CS4400: Database Systems
Homework 6
Database design and application, part 1

Due November 8, 2016, 11:59pm

Instructions: Upload your submission as a PDF (.pdf) and plain text file (.sql) on ICON under Assignments > Homework 6.

Files to submit

- hw6.pdf
- create-hotel.sql OR create-bank.sql

Goals for this assignment

- Design a database, considering the needs of an application.
- Model a scenario using an Entity-Relationship diagram, and use it to inform a relational model.
- Identify and fix anomalies in a relational model that you have designed.

Description

You will design and implement a basic database for one of two applications: e-banking and hotel management.

HW6 and HW7 really comprise a single database project. You will work on the project as a team of 2. You will pick teams in class. **Submit only one HW6 per team.**

You will pick **one of the following** applications to be used in HW6 and HW7.

Hotel Management System

Design a database to manage the rentals of different rooms available in a hotel. Relevant details about each room need to be stored, such as a description and cost. Customers may search this database and use it to make reservations. Managers may use the database to manage housekeeping and check ins and check outs. Feel free to expand upon this basic scenario as appropriate.

Required queries to support

a. view the currently occupied rooms
b. view the room types and costs that are still available
c. calculate the total cost for a guest at checkout time
d. list future reservations for a guest
e. list house-keeping assignments

Required transactions to support

a. check-in a guest (must be day of reservation)
b. check-out a guest (if the checkout is early then calculate the refund to the guest)
c. make a reservation
d. cancel a reservation (denied if same day)
e. record that a room has been serviced for the day; no room should be serviced twice in the same day

E-banking

Design a database for e-banking. Customers will use the database to conduct transactions such as transfers between their account(s). Management will use it to calculate, for example, interest to be deposited at the end of each month and to calculate fees, based on number of transactions, to be deducted. Feel free to expand upon this basic scenario as appropriate.

Required queries to support

a. calculate the total deposits and withdrawals from the whole bank in a given time range
b. view balance of a customer account
c. view transactions from latest month for a customer
d. view the total balance of all of a customer's accounts

Required transactions to support

a. add an account for a customer (every customer must have at least one account)
b. withdraw fees from every customer account and put them into the bank's master account. Fees are assessed according to transactions in the last month
c. deposit interest into every customer account taken from the bank's master account. Interest rate is determined by the account type and the interest rate for that type of account.
d. update current fees for each kind of transaction
e. transfer money between two accounts owned by the same customer

What you must do in part 1

1. (10 points) Create an E/R model for your application. You must adhere to the following requirements.
    a. You should not duplicate attributes. For example, in the University scenario from class, only the Faculty entity set has univID; Evaluations and Departments do not also have the univID attribute. You must rely on relationships to link entities sets together, not attributes.
    b. Your diagram should be legible. A computer-aided drawing is preferred. If you draw it by hand, make it large, use a ruler or graph paper for straight lines, write clearly, and make sure all shapes are distinct.
    c. You should label your E/R diagram with multiplicities on the edges, as well as constraints where necessary for modeling the scenario.
    d. You must include the entity sets and relationships necessary to support all required queries and all required transactions.

2. (10 points) Create a relational model from your E/R. You must:
    - Minimally, include the name and schema of every relation
    - Make sure your relations are in BCNF to the degree possible. You may show the progression of relations you chose as you normalized them.
    - Write about anomalies that came up that you fixed and those that you did not fix and why.
    - For each relation in your final model, list *all* functional dependencies. You need not include trivial functional dependencies.
    - Write create table statements (they should work in PostgreSQL) and include your constraints (e.g., primary/foreign keys, on update, not null, check). Put these statements in a file called create-hotel.sql or create-bank.sql.

3. (5 points) Write a testing plan for your database.
    - List situations you will need to cover in your tests
    - List the specific errors that could occur due to concurrent access to the database
    - In the process of writing this plan, you may find constraints you can add to your model

Looking forward to part 2 (HW7)

You will build your application by writing a database client in either Python or Java for issuing the queries and transactions to PostgreSQL. You need not include the SQL queries/transactions for your application in your HW6 submission, but you might start working on them.