

بسمه تعالي

موضوع مقاله : تريگر در SQL

ارائه دهنده : فهيمه علي اكبر وفا

استاد مربوطه : آقاي مهندس محمد سليمي

Trigger

مفهوم ترگر :

فرض کنید بخواهید مقدار فیلد شماره درس را در جدول Course تغییر دهید یا حذف کنید . ممکن است این مقدار در جدول Grade وجود داشته باشد . در این صورت ، اگر مقدار شماره درس در جدول course تغییر کند یا حذف شود ، اطلاعات جدول Grade که این شماره درس را دارند ، به رکوردهای نا معتبر تبدیل خواهند شد . به عنوان مثال دیگر ، فرض کنید شماره درسی را در جدول grade وارد کنید ، ولی این شماره درس در جدول Course موجود نیست . برای جلوگیری از این مشکلات می توانید از تریگر ها استفاده کنید . ترگر ها ، نوع خاصی از رویه های ذخیره شده اند که در هنگام تغییر داده های جدول برای محافظت از طراحی بانک اطلاعاتی فعال می شوند و از ورود داده های معتبر جلوگیری می کنند.

تریگرها پارامتر ندارند و صریحا فعال نمی شوند و برای فعال سازی آنها تغییرات داده ای لازم است.

دو تفاوت بین ترگر ها و رویه های ذخیره شده:

- ۱- تریگر ها فاقد پارامتر هستند . ولی رویه های ذخیره شده می توانند پارامتر داشته باشند.
- ۲- تریگر ها در هنگام تغییر داده های جدول فعال می شوند ، ولی رویه های ذخیره شده حتما باید فراخوانی شوند تا اجرا گردند.

هر جدول می تواند حداکثر ۳ تریگر داشته باشد که عبارتند از:

0 تریگر در هنگام اضافه کردن رکورد (Insert)

0 تریگر در هنگام به روز رسانی رکورد (Update)

0 تریگر در هنگام حذف رکورد (Delete)

به طور پیش فرض ، کلیه تریگر های جدول پس از تغییرات جدول فعال می شوند . این تریگر ها AFTER نام دارند . تریگر های دیگری نیز وجود دارند که INSTEAD OF نامیده می شوند و به جای تغییر در داده های مورد نظر ، باعث فعال سازی خودشان می شوند . در SQL SERVER تریگر ها از دو جدول Deleted و Inserted استفاده می کنند .

درك جدول Deleted و Inserted

تریگرها از دو جدول inserted و deleted استفاده می کنند. این جداول ساختار مشابهی با جدول پایه یا جدول trigger که تریگر در آن ایجاد می شود دارند. این دو جدول در حافظه RAM قرار می گیرند و جداول منطقی هستند . اگر رکورد جدیدی را به جدول پایه اضافه کنید این رکورد هم در جدول پایه و هم در جدول inserted ثبت می شود. وجود مقادیر در دسترس در جدول inserted به شما امکان می دهد که نیازی به ایجاد متغیر برای نگهداری اطلاعات نداشته باشید و به اطلاعات دلخواه دسترسی بیابید. هنگام حذف یک رکورد، رکورد حذف شده در جدول deleted قرار می گیرد. به روز رسانی نیز یک حذف و درج متوالی است. اگر رکوردی را به روز رسانی کنید، رکورد اصلی در جدول deleted و رکورد تغییر یافته در جدول پایه و جدول inserted ذخیره می شود.

نکته : شما نمی توانید محتوای جداول Deleted و inserted را ببینید.

ایجاد تریگرها با دستور CREATE TRIGGER

نگارش ایجاد تریگرها به شرح زیر است:

کد:

```
CREATE TRIGGER [owner.] trigger name
ON [owner.] table_name | view name
{FOR|AFTER | INSTEAD OF} {INSERT | UPDATE | DELETE}
[WITH ENCRYPTION]
AS sql.statements
```

یک جدول می تواند هر تعداد تریگر از انواع INSERT, UPDATE, DELETE داشته باشد. هر عمل را می توان در یک یا چند تریگر ذخیره کرد. اگر عملی در چند تریگر ذخیره شود باید نام تریگرها منحصر بفرد و یکتا باشد.

trigger name:

نام تریگر را مشخص می کنند

ON {table | view}:

جدول یا دیدگاهی که تریگر برای آن ایجاد می شود را مشخص می کند که می تواند یکی از جداول بانک یا دیدگاه ها باشد.

{FOR | AFTER | INSTEAD OF}:

مشخص می کند که تریگر در چه زمانی اجر شود قبل از عمل ، بعد از عمل یا به جای عمل ..

{[INSERT] [UPDATE] [DELETE]}:

مشخص می کند که برای کدام عمل اصلی اعمال شود که می تواند مقادیر (INSERT , UPDATE , DELETE) باشد

sql_statement:

مجموعه دستوراتی که باید انجام شود را نگه می دارد

نکته: به جای کلمه for می توان از کلمه after , instead of نیز استفاده کرد و کلمه after پیش فرض است.

After : بعد از انجام موفقیت آمیز عملیات (insert/update/delete)

For : در همان لحظه اجرا می شود.

Instead of : ترگیر را به یک قطعه کد T.SQl عادی تبدیل می کند و مانند یک پروسچر معمولی تریگر را می سازد.

تریگرهای Delete و Update و Insert

حال به يك نمونه تریگر برای دستورات INSERT و UPDATE جدول توجه کنید:

کد:

```
CREATE TRIGGER trAddAuthor
ON authors
FOR INSERT, UPDATE
AS raiserror ('%d rows have been modified', 0, 1, @@rowcount)
RETURN
```

کد:

```
CREATE TRIGGER trDelAuthors
ON authors
FOR DELETE
AS raiserror ('%d rows are going to be deleted from this table!', 0, 1, @@rowcount)
```

توجه کنید که پیام را در صورتی مشاهده می کنید که عمل حذف را در محیط کوئری آنالیز انجام دهید.

تریگری برای دیدن محتوای جدول Deleted و inserted

کد:

```
CREATE TRIGGER tr24 ON Teacher
FOR DELETE, update, insert
AS
select * into del from deleted
select * into ins from inserted
```

نام تریگر بالا tr24 است. که بر روی جدول teacher ساخته شده است. خط دوم یعنی وقتی عملیات اضافه یا حذف یا ویرایش بر روی جدول teacher اتفاق افتاد این تریگر فراخوانی شود. خط چهارم رکوردهای موجود در جدول Deleted را در جدول del می ریزد و خط پنجم هم محتوای جدول inserted را در جدول ins می ریزد. توجه کنید که جدولهای del و ins وجود داشته باشند وگرنه پیام خطا میدهد.

بکارگیری تریگرهای INSTEAD OF

هنگام استفاده از تریگرهای INSTEAD OF کد اصلی ایجاد تغییرات در جدول اجرا نمی شود بلکه کد تریگر اجرا می شود. برای مثال شما می توانید تریگری را برای جدول authors ایجاد کنید که به کاربران اطلاع دهد مولفین را نمی توان حذف کرد. این کار با تریگر معمولی FOR ، AFTER نیز قابل انجام است. اما مستلزم آن است که تغییرات واقعی در ابتدا اعمال شوند سپس در کد تریگر عقبگرد شوند. اگر کد تریگر INSTEAD OF را بکار برید، عمل به روز رسانی انجام نمی شود و طبیعی است این تریگر کارا تر است. به مثال زیر توجه کنید.

کد:

```
USE pubs
GO
CREATE TRIGGER trIO_DelAuthors
ON authors INSTEAD OF DELETE
AS
PRINT 'you can not delete authors from the authors table!'
```

برای تست تریگر سعی کند مولفی با نام خانوادگی white را حذف کنید:

کد:

```
USE pubs
GO
DELETE authors WHERE au_name = 'white'
```

ویرایش تریگرها :

توسط دستور ALTER TRIGGER تریگر را ویرایش کنید یا ابتدا آنرا حذف نموده و دوباره ایجاد کنید.
توجه : تغییر تریگر با دستور ALTER TRIGGER ، سبب جایگزینی کامل تریگر قدیم با تریگر جدید می شود. با حذف یک جدول، تریگرها موجود آن نیز به شکل خودکار حذف می شوند

```
ALTER TRIGGER trigger_name
ON ( table | view )
[ WITH ENCRYPTION ]
{
{ ( FOR | AFTER | INSTEAD OF ) { [ DELETE ] [ , ] [ INSERT ] [ , ] [ UPDATE ] }
[ NOT FOR REPLICATION ]
AS
sql_statement [ ...n ]
}
```

در صورتی که بخواهید مقدار فیلد tid را در جدول Teacher حذف کنید، ممکن است این مقدار در جدول Students وجود داشته باشد. با دستور زیر اگر مقدار tid در جدول Teacher حذف شود، مقدار tid جدول Students که این شماره استاد را دارند نیز حذف می شوند.

```
create trigger trdeleteteach_stude
on teacher
after delete
as
delete from students
where students.tid in (select tid from deleted)
```

حذف تریگر :

برای این منظور از دستور DROP TRIGGER استفاده کنید.

```
DROP TRIGGER {trigger} [...n]
```

Trigger...*n*: تریگر هایی هستند که باید حذف شوند