CSCC43 August 8, 2022

Johnson Lai

Database Project

1. **Description of the project (Purpose, Problems, Solutions)**

This project was about creating a database system in MySQL that could manage an AirBnB-like system. The purpose was to utilize SQL techniques and conceptual design practices to create a database that can handle analytics with large data. This means creating databases with an adequate normal form, making sure relevant information updates correctly, and having useful queries.

Some conceptual problems were about whether the calendar should contain just the date and price or other things such as the owner. I decided to include the owner and listing number because a date and price are not unique to a listing.

Another problem was implementing a booking system on top of a calendar system. Every time I had to book a listing, a check had to be made with other bookings as well as the calendar, something that was out of SQL’s scope. The solution was to implement an algorithm in Java.

One egregious problem was whether the Host should be its own relation. There was nothing unique to the Host, as ownership could be done by a renter as well. The solution was to build another relation about ownership which linked a user to a listing.

A problem towards the end was how to find nouns and combine varchars in SQL. It isn’t possible in SQL, so an algorithm in Java must be implemented where nouns are taken from a preset list. A library for nouns would be optimal but I decided not to implement that as it was outside the scope of this course and would take too much time.

1. **Assumptions made**

There are quite a few assumptions I made:

* “Recent” denotes one year from submission date, i.e. 2021-08-08 and more recent.
* The reports about “total bookings by city” etc. means total bookings given a city the user specifies.
* The reports about “per country” means “give a query that returns all countries but grouped by countries”.
* Renters can also be hosts.
* All bookings from here on out will be set in the future.
* The calendar relation must be one to one with listings.
* Update, a relationship in the ER Diagram, didn’t need to be a relation in SQL as the functions were handled by Java
* List filters are only supplied if the user requests it, through a GUI questionnaire
* Amenities can only be requested or denied, no “I might want it” options
* Cancellations from hosts are calculated from their listings that get cancelled by renters.

1. **ER Diagram**

**Diagram

Description automatically generated**

1. **Relation Schema**

Listings (listno, housetype, longitude, latitude, address, postal, city, country)

User (sin, name, job, date\_of\_birth, address)

Calendar (listno, date\_start, date\_end, price)

Booking (listno, date\_start, date\_end, sin, cancel)

Amenities (listno, ac, heat, lake, pets, internet)

Listing\_Comments (listno, timestamp, comments)

Reviews (reviewer, reviewee, rating, comment, type)

Renter (sin, credit\_card)

Rent\_History (listno, sin, type)

Ownership (sin, listno)

House\_History (listno, sin)

1. **User Manual & System Limitations**

User manual for this system: As this was implemented in Java with Eclipse 2020, to run this program you need to

* Open the folder “bnb” in Eclipse
* Select “bnbGUI.java” under src/driver
* Right Click “bnbGUI.java”, then click “Run As” then click “Java Application”.
* The text interface will appear on the console in the IDE. Follow the text instructions, which involve inputting a number and hitting enter.

System limitations are about installed software. Certain computer operating systems are not supported by SQL’s Connector/J or cannot run Eclipse/Java/SQL. In terms of hardware, the limit of memory on the computer may hinder total data size in the database and query runtime.

Improvements to this project can be done with the Java application. As Java programming was not the focus of this course, the main Java app has long functions, is only adequately segmented, and not all errors and constraints have catch statements. This lack of focus on Java also made the interface difficult to progress quickly. A local website attached to a database is one area this could be improved upon.

Regarding SQL, I feel alright with query accuracy except for the final query about nouns – which I have yet to find a direct SQL answer. The SQL queries seem to work for simple questions.