



TRIGGER

(BẦY LỖI)

Khái niệm Trigger

Trigger là một dạng đặc biệt của Store Procedure, dùng khai báo các ràng buộc dữ liệu cho một Table và tự động thực hiện khi một trong ba phát biểu (INSERT, UPDATE hay DELETE) làm thay đổi dữ liệu trên Table.

Trigger không được gọi trực tiếp như Store Procedure mà tự động thực hiện khi một trong ba phát biểu cập nhật dữ liệu được thực hiện, không có tham số đầu vào và không có giá trị trả về như Store Procedure.

✓ Thường dùng Trigger:

- Tính toán, cập nhật giá trị tự động trên nhiều table có liên quan, đảm bảo dữ liệu luôn được toàn vẹn, kiểm tra dữ liệu nhập.
- Giúp mở rộng khả năng kiểm tra các ràng buộc toàn vẹn không thể dùng constraint,

Khái niệm Trigger

- ❑ Các loại Trigger: **UPDATE Trigger**, **INSERT Trigger**, **DELETE Trigger** (theo hành động sẽ kích hoạt trigger)
- ❑ Trigger khi được tạo, kết hợp với các hành động **Insert**, **Update** hay **Delete** trên bảng, tham gia vào **transaction** khởi tạo bởi câu lệnh cập nhật dữ liệu tương ứng.
- ❑ Gọi hành động **Rollback Tran** để hủy thao tác cập nhật khi vi phạm ràng buộc.

Lệnh tạo Trigger (T- SQL)

CREATE TRIGGER *trigger_name*

Tên trigger
cần tạo

ON { *table* | *view* }

Tạo trigger
trên Table hay
view nào

[WITH <dml_trigger_option> [,...*n*]]

{ FOR | AFTER | INSTEAD OF }

Kiểu trigger
cần tạo

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

AS

{ *sql_statement* [;] [,...*n*] }

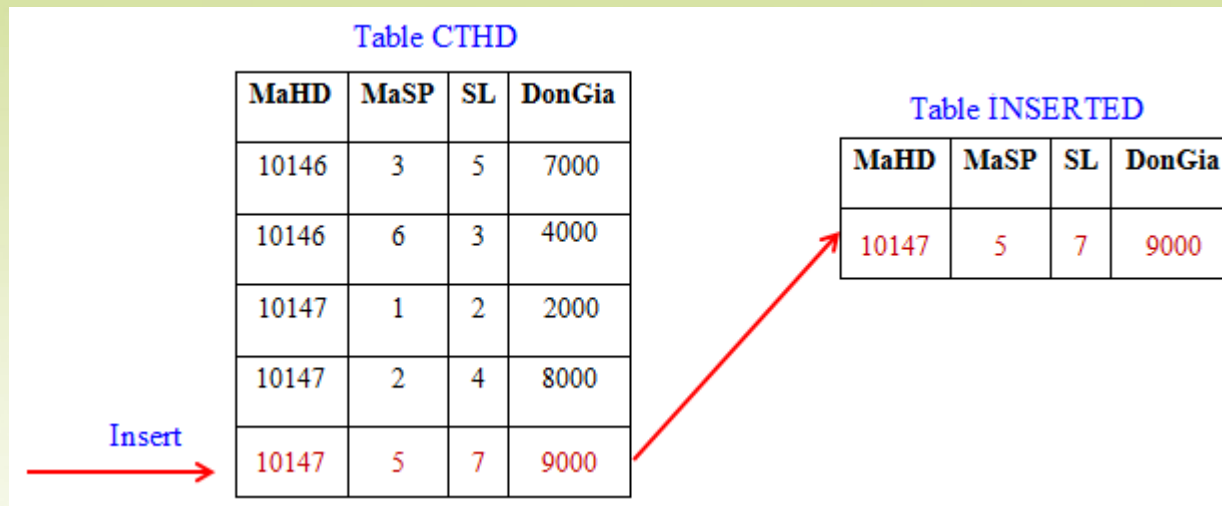
Hành động nào sẽ
kích hoạt trigger

Các kiểu trigger

- ❑ **AFTER trigger:** chạy sau các hành động kiểm tra dữ liệu (constraint, rule).
Thực hiện sau khi hành động **INSERT**, **UPDATE** hay **DELETE** được thực hiện (dữ liệu đã bị tạm thời thay đổi trong bảng). Chỉ định nghĩa được AFTER trigger trên table.
- ❑ **FOR trigger:** tương tự như AFTER trigger.
- ❑ **INSTEAD OF trigger:** Chạy trước các hành động kiểm tra dữ liệu (Constraint, rule). Dữ liệu chưa hề bị thay đổi trong bảng. Có thể thay thế hành động cập nhật dữ liệu bằng các hành động khác. Có thể định nghĩa được cho VIEW (một hoặc nhiều table cơ sở)

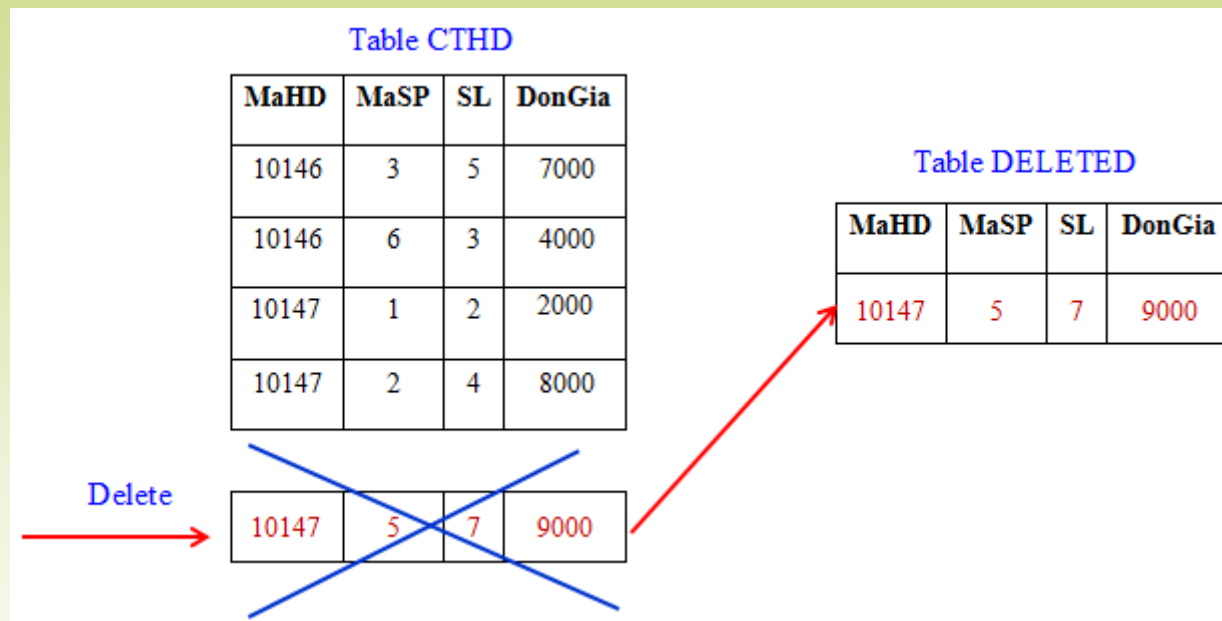
Cơ chế hoạt động của INSERT Trigger

- ❑ Khi thêm mới một dòng vào table, INSERT Trigger sẽ được kích hoạt, dữ liệu của dòng thêm mới cũng được lưu vào Table tạm có tên là INSERTED (lưu trên RAM).
- ❑ Table tạm INSERTED có cấu trúc giống hoàn toàn với Table gắn trigger.



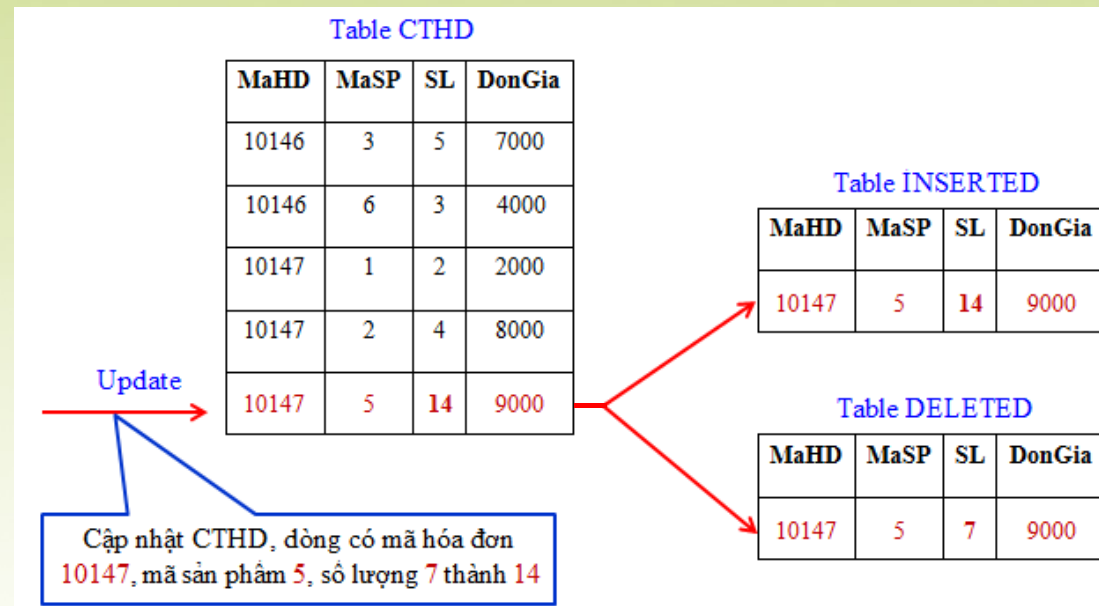
Cơ chế hoạt động của DELETE Trigger

- ❑ Khi DELETE dòng từ table, Delete Trigger sẽ được kích hoạt, dữ liệu của dòng bị xóa sẽ được lưu vào Table tạm có tên là DELETED (lưu trên RAM)
- ❑ Table tạm DELETED có cấu trúc giống hoàn toàn với Table gắn trigger.



Cơ chế hoạt động của UPDATE Trigger

- ❑ Khi UPDATE nội dung dòng từ table, UPDATE Trigger sẽ được kích hoạt:
 - Dữ liệu của dòng trước khi UPDATE sẽ được lưu vào table tạm có tên là DELETED (lưu trên RAM),
 - Dữ liệu của dòng sau khi UPDATE sẽ được lưu vào bảng tạm có tên là INSERTED (lưu trên RAM).
- ❑ Table tạm INSERTED, DELETED có cấu trúc giống hoàn toàn với Table gắn trigger.



Hàm UPDATE()

- ❑ Hàm UPDATE(column_name) trả về giá trị TRUE nếu <column_name> được cập nhật dữ liệu, ngược lại, trả về FALSE

Cú pháp: UPDATE(column_name)

→ Lưu ý: Hàm UPDATE() không dùng trong DELETE trigger

Hàm UPDATE()

- ❑ Cấu trúc hàm UPDATE() thường dùng:

```
CREATE TRIGGER trigger_name
```

```
ON table_name
```

```
{ FOR | AFTER | INSTEAD OF }
```

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

```
AS
```

```
IF UPDATE(column_name) [{AND|OR} UPDATE(column_name)...]
```

```
sql_statements
```

```
GO
```

Tạm dừng Trigger

- ❑ Cú pháp ENABLE | DISABLE trigger:

`ALTER TABLE <table_name> ENABLE | DISABLE TRIGGER ALL | trigger_name[,...n]`

- ✓ Ví dụ: Không thực hiện tất cả Triggers của table CTHD

`ALTER TABLE CTHD DISABLE TRIGGER ALL`

- ✓ Không thực hiện Trigger *tg_SLHang_CTHD* và *tg_TongTien_CTHD* của table CTHD

`ALTER TABLE CTHD DISABLE TRIGGER tg_SLHang_CTHD, tg_TongTien_CTHD`

Chỉnh sửa trigger

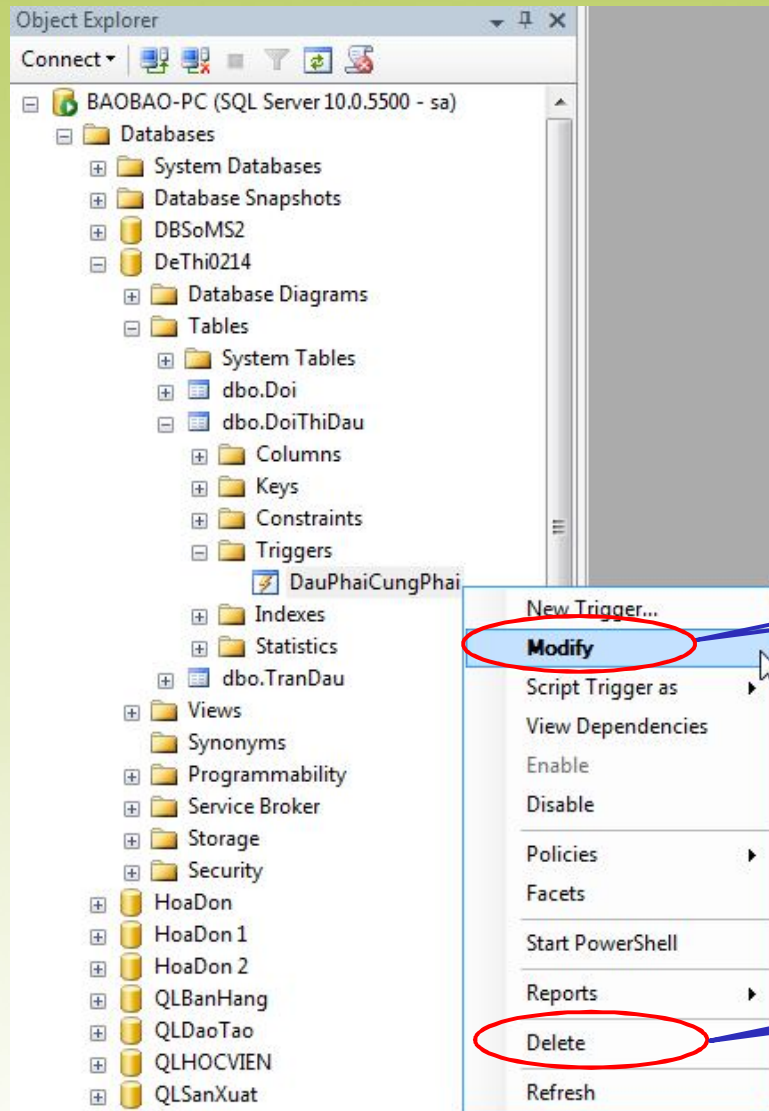
❑ Chỉnh sửa trigger:

```
ALTER TRIGGER trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS
    { sql_statement [ ; ] [ ,...n ] }
GO
```

❑ Xóa trigger:

```
Drop Trigger trigger_name
```

Chỉnh sửa trigger sử dụng giao diện



Mở rộng Table có Trigger cần chỉnh sửa,
Chọn mục Trigger, Right Click chuột lên
Trigger cần thao tác.

Sửa trigger

Xóa trigger

Ví dụ về trigger

- ✓ Ví dụ: Khi thêm mới hóa đơn, mỗi hóa đơn chi tiết chỉ được có tối đa 10 mặt hàng.

Create trigger chk_MatHangHoaDon

on CTHD

after insert

As

If (Select count(*) From CTHD Where MaHD = (Select MaHD From Inserted)) >10

Begin

Raiserror('Một hóa đơn không thể có nhiều hơn 10 mặt hàng',16,1)

Rollback tran

End

Ví dụ về trigger

- ✓ Ví dụ: Nếu có chỉnh sửa bảng CTHD mỗi hóa đơn chi tiết chỉ được có tối đa 10 mặt hàng.

```
Create trigger chk_MatHangHoaDon  
on CTHD  
after update
```

```
As
```

```
    If Update(MaHD)
```

```
    Begin
```

```
        If (Select Count(*) From CTHD, Inserted
```

```
        Where CTHD.MaHD = Inserted.MaHD)>10
```

```
        Begin
```

```
            Raiserror('Một hóa đơn tối đa 10 mặt hàng thôi nhé',16,1)
```

```
            Rollback tran
```

```
        End
```

```
    End
```

Ví dụ về trigger

- ✓ Ví dụ: Không được chỉnh sửa ngày vào làm (NgàyLV) của các nhân viên.

Create trigger chk_SuaNgayVL

on NhanVien

after update

As

If Update(NgayLV)

Begin

Raiserror('Không được sửa ngày vào làm của nhân viên',16,1)

Rollback tran

End