**Deyisenler**

DECLARE @num AS INT

SET @num = 250

PRINT @NUM

**Sorgudan gelen datani deyisesene beraber etmek**

SELECT @totalPages =SUM(Books.Pages)

FROM BOOKS

PRINT 'Total pages : ' + CAST(@totalPages AS NVARCHAR(30))

**PRINT ile cumle cixarda bilerik**

**Conditions syntax**

DECLARE @sum1 AS INT = 20, @sum2 AS INT = 30

if @sum1=@sum2

BEGIN

PRINT 'Equality'

END

ELSE IF @sum1>@sum2

BEGIN

PRINT 'sum1 is greater than sum2'

END

ELSE IF @sum1<@sum2

BEGIN

PRINT 'sum2 is greater than sum1'

END

**WHILE**

DECLARE @i AS INT = 0

WHILE @i < 10

BEGIN

PRINT @i

SET @i+=1

END

**2 cur function var**

**SCALAR** Funsiya - yalniz bir data qaytarit(mes. 1, ‘a’, true)

**TableReturn** function - table qaytarir.

**FUNKSIYA DDL emeliyyati ede bilmerik**

**SCALAR FUNCTION**

CREATE FUNCTION ScalarFunction\_Name(@varINT)

RETURNS INT

AS

BEGIN

. . . Some Codes . . .

END

***FUNCTIONLAR Programmability folderinde olur.***

**Calling scalar function** :

*without parametr*

EXECUTE ScalarFunction\_Name

OR  
EXEC ScalarFunction\_Name

*with parametr*

EXECUTE ScalarFunction\_Name parameter

OR  
EXEC ScalarFunction\_Name parameter

**TableReturn Function**

CREATE FUNCTION TableReturnFunction\_Name(@varINT)

RETURNS TABLE

AS

RETURN(

. . . Some Codes . . .

END)

**Calling tablereturn function:**

SELECT \* FROM TableReturnFunction\_Name(parameter)

SELECT \* FROM TableReturnFunction\_Name

**TableReturnFunction Example**

CREATE FUNCTION ShowBooksByAuthorId

(@author\_id AS INT)

RETURNS TABLE

AS

RETURN(

SELECT A.FirstName,B.Name,B.Pages FROM Books AS B

INNER JOIN Authors AS A

ON B.Id\_Author=A.Id

WHERE A.Id=@author\_id

)

SELECT \* FROM ShowBooksByAuthorId(2)