Introduction to Database Systems

CSE 444

Lecture #1
September, 25 2000

Staff

- Instructor: Dan Suciu
  - Sieg, Room 318, suciu@cs.washington.edu
  - Office hours: Wednesday 2:30-3:30
  - (or by appointment)
- TA: Stefan Saroiu
  - Sieg 226b, tzoompy@cs.washington.edu
  - Office hours: TBA (check mailing list)

Communications

- Web page: http://www.cs.washington.edu/444/
- Mailing list: send email to
  majordomo@cs
  saying:
  subscribe cse444

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 Texts

- Database Management Systems, *Ramakrishnan*
  - very comprehensive
- Fundamentals of Database Systems, *Elmasri and Navathe*
  - very widely used
- Foundations of Databases, *Abiteboul, Hull and Vianu*
  - Mostly theory of databases
- Data on the Web, *Abiteboul,Buneman,Suciu*
  - XML and other new/advanced stuff

Available on reserve, at the library

Traditional Database Application

Suppose we are building a system to store the information about:

- students
- courses
- professors
- who takes what, who teaches what

Why use a DBMS ?

What we need from a database:

- store the data for a long period of time
  - large amounts (100s of GB)
  - protect against crashes
  - protect against unauthorized use
- allow users to query/update:
  - who teaches "CSE142"
  - enroll "Mary" in "CSE444"

- allow several (100s, 1000s) users to access the data simultaneously
- allow administrators to change the schema
  - add information about TAs

Trying Without a DBMS

Why Direct Implementation Won't Work:

- Storing data: file system is limited
  - size less than 4GB (on 32 bits machines)
  - when system crashes we may loose data
  - password-based authorization insufficient
- Query/update:
  - need to write a new C++/Java program for every new query
  - need to worry about performance

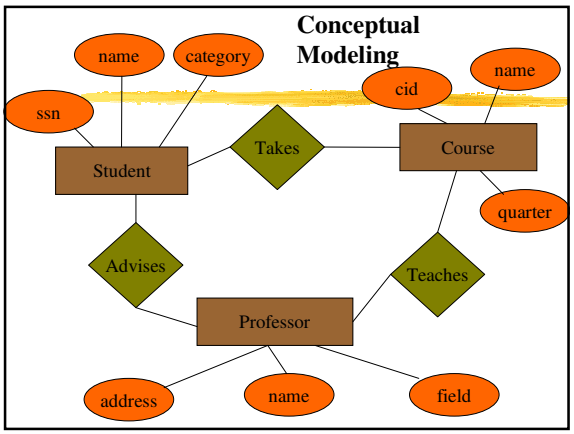
- Concurrency: limited protection
  - need to worry about interfering with other users
  - need to offer different views to different users (e.g. registrar, students, professors)
- Schema change:
  - need to rewrite virtually all applications

Functionality of a DBMS

- Storage management
- Data Manipulation Language - DML
  - query language
- Transaction Management
  - concurrency control
  - recovery
- Data Definition Language - DDL

Building an Application with a DBMS

- Requirements modeling (conceptual, pictures)
  - Decide what entities should be part of the application and how they should be linked.
- Schema design and implementation
  - Decide on a set of tables, attributes.
  - Define the tables in the database system.
  - Populate database (insert tuples).
- Write application programs using the DBMS
  - way easier now that the data management is taken care of.



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.

Querying a Database

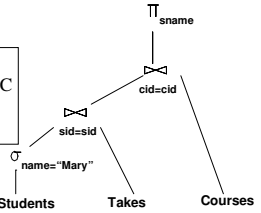
- Find all courses that "Mary" takes
- S(tructured) Q(uery) L(anguage)
  - **select** C.name
  - **from** Students S, Takes T, Courses C
  - **where** S.name="Mary" and
  - S.ssn = T.ssn and T.cid = C.cid
- Query processor figures out how to answer the query efficiently.

Query Optimization

Goal:

Declarative SQL query ⟶ *Imperative query execution plan:*

```
select C.name
from Students S, Takes T, Courses C
where S.name="Mary" and
  S.ssn = T.ssn and T.cid = C.cid
```



Plan: Tree of Relational Algebra operators, with a choice of algorithm implementation for each operator

Ideally: Want to find best plan. Practically: Avoid worst plans!

Current and Future Data Management

- Current Data Management:
  - relational data for enterprise applications
  - storage
  - query processing/optimization
  - transaction processing
- Future Data Management:
  - XML data for exchange on the Web
  - transport
  - query/data translation
  - information retrieval

Database Industry

- Relational databases are a great success of theoretical ideas.
- "Big 3" DBMS companies are among the largest software companies in the world.
- IBM (with DB2) and Microsoft (SQL Server, Microsoft Access) are also important players.
- $20B industry.
- Challenged by object oriented DBMS.

The Study of DBMS

- Several aspects:
  - Modeling and design of databases
  - Database programming: querying and update operations
  - Database implementation
- DBMS study cuts across many fields of Computer Science: OS, languages, AI, Logic, multimedia, theory...

Course (Rough) Outline

- Database design:
  - Entity Relationship diagrams
  - ODL (object-oriented design language)
  - Modeling constraints
- The relational model:
  - Relational algebra
  - Transforming E/R models to relational schemas
- XML: a data format for the Web

Outline (Continued)

- SQL ("intergalactic dataspeak")
  - Views and triggers
- Advanced query languages:
  - Recursive queries and datalog
  - Object-oriented features
  - Queries for XML

Outline (Continued)

- Storage and indexing
- Query optimization
- Transaction processing and recovery
- Advanced topics

Structure

- Prerequisites: Data structures course (CSE-326 or equivalent).
- Work & Grading:
  - Homework 25%: 6 of them, some light programming.
  - Project: 30% - see below.
  - Midterm: 15%
  - Final: 25%
  - Intangibles: 5%

The Project

- Goal: design end-to-end database application.
- Work in groups of 3-4 (start forming now).
- Choose topic on your own.
- Some service projects available.
- Timetable for project milestones.
- Be creative!
- Start soon!!