Fall 2014 CS157A
Introduction to Database Management Systems

Term Project

Instructor: Dr. Kim

In this project, you will implement a database and its application to demonstrate your knowledge in the subjects taught in this course.
In this project you are required to implement a non-trivial database and its application. Initially, you will concentrate on the construction of database. The final project will include a Java GUI application through which a user can request/display/add/cancel information from the database.

You will use MySQL for DBMS, JDBC for connector, and Java for high level language to write the front end application. It is ok for you to choose other alternatives as long as your system can satisfy the following requirements.
I recommend you to start with a small version of your application. For example, after constructing the database (or part of the database), you may want to write 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

- DBA - DBA functions can be done at the DBMS directly. (i.e. You may construct database and relations using MySQL workbench). Therefore, DBA may NOT be considered as a user of your application.
- A public user. An example of such user is a potential guest who wants to make a reservation.
- An administrator can be a user who assesses your application. For example, a hotel manager may use your application to find the statistics of room reservations for this week (or month, etc).

In summary, when you think of the functionality of users, you may want to include functions that will be needed by an administrator.

Minimum Requirements

- **The system should support at least 15 distinct functions to the user excluding functions done by DBA (i.e. excluding functions to construct database etc.)**
- The database involves at least 5 relations and 15 attributes(total). 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 an d 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

Deliverables

1. Wednesday, October 29 11:59
   Submit proj.txt that shows the URL of your google document that will contain
   1. Posting date
   2. Team Name
   3. Project Title
   4. Database Schema
   5. Functional Requirements of your system in English. There are two different users to consider: DBA who can load bulk data into data base, change database schema, and backup relations into the archive. General user may look up his/her reservations and cancel reservations, etc.

   What you reported in this google document can be changed (not significantly) over time while conducting the project. **Do not e-mail proj.txt. Write the url in proj.txt and submit it through Project Submission Link on my course web site. Use one of IDs of team members for the submission. Use the same id consistently for your team to make any submission.**
2. Monday, November 3 11:59
   Update your google document as follows: For each function, write a SQL program that implements the functionality.
3. Sunday November 9 11:59pm
   Update the goole documents with the powerpoint slides for the demo. The slides should 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)
4. Demo: Monday November 10 or Wednesday November 12 in class
   A demo signup sheet will be available soon. Do the demo following this guidelines.

5. Saturday, December 6, 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.
6. Monday, December 8 in class
   Submit the hard copy of final report in class. Here is the requirements of the final report.
7. Thursday, December 11
   Final Demo