



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA HỆ THỐNG THÔNG TIN

*MÔN CƠ SỞ DỮ LIỆU - IT004*

# CHƯƠNG 5 (tt): TRIGGER

ThS. TẠ VIỆT PHƯƠNG  
phuongtv@uit.edu.vn

# Stored procedure

Stored Procedure là đoạn chương trình kịch bản (programming scripts) với các câu lệnh SQL nhúng (embedded SQL) được lưu dưới dạng đã được biên dịch và thi hành trực tiếp bởi MS SQL server.

Một stored procedure là một mã SQL đã được chuẩn bị sẵn mà có thể lưu, vì thế mã này có thể được sử dụng lại nhiều lần.

# Stored procedure

Cú pháp:

```
CREATE PROCEDURE tên procedure
```

```
AS
```

```
Câu lệnh SQL
```

```
GO;
```

Câu lệnh thực thi một Stored Procedure:

```
EXEC tên procedure;
```

# Stored procedure

Ví dụ:

```
CREATE PROCEDURE DanhSachNV  
AS  
BEGIN  
    SELECT HoTen, NTNS  
    FROM NhanVien  
    ORDER BY HoTen;  
END;
```

# Stored procedure

Ví dụ:

```
CREATE PROCEDURE DanhSachNV1 @max_luong INT
AS
BEGIN
    SELECT HoTen, NTNS, Luong
    FROM NhanVien WHERE Luong< @max_luong
    ORDER BY HoTen;
END;
```

## Review Chương 4, phần 2.2

- ❖ **Ràng buộc toàn vẹn trong SQL được chia làm 2 loại chính:**
  - **Loại đơn giản:** Sử dụng **Constraint** để mô tả
  - **Loại phức tạp:** Sử dụng **Trigger** để thực hiện
- ❖ Có thể khai báo RBTV ở mức cột hoặc mức bảng

# Trigger

Trigger là một loại stored procedure đặc biệt được thực thi (execute) một cách tự động khi có một sự kiện thay đổi dữ liệu (data modification) xảy ra như Update, Insert hoặc Delete.

Trigger được dùng để đảm bảo tính toàn vẹn dữ liệu (Data Integrity) hoặc thực hiện các quy tắc nghiệp vụ (business rules) nào đó.

# Trigger

Khi trigger được thực thi, SQL tự động tạo ra 2 bảng tạm với cùng cấu trúc với bảng mà trigger được định nghĩa trên đó.

Bảng INSERTED chứa dữ liệu mới khi thực thi câu lệnh Insert hoặc câu lệnh Update.

Bảng DELETED chứa dữ liệu bị xóa khi thực thi câu lệnh Delete hoặc chứa dữ liệu cũ (old) khi thực thi câu lệnh Update.

Hai bảng này chỉ tồn tại trong thời gian trigger xử lý và cục bộ cho mỗi trigger.

Hoạt động	Bảng INSERTED	Bảng DELETED
INSERT	dữ liệu mới được insert	không có dữ liệu
DELETE	không có dữ liệu	chứa dữ liệu bị xóa
UPDATE	chứa dữ liệu sau khi được cập nhật	chứa dữ liệu trước khi cập nhật



# Trigger

Đối với thao tác insert:

Insert into HOADON values (1004, '01/09/2006', 'KH02', 180000)

HOADON			
SOHD	NGHD	MAKH	TRIGIA
1001	23/07/2006	KH01	320,000
1002	12/08/2006	KH01	840,000
1003	23/08/2006	KH02	100,000
1004	01/09/2006	KH02	180,000

INSERTED			
SOHD	NGHD	MAKH	TRIGIA
1004	01/09/2006	KH02	180,000

DELETED			
SOHD	NGHD	MAKH	TRIGIA

1004	01/09/2006	KH02	180,000
------	------------	------	---------

# Trigger

Đối với thao tác delete

Delete from HOADON where soh=1004

HOADON			
SOHD	NGHD	MAKH	TRIGIA
1001	23/07/2006	KH01	320,000
1002	12/08/2006	KH01	840,000
1003	23/08/2006	KH02	100,000
1004	01/09/2006	KH02	180,000

1004	01/09/2006	KH02	180,000
------	------------	------	---------

INSERTED			
SOHD	NGHD	MAKH	TRIGIA

DELETED			
SOHD	NGHD	MAKH	TRIGIA
1004	01/09/2006	KH02	180,000

# Trigger

Đối với thao tác update:

Update HOADON set makh='kh07', trigia=300000 Where sohd=1004

HOADON			
SOHD	NGHD	MAKH	TRIGIA
1001	23/07/2006	KH01	320,000
1002	12/08/2006	KH01	840,000
1003	23/08/2006	KH02	100,000
1004	01/09/2006	KH07	300,000

INSERTED			
SOHD	NGHD	MAKH	TRIGIA
1004	01/09/2006	KH07	300,000
DELETED			
SOHD	NGHD	MAKH	TRIGIA
1004	01/09/2006	KH02	180,000

# Trigger

Tạo trigger: cú pháp

```
CREATE [OR ALTER] TRIGGER Tên_Trigger  
ON Tên_Table  
AFTER (FOR) | INSTEAD OF INSERT, DELETE, UPDATE  
AS  
    Các _lệnh_của_Trigger
```

Xóa trigger: cú pháp

```
DROP TRIGGER Tên_Trigger
```

# Trigger

Có 2 loại triggers: INSTEAD OF và AFTER (FOR).

## INSTEAD OF:

- Trigger được gọi thực hiện thay cho thao tác delete/insert/update tương ứng.
- Trigger instead of thường được dùng để xử lý cập nhật trên view.

## AFTER (FOR):

- Trigger được gọi thực hiện sau khi thao tác delete/ insert/ update tương ứng đã được thực hiện thành công.
- Có thể quay lui thao tác đã thực hiện bằng lệnh rollback transaction.

# Trigger

**KHACHHANG** (MAKH, HOTEN, NGSINH, NGDK)

**HOADON** (SOHD, NGHD, MAKH, TRIGIA)

Ngày mua hàng (NGHD) của một khách hàng thành viên sẽ lớn hơn hoặc bằng ngày khách hàng đó đăng ký thành viên (NGDK).

	THÊM	XÓA	SỬA
KHACHHANG	-	-	+ (NGDK)
HOADON	+	-	+ (NGHD, MAKH)

# Trigger

Sinh viên tìm hiểu các cấu trúc sau:

- + Khai báo biến: DECLARE
- + Xuất thông tin: PRINT, RAISERROR
- + Cấu trúc điều kiện: IF
- + Cấu trúc lặp: FOR
- + Con trỏ: CURSOR
- + Hủy cập nhật dữ liệu vào hệ thống bộ nhớ: ROLLBACK  
TRAN

# Trigger

```
CREATE TRIGGER nghd_hoadon_insert
ON hoadon
AFTER INSERT
AS
    DECLARE @ng_muahang smalldatetime
    DECLARE @ng_dangky smalldatetime
    SELECT @ng_muahang=nghd, @ng_dangky=ngdk
    FROM khachhang, inserted
    WHERE khachhang.makh=inserted.makh
    IF @ng_muahang< @ng_dangky
    BEGIN
        rollback transaction
        print 'ngay mua hang phai lon hon ngay dang ky'
    END
```



# Trigger

HOẶC:

```
CREATE TRIGGER nghd_hoadon_insert
ON hoadon
AFTER INSERT
AS
    IF (EXISTS (SELECT *
                FROM inserted i JOIN khachhang kh ON i.makh=kh.makh
                WHERE i.nghd<kh.ngdk))
BEGIN
    rollback transaction
    print 'ngày mua hàng phải lớn hơn ngày đăng ký'
END
```

# Trigger

```
CREATE TRIGGER nghd_hoadon_update
ON hoadon
AFTER UPDATE
AS
IF (UPDATE (makh) OR UPDATE (nghd))
BEGIN
    DECLARE @ng_muahang smalldatetime
    DECLARE @ng_dangky smalldatetime
    SELECT @ng_muahang=nghd, @ng_dangky=ngdk
    FROM khachhang, inserted
    WHERE khachhang.makh=inserted.makh
    IF @ng_muahang< @ng_dangky
    BEGIN
        rollback transaction
        print 'ngay mua hang phai lon hon ngay dang ky'
    END
END
END
```

# Trigger

HOẶC:

```
CREATE TRIGGER nghd_hoadon_update
ON hoadon
AFTER UPDATE
AS
IF (UPDATE (makh) OR UPDATE (nghd)
BEGIN
    IF (EXISTS (SELECT *
                FROM inserted i JOIN khachhang kh ON i.makh=kh.makh
                WHERE i.nghd<kh.ngdk))
    BEGIN
        rollback transaction
        print 'ngay mua hang phai lon hon ngay dang ky'
    END
END
END
```

# Trigger

```
CREATE TRIGGER nghd_khachhang_update  
ON KHACHHANG  
AFTER UPDATE  
AS
```

```
    DECLARE @ng_dangky smalldatetime, @makhang char(4)  
    SELECT    @ng_dangky=ngdk, @makhang=makh  
    FROM inserted  
    IF (UPDATE (ngdk))  
    BEGIN  
    IF (EXISTS (SELECT *  
                FROM hoadon  
                WHERE    makh=@makhang AND nghd<@ng_dangky))  
    BEGIN  
        rollback transaction  
    END  
    END
```

# Trigger

HOẶC:

```
CREATE TRIGGER nghd_khachhang_update
ON KHACHHANG
AFTER UPDATE
AS
    IF (UPDATE (ngdk))
    BEGIN
        IF (EXISTS (SELECT *
                    FROM hoadon hd JOIN inserted i ON hd.makh=i.makh
                    WHERE hd.nghd<kh.ngdk))
        BEGIN
            rollback transaction
        END
    END;
END;
```

# Trigger

Trường hợp thì chỉ muốn ngưng kiểm tra trigger thì ta sử dụng lệnh sau:

```
ALTER TABLE Tên_bảng  
DISABLE/ENABLE TRIGGER Tên_trigger (/ALL)
```

# Lược đồ CSDL quản lý giáo vụ

**HOCVIEN** (MAHV, HO, TEN, NGSINH, GIOITINH, NOISINH, CMND, MALOP)

**LOP** (MALOP, TENLOP, TRGLOP, SISO, MAGVCN)

**KHOA** (MAKHOA, TENKHOA, NGTLAP, TRGKHOA)

**MONHOC** (MAMH, TENMH, TCLT, TCTH, MAKHOA)

**DIEUKIEN** (MAMH, MAMH\_TRUOC)

**GIAOVIEN** (MAGV, HOTEN, HOCVI, HOCHAM, GIOITINH, NGSINH, NGVL, HESO, MUCLUONG, MAKHOA)


**GIANGDAY** (MALOP, MAMH, MAGV, HOCKY, NAM, TUNGAY, DENNGAY)

**KETQUATHI** (MAHV, MAMH, LANTHI, NGTHI, DIEM, KQUA)

# Bài tập

1. Trưởng lớp của một lớp phải là học viên của lớp đó.
2. Học viên chỉ được thi lại một môn (lần thi  $>1$ ) khi điểm của lần thi trước đó dưới 5.
3. Ngày thi của lần thi sau phải lớn hơn ngày thi của lần thi trước (cùng học viên, cùng môn học).
4. Mỗi năm học chỉ được giảng dạy tối đa 10 lớp
5. Mỗi học kỳ của một năm học, một lớp chỉ được dạy tối đa 3 môn.





# THANK YOU!

## Q & A

**ThS. TẠ VIỆT PHƯƠNG**  
**[phuongtv@uit.edu.vn](mailto:phuongtv@uit.edu.vn)**