Database Systems (CS-UH 2214)

Fall 2025

Instructor:	Azza Abouzied azza@nyu.edu	Lectures: Office Hours:	Tu Th 11:20 – 12:35 Social Sciences 018 calendly.com/prof-azza/db-students
TA:	Anh Mai anh.mai@nyu.edu	Office Hours:	Wed, A1-1106 D, 2:00 PM - 3:30 PM

Course Description: This course introduces students to the foundations of database systems, focusing on basics such as data models, especially the relational data model, query languages, query optimization and processing, indices and other specialized data structures, as well as transactions and concurrency control. Students build components of a database system and through research readings understand the design complexities of transactional and big data analytical systems.

Course Links:

Course Book	azzadev.github.io/dbcrsbook

The course book is the online central resource for all video lectures, reading materials, links to assigned assessments, etc.

Course textbook:

- Raghu Ramakrishnan and Johannes Gehrke. **Database Management Systems**. 3rd. McGraw-Hill, Aug. 2002.

Reference textbooks:

- Peter Bailis, Joseph Hellerstein, and Michael Stonebraker. **Readings in Database Systems**. 5th. 2015. URL: `http://www.redbook.io/`.

Getting help: The course slack channel is database-systems-org.slack.com. You can send questions to the private fall 2025 channel (all other channels are restricted where course instructors may post general information) or DM the course instructors on slack.

You can also set up an appointment for office hours with any of the instructors. On certain weeks, Thursday's class is an open office hour, lab recitation, or patch-the-gap session. You can book an office hour appointment to meet Prof Azza using calendly.com.

Learning Outcomes:

- Design and implement a database for any application that provides good transactional performance and maintains data consistency guarantees
- Query and analyze data within a database
- Explain data storage, access, index and query optimization techniques employed by relational database systems
- Estimate and evaluate the performance of different query workloads and fine-tune database performance
- Understand the components of a DBMS by implementation through a series of programming-intensive labs
- Explain how a database system can allow multiple users to concurrently query and update the same data and preserve data consistency even in the face of failures
- Explain how distributed databases provide ACID transactions
- Explain the tradeoffs made by different database architectures and newSQL or noSQL systems

- Synthesize a set of design principles that are useful for building large data processing systems
- Study how one can break down fundamental assumptions about hardware or workload expectations to re-design novel database systems.

Teaching Methodologies:

- *Online Lectures*: This course is *flipped*! All lectures will be provided asynchronously, online through the course book. The book is split into 14 modules. Each week covers a specific module.

  In-person class meetings will be a combination of *in-class interactive exercises* where you are expected to participate and *patch the gap* sessions where we cover concepts that are somewhat tricky.

- *Readings*: Each weekly module will begin with the assigned reading material. You should read these in addition to listening to the online video lectures, and reading the online course book notes. By keeping on top of the readings, you will be better prepared to take the online weekly assessment and you can better participate in class and you can clarify concepts you found difficult to understand.
- *Collaborative work*: You will work with your peers to complete your programming-intensive labs. Effective team work is crucial for developing large software systems.
- *Labs*: You will complete a set of programming-intensive group labs to build the different components of SimpleDB[1]. By implementing the building blocks of a database system, you apply the concepts learned in class. A condition of using MIT's SimpleDB code base is not to distribute/share your solutions. All assignment resources are therefore distributed through NYU Drive. Solutions must be submitted securely and you should not publish your solutions online.
- *Weekly Online Assessments*: You have to complete 11-14 online assessments, roughly one for each week. Assessments are due on **Saturday** and discussed in class on the following Tuesday. You have an entire week to complete each assessment. Links for each assessment will be found in the relevant module's course book page.

  This course emphasizes ***independent and active learning***. You need to read the assigned material, and follow the online lectures to successfully complete the assessment. Some Thursday classes are dedicated to help students understand any material they found confusing.

- *Midterm & Final Exam*: The midterm and final are not just for grades: they are a chance to sit down, really learn the material, and see where you are at. The good news is that if you keep up with the weekly assessments and work through them thoughtfully, without taking shortcuts, the midterm and final will feel much more manageable. They also help highlight what to keep working on, since learning does not stop when the course does.

Course Deliverables:

2-3 Labs	20%
2-3 Problem Sets	20%
10-14 Weekly Assessments	10%
Midterm	20%
Final Exam	25%
Bonus and Class Participation	5%

The exact grade breakdown may change. If this happens, the course instructor will notify students in a reasonable time.

Grading Policy: In general, a 90% or above is within the A range, 80%-90% is within the B range and 70%-80% is within the C range.
You have **100 hours of lateness forgiveness** that you can use throughout the course for any problem set or lab submission deadline.

[1] **MIT SimpleDB Labs for 6.830**. URL: `https://github.com/MIT-DB-Class/course-info/blob/master/lab1.md`.

Weekly Assessments Grading The weekly assessments are intended to support your independent learning by offering a structured set of questions. They are low-stakes and graded primarily on a pass/fail basis, so individual scores matter less than consistent engagement. Detailed feedback will not be provided beyond what is covered in the *patch-the-gap* sessions. Please note that relying on AI to complete these assessments is counterproductive: it undermines your learning and will likely leave you unprepared for the midterm and final exams, which build directly on the material covered in these weekly tasks.

Grade Inquires: Grades are determined based on the criteria stated in this syllabus and relevant rubrics. If you believe there has been a genuine error in grading (e.g., a miscalculation or overlooked content), you are welcome to bring it to our attention within 7 days of receiving the grade. However, requests for additional points based on effort, personal need, or comparisons to peers will not be considered. Grade discussions should focus on substance and evidence, not negotiation. **Grade grubbing wastes everyone's time!** Our goal is to focus on learning and growth, not point accumulation.

Course Schedule:

The following is a tentative course schedule. Exact release and due dates of labs and problem sets may change. Weekly assessments are due every Saturday at 9:00 PM Abu Dhabi time. Weekly modules may be extended, shrunk, shifted or re-ordered.

Week	Lectures, Readings, Assignments
0	**Overview & Introduction to Database Systems** *Reading:* Bailis, J. Hellerstein, and Michael Stonebraker, *Readings in Database Systems*, What Goes Around, Comes Around
1	**The Relational Model** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 4-4.25, Chp 2-2.6, Sec 3.5 *Optional Reading:* Codd, "A Relational Model of Data for Large Shared Data Banks"
2	**Querying Languages & SQL** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 5, 6 *Optional Reading:* Cohen et al., "MAD Skills: New Analysis Practices for Big Data" *Assignment:* Problem Set 1 - Analyzing the DBLP dataset (2 weeks)
3	**Schema Design & Normalization** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 19-19.4
4	**Introduction to DB internals** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 9.3-9.7 *Optional Reading:* J. M. Hellerstein, Michael Stonebraker, and Hamilton, "Architecture of a Database System"
5	**Buffer Pools & Memory Management** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 9.4 *Optional Reading:* Chou and DeWitt, "An Evaluation of Buffer Management Strategies for Relational Database Systems" *Assignment:* Lab 1: The Catalog, Heap File and Buffer Pool (2 weeks)
6	**Access Methods** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 10
7	**Indexing & Hashing** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 13, 28.1–28.3.1, 28.6, 11 *Optional Reading:* Beckmann et al., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles" **Midterm** covers weeks 1-7
8	**Query Processing** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 12.4 *Assignment:* Lab 2: Query Operators (2 weeks)
9	**Sorting, Grouping & Aggregating** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 13, 14.5, 14.6
10	**Joins** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 14.4 *Optional Reading:* Shapiro, "Join Processing in Database Systems with Large Main Memories"

11 **Query Optimization**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 12.4-7, 15.1, 15.3–7
Optional Reading: Selinger et al., "Access Path Selection in a Relational Database Management System"
Optional Reading: Mannino, Chu, and Sager, "Statistical Profile Estimation in Database Systems"
Assignment: Problem Set 2: Query Optimization with Postgres (2 weeks)

12 **ACID Transactions & Concurrency Control**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 16.1-3 and 17.1, Chp 16.4-16.6, 17.1–17.5
Optional Reading: Franklin, "Concurrency Control and Recovery"
Optional Reading: Kung and Robinson, "On Optimistic Methods for Concurrency Control"

13 **Recovery**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 18
Optional Reading: Mohan, Haderle, et al., "ARIES: A Transaction Recovery Method Supporting Fine-granularity Locking and Partial Rollbacks Using Write-ahead Logging"

14 **Distributed Databases & Transactions**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 22.6-22.8, 22.11-22.14)
Optional Reading: Mohan, Lindsay, and Obermarck, "Transaction Management in the R* Distributed Database Management System"
Optional Reading: Bailis, J. Hellerstein, and Michael Stonebraker, *Readings in Database Systems*, Chp 6: Weak Isolation and Distribution
Optional Reading: Brewer, "CAP twelve years later: How the "rules" have changed"
Final Exam covers all weeks with emphasis on weeks 8-14.

References for Additional or Optional Course Readings:

- Andrew Pavlo and Matthew Aslett. **What's Really New with NewSQL?** In: *SIGMOD Rec.* 45.2 (Sept. 2016), pp. 45–55. URL: https://doi.org/10.1145/3003665.3003674.
- Azza Abouzeid et al. **HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads**. In: *Proc. VLDB Endow.* 2.1 (Aug. 2009), pp. 922–933. URL: http://dx.doi.org/10.14778/1687627.1687731.
- Jeffrey Cohen et al. **MAD Skills: New Analysis Practices for Big Data**. In: *Proc. VLDB Endow.* 2.2 (Aug. 2009), pp. 1481–1492. URL: http://dx.doi.org/10.14778/1687553.1687576.
- Jeffrey Dean and Sanjay Ghemawat. **MapReduce: Simplified Data Processing on Large Clusters**. In: *Commun. ACM* 51.1 (Jan. 2008), pp. 107–113. URL: http://doi.acm.org/10.1145/1327452.1327492.
- Joseph M. Hellerstein, Michael Stonebraker, and James Hamilton. **Architecture of a Database System**. In: *Found. Trends databases* 1.2 (Feb. 2007), pp. 141–259. URL: http://dx.doi.org/10.1561/1900000002.
- Michael J. Franklin. **Concurrency Control and Recovery**. In: *The Computer Science and Engineering Handbook.* Ed. by Allen B. Tucker. CRC Press, 1997, pp. 1058–1077. URL: http://bit.ly/2g6oC6R.
- C. Mohan, Don Haderle, et al. **ARIES: A Transaction Recovery Method Supporting Fine-granularity Locking and Partial Rollbacks Using Write-ahead Logging**. In: *ACM Trans. Database Syst.* 17.1 (Mar. 1992), pp. 94–162. URL: http://doi.acm.org/10.1145/128765.128770.
- Norbert Beckmann et al. **The R*-tree: An Efficient and Robust Access Method for Points and Rectangles**. In: *SIGMOD Rec.* 19.2 (May 1990), pp. 322–331. URL: http://doi.acm.org/10.1145/93605.98741.
- Michael V. Mannino, Paicheng Chu, and Thomas Sager. **Statistical Profile Estimation in Database Systems**. In: *ACM Comput. Surv.* 20.3 (Sept. 1988), pp. 191–221. URL: http://doi.acm.org/10.1145/62061.62063.
- C. Mohan, B. Lindsay, and R. Obermarck. **Transaction Management in the R* Distributed Database Management System**. In: *ACM Trans. Database Syst.* 11.4 (Dec. 1986), pp. 378–396. URL: http://doi.acm.org/10.1145/7239.7266.
- Leonard D. Shapiro. **Join Processing in Database Systems with Large Main Memories**. In: *ACM Trans. Database Syst.* 11.3 (Aug. 1986), pp. 239–264. URL: http://doi.acm.org/10.1145/6314.6315.
- Hong-Tai Chou and David J. DeWitt. **An Evaluation of Buffer Management Strategies for Relational Database Systems**. In: *Proceedings of the 11th International Conference on Very Large Data Bases - Volume 11.* VLDB '85. Stockholm, Sweden: VLDB Endowment, 1985, pp. 127–141. URL: http://dl.acm.org/citation.cfm?id=1286760.1286772.

- H. T. Kung and John T. Robinson. **On Optimistic Methods for Concurrency Control**. In: *ACM Trans. Database Syst.* 6.2 (June 1981), pp. 213–226. URL: http://doi.acm.org/10.1145/319566.319567.
- P. Griffiths Selinger et al. **Access Path Selection in a Relational Database Management System**. In: *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data*. SIGMOD '79. Boston, Massachusetts: ACM, 1979, pp. 23–34. URL: http://doi.acm.org/10.1145/582095.582099.
- E. F. Codd. **A Relational Model of Data for Large Shared Data Banks**. In: *Commun. ACM* 13.6 (June 1970), pp. 377–387. URL: http://doi.acm.org/10.1145/362384.362685.

Academic Integrity: As set forth in NYU Abu Dhabi's Academic Integrity Policy, the relationship between students and faculty at NYU Abu Dhabi is defined by a shared commitment to academic excellence and is grounded in an expectation of fairness, honesty, and respect, which are essential to maintaining the integrity of the community. Every student who enrolls and everyone who accepts an appointment as a member of the faculty or staff at NYU Abu Dhabi agrees to abide by the expectation of academic honesty. The full policies and procedures relating to Academic Integrity may be found on the NYUAD Student Portal.

Wellness: As a University student, you may experience a range of issues that can interfere with your ability to perform academically or impact your daily functioning, such as: heightened stress; anxiety; difficulty concentrating; sleep disturbance; strained relationships; grief and loss; personal struggles. If you have any well-being or mental health concerns please visit the Counseling Center on the ground floor of the campus center from 9am-5pm Sunday - Thursday, or schedule an appointment to meet with a counselor by calling: 02-628-8100, or emailing: nyuad.healthcenter@nyu.edu. If you require mental health support outside of these hours call NYU's Wellness Exchange hotline at 02-628-5555, which is available 24 hours a day, 7 days a week. You can also utilize the Wellness Exchange mobile chat feature, details of which you can find on the student portal. If you need help connecting to these supports please contact me directly.