



YALOVA ÜNİVERSİTESİ
university of yalova

Öğr. Gör. Cansu AYVAZ GÜVEN

VERİTABANI-II

Fonksiyonlar

Fonksiyonları, View ve Stored Procedurelerle Karşılaştıralım.

- Dışarıdan parametre alan bir view tanımlanamaz. Bu türden ihtiyaçlar için fonksiyon kullanılır.
- Stored procedureleri bir sorgunun parçası olarak kullanamayız. Ya da stored prosedürleri sorgulayamayız. Ancak view ve fonksiyonları sorgulayabiliriz.
- Bir select ifadesi içeren stored prosedürler sorgunun FROM, WHERE, IN gibi kısımlarında kullanılamaz.
- Bir tek select ifadesi ile view oluşturamadığımız zaman fonksiyon kullanılır.

1- Skaler tanımlı (tek değer döndüren) fonksiyonlar

- Skaler değerli fonksiyonlar bir tek değer döndüren fonksiyonlardır. Örnek `getdate()` bir skaler fonksiyondur, çünkü tek bir değer yani şu anın tarih ve saatini döndürür.
- Bazı durumlarda, buna benzer fonksiyonlara ihtiyaç duyulur. Örneğin bir müşterinin sepetinde kaç ürünün bulunduğu, kaç sipariş verdiğini döndüren fonksiyonlar tanımlanabilir.

Genel ifade

CREATE FUNCTION fonksiyonAdi(varsaa
parametreTanimi)

RETURNS geriDonusTipi

AS

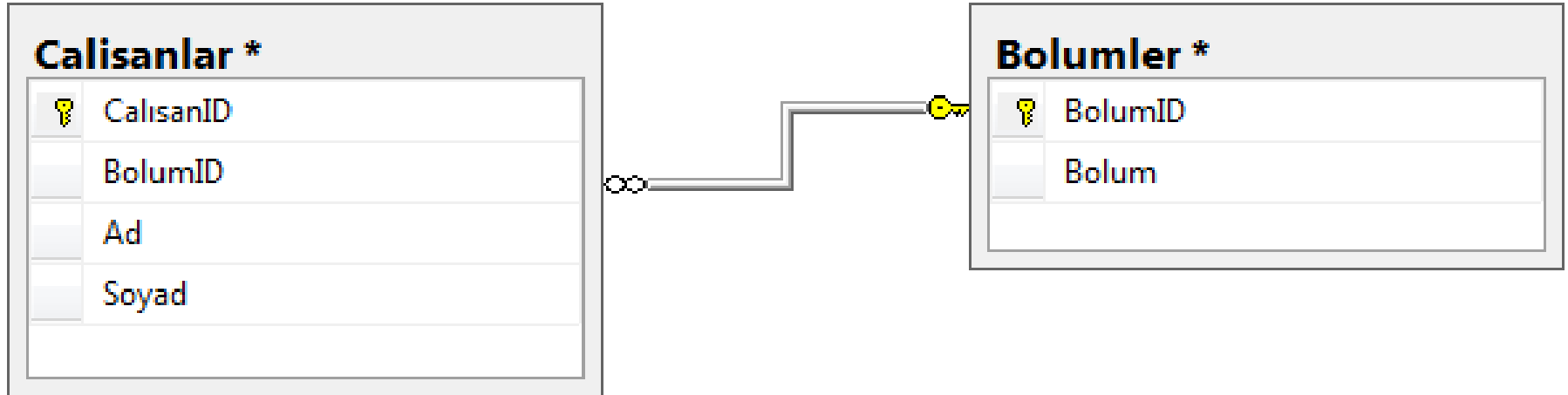
BEGIN

Sql deyimi

RETURN geriDonusDeğeri

END

Örnek Veritabanımız



Örnek

- Bir BölümId girildiğinde, bu bölümde kaç çalışan olduğunu bulan bir fonksiyon yazınız ve bu fonksiyonu çalıştırınız.

```
CREATE FUNCTION bolumCalisanSayisi (@bId int)
RETURNS int
AS
BEGIN
Declare @calisanSayisi int
Set @calisanSayisi = ( SELECT count (CalisanID) FROM Calisanlar group by BolumID
                        Having BolumID=@bId)
RETURN @calisanSayisi
END

Select dbo.bolumCalisanSayisi(6)
```

Örnek:

- bolumCalisanSayisi fonksiyonunu kullanarak, çalışan sayısı 15 den fazla olan bölümleri listeleyiniz.

```
- SELECT * FROM Calisanlar  
  WHERE dbo.bolumCalisanSayisi(BolumID) > 15
```

Örnek

- 1,2 gibi ay numarası gireceğiz. Ocak, şubat gibi ay adını veren fonksiyon yazınız. Ay değeri 12 büyük girildiğinde tanımsız yazsın.


```
CREATE FUNCTION AyAdi (@ay int)
RETURNS nvarchar(20)
AS
BEGIN
RETURN
(
SELECT CASE @ay
WHEN 1 THEN 'Ocak'
WHEN 2 THEN 'Şubat'
WHEN 3 THEN 'Mart'
WHEN 4 THEN 'Nisan'
WHEN 5 THEN 'Mayıs'
WHEN 6 THEN 'Haziran'
WHEN 7 THEN 'Temmuz'
WHEN 8 THEN 'Ağustos'
WHEN 9 THEN 'Eylül'
WHEN 10 THEN 'Ekim'
WHEN 11 THEN 'Kasım'
WHEN 12 THEN 'Aralık'
ELSE 'Tanımlı Değil'
END
)
END
```

```
Select dbo.AyAdi (3)
```

110 %



Results



Messages

	(No column name)
1	Mart

2- Tablo döndüren fonksiyonlar (inline)

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

Genel İfade

```
CREATE FUNCTION fonksiyonAdi (varsaParametreler)
RETURNS TABLE
AS
RETURN selectİfadesi
GO
```

Örnek: Tüm çalışanları listeleyen bir fonksiyon oluşturunuz ve bu fonksiyonu sorgulayınız.

```
- CREATE FUNCTION fn_Calisan ( )  
  RETURNS TABLE  
  AS  
  RETURN select * from Calisanlar  
Go  
  
Select * From fn_Calisan()
```

Tüm çalışanları listeleyen bir stored procedure oluşturunuz ve bu prosedürü sorgulayınız.

```
CREATE PROCEDURE spCalisanlar
AS
SELECT * FROM Calisanlar
Go

EXEC spCalisanlar
```

Tüm çalışanları listeleyen bir view oluşturunuz ve bu viewi sorgulayınız.

```
- CREATE VIEW vwCalisanlar
AS
SELECT * FROM Calisanlar
Go

SELECT * FROM vwCalisanlar
```

Örnek-2

- Oluşturduğunuz view, fonksiyon ve prosedürden faydalanarak 6 nolu bölümdeki çalışanları listeleyiniz.

- vwCalisanlar viewinden 6 nolu bölümdeki çalışanlar
`SELECT * FROM vwCalisanlar WHERE BolumID=6`

- fnCalisanlar fonksiyonunda 6 nolu bölümdeki çalışanlar
`SELECT * FROM fn_Calisan() WHERE BolumID=6`

- spCalisanlar prosedüründen A grubundaki müşteriler.

Bu prosedürden 6 nolu bölümdeki çalışanlar listelenemez. Çünkü prosedürler sorgulanamaz. Yani SELECT deyimi içinde prosedür bulunamaz.

Örnek-3

- Oluşturduğunuz view, fonksiyon ve prosedüre Bno parametresi koyalım ve yeniden oluşturalım

fn_Calisan fonksiyonuna parametre tanımlayalım

```
- ALTER FUNCTION fn_Calisan (@Bno int )  
  RETURNS TABLE  
  AS  
  RETURN select * from Calisanlar where BolumID=@Bno  
Go  
  
--BolumId=3 teki çalışanlar  
  
SELECT * FROM fn_Calisan (3)
```

spCalisan prosedürüne parametre tanımlayalım.

```
ALTER PROCEDURE spCalisanlar
@Bno int
AS
SELECT * FROM Calisanlar WHERE BolumID=@Bno
Go

--BolumId=2 deki çalışanlar

EXEC spCalisanlar 2
```

vwCalisan view'ine parametre tanımlayalım.

□ *Viewe parametre tanımlanamaz.*