PRINT deyimi

Print deyimile yazı yazdırabiliriz.

Print ‘Kenan KILIÇASLAN’

Print 10

Print ‘Hata’

Print @sayi

Print @ad

Print 10,20 -- yanlış

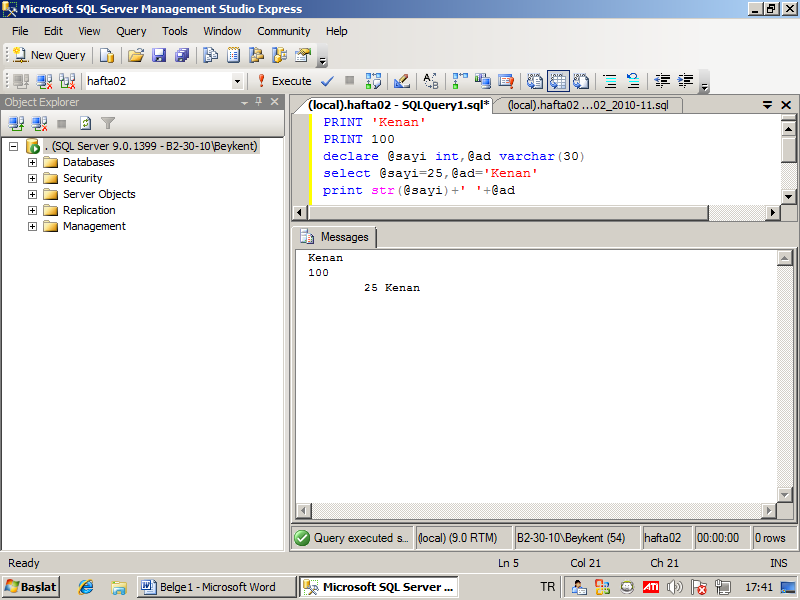
Print @ad,@soyad -- yanlış

@ad ve @soyadı yazmak istersek

Print @ad+’ ’+@soyad -- doğru

Print @ad+@sayi -- yanlış

Print @ad+str(@sayi) – doğru



AKIŞ KONTROL DEYİMLERİ (Devam)

IF … ELSE …

Bu batch içinde kullanılır

Batch: bir dizi sql deyimidir.

C++’daki { … } yerine sql serverda BEGIN … END

IF koşul

Doğru ise çalışacak sql deyimi

IF koşul

BEGIN

Doğru ise çalışacak sql deyimleri

END

IF koşul

Doğru ise çalışacak sql deyimi

ELSE

Koşul yanlış ise çalışacak sql deyimleri

Örnek:

@para isimli değişken tanımlayınız. Bu değişkenin değeri 500 den büyükse personel tablosunu, küçükse satış tablosunu listeyen bir veritabanı programcığı yazınız.

DECLARE @para Money

SELECT @para = 200

IF @para > 500

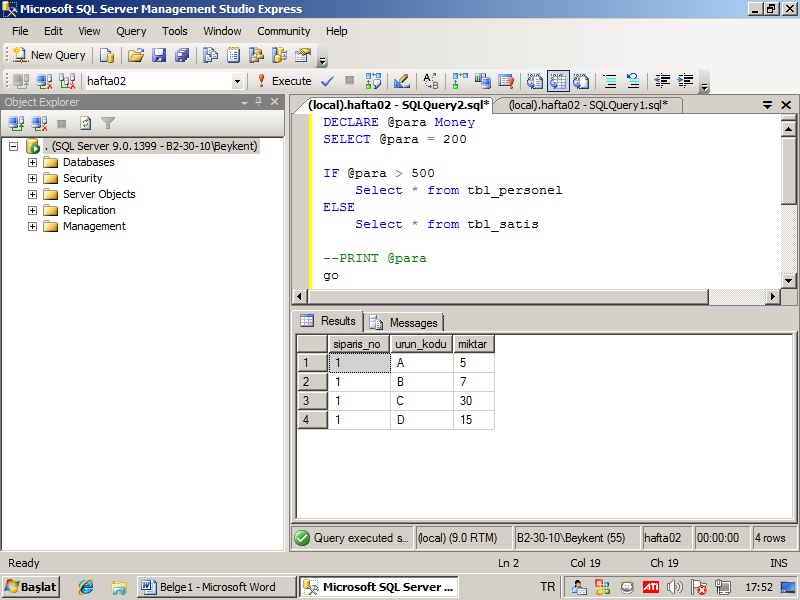
Select \* from tbl\_personel

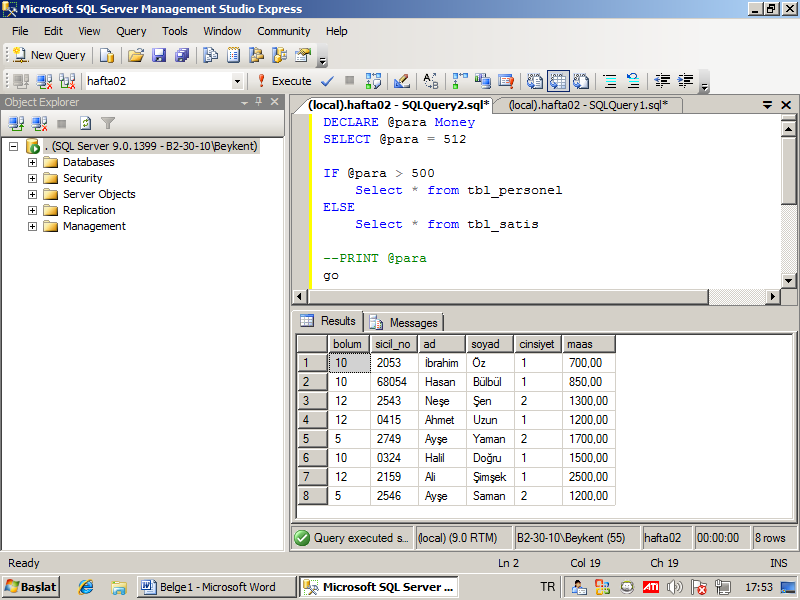
ELSE

Select \* from tbl\_satis

PRINT @para

Go





Örnek:

Personel tablosundaki maaş ortalaması 1000 fazla ise “ortalama maaş yüksek” mesajı versin, az ise “ortalama maaş düşük” mesajı versin

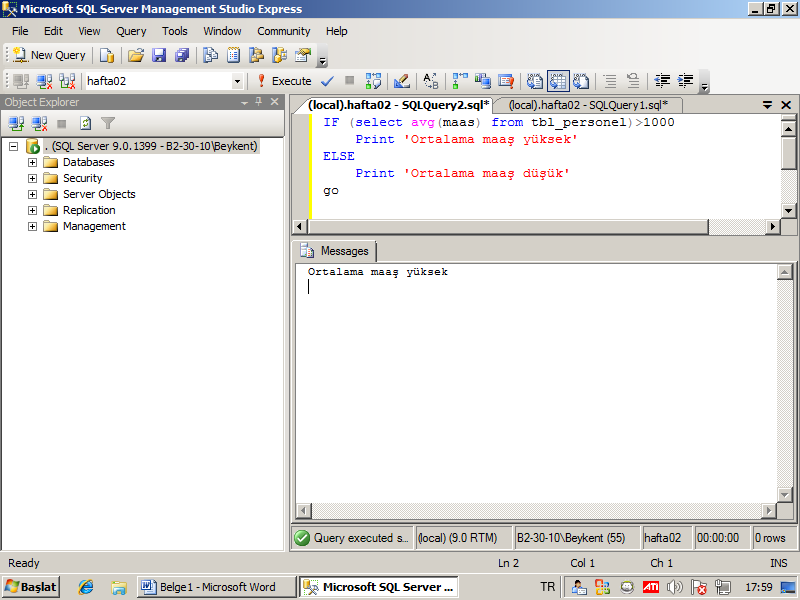
IF (select avg(maas) from tbl\_personel)>1000

Print ‘Ortalama maaş yüksek’

ELSE

Print ‘Ortalama maaş düşük’

go



Yukarıdaki programı aşağıdaki gibi de yazabilirdik.

DECLARE @ortalama Money

SELECT @ortalama = (select avg(maas) from tbl\_personel)

IF @ortalama>1000

Print ‘Ortalama maaş yüksek’

ELSE

Print ‘Ortalama maaş düşük’

go

yukarıda, @ortalama isminde Money tipinde bir değişken tanımladık.

Bu değişkene personel tablosunun ortalamasını atadık. Bu değerin 1000 büyük olup olmamasına göre mesaj verdirdik

TRY…CATCH…

Try…catch … hata kontrolünde kullanılır.

Örnek

BEGIN TRY

BEGIN TRAN

Delete from tbl\_personel where bolum=10

Update tbl\_personel set maas=1000 where bolum=12

Commit tran

END TRY

BEGIN CATCH

ROLLBACK TRAN

END CATCH

Go

Birinci blok try bloğudur. Normalda bu blok çalışır. Diyelim ki “delete from..” satırında hata meydana geldi. Bu komutun devamındaki kodlar çalışmaz. Kodun akışı catch bloğuna girer. Catch bloğundaki komutlar çalışır.

Diyelim ki TRY için hata oluşmadı bu durumda TRY içindeki tüm kodlar çalışır. Hata oluşmadığında CATCH bloğu çalışmaz.

WHILE Döngüsü

SQL deyimlerinin tekrarlanmasını sağlar. Koşul yanlış oluncaya kadar döngü çalışır.

WHILE koşul

Sql deyimi

Veya

WHILE koşul

begin

Sql deyimleri

End

Koşul ise döngüden çıkır, doğru ise döngü devam eder.

Örnek:

1’den 12ye kadar sayıları birer birer yazdırınız.

Declare @sayi int

Select @sayi =1

While @sayi<=12

Begin

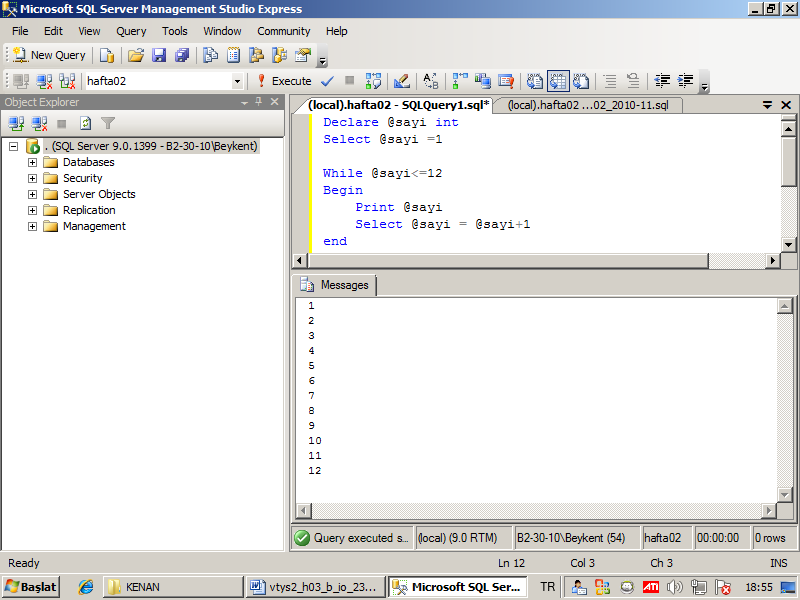
--@sayi değerini yazdır

Print @sayi

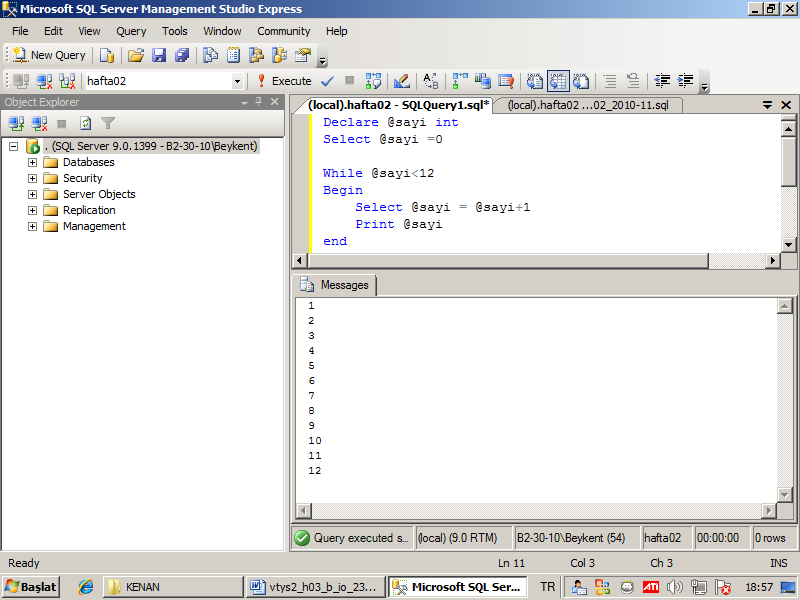
-- sayı değerini bir artır.

Select @sayi = @sayi+1

End



Yukarıdaki kodu aşağıdaki gibi de yazabiliriz.



BREAK: döngü koşulu yerine gelmeden, erken olarak döngüden çıkmayı sağlar

CONTINUE : döngü içinde , sonraki deyimler çalışmadan döngünün başa dönmesini sağlar