

**San José State University**  
**Department of Computer Science**  
**Introduction to Database Management Systems, CS157A, Spring 2026**

**Course and Contact Information**

<b>Instructor:</b>	Prof. Mike Wu
<b>Office Location:</b>	DH 215A
<b>Email:</b>	Ching-seh.Wu@sjsu.edu
<b>Office Hours:</b>	Tuesday & Thursday 5:00 – 6:00 pm <b>(Please drop me an email with time info and subject in advance.)</b>
<b>Class Days/Time:</b>	See MySJSU
<b>Classroom:</b>	See MySJSU
<b>Prerequisites:</b>	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, JSP, JavaScript, 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 (2nd edition) by Garcia-Molina, Ullman, and Widom

[Textbook web site: http://infolab.stanford.edu/~ullman/dscb.html](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
- [DBMS Tutorial](#)

## 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.

### Homework Assignments

Students are expected to learn all 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 and are due at the beginning of the class time on the announced due date; **late homework and reports will NOT be accepted and will be accounted 0 point.** It's student's responsibility to pay attention to my lectures for clearly explanation of the requirements for each homework assignment in class before they are posted on Canvas. Students who ignore lectures will not perform well on homework assignments. **All assignments submitted are expected to be the students' own original work. AI generated HW answers will not be accepted. The instructor may, at any time, ask a student to explain the meaning of any part of any answer that they**

**submitted. If the student can't explain the answer to a question, the penalty for the first incident will be loss of all points on the question. The penalty for the second and subsequent incidents will be loss of all points on the assignment and a report to the Office of Student and Ethical Conduct.**

### **Pop Quizzes & Pop Questions**

Unannounced pop questions and pop quizzes may be given anytime during class. **Each student MUST bring your laptop computer to the classroom for pop quizzes and for hands on exercises.** The purpose of pop quizzes and pop questions is to encourage you to learn, study and review the concepts and materials presented/discussed in the lecture. These will generally be problems covered in that day's or previous 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 several pop questions during the semester. Each pop quiz/pop question 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% (lowest scores) 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.**

### **Midterm 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 books and notes.

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 web-based database applications. A three-tiered architecture or a front-end/back-end framework architecture of the 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 project involves database design, modeling, creation, data population, query, and database application programming using JDBC, Java, JavaScript, JSP, HTML, CSS, 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 GIT & Github account and Github will manage and keep track of your team's 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 and required for each team member to assure that **each team member MUST contribute on 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/>
- Apache Tomcat Server version 9: <https://tomcat.apache.org/download-90.cgi#9.0.93>
- JSP Tutorial: <https://www.tutorialspoint.com/jsp/index.htm>
- HTML & CSS Tutorial: <https://www.w3schools.com/html/default.asp>
- HTML & CSS Basics: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)
- XML Tutorial: <https://www.w3schools.com/xml/default.ASP>
- 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

### Determination of Grades

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

- **Class Participation: 15%** (pop quizzes, pop questions, discussions, etc.)
- **HW Assignments: 20%** (5 Individual HWs, **NO AI generated answers**)
- **Database Team Project: 20%** (Team work with peer evaluations)
- **Midterm exam: 20%**
- **Final exam: 25%** (Accumulative/Comprehensive)

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

97 ~ 100	----	A+
93 ~ 96	----	A
90 ~ 92	----	A-
87 ~ 89	----	B+
83 ~ 86	----	B
80 ~ 82	----	B-
77 ~ 79	----	C+
73 ~ 76	----	C
70 ~ 72	----	C-
67 ~ 69	----	D+
63 ~ 66	----	D
60 ~ 62	----	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.

### Class 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 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 note that obtaining HW solutions from someone or giving/showing your HW solutions to someone is also treated as plagiarism/cheating.)**
- **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.**
- **I reserve the right to make announcements in class that may not appear on the class website/Canvas.**

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.**

### MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found at **Canvas** of SJSU One. **You are responsible for regularly checking with the email system and Canvas through One.SJSU at <http://one.sjsu.edu> to learn of any updates.**

### 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 Introduction to Database Management Systems, Course Schedule

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

Week	Date	Topics, Readings, Assignments, Deadlines
1	1/22	Motivation, Orientation/Syllabus, Course Introduction, Prerequisites Check <b>(Student's Information Form with Signature Due) (Instructor-drop if no show)</b>
2	1/27	Introduction to Database Systems Concepts (Chapter 1 & 2) <b>(Transcript and Honesty Pledge Due)</b>
2	1/29	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.)
3	2/3	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 <b>Project team formation, 3 students per team</b>
3	2/5	Relational Model, SQL Data Definition, XML, DTD (Chapter 4, 7) Project Proposal Discussions
4	2/10	Relational Model, SQL Data Definition, XML, DTD (Chapter 4, 7) <b>Project Proposal &amp; Requirements Document Due</b>

Week	Date	Topics, Readings, Assignments, Deadlines
		(HW1 Out)
4	2/12	Relational Model, SQL Data Definition, XML, DTD (Chapter 4, 7) <b>Git &amp; GitHub (account for team project due, Set to public and send collaborator invitation email to <a href="mailto:mike520124@gmail.com">mike520124@gmail.com</a> by 6pm, Tuesday 2/17)</b>
5	2/17	SQL Programming (Chapter 6) (HW1 Due & Solutions)
5	2/19	SQL Programming (Chapter 6) (Runnable 3-tiered architecture due for demo and code due in Github) (HW2 Out)
6	2/24	SQL Programming (Chapter 6)
6	2/26	SQL Programming (Chapter 6) (HW2 Due & Solutions)
7	3/3	<b>Project Data Model &amp; DB Design Document Due &amp; Presentation</b>
7	3/5	<b>Midterm Exam</b>
8	3/10	Midterm Solutions Functional Dependencies (Chapter 3)
8	3/12	First Project Code Review (revision code due in Github) Functional Dependencies (Chapter 3)
9	3/17	Normalization (Chapter 3)
9	3/19	BCNF, 3 <sup>rd</sup> Normal Form (Chapter 3, 4) (HW3 Out)
10	3/24	BCNF, 3 <sup>rd</sup> Normal Form (Chapter 3, 4) Inference of Dependencies, Multivalued Dependencies
10	3/26	Relational Algebra (Chapter 5) (HW3 Due & Solutions)
	3/31	<b>Spring Recess 3/30 ~ 4/3 (No Class)</b>
	4/2	<b>Spring Recess 3/30 ~ 4/3 (No Class)</b>
11	4/7	Relational Algebra (Chapter 5)
11	4/9	Views and Indexes (Chapter 8) Second Project Code Review (revision code due in Github) (HW4 Out)
12	4/14	Views and Indexes (Chapter 8)
12	4/16	Views and Indexes (Chapter 8) (HW4 Due & Solutions)
13	4/21	Constraints & Triggers (Chapter 7) Third Project Code Review (revision code due in Github) (HW5 Out)
13	4/23	Stored Procedure, Querying XML (Chapter 9)
14	4/28	JSON, RDF, JQuery, SPARQL (Chapter 9)
14	4/30	SQL Injections (Chapter 10) (HW5 Due & Solutions)
15	5/5	Final Project Presentation & Demo
15	5/7	Final Project Presentation & Demo Final Exam Preparation Final Project Code Review (final revision code due in Github)

<b>Week</b>	<b>Date</b>	<b>Topics, Readings, Assignments, Deadlines</b>
Final Exam	TBA	Time; TBA (Final Project Report, Project Peer Evaluation, Final revision code due in Github)