

Data225 DB Systems for Analytics

Lab is for a group project.

Lab 1

We will use Airbnb like vacation rental system for your team projects, we call it SJSU rental Database (SRD). The project keeps track of each listing and host information.

The database keeps track of every listing, host, location, coordinates, room type and price. The database is designed for managerial staff who have instant access to the rentals. Each customer will pay a fixed amount of 9.95 per month. The billing system is automated so that monthly statements can be generated for all valid customers. Assume the account is single which means that only one customer owns an account. You will need to keep track of customer service number whenever a customer reports a problem related to any account related issues.

For Airbnb data set use the following Kaggle data set:

<https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data>

The above dataset contains the following data:

id, name, host id, host name, neighborhood group, neighborhood, latitude, longitude, room type, price, minimum nights, number of reviews, last review, reviews per month, calculated host listings count, availability 365

As the first step, your project team must define the requirements for this application system -- what will it do, what won't it do (i.e., what are its limitations), how should it work from the user's point of view, etc. (Think of Airbnb as your model)

Next you must do a conceptual design of the database using the Entity Relationship model. You should include all the entities and relationships with key attributes. This step is important to get an idea of the size and complexity of the required database.

You and your team must also do a functional analysis to identify the primary functional components that must be implemented for your application system. The functional components are the major building blocks for the application system; they are not individual programming language functions.

For each functional component, identify the interactions with the database that each functional component will require. Write SQL codes/queries for each functional component and list its inputs, outputs, and interactions with the database.

Cloud-native is a modern approach to building and running software applications that exploits the flexibility, scalability, and resilience of cloud computing. The cloud-native approach to building and running software was pioneered by a group of companies commonly referred to as “born in the cloud” — such as streaming giants Netflix and Spotify, ride-hailing company Uber, and accommodation booking platform Airbnb. The cloud-native approach has since been adopted by

other companies looking for similar digital agility and disruptive competitive advantage.
Hence migrate your SRD database to AWS.

Project Report

The project proposal should clearly define the application system that your team is proposing and the results of your requirements analysis and conceptual design. The proposal does not have to be long, three to five typed pages plus supporting diagrams and the like is probably sufficient. Be concise, but thorough.

The project report should include the following sections:

1. Problem Statement - what application system does your team to build and why; why is a database needed as part of the system.
2. Solution Requirements - what are the requirements for a solution, what will the system do, what are its limitations, how will people use the system.
3. Conceptual Database Design - discuss the requirements for the database in your applications system, and the conceptual design for this database. Include your Entity Relationship diagram.
4. Functional Analysis - discuss the functional components of the application system that you are proposing and how they collectively solve the problem. Include database interactions for each.
5. Tables structure, access privileges, SQL codes/Queries, Triggers, Stored Procedures, logging.
6. Move your database to AWS and show that DB connection (Embedded SQL) works

Grading Criteria

1. Requirements and specifications
2. ER diagram (Entities and Relationships), Normalization, Representing Pk and Fk
3. Tables Structure (Fields, attributes, constraints), access privileges
4. Usage of Triggers, Procedures (> 5)
5. SQL Code/Queries (> 10)
6. Connectivity to AWS using Python
7. Logging of DB
8. SQL performance Measurement
9. Submit Project -1 report (Report should be in IEEE format)