



جامعة المنارة

كلية: الهندسة

قسم: المعلوماتية

اسم المقرر: قواعد بيانات ٢

رقم الجلسة (٧)

عنوان الجلسة

القوادح في قواعد البيانات Triggers



العام الدراسي: ٢٠٢٣-٢٠٢٤

الفصل الدراسي : الأول



جدول المحتويات

Contents

رقم الصفحة	العنوان
٣	القواعد في قواعد المعطيات
٤	أنماط القواعد
٥	تصميم القواعد الاستبدالية
٨	تصميم القواعد اللاحقة
١٠	تعديل و حذف القواعد

✓ تعريف الطالب بأنواع القواعد والحاجة إليها وكيفية تنفيذها وربطها بعمليات ال DML المختلفة

القواعد في قواعد المعطيات

Triggers in Databases

استخدام القواعد لفرض القيود :

القواعد بالتعريف هو نوع خاص من الإجراءات المخزنة التي تنفذ تلقائياً عندما تتغير البيانات الموجودة في جدول محدد. إذ يتم استدعاء القواعد استجابة إلى تعليمات Insert, Update, Delete. تتم معاملة القواعد والتعليمات التي حفزته كمناقلة واحدة وهي مناقلة يمكن التراجع عنها من داخل القواعد وذلك في حال حصول خطأ. توفر مخدمات قواعد البيانات طريقتين أساسيتين لفرض قيود العمل وهما القيود Constraints والقواعد Triggers.

مقارنة القواعد بالقيود :

الفائدة الأساسية للقواعد تكمن في احتوائها على تعليمات معالجة معقدة، وبالتالي فهي تشمل القيود في معالجتها.

تكون القواعد مفيدة في الحالات التي لا تحقق فيها القيود الحاجات الوظيفية لتطبيق ما:

- يمكن للقواعد أن تسلسل التعديلات إلى الجداول المرتبطة، ولكن هذه السلسلة يمكن فرضها أيضاً عبر قيود التكامل المرجعي واستخدام التسلسل للقيود المرجعية يكون عادة أكثر فعالية.
- يمكن للقواعد أن تفرض القيود المعقدة التي يصعب فرضها باستخدام قيد الاختبار Check Constraint. لا يمكن لقيد الاختبار أن يشير إلى أعمدة موجودة في جداول أخرى، أما القواعد فهي قادرة على ذلك.
- يمكن للقواعد أن يختبر حالة الجدول قبل وبعد التعديل، وأن يقوم بعمليات معينة بناء على ذلك.
- يمكن إضافة عدة قواعد لنفس نمط التغيير على الجدول.

قبل إنشاء القواعد

- لا يمكن تعريف قواعد الاستبدال للحذف والتعديل على الجداول التي تحوي مفاتيح أجنبية معرفة بأفعال مرتبطة مثل التسلسل CASCADE أو إسناد قيمة NULL أو قيمة افتراضية DEFAULT.
- لا تؤدي تعليمة بتر الجدول Truncate Table (وهي تعليمة مماثلة لتعليمة الحذف DELETE بدون شرط فلترة أي حذف جميع السجلات من الجدول) إلى تحفيز القواعد البديل عن الحذف.

تصميم القوادح

هناك نمطان من القوادح:

- القوادح الاستبدالية Instead Of Triggers: وهي قوادح تنفذ عوضاً عن التغيير الأصلي. أي إذا كان هذا القادح استبدالاً لعملية الإضافة Insert فلا تنفذ عملية الإضافة وإنما ينفذ القادح عوضاً عنه. بمعنى آخر لو أردنا أن تتم عملية الإضافة الأصلية فأن مسؤولية الإضافة تقع على عاتق القادح (أي يجب أن نكتب كود الإضافة في القادح). يمكن تعريف القوادح الاستبدالية على المناظير متعددة الجداول، الأمر الذي يسمح بتوسيع أنماط التحديث الممكنة عبر هذه المناظير.

- القوادح اللاحقة After Triggers: وهي قوادح تنفذ بعد نجاح الفعل الموافق لها (Insert, Update, Delete)، في حال فشلت عملية (الإضافة أو الحذف أو التعديل) فلن يتم استدعاء القادح اللاحق المرتبط.

يتم التحقق من القيود بعد القوادح من النمط Instead Of وقبل القوادح من النمط After. إذا سبب القادح Instead Of اختراقاً لأحد القيود فإن المناقلة الموافقة له تلغى، ولا يتم تنفيذ القادح After الموافق إن وجد.

تنبيه:

يمكن لأي جدول أو منظار أن يحوي قادحاً استبدالياً وحيداً، بينما يمكنه أن يحوي عدة قوادح لاحقة خاصة بكل فعل ممكن.

استخدام جداول السجلات المضافة والمحذوفة Inserted and Deleted Tables

يقوم مخدم البيانات بإنشاء وإدارة الجدولين Inserted, Deleted. هذان الجدولان هما جدولان مؤقتان مخزانان في الذاكرة ويحويان القيم السابقة Deleted واللاحقة Inserted الناتجة عن العملية.

يخزن جدول المحذوفات Deleted نسخة عن السجلات القديمة المتأثرة بكل من عمليتي الحذف والتعديل.

يخزن جدول الإضافات Inserted نسخة عن السجلات الجديدة الناتجة عن كل من عمليتي الإضافة والتعديل.

أثناء عمليتي الحذف والتعديل يقوم المخدم بحذف السجلات المتأثرة بالعملية من الجدول الموافق وإضافتها إلى الجدول Deleted.

أثناء عمليتي الإضافة والتعديل يقوم المخدم بتخزين القيم المحدثة أو الجديدة للسجلات في الجدول Inserted. أي أن هذا الجدول يحوي نسخاً من السجلات الجديدة المراد إضافتها إلى الجدول المرتبط.

بالتالي: جدول المحذوفات لا يحوي أي سجلات للقوادح البديلة عن الإضافة، كما أن جدول المضافات لا يحوي أي سجلات للقوادح البديلة عن الحذف.

بنية الجداول في الأمثلة:

```

CREATE TABLE Employee
( Emp_ID int identity,
  Emp_Name varchar(55),
  Emp_Sal decimal (10,2)
)

Insert into Employee values ('hasan',1000);
Insert into Employee values ('wesam',1200);
Insert into Employee values ('samer',1100);
Insert into Employee values ('lamis',1300);
Insert into Employee values ('fahed',1400);

create table Employee_log
( Emp_ID int,
  Emp_Name varchar(55),
  Emp_Sal decimal(10,2),
  Audit_Action varchar(100),
  Audit_Timestamp datetime
)

```

تصميم القوالب الاستبدالية:

الفائدة الأساسية للقوالب الاستبدالية هي لدعم المناظير القابلة للتحديث. إذا تضمن منظار ما على أكثر من جدول، عندها يجب استخدام القادح الاستبدالي لدعم عمليات الإضافة والحذف والتعديل عبر هذا المنظار للجداول المرتبطة به.

القادح الاستبدالي مع عملية الإضافة: Instead of Insert Trigger

يقوم القادح البديل عن الاستبدال بالتنفيذ عوضاً عن فعل الإضافة العادي المنفذ من قبل المخدم. هذا القادح مفيد عادة لتمكين الإضافة عبر منظار إلى مجموعة الجداول المضمنة فيه (أو بعضها على الأقل). في مثالنا هنا نقوم بمناقشة الموظف المضاف بشرط أن لا يكون راتبه أقل ١٠٠٠.

```

CREATE TRIGGER triger_instead_insert ON Employee
INSTEAD OF Insert
AS
declare @emp_id int, @emp_name varchar(55),
        @emp_sal decimal(10,2), @log_action varchar(100);
select @emp_id=i.Emp_ID from inserted i;
select @emp_name=i.Emp_Name from inserted i;
select @emp_sal=i.Emp_Sal from inserted i;
SET @log_action='Inserted Record -- Instead Of Insert Trigger.';

```

```

BEGIN
  if(@emp_sal<=1000)
  begin
    print 'Cannot Insert where salary < 1000 '
    ROLLBACK;
  end
  else
  begin
  Insert into
    Employee (Emp_Name,Emp_Sal)
    values (@emp_name,@emp_sal);
  Insert into
  Employee_log(Emp_ID,Emp_Name,Emp_Sal,Audit_Action,Audit_Timestamp)
    values( @@identity,@emp_name,@emp_sal
    ,@log_action,getdate() );
  PRINT 'Record Inserted Successfully .'
  END
END
-----ثم جرب الأمثلة التالية-----
insert into Employee values ('wedad',1300)
insert into Employee values ('Ahmad',900)

```

القادح الاستبدالي مع عملية التعديل:

يجب عدم تنفيذ عملية التعديل على الموظف في حال كان الراتب الجديد أقل من ١٠٠٠

```

CREATE TRIGGER trgInsteadOfUpdate ON dbo.Employee
  INSTEAD OF Update
AS
declare @emp_id int, @emp_name varchar(55),
        @emp_sal decimal(10,2), @log_action varchar(100);
select @emp_id=i.Emp_ID from inserted i;
select @emp_name=i.Emp_Name from inserted i;
select @emp_sal=i.Emp_Sal from inserted i;
SET @log_action='Updated Record -- Instead Of Update';
BEGIN
if(@emp_sal<= 1000)
begin
  RAISERROR('Cannot Insert where salary < 1000',16,1);
  ROLLBACK;
end

```

```

else
begin
update Employee set emp_name = @emp_name
                    , emp_sal = @emp_sal
    where emp_id = @emp_id;
insert into Employee_log(Emp_ID,Emp_Name,Emp_Sal
                        ,Audit_Action,Audit_Timestamp)
    values(@emp_id,@emp_name,@emp_sal,
          @log_action,getdate());
    PRINT 'Record Updated successfully';
END
END

```

-----ثم جرب الأمثلة التالية-----

```

update Employee set Emp_Sal = '1400' where emp_id = 6
update Employee set Emp_Sal = '900' where emp_id = 6

```

لعرض رسائل الخطأ يمكن استخدام أي من التعليمتين:

```

RAISERROR('Cannot Insert where salary < 1000',16,1);
print'Cannot Insert where salary < 1000'

```

القادح الاستبدالي مع عملية الحذف:

```

CREATE TRIGGER trgInsteadDelete ON dbo.Employee
INSTEAD OF DELETE
AS
declare @empid int, @empname varchar(55), @empsal decimal(10,2),
@audit_action varchar(100);
select @empid=d.Emp_ID FROM deleted d;
select @empname=d.Emp_Name from deleted d;
select @empsal=d.Emp_Sal from deleted d;
Begin
if(@empsal>1200)
begin
    RAISERROR('Cannot delete where salary > 1200',16,1);
    ROLLBACK;
end
else
begin
delete from Employee where Emp_ID=@empid;
COMMIT;
insert into Employee_log(Emp_ID,Emp_Name,Emp_Sal,

```

```
Audit_Action,Audit_Timestamp)
values(@empid,@empname,@empsal
,'Deleted -- Instead Of Delete Trigger.',getdate() );
PRINT 'Record Deleted -- Instead Of Delete Trigger.'
end
END
```

-----ثم جرب الأمثلة التالية-----

```
DELETE FROM Employee where emp_id = 1
DELETE FROM Employee where emp_id = 6
```

تصميم القوادح اللاحقة:

القادح اللاحق بعد عملية الاضافة After Insert Trigger

```
CREATE TRIGGER trgAfterInsert on Employee
FOR INSERT
AS declare @empid int, @empname varchar(55),
           @empsal decimal(10,2), @audit_action varchar(100);
select @empid=i.Emp_ID from inserted i;
select @empname=i.Emp_Name from inserted i;
select @empsal=i.Emp_Sal from inserted i;
set @audit_action='Inserted Record -- After Insert Trigger.';

insert into Employee_log(Emp_ID,Emp_Name,Emp_Sal,
Audit_Action,Audit_Timestamp)
values (@empid,@empname,@empsal,
@audit_action,getdate());
PRINT 'AFTER INSERT trigger fired.'
-----ثم جرب الأمثلة التالية-----
insert into Employee(Emp_Name,Emp_Sal)values ('sami',1000);
```

ملاحظة: حتى لو قام القادح الاستبدالي بإلغاء عملية الاضافة واستبدالها بأي تعليمة أخرى، يستمر تنفيذ القادح اللاحق لعملية الاضافة بعد تعليمة الاضافة، وكأن عملية الاضافة قد تمت بنجاح.


```

CREATE TRIGGER trgAfterUpdate ON dbo.Employee
FOR UPDATE
AS
declare @empid int, @empname varchar(55),
        @empsal decimal(10,2), @audit_action varchar(100);
select @empid=i.Emp_ID from inserted i;
select @empname=i.Emp_Name from inserted i;
select @empsal=i.Emp_Sal from inserted i;
if update(Emp_Name)
    set @audit_action='Update on Emp_name ';
if update (Emp_Sal)
    set @audit_action='Update on Salary..';
if update(Emp_Name)AND update (Emp_Sal)
    set @audit_action='Update on Salary and NAmE';

insert into
    Employee_log (Emp_ID,Emp_Name,Emp_Sal,
                  Audit_Action,Audit_Timestamp)
    values (@empid,@empname,@empsal,
            @audit_action,getdate());
PRINT 'AFTER UPDATE trigger fired.'
-----ثم جرب الأمثلة التالية-----
update Employee set Emp_Name='Pawan' Where Emp_ID =6;

```

التابع update في السطر: if update(Emp_Name)
يستخدم للتحقق من أن الحقل Emp_Name قد تم تعديل قيمته (أي هو يستخدم لمعرفة الحقل الذي
تغيرت قيمته من الحقل الذي لم تتغير قيمته)

القادح اللاحق بعد عملية الحذف After Delete Trigger

```

CREATE TRIGGER trgAfterDelete ON dbo.Employee
FOR DELETE
AS
declare @empid int, @empname varchar(55),
        @empsal decimal(10,2), @audit_action varchar(100);
select @empid=d.Emp_ID FROM deleted d;
select @empname=d.Emp_Name from deleted d;
select @empsal=d.Emp_Sal from deleted d;
select @audit_action='Deleted -- After Delete Trigger.';

insert into Employee_log

```

```
(Emp_ID,Emp_Name,Emp_Sal,Audit_Action,Audit_Timestamp)
values (@empid,@empname,@empsal,@audit_action,getdate());
PRINT 'AFTER DELETE TRIGGER fired.'
-----ثم جرب المثال التالي-----
DELETE FROM Employee where emp_id = 5
```

تعديل القوادم

لتعديل القادح يمكننا أن نحذفه ومن ثم نعيد تعريفه كما يمكننا تعديل القادح بتعليمة واحدة.

إذا تغير اسم أي غرض مستخدم في القادح فيجب عندها تعديل القادح بما يتوافق، لذلك قبل تعديل اسم أي غرض يجب عرض قائمة الأغراض المرتبطة به لتحديد وجود أي قوادم متأثرة بهذا التغيير.

كما يمكن تعديل أي قادح يملكه المستخدم من قبل المستخدم نفسه (حكما في قاعدة البيانات الحالية) كما يمكن لمدير النظام أن يغير اسم أي قادح لأي مستخدم.

حذف القوادم

عند حذف قادح ما فإن الجدول أو المنظار الموافق لا يتأثر. أما إذا حذفنا جدولا أو منظارا فإن جميع القوادم المرتبطة ستحذف تلقائيا. يحق لمالك القادح فقط إضافة إلى مدير النظام إمكانية حذف القادح.

نستخدم التعليمة DROP TRIGGER كما يبين المثال التالي:

```
DROP TRIGGER InsteadTrigger
```