

# 5.SQL CONSTRAINTS &KEYS

Wednesday, February 4, 2026 12:38 PM

## Definition

Constraints are rules applied to table columns to prevent invalid or unwanted data from being inserted. They ensure that the data stored in a table remains accurate, consistent, and reliable.

## Why Constraints Are Used

- Prevent insertion of incorrect or unwanted data
- Maintain data integrity across the database
- Can be applied on one or multiple columns
- Ensure the stored data remains accurate and meaningful

## Types of Data Integrity Enforced by Constraints

### 1. Entity Integrity

- Ensures each row in a table is uniquely identifiable
- Enforced using:
  - PRIMARY KEY
  - UNIQUE KEY
- Prevents duplicate or NULL values in key columns

### 2. Referential Integrity

- Ensures relationships between tables remain valid
- Enforced using:
  - FOREIGN KEY
- Prevents orphan records and maintains consistency between parent and child tables

### 3. Domain Integrity

- Ensures data values follow defined rules for range, format, and type
- Enforced using:
  - CHECK constraints
- Controls what values are allowed in a column

## Main Types of Constraints

1. Unique Key constraint
2. Not Null constraint
3. Check constraint
4. Primary Key constraint
5. Foreign Key constraint

### 1. Unique Key Constraint

#### Meaning

- Ensures that no duplicate values are allowed in the column.
- Both UNIQUE and PRIMARY KEY enforce uniqueness, but UNIQUE allows NULL values while PRIMARY KEY does not.
- A table can have only one PRIMARY KEY but can have multiple UNIQUE keys.

#### Example (explained in words)

- A table EMP is created with:
  - EID as a unique integer
  - ENAME as a unique text value
  - SALARY as a money value
- This ensures EID and ENAME cannot have duplicate values.

#### CREATE TABLE EMP(

EID INT UNIQUE,

ENAME VARCHAR(50) UNIQUE,

**SALARY INT**

**);**

```
INSERT INTO EMP VALUES(101,'SAI',10000);
INSERT INTO EMP VALUES(NULL,'SAI B',100327)
INSERT INTO EMP VALUES(102,'SAI VARDHAN',100327)
```

## **2. NOT NULL Constraint**

### **Meaning**

- Ensures that a column cannot store NULL values.
- Forces the user to enter a valid value for that column.
- Commonly used for mandatory fields like names, dates, or IDs

```
CREATE TABLE EMP(
    EID INT NOT NULL,
    ENAME VARCHAR(50) NOT NULL,
    SALARY INT
);
```

```
INSERT INTO EMP VALUES(101,'SAI',10000);
INSERT INTO EMP VALUES(NULL,'SAI B',100327)
INSERT INTO EMP VALUES(102,'SAI VARDHAN',100327)
INSERT INTO EMP VALUES(102,NULL,100327)
```

## **3. CHECK Constraint**

### **Meaning**

- Ensures that values in a column follow a specific rule or condition.
- Used to enforce domain integrity (valid range, format, or condition).
- Examples include: salary must be greater than zero, age must be above 18, etc.

```
CREATE TABLE EMP(
    EID INT NOT NULL,
    ENAME VARCHAR(50) NOT NULL,
    SALARY INT CHECK(SALARY >10000)
);
```

```
INSERT INTO EMP VALUES(101,'SAI',10000);
INSERT INTO EMP VALUES(NULL,'SAI B',100327)
INSERT INTO EMP VALUES(102,'SAI VARDHAN',100327)
INSERT INTO EMP VALUES(102,NULL,100327)
```

## **4. Primary Key Constraint**

### **Meaning**

- Uniquely identifies each row in a table.
- Cannot contain NULL values.
- Cannot contain duplicate values.
- Only one PRIMARY KEY is allowed per table.
- Can be a single column or a combination of columns (composite key).

```
CREATE TABLE EMP(
    EID INT PRIMARY KEY,
    ENAME VARCHAR(50),
    SALARY INT
);
```

```
INSERT INTO EMP VALUES(101,'SAI',10000);
INSERT INTO EMP VALUES(NULL,'SAI B',100327)
INSERT INTO EMP VALUES(102,'SAI VARDHAN',100327)
```

## **INSERT INTO EMP VALUES(102,NULL,100327)**

A Foreign Key is used to create a relationship between two tables in a database. It ensures that the value in one table must match an existing value in another table, which helps maintain data accuracy and consistency.

### **Purpose**

- Connects two tables logically
- Ensures that related data exists in the parent table
- Prevents invalid or orphan records
- Helps retrieve related data efficiently across multiple tables

### **How It Works**

- One table acts as the **parent table** (contains the primary key).
- The other table acts as the **child table** (contains the foreign key).
- The foreign key column in the child table must match a value in the parent table's primary key column.

### **Why It Is Important**

- Maintains referential integrity
- Ensures data in related tables stays consistent
- Helps enforce rules about how data is linked
- Supports meaningful joins between tables

#### **Requirements for Creating a Foreign Key**

- Two tables must exist
- Both tables must share a common column
- The common column in the parent table must be a primary key or unique key
- The child table references this column through a foreign key constraint

### **CREATE TABLE DEPARTMENT**

```
(  
DEPTNO INT PRIMARY KEY,  
DNAME VARCHAR(50),  
LOCATION VARCHAR(MAX)  
);  
INSERT INTO DEPARTMENT VALUES (10, 'SALES', 'CHENNAI');  
INSERT INTO DEPARTMENT VALUES (20, 'PRODUCTION', 'MUMBAI');  
INSERT INTO DEPARTMENT VALUES (30, 'FINANCE', 'DELHI');  
INSERT INTO DEPARTMENT VALUES (40, 'RESEARCH', 'HYDERABAD');
```

```
SELECT * FROM DEPARTMENT;
```

### **CREATE TABLE EMPLOYEE**

```
(  
EID INT,  
NAME VARCHAR(50),  
SALARY DECIMAL,  
DEPTNO INT FOREIGN KEY REFERENCES DEPARTMENT(DEPTNO)  
);  
INSERT INTO EMPLOYEE VALUES (101, 'SAI', 35000, 10);  
INSERT INTO EMPLOYEE VALUES (102, 'PAVAN', 45000, 20);  
INSERT INTO EMPLOYEE VALUES (103, 'KAMAL', 74000, 30);  
INSERT INTO EMPLOYEE VALUES (104, 'RAVI', 58000, 40);  
INSERT INTO EMPLOYEE VALUES (104, 'RAVI', 58000, 50);
```