Fall 2015 CS157A
Introduction to Database Management Systems

Term Project

Instructor: Dr. Kim

Overview

In this project, you are going to develop a non-trivial database and its application to demonstrate your knowledge in the subjects taught in this course.

The software stack includes MySQL for DBMS, JDBC for connector, and Java program as a front-end application. The Java application can be console-based. If your application supports a GUI for user interactions, 10 points of extra credit (out of 100 points of the project grade). Since GUI is not a main focus of this project, the GUI of your application may be simple but working to test every function your system supports. Note that you need to stick to one type of input interface either GUI or console not both. As described later, executing SQL DDL scripts to create database, tables, triggers, and stored procedures will be done at MySQL directly. For this, you may use MySQL shell or MySQL workbench.

I recommend you to develop the application in an incremental way. For example, after constructing the database (or part of the database), you may try a small java program that pops up a text field or a button to take a user request and see if the program connects to the database through JDBC correctly and retrieves data by executing the expected sql program.

Definition of User

- A public user. An example of such user is a potential guest who wants to make a reservation.
- An administrator is another type of user accessing your application. For example, a hotel manager may use your application to find the statistics of room reservations for this week (or month, etc).
- DBA - DBA functions can be done at the DBMS directly. In this project, the creation of database, tables, triggers, and stored procedures is going to be done by executing the corresponding DDL scripts at MySQL.

Minimum Requirements

- The system should support at least 15 distinct functions to the users. Here the users means public users and the administrator of the application, not including DBA.
- The database involves at least 5 relations and total 15 attributes. There should be relations connect one relation to at least one other relation. The Loan relation in our case study is such an example, that is, Loan connects User and Book.
- Your system should be able to handle at least 5 significantly different queries involving different relations and attributes. Make sure to have at least one co-related subquery, group by and having, aggregation, outer join, and set operation. At least **three** of them must involve several relations simultaneously.
- All schema should come with a key constraint.
- Reference integrity constraints are imposed on all possible cases to avoid dangling pointers. Please avoid circular constraints.

- Define at least two triggers in the database
- In large database systems, it is very common that their data grows over time and an archive function, which copies older entries into an archive database, will be useful. You will follow a simple approach to implement this function. Supply one additional column called updatedAt to the relation you want to archive from. The value of this column will be set to the current time stamp whenever a tuple is created and modified, respectively. And write a stored procedure that takes a cutoff date as a parameter and copies all tuples that haven't been modified since the cutoff date into the archive. You may create a table serving as an archive when you construct the database in the first place. You must do this using transactions to prevent inconsistencies in your DB due to unsafe deletions. In other words, you need to write a stored procedure that include a transaction that will do the copy and deletion.
- Ideally, you pick a domain where lots of data is already available. However, the goal of this project is to assess your ability to construct the database, write queries to manipulate database and to develop its application program. Therefore, synthetic data can be used as long as they can demonstrate all the futures of your database system and its application.

Project Ideas

- Airline reservation, hotel reservation system, tennis court reservation, music band scheduling, library management, product management. Explore your ideas !
- Data Sets
    - Sample data sets for relational database is posted here.
    - DataSet from CKAN

Deliverable

1. Thursday, October 29 11:59
   Submit a hard copy of report that includes
    1. Team name
    2. Members name
    3. Project Title
    4. Database Schema (DDL to create database and relations)
    5. Functional Requirements of your system in English. These are functions to support the business logic of your application.
2. Monday, November 9 5:00pm
    1. Create a set of power point slides that include
        - Database Schema with constraints, also showing the relation(s) to be archived.
        - At least 15 functional requirements and associated SQL programs
        - Stored procedure(s)
        - Trigger(s)
        - Screenshots to show that at least three different user requests can be taken to the the running **Java application** and then be served successfully.
    2. Create a goole doc and upload the powerpoint slides to the google doc.
    3. Email me
        1. with a subject line [CS157A Section 1|2] the order number assigned to your team goes here

```
Example subject line
[CS157A Section1] 1
```

    2. in mail content, write your team name, names of team members, and the URL of your google doc.

3. Tuesday November 10 or Thursday November 12 in class
   Intermediate Demo: Present the intermeidate demo using powerpoint slides you submitted. See intermediate demo guidelines for your preperation.
4. Saturday, December 5, 11:59
   From the course web site, submit project.zip including all Java files, SQL files, and text files containing data. Delete all package statement from Java files so that I can run it using a default package.
5. Tuesday, December 8 in class
   Submit the hard copy of final report in class. Here is the requirements of the final report.