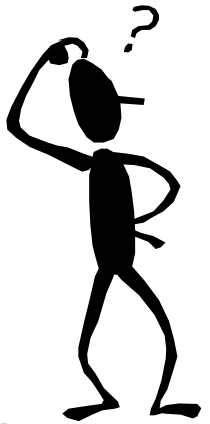
CS392 - Introduction to Database Systems

姚 斌

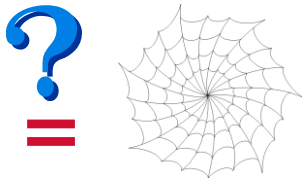
yaobin@cs.sjtu.edu.cn

手机：**15921379438**

What Is a Database *System*?

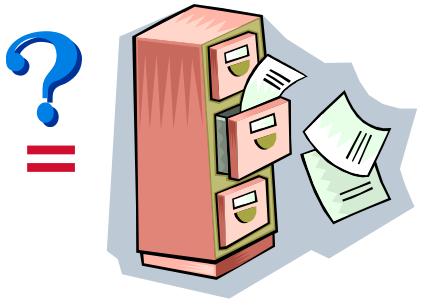


- **Database:**
  a very large, integrated collection of data.
- **Models a real-world *enterprise* (e.g. ER model)**
  - Entities (e.g., students, teams, games)
  - Relationships
    (e.g., faculty teaching courses, use of classroom)
- **A *Database Management System (DBMS)* is a software system designed to store, manage, and facilitate access to databases.**



Is the WWW a DBMS?

- **Fairly sophisticated search available**
  - crawler *indexes* pages on the web
  - Keyword-based search for pages
- **But, currently**
  - data is mostly unstructured and untyped
  - search only:
    - can't modify the data
    - can't get summaries, complex combinations of data
  - few guarantees provided for freshness of data, consistency across data items, fault tolerance, …
  - Web sites (e.g. e-commerce) typically have a DBMS in the background to provide these functions.
- **The picture is changing**
  - New standards like XML can help data modeling
  - Research groups are working on providing some of this functionality *across multiple web sites.*
  - The WWW/DB boundary is blurring!



Is a File System a DBMS?

- **Thought Experiment 1:**
  - You and your project partner are editing the same file.
  - You both save it at the same time.
  - Whose changes survive?

A) Yours B) Partner's C) Both D) Neither E) ???

- **Thought Experiment 2:**
  - You're updating a file.
  - The power goes out.
  - Which of your changes survive?

A) All B) None C) All Since last save D) ???

Q: How do you write programs over a subsystem when it promises you only "???" ?

A: Very, very carefully!!

Why Study Databases??

- **Shift from *computation* to *information***
  - always true for corporate computing
  - Web made this point for personal computing
  - more and more true for scientific computing
- **Need for DBMS has exploded in the last years**
  - Corporate: retail swipe/clickstreams, "customer relationship mgmt", "supply chain mgmt", "data warehouses", etc.
  - Scientific: digital libraries, Human Genome project, NASA Mission to Planet Earth, physical sensors, grid physics network
- **DBMS encompasses much of CS in a practical discipline**
  - OS, languages, theory, AI, multimedia, logic
  - Yet traditional focus on real-world apps

What's the intellectual content?

- **representing information**
  - data modeling
- **languages and systems for querying data**
  - complex queries with real *semantics**
  - over massive data sets
- ***concurrency control* for data manipulation**
  - controlling concurrent access
  - ensuring *transactional semantics*
- **reliable data storage**
  - maintain data semantics even if you pull the plug

* semantics: the meaning or relationship of meanings of a sign or set of signs

Rest of Today

- **A "free tasting" of things to come in this class:**
  - data modeling
  - query languages
  - DBMSs
  - Application development and web data management

- **Today's lecture is from Chapter 1 in R&G**

OS Support for Data Management

- **Data can be stored in RAM**
  - this is what every programming language offers!
  - RAM is fast, and random access
  - Isn't this heaven?
- **Every OS includes a File System**
  - manages *files* on a magnetic disk
  - allows *open, read, seek, close* on a file
  - allows protections to be set on a file
  - drawbacks relative to RAM?

Database Management Systems

- **What more could we want than a file system?**
  - Simple, efficient *ad hoc*[1] queries
  - concurrency control
  - recovery
  - benefits of good data modeling
- **S.M.O.P.[2]? Not really…**
  - as we'll see this semester
  - in fact, the OS often gets in the way!

---

[1] ad hoc: formed or used for specific or immediate problems or needs

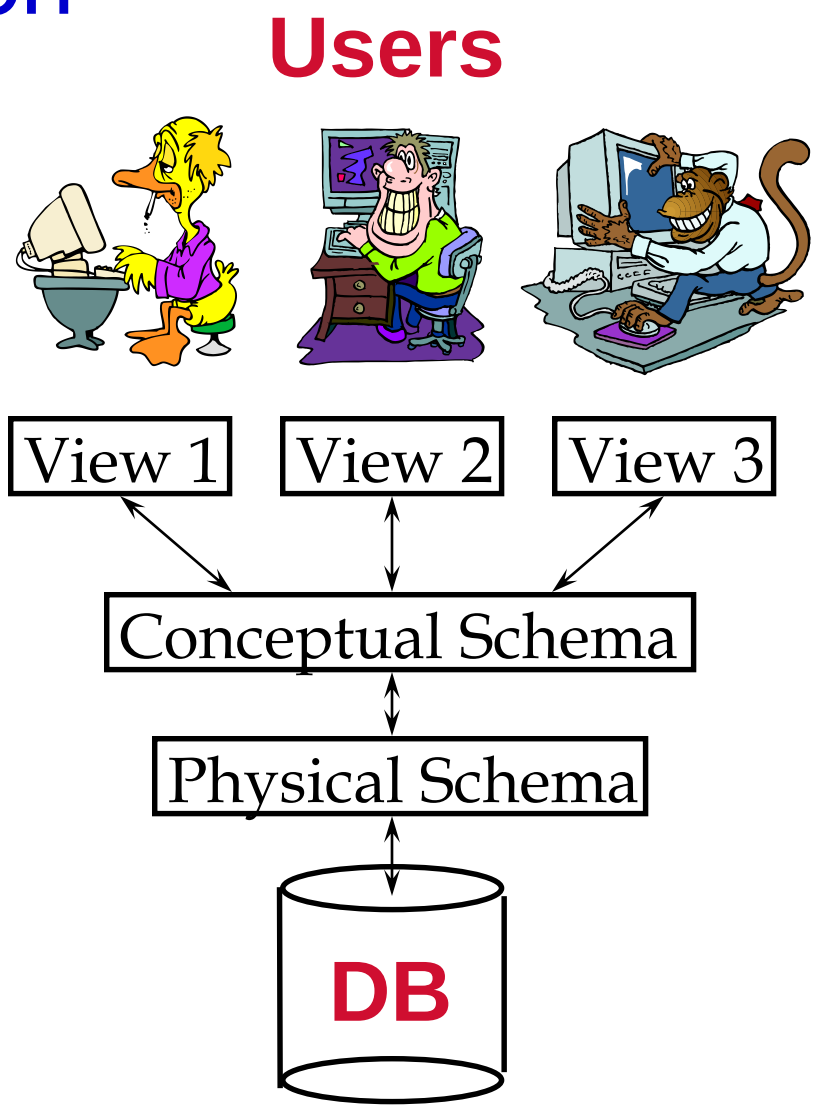
[2] SMOP: Small Matter Of Programming

Describing Data: Data Models

- **A *data model* is a collection of concepts for describing data.**
- **A *schema* is a description of a particular collection of data, using a given data model.**
- **The *relational model of data* is the most widely used model today.**
  - Main concept: *relation*, basically a table with rows and columns.
  - Every relation has a *schema*, which describes the columns, or fields.

Levels of Abstraction

- Views describe how users see the data.
- Conceptual schema defines logical structure
- Physical schema describes the files and indexes used.



Example: University Database

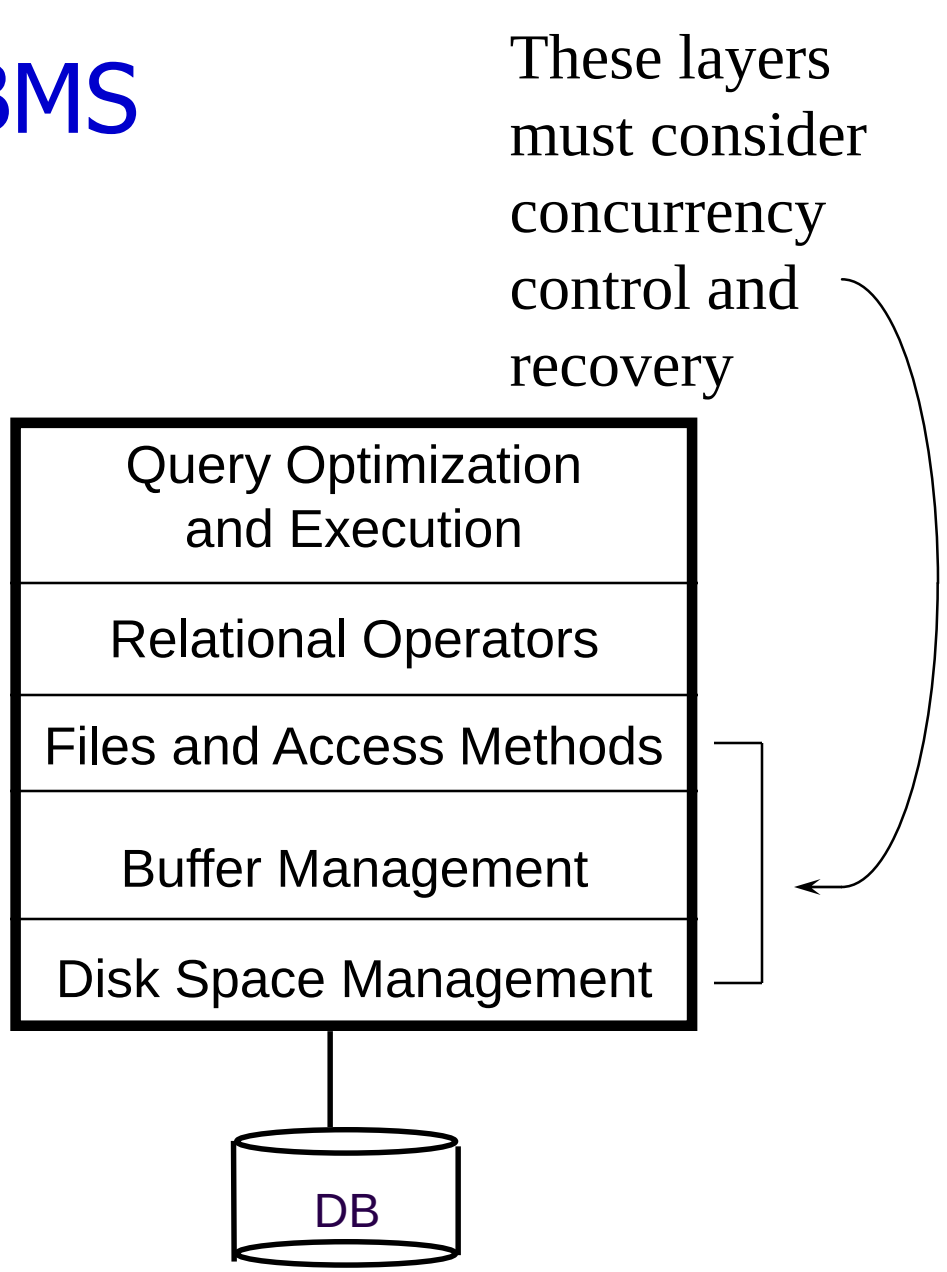
- **Conceptual schema:**
  - *Students(sid: string, name: string, login: string, age: integer, gpa:real)*
  - *Courses(cid: string, cname:string, credits:integer)*
  - *Enrolled(sid:string, cid:string, grade:string)*
- **Physical schema:**
  - Relations stored as unordered files.
  - Index on first column of Students.
- **External Schema (View):**
  - *Course_info(cid:string,enrollment:integer)*

Concurrency Control

- **Concurrent execution of user programs: key to good DBMS performance.**
  - Disk accesses frequent, pretty slow
  - Keep the CPU working on several programs concurrently.
- **Interleaving actions of different programs: trouble!**
  - e.g., account-transfer & print statement at same time
- **DBMS ensures such problems don't arise.**
  - Users/programmers can pretend they are using a single-user system. (called "Isolation")
  - Thank goodness! Don't have to program "very, very carefully".

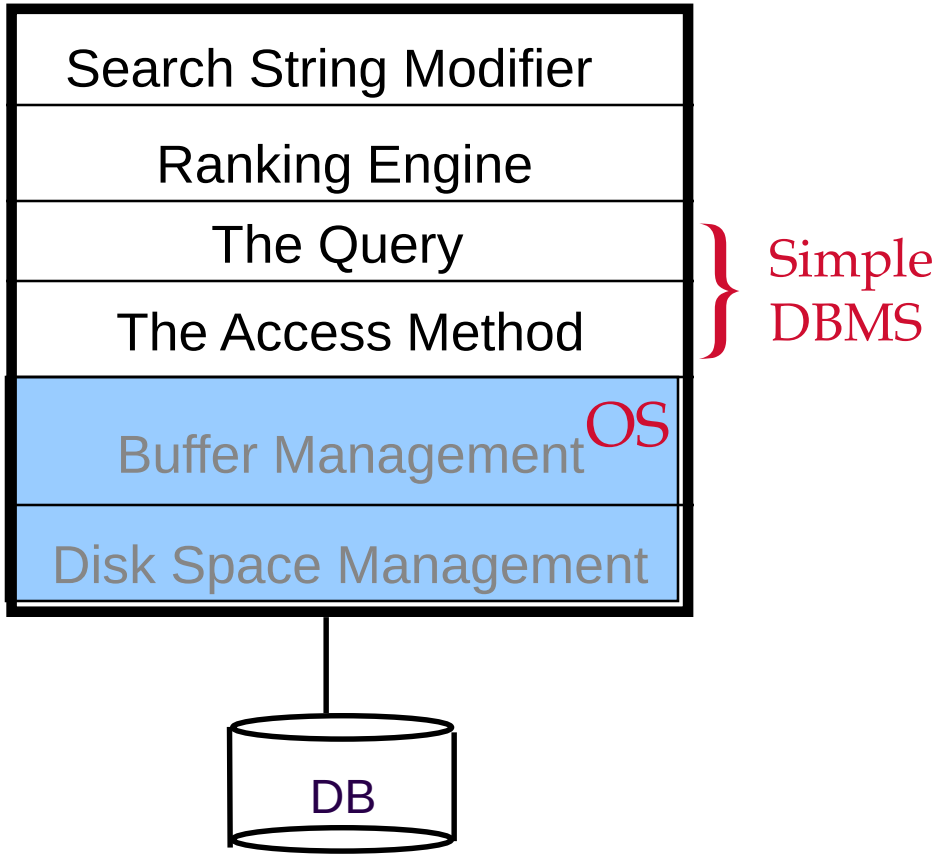
Structure of a DBMS

- **A typical DBMS has a layered architecture.**
- **The figure does not show the concurrency control and recovery components.**
- **Each system has its own variations.**
- **The book shows a somewhat more detailed version.**



FYI: A text search engine

- **Less "system" than DBMS**
  - Uses OS files for storage
  - Just one access method
  - One hardwired query
    - regardless of search string
- **Typically no concurrency or recovery management**
  - Read-mostly
  - Batch-loaded, periodically
  - No updates to recover
  - OS a reasonable choice
- **Smarts: text tricks**
  - Search string modifier (e.g. "stemming" and synonyms)
  - Ranking Engine (sorting the output, e.g. by word or document popularity)
  - no semantics: WYGIWIGY



There may be time to talk about some of these text tricks in this class, but it won't be a focus.

Advantages of a DBMS

- **Data independence**
- **Efficient data access**
- **Data integrity & security**
- **Data administration**
- **Concurrent access, crash recovery**
- **Reduced application development time**
- **So why not use them always?**
  - Expensive/complicated to set up & maintain
  - This cost & complexity must be offset by need
  - General-purpose, not suited for special-purpose tasks (e.g. text search!)

Databases make these folks happy ...

- **DBMS vendors, programmers**
  - Oracle, IBM, MS, Sybase, SUN, …
- **End users in many fields**
  - Business, education, science, …
- **DB application programmers**
  - Build enterprise applications on top of DBMSs
  - Build web services that run off DBMSs
- **Database administrators (DBAs)**
  - Design logical/physical schemas
  - Handle security and authorization
  - Data availability, crash recovery
  - Database tuning as needs evolve

…must understand how a DBMS works

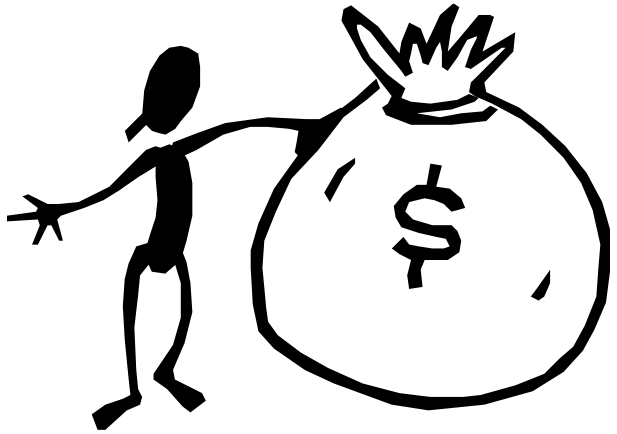
Summary

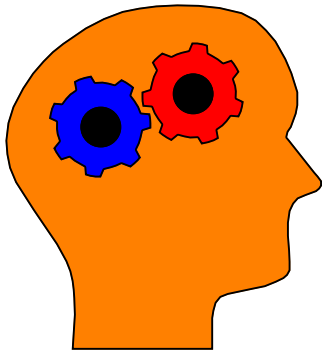
- **DBMS used to maintain, query large datasets.**
  - can manipulate data and exploit *semantics*
- **Other benefits include:**
  - recovery from system crashes,
  - concurrent access,
  - quick application development,
  - data integrity and security.
- **In this course we will explore:**

  How to be a sophisticated user of DBMS technology

Summary, cont.

- DBAs, DB developers the bedrock of the information economy





- DBMS R&D represents a broad, fundamental branch of the science of computation