SQL – TRIGGERS (Tetikleyiciler)

Trigger kelime anlamı olarak tetikleyici demektir. Trigger, SQL'de bir sorgu çalıştırdığımızda, başka bir sorgunun çalışmasını sağlamak için yazılan koddur. Mesela bir tablodan kayıt **sildiğimizde**, başka bir tablodan onunla ilgili bir kaydın silinmesi için kullanabiliriz. Yada bir tabloya **ekleme** yaptığımızda başka bir tabloya da onunla ilgili olan bilgileri eklemek için kullanabiliriz.

Ardı Sıra Tetikleyiciler (After Triggers)
Bir tabloya **UPDATE**, **INSERT** veya **DELETE** işlemi yapıldıktan sonra bir takım işlemlerin yapılması için kullanılan tetikleyicilere **Ardı Sıra Tetikleyici** denir. Bu tür tetikleyiciler pek çok değişik iş yapabilirler. Bir başka tabloya veri girişi yapmak veya tabloyu güncellemek, tablolar arasında uyumu sağlamak için bu tür tetikleyiciler çok uygundur.

Yerine Tetikleyiciler(INSTEAD OF TRIGGER)
Bir INSERT, UPDATE veya DELETE işlemi bir tabloya uygulandığında bu tablo üzerinde , sırasıyla bir Instead Of INSERT, Instead Of UPDATE veya Instead Of DELETE tetikleyici varsa bu işlem tablo üzerinde **gerçekleşmez**. Onun yerine tetikleyici içinde yazılı kodlar yapılır.

by yselim

SQL – TRIGGERS (Tetikleyiciler)

4.3. Tetikleyici Türleri

SQL Server'da iki farklı tür tetikleyici vardır. Bunlar **After** ve **Instead Of** tetikleyicileridir.

4.3.1. After Tetikleyicileri

After tetikleyicileri, kendiyle ilişkili işlem gerçekleştikten hemen sonra ateşlenir. Veritabanındaki temel işlemler için (ekleme, silme ve güncelleme) **After** tetikleyicileri tanımlanabilir. Örneğin, yeni bir personel kaydı silindiğinde farklı tablolarda o personele ait diğer bilgilerin silinmesi **After Delete** tetikleyicisi ile sağlanabilir. Birden fazla tetikleyici, bir iş için tanımlanabildiği gibi, bir tetikleyici de birden fazla iş için tanımlanabilir. **After** tetikleyicileri, sadece tablolar için tanımlanabilir.

4.3.2. Instead Of Tetikleyicileri

Instead of tetikleyicileri, belirlenen işlem gerçekleşirken devreye girer ve kendi içinde tanımlanan komutları icra etmeye başlar. Yani, belirlenen işlemin yerine geçer. **Instead of** tetikleyicileri işlemlerin arasına girebildiğinden kontrol amaçlı kullanılabilirler. Örneğin, ekleme işlemi için tanımlanan bir tetikleyici, ekleme işleminden hemen önce araya girerek uygun koşulların sağlanıp sağlanmadığını kontrol edebilir. Tıpkı **After** tetikleyicileri gibi temel veritabanı işlemleri için **Instead Of** tetikleyicileri tanımlanabilir. Fakat, **After**

SQL – TRIGGERS (Tetikleyiciler)

CREATE TRIGGER'i kullanabilmek için sysadmin, db_owner veya db_ddladmin rolüne sahip olmak gerekir.

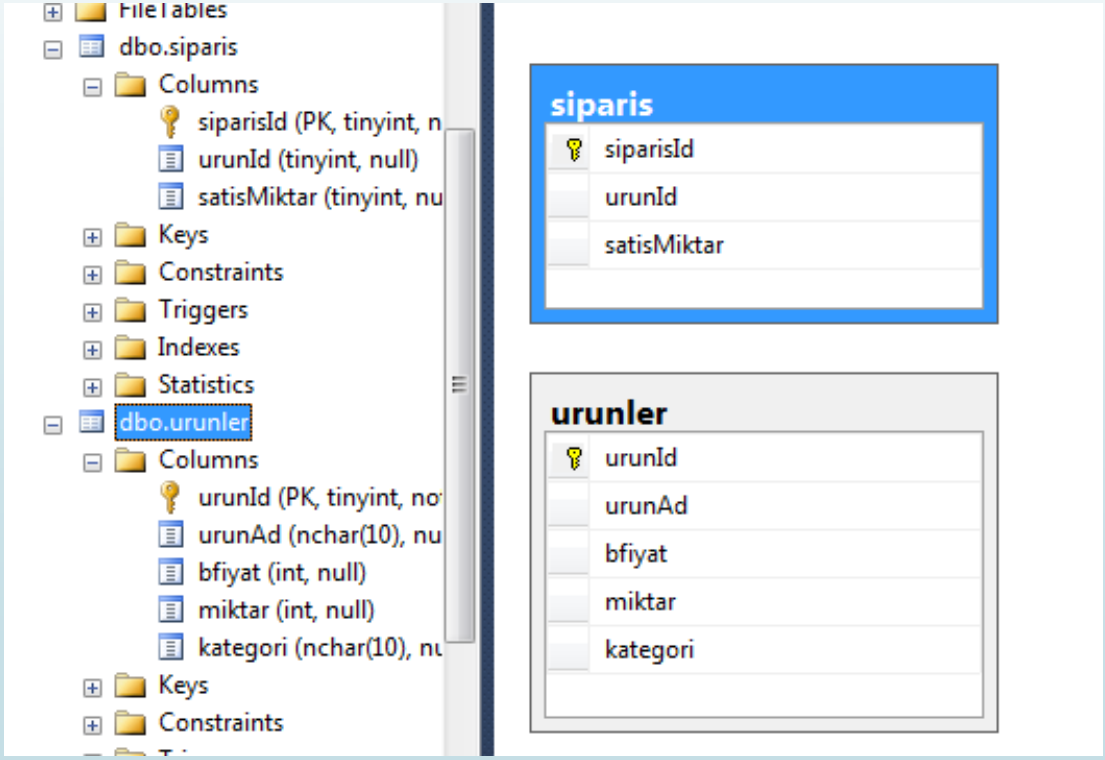
Bir tetikleyici (trigger) başka bir tabloya erişecekse bu tablo için de tetikleyici oluşturan kullanıcının erişim izni veya güncelleme izni olması gerekir.

Yazılışı

CREATE TRIGGER tetikleyici_adi
ON tablo_adi
FOR veya **AFTER** veya **INSTEAD OF** (**INSERT** veya **UPDATE** veya **DELETE**)
AS Sql ifadeleri

SQL – TRIGGERS (Tetikleyiciler)

Aşağıdaki şekilde verilmiş veritabanı için tetikleyici uygulaması yapalım.
Ürünlerle ilgili bir sipariş geldiğinde yani sipariş tablosuna bir insert olduğunda sipariş edilen miktarı otomatik olarak 'urunler' tablosundaki 'miktar' alanından çıkaracak tetikleyiciyi yazalım

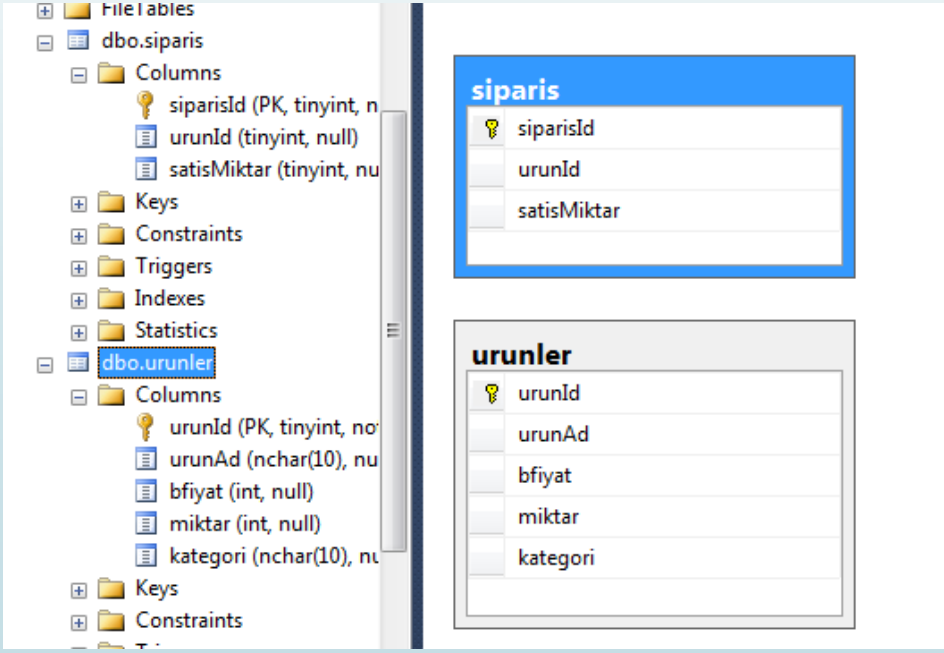


by yselim

SQL – TRIGGERS (Tetikleyiciler)

Sipariş tablosuna yeni bir kayıt girildiğinde **(insert trigger)**, ürünler tablosundaki ilgili ürünün stok miktarından satış miktarı düşülecek.

- Tetikleyici sipariş tablosu için yazılacak.
- urunId'si girilmiş sipariş edilen ürün için satış miktarı , urunler tablosundaki miktar alanından düşülecek.

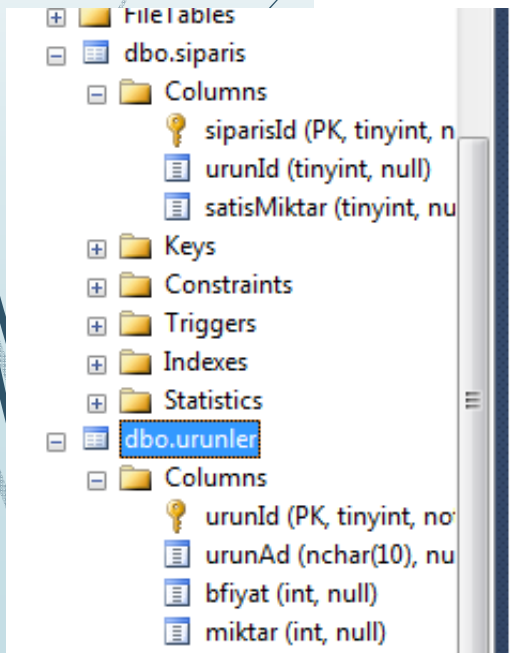


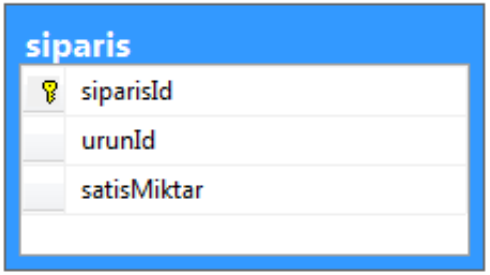
by yselim

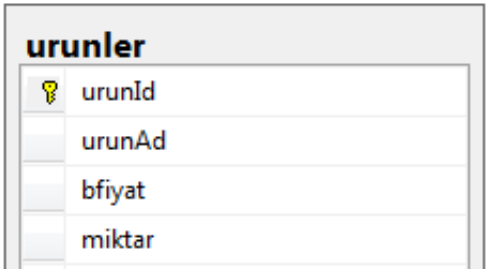
SQL – TRIGGERS (Tetikleyiciler)

Urunler tablosu

	urunId	urunAd	bfiyat	miktar	kategori
1	1	cetvel	120	800	kırtasiye
2	2	kalem	50	2500	sarf
3	3	defter	250	500	kırtasiye
4	4	kartuş	400	40	sarf







Siparis tablosu

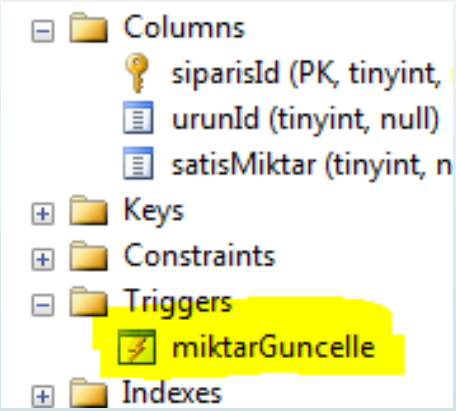
	siparisId	urunId	satisMiktar
1	1	2	100
2	2	1	200
3	3	3	50
4	4	1	100
5	5	1	100
6	6	4	10

by yselim

SQL – TRIGGERS (Tetikleyiciler)

Insert için trigger kodu;
Sipariş verildiğinde sipariş verilen miktarı stoktan düşecek insert trigger'ı yazınız

```
create trigger miktarGuncelle on siparis
for insert as
begin
declare @satis int
declare @urunid tinyint
select @satis=satisMiktar,@urunid=urunId from inserted
update urunler set miktar=miktar-@satis where urunId=@urunid
End
insert into siparis(11,1,100)--- miktarGuncelle çalışır
```



by yselim

SQL – TRIGGERS (Tetikleyiciler)

Trigger oluşturulduktan sonra , sipariş tablosuna yeni bir kayıt girilip , urunler tablosu listelendiğinde stoktaki miktarın satış miktarı kadar azaldığı görülmektedir.

```
select * from urunler

insert into siparis values(5,1,100)

select * from urunler
```

Results | Messages

	urunId	urunAd	bfiyat	miktar	kategori
1	1	cetvel	120	900	kırtasiye
2	2	kalem	50	2500	sarf
3	3	defter	250	500	kırtasiye
4	4	kartuş	400	600	sarf

	urunId	urunAd	bfiyat	miktar	kategori
1	1	cetvel	120	800	kırtasiye
2	2	kalem	50	2500	sarf
3	3	defter	250	500	kırtasiye
4	4	kartuş	400	600	sarf

by yselim

SQL – TRIGGERS (Tetikleyiciler)

4.4.2. DELETE Tetikleyicisi

Tablodan bir kayıt silindiğinde otomatik olarak yapılması istenen işlemler için DELETE tetikleyicisi kullanılır. DELETE tetikleyicisi çalıştıktan sonra silinen kayıt Deleted sahte tablosuna kaydedilir. Deleted tablosunun Inserted tablosundan farkı, asıl tablodan silinen kayıt artık Deleted tablosunda yer almaktadır.

SQL – TRIGGERS (Tetikleyiciler)

```
create trigger sil on siparis
for delete as
begin
declare @satir int
select @satir=count(*) from deleted
if(@satir>3)
begin
print cast(@satir as nvarchar(4))+' adet silme!!!'
raiserror('fazla miktarda silme olmaz',16,1)
rollback transaction
end
End

Delete from urunler
```

SQL – TRIGGERS (Tetikleyiciler)

4.4.3. UPDATE Tetikleyicisi

Tablo üzerindeki kayıt ya da kayıtlarda güncelleme olduğunda devreye girecek olan tetikleyicidir. INSERT ve DELETE tetikleyicilerden biraz farklıdır. Farkı ise UPDATE tetikleyici devreye girdiğinde Inserted sahte tablosu asıl tablodaki kayıtlardan, düzenlenmiş kayıtların kopyasını, Deleted sahte tablosu ise kayıtların düzenleme işleminden önceki hâllerini tutar.

SQL – TRIGGERS (Tetikleyiciler)

Örnek

"tablo_Kitap" tablosunda bulunan kitap fiyatlarından kaçının fiyatının değiştiğini ulacak bir UPDATE tetikleyici yazdığınızı varsayınız.

Bunun için "tablo_Kitap" tablosuna kitap fiyatlarının tutulacağı "Kitap_Fiyat" adında e int tipinde bir sütun oluşturarak sütun içerisine kitap fiyat bilgilerini giriniz.

Kitap_Id	Kitap_Ad	Kitap_Yazar	Kitap_Sayfa	Kitap_Fiyat
123	Visual Basic.NET	Ali CAN	550	45
128	C#.Net	Veli BAL	700	35
150	Access XP	Osman GÜR	450	25
200	VB 6.0 Uygulam...	Sena PINAR	330	12
250	C++ Uygulamaları	Yasemin ZEKİ	270	12
251	Delphi 7.0	Yağmur AK	310	20
NULL	*NULL*	*NULL*	*NULL*	*NULL*

SQL – TRIGGERS (Tetikleyiciler)

```
CREATE TRIGGER indirim
ON tablo_Kitap
FOR UPDATE
AS
raiserror ('%d kayıt üzerinde değişiklik yapılmıştır',0,1,@@rowcount)
RETURN
```

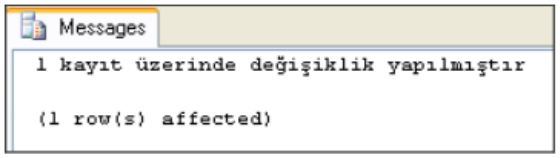
Resim 4.10: UPDATE tetikleyicisi örneği

Bu tetikleyicide kullanılan RAISERROR fonksiyonu, SQL Server'da hata meydana geldiğinde hata mesajı vermek için kullanılır. Burada ise sadece mesaj verme işlemi için kullanılmıştır. @@rowcount ise SQL Serverde en son yapılan işlemden kaç kaydın etkilendiğini verir.

```
UPDATE tablo_Kitap
SET Kitap_Fiyat=Kitap_Fiyat-5
WHERE Kitap_Id>250
```

Resim 4.11: Tetikleyiciyi ateşleyecek SQL kodları

Execute işleminden sonra Messages penceresinde sorgu sonucu kaç kaydın bu mden etkilendiği görüntülenecektir.



SQL – TRIGGERS (Tetikleyiciler)

Aşağıdaki tetikleyicileri istenen şekilde hazırlayıp test ediniz

- 1-Ürünler tablosuna kayıt girildiğinde eğer ürün miktarı 50 ve altı ise kategori alanını otomatik olarak 'sarf' olarak atasın
- 2-Verilen bir sipariş için ödenecek tutarı hesaplattırıp yazdıracak bir tetikleyici yazınız.
  Tutar=bfiyat*satisMiktar
- 3-Sipariş miktarı 500 den büyük tutarlı ürünlerin tutarı hesaplanırken tutar üzerinden %10 indirim yapılarak tutar yazdırılsın.
- 4-Ürün tablosundan 1 nolu ürün miktarı 500'un altına indiğinde birim fiyatını yarıya indiren tetikleyiciyi yazınız
- 5-Sipariş edilen miktar stok miktarını aşıyorsa hata verdiren trigger'ı yazınız
- 6-Sipariş listesinden silinen bir ürün için sipariş miktarı kadar değeri urunlerdeki miktara geri iade eden trigger'ı yazınız.
- Birden fazla kayıt aynı anda silinemez

by yselim

SQL – TRIGGERS (Tetikleyiciler)

```
create trigger miktarGuncelleKontrol on siparis
for insert as
begin
declare @satisMiktar int
declare @urunid tinyint
declare @stokMiktar int
select @satisMiktar=satismiktar from siparis
select @stokMiktar=miktar from urunler where urunId in(select urunId
from inserted)
if(@stokMiktar>@satisMiktar)
begin
select @satisMiktar=satisMiktar,@urunid=urunId from inserted
update urunler set miktar=miktar-@satisMiktar where urunId=@urunid
end
else
begin
rollback transaction
raiserror('girilen sipariş miktarı fazla',16,1)
end
end
```

by yselim

SQL – TRIGGERS (Tetikleyiciler)

Test etmek için;

```
insert into siparis values(6,4,10)
HATA YOK!!
```

```
insert into siparis values(7,4,100)
HATA!!!
Msg 50000, Level 16, State 1, Procedure miktarGuncelleKontrol, Line 17
girilen sipariş miktarı fazla
```

SQL – TRIGGERS (Tetikleyiciler)

4.4.4. INSTEAD OF Tetikleyicisi

INSTEAD OF tetikleyicisi, belirlenen işlem gerçekleşirken devreye girer ve kendi içinde tanımlanan komutları icra etmeye başlar. Yani, belirlenen işlemin yerine geçer.

Bu tetikleyiciden önceki tetikleyiciler, veriler uygun değilse ROLLBACK ile işlemleri geri alırlar. INSTEAD OF, işlem gerçekleşirken verilerin uygunluğunu denetleyecektir.

- Uygulaması yapılacak olan bu tabloda, kayıtlı olan müşteri silinmeye çalışıldığı anda silme işleminin gerçekleştiği tarih ve saat otomatik olarak "SilmeGirisimi" adlı alana yazılacaktır.

SQL – TRIGGERS (Tetikleyiciler)

- "Musteri_Silme" adında bir tetikleyici oluşturarak bu tetikleyicinin tablodan kayıt silinmesinin başlaması durumunda sistem tarih ve saatini "SilmeGirisimi" sütununa yazmasını ve sorguyu EXECUTE ederek tetikleyicinin oluşmasını sağlayınız.

```
CREATE TRIGGER Musteri_Silme
ON Musteriler
INSTEAD OF DELETE
AS
UPDATE Musteriler SET SilmeGirisimi = GETDATE()
WHERE Musteri_No IN (SELECT Musteri_No FROM DELETED)
```

SQL – TRIGGERS (Tetikleyiciler)

- "Musteriler" tablosunda üçüncü ve beşinci kayıtları silecek yeni bir sorgu oluşturunuz ve çalıştırınız.

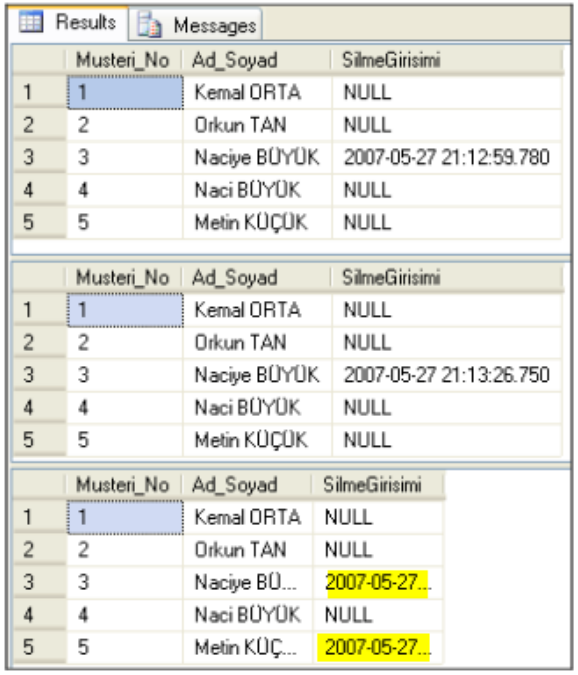
```
SELECT * FROM Musteriler
DELETE Musteriler WHERE Musteri_No=3

SELECT * FROM Musteriler
DELETE Musteriler WHERE Musteri_No=5

SELECT * FROM Musteriler
```

Resim 4.17: Kayıt silme sorgusu

- 3 ve 5 numaralı kayıt silinmek istendiğinde tetikleyici devreye girecek ve "SilmeGirisimi" sütununa sistem tarih ve saatini yazacaktır.
- Uygulamanın sonuç penceresine bakarak yapılan işlemleri görebilirsiniz.

SQL – TRIGGERS (Tetikleyiciler)

- 3 ve 5 numaralı kayıt silinmek istendiğinde tetikleyici devreye girecek ve "SilmeGirisimi" sütununa sistem tarih ve saatini yazacaktır.
- Uygulamanın sonuç penceresine bakarak yapılan işlemleri görebilirsiniz.


Results Messages

	Musteri_No	Ad_Soyad	SilmeGirisimi
1	1	Kemal ORTA	NULL
2	2	Orkun TAN	NULL
3	3	Naciye BÜYÜK	2007-05-27 21:12:59.780
4	4	Naci BÜYÜK	NULL
5	5	Metin KÜÇÜK	NULL

	Musteri_No	Ad_Soyad	SilmeGirisimi
1	1	Kemal ORTA	NULL
2	2	Orkun TAN	NULL
3	3	Naciye BÜYÜK	2007-05-27 21:13:26.750
4	4	Naci BÜYÜK	NULL
5	5	Metin KÜÇÜK	NULL

	Musteri_No	Ad_Soyad	SilmeGirisimi
1	1	Kemal ORTA	NULL
2	2	Orkun TAN	NULL
3	3	Naciye BÜ...	2007-05-27...
4	4	Naci BÜYÜK	NULL
5	5	Metin KÜÇ...	2007-05-27...

SQL – TRIGGERS (Tetikleyiciler)

DELETE işlemi için Instead of TRIGGER örneği;

Her hangi bir silme işlemi gerçekleşmeden önce durduruluyor!!!

```
use urunler
create trigger silInsteadof on siparis
instead of delete as
begin
raiserror('silme olmaz',16,1)
rollback transaction
end


delete from tabloAdı
```

SQL – TRIGGERS (Tetikleyiciler)

Trigger Alter Etmek:

Trigger'ı düzeltmek için ALTER TRIGGER komutu kullanılır.

```
ALTER TRIGGER trigger_ismi ON tablo_ismi FOR
INSERT AS RAISERROR ('hata!!!', 16, 10)
```

SQL – TRIGGERS (Tetikleyiciler)

Trigger Drop Etmek:

Trigger silmek için;

```
DROP TRIGGER trigger_ismi
```

Eğer trigger'ın var olup olmadığını kontrol etmek istersek «`sys.triggers`» içinde trigger'ınızın varlığını kontrol ettirebilirsiniz.

```
declare @isim nvarchar(20)
set @isim='siparisKontrol1'
if  exists(select sys.triggers.name from
sys.triggers where name=@isim)
begin
drop trigger siparisKontrol1
end
else
begin
print @isim+' diye bir trigger yok!!'
end
```

SQL – TRIGGERS (Tetikleyiciler)

Sistemde tanımlı tüm triggerları listelemek

```
declare @trigAd nvarchar(50)
declare @trigId int

declare trigCrs cursor for select sys.triggers.name,sys.triggers.object_id from
sys.triggers
open trigCrs
fetch next from trigCrs into @trigAd,@trigId
while(@@FETCH_STATUS=0)
begin
print 'Trigger ismi:'+@trigAd+char(13)+'Trigger id:'+cast(@trigID as nvarchar(20))
fetch next from trigCrs into @trigAd,@trigId
end
close trigCrs
deallocate trigCrs
```

by yselim