Data Definition Language (DDL) and Integrity Constraints in SQL

DDL (Data Definition Language) commands are used to define and manage the structure of database objects such as databases, tables, and constraints. The most common DDL commands are: CREATE, ALTER, DROP, and TRUNCATE.

1. CREATE DATABASE

- Purpose: Creates a new, empty database to store tables and other database objects.
- Syntax:

```
CREATE DATABASE DatabaseName;
```

- Example:

```
CREATE DATABASE Company;
```

2. CREATE TABLE

- Purpose: Defines a new table, specifying columns, data types, and constraints.
- Syntax:

```
CREATE TABLE TableName (
  Column1 DataType [Constraint],
  Column2 DataType [Constraint],
  ...
);
```

- Example:

```
CREATE TABLE Employees (
  EmployeeID INT PRIMARY KEY,
  FirstName VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL,
  Department VARCHAR(50),
  Salary DECIMAL(10, 2) CHECK (Salary > 0)
);
```

- Explanation: This creates an Employees table with a primary key on EmployeeID, required fields for FirstName and LastName, and a check constraint on Salary.

3. ALTER TABLE

- Purpose: Modifies an existing table by adding, modifying, or dropping columns.

- Examples:

```
ALTER TABLE Employees
ADD DateOfBirth DATE;

ALTER TABLE Employees
MODIFY Salary DECIMAL(12, 2);

ALTER TABLE Employees
DROP COLUMN Department;
```

- Explanation: These commands modify the Employees table by adding a new column, adjusting a data type, and removing a column.

4. DROP TABLE

- Purpose: Completely removes a table, including its structure and data.
- Syntax and Example:

```
DROP TABLE Employees;
```

- Note: This operation is irreversible. All data and structure are permanently deleted.

5. TRUNCATE TABLE

- Purpose: Removes all rows from a table without deleting its structure.
- Syntax and Example:

```
TRUNCATE TABLE Employees;
```

- Note: This command is faster than DELETE for large tables and usually cannot be rolled back.

Integrity Constraints

Integrity constraints are rules that ensure the accuracy and consistency of data within a database table.

- Types of Constraints:

• Domain Constraint: Defines permissible values (e.g., data types, CHECK).

• Entity Constraint: Ensures unique identity (e.g., PRIMARY KEY).

• Referential Constraint: Enforces relationships (e.g., FOREIGN KEY).