



Chương 8

TRIGGER



Mục tiêu

- Sau khi học xong chương này, SV có thể:
 - Biết Trigger là gì và khi nào sử dụng Trigger.
 - Nắm được các loại Trigger.
 - Nắm vững cú pháp tạo, xóa và cập nhật Trigger.
 - Tạo được Trigger khi thêm, xóa và cập nhật dữ liệu.

Nội dung chi tiết

- Trigger là gì?
- Khi nào sử dụng Trigger?
- Các loại Trigger
- Các bảng tạm Inserted và Deleted
- Tạo - Xóa - Cập nhật Trigger
- Tạo Trigger khi thêm, xóa và cập nhật dữ liệu

Trigger là gì?

- Trigger là một Stored Procedure không có tham số.
- Trigger thực thi một cách tự động khi một trong ba câu lệnh Insert, Update, Delete làm thay đổi dữ liệu trên bảng có chứa Trigger.

Khi nào sử dụng Trigger?

- Trigger thường được sử dụng để:
 - Kiểm tra các ràng buộc dữ liệu phức tạp mà không thể dùng constraint trên nhiều bảng hoặc trên nhiều dòng của bảng.
 - Gọi hành động "Rollback Tran" để hoàn tác, hủy thao tác cập nhật khi vi phạm ràng buộc.
 - Tính toán, tự động cập nhật giá trị để đảm bảo tính toàn vẹn dữ liệu.

Ưu điểm của Trigger

- Sử dụng Trigger để kiểm tra tính toàn vẹn của CSDL.
- Trigger có thể bắt lỗi logic ở mức CSDL.
- Có thể dùng Trigger là một cách khác để thay thế việc thực hiện những công việc định giờ theo lịch.
- Trigger rất hiệu quả khi sử dụng để kiểm soát những thay đổi của dữ liệu trong bảng.

Nhược điểm của Trigger

- Trigger chỉ là một phần mở rộng của việc kiểm tra tính hợp lệ của dữ liệu chứ không thể thay thế hoàn toàn được công việc này.
- Trigger hoạt động ngầm trong CSDL, không hiển thị ở tầng giao diện. Do đó, khó chỉ ra được điều gì xảy ra ở tầng CSDL.
- Trigger thực hiện các update lên bảng dữ liệu vì thế nó làm tăng lượng công việc lên CSDL và làm cho hệ thống chạy chậm.

Các loại Trigger

■ **INSTEAD OF Trigger**

- Chạy trước các hành động kiểm tra ràng buộc dữ liệu.
- Dữ liệu chưa bị thay đổi.

■ **FOR (AFTER) Trigger**

- Chạy sau các hành động kiểm tra ràng buộc dữ liệu.
- Dữ liệu đã bị tạm thời thay đổi trong bảng.

Các bảng tạm Inserted và Deleted

- Khi Trigger được thực hiện, SQL tự động tạo ra 2 bảng tạm Inserted và Deleted có cùng cấu trúc với bảng chứa Trigger:
- **Inserted:** chứa dữ liệu được thêm mới khi thực thi câu lệnh Insert hoặc câu lệnh Update.
- **Deleted:** Chứa dữ liệu bị xoá khi thực thi câu lệnh Delete hoặc chứa dữ liệu cũ khi thực thi câu lệnh Update.
- **Hành động Update trong SQL Server:** xoá dữ liệu cũ và thêm dữ liệu mới với thông tin đã cập nhật.
=> Sử dụng cả 2 bảng tạm Inserted và Deleted.

Tạo Trigger

- Cú pháp:

```
CREATE TRIGGER Tên_Trigger ON Tên_bảng  
{ INSTEAD OF | FOR | AFTER } { INSERT | DELETE | UPDATE }  
AS
```

Khởi lệnh xử lý

Quản lý Trigger

- Xóa Trigger:
 - Cú pháp:

DROP TRIGGER Tên_Trigger

- Cập nhật Trigger:
 - Cú pháp:

ALTER TRIGGER ...

Tạo Trigger khi thêm dữ liệu

- Thường dùng để kiểm tra:
 1. Khóa ngoại
 2. Miền giá trị
 3. Liên thuộc tính trong cùng một bảng
 4. Liên thuộc tính của nhiều bảng khác nhau
- 3 loại đầu tiên, chỉ dùng Trigger nếu muốn cung cấp các báo lỗi cụ thể bằng tiếng Việt khi đã khai báo các ràng buộc này bằng constraint.

Tạo Trigger khi thêm dữ liệu

- Các cấu trúc / lệnh thường dùng khi kiểm tra:
 - **if else**
 - **if exists**
 - **raiserror**
 - **rollback tran**

Tạo Trigger khi thêm dữ liệu

- **VD:** Bổ sung cột điểm trung bình (DTB) và số tín chỉ tích lũy (SoTC) trong bảng SINHVIEN.

```
alter table SINHVIEN  
add DTB real, SOTC int  
go
```

- Cập nhật giá trị tại 2 cột này cho các mẫu tin đã có trong bảng SINHVIEN.

```
update SINHVIEN  
set DTB =(select AVG(DIEM)  
          from KETQUA  
          where MASV=SINHVIEN.MASV),  
    SOTC=(select SUM(SOTIET/15)  
          from KETQUA inner join MONHOC on KETQUA.MAMH=MONHOC.MAMH  
          where MASV=SINHVIEN.MASV and DIEM>=5)  
go
```

- Hãy tạo Trigger ứng với hành động thêm mẫu tin mới vào bảng KETQUA: trong đó phải tính lại DTB của SV và kiểm tra nếu Điểm ≥ 5 thì tăng số tín chỉ tích lũy của SV.

Tạo Trigger khi thêm dữ liệu

- VD1: Hãy tạo **Trigger instead of** ứng với hành động **thêm mẫu tin mới** vào bảng KETQUA:

```
create trigger tg_ThemKQ on KETQUA
instead of insert
as
--Khai báo các biến nhận các giá trị của mẫu tin mới sắp thêm từ bảng tạm inserted
declare @masv char(3), @mamh char(2), @diem real
select @masv=MASV, @mamh=MAMH, @diem=DIEM from inserted

--Lần lượt kiểm tra các vi phạm ràng buộc toàn vẹn nếu có trước khi thêm dữ liệu
if not exists(select * from SINHVIEN where MASV=@masv)
begin
    raiserror(N'Mã SV không tồn tại', 16, 1)
    rollback
    return
end
if not exists(select * from MONHOC where MAMH=@mamh)
begin
    raiserror(N'Mã môn học không tồn tại', 16, 1)
    rollback
    return
end
if exists(select * from KETQUA where MASV=@masv and MAMH=@mamh)
begin
    raiserror(N'SV đã có điểm thi môn này', 16, 1)
    rollback
    return
end
```

Tạo Trigger khi thêm dữ liệu

- VD1: Hãy tạo **Trigger instead of** ứng với hành động **thêm mẫu tin mới** vào bảng KETQUA (tt):

```
if @diem not between 0 and 10
begin
    raiserror(N'Điểm phải từ 0 đến 10', 16, 1)
    rollback
    return
end

--Thêm mới mẫu tin vào bảng KETQUA
insert into KETQUA values(@masv,@mamh,@diem)

-- Cập nhật lại ĐTB của SV sau khi thêm kết quả thi môn mới
update SINHVIEN
set DTB=(select AVG(DIEM) from KETQUA where MASV=@masv)
where MASV=@masv

-- Cập nhật lại Số tín chỉ tích lũy của SV sau khi thêm kết quả thi môn mới
if @diem>=5
begin
    declare @sotiet int
    select @sotiet=SOTIET from MONHOC where MAMH=@mamh
    update SINHVIEN set SOTC=SOTC+@sotiet/15 where MASV=@masv
end
go
```


Tạo Trigger khi thêm dữ liệu

- VD2: Hãy tạo **Trigger for** ứng với hành động **thêm mẫu tin mới** vào bảng KETQUA:

```
create trigger tg_ThemKQ on KETQUA
for insert
as
--Khai báo các biến nhận các giá trị của mẫu tin mới đã tạm thêm từ bảng tạm inserted
declare @masv char(3), @mamh char(2), @diem real
select @masv=MASV, @mamh=MAMH, @diem=DIEM from inserted

-- Cập nhật lại ĐTB của SV sau khi thêm kết quả thi môn mới
update SINHVIEN
set DTB=(select AVG(DIEM) from KETQUA where MASV=@masv)
where MASV=@masv

-- Cập nhật lại Số tín chỉ tích lũy của SV sau khi thêm kết quả thi môn mới
if @diem>=5
begin
    declare @sotiet int
    select @sotiet=SOTIET from MONHOC where MAMH=@mamh
    update SINHVIEN set SOTC=SOTC+@sotiet/15 where MASV=@masv
end
go
```

Tạo Trigger khi xóa dữ liệu

- Tương tự Trigger Insert, khi xóa cũng phải kiểm tra vi phạm các ràng buộc toàn vẹn và cập nhật dữ liệu của các bảng liên quan nếu có.
- VD1: Trigger ứng với hành động xóa mẫu tin trong bảng SINHVIEN => Kiểm tra chỉ cho xóa một sinh viên nếu sinh viên này chưa có kết quả thi.
- VD2: Trigger ứng với hành động xóa mẫu tin trong bảng KETQUA => Cập nhật lại điểm trung bình và số tín chỉ tích lũy của SV.
- VD3: Trigger ứng với hành động xóa mẫu tin trong bảng DONHANG => Xóa các chi tiết đơn hàng nếu có và xóa đơn hàng.

Tạo Trigger khi xóa dữ liệu

- VD1: Bổ sung cột số sinh viên (**SOSV**) trong bảng **KHOA**:

```
alter table KHOA  
add SOSV int  
go
```

- Cập nhật giá trị tại cột này cho các mẫu tin đã có trong bảng **KHOA**:

```
update KHOA  
set SOSV =(select COUNT(MASV)  
            from SINHVIEN  
            where MAKH=KHOA.MAKH)  
go
```

- Hãy tạo Trigger ứng với hành động xóa mẫu tin trong bảng **SINHVIEN**: trong đó phải cập nhật lại **Số SV** của Khoa.

Tạo Trigger khi xóa dữ liệu

- VD1a: Hãy tạo **Trigger instead of** ứng với hành động **xóa mẫu tin** trong bảng SINHVIEN:

```
create trigger tg_XoaSV on SINHVIEN
instead of delete
as
--Khai báo các biến nhận các giá trị của mẫu tin sắp xóa từ bảng tạm deleted
declare @masv char(3), @makh char(2)
select @masv=MASV, @makh=MAKH from deleted

--Kiểm tra vi phạm ràng buộc toàn vẹn nếu có trước khi xóa dữ liệu
if exists(select * from KETQUA where MASV=@masv)
begin
    raiserror(N'SV đã có kết quả thi nên không thể xóa', 16, 1)
    rollback
    return
end

--Xóa mẫu tin SV trong bảng SINHVIEN
delete SINHVIEN where MASV=@masv

--Cập nhật lại tổng số SV của Khoa vừa có SV bị xóa
update KHOA set SOSV=SOSV-1 where MAKH=@makh
go
```

Tạo Trigger khi xóa dữ liệu

- VD1b: Hãy tạo **Trigger for** ứng với hành động **xóa mẫu tin** trong bảng SINHVIEN:

```
create trigger tg_XoaSV on SINHVIEN
for delete
as
--Khai báo các biến nhận các giá trị của mẫu tin sắp xóa từ bảng tạm deleted
declare @masv char(3), @makh char(2)
select @masv=MASV, @makh=MAKH from deleted

--Cập nhật lại tổng số SV của Khoa vừa có SV bị xóa
update KHOA set SOSV=SOSV-1 where MAKH=@makh
go
```

Tạo Trigger khi xóa dữ liệu

- VD2a: Hãy tạo **Trigger instead of** ứng với hành động **xóa mẫu tin** trong bảng KETQUA:

```
create trigger tg_XoaKQ on KETQUA
instead of delete
as
--Khai báo các biến nhận các giá trị của mẫu tin sắp xóa từ bảng tạm deleted
declare @masv char(3), @mamh char(2), @diem real
select @masv=MASV, @mamh=MAMH, @diem=DIEM from deleted

--Xóa mẫu tin trong bảng KETQUA
delete KETQUA where MASV=@masv and MAMH=@mamh

-- Cập nhật lại ĐTB của SV sau khi xóa kết quả thi 1 môn của SV
update SINHVIEN
set DTB=(select AVG(DIEM) from KETQUA where MASV=@masv)
where MASV=@masv

-- Cập nhật lại Số tín chỉ tích lũy của SV sau khi xóa kết quả thi 1 môn của SV
if @diem>=5
begin
    declare @sotiet int
    select @sotiet=SOTIET from MONHOC where MAMH=@mamh
    update SINHVIEN set SOTC=SOTC-@sotiet/15 where MASV=@masv
end
go
```

Tạo Trigger khi xóa dữ liệu

- VD2b: Hãy tạo **Trigger for** ứng với hành động **xóa mẫu tin** trong bảng KETQUA:

```
create trigger tg_XoaKQ on KETQUA
for delete
as
--Khai báo các biến nhận các giá trị của mẫu tin sắp xóa từ bảng tạm deleted
declare @masv char(3), @mamh char(2), @diem real
select @masv=MASV, @mamh=MAMH, @diem=DIEM from deleted

-- Cập nhật lại ĐTB của SV sau khi xóa kết quả thi 1 môn của SV
update SINHVIEN
set DTB=(select AVG(DIEM) from KETQUA where MASV=@masv)
where MASV=@masv

-- Cập nhật lại Số tín chỉ tích lũy của SV sau khi xóa kết quả thi 1 môn của SV
if @diem>=5
begin
    declare @sotiet int
    select @sotiet=SOTIET from MONHOC where MAMH=@mamh
    update SINHVIEN set SOTC=SOTC-@sotiet/15 where MASV=@masv
end
go
```

Tạo Trigger khi xóa dữ liệu

- VD3: Hãy tạo **Trigger instead of** ứng với hành động **xóa mẫu tin** trong bảng DONHANG: xóa các chi tiết đơn hàng nếu có và xóa đơn hàng.

```
create trigger tg_XoaDH on DONHANG
instead of delete
as
--Khai báo các biến nhận các giá trị của mẫu tin sắp xóa từ bảng tạm deleted
declare @sodh char(4)
select @sodh=SODH from deleted

--Xóa các chi tiết đơn hàng nếu có
if exists(select * from CTDONHANG where SODH=@sodh)
    delete CTDONHANG where SODH=@sodh

-- Xóa mẫu tin đơn hàng trong bảng DONHANG
delete DONHANG where SODH=@sodh
go
```


Tạo Trigger khi cập nhật dữ liệu

- Không được sửa thông tin khóa chính:
 - Cú pháp kiểm tra cột có bị sửa thông tin không?

if update(Tên_cột)

-- Khối lệnh xử lý

- Khi cập nhật dữ liệu cũng phải **kiểm tra vi phạm các ràng buộc toàn vẹn và cập nhật dữ liệu của các bảng liên quan** nếu có sau khi thực hiện.

Tạo Trigger khi cập nhật dữ liệu

- VD: Hãy tạo **Trigger instead of** ứng với hành động **sửa đổi mẫu tin** trên bảng KETQUA. Trong đó phải kiểm tra:
 - Nếu thay đổi Mã SV và Mã môn học thì báo lỗi không được sửa khóa chính.
 - Nếu Điểm thay đổi thì cập nhật lại cột DTB trong bảng SINHVIEN.
 - Nếu Điểm cũ < 5 và Điểm mới ≥ 5 thì tăng số tín chỉ tích lũy của SV.
 - Nếu Điểm cũ ≥ 5 và Điểm mới < 5 thì giảm số tín chỉ tích lũy của SV.

Tạo Trigger khi cập nhật dữ liệu

- VD1: Hãy tạo **Trigger instead of** ứng với hành động **sửa đổi mẫu tin** trên bảng KETQUA:

```
create trigger tg_SuaKQ on KETQUA
instead of update
as

--Kiểm tra không cho sửa thông tin khóa chính
if update(MASV) or update(MAMH)
begin
    raiserror(N'Không được sửa khóa chính',16,1)
    rollback
    return
end

--Khái báo các biến nhận các giá trị của mẫu tin sắp sửa sửa thông tin từ bảng tạm deleted và inserted
declare @masv char(3), @mamh char(2), @diemcu real, @diemmoi real
select @diemcu=DIEM from deleted
select @masv=MASV, @mamh=MAMH,@diemmoi=DIEM from inserted

--Lần lượt kiểm tra các vi phạm ràng buộc toàn vẹn nếu có trước khi cập nhật dữ liệu
if @diemmoi<0 or @diemmoi>10
begin
    raiserror(N'Điểm sửa phải từ 0 đến 10', 16, 1)
    rollback
    return
end
```

Tạo Trigger khi cập nhật dữ liệu

- VD1: Hãy tạo **Trigger instead of** ứng với hành động **sửa đổi mẫu tin** trên bảng KETQUA (tt):

```
-- Cập nhật mẫu tin trong bảng KETQUA
update KETQUA set DIEM=@diemmoi where MAMH=@mamh and MASV=@masv

-- Cập nhật ĐTB của SV
update SINHVIEN
set DTB=(select AVG(DIEM) from KETQUA where MASV=@masv)
where MASV=@masv

-- Cập nhật lại Số tín chỉ tích lũy của SV sau khi sửa điểm thi mới
declare @sotiet int
select @sotiet=SOTIET from MONHOC where MAMH=@mamh
if @diemcu<5 and @diemmoi>=5
    update SINHVIEN set SOTC=SOTC+@sotiet/15 where MASV=@masv
if @diemcu>=5 and @diemmoi<5
    update SINHVIEN set SOTC=SOTC-@sotiet/15 where MASV=@masv
go
```

Tạo Trigger khi cập nhật dữ liệu

- VD2: Hãy tạo **Trigger for** ứng với hành động **sửa đổi mẫu tin** trên bảng KETQUA:

```
create trigger tg_SuaKQ on KETQUA
for update
as

--Kiểm tra không cho sửa thông tin khóa chính
if update(MASV) or update(MAMH)
begin
    raiserror(N'Không được sửa khóa chính',16,1)
    rollback
    return
end

--Khai báo các biến nhận các giá trị của mẫu tin sắp sửa sửa thông tin từ bảng tạm deleted và inserted
declare @masv char(3), @mamh char(2), @diemcu real, @diemmoi real
select @diemcu=DIEM from deleted
select @masv=MASV, @mamh=MAMH,@diemmoi=DIEM from inserted

-- Cập nhật ĐTB của SV
update SINHVIEN
set DTB=(select AVG(DIEM) from KETQUA where MASV=@masv)
where MASV=@masv

-- Cập nhật lại Số tín chỉ tích lũy của SV sau khi sửa điểm thi mới
declare @sotiet int
select @sotiet=SOTIET from MONHOC where MAMH=@mamh
if @diemcu<5 and @diemmoi>=5
    update SINHVIEN set SOTC=SOTC+@sotiet/15 where MASV=@masv
if @diemcu>=5 and @diemmoi<5
    update SINHVIEN set SOTC=SOTC-@sotiet/15 where MASV=@masv
go
```

Bài tập ứng dụng

1. Sử dụng CSDL Quản lý sinh viên:

- Bổ sung cột tổng số sinh viên (SOSV: int) trong bảng KHOA, cột điểm trung bình (DTB: real) và số tín chỉ tích lũy (SOTC: int) trong bảng SINHVIEN.
- Hãy tạo các Trigger khi thêm – xóa – sửa dữ liệu của các bảng trong CSDL.

Bài tập ứng dụng

2. Sử dụng CSDL Quản lý bán hoa:

- Bổ sung cột tổng số hoa (SOHOA: int) trong 2 bảng LOAIHOA và CHUDE, cột tổng trị giá đơn hàng (TRIGIADH: money) trong bảng DONHANG.
- Hãy tạo các Trigger khi thêm – xóa – sửa dữ liệu của các bảng trong CSDL.

Câu hỏi trắc nghiệm

1. Chọn phát biểu SAI về Trigger?

- a. Trigger có tham số đầu vào và tham số đầu ra.
- b. Trigger là một Stored Procedure không có tham số.
- c. Trigger được sử dụng để kiểm tra các ràng buộc dữ liệu phức tạp mà không thể dùng constraint.
- d. Trigger thực thi một cách tự động khi một trong ba câu lệnh Insert, Update, Delete làm thay đổi dữ liệu trên bảng có chứa Trigger.

Câu hỏi trắc nghiệm

2. Có mấy loại Trigger? Đó là những loại nào?

- a. Có 1 loại Trigger duy nhất để kiểm tra các ràng buộc dữ liệu phức tạp.
- b. Có 2 loại: Trigger Instead of và Trigger For (After).
- c. Có 2 loại: Trigger Inserted và Trigger Deleted.
- d. Có 3 loại: Trigger Inserted, Trigger Deleted và Trigger Updated.

Câu hỏi trắc nghiệm

3. Cú pháp xây dựng Trigger dưới đây có ĐÚNG không?

CREATE TRIGGER Tên_Trigger **ON** Tên_bảng

{ **INSTEAD OF** | **FOR** | **AFTER** } { **INSERT** | **DELETE** | **UPDATE** }

AS

Khởi lệnh xử lý

- a. Đúng
- b. Không

Câu hỏi trắc nghiệm

4. Cú pháp dưới đây dùng để làm gì?

`if update(Tên_cột)`

`-- Khối lệnh xử lý`

- a. Kiểm tra tên cột có chính xác không?
- b. Kiểm tra tên cột có được cập nhật chưa?
- c. Kiểm tra cột có bị đổi tên không?
- d. Kiểm tra cột có bị sửa thông tin không?

Câu hỏi trắc nghiệm

5. Chọn thứ tự ĐÚNG khi tạo Trigger Instead of để thêm dữ liệu mới vào bảng:

1. Cập nhật dữ liệu của các bảng liên quan nếu có.
2. Khai báo các biến nhận các giá trị của mẫu tin mới sắp thêm từ bảng tạm inserted.
3. Lần lượt kiểm tra các vi phạm ràng buộc toàn vẹn nếu có trước khi thêm dữ liệu.
4. Thêm mới mẫu tin vào bảng.

- a. 1, 2, 3, 4.
- b. 4, 3, 2, 1.
- c. 2, 3, 4, 1.
- d. 2, 3, 1, 4.