

SQL Create Database

In SQL, the '**Create Database**' statement is a first step for storing the structured data in the database.

The database developers and the users use this statement in SQL for creating the new database in the database systems. It creates the database with the name which has been specified in the Create Database statement.

Syntax of Create Database statement in SQL

1. **CREATE DATABASE Database_Name;**

In this syntax, **Database_Name** specifies the name of the database which we want to create in the system. We have to type the database name in query just after the 'Create Database' keyword.

Following are the most important points which are required to learn while creating a database:

- **The database we want to create should be a simple and unique name, which can be easily identified.**
- **Database name should be no more than 128 characters.**

Syntax of Create Database statement in MySQL

The same command is used in MySQL to create the new database for storing the structured data.

1. **CREATE DATABASE Database_Name;**

Syntax of Create Database in Oracle

There is no need to create the database in Oracle systems. **In the Oracle database, we can directly create the database tables.**

Examples of Create Database statement in SQL

In this article, we took the following two examples which will help how to run and perform the Create Database query in SQL:

Example 1:

This example creates the **Student** database. To create the Student database, you have to type the following command in Structured Query Language:

1. **CREATE DATABASE Student ;**

When this query is executed successfully, then it will show the following output:

Database created successfully

You can also verify that your database is created in SQL or not by using the following query:

1. **SHOW DATABASE ;**

SQL does not allow developers to create the database with the existing database name. Suppose if you want to create another Student database in the same database system, then the Create Database statement will show the following error in the output:

```
Can't create database 'Student'; database exists
```

So, firstly you have to delete the existing database by using the Drop Statement. You can also replace the existing database with the help of Replace keyword.

If you want to replace the existing Student database, then you have to type the following SQL query:

1. **CREATE OR REPLACE DATABASE Student ;**

Example 2:

Suppose, we want to create the Employee database in the system.

Firstly, we have to type the following command in Structured Query Language:

1. **CREATE DATABASE Employee ;**

When this query is executed successfully, then it will show the following output:

Database created successfully

You can also check that your database is created in SQL by typing the following query:

1. **SHOW DATABASE ;**

We know that SQL does not allow developers to create the database with the existing database name.

Suppose, we want to create another Employee database in the same database system, firstly, we have to delete the existing database using a drop statement, or we have to replace the existing Employee database with the help of the 'replace' keyword.

To replace the existing Employee database with a new Employee database, we have to type the following query in SQL:

1. **CREATE** OR **REPLACE DATABASE** Employee;

SQL DROP Database

The SQL Drop Database statement **deletes the existing database permanently from the database system. This statement deletes all the views and tables if stored in the database**, so be careful while using this query in SQL.

Following are the most important points which are required to learn before removing the database from the database system:

- **This statement deletes all the data from the database. If you want to restore the deleted data in the future, you should keep the backup of data of that database which you want to delete.**
- **Another most important point is that you cannot delete that database from the system which is currently in use by another database user. If you do so, then the drop statement shows the following error on screen:**

1. Cannot **drop database** "name_of_the_database" because it **is** currently in use.

Syntax of Drop Database Statement in SQL

1. **DROP DATABASE** Database_Name;

In this SQL syntax, we have to specify the name of that database which we want to delete permanently from the database system. We have to write the name of the database after the DROP DATABASE keyword in every example.

We can also delete multiple databases easily by using the single DROP syntax:

1. **DROP DATABASE** Database_Name1, [Database_Name2,, Database_Name N] ;

Using this statement, we have no need to write multiple statements for deleting multiple databases. We can specify all the databases by using a comma in a single statement, as shown in the above syntax.

Examples of Drop Database Statement in SQL

In this article, we took the following two examples that will help how to run and perform the Drop Database query in SQL:

Example1:

Suppose, we want to delete the Student database with all its data from the database system so, firstly we have to check that the Student database exists in the system or not by using the following statement:

1. **SHOW DATABASES ;**

If the Student database is shown in the output, then we have to type the following query in SQL for removing the Student database:

1. **DROP DATABASE** Student;

If the Student database does not exist in the database system and we run the above query in SQL, then the query will show the following output:

```
Can't drop database 'Student'; database doesn't exist
```

Example2:

Suppose, we want to delete the College database with all its tables and views from the database system, firstly we have to check that if the College database exists in the system or not by using the following statement:

1. **SHOW DATABASES;**

If the College database is shown in the output, then you have to type the following query in SQL for removing the College database permanently:

1. **DROP DATABASE** College;

If the College database does not exist in the database system, and we run the above query in SQL, then this query will show the following output:

```
Can't drop database 'College'; database doesn't exist
```

SQL RENAME Database

In some situations, database users and administrators want to change the name of the database for some technical reasons. So, the **Rename Database** statement in SQL is used to change the name of the existing database.

Sometimes, the Rename Database statement is used because the developers think that the original name is not more relevant to the data of the database, or they want to give a temporary name to that database.

Syntax of Rename Database in SQL

1. **ALTER DATABASE** old_database_name **MODIFY NAME** = new_database_name;
1. **EXEC** sp_renamedb'old_database_name' , 'new_database_name'

Syntax of Rename Database in MySQL

1. **RENAME DATABASE** old_database_name **TO** new_database_name;

This syntax is used when we want to change the name of the database in MySQL.

Examples of Rename Database in SQL

In this article, we have taken the following two examples which will help you how to run and perform the Rename Database query in SQL:

Example 1:

Suppose we want to rename the Student Database. For this, we have to type the following query in SQL:

1. **ALTER DATABASE** Student **MODIFY NAME** = College ;

This query will change the name of the database from Student to College. To run this query, we must ensure that the database Student exists in the current database server. If not, then it will show an error in the output.

Example 2:

Suppose we want to rename the Department Database. For this, we have to type the following query in SQL:

1. **ALTER DATABASE** Department **MODIFY NAME** = Company ;

This query changes the name of the database from Department to Company. To run this query, we must ensure that the database Department exists in the current database server. If not, then it will show an error in the output.

SQL SELECT Database

Suppose database users and administrators want to perform some operations on tables, views, and indexes on the specific existing database in SQL. Firstly, they have to select the database on which they want to run the database queries.

Any database user and administrator can easily select the particular database from the current database server using the **USE** statement in SQL.

Syntax of USE statement in SQL

1. **USE** database_name;

In this syntax, we have to define the name of the database after the **USE** keyword and the name of the database must be unique.

Syntax of USE statement in MySQL

1. **USE** database_name;

Syntax of USE statement in Oracle

There is no need to select the database in Oracle.

Examples of USE statement in SQL

In this article, we have taken the following three examples which will help you how to run and perform USE statement in SQL:

Example 1: Suppose, you want to work with the **Hospital** database. For this firstly, you have to check that if the Hospital database exists on the current database server or not by using the following query:

1. SHOW DATABASES;

If the Hospital database is shown in the output, then you have to execute the following query to select the Hospital database:

1. USE Hospital;

Example 2: Suppose, you want to work with another College database in SQL. For this firstly, you have to check that the College database exists on the current database server or not by using the following query:

1. SHOW DATABASES;

If the College database is shown in the result, then you have to execute the following query to select the College database:

1. USE College;

Example 3: Suppose you want to work with another School database in SQL. For this firstly, you have to check that the School database exists on the current database server or not by using the following query:

1. SHOW DATABASES;

If the School database is shown in the result, then you have to execute the following query to select the School database:

1. USE School;