**SQL**

* Case Insensitive (i.e. select is same as SELECT)

**Some of The Most Important SQL Commands**

* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index

**Use SQL Server Management Studio**

To create a stored procedure in SSMS:

1. In **Object Explorer**, connect to an instance of SQL Server or Azure SQL Database.

For more information, see the following quick starts:

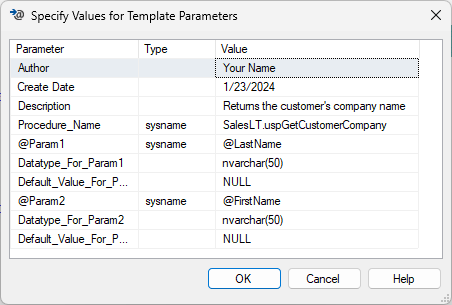
* + [Connect and query a SQL Server instance using SSMS](https://learn.microsoft.com/en-us/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-ver16)
  + [Connect to and query Azure SQL Database or Azure SQL Managed Instance using SSMS](https://learn.microsoft.com/en-us/azure/azure-sql/database/connect-query-ssms)

1. Expand the instance, and then expand **Databases**.
2. Expand the database that you want, and then expand **Programmability**.
3. Right-click **Stored Procedures**, and then select **New** > **Stored Procedure**. A new query window opens with a template for the stored procedure.

The default stored procedure template has two parameters. If your stored procedure has fewer, more, or no parameters, add or remove parameter lines in the template as appropriate.

1. On the **Query** menu, select **Specify Values for Template Parameters**.
2. In the **Specify Values for Template Parameters** dialog box, provide the following information for the **Value** fields:
   * **Author**: Replace Name with your name.
   * **Create Date**: Enter today's date.
   * **Description**: Briefly describe what the procedure does.
   * **Procedure\_Name**: Replace ProcedureName with the new stored procedure name.
   * **@Param1**: Replace @p1 with your first parameter name, such as *@ColumnName1*.
   * **@Datatype\_For\_Param1**: As appropriate, replace int with your first parameter's datatype, such as *nvarchar(50)*.
   * **Default\_Value\_For\_Param1**: As appropriate, replace 0 with your first parameter's default value, or *NULL*.
   * **@Param2**: Replace @p2 with your second parameter name, such as *@ColumnName2*.
   * **@Datatype\_For\_Param2**: As appropriate, replace int with your second parameter's datatype, such as *nvarchar(50)*.
   * **Default\_Value\_For\_Param2**: As appropriate, replace 0 with your second parameter's default value, or *NULL*.

The following screenshot shows the completed dialog box for the example stored procedure:



1. Select **OK**.
2. In the **Query Editor**, replace the SELECT statement with the query for your procedure.

The following code shows the completed CREATE PROCEDURE statement for the example stored procedure:

SQLCopy

-- =======================================================

-- Create Stored Procedure Template for Azure SQL Database

-- =======================================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE SalesLT.uspGetCustomerCompany

(

-- Add the parameters for the stored procedure here

@LastName nvarchar(50) = NULL,

@FirstName nvarchar(50) = NULL

)

AS

/\*

-- =============================================

-- Author: My Name

-- Create Date: 01/23/2024

-- Description: Returns the customer's company name.

-- =============================================

\*/

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON

-- Insert statements for procedure here

SELECT FirstName, LastName, CompanyName

FROM SalesLT.Customer

WHERE FirstName = @FirstName AND LastName = @LastName;

END

GO

1. To test the syntax, on the **Query** menu, select **Parse**. Correct any errors.
2. Select **Execute** from the toolbar. The procedure is created as an object in the database.
3. To see the new procedure listed in **Object Explorer**, right-click **Stored Procedures** and select **Refresh**.

To run the procedure:

1. In **Object Explorer**, right-click the stored procedure name and select **Execute Stored Procedure**.
2. In the **Execute Procedure** window, enter values for all parameters, and then select **OK**. For detailed instructions, see [Execute a stored procedure](https://learn.microsoft.com/en-us/sql/relational-databases/stored-procedures/execute-a-stored-procedure?view=sql-server-ver16#SSMSProcedure).

For example, to run the SalesLT.uspGetCustomerCompany sample procedure, enter *Cannon* for the **@LastName** parameter and *Chris* for the **@FirstName** parameter, and then select **OK**. The stored procedure runs, and returns FirstName **Chris**, LastName **Cannon**, and CompanyName **Outdoor Sporting Goods**.

**Important**

Validate all user input. Don't concatenate user input before you validate it. Never execute a command constructed from unvalidated user input.

**Stored Procedure Commands**

1. **Select Department**

CREATE PROCEDURE selectDepartment

--Declare the Parameters

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Department;

END

1. **Select Department using Function**

--to create functions

CREATE Function FgetDepartment ()

Returns table as

return(

select \* from Department

)

--to call that function

Select \* from FgetDepartment()

1. **Select Department by ID**

CREATE PROCEDURE selectDepartmentById

--Declare the Parameters

@id int

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Department WHERE DepartmentId = @id;

END

1. **Select Department by Name**

CREATE PROCEDURE selectDepartmentByName

--Declare the Parameters

@name NVARCHAR(100)

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Department WHERE DepartmentName = @name;

END

1. **Insert Department Data**

CREATE PROCEDURE insertDepartmentData

--Declare the Parameters

@name NVARCHAR(100)

AS

BEGIN

--Insert statements for procedure here

INSERT INTO Department (DepartmentName)

VALUES (@name);

END

1. **Update the Department Details**

CREATE PROCEDURE updateDepartmentData

--Declare the Parameters

@id int,

@name NVARCHAR(100)

AS

BEGIN

--Insert statements for procedure here

UPDATE Department

SET DepartmentName = @name

WHERE DepartmentId=@id;

END

--to execute this stored procedure, use this syntax

EXEC updateDepartmentData @id=2, @name='Frontend Developer'

Or

EXEC updateDepartmentData 2, 'Frontend Developer'

1. **Delete the Department Details**

CREATE PROCEDURE deleteDepartmentData

--Declare the Parameters

@id int

AS

BEGIN

--Insert statements for procedure here

DELETE FROM Department

WHERE DepartmentId=@id;

END

--to execute above stored procedure use below command:

EXEC deleteDepartmentData @id=5

1. **Select Designation**

CREATE PROCEDURE selectDesignation

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Designation;

END

1. **Select Designation using Function**

--to create functions

CREATE Function FgetDesignation ()

Returns table as

return(

select \* from Designation

)

--to call that function

Select \* from FgetDesignation()

1. **Select Designation by Id**

CREATE PROCEDURE selectDesignationById

--Declare the Parameters

@id Int

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Designation WHERE DesignationId = @id;

EN

1. **Select Designation by Name**

CREATE PROCEDURE selectDesignationByName

--Declare the Parameters

@name NVARCHAR(100)

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Designation WHERE DesignationName = @name;

END

1. **Insert Designation Data**

CREATE PROCEDURE insertDesignationData

--Declare the Parameters

@name NVARCHAR(100)

AS

BEGIN

--Insert statements for procedure here

INSERT INTO Designation (DesignationName)

VALUES (@name);

END

1. **Update the Designation Details**

CREATE PROCEDURE updateDesignationData

--Declare the Parameters

@id int,

@name NVARCHAR(100)

AS

BEGIN

--Insert statements for procedure here

UPDATE Designation

SET DesignationName = @name

WHERE DesignationId=@id;

END

--to Execute this stored procedure type following syntax:

EXEC updateDesignationData @id=5, @name='Mid Level'

1. **Delete Designation Details**

CREATE PROCEDURE deleteDesignationData

--Declare the Parameters

@id int

AS

BEGIN

--Insert statements for procedure here

DELETE FROM Designation

WHERE DesignationId=@id;

END

--to execute above stored procedure use below command:

EXEC deleteDesignationData @id=6

1. **Select Employee**

CREATE PROCEDURE selectEmployee

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Employee;

END

1. **Select Employee Function**

CREATE Function FgetEmployee ()

Returns table as

return(

select \* from Employee

)

1. **Select Employee By Id**

CREATE PROCEDURE selectEmployeeById

@id int

AS

BEGIN

--Insert statements for procedure here

SELECT \* FROM Employee WHERE EmployeeId = @id;

END

1. **Insert Employee Data**

CREATE PROCEDURE insertEmployeeData

--Declare the Parameters

@fname NVARCHAR(50),

@lname NVARCHAR(50),

@email NVARCHAR(255),

@phnum VARCHAR(15),

@departId int,

@desigId int

AS

BEGIN

--Insert statements for procedure here

INSERT INTO Employee (FirstName,LastName, Email, PhoneNumber, DepartmentId, DesignationId)

VALUES (@fname,@lname,@email,@phnum,@departId,@desigId);

END

1. **Update Employee Details**

CREATE PROCEDURE updateEmployeeData

--Declare the Parameters

@id int,

@fname NVARCHAR(50),

@lname NVARCHAR(50),

@email NVARCHAR(255),

@phnum VARCHAR(15),

@departId int,

@desigId int

AS

BEGIN

--Insert statements for procedure here

UPDATE Employee

SET FirstName=@fname,LastName=@lname, Email=@email, PhoneNumber=@phnum, DepartmentId=@departId, DesignationId=@desigId

WHERE EmployeeId=@id;

END

--to use this stored procedure use this syntax

EXEC updateEmployeeData @id=2,@fname='ram',@lname='Sima',@email='ramsima@gmail.com',@phnum='9874444637',@departId=1,@desigId=2;

1. **Delete Employee Details**

CREATE PROCEDURE deleteEmployeeData

--Declare the Parameters

@id int

AS

BEGIN

--Insert statements for procedure here

DELETE FROM Employee

WHERE EmployeeId=@id;

END

--to execute above stored procedure use below command:

EXEC deleteEmployeeData @id=2