TOPS TECHNOLOGY

# Module 4 – Introduction to DBMS

Presented By :

Nandni Vala

# SQL Constraints

## 1.What are constraints in SQL? List and explain the different types of constraints.

- Constraints in SQL are rules enforced on data in a table to maintain its accuracy, integrity, and reliability.
- **Types of Constraints**
- **1. NOT NULL**
- **Purpose**: Ensures that a column cannot have a NULL value.
- **Use Case**: To prevent missing or undefined data in a column.
- **Example**:
- CREATE TABLE employees (
-     id INT NOT NULL,
-     name VARCHAR(50) NOT NULL
- );

- **2.UNIQUE**
- **Purpose**: Ensures that all values in a column or combination of columns are unique.
- **Use Case**: To prevent duplicate values in a column.
- **Example**:
- CREATE TABLE users (
-     email VARCHAR(100) UNIQUE
- );
- **3. PRIMARY KEY**
- **Purpose**: Uniquely identifies each record in a table.
- **Use Case**: Combines NOT NULL and UNIQUE constraints. Only one primary key is allowed per table.

- **Example:**
- CREATE TABLE students (
-     student_id INT PRIMARY KEY,
-     name VARCHAR(50)
- );
- **4. FOREIGN KEY**
- **Purpose**: Links two tables by establishing a relationship between a column in one table and a primary key in another.
- **Use Case**: Maintains referential integrity.
- **Example**:
- CREATE TABLE orders (
-     order_id INT PRIMARY KEY,
-     customer_id INT,
-     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
- );

## 2. How do PRIMARY KEY and FOREIGN KEY constraints differ?

- **PRIMARY KEY**
- **Purpose**: Uniquely identifies each record (row) in a table.
- **Key Features**:
  - **Uniqueness**: The values in the primary key column(s) must be unique across the table.
  - **Non-NULL**: A primary key column cannot contain NULL values.
  - **Single Per Table**: Each table can have only one primary key.
  - **Composite Keys**: A primary key can consist of multiple columns (a composite primary key).
- **Use Case**: Ensures that every row in the table is distinct and can be uniquely identified.

- **Example**:
- CREATE TABLE employees (
-     employee_id INT PRIMARY KEY,
-     name VARCHAR(50)
- );
- **FOREIGN KEY**
- **Purpose**: Establishes a relationship between two tables by linking a column in one table to the primary key in another table.
- **Key Features**:
  - **References Primary Key**: The foreign key column refers to the primary key in another (or the same) table.
  - **Allows NULLs**: A foreign key column can contain NULL values unless restricted by the NOT NULL constraint.
  - **Enforces Referential Integrity**: Ensures that the value in the foreign key column matches a value in the referenced primary key column or is NULL.

  - **Multiple Per Table**: A table can have multiple foreign keys.

- **Use Case**: Creates a parent-child relationship between tables, ensuring data consistency across related tables.
- **Example**:
- CREATE TABLE orders (
-     order_id INT PRIMARY KEY,
-     customer_id INT,
-     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
- );

# 3. What is the role of NOT NULL and UNIQUE constraints?

- **NOT NULL Constraint**
- **Purpose**: Ensures that a column cannot have a NULL value.
- **Key Features**:
  - Prevents missing or undefined data in a column.
  - Applied at the column level.
  - Ensures that every row has a valid (non-NULL) value in the specified column.
- **Use Case**:
  - Ensures critical information is always provided. For example, an email column in a user table should never be empty.
- **Example**:
- CREATE TABLE employees (
-     employee_id INT NOT NULL,
-     name VARCHAR(50) NOT NULL
- );

## UNIQUE Constraint

- **Purpose**: Ensures that all values in a column (or a combination of columns) are distinct.
- **Key Features**:
  - Prevents duplicate values in a column or set of columns.
  - Allows a single NULL value (depending on the database system).
  - Can be applied to one or more columns (known as a composite unique constraint).
- **Use Case**:
  - Ensures that certain data, such as usernames, email addresses, or order IDs, is unique across the table.
- **Example**:
  - CREATE TABLE users (
  -     user_id INT PRIMARY KEY,
  -     email VARCHAR(100) UNIQUE
  - );