LNCT GROUP OF COLLEGES
"WORKING TOWARDS BEING THE BEST"

UNIT 5

Basic Computer Engineering (BT-205)

DBMS

- A database is an organized collection of data that can be easily accessed, managed, and updated.
- Example: Student records, bank account details, online shopping data.
- A Database Management System (DBMS) is software used to store, organize, and manage data in a database.
- It helps users create, read, update, and delete data efficiently.

DBMS Uses

- To avoid data duplication.
- To keep data safe and secure.
- To organize large amounts of data.
- To allow multiple users to access data.
- To ensure data accuracy and consistency.

DBMS Functions

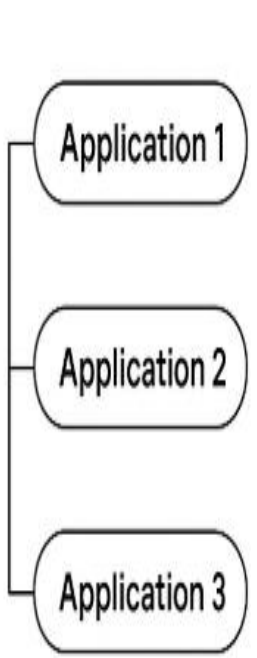
- Data storage
- Data retrieval
- Data updating
- Security & backup
- Controlling access

Examples of DBMS

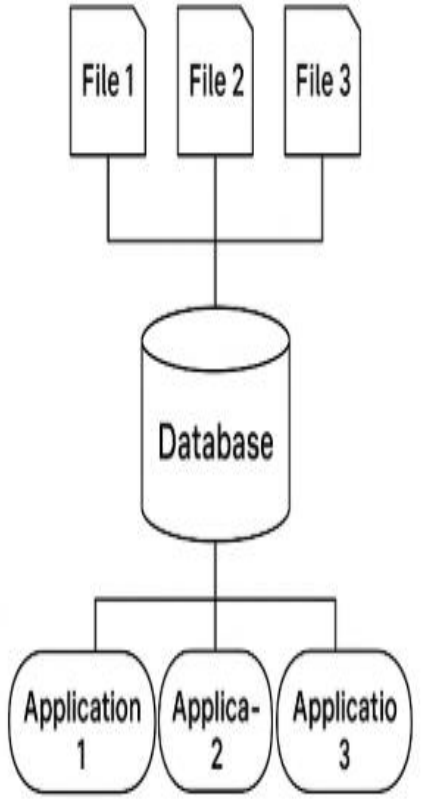
- MySQL
- Oracle
- SQL Server
- MongoDB

Types of Approaches

- File-Oriented Approach: The file-oriented approach stores data in separate files for each application. It often leads to data duplication, inconsistency, and difficult data management.
- Database Approach: The database approach stores all data in a central database managed by a DBMS. It provides controlled access, reduces redundancy, and ensures better data consistency.



File-Oriented Approach



Database Approach

Computer Network

Feature	File-Oriented Approach	Database Approach (DBMS)
Storage	Data stored in separate files	Data stored in a central database
Redundancy	High (same data repeated)	Low (data stored once)
Consistency	Low	High
Security	Weak, difficult to control	Strong, controlled by DBMS
Data Sharing	Difficult	Easy, multi-user supported
Flexibility	Low, hard to modify	High, easy to update
Backup & Recovery	Mostly manual	Automatic and efficient
Data Access	Needs separate programs	Easy through DBMS queries
Maintenance	Complex	Easier and centralized

Data Model

- Data Model is a way to organize, structure, and represent data in a database so that data can be stored, accessed, and managed easily.
- Types of Data Models
- 1. Hierarchical Model
- Data arranged in a tree (parent–child) structure.
- Example: Organization chart.
- 2. Network Model
- Data stored in a graph allowing many-to-many relationships.
- Example: Airline flight routes.

Data Model

- 3. Relational Model
- Data stored in tables (rows & columns) linked by keys.
- Example: Student table & Course table.
- 4. Entity–Relationship (ER) Model
- Data represented using entities, attributes, and relationships.
- used in designing databases.
- Example: Students–Enroll–Courses ER diagram.
- 5. Object-Oriented Model
- Data stored as objects with classes, inheritance, and methods.
- Useful for complex data (images, multimedia).
- Example: Storing shape objects like Circle, Rectangle.

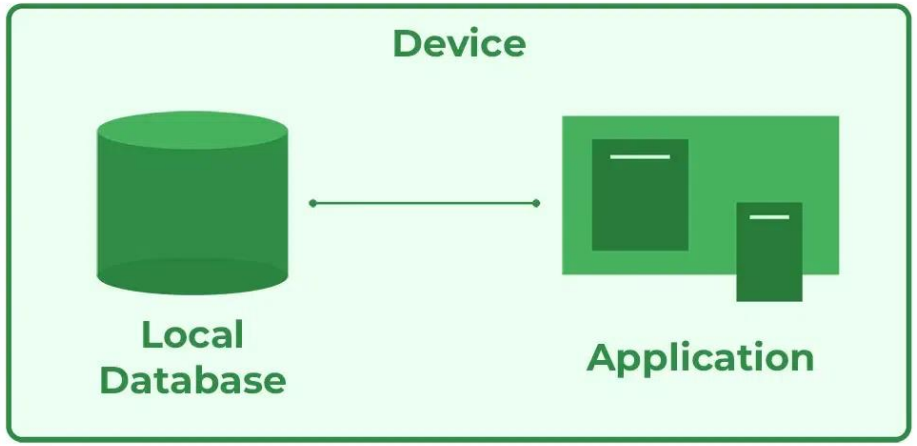
Architecture of Database System

- A DBMS architecture defines how users interact with the database to read, write, or update information. A well-designed architecture and schema (a blueprint detailing tables, fields and relationships) ensure data consistency, improve performance and keep data secure.
- **Types of DBMS Architecture:** There are three types of DBMS Architecture that we use according to the usage requirements.

- 1-Tier Architecture
- 2-Tier Architecture
- 3-Tier Architecture

1-Tier Architecture

- In 1-Tier Architecture, the user works directly with the database on the same system.
- This means the client, server and database are all in one application.
- The user can open the application, interact with the data and perform tasks without needing a separate server or network connection.



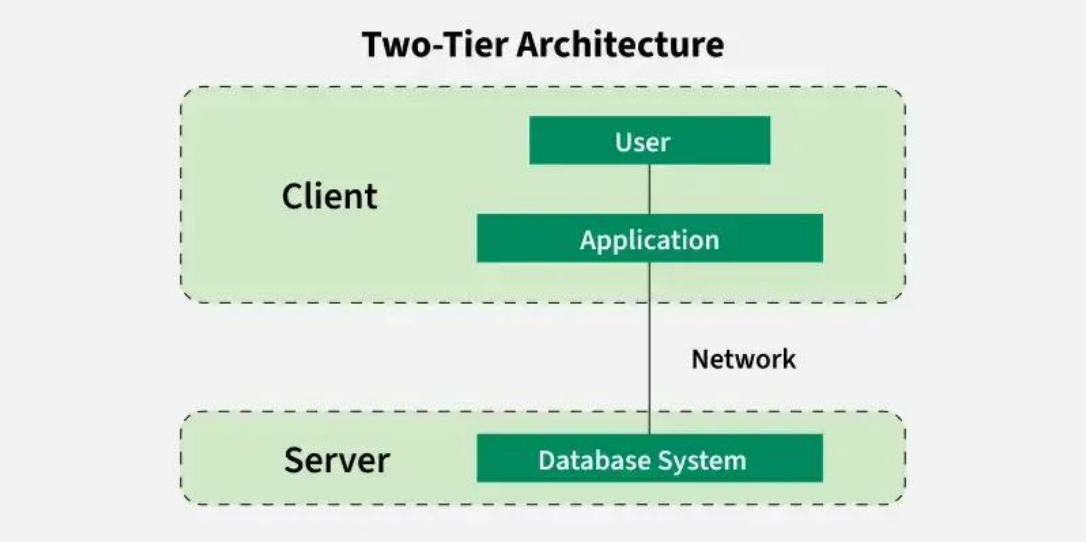
1-Tier Architecture

- A common example is Microsoft Excel.
- Everything from the user interface to the logic and data storage happens on the same device.
- The user enters data, performs calculations and saves files directly on their computer.

Advantages	Disadvantages
Simple Architecture	Limited to Single User
Cost-Effective	No Centralized Control
Easy to Implement	Hard to Share Data

2-Tier Architecture

- The 2-tier architecture is similar to a basic client-server model.
- The application at the client end directly communicates with the database on the server side.
- APIs like ODBC (Open Database Connectivity) and JDBC (Java Database Connectivity) are used for this interaction.



2-Tier Architecture

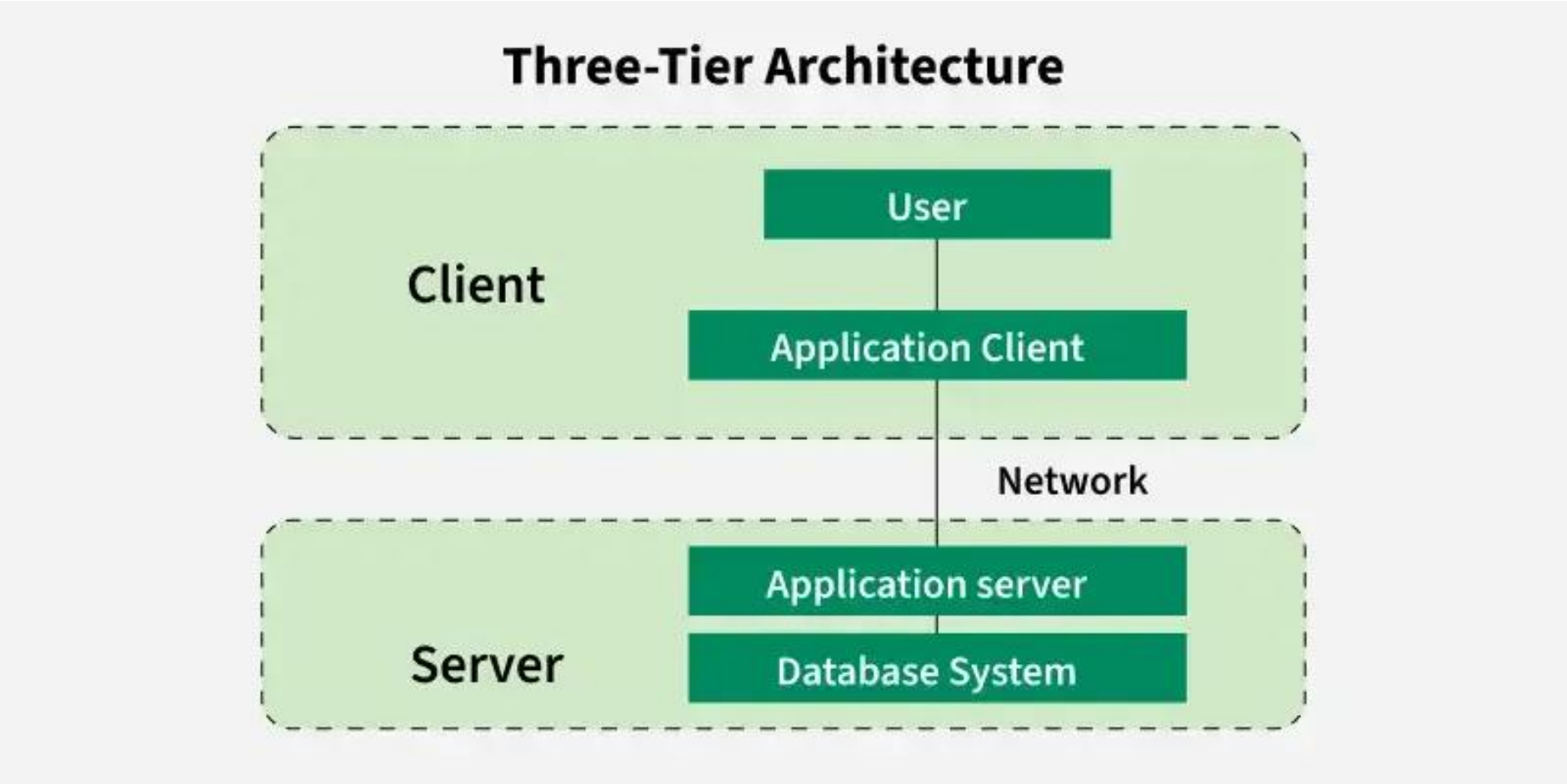
- For Example: A Library Management System used in Colleges.
- Client Layer (Tier 1): This is the user interface that library staff or users interact with.
- Database Layer (Tier 2): The database server stores all the library records such as book details, user information and transaction logs.

Advantages	Disadvantages
Easy to Access	Security Issues
Scalable	Tight Coupling
Low Cost	Difficult Maintenance

3-Tier Architecture

- In 3-Tier Architecture, there is another layer between the client and the server.
- The client does not directly communicate with the server. Instead, it interacts with an application server which further communicates with the database system and then the query processing and transaction management takes place.
- This intermediate layer acts as a medium for the exchange of partially processed data between the server and the client.
- This type of architecture is used in the case of large web applications.

3-Tier Architecture



3-Tier Architecture

- Example: E-commerce Store
- User: You visit an online store, search for a product and add it to your cart.
- Processing: The system checks if the product is in stock, calculates the total price and applies any discounts.
- Database: The product details, your cart and order history are stored in the database for future reference.

3-Tier Architecture

Advantages	Disadvantages
Enhanced scalability	More Complex
Data Integrity	Slower Response Time
Security	Higher Cost

Data Independence

- Data Independence is a fundamental concept in Database Management Systems (DBMS) that refers to the ability to modify the schema at one level of the database without affecting the schema at the next higher level.
- This concept ensures that changes in how data is stored or structured internally do not impact how users or applications access and interact with the data. It is of two types:
- **Physical Data Independence:** Change in the internal level without affecting the logical level.
- **Logical Data Independence:** Change in the logical level without affecting the view level.

LNCT GROUP OF COLLEGES
"WORKING TOWARDS BEING THE BEST"

Data Independence

Physical Data Independence	Logical Data Independence
Focuses on how data is stored physically	Focuses on structure and organization of data.
Deals with the internal schema	Deals with the conceptual schema
Changes don't affect application programs	Changes may require updates in application programs
It tells about the internal schema.	It tells about the conceptual schema.
Easier to achieve	More difficult to achieve
Used for performance and storage optimization	Used for evolving database design
Example: Moving data files or adding indexes	Example: Adding or removing a column in a table

Data Dictionary

- Data Dictionary is a centralized repository that stores definitions of all data elements in a database.
- It is “data about data” where details of tables, columns, data types, constraints, users, etc., are stored.
- Example: Table-STUDENT(Roll_No → Integer,Name → Text,Course → Varchar(30)).These definitions are stored in the Data Dictionary.
- Use of data dictionary:
- It Stores metadata (structure details).
- Helps DBMS in managing the database.
- Used by developers, DBA, and designers.
- Ensures consistency of data definitions.
- Automatically maintained by the DBMS.

DBA (Database Administrator)

- Database Administrator is a person responsible for managing, controlling, and maintaining a database system.
- Uses of DBA:
- Database Design & CreationUser Management (create users, assign permissions)
- Backup & Recovery
- Performance Tuning
- Security Management
- Monitoring database operations
- Ensuring data integrity & availability
- Example:A DBA ensures: Only authorized users can access the student database. Regular backups are taken. Database runs fast without issues.

A Primary Key

- A Primary Key is a special column (or set of columns) in a table that uniquely identifies each record.
- Features of primary key A Primary Key :
- Unique (no two rows can have the same value)
- Not Null (cannot be empty)
- Only one primary key per table
- Ensures data integrity
- Example: Table: STUDENT –(Roll_No (Primary Key), Name)
- Here, Roll_No is the primary key because it uniquely identifies every student.
- It is used to avoid duplicate records , search and update specific rows or maintain accuracy and consistency.

Data Definition Language (DDL)

- DDL stands for **Data Definition Language**.
  It is used to **define, create, alter, and delete** the structure of database objects like tables, views, indexes, etc.
- **Common DDL Commands:**
- **CREATE** – to create tables
- **ALTER** – to modify structure
- **DROP** – to delete table
- **TRUNCATE** – to remove all data
- **RENAME** – to rename objects

Data Manipulation Language(DML)

- DML stands for **Data Manipulation Language**.
  It is used to **insert, update, delete, and retrieve** data from database tables.
- **Common DML Commands:**
- **INSERT** – add new records
- **UPDATE** – modify existing records
- **DELETE** – remove records
- **SELECT** – retrieve data

LNCT GROUP OF COLLEGES
"WORKING TOWARDS BEING THE BEST"

DDL vs DML

DDL (Data Definition Language)	DML (Data Manipulation Language)
Defines and modifies database structure	Manages and manipulates data inside tables
Affects schema of the database	Affects records/rows
AUTO-COMMIT (changes cannot be rolled back)	Can be rolled back using ROLLBACK
Examples: CREATE, ALTER, DROP	Examples: INSERT, UPDATE, DELETE, SELECT
Slower usage frequency (done rarely)	Used frequently by users/programs

LNCT GROUP OF COLLEGES
"WORKING TOWARDS BEING THE BEST"

DDL vs DML

DDL (Data Definition Language)	DML (Data Manipulation Language)
Changes are auto-committed	Changes can be rolled back
Not used frequently (structure changes are rare)	Used very frequently (daily data operations)
Does not use WHERE clause	Uses WHERE clause (UPDATE, DELETE, SELECT)
Does not affect the actual data inside the table	Directly affects the data inside the table
Examples: CREATE, ALTER, DROP, TRUNCATE	Examples: INSERT, UPDATE, DELETE, SELECT
Used by DBA and developers	Used by end users and applications
Helps in creating database design	Helps in maintaining and modifying data
DDL commands cannot be undone by ROLLBACK	DML commands can be undone by ROLLBACK
DDL commands require higher privileges	DML commands need normal user privileges

Cloud Computing

- Cloud Computing is the delivery of computing services (like storage, servers, software, databases) over the Internet, so you can access them anytime, anywhere.
- Cloud computing means using computers, storage, and software over the internet instead of keeping everything on your personal computer.
- Think of it like this:
- Instead of storing photos on your phone → you store them on Google Drive
- Instead of installing software on your computer → you use online apps like Gmail or Microsoft 365.
- Instead of buying big servers → companies rent servers from cloud providers like AWS, Azure, or Google Cloud.

Cloud Infrastructure

- Cloud infrastructure refers to the combination of physical hardware and software resources that together enable cloud computing.
- **Key Components of Cloud Infrastructure:**
- **Hardware:** Physical equipment like servers, routers, switches, firewalls, and load balancers that support cloud operations.
- **Virtualization:** Software technology that abstracts physical resources to create virtual machines, enabling efficient use of hardware.
- **Storage:** Data storage solutions that allow hosting of big data and applications in the cloud rather than on local data centers.
- **Networking:** The network infrastructure that ensures seamless data transfer and accessibility of cloud resources via the internet.

Cloud segments or service delivery models

- There are three Cloud segments or service delivery models:
- **Software as a Service(SaaS):** Software-as-a-Service (SaaS) means using software over the internet instead of installing in on your computer. You don't have to worry about downloading, updating, or maintaining anything. the company that provides the software handles all of that. Eg: Google Docs.
- **Platform as a Service(Pass):** PaaS is a type of cloud service that gives developers the tools they need to build and launch apps online without setting up any hardware and software themselves.
- With PaaS, everything runs on the provider's server and is accessed through a web browser. The provider takes care of things like servers, storage, and operating systems. Developers just focus on writing and managing the app. Eg: Salesforce, Azure, AWS

Cloud segments or service delivery models

- Infrastructure as a Service (IaaS) is a cloud service model where companies rent IT infrastructure instead of buying it. This infrastructure includes Virtual machines (VMs), Servers, Storage, Networking, Firewalls & Load balancers. Users can install their own operating systems, software, databases, and manage everything, while the cloud provider manages the physical hardware. Eg: Azure, AWS.

Types of Cloud

- **Public Cloud:** Cloud services are open for everyone to use. The cloud provider owns and manages everything. Examples: AWS, Google Cloud, Microsoft Azure.
- **Private Cloud:** Used by one single organization only. Not shared with the public. More secure and controlled. Examples: Banks, government, big companies running their own data centres.
- **Hybrid Cloud:** (Combination of Public Cloud + Private Cloud): Some data is kept private, some is stored in the public cloud. Example: A hospital keeps patient data in private cloud but uses public cloud for apps like email.

Pros of Cloud Computing

- **Cost Saving**: You don't need to buy expensive servers or hardware. You only pay for what you use.
- **Accessibility from Anywhere:** You can access files, software, and services from any device with the internet.
- **Scalability:** You can increase or decrease resources anytime (like adding more storage or servers).
- **Automatic Updates:** Cloud providers automatically update hardware and software.
- **Backup & Recovery:** Your data is automatically backed up, reducing the risk of data loss.
- **No Maintenance:** Providers handle hardware repairs, security, and maintenance.

Cons of Cloud Computing

- **Requires Internet:** If the internet is slow or unavailable, you cannot access cloud.
- **Security Concerns:** Storing data online may raise privacy or security risks, depending on the provider.
- **Limited Control:** You cannot control physical hardware; everything is managed by the cloud provider.
- **Downtime Issues:** If the cloud provider faces an outage, your service may stop temporarily.
- **Possible Hidden Costs:** Extra storage, computing power, or network usage may increase the bill.