Önceki dersin özeti

Tablo tipi değişken ve bir tablodan başka bir tabloya veri göndermeyi görmüştük.

Tablo tipi değişkenler

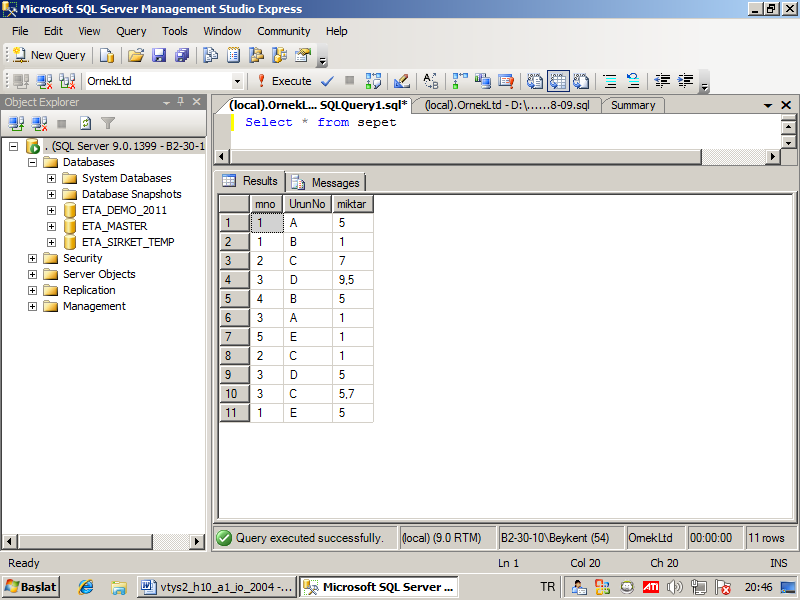
Bu değişkenler tablo gibi davranır. Tablo üzerinde yapılabilecek her şey bu değişkenlere de yapılabilir.

Bu değişkenlere veri eklenebilir, veri silinebilir ve veri güncellenebilir, ayrıca tablo gibi sorgulanabilir.

DECLARE @isim TABLE (kolon tanımı)

Örnek olarak sepet tablosuna benzeyen bir değişken tanımlayalım.

Select \* from sepet



Declare @sepet table (mno int, urunno char(1), miktar float)

Bir tablodaki verinin başka bir tabloya gönderilmesi

insert into tabloAdı (kolon listesi) select ifadesi

yukarıdaki kod, select ifadesinden dönen veriyi tabloya aktarır. Eğer tablonun tüm kolonlarına veri yazılacak ise kolon listesini yazmaya gerek yoktur.

insert into tabloAdı select ifadesi

şimdi @sepet değişkenine sepet tablosunda verilerin tümünü aktaralım.

insert into @sepet select mno,urunno, miktar from sepet

yukarıdakini aşağıdaki gibi de yazabiliriz.

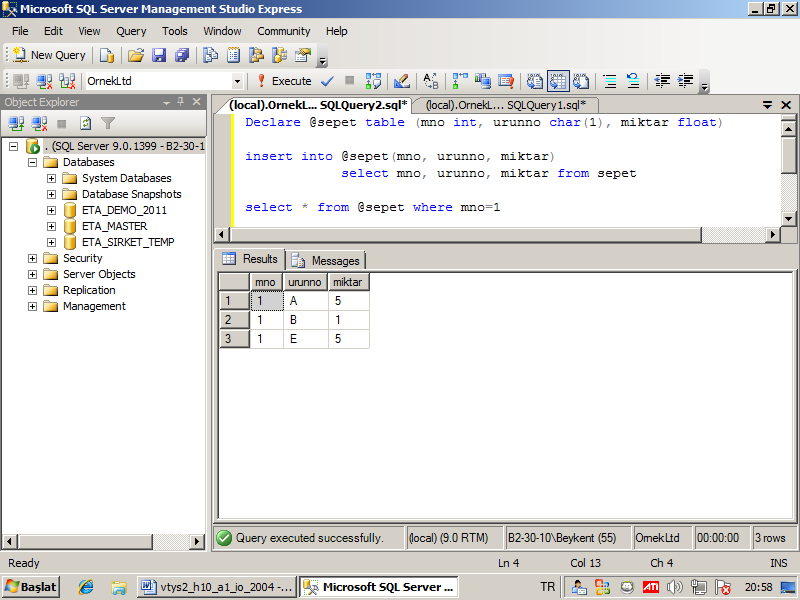
insert into @sepet(mno, urunno, miktar) select mno, urunno, miktar from sepet

yukarıdaki kod tek başına çalışmaz, değişken tanımlı değil mesajı verir.

Declare @sepet table (mno int, urunno char(1), miktar float)

insert into @sepet(mno, urunno, miktar) select mno, urunno, miktar from sepet

select \* from @sepet where mno=1



**KULLANICI TANIMLI FONKSİYONLAR**

Kullanıcı tanımlı fonksiyonlar üç çeşittir.

* Skaler değer (yani tek bir değer) veren fonksiyonlar
* Tablo döndüren fonksiyonlar (bu fonksiyonlar view’e benzer)
* Çok deyimli tablo döndüren fonksiyonlar (multi-statement functions)

Fonksiyonları veritabanı programlamada nereye koyacağımızı anlamak biraz zor olabilir. Bu nedenle view, fonksiyon ve prosedürleri karşılaştıralım.

\* Dışarıdan parametre alan bir view oluşturamayız. Bu türden ihtiyaçlar için kullanıcı tanımlı fonksiyonlar kullanılır.

\* Stored procedure’leri bir sorgunun parçası olarak kullanamazı. Prosedürler select deyimi içinde kullanılamaz. Ancak Fonksiyonlar sorgulanabilir.

\* Bazen bir tek select deyimi ile kodu yazamayız. View’lerde birden fazla kod olamaz. Bu durumda fonksiyonları tercih ederiz.

\* Sql server’da bulunmayan, örneğin replace() gibi fonksiyonları tanımlamak için kullanıcı tanımlı fonksiyonlar kullanılır.

SKALER DEĞERLİ KULLANICI TANIMLI FONKSİYONLAR

Skaler fonksiyonlar bir tek değer döndüren fonksiyonlardır. Örneğin getdate() bir skaler fonksiyondur çünkü tek bir değer verir yani şu anın tarih ve saatini verir.

Bazı durumlarda buna benzer fonksiyonlara ihtiyaç duyulur. Örneğin, bir müşterinin sepetinde kaç ürünün bulunduğunu veren bir kullanıcı tanımlı fonksiyon tanımlayabiliriz.

Genel İfade

Create function fonkAdı (varsaParametre) RETURNS geriDönüşTipi

As

Begin

Sql deyimi

RETURN geriDönüşDeğeri

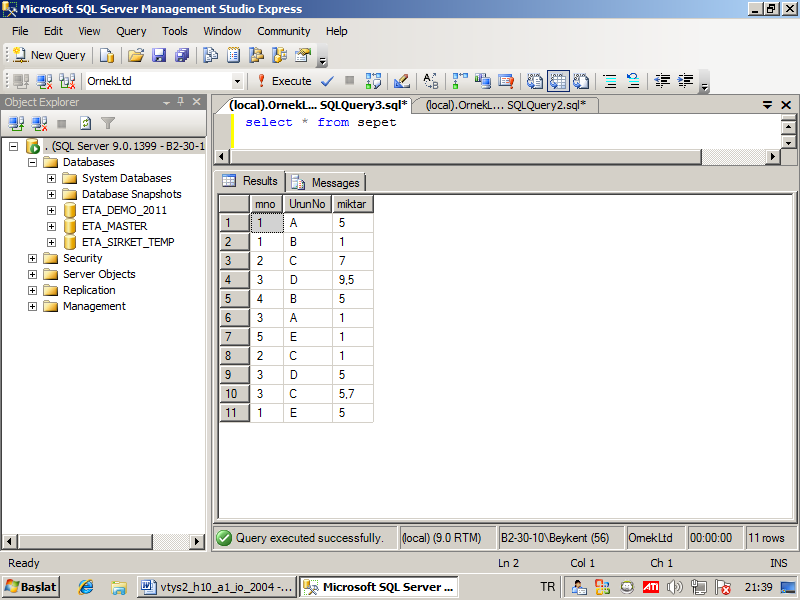
End

Go

Örnek:

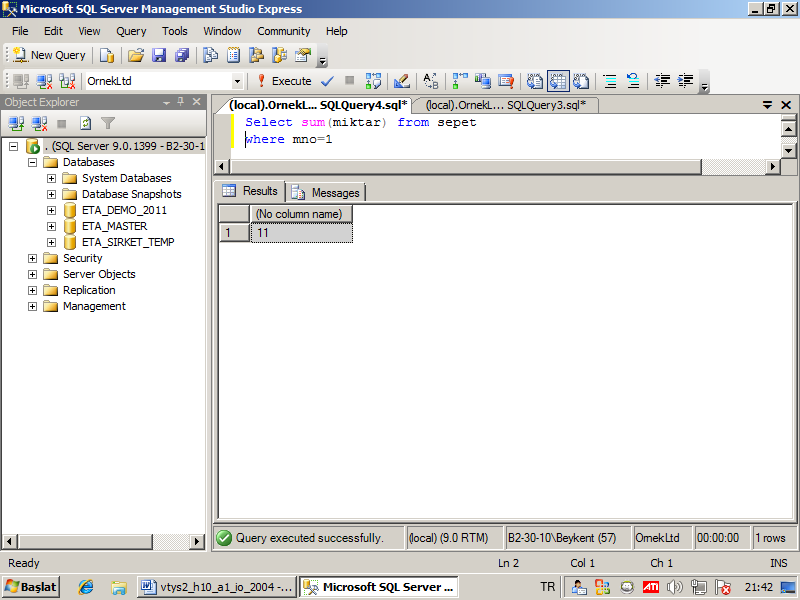
Bir müşteri no girildiğinde, bu müşterinin sepetinde toplam kaç ürün olduğunu bulan bir kullanıcı tanımlı fonksiyon oluşturunuz ve bu fonksiyonu çalıştırınız.

Çözüm:



Fonksiyonu yazmadan önce 1 nolu müşterinin ürün miktarını bulalım.

Select sum(miktar) from sepet where mno=1



Şimdi yukarıdaki fonksiyon tanımı içine koyalım.

Create function fn\_urun\_sayi (@mno int)

Returns float

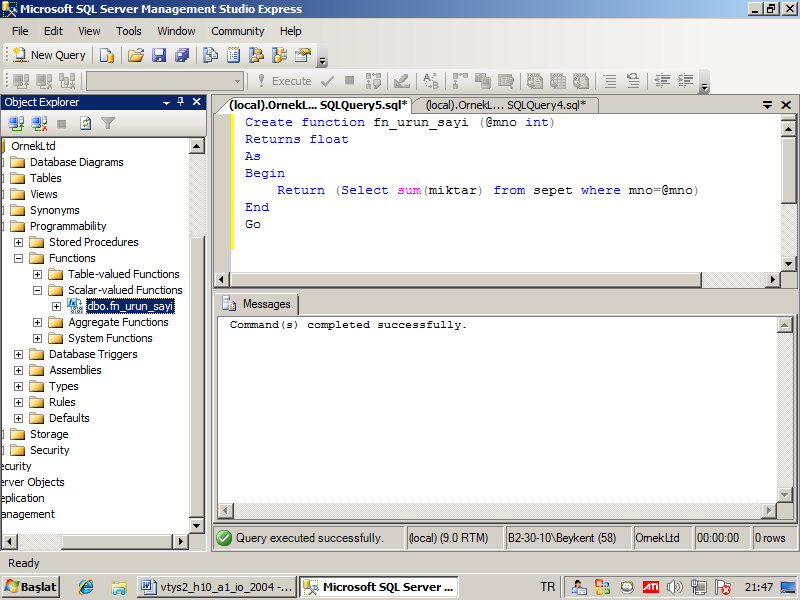
As

Begin

Return (Select sum(miktar) from sepet where mno=@mno)

End

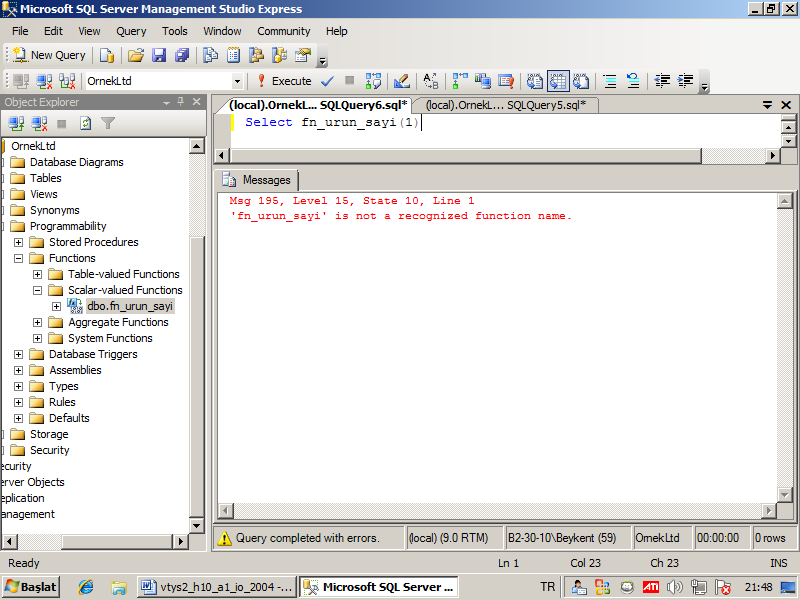
Go



Yukarıdaki fonksiyonu çalıştıralım.

Aşağıdaki kod hata verecek.

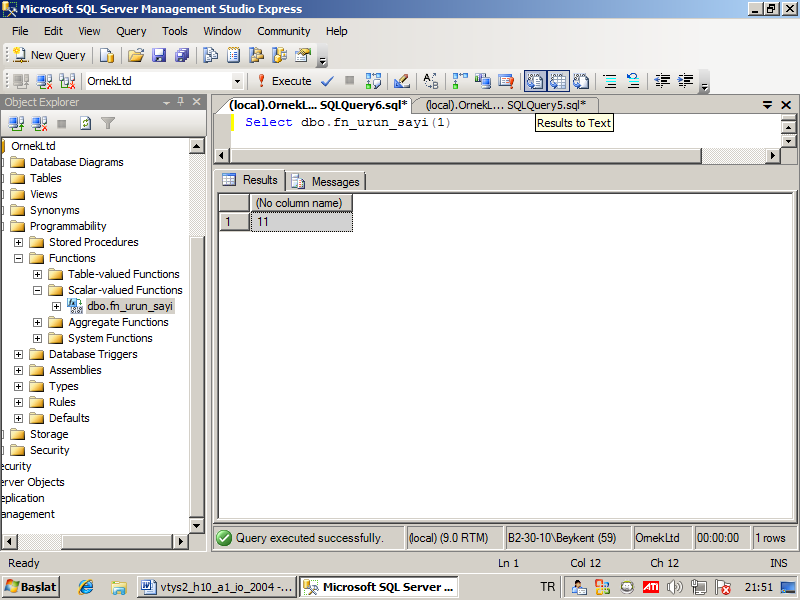
Select fn\_urun\_sayi(1)



Bu fonksiyonu, kendi fonksiyon aradı ve bulamadı. Skaler fonksiyonları şema adı ile yazılmadır. (Şemaları güvenlik ile ilgili bölümde göreceğiz.)

Varsayılan şema dbo’dur.

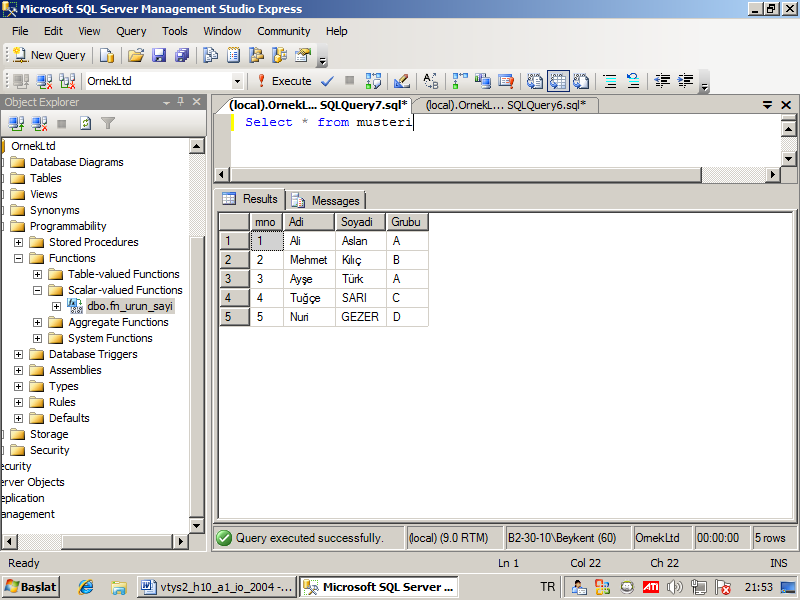
Select dbo.fn\_urun\_sayi(1)



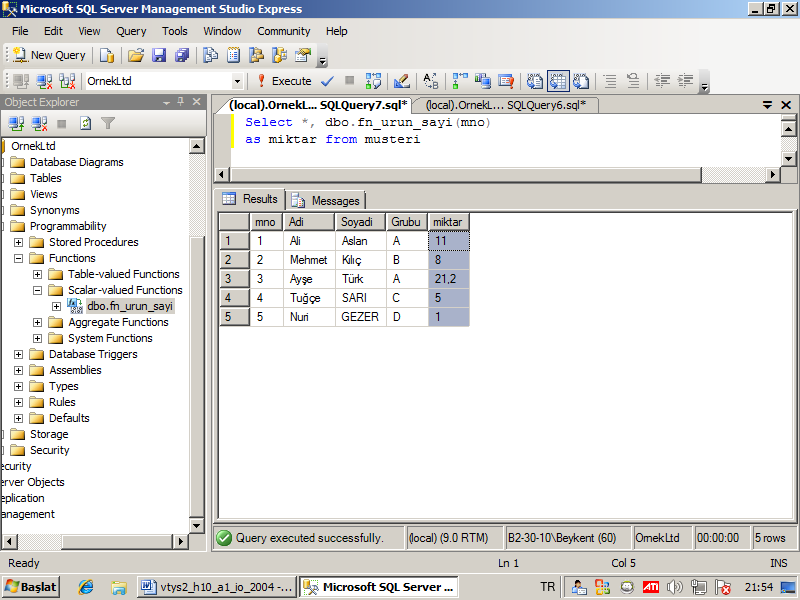
Örnek:

Yukarıdaki fonksiyonu kullanarak, müşterileri sepetindeki ürün miktarı ile birlikte veren sorguyu yazınız.

Select \* from musteri



Select \*, dbo.fn\_urun\_sayi(mno) as miktar from musteri



Örnek:

1,2 gibi ay numarası girilecek, Ocak, Şubat gibi ay adını döndüren bir kullanıcı tanımlı fonksiyon yazınız.

Create Function fn\_ay(@ay tinyint)

Returns varchar(30)

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’

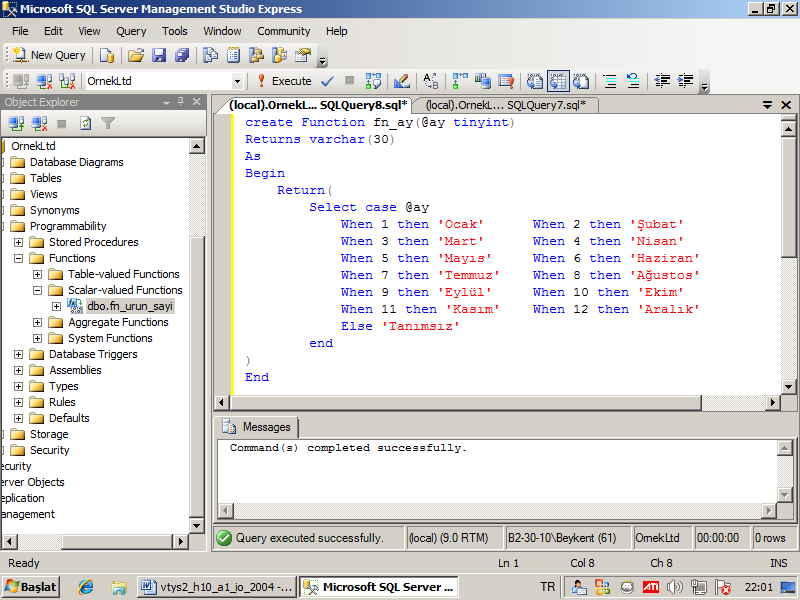
When 12 then ‘Aralık’

Else ‘Tanımsız’

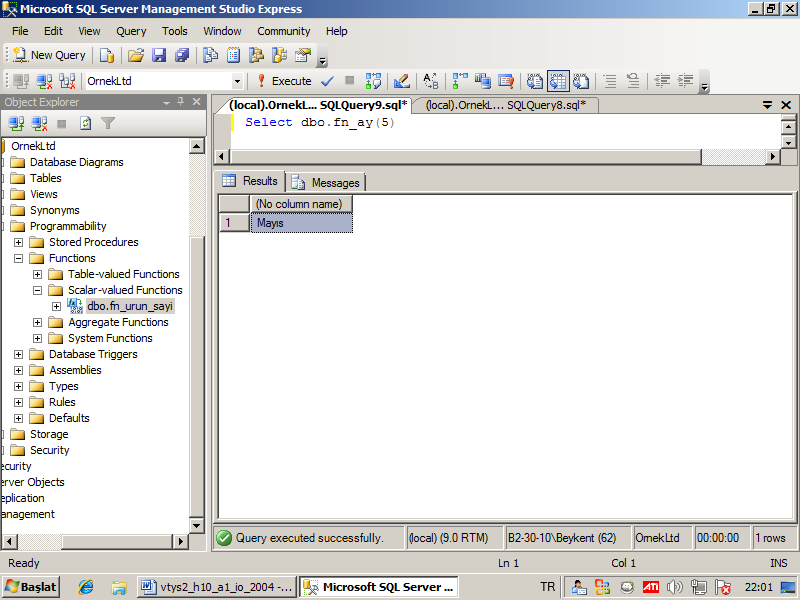
end

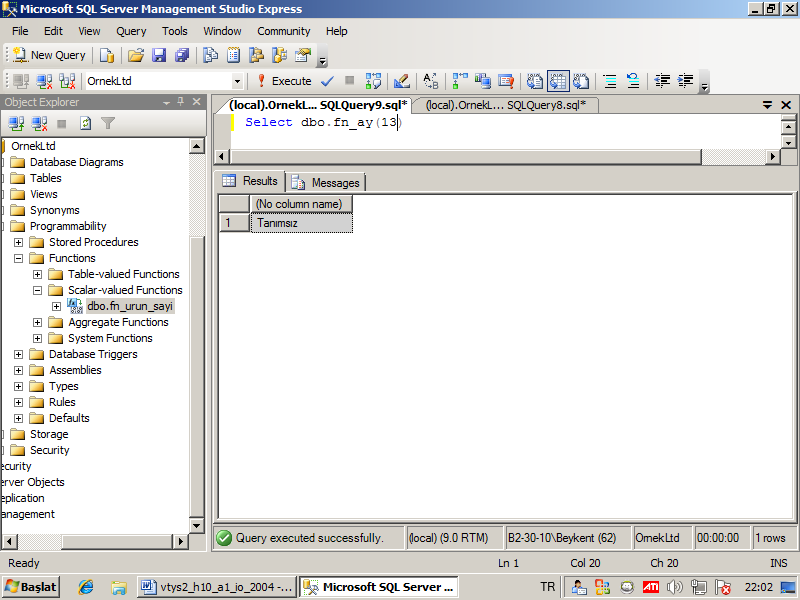
)

End



Select dbo.fn\_ay(5)





Örnek:

Herhangi bir gruba ait müşteri sayısını veren bir fonksiyon yazınız.

Create Function fn\_musteri\_sayisi (@grup char(1))

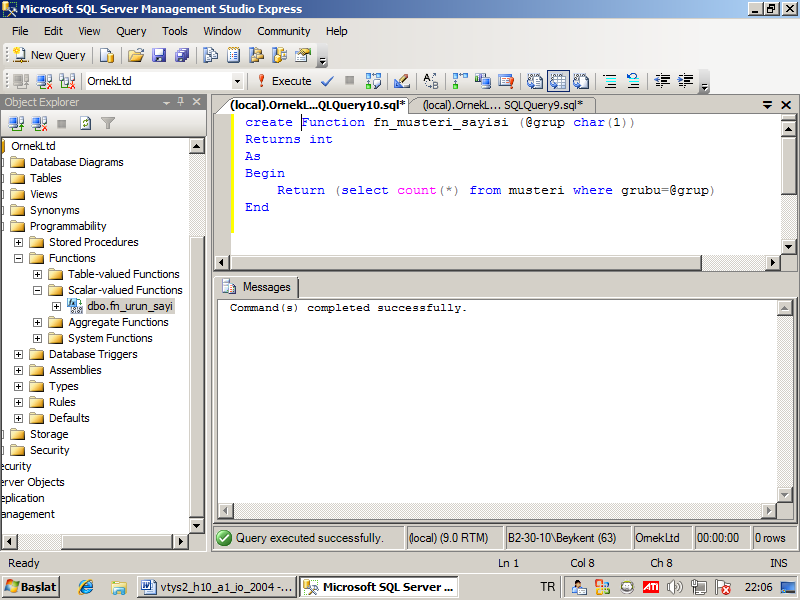
Returns int

As

Begin

Return (select count(\*) from musteri where grubu=@grup)

End



Select dbo.fn\_musteri\_sayisi(‘A’)

