

Kullanıcı tanımlı fonksiyonlar (User Defined Functions)

SQL Server dq üç türlü kullanıcı tanımlı fonksiyon vardır.

Skaler değerli fonksiyonlar, inline tablo değerli fonksiyonlar, birden fazla satır tablo değerli fonksiyonlar.

Skaler değerli fonksiyonlar

Bu fonksiyonların genel yapısı aşağıdaki gibidir.

```
CREATE FUNCTION Fonksiyon_adı(Parametreler)
RETURNS Fonksiyonun_değerinin_tipi
```

```
AS
BEGIN
    Fonksiyonun işlevini programlayan satırlar
    RETURN Fonksiyonun_değeri
END
```

Örnek:

```
CREATE FUNCTION DonemStr(@YIL INT, @DONEM INT)
RETURNS VARCHAR(20)
AS
BEGIN
    DECLARE @DonemStr VARCHAR(20)
    SET @DonemStr = CAST(@YIL + 1 AS VARCHAR(4))
    SET @DonemStr = '-' + RIGHT(@DonemStr, 2)
    SET @DonemStr = CAST(@YIL AS VARCHAR(4)) + '-' + @DonemStr
    IF @DONEM = 1
        SET @DonemStr = @DonemStr + ' Güz'
    ELSE
        SET @DonemStr = @DonemStr + ' Bahar'
    RETURN @DonemStr
END
```

Yukarıdaki fonksiyonu MT487 veri tabanında

```
SELECT SIRANO, dbo.DonemStr(YIL, Donem) AS Dönem, BasNot FROM KUTUK
```

Yazarak deneyelim. Şimdi iki tamsayıdan büyük olanını veren bir fonksiyon yazalım.

```
CREATE FUNCTION Buyuk(@X INT, @Y INT)
RETURNS INT
```

```
AS
```

```
BEGIN
```

```
    IF @X < @Y
```

```
        SET @X = @Y
```

```
    RETURN @X
```

```
END
```

Inline tablo değerli fonksiyonlar

Bu fonksiyonların genel yapısı:

```
CREATE FUNCTION Fonksiyon_adı(Parametreler)
```

```
RETURNS table
```

```
AS
```

```
RETURN (
```

```
    Bir select deyimi    )
```

Şeklindedir.

Bu tür fonksiyonlar daha çok parametre alan bir VIEW olarak düşünülmelidir.

Örneğin tüm il ve ilçeleri veren aşağıdaki VIEW sadece belirli bir il kodu verildiğinde o ile ait ilçeleri veren bir VIEW haline inline tablo değerli bir fonksiyon kullanarak getirilebilir.

```
CREATE VIEW v_IlceBilgi
```

```
AS
```

```
SELECT    IL.ILKOD, IL.ILAD, ILCE.ILCEKOD, ILCE.ILCEAD
```

```
FROM      IL INNER JOIN
```

```
          ILCE ON IL.ILKOD = ILCE.ILKOD
```

```
CREATE FUNCTION fn_IlceBilgi(@ILKOD AS INT)
```

```
RETURNS TABLE
```

```
AS
```

```
    RETURN(SELECT    IL.ILKOD, IL.ILAD, ILCE.ILCEKOD, ILCE.ILCEAD
```

```
    FROM      IL INNER JOIN
```

```
          ILCE ON IL.ILKOD = ILCE.ILKOD
```

```
    WHERE IL.ILKOD = @ILKOD )
```

Yukarıdaki fonksiyonu MT487 veri tabanında deneyelim.

```
SELECT * FROM fn_IlceBilgi(1)
```

Birden fazla satır içeren ve tablo değerli fonksiyonlar

Bu fonksiyonların genel yapısı:

```
CREATE FUNCTION Fonksiyon_adı(Parametreler)
RETURNS @Tablo_adı TABLE(Tablonun tanımı)
```

```
AS
```

```
BEGIN
```

```
    @Tablo_adı isimli tabloyu oluşturan satırlar
```

```
    RETURN
```

```
END
```

Şeklindedir.

Bir örnek verelim.

```
CREATE FUNCTION fn_Ilceler(@ILKOD AS INT)
```

```
RETURNS @fn_Ilceler TABLE( ILKOD INT, ILAD VARCHAR(20), ILCEKOD INT,
ILCEAD VARCHAR(30))
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO @fn_Ilceler(ILKOD,ILAD, ILCEKOD, ILCEAD)
```

```
    SELECT    IL.ILKOD, IL.ILAD, ILCE.ILCEKOD, ILCE.ILCEAD
```

```
    FROM      IL INNER JOIN
```

```
        ILCE ON IL.ILKOD = ILCE.ILKOD
```

```
    WHERE IL.ILKOD = @ILKOD
```

```
    DECLARE @SAY INT
```

```
    SELECT @SAY = COUNT(*) FROM @fn_Ilceler
```

```
    IF @SAY = 0
```

```
    BEGIN
```

```
        SELECT @SAY = COUNT(*) FROM IL
```

```
        WHERE ILKOD = @ILKOD
```

```
        IF @SAY = 0
```

```
            INSERT INTO @fn_Ilceler(ILKOD,ILAD, ILCEKOD, ILCEAD)
```

```
            VALUES(@ILKOD,'İl kodu verilen kod olan bir il yok',0,"")
```

```
        ELSE
```

```
            INSERT INTO @fn_Ilceler(ILKOD,ILAD, ILCEKOD, ILCEAD)
```

```
            SELECT ILKOD, ILAD, 0, ILAD + ' iline bağlı bir ilçe yok'
```

```
            FROM IL
```

```
            WHERE ILKOD = @ILKOD
```

```
    END
```

```
RETURN  
END
```

Bu fonksiyonu MT+(/ içinde aşağıdaki seçme deyimleri ile deneyelim.

```
SELECT * FROM fn_Ilceler(1)  
SELECT * FROM fn_Ilceler(99)  
SELECT * FROM fn_Ilceler(90)
```