

SQL KULLANICI TANIMLI FONKSİYON

Fonksiyonlar tamamen işimizi kolaylaştırmak adına sürekli olarak tekrarladığımız sql sorgularına tek bir noktadan erişmemizi sağlar. Buda bize hızlı bir erişim imkanı, hızlı bir hata kontrol mekanizması, çabuk müdahale, sorgu tekrarlamama gibi imkanları verir.

Sql Serverda çeşitli fonksiyon kavramları vardır;

- 1- Scalar-Valued Function
- 2- Table-Valued Function
- 3- Çoklu ifade ile tablo döndüren fonksiyonlar

Fonksiyonlar istenilen deger tipinde dönüş yapabilir. INT, VARCHAR deger döndürebileceğiniz gibi bir tablo da döndürebilirsiniz.

SQL KULLANICI TANIMLI FONKSİYON

Kullanıcı tanımlı fonksiyonlar SQL2000 ile gelen özelliklerden biridir. Fonksiyonlar tek bir değer veya tablo döndürmek için kullanılır.

Fonksiyonları veri tabanı programlamada nereye koymak gerektiğini anlamak biraz zor olabilir.

Bu nedenle, fonksiyonları view ve stored procedurlerle karşılaştıralım.

- *Dışarıdan parametre alan bir view tanımlamaz. Bu türden ihtiyaçlar için kullanıcı tanımlı fonksiyonlar kullanılır.*
- *Stored procedure'leri bir sorgunun parçası olarak kullanamayız. Bir select ifadesi içinde prosedürler bulunamaz ama kullanıcı tanımlı fonksiyonlar bulunabilir. Kullanıcı tanımlı fonksiyonlar, tablo gibi, view gibi sorgulanabilir.*
- *Bir tek select ifadesi ile view oluşturamadığımız durumda, kullanıcı tanımlı fonksiyon kullanılır.*
- *Sql serverda bulunmayan, örneğin replace() gibi fonksiyonları tanımlamak için kullanıcı tanımlı fonk. kullanılır*

SQL KULLANICI TANIMLI FONKSİYON

1. SCALAR-VALUED FUNCTION

Scaler değerli fonksiyonlar, bir tek değer döndüren fonksiyonlardır. Örnek getdate() bir scaler fonksiyondur, çünkü tek bir değer yani şu anın tarihi ve saatini döndürür.

Bu fonksiyon tanımlanırken integer tipinde bir değer döndürüleceği düşünülerek tanımlanır.

Create komutu ile fonksiyonlar tanımlanır. Aynı fonksiyonu düzenlemek istediğimizde ise create kısmını Alter ile değiştirmemiz yeterli olacaktır.

```
Create Function Fn_ToplamaYap(@sayi1 int,@sayi2 int)
```

```
Returns int
```

```
As
```

```
Begin
```

```
    Declare @toplam int
```

```
    Set @toplam = @sayi1+ @sayi2
```

```
    return @toplam
```

```
End
```

```
-- Çalıştırmak için
```

```
Select Fn_ToplamaYap(5,6)
```

SQL KULLANICI TANIMLI FONKSİYON

1.SCALAR-VALUED FUNCTION

Aynı şekilde bu fonksiyon string yani nvarchar tipinde bir değişkeni döndürmek için yaratılmıştır. Örnek string birleştirme için tanımlanmış olsada benzer amaçlara hizmet eden bir çok fonksiyon tanımlaması kodların çok daha okunabilir olmasını sağlamaktadır.

```
Create Function Fn_Birlestir(@Ad Nvarchar(20),@Soyad Nvarchar(30))
Returns Nvarchar(51)
As
Begin
    Return @Ad + Space(1)+ @Soyad
End
```

```
Çalıştırmak için;
Select dbo.Fn_Birlestir('yselim', 'fatihoglu')
```

SQL KULLANICI TANIMLI FONKSİYON

2. TABLE-VALUED FUNCTION

Tablo Döndüren Fonksiyon(SQL de temp tablo yada deęişken tablo kullanımı bazen performans sorunlarına yol açabilir. Bu durumda Büyük sorgularda kullanılan tablolar tablo döndüren fonksiyonlarla kullanıldığında performansda artmalara yol açabilmektedir.)

Genel ifade:

```
CREATE FUNCTION fonksiyonAdi (varsaParametreler)
RETURNS TABLE
AS
BEGIN
RETURN Select ifadesi
END
```

SQL KULLANICI TANIMLI FONKSİYON

2. TABLE-VALUED FUNCTION(INLINE)

Bu tip fonksiyonlar viewlere çok benzerler. Bir tek select ifadesi içerir. Bu ifadenin sonucunu gösterir. View'den farklı olarak dışarıdan parametre alır.

```
use urunler

create function fn_sipList(@uid tinyint)
returns table
as
return(select * from siparis where siparisId=@uid)

--Test etmek için
select * from dbo.fn_sipList(1)
```

SQL KULLANICI TANIMLI FONKSİYON

3.ALTER FUNCTION

Fonksiyonu düzeltmek için ALTER FUNCTION kullanılır

```
ALTER FUNCTION fnMusteri (@grubu varchar(10))  
RETURNS TABLE  
AS  
RETURN select * from musteriler where  
grubu=@grubu
```

SQL KULLANICI TANIMLI FONKSİYON

4. ÇOKLU İFADE İLE TABLO DÖNDÜREN FONKSİYONLAR (Multi-statement)

Bu türden fonksiyonlar bir öncekine benzer. Farkı dışarıya değer döndüren tablo tanımlanması gereklidir.

Genel ifade

```
CREATE FUNCTION fonksiyonAdi (varsaparametreler)  
RETURNS @deger TABLE (tablo tanımı)
```

```
AS
```

```
BEGIN
```

```
Sql deyimleri
```

```
INSERT INTO @deger selectIfadesi
```

```
INSERT INTO @deger selectIfadesi
```

```
INSERT INTO @deger selectIfadesi ....
```

```
RETURN
```

```
END
```


SQL KULLANICI TANIMLI FONKSİYON

4. ÇOKLU İFADE İLE TABLO DÖNDÜREN FONKSİYONLAR (Multi-statement)

Bu türden fonksiyonlar bir öncekine benzer. Farkı dışarıya değer döndüren tablo tanımlanması gereklidir.

Genel ifade

```
CREATE FUNCTION fonksiyonAdi (varsaparametreler)  
RETURNS @deger TABLE (tablo tanımı)
```

```
AS
```

```
BEGIN
```

```
Sql deyimleri
```

```
INSERT INTO @deger selectIfadesi
```

```
INSERT INTO @deger selectIfadesi
```

```
INSERT INTO @deger selectIfadesi ....
```

```
RETURN
```

```
END
```

SQL KULLANICI TANIMLI FONKSİYON

4. ÇOKLU İFADE İLE TABLO DÖNDÜREN FONKSİYONLAR

```
use urunler
--multi statement

create function fn_sipList1(@uid tinyint)
returns table
as
return select * from siparis where @uid in(1,2)

alter function fn_sipList1(@uid tinyint)
returns table
as
return select * from siparis where @uid in(1,2)

--test etmek için
select * from dbo.fn_sipList1(1)
```

SQL KULLANICI TANIMLI FONKSİYON

5. DROP FUNCTION

Mevcut olan bir fonksiyonu silmek için; **DROP FUNCTION** ifadeleri kullanılır.

SQL KULLANICI TANIMLI FONKSİYON

Örnekler

```
--Round decimal rakamları yuvarlamak için kullanılır.
```

```
SELECT ROUND(123.56, 1)
```

```
--KULLANICI TANIMLI FONKSİYONLAR
```

```
CREATE FUNCTION fnc_adi (@parametre tipi)
```

```
RETURNS dondurecegidegertipi
```

```
AS
```

```
BEGIN
```

```
Return blabla
```

```
END
```

```
--ÖRNEK
```

```
Create Function StringBirlestir(@Isim nVarChar(50), @SoyIsim nVarChar(50))
```

```
Returns nVarChar(100)
```

```
BEGIN
```

```
Declare @Sonuc nVarChar(100)
```

```
Set @Sonuc = @Isim + ' ' + @SoyIsim
```

```
Return @Sonuc
```

```
END
```