SQL Commands

Here's an **overview of the different types of SQL commands**, classified under **DDL, DML, DQL, TCL, and DCL**, with explanations of each:

---

1. DDL (Data Definition Language)

Purpose:

DDL commands are used to define, alter, or drop the structure of database objects such as tables, schemas, indexes, etc. These commands deal with the **schema** or structure of the database.

Key Commands:

- **CREATE**: To create a new table, database, or any database object.
  - Example: `CREATE TABLE employees (id INT, name VARCHAR(50));`
- **ALTER**: To modify the structure of an existing table or database object.
  - Example: `ALTER TABLE employees ADD COLUMN age INT;`
- **DROP**: To delete tables, indexes, or databases.
  - Example: `DROP TABLE employees;`
- **TRUNCATE**: To delete all rows from a table without logging individual row deletions.
  - Example: `TRUNCATE TABLE employees;`

Characteristics:

- Changes made by DDL are usually permanent.
- Impacts the database structure/schema, not the data.

---

2. DML (Data Manipulation Language)

Purpose:

DML commands are used to **manipulate data** within database objects (e.g., tables).

Key Commands:

- **INSERT**: To add new rows to a table.
  - Example: `INSERT INTO employees (id, name) VALUES (1, 'John');`
- **UPDATE**: To modify existing data in a table.
  - Example: `UPDATE employees SET name = 'Jane' WHERE id = 1;`
- **DELETE**: To remove rows from a table.
  - Example: `DELETE FROM employees WHERE id = 1;`

Characteristics:

- DML commands are logged and can be rolled back or committed.
- Focuses only on the data, not the structure.

---

3. DQL (Data Query Language)

Purpose:

DQL commands are used to **query and fetch data** from the database. It involves retrieving specific or all data from one or more tables.

Key Command:

- **SELECT**: To fetch data from a database.
  - Example: `SELECT * FROM employees WHERE age > 30;`

Characteristics:

- Only retrieves data without altering it.
- Often paired with filters, aggregations, and joins to process data efficiently.

---

4. TCL (Transaction Control Language)

Purpose:

TCL commands manage **transactions** in a database, ensuring data integrity and consistency. Transactions represent a unit of work.

Key Commands:

- **COMMIT**: To save changes made in the current transaction.
  - Example: `COMMIT;`
- **ROLLBACK**: To undo changes made in the current transaction.
  - Example: `ROLLBACK;`
- **SAVEPOINT**: To set a point within a transaction to which a rollback can revert.
  - Example: `SAVEPOINT sp1;`
- **SET TRANSACTION**: To define the properties of a transaction.
  - Example: `SET TRANSACTION READ ONLY;`

Characteristics:

- Operates on transactions, allowing partial rollbacks or commits.
- Critical for ensuring data consistency.

---

5. DCL (Data Control Language)

Purpose:

DCL commands are used to control **permissions and access** to the database.

Key Commands:

- **GRANT**: To give privileges to users.
  - Example: `GRANT SELECT ON employees TO user1;`
- **REVOKE**: To remove privileges from users.
  - Example: `REVOKE SELECT ON employees FROM user1;`

Characteristics:

- Manages user permissions and access rights.
- Ensures database security.

Summary Table

	Category	Purpose	Examples of Commands
1	**DDL**	Define database structure	`CREATE`, `ALTER`, `DROP`, `TRUNCATE`
2	**DML**	Manipulate data	`INSERT`, `UPDATE`, `DELETE`
3	**DQL**	Query data	`SELECT`
4	**TCL**	Manage transactions	`COMMIT`, `ROLLBACK`, `SAVEPOINT`, `SET TRANSACTION`
5	**DCL**	Control permissions	`GRANT`, `REVOKE`

+ Neu