PRINT deyimi

Print deyimi ile yazı yazdırılır.

Print ‘Kenan KILIÇASLAN’

Print 10

Print @sayi

Print @tarih

Print @ad,@soyad -- hatalı

Print @ad+’ ‘+@soyad -- doğru

Print 10,20 --yanlış

Print @ad+@sayi -- yanlış (metin ile sayı birleşmez) sayıyı metne çevirmek gerekir.

Print @ad+str(@sayi) -- doğru

Str() : sayıyı metne çeviren sql fonksiyonu



IF… ELSE …

Bu deyim koşula bağlı olarak sql deyimi çalıştırır.

IF koşul

 Doğru ise çalışacak sql deyimi

IF @para>500

 Select \* from personel

IF koşul

 Doğru ise çalışacak sql deyimi

ELSE

 Yanlış ise çalışacak sql deyimi

İf içinde birden fazla sql komutu konacak ise bu komutlar begin…end arasına konur

IF koşul

 BEGIN

 ….

 …

 END

ELSE

 BEGIN

 ….

 ….

 END

IF @para>500

 Select \* from personel --koşula bağlı

Select \* from satis -- if içinde değil

IF @para>500

begin

 Select \* from personel --koşula bağlı

Select \* from satis -- koşula bağlı

End

Örnek

Declare @sayi int

Select @sayi=200

if @sayi>500

 select \* from personel

else

 select \* from satis





Örnek:

Personel tablosunda maaş ortalaması 1000’den büyük ise “maaş ortalaması yüksek”, küçük ise “maaş ortalaması düşük” mesajı versin.

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

 Print ‘Maaş ortalaması yüksek’

ELSE

 Print ‘maaş ortalaması düşük’

Go



Yukarıdaki kodu değişken tanımlayarak yeniden yazalım

DECLARE @ortalama Money

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

IF @ortalama>1000

 Print ‘Maaş ortalaması yüksek’

ELSE

 Print ‘maaş ortalaması düşük’

Print @ortalama

Go



Yukarıdaki kodda

\* @ortalama isimli değişken tanımladık.

\*Bu değişkene personel tablosunun ortalamasını atadık.

Ortalama değişkeninin değerinin 1000’den büyük olup olmadığını kontrol ettik.

Büyükse ‘Maaş ortalaması yüksek’, değilse

‘maaş ortalaması düşük’ mesajı verdirdik ve @ortalama değişkenini yazdırdık.

TRY…CATCH… bloğu

Try…catch .. hata kontrolünde kullanılır. Sql 2005’den sonraki sürümlerde vardır. Bunun alternatifi @@error değişkenidir.

Blok yapısı

BEGIN TRY

 ….

 …. (hata yokken çalışan kodlar)

 …

END TRY

BEGIN CATCH

 … (hata oluştuğunda çalışan kodlar)

END CATCH

Çalışma mekanizması

Try bloğu içinde herhangi satırda hata oluştuğunda devamındaki satırlar çalışmadan, program akışı catch bloğuna geçer. Try içinde hata yoksa try bloğu sonuna kadar çalışır ve catch bloğu içine girilmez.

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



WHILE döngüsü

While deyimi bir sql deyiminin yinelenmesini sağlar

WHILE koşul

 SQL deyimi

Veya

WHILE koşul

begin

 SQL deyimleri

End

Koşul doğru olduğu sürece döngü çalışır.

Örnek:

1 den 12’ye kadar sayıları birer birer yazdırınız.

Declare @sayi int

Select @sayi=1

While @sayi<=12

Begin

 Print @sayi

 Select @sayi=@sayi+1

End

