Constraints in SQL are rules applied to columns in a table to enforce data integrity and ensure the accuracy and reliability of the data stored in the database. Constraints restrict the type of data that can be entered into a table and provide a way to enforce business rules at the database level. Here are the most common types of SQL constraints:

**1. NOT NULL**

* Ensures that a column cannot have a NULL value.
* Example:
* CREATE TABLE Employees (
* EmployeeID INT NOT NULL,
* Name VARCHAR(50) NOT NULL
* );

**2. UNIQUE**

* Ensures that all the values in a column are distinct.
* Example:
* CREATE TABLE Users (
* UserID INT UNIQUE,
* Email VARCHAR(100) UNIQUE
* );

**3. PRIMARY KEY**

* A combination of NOT NULL and UNIQUE. It uniquely identifies each record in a table.
* A table can have only one primary key, and it can consist of one or multiple columns (composite key).
* Example:
* CREATE TABLE Orders (
* OrderID INT PRIMARY KEY,
* ProductName VARCHAR(100)
* );

**4. FOREIGN KEY**

* Establishes a relationship between two tables.
* A foreign key in one table refers to the primary key in another table.
* Example:
* CREATE TABLE Customers (
* CustomerID INT PRIMARY KEY,
* Name VARCHAR(50)
* );
* CREATE TABLE Orders (
* OrderID INT PRIMARY KEY,
* CustomerID INT,
* FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
* );

**5. CHECK**

* Ensures that all values in a column satisfy a specific condition.
* Example:
* CREATE TABLE Products (
* ProductID INT PRIMARY KEY,
* Price DECIMAL(10, 2) CHECK (Price > 0)
* );

**6. DEFAULT**

* Provides a default value for a column when no value is specified.
* Example:
* CREATE TABLE Employees (
* EmployeeID INT PRIMARY KEY,
* Salary DECIMAL(10, 2) DEFAULT 50000
* );

**7. INDEX**

* Not a traditional constraint but ensures quick retrieval of data by creating an index on a column or columns.
* Example:
* CREATE INDEX idx\_name ON Employees(Name);

**Combining Constraints**

* Constraints can be combined to apply multiple rules to a column.
* Example:
* CREATE TABLE Accounts (
* AccountID INT PRIMARY KEY,
* Username VARCHAR(50) UNIQUE NOT NULL,
* Balance DECIMAL(10, 2) DEFAULT 0 CHECK (Balance >= 0)
* );

By using constraints, you can improve data integrity, reduce the need for manual data validation, and maintain consistency in your database.