Quick Introduction to Database Systems

May 30, 2003

Outline

- Why do we need a different kind of system?
- What is a database system?
- Separating the *what* from the *how:*
  - The relational data model
  - Querying the databases: SQL

Note: A typical database system is over 1 million lines of code.

Data Manipulation Tasks

- Many applications require complex manipulation of *data*:

- Bank accounts
- Airline reservations
- Books on Amazon
- Random items on E-Bay
- Students and grades
- Data from scientific experiments
- Preference data on match.com

What do they all have in common?

- Need to store a huge amount of data:
  - Data doesn't fit in main memory. Need disks, sometimes tapes
  - It's not enough to build an array. Need sophisticated methods to access data on disk.
- Data is *valuable:*
  - Needs to be there even after a disk crash or natural disaster.
- Many users are accessing the data concurrently
  - Need to make sure they don't interfere with each other.
- Need to ask complex *queries:*
  - Find books written by authors whose other books sold to customers who spend over $100 on Amazon per month.

Protection from Crashes

- **Suppose we're transferring money between accounts:**

```
•/** Transfer the given amount from otherAccount to this BankAccount */
    public void transfer(double amount, BankAccount otherAccount) {
                balance = balance + amount;
```

System Crash

```
                otherAccount.balance = otherAccount.balance – amount;
                }
```


Supporting Concurrency

Making a reservation:

- Check if the seat is available
- Reserve the seat.

Possible scenario:

Customer 1 - finds a seat empty
Customer 2 - finds the same seat empty

Customer 1 - reserves the seat.
Customer 2 - reserves the seat.

Customer 1 will not be happy.

Trying Without a DBMS

Why Direct Implementation Won't Work:

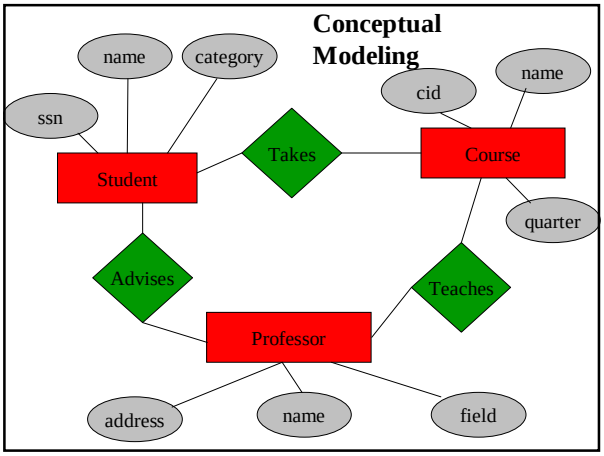
- **Storing data: file system is limited**
  - **size less than 4GB (on 32 bits machines)**
  - **when system crashes we may loose data**
  - **password-based authorization insufficient**
- **Query/update:**
  - **need to write a new Java program for every new query**
  - **need to worry about performance**

Functionality of a DBMS

- **Persistent storage management**
- ***Transaction* management**
- **Resiliency: recovery from crashes.**
- **Separation between logical and physical views of the data.**
  - **High level query and data manipulation language.**
  - **Efficient query processing**
- **Interface with programming languages**

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: SQL

- Find all undergraduates:

```
select  Name
from    Students
where   Category = 'undergrad'
```

Students

SSN	Name	Category
123-45-6789	Charles	undergrad
234-56-7890	Dan	grad
	...	...

Querying a Database: Aggregation

- **Find all how many students there are from each category**

```
select   Category, Count(SSN) as Number
from     Students
group by Category
```

Students

SSN	Name	Category
123-45-6789	Charles	undergrad
234-56-7890	Dan	grad
	...	...

Category	Number
undergrad	523
grad	231
confused	17
	...

Querying a Database: Join

- **Find graduate students taking 142**

```
select   Students.SSN, Name
from     Students, Takes
where    Takes.SSN=Students.SSN AND
         Category = "graduate" AND CID=CSE142
```

Students

SSN	Name	Category
123-45-6789	Charles	undergrad
234-56-7890	Dan	grad
	...	...

Takes

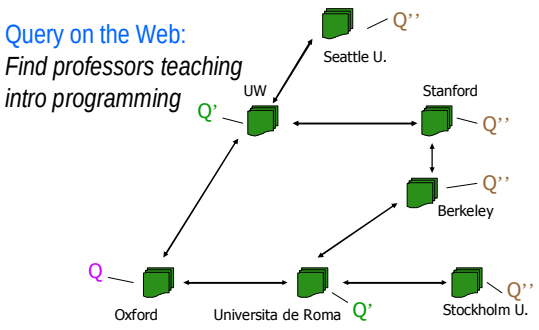
SSN	CID
123-45-6789	CSE444
123-45-6789	CSE444
234-56-7890	CSE142
	...

Database Industry

- **Relational databases are a great success of theoretical ideas.**
- **Big DBMS companies are among the largest software companies in the world.**
- **Oracle**
- **IBM (with DB2)**
- **Microsoft (SQL Server, Microsoft Access)**
- **Sybase**
- **$20B industry.**

The Semantic Web (e.g., our Piazza)

Query on the Web:
Find professors teaching intro programming



Summary

- Databases are an incredibly useful tool, whether you are a computer scientist or not!
- They enable you to focus on what you want to do with the data, not how to store it and how to access it efficiently.
- Three main concepts:
  - Transactions: a set of operations treated as an atomic unit.
  - Logical data model: think of the data as tables, the DBMS will worry how to store and access them
  - SQL: a high level query language