05 SQL Commands

SQL (Structured Query Language) statements are generally categorized into five main types

01 Data Definition Language (DDL)

- **Purpose:** Defines the structure of the database
- **Common Commands**
  - `CREATE` : Creates a new table, database, index, or view
  - `ALTER` : Modifies an existing database object, such as a table or view
  - `DROP` : Deletes an existing table, database, or view
  - `TRUNCATE` : Removes all records from a table, but not the table itself
- **CREATE**

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Department VARCHAR(50)
);
```

- **ALTER**

```
ALTER TABLE Employees
ADD DateOfBirth DATE;
```

- **DROP**

```
DROP TABLE Employees;
```

- **TRUNCATE**

```
TRUNCATE TABLE Employees;
```

02 Data Manipulation Language (DML)

- **Purpose:** Handles the manipulation of data within tables
- **Common Commands**

    - `SELECT`: Retrieves data from one or more tables
    - `INSERT`: Adds new rows of data to a table
    - `UPDATE`: Modifies existing data within a table
    - `DELETE`: Removes data from a table
- **SELECT**

```
SELECT FirstName, LastName, Department
FROM Employees
WHERE Department = 'HR';
```

- **INSERT**

```
INSERT INTO Employees (EmployeeID, FirstName, LastName, Department)
VALUES (1, 'John', 'Doe', 'HR');
```

- **UPDATE**

```
UPDATE Employees
SET Department = 'Finance'
WHERE EmployeeID = 1;
```

- **DELETE**

```
DELETE FROM Employees
WHERE EmployeeID = 1;
```

03 Data Control Language (DCL)

- **Purpose:** Manages access rights to the data within the database
- **Common Commands**
    - `GRANT`: Gives users access privileges to the database
    - `REVOKE`: Removes access privileges from users
- **GRANT**

```
GRANT SELECT, INSERT ON Employees TO User1;
```

- **REVOKE**

```
REVOKE INSERT ON Employees FROM User1;
```

04 Transaction Control Language (TCL)

- **Purpose:** Manages the changes made by DML statements, ensuring data integrity
- **Common Commands**
  - `COMMIT`: Saves the changes made by the current transaction
  - `ROLLBACK`: Reverts the changes made by the current transaction
  - `SAVEPOINT`: Sets a save point within a transaction that you can rollback to
- **COMMIT**

```
BEGIN TRANSACTION;

UPDATE Employees
SET Department = 'IT'
WHERE EmployeeID = 2;

COMMIT;
```

- **ROLLBACK**

```
BEGIN TRANSACTION;

UPDATE Employees
SET Department = 'IT'
WHERE EmployeeID = 2;

ROLLBACK;
```

- **SAVEPOINT**

```
BEGIN TRANSACTION;

SAVEPOINT SavePoint1;

UPDATE Employees
SET Department = 'Marketing'
WHERE EmployeeID = 3;

ROLLBACK TO SavePoint1;
```

05 Data Query Language (DQL)

- **Purpose:** Queries the database for information
- **Common Command**

- `SELECT`: Though part of DML, `SELECT` is sometimes categorized under DQL because it is used primarily for querying the database

- **SELECT**

```
SELECT FirstName, LastName
FROM Employees
WHERE Department = 'IT';
```

DELETE vs TRUNCATE

DELETE Statement

- **Functionality:** Deletes rows from a table based on a condition specified in a `WHERE` clause. If no condition is provided, all rows are deleted

```
DELETE FROM table_name WHERE condition;
```

```
DELETE FROM Employees WHERE Department = 'HR';
```

TRUNCATE Statement

- **Functionality:** Deletes all rows from a table without logging individual row deletions

```
TRUNCATE TABLE table_name;
```

```
TRUNCATE TABLE Employees;
```

Feature	DELETE	TRUNCATE
Conditional Deletion	Yes (using `WHERE` clause)	No (deletes all rows)
Transaction-Safe	Yes (can be rolled back)	No (except in some databases)
Triggers	Fires `DELETE` triggers	Does not fire `DELETE` triggers
Speed	Slower for large data sets	Faster for large data sets
Logging	Logs individual row deletions	Minimal logging
Auto-Increment	Does not reset auto-increment	Resets auto-increment

Feature	DELETE	TRUNCATE
Reset	counters	counters