Introduction to Database Systems

CS432

Instructor: Johannes Gehrke
http://www.cs.cornell.edu/johannes
johannes@cs.cornell.edu

CS432/433: Introduction to Database Systems

- **How does Wal-Mart manage its 200 TB data warehouse?**
- **What is the database technology behind ebay's website?**
- **How do you build an Oracle 9i, IBM DB2 or Microsoft SQL Server database?**
- **How do build a search engine?**

CS432/433: Introduction to Database Systems

- Underlying theme: How do I build a data management system?
- CS432 will deal with the underlying *concepts*
  - No programming assignments
- CS433 will be the *practicum*
  - Build components of a small search engine (C++ programming)

CS432 Course Information

- Information is one of the most valuable resources in this information age
- How do we effectively and efficiently manage this information?
  - Relational database management systems
    - Dominant data management paradigm today
  - Search engines
    - Ubiquitous today
  - 100+ billion dollar a year industry
    - You will see this in the job market!

Prerequisites

- Courses
  - CS212 (Computers and Programming)
  - CS312 (Structure and Interpretation of Computer Programs)

People

- Instructor
  - Johannes Gehrke
- TAs
  - TBA

Access to Instructor and TAs

- Office hours
  - Posted on course web site
  - http://www.cs.cornell.edu/courses/cs432
- Course newsgroup
  - Monitored by TAs
  - Reply within 24 hours on weekdays, 48 hours on weekends
- TA mailing list
  - cs432ta-l@cs.cornell.edu
  - Do not directly email TAs

Course Structure

- Two components
  - Assignments (50%)
  - Two examinations (50%)
- No programming assignments in CS432
  - CS433 will have all programming assignments

Class Lectures

- Textbook: "Database Management Systems" (3rd Edition)
  - By Raghu Ramakrishnan and Johannes Gehrke
  - Required textbook
- Syllabus
  - Defined by class lectures, will be online tonight
  - Not defined by textbook

Course Structure

- Two components
  - Assignments (50%)
  - Examinations (50%)

Assignments

- Five assignments
- Each assignment worth 10% of total grade

Assignment Policies

- Assignments have to be done individually
  - No collaboration with others
- Academic integrity violations taken VERY seriously
  - Read Cornell and CS academic integrity policies
  - Available off course web page
  - Need to sign and hand in form
- Course management system used to post assignment grades

Assignment Policies (contd.)

- No late submissions
  - Will receive 0% of grade for late submissions
  - No exceptions (assignments handed out well in advance of deadline)
- Regrade requests
  - Within 7 days after assignments are graded
  - Hard deadline

Course Structure

- Two components
  - Assignments (50%)
  - Examinations (50%)

Exams

- Mid-term exam (20%)
  - 19 October 2006, 7:30-9:30pm
  - Closed book exam
- Final exam (30%)
  - Examination period
  - Closed book exam
  - Cumulative with emphasis on second half
- Do not schedule other exams on these days

Relationship to CS433

- CS432 is about *concepts* underlying databases
  - No programming assignments
- CS433 is the *practicum* associated with CS432
  - Will actually build a "realistic" search engine
  - C++ programming
- Complementary
  - Suggest that you take both
  - **Can** take CS432 without taking CS433
  - **Cannot** take CS433 without taking CS432

Is CS432/433 a lot of work?

- It depends!
  - Much of the material in CS432 is probably new to you
  - CS433 has substantial programming assignments
- Then why on earth should I take this course?
  - Intellectual argument
    - Big conceptual ideas
    - Meeting of theory and practice
  - Utilitarian argument
    - Many, many real applications (data management, data-driven websites, search engines,…)
    - Job market!

CS530: Architecture of Large-Scale Information Systems

- **How do you build e-commerce websites such as amazon.com?**
- **How do you build a reliable service that scales to millions of users?**
- **How are Internet transactions processed?**
- **How do you manage audio, video and XML data?**

CS530: Architecture of Large-Scale Information Systems

- Underlying theme: How do I build *applications* on top of a database system?
- Will combine coverage of fundamental concepts with "hands-on" experience
- Prerequisite: CS432

CS530: Material Covered

- Three-tier architectures
- Edge caches
- Distributed transaction management
- Web services
- Content management

- Technologies: .NET, JSPs, ASPs, Servlets, Enterprise Java Beans (EJBs), XML, SOAP

Reminder

- Complete academic integrity form (on the web)
  - Need to hand this in for your course management system account

What Is a DBMS?

- A very large, integrated collection of data.
- Models real-world *enterprise.*
  - Entities (e.g., students, courses)
  - Relationships
- A *Database Management System (DBMS)* is a software package designed to store and manage databases.

Files vs. DBMS

- Application must stage large datasets between main memory and secondary storage (e.g., buffering, page-oriented access, 32-bit addressing, etc.)
- Special code for different queries
- Must protect data from inconsistency due to multiple concurrent users
- Crash recovery
- Security and access control

Why Use a DBMS?

- Data independence and efficient access.
- Reduced application development time.
- Data integrity and security.
- Uniform data administration.
- Concurrent access, recovery from crashes.

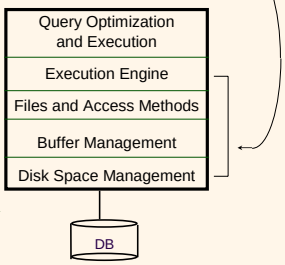
Why Study Databases??

- Shift from *computation* to *information*
  - at the "low end": scramble to webspace (a mess!)
  - at the "high end": scientific applications
- Datasets increasing in diversity and volume.
  - Digital libraries, interactive video, Myspace, YouTube, Google
  - ... need for DBMS exploding
- DBMS encompasses most of CS
  - OS, languages, theory, data mining, multimedia, logic

Structure of a DBMS

These layers must consider concurrency control and recovery

- A typical DBMS has a layered architecture.
- The figure does not show the concurrency control and recovery components.
- This is one of several possible architectures; each system has its own variations.



Summary

- DBMS are used to store and query large datasets.
- Benefits include recovery from system crashes, concurrent access, quick application development, data integrity and security.
- Levels of abstraction give data independence.
- A DBMS typically has a layered architecture.
- Data management R&D is one of the broadest, most exciting areas in CS.