



Trigger

م.عبدالبدیع مراد

محتوى مجاني غير مخصص للبيع التجاري

13/12/2022

RB Informatics;

قواعد المعطيات (2)

Trigger

ماهو ال Trigger؟ وما الفرق بينه وبين ال procedure أو ال function؟

يعد ال Trigger مثل ال procedure أو ال function، إلا أنه يُستدعى بشكل تلقائي عندما يتم كتابة تعليمة DML، ولسنا بحاجة إلى استدعاءه.

أما ال procedure وال function بحاجة execute للاستدعاء، ويتم إنشاء ال Trigger على ال table بشكل مباشر ويكون تابع له، فمثلاً إذا أردت تنفيذ تعليمة insert وكان هناك Trigger تابع لهذا ال Table وخاص بال insert فإنه يستدعى بمجرد قيامنا بعمل insert على هذا ال Table، ومنه نستنتج أن لل Trigger علاقة بنوع عملية DML المستخدمة.

تذكرة:

تعليمات ال DML هي: delete, update, insert

طريقة بناء ال Trigger

```
Create Trigger name
On Table name
For DML (insert, update, delete)
As
Begin
(Body)
End
```

يوجد ضمن ال Trigger 2 objects هما (Inserted, deleted)، وال Body الخاص بهم يتناسب (يتطابق) مع body ال Table، وهم فعلياً جدولين افتراضيين ال body الخاص بهم متغير ومحكوم بال table الذي نتعامل معه.

نستطيع التعامل مع هذا ال table، وتكون طريقة التعامل هي عبارة عن قراءة فقط لا غير ضمن ال body الخاص بال Trigger.



ما محتوى ال inserted وال deleted؟

- كل شيء جديد new value يضاف لل inserted.
- وكل شيء عبارة عن delete أو update لل data يضاف لل deleted.

مثال: ليكن لدينا الجدول table التالي:

t1

Pk	C1	C2	C3
	1	a	b
	2	c	d

Insert into t1
Values (3, g, f)

-عند القيام بالتعليمة:

Trigger insert → insert الخاص بال

Select *
From inserted

3	g	f
---	---	---

Update t1
Set C2 = 'aa'
Where C1 = 1

-وعند القيام بالتعليمة:

Trigger update

Select *
From inserted

1	aa	b
---	----	---

Select *
From deleted

1	a	b
---	---	---

-ومنه نستنتج أنه يتم الاحتفاظ بال new value التي تضاف ضمن ال inserted، ويتم الاحتفاظ بال old value ضمن ال deleted.



■ ملاحظة:

يمكننا عمل Trigger على update و delete مثلاً بنفس الوقت

■ مثال: سنقوم ببناء Trigger بسيط:

```
CREATE TRIGGER test_trigger
ON emp
FOR INSERT
AS
BEGIN
SELECT *
FROM inserted

SELECT *
FROM deleted
end
go
```

-قمنا بإنشاء trigger على عملية insert على جدول employees، سيقوم هذا الـ trigger بعملية select على جدولي الـ inserted والـ deleted عند القيام بعملية insert على جدول الـ employees.

-الآن سنقوم بعملية insert على هذا الجدول:

The screenshot shows a SQL query window with the following text:

```
18 INSERT INTO emp
19 VALUES (9, 'TEST', 'M', 1000, 777, 'damas')]
```

Below the query window, the 'Results' tab is selected, displaying a table with the following data:

emp_id	name	gender	salary	phone	address
9	TEST	M	1000	777	damas

-نلاحظ أنه ارجع جدولين، الأول فيه الـ record الجديد الذي تمت اضافته، بالتالي هو جدول الـ inserted، والثاني فارغ لأننا لم نحذف/نعدل على الجدول، وهو جدول الـ deleted.

■ فائدة الـ Tigger:

الـ Tigger يسمح لنا بالإمساك بالـ new values والـ old values، وفي حال كنا نريد عمل سجل متابعة يساعدنا الـ Tigger في معرفة التغييرات (audit) الخاصة على الـ DB، وهو أمر مهم جداً لمتابعة أي تغيير حدث على row ما ومن قبل من وفي أي وقت ما منع التلاعب في البيانات.

- الآن نريد إنشاء جدول متابعة لجدول الـ dept، أي نريد أن نخزن في الـ Table Audit أي Transaction يحصل على جدول الـ dept.

■ إنشاء جدول Table Audit:

يحتوي على id و description لتوضيح الحاصل الـ Transaction على جدول dept .

```
CREATE TABLE dbo.tbdept_audit (
    id          int identity (1,1),
    description varchar(255));
```

Messages

Commands completed successfully.

Completion time: 2022-12-16T19:22:49.0691451+02:00



■ بناء الـ Tigger على الجدول الذي نريد متابعته (أي dept):

```
CREATE TRIGGER tr_tbdept_forinsert
ON dept
FOR INSERT
AS
BEGIN
    DECLARE @p_depid int
    DECLARE @p_dname varchar(50)

    SELECT @p_depid = depid, @p_dname = dname
    FROM INSERTED

    INSERT INTO tbdept_audit
    VALUES ('New dept id = ' +
        CAST(@p_depid AS NVARCHAR(5)) +
        ' name = ' + @p_dname +
        ' Is Added At ' +
        CAST(GETDATE() AS NVARCHAR(20))
    )
END
```

■ ويقوم هنا الـ Tigger بعمل insert على جدول Audit مع كل مرة يتم فيها عمل insert على جدول Dept ولإمسك الـ value التي يتم تخزينها في Dept نقوم بتعريف متحولين عن طريق تعليمة Declare ثم إسناد القيم المخزنة إلى المتحولات التي قمنا بتعريفها.

عرض محتويات كل من الجدولين Dept و Audit:

```
SELECT * FROM dept
SELECT * FROM tbdept_audit
```

deptid	dname
1	Admin Update
2	Sales
3	Archive

id	description
----	-------------

الآن لو قمنا بعمل insert على جدول Dept نلاحظ 2 row affected والسبب أن في نفس الوقت الذي يتم تخزين القيمة الجديدة في جدول Dept قام الـ Tigger بالتخزين في جدول Audit.

```
INSERT INTO dept VALUES ('New Dept')
```

(1 row affected)

(1 row affected)

Completion time: 2022-12-16T19:42:26.3337934+02:00

```
SELECT * FROM dept
SELECT * FROM tbdept_audit
```

deptid	dname
1	Admin Update
2	Sales
3	Archive
4	New Dept

id	description
1	New dept id = 4 name = New DeptIs Added At Dec ...

إنشاء Tigger من أجل الـ Delete:

```
CREATE TRIGGER tr_tbdept_fordelete
ON dept
FOR DELETE
AS
BEGIN
    DECLARE @p_depid int
    DECLARE @p_dname varchar(50)

    SELECT @p_depid = deptid, @p_dname = dname
    FROM DELETED

    INSERT INTO tbdept_audit
    VALUES ('An Existing dept with id = ' +
        CAST(@p_depid AS NVARCHAR(5)) +
        ' name = ' + @p_dname +
        'Is Deleted At ' +
        CAST(GETDATE() AS NVARCHAR(20))
    )
END
```

■ سنقوم بعمل Delete للسجل الذي قمنا بإدخاله قبل قليل: كذلك نلاحظ 2 row affected

```
DELETE dept
WHERE deptid = 4
```

Messages

(1 row affected)

(1 row affected)

Completion time: 2022-12-16T20:01:04.9381222+02:00

■ ليصبح كل من الجدولين بالشكل:

```
SELECT * FROM dept
SELECT * FROM tbdept_audit
```

Results Messages

	deptid	dname
1	1	Admin Update
2	2	Sales
3	3	Archive

	id	description
1	1	New dept id = 4 name = New DeptIs Added At Dec ...
2	2	An Existing dept with id = 4 name = New DeptIs Dele...

ونلاحظ لولا سجل المتابعة لم يكن هناك وسيلة لمعرفة الأقسام التي كانت موجودة في السابق قبل أن يتم حذفها.

- ملاحظة: في كل من Tigger ال insert و Delete نحن نضمن أن التعديل يحصل على سجل واحد فيرد لنا متحول dept id , dname واحد في كل مرة، فماذا لو تم عمل insert لعدة سجلات في مرة واحدة؟
- عندها عملية الإسناد ستعطي Exception لأن الاستعلام الواحد يرد N Record (مصفوفة من المتحولات) فلا يمكن تخزينها بمتحول واحد.

■ إذاً لو كان التغير الحاصل على أكثر من Row كيف نعالج المشكلة السابقة؟

يكون الحل بتخزين كامل ال values من ال INSERTED في TempDept # (وال # ترمز لكونه جدول مؤقت يتم بناؤه ضمن scoop ال Tigger وبمجرد الخروج من ال scoop يتم التخلص من ال obj فلا يعود له وجود، وتكون بنية هذا الجدول هي نفسها بنية ال INSERTED (أي بنية ال Dept) وبعد تخزين ال INSERTED في هذه الجدول نقوم بعمل حلقة loop: طالما يوجد Data ضمن Table ال Temp، قم بعمل select لأول Row من ال Temp (أي ال INSERTED) وبالتزامن معه نقوم بعمل select (From Deleted) حيث قيمة ال id مساوية لقيمة ال id الذي تم عمل INSERT له، وبعد قراءة كامل سجلات الجدول Temp نقوم بعمل Delete له حتى لا ندخل بـ infinity loop، وهي الطريقة الصحيحة لعمل Tigger، في حين أن كل من ال insert و ال Delete السابقين غير صحيحين عملياً.

■ والآن سنقوم بإنشاء Trigger لعملية ال update بالطريقة التي ذكرناها قبل قليل:

```
CREATE TRIGGER tr_tbdept_forupdate
ON dept
FOR UPDATE
AS
BEGIN
    DECLARE @depid int
    DECLARE @old_dname nvarchar(50), @new_dname nvarchar(50)
    DECLARE @AuditString nvarchar(500)

    SELECT *
    INTO #TempDept
    FROM INSERTED

    WHILE (EXISTS (SELECT 1 FROM #TempDept))
    BEGIN
        SET @AuditString = ''
```



```
        SELECT TOP 1
            @depid = depid,
            @new_dname = dname
        FROM #TempDept

        SELECT @old_dname = dname
        FROM DELETED
        WHERE depid = @depid

        SET @AuditString = 'Dept with id = ' + CAST(@depid AS nvarchar(5)) + 'changed ' +
            ' Name From ' + @old_dname + ' to ' + @new_dname

        INSERT INTO tbdept_audit VALUES (@AuditString)

        DELETE #TempDept
        WHERE depid = @depid
    END
END
```

■ الآن سنقوم بعمل update لكامل قيم dname في جدول ال Dept (اي 3 سجلات):

```
UPDATE dept SET dname = dname + 'updated'
```

Messages

(3 rows affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)|
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (3 rows affected)

Completion time: 2022-12-16T21:21:11.8462677+02:00

■ نلاحظ النتيجة قام ال Tigger بتخزين القيم القديمة مع التحديث الحاصل عليها في جدول Audit:

```
SELECT * FROM dept
SELECT * FROM tbdept_audit
```

Results Messages

	deptid	dname
1	1	Admin Updateupdated
2	2	Salesupdated
3	3	Archiveupdated

	id	description
1	1	New dept id = 4 name = New DeptIs Added At Dec 16 2022 8:10PM
2	2	An Existing dept with id = 4 name = New DeptIs Deleted At Dec 16 2022 8:10PM
3	3	Dept with id =3changed Name From Archive to Archiveupdated
4	4	Dept with id =2changed Name From Sales to Salesupdated
5	5	Dept with id =1changed Name From Admin Update to Admin Updateupdated

■ ولو لم نقوم بعمل loop لما تمكن ال Tigger من تخزين الثلاث تحديثات الحاصلة في آن واحد معاً، وكان أعطى Exception.



نكونُ هُنَا قد وصلنا الى نهاية

القسم العملي من المقرر، كان معكم

فريق بنى المعطيات 2، لا تنسونا من

صالح دعائكم.