**SQL SERVER Advanced 5**

## Introduction SQL Server DROP TRIGGER statements

The SQL Server DROP TRIGGER statement drops one or more triggers from the database. The following illustrates the syntax of the DROP TRIGGER statement that removes [DML triggers](https://www.sqlservertutorial.net/sql-server-triggers/sql-server-create-trigger/):

DROP TRIGGER [ IF EXISTS ] [schema\_name.]trigger\_name [ ,...n ];

Code language: SQL (Structured Query Language) (sql)

In this syntax:

* IF EXISTS conditionally removes the trigger only when it already exists.
* schema\_name is the name of the schema to which the DML trigger belongs.
* trigger\_name is the name of the trigger that you wish to remove.

If you want to remove multiple triggers at once, you need to separate triggers by commas.

To remove one or more [DDL triggers](https://www.sqlservertutorial.net/sql-server-triggers/sql-server-ddl-trigger/), you use the following form of the DROP TRIGGER statement:

DROP TRIGGER [ IF EXISTS ] trigger\_name [ ,...n ] ON { DATABASE | ALL SERVER };

Code language: SQL (Structured Query Language) (sql)

In this syntax:

* DATABASE indicates that the scope of the DDL trigger applies to the current database.
* ALL SERVER indicates the scope of the DDL trigger applies to the current server.

To remove a LOGON event trigger, you use the following syntax:

DROP TRIGGER [ IF EXISTS ] trigger\_name [ ,...n ] ON ALL SERVER;

Code language: SQL (Structured Query Language) (sql)

Notice that when you [drop a table](https://www.sqlservertutorial.net/sql-server-basics/sql-server-drop-table/), all triggers associated with the table are also removed automatically.

## SQL Server DROP TRIGGER examples

### A) SQL Server DROP TRIGGER – drop a DML trigger example

The following statement drops a DML trigger named sales.trg\_member\_insert:

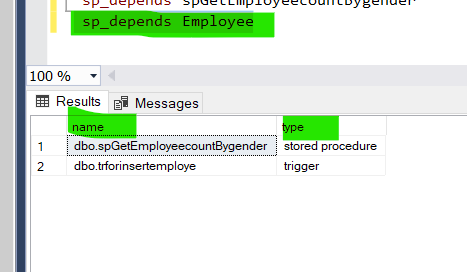
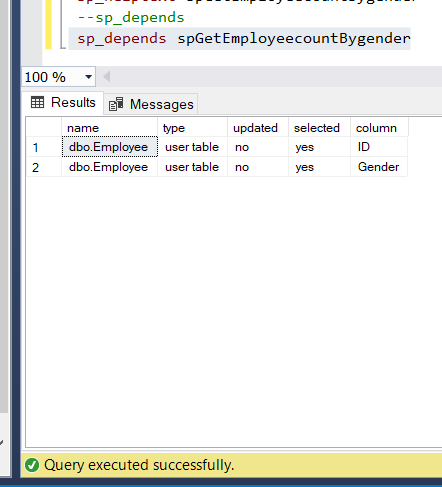
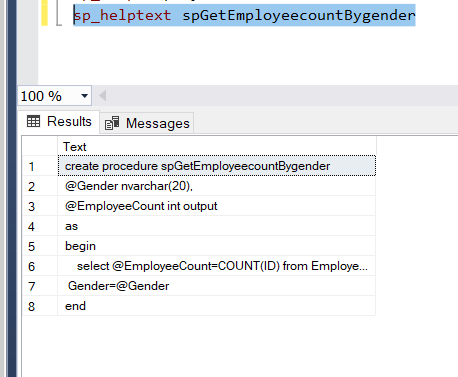
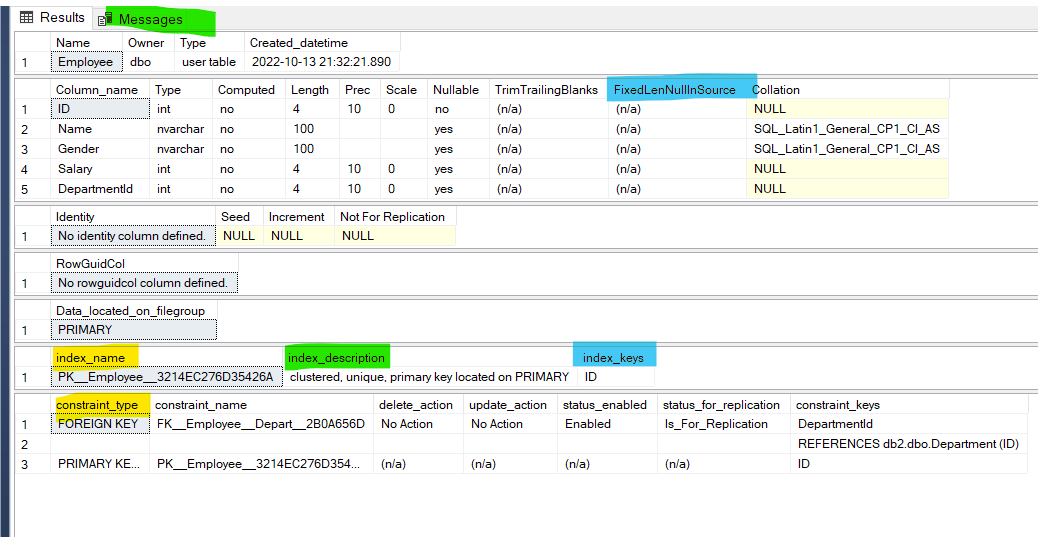
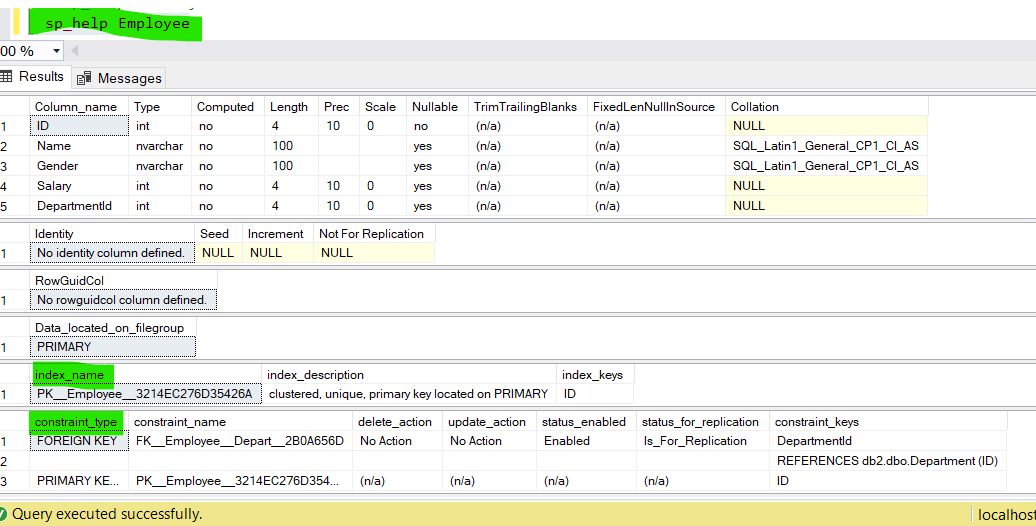
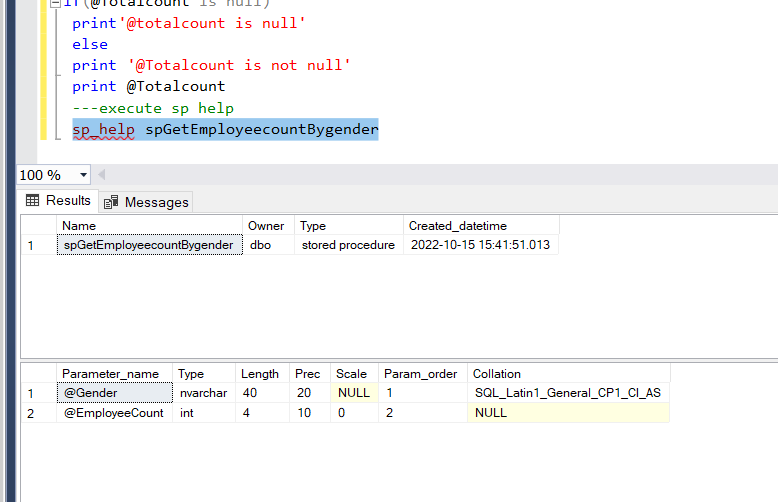
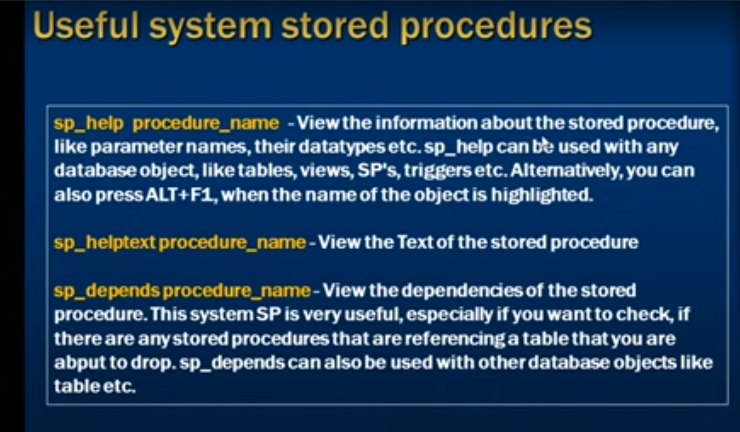
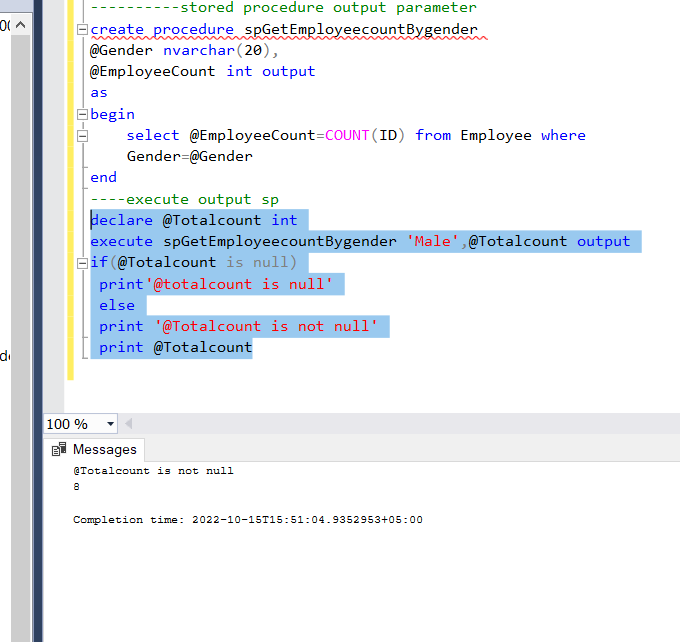
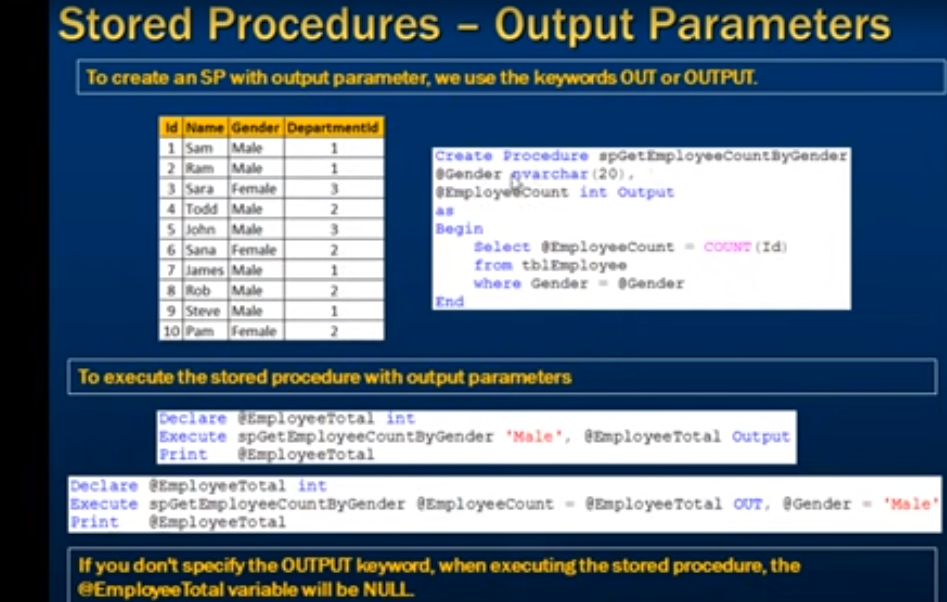
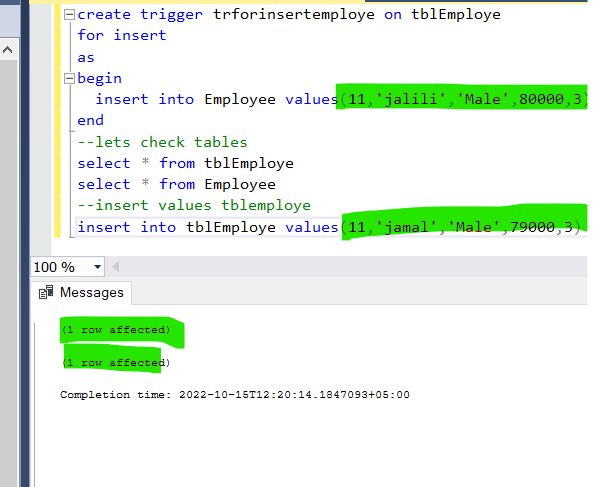
DROP TRIGGER IF EXISTS sales.trg\_member\_insert;

Code language: SQL (Structured Query Language) (sql)

### B) SQL Server DROP TRIGGER – drop a DDL trigger example

The following statement removes the trg\_index\_changes trigger:

DROP TRIGGER IF EXISTS trg\_index\_changes;



create trigger trforinsertemploye on tblEmploye

for insert

as

begin

insert into Employee values(11,'jalili','Male',80000,3)

end

--lets check tables

select \* from tblEmploye

select \* from Employee

--insert values tblemploye

insert into tblEmploye values(11,'jamal','Male',79000,3)

--check

select \* from Employee

----------stored procedure output parameter

create procedure spGetEmployeecountBygender

@Gender nvarchar(20),

@EmployeeCount int output

as

begin

select @EmployeeCount=COUNT(ID) from Employee where

Gender=@Gender

end

----execute output sp

declare @Totalcount int

execute spGetEmployeecountBygender 'Male',@Totalcount output

if(@Totalcount is null)

print'@totalcount is null'

else

print '@Totalcount is not null'

print @Totalcount

---execute sp help

sp\_help spGetEmployeecountBygender

--sp\_help for any table

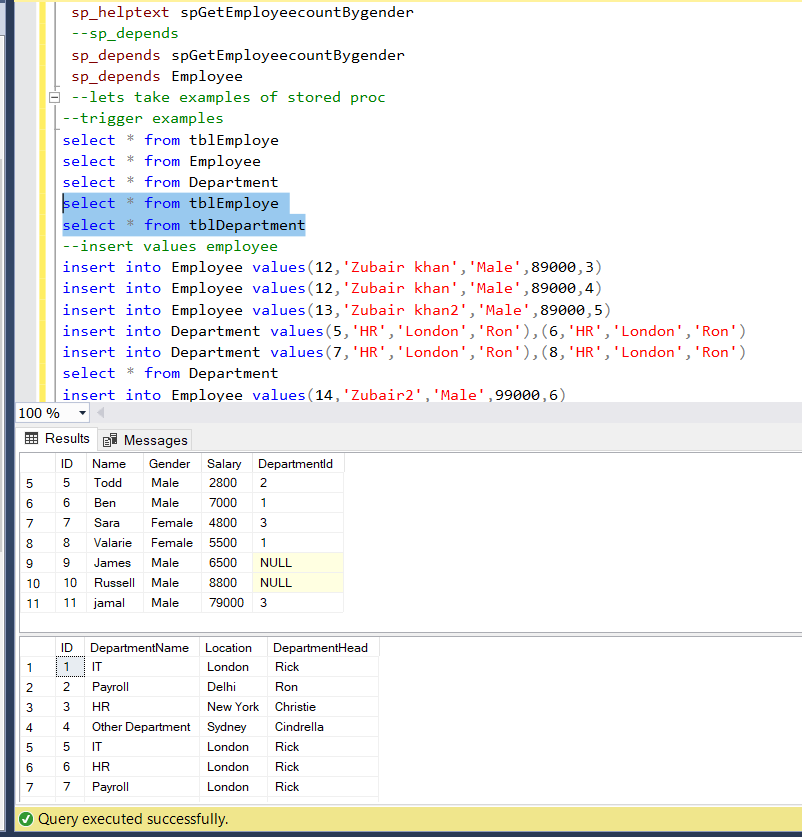
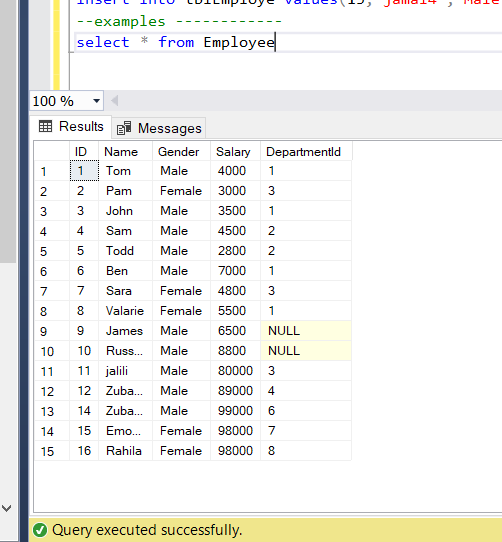
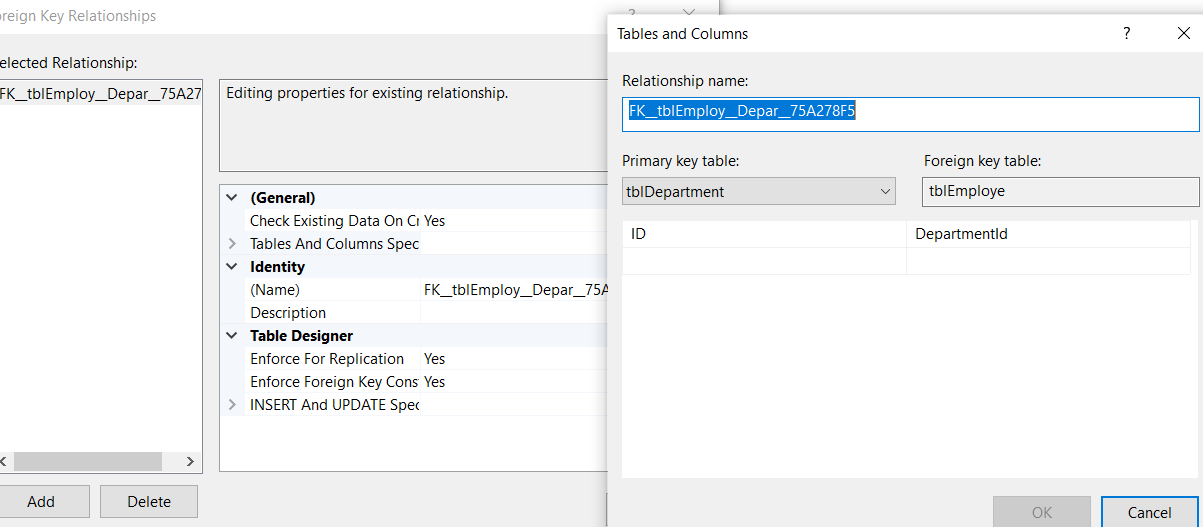
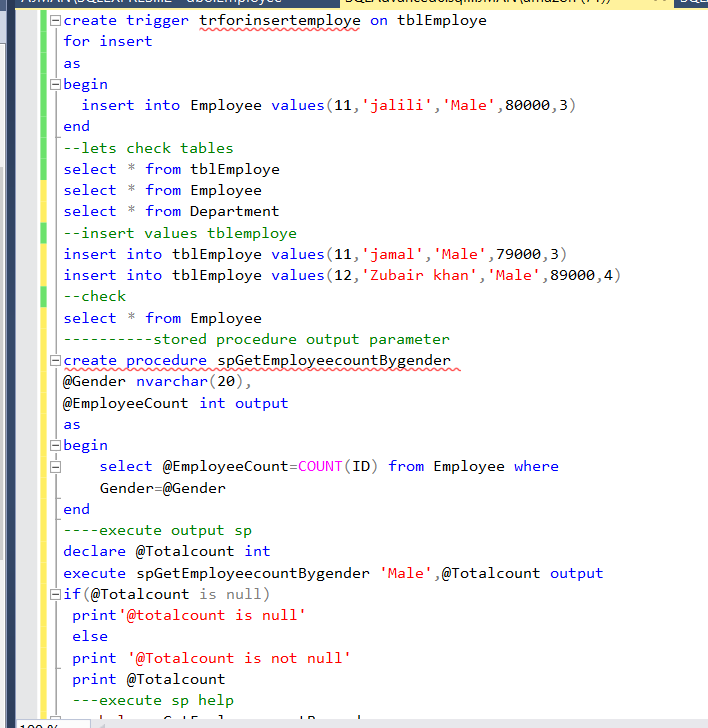
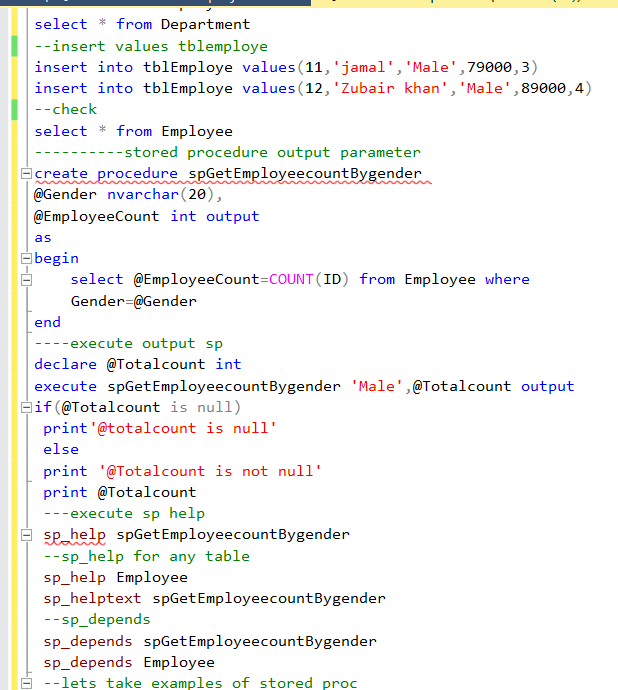
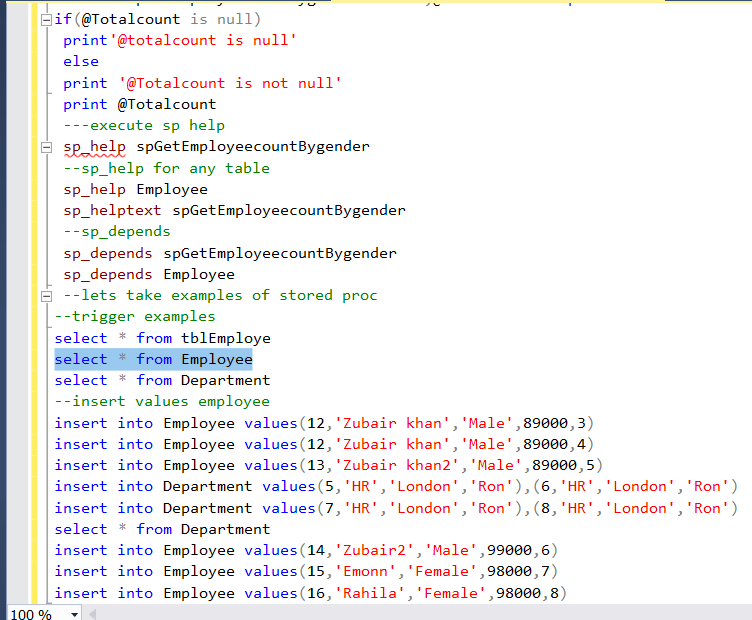
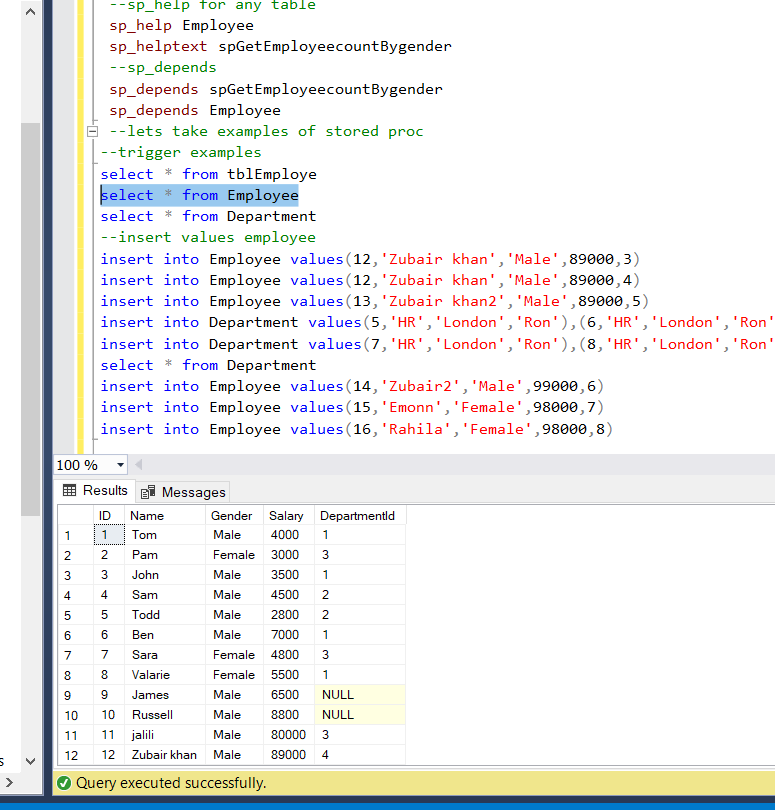
sp\_help Employee

sp\_helptext spGetEmployeecountBygender

--sp\_depends

sp\_depends spGetEmployeecountBygender

sp\_depends Employee



////////////////////////////////////////////////////////////

create trigger trforinsertemploye on tblEmploye

for insert

as

begin

insert into Employee values(11,'jalili','Male',80000,3)

end

--lets check tables

select \* from tblEmploye

select \* from Employee

select \* from Department

--insert values tblemploye

insert into tblEmploye values(11,'jamal','Male',79000,3)

insert into tblEmploye values(12,'Zubair khan','Male',89000,4)

--check

select \* from Employee

----------stored procedure output parameter

create procedure spGetEmployeecountBygender

@Gender nvarchar(20),

@EmployeeCount int output

as

begin

select @EmployeeCount=COUNT(ID) from Employee where

Gender=@Gender

end

----execute output sp

declare @Totalcount int

execute spGetEmployeecountBygender 'Male',@Totalcount output

if(@Totalcount is null)

print'@totalcount is null'

else

print '@Totalcount is not null'

print @Totalcount

---execute sp help

sp\_help spGetEmployeecountBygender

--sp\_help for any table

sp\_help Employee

sp\_helptext spGetEmployeecountBygender

--sp\_depends

sp\_depends spGetEmployeecountBygender

sp\_depends Employee

--lets take examples of stored proc

--trigger examples

select \* from tblEmploye

select \* from Employee

select \* from Department

select \* from tblEmploye

select \* from tblDepartment

--insert values employee

insert into Employee values(12,'Zubair khan','Male',89000,3)

insert into Employee values(12,'Zubair khan','Male',89000,4)

insert into Employee values(13,'Zubair khan2','Male',89000,5)

insert into Department values(5,'HR','London','Ron'),(6,'HR','London','Ron')

insert into Department values(7,'HR','London','Ron'),(8,'HR','London','Ron')

select \* from Department

insert into Employee values(14,'Zubair2','Male',99000,6)

insert into Employee values(15,'Emonn','Female',98000,7)

insert into Employee values(16,'Rahila','Female',98000,8)

insert into tblDepartment values(5,'IT','London','Rick')

insert into tblDepartment values(6,'HR','London','Rick')

insert into tblDepartment values(7,'Payroll','London','Rick')

insert into tblDepartment values(8,'Payroll','Delhi','Ron')

insert into tblEmploye values(12,'jamal3','Male',79000,5)

insert into tblEmploye values(13,'jamal4','Male',99000,6)

--examples ------------

select \* from Employee

--lets execute trigger

create trigger tremployeebynamegender on Employee

for update

as

begin

update Employee set Name='ajman' where ID=5

end

