Introduction to Database Systems
CSE 444

Lecture 1: Introduction
The Staff

- **Instructors:**
  - Wolfgang Gatterbauer (CSE 438)
  - Alexandra Meliou (CSE 444)
    - Email: cse444-instructors@cs
    - Office Hours: Wed 2pm – 3:50pm

- **TAs:**
  - Grad: Rita Sodt (rsodt@cs)
    - Office Hours: Thu 10:30am – 12pm CSE 216
  - Ugrad: Liem Dinh (liemdinh@cs)
    - Office Hours: Tue 1:30pm – 3pm CSE 220
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

- **Message board**
  - Great place to ask assignment-related questions
Textbook

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

- Other Texts:
  - Database Management Systems, Ramakrishnan, Gehrke
  - Foundations of Databases, Abiteboul, Hull, Vianu
  - Fundamentals of Database Systems, Elmasri, Navathe
  - Data on the Web, Abiteboul, Buneman, Suciu
Course Format

- Lectures: MWF, 12:30pm – 1:20pm
  - EEB 045

- Sections: Thu, 8:30am – 9:20am, 9:30am – 10:20am
  - EEB 025

- 4 Mini-Projects
- 3 Homework assignments

- Midterm and final exams
Grading

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

- SQL
- SQL in Java
- Database tuning
- Parallel processing: MapReduce

Check course website for due dates
Three Homework Assignments

- Conceptual Design
- Transactions
- Query execution and optimization

Check course website for due dates
Exams

- Midterm: Wednesday, February 9, in class
- Final: Thursday, March 17, 8:30-10:20am, in class
Outline of Today’s Lecture

- Overview of a DBMS
- A DBMS through an example
- Course content
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?**
  - 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, …
Market Shares

- From 2006 Gartner report:
  - Oracle: 47% market with $7.1BN in sales
  - IBM: 21% market with $3.2BN in sales
  - Microsoft: 17% market with $2.6BN in sales

- From 2008 Gartner study:

<table>
<thead>
<tr>
<th>Database</th>
<th>Currently Deployed</th>
<th>Plan to Deploy Next Year</th>
<th>Plan to Deploy but Not in the Next Year</th>
<th>No Plans to Deploy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>70%</td>
<td>8%</td>
<td>9%</td>
<td>14%</td>
</tr>
<tr>
<td>SQL Server</td>
<td>68%</td>
<td>11%</td>
<td>6%</td>
<td>15%</td>
</tr>
<tr>
<td>MySQL</td>
<td>50%</td>
<td>13%</td>
<td>10%</td>
<td>20%</td>
</tr>
<tr>
<td>DB2</td>
<td>39%</td>
<td>12%</td>
<td>11%</td>
<td>38%</td>
</tr>
<tr>
<td>Informix</td>
<td>18%</td>
<td>13%</td>
<td>14%</td>
<td>55%</td>
</tr>
<tr>
<td>Sybase ASE</td>
<td>15%</td>
<td>14%</td>
<td>14%</td>
<td>57%</td>
</tr>
<tr>
<td>Sybase IQ</td>
<td>14%</td>
<td>14%</td>
<td>13%</td>
<td>59%</td>
</tr>
<tr>
<td>Teradata</td>
<td>11%</td>
<td>15%</td>
<td>13%</td>
<td>61%</td>
</tr>
</tbody>
</table>
An Example

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

- Entities:
  Actors (1.8M), Movies (1.5M), Directors, …

- Relationships:
  who played where, who directed what, …
Required Data Management Functionality

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

- Expensive to implement all these features inside the application
- DBMS provides these features (and more)
- DBMS simplifies application development
### Back to Example: IMDB database

#### Actor

<table>
<thead>
<tr>
<th>id</th>
<th>fName</th>
<th>lName</th>
<th>gender</th>
</tr>
</thead>
</table>

#### Directors

<table>
<thead>
<tr>
<th>id</th>
<th>fName</th>
<th>lName</th>
</tr>
</thead>
</table>

#### Movie

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>year</th>
<th>rank</th>
</tr>
</thead>
</table>

#### Genre

<table>
<thead>
<tr>
<th>mid</th>
<th>genre</th>
</tr>
</thead>
</table>

#### Movie_Directors

<table>
<thead>
<tr>
<th>did</th>
<th>mid</th>
</tr>
</thead>
</table>

#### Casts

<table>
<thead>
<tr>
<th>pid</th>
<th>mid</th>
<th>role</th>
</tr>
</thead>
</table>
## Tables

### Actor:

<table>
<thead>
<tr>
<th>id</th>
<th>fName</th>
<th>lName</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>429073</td>
<td>Tom</td>
<td>Hanks</td>
<td>M</td>
</tr>
<tr>
<td>146871</td>
<td>Amy</td>
<td>Hanks</td>
<td>F</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<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>561300</td>
<td>Toy Story</td>
<td>1995</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Cast:

<table>
<thead>
<tr>
<th>pid</th>
<th>mid</th>
<th>role</th>
</tr>
</thead>
<tbody>
<tr>
<td>429073</td>
<td>561300</td>
<td>Woody</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
SELECT *  
FROM Actor
SELECT count(*)
FROM Actor

This is an aggregate query
SQL

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

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

This query has selections and joins

1.8M actors, 11M casts, 1.5M movies;
How long do we expect it to take?
How Can We Evaluate the Query?

<table>
<thead>
<tr>
<th>Actor:</th>
<th>Cast:</th>
<th>Movie:</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>fName</td>
<td>lName</td>
</tr>
<tr>
<td>...</td>
<td>Hanks</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Plan 1: .... [ in class ]

Plan 2: .... [ in class ]
Evaluating Tom Hanks

\( \sigma_{\text{Name}='Hanks'} \)

\( \sigma_{\text{year}=1995} \)

\( \sigma_{\text{Name}='Hanks'} \)

\( \sigma_{\text{year}=1995} \)
What an RDBMS Does Well (1/2)

- Indexes: on Actor.lName, on Movie.year
- Multiple implementations of joins
- Query optimization
  - Access path selection
  - Join order
  - Join implementation
- Statistics!
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
Client/Server Architecture

- There is a single server that stores the database (called DBMS or RDBMS):
  - Usually a beefy system, e.g. IISQLSRV
  - 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
What This Course Contains

- SQL
- Conceptual Design
- Transactions
- Database tuning and internals (very little)
- Query Optimization
- Distributed databases: a taste of MapReduce
- More stuff depending on time