

Advanced Features of an EMR Database System

Due 05/03/2024

This project is a continuation of Project 1. In this project, we will develop additional features in the database system to facilitate more robust and efficient application development and to support security, privacy, audit trail and other regulatory requirements of an electronic medical record system.

More specifically, in this project, you are to enhance your database design by developing additional tables, columns, stored procedures, triggers, indexes, views and other database functions that are needed to enable the following new requirements.

- Create an application programming interface (API) using stored procedures that allows an EMR system developer to work with the database without having to write SQL statements
 - One simple way to use MySQL stored procedures in an API is to create a text-based interface with multiple options for the user to access your database functionalities without being able to access/modify the data or using MySQL workbench.
 - You can use any programming language to create the text-based interface. It is important to note that you will need to allow these functions according to the user authorization. For example, you may want to allow doctors to update prescriptions, but not patients.
- Add user authentication so that only authorized users can access the database
- Add some form of role-based access control to limit the functions that a user can perform. You should implement at least one use case to demonstrate this capability. For example, you may allow a physician to delete a prescription order, but not a receptionist. In this case, you have two roles, physician and receptionist.
- Add tables and other features to enable audit trail so that every query or change of every record in the database is monitored and the entire history of the data in the database is captured. Basically, every time a record is accessed (queried, inserted, or changed), the user and time of access is recorded. Every time any field of a record is updated or deleted, the previous value of the record is saved.
- Add indexes and views so that frequently used queries and changes to the database are most efficient.

Deliverables

- Presentation (5~10 min video recording) explaining/demonstrating your system features
- Requirements specification (Word doc)
- UML Data model (be hand drawn)
- ER diagram (Workbench)
- Database creation script (with test data)

- SQL scripts to support EMR system functions (including stored procedures, triggers, views, and indexes)
- All materials should be organized into one zip folder (codes should be in SQL script files, not in word/text forms)
- If you have updated your database from Project 1, please make a note and specify the changes in the specification AND presentation, and include your new SQL scripts, UML data model, and ER diagram

Evaluation criteria

- Functionality (must work correctly first)
- Quality
- Documentation
- Note the rubric file for details