
Chương 7

Bẫy lỗi - Trigger

Nội dung trình bày

1. Định nghĩa
2. Sử dụng mệnh đề IF UPDATE trong Trigger
3. Sử dụng Trigger tác động nhiều dòng dữ liệu
4. ROLLBACK TRANSACTION

1. Định nghĩa

1. Định nghĩa

Trigger: là một đối tượng gắn liền với một bảng và được tự động kích hoạt khi xảy ra những giao tác làm thay đổi dữ liệu trong bảng.

Định nghĩa một trigger bao gồm các yếu tố sau:

- Trigger sẽ được áp dụng đối với bảng nào?
- Trigger được kích hoạt khi câu lệnh nào được thực thi trên bảng: INSERT, UPDATE, DELETE?
- Trigger sẽ làm gì khi được kích hoạt?

1. Định nghĩa

Cú pháp:

```
CREATE TRIGGER tên_trigger ON tên_bảng  
FOR { [INSERT] [,] [UPDATE] [,] [DELETE] }  
AS  
    các_câu_lệnh_của_trigger
```

INSERTED: Chứa các trường đã insert hoặc update vào bảng

DELETED: Chứa các trường đã xóa khỏi bảng

1. Định nghĩa

Thao tác	Bảng INSERTED	Bảng Deleted
Insert	Các dòng đã được insert	Trống
Update	Các dòng được update mới	Các dòng trước khi update
Delete	Trống	Các dòng đã được xóa

1. Định nghĩa

Ví dụ: Ta có bảng SACH và bảng DonBanHang. Yêu cầu là khi có bán hàng (thêm 1 dòng vào bảng DonbanHang) thì số lượng sách đã bán sẽ được cập nhật lại ở bảng SACH

DonBanHang

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S01	5
KH03	2022-03-05	S02	20

SACH

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	15
S02	Vật lý	150	20
S03	Toán	200	0

Số lượng đã bán sẽ tăng lên

Thêm 1 dòng mới vào bảng DonBanHang



Khi thực hiện insert dữ liệu thì sẽ thực hiện thao tác

```
insert into DonBanHang(makh, ngayban, masach, soluongban)
values('KH04', '5/4/2022', 'S02', 5)
```

1. Định nghĩa

```
create trigger TG_DonBanHang_Insert on DonBanHang
for insert
as
Begin

    declare @MaSach char(3);
    declare @SoLuong int;

    select @MaSach = MaSach, @SoLuong=SoLuongBan from Inserted;

    update SACH set SACH.SoDaBan = SACH.SoDaBan + @SoLuong
    where MaSach=@MaSach;

end
```


1. Định nghĩa

Ví dụ: Ta có bảng SACH và bảng DonBanHang. Yêu cầu là xóa 1 dòng ở bảng DonbanHang thì số lượng sách đã bán sẽ được cập nhật lại ở bảng SACH

DonBanHang

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S01	5
KH03	2022-05-03	S02	10
KH04	2022-04-05	S03	7

SACH

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

Số lượng đã bán sẽ giảm

Xóa 1 dòng ở bảng DonBanHang



Khi thực hiện delete dữ liệu thì sẽ thực hiện thao tác

```
delete from DonBanHang where MaKH= 'KH04'
```

1. Định nghĩa

```
create trigger TG_DonBanHang_Deleted on DonBanHang
for delete
as
Begin

    declare @MaSach char(3);
    declare @SoLuong int;

    select @MaSach = MaSach, @SoLuong=SoLuongBan from Deleted;

    update SACH set SACH.SoDaBan = SACH.SoDaBan - @SoLuong
    where MaSach=@MaSach;

end
```

1. Định nghĩa

Ví dụ: Ta có bảng SACH và bảng DonBanHang. Yêu cầu là cập nhật dữ liệu 1 dòng ở bảng DonbanHang thì số lượng sách đã bán sẽ được cập nhật lại ở bảng SACH

DonBanHang

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S01	5
KH03	2022-03-05	S02	20

SACH

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	15
S02	Vật lý	150	20
S03	Toán	200	0

Số lượng đã bán sẽ tăng/giảm tương ứng

Cập nhật 1 dòng ở bảng DonBanHang



Khi thực hiện cập nhật dữ liệu thì sẽ thực hiện thao tác

```
update DonBanHang set SoLuongBan = 12  
where makh= 'KH03'
```

1. Định nghĩa

```
create trigger TG_DonBanHang_Update on DonBanHang
for update
as
Begin
    update SACH
    set SACH.SoDaBan = SACH.SoDaBan + inserted.SoLuongBan - deleted.SoLuongBan
    from (inserted inner join deleted on inserted.MaSach=deleted.MaSach)
         inner join SACH on SACH.MaSach=inserted.MaSach;
end
```

2. Sử dụng mệnh đề IF UPDATE trong Trigger

2. Sử dụng mệnh đề IF UPDATE trong Trigger

IF ... UPDATE

Thay vì yêu cầu TRIGGER kích hoạt khi có sự thay đổi dữ liệu trên toàn bảng. Ta sẽ **chỉ định TRIGGER kích hoạt khi có sự thay đổi dữ liệu trên cột nào đó của bảng thôi.**

2. Sử dụng mệnh đề IF UPDATE trong Trigger

```
create trigger TG_DonBanHang_Update on DonBanHang
for update
as
begin
    if update(SoLuongBan)
        update SACH
        set SACH.SoDaBan = SACH.SoDaBan + inserted.SoLuongBan - deleted.SoLuongBan
        from (inserted inner join deleted on inserted.MaSach=deleted.MaSach)
            inner join SACH on SACH.MaSach=inserted.MaSach;
end
```

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

Với các viết Trigger ở slide trước, thì Trigger sẽ chỉ có tác động đến 1 dòng.

Thực tế thì khi thực hiện các thao tác Delete hoặc Update thì sẽ thường xuyên xảy ra trường hợp xóa hoặc cập nhật nhiều dòng dữ liệu 1 lúc.

Hoặc với thao tác Insert dạng: insert into ... Select cũng sẽ thêm nhiều dòng dữ liệu vào bảng 1 lúc.

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

Cách giải quyết có thể tác động nhiều dòng dữ liệu khi Trigger được gọi:

- Sử dụng truy vấn con
- Sử dụng biến con trỏ

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

3.1. Sử dụng truy vấn con

DonBanHang

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S02	5
KH03	2022-05-10	S02	10
KH04	2022-04-05	S03	7

SACH

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

update DonBanHang set SoLuongBan=10
where MASACH = 'S03'

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S02	5
KH03	2022-05-10	S02	10
KH04	2022-04-05	S03	10

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	10

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

3.1. Sử dụng truy vấn con

DonBanHang

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S02	5
KH03	2022-05-10	S02	10
KH04	2022-04-05	S03	7

SACH

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

update DonBanHang set SoLuongBan=10
where MASACH = 'S02'



Cập nhật 2 dòng -> Chạy sai

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S02	10
KH03	2022-05-10	S02	10
KH04	2022-04-05	S03	7

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

```
create trigger TG_DonBanHang_Update on DonBanHang
for update
as
begin
    if update(SoLuongBan)
    begin
        update SACH
        set SACH.SoDaBan = SACH.SoDaBan +
            (select sum(Inserted.SoLuongBan - Deleted.SoLuongBan)
             from Inserted inner join Deleted
              on Inserted.MaSach = Deleted.MaSach and
               Inserted.MaKH = Deleted.MaKH
             where Inserted.MaSach=SACH.MaSach)
        where SACH.MaSach in(select MaSach from Inserted);
    end
end
```

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

3.1. Sử dụng truy vấn con

DonBanHang

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S02	5
KH03	2022-05-10	S02	10
KH04	2022-04-05	S03	7

SACH

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

update DonBanHang set SoLuongBan=10
where MASACH = 'S02' → Cập nhật 2 dòng -> OK

MaKH	NgayBan	MaSach	SoLuongBan
KH01	2022-01-05	S01	10
KH02	2022-02-05	S02	10
KH03	2022-05-10	S02	10
KH04	2022-04-05	S03	7

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

3.2. Sử dụng con trỏ CURSOR

Con trỏ giúp ta duyệt qua tập dữ liệu lần lượt từng dòng. Do vậy thường sẽ sử dụng kết hợp con trỏ với vòng lặp **WHILE**

Sử dụng con trỏ trải qua 4 bước:

Bước 1: Khai báo con trỏ, trỏ đến 1 tập dữ liệu (Lấy được từ câu lệnh select, khai báo bằng từ khóa declare).

Ví dụ duyệt qua danh sách các Sách có trong bảng SACH

DECLARE ConTro **CURSOR FOR** select MaSach, TenSach from SACH

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

Bước 2: Trước khi đọc dữ liệu từng dòng, phải mở con trỏ trước:

```
OPEN ConTro;
```

Bước 3: Khi con trỏ được mở, thì con trỏ mặc định trỏ đến dòng đầu tiên của tập dữ liệu, khi đó có thể đọc dữ liệu bằng lệnh **FETCH**.

```
FETCH NEXT FROM ConTro INTO @MaSach, @TenSach
```

Đọc dữ liệu dòng con trỏ đang trỏ đến và lưu vào 2 biến @MaSach và @TenSach
Để tiếp tục đọc các dòng tiếp theo phải sử dụng vòng lặp WHILE, vòng lặp sẽ lặp khi con trỏ xác định còn dòng dữ liệu chuyển đến

```
WHILE @@FETCH_STATUS=0
```

```
BEGIN
```

```
    print @MaSach + ' ' + @TenSach;
```

```
    FETCH NEXT FROM ConTro INTO @MaSach, @TenSach;
```

```
END
```

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

Bước 4: Khi dùng xong cần đóng con trỏ và giải phóng tài nguyên mà con trỏ đang nắm giữ:

```
CLOSE ConTro;
```

```
DEALLOCATE ConTro;
```

3. Sử dụng Trigger tác động nhiều dòng dữ liệu

```
create trigger TG_DonBanHang_Update on DonBanHang
for update
as
begin
    if update(SoLuongBan)
    begin
        declare @MaSach char(5);
        declare @SoLuong int;
        declare contro Cursor FOR
        select inserted.MaSach,inserted.soluongban-deleted.soluongban as soluong
        from inserted inner join deleted on inserted.MaSach=deleted.MaSach and
                                                inserted.MaKH=deleted.MaKH;

        open contro;
        fetch next from contro into @MaSach,@SoLuong;
        while @@FETCH_STATUS=0
        begin
            update SACH
            set SACH.SoDaBan = SACH.SoDaBan + @SoLuong
            where SACH.MaSach=@MaSach;
            fetch next from contro into @MaSach,@SoLuong;
        end
        close contro;
        deallocate contro;
    end
end
```

4. ROLLBACK TRANSACTION

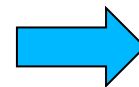
4. ROLLBACK TRANSACTION

Trigger có khả năng nhận biết được sự thay đổi dữ liệu trên bảng, từ đó có thể phát hiện và hủy bỏ các dữ liệu nào không đảm bảo tính toàn vẹn dữ liệu. Trong Trigger muốn hủy bỏ tác dụng của câu lệnh làm kích hoạt Trigger, ta sử dụng:

ROLLBACK TRANSACTION;

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

```
create trigger TG_DonBanHang_Delete
on DonBanHang
for delete
as
begin
    Rollback Transaction;
end
```



```
delete from Sach
where MaSach= 'MS03';
```



Không thể xóa do Trigger đã hủy những thay đổi của lệnh delete

4. ROLLBACK TRANSACTION

Ví dụ: Viết Insert Trigger kiểm tra ràng buộc sau trên bảng SACH:

- Số lượng SX phải lớn hơn hoặc bằng số đã bán
- Số lượng SX phải lớn hơn hoặc bằng 0
- Số đã bán phải lớn hơn hoặc bằng 0

MaSach	TenSach	SoLuongSX	SoDaBan
S01	Âm nhạc	100	10
S02	Vật lý	150	15
S03	Toán	200	7

4. ROLLBACK TRANSACTION

```
create trigger TG_Sach_Insert on SACH
for insert
as
begin
    declare @SoLuongSX int,@SoDaBan int;
    select @SoLuongSX=SoLuongSX,@SoDaBan=SoDaBan
    from inserted;
    if @SoLuongSX<@SoDaBan
    begin
        print N'Số lượng sản xuất nhỏ hơn số đã bán';
        rollback transaction;
    end
    else if @SoLuongSX<0
        begin
            print N'Số lượng sản xuất nhỏ hơn 0';
            rollback transaction;
        end
    else if @SoDaBan<0
        begin
            print N'Số lượng đã bán nhỏ hơn 0';
            rollback transaction;
        end
    end
end
```

Câu hỏi ???