CMPUT 291
File and Database Management Systems

Davood Rafiei

University of Alberta

drafiei@ualberta.ca

Fall 2019

Class Information

- **Lectures:**
  - A1: Mon, Wed & Fri 1:00pm – 1:50pm in ETLC E1007
  - A2: Tue & Thu 9:30 – 10:50am in ETLC E1013
- **Instructors**:
  - Davood Rafiei (drafiei@ualberta.ca)
- **Office hours:**
  - in ATH 436, time: Wed & Fri 2-3pm
- **TA's:**
  - Names and contacts in the course web page

Labs

- **Location:** CSC 219

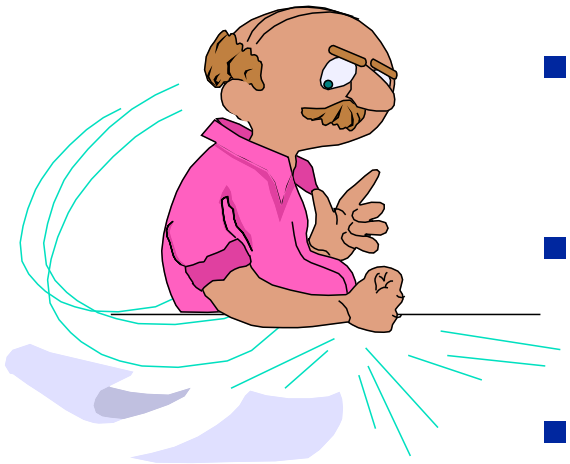
  (CSC B10 and ETLC E2002 for the laptop labs)
- **10 labs:** schedule posted in the course web page

- **Working with:**
  - Unix tools and commands
  - SQLite
  - Berkeley DB
- Access from home
- Assignments and projects are tested on lab machines

Textbook and Slides

- **Recommended Text:**
  - Silberschatz, Korth, Sudarshan: Database Systems Concepts, 7th edition, McGraw Hill, 2019.
- **Alternative Text:**
  - Kifer, Bernstein, Lewis; Database Systems : An Application Oriented Approach , 2nd edition, Addison Wesley, 2005
- **Schedule, slides, readings and forums:**
  - Check the eclass pages at *https://eclass.srv.ualberta.ca/course/view.php?id=53330*

Evaluation, etc.



- **Assignment 1:** 5% (groups of 2-3)
  - Available 2 weeks in advance
- **Assignment 2**: 10% (**individual**)
  - Available 2 weeks in advance
- **Quizzes:** 7% total
  - 2 in the lab (4%), 3 in lectures (3%)
- **Mini-Project 1:** 13% (groups of 2-3)
  - Available 3 weeks in advance
- **Mini-Project 2:** 12% (groups of 2-3)
  - Available 2-3 weeks in advance
- **Term Test:** 23%
- **Final Exam:** 30%

This course

- A balanced mix of theory and practice
  - Theory
    - ✓ query languages with roots in logic
    - ✓ set theory
    - ✓ normal forms
  - Practice
    - ✓ writing queries
    - ✓ building databases
    - ✓ developing database applications
    - ✓ searching files and indexes

What is expected?

- A good understanding of the concepts covered
- Hands on experience
  - Writing queries, developing database applications

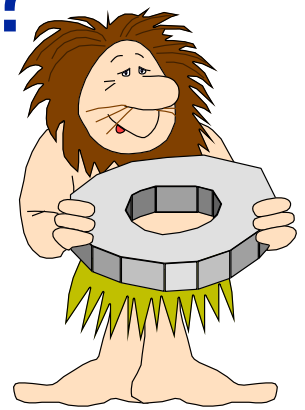
How to succeed?

- For theory
  - Regularly attend the lectures/labs and ask questions
  - Closely follow the course and make use of office hours to resolve problems
- For practice
  - Write as many queries as possible (like a musician, an artist, a programmer, practice is the key!)
  - Attend the labs and make the best use of it
  - Start the assignments and projects early
- Stay on top
  - Topics covered can be overwhelming

Course Outline

- What are they and why are they useful?
  - separate data from programs.
  - factor out common pieces from applications.
- How can I use them?
  - Structuring data: relational data model
  - queries: SQL
- How do they work?
  - Organizing data in files and indexes
  - Query execution
  - Performance tuning

What is a Database?



- A *database* is a very large, integrated collection of data.
- Models real-world enterprise.
  - Entities (e.g. students, courses)
  - Relationships (e.g. Justin Trudeau is taking 291).

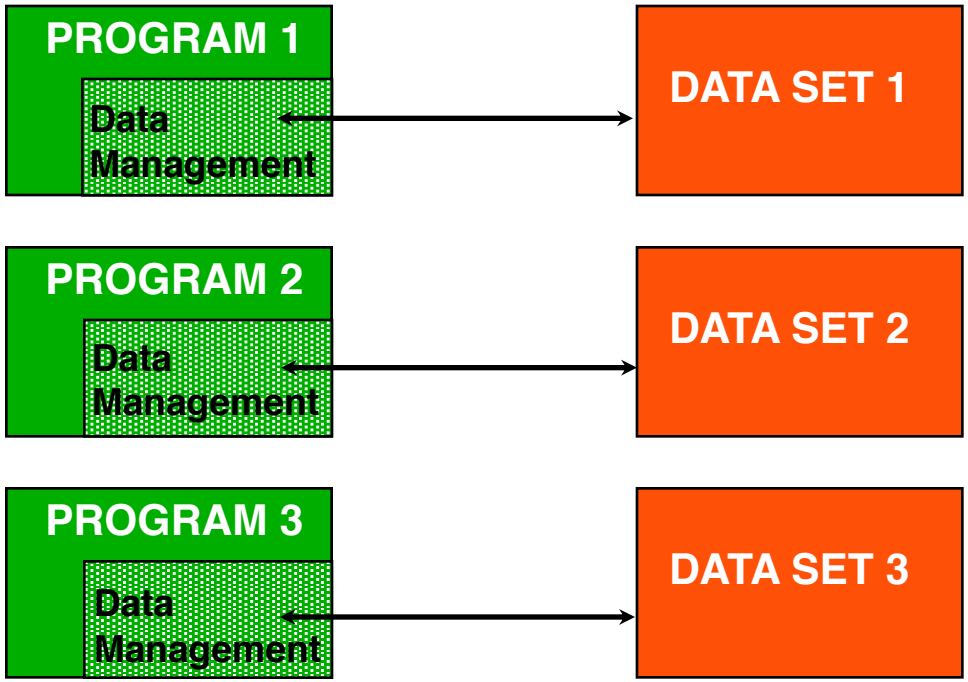
Manufacturing	Product data
University	Student data, courses
Hospital	Patient data, facilities
Bank	Account data

What is a DBMS?

- A *Database Management System (DBMS)* is a software package designed to store and manage databases.

Non-Database Approach

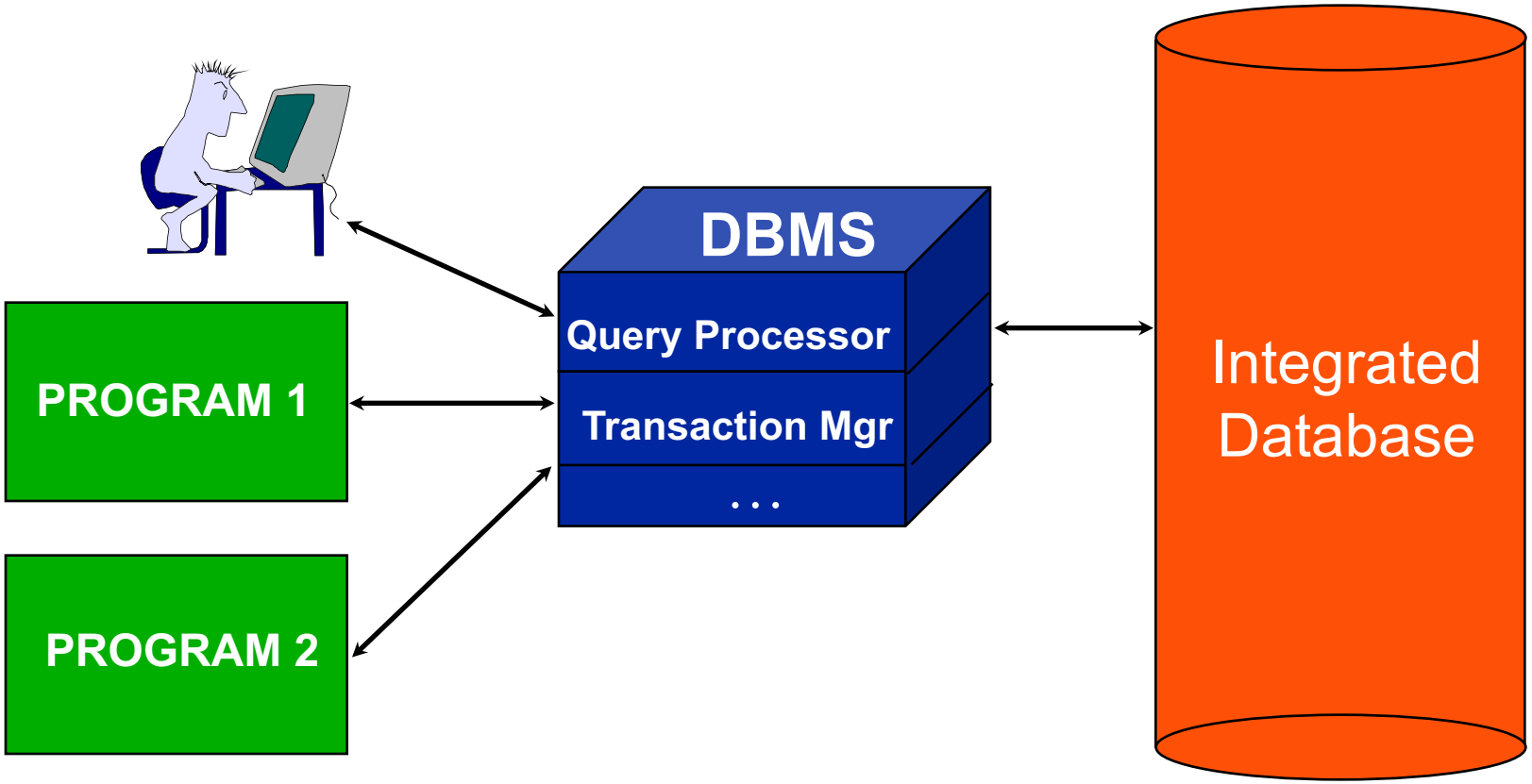
- Common in 60's.
- One data set per program.
- Programmer defines (and implements) storage structures, access methods, etc.



Problems

- With more and more applications we get
  - many files with different structures
  - redundant storage
  - inconsistent copies
  - expensive updates
  - incorrect data
  - data exchange between applications

Database Approach



Basic Idea

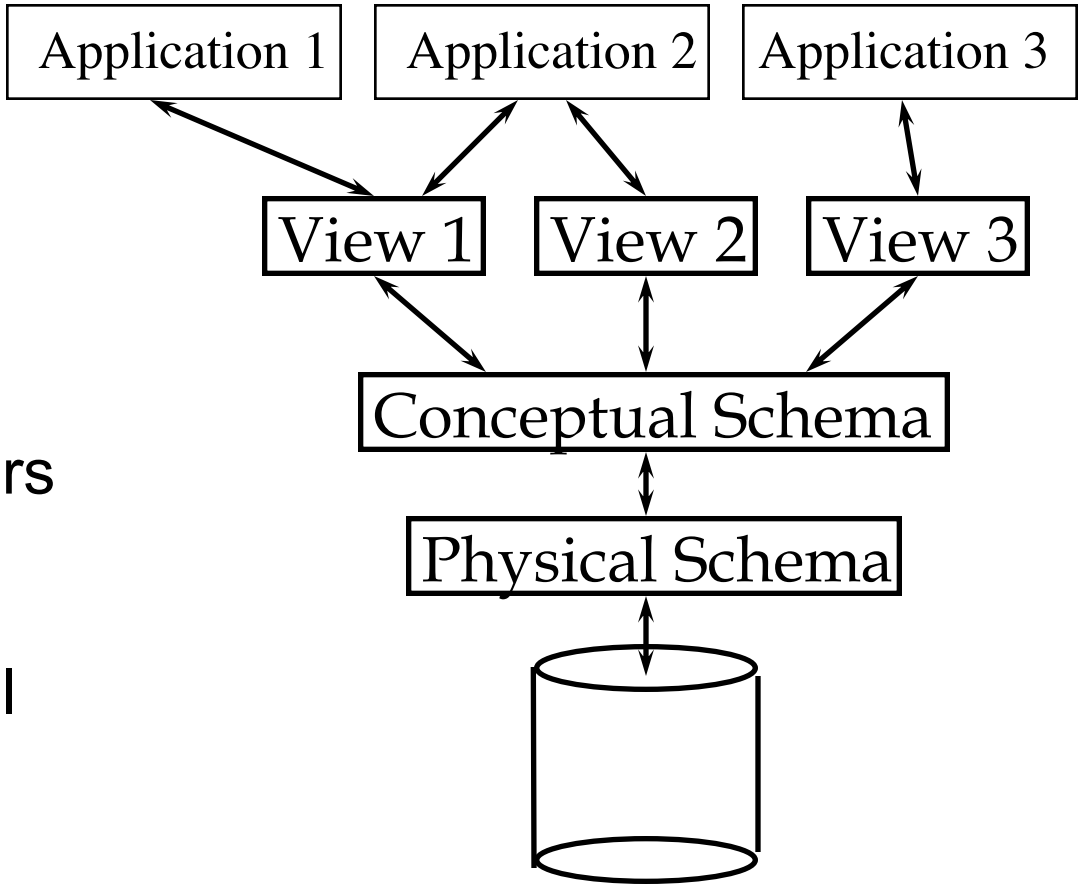
- Remove details related to data storage and access from application programs.
- Concentrate those functions in a single system called *database management system*.
- Have all applications access data through the DBMS.

Why Use a DBMS?

- Reduced redundancy
- Less risk of inconsistency
- Reduced application development time
- Uniform data administration
- Concurrent access, recovery from crashes
- **But,** most importantly
  Data Independence

Three Schema Levels

- **Schema:** a description of the data contents, structures and other aspects.
- External schema: what appl. programs and users see.
- Conceptual schema: description of the logical structure of data.
- Physical schema: file structures and indexes being used.



Example: University Database

- External Schema (View):
  - *Course_info(cid:char(8),enrollment:int)*
- Conceptual schema:
  - *Students(sid: char(8), name: char(16), login: char(8), age: int, gpa:float)*
  - *Courses(cid: char(8), cname:char(16), credits:int)*
  - *Enrolled(sid:char(8), cid:char(8), grade:char(2))*
- Physical schema:
  - Students is stored as an unordered file with file name data.txt.
  - There is an index on the first column of Students.

Data Independence

- Objective: Application programs unaffected by changes in storage structure and access strategy.
- *Logical data independence*: Protection from changes in *logical* structure of data.
- *Physical data independence*: Protection from changes in *physical* structure of data.

☛ *One of the most important benefits of using a DBMS!*

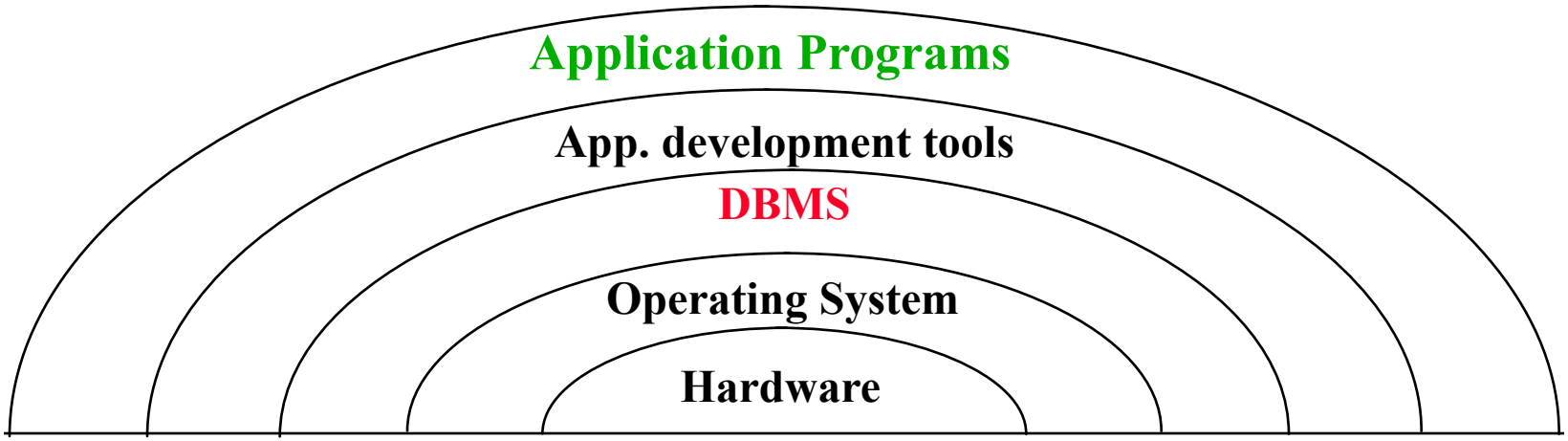
DBMS Functionality

- Data definition facilities
  - provides a data definition language (DDL),
  - stores the definitions in a user-accessible catalog (data dictionary).
- Data manipulation facilities
  - provides a query language for storing, retrieving and updating data.
- Facilities for integrity constraints
  - does the validation check for integrity constraints before updates,
  - different kinds of constraints
    - ✓ primary key constraints (entity integrity),
    - ✓ foreign key constraints (no dangling references),
    - ✓ check constraints.

DBMS Functionality (cont'd)

- Provides concurrency control
  - multiple users simultaneously access/update a database.
- Supports transactions
  - a sequence of user operations to be performed as an atomic action,
  - all operations or none are performed.
- Provides database recovery
  - whatever happens, never lose data.
- Provides query optimization
  - find the best possible plan for executing a query.

Place in a Computer System



Why this Course?

- A paradigm shift from *computation* to *information*
- "Big data" is everywhere!
  - Digital libraries, online stores, the Web, etc.
- Many applications need DBMS functionalities
- One size doesn't fit all

- Challenge: manage this diversity and volume (need more people).
- It is also fun!

Course Outline

- What are they and why are they useful?
  - Introduction
- How can I use them?
  - Relational model
  - ER model
  - Relational query languages
  - Database design
- How do they work?
  - Secondary storage
  - Organizing data in files and indexes
  - Physical database design
  - Database tuning

Summary

- Introduced databases and DBMS
- Reviewed some functionalities of a DBMS and benefits
- Three-layered schema architecture
- Not sold yet?
  - Google returns 122 million hits for “jobs DBA” compared to 171 million for “jobs programmer”,
  - and good DBAs are well-paid!
    - As of Aug 2019, DBAs and DB developers are paid $77k-$167k
      - http://www.itcareerfinder.com/brain-food/it-salaries/database-administrator-salary-range.html
  - DBMS R&D is one of the broadest and exciting areas in CS