Lab Assignments

Overview

Throughout the semester, you will design, implement, and refine a single, cohesive relational database system called the Campus Event and Resource Management System (CERMS). Rather than treating each lab as a standalone assignment, view this assignment as a single system developed in stages. Each lab introduces new database concepts and technical requirements while building on the same underlying design and implementation.

By the end of the semester, you will have a fully implemented and analyzed database system that demonstrates your ability to design, query, secure, and administer a real-world database application, including:

- Analyze an informal problem description and derive precise data requirements
- Design a normalized relational schema using ER diagrams and functional dependencies
- Implement the same logical design across multiple database management systems
- Enforce integrity using keys, constraints, triggers, and stored procedures
- Write complex SQL queries and views for reporting and analysis
- Manage concurrent access using transactions and isolation levels
- Identify and mitigate common database security vulnerabilities
- Configure database users, roles, and privileges
- Perform basic database administration tasks such as indexing, backup, and recovery
- Justify design decisions using database theory and best practices

Structure

The lab is structured into several assignments over the semester. Each session builds on the work from prior labs. Initial design choices influence later implementation, performance, and security. You should revise and improve your work as new concepts are introduced. The detailed labs and specific deliverables for each phase will be separate documents.

Expectations

- You are responsible for maintaining a consistent and evolving schema throughout the semester.
- Expect to revisit and refine earlier work when new requirements or discoveries are introduced.
- Design choices must be justified, especially when trade-offs are involved.
- The database must enforce rules at the database level, not only in application code.
- Academic integrity applies to all work. While discussion is encouraged, all designs and implementations must be your own.

Outcome

At the conclusion of the course, you will have a complete database project suitable for discussion in technical interviews or inclusion in a professional portfolio. More importantly, you will have practical experience working through the full lifecycle of a non-trivial database system.

Important Guidelines

- Ask questions directly if you need clarification. Do not rely on outside instructors or advisors for interpretation.
- No generative AI may be used in this lab (unless otherwise specified).

- Follow the instructions carefully. There is room for flexibility and extra features, but you must meet the minimum requirements.
- The instructions may require you to interpret requirements and make reasonable assumptions, but not all assumptions are correct.
- Exam questions may extend beyond what is explicitly required in the labs.

Submission

- The deliverables will be submitted to GitHub to allow versioning and watching development.
- For each assignment, submit “done” in Moodle after committing your deliverables to GitHub.
- Do not submit assignment deliverables to Moodle unless otherwise indicated.

Scenario and Problem Description

Campus Event and Resource Management System (CERMS)

A mid-sized university is replacing a collection of disconnected spreadsheets, email workflows, and ad-hoc databases used to manage campus events and shared resources. The new system, referred to as the Campus Event and Resource Management System (CERMS), will serve students, faculty, staff, and administrators across multiple departments and student organizations.

The university hosts hundreds of events each semester, including student organization meetings, guest lectures, workshops, performances, and departmental activities. Events often require reserving physical spaces (such as classrooms, auditoriums, labs, and outdoor areas) and shared equipment (such as projectors, sound systems, cameras, and specialized lab hardware). Conflicts, double bookings, and unclear approval responsibilities are common problems with the current system.

Users and Roles

CERMS must support different user categories. Students may participate in events and may also serve as officers of registered student organizations. Faculty and staff may organize events, approve requests, or manage resources. Administrative users are responsible for system oversight, policy enforcement, and reporting.

Not all users have the same permissions. Some users may create events but not approve them. Others may approve of events but not modify event details. Specific administrative actions should be restricted to a small group of users. Users may have multiple roles depending on their affiliation and responsibilities.

Events and Scheduling

An event has a title, description, date and time range, expected attendance, and one or more associated organizations or departments. Some events require formal approval before they become active. Events may span multiple hours and may recur.

Events may require one or more resources. Resources are limited and may not be reserved for more than one event at the same time. Specific resources have capacity limits or usage restrictions. For example, a room may have a maximum occupancy, or equipment may require special authorization.

Registration and Attendance

Some events require registration, while others are open to all. When registration is required, capacity limits must be enforced. In high-demand situations, multiple users may attempt to register simultaneously. The system must prevent over-registration and maintain consistent data even under heavy concurrent usage.

Attendance data may be collected for reporting and auditing purposes. Not all events require attendance tracking.

Approvals and Workflow

Some events and resource reservations require approval. Approval rules may vary depending on the type of event, the resources involved, or the organizing group. An event may require multiple approvals in a defined order. Approved and rejected actions should be recorded for later review.

Security and Data Integrity

The system must enforce appropriate access controls so users can only view or modify data they are authorized to access. Administrative users should be able to define roles and permissions. The system must protect against common security vulnerabilities, including unauthorized data access and improper query execution.

The university is particularly concerned about data integrity and auditability. Changes to critical data, such as event approvals, reservations, and user permissions, should be traceable.

Reporting and Administration

Administrators require reports on system usage, including event counts, resource utilization, approval timelines, and user activity. The system must support backup and recovery. Performance considerations are essential, especially during peak registration periods.

Project Goal

Your task is to analyze this scenario and determine the functional, non-functional, and data requirements for designing and implementing CERMS as a relational database system. You will translate these requirements into an entity-relationship design and ultimately into a working implementation.

You are expected to identify ambiguities, make reasonable assumptions, and justify your design decisions.