CMPS 277:
officially
Relational Databases
but this quarter
Database Implementation

Notes 01: Introduction

Arthur Keller
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

First sign non-disclosure agreement
Megatron 3000

Implementation Details

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

<table>
<thead>
<tr>
<th>Name</th>
<th>#</th>
<th>Dept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>123</td>
<td>CS</td>
</tr>
<tr>
<td>Jones</td>
<td>522</td>
<td>EE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Megatron 3000
Implementation Details

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

```
R1  #  A  #  INT  #  B  #  STR  ...
R2  #  C  #  STR  #  A  #  INT  ...
...
```
Megatron 3000

Sample Sessions

% MEGATRON3000

Welcome to MEGATRON 3000!

&

:

:

& quit

%
Megatron 3000
Sample Sessions

& select *
    from R #

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>SMITH</td>
<td>123</td>
<td>CS</td>
</tr>
</tbody>
</table>

&
& select A,B
from R,S
where R.A = S.A and S.C > 100 #

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>CAR</td>
</tr>
<tr>
<td>522</td>
<td>CAT</td>
</tr>
</tbody>
</table>

&
Megatron 3000

Sample Sessions

& select *
  from R | 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 R,S attributes
  2. Read R file, for each line:
     a. Read S file, for each line:
        i. Create join tuple
        ii. Check condition
        iii. Display if OK
What’s wrong with the Megatron 3000 DBMS?
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., `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
What’s wrong with the Megatron 3000 DBMS?

• Lousy salesman!!
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
  Interoperation, distributed recovery, ...
System Structure

1. Strategy Selector
2. Query Parser
3. Transaction Manager
4. Recovery Manager
5. Lock Table
6. File Manager
7. M.M. Buffer
8. Log
9. User Transaction
10. File Manager
11. M.M. Buffer
12. Log
13. Statistical Data
14. Indexes
15. User Data
16. System Data
Some Terms

- Database system
- Transaction processing system
- File access system
- Information retrieval system
Mechanics

http://www.soe.ucsc.edu/classes/cs277/
Coming soon
Prerequisite

• An introductory database course equivalent to CMPS180
• Knowledge of SQL (theory and practice)
• Algorithms and elementary analysis

• If you do not have the prerequisite, you may want to audit the class instead.
Staff

- **INSTRUCTOR:** Arthur Keller
- **Office:** Baskin Engineering 153A
- **Email:** ark@soe.ucsc.edu – a good way to reach me.
- **Office Hours:** Most Tuesdays, Some Thursdays 4:30-5:30pm, often for a few minutes after class, and by appointment.
- I’m coming from Palo Alto, so I may be late.
- **TEACHING ASSISTANT:** none
- **GRADER:** ?
Details

• LECTURES: Tuesday, Thursday 6-7:45pm, SS II 179

• TEXTBOOK: Garcia-Molina, Ullman, Widom: “DATABASE SYSTEMS, THE COMPLETE BOOK” (second half of book, first half was used for CMPS180).

• ASSIGNMENTS: Seven written homework assignments. No programming. Also readings in Textbook.

• GRADING: Homeworks: 21% (3% each), Survey paper: 19%, Midterm: 20%, Final: 40%.

• WEB SITE: All handouts and assignments will be posted on our Web site at http://www.soe.ucsc.edu/classes/cs277/

• NEWSGROUP: ucsd.class.cmps277 is being set up.
## Tentative Syllabus

<table>
<thead>
<tr>
<th>DATE</th>
<th>CHAPTER</th>
<th>TOPIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue Mar 26</td>
<td></td>
<td>Introduction</td>
</tr>
<tr>
<td>Thu Mar 28</td>
<td></td>
<td>Class cancelled</td>
</tr>
<tr>
<td>Tue Apr 2</td>
<td>Ch. 11</td>
<td>Hardware</td>
</tr>
<tr>
<td>Thu Apr 4</td>
<td>Ch. 12</td>
<td>File and System Structure</td>
</tr>
<tr>
<td>Tue Apr 9</td>
<td>Ch. 12</td>
<td>File and System Structure</td>
</tr>
<tr>
<td>Thu Apr 11</td>
<td>Ch. 13</td>
<td>Indexing and Hashing</td>
</tr>
<tr>
<td>Tue Apr 16</td>
<td>Ch. 13</td>
<td>Indexing and Hashing</td>
</tr>
<tr>
<td>Thu Apr 18</td>
<td>Ch. 14</td>
<td>Indexing and Hashing</td>
</tr>
<tr>
<td>Tue Apr 23</td>
<td>Ch. 15</td>
<td>Query Processing</td>
</tr>
<tr>
<td>Thu Apr 25</td>
<td>Ch. 15</td>
<td>Query Processing</td>
</tr>
<tr>
<td>Tue Apr 30</td>
<td>Ch. 16</td>
<td>Query Processing</td>
</tr>
<tr>
<td>Thu May 2</td>
<td>Ch. 17</td>
<td>Crash Recovery</td>
</tr>
<tr>
<td>Tue May 7</td>
<td></td>
<td>Midterm</td>
</tr>
<tr>
<td>Thu May 9</td>
<td>Ch. 17</td>
<td>Crash Recovery</td>
</tr>
<tr>
<td>Tue May 14</td>
<td>Ch. 18</td>
<td>Concurrency Control</td>
</tr>
<tr>
<td>Thu May 16</td>
<td>Ch. 18</td>
<td>Concurrency Control</td>
</tr>
<tr>
<td>Tue May 21</td>
<td>Ch. 18</td>
<td>Concurrency Control</td>
</tr>
<tr>
<td>Thu May 23</td>
<td>Ch. 19</td>
<td>Transaction Processing</td>
</tr>
<tr>
<td>Tue May 28</td>
<td>Ch. 19</td>
<td>Transaction Processing</td>
</tr>
<tr>
<td>Thu May 30</td>
<td>Ch. 20</td>
<td>Information Integration</td>
</tr>
<tr>
<td>TBA</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Wed Jun 5</td>
<td></td>
<td>Final</td>
</tr>
</tbody>
</table>
Read: All Chapters

- Except following optional material:
  - Sections 11.7.4, 11.7.5
  - Sections 14.3.6, 14.3.7, 14.3.8
  - Sections 14.4.2, 14.4.3, 14.4.4
  - Sections 15.7, 15.8, 15.9
  - Sections 16.6, 16.7
  - In Chapters 15, 16: material on duplicate elimination operator, grouping, aggregation operators
  - Section 18.8
  - Sections 19.4, 19.5, 19.6, 19.7
Next time:

- Hardware
- Read chapter 11