R. C. Patel Institute of Technology, Shirpur

Department of Computer Engineering

Sub: Database Management System(PCCO3040T)

by

Prof. J. S. Sonawane

Teaching Scheme

Teaching Scheme	Examination Scheme
Lectures : 03 Hrs./week	Term Test : 15 Marks
Credits : 03	Teacher Assessment : 20 Marks
	End Sem Exam : 65 Marks
	Total Marks : 100 Marks

Course Objectives:

1. To learn and practice data modelling using the entity-relationship and develop database designs.

2. To understand the use of Structured Query Language (SQL) and learn SQL syntax.

3. To apply normalization techniques to normalize the database.

4. To understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.

5. To introduce principles and foundations of distributed databases, design issues, query processing and optimization.

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Illustrate the fundamentals of a database systems.	L2	Understand
CO2	Design and draw ER and EER diagram for the real-life problem.	L6	Create
CO3	Build relational model from conceptual model and formulate relational algebra queries.	L6	Create
CO4	Design and query database using SQL.	L6	Create
CO5	Analyze and apply concepts of normalization to relational database design and explain the concept of transaction, concurrency and recovery.	L3, L4, L2	Apply, Analyze, Understand
CO6	Summarize the concepts of distributed database.	L2	Understand

Course Contents

Unit-I Introduction to Database Concepts **03 Hrs.**

Introduction, Characteristics of Databases, File System v/s Database System, Users of Database System, Data Independence, DBMS System Architecture, Database Administrator.

Unit-II Entity–Relationship Data Model **08 Hrs.**

The Entity-Relationship (ER) Model: Entity Types: Weak and Strong Entity Sets, Entity Sets, Types of Attributes, Keys, Relationship Constraints: Cardinality and Participation. Extended Entity-Relationship (EER) Model: Generalization, Specialization and Aggregation.

Unit-III Relational Model and Relational Algebra **08 Hrs.**

Introduction to the Relational Model, Relational Schema and Concept of Keys, Mapping the ER and EER Model to the Relational Model. Relational Algebra: Unary and Set Operations, Relational Algebra Queries.

Unit-IV Structured Query Language (SQL) 09 Hrs.

Overview of SQL, Data Definition Commands, Data Manipulation Commands, Data Control Commands, Transaction Control Commands.

Integrity Constraints: Key Constraints, Domain Constraints, Referential Integrity, Check Constraints, Set and String Operations, Aggregate Function, Group By Clause, Having Clause. Views in SQL, Joins, Nested and Complex Queries. Introduction to PL/SQL

Unit-V Relational Database Design 10 Hrs.

Pitfalls in Relational-Database Designs, Concept of Normalization, Functional Dependencies, First Normal Form, 2NF, 3NF, BCNF. Transactions Management and Concurrency: Transaction Concept, Transaction States, ACID Properties, Concurrent Executions, Serializability – Conflict and View, Concurrency Control: Lock-Based, Timestamp-Based Protocols. Recovery System: Introduction to Recovery System.

Unit-VI Distributed Database 04 Hrs.

Introduction to Distributed Database, Features of DDBS, Design Issues in DDBS, Distributed Database Design Concept, Objectives, Data Fragmentation, Transparencies in Distributed Database Design.

Text Books:

1. **Korth, Silberchatz, Sudarshan, Database System Concepts, 6th Edition, McGraw – Hill.**

2. **Elmasri and Navathe, Fundamentals of Database Systems, 5th Edition, Pearson Education.**

3. Peter Rob and Carlos Coronel, Database Systems Design, Implementation and Management, 5th Edition, Thomson Learning.

4. Chhanda Ray, Distributed Database System, Pearson Education India.

5. G. K. Gupta, Database Management Systems, McGraw – Hill.

Reference Books:

1. Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g, Black Book, Dreamtech Press.

2. Gillenson, Paulraj Ponniah, Introduction to Database Management, Wiley Publication.

3. Sharaman Shah, Oracle for Professional, SPD.

4. Raghu Ramkrishnan and Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw – Hill.

Unit I

Introduction Database Concepts

What is a Database?

- **Data**: Facts, figures, statistics etc. having no particular meaning (e.g. 1, ABC, 19 etc).
- **Record**: Collection of related data items, e.g. in the above example the three data items had no meaning. But if we organize them in the following way, then they collectively represent meaningful information.

Roll	Name	Age
1	ABC	19

- **Table** or **Relation**: Collection of related records.

Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

The columns of this relation are called **Fields**, **Attributes** or **Domains**. The rows are called **Tuples** or **Records**.

- **Database**: Collection of related relations. Consider the following collection of tables:

T1

Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

T2

Roll	Address
1	KOL
2	DEL
3	MUM

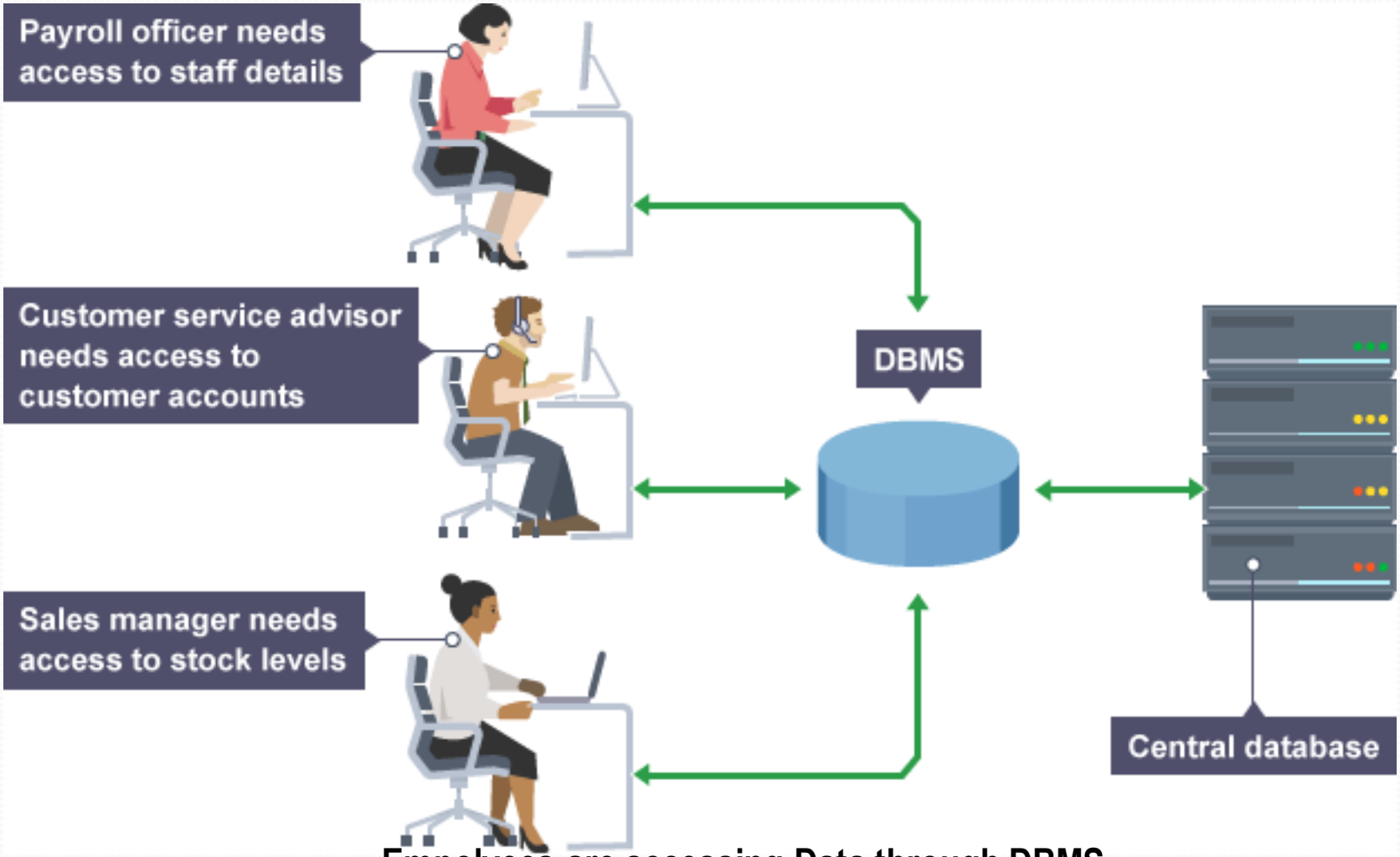
T3

Roll	Year
1	I
2	II
3	I

T4

Year	Hostel
I	H1
II	H2

- A database in a DBMS could be viewed by lots of different people with different responsibilities.



Empolyees are accessing Data through DBMS

Introduction Database Concepts

- A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data.
- The collection of data, usually referred to as the database, contains information relevant to an enterprise.
- 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.

For example: The college Database organizes the data about the admin, staff, students and faculty etc. Using the database, you can easily retrieve, insert, and delete the information.

Characteristics of the Database Approach

- Some of the most important characteristics of the database approach to the file processing approach are the following as follows.

1. Self-Describing Nature of a Database System

- One of the most fundamental characteristics of the database approach is that the database system contains not only the database itself but also an entire definition or description of the database structure and constraints also known as metadata of the database.
- This definition is stored within the DBMS catalog, which contains information like the structure of every file, the sort and storage format of every data item, and various constraints/rules on the information.

- The knowledge stored within the catalog is named meta-data, and it describes the structure of the first database The catalog is employed by the DBMS software and also by database users such as database administrators who required to know the information about the database structure.

2. **Isolation between Programs and Data, and Data Abstraction**

- In a traditional file processing system, the structure of database knowledge files is embedded within the application programs, so any changes to the structure of a file may require changing all programs that access that file.
- Against this, DBMS access programs don’t require such changes in most cases, so independence is achieved between them.
- The structure of knowledge files is stored within the DBMS catalog separately from the programs that access them. We call this property program-data independence.
- The characteristic that allows program-data independence and program-operation independence is known as data abstraction.

- A DBMS provides users with a conceptual representation of knowledge that doesn’t include much of the small print of how the information is stored or how the operations are implemented internally.
- The information model uses logical concepts, like objects, their properties, and their relationships between them, which will be easier for many users to know than memory concepts or storage concepts. Hence, the information model hides storage and implementation details that are not of interest to most database users, so unnecessary complications are hidden from them

3. Support for Multiple Views of the Data

- A database sometimes has many users, each of whom may require a special perspective or view of the database.
- A view could also be a subset of the database, or it's going to contain virtual data that is derived from the database files but isn't explicitly stored.
- Some users might not get to remember whether the information they ask for is stored or derived.
- A multi-user DBMS whose users have a spread of distinct applications must provide facilities for outlining multiple views. This provides many benefits for large databases such as the Aadhaar database.

4. Sharing of knowledge and Multi-user Transaction Processing:

- A multi-user DBMS as its name implies, must allow multiple users to access the database at an equivalent time or concurrently.
- This is often essential if data for multiple applications is to be integrated and maintained during a single database such as the latest feature of WhatsApp integration with Facebook.
- The DBMS must implement concurrency control in the software to make sure that several users trying to update equivalent data do so in a controlled manner in order that the results of the updates are correct.
- For instance, when several reservation agents attempt to assign a seat on an airline flight, the DBMS should make sure that each seat is often accessed by just one user agent at a single time for an assignment to a passenger.

- These sorts of applications are generally called online transaction processing (OLTP) applications. A fundamental role of multi-user DBMS software is to make sure that concurrent transactions operate correctly and efficiently with no inconsistency.
- The concept of a transaction has become central to several database applications. A transaction is an executing program or process that has one or more database accesses, like reading or updating of database records or inserting new records.
- The isolation property ensures that every transaction appears to execute in isolation from other transactions, many transactions could also be executed concurrently without affecting each other.
- The atomicity property ensures that either all the database operations during a transaction are executed or none are, these all ACID properties we know.

File system v/s Database system

DBMS	File System
DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	File system is a collection of data. In this system, the user has to write the procedures for managing the database.
DBMS gives an abstract view of data that hides the details.	File system provides the detail of the data representation and storage of data.
DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from the system failure.	File system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will lost.
DBMS provides a good protection mechanism.	It is very difficult to protect a file under the file system.
DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	File system can't efficiently store and retrieve the data.
DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while other deleting some information or updating some information.

Users of Database system

There are four different types of database-system users, differentiated by the way they expect to interact with the system. Different types of user interfaces have been designed for the different types of users.

- **Naive users** are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.
- They don't have any DBMS knowledge but they frequently use the data base applications in their daily life to get the desired results.
- For examples, Railway's ticket booking users are naive users. Clerks in any bank is a naive user because they don't have any DBMS knowledge but they still use the database and perform their given task.

- **Application programmers** are computer professionals who write application programs.
- They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, C++, JAVA, Pascal etc.
- They develop application programs and provide a user interface through applications so other user can interact with the database.

- **Sophisticated users** interact with the system without writing programs.
- They are those users who interact with the system without writing the program instead they form their request in database query language.
- They are the SQL programmers, who are going to deal directly with the database. They write queries to delete or select or insert and update the database.

- **Specialized users** are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.
- They are the developers who develop the complex programs to the requirement.
- Among these applications are computer-aided design systems, knowledgebase and expert systems, systems that store data with complex data types (for example, graphics data and audio data), and environment-modeling systems.

Data model Schema and Instance

- The data which is stored in the database at a particular moment of time is called an instance of the database.
- The overall design of a database is called schema.
- A database schema is the skeleton structure of the database. It represents the logical view of the entire database.
- A schema contains schema objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.
- A database schema can be represented by using the visual diagram. That diagram shows the database objects and relationship with each other.
- A database schema is designed by the database designers to help programmers whose software will interact with the database. The process of database creation is called data modeling.

- A schema diagram can display only some aspects of a schema like the name of record type, data type, and constraints. Other aspects can't be specified through the schema diagram. For example, the given figure neither show the data type of each data item nor the relationship among various files.

STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

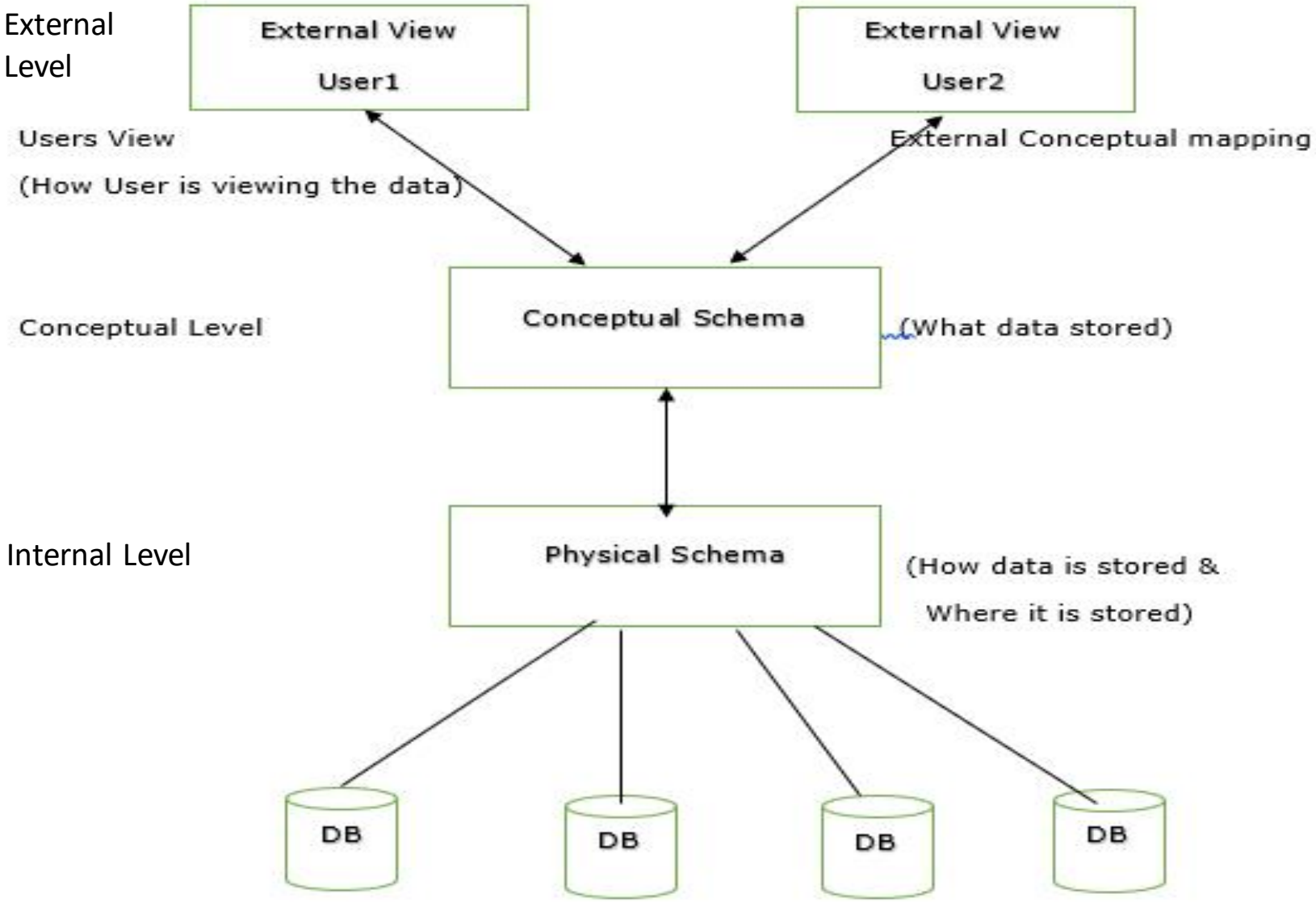
In the database, actual data changes quite frequently. For example, in the given figure, the database changes whenever we add a new grade or add a student. The data at a particular moment of time is called the instance of the database.

Three schema Architecture

- The three-schema architecture divides the database into three-level used to create a separation between the physical database and the user application.
- In simple terms, this architecture hides the details of physical storage from the user.
- The database administrator (DBA) responsibility is to change the structure of database storage without affecting the user's view. It deals with the data, the relationship between them and the different access methods implemented on the database.

- This architecture contains three layers of database management system, which are as follows –
  - External level
  - Conceptual level
  - Internal level

- **The three-schema architecture is as follows:**



- **Internal or Physical 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 in a block.
  - The physical level is used to describe complex low-level data structures in detail.

- **Conceptual or Logical level**
  - The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
  - The conceptual schema describes the structure of the whole database.
  - The conceptual level describes what data are to be stored in the database and also describes 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/ View level**

  - This is the highest level of database abstraction. It includes a number of external schemas or user views.
  - This level provides different views of the same database for a specific user or a group of users.
  - An external view provides a powerful and flexible security mechanism by hiding the parts of the database from a particular user.

Data Independence

Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

There are two types of data independence:

1. Logical Data Independence

- Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

2. Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.

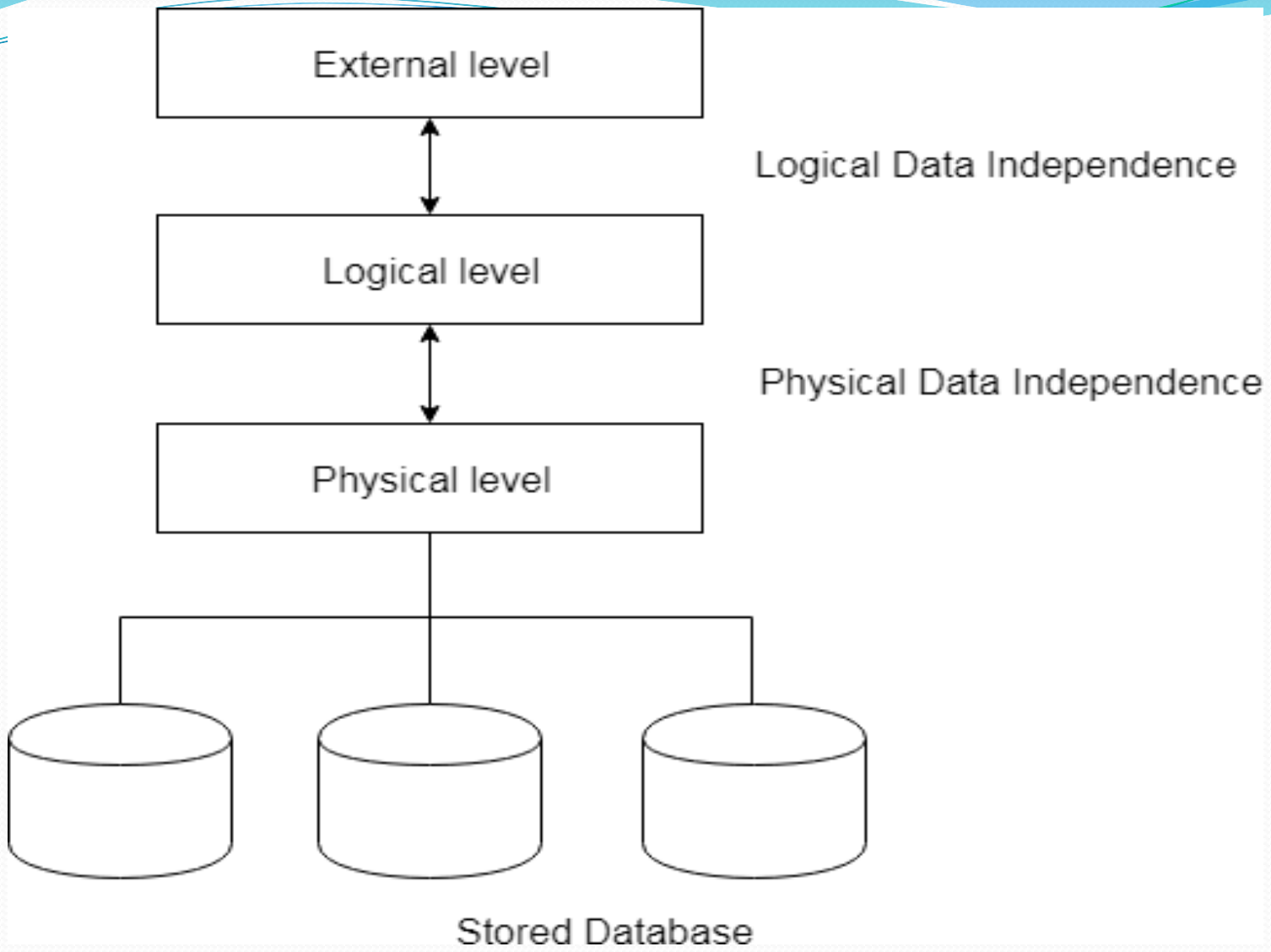
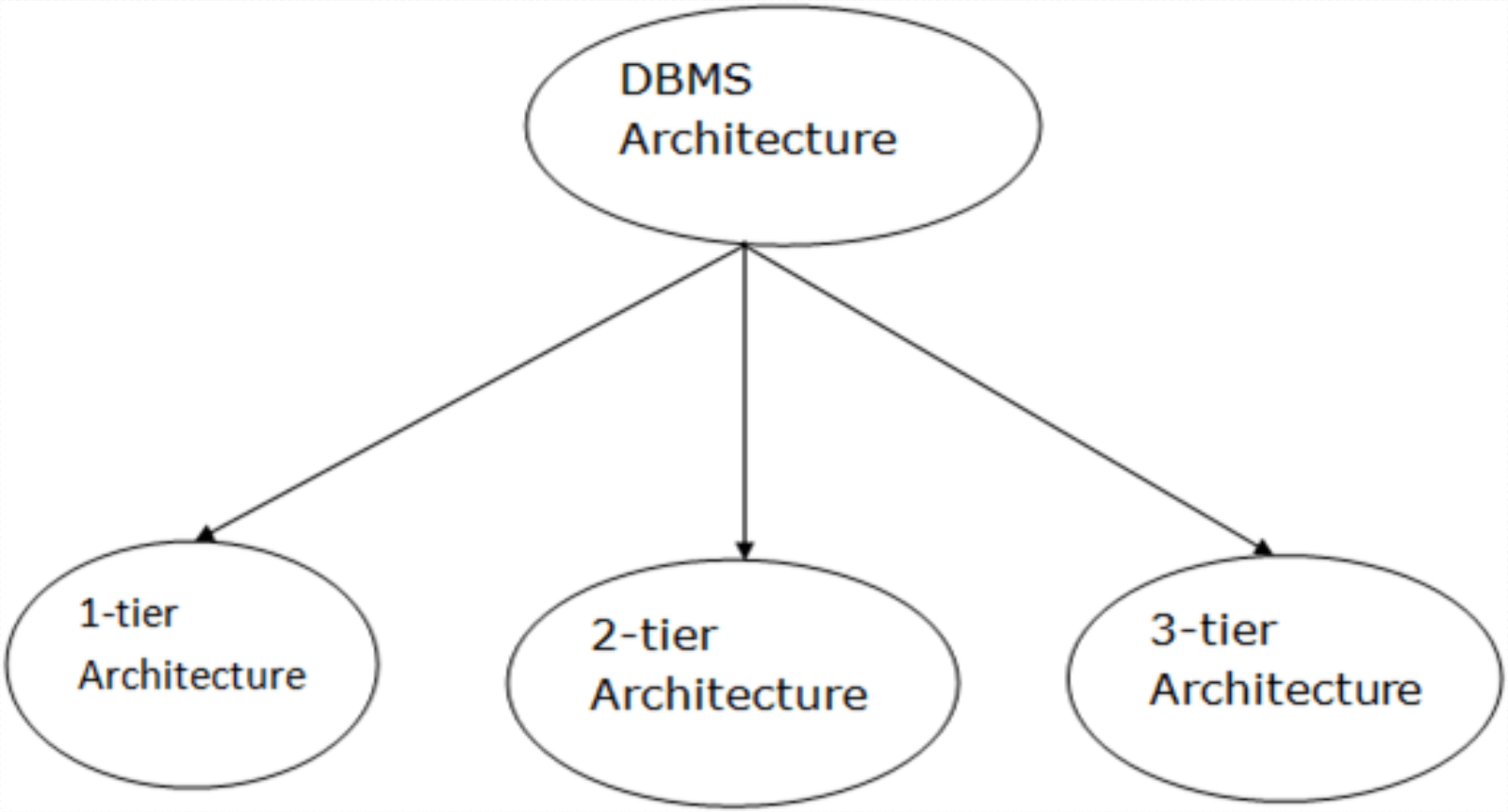


Fig: Data Independence

DBMS Architecture

- A **Database Architecture** is a representation of DBMS design.
- It helps to design, develop, implement, and maintain the database management system.
- A DBMS architecture allows dividing the database system into individual components that can be independently modified, changed, replaced, and altered.
- It also helps to understand the components of a database.
- A Database stores critical information and helps access data quickly and securely. Therefore, selecting the correct Architecture of DBMS helps in easy and efficient data management.

Types of DBMS Architecture



1-Tier Architecture

- **1 Tier Architecture** in DBMS is the simplest architecture of Database in which the client, server, and Database all reside on the same machine.
- A simple one tier architecture example would be anytime you install a Database in your system and access it to practice SQL queries. But such architecture is rarely used in production.

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.

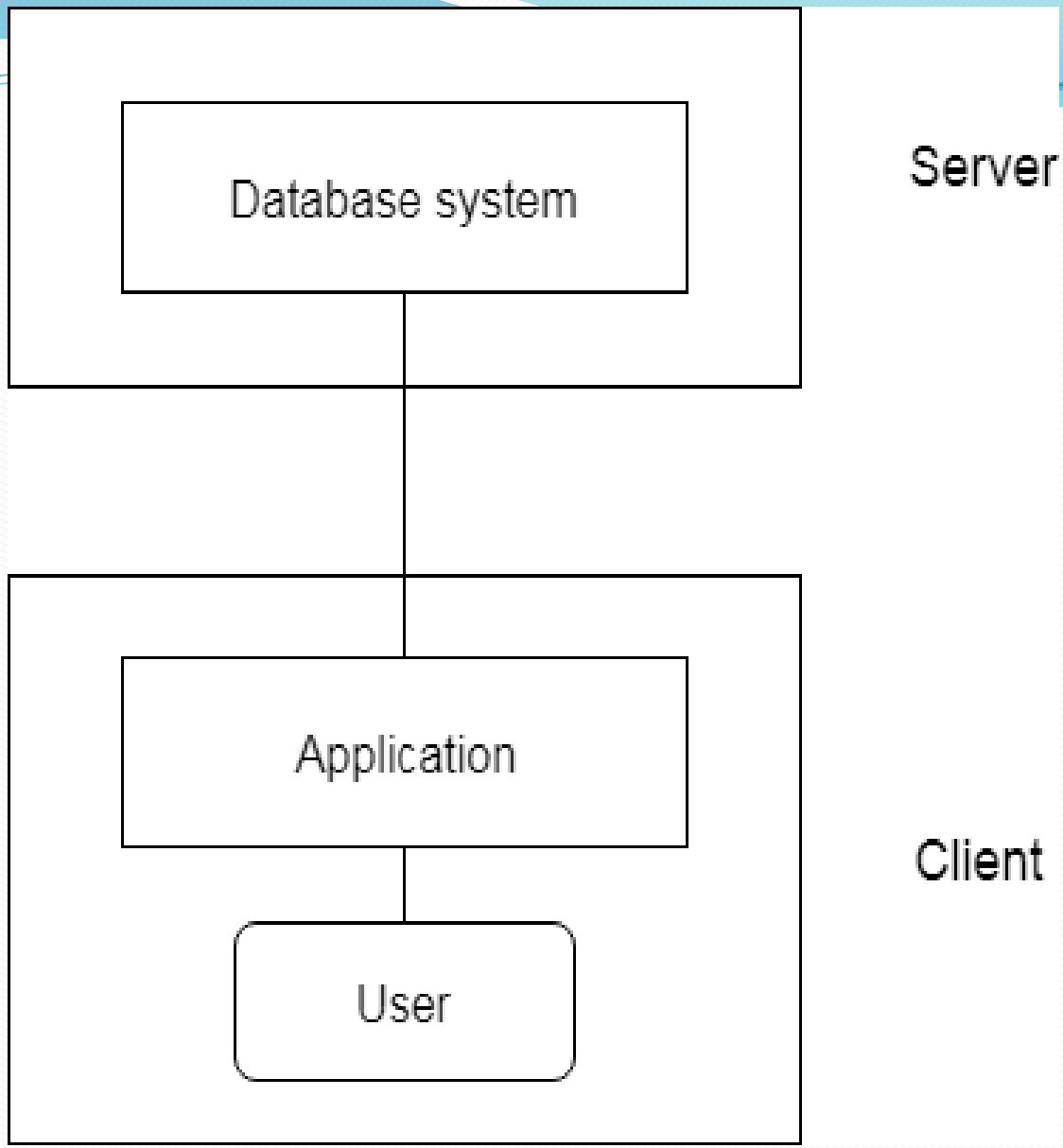


Fig: 2-tier Architecture

- **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 application.

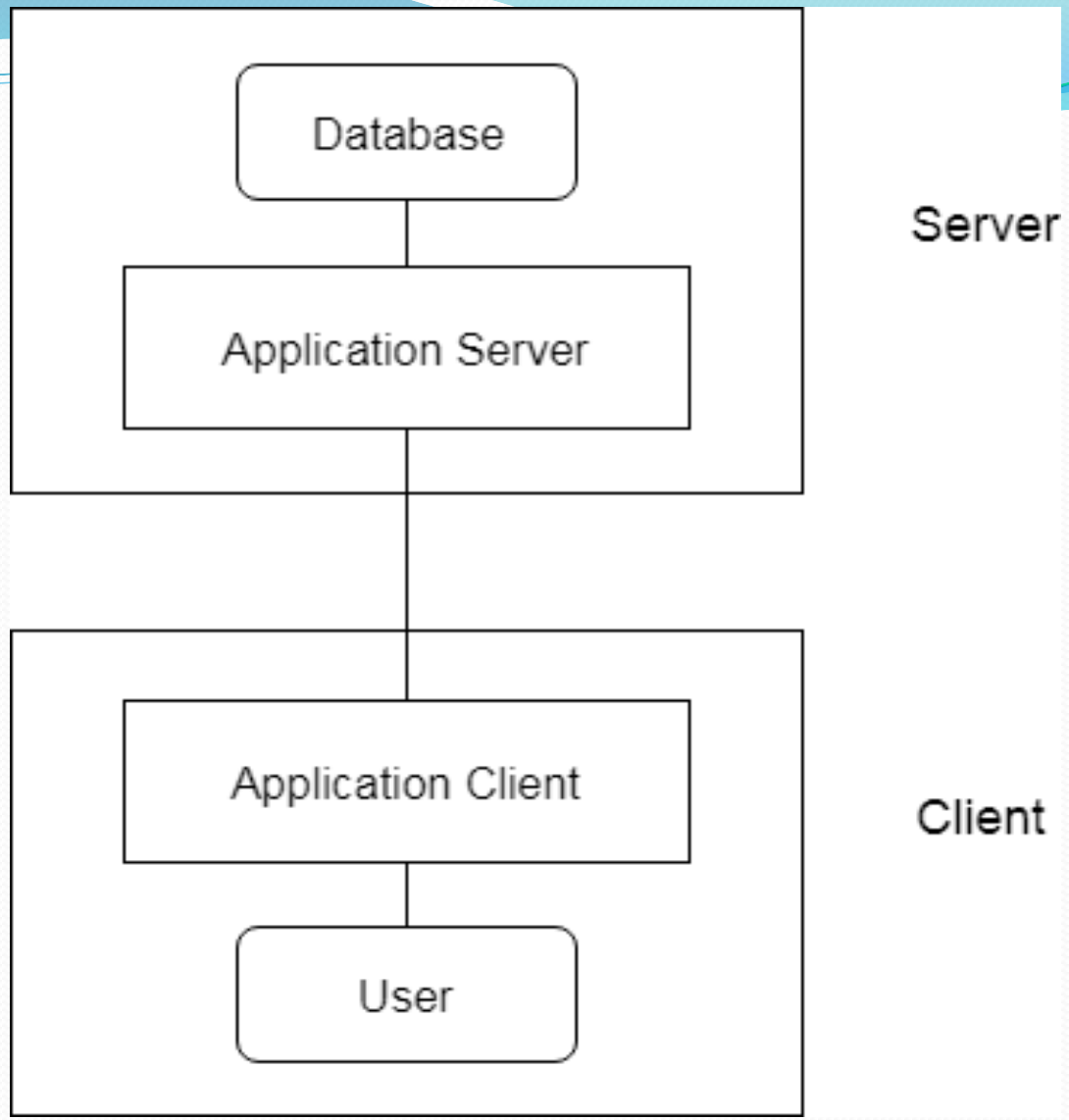


Fig: 3-tier Architecture

Database Administrator

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA).

The functions of a DBA include:

- **Schema definition**. The DBA creates the original database schema by executing a set of data definition statements in the DDL.
- **Storage structure and access-method definition.**
- **Schema and physical-organization modification**. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

- **Granting of authorization for data access**. By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.
- **Routine maintenance**. Examples of the database administrator's routine maintenance activities are: ◦ Periodically backing up the database, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.
- Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some 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.
- **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 electric failure or database corruption then the data may be lost forever.