



Apply theoretical knowledge and practical skills to develop database applications using DBMS and SQL language

Effectively use the Entity Relationship Diagram for the representation of conceptual schemas.

Identify functional dependencies and apply normalization algorithms.

Use Data Definition Language to define database schemas.

Write SQL commands to create databases, create tables, insert/update/delete/retrieve rows in a common database management system.

Unannounced pop questions and pop quizzes may be given anytime during class. The purpose of pop questions and pop quizzes is to encourage you to learn, study and review the concepts and materials presented/discussed in the lecture. You will be called to answer pop questions anytime during the lecture. If your name is called and no response or you give incorrect answers, 0 points will be recorded. There will be approximately 8-10 pop quizzes and many pop questions during the semester. Each pop quiz/pop question will be scored/weighted evenly. In the end of semester, the top 80% scores of your pop quizzes will be selected for calculating your final grade of your pop quizzes. In other words, you can drop the bottom 20% of your pop quizzes. However, **if you miss a class and miss a pop quiz, it counts 0 point and it cannot be dropped.** Each missing pop quiz is

programming languages of your choice may be used to build the necessary user interface. Teams must be able to run their projects in class on the day of project presentations and demos.

The following deliverables for the database design and implementation project are due on specified days in the schedule (each report/document format will be provided in the class):

1. Each team must propose a web-based database application and submit a 6~8 pages long narrative , detailing the preliminary analysis for the application, including its basic requirements. The narrative should focus on application functionality, rather than implementation details.
2. Each team must submit a data model with the requirement of minimum 10 relations for the application. The model should be sufficiently complete to serve as the basis for further development. Teams will present their data models to the class.
3. Each team must submit a database design, including tables and relationships among tables. The tables and relationships should be essentially the same as will be used in the final application. Normalization process to the level of BCNF is expected. Teams will present their database designs to the class.
4. Each team must submit the database application they develop. Teams will present their applications to the class.  
It is required to provide a graphical/polished user interface and the application should provide all required functionality, including that needed to maintain data.

Better presentations will demonstrate clearly what the application does and how the team accomplished the project. Applications that do not run will receive poor marks.

Projects must be done in teams and team membership assigned will be considered fixed no switching teams allowed.

By submitting/presenting a project, team members attest that they all participated in the conceptualization and accomplishment of the project. It is incumbent on team members to assure that **each team member MUST contribute on writing program code and documents**

file of code and document), ugh the project. If problems arise during the term, upon consultation with team members, the instructor will remove non-participating team members from their teams. Individuals removed from teams will not receive points on the team assignment.

**Database Team Project: 20%** (Team with peer evaluations)

**Midterm exam: 20%**

**Final exam: 25%** (Accumulative/Comprehensive)

Digit number grades will be assigned according to the following policy:



