

# SQL

---

## Introduction to Sql

### What is SQL ?

SQL (Structured Query Language) is a computer language aimed to store, manipulate, and retrieve data stored in relational databases. It is a language to communicate with the database.

SQL is the standard language for Relational Database Systems.

Relational database means the data is stored as well as retrieved in the form of relations (tables).

### Why is SQL needed?

- SQL is an exceptional programming language that is utilized to interface with databases. It works by understanding and analyzing databases that include data fields in their tables.
- SQL is simple and easy to learn
- Allows users to access information on relational database management systems.
- Allows users to interpret data.
- Allows users to define data in a database and use that data. Allows embedding in other languages using SQL modules, libraries and pre-compilers.
- Allows users to create and drop data and tables.
- Allows users to set permissions for tables and views.

### SQL Commands

- DDL - Data Definition Language
  - DML - Data Manipulation Language
  - DCL - Data Control Language
  - TCL - Transaction Control Language
-

### **DDL (Data Definition language)**

The data definition language (DDL) is used to specify the relation schemas as well as other information about the relations.

#### **Commands:**

Create : Create a new table, a view of the table or other objects in the database.

Alter : Modifies existing database objects such as tables.

Drop : Deletes an entire table, a view of table or other objects in the database.

Rename : Rename object.

### **DML (Data Manipulation Language)**

The DML statements used for managing data within schema objects which deals with data manipulation.

#### **Commands:**

Select : retrieve data from a database

Insert : Creates a record

Update : Modifies record, updates existing data within a table

Delete : Deletes a record

### **DCL (Data Control Language)**

The DCL Commands is used to control user access in a database. Using DCL command, it allows or restricts the user from accessing data in database schema.

#### **Commands:**

Grant : Gives Privilege to users.

Revoke : Take back privileges granted from users.

### **TCL (Transaction Control Language)**

The TCL commands for specifying the beginning and ending of transactions. These are used to manage the changes made by DML statements.

#### **Commands:**

Commit : Permanently save any transaction into the database.

Rollback : Restores the database to the last committed state.

---

## Creating Tables

### CREATE TABLE Statement

The CREATE TABLE statement is used to create a new table in a database.

To create a new table, you must have the CREATE TABLE privilege and a storage area for it.

The database administrator uses data control language (DCL) statements to grant this privilege to users and assign a storage area.

#### Syntax:

```
CREATE TABLE table_name  
(  
  column_name1 data_type(size),  
  column_name2 data_type(size),  
  column_name3 data_type(size),  
  ....  
);
```

The column\_name parameters specify the names of the columns of the table.

The data\_type parameter specifies what type of data the column can hold (e.g. varchar, integer, decimal, date, etc.).

The size parameter specifies the maximum length of the column of the table.

#### Example:

```
CREATE TABLE Person (  
    PID int,  
    FName varchar(255),  
);
```

#### Person

|     |       |
|-----|-------|
| PID | FName |
|-----|-------|

## INSERT INTO Statement

The **INSERT INTO** statement is used to insert new records in a table.

### SQL INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two forms.

The first form does not specify the column names where the data will be inserted, only their values:

**Syntax:**

```
INSERT INTO table_name  
VALUES (value1,value2,value3,...);
```

The second form specifies both the column names and the values to be inserted:

**Syntax:**

```
INSERT INTO table_name (column1,column2,column3,...)  
VALUES (value1,value2,value3,...);
```

**Example:**

```
INSERT INTO Customers (PID, FName, Country)  
VALUES '1', 'Robot', 'India');
```

| PID | FName | Country |
|-----|-------|---------|
| 1   | Robot | India   |

## ALTER TABLE STATEMENT

The SQL ALTER TABLE statement is used to add, modify, or drop/delete columns in a table. The SQL ALTER TABLE statement is also used to rename a table.

### 1. ADD TABLE

To add a column in a table, the ALTER TABLE syntax in SQL is:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

---

EXAMPLE:

```
ALTER TABLE Person  
ADD LName varchar(255);
```

## 2.DROP TABLE

To drop a column in an existing table, the SQL ALTER TABLE syntax is:

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

EXAMPLE:

```
ALTER TABLE Person  
DROP COLUMN Country;
```

# SQL Data Types

## Character Data Types

### Char

Char data types contains non-binary strings. length of character is fixed while creating a table.string is right-padded with spaces to the specified length when stored.A fixed-length string between 0 and 255 characters in length ,by default is 1.

**Syntax: char(size)**

Where size is length of string;

### Varchar

Varchar contains contains non-binary strings. Columns are variable-length strings. A variable size between 1 and 255 characters in length. You must define a length when creating a VARCHAR field.

**Syntax: VARCHAR(SIZE)**

Where size is length of string;

### Text

---

Text values are considered as character strings having a character set. Variable length storage with maximum size of 2GB data

**Syntax:** *TEXT(size)*

## Numeric Data Types

### **SMALLINT(size)**

A small integer that can be signed or unsigned. Small signed range is from -32768 to 32767. Unsigned range is from 0 to 65535.

### **INTEGER(size)**

An integer can be signed or unsigned. Signed Integer range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The size parameter specifies the maximum display width (which is 255).

### **BIGINT(size)**

A BIG (large) integer that can be signed or unsigned. If signed, the allowable range is from -9223372036854775808 to 9223372036854775807. If unsigned bigint, the allowable range is from 0 to 18446744073709551615.

### **FLOAT(L,D)**

A FLOAT number that cannot be unsigned. we have to define the display length (L) and the number of decimals (D). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a FLOAT.

### **FLOAT(v)**

A floating point number. It uses the v value to determine whether to use FLOAT or DOUBLE for the resulting data type. If p is from 0 to 24, the data type becomes FLOAT(). If v is from 25 to 53, the data type becomes DOUBLE().

### **DECIMAL(L,D)**

---

An unpacked floating-point number that cannot be unsigned. In the unpacked decimals, each decimal corresponds to one byte. Defining the display length (L) and the number of decimals (D) is required. NUMERIC is a synonym for DECIMAL.

### **DOUBLE(L,D)**

A double precision floating-point number cannot be unsigned. DOUBLE(L,D); where L is length and D is the number of decimals. By default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.

## **Date and Time Data Types:**

### **DATE**

A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31.  
For example, NOVEMBER 20th, 1999 would be stored as 1999-11-20.

### **TIME**

It stores the time in a HH:MM:SS format.

### **DATETIME**

It is a combination of date and time in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59.  
For example, 1:15 in the afternoon on November 20th, 1999 would be stored as 1999-11-20 13:15:00

### **TIMESTAMP**

A timestamp between midnight, January 1st, 1970 and sometime in 2037. This looks like the previous DATETIME format, only without the hyphens between numbers; 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 (YYYYMMDDHHMMSS).

### **YEAR**

It stores the year in 2 digit or 4 digit format. Range 1901 to 2155 in 4-digit format. Range 70 to 69, representing 1970 to 2069.

---

## **BINARY**

Fixed length with maximum length of 8,000 bytes

## **BIT**

A BIT data type is used to store bit values from 1 to 64. So, a BIT field can be used for booleans, providing 1 for TRUE and 0 for FALSE.