Introduction to Database Systems
CSE 444

Lecture #1
March 31, 2008

Staff

• Instructor: Hal Perkins
  – CSE 548, perkins at cs
  Office hours: Mon 4-4:30+, Wed 4:30-5:00+, CSE006 lab + dropins + appointments

• TAs:
  – Huei-hun Elizabeth Tseng, lachesis at cs
  – Zackary Allred, jaerys at cs
  – Office hours tbd

Communications

• Web site: www.cs.washington.edu/444
  – Lectures available here (usually the morning before class)
  – Homework posted here (HW0 & HW1 are posted now)
  – The project description is also here (Project phases 0 and 1 are posted!)

• Discussion board
  – Will be linked from web site
  – Please watch, contribute

• Mailing list
  – Everyone automatically subscribed
  – Mainly for announcements from course staff, etc.

Textbook(s)

Main textbook, available at the bookstore:

  – Most chapters are good. Some are not as great (functional dependencies).

• COME TO CLASS ! ASK QUESTIONS ! READ SLIDES !
Other Texts
Available at the Engineering Library (not on reserve):

- *Database Management Systems*, Ramakrishnan
- *XQuery*, Walmsley
- *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. Homeworks 0 & 1, Project phases 0 & 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/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, …

RDBMS Market Shares

In 2006, www.gartner.com

• Oracle: 47% market share, $7.2BN in sales
• IBM: 21% market share with $3.2BN 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, …
Tables

<table>
<thead>
<tr>
<th>Directors:</th>
<th>Movie_Directors:</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>fName</td>
</tr>
<tr>
<td>15901</td>
<td>Francis Ford</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Movies:

<table>
<thead>
<tr>
<th>mid</th>
<th>Title</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>130128</td>
<td>The Godfather</td>
<td>1972</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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

Possible Organizations

- Files
- Spreadsheets
- DBMS

1. Create/store Large Datasets

- Files \[\text{Yes, but…}\]
- Spreadsheets \[\text{Not really…}\]
- DBMS \[\text{Yes}\]
2. Search/Query/Update

- Files
  - Simple queries (grep);
  - Updates are difficult
- Spreadsheets
  - Simple queries;
  - Simple updates
- DBMS
  - All

Updates: generally OK

3. Change the Structure

Add Address to each Actor

- Files
  - Very hard
- Spreadsheets
  - Yes
- DBMS
  - Yes

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?

A logical action consists of multiple updates

5. Recover from crashes

- 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);
```

CRASH!

What is the problem?
6. Security

- Files
  - File-level access control

- Spreadsheets
  - Same [?]

- DBMS
  - Table/attribute-level access control

Enters a DMBS

"Two tier system" or "client-server"

Data Independence

<table>
<thead>
<tr>
<th>Directors:</th>
<th>Movie_Directors:</th>
<th>Movies:</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>fName</td>
<td>id</td>
</tr>
<tr>
<td>15901</td>
<td>Coppola</td>
<td>mid</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>15901</td>
</tr>
<tr>
<td></td>
<td></td>
<td>130128</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>mid</td>
<td>Title</td>
<td>Year</td>
</tr>
<tr>
<td>130128</td>
<td>The Godfather</td>
<td>1972</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

SQL DML
SQL DDL
Transactions
ACID
Grant, Revoke, Roles
Course Outline - TENTATIVE !!

1. 3/31: SQL
2. 4/7: Views, Constraints, SQL in C#
3. 4/14: Database Design: E/R, NF
4. 4/21: XML/XPath/XQuery
5. 4/28: Midterm, security
6. 5/5: Transactions, recovery, concurrency
7. 5/19: Database storage, indexes, query execution
8. 5/28: Physical operators, optimization
   • Calendar on web site – updated as we go

Grading (TENTATIVE)

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

Late days: Up to 4 total per quarter, at most 2 on any particular assignment/project phase. Otherwise no late assignments accepted

Reading Assignment

• Reading assignment for Fri, Sept.28

• This is a one-time assignment, no grading, BUT very instructive and lots of fun reading

Homework

• Homework 0:
  – Due this Friday! (Don’t panic – page with your name/picture/etc.)
• Homework 1:
  – SQL Queries
  – Due Friday, April 11
  – It is posted already!
• Homework 2:
  – Conceptual design: E/R diagrams, Normal Forms
  – Due Friday, April 25
• Homework 3:
  – XML/Xquery
  – Due Friday, May 9
• Homework 4:
  – Transactions: concurrency control and recovery
  – Due Friday, May 23
The Project: Boutique Online Store

- Phase 0:
  - Partner details; due middle of next week
- Phase 1:
  - Design a Database Schema, Build Related Data Logic
    - Due Friday, April 18
- Phase 2:
  - Import data, Web Inventory Data Logic
    - Due Friday, May 2
- Phase 3:
  - Checkout Logic
    - Due Friday, May 16
- Phase 4:
  - Database Tuning
    - Due Friday, May 30

Project

SQL Server, C#, ASP.NET
- Supported
- Will provide starter code in C#, ASP.NET
- The import data is in SQL/XML on SQL Server

Alternative technologies: MySQL, postgres, PHPs
- Technically possible
- Not support or encouraged. Talk to instructor if you think you have a compelling reason for even considering this.
  - Religious commitment to LAMP is not a compelling reason

Accessing SQL Server – Today!

SQL Server Management Studio
- Server Type = Database Engine
- Server Name = IISQLSRV
- Authentication = SQL Server Authentication
  - Login = your UW CSE login id
  - Password = 2008#cse444
  - Change your password on first login (must be “secure”)
- Details on the course web

Then play with IMDB, start thinking about HW1

Until Next Time…

- Homework 0
- Log on to SQL server – let us know immediately if you have problems
- Find a partner for the project (all quarter)
- Look at homework 1
- Start reading about SQL online and in the book