

Introduction to SQL

- **SQL Data Types-**

We store data in SQL in the form of tables. Tables are the simplest objects (structures) for data storage that exist in a database.

Below we have given an example of table - Defining a table using SQLCommands -

```
CREATE TABLE CUSTOMERS'  
(  
  ID INT NOT NULL,  
  Ninja_Name VARCHAR (20) NOT NULL,  
  Ninja_Age INT NOT NULL,  
  Ninja_Address CHAR (25) ,  
  Ninja_Goal CHAR(25),  
  PRIMARY KEY (ID)  
);
```

This command will result into creation of a table -

ID	Ninja_Name	Ninja_Age	Ninja_Address	Ninja_Goal

From a SQL command for each column attribute we have defined a specific data type. In case of ID it is INT i.e Integer and similarly in case of Ninja_Name it is VARCHAR i.e a variable length string.

While choosing a data type for a column attribute we have to think of

1. Value we want to represent - like if it is a name(Character data type will be most suitable, VARCHAR) or Age(Numerical data type will be more relevant, INT)
2. Space or Memory that value will occupy - like if we are taking customer feedback we need to store value in a variable character data type, VARCHAR.

Below we have given three main types of data types -

- **String Data Types -**

Datatype	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the maximum column length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the maximum column length in bytes.

- **Numeric Data Types:**

Datatype	Description
BIT(size)	A bit-value type. The number of bits per value is specified in <i>size</i> . The <i>size</i> parameter can hold a value from 1 to 64. The default value for <i>size</i> is 1.
TINYINT(size)	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The <i>size</i> parameter specifies the maximum display width (which is 255)
INT(size)/ INTEGER(size)	Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The <i>size</i> parameter specifies the maximum display width (which is 255)
FLOAT(p)	A floating point number. MySQL uses the <i>p</i> value to determine whether to use FLOAT or DOUBLE for the resulting data type. If <i>p</i> is from 0 to 24, the data type becomes FLOAT(). If <i>p</i> is from 25 to 53, the data type becomes DOUBLE()
DECIMAL(size, d)	The number of digits after the decimal point is specified in the <i>d</i> parameter. The maximum number for <i>size</i> is 65. The maximum number for <i>d</i> is 30. The default value for <i>size</i> is 10. The default value for <i>d</i> is 0.

- **Date and Time Data Types:**

Datatype	Description
DATE	Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'
DATETIME	This is a date and time combination in the format: YYYY-MM-DD hh:mm:ss. The supported range starts from '1000-01-01 00:00:00' and end at '9999-12-31 23:59:59'.
TIME	This is a time Format: hh:mm:ss. The supported range is starts from '-838:59:59' and ends at '838:59:59'
TIMESTAMP	TIMESTAMP values are stored as the number of seconds. Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC.

MySQL retrieves and displays DATETIME values in 'YYYY-MM-DD HH:MM:SS' format. But we can use the DATE_FORMAT() function to format the date time when we want to display it.

Eg:

```
SELECT DATE_FORMAT (column_name, '%m/%d/%Y %H:%i')
```

```
FROM tablename
```

TINYINT: The Boolean data type is not specifically defined in MySQL, but it uses this boolean value from the already existing Numeric data type explained above. The values to be stored in this case will be 0 and 1.

Syntax:

```
CREATE TABLE <table-name>
```

```
(
```

```
<column-name> BOOLEAN
```

```
);
```

There are two Advanced Data types in MySQL. They are:

1. Spatial types - Using this type, we can store single or multi-point data (geometrical or geographical).
2. JSON types - JSON stands for JavaScript Object Notation. JSON has some standard format that we need to follow in order to easily store information. This structure is in the form of a KEY-VALUE pair.