KÜRSÖRLER

Bir tablo üzerinde satır, satır işlem yapılmasını sağlar.

Kürsörlerin tanımlanması

DECLARE …1… CURSOR FOR …2…

Veya

DECLARE …1… SCROLL CURSOR FOR …2…

1 nolu yere, kürsöre verdiğimiz isim yazılır.

2 nolu yere, kürsörün içinde dolaşacağı SELECT sorgusu

İkinci ifadedeki SCROLL kelimesi bulunmaz ise, kürsör sadece ileri hareket eder.

Eğer SCROLL kelimesi varsa, kürsör istediği yere konumlanabilir, ileri gidebilir, geri gidebilir, ilk kayıta gidebilir, son kayıta gidebilir.

Eğer biz kürsör olan bir program yazarsak, aşağıdaki işlemler olmalıdır.

* Kürsörü tanımla
* Kürsörü aç
* Kürsörle yapılacak işlemleri yap
* Kürsörü kapat
* Kürsörü yok et

Kürsörü açmak

OPEN kürsörAdı

Kürsörü kapatmak

CLOSE kürsörAdı

Kürsörün yok edilmesi(hafızadan kaldırılması)

DEALLOCATE kürsörAdı

Kürsörün İçinde hareket

Aşağıdaki deyim ile kürsör istenilen yere konumlanır.

FETCH …1… FROM …2… INTO …3…

1 nolu yere aşağıdaki kelimeler yazılabilir.

NEXT : bir sonraki kayıta git.

İkinci tip kürsörlerde (SCROLL yazılı)

PRIOR : bir önceki kayıta git

FIRST : ilk kayıta git

LAST : son kayıta git

ABSOLUTE n : n. inci kayıta git.

RELATIVE n : mevcut pozisyondan n kayıt ileri git.

2 nolu yere, içinde hareket edilen kürsörün adı

3 nolu yere, bulunulan pozisyondaki bilgilerin aktarıldığı değişkenler.

Örnek:

Kürsörleri kullanarak, personel tablosunda 4 nolu kayıttaki personeli, adı, soyadı ve sicil numarasını veren kodu yazınız.

DECLARE per\_kursor SCROLL CURSOR FOR SELECT ad, soyad, sicilno FROM Personel

OPEN per\_kursor

DECLARE @ad varchar(30), @soyad varchar(30), @sicilno varchar(20)

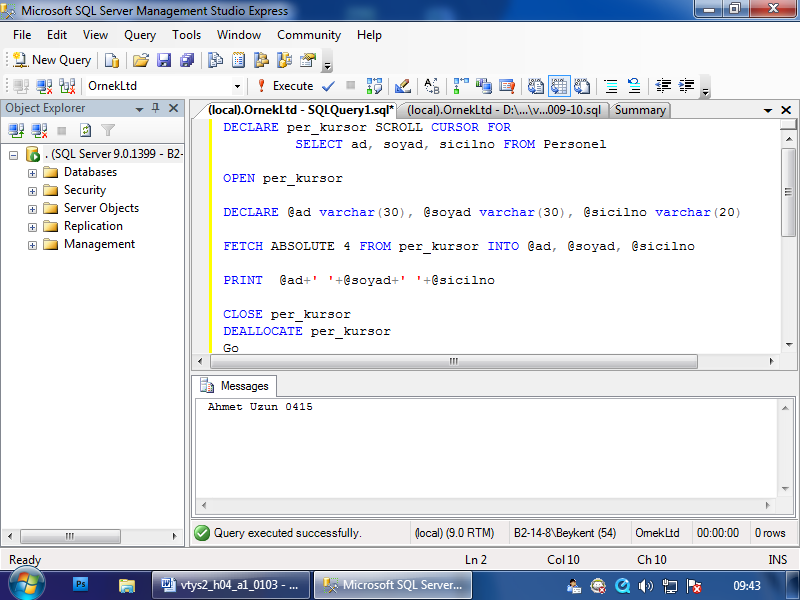
FETCH ABSOLUTE 4 FROM per\_kursor INTO @ad, @soyad, @sicilno

PRINT @ad+’ ‘+@soyad+’ ‘+@sicilno

CLOSE per\_kursor

DEALLOCATE per\_kursor

Go



DECLARE per\_kursor SCROLL CURSOR FOR SELECT ad, soyad, sicilno FROM Personel

OPEN per\_kursor

DECLARE @ad varchar(30), @soyad varchar(30), @sicilno varchar(20)

FETCH ABSOLUTE 4 FROM per\_kursor INTO @ad, @soyad, @sicilno

PRINT @ad+’ ‘+@soyad+’ ‘+@sicilno

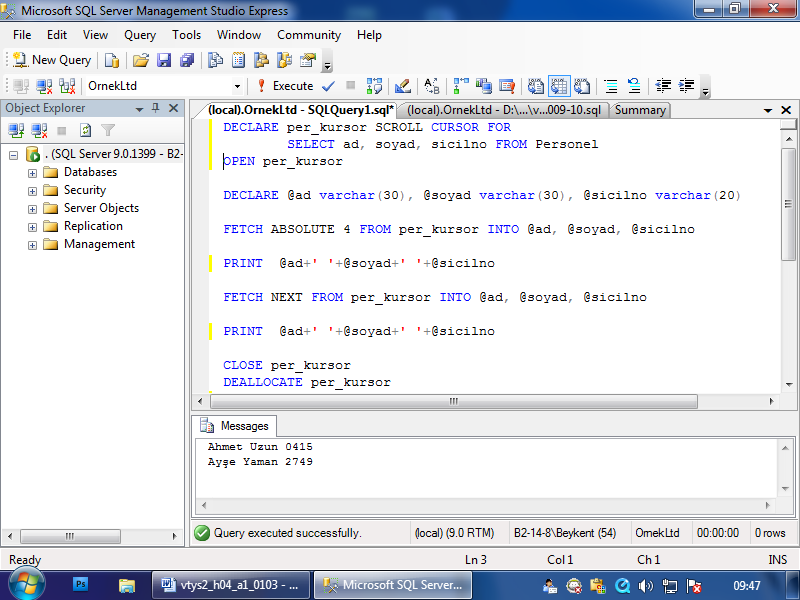
FETCH NEXT FROM per\_kursor INTO @ad, @soyad, @sicilno

PRINT @ad+’ ‘+@soyad+’ ‘+@sicilno

CLOSE per\_kursor

DEALLOCATE per\_kursor

Go



@@FETCH\_STATUS : kürsör içinde istediğim hareket gerçekleşirse değeri sıfır, gerçekleşmezse değeri sıfırdan farklıdır.

Örnek:

Personel tablosundaki bayanların listesini, satır satır yazdırınız. (cinsiyet=2)

DECLARE per\_kursor CURSOR FOR select ad, soyad, sicilno from personel where cinsiyet=2

DECLARE @ad varchar(30), @soyad varchar(30), @sicilno varchar(10)

OPEN per\_kursor

FETCH NEXT FROM per\_kursor INTO @ad, @soyad, @sicilno

WHILE @@fetch\_status = 0

BEGIN

PRINT @ad+’ ‘+ @soyad+’ ‘+ @sicilno

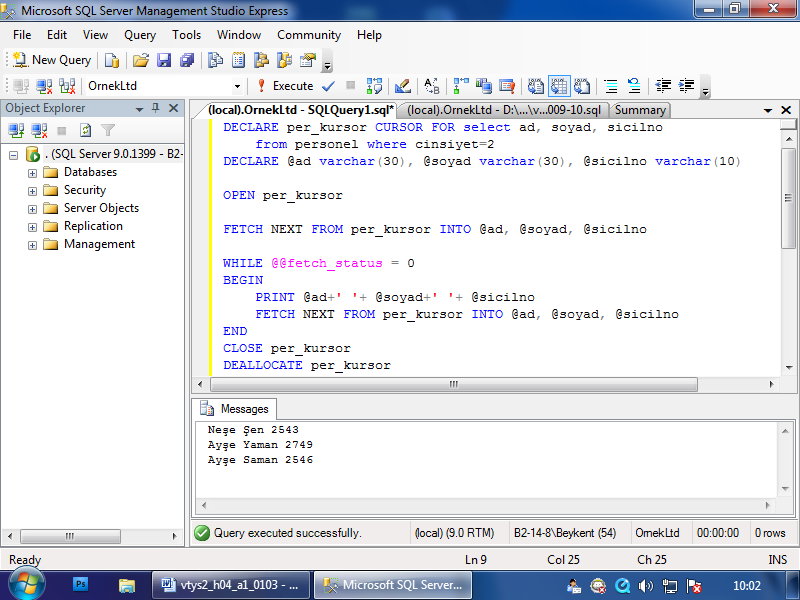
FETCH NEXT FROM per\_kursor INTO @ad, @soyad, @sicilno

END

CLOSE per\_kursor

DEALLOCATE per\_kursor

GO



Soru:

Her müşterinin faturalarını ayrı ayrı listeleyiniz. (kürsör kullanarak)

DECLARE musteri\_kursor CURSOR FOR SELECT mno FROM musteri

DECLARE @mno int

OPEN musteri\_kursor

FETCH NEXT FROM musteri\_kursor INTO @mno

WHILE @@fetch\_status=0

BEGIN

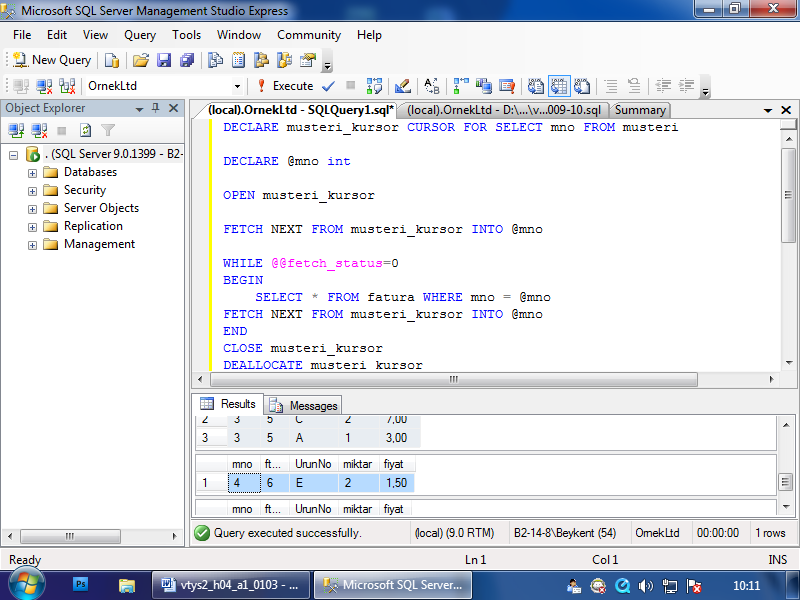
SELECT \* FROM fatura WHERE mno = @mno

FETCH NEXT FROM musteri\_kursor INTO @mno

END

CLOSE musteri\_kursor

DEALLOCATE musteri\_kursor



STORED PROCEDURE (saklı yordam)

Bir stored prosedür oluşturduktan sonra veritabanı sunucusunda saklanır. Her ihtiyaç duyulduğunda defalarca çağrılabilir.

Ağ bazlı uygulamalarda, ağ trafiğini azaltır, sistem performansını artırır. Sunucu bazlı uygulama imkanı verir.

Yapısı ve özellikleri

* parametreli veya parametresiz olabilir.
* İstendiğinde parametreden veri geri dönebilir.
* İstendiğinde parametreye varsayılan değer atanabilir
* İçinde türlü sql deyimi bulunabilir.
* Sunucu tarafında çalışması istenen uygulamalar yapılabilir.

Örnek :

CREATE PROCEDURE sp\_personel @departman int

AS

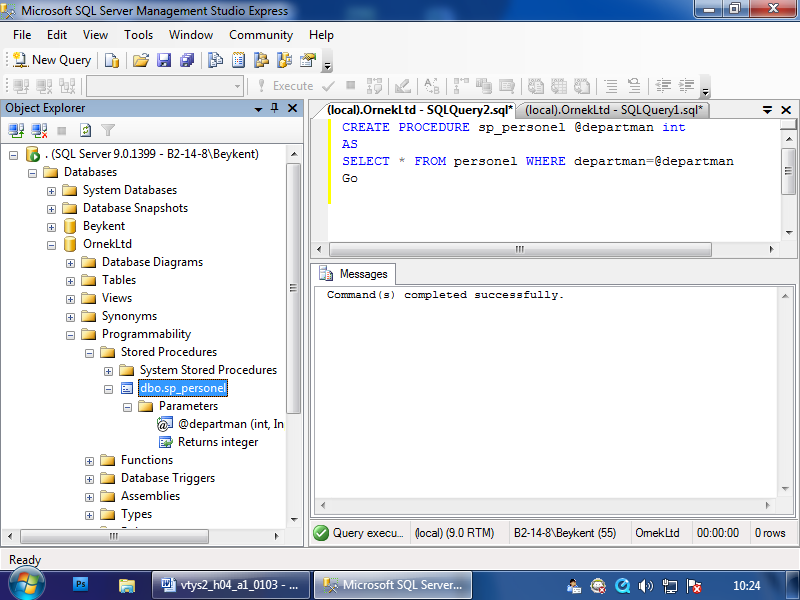
SELECT \* FROM personel WHERE departman=@departman

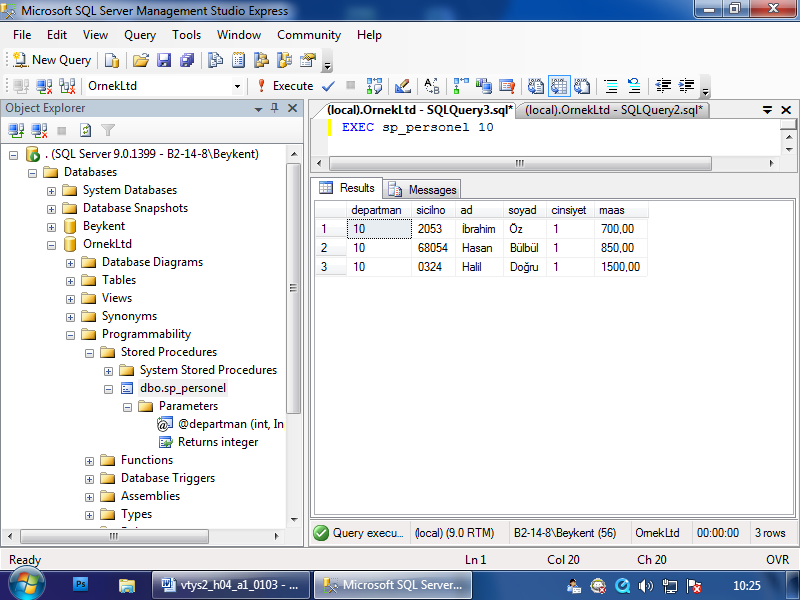
Go

Yukarıda basit bir prosedür var. Bu prosedür, personel tablosunu listeler. @departman isimli parametresi vardır. Bu parametre girilen departmanı listeler.

Yukarıdaki prosedürü çalıştıralım

EXEC sp\_personel 10





Departman değişkenine varsayılan değer verelim.

ALTER PROCEDURE sp\_personel @departman int = NULL

AS

IF @departman is null

SELECT \* FROM personel

ELSE

SELECT \* FROM personel WHERE departman=@departman

Go

