CSC 370

Database Systems: Intro

Course admin

- Instructor: Mike Zastre
  - E-mail: zastre@cs.uvic.ca
  - Please include "CSC 370" in subject line
  - Office: ECS 528
  - Office hours (Fall 2013): Tuesdays 2:30 to 3:30; Thursdays 1:00 to 2:30
- Online materials: via conneX (connex.csc.uvic.ca)
- Official course outline: link is at the conneX site
  - Here is the course-outline link (longish): http://courses.seng.uvic.ca/courses/2013/fall/csc/370

Course admin (continued)

- Four assignments
  - Some will involve theoretical material
  - Some will involve programming
- One midterm (scheduled for Friday, October 18th)
- Required text: "Database Systems: The Complete Book", second edition (Garcia-Molina, Ullman, Widom)
  - Order of material is significantly different from the first edition

Warning

CSC370 ≠ SQL

(There is a life beyond CRUD!)

... Yet SQL is still very important, and we will cover it in some detail.

What is this course about?

- Theory supporting the design of databases
  - Relational algebra
  - Normal forms
  - E/R, UML model
- Database Programming
- Some topics in database-system implementation
- Goal:
  - Provide you with enough theory and practice to better understand the use of databases
  - Ensure you have a chance of grasping what happens as database technology evolves

Databases (once upon a time)

- Back in the day, databases were workhorses for typical corporate/institutional tasks
  - Employee records
  - Bank records
- That said, a company's database administrator (DBA) was usually highly respected
  - Poorly configured databases could seriously impact operations
- Full DBMS implementations required expensive equipment.

Databases (today)

- This topic in computing is now relevant nearly everywhere computers themselves are used!
- Examples:
  - Web searching
  - Data mining
  - E-commerce
  - Scientific and medical databases
  - Programming frameworks such as Ruby on Rails

Databases (some further observations)

- We will see how "database programming" depends upon very limited programming languages
  - SQL is non-Turing complete.
  - We'll see some of the theory behind why this works.
  - Good news: succinct programming of queries
  - Bad news: queries often depend on back-end optimizations for acceptable performance (i.e., we need to learn what makes a "badly formed query" so darn bad)

Important abstraction: Data Model

- **Data model**: notation for describing data or information
  - Note that we may need to shoehorn the real-world data into the chosen model
  - Generally we will try to do as little violence as possible to the real-world data
- Three parts to any data model:
  1. **Structure**
  2. **Operations**
  3. **Constraints**

Databases (some further observations)

- Many people may not notice it...
- ... yet databases are behind almost everything done on the web (i.e., lots of simultaneous users)
  - Google search
  - Amazon queries
  - eBay auctions
  - Blog postings
  - Twitter
- Databases often have unique concurrency-control problems
  - Many activities (transactions) active the database at all times...
  - ... Yet must ensure combined effect of the activities do not produce incorrect results (think of two withdrawals from a bank account).
  - And must sometimes also think about scalability!