Introduction to Database Systems

CSE 444

Lecture #1
Jan 3 2001

Staff

Instructor: Surajit Chaudhuri
Contact: surajitc@microsoft.com
Office hours: MW 4.50-5.20 (in Class)

TA: Yana Kadiysk
Sieg 226B, ykadiysk@cs.washington.edu
Office hours: Tu 9-10am, Fr 10-11am

NOTE: Your email to either of us must have CSE444 as the first word in the Subject line. Otherwise, it will be ignored

Textbook(s)

- A First Course in Database Systems
  by Jeff Ullman and Jennifer Widom
- Database Implementation
  by Hector Garcia-Molina, Jeff Ullman and Jennifer Widom
- Available in a shrink-wrapped package at the book store
  not available in that form for non-students

Other Reference Books

- Database Management Systems
  Ramakrishnan
- Fundamentals of Database Systems, Elmasri and Navathe
- Both are on hold in the library
Misc Administrative Issues

- Homework
  - See homepage for homework deadlines
  - No extension granted
- Course credit
  - Project 25%
  - Homework 15%
  - Programming Assignments 10%
  - Midterm 15%
  - Final 30%
- Prerequisites: CSE-326 or equivalent

Resolving Questions

- Follow the Sequence:
  1. Look at the CSE444 Hypermail archive
  2. If unresolved, determine whom you should contact
     - Project, Software, Homework Assignments: Yana
     - Concepts, Class Lectures: Surajit
  3. Try to come for the office hour of the right contact
  4. Send email to the right contact
     - NOTE: Your email to either of us must have CSE444 as the first word in the Subject line. Otherwise, it will be ignored

Wide World of Information

- Text Documents
  - Text, Word, Powerpoint Files, HTML pages
  - Indexed and searched by "Search Engines"
- Structured Information
  - Databases, Spreadsheets
  - Drives businesses
  - Focus of this course
- Future: Richer Integration

An Architecture for Structured Information Systems

- Web Browser as the user interface
- Web Server talks to an application-server
  - Supports business objects
- Application Server talks to a database server
  - Supports data objects
  - Focus of this course
Examples of Structured Information Systems

- Banking System
- Airline Reservation System
- Inventory Management
- Amazon.com, Dell.com, Etrade.com

Example: SCBook.com

- Data Structures
  - (Bookid, Publisherid, Title, ISBN, Price, topic)
  - (Bookid, Count)
  - (Publisherid, Pub_Price)
  - (Orderid, Publisherid, Bookid, Order_Count)
  - (Custid, Name, Address1, City)

- Applications
  - Report Sales by City and Topic
  - Order/receive more copies of a book
  - Buy a book

Some Characteristics

- Large Volumes of structured data
- Multi-user, Multi-application system
- Key Issues
  - Data structure
  - Application Development
  - Concurrency
  - Recovery
- DBMS: Software to simplify development of information systems

Why not use File System?

- Problems with virtual memory
  - Database sizes > 10T
  - Need advanced storage management
- Applications need to be smart to deal with large volumes of data
  - Good performance is crucial
  - Support high degree of parallelism
- Multiple applications
  - Different views to different applications
Why not use a File System?

- Data Integrity is key
  - Failure, Concurrency tolerant
  - Fine-Grained security
- Evolution in data structures
  - Need to rewrite applications

Key Observations

- Tabular data: simplest, widely used
- Tabular data in, tabular data out
  - Add/Remove/Update rows
  - Select subset of rows and columns
  - Combine information from multiple tables
  - Produce Reports
- Pick data structures carefully
- Serialize all user interactions
  - Success or Failure
  - Successful actions are permanent

Services from a DBMS

- High Level Programming on Relations
  - Query language: Set-Oriented Access
  - Data Definition Language - DDL
  - Data Manipulation Language - DML
  - Physical Data Independence
  - Data Integrity
- Transaction Management
  - Concurrency control
  - Recovery
- Storage Management
  - Indexes, Clustering

Questions the Course Addresses

- What are the services rendered by a DBMS?
  - High-Level Programming, Data Integrity
  - Transaction
  - Storage
- How do we use a commercial DBMS to implement an information system?
  - Design and Implementation
  - Web-based application
  - Hands-on experience (The Project)
- How is a DBMS built?
Building a Database for an Information System

- Model data from an information-centric viewpoint
  - Conceptual Database Design (ER Diagrams)
- Define Relational Schema
- Develop Application(s) using Query Languages
  - Views (virtual schema)
  - Stored Procedures
- Physical Database Design (indexes, clustering)

Abstraction: Logical Schema and Views

- Views describe how users see the data.
- Logical schema defines logical structure using relational data model
- Physical schema describes the files and indices used.

Example: University Database

- Logical Schema:
  - Students(sid: string, name: string, login: string, age: integer, gpa: real)
  - Courses(cid: string, cname: string, credits: integer)
  - Enrolled(sid: string, cid: string, grade: string)
- A possible Physical Schema:
  - Relations stored as unordered files.
  - Index on first column of Students.
- An External Schema (View):
  - Course_info(cid: string, enrollment: integer)
### Schema Design and Implementation

- **Tables:**
  - **Students:**
    - SSN | Name | Category
    - 123-45-6789 | Charles | undergrad
    - 234-56-7890 | Dan | grad
  - **Takes:**
    - SSN | CID
    - 123-45-6789 | CSE444
    - 123-45-6789 | CSE444
    - 234-56-7890 | CSE142
  - **Courses:**
    - CID | Name | Quarter
    - CSE444 | Databases | Fall
    - CSE541 | Operating systems | Winter

- Separates the logical view from the physical view of the data
- Build appropriate indexes

---

### Data Independence

- Applications insulated from how data is structured and stored.
- **Logical data independence**: Protects views from changes in logical (conceptual) structure of data.
- **Physical data independence**: Protects conceptual schema from changes in physical structure of data.

One of the most important benefits of using a DBMS!

---

### Building Applications: Querying a Database

- Find all courses that "Mary" takes
- S(tructured) Q(uery) L(anguage)
  ```sql
  select C.name 
  from Students S, Takes T, Courses C 
  where S.name = "Mary" and 
  S.ssn = T.ssn and T.cid = C.cid
  ```

---

### Inside a DBMS

- 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.
Looking Ahead: Role of XML

- XML as the universal transport
  - Semi-structured and hierarchical
  - Efficient publishing of information in XML
  - Efficient storage of information in XML
  - "XML Stores" and/or "Native Stores"

Database Professionals

- Server Implementers
  - Application Developers
  - Database Administrators
    - Use knowledge of server and applications to tune databases
    - Physical design, security,..
  - End-Users of Applications

Database Industry

- Relational databases are a great success
- Servers
  - Oracle, IBM, Microsoft, Sybase, Informix, SQL, Compaq,..
- Client Tools for Database development
  - Many ISV-s
- Major Application vendors
  - SAP, Peoplesoft, ..

Course Outline

- High-Level Programming on Databases using SQL
  - Query Language (including views)
  - Web-based end-to-end application
- Database Design
  - Entity Relationship diagrams
  - Transforming E/R models to relational schemas
  - Normalization
Course Outline (2)

- Transactions
- Inside a DBMS
  - Storage and Indexes
  - Query Processing
  - Query Optimization
- Information Exchange on Internet: XML
- Special Topics

Course Project

- Goal: design a database application using ASP
- Choose topic on your own.
  - Some service projects available.
- Work in groups of 3-4 (start forming now)