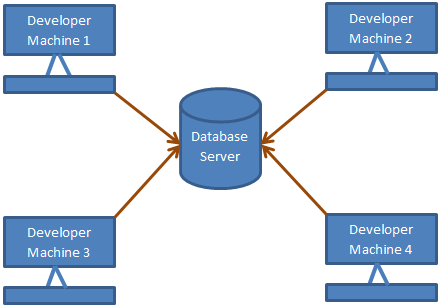
**Creating, altering and dropping a database - Part 2**

In Part 1 of SQL Server, we have seen, using SSMS to connect to SQL Server. In this part we will learn creating, altering and dropping a database.  
  
  
**A SQL Server database can be created, altered and dropped**  
1. Graphically using SQL Server Management Studio (SSMS) or  
2. Using a Query

**To create the database graphically**  
1. Right Click on Databases folder in the Object explorer  
2. Select New Database  
3. In the New Database dialog box, enter the Database name and click OK.  
  
  
**To Create the database using a query**  
Create database DatabaseName  
  
  
**Whether, you create a database graphically using the designer or, using a query, the following 2 files gets generated.**  
.MDF file - Data File (Contains actual data)  
.LDF file - Transaction Log file (Used to recover the database)  
  
  
**To alter a database, once it's created**  
Alter database DatabaseName Modify Name = NewDatabaseName  
  
  
**Alternatively, you can also use system stored procedure**  
Execute sp\_renameDB 'OldDatabaseName','NewDatabaseName'  
  
  
**To Delete or Drop a database**  
Drop Database DatabaseThatYouWantToDrop  
  
  
**Dropping a database, deletes the LDF and MDF files.**  
  
  
You cannot drop a database, if it is currently in use. You get an error stating - Cannot drop database "NewDatabaseName" because it is currently in use. So, if other users are connected, you need to put the database in single user mode and then drop the database.  
Alter Database DatabaseName Set SINGLE\_USER With Rollback Immediate  
  
  
With Rollback Immediate option, will rollback all incomplete transactions and closes the connection to the database.

**SQL Server Management Studio (SSMS)**, is the client tool that can be used to write and execute SQL queries. To connect to the SQL Server Management Studio  
**1.** Click Start   
**2.** Select All Programs  
**3.** Select Microsoft SQL Server 2005, 2008, or 2008 R2 (Depending on the version installed)  
**4.** Select SQL Server Management Studio

**You will now see, Connect to Server window.**  
**1.** Select Database Engine as the Server Type. The other options that you will see here are Analysis Services(SSAS), Reporting Services (SSRS) and Integration Services(SSIS).  
**Server type = Database Engine**  
  
  
**2.** Next you need to specify the Server Name. Here we can specify the name or the server or IP Address.If you have SQL Server installed on your local machine, you can specify, (local) or just . (Period) or 127.0.0.1  
**Server name = (local)**  
  
  
**3.** Now select Authentication. The options available here, depends on how you have installed SQL Server. During installation, if you have chosen **mixed mode authentication**, you will have both Windows Authentication and SQL Server Authentication. Otherwise, you will just be able to connect using windows authentication.  
  
  
4. If you have chosen Windows Authentication, you dont have to enter user name and password, otherwise enter the user name and password and click connect.

You should now be connected to SQL Server. Now, click on**New Query**, on the top left hand corner of SSMS. This should open a new query editor window, where we can type sql queries and execute.  
  
  
SSMS is a client tool and not the Server by itself. Usually database server (SQL Server), will be on a dedicated machine, and developers connect to the server using SSMS from their respective local (development) computers.  
  
  
Developer Machines 1,2,3 and 4 connects to the database server using SSMS.  


## Create Table

The CREATE TABLE statement is used to create a new table in a database. In that table, if you want to add multiple columns, use the below syntax.

**Syntax**

1. **CREATE** **TABLE** table\_name (
2. column1 datatype,
3. column2 datatype,
4. column3 datatype,
5. ....
6. );

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

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

**Create Table Example**

1. **CREATE** **TABLE** Employee(
2. EmpId **int**,
3. LastName **varchar**(255),
4. FirstName **varchar**(255),
5. Address **varchar**(255),
6. City **varchar**(255)
7. );

The EmpId column is of type int and will hold an integer.

The LastName, FirstName, Address, and City columns are of type varchar and will hold characters and the maximum length for these fields is 255 characters.

## Insert Value in this Table

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

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

**Syntax**

The first way specifies both the column names and the values to be inserted.

If you are adding values for all the columns of the table, then no need to specify the column names in the SQL query. However, make sure that the order of the values is in the same order as the columns in the table.

1. **INSERT** **INTO** table\_name (column1, column2, column3, ...)
2. **VALUES** (value1, value2, value3, ...);
4. '2nd way
5. **INSERT** **INTO** table\_name
6. **VALUES** (value1, value2, value3, ...);

**Example**

Insert value in a 1st way. The column names are used here

1. **INSERT** **INTO** Employee    (EmpId,LastName,FirstName,ADDRESS,City)
2. **VALUES** (1, 'XYZ', 'ABC', 'India', 'Mumbai' );
3. **INSERT** **INTO** Employee (EmpId,LastName,FirstName,ADDRESS,City)
4. VALUES (2, 'X', 'A', 'India', 'Pune' );

Insert value in a 2nd way.

1. **INSERT** **INTO** Employee
2. **VALUES** (3, 'XYZ', 'ABC', 'India', 'Mumbai' );

## Select Statment in SQL

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

1. **SELECT** column1, column2, ...
2. **FROM** table\_name;

Here, column1, column2, ... are the field names of the table you want to select from the data. If you want to select all the fields available in the table, use the following syntax:

1. **SELECT** \* **FROM** table\_name;

If the above query is executed, then all record is displayed.

**Example**

1. **Select** EmpId, LastName **from** Employee;
3. **Select** \* **from** Employee;

## Update Table

The UPDATE statement is used to modify the existing records in a table.

**Syntax**

1. **UPDATE** table\_name
2. **SET** column1 = value1, column2 = value2, ...
3. **WHERE** condition;

**Example**

1. **UPDATE** Employee
2. **SET** FirstName= 'KS', City= 'Pune'
3. **WHERE** EmpId= 1;

If the above query is executed then for EmpId= 1, "Firstname" and "City" column data will be updated.

## Update Multiple Rows

It is the WHERE clause that determines how many records will be updated.

1. **UPDATE** Employee
2. **SET** City='Pune'

## Delete Statment in SQL

The DELETE statement is used to delete existing records in a table for a particular Record.

**Syntax**

1. **DELETE** **FROM** table\_name **WHERE** condition;

**Example**

1. **DELETE** **FROM** Employee **WHERE** EmpId=1;

In Employee table EmpId = 1 record gets deleted.

## Delete All Records

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes, and indexes will be intact,

1. **DELETE** **FROM** table\_name;
3. **DELETE** **From** Employee  ;

When the above query is executed, only table Data gets deleted.