# **SQL Datatypes and Constraints.**

The data entered into the table can be validated in 2 steps.

1. By assigning Datatypes.
2. By assigning Constraints.

Datatypes are mandatory, whereas Constraints are Optional.

**Datatypes**

Just like other programming languages, facilities of defining data of various types are available in **SQL** also. **SQL** supports the following data types for the specification of various data-items or fields of a relation/table. In SQL, each column of the table is assigned a datatype which conveys the kind of value that will be stored in the column.

### **Types of SQL Data Types**

1. **Numeric data type**:  
   It includes datatypes like int, tinyint, bigint, float, real, etc.
2. **Date and Time data type**:  
   It includes datatypes like Date, Time, Datetime, etc.
3. **Character and String datatype**:  
   It includes data types like char, varchar, text, etc.
4. **Miscellaneous datatype**:  
   It includes datatypes like clob, blob, xml,  etc.

|  |  |
| --- | --- |
| Datatype | Range |
| Int | **-2,147,483,648 to 2,147,483,647.** |
| SMALLINT | **-32768 to 32767**. |
| bigint | -9223372036854775808- 9223372036854775807 |

**NUMBER**

It is used to store numeric values .

SYNTAX: NUMBER ( Precision , [ Scale ] )

[ ] - Not Mandatory .

Precision : it is used to determine the number of digits used To store integer value .

Scale : : it is used to determine the number of digits used to store Decimal ( floating ) value within the precision . Scale is not mandatory , and the default value of scale Is zero ( 0 ) .

Example : Number ( 3 ) +/- 999

Example : Number ( 5 , 0 ) +/- 99999

Example : Number ( 5 , 2 ) +/- 999.99

### **DATE**

This data type is used to store a date in '**yyyy/mm/dd**' format. It stores a year, month and date values. **DATE** values can be compared with each other only. The date decimal point values to be entered are to be enclosed in **{ }** or with **single quotation marks**.

### **TIME**

This data type is used to store time in**hh:mm:ss**format. It stores hour, minute, and second values.

**CHAR**

In character datatype we can store 'A-Z' , 'a-z' , '0-9' And Special Characters( $ , & , @ , ! … ) .

A maximum of 254 characters can be stored in a string.

size is the number of characters to store which is of fixed length.

If you store strings that are not as long as the '*size*' parameter value, the remaining spaces are left unused.

Syntax: CHAR(size)

Example:

if you specify CHAR (10), strings such as "ram" and "technology" are each stored as 10 characters. However, a student *admission\_no* is 6 digits long in a school, so CHAR(6) would be appropriate to store the *admission\_no* of all the students. This data type is suitable where the number of characters to store is fixed. The values for CHAR data type have to be enclosed in single or double quotation marks.

**VARCHAR**

This data type is used to store variable-length alphanumeric data.

Syntax: VARCHAR(Size)

Example:

The address of a student can be declared as **VARCHAR (25)** to store the address up to 25 characters long. The advantage of using this data type is that **VARCHAR** will not leave unused spaces. It releases the unused memory spaces.

**CONSTRAINTS:**

It is a rule given to a column for validation .

Types of Constraints :

1. UNIQUE
2. NOT NULL
3. CHECK
4. PRIMARY KEY
5. FOREIGN KEY .
6. DEFAULT
7. **UNIQUE :** "It is used to avoid duplicate values into the column "
8. **NOT NULL :** "It is used to avoid Null "
9. **CHECK :** "It is an extra validation with a condition If the condition is satisfied then the value is accepted else Rejected "
10. **PRIMARY KEY :** "It is a constraint which is used to identify a record Uniquely from the table " .

Characteristics of Primary key :

* + - * We can have only 1 PK in a table
      * PK cannot accept duplicate / repeated values .
      * PK cannot accept Null
      * PK is always a combination of Unique and Not Null Constraint

1. **FOREIGN KEY :** "It is used to establish a connection between the The tables”

Characteristics of Foreign key :

* We can have Multiple FK in a table
* FK can accept duplicate / repeated values .
* FK can accept Null
* FK is not a combination of Unique and Not Null Constraint.
* For an Attribute ( column ) to become a FK ,it is mandatory That it must be a PK in its own table .

**6**.**DEFAULT: “**It is used to give a default value to column”

NOTE: NULL

* Null Is a keyword which is used to represent Nothing / Empty Cell.
* Null doesn’t represent 0 or Space.
* Any operations performed on a Null will result in Null itself.