EXISTS ve NOT EXISTS fonksiyonu

DECLARE @sipno int

Select @sipno=1

IF EXISTS (select \* from satis where sipno=@sipno)

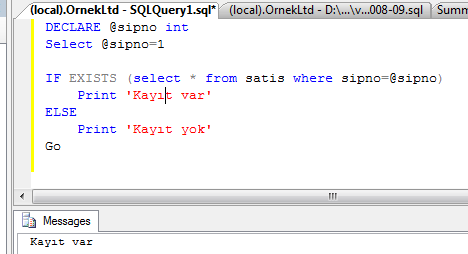
Print ‘Kayıt var’

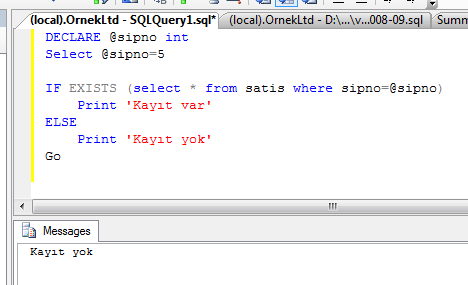
ELSE

Print ‘Kayıt yok’

Go

Yukarıdaki kodu çalıştıralım.





DECLARE @sipno int

Select @sipno=1

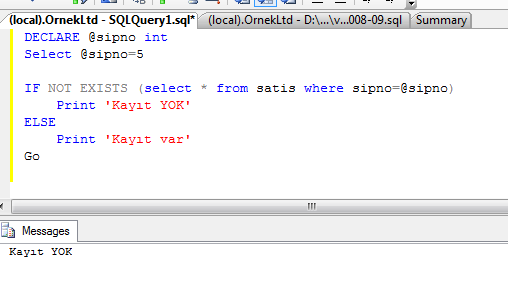
IF NOT EXISTS (select \* from satis where sipno=@sipno)

Print ‘Kayıt yok’

ELSE

Print ‘Kayıt var’

Go



Trigger Örneği

Satış tablosuna veri girildiğinde çalışan ve veri girildi mesajı veren trigger oluşturunuz. Satış tablosuna veri girerek bu trigger’ın çalışmasını sağlayınız.

CREATE TRIGGER trg\_satis\_gir

ON satis

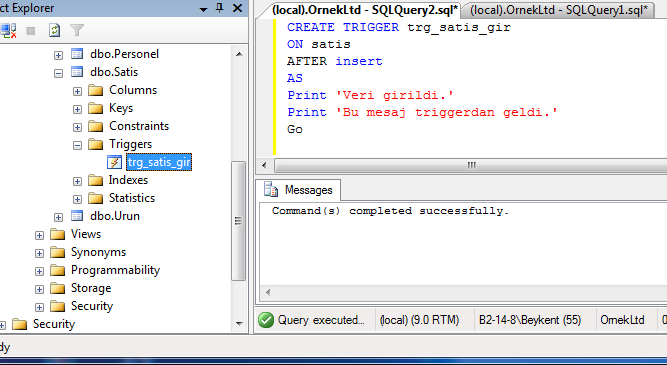
AFTER insert

AS

Print ‘Veri girildi.’

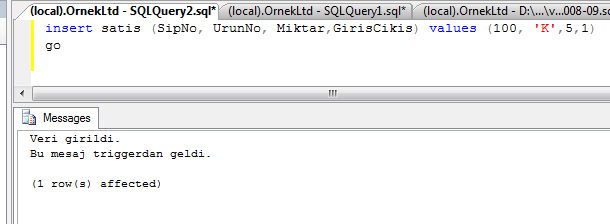
Print ‘Bu mesaj triggerdan geldi.’

Go



insert satis (SipNo, UrunNo, Miktar,GirisCikis) values (100, ‘K’,5,1)

go



Yukarıdaki trigger’ı değiştirelim. Girilen verileri bir değişkene atayalım ve bu değişkenin değerini yazdıralım.

ALTER TRIGGER trg\_satis\_gir

ON satis

AFTER insert

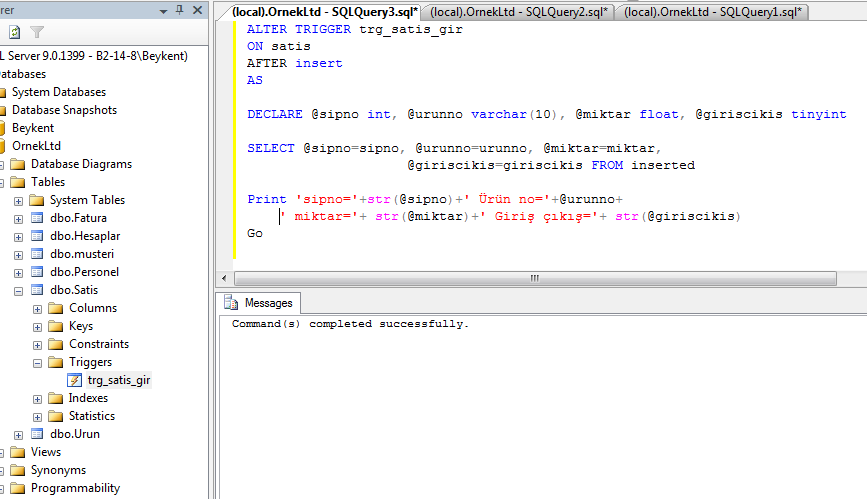
AS

DECLARE @sipno int, @urunno varchar(10), @miktar float, @giriscikis tinyint

SELECT @sipno=sipno, @urunno=urunno, @miktar=miktar, @giriscikis=giriscikis FROM inserted

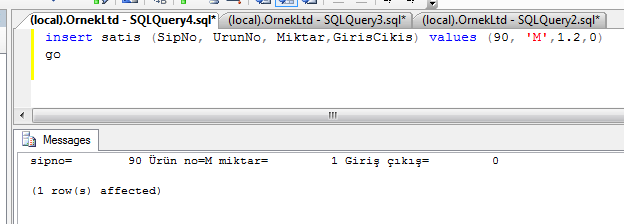
Print ‘sipno=’+str(@sipno)+  
’ Ürün no=‘+@urunno+’ miktar=’+ str(@miktar)+’ Giriş çıkış=‘+ str(@giriscikis)

Go



insert satis (SipNo, UrunNo, Miktar,GirisCikis) values (90, ‘M’,1.2,0)

go



Yukarıdaki trigger’ı değiştireceğiz. Bu trigger girilen ürün, ürün tablosunda yok ise girilen satis kaydını iptal edecek.

ALTER TRIGGER trg\_satis\_gir

ON satis

AFTER insert

AS

-- girilen verideki ürün kodunu bulalım.

-- ürün kodu tanımını yapalım.

Declare @urunno varchar(10)

-- bu değişkene girilen ürünno değerini atayalım

Select @urunno = urunno from inserted

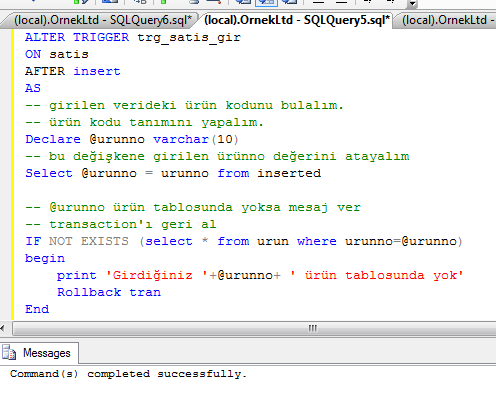
IF NOT EXISTS (select \* from urun where urunno=@urunno)

begin

print ‘Girdiğiniz ’+@urunno+ ‘ ürün tablosunda yok’

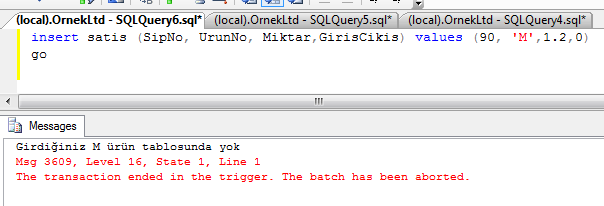
Rollback tran

End

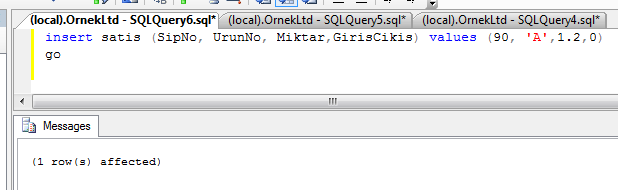


insert satis (SipNo, UrunNo, Miktar,GirisCikis) values (90, ‘M’,1.2,0)

go



Yukarıdaki veri girişini inceleyelim. Biz M kodlu ürünü satis tablosuna girmek istedik. Trigger bu M kodlu ürünün, ürün tablosunda olup olmadığına baktı. Olmadığı için, bize mesaj verdi. Kayıt gerçekleşmedi.



Yukarıdaki veri girişinde A kodlu ürün var mı diye ürün tablosuna sordu. Var olduğu için kayıt gerçekleşti.

Örnek:

Satış tablosuna veri girildiğinde çalışan bir trigger yazınız. Bu trigger girilen ürün, ürün tablosunda yok ise, ürün tablosuna ekleyecektir. Var ise giriş çıkış değerine bağlı olarak ürün miktarını artırıp, azaltacaktır. Giriş çıkış değeri 0 ise artsın, değilse azalsın.

ALTER TRIGGER trg\_satis\_gir

ON satis

AFTER insert

AS

-- değişken tanımlarını

Declare @urunno varchar(10), @miktar float, @giriscikis tinyint

Select @urunno= urunno, @miktar=miktar, @giriscikis=giriscikis FROM inserted

-- girilen ürün ürün tablosunda var mı?

IF not exists(select \* from urun where urunno =@urunno)

-- ürün tablosunda yok, yeni kayıt ekle

insert urun (urunno, stokmiktar) values (@urunno, @miktar)

ELSE

BEGIN

IF @giriscikis=0

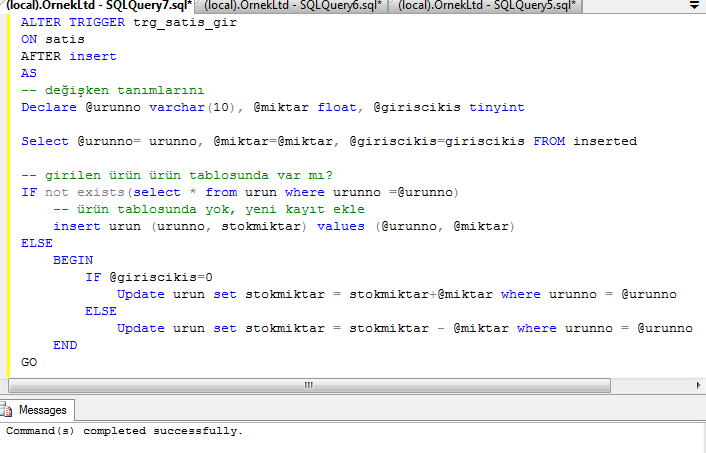
Update urun set stokmiktar = stokmiktar+@miktar where urunno = @urunno

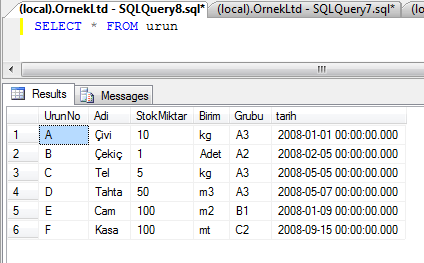
ELSE

Update urun set stokmiktar = stokmiktar - @miktar where urunno = @urunno

END

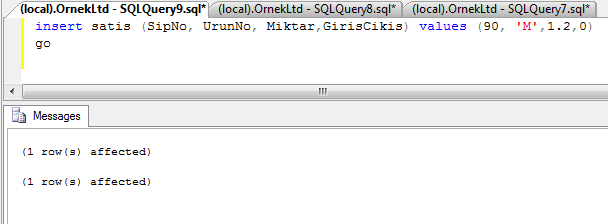
GO



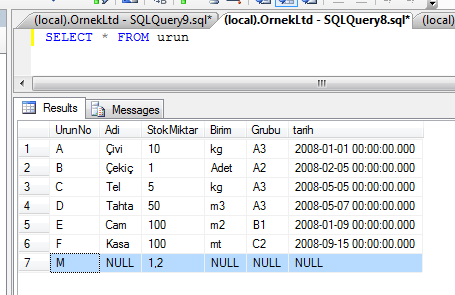


insert satis (SipNo, UrunNo, Miktar,GirisCikis) values (90, ‘M’,1.2,0)

go

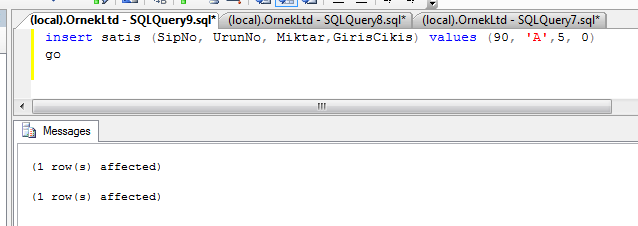


M kodlu ürün, ürün tablosunda yok. Bu nedenle trigger ürün tablosuna kaydı ekledi.

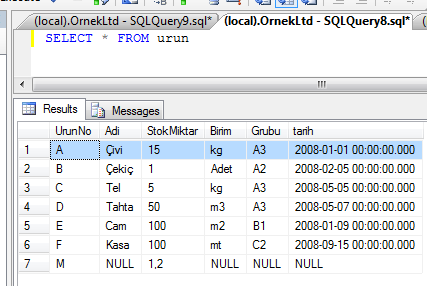


insert satis (SipNo, UrunNo, Miktar,GirisCikis) values (90, ‘A’,5, 0)

go

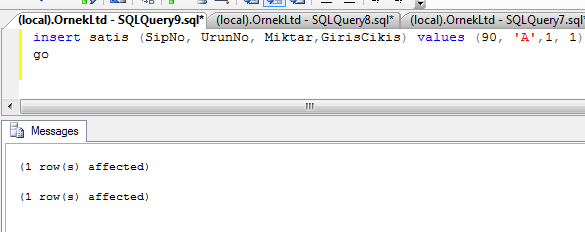


A kodlu ürün ürün tablosunda var, trigger girilen giriscikis değeri 0 olduğu için miktar A’nın miktarını artırdı.

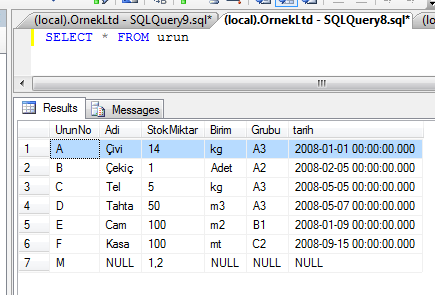


insert satis (SipNo, UrunNo, Miktar,GirisCikis) values (90, ‘A’,1, 1)

go



A kodlu ürün ürün tablosunda var, trigger girilen giriscikis değeri 1 olduğu için miktar A’nın miktarını azalttı.



Stored Prosedür örneği

Örnek:

Fatura tablosunda veri değiştiren bir stored procedure yazınız. Bu prosedür kaç kaydın güncellendiğini parametreden geri döndürsün

CREATE PROC sp\_futura\_guncelle

-- Giriş parametreleri

@mno int,

@urunno varchar(10),

@miktar float,

@fiyat Money,

-- çıkış parametresi

@kayitsayisi int OUTPUT

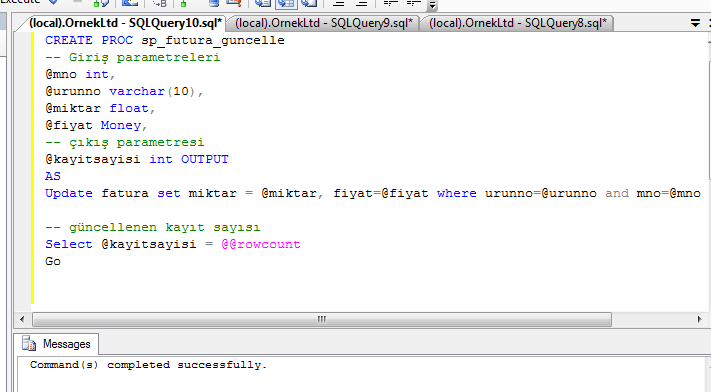
AS

Update fatura set miktar = @miktar, fiyat=@fiyat where urunno=@urunno and mno=@mno

-- güncellenen kayıt sayısı

Select @kayitsayisi = @@rowcount

Go

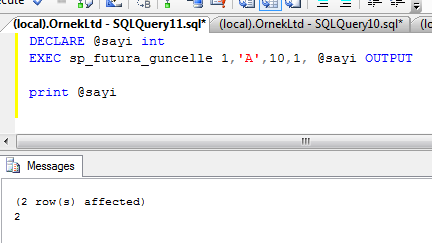


Yukarıdaki prosedürü çalıştıralım. Output parametresi için değişken tanımlayalım

DECLARE @sayi int

EXEC sp\_futura\_guncelle 1,’A’,10,1, @sayi OUTPUT

Select @sayi



Yukarıdaki prosedürü parametre isimleri ile de çalıştırabiliriz.

DECLARE @sayi int

EXEC sp\_futura\_guncelle @mno =1, @urunno=’A’, @miktar =10, @fiyat =1, @kayitsayisi=@sayi OUTPUT

Select @sayi

