Introduction to Database Systems
CSE 444

Lecture #1

March 29, 2010

Staff

- Instructor: Dan Suciu
  - CSE 662, suciu@cs.washington.edu
    Office hours: Mondays 1:30-2:30

- Grad TA: Jessica Leung joyleung@cs.washington.edu

- UGrad TA: Daniel Swisher, Derek Cheng

Communications

- Web page: http://www.cs.washington.edu/444/
  - Lectures will be available here
  - The project description will be here
  - Homework will be posted 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 important: COME TO CLASS & SECTIONS !

Other Texts

Available at the Engineering Library (<u>not</u> on reserve):

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

Course Format

- Lectures MWF, 1:30-2:20, MGH 241

- Quiz sections:
  - Th 8:30-9:20, JHN 026 - Jessica
  - Th 9:30-10:20, JHN 026 - Jessica
  - Th 9:30-10:20, MEB 235 – Daniel/Derek/Dan (alternate ?)

- 4 Mini-projects

- 4 homework assignments

- Midterm, final

Grading

- Homework 30%
- Project   30%
- Midterm   15%
- Final   25%

4 Mini Project

1. SQL (already posted)
2. SQL in Java
3. Database tuning
4. Map/reduce

Due: Wednesday's every other week by 11:59pm

4 Homework Assignments

1. Conceptual Design

2. Transactions

3. Query execution and optimization

4. XML

Due: Wednesday's every other week in class: 12:30pm

Exams

Midterm:

- Friday, April 30, 12:30-1:20 (in class: MGH241)

Final:

- Thursday, June 10, 8:30-10:20 (MGH241)

Rest of Today's Lecture

Overview of DBMS

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, …

SQL for Nerds, Greenspun, http://philip.greenspun.com/sql/ (Chap 1,2)

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, …

Tables

Actor:

id	fName	lName	gender
195428	Tom	Hanks	M
645947	Amy	Hanks	F
. . .			

Casts:

pid	mid
195428	337166
. . .	

Movie:

id	Name	year
337166	Toy Story	1995
. . .	. . .	. ..

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*

SQL

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*

817k actors, 3.5M casts, 380k movies;

How can it be so fast ?

Optimization and Query Execution

- Indexes: on Actor.lName, on Movie.year

- Query optimization

  – Access path selection

  – Join order

- Statistics

- Multiple implementations of joins

Recovery

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

$$X = Read(Account\_1);$$
$$X.amount = X.amount - 100;$$
$$Write(Account\_1, X);$$

$$Y = Read(Account\_2);$$
$$Y.amount = Y.amount + 100;$$
$$Write(Account\_2, Y);$$

Recovery

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

X = Read(Account_1);
X.amount = X.amount - 100;
Write(Account_1, X);

Y = Read(Account_2);
Y.amount = Y.amount + 100;
Write(Account_2, Y);

CRASH !

What is the problem ?

Concurrency Control

- ## How to overdraft your account:

User 1

```
X = Read(Account);
if (X.amount > 100)
   { dispense_money( );
     X.amount = X.amount – 100;
   }
else error("Insufficient funds");
```

User 2

```
X = Read(Account);
if (X.amount > 100)
   { dispense_money( );
     X.amount = X.amount – 100;
   }
else error("Insufficient funds");
```

What can go wrong ?

Transactions

- Recovery

- Concurrency control

ACID =
- Atomicity (= recovery)
- Consistency
- Isolation   (= concurrency control)
- Durability

Client/Server Database Architecture

- There is one single *server* that stores the database (called DBMS or RDBMS):
  - Usually a beefed-up system, e.g. IISQLSRV1
  - But can be your own desktop…
  - … or a huge cluster running a parallel dbms
- Many *clients* running apps and connecting to DBMS
  - E.g. Microsoft's Management Studio
  - Or psql (for postgres)
  - More realistically some Java or C++ program
- The client "talks" to the server using JDBC protocol

Data Management

- Data Management is more than databases !

A Data Management QUIZ:
- Alice sends Bob in random order all the numbers 1, 2, 3, …, 1000000000000000000000
- She does not repeat any number
- But she misses _exactly one_
- Help Bob find out which one is missing !

After you solve it, make it a bit harder:
- Alice misses _exactly ten_ numbers

Accessing SQL Server

SQL Server Management Studio
- Server Type = Database Engine
- Server Name = IISQLSRV
- Authentication = SQL Server Authentication
  - Login = your UW email address (*not* the CSE email)
  - Password = [in class]

Change your password !!

Then play with IMDB, start working on project 1