

## SQL Constraints

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

### Syntax:

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    column3 datatype constraint,  
    ....  
);
```

The following constraints are commonly used in SQL:

- a) **NOT NULL** - Ensures that a column cannot have a NULL value.  
The following SQL ensures that the "ID", "LastName", and "FirstName" columns will NOT accept NULL values:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);
```

- b) **UNIQUE** - Ensures that all values in a column are different.  
The following SQL creates a UNIQUE constraint on the "ID" column when the "Persons" table is created:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    UNIQUE (ID)  
);
```

- c) **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.  
The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created:

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
);
```

- d) **FOREIGN KEY** - Uniquely identifies a row/record in another table.

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

- e) **CHECK** - Ensures that all values in a column satisfies a specific condition.  
The following SQL creates a CHECK constraint on the "Age" column when the "Persons" table is created. The CHECK constraint ensures that you can not have any person below 18 years:

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CHECK (Age>=18)
);
```

- f) **DEFAULT** - Sets a default value for a column when no value is specified.  
The following SQL sets a DEFAULT value for the "City" column when the "Persons" table is created:

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    City varchar(255) DEFAULT 'Birtamode'
);
```

### **Assignment for Students:**

Write SQL Query to create following table (Student).

<b>Filed</b> s	<b>Datatype</b>	<b>Null</b>	<b>Key</b>	<b>Default</b>	<b>Check</b>	<b>Extra</b>
student_id	int(11)	No	Primary			Auto_Increment
name	varchar(50)	No				Unique
address	varchar(100)	No		Birtamode		
class_id	int(11)	No	Foreign			
section	varchar(50)	Yes				
Age	Int(11)	No		16	Age>=15	

**Note:** Foreign key references to (Class) Table.