
DATA TYPES, CONSTRAINTS IN SQL

SQL Series Part 3

-Mayuri Dandekar

WHAT IS DATA TYPE?

Data type of a column defines what **value the column can store** in table.

Data types are defined while creating tables in database.

Data types are mainly classified into **three categories**

- **String:** char, varchar, etc
- **Numeric:** int, float, bool, etc
- **Date and time:** date, datetime, etc

COMMONLY USED DATA TYPES

- **Int:** used for the integer value (1,2,3,...)
- **Float:** used to specify a decimal point number (1.2, 2.5, 5.0,...)
- **Bool:** used to specify Boolean values true and false
- **Char:** fixed length string that can contain numbers, letters, and special characters
- **Varchar:** variable length string that can contain numbers, letters, and special characters
- **Date:** date format YYYY-MM-DD
- **Datetime:** date & time combination, format is YYYY-MM-DD hh:mm:ss

WHAT IS CONSTRAINTS?

Constraints are used to **specify rules for data** in a table. This ensures the **accuracy and reliability** of the data in the table

Constraints can be specified when the table is created with the **CREATE TABLE statement**, or after the table is created with the **ALTER TABLE statement**.

If there is any violation between the constraint and the record 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,
 );

COMMONLY USED CONSTRAINTS

NOT NULL - Ensures that a column cannot have a NULL value

UNIQUE - Ensures that all values in a column are different

PRIMARY KEY - A combination of a NOT NULL and UNIQUE

FOREIGN KEY - Prevents actions that would destroy links between tables (used to link multiple tables together)

CHECK - Ensures that the values in a column satisfies a specific condition

DEFAULT - Sets a default value for a column if no value is specified

CREATE INDEX - Used to create and retrieve data from the database very quickly

NOT NULL CONSTRAINT

The NOT NULL constraint enforces a column **NOT to accept NULL values**.

This imposes a field always to contain a value, which means that the user cannot insert a new record in a table or update a record without adding a value to this field.

NOTE: By default, a column can hold NULL values

```
1 • create database demo;
2 • use demo;
3 • CREATE TABLE student (
4     id INT NOT NULL,
5     first_name VARCHAR(25) NOT NULL,
6     last_name VARCHAR(25) NOT NULL,
7     age INT
8 );
9 • ALTER TABLE student
10  MODIFY age int NOT NULL;
11
```

UNIQUE CONSTRAINT

The UNIQUE constraint in SQL ensures that all **values in a column are distinct**.

UNIQUE and PRIMARY KEY constraints both provides a **guarantee for uniqueness** for a column or group of columns.

A PRIMARY KEY constraint, by default, has a UNIQUE constraint.

However, the user can have **many UNIQUE constraints per table**, but **only one PRIMARY KEY** constraint per table.

```
12  -- unique constraints --
13 • CREATE TABLE person (
14     id int NOT NULL,
15     last_name varchar(255) NOT NULL,
16     first_name varchar(255),
17     age int,
18     UNIQUE (ID)
19 );
20  -- add unique to firstname when table already created --
21 • ALTER TABLE person
22     ADD UNIQUE (first_name);
23
```

PRIMARY KEY CONSTRAINT


The **PRIMARY KEY** constraint uniquely identifies each of the records in a table.

Only ONE primary key can have in a table.

And also, in the table, this primary key can consist of **single or multiple columns** (fields).

Primary keys should **contain UNIQUE values**, and **cannot contain NULL values**.

```
24      -- primary key constraints --
25 • ○ CREATE TABLE employee (
26      ID INT NOT NULL,
27      last_name VARCHAR(255) NOT NULL,
28      first_name VARCHAR(255),
29      age INT,
30      PRIMARY KEY (ID)
31  );
```



FOREIGN KEY CONSTRAINT

A FOREIGN KEY is used to **link two tables** together. It is also called a **referencing key**.

Foreign Key is a combination of columns (can be single column) whose **value matches a Primary Key** in the different tables.

The relationship between two tables matches the **Primary Key in one of the tables with a Foreign Key in the second table**.

If the table contains a primary key defined on any field, then the user should not have two records having the equal value of that field.

```
33      -- foreign key constraints --
34  ● ○ CREATE TABLE customer (
35      C_Id INT NOT NULL,
36      Name VARCHAR(20) NOT NULL,
37      Age INT NOT NULL,
38      Address VARCHAR(25),
39      Salary DECIMAL(18 , 2 ),
40      PRIMARY KEY (C_Id)
41  );
42  ● ○ CREATE TABLE Orders (
43      OrderID INT NOT NULL,
44      OrderNumber INT NOT NULL,
45      Customer_Id INT,
46      PRIMARY KEY (OrderID),
47      FOREIGN KEY (Customer_Id)
48          REFERENCES customer (C_Id)
49  );
```

CHECK CONSTRAINT

The CHECK CONSTRAINTS is used to **limit the range of value** that can be placed in a column if the user defines a CHECK constraint on a single column, it **allows only specific values** for the column.

If the user defines a CHECK constraint on a table, it can limit the values in particular columns based on values in another column in the row.

```
51      -- check constraints --
52 • ○ CREATE TABLE booking (
53     ID int NOT NULL,
54     LastName varchar(255) NOT NULL,
55     FirstName varchar(255),
56     Age int,
57     CHECK (Age>=18)
58 );
```

DEFAULT CONSTRAINT

The DEFAULT constraint in SQL is used to provide a **default value for a column** of the table.

The default value will be **added to every new record** if no other value is mentioned.

```
60      -- default constraints --
61 • ○ CREATE TABLE student_new (
62      ID int NOT NULL,
63      LastName varchar(255) NOT NULL,
64      FirstName varchar(255),
65      Age int,
66      City varchar(255) DEFAULT 'Mumbai'
67  );
```

DEFAULT CONSTRAINT

CREATE INDEX statement in SQL is used to **create indexes in tables**.

The indexes are used to **retrieve data from the database more quickly than others**.

The user can not see the indexes, and they are just used to **speed up queries /searches**.

Note: Updating the table with indexes takes a lot of time than updating a table without indexes. It is because the indexes also need an update. So, only create indexes on those columns that will be frequently searched against.

Syntax-

```
CREATE INDEX index_name  
ON table_name (column1, column2, ...);
```

```
68  
69      -- index constraints --  
70 •   CREATE INDEX idx_lastname  
71      on Person (LastName);  
72  
73
```



THANK YOU!!!

SQL Series Part 3

-Mayuri Dandekar