

TRIGGER KAVRAMI

Öncelikle 3 çeşit trigger vardır. Bunlar:

1. Insert Trigger (**ilgili tabloda ekleme işlemi olduğu anda**)
- 2- 2. Delete Trigger (**ilgili tabloda silme işlemi olduğu anda**)
- 3- 3. Update Trigger (**ilgili tabloda güncelleme işlemi olduğu anda**)

Peki trigger nedir? Triggerin türkçe anlamı tetikleyicidir. Adından da anlaşılacağı gibi bir tablodaki değişikliği başka bir tabloda istediğimiz şekilde otomatik olarak değiştirmek istediğimizde trigger kullanırız. Buradan da anlaşılacağı gibi bu uygulama (trigger) 2 farklı tablo kullanmaktadır.

Hemen örneklerimiz üzerinden gidelim. Konunun daha iyi anlaşılması için örneğimde aşağıdaki gibi ürünler ve satışlar adında basit 2 tablo oluşturdum.

Table - dbo.urunler	Table - dbo.satilar	Summary
urunid	urunAdi	urunAdedi
1	kitap	50
2	defter	100
3	kalem	255
**	NULL	NULL

Table - dbo.satilar	Summary	
satisid	urunid	satisAdedi
5	1	10
**	NULL	NULL

Tablolarımızı iyice inceleyelim. Sütun adlarına dikkat edelim.

INSERT TRIGGER

Şimdi yapacağımız ilk trigger insert trigger olsun. Buradaki amaç trigger ekleyeceğimiz tabloda herhangi bir ekleme işlemi olduğu anda diğer tabloda da yazdığımız algoritmanın otomatik olarak etki etmesini sağlamak.

Örneğimizden yola çıkalım. Kitap ürünümüzün stoktaki adedi 50 tane ama biz sunu istiyoruz. Kitaptan 10 adet sattığımızda stoktaki sayısının otomatik olarak 10 tane azalmasını istiyoruz. Hiçbir işlem yapmazsak bu mümkün değil. Tablomuzda bakarsak ben 10 tane kitap sattığım halde stoktaki sayım hala 50 tane görünüyor. Şimdi bu olayı insert trigger ile halledelim.

Yeni bir sql query oluşturalım ve Insert trigger için aşağıdaki kodları yazalım.

```
create trigger urunAdediGuncelle on satilar --1
for insert --2
as begin --3
declare @urunid int, @satisadedi int --4
select @urunid =urunid, @satisAdedi=satisAdedi from inserted --5
update urunler set urunadedi=urunadedi-@satisadedi where urunid=@urunid --6
end --7
```

Şimdi yazdığımız kodları açıklayalım.

--1. Yeni bir trigger oluşturmak için create trigger diyoruz, triggerimizin adını yazıyoruz ve satışlar tablosunda ekleme olduğu anda bir kod bloğu çalışacağı için on satışlar diyoruz.

--2. Trigger çeşidimizi belirtiyoruz. Satışlar tablosunda herhangi bir ekleme işlemi olduğu anda sql server "inserted" adında bir tablo oluşturur. Bu tablonun adını veya yerini değiştiremeyiz.

--3. Kodlarımızı yazacağımız bloğu açıyoruz.

--4. Ununid ve satisadedi adında int türünde 2 tane değişken tanımladık.

--5. Sql server arka planda inserted adında bir tablo oluşturur demistik. Şimdi bu tablodan (inserted tablosu) tanımladığımız değişkenlere değerlerini atıyoruz.

--6. Bu kısımda da diğer tabloda (bizim için ürünler tablosu) yapılması gereken değişikliği sorgu biçiminde yazıyoruz. Sorgumuzda ürünler tablosundaki urunadedinin girdiğimiz satisadedi kadar azalmasını sağlamışız. Tabii ki daha sonra where koşulu ile bunun satışını yaptığımız üründe gerçekleşmesini sağlamışız.

--7. Sonlandırıyoruz.

Triggerimizi execute edelim, yeni bir satış girelim ve çalıştığını görelim.

Table - dbo.urunler	Table - dbo.satilar	Summary
satisid	urunid	satisAdedi
5	1	10
6	1	10
**	NULL	NULL

Table - dbo.urunler	Table - dbo.satilar	Summary
urunid	urunAdi	urunAdedi
1	kitap	40
2	defter	100
3	kalem	255
*	NULL	NULL

Kalemden 10 adet daha sattık ve bu kez kalemin stok adedinin otomatik olarak 10 azaldığını gördük.

DELETE TRIGGER

Şimdi diğeri bir trigger çeşidi olan delete triggerı inceleyelim. Burdaki amaç trigger ekleyeceğimiz tabloda herhangi bir silme işlemi olduğu anda diğeri tabloda da, yazdığımız sorgunun otomatik olarak etki etmesini sağlamak. Örneğimizle ilişkilendirecek olursak, herhangi bir ürün sattığımızda bu ürünün stok adedi sattığımız adet kadar düşecektir. Bunu insert trigger ile yaptık. Sattığımız ürünü geri iade alırsak ve satışı iptal edersek stogumuzun eski haline dönmelerini sağlamamız gerek. Yani sattığımız ürün kadar stoga eklenmesi gerek. Burada delete trigger ile yapıyoruz.

Hemen delete triggerımızı yazalım

```
create trigger satisIptalEt on satislar --1
for delete --2
as
begin
declare @urunid int,@satisadedi int
select @urunid =urunid, @satisadedi=satisAdedi from deleted --3
update urunler set urunadedi=urunadedi+@satisadedi where urunid=@urunid --4
end
```

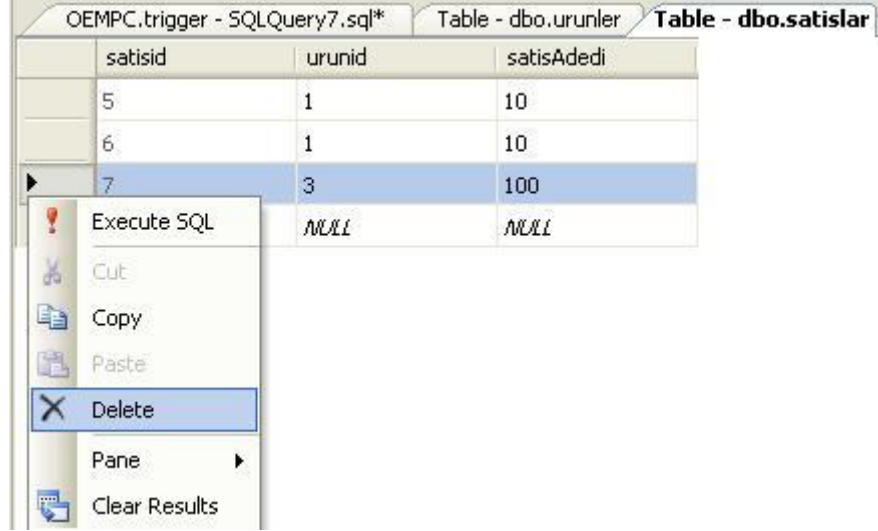
--1. SatisIptalEt adında yeni bir trigger oluşturuyoruz. Tabi bu satislar tablosu üzerinde bir silme olduğunda çalışacağı için on satislar diyoruz.

--2. Trigger çeşidimizi belirtiyoruz. Burada sql server arka planda “deleted” adında bir tablo oluşturacaktır.

--3. Tanımladığımız değişkenlerin değerlerini deleted tablosundan alıyoruz.

--4. Yine sorgumuzu yazıyoruz. Bu kez sildığımız satışın satış adedi kadar o ürünün ürün adedine eklenmesini sağlıyoruz.

Triggerimizi execute edelim, bir satış silelim ve nasıl etki ettiğini görelim.



satisid	urunid	satisAdedi
5	1	10
6	1	10
7	3	100

Kalem adlı üründen 100 adet sattığımızda ilk önce 255 olan ürün adedi 155 e düşmüş olacaktır. Daha sonra bu satış sildiğimizde bu ürünün stok adedinin otomatik olarak tekrar 255 olduğunu görebilirsiniz.

UPDATE TRIGGER

Şimdi son olarak update triggeri inceleyelim. Burdaki amaç trigger ekleyeceğimiz tabloda herhangi bir güncelleme işlemi olduğu anda diğeri tabloda da, yazdığımız sorgunun otomatik olarak etki etmesini sağlamak.

Örneğimizde de daha önce satışını yaptığımız bir ürünün satış adedini değiştirelim. Yani var olan bir satışın satış adedini değiştirerek bunun ürünler tablosundaki ürün adedine nasıl yansidiğini inceleyelim.

Yeni bir sql query oluşturarak update triggerimizi yazmaya başlayalım.

```
create trigger SatisGuncelle on satislar
for update --1
as
begin
declare @urunid int, @eskisatisadedi int, @yenisatisadedi int, @fark int
select @urunid =urunid, @eskisatisadedi=satisAdedi from deleted --2
select @yenisatisadedi=satisadedi from inserted --3
set @fark=@yenisatisadedi - @eskisatisadedi --4
update urunler set urunadedi=urunadedi-@fark where urunid=@urunid --5
```

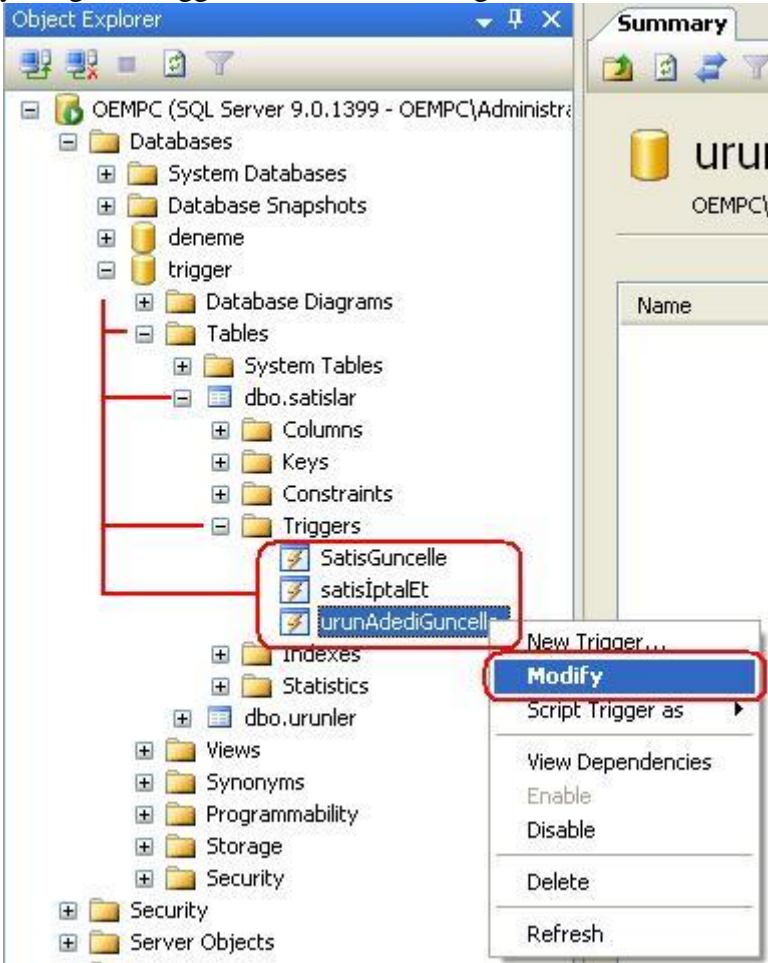
end

- 1. Trigger çesidimizi belirtiyoruz. Satislar tablosu üzerinde herhangi bir güncelleme olduğu anda çalışacaktır. Burada bilmeniz gereken update triggerin insert ve delete triggerdan biraz daha farklı çalışmasıdır. Triggerda direk bir güncelleme işlemi olmaz. Güncelleme yapabilmek için önce güncellenen tabloyu siler ve daha sonra güncellenmiş şekli ile tabloyu tekrar ekler.
 - 2. Yukarıda da belittığımız gibi önce silme işlemi yapıyoruz ve silinen tablodan eski satış adedini oluşturduğumuz değişkene atıyoruz.
 - 3. Şimdide insert tablosundan güncellenen satış adedini yenisatısadedi adlı değişkene aktarıyoruz.
 - 4. Yeni satış adedinden eski satış adedini çıkartarak aradaki farkı fark adlı bir değişkene aktarıyoruz.
 - 5. Son olarak ürünler tablosundaki ürün adedini aradaki fark kadar azaltıyoruz. Tabiki yine where koşulu ile hangi üründe değişiklik yapılacağını belirtiyoruz.
- Yaptığımız işlemi daha iyi anlamak için tablolarda olan değişiklikleri gözlemleyelim.

Table - dbo.urunler	Table - dbo.satislar	Summary
satisid	urunid	satisAdedi
5	1	10
6	1	20
NULL	NULL	NULL

Table - dbo.urunler	Table - dbo.satislar	Summary
urunid	urunAdi	urunAdedi
1	kitap	30
2	defter	100
3	kalem	255
NULL	NULL	NULL

10 adet olan kitap satışımızı aynı satış üzerinde 20'ye çıkartıyoruz ve ürünler tablosundaki 40 adet olan stok miktarının otomatik olarak yazdığımız trigger sayesinde 30'a düştüğünü göreceğiz. Makelemizin sonlarına doğru oluşturduğumuz bir triggerin nasıl değiştirileceğini görelim. Bunun için trigger yazdığımız tablo altından triggers kabini açarak değiştirmek istediğimiz trigger üzerine sağtık yapıp **modify** dersek yazdığımız trigger kodları karşımıza gelecektir.



Not: Yazdığımız trigger üzerinde değişiklik yaptığımızda create trigger yazdığımız kısmı alter trigger olarak değiştirdikten sonra execute yapmamız gerekecektir.

KAYNAK: <http://www.cozumpark.com/forums/thread/335010.aspx>