Abstract on Finalization Phase of Development of a Data Mart using MySQL

Structuring of stored information or data is an essential part of how companies, corporations, organizations etc. work in today's connected world, important information can be inferred from stored data. Various tools have been developed for analysis of stored information in order to make important business and strategic decisions such as Microsoft Power BI. Databases are widely used method of managing stored structured data. This project is about development of a database which will be used to store and manage structured data of rental services such as Airbnb, the development of this data mart project has greatly improved my understanding of databases using MySQL. My understanding of databases by working on this project using MySQL includes but not limited to: -

- ER modelling of database systems
- Database Normalization
- Database creation
- Database population with data.

The development of this project was carried out in the following stages: -

Conception phase: - In this phase, I researched about online rental organization by visiting Airbnb online website to get information about reservation types such as request and instant book. Room types such as entire place, private rooms, shared rooms and Hotel rooms, Locations, Facilities, Rules and so on.

After getting information about rental organizations, I proceeded by designing an Entity Relation (ER) diagram using UML class diagram using an online resource, this diagram displayed tables, columns and relationships between them. This diagram would form the basis on which the working database will be built upon in development phase. The UML diagram contains the following tables: -

- Customer, address, country and dependents tables: - these tables are used to store information about customers names, age, address, dependents and so on.
- Login, login details and security questions tables: - these tables are used to store information about customer login details such as username, email address, password, login history, security questions and so on.
- Reservation, payment and review tables: - these tables are used to store information such as customer's type of reservation, type of apartment, mode of payment, payment status, reviews and so on.
- Room, room address, facilities, amenities, entertainment and rules tables:- these tables are used to store information about type of rooms, total bedrooms and bathrooms, location of room, room rules, facilities and amenities that the room contains such as TV's, sound systems, pools, parking space, beach front and so on.
- Room owner and room owner address tables: - these tables contain information about the owner of rented rooms such as name, sex, age, address and so on
- Staff, post inspection, entertainment after use, facilities after use and amenities after use tables: - these tables contain information about the rental organization staffs assigned to each customer to inspect their rental locations after they check out to take a record of facilities, amenities and entertainment instrument statuses.

Development phase: - In this phase, I started firstly by normalizing the database. Normalization optimizes the structure of a database and the normal form of a database is a measure of this structure. It provides information on how redundant data are stored in a database (Kneuper. pg. 65. 2021). This process led to some modifications from the previous UML class diagram designed from conception phase, these changes consist of: -

- Removal of city table
- Addition of amenities after use table which is used to record data of state of amenities after a customer has checked out
- Addition of facilities after use table which is used to record data of state of amenities after a customer has checked out
- Addition of entertainment after use table which is used to record data of state of amenities after a customer has checked out
- Renaming of attributes in security questions table from first, second and third to nickname, pet name and birth city respectively.
- Adding of more foreign keys in order to maintain referential integrity on deletion of records which means that after deletion, there must be a primary key for every foreign key.

In the process of formulating queries for creation of tables in the database, I discovered that tables with no or less foreign keys needs to be created first before tables with many foreign keys so as to avoid errors when executing queries for tables creation, still not all foreign keys could be added in create table statements which lead to executing of alter table queries in order to add remaining foreign keys after tables has been created.

After tables has been created in the database, insert statements were executed to insert data across tables, I inserted data across tables at once using transactions so as to maintain uniformity across tables, maintenance of referential integrity and in order to have the flexibility of using rollback statement to undo changes when error occurs in the process of inserting data across tables which made data insertion easier and faster, this method of inserting data into the database made error handling while executing queries easier and faster.

While inserting data into tables in the database, I realized that all foreign keys could not be inserted all at once while inserting data, this led to the execution of update statements after insertion of data into the database. I also executed update statements using transactions in order to have access to the flexibility of rollback when error occurs. For example, while executing update statements, I made mistake of repeating of foreign key values into tables, However, this attribute has been restricted by using “UNIQUE” statement which led to errors. A rollback was initiated after which I corrected these values and the query was successfully executed later.

As a submission requirement and in order to infer useful information from data stored in a structured manner, sometimes data need to be selected and joined across tables in a single query. To that effect, test cases were formed by formulating select statement that joins data across tables.

References

Prof. Dr. Ralf Kneuper., (2021) DLBCSDMDS01: Database Management and Database Systems Coursebook. IUBH Internationale Hochschule GmbH