Uy Tran, Samantha Vu
CS 166 Fall 2019
December 6th, 2019

CS 166 Final Project:
Hotel Database Management System

Introduction

This project designed and implemented a hotel database management system. This system allows hotels to store, organize, and access important information on things such as the booking of rooms, the maintenance of rooms, hotel management, and their customers.

ER Design

Linked here is the ER model.

Folder Structure

- `data` - contains the data in csv files.
- `java` - contains the Java Database Connector code.
  - `DBproject.java` - implements user interface and queries.
  - `compile.sh` - shell script to compile and run the Java program.
- `postgresql` - contains shell scripts to start, set up, and stop the database.
- `sql` - contains `create.sql`, which creates the relational schema.

Running the Database

1. Start and setup the database by running:
   ```
   $ source ./startPostgreSQL.sh
   $ source ./createPostgreDB.sh
   ```
2. Copy the data files into the database's temp folder:
   ```
   $ cp *.csv /tmp/$USER/myDB/data/
   $ psql -h localhost -p $PGPORT $USER"_DB" < create.sql
   ```
3. Run the client application:
   ```
   $ ./compile.sh
   ```

Queries: Implementation, Assumptions, and Challenges

1. Query 1: Add new customer

   In order to add a new customer, we created a SQL query to insert a new value into the Customer table. The user is prompted to enter the values of the Customer's attributes, such as their first name, address, phone number, etc. Additionally, a unique Customer ID

is generated for the new Customer via the `executeQuery0` function. This function returns the last used Customer ID, and the incremented value is used for the next added Customer. Finally, if the query does not execute, such as because of an input error (e.g. wrong data type), it will return with an error message.

2. Query 2: Add new room

   To add a new room, a SQL query is generated to insert a new value into the Room table. The user is prompted to enter the values of the Room's attributes, and a unique room number is generated for it. If the query does not execute, any error messages will be printed out. A possible error for this query would be if the hotel ID the user enters does not exist, because Room is dependent on the Hotel entity.

3. Query 3: Add new maintenance company

   To add a new maintenance company, a SQL query is generated to insert a new value into the MaintenanceCompany table. The user is prompted to enter the values of the Maintenance Company's attributes, and a unique company ID is generated for it. If the query does not execute, any error messages will be printed out.

4. Query 4: Add new repair

   To add a new repair, a SQL query is generated to insert a new value into the Repair table. The user is prompted to enter the values of the Repair's attributes, and a unique repair number is generated for it. If the query does not execute, any error messages will be printed out.

5. Query 5: Add new booking

   To book a room, a SQL query is generated to insert a new value into the Booking table. The user is prompted to enter the values of the Booking's attributes, and a unique booking number is generated for it. If the query does not execute, any error messages will be printed out.

6. Query 6: Assign house cleaning staff to a room

   To add a new house cleaning assignment, a SQL query is generated to insert a new value into the Assigned table. The user is prompted to enter the values of the Assigned attributes, and a unique assignment ID is generated for it. If the query does not execute, any error messages will be printed out.

7. Query 7: Raise a repair request

   To add a new repair request, a SQL query is generated to insert a new value into the Request table. The user is prompted to enter the values of the Request's attributes, and a unique repair ID is generated for it. If the query does not execute, any error messages will be printed out.

8. Query 8: Get number of available rooms

   To get the number of available rooms, the user is first prompted for the query's hotel ID. Then, a SQL query is generated by counting the number of rooms in a hotel that are not

in the Booking table . If the query does not execute, any error messages will be printed out.

9. Query 9: Get number of booked rooms
   Given a hotel ID, this returns the amount of rooms booked by looking into the Booking table. This verifies that the input is of type int, but does not take into consideration if the inputted ID is a preexisting hotel ID. There may be multiple bookings for the same room on different days, but that is taken care of by using the "DISTINCT" operator.
10. Query 10: Get room availability for a week
    Given a hotel ID and a date, this finds all rooms that are available for the week, starting for the day. This verifies that something resembling a hotel ID and a date are inputted. However, this does not verify that the begin date is before the end date.
11. Query 11: Get top k rooms with highest prices for a date range
    Given a number K and a date range, this finds all of the rooms and sorts them by price, descending. This query then returns the first K of them.
12. Query 12: Get top k highest booking prices for a customer
    Given a number K and a customer name, this finds a customer ID that has the same name as the given name. The booking prices of the customer are found and sorted to return the top K prices.
13. Query 13: Get customer total cost incurred for a given date range
    Given customer name and a date range, this finds a customer ID that has the same name and sums up their expenses over the date range.
14. Query 14: List the repairs made by maintenance company
    Given a maintenance company, this finds all IDs that correspond to companies with that name and lists all of their collective repairs. This may have a problem if multiple IDs have the same name.
15. Query 15: Get top k maintenance companies based on repair count
    This sorts all the Maintenance companies based on how many repairs they made and then returns the top K of them.
16. Query 16: Get number of repairs incurred per year for a given hotel room
    Given a hotel ID and room number, this returns the amount of repairs that room has had for each year.

Additions to Error Handling

- Date and int validation on the last 8 items
- Specific values are not verified, but the types and format are
- If improper format, loops until it is correct

Side notes:

- Look up with customer name is possible implemented in code but customer names might not be unique
- Maintenance company names are also used when they might not also be unique

Author Contributions

- Samantha: Queries #1-8 and Report (‘Introduction,’ ‘ER Design’, ‘Folder Structure,’ ‘Running the Database,’ and ‘Implementation of Queries #1-8’)
- Uy: Queries #9-16, extra error handling, and Report (‘Implementation of Queries #9-16’, ‘Additions to Error Handling’, ‘Side Notes’)