San José State University
College of Science / Department of Computer Science
Introduction to Database Management Systems, CS157A-01, Fall 2026

Course and Contact Information

Instructor:	Prof. Mike Wu
Office Location:	DH 215A
Email:	Ching-seh.Wu@sjsu.edu
Class Days/Time:	Tuesday and Thursday 4:30pm ~ 5:45pm
Office Hours:	Tuesday & Thursday 5:45pm ~ 6:45pm **(Please drop me email with time info and subject.)**
Classroom:	In-Person, Duncan Hall 415
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 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 **Google's Team Project Based Learning (PBL)** to develop a web-based 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).
- Write SQL commands to create databases, create tables, insert/update/delete/retrieve rows in a common database management system.
- Develop data retrieval procedures using Relational Algebra.
- Write simple transactions using JDBC or similar programmer interfaces in other languages.

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 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.

General Expectations & Attendance

- Students are expected to master all material presented during lectures.
- Detailed requirements for each homework assignment will be explained in class before they are officially posted on Canvas. It is the student's responsibility to attend lectures and pay close attention to these instructions, as skipping lectures will negatively impact assignment performance.

Submission & Late Policy

- Written homework and project reports must be submitted on time.
- Assignments are due exactly at the **beginning of class** on the announced due date.
- **Late homework and reports will NOT be accepted** and will receive a score of zero (0).

Assessment Structure (The 50/50 Rule)

To ensure comprehensive understanding, homework grades are divided into two parts:

- **Submission (50%):** The written homework assignment itself.

- **HW Quiz (50%):** Each homework submission will be followed by an in-class pop quiz featuring questions that are identical or highly like the homework questions.

Academic Integrity & Verification

- All submitted assignments must be entirely the student's own original work. **AI-generated homework answers will not be accepted** and will be explicitly treated as cheating and plagiarism.
- **Verbal Verification:** The instructor reserves the right, at any time, to ask a student to explain the meaning or reasoning behind any part of their submitted answer.
    - *First Incident:* If a student cannot explain their answer, they will lose all points for that specific question.
    - *Second and Subsequent Incidents:* If it happens again, the student will lose all points for the entire assignment and will be formally reported to the Office of Student and Ethical Conduct.

Pop Quizzes and Class Participation Policy

To encourage consistent studying and review of the materials presented in lecture, unannounced pop questions and pop quizzes may be given at any time during class. These assessments will generally cover problems from the current or previous lectures.

- Pop Questions (Verbal)
    - You may be called upon randomly during the lecture to answer pop questions.
    - If your name is called and you provide an incorrect answer or fail to respond, a score of 0 points will be recorded.
- Pop Quizzes & Grading
    - **Frequency:** There will be approximately 10 –12 pop quizzes and several pop questions throughout the semester. Each quiz and question will be scored and weighted evenly.
    - **Standard Drop Policy:** At the end of the semester, your lowest scores (the bottom 20%) will be dropped. Your final pop quiz grade will be calculated using only your top 80%.
    - **Strict Absence Penalty: If you miss a class and miss a pop quiz, you will receive a score of 0. This specific 0 cannot be dropped and must be factored into your final score.**

Mid-Term and Final Exams

Exam Format & Expectations

Exams are designed to rigorously assess your mastery of the course material.

- **Conceptual Knowledge:** Tested via essay or multiple-choice questions.
- **Applied Problems:** You will be required to write or debug SQL code, design data models, or produce similar technical outputs.
- **Testing Environment:** All exams are strictly **closed-books and closed-notes**.

Absence & Make-Up Policy

Accommodations for missed exams are only granted for emergency with legitimate documents, and verifiable reasons.

- **Planned Absences:** You must notify the instructor well in advance and provide written documentation prior to the absence.
- **Illnesses & Emergencies:** You must notify the instructor **before the exam begins**. Upon your return, you must present formal documentation to request a make-up exam.
- **Point Reallocation:** If a make-up exam cannot be accommodated, the points for the missed exam will be reallocated to other exams at the instructor's discretion. Failure to follow the notification and documentation procedures will result in a zero.

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 Google's Team 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 lectures and obtain practical, hands-on experience. Students, in randomly selected, 2/3-member teams, will design and implement a small web-based database application. Team may choose any applications that are appropriate in size and complexity. Appropriateness will be determined by the instructor.

The project involves database design, modeling, creation, data population, query, and database application programming using JDBC, Java, JavaScript, JSP, XML, etc. 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 GitHub account, and GitHub will manage and keep track of your project work. Other development tools and 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 "Proposal & Requirements Report", 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 to 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 Workbench at http://www.mysql.com/products/workbench
  - MySQL Connectors - JDBC Driver for MySQL (Connector/J) at http://www.mysql.com/products/connector/
  - MySQL server at http://dev.mysql.com/downloads/mysql/
- 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

Grading Information

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

- Class Participation: **20%** (pop quizzes, pop questions, discussions, and interaction with instructor)
- HW Assignments: **20%** (10% on HWs and 10% on tests)
- Database application development project: **20%** (Teamwork with peer evaluations)
- Midterm exam: **20%**
- Final exam: **20%**

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 the first two lectures will be instructor-dropped out from the class.**
- 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 draw 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 note that obtaining HW solutions from someone or giving/showing your HW solutions to someone is also treated as cheating/ plagiarism.)**
- **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.

CS157A-01 Introduction to Database Management Systems, Fall 2026, Course Schedule

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

Week	Date	Topics, Readings, Assignments, Deadlines
1	8/20	Motivation, Orientation/Syllabus, Course Introduction, Prerequisites Check (Student's Information Due)
1	8/25	Syllabus and Class Activities Introduction to Database Systems Concepts (Chapter 1 & 2) (Transcript and Honesty Pledge Due)
2	8/27	Introduction to Database Systems Concepts (Cont.) Relational Model, SQL Data Definition, XML, DTD (Chapter 4 & 11) (Each student MUST install MySQL Server & Workbench & Tomcat Server on your laptop computer by next class time.)

Week	Date	Topics, Readings, Assignments, Deadlines
2	9/1	MySQL DBMS installation, basic administration, create database and tables, Java code connecting to MySQL using JDBC (Chapter 9) Three-tiered Architecture, Front-end/Back-end Framework Architecture (Chapter 9) Git & Github **Project team formation, 3 students per team**
3	9/3	Relational Model, SQL Data Definition, XML, DTD (Chapter 4, 7) Project Proposal Discussions
3	9/8	Relational Model, SQL Data Definition, XML, DTD (Chapter 4, 7) **Project Proposal & Requirements Document Due** (HW1 Out & Test)
4	9/10	**Git & GitHub account for team project due (Set to public and send collaborator invitation email to mike520124@gmail.com by 1pm, Tuesday 9/15)**
4	9/15	SQL Programming (Chapter 6) (HW1 Due & Solutions) **Last day to drop courses without an entry**
5	9/17	SQL Programming (Chapter 6) (Runnable 3-tiered architecture due for demo and code due in Github) (HW2 Out)
5	9/22	SQL Programming (Chapter 6)
6	9/24	SQL Programming (Chapter 6) (HW2 Due & Test, Solutions)
6	9/29	**Project Data Model & DB Design Document Due & Presentation**
7	10/1	**Midterm Exam**
7	10/6	Midterm Solutions Functional Dependencies (Chapter 3)
8	10/8	1st Project Code Review (revision code due in Github) Functional Dependencies (Chapter 3)
8	10/13	Normalization (Chapter 3)
9	10/15	BCNF, 3rd Normal Form (Chapter 3, 4) (HW3 Out)
9	10/20	Inference of Dependencies, Multivalued Dependencies BCNF, 3rd Normal Form (Chapter 3, 4)
10	10/22	Relational Algebra (Chapter 5) (HW3 Due & Test, Solutions)
10	10/27	Relational Algebra (Chapter 5)
11	10/29	Views and Indexes (Chapter 8) 2nd Project Code Review (revision code due in Github) (HW4 Out)
11	11/3	Views and Indexes (Chapter 8)
12	11/5	Views and Indexes (Chapter 8) (HW4 Due & Test, Solutions)
12	11/10	Constraints & Triggers (Chapter 7)
13	11/12	Constraints & Triggers (Chapter 7)
13	11/17	Stored Procedure, Querying XML (Chapter 9) 3rd Project Code Review (revision code due in Github)

Week	Date	Topics, Readings, Assignments, Deadlines
		(HW5 Out)
14	11/19	JSON, RDF, JQuery, SPARQL (Chapter 9)
14	11/24	SQL Injections (Chapter 10) (HW5 Due & Test, Solutions)
	11/26	**Thanksgiving Holiday - Campus Closed**
15	12/1	Final Project Presentation & Demo
15	12/3	Final Project Presentation & Demo Final Exam Preparation Final Project Code Review (final revision code due in Github)
Final Exam	12/10	Thursday 3:15pm – 5:15pm (Final Project Report & Source code files – zip file, Project Peer Evaluation, Final revision code due in Github at 3:15pm)