

كلية الهندسة المعلوماتية

السنة الرابعة - هندسة البرمجيات

Trigger

STORAGE TECHNOLOGY SAN AND TECHN

م.عبدالبديع مراد

محتوى مجاني غير مخصص للبيع التجاري

قواعد المعطيات (2)

RBOInformatics;

Trigger

ماهو الـ Trigger؟ ومالفرق بينه وبين الـ procedure أو الـ Trigger

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

أما الـprocedure والـ function بحاجة execute للاستدعاء، ويتم إنشاء الـ Trigger على الـ table بشكل مباشر ويكون تابع له، فمثلاً إذا أردت تنفيذ تعليمة insert وكان هناك Trigger تابع لهذا الـTrigger وخاص بالـ insert فإنه يستدعى بمجرد قيامنا بعمل insert على هذا الـTrigger، ومنه نستنتج أن للـTriggerعلاقة بنوع عملية DML المستخدمة.

تذكرة:

تعلیمات الـ DML هی: delete, update, insert

طريقة بناء الـ Trigger

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

- يوجد ضمن الـaobjects Trigger هما (Inserted, deleted)، والـBody الخاص بهم يتناسب (يتطابق) مع body الـtable الذي نتعامل body الـtable الذي نتعامل معه.
 - نستطيع التعامل مع هذا الـtable، وتكون طريقة التعامل هي عبارة عن قراءة فقط لا غير ضمن الـbody
 الخاص بالTrigger.







ما محتوى الـ Inserted والـ deleted؟

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

مثال: ليكن لدينا الجدول table التالي:

t1

Pk		
C1	C2	C3
1	а	ь
2	С	d

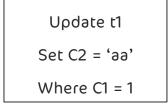
-عند القيام بالتعليمة:

Insert into t1
Values (3, g, f)

الـ Trigger insert → insert الخاص بال Trigger



-وعند القيام بالتعليمة:

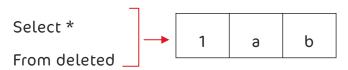


Trigger update

Select *

1 aa b

From inserted





-ومنه نستنتج أنه يتم الاحتفاظ بالـ new value التي تضاف ضمن ال Inserted، ويتم الاحتفاظ بالـ old value ضمن الـ deleted.





ملاحظة:

يمكننا عمل Trigger على update و delete مثلاً بنفس الوقت

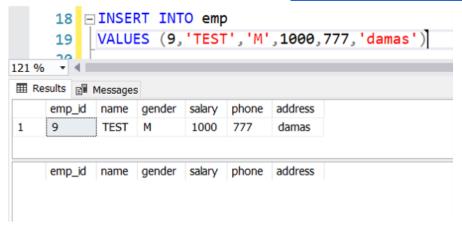
مثال: سنقوم بيناء Trigger بسيط:

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

SELECT *
FROM deleted
end
go
```

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

-الأن سنقوم بعملية insert على هذا الجدول:



-نلاحظ أنه ارجع جدولين، الأول فيه الـrecord الجديد الذي تمت اضافته، بالتالي هو جدول الـinserted، والثاني فارغ لأننا لم نحذف/نعدل على الجدول، وهو جدول الـdeleted.

فائحة الـ Tigger:

الـTigger يسمح لنا بالإمساك بالـnew values والـold values، وفي حال كنا نريد عمل سجل متابعة يساعدنا الـ Tigger في معرفة التغييرات (audit) الحاصة 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 أي Transaction



إنشاء جدول Table Audit:

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

```
CREATE TABLE dbo.tbdept_audit (
                      int identity (1,1),
         description varchar(255));
175 % + 4
Messages
   Commands completed successfully.
   Completion time: 2022-12-16T19:22:49.0691451+02:00
```

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

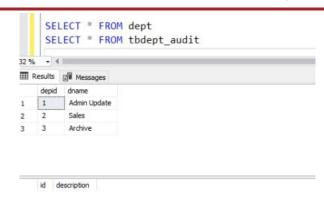
```
☐ CREATE TRIGGER tr_tbdept_forinsert
 ON dept
 FOR INSERT
 AS
BEGIN
     DECLARE @p_depid int
     DECLARE @p_dname varchar(50)
     SELECT @p_depid = depid, @p_dname = dname
     FROM INSERTED
    INSERT INTO tbdept audit
     VALUES ('New dept id = ' +
             CAST(@p_depid AS NVARCHAR(5)) +
             ' name = ' + @p_dname +
             'Is Addded At ' +
             CAST(GETDATE() AS NVARCHAR(20))
 END
```

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

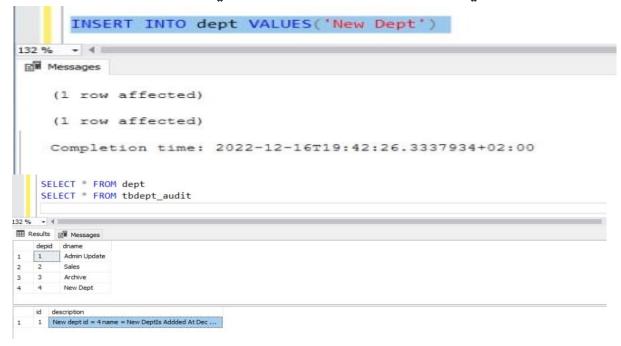




عرض محتويات كل من الجدولين Dept و Audit:



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



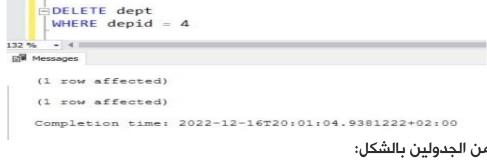
إنشاء Tigger من أجل الـ Delete:

```
CREATE TRIGGER tr_tbdept_fordelete
  ON dept
  FOR DELETE
BEGIN
      DECLARE @p_depid int
      DECLARE @p_dname varchar(50)
      SELECT @p_depid = depid, @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
```

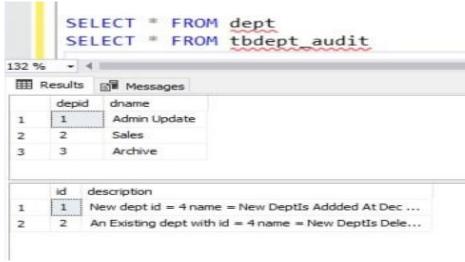




سنقوم بعمل Delete للسجل الذى قمنا بإدخاله قبل قليل: كذلك نلاحظ Delete سنقوم بعمل



ليصبح كل من الجدولين بالشكل:



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

- ملاحظة: في كل من Tigger الـ insert و Delete نحن نضمن أن التعديل يحصل على سجل واحد فيرد لنا متحول tosert واحد في كل مرة، فماذا لو تم عمل insert لعدة سجلات فى 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 الـ values من الـ INSERTED في TempDept (والـ # ترمز لكونه جدول مؤقت يتم بناؤه ضمن scoop الـ Tigger وبمجرد الخروج من الـ scoop يتم التخلص من الـ dobj الله يعود له وجود، وتكون بنية هذا الجدول هي نفسها بنية الـ INSERTED (أي بنية الـ Dept) وبعد تخزين الـ INSERTED في هذه الجدول نقوم بعمل select هي نفسها بنية الـ Table ضمن Data الـ Temp من الـ Row من الـ Row من الـ Temp فقوم بعمل حلقة وloop: طالما يوجد bata ضمن Table الـ Temp ، قم بعمل select لأول Row من الـ Prom INSERTED (أي From INSERTED) حيث قيمة الـ bi مساوية لقيمة الـ أن INSERT وبالتزامن معه نقوم بعمل select بعمل Temp نقوم بعمل Delete الله ندخل بـ dolete المحيحة المحيحة العمل Tigger ، في حين أن Tigger كل من الـ insert و الـ Delete السابقين غير صحيحين عمليًا.



والأن سنقوم بإنشاء Tigger لعملية الـ 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
```

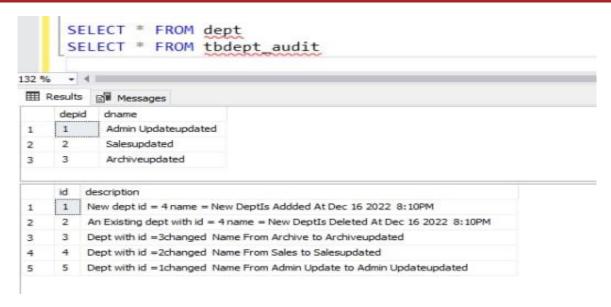




الأن سنقوم بعمل update لكامل قيم dname في جدول الـ Dept (اي 3 سجلات):

```
UPDATE dept SET dname = dname + 'updated'
(3 rows affected)
(1 row affected)
(3 rows affected)
Completion time: 2022-12-16T21:21:11.8462677+02:00
```

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



■ ولو لم نقم بعمل ١٥٥ρ لما تمكن الـ Tigger من تخزين الثلاث تحديثات الحاصلة في آن واحد معاً، وكان أعطى •Exception



نكونُ هُنا قد وصلنا الى نهاية القسم العملي من المقرر، كان معكم فريق بنى المعطيات 2، لا تنسونا من

صالح دعائكم.