CS434 – Data Base Theory and Design

Project #3

Author: Mark McKenney
Organization: CS434, Department of Computer Science, SIUE
Points: This assignment is worth 25 points.

Team Database Application (TDA): Part 3 - Schema Creation and Testing

Description

In this project, you will bring your TDA "live," i.e., you will implement the schema and populate it with a few test tuples.

Please make sure to read the assignment carefully and understand exactly what to hand in for each part!

1. *(0 points)* Familiarize yourself with your DBMS of choice, logging into the DBMS, trying some test queries, and experimenting with the help command.

2. *(2 points)* Attach a copy of your most up-to-date E/R schema to your submission of this assignment. Remember, if you have modified your design because of feedback (or any other reason), please hand in the modified design instead; the new design will not be graded but will be compared with your relational implementation.

3. *(13 points)* Implement the SQL database schema for your TDA, using CREATE TABLE commands. Pick suitable data types for each attribute. **Declare foreign keys**. Check your DBMSs manual for data type descriptions.

   - *Hand in a printout of the commands you use to create your database schema* (it is a good idea to keep this file for the remainder of the course). Refer to DBMS documentation for how to enter commands using a typescript rather than interactively. Print out and hand in the contents of your file.
   - In addition, show the response of the DBMS to the request to describe EACH of your relation schemas using the DBMSs describe command *by printing out and submitting a hardcopy of your interactive session* (or creating a PDf of it). For example, to see the schema for relation foo in Oracle type

     ```
     >  describe foo;
     ```

In Postresql type:

```
> \d foo;
```

Alternatively, you can include a screenshot of your database reporting success for each CREATE TABLE statment.

Required: Make sure your schema enforces all the referential integrity constraints that should hold on the relations.

Use **unique**, **primary key**, and **foreign key**-referencing clauses within your create table statements so that the DBMS will enforce the uniqueness of keys and will enforce referential integrity. You may use the default option for handling referential integrity violations (i.e., violations are not allowed and generate errors). In addition, you may specify any other attribute value constraints that you want the DBMS to enforce for you.

4. *(10 points)* For each relation in your TDA, create a file containing a few (approximately 5-10) records of your **real data** (you can extract this data by hand if you do not have auto extraction completed). You can go ahead and enter ALL of your data if you have that capability at this point (then you will be ahead of the game!). Then insert those records as tuples into your relations. Create a script or a file of insert statements that can be executed by the DBMS interface. Some DBMSs provide a bulk loading facility that may be used for this purpose.

5. Turn in a listing showing the contents of the files you created, the successful loading of the data into your database (using a typescript or screenshots of the database reporting success), and the execution of "SELECT *" commands to show the contents of each relation.

Final Comments

Questions 3 and 4 of this project part each tell you what should be recorded in the script that you turn in. In this and all subsequent project parts, the material you turn in should be **clearly formatted** and **delineated**, and should include comments for any aspects that are not crystal clear. **Poorly assembled or documented material will not receive full credit**, even if it is correct. Other than comments, truncation, and simple formatting, it is an Academic Honesty Code violation to edit scripts before turning them in.

When grading, I am looking for sufficient evidence to convince me that you have completed the scope of work outlined in this project description. You can provide forms of evidence other than those listed above (e.g., screenshots of a GUI showing a table exists and has an appropriate number of rows), it must simply prove that you have completed the work.

What to Hand in

As always, you only need to hand in one deliverable per team but make sure you write the **names and e-mail addresses** of all teams members on the assignment.

Don't forget. If your submission contains multiple pages, please staple them together (clearly this only applies if turn in a hard copy in class). Also, it is a good idea to make copies of all your submissions before handing them in.

You must provide enough documentation to convince us that the required items listed above have been completed. This can include screenshots showing the amount of tuples in each table in the database, a typescript showing correct execution of the inserts (and the number of rows affected), or screenshots of the showing the correct execution of the insert statements (and the number of rows affected). We must simply be convinced that the tables are created and the data is in the database.

You **MUST** include your CREATE statements with foreign key/referential integrity constraints and your INSERT statements (or bulk loader data files) with your submission.

Deliverables

A report containing the information outlined above. Also, attach a copy of your most up-to-date **ER Diagram**.