. Each member of the group should be familiar with the entire project, including the code and algorithms used. If an irreconcilable problem arises in your group it is your responsibility to contact the instructor at once. After the project is due it would be too late.

- Assumptions: You might have to make assumptions for your design. You can do so, provided 1) they are stated in your reports, 2) they don't conflict with other requirements; 3) they are 'reasonable'. If you have questions about the acceptability of your assumptions you can ask the instructor or the TA's about them.
- Implementation: The database and your application will be implemented in Java embedded SQL using MySQL. Information to download and install MySQL is posted on the course web page along with example code.

## **Project Report**

Due July 29th. The report will include:

- A short description of the purpose of the project. A description of the conceptual problems encountered and the justification of your solutions.
- The assumptions you did (if any).
- An ER diagram of the enterprise, specifying attributes, entity sets, relationships and primary keys.
- The relation schema and the keys.
- DDL statements to create the schema in some appropriate normal form, specifying the types and the constraints that you identify and you wish to maintain.

- The source code for your implementation. You should have a file to create and drop the schema of your database. Also sample data files to populate the tables in your database. This is important as for your final course demo you should be able to demonstrate all queries by bulk loading enough data in your database.
- You should also include and hand in separately a user manual and a description of the system limitations and possibility for improvement.

Demos will be scheduled every 15 minutes or so on July 29<sup>th</sup>. The demo will test all aspects of the project and each query specified in the description should be working. Make sure you explain any algorithms you used for the host toolkit so the TAs can understand what you did. The oral test would ask you to identify specific areas your code where a query is implemented and then show the suitable queries running live in your program.

### **Implementation Tips**

What is expected is to have at a minimum text based interface on the terminal screen. It is imperative to make sure all the queries/parts of the system work fine before you invest time to build a sophisticated GUI. The host toolkit should have a working function and the algorithm can be very simple. However if you have a well-documented and innovative solution for this feature you will receive extra credit.