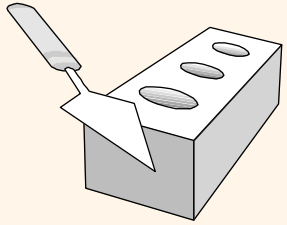
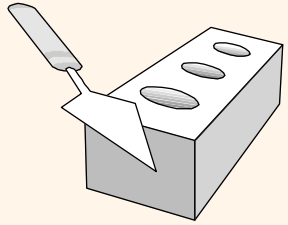


Database Management Systems

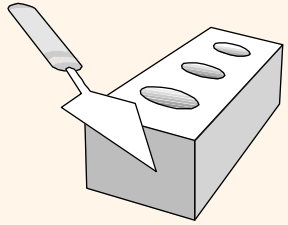
Chapter 1: Overview of Database Systems

Instructor: Azadeh Shakery
shakery@ut.ac.ir



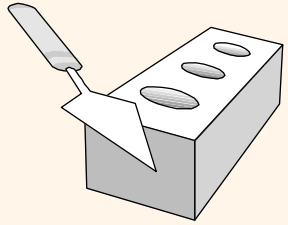
What Is a DBMS?

- A *Database* is a very large, integrated collection of data.
- Models real-world *enterprise.*
  - Entities (e.g., students, courses)
  - Relationships (e.g., John is taking ECE459)
- A *Database Management System (DBMS)* is a software package designed to store and manage databases.



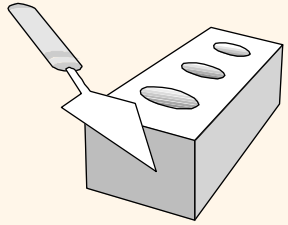
Files vs. DBMS

- Application must stage large datasets between main memory and secondary storage (e.g., buffering, page-oriented access, 32-bit addressing, etc.)
- Special code for different queries
- Must protect data from inconsistency due to multiple concurrent users
- Crash recovery
- Security and access control



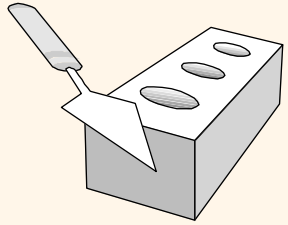
Why Use a DBMS?

- Data independence (abstract view of data) and efficient access.
- Reduced application development time.
- Data integrity (enforce constraints) and security.
- Uniform (central) data administration.
- Concurrent access, recovery from crashes.



Why Study Databases??

- Data is meaningless without the tools to extract information from the data
  - Optimal pricing of an airline ticket, gene finding, ...
- Datasets increasing in diversity and volume.
  - Digital libraries, interactive video, Human Genome project, EOS project
  - ... need for DBMS exploding
- DBMS intersects with most of CS
  - OS, languages, theory, AI, multimedia, logic

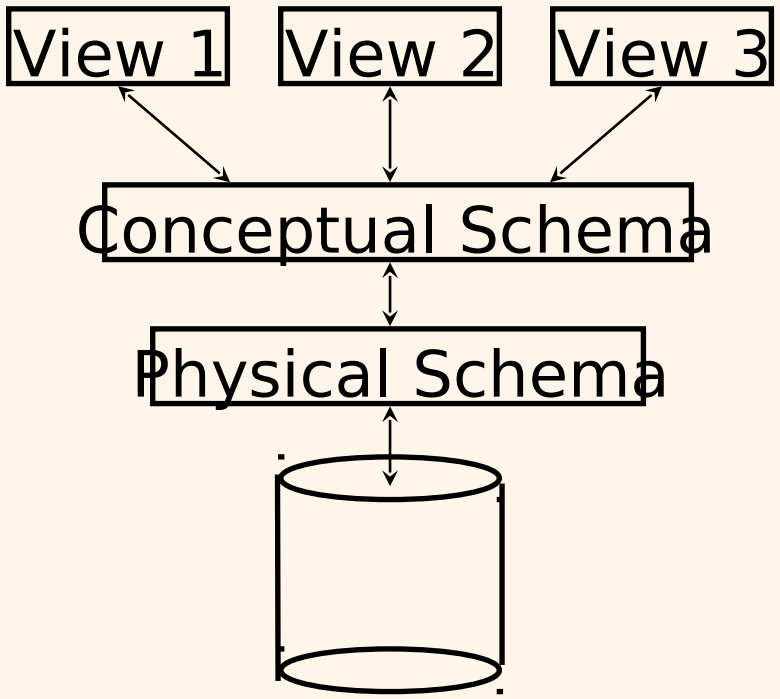


Data Models

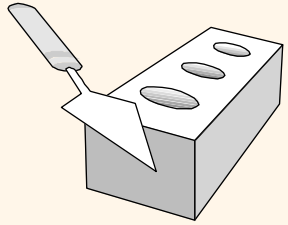
- A *data model* is a collection of concepts for describing data.
- A *schema* is a description of a particular collection of data, using the given data model and its data definition language.
- The *relational model of data* is the most widely used model today.
  - Main concept: *relation*, basically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

Levels of Abstraction

- Many *views*, single *conceptual (logical) schema* and *physical schema*.
  - Views describe how users see the data.
  - Conceptual schema defines logical structure
  - Physical schema describes the files and indexes used.



- *Schemas are defined using DDL; data is modified/queried using DML.*



Example: University Database

- Conceptual schema:

  Students(sid: string, name: string, login: string, age:integer, gpa:real)

  Faculty(fid: string, fname: string, sal:real)

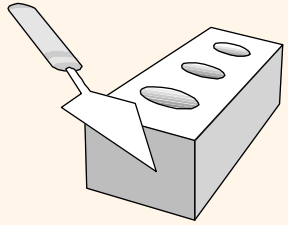
  Courses(cid: string, cname:string, credits:integer)

  Teaches(fid: string, cid:string)

  Enrolled(sid:string, cid:string, grade:string)

- Physical schema:
  - Relations stored as unordered files.
  - Index on first column of Students.
- External Schema (View):

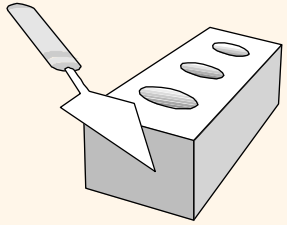
  Course_info(cid:string, fname: string, enrollment: integer)



Data Independence *

- Applications insulated from how data is structured and stored.
- *Logical data independence*: Protection from changes in *logical* structure of data (e.g., adding new fields).
- *Physical data independence*: Protection from changes in *physical* structure of data (e.g., sorting, indexing, compressing).

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

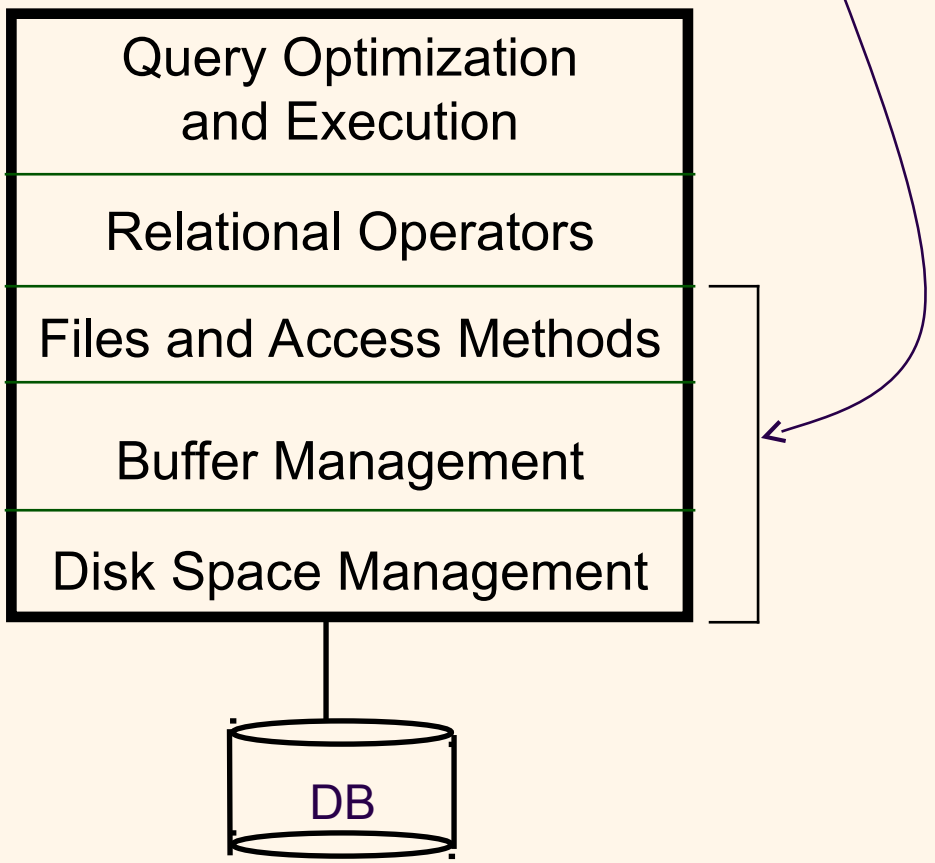


Concurrency Control

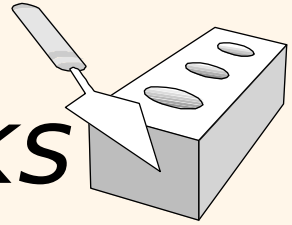
- Concurrent execution of user programs is essential for good DBMS performance.
  - Because disk accesses are frequent, and **relatively slow**, it is important to keep the cpu humming by working on several user programs concurrently.
- Interleaving actions of different user programs can lead to inconsistency: e.g., check is cleared while account balance is being computed.
- DBMS ensures such problems don't arise: users can pretend they are using a single-user system.

Structure of a DBMS

- A typical DBMS has a layered architecture.
- The figure does not show the concurrency control and recovery components.
- This is one of several possible architectures; each system has its own variations.



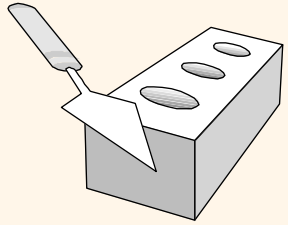
Databases make these folks happy ...



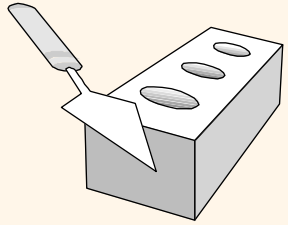
- End users and DBMS vendors
- DB application programmers
- *Database administrator (DBA)*
  - Designs logical /physical schemas
  - Handles security and authorization
  - Data availability, crash recovery
  - Database tuning as needs evolve

Must understand how a DBMS works!

What We Will See In This Course

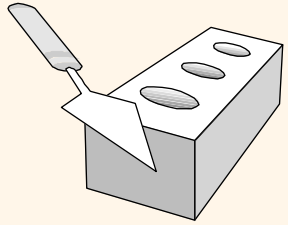


- Foundations
- Storage and Indexing
- Query Evaluation
- Some Additional Topics



Course Format

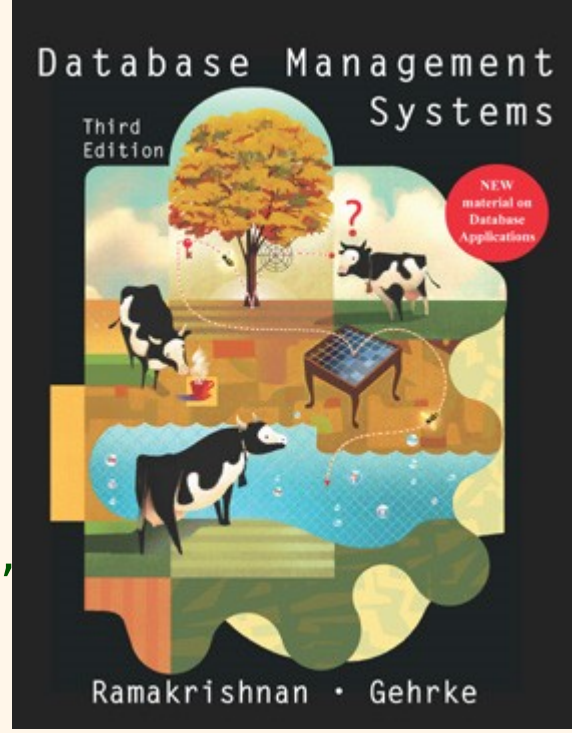
- Lecture-based
- Assignments
- Quizzes
- Midterm
- Final



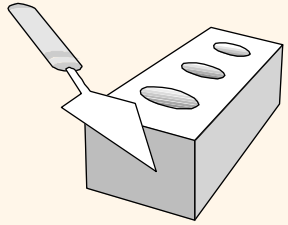
Course policy and grading

- Homework
  - Due at the beginning of the class
  - Late policy: 30% penalty
  - Quiz
    - Both announced an unannounced
- Grading (approximate):
  - Homework: 20%
  - Quiz: 10%
  - Midterm exam: 35%
  - Final exam: 35%

References

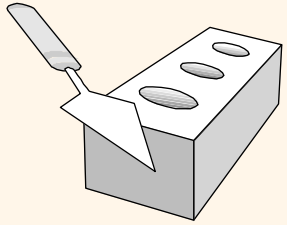


- Main textbook:
  - *Database Management Systems, 3rd Edition*, by Ramakrishnan and Gehrke, 2002
- Other textbooks:
  - *A First Course in Database Systems, 2nd Edition*, by Ullman and Widom, 2002.
  - *Database System Concepts, 5th Edition*, by Silberschatz et al , 2005.
  - *An Introduction to Database Systems, 7th Edition*, by Date, 2000.
  - *Fundamentals of Database Systems, 4th Edition*, by Elmasri and Navathe, 2003



Questions?

- Ask the TAs
  - Head TA: Emad Kebriaei (emadkebriaei@gmail.com)
- Come to my office (with prior appointment)
- Email me at: shakery@ut.ac.ir



Question?