

Database System Internals

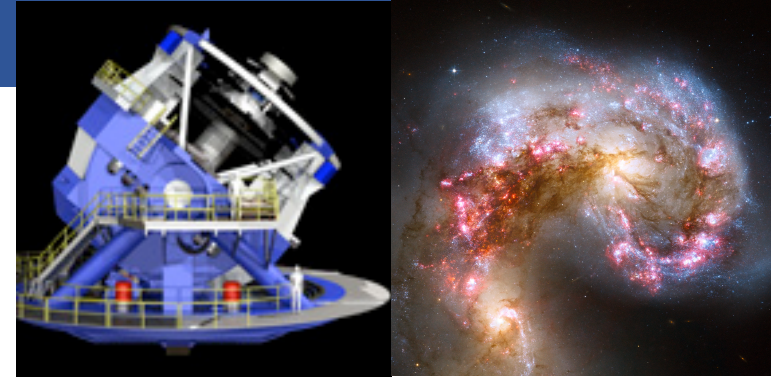
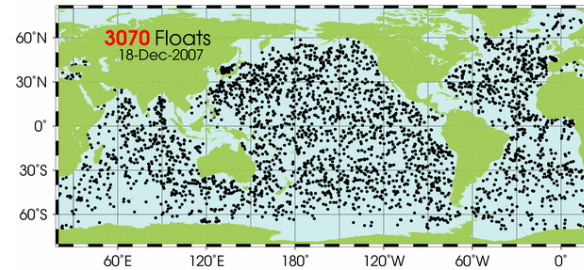
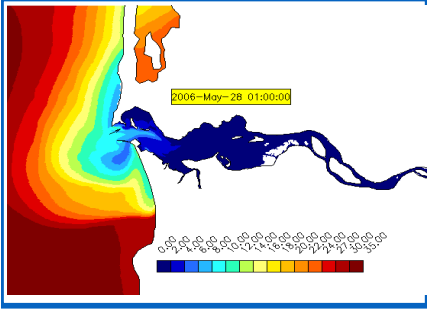
Introduction

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Course Staff

- Instructors:
 - Ryan Maas
- TAs:
 - Saksham Aggarwal
 - Apurv Goel
 - Moe Kayali
 - Kexuan Liu
 - Pranay Mundra
 - Yiwen Qiu
 - Ying Wang
- Email addresses and office hour times and locations will be on the course website and on message board
 - Every day one or more of us will have office hours

Course Goals



- The world is drowning in data!
- Need computer scientists to help manage this data
 - Help domain scientists achieve new discoveries
 - Help companies provide better services
 - Help governments become more efficient
- This class: **principles of building data mgmt systems**
 - Learn how classical DBMSs are built
 - Learn key principles and techniques
 - Get hands-on experience building a working DBMS

Course Format

- Lectures MWF @ 12:30pm
- Sections: Thursday morning
- Homeworks
 - 5 Labs + 6 Written homeworks
- Quizzes:
 - 2 short quizzes on Gradescope

Communication (part 1)

- Web page: <http://www.cs.washington.edu/444>
 - Lectures/Sections slides will be posted there
 - Homeworks/Labs will be available there
- Mailing list
 - Announcements, group discussions
 - Your @uw.edu address is already subscribed

Communication (part 2)

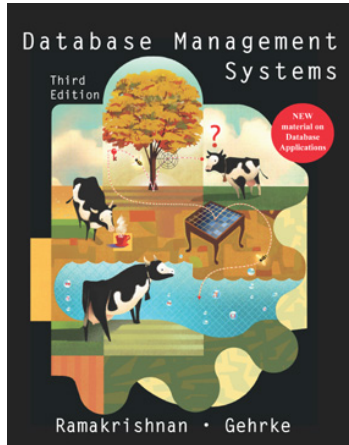
Message Board:

- <https://edstem.org/us/courses/3123/discussion/>
- Ask questions about the course, labs, homeworks
 - Feel free to answer questions too! If you think you know how to answer but are not sure, simply say so
 - Staff will check & answer questions regularly
 - If your question has not been answered in 12 hours, let me know
- Do not post any fragments of your code

Communication (part 3)

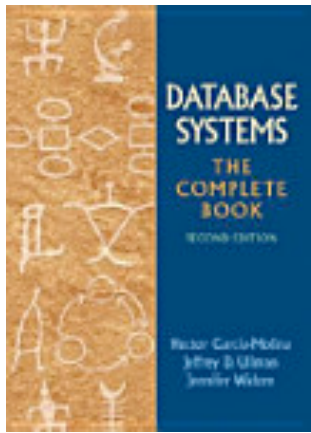
- Do **not** send questions by email unless
 - You need to discuss a personal matter
 - You want to setup an appointment
 - A question has not been answered on the board

Textbooks



Recommended textbook (pick one)

- **Database Management Systems. Third Ed.** Ramakrishnan and Gehrke. McGraw-Hill.



- *Database Systems: The Complete Book*, Hector Garcia-Molina, Jeffrey Ullman, and Jennifer Widom. **Second edition.**

See course website for recommended chapters

Other Readings

- See Website
- There is a section on reading assignments for 544M only

Grading CSE 444

- Labs: 43%
 - Includes final project lab
- Final project report 7%
- Six written assignments: 30%
- Two quizzes 20%

(above subject to +/- 5% adjustment)

Grading CSE 544M

- Same as CSE 444 plus
- Another 10% for the 4 paper reviews
- Then re-normalize to add up to 100%

- Graded separately from CSE 444

Five Labs

Acks: SimpleDB lab series originally developed by Prof. Sam Madden at MIT. We work with them on improving/extending.

- Lab 1: Build a DBMS that can scan a relation on disk
 - **Releasing later tonight! Part 1 of this lab is due on Monday!**
- Lab 2: Build a DBMS that can run simple SQL queries and also supports data updates
- Lab 3: Add a lock manager (transactions)
- Lab 4: Add a write-ahead log (transactions)
- ~~Lab 5: Add a query optimizer (not this quarter)~~
- Lab 6: Add support for parallel processing

About the Labs

Warning: I **will** run cheating-detecting software!
I have solutions from past years too.

Managed on GitLab:

[https://gitlab.cs.washington.edu/cse444-21wi/simple-db-\[your gitlab id\]](https://gitlab.cs.washington.edu/cse444-21wi/simple-db-[your gitlab id])

Will release tomorrow afternoon

Logistics:

- To be done **INDIVIDUALLY!**
- Each lab will take a **significant** amount of time
- Labs build on each other

Purpose

- Hands-on experience building a DBMS
- Deepen your understanding significantly
- We will build a *classical* DBMS

Six Homeworks

- Homework 1 releases tomorrow. Due next week
- Written assignments – **Print out pdf and fill in answers**
- Help review material learned in class
- Prepare you for the labs
 - One homework before each corresponding lab
- Go beyond what we implement in labs
- To be done **INDIVIDUALLY**

Exams

- No midterm!
- No final!
- Short take-home quizzes

- Quizzes represent knowledge from labs 1-4
- Tests depth of your knowledge
 - Only one or two open-ended questions
 - Example: “Explain how data is stored in SimpleDB”
 - Grades:
 - 9-10: Strength! Exceptional understanding and explanations
 - 8: You got it!
 - 7 or less: Developing knowledge – some gaps
 - 0: Did not show up or wrote nothing
 - Important: We grade based on the **depth of knowledge demonstrated in your answer**

Late Days

- Total of **6 late-days**
- Use in 24-hour chunks on hws or labs
- **At most 2 late-days per assignment**
- **No late-days can be applied to the final project due during finals week**

Outline (this lecture and next)

- Review of DBMS goals and features
- Review of relational model
- Review of SQL

Review: DBMS

- **What is a database?** Give examples
 - A collection of related files
 - E.g. payroll, accounting, products
- **What is a database management system?**
Give examples
 - A program written by someone else that manages the database; PostgreSQL, Oracle, ...
 - In 444 you are that “someone else”, implementing SimpleDB

Review: Data Model

- What is a data model?
 - A mathematical formalism for data
- What is the relational data model?
 - Data is stored in tables (aka relations)
 - Data is queried via relational queries
 - Queries are *set-at-a-time*

Review: Transactions

- What is a transaction?
 - A set of instructions that must be executed all or nothing
- What properties do transactions have?
 - ACID
 - Better: Serialization, recovery

Review: Data Independence

Review: Data Independence

The application should not be affected by changes of the physical storage of data

- Indexes
- Physical organization on disk
- Physical plans for accessing the data
- Parallelism: multicore, distributed

Key Data Management Concepts

- Data models: Relational, XML, graph data (RDF)
- Schema vs. Data
- Declarative query languages
 - Say what you want not how to get it
- Data independence
 - Physical: Can change how data is stored on disk without maintenance to applications
- Query compiler and optimizer
- Transactions: isolation and atomicity

Course Content

Focus: how to build a classical relational DBMS

- Review of the relational model (lecture 1 and 2)
- DBMS architecture and deployments (lecture 3)
- Data storage, indexing, and buffer mgmt (lectures 4-6)
- Query evaluation (lectures 7-8)
- Query optimization (lectures 9-12)
- Transactions (lectures 13-19)
- Parallel query processing (lectures 20-23)
- Replication and distribution (lectures 24-25)
- NoSQL and NewSQL (lectures 26-27)

Relational Model...

- The foundation of our traditional database management system
- We'll continue our review of the relational model next lecture ...