

3.5. TRIGGER

Khái niệm Trigger

- Trigger là một kiểu stored procedure đặc biệt
 - Trigger không có tham số đầu vào, đầu ra
 - Không thể thực thi bằng tay bằng lệnh EXECUTE
 - Trigger được kích hoạt thực hiện một cách tự động khi có các sửa đổi trên dữ liệu (INSERT, UPDATE, DELETE) hoặc các sửa đổi lược đồ dữ liệu (CREATE, ALTER, DROP) liên quan đến Trigger

Khái niệm Trigger (tiếp)

- Trigger được sử dụng trong việc
 - Kiểm tra dữ liệu nhập, đảm bảo sự toàn vẹn cho dữ liệu bằng cách ngăn không cho những thay đổi không nhất quán được thực hiện
 - **Ví dụ:** Lương của nhân viên không thể cao hơn lương của người quản lý
 - Dùng để tính toán, cập nhật dữ liệu tự động
 - **Ví dụ:** Tính lại giá trị trường **ThanhTien** trong bảng **DonHang** khi giá trị **DonGia** của sản phẩm được thay đổi.

Các loại Trigger

- Trigger được chia thành 2 nhóm
 - DML triggers (hay Standard Triggers): thực thi khi người sử dụng sửa đổi dữ liệu thông qua các lệnh thao tác dữ liệu INSERT, UPDATE, DELETE trên bảng hoặc View
 - DDL Triggers: thực thi khi có các sự kiện định nghĩa lược đồ dữ liệu thông qua các lệnh CREATE, ALTER và DROP.

Một số quy tắc

- Trigger chỉ được thực thi khi phát biểu buộc chúng kích hoạt được hoàn tất
 - VD: Update trigger không được thực thi cho đến khi phát biểu Update được hoàn tất
- Nếu một phát biểu cố gắng thực thi một hoạt động mà gây xung đột ràng buộc trên bảng hoặc một số lỗi khác thì trigger sẽ không được kích hoạt.
- Một trigger là một phần của giao dịch đơn với phát biểu gọi nó. Vậy nên một phát biểu quay lui trong trigger sẽ thực hiện quay lui cả trigger và sự kiện thay đổi dữ liệu.
 - VD: Nếu người dung ngắt kết nối SQL server tự động thực hiện quay lui toàn bộ giao dịch
- Trigger chỉ kích hoạt một lần cho một phát biểu, ngay cả khi phát biểu đó ảnh hưởng đến nhiều dòng dữ liệu

Tạo DML Trigger

- Cú pháp:

```
CREATE TRIGGER Tên_trigger  
ON Tên_bảng_hoặc_tên_view [WITH  
ENCRYPTION]  
{FOR | AFTER | INSTEAD OF}  
{[DELETE] [,] [INSERT] [,] [UPDATE]}  
AS Câu_lệnh_SQL
```

Tạo DML Trigger (tiếp)

- AFTER:

- Trigger được thực thi sau khi tất cả các câu lệnh SQL gây ra trigger được thực thi thành công
- AFTER là kiểu mặc định nếu từ khóa FOR được dùng
- Không thể định nghĩa AFTER Trigger cho view

- INSTEAD OF:

- Trigger được thực thi thay cho các câu lệnh SQL gây ra trigger
- INSTEAD OF trigger dùng được cho cả bảng và view

Tạo DML Trigger (tiếp)

- DELETE, INSERT, UPDATE:

- Xác định câu lệnh mà khi thực thi trên bảng hoặc view sẽ gây ra Trigger

- WITH ENCRYPTION:

- Mã hóa nội dung, ngăn chặn người dùng khác xem nội dung của trigger

Chú ý

- SQL Server tạo ra 2 bảng logic đặc biệt lưu trong bộ nhớ có cùng cấu trúc với bảng mà trên đó trigger được tạo
 - Bảng **INSERTED**: chứa các giá trị đang được add vào bảng
 - Bảng **DELETED**: chứa các giá trị đang bị xóa từ bảng
- Với **INSERT** Trigger, chỉ có bảng **INSERTED** được tạo
- Với **DELETE** Trigger, chỉ có bảng **DELETED** được tạo
- Với **UPDATE** trigger, cả 2 bảng **INSERTED** và **DELETED** được tạo

Chú ý (tiếp)

- Trong câu lệnh SQL của trigger, có thể thực hiện truy vấn 2 bảng INSERTED và DELETED để kiểm tra điều kiện toàn vẹn dữ liệu
- Trong câu lệnh SQL của trigger, có thể sử dụng:
 - IF UPDATE(tên cột): được dùng trong các trigger INSERT, UPDATE được dùng để kiểm tra xem có các sửa đổi trên cột chỉ định không
 - IF UPDATE(tên cột) [{AND | OR} UPDATE (tên_cột)]: được dùng khi kiểm tra các sửa đổi trên nhiều cột

INSERT TRIGGER

- Thực hiện bất cứ khi nào có lệnh INSERT được thực hiện.
- Bản sao của dữ liệu mới được thêm vào INSERTED table.
- Kiểm tra xem những bản ghi mới đó có hợp lệ hay không.
- Nếu hợp lệ thì thêm bản ghi đó vào trigger table

Ví dụ: cho bảng SinhVien (MaSV, HoTen, MaK, DiaChi, NgaySinh, DiemTB), HOC (MaSV, MaMH, Diem). Viết 1 trigger đảm bảo rằng DTB của SV phải ≥ 0 và ≤ 10

DELETE TRIGGER

- Thực hiện bất cứ khi nào có lệnh DELETE được thực hiện.
- Bản sao của dữ liệu bị xóa được thêm vào DELETED table.
- Kiểm tra xem những bản ghi bị xóa đó có hợp lệ hay không.
- Nếu hợp lệ thì xóa bản ghi đó khỏi trigger table

Ví dụ: Viết 1 trigger không cho phép xóa cùng lúc >2 sinh viên

UPDATE TRIGGER

- Thực hiện bất cứ khi nào có lệnh UPADTE được thực hiện.
- Bản sao của dữ liệu trước khi sửa đổi được thêm vào DELETED table.
- Bản sao của dữ liệu sau khi sửa đổi được thêm vào INSERTED table.

Ví dụ: Viết 1 trigger không cho phép cập nhật những sinh viên có địa chỉ tại Hà Nội

Insert trigger: ví dụ

- VD1: Tạo trigger hiển thị thông báo mỗi khi thực hiện chèn thành công một bản ghi vào bảng KhachHang của CSDL QuanLyKhachHang

```
CREATE TRIGGER insert_trigger_KH
```

```
on KhachHang
```

```
FOR INSERT
```

```
AS Print N'Bạn đã chèn thành công khách hàng'
```

```
insert into KhachHang values (N'Phạm Văn
```

```
HoàngA', 'Nam', N' Đống Đa', 'hoang@jhf', '3545')
```

Insert trigger: ví dụ

- VD2: Tạo trigger thực hiện tự động tính trường ThanhTien của bảng SP_DonHang(IDDonHang, IDSanPham, SoLuong, ThanhTien) khi thêm một bản ghi mới gồm IDDonHang, IDSanPham và SoLuong

```
Create trigger ThanhTien
on SP_donhang
For INSERT As
    if ((select IDSanpham from inserted) is not null)
Begin
    Update SP_DonHang
    Set thanhTien= soLuong* DonGia
    From SanPham, (select IDSanpham, IDDonHang from inserted)
                                as I
    where SanPham.IDSanPham=SP_DonHang.IDSanpham and
           SP_DonHang.IDSanpham=I.IDSanpham and
           SP_DonHang.IDDonhang=I.IDDonhang
end
```

Insert trigger: ví dụ

- Trigger trong ví dụ sẽ được kích hoạt khi câu lệnh insert vào bảng SP_DonHang được thực hiện. Ví dụ

```
insert into SP_DonHang(IDDonHang, IDSanPham, SoLuong)  
values (1,3,2)
```


Insert trigger: ví dụ

VD3. Cho bảng NHANVIEN (MaNV, Hoten, Ngaysinh,...).Viết một trigger để đảm bảo rằng khi thêm một nhân viên mới vào thì tuổi của nhân viên không được ≥ 45

```
create trigger check_age
on Nhanvien
after insert
as
begin
declare @tuoi_nvmoi int
set @tuoi_nvmoi = ( select year(getdate())-year(Ngaysinh) as
tuoi_nv from inserted)
if (@tuoi_nvmoi > 45)
begin
raiserror ('nhan vien khong duoc qua 45 tuoi',16,1)
end
end
```

Insert trigger: ví dụ

Thực hiện:

```
insert into Nhanvien(MaNV,TenNV,Ngaysinh,Diachi,MaP,Luong,MATP)
values ('NV7','Nguyen TUAN A','1929-1-3','Ha
Noi','P1',1000000,'024')
```

Kết quả:

Msg 50000, Level 16, State 1, Procedure check_age, Line 10
nhan vien khong duoc qua 45 tuoi

(1 row(s) affected)

Update trigger: ví dụ

- VD4: Tạo trigger hiển thị thông báo mỗi khi thực hiện cập nhật thành công một bản ghi của bảng KháchHang của CSDL QuanLyKhachHang

```
CREATE TRIGGER update_trigger_KH  
ON KháchHang  
FOR Update  
AS  
Print N'Bạn đã cập nhật thành công bảng KháchHang'
```

Update trigger: ví dụ 2

- VD5. Tạo trigger kiểm tra nếu người dùng muốn sửa

IDKhachHang của bảng khách hàng thì không cho phép và hiển

thị thông báo

```
CREATE TRIGGER update_IDKH
```

```
ON KháchHang
```

```
FOR Update AS
```

```
If UPDATE(IDKhachHang)
```

```
Begin
```

```
Print N'Không thể thay đổi giá trị của trường IDKhachHang'
```

```
ROLLBACK TRANSACTION
```

```
End
```

Update trigger: ví dụ 2

- Trigger trong ví dụ 2 sẽ được kích hoạt khi có người dùng muốn sửa đổi trường IDKhachHang trong bảng KhachHang

```
Update KhachHang
set IDKhachHang = 2
where IDKhachHang = 1
```

Delete trigger: ví dụ

- Tạo trigger hiển thị thông báo mỗi khi thực hiện xóa thành công một bản ghi của bảng KháchHang của CSDL QuanLyKhachHang

```
CREATE TRIGGER delete_trigger_KH
ON KháchHang
FOR DELETE
AS
Print N'Bạn đã xóa thành công bảng KháchHang'
```

Delete trigger: ví dụ 2

- Tạo trigger sao cho khi xóa 1 đơn hàng trong bảng DonHang, tất cả các dòng tương ứng trong bảng SP_DonHang cũng bị xóa

```
CREATE TRIGGER delete_trigger_DonHang
ON DonHang
FOR DELETE
AS
BEGIN
    DELETE from SP_DonHang
    where IDDonHang = (Select IDDonHang from Deleted)
    Print N'Xóa thành công'
END
```

Sửa đổi Trigger

- Cú pháp:

ALTER TRIGGER Tên_trigger

ON Tên_bảng_hoặc_tên_view

[WITH ENCRYPTION]

{FOR | AFTER | INSTEAD OF}

{[DELETE] [,] [INSERT] [,] [UPDATE]}

AS Câu_lệnh_SQL

Xóa Trigger

- Cú pháp:

DROP TRIGGER Tên_trigger

INSTEAD OF TRIGGER

- Trên read-only view không thể thực hiện được các thao tác INSERT, UPDATE, DELETE.
- Khắc phục: sử dụng INSTEAD OF TRIGGER.
- INSTEAD OF TRIGGER chứa mã lệnh thay thế cho những câu lệnh nguyên thủy INSERT, UPDATE, DELETE trên view

INSTEAD OF TRIGGER

Ví dụ: Tạo một view có tên NV_P_view gồm TenNV, TenP.

Sử dụng lệnh DELETE để xóa một nhân viên trên bảng cơ sở NHANVIEN thông qua view trên.

⇒ Có thể xóa được không?

⇒ Phải làm thế nào để xóa được?

INSTEAD OF Trigger

- INSTEAD OF trigger thường dùng cho View nhằm:
 - Cập nhật nhiều bảng một lúc trong một khung nhìn
 - Tăng điều kiện ràng buộc trên các thuộc tính so với CHECK
 - Đánh giá trạng thái của bảng trước hoặc sau khi cập nhật dữ liệu và thực thi một số nhiệm vụ như in thông báo lỗi, sửa đổi bảng khác
 - Cho phép một phần tập hợp câu lệnh bị từ chối trong khi các phần còn lại vẫn được thực thi thành công
- INSTEAD OF Trigger cũng sử dụng các bảng logic Inserted, Deleted để lưu những thay đổi về dữ liệu khi Trigger đang được thực thi

INSTEAD OF Trigger: ví dụ

- Có View Hang_LoiHang lấy thông tin từ 2 bảng MatHang và LoiHang

Create View View_Hang_LoiHang

AS

Select mathang.mahang, tenhang, macongty, soluong,
donvitinh, giahang, mathang.maloaihang,
loaihang.tenloaihang

from mathang inner join loaihang on

mathang.maloaihang=loaihang.maloaihang

INSTEAD OF Trigger: ví dụ

- Tạo trigger trên View NV_DV dùng để chèn dữ liệu vào các bảng tương ứng khi chèn một bản ghi vào view

```
Create Trigger trigger_chen_Hang_LoaiHang
On View_Hang_LoaiHang Instead of Insert
As
Begin
    Insert into loaihang(maloaihang,tenloaihang)
    Select maloaihang,tenloaihang from Inserted
    Insert into mathang (mahang, tenhang, macongty,
                        soluong, donvitinh, giahang,maloaihang)
    Select mahang, tenhang, macongty, soluong,
donvitinh,          giahang, maloaihang from
Inserted
End
```