COURSE NAME: DBMS
COURSE CODE:23AD2102A

Topic: **DBMS ARCHITECTURE**

Session - 3

AIM OF THE SESSION

To familiarize students with the basic concept

- ❖ Three level architecture of DBMS, Data Independence and DBMS languages

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

- ❖ DBMS Three-Level Architecture
- ❖ Data Independence
- ❖ Database Languages

LEARNING OUTCOMES

At the end of this session, you should be able to:

- ❖ Know the three level architecture of DBMS
- ❖ Know the Data Independence and different levels
- ❖ Different types of DBMS languages

SESSION INTRODUCTION

DBMS – Three Level Architecture

- A database management system that provides three level of data is said to follow three-level architecture .
  - External level (i.e. External Schema)
  - Conceptual level (i.e. Conceptual Schema)
  - Internal level (i.e. Internal Schema)

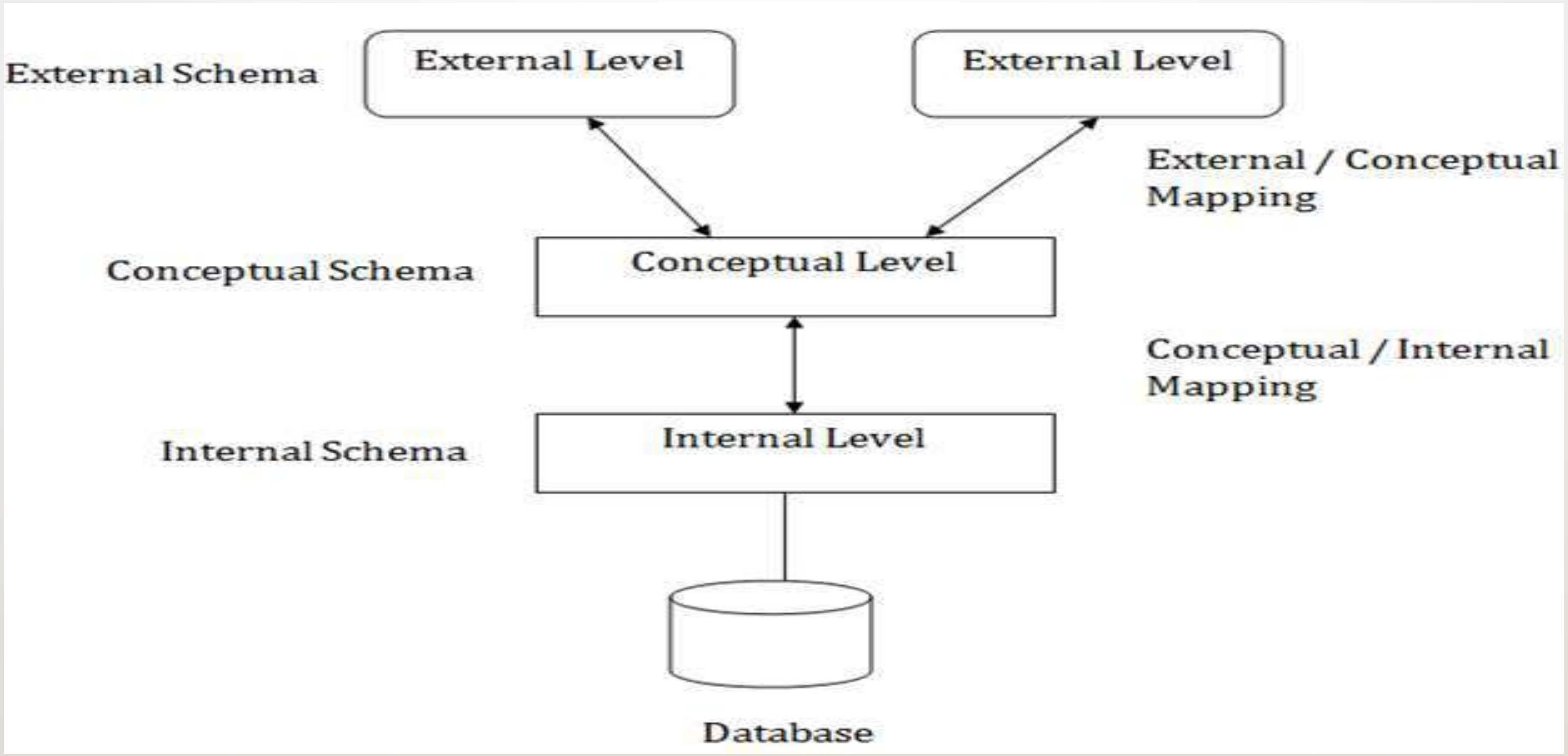
Data Independence

Data independence is the ability to modify the schema without effecting the programs and the application to be rewritten.

DBMS Languages

- A DBMS has appropriate languages and interfaces to express database queries and updates.
- Database languages can be used to read, store and update the data in the database.

DBMS Architecture



External Level

- The external level is at the highest level of database abstraction . At this level, there will be many views define for different users requirement. A view will describe only a subset of the database. Any number of user views may exist for a given global or sub schema.
- For example , each student has different view of the time table. The view of a student of BTech (CSE) is different from the view of the student of BTech(ECE).Thus this level of abstraction is concerned with different categories of users.
- Each external view is described by means of a schema called sub schema or external schema.

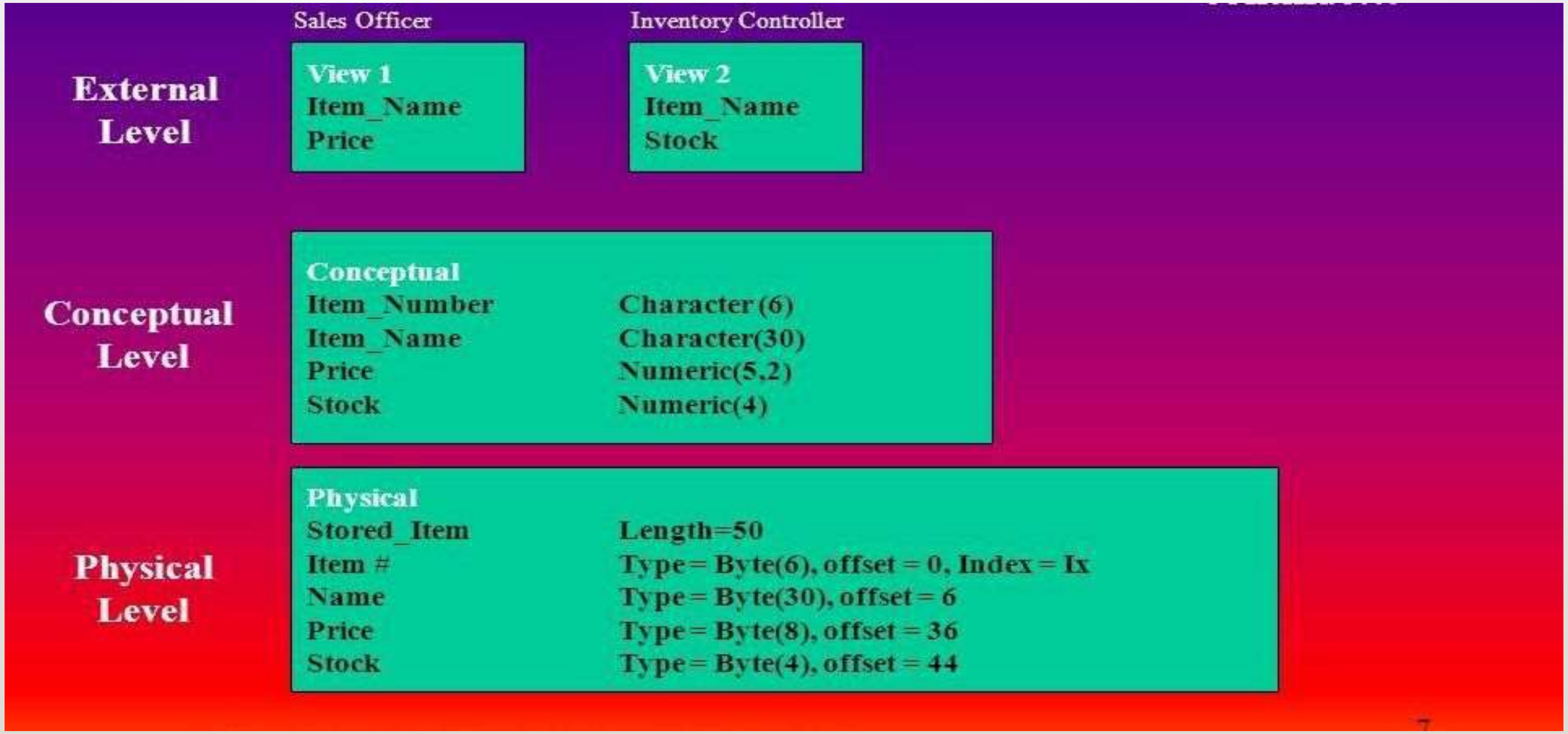
Conceptual Level

- At this level of database abstraction all the database entities and the relationships among them are included . One conceptual view represents the entire database . This conceptual view is defined by the conceptual schema.
- The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations and constraints.
- It describes all the records and relationships included in the conceptual view . There is only one conceptual schema per database . It includes feature that specify the checks to relation data consistency and integrity.

INTERNAL LEVEL

- It is the lowest level of abstraction closest to the physical storage method used . It indicates how the data will be stored and describes the data structures and access methods to be used by the database .The internal view is expressed by internal schema.
- The following aspects are considered at this level:
  - Storage allocation eg: B-tree, hashing.
  - Access paths eg. specification of primary and secondary keys, indexes etc.
  - Miscellaneous eg. Data compression and encryption techniques, optimization of the internal structures.

EXAMPLE FOR EACH LEVEL



DATA INDEPENDENCE

- Data independence is the ability to modify the schema without effecting the programs and the application to be rewritten.
- Data is separated from the programs, so that the changes made to the data will not effect the program execution and the application.
- **The main purpose of the three levels of data abstraction is to achieve data independence.** If the database changes and expands over time, it is very important that the changes in one level should not affect the data at other levels of the database. This would save time and cost required when changing the database.

LEVELS OF DATA INDEPENDENCE

- There are two levels of data independence based on three levels of abstraction. These are as follows −
  - ✓ Physical Data Independence
  - ✓ Logical Data Independence

Physical Data Independence

- Physical data independence is the ability to modify the physical scheme without making it necessary to rewrite application programs. Such modifications include changing from unblocked to blocked record storage, or from sequential to random access files.

Logical Data Independence

- Logical data independence is the ability to modify the conceptual scheme without making it necessary to rewrite application programs. Such a modification might be adding a field to a record; an application program's view hides this change from the program.

SESSION DESCRIPTION

DATABASE SYSTEM ENVIRONMENT

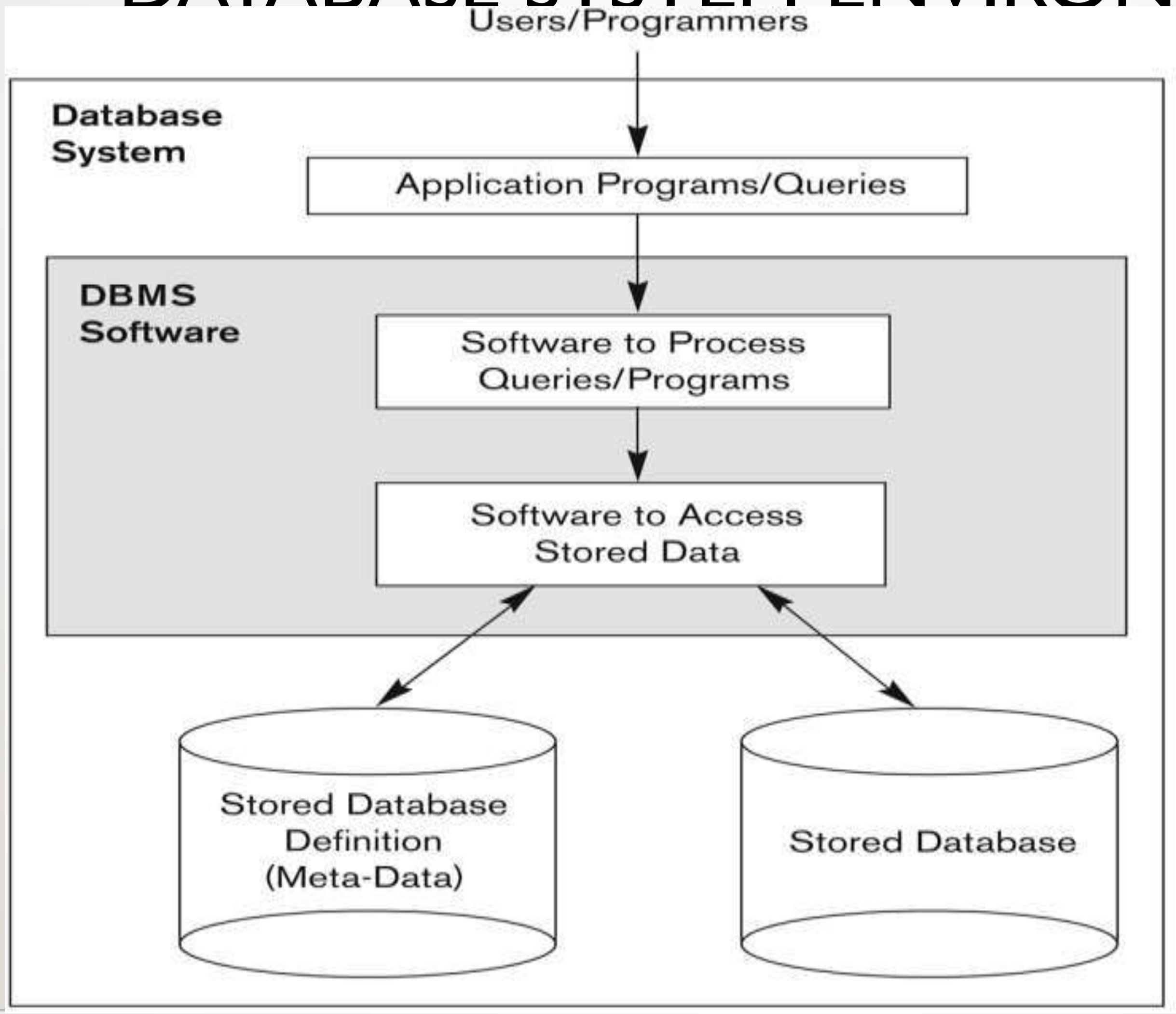

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

Database System Environment

Database system refers to an organization of components that defines and regulate the following:

- Collection
- Storage
- Management
- Use of data within the environment

Database System Environment

The Database System involves the following components:

- Hardware
- Software
- People
- Procedure
- Data.

TYPICAL DBMS FUNCTIONALITY

- ***Define** **a particular database*** in terms of its data types, structures, and constraints
- ***Construct** **or Load the initial database contents*** on a secondary storage medium
- ***Manipulating** **the database***:
    - **Retrieval**: Querying, generating reports
    - **Modification**: Insertions, deletions and updates to its content
    - **Accessing** the database through Web applications
- ***Processing** **and** ***Sharing** **by a set of concurrent users*** and application programs – yet, keeping all data valid and consistent

TYPICAL DBMS FUNCTIONALITY

- Other features:
  - **Protection or Security** measures to prevent unauthorized access
  - **“Active” processing** to take internal actions on data
  - **Presentation and Visualization** of data
  - **Maintaining the database** and associated programs over the lifetime of the database application
    - Called database, software, and system maintenance

DATABASE SYSTEM STRUCTURE

- A database system is partitioned into modules that deal with each of the responsibilities of the overall system.
- The functional components of a database system can be broadly divided into the storage manager and the query processor components.

DATABASE SYSTEM STRUCTURE

- A database system is partitioned into modules that deal with each of the responsibilities of the overall system.
- The functional components of a database system can be broadly divided into the storage manager and the query processor components.

SESSION DESCRIPTION

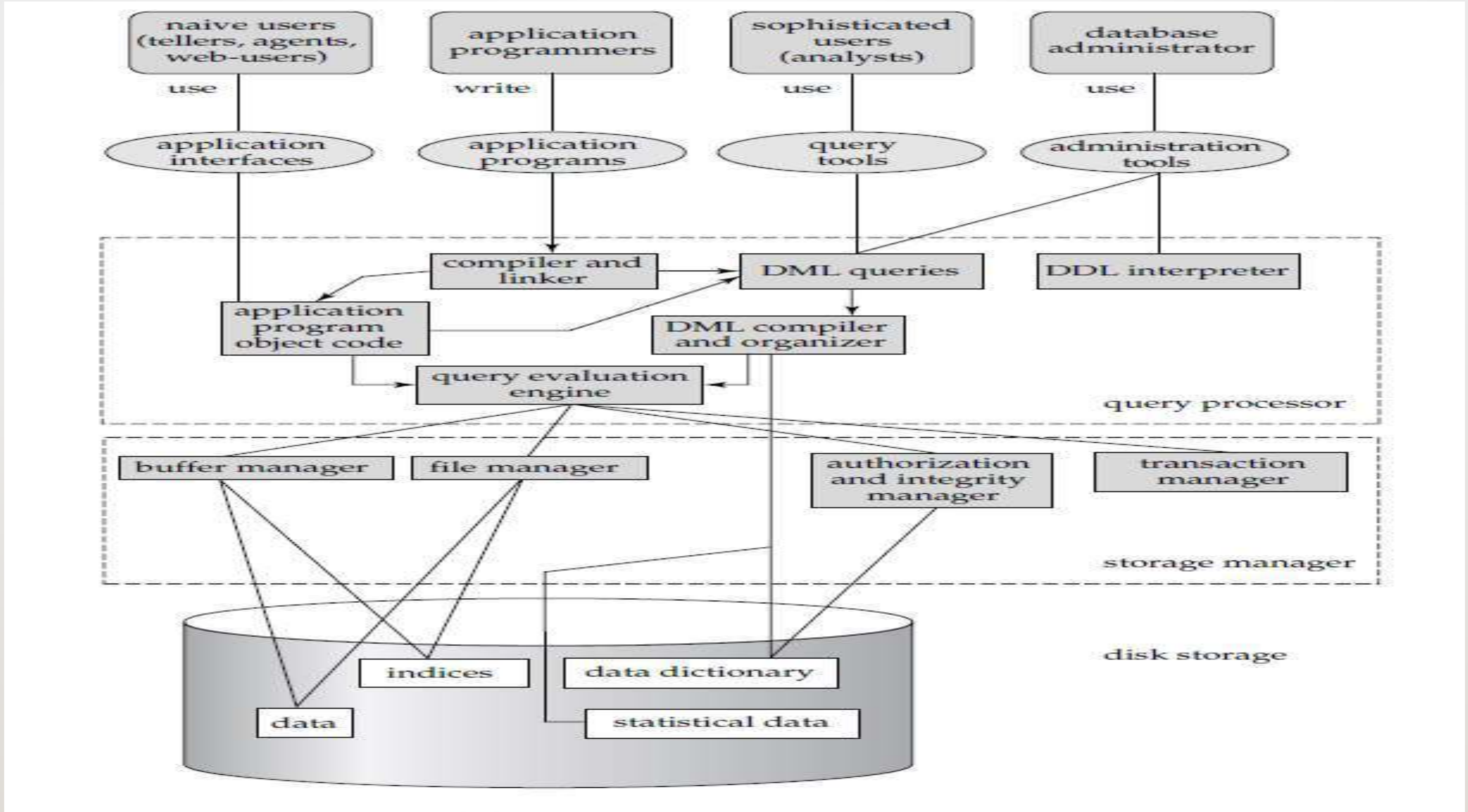


Figure 1.4 System structure.

SESSION DESCRIPTION

STORAGE MANAGER

- A *storage manager* is a program module 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.
- The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system.
- The storage manager translates the various DML statements into low-level file-system commands.
- Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.

THE STORAGE MANAGER COMPONENTS INCLUDE

- **Authorization and integrity manager**, which tests for the satisfaction of integrity constraints and checks the authority of users to access data.
- **Transaction manager**, which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.
- **File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
- **Buffer manager**, which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory.
- The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory

STORAGE MANAGER

- The storage manager implements several data structures as part of the physical system implementation
- **Data files**, which store the database itself.
- **Data dictionary**, which stores metadata about the structure of the database, in particular the schema of the database.
- **Indices**, which provide fast access to data items that hold particular values

THE QUERY PROCESSOR

- The query processor components include
- **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.
- A query can usually be translated into any of a number of alternative evaluation plans that all give the same result.
- 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.

DATABASE USERS

- Actors on the scene
  - **Database administrators:**
    - Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.
  - **Database Designers:**
    - Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

CATEGORIES OF END-USERS

- Actors on the scene (continued)
  - **End-users:** They use the data for queries, reports and some of them update the database content. End-users can be categorized into:
    - **Casual**: access database occasionally when needed
    - **Naïve** or Parametric: they make up a large section of the end-user population.
      - They use previously well-defined functions in the form of "canned transactions" against the database.
      - Examples are bank-tellers or reservation clerks who do this activity for an entire shift of operations.

SESSION DESCRIPTION

CATEGORIES OF END-USERS

- **Sophisticated:**
  - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
  - Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone:**
  - Mostly maintain personal databases using ready-to-use packaged applications.
  - An example is a tax program user that creates its own internal database.
  - Another example is a user that maintains an address book

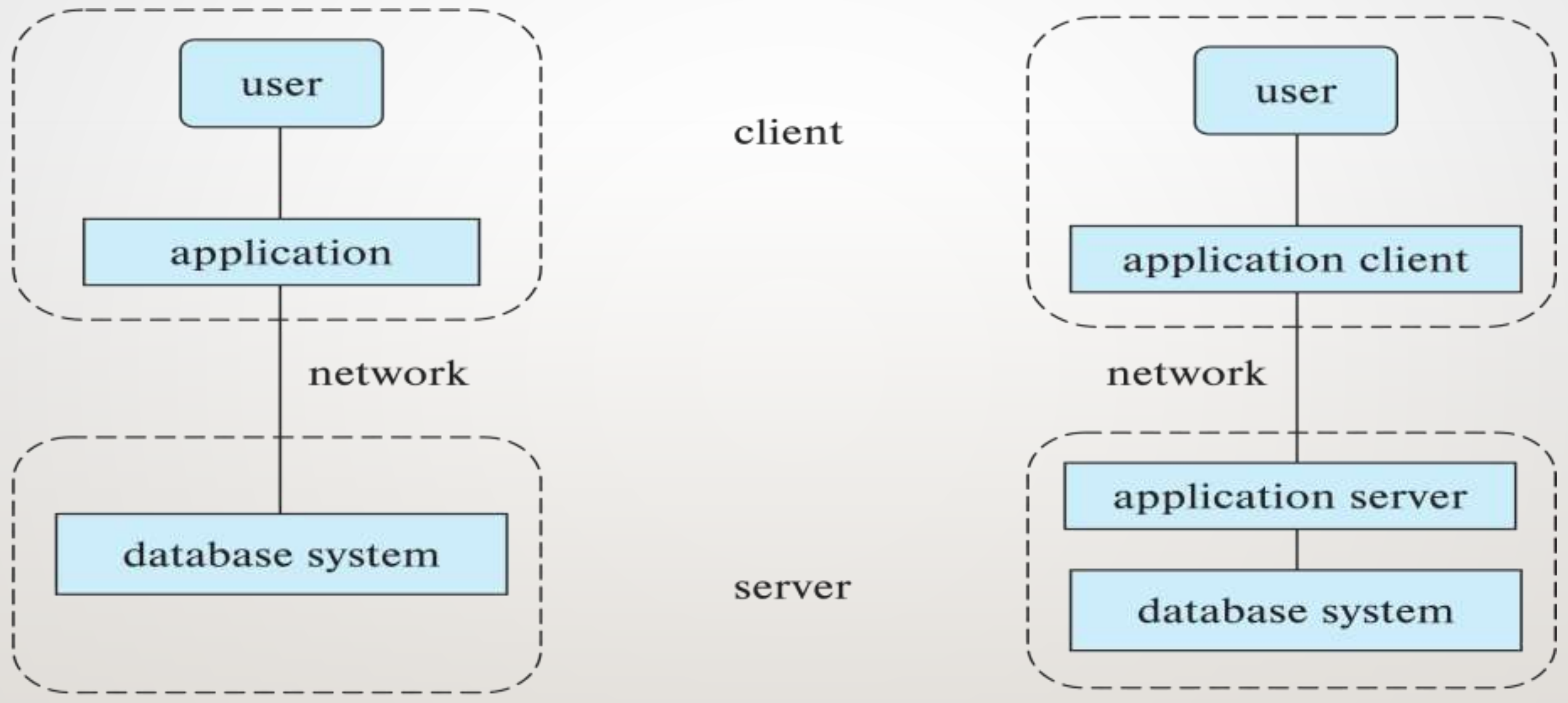
DATABASE ADMINISTRATOR

A person who has central control over the system is called a **database administrator (DBA).**

Functions of a DBA include:

- Schema definition
- Storage structure and access-method definition
- Schema and physical-organization modification
- Granting of authorization for data access
- Routine maintenance
- Periodically backing up the database
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required
- Monitoring jobs running on the database

TWO-TIER AND THREE-TIER ARCHITECTURES



(a) Two-tier architecture (b) Three-tier architecture

SESSION DESCRIPTION

Two-Tier Database Architecture

- A two-tier architecture either buries the application logic inside a UI (User interface) of the concerned client or the database of a given server or both of them. One can feasibly locate the user's system interface in the desktop environment of the user with a two-tier architecture of a server/client.
- The DB management system usually resides in a server that hosts more powerful machines capable of providing service to many clients.

SESSION DESCRIPTION

Three-Tier Database Architecture

The process or application logic in a three-tier architecture resides in the very middle-tier. It stays totally separated from the UI and the data. These types of systems are comparatively more flexible, robust, and scalable. Added to it, they can also easily integrate data and info from various different sources. In a three-tier architecture, the middle-tier exists between the client environment of the user system interface and the server environment of the database management. One can implement this middle tier in a number of ways, such as application servers, message servers, or processing monitors.

SUMMARY

- The architecture can be designed as centralized, decentralized, or hierarchical.
- The three levels present in this architecture are Physical level, Conceptual level and External level.

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.

ACTIVITIES/ CASE STUDIES/ IMPORTANT FACTS RELATED TO THE SESSION

- ❖Similar to a client-server architecture.
- ❖ Faster access, Easier to maintain and can handle multiple users simultaneously.
- ❖Used when we wish to access DBMS via applications and APIs.
- ❖Has scalability and security issues because of direct client-server connection.

SELF-ASSESSMENT QUESTIONS

1. Which is the lowest level of abstraction that describes how the data are actually stored?

(a) Physical
(b) Conceptual
(c) View
(d) User

Ans: a

2. The ______ can be used to explain data independence.

(a) One-schema architecture
(b) Two-schema architecture
(c) Three-schema architecture
(d) Four-schema architecture

Ans: c

SELF-ASSESSMENT QUESTIONS

3.Database languages can be used to ____________________ the data in the database.

(a) Read

(b) Store

(c) Update

(d) All of the above

ANS: d

4. What is TRUE about DDL?

(a) Database structure or patterns are defined by it.
(b) A schema, table, index, constraint, etc. are created in a database using it.
(c) A database skeleton can be created with DDL statements.
(d) All of the above

Ans: d

TERMINAL QUESTIONS

1. Describe the Three-level architecture of DBMS.

2. List out five responsibilities of a database management system. For each responsibility, explain the problems that would arise if the responsibility were not discharged.

3. Analyze the difference between physical and logical data independence.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Reference Books:

1. Database System Concepts, Sixth Edition, Abraham Silberschatz, Yale University Henry, F. Korth Lehigh University, S. Sudarshan Indian Institute of Technology, Bombay.
2. Fundamentals of Database Systems, 7th Edition, RamezElmasri, University of Texas at Arlington, Shamkant B. Navathe, University of Texas at Arlington.

Sites and Web links:

1. https://nptel.ac.in/courses/106105175
2. https://beginnersbook.com/2018/11/dbms-three-level-architecture/

THANK YOU

Team - DBMS