**How to Create Database in SQL Server**

A database in Sql Server is a storage location where we can store our business data. The SQL database uses tables to store our information in a normalizes way. So by creating database in Sql Server, We can easily Select, Update and Delete the business data.

The basic syntax for SQL create Database is:

CREATE DATABASE Database\_Name

Example

-- Code for SQL Create Database

CREATE DATABASE New\_Database;

**NOTE:** In an organization, we may or may not have the privileges to know the available databases in SQL Server. So, it is always advisable to check whether the database name already exists or not

-- SQL Create Database Example

IF NOT EXISTS

(

SELECT name FROM master.dbo.sysdatabases

WHERE name = N'New\_Database'

)

CREATE DATABASE [New\_Database]

OR

-- Code to Sql Server Create Database

IF EXISTS

(

SELECT name FROM master.dbo.sysdatabases

WHERE name = N'New\_Database'

)

BEGIN

SELECT 'Database Name already Exist' AS Message

END

ELSE

BEGIN

CREATE DATABASE [New\_Database]

SELECT 'New Database is Created'

END

## How to Delete Database in SQL Server

**The syntax** for SQL Delete database or Drop Database in SQL server is:

DROP DATABASE [Database Name]

Better approach

IF EXISTS

(

SELECT name FROM master.dbo.sysdatabases

WHERE name = N'New\_Database'

)

DROP DATABASE [New\_Database]

## How to Rename Database in SQL Server

**The syntax** to rename database in Sql Server is:

SP\_RENAMEDB [Old Database Name],[New Database Name]

# Get Database Names from SQL Server

-- Query to get SQL Server Database Names

USE master

GO

SELECT name FROM sys.databases

You can also use sysdatabases to get the list of databases in SQL Server.

-- Query to get SQL Server Database Names

USE master

GO

SELECT name FROM sysdatabases

Or, use sp\_databases [stored procedure](https://www.tutorialgateway.org/stored-procedures-in-sql/) to get a list of databases in SQL Server.

-- Query to get SQL Server Database Names

USE master

GO

EXEC sp\_databases

If you know the database id, use the following query to display the list of databases except for system databases

-- Query to get SQL Server Database Names

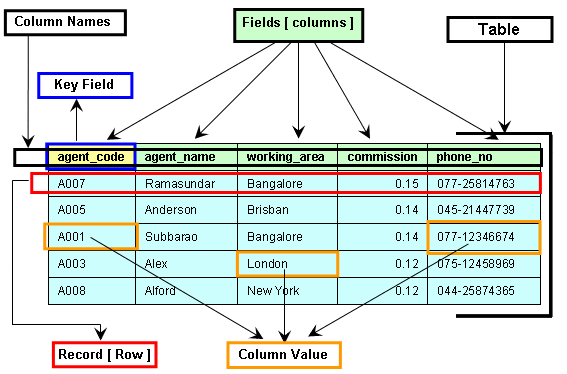
USE master

GO

SELECT database\_id, name FROM sys.databases

WHERE database\_id > 4

**RDBMS** is used to manage Relational database. **Relational database** is a collection of organized set of tables related to each other, and from which data can be accessed easily.



SQL Commands (language)

Structured Query Language(SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert etc. to carry out the required tasks.

These SQL commands are mainly categorized into four categories as:

1. DDL – Data Definition Language
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language
5. TCL – Transaction Control Language
6. **DDL(Data Definition Language) :**DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

**Examples of DDL commands:**

* + [**CREATE**](https://www.geeksforgeeks.org/sql-create/) – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
  + [**DROP**](https://www.geeksforgeeks.org/sql-drop-truncate/) – is used to delete objects from the database.
  + [**ALTER**](https://www.geeksforgeeks.org/sql-alter-add-drop-modify/)-is used to alter the structure of the database.
  + [**TRUNCATE**](https://www.geeksforgeeks.org/sql-drop-truncate/)–is used to remove all records from a table, including all spaces allocated for the records are removed.
  + [**COMMENT**](https://www.geeksforgeeks.org/sql-comments/) –is used to add comments to the data dictionary.
  + [**RENAME**](https://www.geeksforgeeks.org/sql-alter-rename/)–is used to rename an object existing in the database.

1. **DQL (Data Query Language) :**

DML statements are used for performing queries on the data within schema objects. The purpose of DQL Command is to get some schema relation based on the query passed to it.

**Example of DQL:**

* + [**SELECT**](https://www.geeksforgeeks.org/sql-select-clause/) – is used to retrieve data from the a database.

1. **DML(Data Manipulation Language) :**The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

**Examples of DML:**

* + [**INSERT**](https://www.geeksforgeeks.org/sql-insert-statement/) – is used to insert data into a table.
  + [**UPDATE**](https://www.geeksforgeeks.org/sql-update-statement/) – is used to update existing data within a table.
  + [**DELETE**](https://www.geeksforgeeks.org/sql-delete-statement/) – is used to delete records from a database table.

1. **DCL(Data Control Language) :**DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.

**Examples of DCL commands:**

* + **GRANT**-gives user’s access privileges to database.
  + **REVOKE**-withdraw user’s access privileges given by using the GRANT command.

1. **TCL(transaction Control Language) :**TCL commands deals with the [transaction within the database](https://www.geeksforgeeks.org/sql-transactions/).

**Examples of TCL commands:**

* + **COMMIT**– commits a Transaction.
  + [**ROLLBACK**](https://www.geeksforgeeks.org/sql-transactions/)– rollbacks a transaction in case of any error occurs.
  + **SAVEPOINT**–sets a savepoint within a transaction.
  + **SET TRANSACTION**–specify characteristics for the transaction.

The basic syntax for SQL Server create table is as shown below:

-- SQL Server Create Table Syntax

CREATE TABLE [Table Name]

(

Column\_Name1 Data\_Type(Size) [NULL | NOT NULL],

Column\_Name2 Data\_Type(Size) [NULL | NOT NULL],

…

Column\_NameN Data\_Type(Size) [NULL | NOT NULL]

);

* **Table Name:** Please provide Unique table name here. If you write the already existing table name, it will throw an error
* **Column\_Name:** Please specify the Unique Column Names required for this table
* **Data Type:** Please specify the valid [Data type](https://www.tutorialgateway.org/sql-data-types/) that the column will hold. For example, Int, Money, Varchar, Nvarchar and Date
* **Size:** Data types like Varchar, nvarchar, Char expects the size So, Please provide valid number here
* **NULL or NOT NULL:** If you select the NULL option then column will accept both normal values and NULL values. Otherwise, it will throw an error saying Column should not be empty

-- SQL Server Create Table Example

CREATE TABLE [Customer]

(

[CustomerKey] [int] NOT NULL,

[Name] [varchar](150) NULL,

[DateOfBirth] [date] NULL,

[EmailAddress] [nvarchar](50) NULL,

[Profession] [nvarchar](100) NULL

)

GO

**NOTE:** Before you start creating a table in Sql Server, It is always advisable to [check if a Table exists or not](https://www.tutorialgateway.org/how-to-check-if-a-table-exists-in-sql-server/).

Apporach 1:

-- Query:- Sql Server check table exists before creating

USE [SQLTEST]

GO

IF OBJECT\_ID('dbo.Employees', 'U') IS NOT NULL

BEGIN

PRINT 'Table Exists in SQL Test Database'

END

ELSE

BEGIN

PRINT 'Table Does not Exists'

END

Approach 2:

-- Query:- Sql Server check table exists before creating

USE [SQLTEST]

GO

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME = N'Employees')

BEGIN

PRINT 'Table Exists in SQL Test Database'

END

ELSE

BEGIN

PRINT 'Table Does not Exists'

END

Approach 3:

-- Query:- SQL check if table exists before creating

USE [SQLTEST]

GO

IF EXISTS(SELECT 1 FROM sys.Objects

WHERE Object\_id = OBJECT\_ID(N'dbo.Employees')

AND Type = N'U')

BEGIN

PRINT 'Table Exists in SQL Test Database'

END

ELSE

BEGIN

PRINT 'Table Does not Exists'

END

Approach 4:

-- Query:- Sql Server check table exists before creating

USE [SQLTEST]

GO

IF EXISTS(SELECT 1 FROM sys.Tables

WHERE Name = N'Employees')

BEGIN

PRINT 'Table Exists in SQL Test Database'

END

ELSE

BEGIN

PRINT 'Table Does not Exists'

END

### Create Table with Identity Column

-- SQL Server Create Table Example

CREATE TABLE [Customer11]

(

[CustomerKey] [int] IDENTITY(1,1) NOT NULL,

[FirstName] [varchar](50) NULL,

[LastName] [varchar](50) NULL,

[BirthDate] [date] NULL,

[EmailAddress] [nvarchar](50) NULL,

[Yearly Income] [money] NULL,

[Profession] [nvarchar](100) NULL

)

GO

Constraints

* [Primary key](https://www.sqlservertutorial.net/sql-server-basics/sql-server-primary-key/)  – explain you to the primary key concept and show you how to use the primary key constraint to manage a primary key of a table.
* [Foreign key](https://www.sqlservertutorial.net/sql-server-basics/sql-server-foreign-key/) – introduce you to the foreign key concept and show you use the FOREIGN KEYconstraint to enforce the link of data in two tables.
* [NOT NULL constraint](https://www.sqlservertutorial.net/sql-server-basics/sql-server-not-null-constraint/) – show you how to ensure a column not to accept NULL.
* [UNIQUE constraint](https://www.sqlservertutorial.net/sql-server-basics/sql-server-unique-constraint/) – ensure that data contained in a column, or a group of columns, is unique among rows in a table.
* [CHECK constraint](https://www.sqlservertutorial.net/sql-server-basics/sql-server-check-constraint/) – walk you through the process of adding logic for checking data before storing them in tables.

## **Introduction to SQL Server PRIMARY KEY constraint**

A primary key is a column or a group of columns that uniquely identifies each row in a table. You create a primary key for a table by using the PRIMARY KEY constraint.

CREATE TABLE table\_name (

    pk\_column data\_type PRIMARY KEY,

    ...

);

In case the primary key has two or more columns, you must use the PRIMARY KEY constraint as a table constraint:

CREATE TABLE table\_name (

    pk\_column\_1 data\_type,

    pk\_column\_2 data type,

    ...

    PRIMARY KEY (pk\_column\_1, pk\_column\_2)

);

Each table can contain only one primary key. All columns that participate in the primary key must be defined as NOT NULL. SQL Server automatically sets the [NOT NULL](https://www.sqlservertutorial.net/sql-server-basics/sql-server-not-null-constraint/) constraint for all the primary key columns if the NOT NULL constraint is not specified for these columns.

CREATE TABLE sales.activities (

    activity\_id INT PRIMARY KEY,

    activity\_name VARCHAR (255) NOT NULL,

    activity\_date DATE NOT NULL

);

CREATE TABLE sales.participants(

    activity\_id int,

    customer\_id int,

    PRIMARY KEY(activity\_id, customer\_id)

);