

Kullanıcı Tanımlı Fonksiyon Uygulamaları

NOT: `vtyss2_h10_2008-09.sql` isimli dosyayı sql server'da çalıştırınız.

Örnek 1

Bir müşteri no girildiğinde, bu müşterinin sepetinde toplam kaç ürün olduğunu bulan, bir kullanıcı tanımlı fonksiyon oluşturunuz ve bu fonksiyonu çalıştırınız.

```
USE OrnekLtd
GO
CREATE FUNCTION sepetUrunSayi (@mno int)
RETURNS float
AS
BEGIN
    RETURN ( SELECT sum(miktar) FROM
Sepet WHERE mno=@mno)
END
Go
```

Örnek 2

Örnek 1 deki `sepetUrunSayi` fonksiyonunu kullanarak Sepetteki ürün miktarını bulan sorguyu yazınız.

```
USE OrnekLtd
GO
SELECT dbo.sepetUrunSayi (1)

SELECT dbo.sepetUrunSayi (3)
go
```

Örnek 3

Örnek 1 deki fonksiyonu kullanarak, müşteri tablosunu, sepetteki toplam ürün ile birlikte listeleyiniz.

```
USE OrnekLtd
GO
SELECT *, dbo.sepetUrunSayi(mno) as
urunsayi FROM musteriler
go
```

Örnek 4

Örnek 1 deki fonksiyonu kullanarak , sepetinde 10 dan fazla ürün bulunan müşterileri listele.

```
SELECT *, dbo.sepetUrunSayi(mno) as
urunsayi FROM musteriler WHERE
dbo.sepetUrunSayi(mno) >10
go
```

Örnek 5

1,2 gibi ay numarası gireceğiz, Ocak, Şubat gibi ay adını veren bir fonksiyon yazınız. Ay değeri 12 den büyük girilirse Tanımsız yazsın.

```
USE OrnekLtd
GO
CREATE FUNCTION AyAdi (@ay int)
RETURNS varchar (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 'Tanımsız'
        )
    )
END
go
```

Örnek 6

Örnek 5 deki fonksiyonu ile `@ay` parametresini 1 ve 7 olarak sorgulayınız.

```
select dbo.AyAdi (1)

select dbo.AyAdi (7)
```

Örnek 7

Herhangi bir gruba ait müşteri sayısını veren fonksiyon yazınız. (Grup parametre olacak)

```
CREATE FUNCTION MusteriSayisi (@Grup
varchar (10))
RETURNS int
AS
BEGIN
    RETURN (SELECT count(*) FROM
musteriler WHERE Grubu=@Grup)
END
```

Kullanıcı Tanımlı Fonksiyon Uygulamaları

Örnek 8

Örnek 7 deki fonksiyonu kullanarak, A grubundaki ve B grubundaki Müşteri sayısını bulunuz.

```
-- A grubundaki müşteri sayısı
select dbo.MusteriSayisi('A')

-- B grubundaki müşteri sayısı
select dbo.MusteriSayisi('B')
```

Örnek 9

Herhangi bir personelin maaşını veren bir fonksiyon oluşturunuz ve bu fonksiyonu çalıştırınız. Parametre sicilno olacak

```
USE OrnekLtd
GO
CREATE FUNCTION Maas(@sicilno
varchar(10))
returns money
AS
BEGIN
RETURN (SELECT MAAS FROM Personel
WHERE SicilNo=@sicilno)
END
Go
```

Örnek 10

Örnek 9 daki fonksiyonu kullanarak, 2053 nolu personelin maaşını bulunuz.

```
USE OrnekLtd
GO
SELECT dbo.Maas('2053')
```

Örnek11

Tüm müşteri listesini veren vwMusteri adında bir view oluşturunuz.

```
USE OrnekLtd
GO
CREATE VIEW vwMusteri as
SELECT * FROM Musteri
```

Örnek 12

Tüm müşteri listesini veren fonksiyonu oluşturunuz.

```
USE OrnekLtd
GO
```

```
CREATE FUNCTION fnMusteri()
RETURNS TABLE
as
RETURN (SELECT * FROM Musteri)
```

Örnek 13

Tüm müşteri listesini veren spMusteri adında Stored Procedure oluşturunuz.

```
CREATE PROCEDURE spMusteri
AS
SELECT * FROM Musteri
```

Örnek 14

Örnek 11 deki view'i, örnek 12 deki fonksiyonu sorgulayınız ve örnek 13 deki prosedürü çalıştırınız.

```
USE OrnekLtd
GO
-- vwMusteri sorgula
SELECT * FROM vwMusteri

-- fnMusteri sorgula
SELECT * FROM fnMusteri()

-- spMusteri Çalıştır
EXEC spMusteri
```

Örnek 15

Örnek 11 deki view, örnek 12 deki fonksiyon ve örnek 13 deki prosedürü sorgulayarak Grubu A olan müşterileri listeleyiniz

```
USE OrnekLtd
GO
-- vwMusteri sorgula
SELECT * FROM vwMusteri WHERE Grubu='A'

-- fnMusteri sorgula
SELECT * FROM fnMusteri() WHERE
Grubu='A'

-- spMusteri Sorgulanamaz.Bu nedenle bu
procedur ile Grubu A listenemez.
```

Kullanıcı Tanımlı Fonksiyon Uygulamaları

Örnek 16

Örnek 11 deki view, örnek 12 deki fonksiyon ve örnek 13 deki prosedürü Grubu parametre olacak şekilde değiştiriniz.

```
--viewlere parametre tanımlanamaz.

--Grubu parametre tanımlanmış fonksiyon
ALTER FUNCTION fnMusteri (@Grubu
varchar(10))
RETURNS TABLE
AS
RETURN (SELECT * FROM Musteri WHERE
Grubu=@Grubu)
go

--Grubu parametre tanımlanmış prosedür
ALTER PROCEDURE spMusteri @Grubu
varchar(10)
AS
SELECT * FROM Musteri WHERE Grubu=@Grubu
go
```

Örnek 17

Örnek 16daki fonksiyon ve prosedüre parametre değeri olarak A ve C giriniz

```
-- Grubu A olanların Listesi (Fonksiyon
-- sorgulayarak)
SELECT * FROM fnMusteri ('A')

-- Grubu A olanların Listesi
-- Prosedür çalıştırarak
EXEC spMusteri 'A'

-- Grubu C olanların Listesi
-- Fonksiyon sorgularak
SELECT * FROM fnMusteri ('C')

-- Grubu C olanların Listesi
-- Prosedür çalıştırarak
EXEC spMusteri 'C'
```

Örnek 18

mno parametre olacak, sepet tablosunu listeleyen fnSepet adında fonksiyon oluşturunuz.

```
CREATE FUNCTION fnSepet (@mno int)
RETURNS TABLE
AS
RETURN SELECT * FROM Sepet WHERE
mno=@mno
Go
```

Örnek 19

Örnek 18 deki fonksiyon ile Urun tablosunu UrunNo kolonları arasında birleştirerek sorgulayınız. (1 nolu müşteri ve 2 nolu müşteri için)

```
USE OrnekLtd
GO
-- mno değeri 1
SELECT * FROM fnSepet(1) s inner join
Urun u on s.UrunNo=u.UrunNo

-- mno değeri 2
SELECT * FROM fnSepet(2) s inner join
Urun u on s.UrunNo=u.UrunNo
Go
```

Örnek 20

Personel ve Müşteri tablolarını birbirinin devamı olarak birleştiren fonksiyon yazınız. Bu fonksiyonun çıktısında Tipi kolonu olsun, Personel tablosundan gelen için değeri Personel, Müşteri tablosundan gelenin değeri Müşteri olsun.

```
CREATE FUNCTION fnMusteriPersonel ()
RETURNS @deger TABLE (Numara
varchar(10), Adi varchar(20), Soyadi
varchar(20), Tipi varchar(10))
AS
BEGIN
INSERT INTO @deger SELECT
str(mno), Adi, Soyadi, 'Musteri' FROM
Musteri
INSERT INTO @deger SELECT
SicilNo, Ad, Soyad, 'Personel' FROM
Personel

RETURN
END
```

Örnek 21

Örnek 20 deki fonksiyonu sorgulayınız.

```
SELECT * FROM fnMusteriPersonel ()
```