**MENTORNESS ARTICLE**

**TASK 1**

**Batch Name:-MIP-DA-04**

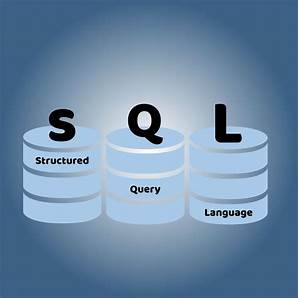
**CONSTRAINTS IN SQL**

**By Vaishnavi Patole**



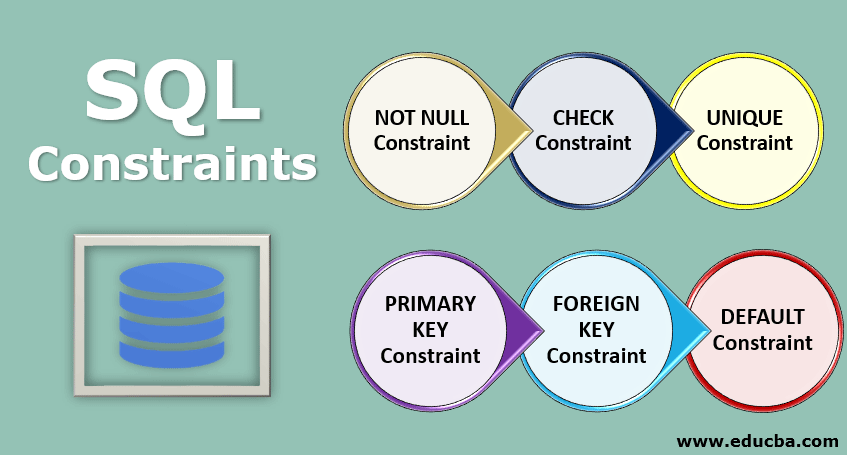
**INTRODUCTION**

SQL constraints have two functions: they guarantee data integrity and consistency. By imposing guidelines and limitations on the data kept in tables, they guard against anomalies and maintain the logical connections between different entities. Every restriction helps to preserve the integrity and quality of the database, whether it's creating foreign keys to provide referential integrity, defining primary keys to uniquely identify entries, or adding check constraints to restrict the range of permissible values.

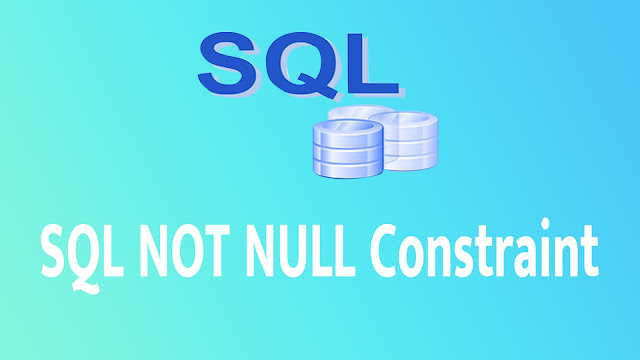


**CONSTRAINTS :-**

These are the rules given for column validation. A constraint in SQL is a rule or condition that is applied to a column or group of columns in a table to guarantee data consistency and integrity. Incorrect, inaccurate, or inconsistent data cannot be added to or altered in a database because constraints make sure that the data is kept according to specified guidelines .There are six types of constraints in SQL Which are given below:



1. **NOT NULL Constraint:-**



In SQL, a column cannot contain NULL values thanks to the NOT NULL constraint. Every row in the table must have a valid (non-NULL) value for each column to which this constraint is applied. In order to preserve data integrity and avoid unexpected behavior during database operations, this requirement is necessary. It is frequently utilized for columns that need to always have a value, such main keys or necessary fields in tables.

**Syntax:**

For creating a table:

CREATE TABLE table\_name (

column1 datatype NOT NULL,

column2 datatype NOT NULL,

...

);

**Example :**

For creating a table of Employees:

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

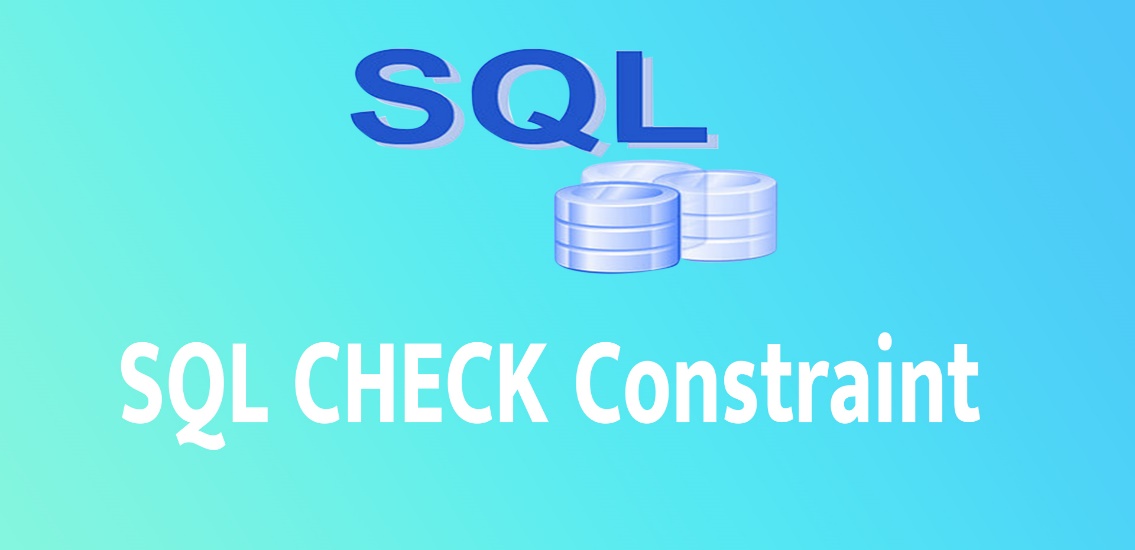
FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Department VARCHAR(50)

);

1. **CHECK Constraint:-**



It is used for extra validation. If a condition satisfy it will accept the value else reject the value. An SQL rule that restricts the values that may be recorded in a column is known as a check constraint. For any data to be entered or changed in the column, a requirement must be met. Invalid or unacceptable values cannot be saved in a database thanks to check constraints, which are used to protect data integrity. Commonly, they are employed to enforce data format standards, limit the range of permissible values, or guarantee that data complies with certain business rules.

**Syntax:**

For creating a table:

CREATE TABLE table\_name (

column\_name data\_type CONSTRAINT constraint\_name CHECK (condition),

...

);

**Example :**

Example of creating a table with a check constraint.

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Age INT CHECK (Age >= 18)

);

1. **UNIQUE Constraint:-**



It is used to avoid duplicate or repeated value into the column. A unique constraint in SQL ensures that all values in a column or a combination of columns are unique within a table. This constraint prevents duplicate values from being entered into the specified column(s), ensuring data integrity and consistency. Unique constraints can be applied to single columns or combinations of columns, allowing each row in the table to have a distinct value in the specified column(s).

**Syntax:**

For creating a table:

CREATE TABLE table\_name (

column\_name data\_type UNIQUE,

...

);

**Example :**   
Example of creating a table with a Unique constraint.

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

Email VARCHAR(50) UNIQUE,

...

);

1. **PRIMARY KEY Constraint:-**



It is constraint which is used to identify uniquely records from the table. To uniquely identify each record in a table, SQL uses a primary key requirement. It guarantees that the provided column or columns' values are distinct and cannot include null values. Furthermore, the performance of data retrieval operations is enhanced by a primary key constraint, which automatically generates a clustered index on the column or columns. In order to preserve data integrity and create relationships between tables in a relational database, this constraint is essential.

**Syntax:**

The syntax for defining a primary key constraint in SQL typically follows the column definition within a CREATE TABLE statement. Here's the basic syntax:

CREATE TABLE table\_name (

column\_name data\_type PRIMARY KEY,

...

);

**Example :**   
Example of creating a table with a Primary Key Constraint.

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

...

);

1. **FOREIGN KEY Constraint:-**



It is used to establish a connection between two or more tables. Referential integrity is maintained between two tables in a relational database via a foreign key requirement in SQL. The primary key of one table and the values in a column or columns of another are guaranteed to match when it comes to the values in a column or columns of the foreign key. In addition to eliminating orphaned or incorrect data, this constraint preserves data integrity by establishing a link between the tables.

**Syntax:**

In SQL, the definition of a foreign key constraint usually comes after the declaration of a column in a CREATE TABLE statement. This is the fundamental syntax:

CREATE TABLE table\_name (

column\_name data\_type,

...

FOREIGN KEY (column\_name) REFERENCES referenced\_table(referenced\_column)

);

**Example :**

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

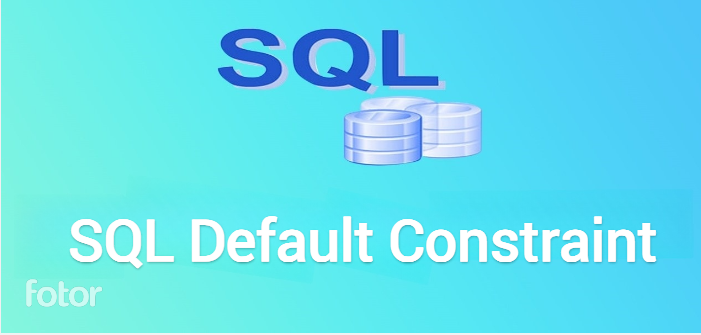
OrderDate DATE,

...

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

1. **DEFAULT Constraint:-**

****

To define a default value for a column in SQL, use a default constraint. The default value stated in the constraint is automatically entered into a column that has a default constraint when a new row is added to a table and the column has no value specified. This guarantees that, even in the event that a value is not supplied explicitly during insertion, the column will always have one. By offering a fallback value when needed, default constraints help to preserve data consistency and integrity.

**Syntax:**

For creating a table with a default constraint:

CREATE TABLE table\_name (

column\_name data\_type DEFAULT default\_value,

...

);

**Example :**

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Department VARCHAR(50) DEFAULT 'Sales'

);

**CONCLUSION**

SQL constraints are essential for guaranteeing data consistency and integrity inside a database, to sum up. They maintain referential integrity, avoid mistakes, and improve overall data quality by enforcing rules that control the validity and correctness of data. Database managers may protect against inconsistent data and guarantee the dependability of their database systems by carefully designing constraints such as main keys, foreign keys, unique constraints, and check constraints.