



TRIGGER

Sau bài học này, sinh viên có thể:

- Hiểu được trigger là gì, công dụng của nó
- Tạo trigger.
- Xoá trigger
- Thay đổi trigger



Nội dung bài học

- ❖ Giới thiệu
- ❖ Tạo Trigger
- ❖ Các loại Trigger
- ❖ Quản lý Trigger
- ❖ Bài tập



Giới thiệu

- Trigger là một **stored procedure** đặc biệt, nó được định nghĩa để tự động thực thi khi có một câu lệnh **Update**, **Insert**, hoặc **Delete** được phát ra trên bảng hoặc View.
- Trigger là một công cụ mạnh dùng để ràng buộc các qui tắc quản lý một cách tự động khi dữ liệu bị hiệu chỉnh.
- Trigger tự động thực thi, không thể gọi một trigger thi hành một cách trực tiếp.



Giới thiệu

- Ràng buộc toàn vẹn dữ liệu cho phù hợp với mô hình quan hệ CSDL.
- Kiểm soát dữ liệu hiện tại khi có thay đổi đến giá trị trong mẫu tin của bảng.
- Kiểm tra dữ liệu nhập vào phù hợp với mỗi quan hệ dữ liệu giữa các bảng.
- Định nghĩa thông báo lỗi của người dùng.
- So sánh trạng thái dữ liệu trước và sau hiệu chỉnh.



Tạo trigger

```
CREATE TRIGGER tên_trigger
ON tên_bảng
{FOR|AFTER|INSTEAD OF}{ [ INSERT ] [, ] [ UPDATE
] [, ] [ DELETE ] }
AS
[ { IF UPDATE ( tên_cột )
[ AND UPDATE ( tên_cột ) | OR UPDATE ( tên_cột )
... ]
các_câu_lệnh_của_trigger
```

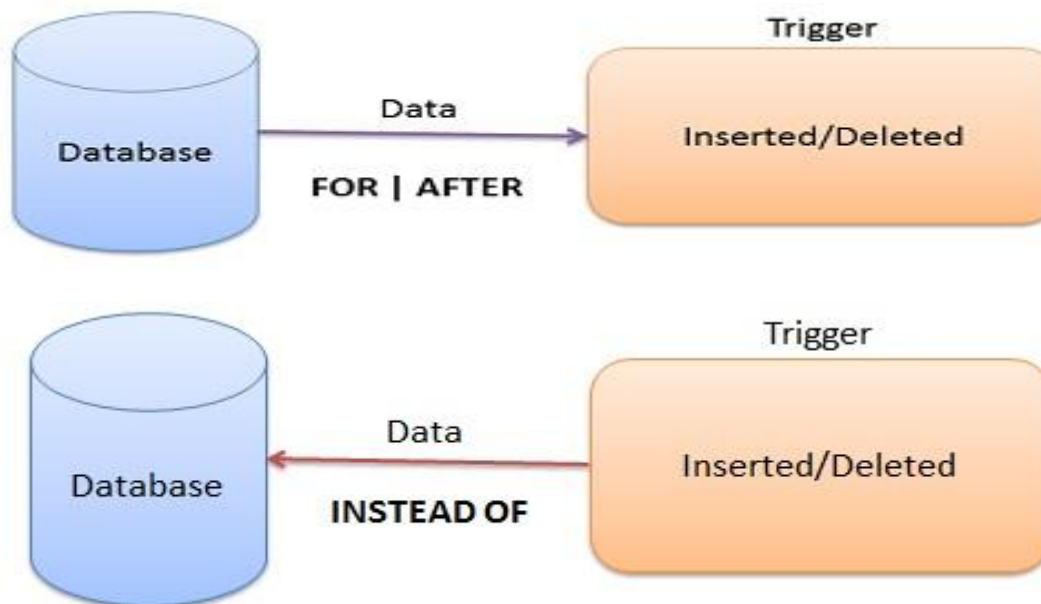


Tham số

- **trigger_name**: nếu trong 1 bảng có nhiều trigger thì tên các trigger phải khác nhau.
- **ON tên_table**: chỉ định table được áp dụng trigger.

Tham số (tt)

FOR, AFTER, INSTEAD OF { [DELETE] [,] [INSERT] [,] [UPDATE] } : chỉ định action gắn với trigger. Đối với INSTEAD OF triggers, action **DELETE** không cho phép trên table mà có relationship mà chỉ định **CASCADE ON DELETE**. Tương tự, action **UPDATE** không cho phép trên table có relationships mà **CASCADE ON UPDATE**.
Triggers không trả kết quả về.





Các loại Trigger

- **Delete trigger**: được thực thi khi có mẫu tin bị xóa, table **Deleted** được sinh ra, để lưu mẫu tin bị xóa, ta có thể truy xuất dữ liệu trong table Deleted.
- **Insert trigger**: được thực thi khi chèn mẫu tin, table **Inserted** được sinh ra, để lưu mẫu tin sắp chèn, ta có thể truy xuất dữ liệu trong table Inserted .
- **Update trigger**: được thực thi khi cập nhật mẫu tin, có 2 table **Inserted** và **Deleted được sinh ra**. Mẫu tin bị cập nhật sẽ lưu trong bảng Inserted (chứa giá trị mới) và Deleted (chứa giá trị cũ)



Quản lý trigger

- **Alter Trigger**: hiệu chỉnh trigger
- **Drop Trigger**: xóa trigger
- **Sp_rename**: đổi tên
- **Sp_helptrigger, Sp_helptext**: xem code trigger
- **DISABLE / ENABLE TRIGGER** trong câu lệnh Alter Table, để cho phép/không cho phép trigger thực thi.

Ví dụ: Tạo một trigger không được xóa nhiều hơn 1 khoa

```
create trigger trgKhoaDelte
on KHOA
for delete
as
if(Select COUNT(*) from deleted ) >1
begin
    raiserror ('ban khong the xoa khoa hon 1 khoa',16,1)
    rollback transaction
end
go
delete from KHOA where MAKHOA = 'a' or MAKHOA = 'b'
```



Ví dụ trigger dạng insert

- Xây dựng trigger nếu số lượng điểm của 1 sinh viên trong bảng điểm > 5 cột điểm thì không cho thêm vào nữa

Create **Trigger** trglon5sv

on ketqua

for **insert**

as | declare @sosv int ,

@masv nvarchar(20)

select @masv=masv from **inserted**

select @sosv = COUNT(*) from KETQUA where
MASV = @masv

if(@sosv>5)

begin

print ' lon hon 5 khong them dc nua'

rollback transaction

end



Ví dụ update

- Số cán bộ giảng dạy ≥ 10



```
create trigger trg_Socbgd
```

```
on KHOA
```

```
AFTER INSERT, UPDATE
```

```
AS
```

```
BEGIN
```

```
    IF(EXISTS (SELECT * FROM inserted  
WHERE SOCBGD <= 10))
```

```
        BEGIN
```

```
            PRINT 'SO CAN BO >=10'
```

```
            ROLLBACK TRAN
```

```
        END
```

```
END
```



Bài tập

- 1) Sinh viên ≥ 18 tuổi
- 2) Điểm thi ≥ 0 và ≤ 10
- 3) Sinh viên chỉ thi 1 môn tối đa 2 lần
- 4) Có thi lần 1 mới thi lần 2



```
CREATE TRIGGER TR_CAU2
```

```
    ON KETQUA
```

```
    AFTER INSERT, UPDATE
```

```
AS
```

```
BEGIN
```

```
    DECLARE @DIEM FLOAT
```

```
    SET @DIEM=(SELECT DIEM FROM INSERTED)
```

```
    IF(@DIEM<0 OR @DIEM>10)
```

```
        BEGIN
```

```
            PRINT 'DIEM KO HOP LE'
```

```
            ROLLBACK TRAN
```

```
        END
```

```
END
```