Introduction to Database Systems
CSE 444

Lecture #1

September 27, 2006

About Me

Dan Suciu:

- Joined the department in 2000
- Before that: Bell Labs, AT&T Labs

Research:

- Past: XML and semi-structured data:
  - Query language: XML-QL (later XQuery)
  - Compressor: XMill
  - Theory: XPath containment, XML typechecking
- Present: Probabilistic databases: MystiQ

Staff

- Instructor: Dan Suciu
  - Allen, Room 662, suciu@cs.washington.edu
    Office hours: Wednesdays 11:30 (appointment strongly recommended)

- TAs:
  - Jue Wang, juewang@cs.washington.edu
    Office hours: Fridays 1:00-2:00, Room TBA

Communications

- Web page:
  http://www.cs.washington.edu/444/
  - Lectures will be available here
  - Homeworks will be posted here (HW1 is posted)
  - The project description will be here

- Mailing list:
  - Announcements, group discussions
  - Please subscribe

Textbook(s)

Main textbook, available at the bookstore:

- *Database Systems: The Complete Book*,
  Hector Garcia-Molina,
  Jeffrey Ullman,
  Jennifer Widom

Most chapters are good. Some are not (functional dependecies).
COME TO CLASS ! ASK QUESTIONS ! READ SLIDES !

Other Texts

Available at the Engineering Library (not on reserve):

- *Database Management Systems*, Ramakrishnan
- *XQuery from the Experts*, Katz, Ed.
- *Fundamentals of Database Systems*, Elmasri, Navathe
- *Foundations of Databases*, Abiteboul, Hull, Vianu
- *Data on the Web,* Abiteboul, Buneman, Suciu

Outline of Today's Lecture

1. Overview of DBMS
2. DBMS through an example
3. Course outline
4. Assignment 1, Homework 1

Database

What is a database ?

Give examples of databases

Database

What is a database ?

- A collection of files storing related data

Give examples of databases

- Accounts database; payroll database; UW's students database; Amazon's products database; airline reservation database

Database Management System

What is a DBMS ?

Give examples of DBMS

Database Management System

What is a DBMS ?

- *A big C program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time*

Give examples of DBMS

- DB2 (IBM), SQL Server (MS), Oracle, Sybase
- MySQL, Postgres, …

Market Shares

From 2004 www.computerworld.com

- IMB: 35% market with $2.5BN in sales
- Oracle: 33% market with $2.3BN in sales
- Microsoft: 19% market with $1.3BN in sales

An Example

The Internet Movie Database
http://www.imdb.com

- Entities:
  Actors (800k), Movies (400k), Directors, …
- Relationships:
  who played where, who directed what, …

Want to store and process locally; what functions do we need ?

What the Database Systems Does

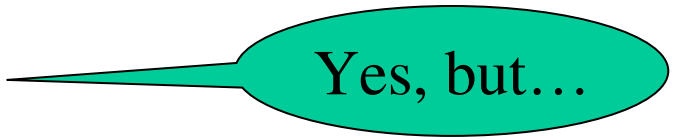
1. Create/store large datasets
2. Search/query/update
3. Change the structure
4. Concurrent access to many user
5. Recover from crashes
6. Security (not here, but in other apps)

Possible Organizations

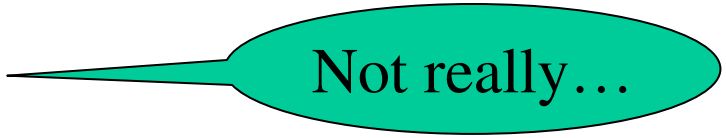
- Files
- Spreadsheets
- DBMS

1. Create/store Large Datasets

- Files



- Spreadsheets



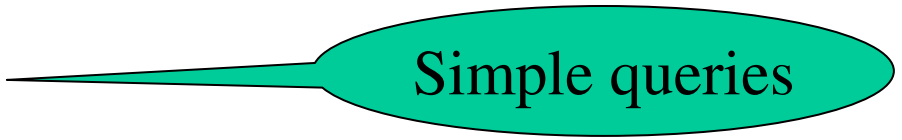
- DBMS

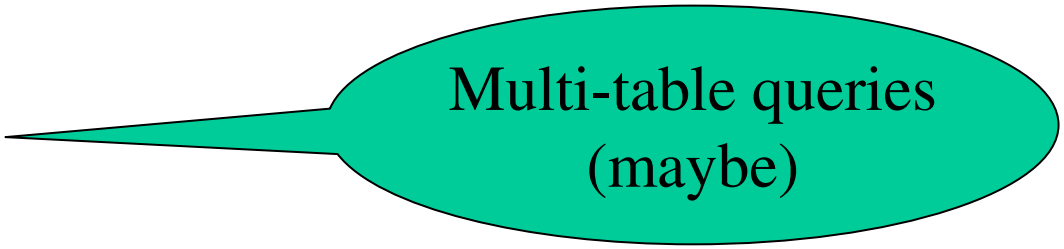
2. Search/Query/Update

- Simple query:
  - In what year was *'Rain man'* produced ?
- Multi-table query:
  - Find all movies by *'Coppola'*
- Complex query:
  - For each actor, count her/his movies
- Updating
  - Insert a new movie; add an actor to a movie; etc

2. Search/Query/Update

- Files
- Spreadsheets
- DBMS



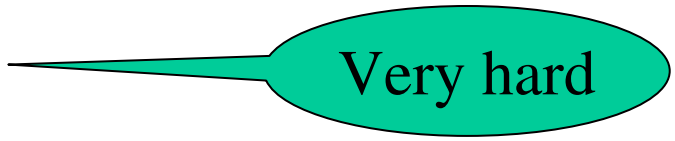


Updates: generally OK

3. Change the Structure

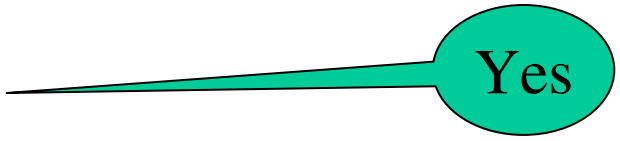
Add *Address* to each Actor

- Files



- Spreadsheets

- DBMS



4. Concurrent Access

Multiple users access/update the data concurrently

- What can go wrong ?
- How do we protect against that in OS ?
- This is insufficient in databases; why ?

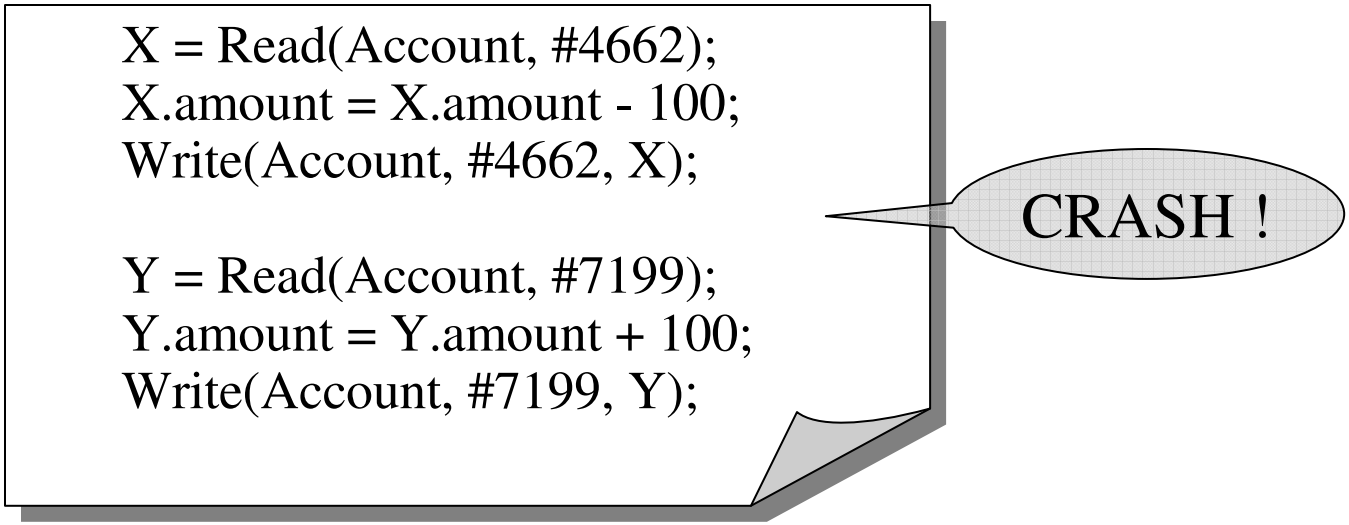
4. Concurrent Access

Multiple users access/update the data concurrently

- What can go wrong ?
  - Lost update; resulting in inconsistent data
- How do we protect against that in OS ?
  - Locks
- This is insufficient in databases; why ?
  - A logical action consists of *multiple* updates

5. Recover from crashes

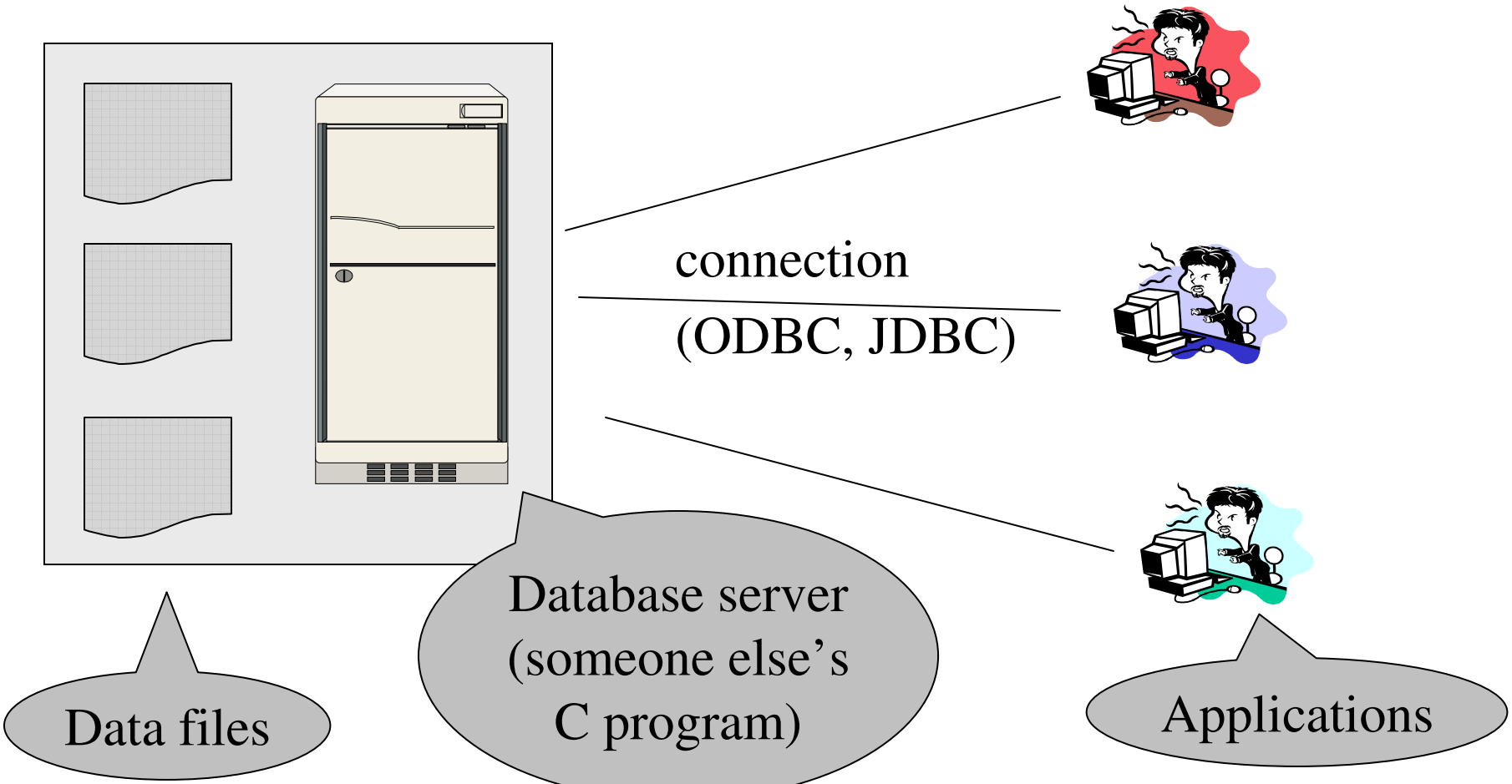
- Transfer $100 from account #4662 to #7199:



What is the problem ?

Enters a DMBS

"Two tier system" or "client-server"



DBMS = Collection of Tables

Directors:

id	fName	lName
15901	Francis Ford	Coppola
. . .		

Movie_Directors:

id	mid
15901	130128
. . .	

Movies:

mid	Title	Year
130128	The Godfather	1972
. . .		

Still implemented as files,
but behind the scenes can be quite complex

"*data independence*"

1. Create/store Large Datasets

Use SQL to create and populate tables:

```
CREATE TABLE Actors (
   Name CHAR(30)
   DateOfBirth CHAR(20)
)   . . .
```

```
INSERT INTO Actors
VALUES('Tom Hanks', . . .)
```

Size and physical organization is handled by DBMS

We focus on modeling the database

Will study data modeling in this course

2. Searching/Querying/Updating

- Find all movies by 'Coppola'

```
SELECT  title
FROM    Movies, Directors, Movie_Directors
WHERE   Directors.lname = 'Coppola' and
        Movies.mid = Movie_Directors.mid and
        Movie_Directors.id = Directors.id
```

We will study SQL in gory details in this course

- What happens behind the scene ?

We will discuss the query optimizer in class.

3. Changing the Structure

Add *Address* to each Actor

```
ALTER TABLE Actor
    ADD address CHAR(50)
    DEFAULT 'unknown'
```

Lots of cleverness goes on behind the scenes

3&4 Concurrency&Recovery: Transactions

- A *transaction* = sequence of statements that either all succeed, or all fail
- E.g. Transfer $100

```
BEGIN TRANSACTION;

UPDATE Accounts
SET amount = amount - 100
WHERE number = 4662

UPDATE Accounts
SET amount = amount + 100
WHERE number = 7199

COMMIT
```

Transactions

- Transactions have the ACID properties:

  A = atomicity

  C = consistency

  I = isolation

  D = durability

4. Concurrent Access

- Serializable execution of transactions
  - The I (=isolation) in ACID

We study three techniques in this course

Locks

Timestamps

Validation

5. Recovery from crashes

- Every transaction either executes completely, or doesn't execute at all
  - The A (=atomicity) in ACID

We study three types of log files in this course

Undo log file

Redo log file

Undo/Redo log file

Course Outline

Part I

- SQL, Relational model, database design
- XML, XPath, Xquery
- **Midterm**: Friday, October 27 (in class)

Part II

- Database security, Transactions
- Concurrency control and recovery
- Query execution and optimization

Final: Monday, December 11, 8:30-10:20(this room)

Grading

- Homework: 30%
- Project: 25%
- Midterm: 15%
- Final: 25%
- Intangibles: 5%

The Project

- Models data management needs of a company
- Will have four phases
- We use SQL Server, C#, .NET
- First phase: handed out next week

Assignment 1, Homework 1

- Reading assignment for Friday:
  - **Introduction** from **SQL for Web Nerds,** by Philip Greenspun, http://philip.greenspun.com/sql/

- Login SQL Server
  - User name = your U email address
  - Password = "studentID" + "!A"

- Homework 1: due Wednesday, October 11 http://www.cs.washington.edu/education/courses/cse444/CurrentQtr/hw/index.htm

Accessing SQL Server

SQL Server Management Studio

- Server Type = Database Engine
- Server Name = IISQLSRV
- Authentication = SQL Server Authentication
  - Login = your email address
  - Password = 11111111

Change your password !!

Then play with IMDB