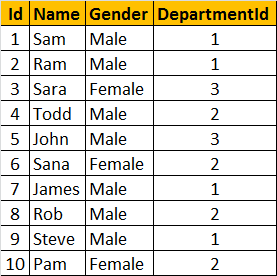
**Stored procedures with output parameters - Part 19**

In this video, we will learn about, creating stored procedures with output parameters. [Please watch Part 18 of this video series, before watching](https://www.blogger.com/goog_350135183)[this video.](http://csharp-video-tutorials.blogspot.com/2012/08/stored-procedures-part-18.html)  
  
  
  
**To create an SP with output parameter**, we use the keywords OUT or OUTPUT. @EmployeeCount is an OUTPUT parameter. Notice, it is specified with OUTPUT keyword.   
Create Procedure spGetEmployeeCountByGender  
@Gender nvarchar(20),  
@EmployeeCount int Output  
as  
Begin  
 Select @EmployeeCount = COUNT(Id)   
 from tblEmployee   
 where Gender = @Gender  
End  
  
  
  
  
  
  
  
  
**To execute this stored procedure with OUTPUT parameter**  
  
**1.** First initialise a variable of the **same datatype** as that of the **output parameter**. We have declared @EmployeeTotal integer variable.   
**2.** Then pass the @EmployeeTotal variable to the SP. You have to specify the **OUTPUT** keyword. If you don't specify the OUTPUT keyword, the variable will be **NULL**.   
**3.** Execute  
  
Declare @EmployeeTotal int  
Execute spGetEmployeeCountByGender 'Female', @EmployeeTotal output  
Print @EmployeeTotal  
  
If you don't specify the OUTPUT keyword, when executing the stored procedure, the @EmployeeTotal variable will be NULL. Here, we have not specified OUTPUT keyword. When you execute, you will see **'@EmployeeTotal is null'** printed.  
  
Declare @EmployeeTotal int  
Execute spGetEmployeeCountByGender 'Female', @EmployeeTotal  
if(@EmployeeTotal is null)  
 Print '@EmployeeTotal is null'  
else  
 Print '@EmployeeTotal is not null'  
  
**You can pass parameters in any order, when you use the parameter names.** Here, we are first passing the OUTPUT parameter and then the input @Gender parameter.  
  
Declare @EmployeeTotal int  
Execute spGetEmployeeCountByGender @EmployeeCount = @EmployeeTotal OUT, @Gender = 'Male'  
Print @EmployeeTotal  
  
**The following system stored procedures, are extremely useful when working procedures.**  
**sp\_help** SP\_Name : View the information about the stored procedure, like parameter names, their datatypes etc. sp\_help can be used with any database object, like tables, views, SP's, triggers etc. Alternatively, you can also press ALT+F1, when the name of the object is highlighted.  
  
**sp\_helptext** SP\_Name : View the Text of the stored procedure  
  
**sp\_depends** SP\_Name : View the dependencies of the stored procedure. This system SP is very useful, especially if you want to check, if there are any stored procedures that are referencing a table that you are abput to drop. sp\_depends can also be used with other database objects like table etc.