CS 4432
Database Systems II

Lecture 1: Introduction

Professor Elke A. Rundensteiner
Today: Tim Sutherland
Recommended Background

• Beginning database design knowledge as gained in say CS3431 (in particular knowledge of the relational data model and SQL) and some knowledge of software engineering (we will be using Java), such as CS3733.
Staff

- **INSTRUCTOR:** Professor Elke A. Rundensteiner
- **Office Hours:** Mondays noon-1pm. Thursdays 9:15-10:15pm

- **TEACHING ASSISTANTS**
  - Tim Sutherland
    - tims@cs.wpi.edu
  - Luping Ding
    - lisading@cs.wpi.edu
  - Yali Zhu
    - yaliz@cs.wpu.edu

See course web Page (http://my.wpi.edu) for office location & hours.
Protocol for Communication

1. Post to the discussion board!

2. Come to Office Hours!!
   - We have 8 office hours spread throughout the week, on Mondays-Thursdays

3. E-Mail cs4432-staff@cs.wpi.edu
   - Expect *at least* a 24 hour response time

4. Schedule an office hour with the course staff.
Details

- **LECTURES:** Monday, Tuesday, Thursday, Friday 2-3pm FL320
- **TEXTBOOK:** Garcia-Molina, Ullman, Widom;
  either "DATABASE SYSTEM IMPLEMENTATION"
  or  "DATABASE SYSTEMS, THE COMPLETE BOOK"

- **ASSIGNMENTS:** 4-5 Written Homework Assignments. 3 Group Projects.
- **GRADING:**
  - **Midterm:** 20%
  - **Final Exam:** 30%
  - **Homework Assignments:** 20%.
  - **Projects:** 30%.
  - **Class participation:** +/-

- **WEB SITE:** [http://my.wpi.edu](http://my.wpi.edu)
- **Homework Submission:** WPI’s Turnin Program (NO EMAIL)
- **Please check it daily for last minute announcements.**
DB Material at WPI

- **A,C terms**: CS 3431
  - D term (alternate)
  - CS 4432
  - you are here

  - CS 542 Grad. DB
    - DON'T TAKE!
  - CS 561 Advanced DB
    - Spring
  - MQP Selected
    - Any time
  - ISP Independent DB Project
    - Any time
  - DSRG DB Research at WPI
    - year round

Notes 1
Isn’t Implementing a Database System Simple?

Relations → Statements → Results
Introducing the MEGATRON 3000
Database Management System

- The latest from Megatron Labs
- Incorporates latest relational technology
- UNIX compatible
Megatron 3000 Implementation Details

- Relations stored in files (ASCII)
e.g., relation Students is in /usr/db/Students

<table>
<thead>
<tr>
<th>Students</th>
<th>Depts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith # 123</td>
<td>CS # Fuller Labs</td>
</tr>
<tr>
<td>Jones # 522</td>
<td>EE # Atwater Kent</td>
</tr>
<tr>
<td></td>
<td>PH # Olin Hall</td>
</tr>
<tr>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
</tr>
</tbody>
</table>
Megatron 3000 Implementation Details

- Directory file (ASCII) in /usr/db/schema

\[
\begin{align*}
\text{Students} & \# \text{name} \# \text{STR} \# \text{id} \# \text{INT} \# \text{dept} \# \text{STR} \\
\text{Depts} & \# \text{name} \# \text{STR} \# \text{office} \# \text{STR} \\
 & \vdots
\end{align*}
\]
Megatron 3000
Sample Sessions

% MEGATRON3000
  Welcome to MEGATRON 3000!
&
  :
  :
& quit
%
Megatron 3000
Sample Sessions

& select *
from Students #

<table>
<thead>
<tr>
<th>Relation Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>SMITH</td>
</tr>
<tr>
<td>JONES</td>
</tr>
</tbody>
</table>

&
Megatron 3000
Sample Sessions

& select Students.name, Depts.office
from Students, Depts
where Students.dept = Depts.name
Students.id > 300

Smith # 123 # CS # CS # Fuller Labs
Smith # 123 # CS # EE # Atwater Kent
Smith # 123 # CS # PH # Olin Hall
Jones # 522 # EE # CS # Fuller Labs
Jones # 522 # EE # EE # Atwater Kent
Jones # 522 # EE # PH # Olin Hall
Megatron 3000
Sample Sessions

& select *
  from Students | LPR #
&

Result sent to LPR (printer).
Megatron 3000
Sample Sessions

& select *
  from R
  where R.A < 100  |  T  #
&

New relation T created.
Megatron 3000

- To execute "select * from R where condition":
  1. Read dictionary to get R attributes
  2. Read R file, for each line:
     (a) Check condition
     (b) If OK, display
Megatron 3000

• To execute "select * from R
  where condition | T":

  (1) Process select as before
  (2) Write results to new file T
  (3) Append new line to dictionary
Megatron 3000

- To execute “select A,B from R,S where condition”:
  1. Read dictionary to get Students,Depts attributes
  2. Read Students file, for each line:
     a. Read Depts file, for each line:
        i. Create join tuple
        ii. Check condition
        iii. Display if OK
What’s wrong with the Megatron 3000 DBMS?

GROUP EXERCISE (15 mins):

On your Syllabus is a number from 1-10:

1. Find all the members of your group
2. Find as many problems with this design as possible.
3. With remaining time, come up with a potential solution to each problem.
4. Have one person come up to board to write down problems.
What’s wrong with the Megatron 3000 DBMS?

• Tuple layout on disk
  e.g.,
  - Change string from ‘Cat’ to ‘Cats’ and we have to rewrite file
    - ASCII storage is expensive
    - Deletions are expensive
What’s wrong with the Megatron 3000 DBMS?

- Search expensive; no indexes
  - Cannot find tuple with given key quickly
  - Always have to read full relation
What’s wrong with the Megatron 3000 DBMS?

• Brute force query processing
  e.g.,
  
  ```sql
  select *
  from R, S
  where R.A = S.A and S.B > 1000
  ```
  - Do select first?
  - More efficient join?
What’s wrong with the Megatron 3000 DBMS?

• No buffer manager
  e.g., Need caching
What’s wrong with the Megatron 3000 DBMS?

• No concurrency control
What’s wrong with the Megatron 3000 DBMS?

- No reliability
  
  e.g.,
  - Can lose data
  - Can leave operations half done
What’s wrong with the Megatron 3000 DBMS?

• No security
  e.g.,
  - File system insecure
  - File system security is coarse
What’s wrong with the Megatron 3000 DBMS?

• No application program interface (API)
  e.g., How can a payroll program get at the data?
What’s wrong with the Megatron 3000 DBMS?

• Cannot interact with other DBMSs.
What’s wrong with the Megatron 3000 DBMS?

- Poor dictionary facilities
What’s wrong with the Megatron 3000 DBMS?

• No GUI
System Structure

- Strategy Selector
- Query Parser
- User
- User Transaction
- Transaction Manager
- Concurrency Control
- Buffer Manager
- Recovery Manager
- Lock Table
- File Manager
- M.M. Buffer
- Log

Statistical Data
- Indexes
- User Data
- System Data
Course Overview

• File & System Structure
  Records in blocks, dictionary, buffer management,…

• Indexing & Hashing
  B-Trees, hashing,…

• Query Processing
  Query costs, join strategies,…

• Crash Recovery
  Failures, stable storage,…
Course Overview

- Concurrency Control
  Correctness, locks,…

- Transaction Processing
  Logs, deadlocks,…

- Security & Integrity
  Authorization, encryption,…

- Distributed Databases/Streaming Data
  Interoperation, distributed recovery,…
Next time:

- Hardware
- Read chapters 1 and 2