

## Creating tables

This reading covers the process of creating a table in a database. You've learned about creating and dropping a database using **CREATE DATABASE** and **DROP DATABASE** SQL commands. You also learned the SQL syntax of creating a table in the database using the **CREATE TABLE** syntax and modifying the structure of an already created table using **ALTER TABLE**.

The main objective of this reading is to help you develop a more practical understanding of how the **CREATE TABLE** statement works. You'll also learn how to use its syntax correctly. Furthermore, this reading also highlights some important points to note when creating a database table.

### Important points on creating a table

You know already that in SQL, the **CREATE TABLE** syntax is available in the Data Definition Language (DDL) subset which is used to create a new database table.

Here are some important points to remember when using the **CREATE TABLE** syntax. These points are also useful for creating tables in general.

Always give a meaningful name to your table as well as the columns or fields of the table.

The data types of columns or fields may vary from one database system to another. For example, **NUMBER** is supported in Oracle database for integer values whereas **INT** is supported in MySQL.

It's good practice to refer to the list of data types supported by the database system that you are using, namely MySQL, SQL Server and Oracle.

Remember to specify the appropriate length for data types where applicable.

- For example, **VARCHAR** is the preferred data type to use if you are storing text-based data because it saves space. **VARCHAR** takes up 1 byte per character, plus another 2 bytes to hold length. So, for example, if you set a **VARCHAR(100)** data type to a first name column and store a value of **Jen** in it, then it would take up 3 bytes (for J, E, and N) plus another 2 bytes, that is 5 bytes in total to store that value.

### Using the CREATE TABLE statement

Let's inspect the **CREATE TABLE** syntax more closely. In this example, let's look at the syntax to create a table named **customers** in the sample database. The table includes the following columns/fields:

**CustomerId**

**FirstName**

**LastName**

**Company**

**Address**

**City**

**State**

**Country**

**PostalCode**

**Phone**

**Fax**

**Email**

**SupportRapid**

The table is created using the following syntax:

```
1 CREATE TABLE customers
2   (CustomerId INT,
3     FirstName VARCHAR(40),
4     LastName VARCHAR(20),
5     Company VARCHAR(80),
6     Address VARCHAR(70),
7     City VARCHAR(40),
8     State VARCHAR(40),
9     Country VARCHAR(40),
```

```
10 | PostalCode VARCHAR(10),
11 | Phone VARCHAR(24),
12 | Fax VARCHAR(24),
13 | Email VARCHAR(60),
14 | SupportRapid INT );
```

The main syntax here is the **CREATE TABLE** syntax.

You then provide the name of the table, in this case **customers**.

Then a pair of parentheses should follow. Within the parenthesis the column or field names are given followed by the data type of each and the size or length of data that can be stored.

**CustomerId** and **SupportRapidfiles** are of type **INT** because they store numeric values. The rest of the fields have a data type of **VARCHAR** but have different lengths.

The lengths are specified according to the amount of data (the number of characters) that need to be stored in these fields.

At the end of the statement, remember to put the semicolon.

Mark as completed

---



Like



Dislike



Report an  
issue