Emergency Room Management System (ERMS)

CS4400: Introduction to Database Systems
Course Project: Fall 2025 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

	Phase I	Phase II	Phase III
We provide you with...	• Scenario Description • Sample Data	• Solution EERD • Initial Data Set (in a Non-Normalized Format)	• Physical Database Schema with Data Sets Inserted • View & Stored Procedure Outlines
You turn in to us... (+ CATME ratings due after each phase)	• Enhanced Entity Relationship Diagram (EERD)	• Relational Schema • Physical Database Schema with Data Sets Inserted • Unhandled constraints	• Implemented Views & Stored Procedures

Phase III Directions

In Phase III, your tasks are to:

- Implement five (5) views which provide information to the system operators about the database state from various "points of view" (i.e., hospital staff roles)
- Implement fifteen (15) stored procedures which allow the system operators to query and modify the database state in accordance with the problem scenario
- Use the “Phase III SP+Views Template.sql” file to write your solutions

Implementation Details

- The only thing you will turn in is your completed procedures/views. The remaining files we've provided are to support your efforts to develop your views and stored procedures.
- Each individual test case for a view or stored procedure will be graded and based in large part on the number of test cases that it completes successfully. 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 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 often be NULL. 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.
- Refer to the autograder tutorial for how to use the autograder in conjunction with the project.

Submission Checklist

Each team needs **one of its members** to upload the deliverables to Gradescope. When submitting, ensure that **all members are added to the submission**. Failure to do so will result in a **2-point penalty** to their overall project score. The other team members should log in to their individual Gradescope accounts and check to ensure that all files have been uploaded correctly and that they are linked to their team's submission. Include your team number in the file name.

Your submission must include the following **file**:

- A file named **cs4400_phase3_stored_procedures_team#.sql** containing your completed stored procedures and views
    - **You must submit the original MySQL statements that you've hand typed, NOT the SQL dump/export file. The file format MUST be ".sql".**
    - **Your SQL file must run in MySQL Workbench without error for you to receive credit for these statements.**

IMPORTANT NOTE: Your submission is graded with hundreds of additional, hidden test cases. ***DO NOT*** assume that you've gotten a 100 once you're passing the basic autograder. You should thoroughly test your stored procedures and views to ensure correctness.

Version History

Version	Date	Notes
1	October 15, 2025	Typo corrected in submission section