

UPDATE() Fonksiyonu

Kolondaki verinin değişip değişmediği kontrol edilir. Kolondaki veri değişmiş ise değer true (doğru), değişmemiş ise değeri false (yanlış) tır.

Örnek:

Miktar kolonundaki veri değiştiğinde trigger içinde yazılacak kod aşağıdadır.

```
IF UPDATE(miktar)
BEGIN
    ... sql kodları
END
```

Örnek:

Satış tablosunda veri değiştiğinde çalışacak bir trigger yazınız. Miktar kolonunda veri değiştiğinde, değişen ürünü, ürün tablosuna ekleyen bir trigger yazınız.

Çözüm:

İhtiyaçlar

Miktar,
UrunNo

-- Değişken tanımlayacağız.

```
DECLARE @miktar float, @UrunNo  
varchar(10)
```

-- Değişen kayıtlardaki miktar ve ürünno bilgisini, tanımladığımız değişkenlere akatacağız.

```
Select @miktar=miktar,  
@UrunNo=UrunNo from inserted
```

-- miktar kolonundaki veri değişti mi?

```
IF UPDATE(miktar)  
BEGIN  
    PRINT 'Miktar kolonundaki veri  
değişti'  
    INSERT urun(UrunNo, StokMiktar)  
values (@UrunNo, @miktar)  
END  
ELSE
```

PRINT 'Miktar kolonundaki veri
değişmedi'

Toplu olarak yazalım.

```
CREATE TRIGGER trgSatisDegistir
ON Satis
AFTER update
As
BEGIN
    -- değişken tanımla
    DECLARE @UrunNo varchar(10),
    @miktar float
    -- değişen kayıttaki bilgiyi değişkene
    aktar
    Select @UrunNo=UrunNo,
    @miktar=miktar from inserted
    -- miktar kolonu değişti mi?
    IF UPDATE (miktar)
    BEGIN
        PRINT 'Miktar kolonundaki veri
değişti'
```

```
insert urun (UrunNo, StokMiktar)
values (@UrunNo, @miktar)
END
ELSE
Print 'Miktar kolonunda veri
değişmedi'
END
Go
```

Yukarıdaki triggerın çalışmasını sağlamak için aşağıdaki kodu çalıştırınız.

```
UPDATE Satis SET GirişÇıkış=1,
Miktar=1200 where UrunNo='EH'
```

Soru :

Satis tablosunda olup, ürün tablosunda bulunmayan ürünleri veren sorguyu yazınız. (Küresörlerden faydalanarak)

Çözüm:

```
-- Bulunulan posisyondaki UrunNo
değerini aktarmak için @UrunNo
değişkenini tanımlayalım
DECLARE @UrunNo varchar(10)

-- Her satırı okumak için kursor tanımla
DECLARE SatisKursor CURSOR FOR
select UrunNo from Satis

-- kursorü aç
OPEN SatisKursor

-- ilk kayıda konumlan
-- bulunulan satıraki bilgiyi @UrunNo
değişkenine ata
FETCH NEXT FROM SatisKursor INTO
@UrunNo

WHILE @@fetch_status = 0
BEGIN
    IF NOT EXISTS (select * from Urun
where UrunNo=@UrunNo)
```

Print @UrunNo + ' Urun tablosunda
yok '

```
-- bir sonraki kayıta git  
Fetch next from SatisKursor into  
@UrunNo  
END  
Close SatisKursor  
Deallocate SatisKursor
```

Soru :

Fatura tablosunda veri değiştiren bir stored procedure yazınız. ÜrünNo ve mno'su bilinen kayıtlarda miktar ve fiyatı değiştiren ve kaç kayıttın değiştiği bilgisini parametreden geri döndüren bir prosedür olsun.

Çözüm:

Giriş parametreleri

Parametreler:

UrunNo, mno

Miktar, fiyat
Geri dönüş parametresi
KayıtSayisi

```
CREATE PROCEDURE spDegistir
@UrunNo varchar(10),
@mno int,
@miktar float,
@fiyat Money,
-- Geri dönüş parametresi
@KayıtSayisi int OUTPUT
AS
UPDATE Fatura SET miktar=@miktar,
fiyat=@fiyat WHERE UrunNo=@UrunNo
and mno=@mno

SELECT @KayıtSayisi = @@rowcount
Go
```

Yukarıdaki prosedürü aşağıdaki kod ile çalıştıralım.

```
DECLARE @KayitSayisi int
EXEC spDegistir @miktar=1200,
@fiyat=11520,
@KayitSayisi=@KayitSayisi OUTPUT,
@UrunNo='A',@mno=1
```

```
SELECT @KayitSayisi as KayitSayi
```