**Chapter 1:**

What is a DBMS?

* Is storage of data;
* Maintains a *very large, integrated* collection of data

A database models real-world activities in various scenarios

* Entities (e.g., students, courses)
* Relationships (e.g., John is taking CMPSC431W)

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

**Database Applications**

* Banking: deposit, withdrawal, all transactions
* Airlines: reservations, flights
* So many examples

**Files vs. DBMS**

In the early days, database applications were built on top of file systems

* 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?**

* Capability to store massive amounts of data
* Efficient access
* Data independence
* Reduced application development time
* Data integrity
* Uniform data administration
* Concurrent access
* Recovery from crashes
* Security

**Why Study Databases?**

* *Big Data*
  + Database is an essential component
* Shift from *computation* to *information*
  + At the “low end”: scramble to web space (a mess)
  + At the “high end”: scientific applications
* Datasets increasing in diversity and volume
  + Digital libraries, interactive video, human genome project, EOS project
  + … need for DBMS is exploding
* DBMS encompassing most of CS
  + OS, languages, theory, AI, multimedia, logic

**Data Models**

* A **data model** is a collection of high-level constructs for describing stored data that hides low-level storage details
* A **schema** is a description of a particular collection of data, based on a given data model
* Three major data models
  + Network data model
  + Hierarchical data model
  + Relational data model \*\*\*\*\*\*\*

**Relational Data Model**

* The ***relational data model*** is the most widely used data model today
  + Main concept: ***relation,*** basically a table with rows and columns
  + Every relation has a ***schema,*** which describes the columns, or fields, and generally the structure of the table
* For examples: students (sid: string, name: string, login: string, age: integer, gpa: real)

**Levels of Abstraction**

* Many ***views (external schema)***, single ***logical (conceptual) schema*** and ***physical schema***
  + Views describe how users see the data. Support security and access control
  + Logical schema (also referred to as conceptual schema in textbook) defines logical structure based on data model
  + Physical schema describes the files and indexes used (i.e., the storage details)
* Conceptual & external schemas are defined using DDL
* Data is modified/queried using DML

**Schema** – the logical structure of the database

* Analogous to type information of a variable in a program

**Instance** – the actual content of the database at a particular point in time

* Analogous to the value of a variable

**Physical Schema:**

* Relations stored as ***unordered files***
* The second column of students is indexed by a ***B-tree***

**Logical Schema:**

* Students (sid: string, name: string, login: string, etc)
* Courses (cid: string, cname: string, credits: integer)
* Enrolled (sid: string, etc)

**External Schema (view):**

* Course\_info (<cid:string>, enrollment:integer)

**Data independence**

* Applications are insulated from how data is structured and stored
* ***Logical data independence:*** protection from changes in *logical* structure of data
  + Achieved by revising view definition in accordance with new logical schema. Users don’t see the change
* ***Physical data independence:*** protection from changes in *physical* structure of data
  + Logical structures (e.g., tables in RDB) are supported by different physical storage structures
* ***This is one of the most important benefits of using a DBMS!***

**Semantic Data Models**

* Data models of databases (**system-oriented)**
  + while hiding details of how data physically stored, they still closely related to how data are seen (represented) in the databases
  + for example, data are organized as tables in relational databases
* *Semantic data models* (**application-oriented**)
  + More abstract, high-level
  + Capture concepts and represent data closer to the real world
  + For example, ***entity-relationship (ER) data model***
  + Data modeled with a semantic data model can be stored in various kinds of databases (based on different data models)

**Queries in a DBMS**

* Some questions a user might ask:
  + What is the average salary of professors who teach 400-level CSE courses?
  + How many students are enrolled in CMSPC431W
* Questions involving the data stored in a DBMS are called ***queries***
* A ***query language*** is used to pose queries
* ***Structural Query Language (SQL),*** which supports a rich class of queries, has contributed greatly to the success of relational DBMS
* ***Relational Algebra*** and ***relational calculus*** are two formal query languages providing theoretical foundation for relational DBMS

**Databases and the Web**

* We have entered the internet age, DBMSs store data in support of internet systems and web applications
* Accessing databases through web interfaces
  + Java programming interface (JDBC)
  + Embedding into HTML pages (JSP)
  + Access through https protocol (web services)
* Using web document formats for data definition and manipulation
  + XML, Xquery, Xpath
  + XML databases and messaging systems

**Web-Based Database Applications**

* Can be divided into three separate functional components:
  + Data management
  + Application logic
  + Presentation
* The system architecture determines whether these three components reside on a single system (“tier) or are distributed across several tiers
  + Single-tier architecture
  + Two-tier architecture
    - Think and thick clients
  + three-tier architecture

**Transaction Management**

* Many enterprises use databases to store information about their state
  + E.g., balances of all depositors
* The occurrence of a real-world event that changes the enterprise state requires the execution of a program that changes the database state in a corresponding way
  + E.g., balance must be updated when you deposit
* A ***transaction*** is a program that accesses the database in response to real-world events
* Concurrent execution of a transaction is essential for good DBMS performance

**Folks interact with Databases**

* **End users**
  + You and me
* **DB application programmers**
  + Needed in all domains
* ***Database administrators (DBA)***
  + Designs logical/physical schemas
  + Handles security and authorization
  + Data availability, crash recovery
  + Database tuning as needs evolve
* **DBMS vendors**
  + IBM, Oracle, Microsoft