TOPS TECHNOLOGY

Module 4 – Introduction to DBMS

Presented By :

Nandni Vala

Transaction Control Language (TCL)

1.What is the purpose of the COMMIT and ROLLBACK commands in SQL?

- **COMMIT Command**
- **Purpose**: The **COMMIT** command is used to **save all changes** made during a transaction to the database permanently.
- Once a **COMMIT** is issued, all modifications (such as **INSERT**, **UPDATE**, **DELETE**) made to the database are permanent and cannot be undone.
- **Key Points**:
- A **COMMIT** marks the end of a transaction.
- After a **COMMIT**, the changes are visible to other users and cannot be rolled back.
- It ensures that all changes made during the transaction are saved to the database.

- **Example**:
- BEGIN TRANSACTION;
- UPDATE employees
- SET salary = salary + 5000
- WHERE department = 'HR';
- COMMIT;
- **ROLLBACK Command**
- **Purpose**: The **ROLLBACK** command is used to **undo** or **revert** all changes made during a transaction, returning the database to its state before the transaction started.
- A **ROLLBACK** can be issued at any point during a transaction if an error occurs, or if the user decides not to save the changes.
  - **Key Points**:
- A **ROLLBACK** undoes all changes made since the start of the transaction.
- It is used to maintain database consistency in case of an error or if a transaction should not be finalized.

2.Explain how transactions are managed in SQL databases.

- In SQL databases, transactions are used to group multiple operations into a single, atomic unit of work. This ensures data integrity and consistency. Transactions are managed using the following commands:
- **BEGIN TRANSACTION**: Marks the beginning of a transaction.
- **COMMIT**: Saves all changes made during the transaction permanently to the database.
- **ROLLBACK**: Undoes all changes made during the transaction, reverting the database to its state before the transaction started.
- **SAVEPOINT**: Sets a point within a transaction that allows partial rollback to that point.
- **RELEASE SAVEPOINT**: Removes a savepoint within a transaction.
- Transactions follow the **ACID** properties:
- **Atomicity**: All or nothing — changes are either fully committed or fully rolled back.
- **Consistency**: The database maintains a valid state.
- **Isolation**: Transactions are isolated from each other to prevent interference.
- **Durability**: Once committed, changes are permanent and survive system failures.