

CÁC HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

TRIGGER

1

Giảng viên: ThS. Nguyễn Thị Uyên Nhi
Email: uyennhisgu@gmail.com

KHÁI NIỆM TRIGGER

- Cấu trúc gần giống như một thủ tục nội tại nhưng
 - Không có tham số đầu vào và đầu ra
 - Phải được liên kết với một bảng/ bảng ảo trong CSDL
- Không thể gọi mà được thực hiện tự động. Sử dụng trong việc:
 - Tính toán, cập nhật giá trị tự động
 - Kiểm tra dữ liệu nhập
- Khai báo sử dụng
 - Kết hợp với các hành động INSERT/ UPDATE/ DELETE trên bảng hay bảng ảo
 - Khi tạo ra, tham gia vào transaction khởi tạo bởi câu lệnh cập nhật dữ liệu tương ứng

SỬ DỤNG TRIGGER

Trigger được sử dụng trong các cách sau:

- Có thể thay đổi đồng loạt các table có liên quan với nhau trong CSDL
- Có thể không cho phép hoặc hủy bỏ những thay đổi vi phạm ràng buộc toàn vẹn tham chiếu và các giao dịch sửa đổi dữ liệu.

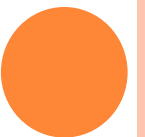
SỬ DỤNG TRIGGER

Trigger được sử dụng trong các cách sau:

- ❑ Có thể thay đổi đồng loạt các table có liên quan với nhau trong CSDL
- ❑ Có thể không cho phép hoặc hủy bỏ những thay đổi vi phạm ràng buộc toàn vẹn tham chiếu và các giao dịch sửa đổi dữ liệu.
- ❑ Có thể áp đặt các giới hạn phức tạp hơn những giới hạn được định nghĩa bằng ràng buộc CHECK và có thể tham chiếu đến các cột trong các bảng khác
- ❑ Có thể tìm sự khác biệt giữa các trạng thái của một table trước và sau khi sửa đổi dữ liệu và lấy ra những tác động dựa trên sự thay đổi đó

CÁC HẠN CHẾ TRÊN TRIGGER

- ❑ Không được tạo và tham chiếu bảng tạm
- ❑ Không tạo hay thay đổi, xoá cấu trúc các đối tượng sẵn có trong CSDL
 - ❑ CREATE/ALTER/DROP
- ❑ Không gán, cấp quyền cho người dùng
 - ❑ GRANT/REVOKE



CƠ CHẾ HOẠT ĐỘNG TRIGGER

- 3 biến cố kích hoạt 1 trigger
 - INSERT
 - UPDATE
 - DELETE
- Trigger lưu trữ dữ liệu của mẫu tin vừa thêm vào một table mới có tên là INSERTED.
- Trigger lưu trữ dữ liệu của mẫu tin vừa xoá vào một table có tên là DELETED.
- Trigger lưu trữ dữ liệu của mẫu tin vừa cập nhật là sự phối hợp của 2 table DELETED và INSERTED

LỆNH CREATE TRIGGER

CREATE TRIGGER <trigger_name> **ON** <table
name> **[WITH ENCRYPTION]**

AFTER | FOR {DELETE, INSERT, UPDATE}

AS <Các phát biểu T-sql>

- Tập con của { **DELETE, INSERT, UPDATE**} dùng chỉ định những phát biểu cập nhật nào trên Table sẽ kích hoạt Trigger.

CÁC LOẠI TRIGGER

- Có hai loại trigger

- Trigger thông thường: AFTER (FOR) Trigger

- Chạy sau các hành động kiểm tra dữ liệu của các Rule, Constraint
 - Dữ liệu đã bị tạm thời thay đổi trong bảng

- INSTEAD OF TRIGGER

- Chạy trước các hành động kiểm tra dữ liệu
 - Dữ liệu chưa hề bị thay đổi
 - Có thể thay thế hành động cập nhật dữ liệu bằng các hành động khác



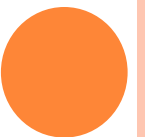
VÍ DỤ

```
CREATE TRIGGER Them_HH  
ON HANG_HOA  
AFTER INSERT  
AS
```

```
    Select * From Inserted
```

□ Thêm dữ liệu

```
INSERT HANG_HOA(MaHH, TenHH)  
VALUES('TV01', 'Tivi Sony')
```



VÍ DỤ

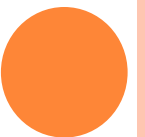
```
CREATE TRIGGER SUA_HH  
ON  HANG_HOA  
AFTER UPDATE  
AS
```

```
Select * From Inserted
```

```
Select * From Deleted
```

❖ Cập nhật dữ liệu

```
UPDATE HANG_HOA  
SET  Ten_HH = 'Man Hinh Sony'  
WHERE MaHH = 'TV01'
```



VÍ DỤ

```
CREATE TRIGGER Xoa_HH  
ON  HANG_HOA  
AFTER DELETE  
AS
```

```
Select * From Inserted
```

```
Select * From Deleted
```

❖ Xóa dữ liệu

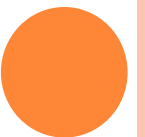
```
DELETE HANG_HOA  
WHERE MaHH = 'TV01'
```



CÁC THAO TÁC TRIGGER PHỔ BIẾN

Kiểm tra ràng buộc toàn vẹn khi:

- ❖ Thêm mới mẫu tin
- ❖ Xóa mẫu tin
- ❖ Sửa mẫu tin



A) TRIGGER – THÊM MẪU TIN MỚI

- Thường dùng để kiểm tra
 - Khóa ngoại,
 - Miền giá trị,
 - Liên bộ trong cùng một bảng
 - Liên thuộc tính trong cùng một bảng
 - 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, n/ếu đã khai báo các ràng buộc này bằng constraint
- Các cấu trúc lệnh thường dùng khi kiểm tra
 - If Else
 - If Exists
 - Raiserror
 - Rollback Tran



A) TRIGGER – THÊM MẪU TIN MỚI

Raiserror : trả thông báo lỗi cho ứng dụng

Raiserror(Tbao_loi, muc_do, trang_thai[, cac_tham_so])

- Tbao_loi: mã thông báo lỗi do người dùng định nghĩa
- Muc_do: 0-25 thể hiện mức độ nghiêm trọng của lỗi
- Trang_thai: 1-127, xác định vị trí lỗi khi sử dụng cùng 1 tbao_loi tại nhiều điểm khác nhau
- cac_tham_so: hỗ trợ các tbao_loi khi cần tham số



A) TRIGGER – THÊM MẪU TIN MỚI

VÍ DỤ:

HOADON_DH(MaHD, NgayDH, MaKH)

PHIEU_XUAT(MaPX, NgayXuat, MaHD)

CHITIET_DH(MAHD, MaHH, SoLuong, DonGia)

Xây dựng trigger trong bảng PHIEU_XUAT để kiểm tra các ràng buộc toàn vẹn dữ liệu khi người dùng thêm mới thông tin của một phiếu xuất hàng cho một bảng hoá đơn đặt hàng trước đó. Các ràng buộc toàn vẹn dữ liệu bao gồm.

- Khoá ngoại: cần kiểm tra số đặt hàng phải tồn tại trong bảng đơn đặt hàng.
- Miền giá trị: cần kiểm tra ngày giao hàng phải ở sau ngày đặt hàng.



A) TRIGGER – THÊM MẪU TIN MỚI

```
CREATE TRIGGER    tg_PhieuXuat_Insert
ON    PHIEU_XUAT
FOR  INSERT AS
DECLARE @NgayHD datetime, @ErrMsg varchar(200)
-- Kiểm tra số hoá đơn đã có trong bảng DONDH không?
IF NOT EXISTS(Select *
                From Inserted I, HOADON_DH D
                Where I.MaHD= D.MaHD)
Begin
    Rollback Tran
    Raiserror('Số đơn đặt hàng không tồn tại', 16,1)
    Return
End
```



A) TRIGGER – THÊM MẪU TIN MỚI

--Tính ra ngày đặt hàng

Select @NgàyDH=NgàyDH

From HoaDon_DH D, Inserted I

Where D.MaHD = I.MaHD

-- Kiểm tra ngày giao hàng phải sau ngày đặt hàng

IF @NgàyDH < (Select ngayxuat From Inserted)

Begin

Set @ErrMsg = 'ngày giao hàng phải ở sau ngày:'

+ Convert(char(10), ngayDH, 103)

Raerror(@ErrMsg,16,1)

Rollback tran

End



B) TRIGGER – XÓA MÀU TIN MỚI

Tương tự, kiểm tra các ràng buộc như trigger INSERT, đặc biệt kiểm tra ràng buộc khóa ngoại

Ví dụ: khi xoá một số hoá đơn đặt hàng trong bảng HOADON_DH cần phải kiểm tra các RBTV dữ liệu sau:

- Kiểm tra xem đơn đặt hàng bị xoá đã được xuất hàng chưa? Nếu đã được xuất rồi thì thông báo không thể xoá đơn đặt hàng được.
- Ngược lại thì xoá dữ liệu liên quan bên bảng chi tiết đơn đặt hàng (CHITIET_HD)



B) TRIGGER – XÓA MÀU TIN MỚI

```
CREATE TRIGGER tg_HOADON_Delete
ON HOADON_DH
FOR DELETE
AS
DECLARE @SoPX char(5), @ErrMsg char(200), @Delete_Err int
-- Kiểm tra xem đơn hàng đã được xuất chưa
IF EXISTS(Select MaPX From PHIEU_XUAT
Where MaHD IN(Select MaHD From Deleted))
Begin
    Select @MaPX = MaPX From PHIEU_XUAT
    Where MaHD In(Select MaHD From Deleted)
    Set @ErrMsg = 'Đơn đặt hàng đã được nhập theo ' + 'số xuất
hàng ' + @SoPX + char(13) + '.Không thể huỷ được'
    RaiseError(@ErrMsg,16,1)
    Rollback tran
End
```



B) TRIGGER – XÓA MẪU TIN MỚI

Else

Begin

-- Xóa tự động chi tiết các đơn đặt hàng liên quan

Delete FROM CHITIET_DH

Where MaHD In(Select MaHD From DELETED)

Set @Delete_Err = @@ERROR

IF @Delete_Err <> 0

Begin

Set @ErrMsg = 'Lỗi vi phạm xóa trên bảng chi tiết đặt hàng'

RaisError(@ErrMsg, 16, 1)

Rollback Tran

End

End



c) TRIGGER – SỬA ĐỔI MẪU TIN

- Tương tự, kiểm tra các ràng buộc như trigger INSERT, ràng buộc khoá ngoại có thể sử dụng UPDATE để thực hiện tự động.
- Hàm Update: kiểm tra dữ liệu của cột bên trong bảng có bị thay đổi trong các trigger sửa đổi dữ liệu
- Cú pháp : UPDATE (tên_cột) (biểu thức luận lý)
 - Tên_cột: tên cột mà chúng ta muốn kiểm tra xem dữ liệu tại đó có bị sửa đổi trong trigger không.
 - Biểu thức luận lý: trả về True khi giá trị dữ liệu của cột đã bị sửa đổi, ngược lại trả về False khi giá trị dữ liệu của cột không bị sửa đổi



c) TRIGGER – SỬA ĐỔI MẪU TIN

Sửa đổi thông tin của một số đặt hàng bên trong bảng HOADON_DH cần phải kiểm tra các ràng buộc toàn vẹn dữ liệu sau:

- ❑ Không cho phép sửa đổi dữ liệu tại cột MaDH hoặc MaKH vì khi đó dữ liệu sẽ ảnh hưởng đến nhiều bảng.
- ❑ Sửa đổi giá trị cột ngày đặt hàng thì phải đảm bảo luôn luôn trước ngày giao hàng đầu tiên của số đặt hàng đó (nếu đơn đặt hàng đã có giao hàng).



c) TRIGGER – SỬA ĐỔI MẪU TIN

```
CREATE TRIGGER tg_HOADON_DH_Update  
ON HOADON_DH  
FOR UPDATE
```

```
AS Declare @MinNgayXH date, @ErrMsg varchar(200)
```

```
-- Khi sửa đổi các cột MaDH hoặc MaKH
```

```
IF Update(MaDH) OR Update(MaKH)
```

```
Begin
```

```
    Rollback Tran
```

```
    Set @ErrMsg = 'Không thể thay đổi số đặt hàng hoặc  
    mã khách hàng'
```

```
    RaisError(@ErrMsg, 16, 1)
```

```
    Return
```

```
End
```



c) TRIGGER – SỬA ĐỔI MẪU TIN

-- Khi sửa đổi ngày đặt hàng

IF Update(NgayDH)

Begin

-- Kiểm tra đơn đặt hàng đã được xuất chưa

IF EXISTS (Select MaPX

From PHIEU_XUAT PX, Deleted d

where px.madh=d.madh)

Begin

-- Tính ra ngày nhập hàng đầu tiên

Select @MinNgayXH = Min(NgayXuat)

From PHIEU_XUAT PX, DELETED D

Where PX.MaDH = D.MaDH



c) TRIGGER – SỬA ĐỔI MẪU TIN

--kiểm tra giá trị ngày đặt hàng sau khi sửa đổi phải luôn trước ngày giao hàng đầu tiên

```
IF @MinNgayXH < (Select NgayDH From Inserted)
```

```
Begin
```

```
Rollback tran
```

```
Set @ErrMsg = 'Ngày đặt hàng phải ở trước ngày:'
```

```
+ Convert(char(10), @MinNgayXH, 103)
```

```
RaisError(@ErrMsg, 16, 1)
```

```
End
```

```
End
```

```
End
```



CHỈ ĐỊNH THỨ TỰ THỰC HIỆN CÁC TRIGGER

- Nếu cùng một thao tác trên một Table mà có nhiều Trigger (khác tên nhau). Khi đó thứ tự thực hiện các trigger được xác định dựa trên thứ tự tạo ra chúng.
- Để thay đổi thứ tự thực hiện mặc định này, sử dụng thủ tục:
`sp_settriggerorder[@triggername =] 'Tên trigger'`
`, [@order =] 'First | Last | None'`
`, [@stmttype =] 'Insert | Update | Delete'`

Ví dụ: `sp_SetTriggerOrder` itrg_SoCTHD, 'First', 'Insert'

`sp_SetTriggerOrder` itrg_GiamTon, 'Last', 'Insert'

- SP này chỉ có thể chỉ định trigger nào được thực hiện đầu tiên và Trigger nào được thực hiện cuối cùng. Các Trigger còn lại sẽ thực hiện theo thứ tự tạo ra chúng.

THỰC HIỆN HAY KHÔNG THỰC HIỆN TRIGGER

ALTER TABLE <tên table>

ENABLE | DISABLE TRIGGER ALL | <tên trigger>[,...*n*]

Ví dụ: Không thực hiện tất cả Triggers của table

HOADON_DH

ALTER TABLE HOADON_DH DISABLE TRIGGER

ALL

Ví dụ: Không thực hiện tg_HOADON_Delete và
tg_HOADON_DH_Update của table HOADON_DH

ALTER TABLE HOADON_DH

DISABLE TRIGGER

tg_HOADON_DH_Update

tg_HOADON_Delete,

SỬA, XÓA TRIGGER

1. Sửa Trigger:

ALTER TRIGGER <trigger_name> ON <table name>

[WITH ENCRYPTION]

AFTER | FOR {DELETE, INSERT, UPDATE}

AS<Các phát biểu T-sql>

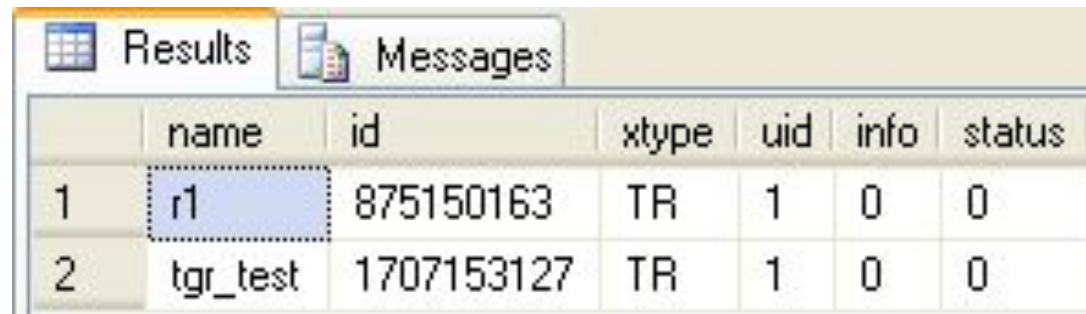
2. Xóa Trigger:

DROP TRIGGER <tên_trigger> [...n]

HIỂN THỊ THÔNG TIN VỀ CÁC TRIGGER

Tất cả các đối tượng trong CSDL được liệt kê trong bảng hệ thống sysobjects. Cột type trong sysobjects xác định các trigger với chữ viết tắt là TR.

```
SELECT *  
FROM sysobjects  
WHERE type='TR'
```



The screenshot shows a SQL Server Enterprise Manager window with the 'Results' tab selected. It displays a table with 7 columns: name, id, xtype, uid, info, and status. There are two rows of data, both representing triggers (xtype = TR). The first row has name 'r1' and id '875150163'. The second row has name 'tgr_test' and id '1707153127'. Both have uid '1', info '0', and status '0'.

	name	id	xtype	uid	info	status
1	r1	875150163	TR	1	0	0
2	tgr_test	1707153127	TR	1	0	0

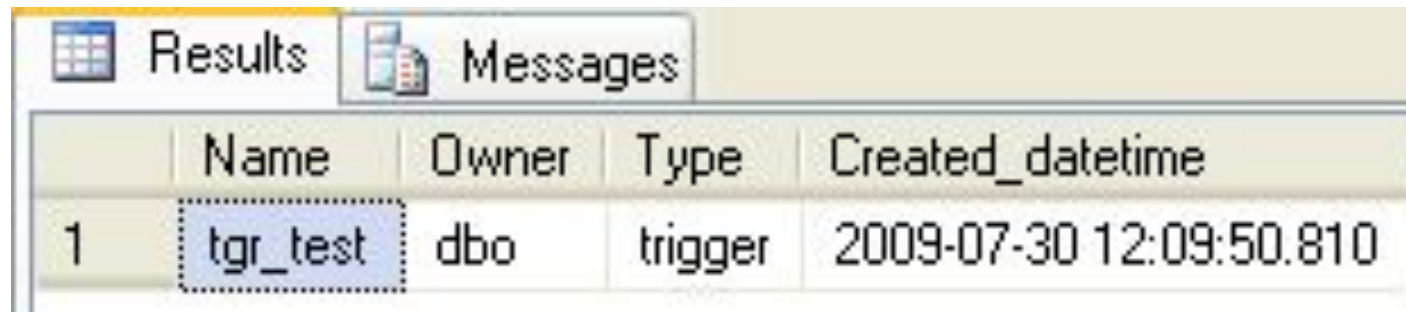
HIỂN THỊ THÔNG TIN VỀ CÁC TRIGGER

- Cú pháp hiển thị thông tin về trigger:

sp_help *tên_trigger*

- Hiển thị thông tin trigger tgr_test:

Sp_help tgr_test



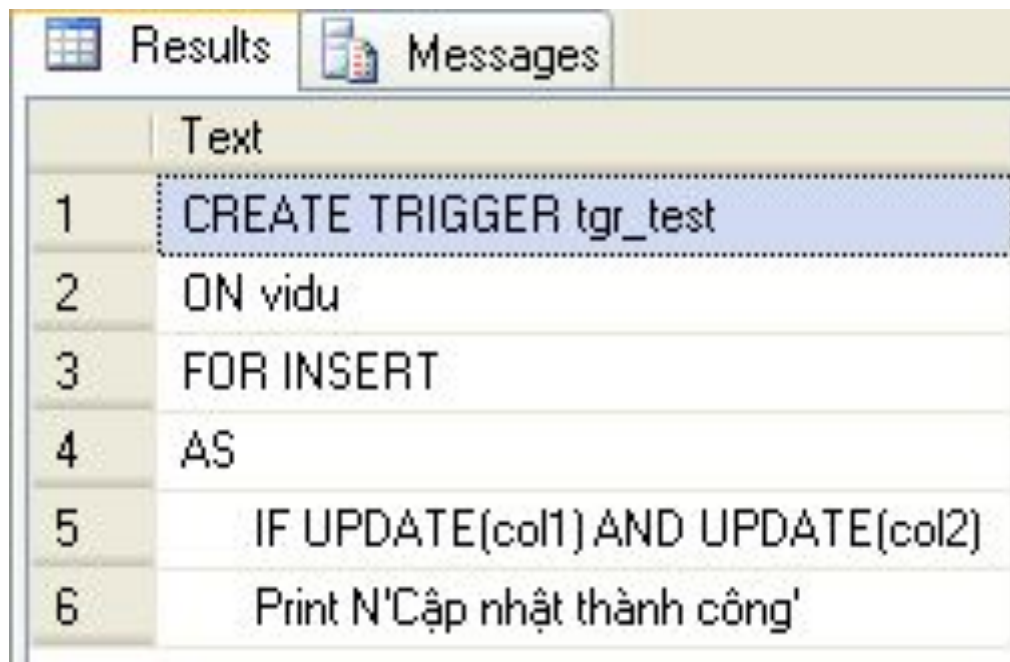
Results		Messages		
	Name	Owner	Type	Created_datetime
1	tgr_test	dbo	trigger	2009-07-30 12:09:50.810

HIỂN THỊ THÔNG TIN VỀ TRIGGER

Câu lệnh Create trigger của mỗi trigger được lưu trữ trong bảng hệ thống syscomments. Người dùng có thể hiển thị nội dung câu lệnh trigger bằng cách sử dụng thủ tục **sp_helptext**

Hiển thị nội dung trigger tgr_test:

Sp_helptext tgr_test



	Text
1	CREATE TRIGGER tgr_test
2	ON vidu
3	FOR INSERT
4	AS
5	IF UPDATE(col1) AND UPDATE(col2)
6	Print N'Cập nhật thành công'