Staff

• **Instructor: Hal Perkins**
  – CSE 548, perkins@cs.washington.edu
  Office hours: labs tba, office drop-ins and appointments welcome

• **TA: David Broderick**
  – dbroder@cs.washington.edu
  – Office hours: labs tba
Communications

- **Web page**: http://www.cs.washington.edu/444
  - Lectures, homework, projects will be available there

- **Discussion list**
  - See the web page
  - Discussions about the course, databases, etc. Stay in touch outside class

- **Mailing list**
  - Mostly announcements, intent is fairly low traffic
  - You are already subscribed
Textbook

Main textbook, available at the bookstore:

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

Most important: COME TO CLASS ! ASK QUESTIONS !
Other Texts

Available at the Engineering Library
(not on reserve except for textbook, Ramakrishnan):
• *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, 10:50-11:50 am
• Quiz sections: Th 9:40-10:40, 10:50-11:50
  – EEB 042 (changed from original assigned rooms)

• 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
   (we may alter this due to summer schedule)

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: Monday, July 20 or 27, in class (tentative, will try to pin this down sooner rather than later)

• Final: Friday, August 21, in class (no separate finals week in summer)
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
• Our interest is mostly in “structured” 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 2006 Gartner report:

- IBM: 21% market with $3.2BN in sales
- Oracle: 47% market with $7.1BN in sales
- Microsoft: 17% market with $2.6BN 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>
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, 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

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

**Actor:**

<table>
<thead>
<tr>
<th>id</th>
<th>fName</th>
<th>lName</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td>Hanks</td>
<td></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>...</td>
<td></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>...</td>
<td></td>
<td>1995</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Plan 1: . . . [ in class ]

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

σ_{lName='Hanks'}

σ_{year=1995}

σ_{lName='Hanks'}

σ_{year=1995}

Actor  Cast  Movie  Actor  Cast  Movie
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 August
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 July
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, C#, or C++ program
- Clients “talk” to server using JDBC protocol
What This Course Contains

• SQL
• Conceptual Design
• Transactions
• Database tuning and internals (very little)
• Distributed databases: a taste of MapReduce
• More data management if we have time
  – 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