ÖNCEKİ DERSİN ÖZETİ

Tablo tipi değişkenler

Yapısı tabloya benzer, tablo gibi satır ve sütunlardan oluşur. Tablo üzerinde yapılan tüm işlemler bu değişkenlere uygulanabilir.

Tanımı

DECLARE @isim TABLE (kolon tanımlaması)

şeklinde tanımlanır.

Declare @personel TABLE (sicilno varchar(10), ad varchar(30), soyad varchar(30), d\_tarihi datetime)

Yukarıda bir değişken tanımı örneği vardır.

II. Veri girme metodu : insert into …

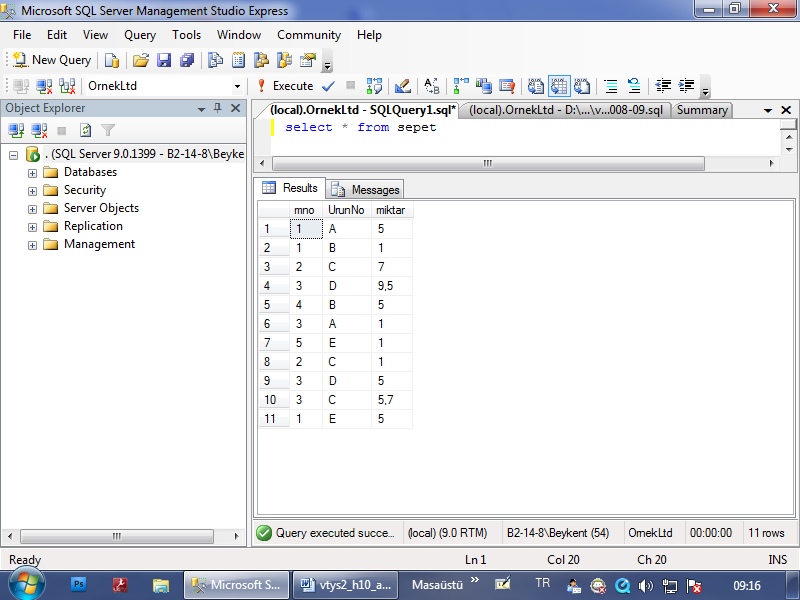
insert into tabloAdı (kolon listesi) select deyimi

Eğer tüm kolonlara veri girilecekse kolon listesini yazmaya gerek yoktur.

insert into tabloAdı select deyimi

Yukarıdaki ifade select deyimi ile dönen veriyi, tabloya aktarır. (Bir çeşit veriyi kopyalamış olur.)

Buna örnek verelim.



-- sepet değişkeni tanımla

Declare @sepet TABLE (mno int, urun\_no varchar(10),miktar float)

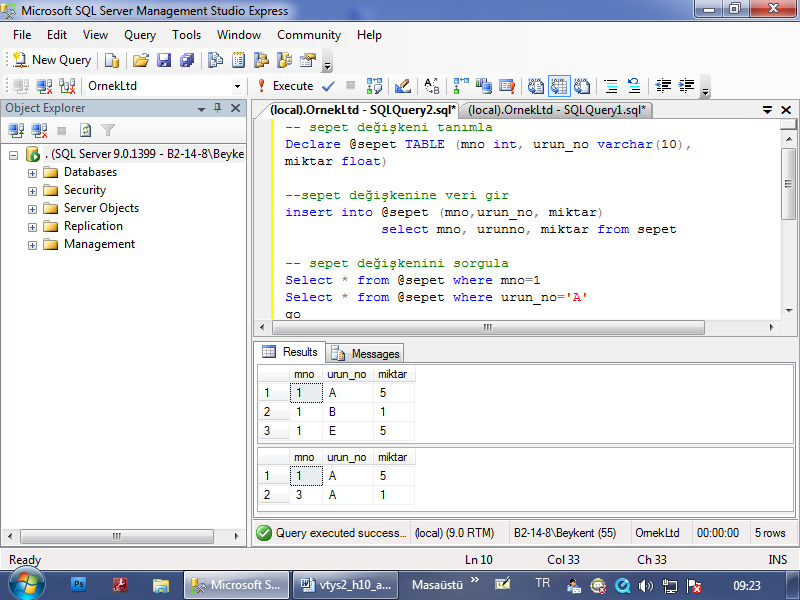
--sepet değişkenine veri gir

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

-- sepet değişkenini sorgula

Select \* from @sepet where mno=1

Go



**KULLANICI TANIMLI FONKSİYONLAR**

Programcının oluşturduğu fonksiyonlardır. Üç çeşittir.

1. Skaler değer (yani tek bir değer) veren fonksiyonlar
2. Tablo döndüren fonksiyonlar, ki bu fonksiyonlar view’e benzerdir.
3. Çok deyimle yazılan multi-statement fonksiyonlar.

Kullanıcı tanımlı fonksiyon SQL Server 2000 ‘den sonra çıkmıştır.

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

\* Dışarıdan parametre alan bir view oluşturamayız. Yani viewin parametresi yoktur ama fonksiyonun vardır.

\* stored prosedürleri sorgulayanayız. Yani select deyimi ile prosedürü sorgulayamayız. Ancak Fonksiyonlar view gibi sorgulanabilir.

\* View’ler tek bir select deyiminden meydana gelir, bazı durumlarda tek bir select ile işimizi göremeyebiliriz bu durumda fonksiyon kullanılır.

\* sql serverda bulunmayan , örneğin replace() gibi fonksiyonları tanımlamak için.

SCALER DEĞER DÖNDÜREN KULLANICI TANIMLI FONKSİYONLAR

Skaler fonksiyonlar, tek bir değer döndüren fonksiyonlardır. Örnek getdate() bir scaler fonksiyondur. Çünkü bu fonksiyon tek bir değer döndürür yanı şu anın tarih ve saatini verir.

Genel oluşturma ifadesi

CREATE FUNCTION fonkAdı   
(varsaParametre )

RETURNS geriDonuşTipi

AS

BEGIN

SQL deyimleri

RETURN geriDönüşDeğeri

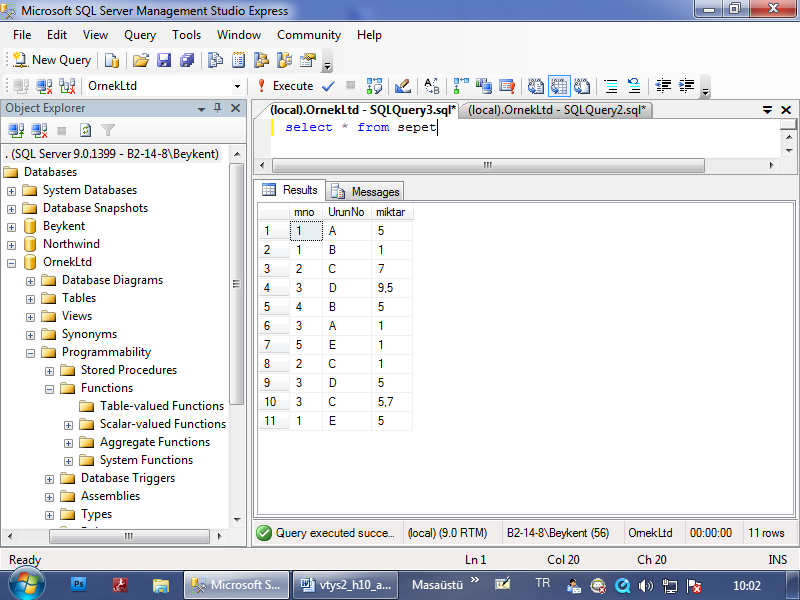
END

Go

Örnek:

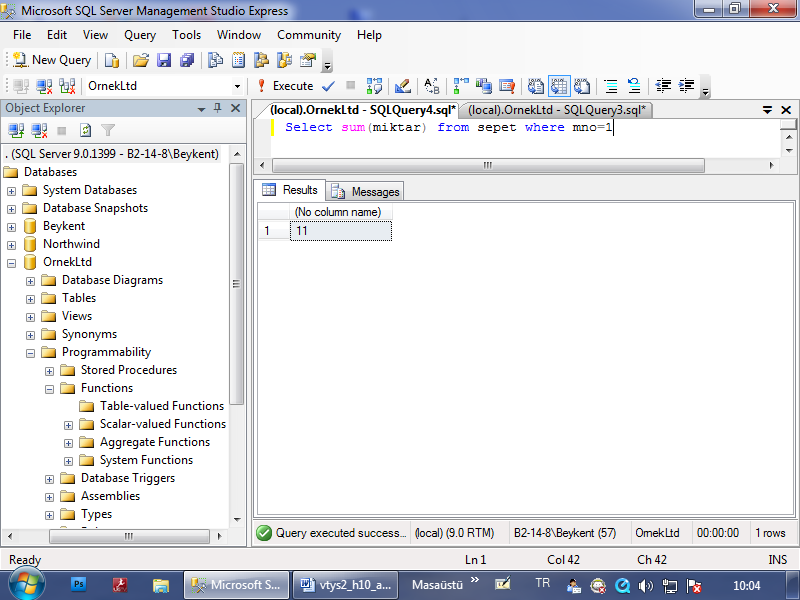
Bir müşteri no girildiğinde, bu müşterinin sepetinde toplam kaç ürün olduğunu bulan bir kullanıcı tanımlı fonksiyon oluşturunuz ve bu fonksiyonu çalıştırınız.

Çözüm:



Örneğin 1 nolu müşterinin miktar toplamını veren sorguyu yazalım.

Select sum(miktar) from sepet where mno=1



Yukarıdaki sorguyu fonksiyon içine koyalım.

CREATE FUNCTION fn\_urun\_sayi (@mno int) -- parametre

RETURNS float

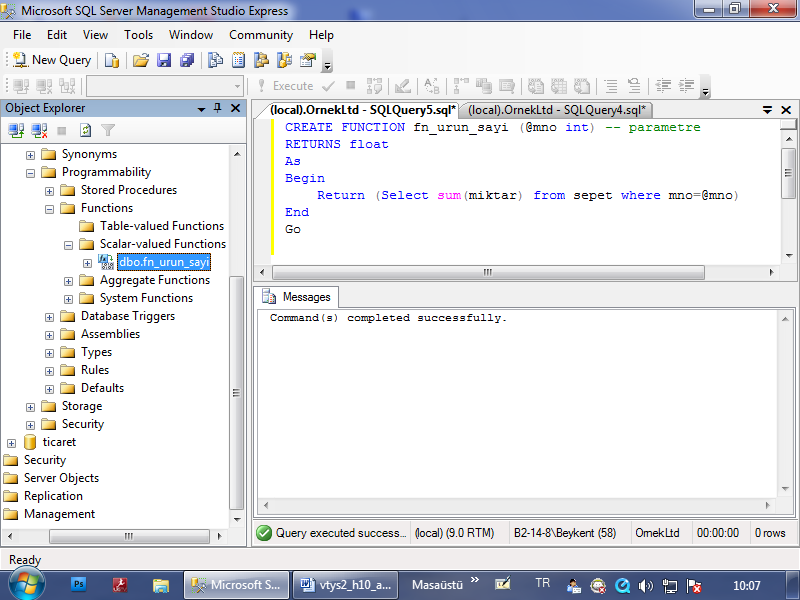
As

Begin

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

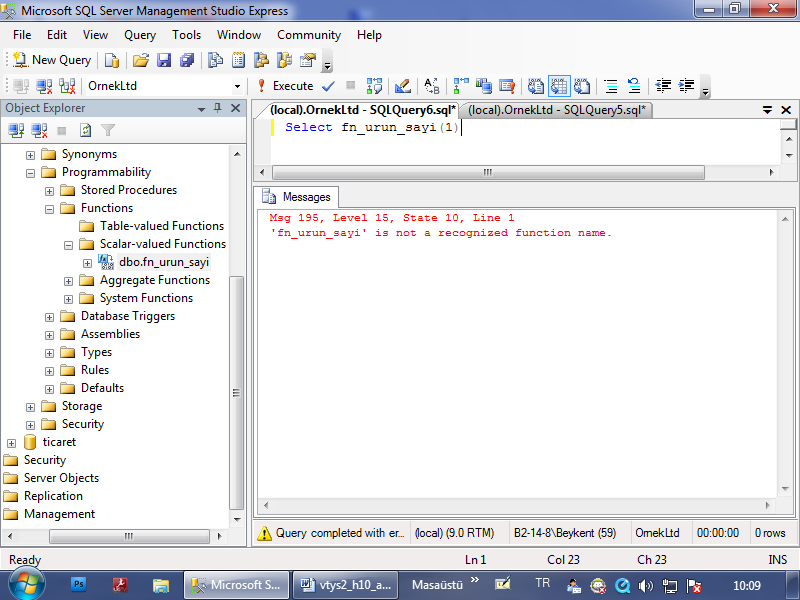
End

Go



Yukarıdaki fonksiyonu çalıştıralım. Mesaja bakalım. Hatayı giderelim.

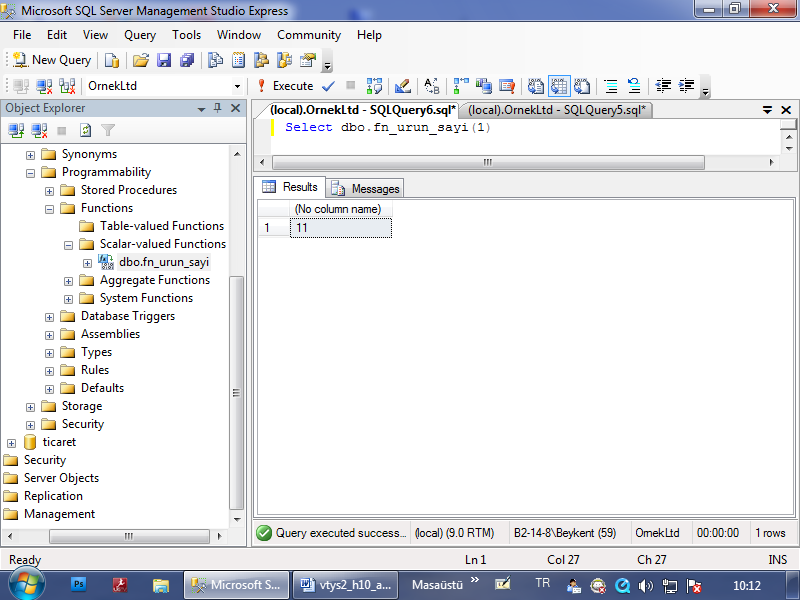
Select fn\_urun\_sayi(1)

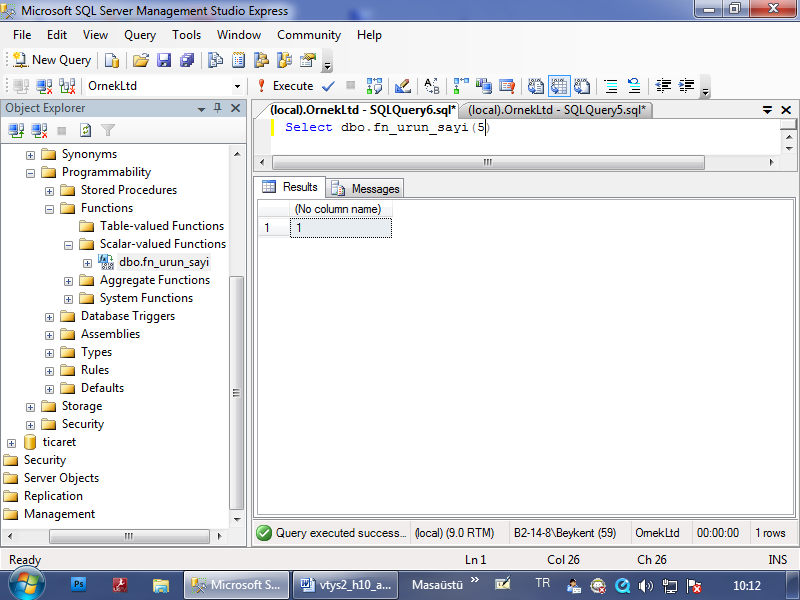


Fonksiyon adı tanımlı değil mesajı. Biz kullanıcı tanımlı skaler fonksiyonu kullanırken şema adını da yazmalıyız.

Varsayılan şema dbo’dur. Bir sonraki konuda göreceğiz.

Select dbo.fn\_urun\_sayi(1)

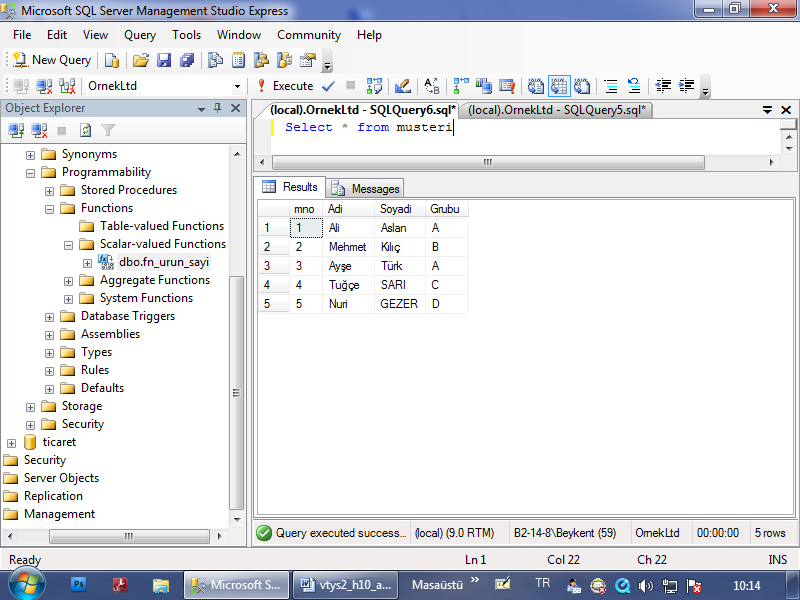




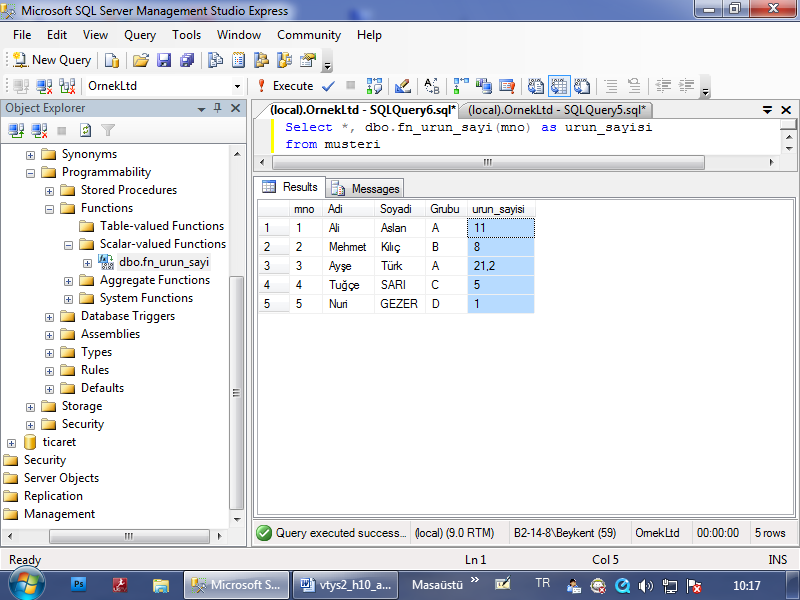
Örnek:

müşterileri listeleyiniz ve Yukarıdaki fonksiyonu kullanarak sepetindeki ürün miktarını veren sorguyu yazınız.

Select \* from musteri



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



Örnek: 1,2 gibi ayın numarasını gireceğiz, Ocak, Şubat gibi ay adını döndüren bir fonksiyon oluşturunuz.

Create function fn\_ay\_adi (@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’

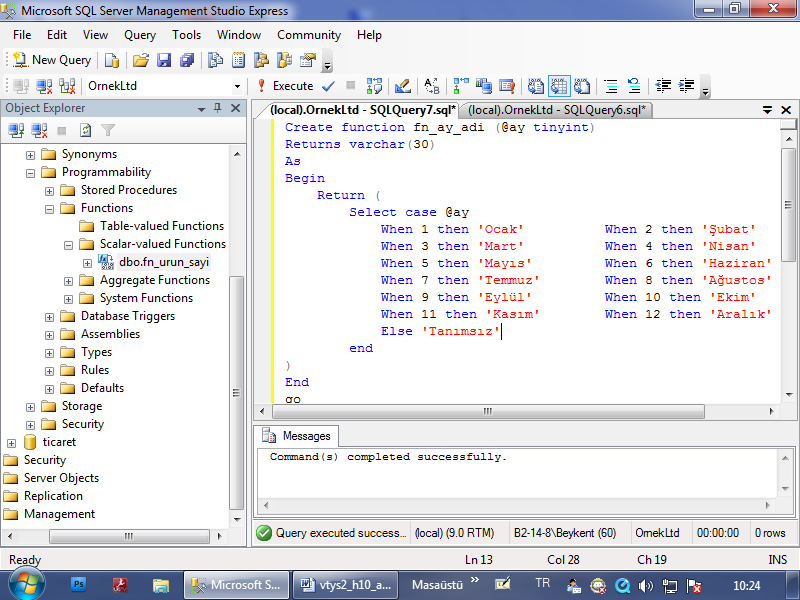
Else ‘Tanımsız’

end

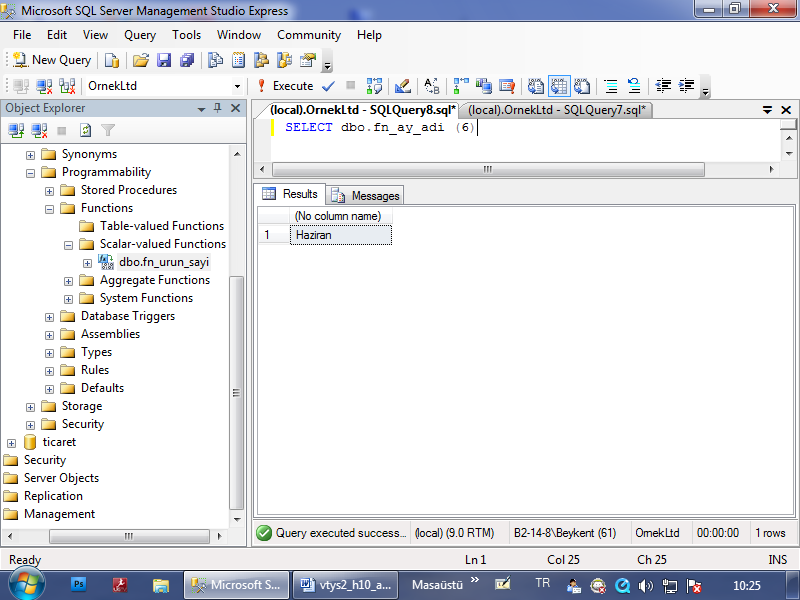
)

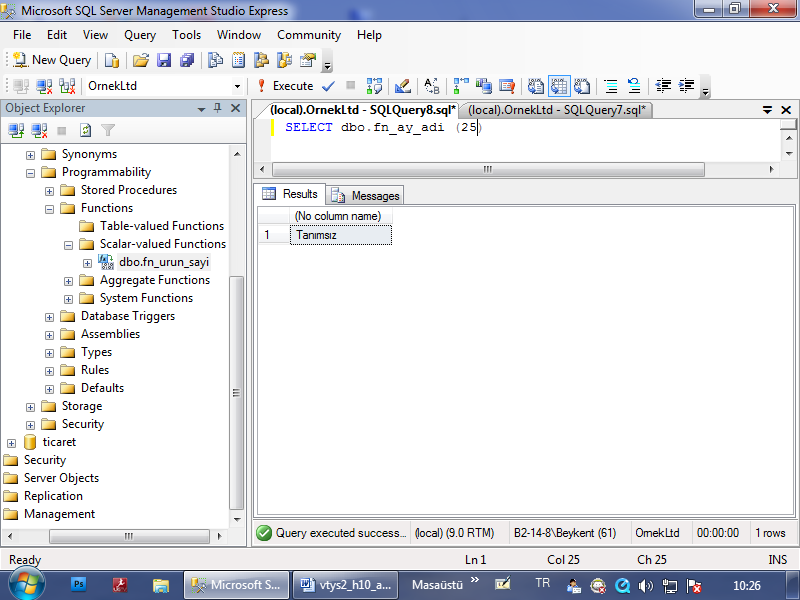
End

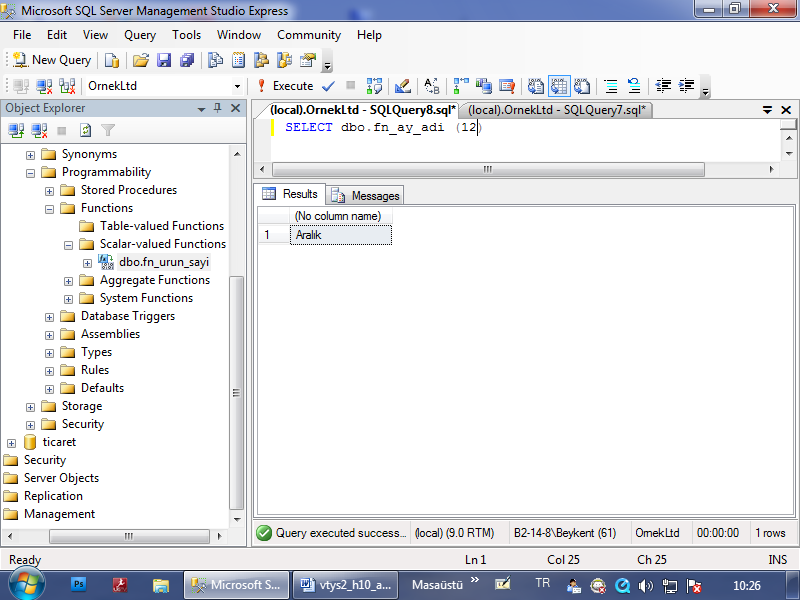
Go



SELECT dbo.fn\_ay\_adi (6)







Örnek:

Herhangi bir gruba ait müşteri sayısını veren bir fonksiyon oluşturunuz.

Create function fn\_musteri\_sayi (@grup varchar(10))

Returns int

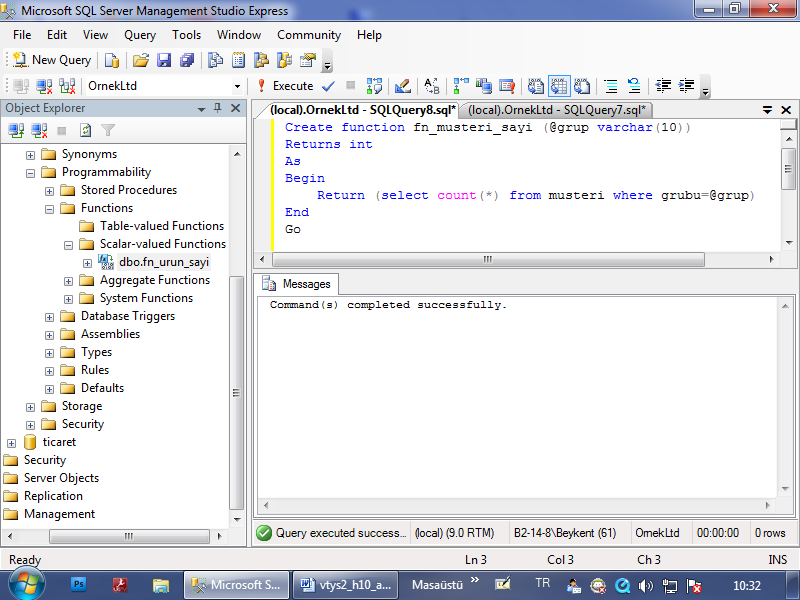
As

Begin

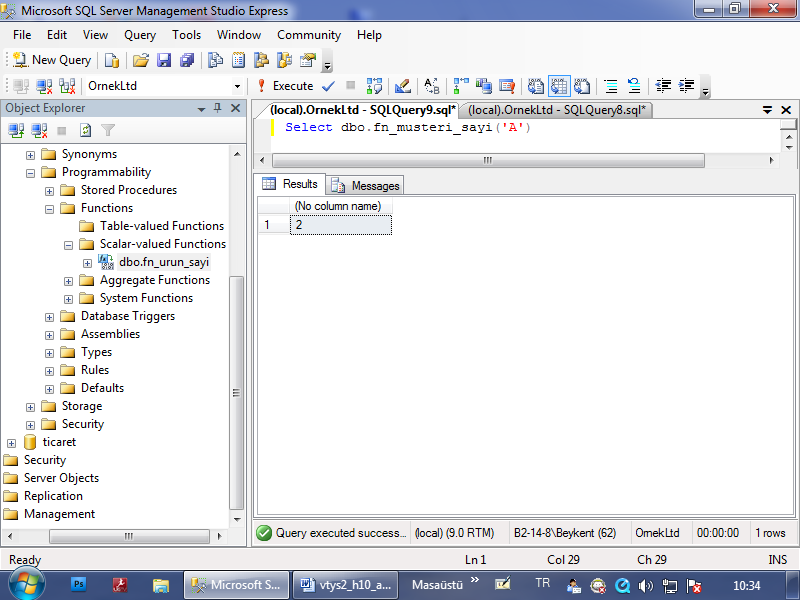
Return (select count(\*) from musteri where grubu=@grup)

End

Go



Select dbo.fn\_musteri\_sayi(‘A’)



Create function fn\_tum\_musteri\_sayi ()

Returns int

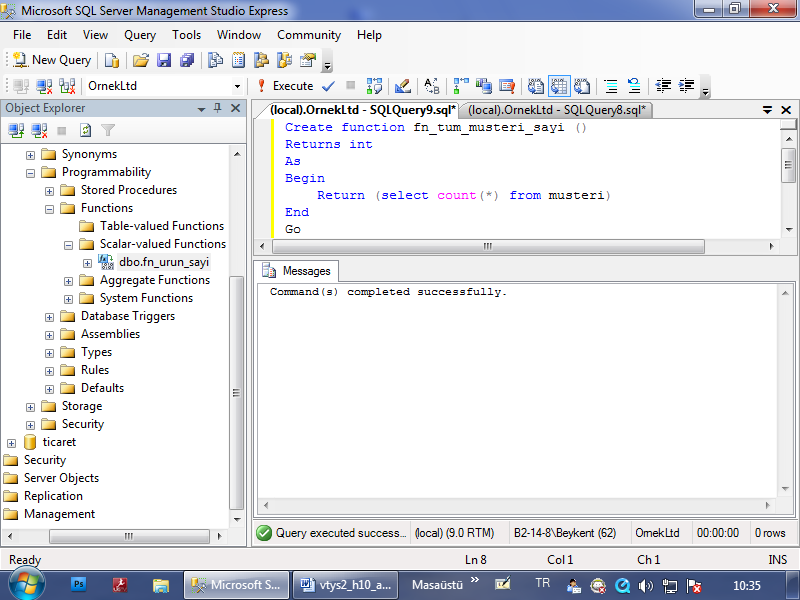
As

Begin

Return (select count(\*) from musteri)

End

Go



Select dbo.fn\_tum\_musteri\_sayi ()

