Why study databases?

Database Management System (DBMS)

Storage

Attending first lecture of csc343.. meh. | — 😞feeling bored.
Why study databases?

Database Management System (DBMS)

Storage
Why study databases?

- Interesting concepts and techniques.
- Spans computer science, including OS, languages, theory, AI, multimedia, logic.
- Databases have become increasingly important
  - shift from a focus on computation to information
  - data increases in volume and diversity.
- Jobs: In demand and well paid.
- Research: Many open problems.
Our first hour or so..

- Some key concepts
- Examples to motivate the course
- Admin info
Databases and DBMSs

- Databases are everywhere, often behind the scenes.
- DBMS (Database Management System): “A powerful tool for creating and managing large amounts of data efficiently and allowing it to persist over long periods of time, safely.” [Ullman and Widom, FCDB]
- Database: a collection of data managed by a DBMS.
Data models

- Every DBMS is based on some data model: a notation for describing data, including
  - the structure of the data
  - constraints on the content of the data
  - operations on the data

- Some specific data models:
  - network & hierarchical data models — of historic interest
  - relational data model
  - semistructured data model
The relational data model

- Main concept is a “relation.”
  Based on the concept of relations in *math*.
- Can think of as *tables* of rows and columns.

### Teams

<table>
<thead>
<tr>
<th>Name</th>
<th>Home Field</th>
<th>Coach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rangers</td>
<td>Runnymede CI</td>
<td>Tarvo Sinervo</td>
</tr>
<tr>
<td>Ducks</td>
<td>Humber Public</td>
<td>Maeve Mahar</td>
</tr>
<tr>
<td>Choppers</td>
<td>High Park</td>
<td>Tom Cole</td>
</tr>
</tbody>
</table>

### Games

<table>
<thead>
<tr>
<th>Home team</th>
<th>Away team</th>
<th>Home goals</th>
<th>Away goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rangers</td>
<td>Ducks</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Ducks</td>
<td>Choppers</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Rangers</td>
<td>Choppers</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Choppers</td>
<td>Ducks</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>
Example ...

- A dataset scraped from Twitter

Who has the most followers?

Who is in Toronto, mentions DJ in their bio, and has more than 1000 followers?
Example ...

- A dataset scraped from **Twitter**
- Defining a **schema** that expresses its structure
Example ...

- A dataset scraped from Twitter
- Defining a schema that expresses its structure
- Creating an instance that contains the data
- Writing some queries on the data…

```sql
>> select id, name, location
from profile
where location = 'Toronto'
and bio like '%DJ%';
```

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td>vorchopolis</td>
<td>Jorge Aranda</td>
<td>Toronto</td>
</tr>
<tr>
<td>lance</td>
<td>lance underscore</td>
<td>Toronto</td>
</tr>
<tr>
<td>zuzelyp</td>
<td>Zuzel Vera</td>
<td>Toronto</td>
</tr>
<tr>
<td>karenreid</td>
<td>karenreid</td>
<td>Toronto</td>
</tr>
<tr>
<td>torontoist</td>
<td>torontoist</td>
<td>Toronto</td>
</tr>
<tr>
<td>dianelynhorton</td>
<td>dianelynhorton</td>
<td>Toronto</td>
</tr>
</tbody>
</table>
What a DBMS provides..

- Ability to specify the logical structure of the data
  - explicitly
  - and have it enforced

- Ability to query or modify the data.

- Good performance under heavy loads
  (huge data, many queries).

- Durability of the data.

- Concurrent access by multiple users/processes.
Overall architecture of a DBMS

- The DBMS sits between the data and the users or between the data and an application program.
Overall architecture of a DBMS

- No like, seriously..?
Overall architecture of a DBMS

- The DBMS sits between the data and the users or between the data and an application program.
- Within the DBMS are layers of software for:
  - parsing “queries”
  - implementing the fundamental operations
  - optimizing queries
  - maintaining indices on the data
  - accessing the files that store the data and indices
  - management of buffers
  - management of disk space
A “semi-structured” example ...

- An **xml dataset** scraped from imdb.com
- No schema required, no instance made
- We can immediately write queries on the data
- A much *looser* approach
A “semi-structured” example ...

<books search-terms="database+design">
  <book>
    <title>Database Design for Mere Mortals</title>
    <author>Michael J. Hernandez</author>
    <date>13/03/2003</date>
  </book>
  <book id="B2">
    <title>Beginning Database Design</title>
    <subtitle>From Novice to Professional</subtitle>
    <author>Clare Churcher</author>
  </book>
</books>
What this course is about

- **csc443** is about **implementation** of the DBMS itself
- **csc343** is about **using** DBMSs:
  - defining schemas and instances
  - writing queries
  - connecting to code written in a general-purpose language (e.g. Java!)
  - rigorous underlying principles
CSC343 - Administrative Info!
Admin Stuff..

Important: Read the course syllabus

- **Contact:**
  - website and Piazza: required reading
  - your questions: to Piazza please
  - *personal* matters: email or visit me in O.H.

- **Office hours:**
  - Tuesdays 3-5pm
  - Room: BA 3219
Prerequisites

- For **A&S** students, the prerequisites are:
  1. CSC165H1/CSC240H1/(MAT135H1, MAT136H1)/MAT135Y1/MAT137Y1/MAT157Y1; (2) CSC207H1
- Prerequisite for Engineering students only: ECE345H1/CSC190H1/CSC192H1
- Email me immediately if you don’t have the prerequisites (nosayba@cs.toronto.edu). Include your unofficial ROSI transcript.
- **Engineering** students, contact me if you need permission.
Active lectures (kind of..)

- Goal: get your gears turning in class!

- Activities like:
  - team problem solving, reviewing other students’ solutions, and short quizzes.

- Weekly “lecture prep activities” will get you ready.
  - exercises, reading, watching videos

- All three hours will be here, with me.
  - Relax: some weeks will have tutorials delivered by TAs ;-)

[Image]
Benefits of active learning

- Exercise your knowledge and skills in class, with support.
- We’ll know where the difficulties are.
- Get more from when I’m lecturing.

What it requires

- Doing the lecture prep.
- Being active in class, including working with others and looking at each other’s solutions to problems.
- A positive, encouraging environment.
# Course Marking Scheme

<table>
<thead>
<tr>
<th>Work</th>
<th>Weight</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 assignments</td>
<td>30%</td>
<td>10% each</td>
</tr>
<tr>
<td>weekly lecture prep</td>
<td>7%</td>
<td>due Sundays 11pm</td>
</tr>
<tr>
<td>weekly in-class exercise</td>
<td>3%</td>
<td>due in lectures</td>
</tr>
<tr>
<td>midterm</td>
<td>15%</td>
<td>Oct 27</td>
</tr>
<tr>
<td>Final exam</td>
<td>45%</td>
<td>You must get &gt;= 40% in exam mark to pass the course</td>
</tr>
</tbody>
</table>
Recommended Resources

- Ullman and Widom,
- Jennifer Widom’s online mini-courses from Stanford.
Assignment Policies

- You may work with a partner on assignments.
- Can be from any section on StGeorge campus.
- Can change partners between assignments.
- You may **not** dissolve a partnership without permission.
- Assignments must be submitted via **MarkUs**.
- Your code must run on our lab computers ("cdf").
- **Late policy:**
  - You have **6 grace tokens** that can be used for **2-hour** extension each.
  - No submission allowed after all tokens are exhausted.
Your To-do list

- Anyone new to the **cdf** labs:
  - Find out your account on our cdf machines. See the course website for details.
  - Try logging in.

- Read the course syllabus.
- Bookmark the course website.
- Do the class prep due Sunday night.