

Trigger

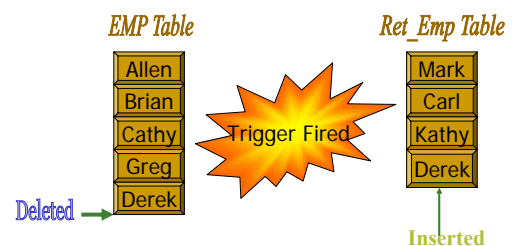
Nội dung:

- Trigger là gì?
- Các dạng Trigger
- Cú pháp tạo Trigger
- Bảng Inserted & Deleted
- Hướng dẫn tạo Trigger
- Trigger AFTER với Insert, Delete và Update
- Trigger INSTEAD OF

Trigger

- Trigger là một dạng thủ tục lưu, chúng được thực thi để đáp ứng lại các thao tác thêm, cập nhật, hoặc xóa trong một bảng.
- Các trigger được kích hoạt một cách tự động khi có một trong các thao tác nói trên xuất hiện.
- Các trigger thường được sử dụng để đảm bảo các quy tắc nhất định.
- Các trigger đảm bảo tính toàn vẹn và tính nhất quán của dữ liệu.

Triggers



Các thao tác thêm, sửa và xóa thì Trigger được kích hoạt để kiểm tra tính toàn vẹn của dữ liệu.

Các dạng Trigger cơ bản

- AFTER Trigger:
 - Insert
 - Delete
 - Update

Không thể tạo cho view
- INSTEAD OF Trigger
Có thể tạo với view hay bảng

Cú pháp để tạo Trigger

```
CREATE TRIGGER trigger_name
ON { table | view }
[ WITH ENCRYPTION ]
{
  { { FOR | AFTER | INSTEAD OF } { [ INSERT ] [, ] [ UPDATE ] }
    [ WITH APPEND ]
    [ NOT FOR REPLICATION ]
    AS
    [ { IF UPDATE ( column )
      [ { AND | OR } UPDATE ( column ) ]
      [ ...n ]
    | IF ( COLUMNS_UPDATED ( ) { bitwise_operator }
      updated_bitmask )
      { comparison_operator } column_bitmask [ ...n ]
    } ]
    sql_statement [ ...n ]
  }
}
```

Bảng Inserted & Deleted

- Là 2 bảng logic được truy cập bởi trigger
- Chứa ảnh của dữ liệu trước và sau khi cập nhật
- Dữ liệu trong bảng không bị tác động bởi thao tác cập nhật sẽ không nằm trong bảng Inserted và Deleted.

Hướng dẫn tạo trigger

- Một trigger có thể kết hợp ba thao tác thực hiện trên một bảng như INSERT, UPDATE, và DELETE.
- Một trigger thích ứng với một bảng.
- Tự chọn WITH ENCRYPTION có thể được sử dụng để ẩn các khai báo của một trigger đối với người sử dụng. Tuy nhiên, một trigger đã được mã hoá sẽ không thể giải mã.
- Một trigger có thể tham chiếu đến một view hoặc một bảng tạm thời, nhưng không thể kết hợp với chúng.

Hướng dẫn tạo trigger

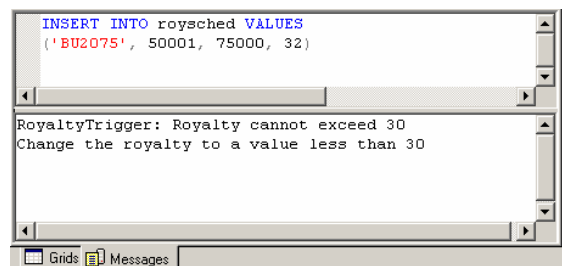
- Một trigger có thể bao gồm các câu lệnh SQL với số lượng bất kì.
- Mặc định, chỉ có người tạo ra cơ sở dữ liệu đó mới có quyền tạo trigger. Quyền này không thể chuyển đổi.
- Một trigger chỉ có thể được tạo trong cơ sở dữ liệu hiện hành. Tuy nhiên, trigger có thể tham chiếu đến các đối tượng nằm trong cơ sở dữ liệu khác.

Trigger INSERT

- Một trigger INSERT được thực thi bất cứ khi nào có thao tác chèn thêm dữ liệu vào một bảng.
- Trigger INSERT thực thi các thao tác dưới đây:
 - Chèn một bản sao của hàng mới vào bảng Inserted.
 - Kiểm tra hàng mới trong bảng Inserted, để xác định xem dữ liệu được chèn vào có hợp lệ hay không.
 - Nếu các giá trị trong bảng Inserted hợp lệ, chèn chúng vào trong bảng trigger.

Trigger INSERT – Ví dụ

```
USE PUBS
-----
CREATE TRIGGER CheckRoyalty
ON roysched
FOR INSERT AS
IF (SELECT royalty FROM inserted) > 30
BEGIN
    PRINT 'RoyaltyTrigger: Royalty cannot exceed 30'
    PRINT 'Change the royalty to a value less than 30'
    ROLLBACK TRANSACTION
END
```



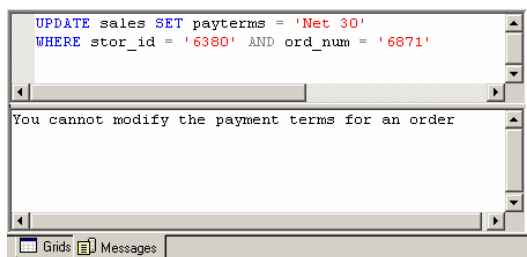
Trigger UPDATE

- Trigger UPDATE được thực hiện khi một tác vụ cập nhật thực thi trên bảng.
- Nó thực thi các tác vụ dưới đây :
 - Di chuyển hàng dữ liệu gốc vào bảng logic Deleted.
 - Chèn một hàng mới vào bảng Inserted, và bảng trigger.
 - Tính toán các giá trị trong các bảng Deleted và Inserted để xác định tính hợp lệ của cần thiết.
- Các trigger UPDATE có thể được tạo ra để xác nhận việc cập nhật trên một cột đơn, hoặc trên toàn bộ bảng.

Trigger UPDATE trên cột

```
USE PUBS
----
CREATE TRIGGER NoUpdatePayterms
ON sales
FOR UPDATE AS
IF UPDATE (payterms)
BEGIN
    PRINT 'You cannot modify the payment
    terms for an order'
    ROLLBACK TRANSACTION
END
```

Trigger UPDATE trên cột



Trigger UPDATE trên bảng

```
CREATE TRIGGER NoUpdateDiscount
ON discounts
FOR UPDATE AS
IF (SELECT discount FROM inserted) > 12
BEGIN
    PRINT 'You cannot assign a discount greater than
    12 percent'
    ROLLBACK TRANSACTION
END
```

```
UPDATE discounts SET discount = 13  
WHERE discounttype = 'Volume Discount'
```

You cannot assign a discount greater than 12 percent

Grids Messages

Trigger DELETE

- Một trigger DELETE được thực hiện mỗi khi xoá các hàng từ một bảng.
- Trigger DELETE thực hiện các công việc sau :
 - Xoá các hàng từ bảng trigger.
 - Chèn các hàng bị xoá vào bảng Deleted.
 - Kiểm tra các hàng trong bảng Deleted để xác định các tác vụ trigger sẽ được thực hiện như thế nào.

Trigger DELETE – Ví dụ

```
USE PUBS  
-----  
CREATE TRIGGER NoDelete9901  
ON pub_info  
FOR DELETE AS  
IF (SELECT pub_id FROM deleted) = 9901  
BEGIN  
    PRINT 'You cannot delete the details of publisher 9901'  
    ROLLBACK TRANSACTION  
END
```

```
DELETE pub_info WHERE pub_id = 9901
```

You cannot delete the details of publisher 9901

Grids Messages

Trigger INSTEAD OF

- Đảm bảo toàn vẹn tham chiếu
- Khi một thay đổi xuất hiện trong một bảng, các trigger INSTEAD OF sửa đổi dữ liệu trong các bảng liên quan.
- Trigger dạng này không thể được sử dụng để thực hiện việc cập nhật dây chuyền hoặc xoá mà ảnh hưởng đến các ràng buộc của các khoá chính và khoá ngoài.
- Thực thi sau khi ràng buộc được kiểm tra

Trigger INSTEAD OF

- Một trigger INSTEAD OF chứa đoạn mã thay thế các lệnh thao tác với cơ sở dữ liệu gốc.
- Các trigger INSTEAD OF hữu dụng nhất khi việc sửa đổi dữ liệu được thực thi trên view mà chúng không thể cập nhật theo cách thông thường.
- Các trigger INSTEAD OF chỉ có thể dựa trên một tác vụ sửa đổi dữ liệu

INSTEAD OF Insert – Ví dụ 1:

```
Create trigger InsertEmpTerrInstead
On EmployeeTerritories WITH ENCRYPTION
Instead of insert
AS
Begin
    Select EmployeeID, TerritoryID From INSERTED
    Select EmployeeID, TerritoryID From DELETED
    Select * From EmployeeTerritories
    Where EmployeeID=(Select EmployeeID From Inserted)
End
--
Insert into EmployeeTerritories values(1,40222)
```

INSTEAD OF Insert – Ví dụ 2:

```
CREATE TABLE BaseTable
(PrimaryKey int IDENTITY(1,1),
Color nvarchar(10) NOT NULL,
Material nvarchar(10) NOT NULL,
ComputedCol AS (Color + Material)
)
GO
--Create a view that contains all columns from the base table.

CREATE VIEW InsteaView
AS SELECT PrimaryKey, Color, Material, ComputedCol
FROM BaseTable
```

INSTEAD OF Insert – Ví dụ 2:

--Create an INSTEAD OF INSERT trigger on the view.

```
CREATE TRIGGER InsteadTrigger on InsteadView
INSTEAD OF INSERT
AS
BEGIN
    --Build an INSERT statement ignoring inserted.PrimaryKey and
    --inserted.ComputedCol.
    INSERT INTO BaseTable
        SELECT Color, Material
    FROM inserted
END
```

INSTEAD OF Insert – Ví dụ 2:

--Minh họa phản ứng của Trigger.

```
INSERT INTO BaseTable (Color, Material)
VALUES ('Red', 'Cloth')
----
INSERT INTO InsteadView (PrimaryKey, Color, Material, ComputedCol)
VALUES (999, 'Blue', 'Plastic', 'XXXXXX')
```

--View the results of the INSERT statement.

```
SELECT PrimaryKey, Color, Material, ComputedCol
FROM InsteadView
--
select * from BaseTable
```

INSTEAD OF Delete – Ví dụ 2:

```
Create table T1
(a varchar(2) primary key,
b varchar(2) not null,
c varchar(2) not null,
)
```

```
-----
Create table T2
(a varchar(2) primary key,
b varchar(2) not null,
c varchar(2) not null,
)
```

```
-----
Insert T1 values('a1','b1','c1')
Insert T1 values('a2','b2','c2')
Insert T1 values('a3','b3','c3')
Insert T1 values('a4','b4','c4')
```

INSTEAD OF Delete– Ví dụ 2:

```
Create trigger DeleteT1
On T1
Instead of delete
AS
Begin
    Insert T2
        select * from Deleted
    Delete from T1 where T1.a= (select deleted.a from deleted)
End
```

----Minh họa phản ứng của Trigger:

```
Delete from T1 where a='a1'
Select * from T2
```

Bài tập: Viết hàm cho phép phục hồi dữ liệu của bảng T1 từ bảng T2.

Trigger và Hiệu suất

- Chi phí dành cho các trigger là nhỏ.
- Hầu hết thời gian đòi hỏi để chạy một trigger là dành cho việc tham chiếu đến các bảng khác.
- Bảng Deleted và inserted luôn tồn tại trong bộ nhớ