San José State University
College of Science / Department of Computer Science
Introduction to Database Management Systems, CS157A-S1, Fall, 2019

Course and Contact Information

Instructor:	Dr. Mike Wu
Office Location:	MacQuarrie Hall 211
Telephone:	(408)924-8144 (Preferred mode of contact is via email.)
Email:	Ching-seh.Wu@sjsu.edu
Office Hours:	Tuesday 1:45pm ~ 2:45pm and Thursday 4:30pm ~ 5:30pm **(Please drop me email with time info and subject.)**
Class Days/Time:	Tuesday and Thursday 3:00pm ~ 4:15pm
Classroom:	Sweeney Hall 100
Prerequisites:	CS 146 Data Structures and Algorithms (with a grade of "C-" or better); Computer Science, Applied and Computational Math major.

Course Description

General: Current, classical database systems. Entity-relationship and enhanced entity models. Relational model, algebra, calculus. Current, emerging SQL standard. Embedded, Dynamic SQL. Application perspective on transactions and security. Interactive and programmatic interfaces to database systems. Application programming project using commercial database system. Prerequisite: CS 146 (with a grade of "C-" or better); Computer Science, Applied and Computational Math, or Software Engineering majors only; or instructor consent.
Course-specific: Database systems represent the primary choice for managing large amounts of data and are used extensively for scientific, engineering, industrial, governmental, financial, and educational applications. This course is an introductory course in database management systems focusing on the application-related aspects of designing, administering, tuning and using databases for engineering data-intensive applications. This course also applies industry-oriented (such as Google, etc.) Project Based Learning (PBL) approach to develop database applications. Students will learn how to design a relational database schema, to create database tables and populates them with real-world data, to use the standard language (SQL) to query the data, to access the data from the application, and to tune the database for specific applications (through schema redesign and creation of indexes and views). While the course primarily focuses on relational databases (SQL), it also introduces students to advanced and emerging technologies such as XML, XQuery, XPath, JSON, RDF, SPARQL, OLAP and NoSQL.

Course Objectives

- To introduce students to the purpose of Database systems and databases, as well as common users of such systems.
- To teach students about the relational model and relation algebra.
- To teach students about design theory (such as normalization, etc.) and algorithms that help

- determine if a given database's tables are organized in a reasonable way.
- To teach students about real-world database system usage, architectures and components. Some example systems that might be considered are: Oracle, MySql, Postgres, Access, DB2, and SQL Server.
- To teach students about SQL, the standard language for interacting with a database.
- To teach students how to interact with a database system from a programming language such as Java, C, PHP, Perl, etc.
- To apply industry-oriented Project Based Learning (PBL) to develop database application software.

Course Learning Outcomes (CLO)

Upon successful completion of this course, students should be able to:

- 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.
- Construct data retrieval procedures using the Data Manipulation Language (schema, index, normalization, view, trigger, constraints).
- Develop data retrieval procedures using Relational Algebra.
- Write simple transactions using JDBC and ODBC, or similar programmer interfaces in other languages.
- Know different types of databases (relational, transactional, big data)
- and how to use the database (querying, application-level data access, transaction, security, authorization) as the backend for data-intensive applications

BS in Computer Science Program Outcomes Supported

These are the BSCS Program Outcomes supported by this course:

- An ability to apply knowledge of computing and mathematics to solve problems
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- An ability to use current techniques, skills, and tools necessary for computing practice

Required Texts/Readings

Textbook

Database Systems: The Complete Book (2nd edition) by Garcia-Molina, Ullman, and Widom
Textbook web site: http://infolab.stanford.edu/~ullman/dscb.html

References

- Database System Concepts (6th edition) by Silberschatz, Korth, and Sudarsha
- Fundamentals of Database Systems (6th edition) by Elmasri and Navathe

Course Requirements and Assignments

Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, and so on.

Assignments

You are expected to learn all of the material presented in the lectures. Written homework and project reports are also a requirement of the course. Homework and project reports must be turned in on time; **late homework and reports will NOT be accepted.** Both homework and project assignments are due at the beginning of the class period on the announced due date.

Pop Quizzes

Unannounced quizzes may be given anytime during class. The purpose of pop quizzes is to encourage you to learn, study and review the concepts and materials we discussed in the previous lecture. These will generally be problems covered in the previous lecture. There will be proximately 8~10 pop quizzes during the semester. Each pop quiz will be scored/weighted evenly. In the end of semester, the top 80% of your pop quizzes will be selected for calculating your final score 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 scored as 0 point and must be used to calculate your final score.

Mid-Term and Final Exams

Exams will consist of questions and problems aimed at assessing student mastery of course topics. Conceptual questions may be in the form of essay or multiple-choice format. Problems will require the production of (or correction of) SQL code, data models, or similar output. All exams are closed book and note.
If you are unable to attend any one of the exams, arrangements may be made only if you have a legitimate reason. You need to inform your instructor ahead of time and have written documentation available. If you are unable to attend the exam due to illness or emergency, you also need to inform your instructor **before the exam** and bring documentation afterwards to request a make-up exam, or the points for that exam will be allocated to other exams.

Database Design and Implementation Project

The course achieves a balance between establishing a theoretical foundation for DBMS and pragmatic applications of DBMS in a real-world environment. A significant semester-long project reinforces lectures and is designed by applying Project Based Learning (PBL) derived from Google's software engineering best practices. The course places particular emphasis on the logical design of relational database systems. In this project, you will apply concepts presented in class and obtain practical, hands-on experience. Students, in randomly-selected, 3-member teams, will design and implement small database applications. A three-tiered architecture of DBMS application will be designed and implemented as a term project for this course. Team may choose any applications that are appropriate in size and complexity. Appropriateness will be determined by the instructor.

The goal of the class project is to implement a database system application. The project involves database design and modeling, creation, data population, query, and database application programming using JDBC. The project includes the following activities spread over the entire semester:

- Identify key features of web-based application area for which database systems may prove beneficial,
- Determine the functionalities of the database application,
- Model the data stored in the database (Identify the entities, roles, relationships, constraints, etc.),
- Design, normalize, and perfect the relational database schema,

- Write the SQL commands to create the database, find appropriate data, populate and manipulate the database, and
- Finally, and most importantly, write the software needed to embed the database system in the application.

The end result should be a functioning application that runs on the web and that uses your database to allow useful functionality.
All projects MUST be implemented with **MySQL** as the DBMS and all project artifacts including **programming code**, **meeting minutes**, and **report documents MUST be stored in Github**. Each student must have a Guthub account and Github will manage and keep track of your project work. Other development tools of your choice may be used to build the necessary user interface. Students 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 an application by submitting a 2-3 page explanation of it.
2. Each team must submit a longer (6~8 page) narrative, detailing the preliminary analysis for the application, including its basic requirements. The narrative should focus on application functionality, rather than implementation details. If the application will support a "real" operation, the team should interview potential users to ascertain their needs. If the application is for a "made-up" scenario, any statement of user needs must be supported with acceptable reasoning.
3. 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.
4. 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.
5. 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 in writing program code and documents** (Github will show each member's contribution to each file of code and document), no one on the team "free rides" through 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.

Available Software and useful Links

- MySQL (http://dev.mysql.com/downloads/installer/5.6.html)
  - MySQL server at http://dev.mysql.com/downloads/mysql/
  - MySQL Connectors - JDBC Driver for MySQL (Connector/J) at http://www.mysql.com/products/connector/
  - MySQL WorkBench at http://www.mysql.com/products/workbench

- Github Account Creation (https://github.com/join)
    - Project Management Using Github: https://github.com/features/project-management/
    - What is Git | What is GitHub | Git Tutorial | GitHub Tutorial | Devops Tutorial | Edureka: https://www.youtube.com/watch?v=xuB1Id2Wxak
    - Github Tutorial For Beginners: https://www.youtube.com/watch?v=0fKg7e37bQE

- XML Editing and Validation Tool: XML Copy Editor at http://xmlcopy-editor.sourceforge.net/
- XSLT and XQuery Processor: SAXON at http://saxon.sourceforge.net/
- GUI Interface to use SAXON: Kernow at http://kernowforsaxon.sourceforge.net/

Grading Information

Determination of Grades

The components of the final grade will be distributed as follows:

- Class Participation: 10% (pop quizzes, pop questions, discussions, interaction with instructor, etc.)
- HW Assignments: 15% (Individual)
- Database term project: 25% (Team)
- Midterm exam: 25%
- Final exam: 25%

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

96 ~ 100	----	A+
92 ~ 95	----	A
90 ~ 91	----	A-
86 ~ 89	----	B+
82 ~ 85	----	B
80 ~ 81	----	B-
76 ~ 79	----	C+
72 ~ 75	----	C
70 ~ 71	----	C-
66 ~ 69	----	D+
62 ~ 65	----	D
60 ~ 61	----	D-
0 ~ 59	----	F

Each assignment, project, and exam will be scored (given points) but not assigned a letter grade.
Final individual class letter grades will be assigned based on the class curve. Your final class grade can be adjusted up or down depending on your level and quality of class / project performance.

Classroom Protocol and Other Notes

- **Absences in attending anyone of the first two lectures will be instructor-dropped out from the class.**
- Every student must attend class and participate actively.
- You will be called in most class sessions for Pop questions and to discuss material contained in lectures by using Random Roster Checker.
- **When emailing me, please always start your email subject line with "CS157A: XXXXX" to get my attention. (XXXXX: Subject, for example: CS157A:HW1 Question)**

- **Plagiarism/Cheating will not be tolerable: 'F' will be given to your FINAL COURSE GRADE and will be reported to the Department and the University. (please be noted: Obtaining HW solutions from someone or giving/showing your HW solutions to someone is also treated as Plagiarism/cheating.)**
- **Your laptop must remain closed** (preferably in your backpack and not on your desk). You will lose points from Class Participation.
- To encourage participation from students, **no recording** is allowed.
- Students must be respectful of the instructor and other students. For example: turn off/silence **cell phones and other mobile devices**.
- **Attendance is crucial to doing well on pop quizzes, assignments and examinations.**
- Students are responsible for all materials distributed on Canvas and discussed in the class.

Attendance: University policy F69-24 at http://www.sjsu.edu/senate/docs/F69-24.pdf states that students should attend all meetings of their classes, not only because they are responsible for material discussed therein, but because active participation is frequently essential to insure maximum benefit for all members of the class.

Consent for Recording of Class and Public Sharing of Instructor Material: University Policy S12-7, http://www.sjsu.edu/senate/docs/S12-7.pdf, requires students to obtain instructor's permission to record the course: Common courtesy and professional behavior dictate that you notify someone when you are recording him/her. You **must** obtain the instructor's permission to make audio or video recordings in this class. Such permission allows the recordings to be used for your private, study purposes only. The recordings are the intellectual property of the instructor; you have not been given any rights to reproduce or distribute the material. Course material cannot be shared publicly without his/her approval. **You are not allowed to publicly share or upload instructor generated material for this course such as exam questions, lecture notes, or homework solutions without instructor consent.**

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/" Make sure to review these policies and resources.

Introduction to Database Management Systems, Fall 2019, Course Schedule

Tentative Course Schedule (This schedule is subject to change with fair notice.)

Week	Date	Topics, Readings, Assignments, Deadlines
1	08/22	Motivation, Orientation /Syllabus, Course Introduction (Student's Information Due)
1	08/27	Introduction to Database Systems Concepts (Github account creation for team project due)
2	08/29	Relational Model, SQL Data Definition, XML, DTD (Project team assignment and formation, 3 students per team)
2	09/03	Entity/Relationship Model Project Proposal Discussions
3	09/05	Entity/Relationship Model (Project Proposal Doc Due in Github) (HW Assignment 1 Out)

Week	Date	Topics, Readings, Assignments, Deadlines
3	09/10	Speakers (Google FIR Team) from Google Headquarters, Mountain View, CA (Google opportunities, hiring process, interview, Project Based Learning, etc.)
4	09/12	MySQL Installation and Basic Administration Functional Dependencies (HW Assignment1 Due)
4	09/17	Functional Dependencies (Project Requirement Document Due in GitHub)
5	09/19	Normalization, BCNF (Runnable three tiered architecture due for demo and code due in Github)
5	09/24	3rd Normal Form (HW Assignment 2 Out)
6	09/26	Inference of Dependencies, Multivalued Dependencies Project Discussion: Schema Design, ER Conversion
6	10/01	SQL Programming (HW Assignment 2 Due)
7	10/03	SQL Programming (Project Data Model & DB Design Document Due in Github)
7	10/8	Constraints & Triggers
8	10/10	Views and Indexes (HW Assignment 3 Out)
8	10/15	Views and Indexes
9	10/17	First Project Code Review (revision code due in Github) (HW Assignment 3 Due)
9	10/22	JDBC, PHP, CLI, Embedded SQL
10	10/24	**Midterm Exam**
10	10/29	Midterm Solutions
11	10/31	Relational Algebra Second Project Code Review (revision code due in Github)
11	11/05	Relational Algebra
12	11/07	XQUERY, XPATH (HW Assignment 4 Out)
12	11/12	Querying XML, JSON, RDF
13	11/14	JQuery, SPARQL (HW Assignment 4 Due)
13	11/19	OLAP, Dynamic DB Third Project Code Review (revision code due in Github)
14	11/21	Fundamental concept of Big Data and NoSQL
14	11/26	Final Project Presentation & Demo (HW Assignment 5 Out)
15	11/28	Thanksgiving Holiday - Campus Closed
15	12/03	Final Project Presentation & Demo (HW Assignment 5 Due)
16	12/05	Final Project Presentation & Demo Final Exam Preparation & QA Final Project Code Review (final revision code due in Github)

Week	Date	Topics, Readings, Assignments, Deadlines
Final Exam	12/11	Wednesday 2:45pm – 5:00pm (Final Project Report Due in Github)