Database Systems I (91.309)

- Content:
  - How to build a database application
  - Principles of database-system implementation

- Texts:
  - Ramakrishnan and Gehrke, *Database Management Systems*, third edition
  - Oracle handouts
  - Sieg, *How To Build a Database System* (draft)

- Web site: www.cs.um.edu/~john/91.309/
- If you didn’t receive “distribution list test” email, please ask to be added to e-mail list.

Prof. John Sieg

- Research in query optimization and scheduling transactions
- Office: Olsen 211
- E-mail: john@cs.um.edu
- Home page (including office hours, etc.): www.cs.um.edu/~john

Objectives

- Students will learn to effectively design and implement database applications.
- Students will become fluent in design methodologies and familiar enough with database system internals that they can make informed design and implementation decisions.

91.309 - Grading

- Application Project 20 %
- Systems Project 20 %
- Quiz 1 20 %
- Quiz 2 20 %
- Quiz 3 20 %

Quizzes

- three, non-cumulative, last one during final exams
- closed book
- Except for documented emergencies, quizzes must be taken on (before) quiz date.

Application Project

- Design and build a database application using the Oracle database system
- Deliverables: design document and diskette(s)
- See project links on course web page (also: handout).
Systems Project

- to be determined
- Language: C++, platform: Windows .NET
- Deliverables: document and diskette(s)
- See system project link on course web page.

Definitions

- *database* - a shared, persistent, conveniently and efficiently accessible collection of data
- *database system* - the software that maintains and accesses databases (e.g., Microsoft Access)
- *database application* - software that solves specific problems by using a database system

Definitions, cont’d

- *relational database* - a database in which data are organized in tables (relations)
- *query* - a request for data from the database; can use a GUI or a query language
- *QBE* - Query-By-Example, a GUI for querying relational databases
- *SQL* - the standard query language for relational databases

People

- Application User
- Designer
- Developer
- Database Administrator
- Database System Administrator

Database System Services: Convenience

- query language support
- design tools
- software development tools
- data independence
- programming language access
- support for building open systems
- support for decision making
- Web publishing

Database System Services: Under the Covers

- efficient data storage and access
- buffer management
- query optimization
- security enforcement
- integrity control
- concurrency control
- recovery
Using Files vs. Using Databases

- Files are OS-dependent.
- Files usually must be accessed via programs.
- Concurrency, recovery, and other services are primitive in file systems.
- File systems do not support data independence.
- Files are not good for modeling interesting data.
- Etc.!

Three-Level DB Architecture

Client-Server Architecture

- Clients run GUIs to interact with user, e.g., to display data.
- Servers access DB and contain most of the database system services.

Three-Tiered Architecture

- Middle layer ("middleware") contains Web server and manages database system access across multiple databases.