Database Systems (CS-UH 2214)

Spring 2020

Instructor:	Azza Abouzied azza@nyu.edu	Lectures:	MW 10:25 – 11:40 @ A2 005.
TA:	Miro Mannino miro.mannino@nyu.edu	Hrs:	T 2:00 PM – 5:00 PM @ A1 1102G

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 Page: The Piazza class forum will serve as the central online resource for this course. You can sign up for the class using access code nyuad-db or by accepting the class invitation sent via email.

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: Email any questions to the class forum on Piazza for the most immediate help from instructors or peers. I'm available for an hour after every class or at other times by appointment.

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
- Analyze and critique research articles
- Study how one can break down fundamental assumptions about hardware or workload expectations to redesign novel database systems.

Teaching Methodologies:

- *Lectures*: In class lectures will cover fundamental database system concepts. Most lectures have *in-class interactive exercises* and you are expected to participate.
- *Readings*: The course schedule lists sections of the course textbook as well as research papers that students should read prior to class. By keeping on top of the readings, you will make the best use of lecture time: you can clarify concepts you found difficult to understand and you can better participate in class discussions and exercises.
- *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.
- *Class Project*: The group class project is meant to help you apply concepts learned in class to other problems. Your project can either involve a research component such as a thorough comparison & evaluation of different database systems or a novel idea applied to a specific database systems problem, or involve an engineering component such as implementing an existing research algorithm in an open source database like Postgres.

Course Deliverables:

Labs & Problem Sets	35%
Course Project	35%
Midterm 1	10%
Midterm 2	15%
Bonus and Class Participation	≈5%

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 project submission deadline.

Project Milestones:

Proposal A 1-page proposal (single spaced, 11pt) with the following
- team members
- short description of the project & motivation
- three to five references from data conferences such as VLDB, PODS, SIGMOD, ICDE, EDBT, KDD, InfoVis, etc.
- tools, systems, etc you plan to use

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

Milestone Report

Extend your proposal into a 3-4 page report, formatted as an ACM Conference paper with the following

- extended problem description (1-2 page)
- proposed solution sketch (1-2 page)
- a summary of related work
- list of completed project steps so far
- problems faced and unexpected findings if any
- future action plan such as remaining modules to code, user studies to conduct, experiments to run, etc.

It is also time to start learning how to write a good research paper: See "How to write a good research paper by Simon Peyton-Jones"

Presentations

Each project gets 15 minutes: 10 minutes presentation + 5 minutes of Q/A. Not sure how to give a good research talk? See Microsoft's excellent series.

Final Report

Extend your previous report into a 6-7 page report:

- improve and expand on all sections
- include the evaluation section
- add a conclusion

Course Schedule:

Week	Lectures, Readings, Case Studies, Assignments
1	**Introduction to Database Systems** *Reading:* Bailis, J. Hellerstein, and Michael Stonebraker, *Readings in Database Systems*, What Goes Around, Comes Around
2	**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" **Schema Design and Normalization** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 19-19.4
3	**Querying Basics: SQL** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 5 *Optional Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 6 **More SQL** *Reading:* Cohen et al., "MAD Skills: New Analysis Practices for Big Data" *Assignment:* Preparing and Querying the DBLP datasets (1 week)
4	**Introduction to DB internals** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 9.3-9.7 *Reading:* J. M. Hellerstein, Michael Stonebraker, and Hamilton, "Architecture of a Database System" **From a Declarative Query to a Plan of Operators** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 12
5	**Query Optimization (Cost-based techniques)** *Reading:* Ramakrishnan and Gehrke, *Database Management Systems*, Chp 14, 15 *Reading:* Selinger et al., "Access Path Selection in a Relational Database Management System" **Join Algorithms** *Reading:* Shapiro, "Join Processing in Database Systems with Large Main Memories" *Optional Reading:* Mannino, Chu, and Sager, "Statistical Profile Estimation in Database Systems"

6 **Buffer Pools & Memory Management**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 20.1-20.7
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Sec 13.4
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)

7 **Indexing & Access Methods**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 10
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 13
R-trees (Spatial Data) and Hashing techniques
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 11
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Sec 28.1-28.3.1, 28.6
Reading: Beckmann et al., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles"
Project: Discussion and Proposals due
Assignment: Query Optimization with Postgres (1 week)

8 **One size fits all? OLTP vs OLAP & Column Stores**
Reading: Bailis, J. Hellerstein, and Michael Stonebraker, *Readings in Database Systems*, Chp 4: New DBMS Architectures
Reading: Mike Stonebraker et al., "C-store: A Column-oriented DBMS"
Midterm 1
Assignment: Lab 2: Query Operators (2 weeks)

9 **ACID Transactions**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 16
Case Study: MongoDB & NoSQL systems: What do transactions give you?

10 **Concurrency Control**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 17
Reading: Franklin, "Concurrency Control and Recovery"
Reading: Kung and Robinson, "On Optimistic Methods for Concurrency Control"
Recovery
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 18
Reading: Mohan, Haderle, et al., "ARIES: A Transaction Recovery Method Supporting Fine-granularity Locking and Partial Rollbacks Using Write-ahead Logging"

11 **Distributed Databases & Transactions**
Reading: Ramakrishnan and Gehrke, *Database Management Systems*, Chp 22
Reading: Mohan, Lindsay, and Obermarck, "Transaction Management in the R* Distributed Database Management System"
Trade-offs & CAP
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"
Project: Milestone Update and Report

12 **Research Discussion: Deconstruct 2PC: Examining Calvin -**
Reading: Thomson et al., "Calvin: Fast Distributed Transactions for Partitioned Database Systems"
Research Discussion: What's new about NewSQL?
Reading: Pavlo and Aslett, "What's Really New with NewSQL?"
Optional Reading: Abouzeid et al., "HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads"

13 **Research Discussion: Web Search Engines & Databases: A good fit?**
Ramakrishnan and Gehrke, *Database Management Systems*, Sec 27.1-27.5
Bailis, J. Hellerstein, and Michael Stonebraker, *Readings in Database Systems*, Combining Systems and Databases: A Search Engine Retrospective by Eric Brewer
Assignment: MapReduce and Building Inverted Indexes (1 week)

14 **Project Presentations**
Midterm 2
Project: Final Reports Due

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.