Lecture 1 Notes

Introduction To DBMS

Discussion Points:

- ❖ Data, Database and Database Management System
- ❖ Database-System Applications
- ❖ View of Data: Data Abstraction
- ❖ Instances and Schemas
- ❖ Database Architecture
- ❖ Data Models
- ❖ SQL: Introduction

❖ Data, Database and Database Management System

- *Data:* are observations or measurements (unprocessed or processed) represented as text, numbers, or multimedia.
- *Dataset:* is a structured collection of data generally associated with a unique body of work.
- *Database:* The collection of data, usually referred to as the database, contains information relevant to an enterprise. A database is an organized collection of data stored as multiple datasets.
- *Database Management System:* A database-management system is a collection of interrelated data and a set of programs to access those data. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.

❖ Database-System Applications

Databases are widely used. Here are some representative applications:

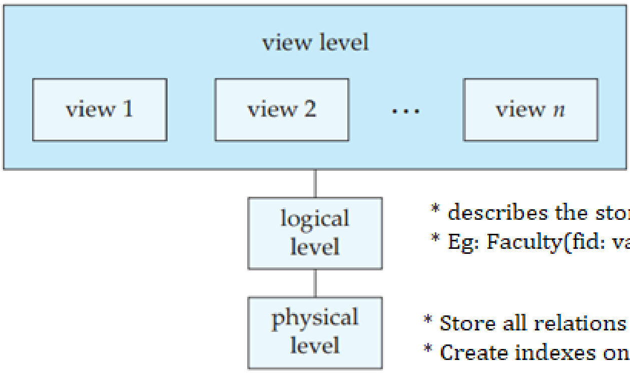
- *Enterprise Information*
  - *Sales:* For customer, product, and purchase information.
  - *Accounting:* For payments, receipts, account balances, assets and other accounting information.
  - *Human resources:* For information about employees, salaries, payroll taxes, and benefits, and for generation of paychecks.
  - *Manufacturing:* For management of the supply chain and for tracking production of items in factories, inventories of items in warehouses and stores, and orders for items.
  - *Online retailers:* For sales data noted above plus online order tracking, generation of recommendation lists, and maintenance of online product evaluations.
- *Banking and Finance*
  - *Banking:* For customer information, accounts, loans, and banking transactions.
  - *Credit card transactions:* For purchases on credit cards and generation of monthly statements.
  - *Finance:* For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds; also, for storing real-time market data to enable online trading by customers and automated trading by the firm.
- *Universities:* For student information, course registrations, and grades (in addition to standard enterprise information such as human resources and accounting).
- *Airlines:* For reservations and schedule information. Airlines were among the first to use databases in a geographically distributed manner.
- *Telecommunication:* For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.
- Many more..

View of Data: Data Abstraction

Since many database-system users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users' interactions with the system:

- *Physical level.* The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in detail.
- *Logical level.* The intermediate level of abstraction describes what data are stored in the database, and what relationships exist among those data. The logical level thus describes the entire database in terms of a small number of relatively simple structures.

  Although implementation of the simple structures at the logical level may involve complex physical-level structures, the user of the logical level does not need to be aware of this complexity. This is referred to as **physical data independence**.
- *View level.* The highest level of abstraction describes only part of the entire database. The view level of abstraction exists to simplify the user's interaction with the system. The system may provide many views for the same database.



* Each external schema consists of a collection of one or more views and relations from the conceptual schema.
* A view is conceptually a relation, but the records in a view are not stored in the DBMS.

* describes the stored data in terms of the data model
* Eg: Faculty(fid: varchar, fname: varchar, sal: integer)

* Store all relations as unsorted files of records
* Create indexes on the columns

Instances and Schemas

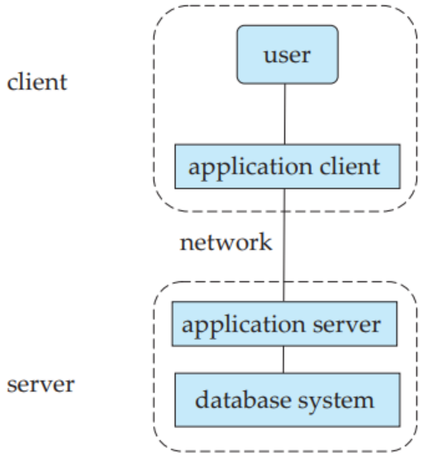
- *Instance:* Databases change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an instance of the database.
- *Schema:* The overall design of the database is called the database schema.

Database systems have several schemas, partitioned according to the levels of abstraction:

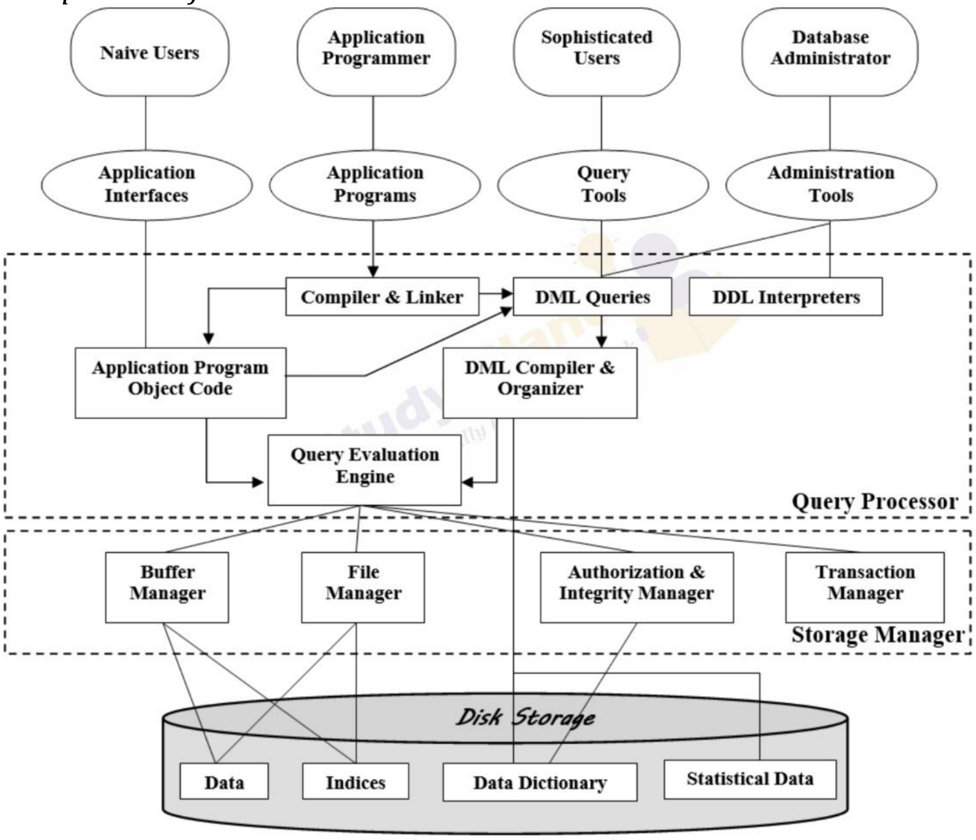
- The physical schema describes the database design at the physical level.
- While the logical schema describes the database design at the logical level.
- A database may also have several schemas at the view level, sometimes called subschemas, that describe different views of the database.

Database Architecture

- In a three-tier architecture, the client machine acts as merely a front end and does not contain any direct database calls. Instead, the client end communicates with an application server, usually through a forms interface. The application server in turn communicates with a database system to access data. The business logic of the application is embedded in the application server, instead of being distributed across multiple clients.



- *Components of DBMS:*



- *Terms:*
    - *Data files:* store the database itself.
    - *Data dictionary:* stores metadata about the structure of the database, in particular the schema of the database.
    - *Indices:* which can provide fast access to data items. Like the index in this textbook, a database index provides pointers to those data items that hold a particular value.
    - *Storage Manager:* The storage manager is the component of a database system that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system. The storage manager is responsible for the interaction with the file manager.

- *Authorization and Integrity manager:* tests for the satisfaction of integrity constraints and checks the authority of users to access data.
- *File manager:* manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
- *Buffer Manager:* is responsible for fetching data from disk storage into main memory and deciding what data to cache in main memory.
- *Transaction:* is a collection of operations that performs a single logical function in a database application.
- *Atomicity:* The all-or-none requirement is called atomicity.
- *Consistency:* The correctness requirement is called consistency.
- *Durability:* The persistence requirement is called durability.
- *Transaction Manager:* ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting. It consists of the concurrency-control manager and the recovery manager.
- *Recovery Manager:* Ensures that the atomicity and durability properties is the responsibility of the database system itself—specifically, of the recovery manager.
- *Concurrency-control Manager*: Controls the interaction among the concurrent transactions, to ensure the consistency of the database.
- *DDL interpreter,* which interprets DDL statements and records the definitions in the data dictionary.
- *DML compiler,* which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands. The DML compiler also performs query optimization; that is, it picks the lowest cost evaluation plan from among the alternatives.
- *Query evaluation engine,* which executes low-level instructions generated by the DML compiler.

❖ Data Models

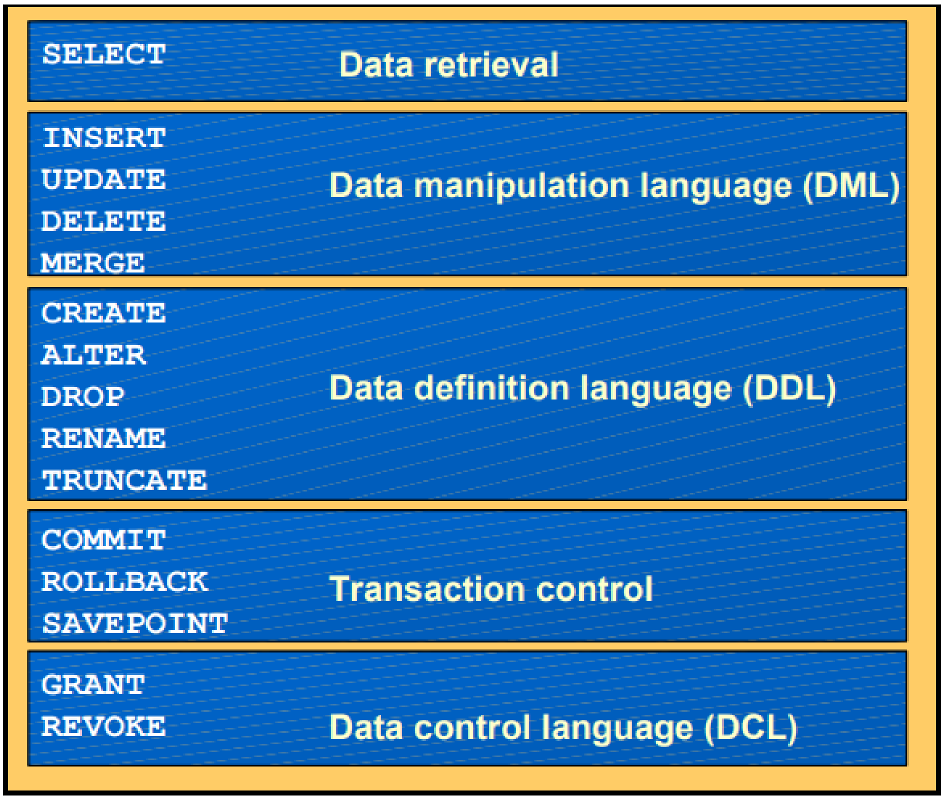
- o *Data Model:* a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. A data model provides a way to describe the design of a database at the physical, logical, and view levels.
- o Data models can be classified into four different categories:
    - *Relational Model.* The relational model uses a collection of tables to represent both data and the relationships among those data. Tables are also known as relations. The relational model is an example of a record-based model. The relational data model is the most widely used data model, and a vast majority of current database systems are based on the relational model.
    - *Entity-Relationship Model.* The entity-relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects. An entity is a “thing” or “object” in the real world that is distinguishable from other objects.
    - *Object-Based Data Model.* Object-oriented programming (especially in Java, C++) has become the dominant software-development methodology. This led to the development of an object-oriented data model that can be seen as extending the E-R model with notions of encapsulation, methods (functions), and object identity. The object-relational data model combines features of the object-oriented data model and relational data model.

- *Semi-structured Data Model.* The semi-structured data model permits the specification of data where individual data items of the same type may have different sets of attributes. This contrasts with the data models mentioned earlier, where every data item of a particular type must have the same set of attributes.

❖ SQL Introduction

- SQL is the most widely used commercial relational database language.
- Features of Structured Query Language (SQL):
  - *Data-definition language (DDL).* The SQL DDL provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
  - *Data-manipulation language (DML).* The SQL DML provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.
  - *Integrity.* The SQL DDL includes commands for specifying integrity constraints that the data stored in the database must satisfy. Updates that violate integrity constraints are disallowed.
  - *View definition.* The SQL DDL includes commands for defining views.
  - *Transaction control.* SQL includes commands for specifying the beginning and ending of transactions.
  - *Embedded SQL and dynamic SQL.* Embedded and dynamic SQL define how SQL statements can be embedded within general-purpose programming languages, such as C, C++, and Java.
  - *Authorization.* The SQL DDL includes commands for specifying access rights to relations and views.

○ SQL Statements



SELECT	Data retrieval

INSERT
UPDATE
DELETE
MERGE
	Data manipulation language (DML)

CREATE ALTER DROP RENAME TRUNCATE	Data definition language (DDL)
COMMIT ROLLBACK SAVEPOINT	Transaction control
GRANT REVOKE	Data control language (DCL)

- **Acronyms Used:**
  - DBMS: Data Base Management System
  - DCL: Data Control Language
  - DDL: Data Definition language
  - DML: Data Manipulation Language
  - DQL: Data Query Language
  - ER Model: Entity-Relationship Model
  - OOP: Object-Oriented Programming
  - SQL: Structured Query Language
  - TCL: Transaction Control Language

- **References:**

  - "Database System Concepts", Avi Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
  - "Database Management Systems", Raghu Ramakrishnan, Johannes Gehrke, McGraw-Hill.
  - Oracle SQL Resources
  - Other Internet Sources