

CAS CS 660: Graduate Introduction to Database Systems

Boston University

Fall 2023

Class Syllabus

Course Description: Database systems provide the necessary infrastructure to manage huge data collections. This class serves as a comprehensive introduction in the key concepts of the architecture of modern database systems. We will discuss both traditional approaches used modern trends that shape the data management industry today. The primary focus of the course will be on the core concepts of the *internals* of database systems, covering SQL and Relational Algebra, file organization, storage and memory management, indexing and hashing, query optimization, query processing, transaction processing, concurrency control and recovery. Finally, we will cover new trends in data management including Big Data and NoSQL databases and data management on the Cloud, and we will discuss the history of database systems and their evolution over the years.

An important part of the course is a semester-long project where students will have to implement parts of a small educational relational database system. Special attention on this course will be placed on the “systems” aspect of data management systems.

Prerequisites: CS background and working knowledge of a high-level programming language (C/C++) is required for the project. Basic knowledge of Algorithms and Data Structures at the CS undergraduate level. Students must also have programming skills and experience in Unix environment.

Instructor: Manos Athanassoulis (mathan@bu.edu)

Office Hours: Tue @ 10-11am / Thu @ 2-3 pm (after class) at CDS 928 or by appointment

Teaching Fellows: Konstantinos Karatsenidis , Aneesh Raman , Zichen Zhu

Office Hours: (check Piazza)

Meeting Times and Places

lectures: Tue/Thu, 12:30-1:45 pm, CAS 313

labs: Fri, 12:20-1:10, HAR 211 & 1:25-2:15, PSY B35 & 4:40-5:30, BRB 122 (*tentative*)

Course Website: <https://bu-disc.github.io/CS660/>

All class assignments, schedules, and lecture notes can be found on this page. We will also use Piazza for discussions and other material distribution.

Required Textbook: R. Ramakrishnan and J. Gehrke. [Database Management Systems](#). Third Edition. McGraw-Hill 2002. Throughout the class we will cover a few topics from recent research and survey papers.

Additional Reading Material: The following are excellent sources for additional reading.

- [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
- [Modern B-Tree Techniques](#), by Goetz Graefe, Foundations & Trends in Databases, 2011

Course Requirements

A. Collaborative Notes: Starting from class 3, the students will contribute towards a shared collaborative document. Each student will contribute notes to at least one class, and each class will have at least two contributors.

B. Programming Assignments: During the semester there will be two programming assignments. The first one will consist of building an application for managing data of a real-life application. The second project will help the students understand how query optimization works via hands-on experimentation with real-life database systems. The projects will be based on topics taught in class, and there will be additional hands-on support during the labs.

C. Written Assignments: Approximately every two weeks there will be one written assignment based on concepts taught in class. The assignments will be graded on completion basis, that is, you will receive full credit as long as you submit them on time and receive a grade above 70%.

D. Hands-on test: There will be a hands-on test for extra credit. The students will have a short amount of time during class to write a few queries (SQL) based on material taught in class and discussed in labs.

Grading Policy: The course grade will break down as follows (minor changes may occur):

- Notes & Class participation: 5%
- Written Assignments: 10%
- Programming Assignments: 40%
- Midterm: 20%
- Final: 25%
- SQL Hands-on Test (bonus): 5%

AI Tools Use Policy: Generative AI Tools like ChatGPT that use foundation models are widely available and offer a useful tool for “start-up tasks” like brainstorming or writing boilerplate code. My philosophy is to limit the use of such tools for any other reason other than start-up tasks and getting feedback (e.g., spelling and grammar checks). Note that the reason is that the

goal is to train your mind and not use the training of the AI model. This is important because: (i) all large models have a tendency to make up incorrect facts and fake citations, (ii) code generation models have a tendency to produce inaccurate outputs, and (iii) in order to eventually develop the ability to identify when the tool should be trusted, you need to be able to judge its output. In case you use any input from a foundational model, its contribution must be acknowledged to avoid plagiarism penalties. Having said all these disclaimers, I urge you to explore the capabilities and the shortcomings of foundation models and in this direction, I plan to experimentally add an assignment using foundational models in this class.

Late Policy: Students needing additional time may submit programming assignments late. There will be no penalty for up to four (4) total late days in the semester. Any additional late days would result to a zero in the corresponding assignment. There are no late days allowed for written assignments.

Important Dates for all classes

September 18th, last day to add a class

October 10th, last day to drop (without a “W”)

November 13th, last day to drop (with a “W”) & to designate the course as Pass/Fail

Tentative Schedule

Week #	Topics	Readings
1	Introduction & Relational Model	Chapters 1, 3
2	SQL, Storage and Files	Chapters 5, 9
3	Storage and Indexing	Chapter 8
4	Tree Structured Indexing	Chapter 10
5	Hash-Based Indexing	Chapter 11
6	External Sorting	Chapter 13
7	Query Processing	Chapters 12, 14
8	Query Processing (continued)	Chapters 12, 14
9	Query Optimization	Chapter 15
10	Transactions	Chapter 16
11	Concurrency Control	Chapter 17
12	Recovery	Chapter 18
13	NoSQL and Big Data	Chapter 25
14	Cloud Data Management & Distributed Databases	Chapter 22

Collaboration Policy

You are strongly encouraged to collaborate with one another in studying the lecture materials and preparing for reviews and presentations.

You may discuss ideas and approaches to the 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 Misconduct

We will assume that you understand BU's Academic Conduct Code:

<http://www.bu.edu/academics/policies/academic-conduct-code>

Prohibited behaviors include:

- copying all or part of someone else's work, even if you subsequently modify it; this includes cases in which someone tells you what you should write for your solution
- viewing all or part of someone else's work
- showing all or part of your work to another student
- consulting solutions from past semesters, or those found online or in books
- posting your work where others can view it (e.g., online).

Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC). The ACC may suspend/expel students found guilty of misconduct. *At a minimum, students who engage in misconduct will have their final grade reduced by one letter grade (e.g., from a B to a C).*