DBMS Basic Concepts

DBMS

- Database Management System
- Term Database requires understanding of data and information
- **Data:** It can be anything like name, place or number, etc. Data usually refers to raw data, or unprocessed data.
- **Information:** It is organized or classified data so that it has some meaningful values to the receiver.
  - Information is the processed data on which decisions and actions are based.

Difference between Data and Information?

Data	Information
Data is raw facts and figures	Information is a processed form of data
For example: 12 is data	For example: When 12 is stored in row column form as shown it is information. Age 12
Data are atomic level pieces of information	Information is a collection of data
Data does not help in decision making	Information helps in decision making

Database

- A database is a shared collection of logically related data designed to meet the information needs of an organization
- The related information when placed is an organized form makes a database.
- The organization of data/information is necessary because unorganized information has no meaning.

Purpose of DBMS

An example

- University Database:

Data about students, faculty, courses, research-laboratories, course registration/enrollment etc. Reflects the state of affairs of the academic aspects of the university.

Purpose: To keep an accurate track of the academic activities of the university.

Purpose of DBMS

Before DBMSs were introduced, organizations usually stored information in file processing system which has a number of disadvantages:

- Data redundancy and inconsistency
- Difficulty in accessing data
- Data isolation
- Integrity problems
- Atomicity problems
- Concurrent-access anomalies
- Security problems

Database Management System

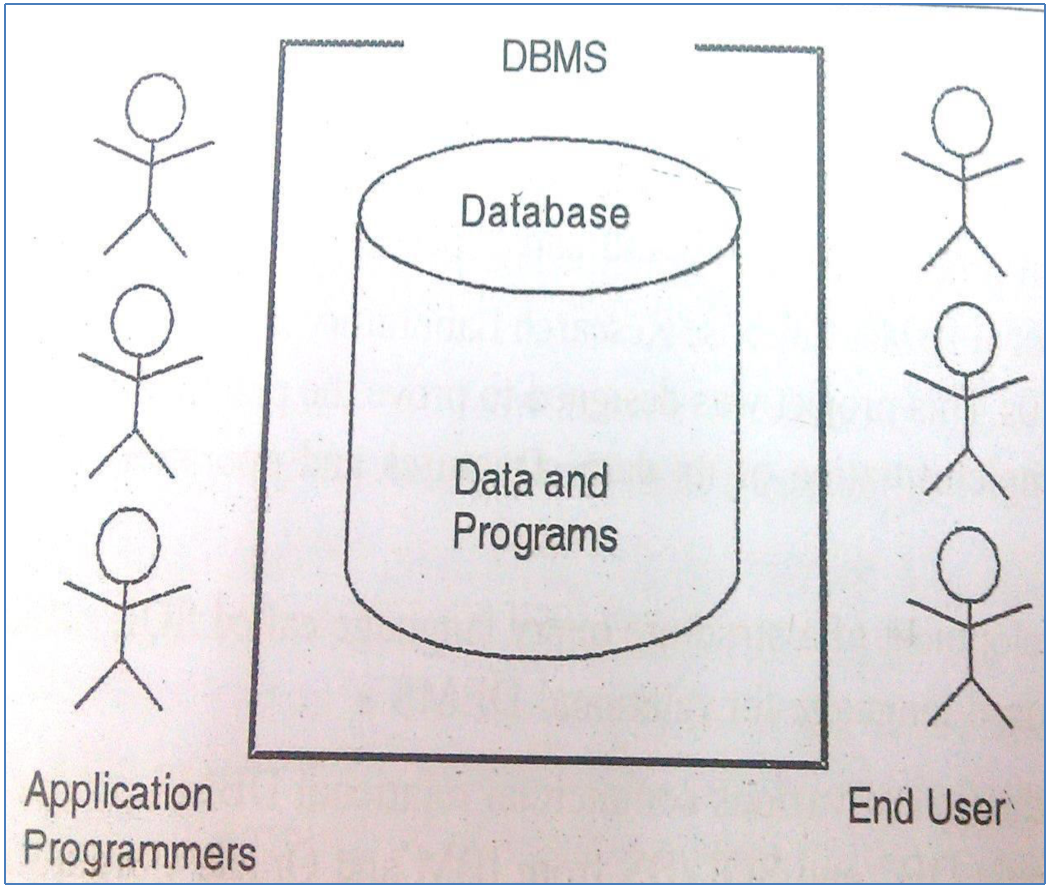
- **DBMS** A database management system is the software system that allows users to define, create and maintain a database and provides controlled access to the data.
- A database management system (DBMS) is basically a collection of programs that enables users to store, modify, and extract information from a database as per the requirements.

Operations on databases

- To add new information
- To view or retrieve the stored information
- To modify or edit the existing
- To remove or delete the unwanted information
- Arranging the information in a desired order etc.

Components of Database

- Five major components in database system environment:
  - Hardware
  - Software
  - Data
  - Users
  - Procedures



Components of Database System

- **Hardware:** It is the actual computer system used for keeping and accessing the database. DBMS hardware consists of secondary storage devices like hard disks.
- **Software:** It is the actual DBMS. Between the physical database itself and the users of system is a layer of software, called DBMS.
- **Data:** Data acts as the bridge between the machine components and user components.

Components of Database System

- **Users:** There are number of users who can access or retrieve data on demand using the applications and the interfaces provided by DBMS. The users can be:
  - Naïve users
  - Online users
  - Application Programmers
  - Sophisticated Users
  - Data base Administrator (DBA)

Components of Database System

- **Procedures:** It refers to the instructions and rules that govern the design and the use of the database. The users of the system and the staff that manage the database requires documented procedures on how to use or run the system.

Applications of DBMS

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

Data models, Schemas, and Instances

- **Data model:-**A set of concepts to describe the *structure* of a database, and certain *constraints* that the database should obey.
- **Schema:-** The overall description of the database is called the Database Schema.
  - A schema is defined as an outline or a plan that describes the records and relationships existing at the particular level.
- **Instance:-** Data in the database at a particular moment in time.

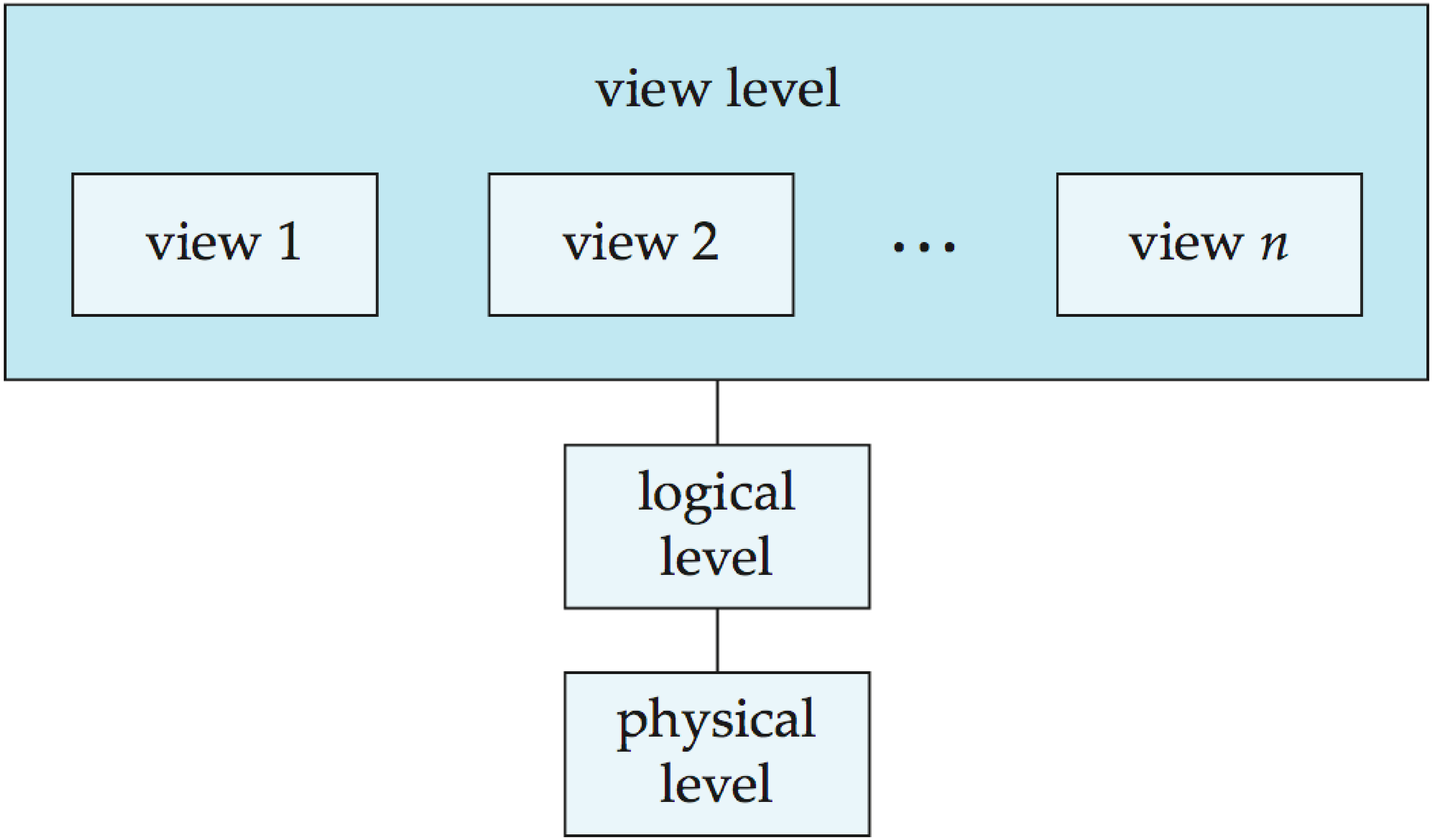
Data abstraction

- A major purpose of database system is to provide user with an abstract view of data. That is, system hides certain details of how the data are stored and maintained.

Levels of Abstraction(view of data)

- **Physical level:** describes how a record (e.g., customer) is stored.
- **Logical level:** describes what data stored in database, and the relationships among the data. DBA, who decides what information to keep in the database, use the logical level of abstraction.
- **View level:** describe only part of database. application programs hide details of data types. Complexity remain due to variety of information stored. Views can also hide information (such as an employee’s salary) for security purposes.

View of Data



An architecture for a database system

Database Languages

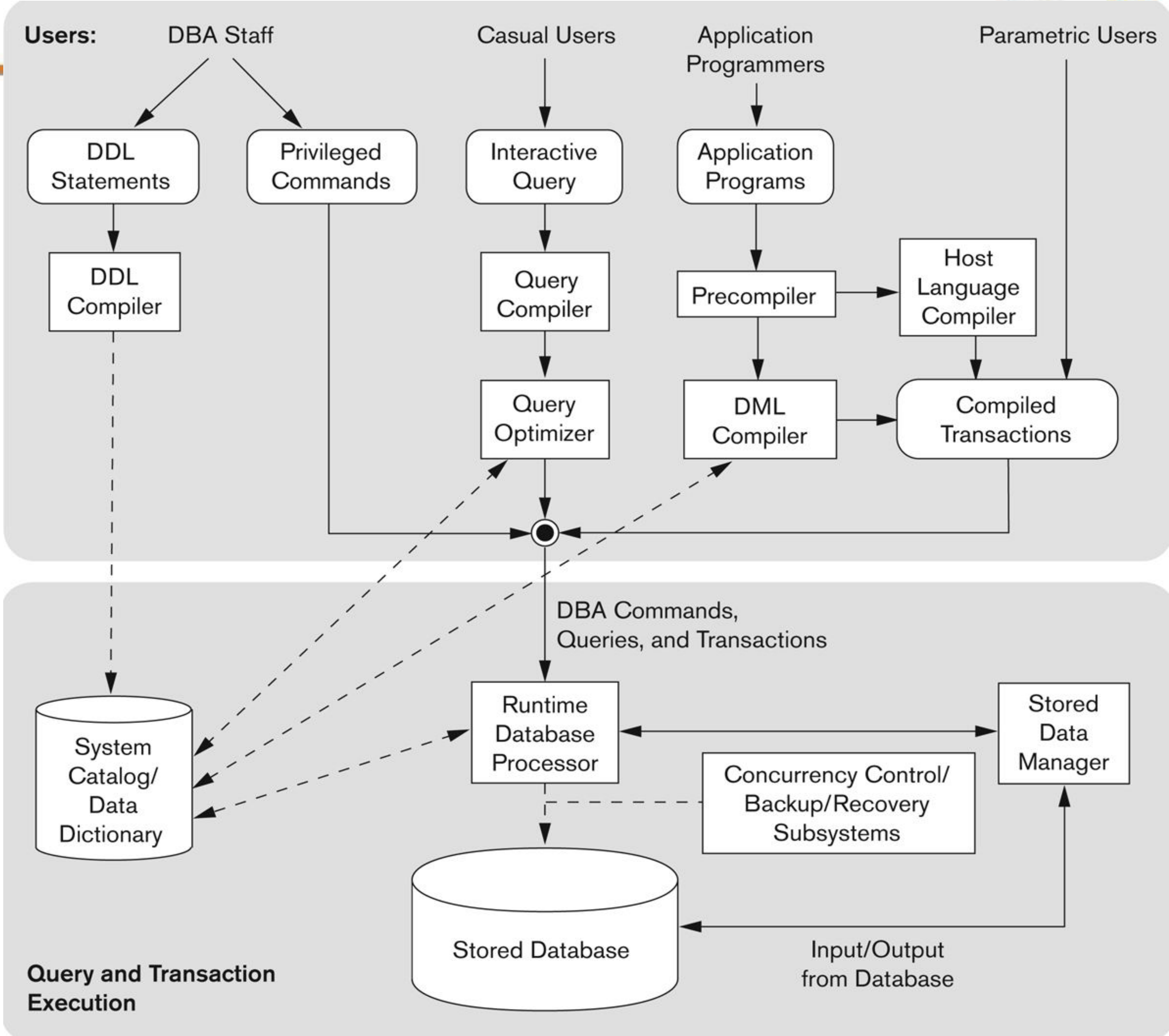
Database languages are used to create and maintain database on computer.

- **Data Definition Language(DDL):** It is a language that allows user to define data and their relationship to other types of data.
  - CREATE
  - ALTER
  - DROP
  - TRUNCATE
  - RENAME
- **Data Manipulation Language(DML):**It provides a set of operations to support the basic data manipulation operations on the data held in databases. It allows user to insert, update, delete and retrieve data from the database.
  - DELETE
  - INSERT
  - SELECT
  - UPDATE

Database Languages

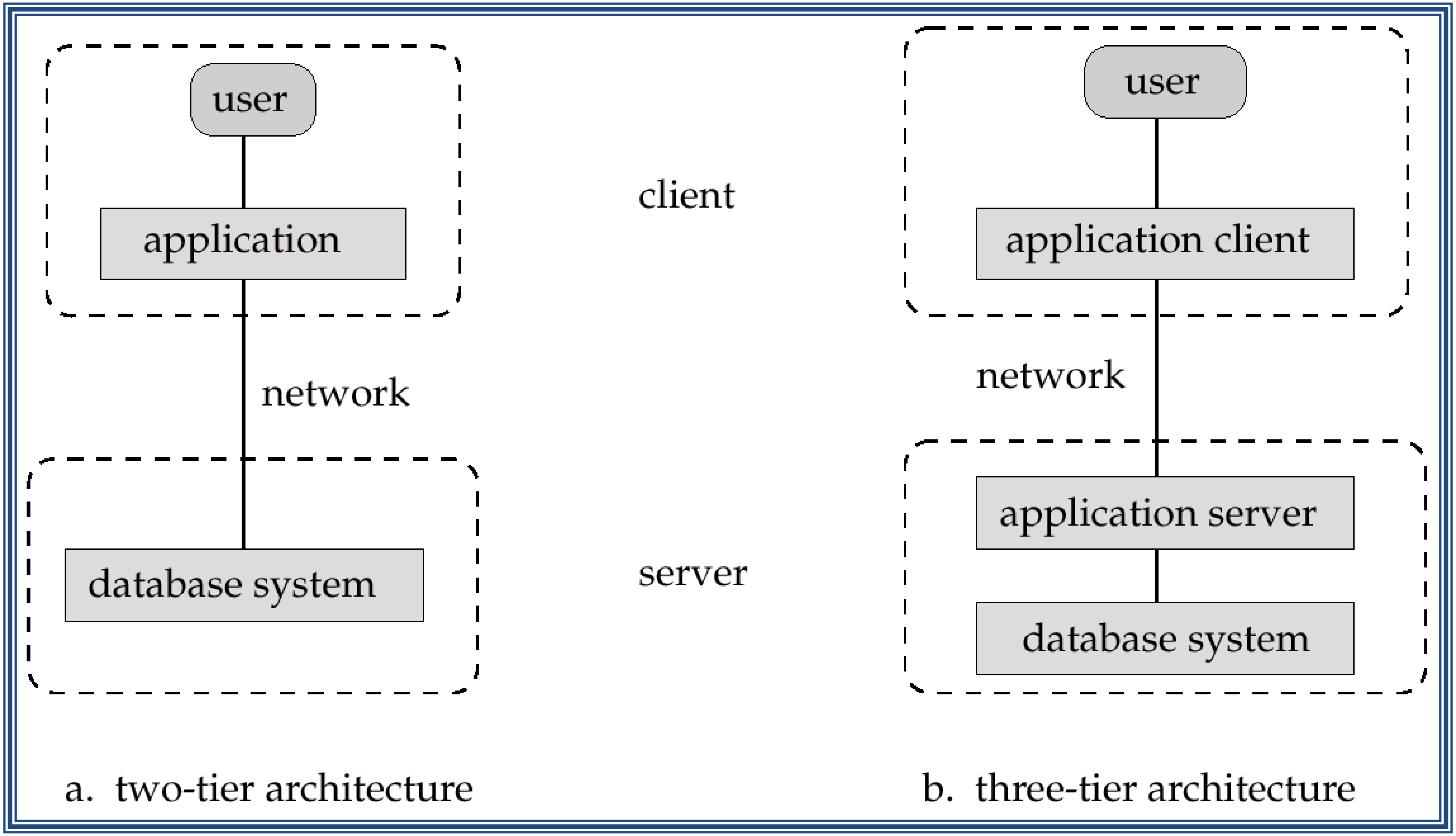
- **Data Control Language(DCL):** DCL statements control access to data and the database
  - GRANT
  - REVOKE
  - COMMENT
- **Transaction Control Language(TCL):** TCL statements manage the change made by DML statements, and group DML statements into transactions
  - COMMIT
  - ROLLBACK
  - SAVEPOINT
  - SET TRANSACTION

Structure and Components of DBMS

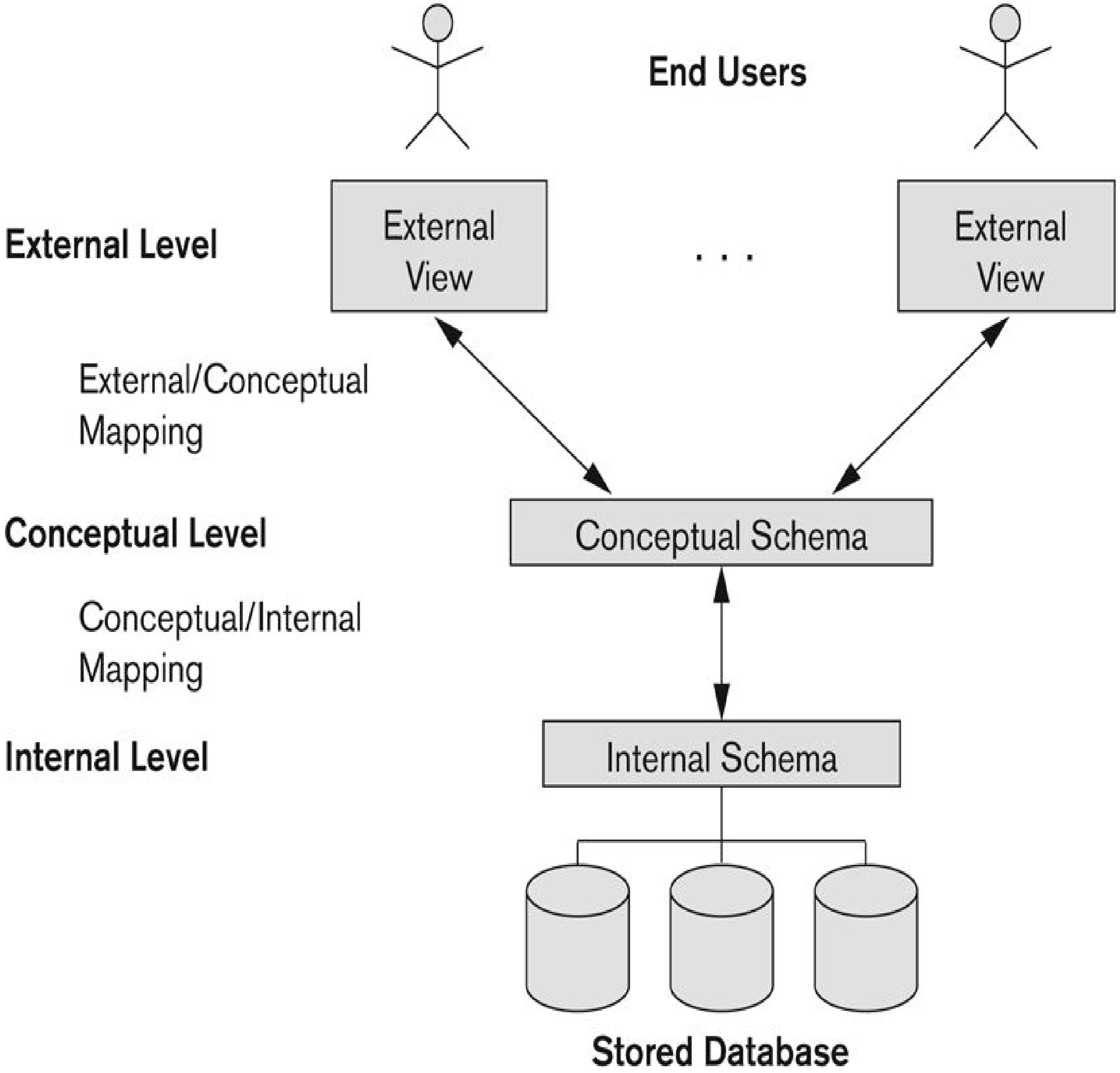

Users:
DBA Staff
Casual Users
Application Programmers
Parametric Users
DDL Statements
Privileged Commands
Interactive Query
Application Programs
DDL Compiler
Query Compiler
Precompiler
Host Language Compiler
Query Optimizer
DML Compiler
Compiled Transactions
DBA Commands, Queries, and Transactions
System Catalog/ Data Dictionary
Runtime Database Processor
Stored Data Manager
Concurrency Control/ Backup/Recovery Subsystems
Stored Database
Input/Output from Database
Query and Transaction Execution

Database Architecture

Two-tier and three-tier architecture



Three-tier architecture



- **External or View level:** It is the users' view of the database. This level describes that part of the database that is relevant to each user.
  - For example, one user may view dates in the form (day, month, year), while another may view dates as (year, month, day).
- **Conceptual or logical level:** It is the community view of the database. This level describes what data is stored in the database and the relationships among the data.
- It represents:
  - All entities, their attributes, and their relationships;
  - The constraints on the data;
  - Security and integrity information.
- **Internal or storage level:** It is the physical representation of the database on the computer. This level describes how the data is stored in the database.

Data Independence-Achievement of Layered Architecture of DBMS

- Two kinds of data independence:
  - – Logical data independence
  - – Physical data independence

Data Independence

- **Logical Data Independence**: The capacity to change the conceptual schema without having to change the external schemas and their application programs.
- **Physical Data Independence**: The capacity to change the internal schema without having to change the conceptual schema.

Data Independence

- The processes of transforming requests and results between the levels are called **mappings.**
- When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are *unchanged*. Hence, the application programs need not be changed since they refer to the external schemas.

Data Independence

- The processes of transforming requests and results between the levels are called **mappings.**
- When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are *unchanged*. Hence, the application programs need not be changed since they refer to the external schemas.