

Trigger ve Stored Procedure Uygulamaları

Not

vtys2_h07_2010-11.sql isimli dosyayı sql server'da çalıştırınız.

Örnek 1

Personel tablosunu listeleyiniz. Bu listede cinsiyet değeri 1 ise Erkek, 2 ise Kız olarak görünsün.

```
use OrnekLtd
go
SELECT *,
CASE cinsiyet
    WHEN 1 THEN 'ERKEK'
    WHEN 2 THEN 'KIZ'
END AS cinsiyetYazi
FROM PERSONEL
```

Örnek 2

Stok miktarı A3 grubunun ortalamasından fazla olanların listesini veren sorguyu yazınız.

A3 grubunun ortalaması aşağıdaki sorgu ile bulunur.

ÇÖZÜM:
SELECT avg(StokMiktar) FROM Urun WHERE Grubu='A3'

Stok Miktarı yukarıdaki sorgudan fazla veya eşit olanları aşağıdaki yere yerleştirelim.

```
SELECT * FROM Urun WHERE StokMiktar>=
(AltSorgu)
```

Alt sorgu yazan yere yukarıdaki sorguyu yazalım.

SONUÇ:
SELECT * FROM Urun WHERE StokMiktar>=
(SELECT avg(StokMiktar) FROM Urun WHERE Grubu='A3')

Örnek 3

Aşağıdaki transaction ve try... catch örneğini ard arda çalıştırınız. (en az 5 kez)

```
DECLARE @Para money, @HesapNo int

select @Para = 200, @HesapNo=12345
BEGIN TRAN
BEGIN TRY
```

```
UPDATE Verdiklerim SET Para =
Para-@Para WHERE HesapNo=@HesapNo
UPDATE Aldiklarim SET Para =
Para+@Para WHERE HesapNo=@HesapNo
COMMIT TRAN
PRINT 'İşlem Başarılı'
END TRY
BEGIN CATCH
ROLLBACK TRAN
PRINT 'İşlem Başarısız'
END CATCH

SELECT * FROM Verdiklerim
SELECT * FROM Aldiklarim
```

Örnek 4

Urun tablosunu Tarih içinde bulunan Ay'ın adını yazdırarak listesini alınız.

```
SELECT *,
CASE month(Tarih)
    WHEN 1 THEN 'OCAK'
    WHEN 2 THEN 'ŞUBAT'
    WHEN 3 THEN 'MART'
    WHEN 4 THEN 'NİSAN'
    WHEN 5 THEN 'MAYIS'
    WHEN 6 THEN 'HAZİRAN'
    WHEN 7 THEN 'TEMMUZ'
    WHEN 8 THEN 'AĞUSTOS'
    WHEN 9 THEN 'EYLÜL'
    WHEN 10 THEN 'EKİM'
    WHEN 11 THEN 'KASIM'
    WHEN 12 THEN 'ARALIK'
END As AyAdi
FROM Urun
```

Örnek 5

Urun tablosunu Mevsim adı ile listeleyiniz.

```
SELECT *,
CASE month(Tarih)
    WHEN 1 THEN 'KIŞ'
    WHEN 2 THEN 'KIŞ'
    WHEN 3 THEN 'İLKBAHAR'
    WHEN 4 THEN 'İLKBAHAR'
    WHEN 5 THEN 'İLKBAHAR'
    WHEN 6 THEN 'YAZ'
    WHEN 7 THEN 'YAZ'
    WHEN 8 THEN 'YAZ'
    WHEN 9 THEN 'SONBAHAR'
    WHEN 10 THEN 'SONBAHAR'
    WHEN 11 THEN 'SONBAHAR'
    WHEN 12 THEN 'KIŞ'
END As AyAdi
FROM Urun
```

Trigger ve Stored Procedure Uygulamaları

Örnek 6

Müşteri tablosundaki tüm verileri satır satır okuyarak yazdırınız.

```
DECLARE @mno int, @Adi varchar(30),
@Soyadi varchar(30), @Grubu char(1)
DECLARE musterCursor CURSOR for SELECT
mno,Adi,Soyadi,Grubu FROM muster

OPEN musterCursor
FETCH NEXT FROM musterCursor INTO @mno,
@Adi, @Soyadi, @Grubu

WHILE @@FETCH_STATUS=0
BEGIN
    SELECT @mno, @Adi, @Soyadi, @Grubu
    FETCH NEXT FROM musterCursor INTO
@mno, @Adi, @Soyadi, @Grubu
END
CLOSE musterCursor
DEALLOCATE musterCursor
```

Örnek 7

Müşteri tablosunun tüm içeriğini listeleyen bir view oluşturunuz.

```
-- istenen kod
CREATE VIEW vwMusteri AS
SELECT * FROM muster
go
```

Örnek 8

Örnek 11 deki view'ı solumayınız.

```
SELECT * FROM vwMusteri
go
```

Örnek 9

Tüm müşteri içeriğini listeleyen Stored Procedure oluşturunuz.

```
CREATE PROCEDURE spMusteri AS
SELECT * FROM muster
go
```

Örnek 10

Örnek 13'de oluşturulan prosedürü çalıştırınız.

```
EXEC spMusteri
```

Örnek 11

mno parametre olacak, verilen mno'lu müşteriyi listeleyen Stored Procedure oluşturunuz.

```
ALTER PROCEDURE spMusteri @mno int AS
SELECT * FROM muster WHERE mno=@mno
Go
```

Örnek 12

Örnek 11 daki prosedürü değişik mno girerek ayrı ayrı çalıştırınız.

```
EXEC spMusteri 2
GO
```

```
EXEC spMusteri 3
GO
```

Örnek 13

Müşteri tablosuna veri ekleyen prosedür yazınız.

```
CREATE PROCEDURE spMusteriEkle
@mno int,
@Adi varchar(30),
@Soyadi varchar(30),
@Grubu char(1)='G'
AS
INSERT muster (mno, Adi, Soyadi, Grubu)
VALUES (@mno, @Adi, @Soyadi, @Grubu)
GO
```

Örnek 14

Örnek 13 deki prosedürü çalıştırınız.

```
EXEC spMusteriEkle 7, 'Ahmet', 'Ak', 'E'
```

```
Go
```

```
EXEC spMusteriEkle @mno=11,@Adi='Alper',
@Soyadi='KILIÇ', @Grubu='Y'
```

```
go
```

```
EXEC spMusteriEkle @mno=13,@Adi='Alper',
@Soyadi='KILIÇ'
```

```
-- Verinin kaydedilginini sinamak için
SELECT * FROM muster
```

Trigger ve Stored Procedure Uygulamaları

Örnek 15

Örnek 13 deki prosürde TRY .. CATCH ile hata kontrolü yapınız ve Hata olduğunda hata var mesajı veriniz.

```
ALTER PROCEDURE spMusteriEkle
    @mno int,
    @Adi varchar(30),
    @Soyadi varchar(30),
    @Grubu char(1)='G'
AS
BEGIN TRY
    INSERT musteriler (mno, Adi, Soyadi,
    Grubu) VALUES (@mno, @Adi, @Soyadi,
    @Grubu)
END TRY
BEGIN CATCH
    PRINT 'Hata var Kayıt Yapılamadı.'
END CATCH
GO
```

Örnek 16

Örnek 13 deki prosedürü çalıştırınız.

```
EXEC spMusteriEkle 7, 'Ahmet', 'Ak', 'E'

Go
EXEC spMusteriEkle @mno=11, @Adi='Alper',
    @Soyadi='KILIÇ', @Grubu='Y'
go
EXEC spMusteriEkle @mno=17,
    @Soyadi='ER', @Adi='Mehmet'

-- Verinin kaydedilmesini sınamak için
SELECT * FROM musteriler
```

Örnek 17

Satis tablosuna veri girildiğinde çalışan bir trigger yazınız. Bu trigger girilen ürün, ürün tablosunda yok ise kaydı iptal etsin.

```
ALTER TRIGGER trgSatisGir
ON Satis
AFTER insert
AS
BEGIN
    -- Değişken tanımlama
    DECLARE @UrunNo varchar(10)

    -- Girilen ürünü değişkene atayalım
    SELECT @UrunNo=UrunNo from inserted
```

```
-- değişken içindeki ürün yok ise
IF NOT EXISTS (select * from Urun where
    UrunNo=@UrunNo)
BEGIN
    PRINT @UrunNo+' kodlu ürün, Urun
    Tablosunda Yok. İşlem iptal edildi'
    ROLLBACK TRAN
END
END
```

Örnek 18

Örnek 17 deki trigger'ın çalışmasını sağlamak için, bir kaç tane farklı Satis tablosuna veri giriniz.

```
INSERT Satis (SipNo, UrunNo,
    Miktar, GirişÇikis) values
    (30, 'G', 11, 0)
go
INSERT Satis (SipNo, UrunNo,
    Miktar, GirişÇikis) values
    (30, 'A', 11, 0)
Go
```

Örnek 19

```
ALTER TRIGGER trgSatisGir
ON Satis
AFTER insert
AS
BEGIN
    -- Değişkenler
    DECLARE @UrunNo varchar(10), @miktar
    float, @GirisCikis tinyint

    -- Değişkenlere bilgiyi aktaralım
    Select @UrunNo=UrunNo, @miktar=miktar,
    @GirisCikis=GirisCikis from inserted

    -- Ürün varmı?
    IF exists(select * from urun where
    urunno=@UrunNo)
    Begin
        -- var
        IF @GirisCikis=0
        Update urun set StokMiktar=StokMiktar+
        @miktar where UrunNo=@UrunNo
        ELSE
        Update urun set StokMiktar = Stokmiktar
        - @miktar where UrunNo=@UrunNo
    End
    ELSE -- Urun Yok
    INSERT URUN(UrunNo, StokMiktar) values
    (@UrunNo, @miktar)
END
```

Trigger ve Stored Procedure Uygulamaları

Örnek 20

Örnek 19 deki trigger'ın çalışmasını sağlamak için bir kaç tane farklı Satis tablosuna veri giriniz.

```
INSERT Satis (SipNo, UrunNo,
Miktar,GirisCikis) values
(30, 'G',11,0)
go
INSERT Satis (SipNo, UrunNo,
Miktar,GirisCikis) values
(30, 'A',11,0)
go

-- Ürün tablosunu listeleyerek.
inceleyiniz.
SELECT * FROM Urun
```

Örnek 21

Fatura tablosunda veri değiştiren bir stored procedure yazınız. ÜrünNo ve mno' su bilinen kayıtlarda miktar ve fiyatını değiştiren ve kaç kayıttın değiştiği bilgisi parametreden geri dönsün.

```
CREATE PROCEDURE spDegistir
-- Giriş parametreleri
    @mno int,
    @UrunNo varchar(10),
    @Miktar float,
    @Fiyat float,
-- Geridönüp parametresi
    @KayitSayisi int OUTPUT
AS
UPDATE Fatura SET miktar=@Miktar,
fiyat=@Fiyat WHERE mno=@mno and
UrunNo=@UrunNo

-- Update edilen kayıt sayısı
SELECT @KayitSayisi=@@ROWCOUNT
GO
```

Örnek 22

Örnek 13 deki prosedürü çalıştırınız.

```
DECLARE @Sayi int

-- @Sayi prosedürden geri dönen değer
EXEC spDegistir 1, 'A',10,1,@Sayi OUTPUT

-- Geri dönen değerın yazılması
SELECT @Sayi

SELECT * FROM Fatura
```

Örnek 23

Satis Tablosunda olup, Urun tablosunda bulunmayan ürünleri veren sql kodunu yazınız.

```
-- Satis tablosundaki tüm ürün
-- numaralarına erişmek için kursor
-- tanımlayalım.
DECLARE SatisKursor CURSOR FOR SELECT
UrunNo FROM Satis

-- Kursorde okuduğumuz değeri değikene
-- Aktarmak için değişken tanımlı
DECLARE @UrunNo char(10)

-- Kursorü aç
OPEN SatisKursor

-- İlk pozisyona git ve bulunulan
-- Pozisyondaki değeri @UrunNo ya
-- aktar
FETCH NEXT FROM SatisKursor INTO @UrunNo

-- Her satıra tek tek ulaş
WHILE @@FETCH_STATUS=0
BEGIN

-- Kursorde okunan @UrunNo ürün
-- tablosunda var mı?
IF NOT EXISTS (SELECT * FROM Urun WHERE
UrunNo = @UrunNo)
    PRINT @UrunNo + ' Kodlu Ürün Urun
                        Tablosunda Yok'

FETCH NEXT FROM SatisKursor INTO @UrunNo
END

CLOSE SatisKursor
DEALLOCATE SatisKursor
GO
```