



YALOVA ÜNİVERSİTESİ
university of yalova

Öğr. Gör. Cansu AYVAZ GÜVEN

VERİTABANI-II

TRY-CATCH
TRANSACTION

SQL Server Hata Yakalama Mekanizması (TRY-CATCH)

- Programlama dillerinde olduğu gibi t-sql de de try catch blokları oluşan hatayı yakalamamızı sağlar ve oluşan hata hakkında bilgi sahibi olmamıza imkan verir.
- **Hata yakalamak için TSQL'de aşağıdaki örnek syntax yapısını kullanırız.**

BEGIN TRY

Sql ifadeleri..

END TRY

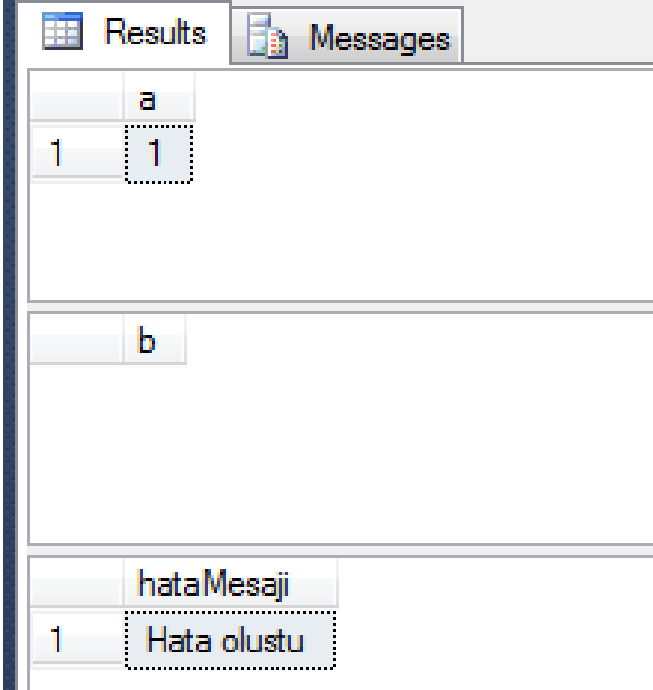
BEGIN CATCH

Hata oluştuğunda yapılması istenen sql ifadeleri..

END CATCH

Örnek;

```
BEGIN TRY
SELECT a = 1
SELECT b = 1/0
SELECT c = 3
END TRY
BEGIN CATCH
SELECT 'Hata olustu' hataMesaji
END CATCH
```



The screenshot shows the SQL Server interface with two windows: Results and Messages. The Results window displays the output of the first two SELECT statements, and the Messages window displays the error message 'Hata olustu'.

	a
1	1

	b
--	---

	hataMesaji
1	Hata olustu

TRY...CATCH kullanıldığında, bir hata oluştuğu zaman bu hata istemciye geri döndürülmez, hata CATCH bloğu içinde işlenir.

Hata Fonksiyonları

```
- BEGIN TRY
  SELECT Sonuc = 5/0
END TRY
BEGIN CATCH
- SELECT [Hata_Satiri] = ERROR_LINE(),
  [Hata_Sayisi] = ERROR_NUMBER(),
  [Hata_Onemi] = ERROR_SEVERITY()
  SELECT [Hata_Mesaji] = ERROR_MESSAGE()
END CATCH
|
```

Sonuç

Results		Messages	
Sonuc			
	Hata_Satin	Hata_Sayisi	Hata_Onemi
1	2	8134	16
Hata_Mesaji			
1	Divide by zero error encountered.		

CATCH bloğunda kullanılan fonksiyonlar ve açıklamaları şu şekilde;

- ❑ **ERROR_LINE**: Hatanın hangi satırda oluştuğunu gösterir.
- ❑ **ERROR_NUMBER**: Oluşan hata numarası
- ❑ **ERROR_SEVERITY**: Hatanın önem düzeyini gösterir. CATCH bloğu önem seviyesi 11 ve üzerinde olan hataları yakalar. 11 - 16 arasında olan hatalar kullanıcı veya kod hatasıdır. 17-25 arasında olan hatalar ise yazılım ve donanımdan kaynaklanan hatalardır.
- ❑ **ERROR_MESSAGE**: Oluşan hataya ait hata mesajı.

Derleme veya yazım hataları TRY...CATCH ile yakalanamaz.

```
BEGIN TRY
SELECT * FROM tbl_der -- tbl_der adında tablo yok (tbl_ders)
END TRY
BEGIN CATCH
PRINT 'Hata Mesaji: ' + ERROR_MESSAGE()
END CATCH
```

00 %

Messages

```
Msg 208, Level 16, State 1, Line 2
Invalid object name 'tbl_der'.
```

Transaction

TRANSACTION

- Bir transaction bir veya daha fazla SQL ifadesinden meydana gelen tek bir işlemdir. SQL ifadelerinin tamamı bir bütün olarak düşünülür ve daha küçük İş parçalarına ayrılamaz. Özellikle ardı ardına gelen ve birbirleriyle bağımlı birden fazla işlemin tek bir İşlem olarak kullanılmasını sağlar.
- Bir transaction içerdiği SQL ifadelerini tamamını veya hiçbirini gerçekleştirir. SQL ifadelerini bir bölümünün uygulanması gibi bir İşlem söz konusu olamaz. İşlemlerin tamamı gerçekleşmediği sürece işlemlerin hiçbiri gerçekleşmemiş sayılır.

TRANSACTION

- Bir müşteri kendi hesabından başka bir hesaba 1500 TL para transferi gerçekleştiriyor. Önce kişinin hesabından transfer edilecek olan 1500 TL tutar düşülmelidir. Sonra diğer kişinin hesabına transfer edilecek olan 1500 TL tutar eklenmelidir.
- Bu işlemlerden birincisi gerçekleştikten sonra herhangi bir sorundan dolayı İkinci işlem gerçekleşmezse hesaplarla ilgili ciddi sorunlar yaşanabilir. Gönderenin hesabından 1500TL tutar düşer ama alıcının hesabına bu para aktarılamaz.

TRANSACTION

- Bu tür sorunları Önlemek için transaction yapılan kullanılır.
- Transaction her iki işlemi de tek bir işlem olarak ele alacağı için herhangi birisi gerçekleşmediği zaman diğer gerçekleşen işlemleri de yok sayacaktır. Yani gerçekleşen işlemi geri alacaktır(rollback). Eğer işlemlerin tamamı sorunsuz bir şekilde gerçekleşirse, tüm İşlemleri kalıcı(commit) hale getirecektir.
- Transaction yapıları transaction logları adı verilen yöntemi kullanır.

Örnek Tablo



Results



Messages

	HesapID	Adı	Soyadı	SubeKodu	Bakiye
1	284598	Hakan	Kul	78	10000,00
2	284599	Selvi	Naz	97	50,00

```
= CREATE PROC sp_ParaTransfer(  
    @GonderenID CHAR(10),  
    @AliciID CHAR(10),  
    @Tutar MONEY  
)  
AS  
= BEGIN  
    = BEGIN  
        = BEGIN TRANSACTION  
            = UPDATE hesaplar  
                SET Bakiye = Bakiye - @Tutar  
                WHERE HesapID = @GonderenID  
            = IF @@ERROR <> 0  
                ROLLBACK  
            = UPDATE hesaplar  
                SET Bakiye = Bakiye + @Tutar  
                WHERE HesapID = @AliciID  
            = IF @@ERROR <> 0  
                ROLLBACK  
            COMMIT  
        = END  
    = END  
= END;
```

Sonuç;

```
exec sp_ParaTransfer 284598,284599,5000
```

```
select * from hesaplar
```

100 %



Results



Messages

	HesapID	Adı	Soyadı	SubeKodu	Bakiye
1	284598	Hakan	Kul	78	5000,00
2	284599	Selvi	Naz	97	5050,00