CS525: Advanced Database Organization

Notes 1: Introduction

Yousef M. Elmehdwi

Department of Computer Science

Illinois Institute of Technology

yelmehdwi@iit.edu

January 8, 2018

Slides: adapted from a course taught by Hector Garcia-Molina, Stanford

Core Terminology Review

- Data
  - any information worth preserving, most likely in electronic form
- Database
  - a collection of data, organized for access and modification, preserved over a long period.
- Query
  - an operation that extracts specified data from the database.
- Relation
  - an organization of data into a two-dimensional table, where rows (tuples) represent basic entities or facts of some sort, and columns (attributes) represent properties of those entities.
- Schema
  - a description of the structure of the data in a database, often called "metadata"
- Database Management System (DBMS)
  - software that enables easy creation, access, and modification of databases for efficient and effective database management.

Advanced Database Organization?

- =Database Implementation
- =How to implement a database system
- and have fun doing it ;-)

Isn't Implementing a Database System Simple?

- Relation $\Rightarrow$ Statements $\Rightarrow$ Results

Introduction the MEGATRON 3000 Database Management System

- "Imaginary" database System
- The latest from MEGATRON Labs
- Incorporates latest relational technology
- UNIX compatible
- Lightweight & cheap!

Megatron 3000 Implementation Details

- Megatron 3000 uses the file system to store its relations
- Relations stored in files (ASCII)
  e.g., relation `Students` is in `/usr/db/Students`

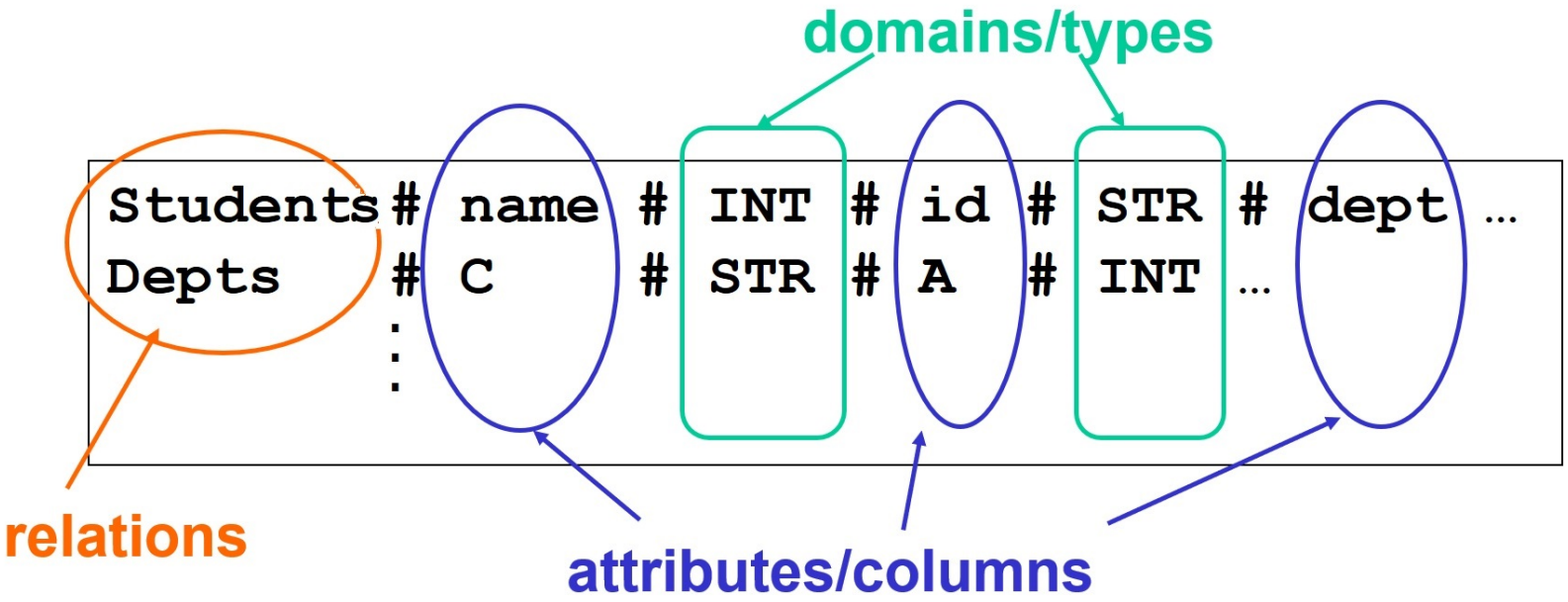
Smith	#	123	#	CS
Jonson	#	522	#	EE
		⋮		

Megatron 3000 Implementation Details

- The database schema is stored in a special file
- Schema file (ASCII) in `/usr/db/schema`

```
Students # name # INT # id # STR # dept ...
Depts    # C    # STR # A  # INT ...
                    ⋮
```

Megatron 3000 Implementation Details



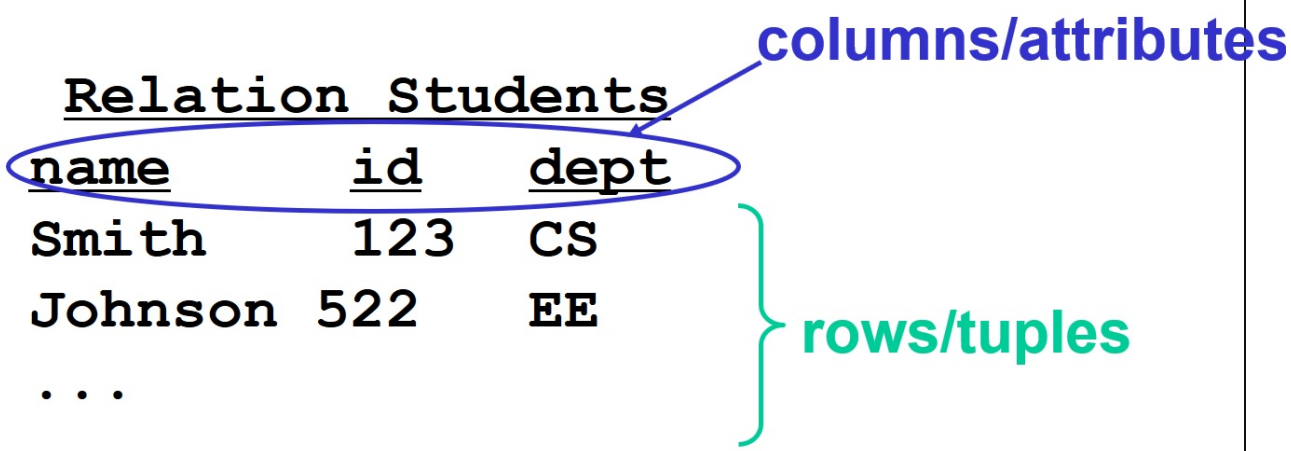
Megatron 3000 Sample Sessions

```
% MEGATRON3000
   Welcome to MEGATRON 3000!
&
   :
   :
& quit
%
```

Megatron 3000 Sample Sessions

```
& select * from Students #

    Relation Students
  name      id    dept
  Smith     123   CS
  Johnson   522   EE
  ...
&
```



Megatron 3000 Sample Sessions

- Execute a query and send the result to printer

```
& select *
  from Students | LPR #
&
```

- Result sent to LPR (printer).

Megatron 3000 Sample Sessions

- Execute a query and store the result in a new file

```
& select *
  from Students
  where id < 100 | LowId #
&
```

- New relation `LowId` created.

Megatron 3000 Query Execution

- To execute

  `SELECT * FROM R WHERE <condition>`

  1. Read schema to get attributes of R
  2. Check validity of condition
  3. Display attributes of R as the header
  4. Read file R; for each line:
     a. Check condition
     b. If TRUE, display

Megatron 3000 Query Execution

- 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 `usr/db/schema`

Megatron 3000 Query Execution

- Consider a more complicated query, one involving a join of two example relations `R`, `S`
- To execute

```
SELECT A,B FROM R,S WHERE <condition>
```

1. Read schema to get `R,S` attributes
2. Read `R` file, for each line `r`:
   a. Read `S` file, for each line `s`:
      1. Create join tuple `r & s`
      2. Check condition
      3. If TRUE, Display `r,s[A,B]`

What's wrong with MEGATRON 3000 DBMS?

- DBMS is not implemented like our "imaginary" MEGATRON 3000
- Described implementation is inadequate for applications involving significant amount of data or multiple users of data
- Next: Partial list of problems follows

What's wrong with Megatron 3000 DBMS?

- Tuple layout on disk is inadequate with no flexibility when the database is modified
- 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 MEGATRON 3000 DBMS?

- Search expensive; no indexes
  - e.g., Cannot find tuple with given key quickly
  - Always have to read full relation

What's wrong with Megatron 3000 DBMS?

- Brute force query processing
- e.g.,

```
SELECT * FROM R,S WHERE R.A = S.A and S.B > 1000
```

  - Much better if use index to select tuples that satisfy condition (Do select using S.B >1000 first)
  - More efficient join (Sort both relations on A and merge)

What's wrong with Megatron 3000 DBMS?

- No buffer manager
  - There is no way for useful data to be buffered in main memory; all data comes off the disk, all the time
  - e.g., Need caching.

What's wrong with Megatron 3000 DBMS?

- No concurrency control
  - Several users can modify a file at the same time with unpredictable results.

What's wrong with Megatron 3000 DBMS?

- No reliability
- e.g., In case of error/crash, say, power failure or leave operations half done
  - Can lose data

What's wrong with Megatron 3000 DBMS?

- No security
- e.g., File system security is coarse
  - Unable to restrict access, say, to some fields of relations

What's wrong with Megatron 3000 DBMS?

- No application program interface (API)
  - e.g., How can a payroll program get at the data?

What's wrong with Megatron 3000 DBMS?

- Cannot interact with other DBMSs.

What's wrong with Megatron 3000 DBMS?

- No GUI

This Course

- Introduce students to better way of building a database management systems.

Reading assignment

- Refresh your memory about basics of the relational model and SQL
  - from your earlier course notes
  - from some textbook
  - Google

Next

Notes 2: Hardware