CS 436/636: Database Application Development

University of Massachusetts Boston - Spring 2026

Class Syllabus

Course Description

Data is everywhere. As scientists, users, and citizens, we are both generating and exploiting large, ever-growing, diverse sets of data. For several applications – ranging from scientific discovery to business analysis, governance, and everyday activities – we are directly using and indirectly affecting hundreds of data systems! The big challenge is to turn data into useful knowledge, and to do so quickly, in order to increase the impact of the new insights. From an application perspective, achieving these goals comes with a number of technical challenges. How to exploit the continuously evolving hardware (storage, computation, network)? How to collect all incoming data efficiently? How to query dynamic collections of data that keep accumulating incoming data? How to parallelize query processing from one core to a few (scale-up), and then to thousands (scale out)? What are the needs of evolving workloads (hybrid transactional/analytical processing, graph analytics, Internet-of-Things, micro-payments, monitoring)? In this course, we will discuss how to design data systems/applications that can address these challenges. We will start with the core concepts of database internals, covering entity-relationship and relational data models, commercial relational query languages (SQL and relational algebra), file organization, storage and memory management, indexing and hashing, and query optimization. We will then study in detail aspects related to developing database applications, that is, software systems that solve a particular real-world problem and hold their data in a relational database. The application has a realistic user interface. Students will work in small groups on a real-world project specified and implemented during the term. Topics include system specification from user needs, analysis of data flow and workflow, object design, database design, client-server techniques, and rapid prototyping systems. Finally, we will cover new research and trends in data management including Big Data and NoSQL databases.

Course Objective

The objective of this course is to provide students with the core concepts and practical skills required to design and develop effective database-driven applications. Students will learn how to model, store, and query data using relational databases, understand key database internals, and address performance challenges. Through hands-on, team-based projects, the course emphasizes translating real-world requirements into complete applications, while also introducing emerging trends such as Big Data, NoSQL systems, and cloud-based data management.

Prerequisites

CS 310, CS 430/630. Knowledge of C++ and Java, data structures, and algorithms is required.

Instructor

Tarikul Islam Papon
Website: https://www.cs.umb.edu/~tpapon/
Email: t.papon@umb.edu
Office hours: Tue and Thu, 9 – 10 am or by appointment
Office: McCormack Hall, 3rd Floor, M-3-201-28

Lecture Times and Places

Tue/Thu, 2:00 - 3:15 pm, room Wheatley-Peters W01-0006

Textbooks and Readings

- R. Ramakrishnan, J. Gehrke, "Database Management Systems", 3rd edition.
- Supporting Material (Including solved exercises)
- Architecture of a Database System, by J. Hellerstein, M. Stonebraker and J. Hamilton
- The Design and Implementation of Modern Column-store Database Systems, by D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden

Web Resources

- SQL Tutorial
- SQLPlus quick guide: http://www.orafaq.com/wiki/SQL*Plus_FAQ
- Supabase Tutorial & Supabase for Beginners
- Supabase Docs (Official)
- Next.js Docs
- Supabase + Next.js Tutorial
- Intro to PL/SQL (from Stanford Infolab)
- A PL/SQL Tutorial: http://plsql-tutorial.com/

Course Website | Canvas | Piazza

All assignments, announcements, and lecture notes can be found on Canvas. Piazza will be used for discussion.

Course Requirements & Grading

The course grade will break down as follows (minor alterations may occur):

Class Participation	5%
3 Written Assignments	25%
Midterm 1	20%
Paper Presentation	10%
Midterm 2	15%
Project	25%

Written Assignments: Approximately every three weeks, there will be one written assignment based on concepts taught in class.

Paper Presentation: Each student (or small group) will present a selected research paper related to database systems. The presentation will focus on clearly explaining the problem, key ideas, and contributions of the paper, and ensure that the class is engages is meaningful debate. Please note that in CS 436, the paper presentation is optional.

Class Project: Students will work in small groups on a course project that involves designing, implementing, or evaluating an application on top of a database system. The project will emphasize applying course concepts and may include a written report and a final presentation.

Important Dates for CS 436/636

February 19: submit assignment 1
March 5: submit assignment 2
March 12: midterm 1
April 2: submit assignment 3
April 7-30: paper presentation
May 5: midterm 2
May 12: project submission

Attendance Policy

Class attendance is mandatory, and students are expected to participate actively in class by asking and answering questions. In case of missed class, students are responsible to get up-to-date with course materials and announcements.

Collaboration Policy

You are strongly encouraged to collaborate with one another in studying the lecture materials. You may discuss ideas and approaches to the assignments and projects with others (provided that you acknowledge doing so in your solution), but such discussions should be kept at a high level, and should not involve actual details of the code or of other types of answers. **You must complete the actual solutions on your own.**

Academic Integrity Statement

Education at UMass Boston is sustained by academic integrity. Academic integrity requires that all members of the campus community are honest, trustworthy, responsible, respectful, and fair in academic work at the university. As part of being educated here, students learn, exercise, increase, and uphold academic integrity. Academic integrity is essential within all classrooms, in the many spaces where academic work is carried out by all members of the UMass Boston community, and in our local and global communities where the value of this education fulfills itsrole as a public good. Students are expected to adhere to the Student Code of Conduct, including policies about academic integrity, delineated here: https://www.umb.edu/life_on_campus/dean_of_students/student_conduct

Accommodation Statement

UMass Boston is committed to creating learning environments that are inclusive and accessible. If you have a personal circumstance that will impact your learning and performance in this class, please let me know as soon as possible, so we can discuss the best ways to meet your needs and the requirements of the course. If you have a documented disability, or would like guidance about navigating support services, contact the Ross Center for Disability Services by email (ross.center@umb.edu), phone (617-287-7430), or in person (Campus Center, UL Room 211). To receive accommodations, students must be registered with the Ross Center and must request accommodations each semester that they are in attendance at UMass Boston. For more information visit: www.rosscenter.umb.edu. Please note that the Ross Center will provide a letter for your instructor with information about your accommodations only and not about your specific disability.

Tentative Schedule

Date	Topic	Deliverables Due
January 27	Class Cancelled Due to Bad Weather	
January 29	Course Logistics & Overview	
February 3	Overview of a Database Management System	
February 5	Entity-Relationship Model	
February 10	ER Contd.	
February 12	Relational Model & Relational Algebra	
February 17	Database Query Language - SQL	
February 19	SQL for Application Development	Assignment 1
February 24	Storage Engine Internals	
February 26	Log-Structured Merge (LSM) Trees	
March 3	Indexing in Database Application	

March 5	Quert Processing	Assignment 2
March 10	Query Optimization in App Development	
March 12	**Midterm 1**	
March 17	**Spring Break**	
March 19		
March 24	App Development: Supabase	
March 26	Supabase Contd.	
March 31	App Development: Node.js + Next.js + React	
April 2	Node.js + Next.js + React Contd.	Assignment 3
April 7	User Interface Design	
April 9	UI Design Contd.	
April 14	Client-Server Techniques	
April 16	Paper Presentation: DB Application	
April 21	Paper Presentation: Serverless Techniques	
April 23	Paper Presentation: UI Design	
April 28	Paper Presentation: Modern Hardware	
April 30	Paper Presentation: Storage Layer	
May 5	**Midterm 2**	
May 7	Project Demonstration	
May 12	Final Review	Final Project Submission