Banking Management Service

CS 4400: Introduction to Database Systems
Course Project: Spring 2022 Semester

Project Purpose

In this project you will analyze, specify, design, implement, and document an online system based on the provided scenario description. 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 8.0 or above) to implement your database and the application. You cannot use any other software like Access or SQLite. Please ask the Instructors and TAs if you have questions.

Project Phases

Inputs (we provide to you…)

Phase I	Phase II	Phase III	Phase IV *(optional)*
• Scenario description • Sample data elements	• Enhance ERD (EERD) • Initial Data Set (in a non-normalized format)	• Physical database schema with initial data set • View & stored procedure shells	• User interface specification
• Enhanced entity relationship diagram (EERD) • List of assumptions (optional)	• Relational schema • Physical database schema with initial data set • Unhandled exceptions list	• Implemented views & stored procedures • Any supporting views and related structures	• Fully functional application integrated with database system • Application source code

Outputs (…you turn in to us)

Phase III Directions

In Phase III, your tasks are to:

- **Implement 19 stored procedures** which allow the system operators to modify the database state in accordance with the main use case (i.e., managing bank accounts and assets)
- **Implement 5 views** which provide information to the system operators about the database state from various "points of view" (i.e., system operator roles)

Implementation Details

- The only thing you will turn in is your completed procedure shell. The remaining material that we've provided is to support your efforts to develop your views and stored procedures.
- Each view or stored procedure will be graded on an "all-or-nothing" basis. We encourage you to develop a solid plan for implementing the required structures (in general) from the easiest to the most difficult. Submitting fewer procedures that have been thoroughly tested and work well is usually better than lots of procedures that work in an inconsistent/haphazard manner.

- Your views and stored procedures must be based on the underlying tables and foreign keys that we've provided. You are not permitted to modify the tables and foreign keys that we've provided in any way.
- You must maintain the state of the system using the tables that we've provided. There might be other systems that use those same tables to monitor the state of the system, and that shared data must be updated and maintained in a consistent manner. Also, our testing systems will monitor the provided tables to ensure that transactions have been updated correctly.
- You must use the stored procedure and view specifications that we've provided in the shell (i.e., view and stored procedure names, along with the stored procedure parameter names, types and order). These are essential to the interface of your application: any changes might impact how users are able to access your application, and might also adversely impact our evaluation of your submission.
- Keep in mind that some of the input values for some of the stored procedure parameters might be empty. In those cases, the input value will be a NULL. Be sure to check this case and all other potential cases. Review the scenario description and the database structure for more information about the valid ranges for the different parameters.
- You are welcome (and encouraged) to develop other views, functions, and stored procedures as desired. Developing these "supporting database structures" can help make implementing the required views and stored procedures easier by dividing a potentially complex problem into simpler tasks.

Autograder

An autograder will be released later via Gradescope, and we will notify you when it is released. A couple notes regarding the auto grader:

- The autograder is meant to test the main cases or the “happy path” cases. These are the cases that will occur most often in the system. It DOES NOT test all cases, and it is up to you to figure out, test, and implement any edge cases
- If you pass all of test cases on the autograder, that is a good sign that you are on track
- Your score on the autograder is NOT your final score on this phase of the project. It is only an indicator. The teaching staff will use a more advanced autograder, testing all the cases to determine your final score
- To use the autograder, upload your shell file to Gradescope with the name “cs4400_phase3_shell.sql” minus the quotes. If you do not use this name, the autograder will not run. Additionally, if your code has errors, the autograder will not run. Make sure you run your code successfully in workbench before uploading to the autograder
- More details about the autograder will be shared upon its release

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 must include the following deliverables compiled into one document:

- A file named cs4400_phase3_shell_team#.sql containing your completed views and stored procedures, along with any other "supporting database structures"
- Your SQL file must run in MySQL Workbench without error for you to receive credit for these statements – PLEASE test your code before your final submission

Version History

Version	Data	Notes
0	March 18th, 2022	Initial Release