**DAY 18**

**NAME: ANNIE JOHN**

**USER ID:27739**

**Batch: 25VID0885_DC_Batch4**

## TITLE: DATABASE ADMINISTRATION

- **What is a Database?**

A **database** is a structured collection of interrelated data that allows efficient **retrieval**, **insertion**, **deletion**, and **management** of data. Data is typically organized in the form of **tables**, **views**, **schemas**, and **reports**.

**Example:** A university database may contain information about students, faculty, and administrative staff, organized to allow quick access and management.

- **What is DBMS?**

A **Database Management System (DBMS)** is software that enables users to create, manage, and interact with databases. It ensures data is organized, stored, and retrieved efficiently, while also providing **security**, **concurrency control**, and **data integrity**.

- **Key Features of DBMS**

  - **Data Modelling:** Tools for designing the structure and relationships of data.
  - **Data Storage & Retrieval:** Efficient methods to store, search, and retrieve data.
  - **Concurrency Control:** Manages simultaneous access by multiple users without conflicts.
  - **Data Integrity & Security:** Enforces rules to maintain accuracy and restrict unauthorized access.
  - **Backup & Recovery:** Safeguards data by enabling restoration after system failures.

- **DBMS can be classified into two types:**
  - Relational Database Management System (RDBMS) - Data is organized in the form of tables and each table has a set of rows and columns. The data are related to each other through primary and foreign keys.
  - Non-Relational Database Management System (NoSQL or Non-SQL)- Data is organized in the form of key-value pairs, documents, graphs, or column-based. These are designed to handle large-scale, high-performance scenarios.
- **Types of DBMS**

1. **Relational DBMS (RDBMS):**
    - Stores data in **tables (relations)** with rows and columns.
    - Uses **primary and foreign keys** to define relationships.
    - Uses **SQL (Structured Query Language)** for data manipulation.
    - **Examples:** MySQL, Oracle, PostgreSQL, Microsoft SQL Server.
2. **Non-Relational DBMS (NoSQL):**
    - Stores data in formats like **key-value pairs, documents, graphs**, or **columns**.
    - Ideal for **large-scale**, **distributed**, and **high-performance** applications.
    - **Examples:** MongoDB (document), Redis (key-value), Cassandra (column), Neo4j (graph).
3. **Object-Oriented DBMS (OODBMS):**
    - Stores data as **objects**, similar to those in object-oriented programming languages.
    - Supports **complex data types and relationships**.
    - Useful for applications that require **tight integration between application and database** logic.

- **Database Languages**

1. **Data Definition Language (DDL)-**DDL is used to **define or change the structure of the database**, like creating or modifying tables.

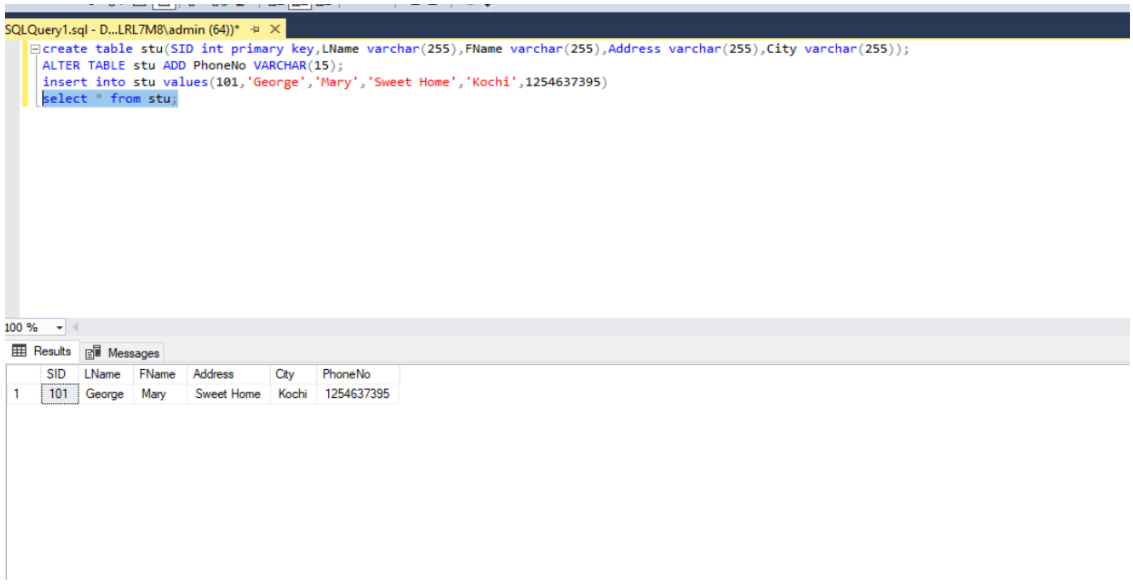
- **Common Commands:**

a. CREATE – to create a new table or database
b. ALTER – to modify an existing table (add or remove columns)
c. DROP – to delete a table or database
d. TRUNCATE – to delete all data from a table but keep the structure


2. **Data Manipulation Language (DML)**-DML is used to **manipulate data** stored in the database, like adding, editing, or deleting records.

➢ **Common Commands:**

a. INSERT – to add new data
b. SELECT – to fetch data
c. UPDATE – to modify existing data
d. DELETE – to remove data



3. **Data Control Language (DCL)-**DCL is used to **control access to the data**, like giving or taking away user permissions.

- **Common Commands:**
  a. GRANT – to give permissions to a user
  b. REVOKE – to remove permissions

4. **Transaction Control Language (TCL)-**TCL is used to **manage transactions** in the database, ensuring data is consistent and correct.

- **Common Commands:**
  a. COMMIT – to save the changes made during the transaction
  b. ROLLBACK – to undo changes if there's an error
  c. SAVEPOINT – to set a point in a transaction to roll back to

- **Advantages of DBMS:**

1. Data Redundancy Control-Minimizes duplication by centralizing data storage.
2. Data Consistency-Updates made in one place are reflected across the system, ensuring data accuracy.
3. Improved Data Security-Access can be restricted using user roles, passwords, and permissions.
4. Data Sharing-Multiple users can access the data concurrently with proper synchronization.
5. Backup and Recovery-Automatic backup and recovery mechanisms protect against data loss.
6. Data Integrity-Enforces rules to maintain the correctness and validity of data.
7. Efficient Query Processing-Optimized to retrieve and manipulate data quickly using query languages (e.g., SQL).
8. Data Independence-Application programs are insulated from changes in data structure.

- **Disadvantages of DBMS:**

1. High Cost-Software, hardware, and skilled personnel can be expensive.

2. Complexity-Setup, configuration, and maintenance require expert knowledge.
3. Performance Overhead-For small or simple tasks, a DBMS might be slower due to its general-purpose design.
4. Size-DBMS software can be large and consume significant system resources.
5. Vulnerability to Failure-A failure in the DBMS can affect the entire data system, though recovery mechanisms help mitigate this.
6. Frequent Maintenance-Requires regular updates, tuning, and monitoring for optimal performance.

- **RDBMS (Relational Database Management System)**

  An **RDBMS** is a type of **DBMS** that stores data in **tables** (also called **relations**) and allows relationships between them using **keys** (like primary and foreign keys). It uses **SQL** (Structured Query Language) to access and manage the data.

  **Popular RDBMS Examples:** MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, IBM Db2

  - **Primary Key**

    A **Primary Key** is a column (or set of columns) in a table that **uniquely identifies each row** in that table.

    **Key Features:**

    a. Must be **unique** for each record.
    b. **Cannot contain NULL** values.
    c. Each table can have **only one** primary key.
    d. Often used to establish relationships with other tables.

  - **Foreign Key**

    A **Foreign Key** is a column (or set of columns) that creates a **link between two tables**. It refers to the **primary key** in another table.

    **Key Features:**

    a. Used to **enforce referential integrity** between tables.
    b. Can contain **duplicate** values.
    c. Can contain **NULL** values unless specified otherwise.
    d. A table can have **multiple foreign keys**.

  - **Advantages of RDBMS:**

1. RDBMS ensures data accuracy and consistency through the use of primary and foreign keys.

2. It allows multiple users to access and work with data at the same time without conflicts.
3. It uses SQL, a standard and powerful language, to manage and query data easily.

- **Disadvantages of RDBMS:**

1. RDBMS software can be expensive to purchase, set up, and maintain.
2. It can become slow and less efficient when handling extremely large or complex datasets.
3. The rigid table structure makes it hard to store and manage unstructured data like images or documents.