Introduction to Database Systems
CSE 444

Lecture 1
Introduction
About Me: General

Prof. Magdalena Balazinska (magda)
- At UW since January 2006
- PhD from MIT
- Born in Poland
- Grew-up in Poland, Algeria, and Canada
About Me: Research

• **Past: Stream Processing**
  – Distributed stream processing (Borealis)
  – Load management and fault-tolerance
  – RFID data management
  – Probabilistic event processing

• **Now: Cloud computing and scientific data mgmt**
  – Collaboration with astronomers, oceanographers, etc.
  – Making large-scale data analysis interactive
  – Collaborative query management
Staff

• **Instructor:** Magdalena Balazinska
  – CSE 550, magda@cs.washington.edu
  Office hours: Mondays 1:30pm-3:20pm

• **Grad TA:** Nodira Khoussainova
  – nodira@cs.washington.edu
  – Office hours: TBA

• **Ugrad TA:** Michael Rathanapinta
  – michaelr@cs.washington.edu
Communications

• **Web page:** [http://www.cs.washington.edu/444](http://www.cs.washington.edu/444)
  - Lectures will be available there
  - The mini-projects description will be there
  - Homeworks will be posted there

• **Mailing list**
  - Announcements, group discussions
  - You are already subscribed
Textbook

Main textbook, available at the bookstore:


Most important: COME TO CLASS! ASK QUESTIONS!
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
Course Format

• Lectures MWF, 12:30-1:20pm
• Quiz sections: Th 9:30-10:20, 10:30-11:20
  – Location to be announced

• 4 Mini-projects
• 3 homework assignments

• Midterm and final
Grading

- Homeworks  30%
- Mini-projects  30%
- Midterm  15%
- Final  25%
Four Mini-Projects

1. SQL
2. SQL in Java
3. Database tuning
4. Parallel processing: MapReduce

Due: Wednesdays every other week
Three Homework Assignments

1. Conceptual Design
2. Transactions
3. Query execution and optimization

Due: Wednesdays every other week
Exams

• Midterm: Friday, May 8, in class

• Final: Thursday, June 11, 8:30-10:20am, in class
Outline of Today’s Lecture

1. Overview of a DBMS
2. A DBMS through an example
3. Course content
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 DBMSs
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 DBMSs

• DB2 (IBM), SQL Server (MS), Oracle, Sybase
• MySQL, PostgreSQL, ...

We will focus on relational DBMSs most quarter
Market Shares

From 2004 www.computerworld.com

• IBM: 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, …
Required Data Management Functionality

1. Describe real-world entities in terms of stored data
2. Create & persistently store large datasets
3. Efficiently query & update
   1. Must handle complex questions about data
   2. Must handle sophisticated updates
   3. Performance matters
4. Change structure (e.g., add attributes)
5. Concurrency control: enable simultaneous updates
6. Crash recovery
7. Security and integrity
DBMS Benefits

• Expensive to implement all these features inside the application

• DBMS provides these features (and more)

• DBMS simplifies application development

How to decide what features should go into the DBMS?
### Back to Example: Tables

#### Actor:

<table>
<thead>
<tr>
<th>id</th>
<th>fName</th>
<th>lName</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>195428</td>
<td>Tom</td>
<td>Hanks</td>
<td>M</td>
</tr>
<tr>
<td>645947</td>
<td>Amy</td>
<td>Hanks</td>
<td>F</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Cast:

<table>
<thead>
<tr>
<th>pid</th>
<th>mid</th>
</tr>
</thead>
<tbody>
<tr>
<td>195428</td>
<td>337166</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Movie:

<table>
<thead>
<tr>
<th>id</th>
<th>Name</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>337166</td>
<td>Toy Story</td>
<td>1995</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

CSE 444 - Spring 2009
SQL

SELECT *
FROM Actor
SQL

```
SELECT count(*)
FROM Actor
```

This is an aggregate query
SQL

```
SELECT *
FROM Actor
WHERE lName = 'Hanks'
```

This is a *selection query*
SELECT *
FROM Actor, Cast, Movie
WHERE lname='Hanks' and Actor.id = Cast.pid and Cast.mid=Movie.id and Movie.year=1995

This query has selections and joins

We will learn SQL in all its glory in 4 lectures!
How Can We Evaluate the Query?

<table>
<thead>
<tr>
<th>Actor:</th>
<th>Cast:</th>
<th>Movie:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id</strong></td>
<td><strong>pid</strong></td>
<td><strong>id</strong></td>
</tr>
<tr>
<td><strong>fName</strong></td>
<td><strong>mid</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>lName</strong></td>
<td></td>
<td><strong>year</strong></td>
</tr>
<tr>
<td><strong>gender</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Hanks</td>
<td></td>
<td>1995</td>
</tr>
</tbody>
</table>

Plan 1:  . . . [ in class ]

Plan 2:  . . . [ in class ]
Evaluating Tom Hanks
What an RDBMS Does Well (1/2)

- Indexes: on Actor.lName, on Movie.year
- Multiple implementations of joins
- Query optimization (which join order ?)
- Statistics!

We’ll learn all about this in May
Now Let’s See Database Updates

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

X = Read(Account, #4662);
X.amount = X.amount - 100;
Write(Account, #4662, X);

Y = Read(Account, #7199);
Y.amount = Y.amount + 100;
Write(Account, #7199, Y);
Now Let’s See Database Updates

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

X = Read(Account, #4662);
X.amount = X.amount - 100;
Write(Account, #4662, X);

Y = Read(Account, #7199);
Y.amount = Y.amount + 100;
Write(Account, #7199, Y);

What is the problem?
What a RDBMS Does Well (2/2)

Transactions!

- Recovery
- Concurrency control

We will learn all that in April
Client/Server Architecture

• There is a single server that stores the database (called DBMS or RDBMS):
  – Usually a beefy system, e.g. IISQLSRV1
  – But can be your own desktop…
  – … or a huge cluster running a parallel dbms

• Many clients run apps and connect to DBMS
  – E.g. Microsoft’s Management Studio
  – Or psql (for postgres)
  – More realistically some Java or C++ program

• Clients “talk” to server using JDBC protocol
Data Management v.s. Databases

• There is more to Data Management!

A Data Management QUIZ:
• Alice sends Bob in random order all the numbers 1, 2, 3, …, 100000000000000000000
• She does not repeat any number
• But she misses exactly one!
• Help Bob find out which one is missing!
What This Course Contains

• SQL
• Conceptual Design
• Transactions
• Database tuning and internals (very little)
• Distributed databases: a taste of MapReduce
• More data management
  – Sampling, data cleaning, etc.
• XML: Xpath, Xquery
Accessing SQL Server

SQL Server Management Studio
- Server Type = Database Engine
- Server Name = IISQLSRV
- Authentication = SQL Server Authentication
  - Login = your UW email address (not CSE email)
  - Password = seattle

Change your password !!

Then play with IMDB, start working on PROJ1