SQL – Constraints

Definition:

The SQL CONSTRAINTS are an integrity which defines some conditions that restrict the column to remain true while inserting or updating or deleting data in the column.

The SQL CONSTRAINTS are used to implement the rules of the table.

The SQL provides following types of CONSTRAINTS :

Constraint	Description
NOT NULL	This constraint confirms that a column cannot store NULL value.
UNIQUE	This constraint ensures that each row for a column must have a different value.
PRIMARY KEY	This constraint is a combination of a NOT NULL constraint and a UNIQUE constraint. This constraint ensures that the specific column or combination of two or more columns for a table have a unique identity which helps to find a particular record in a table more easily and quickly.
CHECK	A check constraint ensures that the value stored in a column meets a specific condition.
DEFAULT	This constraint provides a default value when specified none for this column.
FOREIGN KEY	A foreign key constraint is used to ensure the referential integrity of the data. in one table to match values in another table.

1. NOT NULL EX:

```
CREATE TABLE CUSTOMERS(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2)
);
```

If CUSTOMERS table has already been created, then to add a NOT NULL constraint:

```
ALTER TABLE CUSTOMERS
   MODIFY SALARY  DECIMAL (18, 2) NOT NULL;
```

2. UNIQUE Constraint:

```
CREATE TABLE CUSTOMERS(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              UNIQUE,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2)

);
```

3. CHECK Constraint:

```
CREATE TABLE CUSTOMERS(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL CHECK (AGE >= 18),
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2)

);
```

If the CUSTOMERS table has already been created, then to add a CHECK constraint :

```
ALTER TABLE CUSTOMERS
   MODIFY AGE INT NOT NULL CHECK (AGE >= 18 );
```

4. DEFAULT Constraint:

```
CREATE TABLE CUSTOMERS(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2) DEFAULT 5000.00,
   PRIMARY KEY (ID)
);

INSERT INTO CUSTOMERS(ID,NAME,AGE,ADDRESS)VALUES (1001,
'KUMAR',25,'HYD');

1 row inserted.

Select * from customers;
ID     NAME   AG     ADD    SALARY
1001   KUMAR  25     HYD    5000.00
```

5. PRIMARY KEY:

```
CREATE TABLE CUSTOMERS(
   ID   INT              PRIMARY KEY,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2)

);
```

OR

```
CREATE TABLE CUSTOMERS(
   ID   INT,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

To create a PRIMARY KEY constraint on the "ID" column when the CUSTOMERS table already exists

```
ALTER TABLE CUSTOMER ADD PRIMARY KEY (ID);
```

6. FOREIGN KEY Constraint:

CUSTOMERS table

```
CREATE TABLE CUSTOMERS(
   ID   INT,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

ORDERS table

```
CREATE TABLE ORDERS (
   ID          INT         NOT NULL,
   DATE        DATETIME,
   CUSTOMER_ID INT references CUSTOMERS(ID),
   AMOUNT      double,
   PRIMARY KEY (ID)
);
```

If the ORDERS table has already been created and the foreign key has not yet been set:

```
ALTER TABLE ORDERS
   ADD FOREIGN KEY (Customer_ID) REFERENCES CUSTOMERS (ID);
```

DROP a FOREIGN KEY Constraint

To drop a FOREIGN KEY constraint, use the following SQL syntax.

```
ALTER TABLE ORDERS
   DROP FOREIGN KEY;
```