Database Design Term Project

CS6360 Fall 2017

Project Description

Design, develop, and test a database for ABC hotel. The project consists of four parts: conceptual database design (Phase I), logical database design (Phase II), Oracle relational database implementation (Phase III), and final report &demo (Phase IV).

1. The hotel has a group of employees. Each employee has a unique ID number (9-digit number), name, age, address (street number, street name, city, state, zip code), and salary rate. The employees are categorized as discussed below.
2. Employees are categorized based on their job functions: management, reception, dining, housekeeping and concierge, there are also other types (tech support, accountants, etc.) of employees who do not fall into one of the above main types.
3. For each management team member, the system records his/her title (general manager, customer relationship manager, revenue executive, event manager, etc.).
4. For each receptionist, the system tracks a list of languages he/she can speak and read.
5. For each dining staff, the system keeps his/her shift (morning, afternoon, evening, night). Dining staff can be further categorized based on the type of dining they serve: lounge/bar, fine dining restaurant, buffet, catering, etc.
6. For housekeeping staff and concierge, the system tracks their years of experience (integer between 0 to 50). For tech support and accountants, the system tracks the licenses they have obtained.
7. The hotel has two main types of clients: individuals and organizations.
8. The system tracks the information of each individual customer: unique ID (6-digit number), name, sex, phone (may be multiple phone numbers, in the format of '(xxx)xxx-xxxx'), and date of birth (both in the format of 'MM/DD/YYYY').
9. An individual customer may or may not be a member of the hotel's rewards program. Information about the membership is recorded for a customer if he/she has one. A

customer may have multiple membership numbers.

10. The hotel tracks the information of hotel rooms. Each room has a unique room number (4-digit number), bed type (twin, king, double queen, etc.), room type (standard, premium, suite, etc.), and per-night price (in USD).

11. Housekeepers clean hotel rooms, the date, time and the housekeeper's ID are recorded every time a room is cleaned.

12. Receptionists help individual customers check in to their hotel rooms. An individual customer can be helped by different receptionists check in to different hotel rooms during multiple stays at the hotel. Each time a customer is checked in, the check-in date ('MM/DD/YYYY'), time ('HH-MM-SS'), key type (card key or digital key), lounge access (Yes/No), and length of the stay are recorded.

13. An individual customer needs to pay his/her bill at check out. Date issued, check-in and checkout date, bill amount in USD are recorded for each bill. A customer may make multiple payments to pay off one bill. The date, time and the payment amount in USD are recorded for each payment.

14. An organization client is uniquely identified by an ID (6-digit number), and may or may not have a direct bill account with the hotel. The account number is recorded if a client has one. A client may have multiple direct account numbers, but the account number itself cannot uniquely identify the account.

15. Organization clients may hold events at the hotel. An organization can hold multiple events, and an event can be held by multiple organizations. For each event, a deposit must be paid to the hotel, and the amount in USD is recorded.

16. The hotel assembles event staff to help serve the events. Four main types of event staff come from management, catering, tech support and accountants. Every event staff member is equipped with an on-call speaker and the on-call number (4-digit number) is recorded in the system. An event staff member can serve multiple events. Each event can be served by a group of event staff, and is uniquely identified by an event ID (4-digit number). The date and time of an event are also recorded. An event manager from the management team is assigned for each event; the manager's ID (9-digit number) is recorded.

17. An accountant prepares a bill for each event. Each bill has a unique ID (6-digit

number), date issued ('MM/DD/YYYY') and total amount in USD. An organization may have multiple bills to pay, and each bill may be paid by multiple organizations. An organization can make multiple payments to a bill. Each time an organization makes a payment, the system records the type of payment (cash, check, credit cards, etc.), amount in USD, date and time of the payment.

Project Questions

a) Can you think more rules (other than the one explicitly described above) that are likely to be used in the system?

b) Is the ability to model super-class/subclass relationships likely to be important in such environment? Why or why not?

c) Justify using a Relational DBMS (eg. Oracle) for this project.

Project Phases

I. Draw an EER diagram to accurately represent this set of requirement. This will be your Conceptual Design. Clearly specify any assumption that you are making. You can use any tools (software) to draw the EER diagram. You don't need to describe the value constraints of the attributions in the EER diagram. (20%)

II. Design Logical model for the database. Perform the following steps. (20%)

a. Convert your Conceptual model to an implementation model that can be implemented in a relational DBMS like Oracle. During this process you replace M-N relationships and multi-valued attributes with constructs that can be implemented in the relational DBMS. Draw EER diagram after your modifications. Feel free to change your conceptual model (first delivery) if needed.

b. Document your design in Database Schema format like the one we discussed in the class. Briefly justify your choice of mapping rules.

III. Use a relational DBMS to implement the database. (45%)

a. Normalize all of your tables to third normal form. Make any necessary changes to the EER. Explain why these changes needed to be made.

b. Draw a dependency diagram for each table.

c. Write SQL statements to create database, tables and all other structures. Primary keys and foreign keys must be defined appropriately. The quantity constraints of the relation between the entities, which should be described in EER diagram, are not required.

d. Populate your database to test the following views/queries. You don't need to submit your data in report though.

e. Use the Create View statement to create the following views:

1. Available rooms: show the available rooms in the hotel.
2. Popular event manager: show the popular event managers who have helped organize more than 10 events in one month.
3. Frequent customers: show the individual customers who checked in at least 10 times this year.
4. Popular rooms: show the rooms that were checked in at least 30 times this year.

f. Answer the following Queries. Feel free to use any of the views that you created in previous part.

1. Retrieve the number of employees who work at the lounge/bar.
2. Retrieve the average salary of the receptionists.
3. Retrieve the information of individual customers who have been billed more than $1,000 in total this year.
4. For each individual, retrieve his/her bill amount in ascending order of each check-in date.
5. Retrieve the information of the frequent customers who have stayed for at least 15 nights this year.
6. Retrieve the average age of individual customers who were helped by a receptionist who only speaks Spanish.
7. Retrieve the information of the organization that organized at least two events and got bills of over $2000 in total.
8. Retrieve the highest amount of bill of the events helped by the most popular event manager.

9. Retrieve information of the event that each of its organizers pays the highest amount for the event (suppose organizers of the same event pay the bill evenly).
10. Retrieve the date and time the most popular room was last checked in.

IV. Document the final term project report and demo. (15%)

a. Problem description (Copy it from project description).
b. Project questions (Answer questions listed in this project).
c. EER diagram with all assumptions.
d. Relation schema after normalization. All relations must be in 3NF. The relation schema should include primary keys as well as foreign keys (if any) for all relations.
e. All requested SQL statements.
f. Dependency diagram.
g. Demo.