Ankara Üniversitesi
Elmadağ Meslek Yüksekokulu

EBP 242 Veri Tabanı II

HATA DENETİMİ, TRY-CATCH YAPILARI

ÖĞR. GÖR. DR. YUNUS KÖKVER

GOTO İfadesi

GOTO ifadesi kullanıcı tarafından tanımlanan etiketler için koşulsuz dallanmayı sağlar. GOTO ifadesi kullanıldığında kodların akışı belirtilen etiketten sonra devam eder.

Kullanım Şekli:

Etiket_adı:

…

GOTO etiket_adı

Örnek:

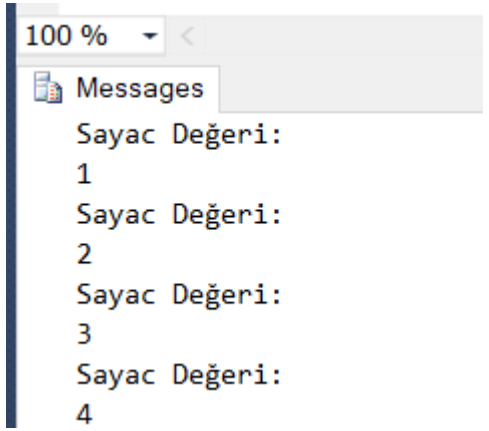
1’den 4’e kadar olan rakamları yazdıralım.

```
DECLARE @sayac int
SET @sayac=1

yenile:
PRINT 'Sayac Değeri: '
PRINT @sayac

SET @sayac=@sayac+1

WHILE @sayac <=4
GOTO yenile
```



RETURN İfadesi

RETURN ifadesi hazırlanan T-SQL kodlarının çalışmasını sonlandırır.

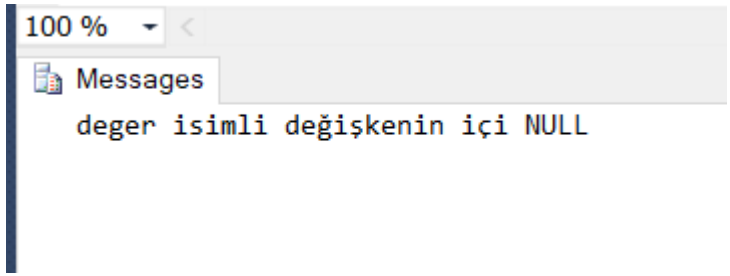
Örnek:

```
DECLARE @deger int
DECLARE @deger2 int
SET @deger2=23

if @deger is null
Begin
Print 'deger isimli değişkenin içi NULL'
RETURN
End

Else
Print @deger

if @deger2 is not null
print @deger2
```



WAITFOR İfadesi

Hazırlanan kod bloğunun çalışmasını belirlenen bir zamana kadar veya belirlenen süre kadar bekletir.

DELAY veya TIME parametreleriyle ve saat, dakika veya saniye birimleriyle kullanılır.

Delay: Parametre olarak verilen zaman kadar sonra çalıştırır.

Time: Parametre olarak verilen zamanda çalıştırır.

Kullanım Şekli:

WAITFOR DELAY/TIME ‘saat:dakika:saniye’

WAITFOR TIME ’12:15:00’ → Kodlar saat 12:15’de çalıştırılır.

WAITFOR DELAY ’12:15:00’ → Kodlar 12 saat 15 dakika sonra çalıştırılır.

Örnek:

```
WAITFOR DELAY ’00:00:10’

SELECT * FROM  personel
```

Kodlar çalışmaya başladıktan 10 saniye sonra personel tablosundaki kayıtları listeletecektir.

T-SQL HATA DENETİMİ

Yazılan SQL ifadeleri çalıştırıldığında oluşan hatalar kayıt altına alınabilir.

Hata mesajları ve kodlarının tamamı sys.messages sistem tablosunda tutulur.

SELECT * FROM sys.messages ifadesi ile SQL üzerinde tanımlı olan hata mesajları ve anlamları incelenebilir.

Bu hatalar aynı zamanda @@ERROR sistem fonksiyonu altında tutulmaktadır.

@@ERROR ile elde edilen hata kodunun karşılığı sys.messages tablosundan bulunabilir.

```
SELECT * FROM sys.messages
```

100 %

Results | Messages

	message_id	language_id	severity	is_event_logged	text
1	21	1033	20	0	Warning: Fatal error %d occurred at %S_DATE. Not...
2	101	1033	15	0	Query not allowed in Waitfor.
3	102	1033	15	0	Incorrect syntax near '%.*ls'.
4	103	1033	15	0	The %S_MSG that starts with '%.*ls' is too long. Max...
5	104	1033	15	0	ORDER BY items must appear in the select list if the...
6	105	1033	15	0	Unclosed quotation mark after the character string '...
7	106	1033	16	0	Too many table names in the query. The maximum ...
8	107	1033	15	0	The column prefix '%.*ls' does not match with a table...
9	108	1033	15	0	The ORDER BY position number %ld is out of range...
10	109	1033	15	0	There are more columns in the INSERT statement t...
11	110	1033	15	0	There are fewer columns in the INSERT statement t...
12	111	1033	15	0	'%ls' must be the first statement in a query batch.
13	112	1033	15	0	Variables are not allowed in the %ls statement.
14	113	1033	15	0	Missing end comment mark '*/'.
15	114	1033	15	0	Browse mode is invalid for a statement that assigns ...

@@ERROR yalnızca en son oluşan hata kodunu tutar.
Bu nedenle her hatayı görebilmek için her sorgu sonuna @@ERROR fonksiyonu eklenmelidir.

@@ERROR sistem fonksiyonu ile yapılan hata denetimlerinde her SQL ifadesinden sonra hata denetimi yapılmalıdır.

```
DELETE FROM KITAP_YAZAR WHERE yazar_no=2
DELETE FROM YAZARLAR WHERE yazar_no=2

IF @@ERROR <> 0
PRINT (‘Hata Oluştu’)
```

Mantıksal olarak hatalıdır. Her SQL ifadesinden sonra if yapısı ile hata kontrolü yapılmalıdır.

TRY- CATCH YAPISI

TRY
bloğu içerisinde çalıştırılacak SQL ifadeleri
CATCH
bloğu içerisindeyse hata oluşması durumunda çalıştırılacak kodlar bulunur.

TRY bloğu içerisinde hata yoksa CATCH bloğundaki kodlar es geçilecek ve devamında gelen kodlar işleme alınacaktır.

Kullanım Şekli:

BEGIN TRY
SQL ifadeleri
END TRY

BEGIN CATCH
Hata durumunda çalıştırılacak kod bloğu
END CATCH

Oluşan hata hakkında ayrıntılı bilgi almak için aşağıdaki sistem fonksiyonları kullanılır:

ERROR_NUMBER (): Hatanın kodunu verir.

ERROR_SEVERITY (): Hatanın önem derecesini verir.

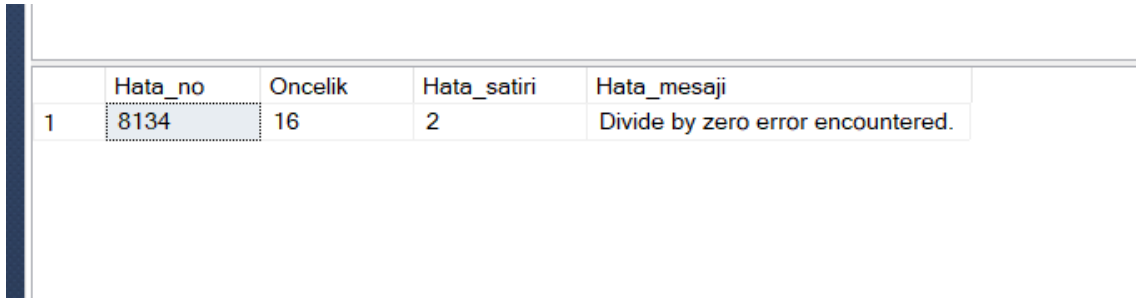
ERROR_LINE () : Hatanın satır numarasını verir.

ERROR_MESSAGE(): Hatanın tamamını verir.

Örnek:

1 değerinin 0'a bölünmesi işleminde hata denetimi işlemi

```
BEGIN TRY
SELECT 1/0
END TRY
BEGIN CATCH
SELECT
ERROR_NUMBER () AS Hata_no,
ERROR_SEVERITY() AS Oncelik,
ERROR_LINE()  AS Hata_satiri,
ERROR_MESSAGE() AS Hata_mesaji
END CATCH
```



	Hata_no	Oncelik	Hata_satiri	Hata_mesaji
1	8134	16	2	Divide by zero error encountered.

Her türlü hata TRY… CATCH ifadesi ile yakalanamaz.

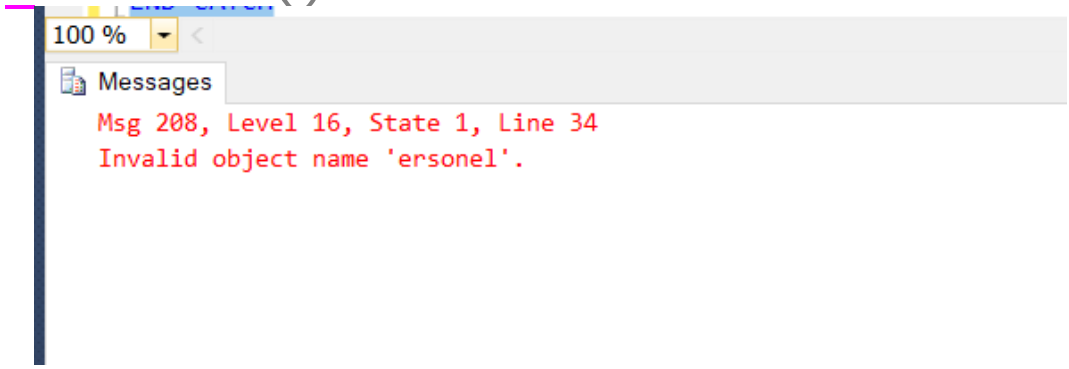
Verilen öncelik değerleri sys.messages tablosunda bulunan **severity** sütunu ile tutulmaktadır. Bu hata mesajının bir öncelik değeri vardır.

TRY… CATCH ile Tespit Edilemeyen Hatalar:

- 10 veya daha düşük öncelik değerine sahip uyarı ve bilgi mesajları.
- SQL Server bağlantısı sorunlu olduğunda 20 veya daha fazla öncelik değerine sahip hata mesajları.
- İstemci bağlantısı koptuğunda oluşan hata mesajları.
- Yönetici tarafından kullanıcı oturumu sonlandırıldığında oluşan hatalar.
- Derleme veya yazım hataları.

Verilen SQL ifadesi ile personel tablosunun içeriği listelenmek istenmiştir ama ‘personel’ yerine ‘ersonel’ yazıldığı için hata oluşacaktır. Bu tür hata TRY-CATCH ile yakalanamaz.

```
BEGIN TRY
SELECT * FROM ersonel
END TRY
BEGIN CATCH
PRINT 'Hata Mesajı: '+ERROR_MESSAGE ()
END CATCH
```



DİNAMİK SQL İFADELERİ HAZIRLAMAK

Değişkenlerden veya diğer SQL ifadelerinden elde edilen değerlere göre hazırlanan SQL ifadeleri dinamik SQL ifadeleri olarak adlandırılır.

Dinamik olarak hazırlanan SQL ifadelerini çalıştırmak için **EXECUTE** fonksiyonu kullanılır.

```
DECLARE @tablo VARCHAR(20)
        SET @tablo='KITAPLAR'
        EXECUTE('SELECT * FROM ' + @tablo)
```

veya

```
        DECLARE @tablo VARCHAR(20)
        DECLARE @sorgu VARCHAR(MAX)
        SET @tablo='KITAPLAR'
        SET @sorgu= 'SELECT * FROM ' + @tablo
        EXECUTE (@sorgu)
```

Ders Notu Hazırlanırken Kullanılan Kaynaklar

Ramakrishnan, R. and Gehrke J., Database Management Systems (Third Edition), WCB/McGraw Hill, ISBN: 0-07-232206-3

Veritabanı Yönetim Sistemleri II (2019)., ÖZSEVEN TURGUT, Ekin Yayınevi, Sayfa Sayısı: 351, Türkçe(Ders Kitabı), (Yayın No: 32619)