Unit – 1

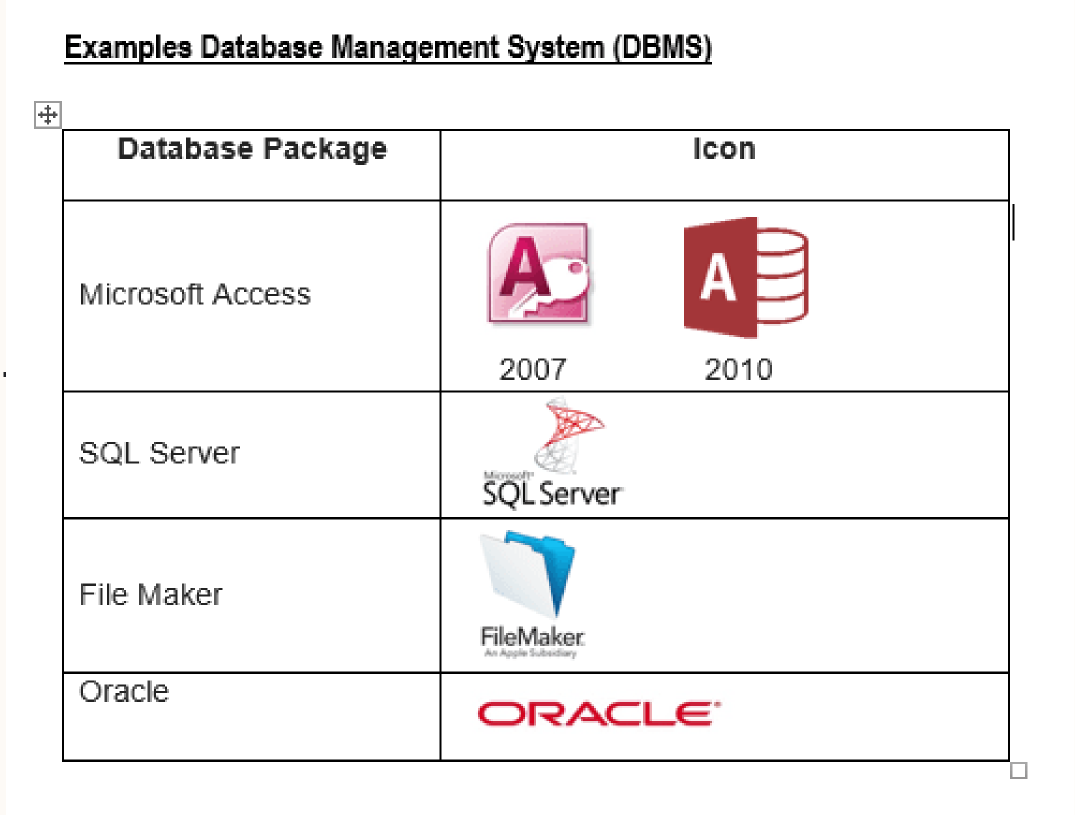
Database Concepts and Architecture

WHAT IS DATABASE ?

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS).

WHAT IS DATABASE MANAGEMENT SYSTEM?

- A database management system (DBMS) is a software tool that enables users to manage a database easily.
- A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.



Examples Database Management System (DBMS)

Database Package	Icon
Microsoft Access	2007 2010
SQL Server	Microsoft SQL Server
File Maker	FileMaker An Apple Subsidiary
Oracle	ORACLE

DATABASE USERS AND ADMINISTRATORS:

The people who are working with the database are called database users and administrators

TYPES OF DATABASE USERS

The database users also can be categorized again into five groups according to how they interact with the database. They are:

- **Native/Naive Users :** Database users who communicate with database through an already written program.
- **Application Programmers :** The software developers and programming professionals who write the program codes and also develop user interfaces.
- **Sophisticated Users :** The users who are creating the databases using query languages like SQL. These types of users doesn’t write program code.
- **Specialized Users :** The sophisticated users who write special database application programs are called specialized users. They write complex programs for the specific complex requirements.
- **Stand-Alone User:** Those who are using database for personal usage is called stand-alone users.

DATABASE ADMINISTRATORS (DBA):

A database administrator (DBA) is a specialized computer systems administrator who maintains a successful database environment by directing or performing all related activities to keep the data secure.

The major responsibilities of DBA in a database system are as listed below:

- **Installing and Configuration of database:**
  - ✓ DBA is responsible for installing the database software, configure the software of database and then upgrade it if needed.
  - ✓ There are many database software like ORACLE, Microsoft SQL and MySQL in the industry so DBA decides how the installing and configuration of these database software will take place.
- **Deciding the hardware device:**
  - ✓ Depending upon the cost, performance and efficiency of the hardware, it is DBS who have the duty of deciding which hardware device will suit the company requirement.
- **Managing data integrity:**
  - ✓ Data integrity should be managed accurately because it protects the data from unauthorized user.
  - ✓ DBA manages relationship between the data to maintain data consistency
- **Capacity Issue:**
  - ✓ All the databases have their limits of storing data in it and the physical memory also has some limitations. DBA has to decide the limit and capacity of database and all the issues related to it.

ADVANTAGES OF DATABASES

There are numerous advantages of database for managing and organizing data efficiently:

1. **Data Integrity:** Databases enforce data integrity by implementing constraints such as unique keys, foreign keys, and data types, ensuring that data entered into the database meets certain standards and rules. This helps in maintaining accurate and reliable data.

2. **Data Consistency:** With a centralized database, data consistency is ensured as changes made to the data are immediately reflected across all applications and users accessing the database.

3. **Data Security:** Databases provide mechanisms for access control and authentication to ensure that only authorized users can access and manipulate data.

4. **Concurrency Control:** Databases handle multiple users accessing and modifying data simultaneously through concurrency control mechanisms.

ADVANTAGES OF DATABASES

5. **Data Recovery and Backup :** Databases provide features for data backup and recovery, allowing organizations to restore data to a previous state in the event of data loss or corruption.

6. **Data Sharing :** Databases facilitate data sharing among multiple users and applications within an organization.

7. **Query Capabilities :** Databases offer powerful query capabilities, allowing users to retrieve and analyze data using query language like SQL.

8. **Data Independence :** Databases provide a layer of abstraction between applications and data, allowing changes to the database structure (schema) without affecting the applications that use it.

Data Abstraction

- Database systems are made-up of complex data structures.
- To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.
- There are mainly three levels of data abstraction:

  1.View Level

  2.Conceptual Level

  3.Physical Level

Real Life Example of Abstraction

DATA MODELS

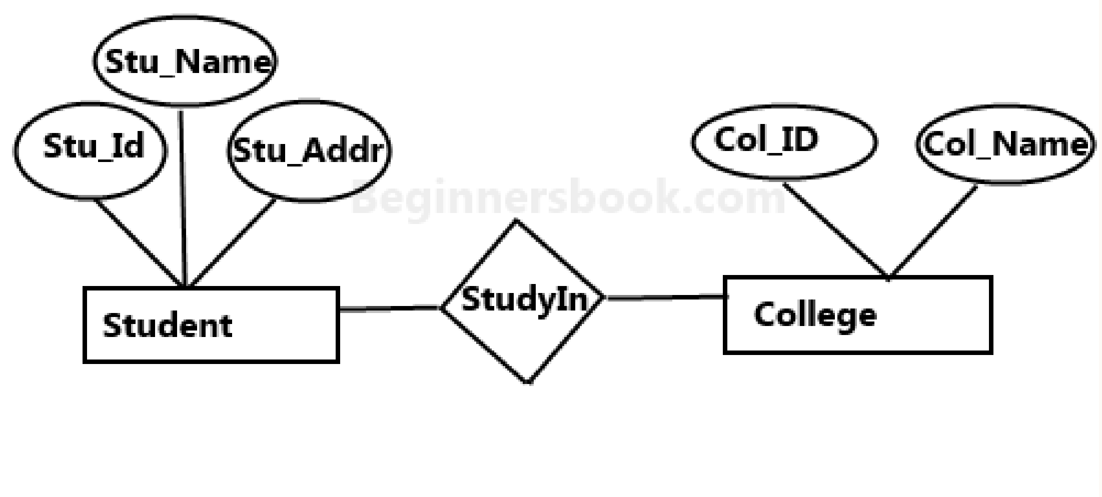
Data Models are **used to show how data is stored, connected, accessed and updated in the database management system**.

- E-R MODEL
- OBJECT ORIENTED MODEL
- RELATIONAL MODEL
- NETWORK MODEL
- HIERARCHICAL MODEL
- PHYSICAL MODEL

Data Models are fundamental entities to introduce abstraction in a DBMS.

ER MODEL

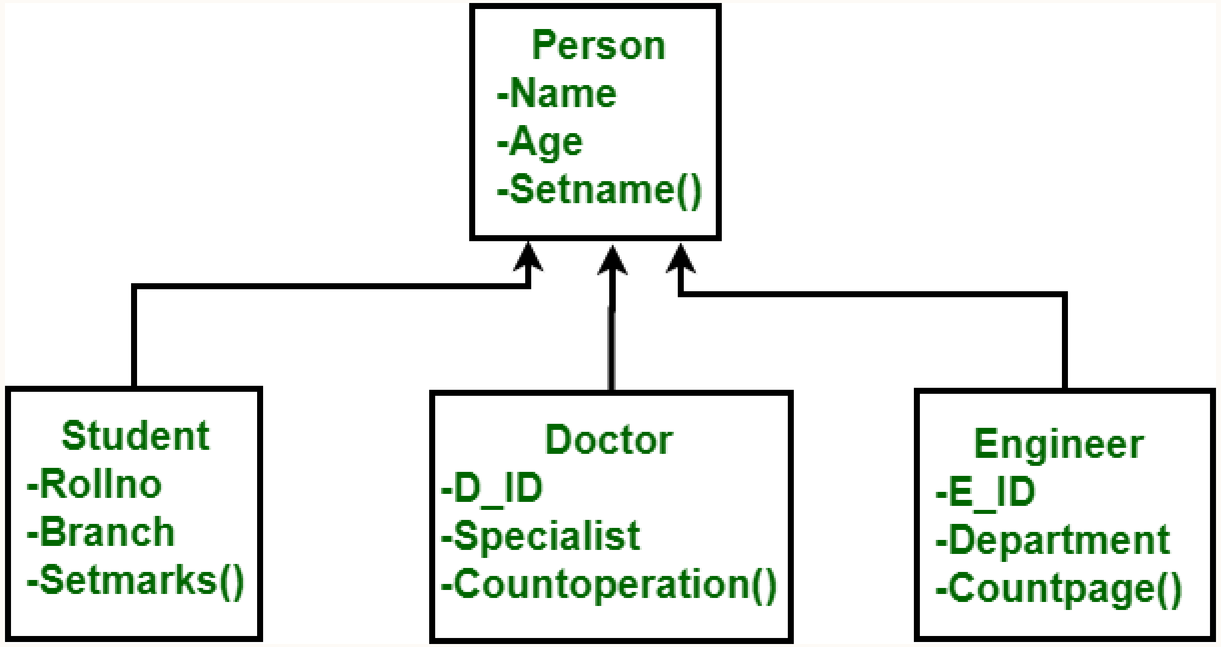
- In terms of DBMS, an entity is a table or attribute of a table in a database.
- The association between two entity types is called a relationship
- An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).
- An ER model is a design or blueprint of a database that can later be implemented as a database



Sample E-R Diagram

OBJECT ORIENTED MODEL

- In Object Oriented Model, the scenarios are represented as objects.
- The objects with similar functionalities are grouped together and linked to different other objects.



RELATIONAL MODEL

- The relational model in DBMS is an abstract model used to organize and manage the data stored in a database.
- It stores data in two-dimensional inter-related tables, also known as relations in which each row represents an entity, and each column represents the properties of the entity.
- Relational database systems use a model that organizes data into tables of rows (also called records or tuples) and columns (also called attributes or fields).

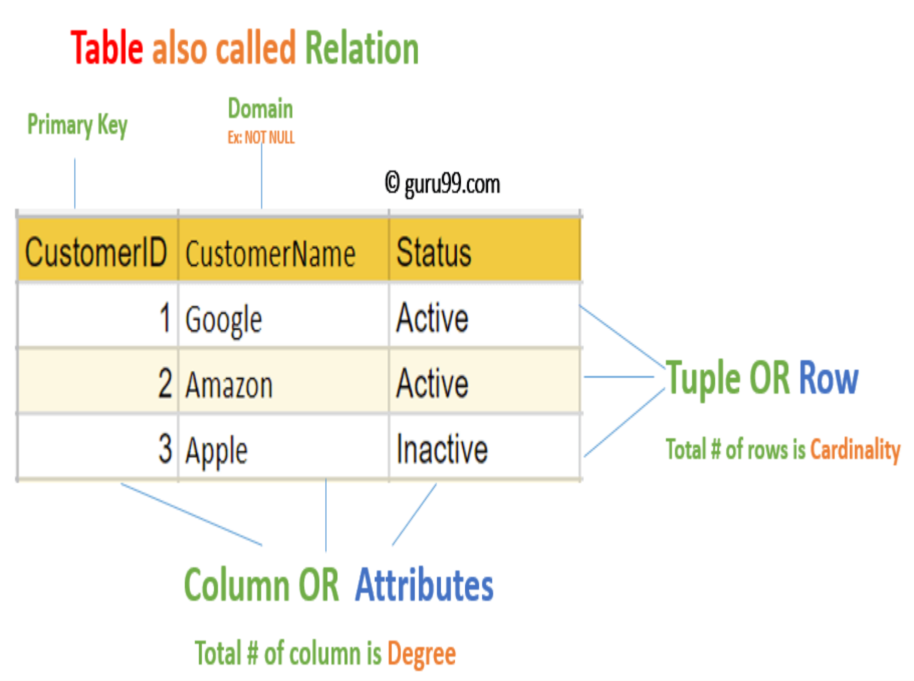
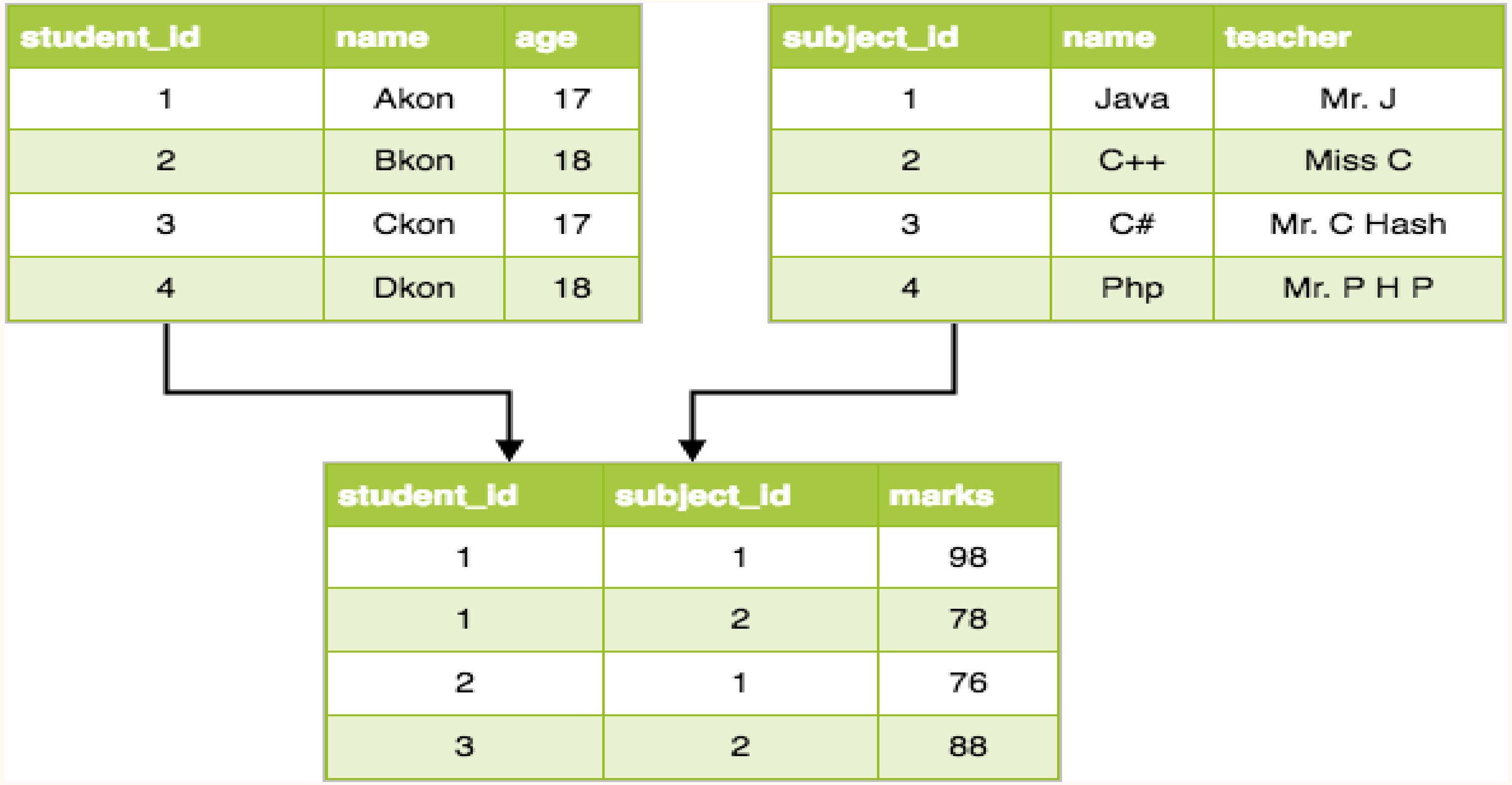


Table also called Relation
Primary Key
Domain
Ex: NOT NULL
© guru99.com
Tuple OR Row
Total # of rows is Cardinality
Column OR Attributes
Total # of column is Degree

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive



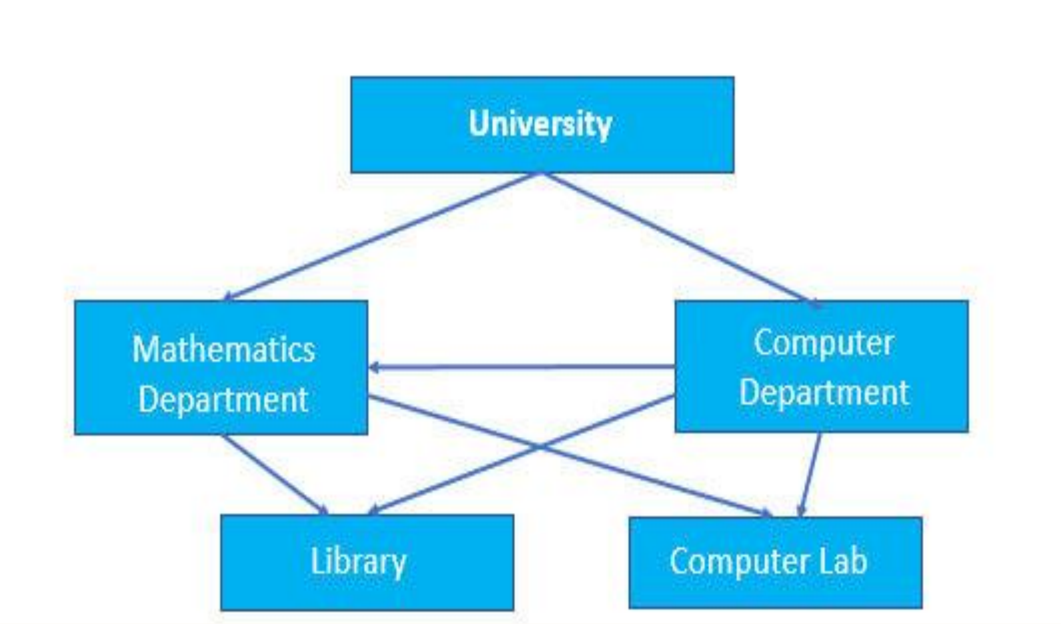
student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

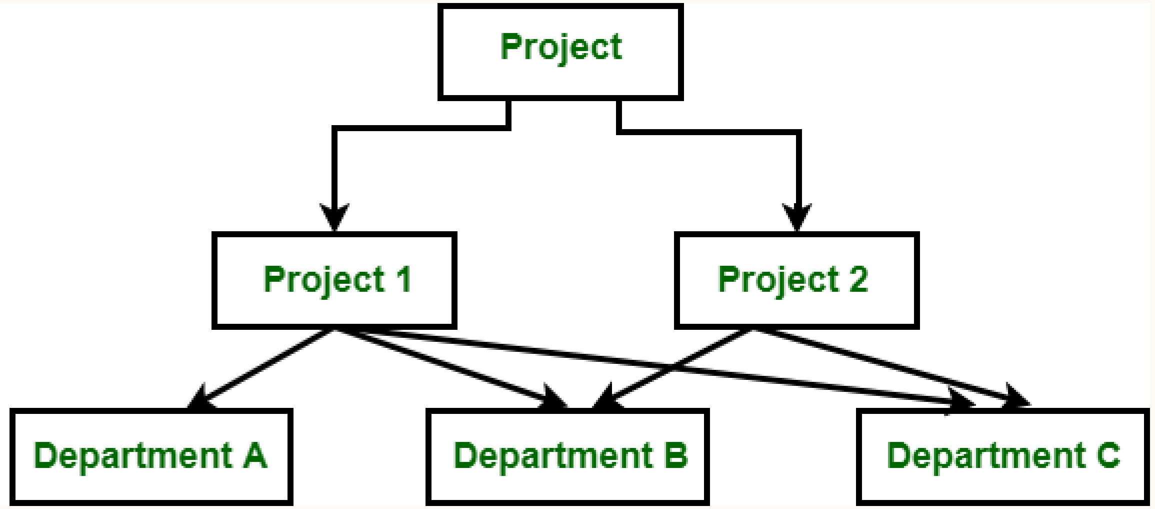
subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

NETWORK MODEL

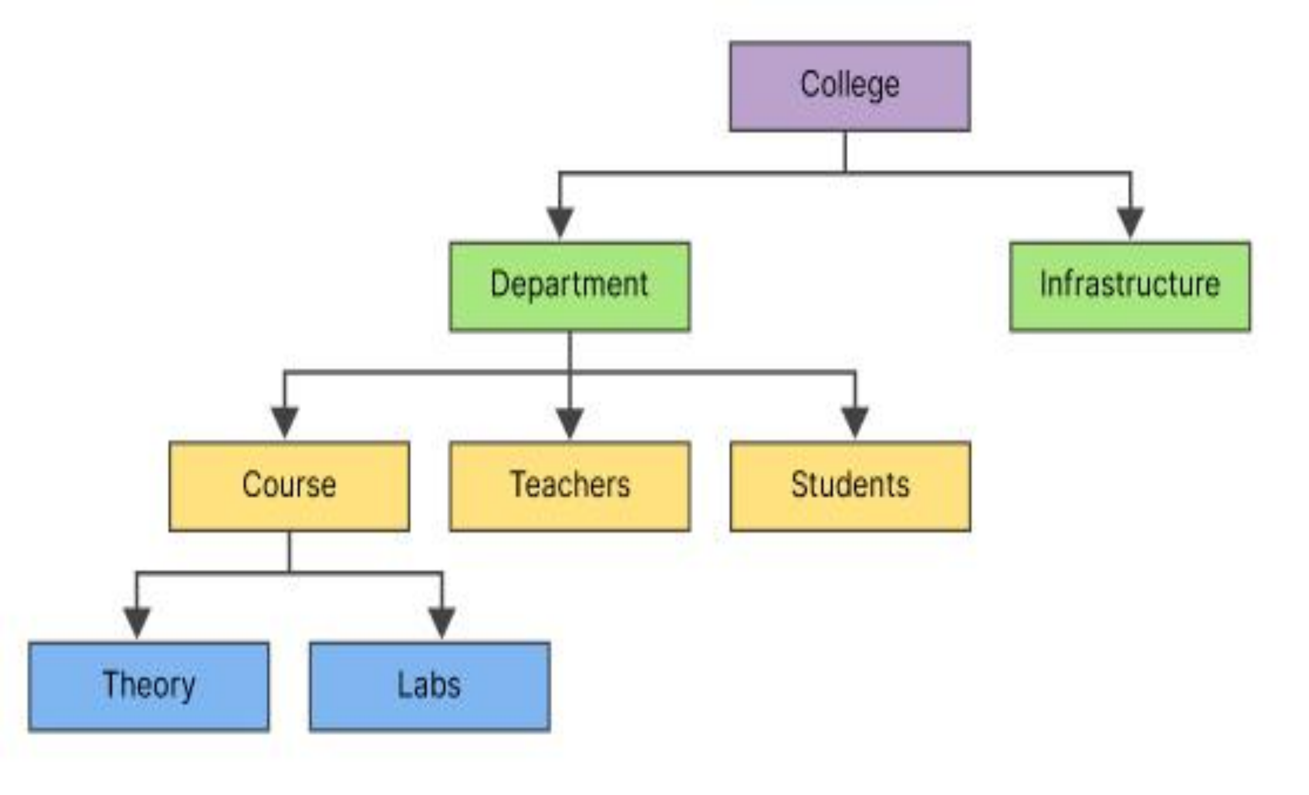
The network model describes the architecture, components, and design used to establish communication between the source and destination systems.





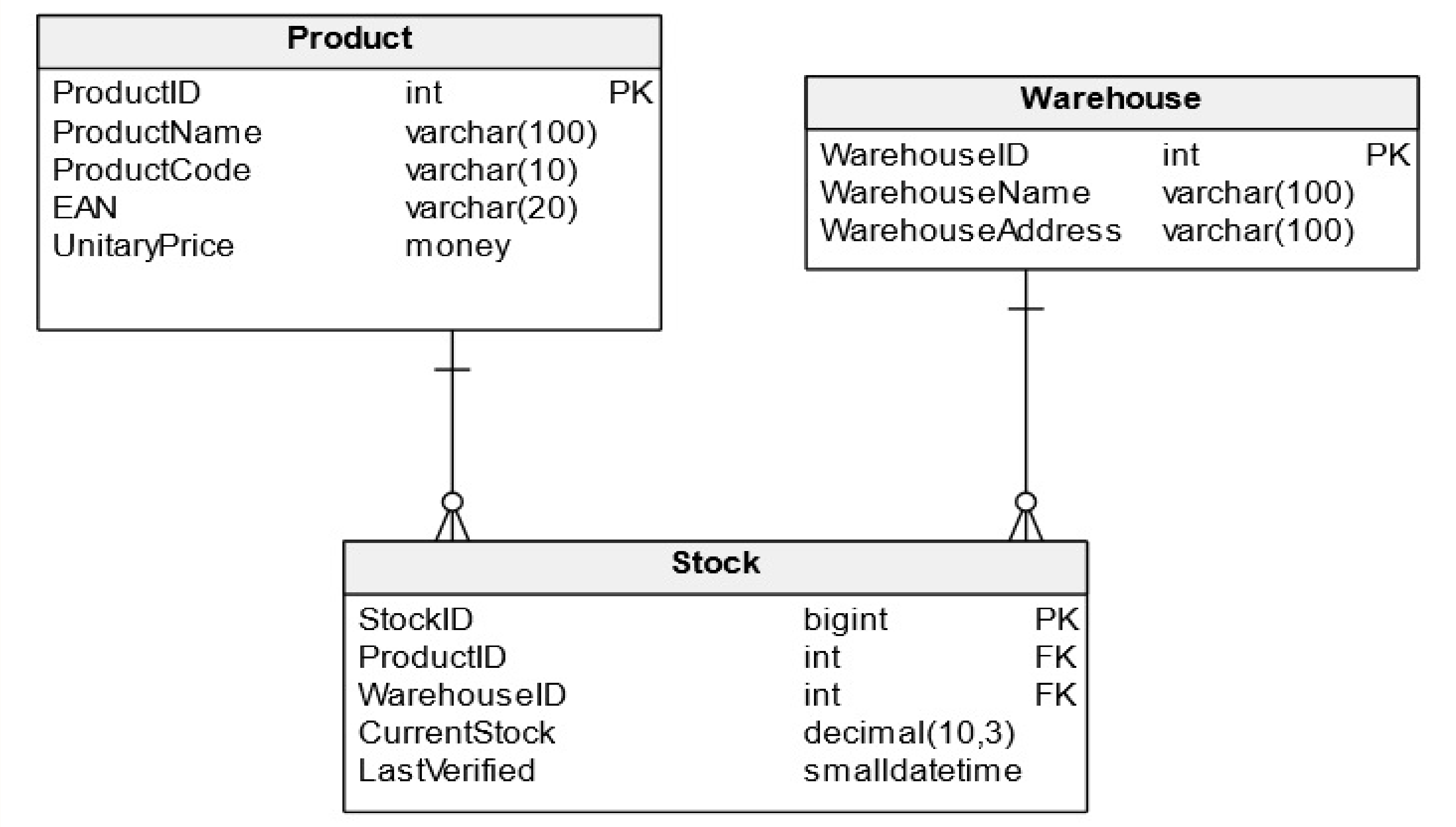
HIERARCHICAL MODEL

A hierarchical database is a data model in which data is stored in the form of records and organized into a tree-like structure, or parent-child structure, in which one parent node can have many child nodes connected through links.



PHYSICAL MODEL

- A physical data model represents the data at data layer or internal layer.
- It represents each table, their columns and specifications, primary key, foreign key etc.
- It basically represents how each table are built and related to each other.



Product		
ProductID	int	PK
ProductName	varchar(100)	
ProductCode	varchar(10)	
EAN	varchar(20)	
UnitaryPrice	money	

Warehouse		
WarehouseID	int	PK
WarehouseName	varchar(100)	
WarehouseAddress	varchar(100)	

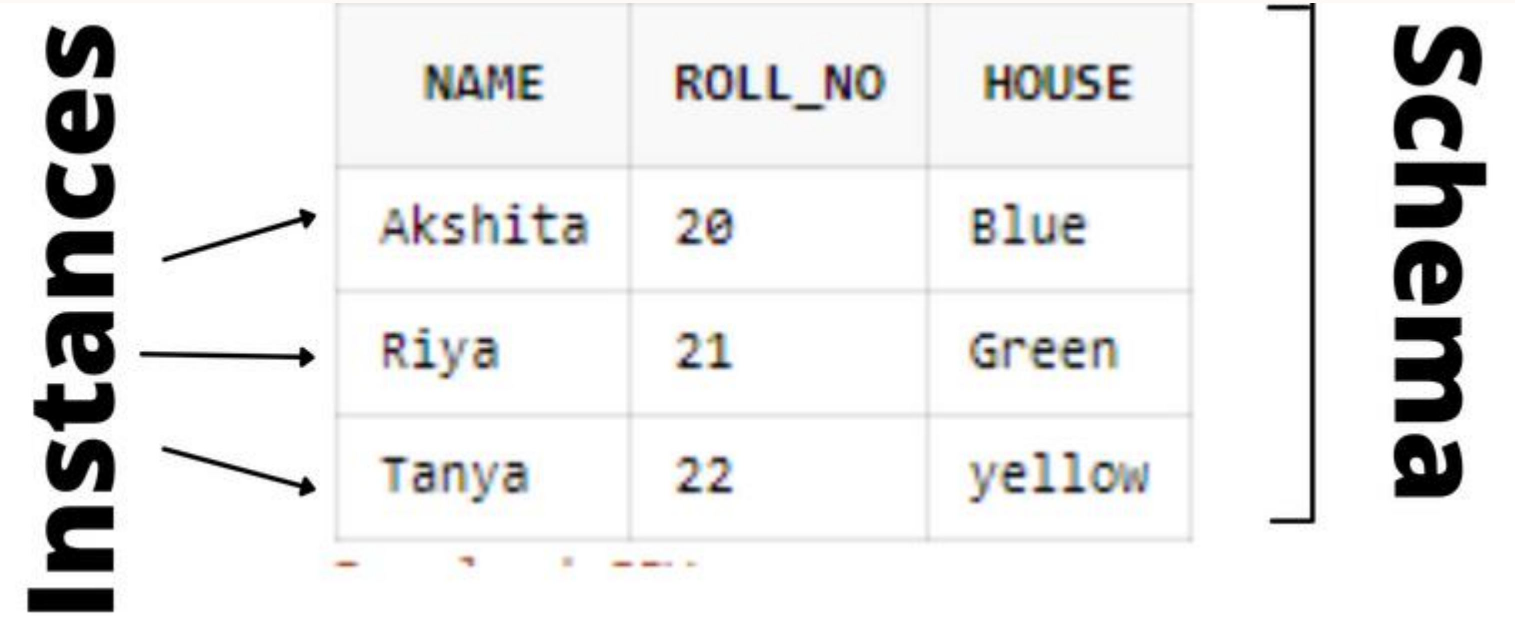
Stock		
StockID	bigint	PK
ProductID	int	FK
WarehouseID	int	FK
CurrentStock	decimal(10,3)	
LastVerified	smalldatetime	

INSTANCES:

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

SCHEMA:

The overall design of the database is called the database schema.



NAME	ROLL_NO	HOUSE
Akshita	20	Blue
Riya	21	Green
Tanya	22	yellow

THREE – SCHEMA ARCHITECTURE

The Three Schema Architecture, also known as the ANSI/SPARC architecture, is a conceptual framework that describes the structure and organization of a database management system (DBMS). This architecture consists of three levels of abstraction or schemas, each representing a different perspective of the database and it provides a clear separation between different levels of abstraction, allowing for data independence, flexibility, and security in database systems.

1.**External Schema (View Level)**:

- This is the highest level of abstraction and represents the user's view of the database.
- Each external schema describes a subset of the database, hiding irrelevant details and providing a simplified and customized view of the data.
- External schemas are often defined using data definition languages (DDL) and allow users to interact with the database without needing to understand its underlying structure.

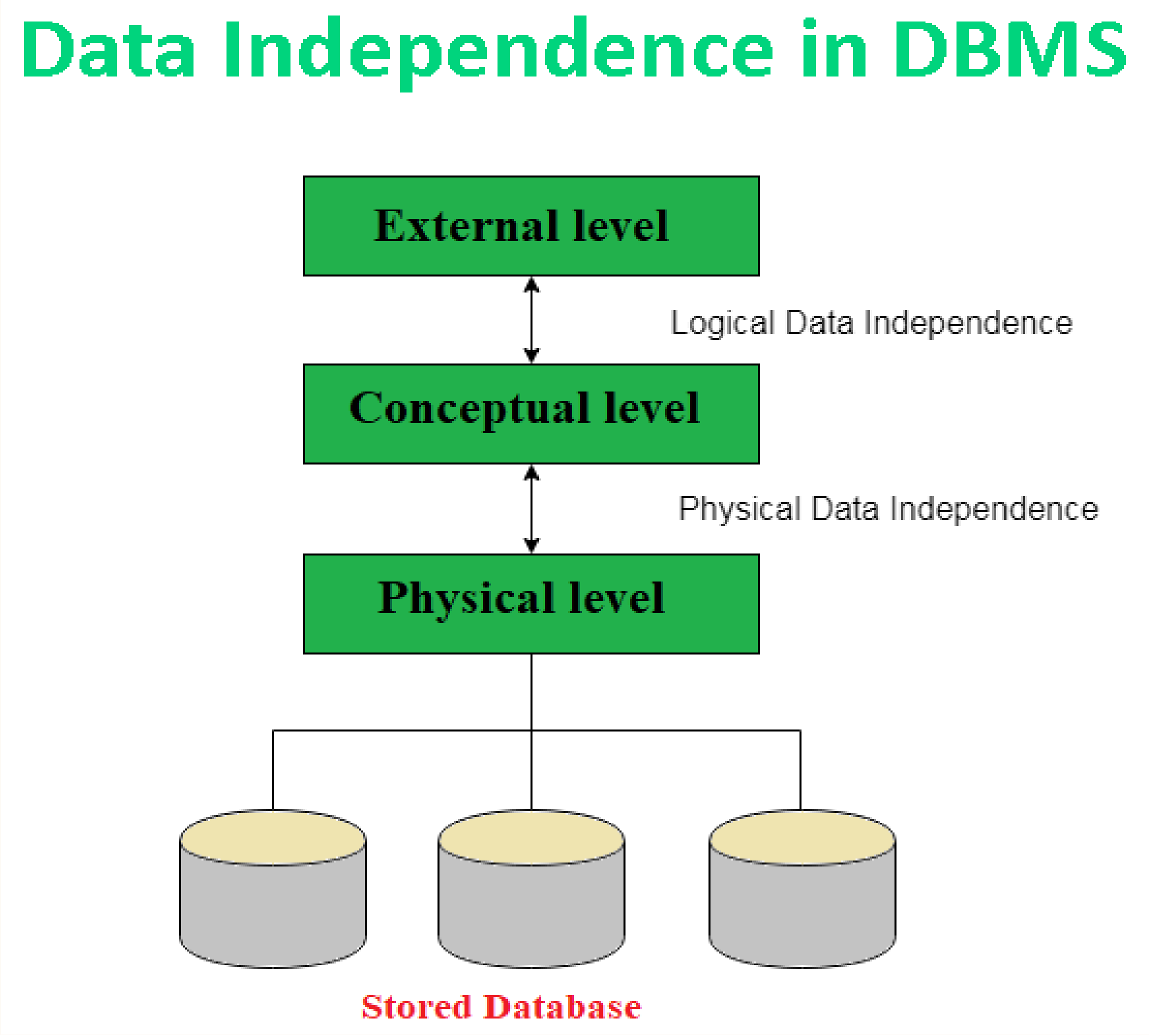
2. **Conceptual Schema (Logical Level):**
    - The conceptual schema represents the logical structure of the entire database, independent of any specific user views or applications.
    - It defines the entities, their attributes, and the relationships between them in the database model, such as an entity-relationship (ER) or relational model.
    - The conceptual schema serves as an intermediate level of abstraction that maps the external schemas to the internal schema.

3. **Internal Schema (Physical Level):**
    - The internal schema represents the physical storage and organization of data within the database system.
    - It describes how the data is stored on the underlying storage devices, including data structures, indexing methods, storage allocation, and access paths.
    - Changes made to the internal schema are transparent to both the conceptual schema and external schemas, preserving data independence and allowing for efficient data management.

DATA INDEPENDENCE

- Data independence can be explained using the three-schema architecture.

- Data independence is defined as the property of DBMS for being able to modify the schema at one level of the database system without affecting the schema at the next higher level.
- In DBMS there are two types of data independence
  1. Physical Data independence
  2. Logical Data Independence.


Data Independence in DBMS
External level
Logical Data Independence
Conceptual level
Physical Data Independence
Physical level
Stored Database

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.

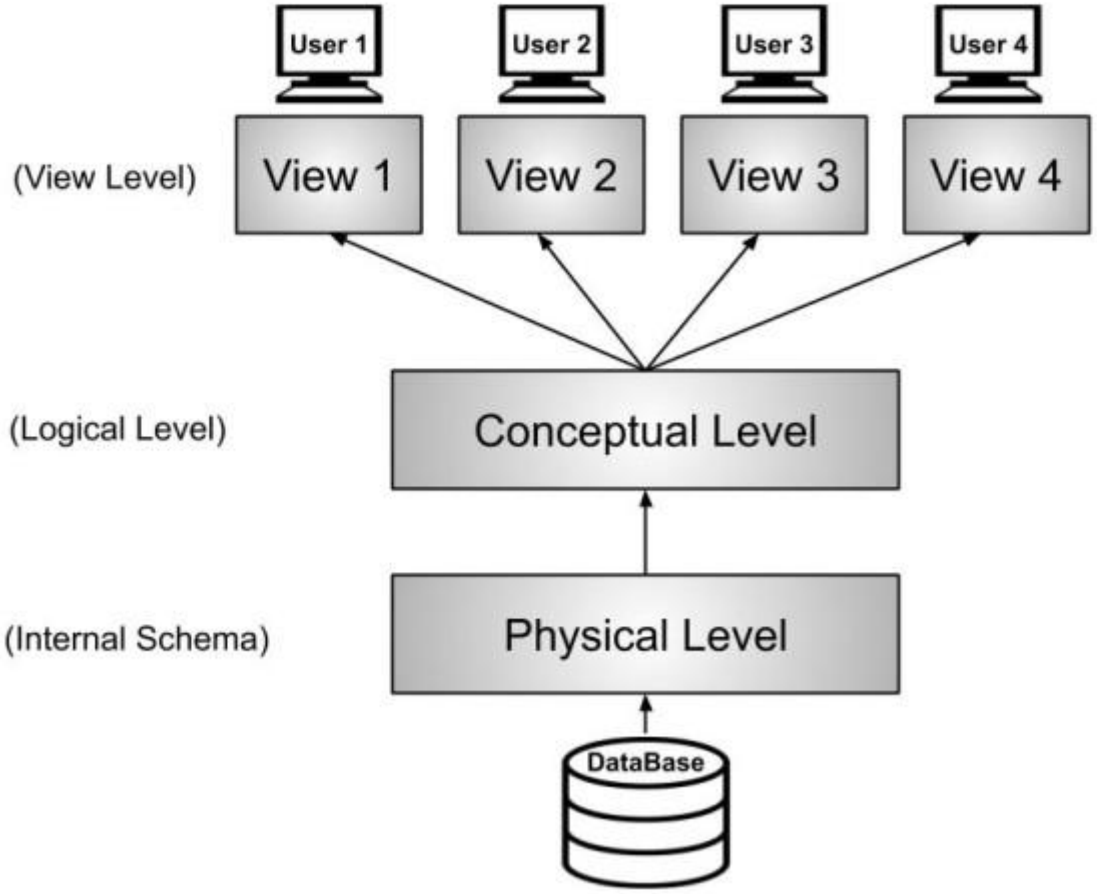
Three levels of Data Abstraction

- The three levels of data abstraction are:

  1.View Level :
  - This level tells the application about how the data should be shown to the user.

  2.Conceptual Level:
  - This level tells how the data is actually stored and structured. We have different data models by which we can store the data.

  3.Physical Level
  - As the name suggests, the Physical level tells us that where the data is actually stored i.e.; it tells the actual location of the data that is being stored by the user.



Levels of Data Abstraction

Database Languages

- Database languages are specialized programming languages used to communicate and manipulate databases.
- Database languages allow users to create, retrieve, update, and delete data within a database management system (DBMS).
- Database languages and interfaces are essential components in managing and interacting with databases.

Types of Database Languages:

There are mainly four types of database languages which are listed below:

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)
3. Data Control Language (DCL)
4. Transaction Control Language (TCL)

Database Interfaces

- Database interfaces serve as intermediaries between users or applications and the underlying database management system (DBMS).
- It provide a means for users to interact with databases, perform operations such as querying, inserting, updating, and deleting data, and manage the database structure.

Types of Database Interface:

There are many types of database interface provided by different DBMS most common are listed below:

1. Character User Interface (CLI)
2. Graphical User Interface (GUI)
3. Web Based Interfaces
4. Natural Language Interface

Command Line Interface (CLI)

- Command-line interfaces provide a text-based environment for interacting with databases. Users input commands directly into the terminal or command prompt to perform operations.
- CLI interfaces are often lightweight and efficient for performing quick tasks
- Examples : MySQL Command-Line Client, PostgreSQL's psql etc.

Graphical User Interface (GUI)

- GUI interfaces offer a visual environment with menus, buttons, and forms for interacting with databases.
- GUI interfaces are user-friendly and often preferred by users who are not comfortable with command-line interfaces.
- Examples : phpMyAdmin, pgAdmin, Microsoft SQL Server Management Studio (SSMS), and Oracle SQL Developer.

Web Based Interface (WBI)

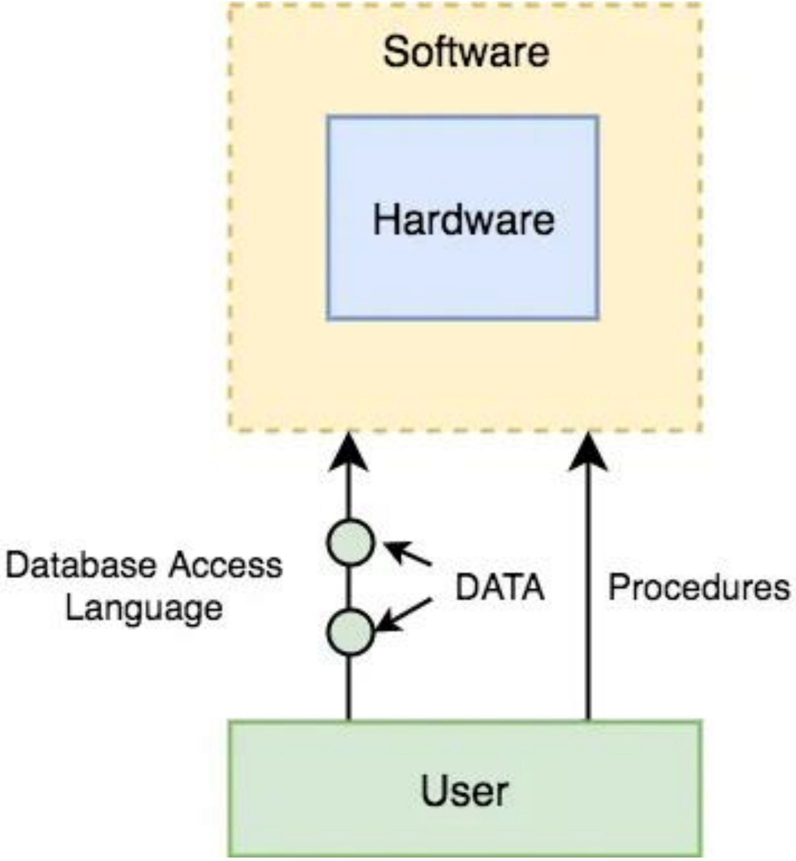
- Web-based interfaces allow users to interact with databases through a web browser.
- They are accessible from anywhere with an internet connection and often provide collaboration features for teams.
- Web interfaces may offer similar functionality to desktop GUI tools, including query execution, data visualization, and administrative tasks.
- Examples include phpMyAdmin, Adminer, Rockerbox, and Firebase Console.

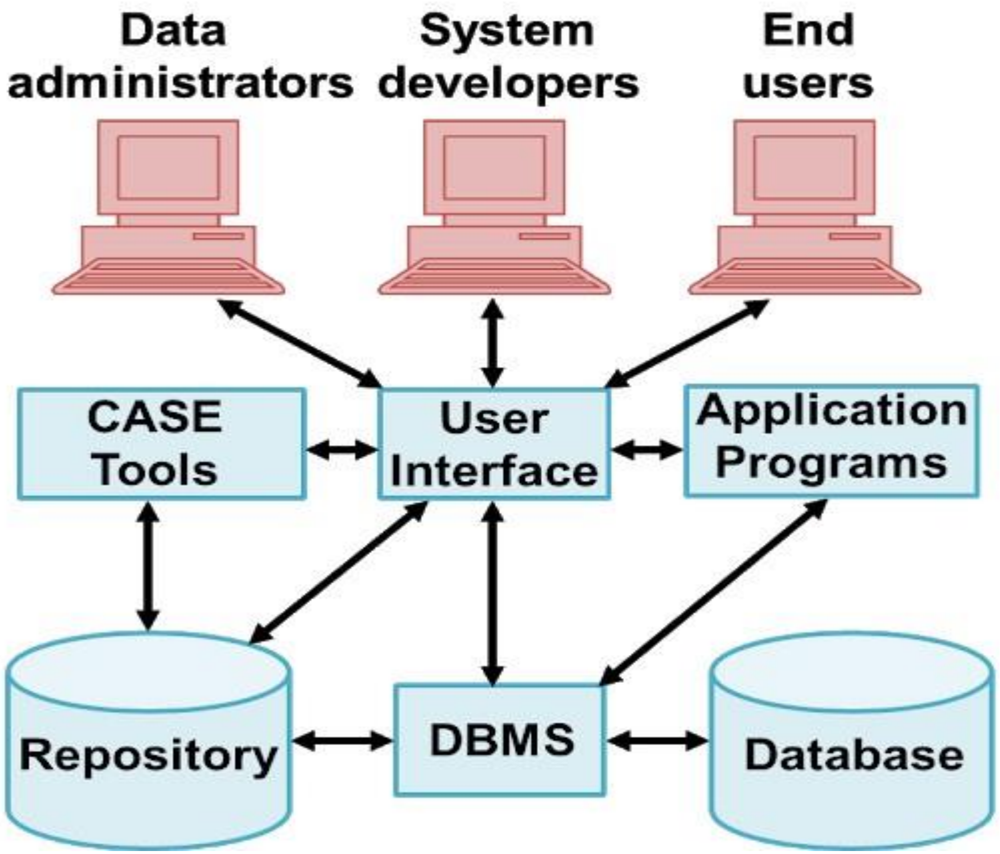
Natural Language Interface (NLI)

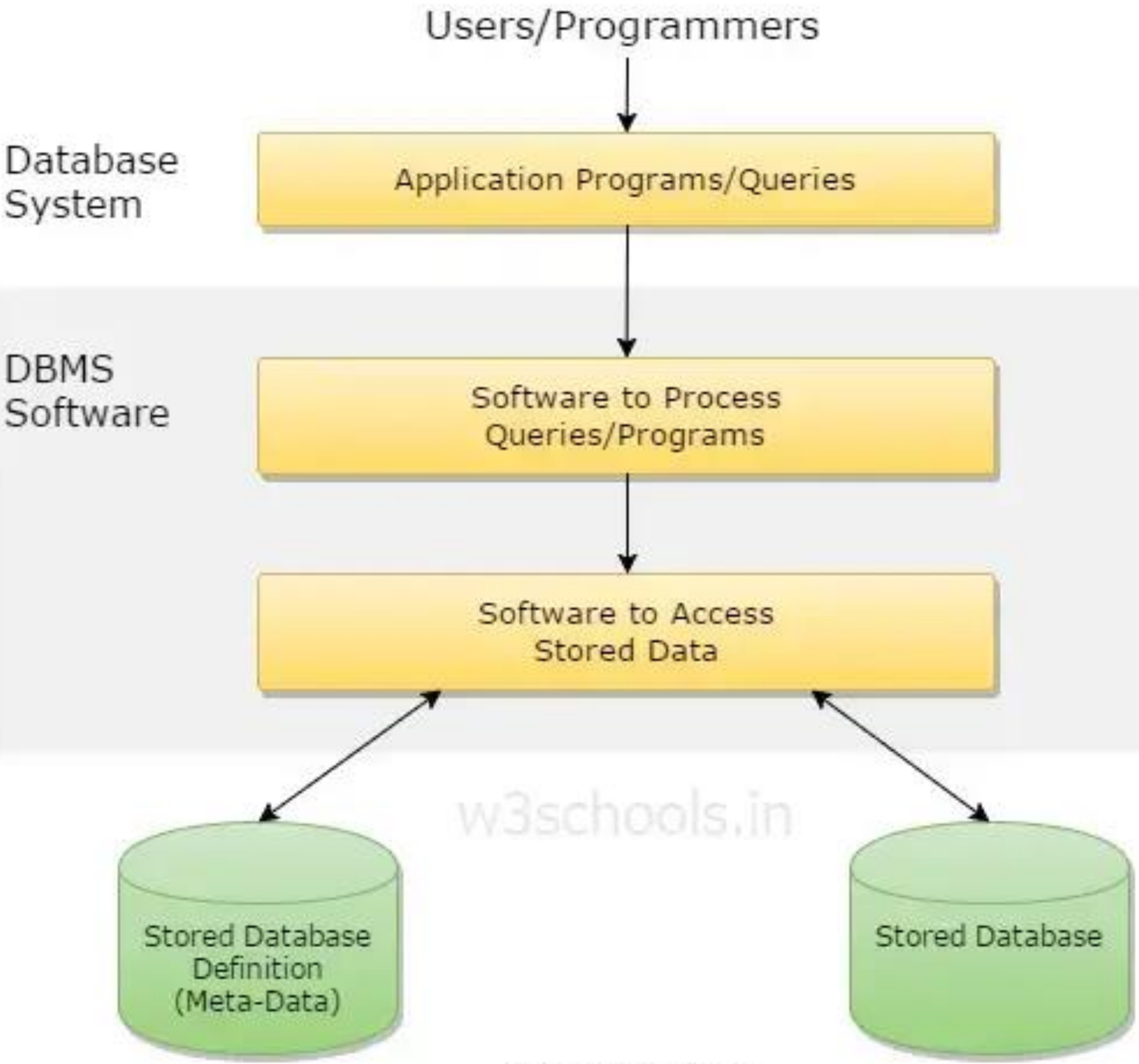
- This interface enables users to communicate with the database in the natural languages.
- The system will convert user commands into SQL instructions and carry them out if they are entered in plain English.
- Users who are unfamiliar with SQL commands might benefit from using natural language interfaces.
- Examples : Oracle Digital Assistant and Microsoft Power BI etc.

Database System Environment

- The Database System Environment refers to the overall infrastructure and components that support the functioning of a database management system (DBMS).
- It encompasses various elements, including hardware, software, networks, and other resources, that collectively provide a platform for storing, managing, and accessing data







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

Database System Environment

Components of Database System Environmnet

- **Hardware:**
  - ✓ This includes the physical devices that host the database system, such as servers, storage devices and networking equipment.

- **Software:**
  - ✓ This is the set of programs used to control and manage the overall database.
  - ✓ It includes the DBMS software itself, the operating system, the network software being used to share the data among users, and the application program used to access data in the DBMS.
  - ✓ It is the main component, as this the program which controls everything.
  - ✓ The software in DBMS environment includes operating system, database management system, application programs, support utility programs.

Components of Database System Environmnet (Con.)

- **Data :**
  - ✓ The main task of DBMS is to process the data.
  - ✓ It is the most important component of the database management system.

- **Procedure :**
  - ✓ These are the instructions and rules that assist on how to use the DBMS, and in designing and running the database, using documented procedures, to guide the users that operate and manage it.

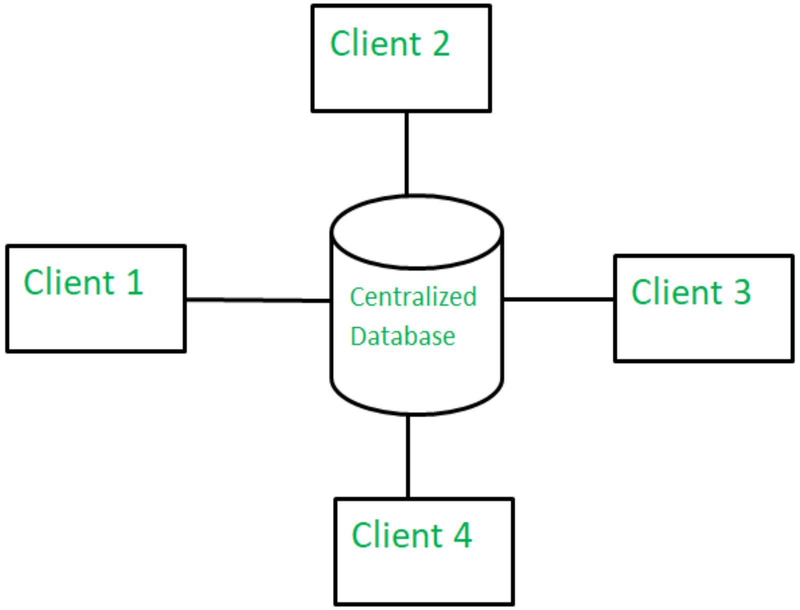
- **User :**
  - ✓ Users are the people are who control and manage the databases and perform different types of operations on the database in the DBMS.
  - ✓ The people includes databse administrator, software developer and End user.

Centralized and Client/Server Architecture for DBMS

- Centralized and client/server architectures are the two most commonly used models for organizing the components of a database management system (DBMS) that dictates how data is stored, retrieved, and updated.
- Each architectural model in DBMS has its own advantages and drawbacks, and the choice of architecture depends on factors such as scalability requirements, performance goals, fault tolerance needs, and the complexity of the application being developed.

Centralized Architecture for DBMS

- A centralized architecture for DBMS is one in which all data is stored on a single server, and all clients connect to that server in order to access and manipulate the data.
- This type of architecture is also known as a monolithic architecture.
- The main advantages of a centralized architecture is its simplicity - there is only one server to manage, and all clients use the same data.
- The main drawback is that, because all data is stored on a single server, if the server goes down for any reason, all clients lose access to the data.

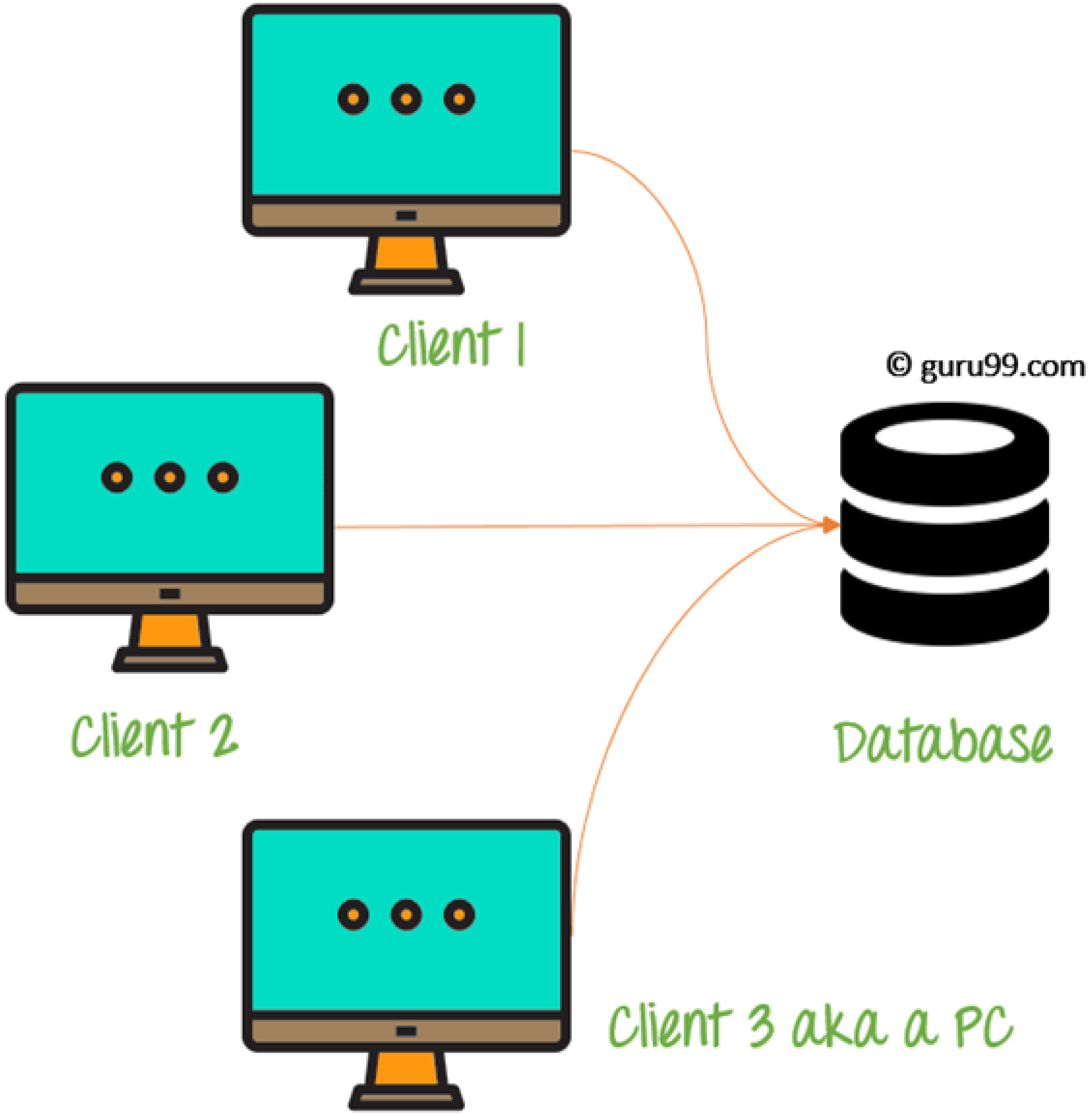


Client/Server Architecture for DBMS

- In a client/server architecture, the components of the database system are distributed across multiple computers or servers, typically divided into client and server roles.
- Clients are responsible for interacting with users and executing application logic, while servers manage data storage and processing.
- Client/server architectures are more scalable compared to centralized architectures because the workload can be distributed across multiple servers which leads to better performance.
- Client/server architecture can be more complex than a centralized system.
- On the basis of the client/server framework, two tier and three tier architecture are built.

Two Tier Architecture:

- The 2-tier Architecture is based on a client-server machine.
- In this type of architecture, the applications on client-side interact directly with the database present at the server-side.
- It helps to design, develop, implement, and maintain the database management system.



Advantages of 2-Tier Architecture

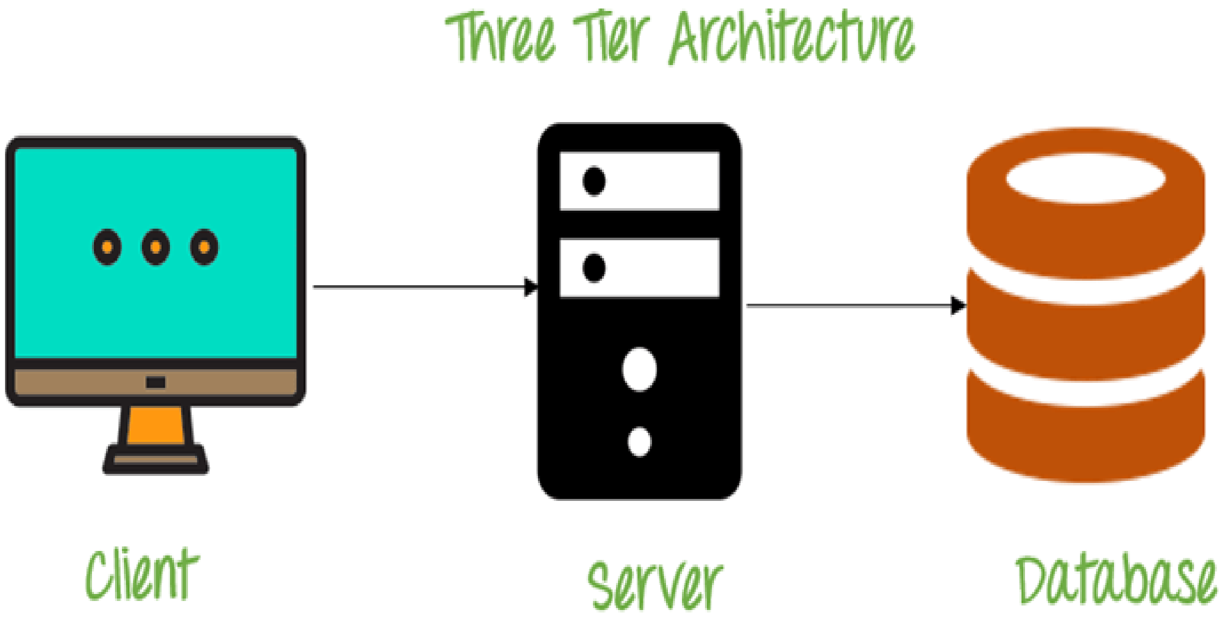
- Easy to understand as it directly communicates with the database.
- Requested data can be retrieved very quickly, when there is less number of users.
- Easy to modify
- Easy to maintain

Disadvantages of 2-Tier Architecture

- It would be time consuming, when there is huge number of users.
- The architecture would be less cost effective.

Three Tier Architecture:

- A 3 Tier Architecture in DBMS is the most popular client server architecture in DBMS in which the development and maintenance of functional processes, logic, data access, data storage, and user interface is done independently as separate modules.
- 3-Tier database Architecture design is an extension of the 2-tier client-server architecture.
- A 3-tier architecture has the following layers:
    1. Presentation layer (your PC, Tablet, Mobile, etc.)
    2. Application layer (server)
    3. Database Server



Advantages of 3-Tier Architecture

- Easy to maintain and modify: Any changes requested will not affect any other data in database. Application layers will do all the validations.
- Improved security: Since there is no direct access to the database, data security is increased.
- Good performance

Disadvantages of 3-Tier Architecture

- It is more complex.
- More effort is required in terms of hitting the database.

Classification of Database Management Systems

Database Management Systems (DBMS) can be classified based on various criteria such as data model, architecture, scalability, user number and database distributtion etc. Some of them are explained below:

- **DBMS based on the user number it supports:**

  - ✓ **Single-User DBMS:**
    - It is designed to support only one user accessing the database at a time. These systems are typically used in personal computing environments or small-scale applications where only one user interacts with the database.
    - **Examples**: Microsoft Access, SQLite.
  - ✓ **Multi-User DBMS:**
    - Multi-user DBMS allows multiple users to access and interact with the database simultaneously. These systems provide concurrency control mechanisms to manage access and ensure data consistency.
    - **Examples**: Oracle Database, SQL Server, MySQL, PostgreSQL.

Classification of Database Management Systems

- **DBMS based on Data Model are:**

✓ **Relational Database Management System (RDBMS):**

- In RDBMS, data is organized into tables consisting of rows and columns and the relationships between tables are established using keys.
- It support SQL (Structured Query Language) for querying and manipulating data and ACID (Atomicity, Consistency, Isolation, Durability) properties ensure data integrity.
- **Examples**: MySQL, PostgreSQL, Oracle Database, SQL Server.

✓ **Object-Oriented Database Management System (OODBMS):**

- OODBMS stores data in the form of objects, which consist of attributes and methods. It extends the object-oriented programming paradigm to database management.
- It supports complex data types, inheritance, encapsulation, and polymorphism.
- **Examples**: db4o, ObjectDB, Versant Object Database.

Classification of Database Management Systems

- **DBMS Based on Architecture:**

  ✓ **Client/Server DBMS:**
  - These systems separate the database management tasks between client and server machines.
  - **Examples** : MySQL, SQL Server, Oracle Database.

  ✓ **Peer-to-Peer DBMS:**
  - These systems distribute database management tasks across multiple peer nodes with no centralized control.
  - **Examples** : CouchDB, Cassandra.

Classification of Database Management Systems

- **DBMS Based on database distribution:**

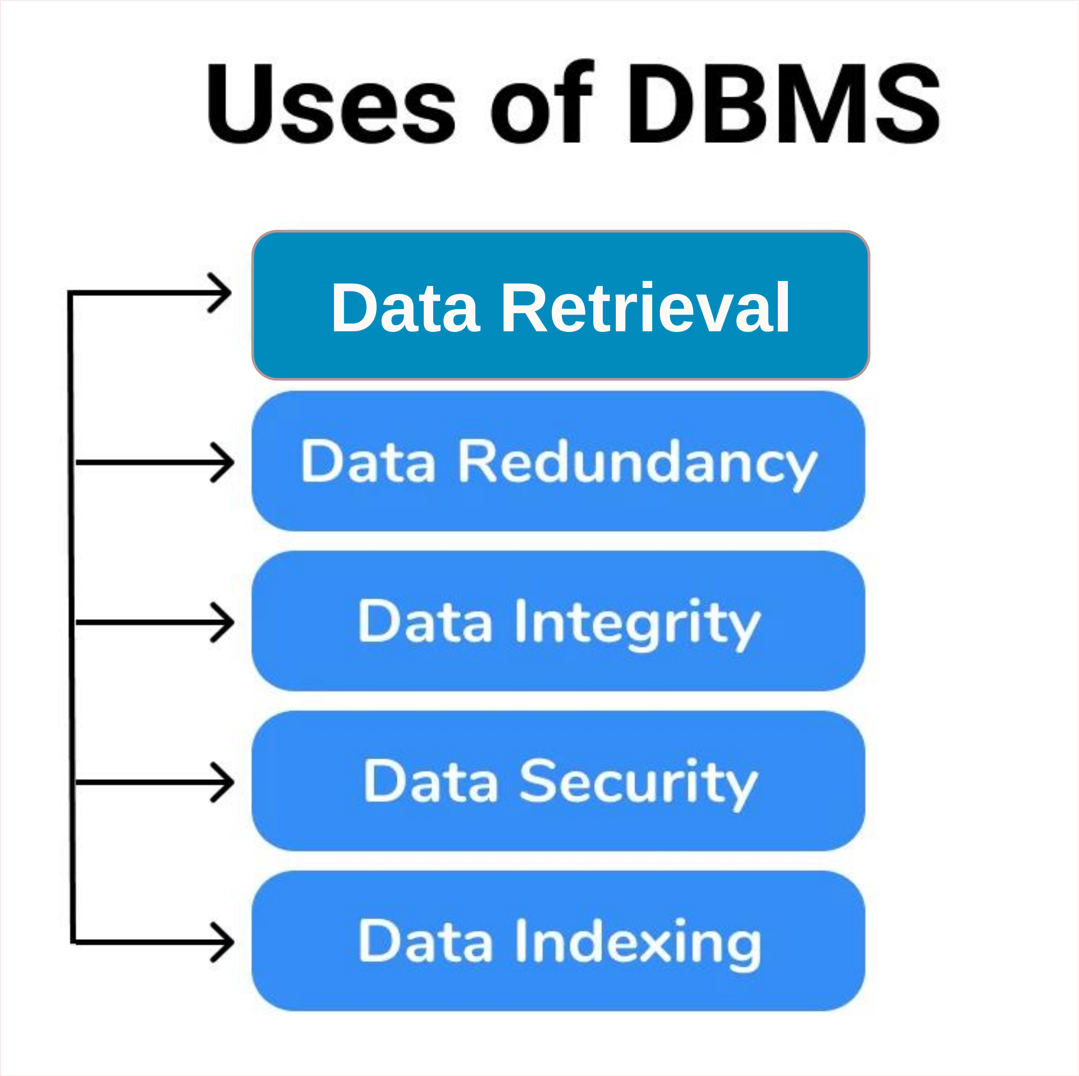
  ✓ **Centralized DBMS:**
  - In a centralized DBMS, all database components, including data storage, processing, and management, reside on a single server or computing instance.
  - **Examples**: Traditional relational database systems like MySQL, PostgreSQL.

  ✓ **Distributed DBMS (DDBMS):**
  - Distributed DBMS distribute data across multiple nodes or servers, allowing for parallel processing and improved scalability and fault tolerance.
  - **Examples**: Apache Cassandra, MongoDB, Google Spanner.

Other Important Topics

PURPOSE OF DATABASE MANAGEMENT SYSTEM

The main purpose of the Database Management System is to store, retrieve, and run queries on data.



Uses of DBMS

Data Retrieval

Data Redundancy

Data Integrity

Data Security

Data Indexing

APPLICATION AREAS OF DATABASE SYSTEM

Database systems are widely used in different areas because of their numerous advantages. Some of the most common database applications are:

TELECOM

There is a database to keep track of the information regarding calls made, network usage, customer details etc. Without database systems it is hard to maintain that huge amount of data that keeps updating every millisecond

AIRLINES

To travel through airlines, we make early reservations, this reservation information along with flight schedule is stored in database.

INDUSTRY

- Where it is a manufacturing unit, warehouse or distribution Centre, each one needs a database to keep the records of ins and outs.
- Without a centralized place to store all this information, you have no clue what is occurring within your business.

BANKING SYSTEM

For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of database management systems.

SALES

To store customer information, production information and invoice details.

HUMAN RESOURCE

Organizations use databases for storing information about their employees, salaries, benefits, taxes, and for generating salary checks.

Q. List four significant differences between a file processing system and a DBMS.
Ans.

Some main differences between a database management system and a file-processing system are :

- Both systems contain a collection of data and set of programs which access that data. A database management system coordinates both the physical and the logical access to the data, whereas a file processing system coordinates only the physical access.
- A DBMS reduces the amount of data duplication by ensuring that a physical piece of data is available to all programs authorized to have access to it, where as data written by one program in a file processing system may not be readable by another program.
- A DBMS is designed to allow flexible access to data (i.e. queries), whereas a file-processing system is designed to allow predetermined access to data (ie., compiled programs).
- A database management system is designed to coordinate multiple users accessing the same data at the same time. A file processing system is usually designed to allow one or more programs to access different data files at the same time. In a file processing system, a file can be accessed by two programs concurrently only if both programs have read-only access to the file.

Q. Explain the difference between Physical and Logical Data Independence.

Ans.

- **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** 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 hide this change from the program.

Q. List six major steps that you would take in setting up a database for a particular enterprise.

Ans.

Six major steps in setting up a database for a particular enterprise are:

- Define a model containing all appropriate types of data and data relationships.
- Define the integrity constraints on the data.
- Define the conceptual schema for the model.
- Define the physical level.
- For each known problem to be solved on a regular basis (e.g., weekly inventory), define a view of the database and write the necessary application programs.
- Create and initialize the database.

End