Georgia PPE Management

CS 4400: Introduction to Database Systems

Summer 2020: Semester Project

Project Purpose

In this project you will analyze, specify, design, implement, document, and demonstrate an online system. You are required to use the classical methodology for relational database development. The system will be implemented using a relational DBMS that supports standard SQL queries. You will use your localhost MySQL Server (Version 5.1 or above) to implement your database and the application. You also cannot use any other software like Access or SQLite. Ask the professors or TAs if you have questions.

Project Phases

Inputs (we give you)

• Text description	• Revised text description • Advanced ERD • Raw initial data	• Database schema • Initialized database • Procedure shell
Phase I	Phase II	Phase III
• Entity Relationship Diagram • Logical constraints	• Relational schema • Database schema • Initialized database	• Implemented procedures

Outputs (you turn in)

Version History

Version	**Date**	**Notes**
1	7/7/20	Initial Release

Directions for Phase III

In this phase, each team will implement a list of stored procedures for the Georgia PPE Management System database.

We are providing you with the following:

Title	Description	When is it available?
Procedure Bank	Provides descriptions, points values, and notes for each of the specified procedures.	Now
Procedure Shell	Stubs out all of the procedures and provides space for you to fill in your implementations.	Now
Stored Procedure Guide	Briefly reviews the various parts of stored procedures in a concise one-pager.	Now
Project Description	The same text description from Phase II for your reference.	Now
Database initialization	Provides the DDL to set up the database you will be working with as well as inserts the initial data from Phase II.	Thu, Jul 9, 12:01 am
Basic autograder	Provides a script to sanity check your procedure implementation. More details to come.	Mon, Jul 13, 12:01 am

The only thing you will turn in is the procedure shell. The rest is for your use.

While the Phase II grace period is going on, you may use your own schema to get started, but ultimately, we expect your procedures to be compatible with the schema we release. The schema we give you will be the one we use to initialize our database before testing your procedures.

For each type of procedure, we expect you to hit a certain number of points for full credit, listed below. You are welcome to turn in as many points as you want, but your grade will be capped at 100%. Each procedure will be graded on an all or nothing basis, so you may attempt a few more "just in case." Please delete any procedures you choose not to attempt.

	Full credit	Points available
INSERT	13	15
DELETE	5	5
UPDATE	10	12
SELECT	38	46

You'll notice we have also provided an example procedure for each procedure type. These ones have their implementations filled in and are worth 0 points.

A few reminders:

- Don't change the stored procedure names, any of the parameter names, the order of the parameters, nor the parameter types. These are essential to the interface of your application: if you change them, you would impact how users are able to access your application, and you could potentially impact other systems (e.g., our testing application) that are used to evaluate your submission.
- Some of the parameters for the stored procedures might be empty. In those cases, the input values might be an empty string (i.e., "") or a null value, usually depending on the parameter type. Also, certain parameter values might be limited to specific string values as indicated in an enumeration type. Please refer to the Project Instructions and the stored procedure script that we've provided for more details.
- For deletions, do not try to circumvent referential ON DELETE triggers by manually deleting parent rows. This ON DELETE restrictions are deliberate.
- For selects, the "SELECT * FROM USER" is just a dummy query to get the file to run. You will need to replace that line with your solution.
- For selects, the stored procedure creates a table that has the same name as the stored procedure along with the string "_result" appended to the end. Your query return results must be stored into that table.
    - Do not change the names or parameters of the tables that are used to return the results of your queries.
    - However, you are welcome to move around the “INSERT INTO ___result” command wherever you need it to be, as long as the results of your SELECT are loaded into that result table.

Submission Checklist

Each team needs **one of its members** to upload the deliverables to Canvas. The other team members should log in and check to ensure that all files have been uploaded correctly. Please include your team numbers in the file names.

Your submission should include the following:

1. Procedure shell filled in with your implementations (mysql_team#.sql)
    a. Delete (completely) any procedures you chose not to implement
    b. The .sql file must run in MySQL Workbench without error for you to receive credit for these statements