Database Management System (MCA-112)
MCA 1st Sem (2020-21)

D. S. Singh
Associate Professor
Department of ITCA
MMMUT Gorakhpur
Email: dssitca@mmmut.ac.in

Database Management System

Course Objectives

- know and understand the fundamental concepts, techniques, and terminology of the database management system.
- know and understand the basic syntax, semantics, and pragmatics of SQL & PL/SQL.
- can analyze problems and apply DBMS concepts and techniques to develop appropriate programs to solve the problems,

Database Management System

- can evaluate alternative database designs to determine which are better according to selected criteria.
- know and understand the basic features of database transactions and concurrency control.
- are able to reason about and manipulate concurrency control techniques.

Database Management System

Learning Outcomes

- ❖ list and define the fundamental concepts of DBMS.
- ❖ manually execute a simple database design a transaction over it.
- ❖ manually infer the type of a given database transaction.
- ❖ implement algorithms and data structures as database transaction.
- ❖ design databases that are modular and have reusable components.
- ❖ explain concurrency control and concurrent transactions.

Unit-I

Introduction

An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree

Unit-II

Relational data Model and Language

Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus

Introduction to SQL:

Characteristics of SQL. Advantage of SQL. SQL data types and literals. Types of SQL commands. SQL operators and their procedure. Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL. induction over

Unit-III

Data Base Design & Normalization

Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Unit-IV

Transaction Processing Concepts

Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

Concurrency Control Techniques

Concurrency control, locking techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

Books & References

- Date C J, “An Introduction To Database System”, Addision Wesley.
- Korth, Silberchatz, Sudarshan, “Database Concepts”, McGraw Hill.
- Elmasri, Navathe, “Fundamentals Of Database Systems”, Addision Wesley.
- Leon & Leon, “Database Management System”, Vikas Publishing House
- Bipin C. Desai, “An introduction to Database Systems”, Galgotia Publication.
- Majumdar & Bhattacharya, “Database Management System”, TMH.
- Ramakrishnan, Gehrke, “Database Management System”, McGraw Hill.
- Kroenke, “Database Processing: Fundamentals, Design and Implementation”, Pearson Education.
- Maheshwari Jain, “DBMS: Complete Practical Approach”, Firewall Media, New Delhi.

Overview of Database Management System

- **Data** is a collection of facts and figures that can be processed to produce the information.
- Processed data is known as **information.**
- **Database** is a collection of inter-related data.
- Database Management System = Database + A set of program to access the data/information from database.
- A **DBMS** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

Characteristics of DBMS

- **Real-world entity**
- **Relation-based tables**
- **Isolation of data and application**
- **Less redundancy**
- **Consistency**
- **Query Language**
- **ACID Properties:** ***Atomicity, Consistency, Isolation and Durability***
- **Multiuser and Concurrent Access**
- **Multiple views**
- **Security**

Users of DBMS

- **Administrator**
- **Designers**
- **End users**

Advantages of DBMS

- **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.

- **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.

- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.

Advantages of DBMS

- **Reduce time:** It reduces development time and maintenance need.

- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.

- **Multiple User Interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces.

Disadvantages of DBMS

- **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.
- **Size:** It occupies a large space of disks and large memory to run them efficiently.
- **Complexity:** Database system creates additional complexity and requirements.
- **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to power failure or database corruption then the data may be lost forever.

Database Management Vs File Management System

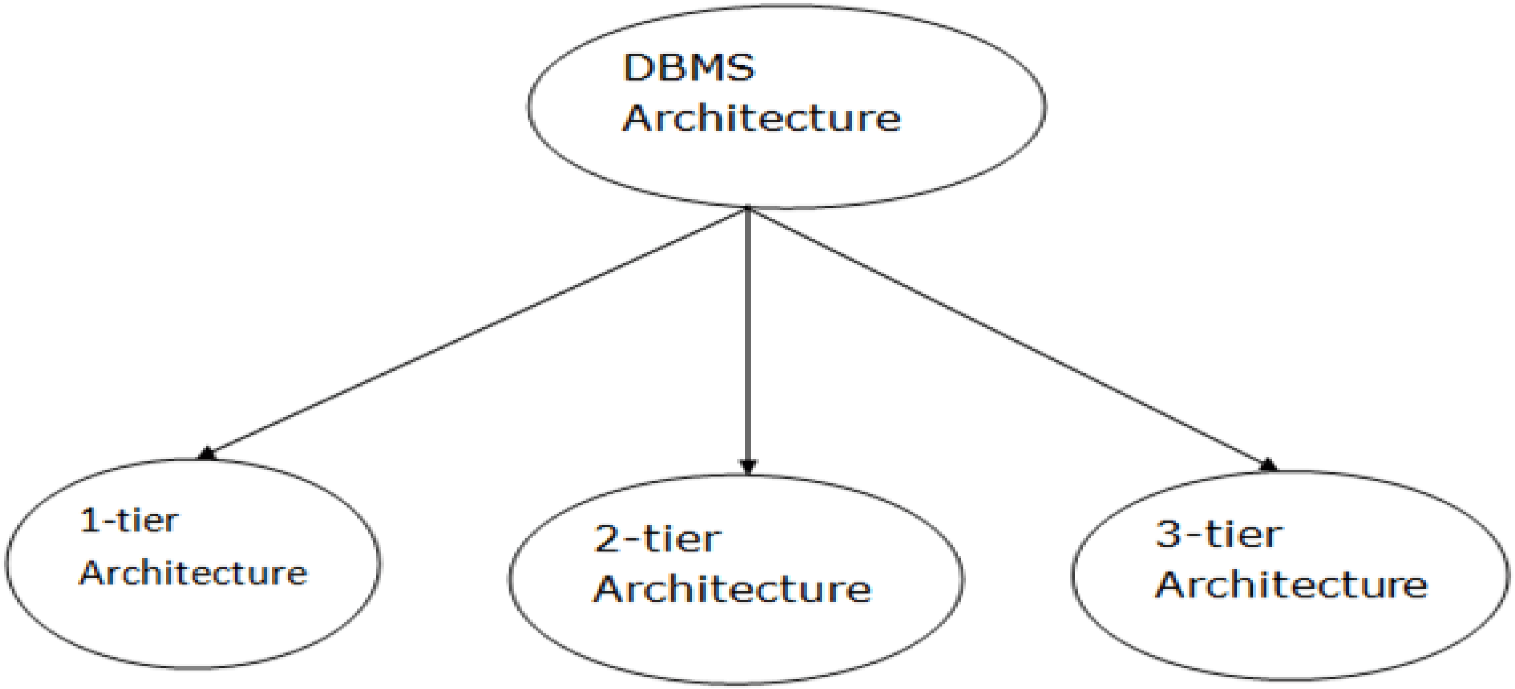
File Management System	Database Management System
Small systems like C++ or Java program.	Large systems like Oracle or Sybase.
Relatively cheap	Relatively expensive
Have simple structure	Have complex structure
Needs very little preliminary design.	Needs vast preliminary design
Not secure	More secure
Often single user- oriented.	Multiple user-oriented.
Have isolated data and simple backup/recovery mechanism	Have shared data and complex & sophisticated backup/recovery mechanism.

Database System Concepts and Architecture

DBMS Architecture

- 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.

Types of DBMS Architecture



1-Tier Architecture

- In this architecture, the database is directly available to the user. It means the user can directly have the DBMS and use it.

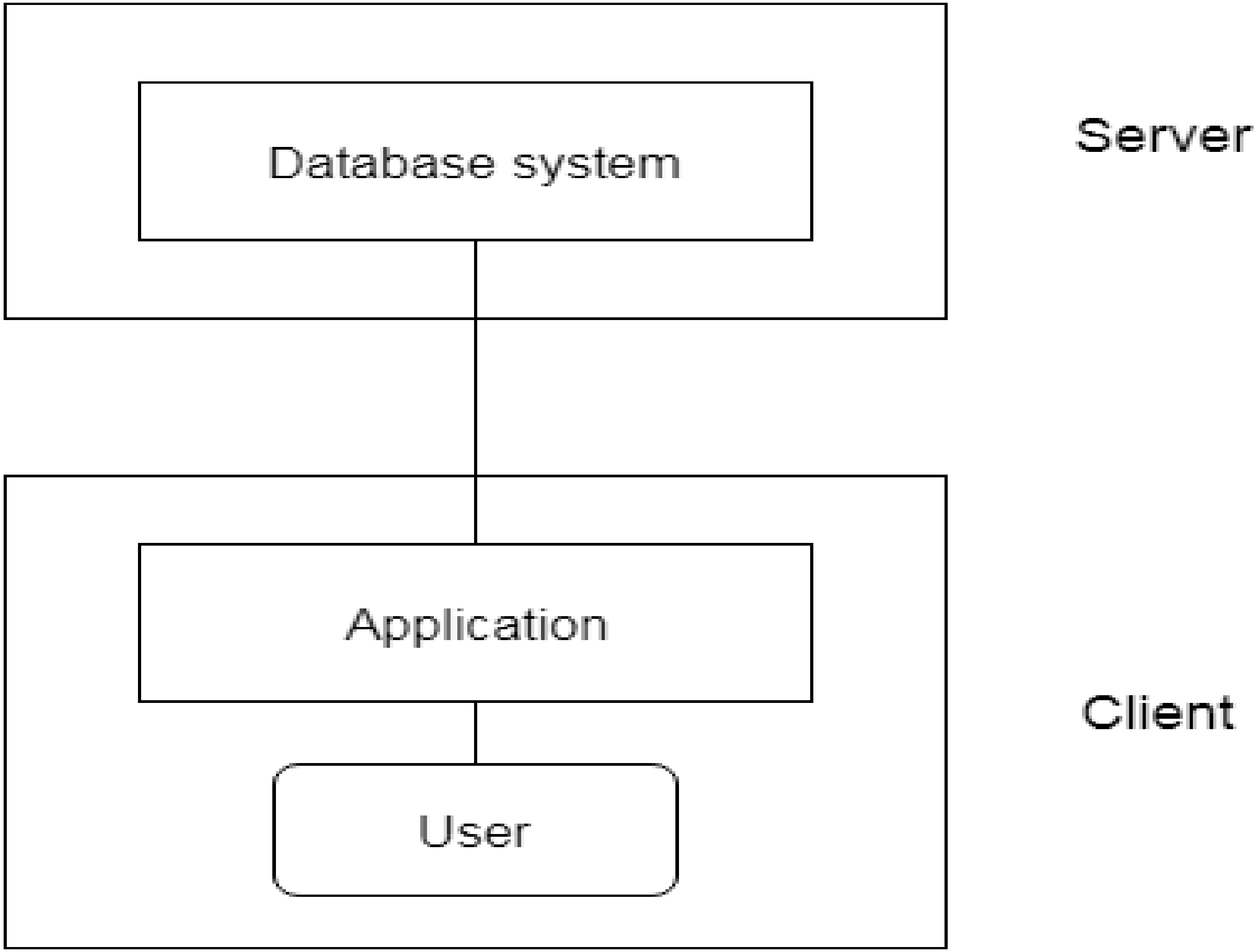
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.

- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-Tier Architecture

- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: ODBC, JDBC are used.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.

2-Tier Architecture

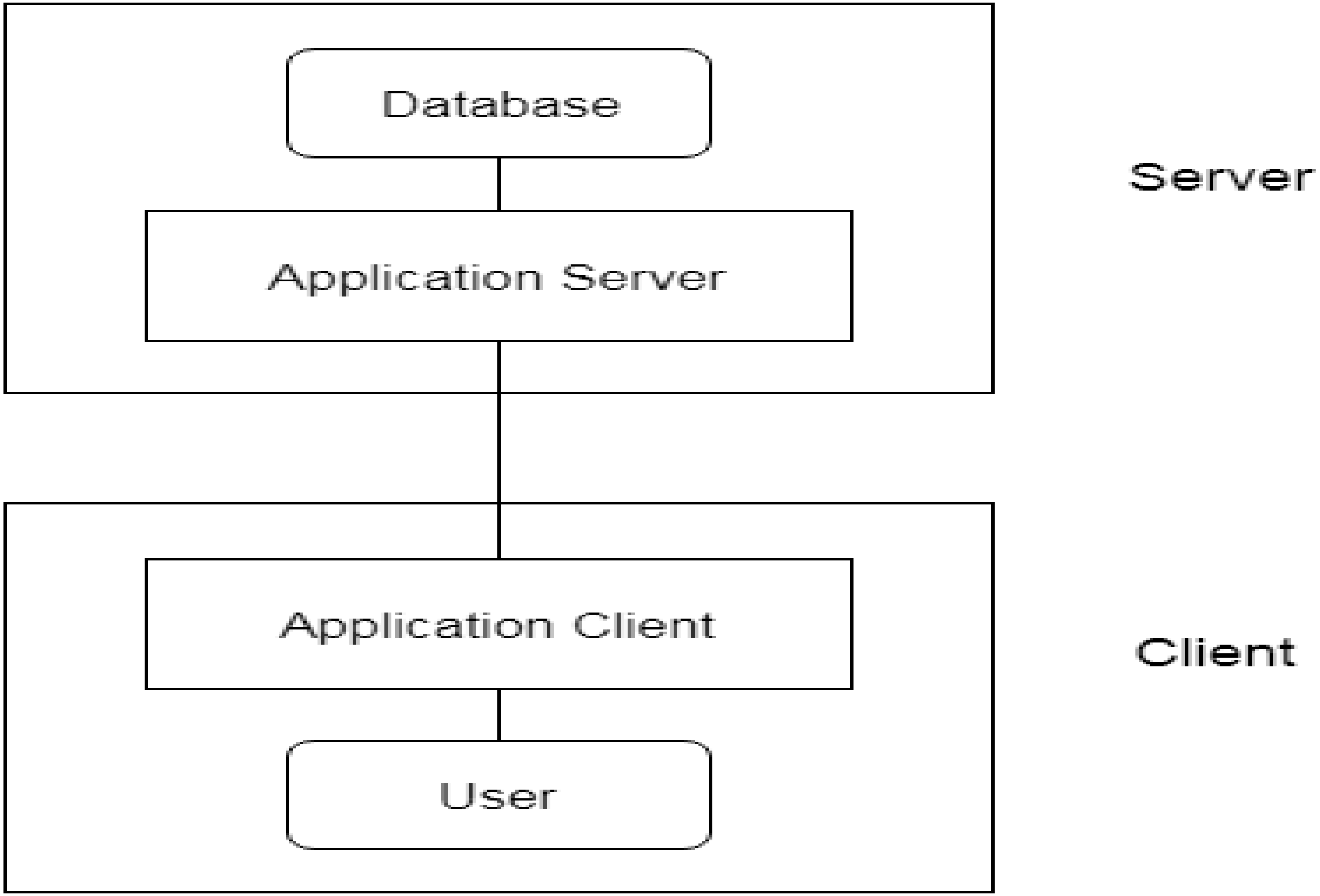


3-Tier Architecture

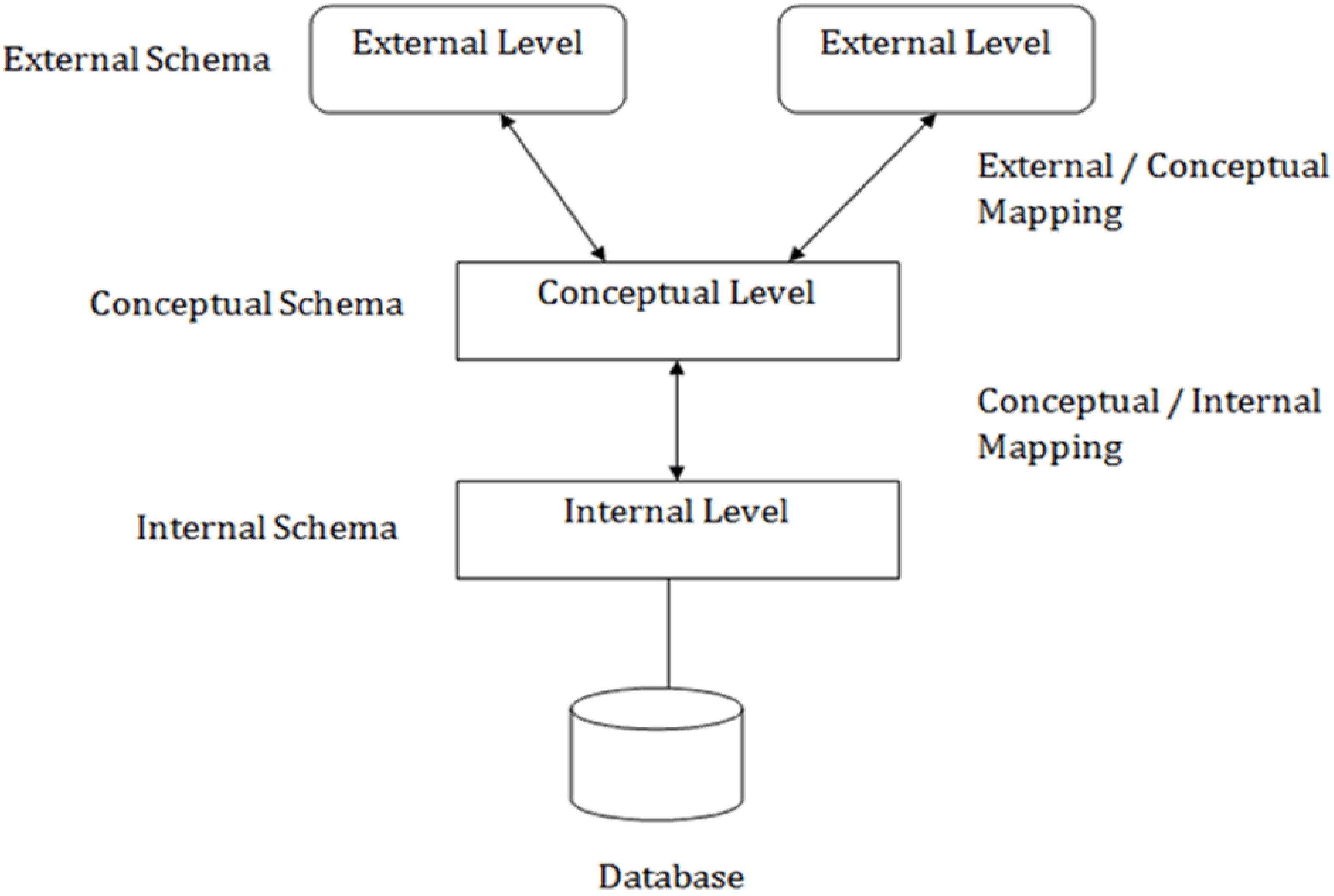
- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.

- The 3-Tier architecture is used in case of large web

3-Tier Architecture



Three Level Architecture of DBMS


External Schema

External Level

External Level

External / Conceptual Mapping

Conceptual Schema

Conceptual Level

Conceptual / Internal Mapping

Internal Schema

Internal Level

Database

Three Level Architecture of DBMS

- It is also known as three schema architecture of DBMS.
- Mapping is used to transform the request and response between various database levels of architecture.
- Mapping is not good for small DBMS because it takes more time.
- In External/Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual/Internal mapping, DBMS transform the request from the conceptual to internal level.

Internal Level

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored on the disk.
- Physical level is used to describe complex low-level data structures in detail.

Conceptual Level

- It describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- It also describes the structure of the whole database.
- It also describes what data are to be stored in the database and what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

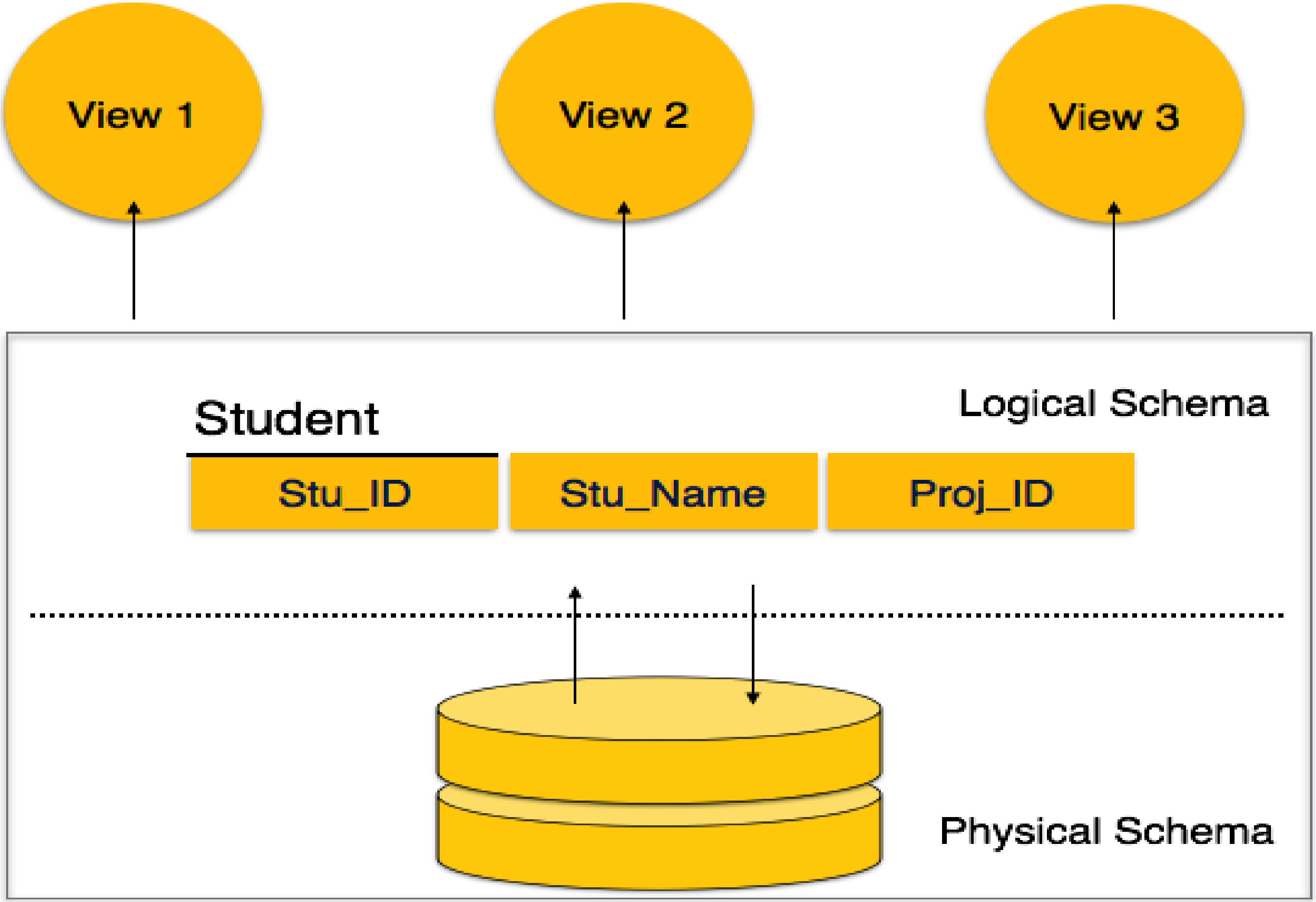
External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

Database Schema

- Skeleton that represents the logical view of the entire database.
- Defines how the data is organized and how the relations among them are associated.
- Formulates all the constraints that are to be applied on the data.
- Defines its entities and the relationship among them.
- Contains a descriptive detail of the database, which can be depicted by means of schema diagrams.
- It’s the database designers who design the schema to help programmers to understand the database and make it useful.

Database Schema



Database Schema

A database schema can be divided broadly into two categories:

- Physical Database Schema: This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- Logical Database Schema: This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Database Instance

- Database schema is the skeleton structure of database.
- It is designed when the database doesn't exist at all.
- Once the database is operational, it is very difficult to make any changes to it.
- A database schema does not contain any data or information.
- A database instance is a state of operational database with data at any given time.

Database Instance

- A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed.
- If a database system is not multi-layered, then it becomes difficult to make any changes in the database system. Database systems are designed in multi-layers.

The collection of information stored in the database at a particular moment is called as an instance of the database.

Data Models

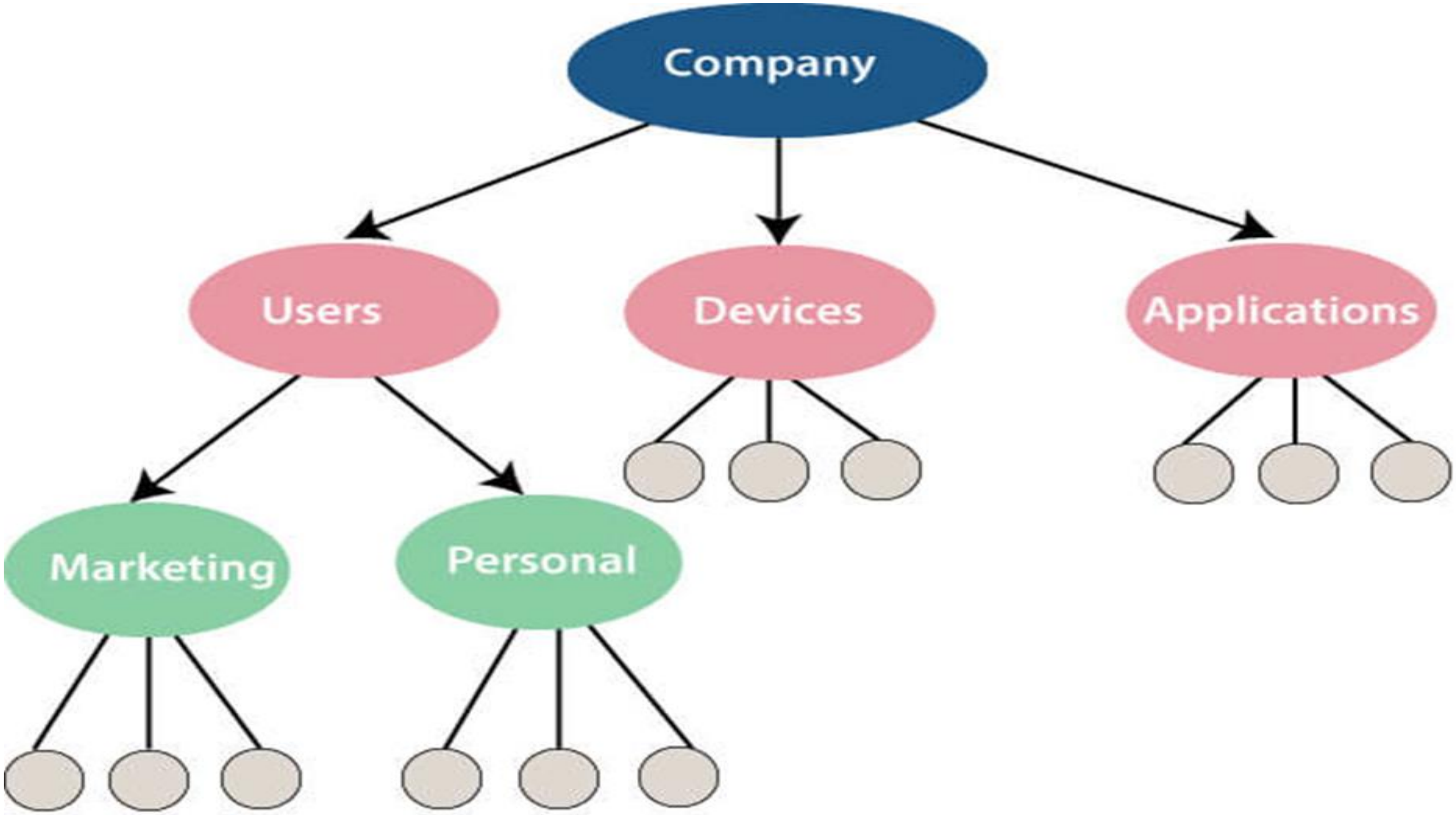
- Data models define how the logical structure of a database is modeled.
- Data Models are fundamental entities to introduce abstraction in a DBMS.
- Data models define how data is connected to each other and how they are processed and stored inside the system.
- The very first data model could be flat data-models, where all the data used are to be kept in the same plane.
- Earlier data models were not so scientific; hence they were prone to introduce lots of duplication and update anomalies.

Hierarchical Data Model

- In this model, files are related in a parent/child manner.
- Like file system, this model also had some limitations like complex implementation, lack structural independence.
- It can't easily handle a many-many relationship, etc.
- It is also called IMS (Information Management System).
- Below diagram represents Hierarchical Data Model. Small circle represents objects.

Hierarchical Data Model

Example of HDM



Network Data Model

- In this model, files are related as owners and members.

- **Network data model identified the following components:**

    - Network schema (Database organization)
    - Sub-schema (views of database per user)
    - Data management language (procedural)

- This model also had some limitations like system complexity and difficult to design and maintain.

Relational Data Model

- Relational Data Model has two main terminologies called instance and schema.
- The instance is a table with rows or columns.
- Schema specifies the structure like name of the relation, type of each column and name.
- This model uses some mathematical concept like set theory and predicate logic.
- During the era of the relational database, many models introduced such as object-oriented model, object-relational model, etc.

Data Independence

- A database system normally contains a lot of data in addition to users’ data. For example, it stores data about data, known as metadata, to locate and retrieve data easily.
- It is rather difficult to modify or update a set of metadata once it is stored in the database.
- But as a DBMS expands, it needs to change over time to satisfy the requirements of the users.
- If the entire data is dependent, it would become a tedious and highly complex job.

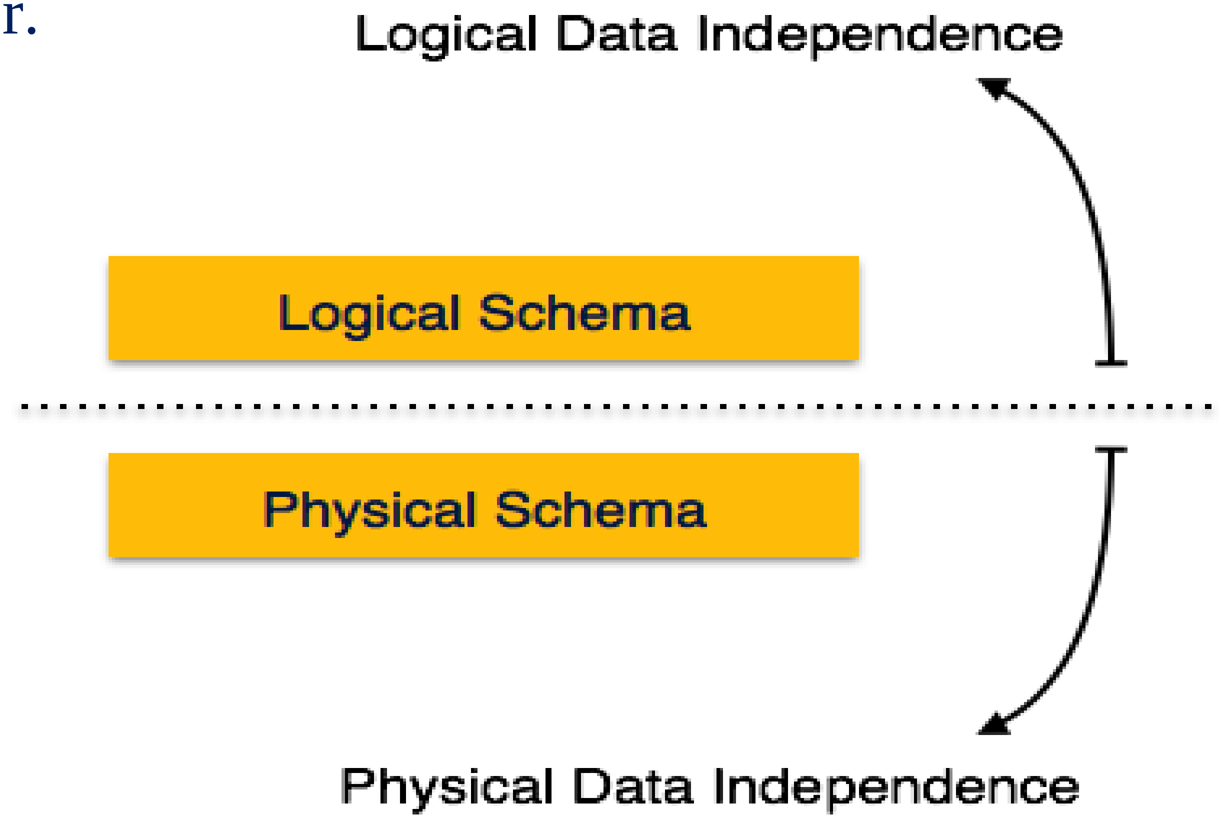
Data Independence

- The major objective of 3-level architecture of DBMS is to provide data independence which means that upper levels are unaffected by changes in lower levels.

- There are two types of data independence:
  - ✓ Physical Data Independence
  - ✓ Logical Data Independence

Data Independence

- Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.



Physical Data Independence

- It indicates that the physical storage structures or devices could be changed without affecting the conceptual schema.

- All the schemas are logical, and the actual data is stored in bit format on the disk.

- Physical data independence is the power to change the physical data without impacting the schema or logical data.

- For example, in case we want to change or upgrade the storage system itself − suppose we want to replace hard-disks with SSD − it should not have any impact on the logical data or schemas.

Logical Data Independence

- Logical data independence means that the conceptual schema (middle level) can be changed without affecting the existing external schemas.

- Logical data is data about database, that is, it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation.

- Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

References

1. Date C J, “An Introduction To Database System”, Addision Wesley.
2. Korth, Silberchatz, Sudarshan, “Database Concepts”, McGraw Hill.
3. https://www.javatpoint.com/dbms-tutorial
4. https://www.tutorialspoint.com/dbms/index.htm