STORED PROCEDURE

Birden fazla işlemi, paketlenmiş bir halde bir tek komut ile çalıştırmak istediğimizde Stored Procedure’leri kullanırız.

Stored prosedürlerin içinde tüm Transact-SQL deyimleri bulunabilir.

Bir prosedür oluştuğunda veritabanı sunucusunda saklanır. Her ihtiyaç duyulduğunda defalarca çağrılabilir.

Prosedürler ağ bazlı çalışmalarda ağ trafiğini azaltır.

Genel ifade:

CREATE : oluşturmak

ALTER : yapıyı değiştirmek

DROP : ilgili nesneyi silmek

CREATE PROC yada

CREATE PROCEDURE

CREATE PROC prosedürAdı

varsaParametreler

AS

….

….

SQL deyimleri

….

Go

Örnek:

Hesaplar arası para transferi yapan bir prosedür oluşturunuz. Ancak bakiyenin sıfırın altına inmesine izin vermesin

Çözüm:

Kullanacağımız parametreler.

1-Alacaklı hesabın numarası

2- Borçlu hesabın numarası

3- Göndereceğimiz para miktarı

Prosedürün adı: sp\_para\_transfer

CREATE PROC sp\_para\_transfer

@alacakli\_hesap varchar(10),

@gonderen\_hesap varchar(10),

@para\_miktari Money

AS

….

Sql deyimleri

…

Go

Şimdi prosedürün içindeki sql komutlarını yazalım.

Gonderen hesabın sıfırın altına düşüp düşmediğini, yani yeterli bakiyenin olup olmadığını hesaplayalım.

IF (Gönderen hesabın bakiyesi)>= göndereceğim para

Parayı gönder

ELSE

Yeterli bakiye yok mesajı ver

Yukarıdaki IF’i yazalım

IF (select bakiye from Hesaplar where hesapno=@gonderen\_hesap)> @para\_miktari

BEGIN

--Parayı gönder

-- gönderenin parasını eksilt

Update hesaplar set bakiye=bakiye - @para\_miktari where hesapno=@gonderen\_hesap

-- alıcı hesabın parasını artır

Update hesaplar set bakiye=bakiye + @para\_miktari where hesapno=@alacakli\_hesap

END

ELSE

Print ‘Bakiye yetersiz’

Yukarıdaki iki kodu birleştirelim.

CREATE PROC sp\_para\_transfer

@alacakli\_hesap varchar(10),

@gonderen\_hesap varchar(10),

@para\_miktari Money

AS

IF (select bakiye from Hesaplar where hesapno=@gonderen\_hesap)> @para\_miktari

BEGIN

--Parayı gönder

-- gönderenin parasını eksilt

Update hesaplar set bakiye=bakiye - @para\_miktari where hesapno=@gonderen\_hesap

-- alıcı hesabın parasını artır

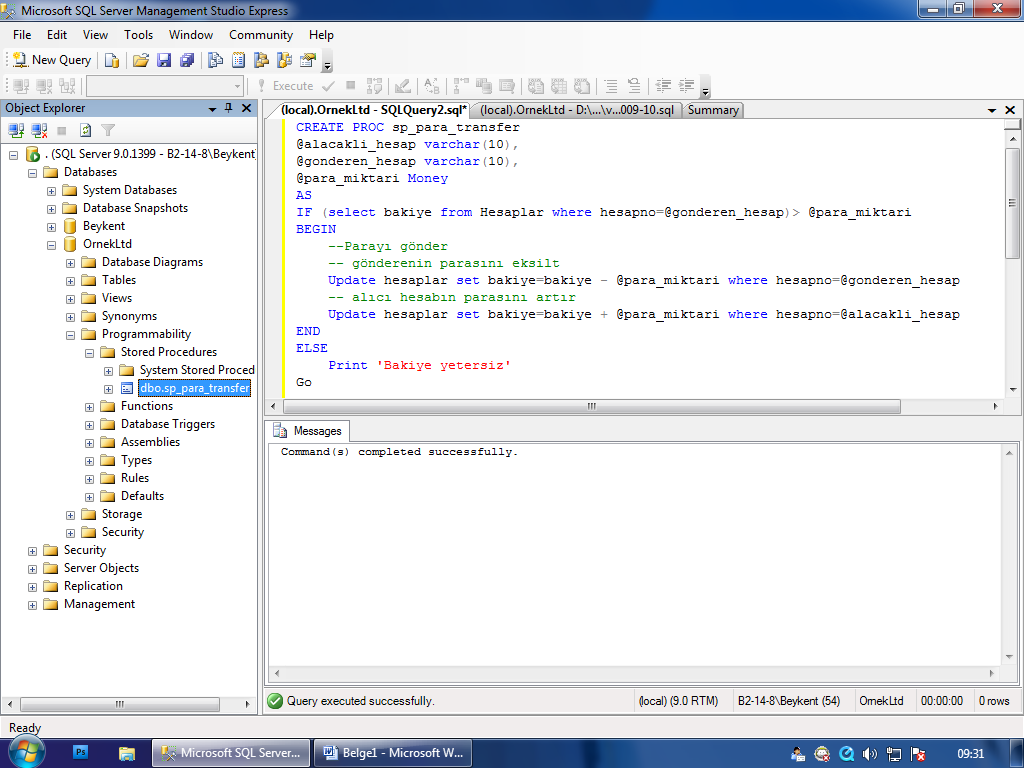
Update hesaplar set bakiye=bakiye + @para\_miktari where hesapno=@alacakli\_hesap

END

ELSE

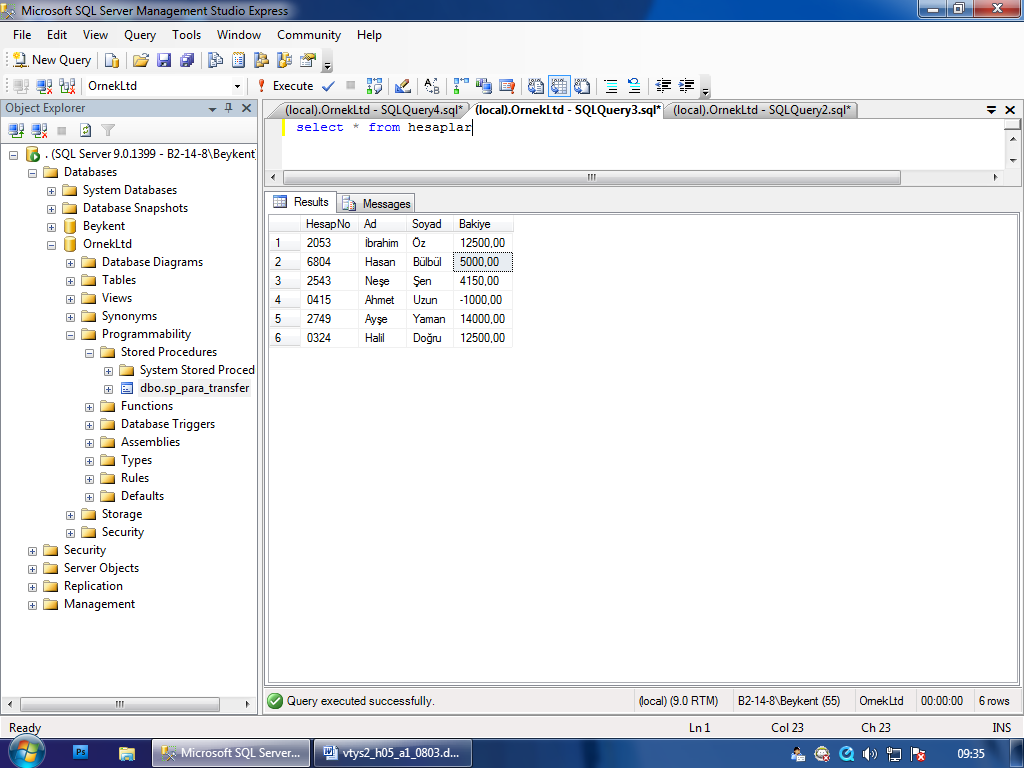
Print ‘Bakiye yetersiz’

Go

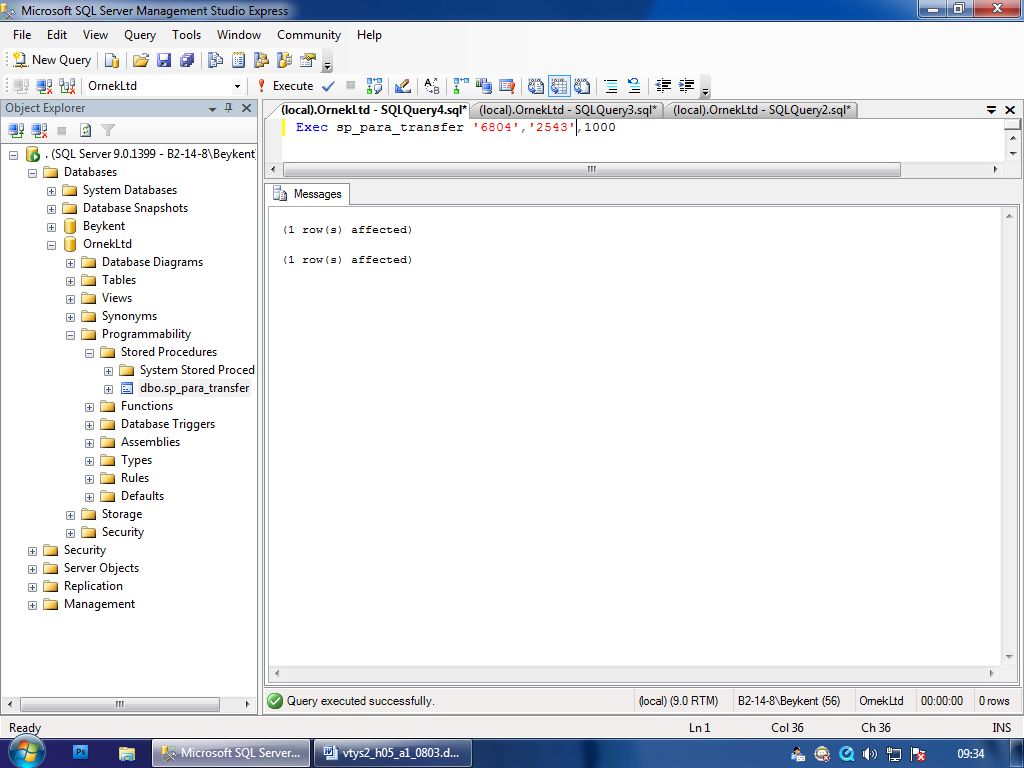


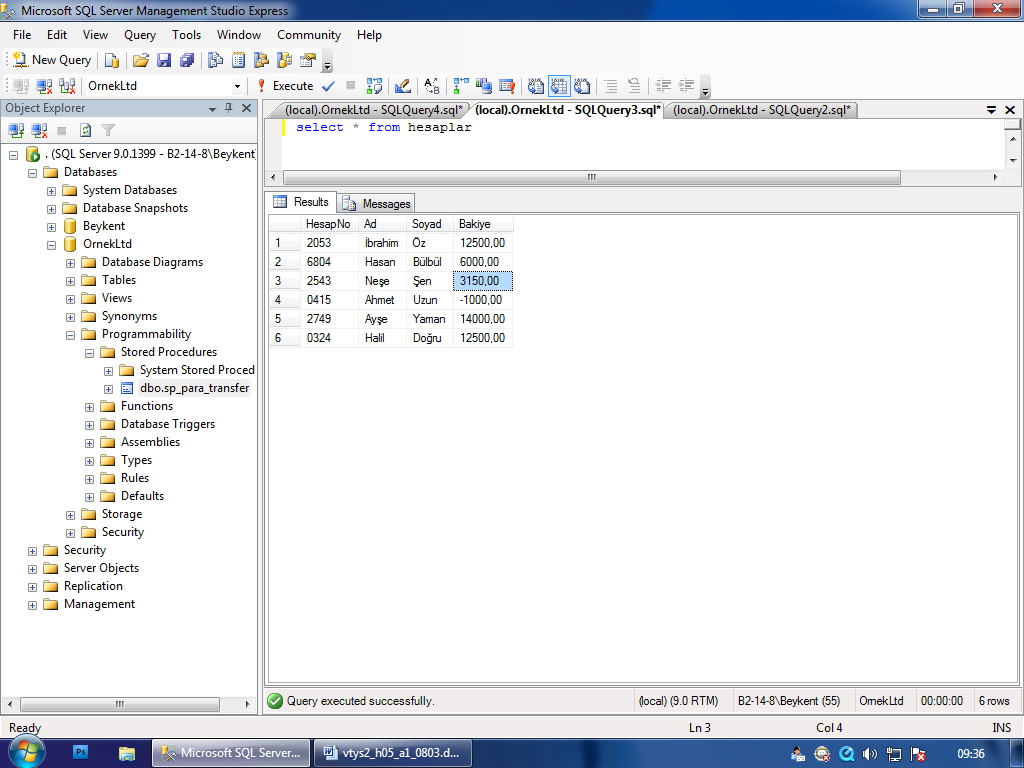
Yukarıdaki prosedürü çalıştıralım.

2543 nolu hesaptan, 6804 nolu hesaba 1000 TL gönderelim.



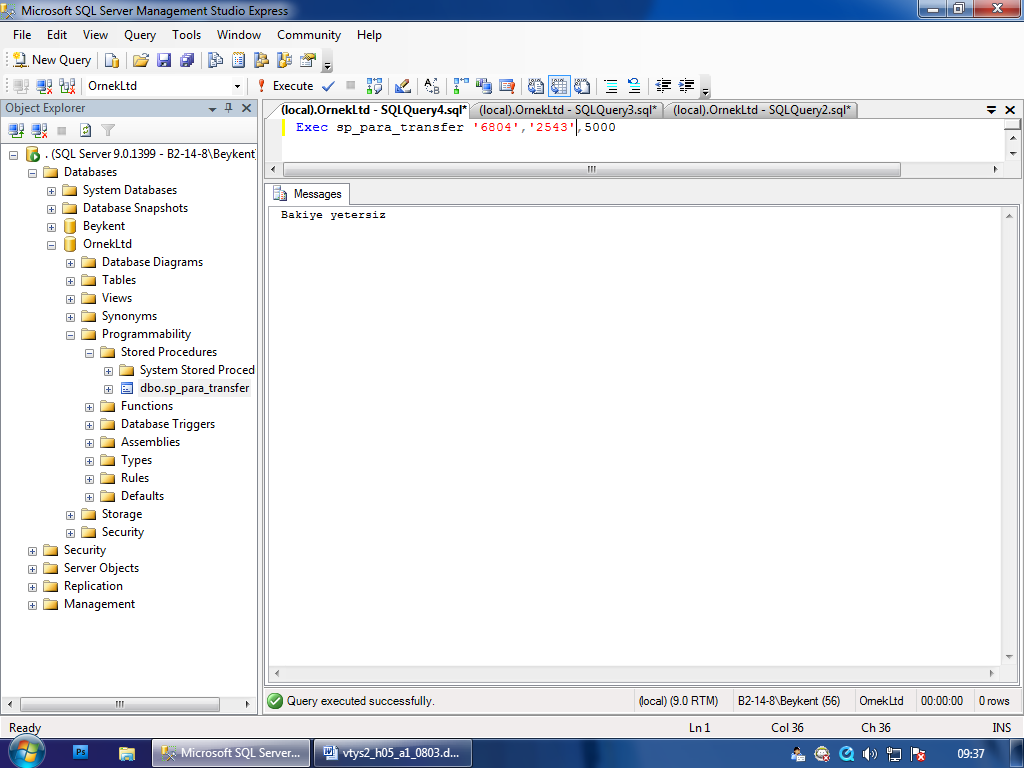
Exec sp\_para\_transfer ‘6804’,’2543’,1000





2543 nolu hesaptan, 6804 nolu hesaba 5000 TL gönderelim.

Exec sp\_para\_transfer ‘6804’,’2543’,5000



Prosedürlerin çalıştırılması

EXEC prosedürAdı varsaSırasıİlePametreye atanan değerler

Exec sp\_para\_transfer ‘6804’,’2543’,5000

Yukarıdaki kodda parametreye gönderilen değer sırası yazıldı. Parametre gönderilen değer parametre adı ile birlikte yazılabilir. Bu durumda sıralama önemli değildir.

Exec sp\_para\_transfer @alacakli\_hesap=‘6804’, @gonderen\_hesap=’2543’,   
@para\_miktari =5000

Veya sıralama değişik

Exec sp\_para\_transfer @para\_miktari=5000,

@alacakli\_hesap=‘6804’, @gonderen\_hesap=’2543’

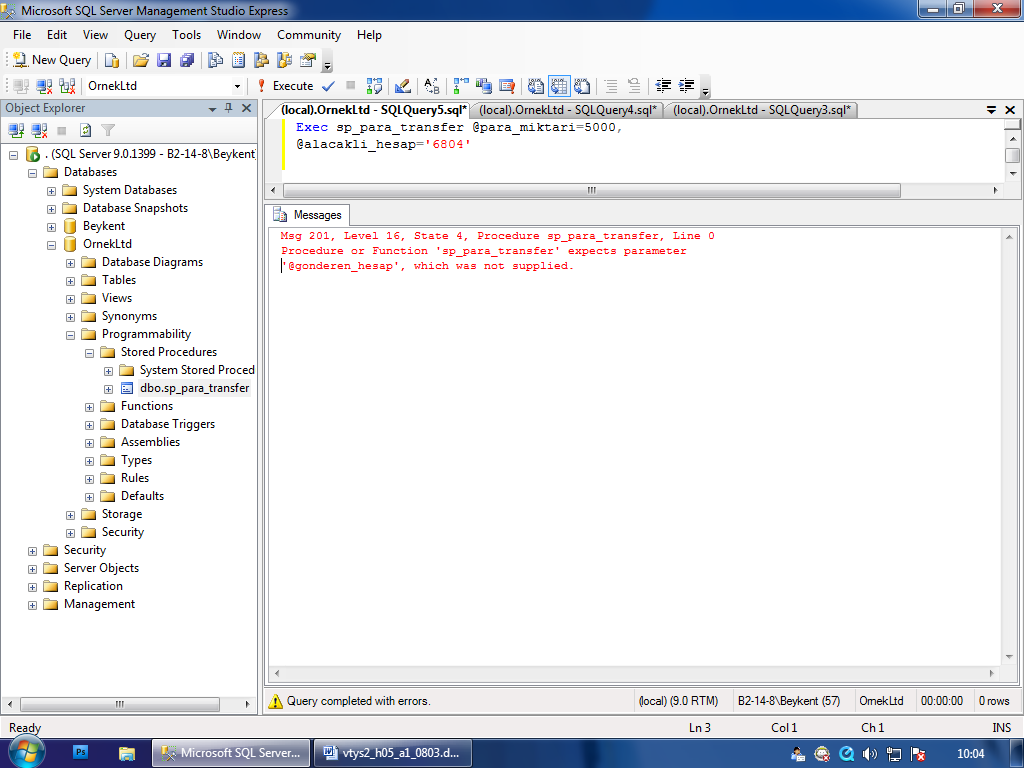
Şimdi, yukarıdaki prosedürdeki parametrelere varsayılan değer olarak NULL değerini atayalım.

Bir paramtreye varsayılan değer aşağıdaki gibi atanır.

@parametre veritipi = varsayılan

@para Money = 5000

Varsayılan değer atamadan önce eksik parametre ile yukarıdaki prosedürü çalıştıralım.



Hata mesajı verdi

ALTER PROC sp\_para\_transfer

@alacakli\_hesap varchar(10)=NULL,

@gonderen\_hesap varchar(10)=NULL,

@para\_miktari Money = 0

AS

IF (select bakiye from Hesaplar where hesapno=@gonderen\_hesap)> @para\_miktari

BEGIN

--Parayı gönder

-- gönderenin parasını eksilt

Update hesaplar set bakiye=bakiye - @para\_miktari where hesapno=@gonderen\_hesap

-- alıcı hesabın parasını artır

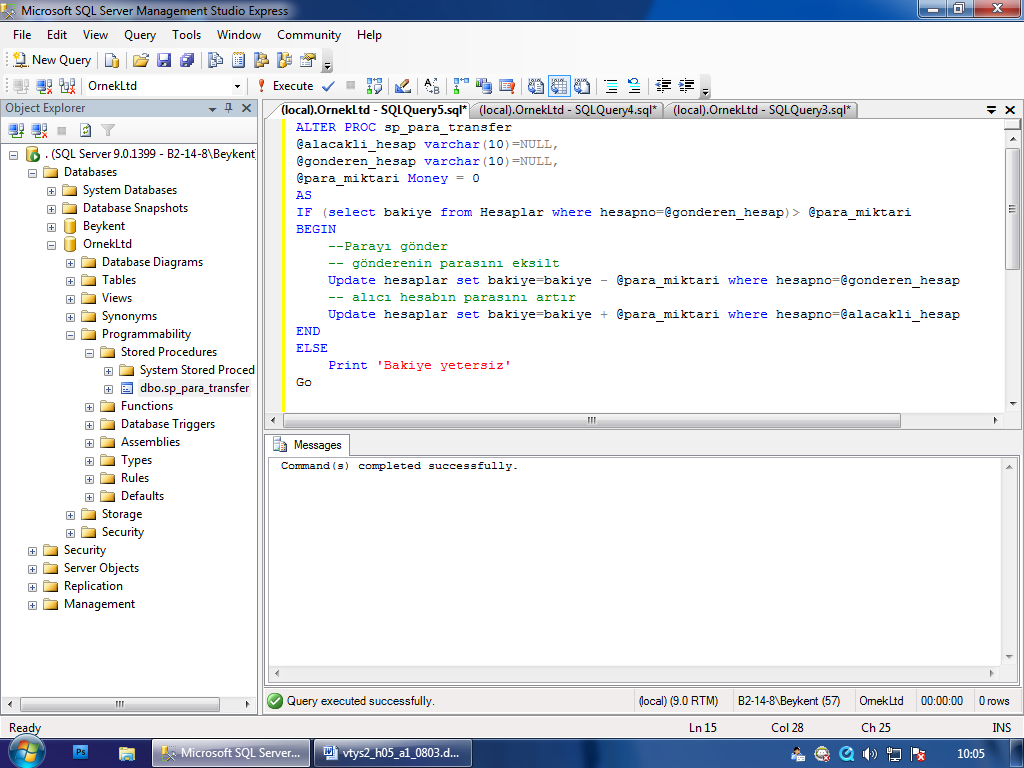
Update hesaplar set bakiye=bakiye + @para\_miktari where hesapno=@alacakli\_hesap

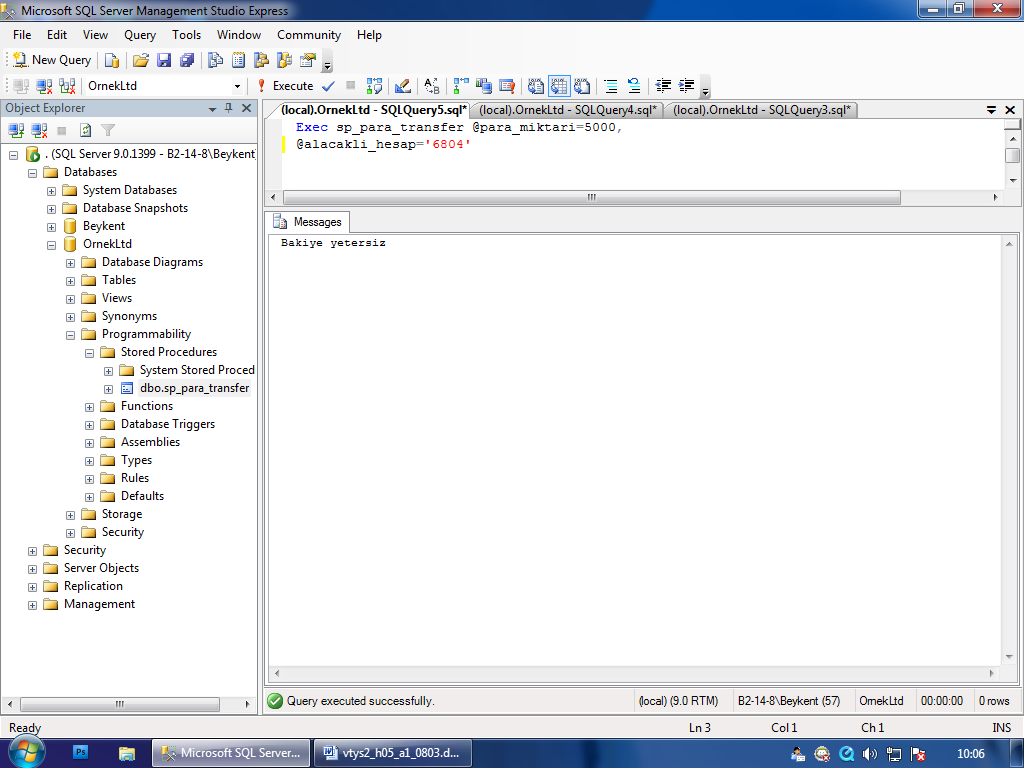
END

ELSE

Print ‘Bakiye yetersiz’

Go





Parametre eksik kullanıldığında exec deyiminde mutlaka parametre adı yazılmalıdır.

Örnek:

Verilen bir hesap numarasının bakiyesini artırıp azaltan bir prosedür yazınız. Bu prosedürde parametreler: hesap\_no, para değeri, işlem tipi olsun. (işlem tipi 1 ise azalsın 0 ise artsın)

Çözüm:

CREATE PROC sp\_bakiye\_degistir

@hesap\_no varchar(10),

@para Money,

@islem tinyint=1 --varsayılan

AS

….

Sql komutları

….

Go

Sql komutlarını yazalım

IF @islem = 1

Bakiye azalt

Else

Bakiye artır

-- bakiye azalt

Update hesaplar set bakiye=bakiye-@para where hesapno=@hesap\_no

-- bakiye artır

Update hesaplar set bakiye=bakiye+@para where hesapno=@hesap\_no

Yukarıdakileri kodun içine yerleştirelim.

CREATE PROC sp\_bakiye\_degistir

@hesap\_no varchar(10),

@para Money,

@islem tinyint=1 --varsayılan

AS

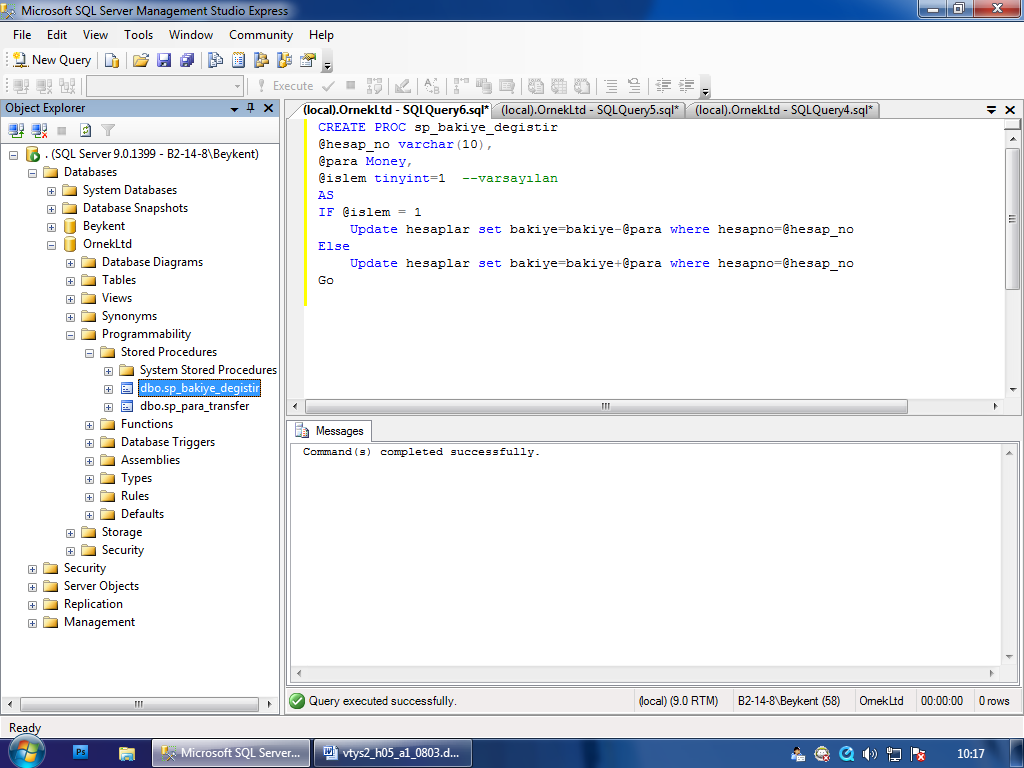
IF @islem = 1

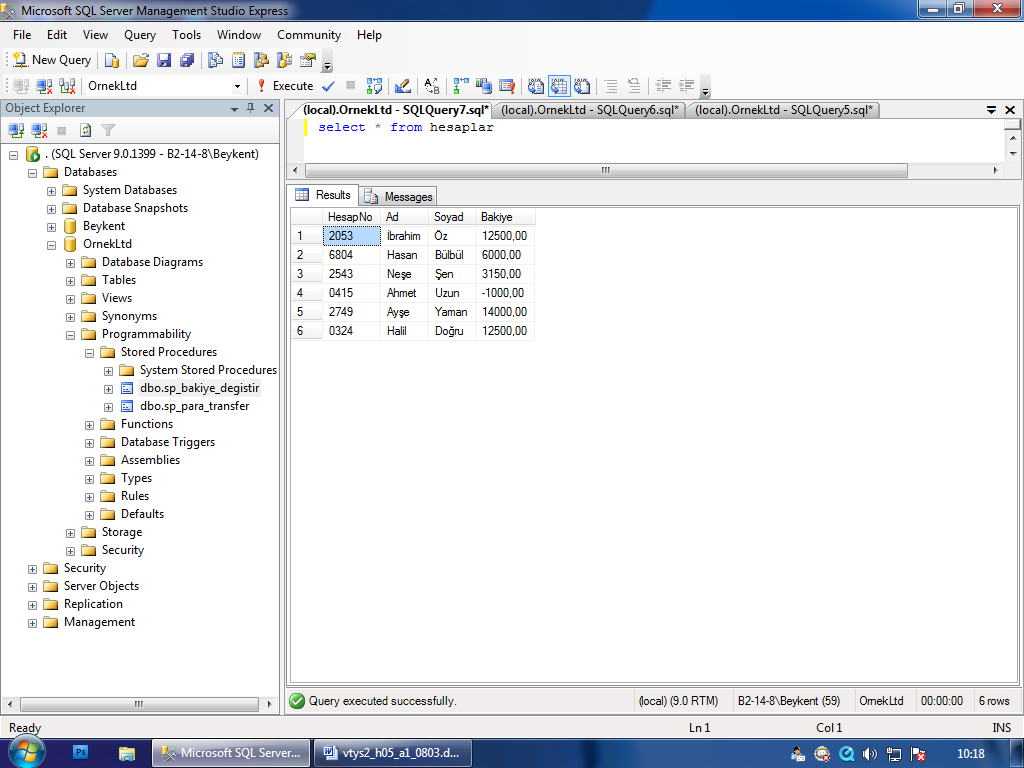
Update hesaplar set bakiye=bakiye-@para where hesapno=@hesap\_no

Else

Update hesaplar set bakiye=bakiye+@para where hesapno=@hesap\_no

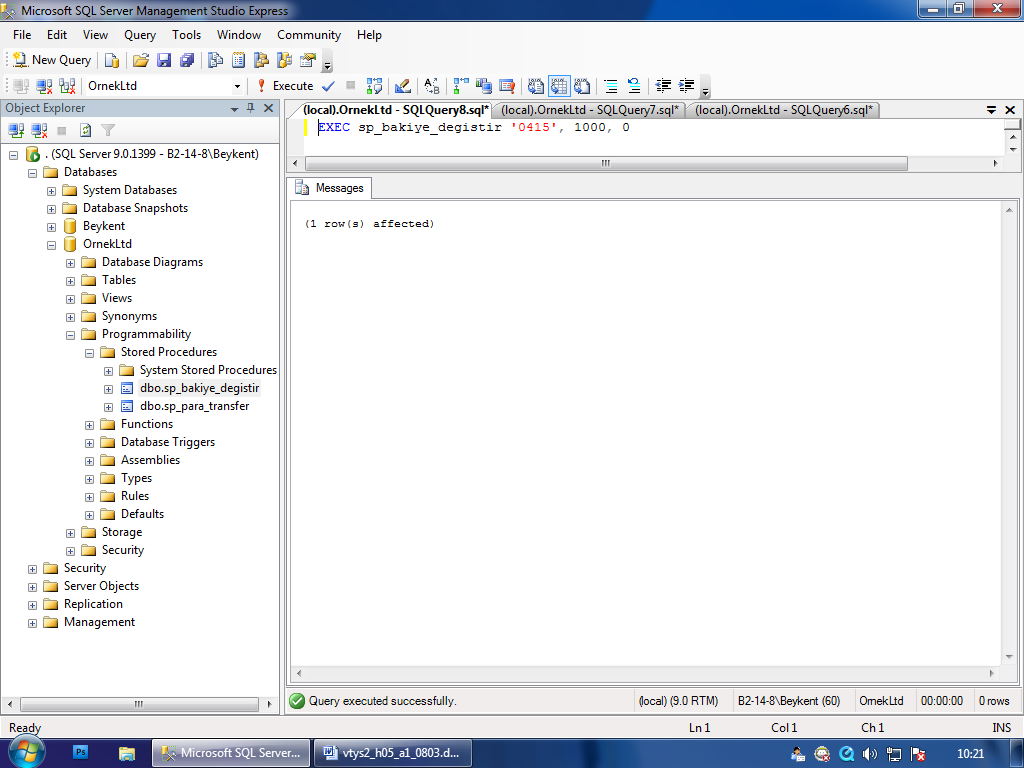
Go

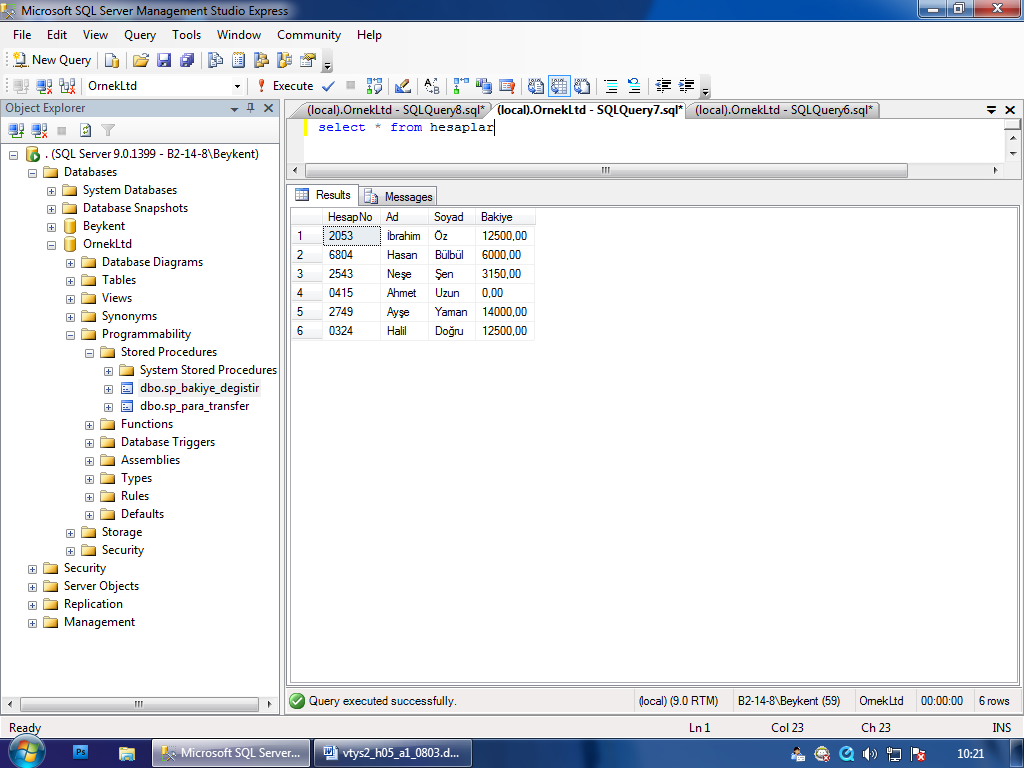




0415 nolu hesabı 1000TL artıralım

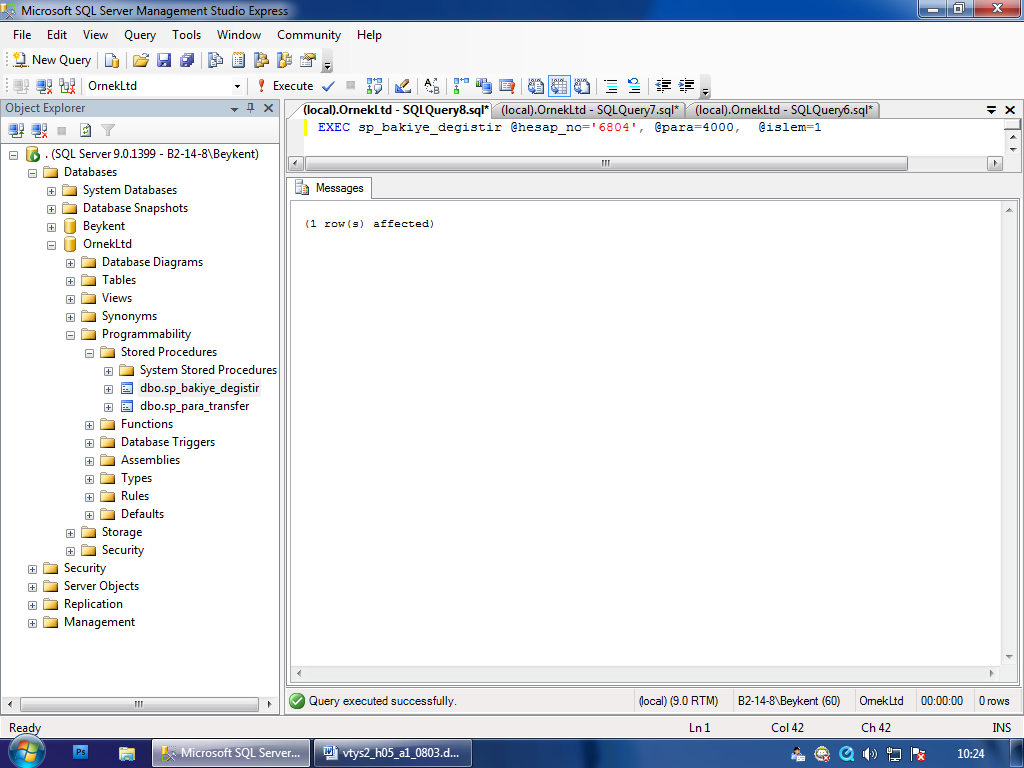
EXEC sp\_bakiye\_degistir ‘0415’, 1000, 0

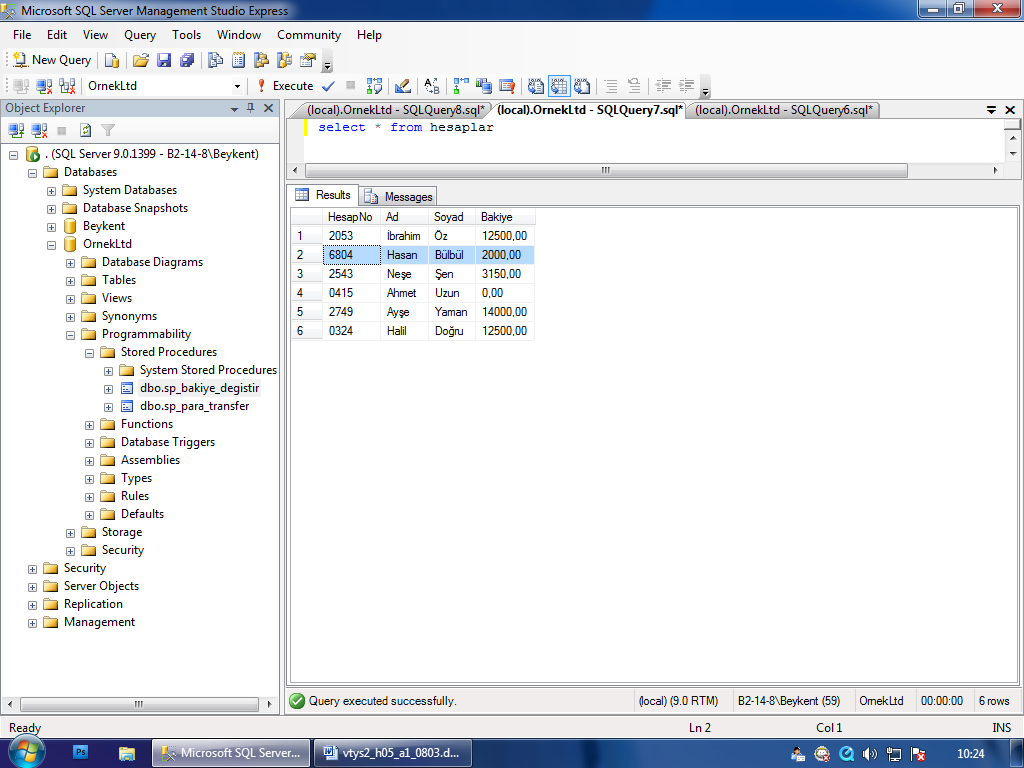




6804 nolu hesabın bakiyesini 4000 azaltalım.

EXEC sp\_bakiye\_degistir @hesap\_no=‘6804’, @para=4000, @islem=1





EXEC sp\_bakiye\_degistir @hesap\_no=‘6804’, @para=4000

