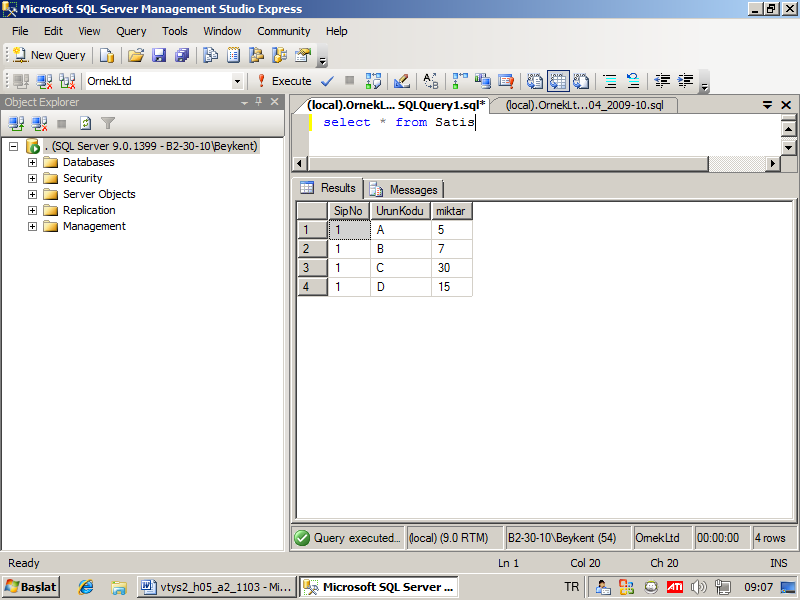
STORED PROCEDURE (özet)

Örnek:

Satış tablosuna veri giren bir stored prosedür oluşturunuz.



CREATE PROC sp\_satis\_veri\_gir

@sipno int,

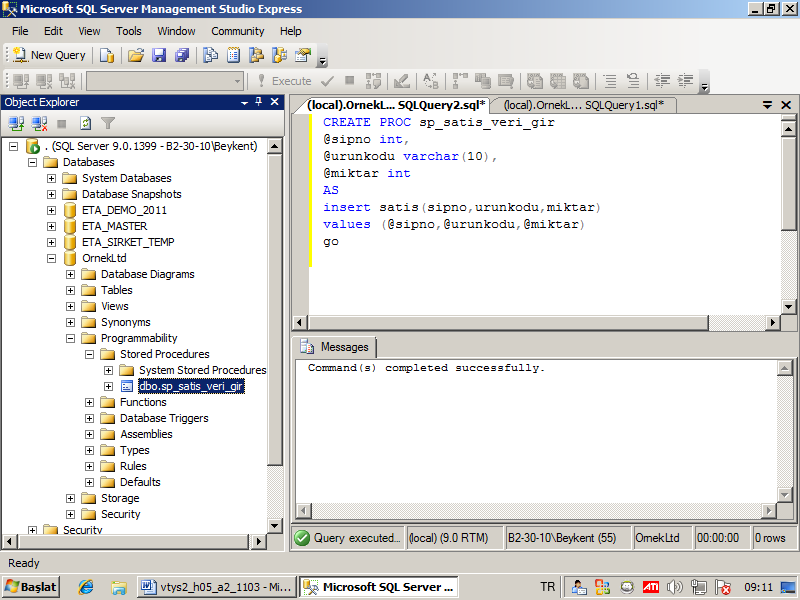
@urunkodu varchar(10),

@miktar int

AS

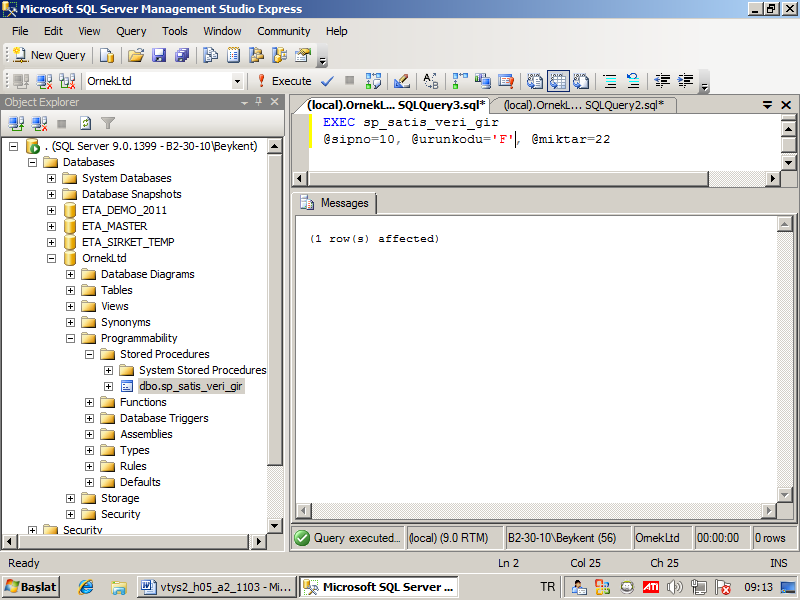
insert satis(sipno,urunkodu,miktar) values (@sipno,@urunkodu,@miktar)

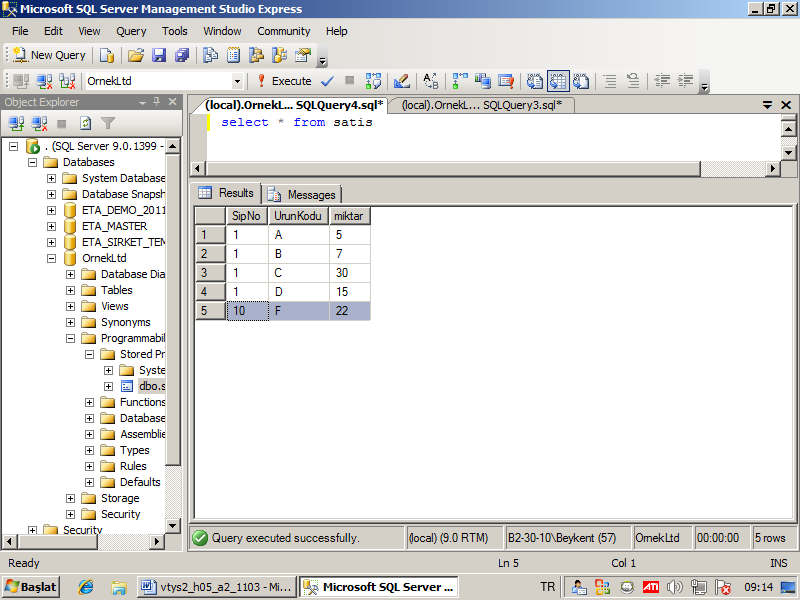
go



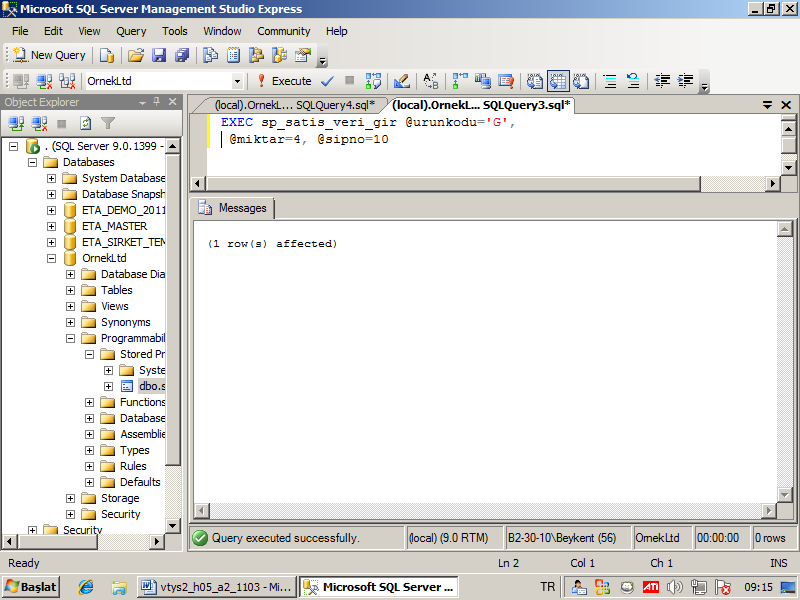
Yukarıdaki prosedürü kullanarak satis tablosuna veri giriniz.

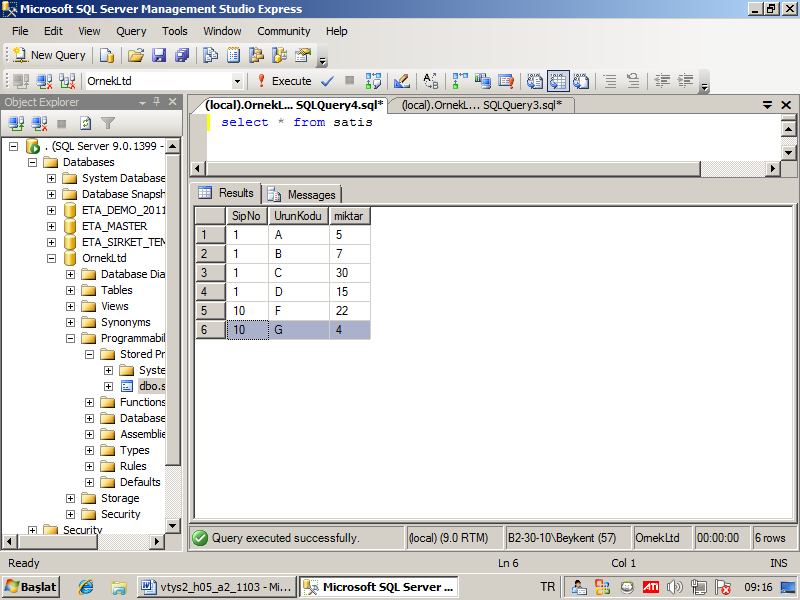
EXEC sp\_satis\_veri\_gir @sipno=10, @urunkodu=’F’, @miktar=22





EXEC sp\_satis\_veri\_gir @urunkodu=’G’, @miktar=4, @sipno=10





Örnek:

Satış tablosunda Sipno ve urun kodu verilecek ve miktar değerini güncelleyen bir prosedür oluşturunuz.

CREATE PROC sp\_satis\_guncelle

@sipno int = null,

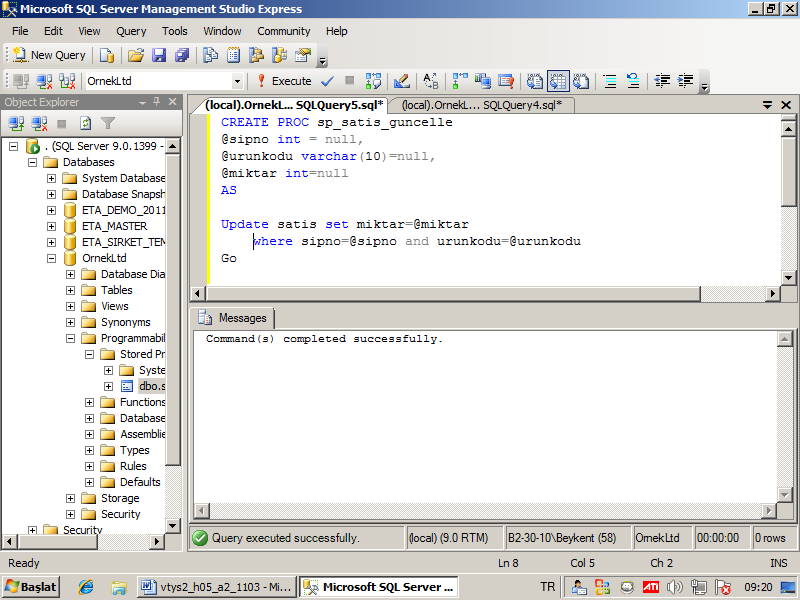
@urunkodu varchar(10)=null,

@miktar int=null

AS

Update satis set miktar=@miktar where sipno=@sipno and urunkodu=@urunkodu

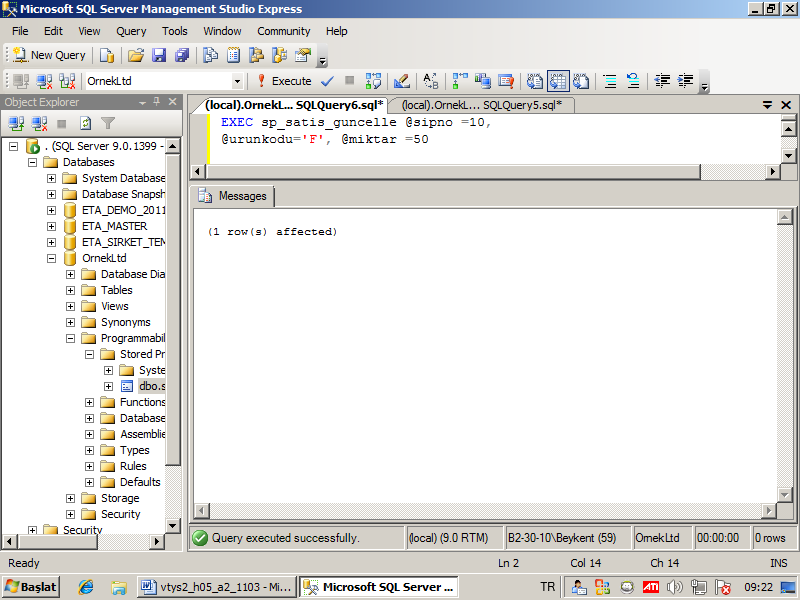
Go

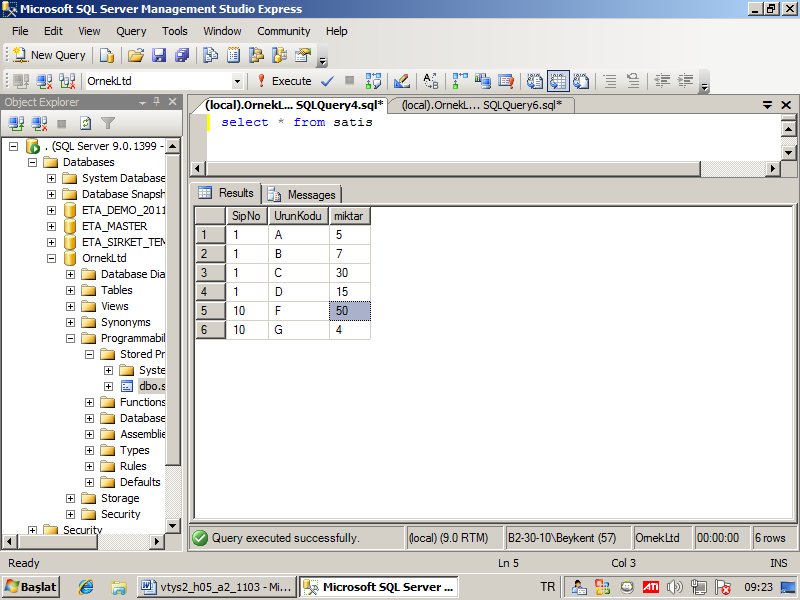


10 nolu satistaki F ürününün miktarını 50 yapınız.

EXEC sp\_satis\_guncelle @sipno =10,

@urunkodu=’F’, @miktar =50





EXISTS ve NOT EXISTS fonksiyonları

EXISTS(sorgu) : Eğer sorgudan bir yada daha fazla kayıt dönerse değer true, dönmezse değer false olur. IF deyimi içinde kullanılır.

NOT EXISTS (sorgu) : exists() fonksiyonun olumsuzudur.

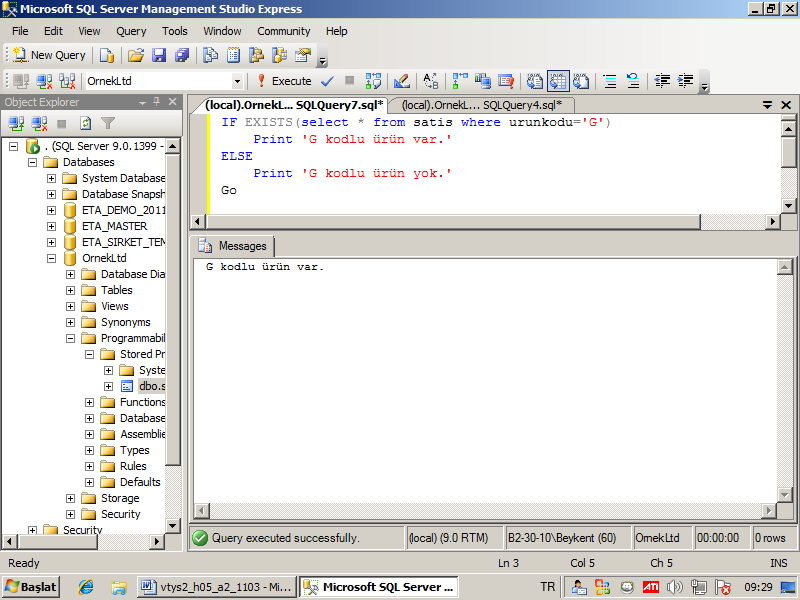
IF EXISTS(select \* from satis where urunkodu=’G’)

Print ‘G kodlu ürün var.’

ELSE

Print ‘G kodlu ürün yok.’

Go



Örnek:

Satis tablosuna duruma göre veri giren veya güncelleyen bir prosedür oluşturunuz. Ürün varsa güncellesin, yoksa yeni eklesin

CREATE PROC sp\_verigir\_guncelle

@sipno int=null,

@urunkodu varchar(10)=null,

@miktar int = null

AS

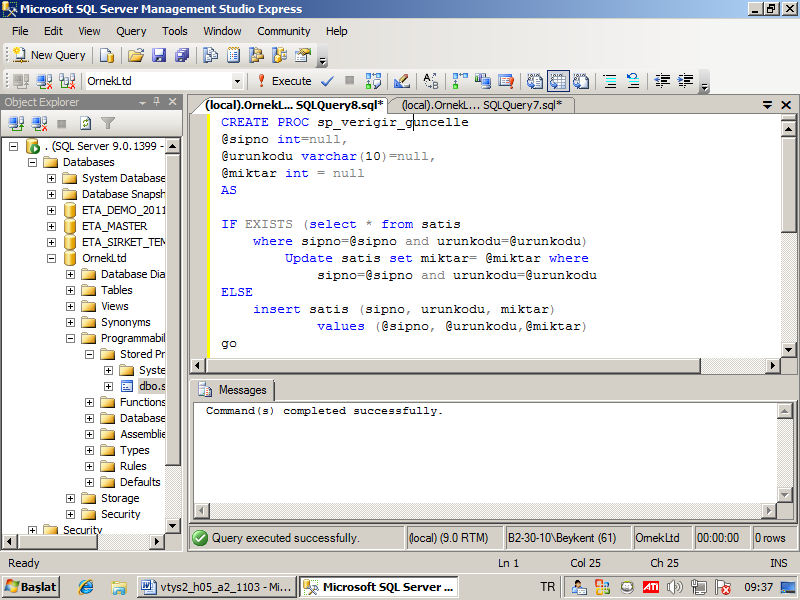
IF EXISTS (select \* from satis where sipno=@sipno and urunkodu=@urunkodu)

Update satis set miktar= @miktar where sipno=@sipno and urunkodu=@urunkodu

ELSE

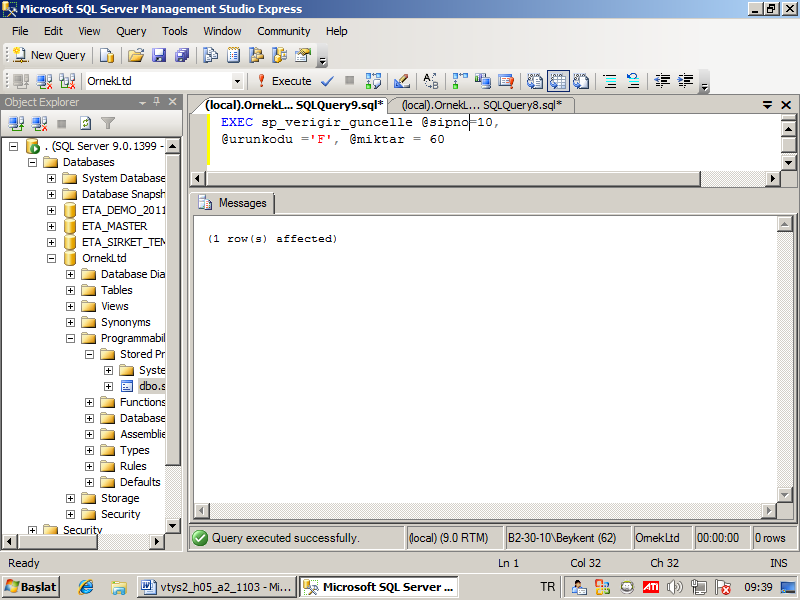
insert satis (sipno, urunkodu, miktar) values (@sipno, @urunkodu,@miktar)

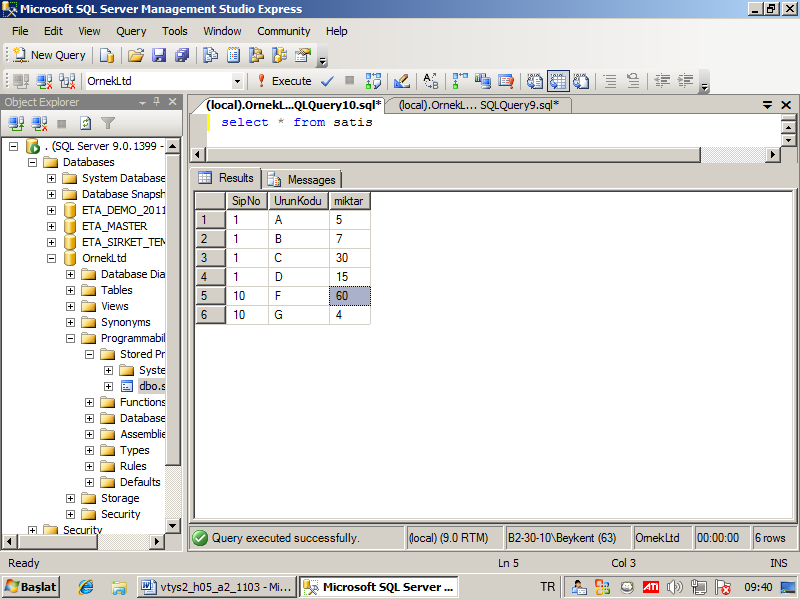
go



EXEC sp\_verigir\_guncelle @sipno=10,

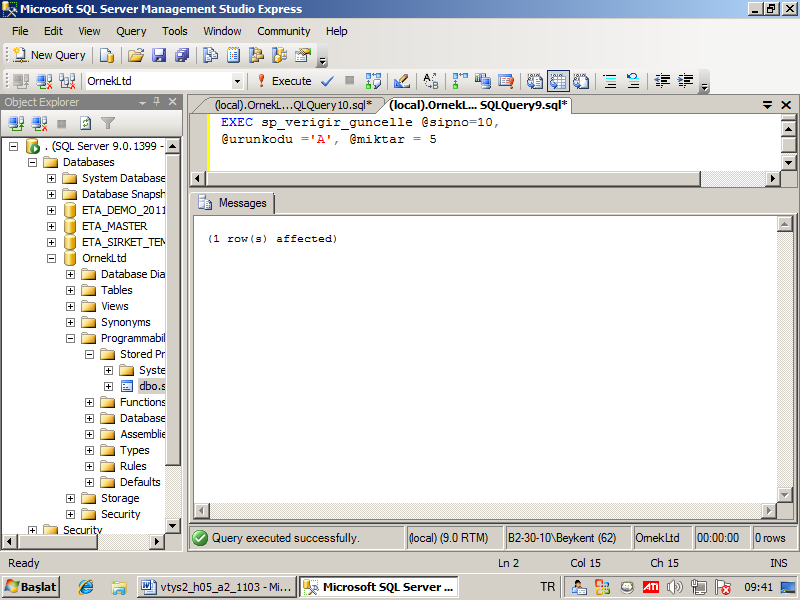
@urunkodu =’F’, @miktar = 60

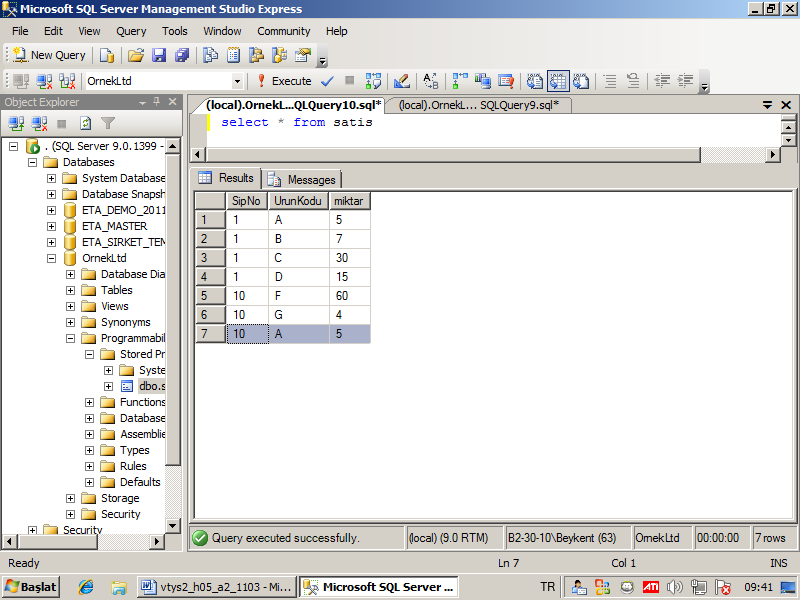




EXEC sp\_verigir\_guncelle @sipno=10,

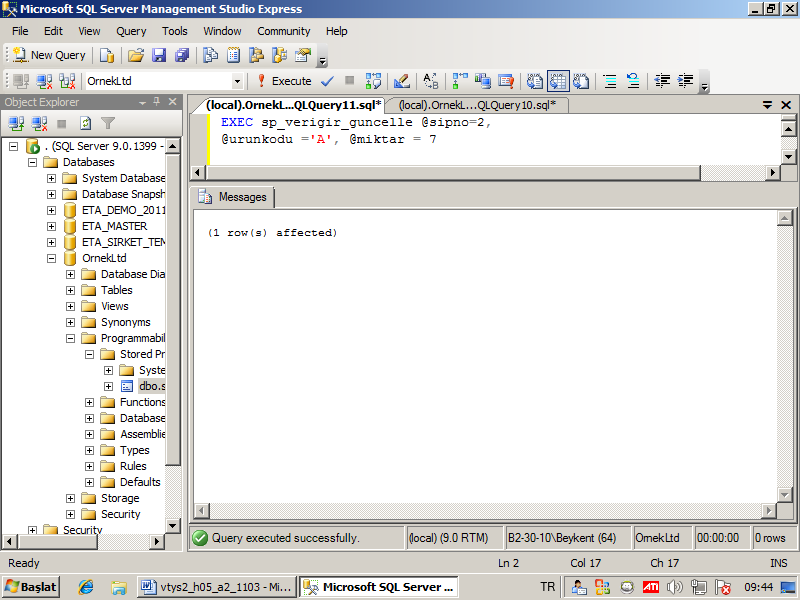
@urunkodu =’A’, @miktar = 5

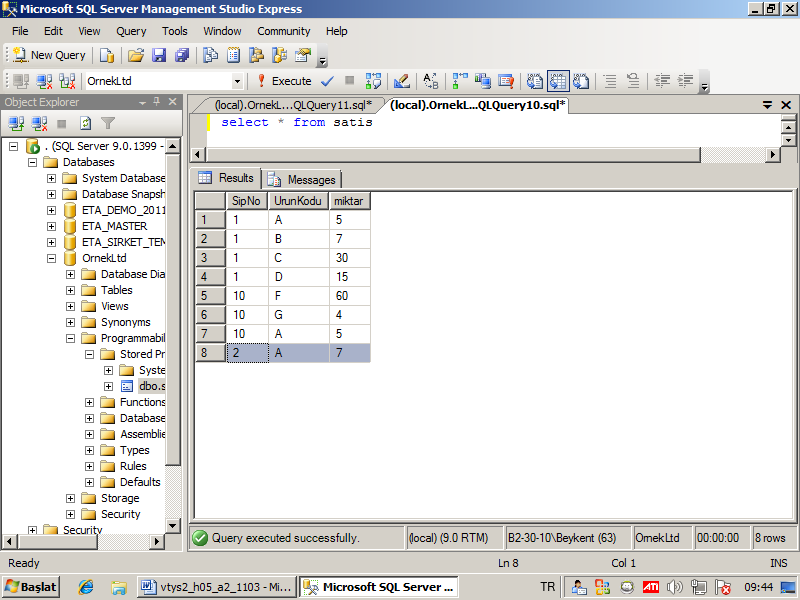




EXEC sp\_verigir\_guncelle @sipno=2,

@urunkodu =’A’, @miktar = 7





TRIGGER

Bir çeşit prosedürdür. Bir olay neticesinde kendiliğinden çalışır. Trigger’lar tablo üzerinde oluşturulur. Trigger’ı tetikleyen olaylar; veri girilmesi, güncellenmesi veya silinmesidir.

Herhangi bir tablo üzerinde yapılan insert, update veya delete işlemi sonrasında otomatik olarak aynı tablo veya başka bir tabloya veri ekleme yapmak veya güncelleme yapmak veya silmek istendiğinde trigger kullanılır.

Trigger’larla girilen veriler kontrol edilebilir. Uygun olmayan veri reddedilebilir.

Trigger çeşitleri

* after trigger : insert, update veya delete olayı gerçekleştiğinde çalışan triggerdır.
* İnstead of trigger : olay gerçekleşmeden önce çalışan triggerdir.

Çoğunlukla after trigger kullanılır ve biz buna odaklanacağız.

Genel ifade

CREATE TRIGGER triggerAdı

ON tabloAdı

AFTER olay

AS

…

Sql deyimi

….

Go

Olaylar : insert, update veya delete olaylarından biri yada bir kaçı yazılabilir.

Örnek:

Satış tablosuna veri girildiğinde çalışan “Yeni veri eklendi” mesajı veren bir trigger yazınız. Tabloya veri girerek trigger’ın çalıştığını görünüz.

CREATE TRIGGER trg\_veri\_gir

ON satis

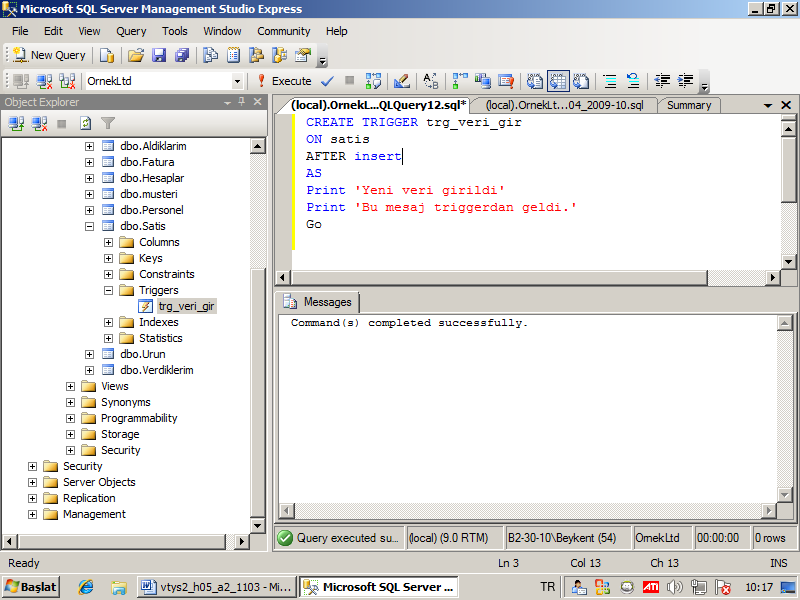
AFTER insert

AS

Print ‘Yeni veri girildi’

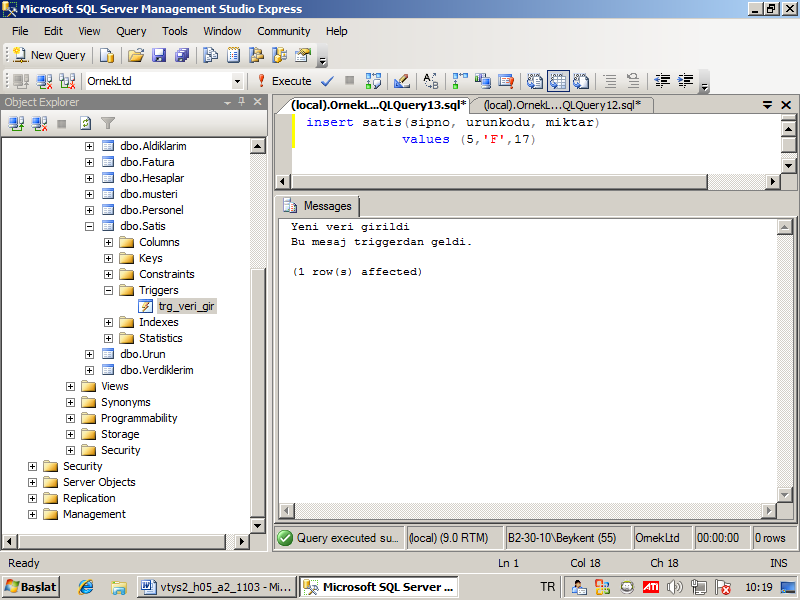
Print ‘Bu mesaj triggerdan geldi.’

Go



Veri girelim

insert satis(sipno, urunkodu, miktar) values (5,’F’,17)



SON EKLENEN, DEĞİŞEN VEYA SİLİNEN kayıtların içindeki bilgilere erişmek:

Trigger son yapılan işlemden etkilenen kayıtlardaki bilgilere ulaşmak için iki tane sanal tablo kullanır. Bu sanal tablolar sadece trigger içinde kullanılır.

* inserted tablosu
* deleted tablosu

inserted tablosunda hangi bilgiler bulunur?

* insert deyimi çalıştığında girilen veri
* update deyimi çalıştığında; güncellenen yeni değer bulunur.

deleted tablosunda hangi bilgiler bukunur?

* Silinen kayıt bulunur.
* Update deyimi çalıştığında , güncelleme öncesi değer bulunur.

Bu tabloları listeleyen bir trigger yazalım.

CREATE TRIGGER trg\_satis

ON satis

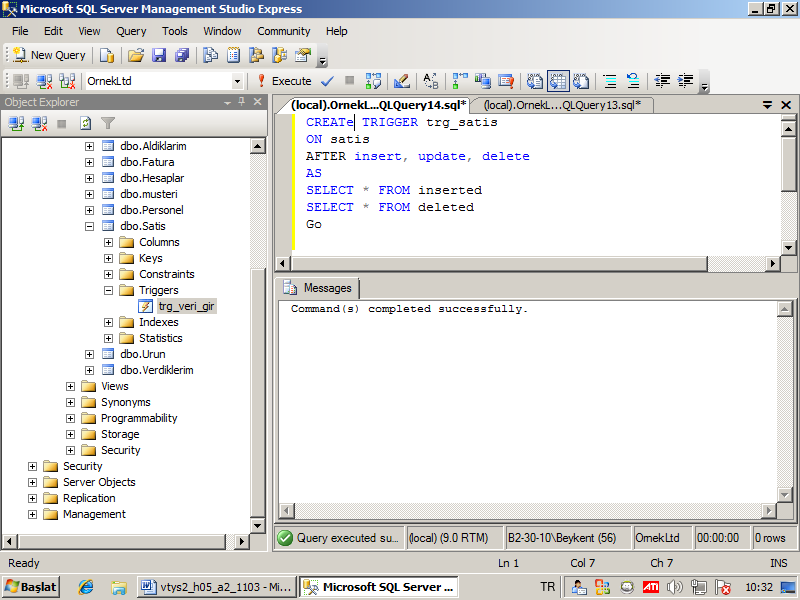
AFTER insert, update, delete

AS

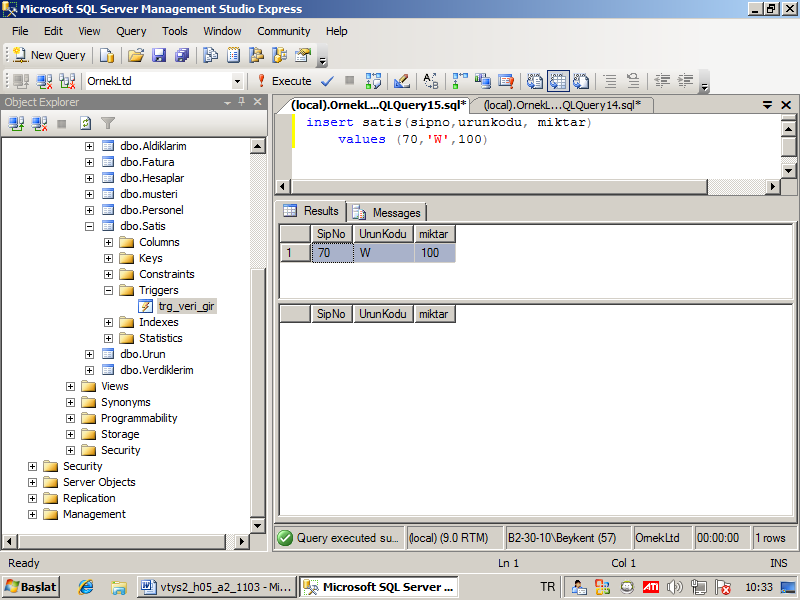
SELECT \* FROM inserted

SELECT \* FROM deleted

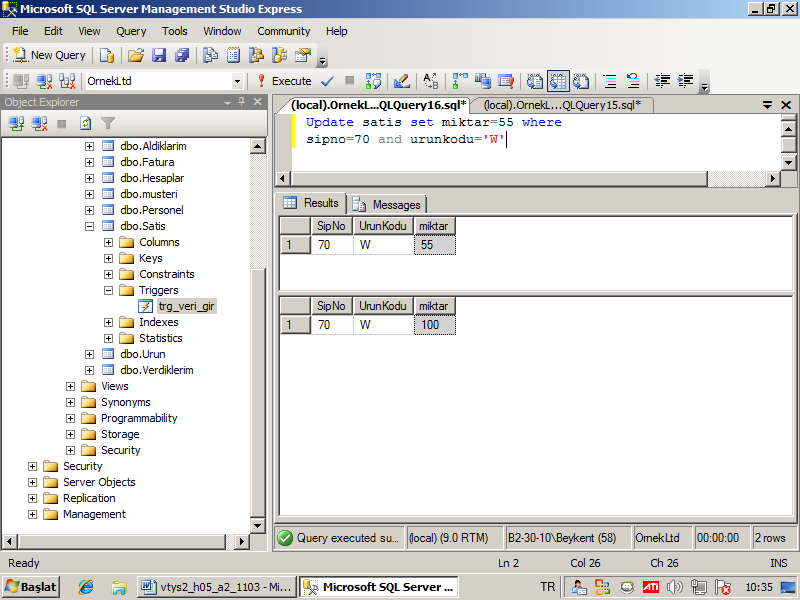
Go



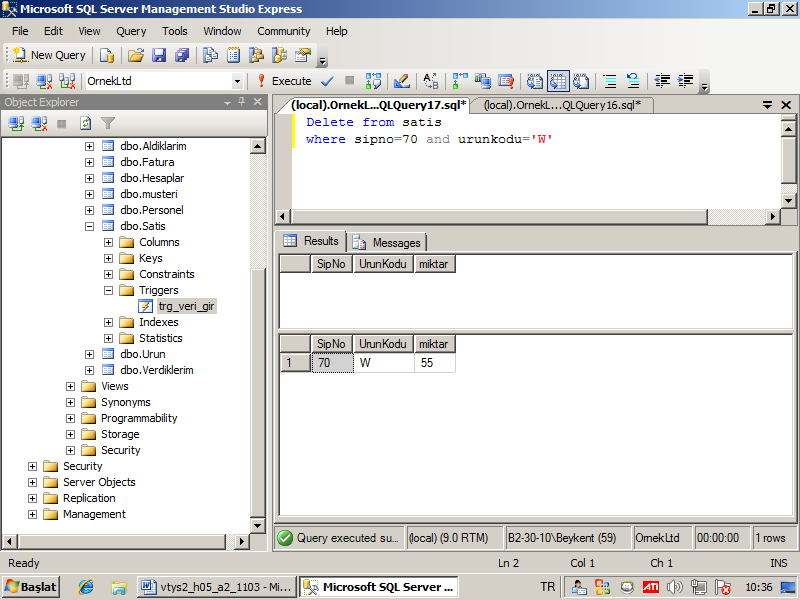
insert satis(sipno,urunkodu, miktar) values (70,’W’,100)



Update satis set miktar=55 where sipno=70 and urunkodu=’W’



Delete from satis where sipno=70 and urunkodu=’W’



Oluşturduğumuz triggerlerı silelim

DROP TRIGGER trg\_satis

DROP TRIGGER trg\_veri\_gir