* What is Store Procedure?
* Why do we use SET NOCOUNT ON in stored procedure?
* How many types of Store procedure are there?
* How to write comment in SQL SERVER?
* What is naming convention for store procedure?
* Step by Step for How to create SELECT QUERY base store procedure which return all records?
* How to execute store procedure in SQL SERVER?
* What is Parameter?
* Step by Step for How to create PARAMETER base SELECT QUERY store procedure which return records as per parameter passed.?
* Step by Step for How to create a INSERT QUERY base store procedure?
* Step by Step for How to create a UPDATE QUERY base store procedure?
* Step by Step for How to create a DELETE QUERY base store procedure?

# What is Store Procedure?

A store procedure is a collection and set of sql statements and sql command logic which is compiled and stored in the database. A store procedure is one kind of database object which is available under programmability section.

We can reuse one Store procedure where-ever we require it in programming because in store procedure we are writing processing, and insert and update, and delete.

Please refer to the following link to know more,

*https://docs.microsoft.com/en-us/sql/t-sql/statements/create-procedure-transact-sql*

# Why do we use SET NOCOUNT ON in stored procedure?

While we set SET NOCOUNT ON it means there is no messages which shows the number of rows affected.

NOCOUNT means do not count that is ON.

Now you will come know what happened when SET NOCOUNT OFF.

# How many types of Store procedure?

1. User defined store procedure.
2. System store procedure.

User defined store procedure  
In programming mostly we are using this type of store procedure. We are creating this store procedure for receiving return value in tabular or scalar result. User defined store procedure can take input parameters and return output parameters. User defined store procedure is mixture of DDL (Data Definition Language) and DML (Data Manipulation Language ) commands.

User defined store is further classified into two types,

* *T-SQL Store procedure*  
  As the name suggests, T-SQL (Transact SQL) which receives and returns parameter. Processed insert, update and delete query with parameter or without parameter. Returning the rows of output as per query result.
* *CLR Store procedure*  
  CLR (Common Language Runtime) base store procedure which directly related to .NET framework. CLR stored procedure can be written in C#, VB.NET or in any other language which is supported by the .NET Framework. Because of this is written under CLR language thats why its MANAGED CODE. Its got full support and power of .NET framework.

System store procedure  
  
Those store procedure used and run for managing administrative activities of SQL SERVER called System Store Procedure.

Please refer following link to know more about Types of Store procedures,

*https://technet.microsoft.com/en-us/library/ms187644(v=sql.105).aspx*

**STEP BY STEP SECTION**

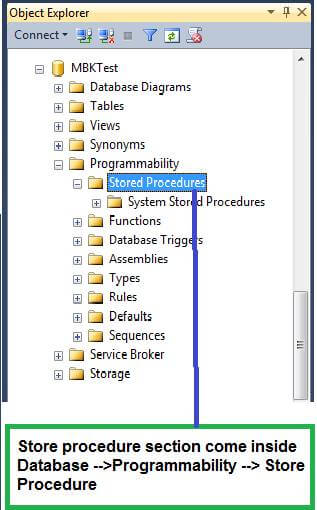
In this section we will do practical of following things,

* Step by Step for How to create SELECT QUERY base store procedure which return all records?
* Step by Step for How to create PARAMETER base SELECT QUERY store procedure which return records as per parameter passed.?
* Step by Step for How to create a INSERT QUERY base store procedure?
* Step by Step for How to create a UPDATE QUERY base store procedure?
* Step by Step for How to create a DELETE QUERY base store procedure?

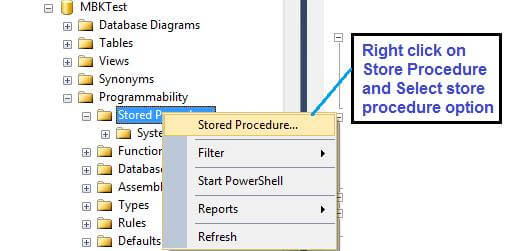
Login in SQL SERVER with your Server Name, Login and Password.



Switch to your database, My Database(DB) name is MBKTest.



Empty template structure of store procedure will be created automatically by following the instructions given in image,



**Empty template structure of store procedure**

1. -- ================================================
2. -- Template generated from Template Explorer using:
3. -- Create Procedure (New Menu).SQL
4. --
5. -- Use the Specify Values for Template Parameters
6. -- command (Ctrl-Shift-M) to fill in the parameter
7. -- values below.
8. --
9. -- This block of comments will not be included in
10. -- the definition of the procedure.
11. -- ================================================
12. **SET** ANSI\_NULLS **ON**
13. GO
14. **SET** QUOTED\_IDENTIFIER **ON**
15. GO
16. -- =============================================
17. -- Author:      <Author,,Name>
18. -- Create date: <Create Date,,>
19. -- Description: <Description,,>
20. -- =============================================
21. **CREATE** **PROCEDURE** <Procedure\_Name, sysname, ProcedureName>
22. -- Add the parameters for the stored procedure here
23. <@Param1, sysname, @p1> <Datatype\_For\_Param1, , **int**> = <Default\_Value\_For\_Param1, , 0>,
24. <@Param2, sysname, @p2> <Datatype\_For\_Param2, , **int**> = <Default\_Value\_For\_Param2, , 0>
25. **AS**
26. **BEGIN**
27. -- SET NOCOUNT ON added to prevent extra result sets from
28. -- interfering with SELECT statements.
29. **SET** NOCOUNT **ON**;
31. -- Insert statements for procedure here
32. **SELECT** <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
33. **END**
34. GO

**How to write comment in SQL SERVER?**

1. -- (two hyphens / dash) for single line commenting.
2. /\* ……. end with  \*/    for multi line commenting.

# What is naming convention for store procedure?

For user defined store procedure naming convention my suggestions are as follows,

1. stp
2. stp\_
3. udstp
4. udstp\_

Naming convention is just to identify the object.

Please refer to the following link for more knowledge comment,

*https://technet.microsoft.com/en-us/library/ms188621(v=sql.105).aspx*

My Table Name is “tblMembers” and following is table structure of tblMembers,

1. USE [MBKTest]
2. GO
4. /\*\*\*\*\*\* Object:  **Table** [dbo].[tblMembers]    Script **Date**: 18-Nov-17,Sat 6:47:55 PM \*\*\*\*\*\*/
5. **SET** ANSI\_NULLS **ON**
6. GO
8. **SET** QUOTED\_IDENTIFIER **ON**
9. GO
11. **SET** ANSI\_PADDING **ON**
12. GO
14. **CREATE** **TABLE** [dbo].[tblMembers](
15. [MemberID] [**int**] IDENTITY(1,1) NOT NULL,
16. [MemberName] [**varchar**](50) NULL,
17. [MemberCity] [**varchar**](25) NULL,
18. [MemberPhone] [**varchar**](15) NULL
19. )
21. GO
23. **SET** ANSI\_PADDING **OFF**
24. GO

**Step by Step for How to create SELECT QUERY base store procedure which return all records?**

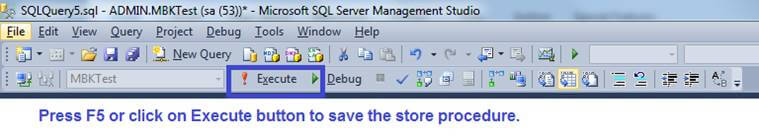
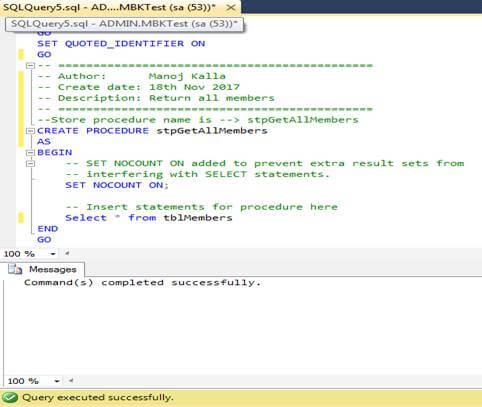
A very simple kind of store procedure which returns all records of table and joining tables.

Click on your Database and expand “Programmability” tab and right click on “Store Procedures” or press CTRL + N to get new query window with current connection.

**Store procedure CODE for return all records of table.**

1. **SET** ANSI\_NULLS **ON**
2. GO
3. **SET** QUOTED\_IDENTIFIER **ON**
4. GO
5. -- =============================================
6. -- Author:      Manoj Kalla
7. -- Create date: 18th Nov 2017
8. -- Description: Return all members
9. -- =============================================
10. --Store procedure name is --> stpGetAllMembers
11. **C REATE** **PROCEDURE** stpGetAllMembers
12. **AS**
13. **BEGIN**
14. -- SET NOCOUNT ON added to prevent extra result sets from
15. -- interfering with SELECT statements.
16. **SET** NOCOUNT **ON**;
18. -- Insert statements for procedure here
19. **Select** \* **from** tblMembers
20. **END**
21. GO

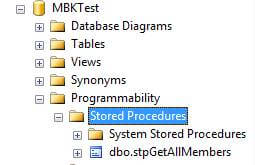
After writing above code press F5 or click on EXECUTE button see image for more detail.

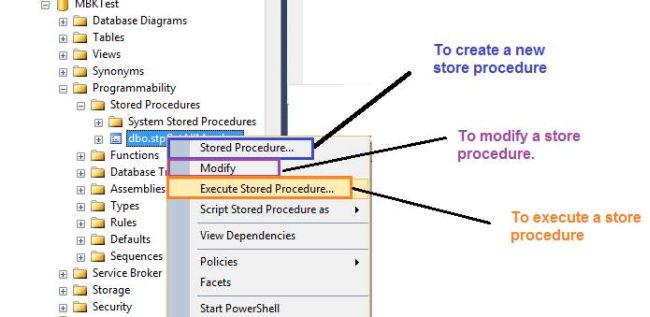
As you can see in image if your store procedure is saved successfully in Messages window you will get message “Command(s) completed successfully.”

Now again click on Programmability -->Store Procedure and right click on store procedure and select REFRESH.

You can see in image store procedure called stpGetAllMembers is created.



**How to execute store procedure in SQL SERVER?**

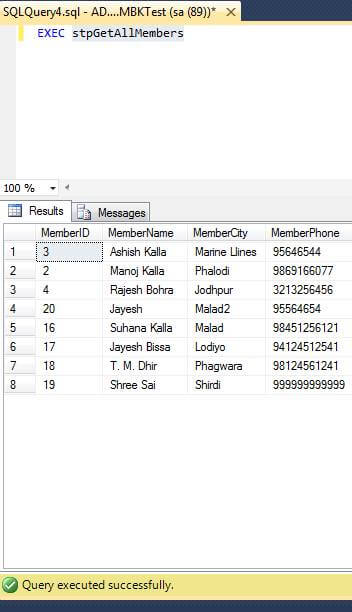


Or simply type following,

* Syntax - EXEC <store procedure name>
* Example - EXEC stpGetAllMembers

For more detail you can visit the following link,

*https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/execute-a-stored-procedure*



# What is Parameter?

Exchanging data between store procedure.

1. Input Parameter - Here we pass the value to store procedure.
2. Output Parameter - Here we receive the value from store procedure.

Please visit the following link for more detail,

*https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/parameters*

**Step by Step for How to create PARAMETER base SELECT QUERY store procedure which return records as per parameter passed.?**

Here Parameter means we want or check desired records is available in table or not by sending parameter value inside store procedure.

Click on your Database and expand “Programmability” tab and right click on “Store Procedures” or press CTRL + N to get new query window with current connection and type following codes.

**Store procedure CODE for return specific records which satisfy the condition of parameters in table.**

1. **SET** ANSI\_NULLS **ON**
2. GO
3. **SET** QUOTED\_IDENTIFIER **ON**
4. GO
5. -- =============================================
6. -- Author:      Manoj Kalla
7. -- Create date: 20-Nov-2017
8. -- Description: Return specifc city records
9. -- =============================================
10. **CREATE** **PROCEDURE** stpGetMembersByCityName
11. -- Add the parameters for the stored procedure here
12. @CityName nvarchar(30)
14. **AS**
15. **BEGIN**
16. -- SET NOCOUNT ON added to prevent extra result sets from
17. -- interfering with SELECT statements.
18. **SET** NOCOUNT **ON**;
20. **Select** \* **From** tblMembers
21. **where** MemberCity like '%'+@CityName+'%'
23. **END**
24. GO

Now we run our store procedure called stpGetMembersByCityName

Refresh the Database or Programmability option, you can easily see your stpGetMembersByCityName

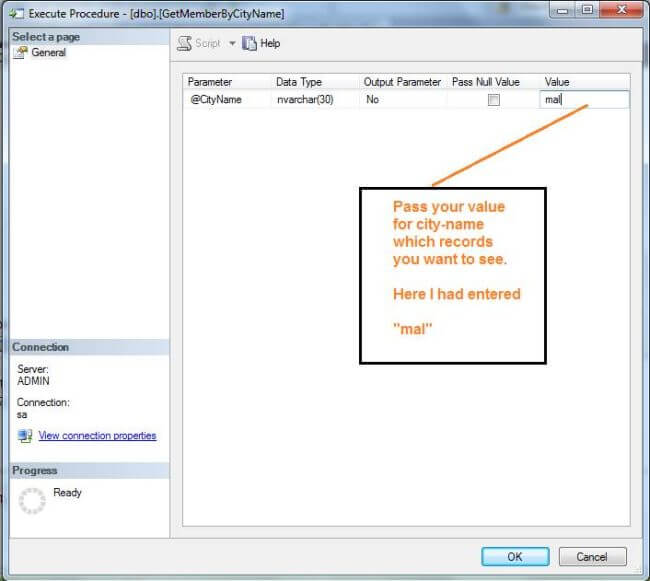
Store procedure under Store procedure option.

You can run above store procedure with coding / manually or UI.

**By Coding / Manually**

*EXEC GetMemberByCityName   @CityName = 'mal'*

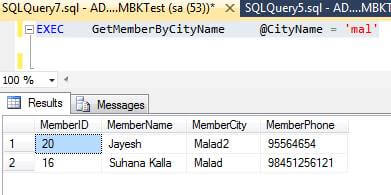
By UI means store procedure will execute your procedure with required parameter.



Folllowing is written by sql server management studio in behind to run and execute the store procedure.

1. USE [MBKTest]
2. GO
4. **DECLARE** @return\_value **int**
6. **EXEC**    @return\_value = [dbo].[GetMemberByCityName]
7. @CityName = N'mal'
9. **SELECT**  'Return Value' = @return\_value
11. GO

**OUTPUT**



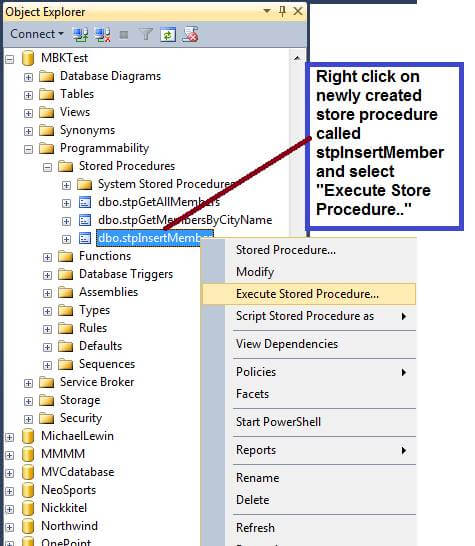
**Step by Step for How to create a INSERT QUERY base store procedure?**

In this step by step you will learn how to insert an new record with the help of store procedure. We will going to write INSERT query inside store procedure.

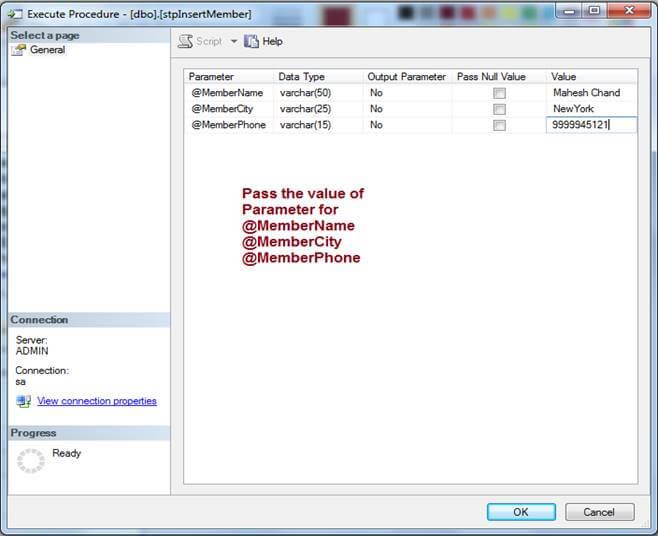
In the following code you can see there is no mention of MEMBER ID column parameter because that field / column is set as identity column and auto increment.

1. **SET** ANSI\_NULLS **ON**
2. GO
3. **SET** QUOTED\_IDENTIFIER **ON**
4. GO
5. -- =============================================
6. -- Author:      Manoj Kalla
7. -- Create date: 20-Nov-2047
8. -- Description: To create a new member
9. -- =============================================
10. **CREATE** **PROCEDURE** stpInsertMember
11. @MemberName **varchar**(50),
12. @MemberCity **varchar**(25),
13. @MemberPhone **varchar**(15)
15. **AS**
16. **BEGIN**
17. -- SET NOCOUNT ON added to prevent extra result sets from
18. -- interfering with SELECT statements.
19. **SET** NOCOUNT **ON**;
21. **Insert** **into** tblMembers (MemberName,MemberCity,MemberPhone)
22. **Values** (@MemberName,@MemberCity, @MemberPhone)
24. **END**
25. GO

Right click on Store Procedure inside Object Explorer of your Database and select REFRESH



Pass the value of parameter in Execute dialog box. Please refer to the following screen shot,

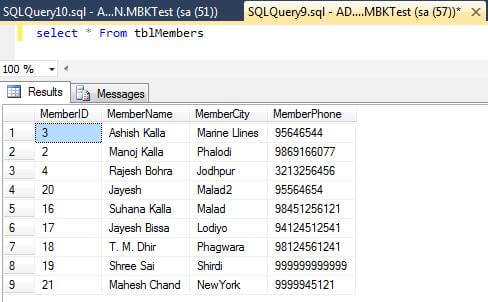


Following code is automatically written by SQL SERVER MANAGEMENT STUDIO,

1. USE [MBKTest]
2. GO
4. **DECLARE** @return\_value **int**
6. **EXEC**    @return\_value = [dbo].[stpInsertMember]
7. @MemberName = N'Mahesh Chand',
8. @MemberCity = N'NewYork',
9. @MemberPhone = N'9999945121'
10. **SELECT**  'Return Value' = @return\_value
11. GO

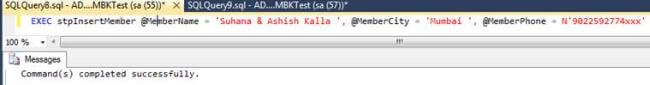
**OUTPUT**

In query window you can check by query for above record to see if  Mahesh Chand sir is created or not.



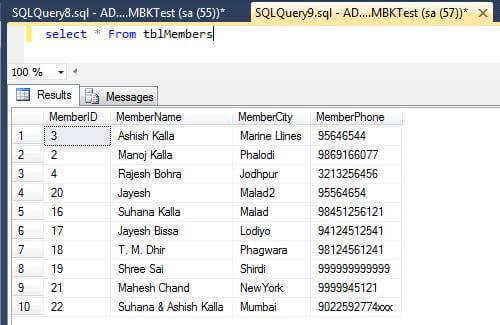
By using Manual you can run and insert store procedure by this way,

*EXEC stpInsertMember @MemberName = 'Suhana & Ashish Kalla ', @MemberCity = 'Mumbai ', @MemberPhone = N'9022592774xxx'*



**OUTPUT**

You can check “Suhana & Ashish Kalla” record added successfully.



**Step by Step for How to create a UPDATE QUERY base store procedure?**

In this step by step you will learn how to update an existing record with the help of store procedure by passing ID of record. We are going to write UPDATE query inside store procedure.

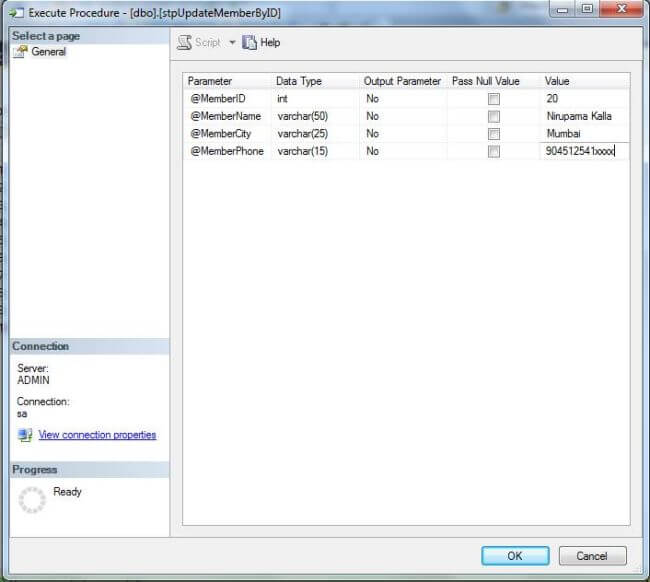
In the following code you can see the code mentions MEMBER ID column parameter

1. **SET** ANSI\_NULLS **ON**
2. GO
3. **SET** QUOTED\_IDENTIFIER **ON**
4. GO
5. -- =============================================
6. -- Author:      Manoj Kalla
7. -- Create date: 20-Nov-2017
8. -- Description: Update a member detail by ID
9. -- =============================================
10. **CREATE** **PROCEDURE** stpUpdateMemberByID
11. @MemberID **int**,
12. @MemberName **varchar**(50),
13. @MemberCity **varchar**(25),
14. @MemberPhone **varchar**(15)
16. **AS**
17. **BEGIN**
18. -- SET NOCOUNT ON added to prevent extra result sets from
19. -- interfering with SELECT statements.
20. **SET** NOCOUNT **ON**;
22. **UPDATE** tblMembers
23. **Set** MemberName = @MemberName,
24. MemberCity = @MemberCity,
25. MemberPhone = @MemberPhone
26. **Where** MemberID = @MemberID
27. **END**
28. GO

Right click on Store Procedure inside Object Explorer of your Database and select REFRESH

**RUN UPDATE STORE PROCEDURE BY UI**

Now again right click on store procedure and select Execute Store procedure…



Following code is automatically written by SQL SERVER MANAGEMENT STUDIO,

1. USE [MBKTest]
2. GO
4. **DECLARE** @return\_value **int**
6. **EXEC**    @return\_value = [dbo].[stpUpdateMemberByID]
7. @MemberID = 20,
8. @MemberName = N'Nirupama Kalla',
9. @MemberCity = N'Mumbai',
10. @MemberPhone = N'904512541xxxx'
12. **SELECT**  'Return Value' = @return\_value
14. GO

**RUN UPDATE STORE PROCEDURE BY MANUALLY (CODING)**

*EXEC stpUpdateMemberByID  17,'Gopal Madhavrai','Bikaner','90454564xxx'*

You can check previous image and see there are other details on Member ID = 20,17, Now it's updated.



**Step by Step for How to create a DELETE QUERY base store procedure?**

In this step by step you will learn to delete a record through store procedure. So far now you have seen how to  Insert and update the record via store procedure.

**Delete Store Procedure Code**

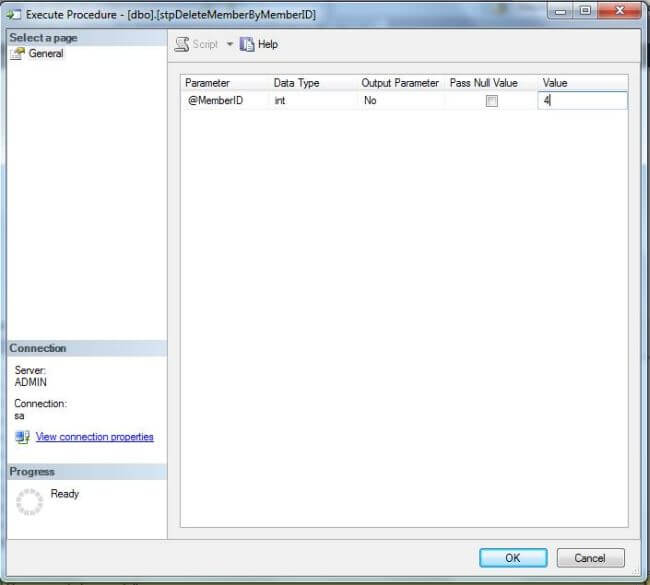
1. **SET** ANSI\_NULLS **ON**
2. GO
3. **SET** QUOTED\_IDENTIFIER **ON**
4. GO
5. -- =============================================
6. -- Author:      Manoj Kalla
7. -- Create date: 21-Nov-2017
8. -- Description: Delete a Member by Member ID
9. -- =============================================
10. **CREATE** **PROCEDURE** stpDeleteMemberByMemberID
11. @MemberID **int**
12. **AS**
13. **BEGIN**
14. -- SET NOCOUNT ON added to prevent extra result sets from
15. -- interfering with SELECT statements.
16. **SET** NOCOUNT **ON**;
18. **Delete** **from** tblMembers
19. **where** MemberId = @MemberID
21. **END**
22. GO

Right click on Store Procedure inside Object Explorer of your Database and select REFRESH

**RUN STORE PROCEDURE BY UI**

Now again right click on store procedure and select Execute Store procedure…

As you can see in the image I passed @MemberID parameter value = 4.



**RUN DELETE STORE PROCEDURE BY MANUALLY (CODING)**

*EXEC stpDeleteMemberByMemberID  2*

**OUTPUT**

You can see in image MemberID = 4 record has deleted successfully.  
  
