Unit 1

Syllabus:

UNIT-I (15 hrs) Basics of DBMS: Database Concept, Characteristics and architecture of DBMS, Database users, 3-tier architecture of DBMS-its advantages over 2-tier, Introduction of Parallel, Distributed Databases, Mobile databases and Cloud databases. Data independence. Physical data organization, Indexing-introduction and types of indexing.

## Characteristics of database:

1. **Real-World Entity**: DBMS represents real-world entities and their relationships, making it easier to model complex data structures. For example, a student database might include entities like students, courses, and instructors.

2. **Self-Describing Nature**: A DBMS contains not only the database itself but also metadata, which is data about data. This metadata describes the structure of the database, making it self-explanatory and easily understandable.

3. **Atomicity of Operations (ACID Properties):** DBMS ensures that all operations (transactions) are atomic, consistent, isolated, and durable. which maintains data integrity and reliability.

4. **Concurrent Access:** Multiple users can access and modify the database simultaneously without affecting each other's work. This is crucial for systems like banking or airline reservations.

5. **Data Integrity:** DBMS enforces data integrity constraints to ensure the accuracy and consistency of data. For example, it can enforce rules like unique IDs for each student.

6**. Security:** DBMS provides security features to protect data from unauthorized access. This includes user authentication, access controls, and encryption.

7**. Data Abstraction and Independence:** DBMS provides a level of abstraction between the physical storage of data and the logical view of the data. This means changes in the database structure do not affect the application programs.

8**. Support for Multiple Views**: DBMS allows different users to have different views of the database according to their needs. For example, a student might see their grades, while an instructor sees the grades of all students.

## Types of Database Users:

In a Database Management System (DBMS), there are several types of users, each with distinct roles and responsibilities. Here are the main categories:

1**. Database Administrators (DBA):**

- **Role**: DBAs are responsible for managing the overall database system. They define the schema, control access, ensure data integrity, and handle backup and recovery.

- **Responsibilities**: Creating user accounts, setting permissions, monitoring performance, and maintaining security.

2. **Database Designers**:

- **Role**: They design the structure of the database, including tables, indexes, views, and constraints.

- **Responsibilities**: Ensuring the database meets the needs of all user groups and is optimized for performance.

**3. System Analysts**:

- **Role**: They analyse the requirements of end users and ensure that the database system meets these needs.

- **Responsibilities**: Bridging the gap between users and developers, and ensuring the system's functionality aligns with user requirements.

4. **Application Programmers**:

- **Role**: Also known as back-end developers, they write the code for applications that interact with the database.

- **Responsibilities**: Developing and maintaining application programs, ensuring efficient data retrieval and manipulation.

**5. Naive/Parametric Users**:

- **Role**: These are end users who interact with the database through predefined applications without needing to understand the underlying database structure.

- **Responsibilities**: Performing routine tasks like data entry and retrieval using user-friendly interfaces.

6. **Sophisticated Users**:

- **Role**: These users are familiar with the database and can write their own queries to retrieve data.

- **Responsibilities**: Conducting complex data analysis and generating reports.

7. **Casual Users**:

- **Role**: These users occasionally interact with the database, often for specific queries or reports.

- **Responsibilities**: Accessing data as needed, typically through query languages like SQL.

| **User Type** | **Description** |
| --- | --- |
| **Database Administrator (DBA)** | **Manages database** schema, security, backups, and technical support. |
| **Naive / Parametric Users** | Unsophisticated users who **interact with database applications**. |
| **System Analysts** | **Analyse user requirements** and ensure they are satisfied. |
| **Sophisticated Users** | Familiar with databases, write custom queries, and **develop applications.** |
| **Database Designers** | **Create the database** schema, tables, indexes, and constraints. |
| **Application Programmers** | **Write code for application programs** that interact with the database. |

## Parallel database:

## A system that uses multiple processors to manage and process data simultaneously is called parallel database.

## Data is divided across multiple nodes for parallel processing

**Advantages:**

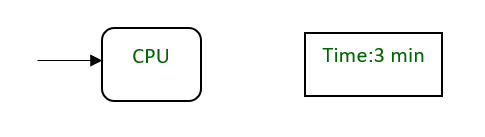
1. **Performance Improvement –**  
   By connecting multiple resources like CPU and disks in parallel we can significantly increase the performance of the system.
2. **High availability –**  
   In the parallel database, nodes have less contact with each other, so the failure of one node doesn’t cause for failure of the entire system. This amounts to significantly higher database availability.
3. **Proper resource utilization –**  
   Due to parallel execution, the CPU will never be idle. Thus, proper utilization of resources is there.
4. **Increase Reliability –**  
   When one site fails, the execution can continue with another available site which is having a copy of data. Making the system more reliable.

**Disadvantages:**

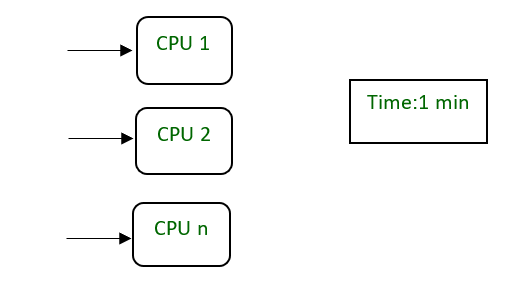
1. Complexity in design and management
2. Costly implementation

**Diagram**:

**fig. A CPU requires 3 minutes to execute a process**



**fig. ‘n’ CPU requires 1 min to execute a process by dividing into smaller tasks**



**Examples –**

Teradata, Oracle Parallel Server

## Distributed Databases

* A database that is spread across multiple locations, either on the same network
* Data is distributed across multiple nodes but appears as a single database to users
* Each node can be located in different geographical locations
* It is designed to store data across different locations, providing transparency to users who perceive it as a single database.

**Advantages**

* 1. High availability and fault tolerance
  2. Better load balancing
  3. Scalability by adding more nodes

**Disadvantages**

* 1. Complexity in data synchronization
  2. Potential for data consistency issues

**Examples**

- MongoDB, Apache Cassandra, Google Spanner

**Diagram**

