

# STORED PROCEDURE, FUNCTION and TRIGGER

Phan Anh Phong – Vinh University

1

## Nội dung

- Stored Procedure
- Function
- Trigger

Phan Anh Phong – Vinh University

2

## Stored procedure

- Stored procedure (SP) là một chương trình con - nhóm các câu lệnh SQL để có thể gọi lại. Các DBMS đều hỗ trợ kỹ thuật này.
- Ý nghĩa của SP:
  - Chia để trị
  - Giảm lưu lượng trao đổi dữ liệu giữa client và DBMS
  - Tăng tốc độ thực thi vì DBMS không cần phân tích/biên dịch lại câu lệnh ở mỗi lần thực thi SP
  - Tăng an toàn dữ liệu, có thể gán quyền trên các SP thay vì gán trên các bảng

## Stored procedure

- Phân loại:
  - System SP: các chương trình con được hệ thống xây dựng sẵn. Ví dụ trong SQL Server: sp\_addlogin, sp\_adduser, sp\_databases,, ...
  - SP do người người tự viết
- Một cách phân loại khác:
  - SP không có tham số
  - SP có tham số

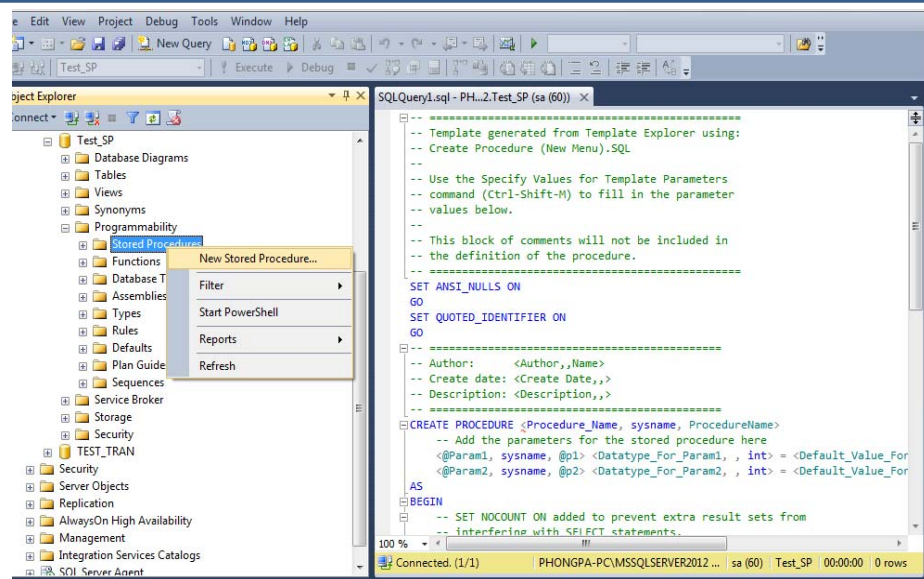
## Tạo SP trong SQL Server

- Cách 1. Sử dụng SQL Server Management Studio (SSMS)
- Cách 2: Tạo SP bằng script

Phan Anh Phong – Vinh University

5

## Tạo SP trong SQL Server bằng SSMS



Phan Anh Phong – Vinh University

6

## Tạo SP trong SQL Server bằng script

- Tạo thủ tục: CREATE PROCEDURE tên\_thủ\_tục  
    [@tham\_số\_1 Kiểu\_dữ\_liệu [out],  
    @tham\_số\_2 Kiểu\_dữ\_liệu [out], ...]  
AS  
BEGIN  
    [khai báo các biến sử dụng lệnh DECLARE @tên\_biến  
    Kiểu\_dữ\_liệu]  
    {các câu lệnh}  
END;  
  - Phần [out] có thể không có. Nếu không có out thì hiểu đó là tham số để truyền giá trị vào cho SP. Nếu có out được hiểu là truyền giá trị ra cho SP

Phan Anh Phong – Vinh University

7

## SP trong SQL Server

- Xóa thủ tục:  
DROP PROC tên\_thủ\_tục;
- Lời gọi một thủ tục  
EXEC tên\_thủ\_tục [danh\_sách\_tham\_số\_thực\_sự,]
- Chỉnh sửa một thủ tục đã có  
ALTER PROC tên\_thủ\_tục  
    [danh sách tham số]  
AS  
BEGIN  
    {các lệnh sql}  
END;

Phan Anh Phong – Vinh University

8

## Ví dụ 1 – Một ví dụ về SP đơn giản

--Tạo thủ tục có tên usp\_XinChao, với 1 tham số vào kiểu là nvarchar,

```
CREATE PROCEDURE usp_XinChao
@ten nvarchar(50) --sử dụng nvarchar/nchar hỗ trợ UNICODE
AS
BEGIN
    PRINT N'Xin chào ' + @ten --N' hỗ trợ UNICODE (N in hoa)
END

EXEC usp_XinChao N'Các bạn 55K'
```

## Một số lệnh cơ bản trong T-SQL

- Biến:
  - Biến hệ thống bắt đầu @@
  - Biến do người dùng định nghĩa -> khai báoKhai báo biến  
DECLARE @<Tên\_biến> <Kiểu\_dữ\_liệu>  
[=<giá\_trị\_khởi\_tạo>];
- Phép gán giá trị cho 1 biến:  
SET @<tên\_biến> = <biểu\_thức>;  
hoặc  
SELECT @<tên\_biến> = <biểu\_thức>;

## Ví dụ 2 về SP không có tham số

Giả sử có bảng:

```
CREATE TABLE canbo
```

```
(macb int primary key, ten varchar(15), namsinh int );
```

Với nội dung:

```
INSERT INTO canbo VALUES
```

```
(1,'Khoa', 1990),
```

```
(2,'Hoc', 1991),
```

```
(3,'Chan', 1992),
```

```
(4,'Chinh', Null);
```

Yêu cầu: Viết một SP không có tham số để đưa thông tin của tất cả các cán bộ

## Ví dụ 2 về SP không có tham số (tiếp)

```
CREATE PROC sp_xem_can_bo
```

```
AS
```

```
BEGIN
```

```
    SELECT * FROM canbo
```

```
END;
```

Sử dụng SP đã viết: EXEC sp\_xem\_can\_bo;

## Tham số trong Stored procedure

- Trong stored procedure có 2 tham số chính:
  - Tham số đầu vào
  - Tham số đầu ra
- Tham số đầu vào là loại tham số ngầm định, để truyền các giá trị vào trong stored procedure
- Tham số đầu ra dùng để nhận kết quả trả về từ stored procedure. Sử dụng từ khóa OUTPUT (hoặc viết tắt là OUT) để xác định tham số loại này

Phan Anh Phong – Vinh University

13

## Ví dụ 3 về SP có tham số vào

Cho quan hệ canbo như sau:

```
CREATE TABLE canbo
```

```
(macb int primary key, ten varchar(15), namsinh int );
```

### **Yêu cầu:**

Viết một SP để đưa ra thông tin của cán bộ tương ứng với mã cán bộ được nhập từ ngoài (giả sử từ bàn phím)

### **Phân tích:**

-- 1 biến để lưu mã cán bộ: @cb\_id int;

--Tham số truyền vào: @cb\_id

-- câu lệnh SQL: SELECT \* FROM canbo WHERE  
macb=@cb\_id

Phan Anh Phong – Vinh University

14

### Ví dụ 3 về SP có tham số vào (tiếp)

```
CREATE PROC sp_hien_thi_cb
    @cb_id int
AS
BEGIN
    SELECT * FROM canbo
    WHERE macb=@cb_id
END;

--Gọi trực tiếp thủ tục Hien_thi_nv
EXEC usp_hien_thi_cb 1;
--Gọi thủ tục Hien_thi_cb qua biến
DECLARE @maso_cb INT;
SELECT @maso_cb=1;
EXEC usp_hien_thi_cb @maso_cb;
```

Phan Anh Phong – Vinh University

15

### Ví dụ 4 về SP có tham số vào, ra

Viết SP đếm số lượng cán bộ trong bảng canbo theo năm sinh được nhập từ ngoài

Tổ chức thủ tục:

Tham số vào có không? Cái gì? năm sinh,

Tham số ra có không, cái gì? số lượng cán bộ tương ứng với năm sinh

Tổ chức 1 tham số vào là ns để lưu giá trị năm sinh truyền vào thủ tục và 1 tham số ra so\_luong để lấy giá trị ra khi thủ tục thực hiện xong (sử dụng từ khóa OUT)

Phan Anh Phong – Vinh University

16



## Ví dụ 4 về SP có tham số vào, ra (tiếp)

```
create procedure so_can_bo_theo_nam_sinh
    @ns int,
    @so_luong int out
as
begin
    select @so_luong = count(*) from canbo
    where namsinh = @ns;
end;

--Gọi lại thủ tục:
declare @so_cb int;
exec so_can_bo_theo_nam_sinh 1990, @so_cb out;
select @so_cb;
```

Để lấy kq ra

Phan Anh Phong – Vinh University

17

## SQL Server

### Một số chú ý:

- Các tham số phải có ký hiệu @ ở trước (tham biến)
- Từ khoá “out” khai báo ở sau tên và kiểu của các tham số
- Khi gọi lại SP cần có từ khóa out ở sau tên biến để lấy kết quả ra
- Không thể lấy trực tiếp từ tên biến của thủ tục, vì nó là biến địa phương

Phan Anh Phong – Vinh University

18

## Trả về giá trị trong Stored Procedure

- Ngoài cách sử dụng tham số đầu ra để trả về giá trị. Lệnh RETURN Có thể sử dụng để trả về giá trị từ stored procedure hoặc các câu lệnh SELECT khi truy vấn dữ liệu.

## Trả về giá trị trong Stored Procedure

- Trả giá trị về bằng lệnh RETURN:
  - Lệnh RETURN được sử dụng để trả về giá trị từ stored procedure mà không cần sử dụng tham số đầu ra. Giá trị trả về này có một số đặc điểm:
    - Giá trị trả về chỉ có thể là số nguyên. Nếu trả về các loại giá trị khác thì lúc thực thi stored procedure sẽ báo lỗi (ngoại trừ 1 số kiểu dữ liệu được tự động chuyển đổi sang kiểu số nguyên như: float, double,...).
    - Giá trị trả về ngầm định là 0.
    - Có thể nhận giá trị trả về này bằng 1 biến.
    - Sau khi gọi RETURN, stored procedure sẽ trả về giá trị và kết thúc thực hiện stored procedure luôn

## Ví dụ 5 – Sử dụng Return trong SP

```
--Chuẩn bị dữ liệu
create table donvi
(madv char(10) primary key,tendv nvarchar(30),dienthoai char(15))

create table nhanvien
(manv int primary key,tennv nvarchar(30),namsinh int, madv char(10),
foreign key (madv) references donvi)
```

Phan Anh Phong – Vinh University

21

```
create table donvi
(madv char(10) primary key,tendv nvarchar(30),dienthoai char(15))
CREATE PROC ThemDonVi
@mdv char (10),
@tdv nvarchar(30),
@dt char(15)
AS
BEGIN
IF( @mdv in(SELECT madv FROM donvi))
BEGIN
PRINT N'Mã số đơn vị ' + @mdv + N' đã có'
RETURN 1
END
INSERT INTO DonVi VALUES(@mdv, @tdv, @dt)
RETURN 0
END
GO
```

22

## Ví dụ 5 – Sử dụng Return trong SP

### ▪ Ví dụ về RETURN

- Viết 1 SP để thêm một đơn vị mới vào CSDL, nếu trùng mã thì thông báo về sự tồn tại của đơn vị và thoát, ngược lại thì thực hiện bổ sung các thông tin của đơn vị đó vào CSDL
- Phân tích:
  - Tên SP?
  - Số tham số?
  - Sử dụng cấu trúc IF kết hợp RETURN để trả kết quả về ứng với từng trường hợp
  - Kiểm tra trùng mã có thể dùng lệnh EXISTS

## Ví dụ 5 – Sử dụng Return trong SP

```
CREATE PROC ThemDonVi
@mdv char (10),
@tdv nvarchar(30),
@dt char(15)
AS
BEGIN
    IF(EXISTS(SELECT * FROM donvi d WHERE d.madv = @mdv))
    BEGIN
        PRINT N'Mã số đơn vị ' + @mdv + N' đã có'
        RETURN 1
    END
    INSERT INTO DonVi VALUES(@mdv, @tdv, @dt)
    RETURN 0 /* procedure tự trả về 0 nếu không RETURN */
END
GO
```

## Ví dụ 5 – Sử dụng Return trong SP

--Chuẩn bị dữ liệu

create table donvi

(madv char(10) primary key, tendv nvarchar(30), dienthoai char(15))

create table nhanvien

(manv char(10) primary key, tennv nvarchar(30), namsinh int, madv char(10),

foreign key (madv) references donvi)

--Yêu cầu:

Viết một SP để thêm một nhân viên mới với yêu cầu:

- Nếu trùng mã nhân viên thì trả về giá trị 1 và thông báo lỗi
- Nếu mã đơn vị của nhân viên mới chưa có trong CSDL thì trả về giá trị 2 và thông báo lỗi
- Nếu không vi phạm các lỗi trên thì trả về giá trị 0 và thực hiện bổ sung nhân viên mới vào CSDL

Phan Anh Phong – Vinh University

25

## Ví dụ 6 – Sử dụng Return trong SP

▪ Viết một SP để thêm một nhân viên mới với yêu cầu:

- Nếu trùng mã nhân viên thì trả về giá trị 1 và thông báo lỗi
- Nếu mã đơn vị của nhân viên mới chưa có trong CSDL thì trả về giá trị 2 và thông báo lỗi
- Nếu không vi phạm các lỗi trên thì trả về giá trị 0 và thực hiện bổ sung nhân viên mới vào CSDL

▪ Phân tích:

- Tên SP?
- Số tham số?
- Sử dụng cấu trúc IF kết hợp RETURN để trả kết quả về ứng với từng trường hợp
- Kiểm tra trùng mã nhân viên, có thể dùng lệnh EXISTS
- Kiểm tra mã đơn vị chưa tồn tại, có thể dùng NOT EXISTS

Phan Anh Phong – Vinh University

26

## Ví dụ 6 – Sử dụng Return trong SP

```
create table nhanvien
(manv char(10) primary key,tennv nvarchar(30),namsinh int, madv char(10),
foreign key (madv) references donvi)
CREATE PROC ThemNhanVien (@msnv char(10), @ht nvarchar(30),
@ns int, @dv char(10))
AS
BEGIN
IF(EXISTS(SELECT * FROM nhanvien n WHERE n.manv = @msnv)) BEGIN
    PRINT N'Mã số nhân viên ' + cast(@msnv as varchar(10)) + N' đã tồn tại'
    RETURN -1
END
IF(NOT EXISTS(SELECT * FROM donvi d WHERE d.madv=@dv))
BEGIN
    PRINT N'Mã đơn vị chưa có'
    RETURN -2
END
INSERT INTO NhanVien(manv, tennv, namSinh,madv) VALUES(@msnv, @ht, @ns,@dv)
RETURN 0
END
```

Phan Anh Phong – Vinh University

27

## Ví dụ 6 – Sử dụng Return trong SP

- Select \* from Nhanvien
- exec ThemNhanVien '1', 'Giáp', 1992, 'IT';
- exec ThemNhanVien '2014', 'Quý', 1992, 'ECO';

Phan Anh Phong – Vinh University

28

## Hàm

- Hàm là 1 đối tượng trong SQL Server, tương tự SP
- Hàm khác SP ở chỗ hàm trả về 1 giá trị qua tên của hàm
- Hàm do DBMS cung cấp sẵn (built-in function)
  - Ví dụ: getdate() – trả về ngày, giờ hiện tại của hệ thống
  - Các hàm gộp: min(), max(), avg(), sum(), count()
- Hàm do người sử dụng tự viết
- Có 3 dạng hàm:
  - Hàm gộp:
  - Hàm trả về giá trị vô hướng
  - Hàm trả về một bảng
- Sử dụng hàm:
  - Hàm thường được dùng trong mệnh đề SELECT, hoặc là một thành phần của một biểu thức.

Phan Anh Phong – Vinh University

29

## Hàm trả về giá trị vô hướng

```
CREATE FUNCTION Người_sở_hữu.tên_hàm  
    ([danh_sách_tham_số])  
RETURNS <kiểu_trả_về_của_hàm>  
AS  
    BEGIN  
        {các_câu_lệnh_của_hàm}  
    RETURN <giá_trị_trả_về>  
END
```

Phan Anh Phong – Vinh University

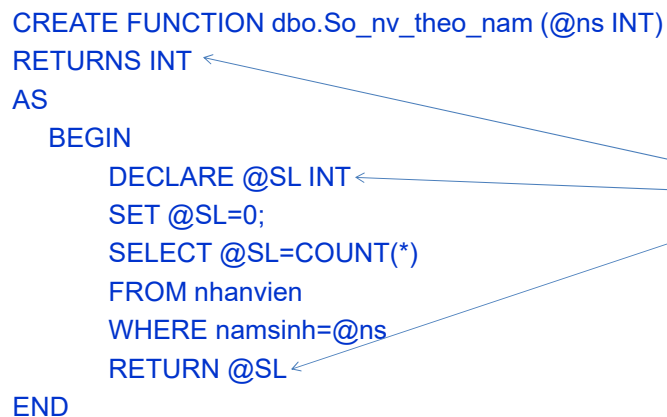
30

## Ví dụ 7 - Hàm trả về giá trị vô hướng

- Viết một hàm đếm số nhân viên có trong CSDL theo năm sinh
- Phân tích:
  - Người sở hữu
  - Tên hàm
  - Tham số
  - Kiểu dữ liệu trả về
  - Thuật toán: sử dụng bảng nào? Câu lệnh SQL ra sao? Tổ chức biến?

## Ví dụ 7 - Hàm trả về giá trị vô hướng

```
CREATE FUNCTION dbo.So_nv_theo_nam (@ns INT)
RETURNS INT
AS
BEGIN
    DECLARE @SL INT
    SET @SL=0;
    SELECT @SL=COUNT(*)
    FROM nhanvien
    WHERE namsinh=@ns
    RETURN @SL
END
```



Sử dụng hàm:

```
select dbo.so_nv_theo_nam(1992)
```



## Hàm trả về giá trị là một BẢNG

```
CREATE FUNCTION Người_sở_hữu.tên_hàm  
    ([danh_sách_tham_số])  
RETURNS TABLE  
AS  
RETURN (Mệnh đề SELECT )
```

Từ SQL SERVER 2005 mới có  
Trả về 1 bảng, do đó, nó được sử dụng như 1 bảng

## Ví dụ 8 - Hàm trả về giá trị là một BẢNG

- Tạo một hàm trả về 1 bảng cho xem tên, năm sinh và tên đơn vị của các nhân viên dựa vào mã đơn vị
- ```
CREATE FUNCTION FUNC_XEM_CB(@mdv char(10))  
RETURNS TABLE  
AS  
    RETURN(SELECT tennv, namsinh, tendv  
    FROM donvi d INNER JOIN nhanvien N ON  
    d.madv=n.madv WHERE n.madv=@mdv)
```
- Sử dụng hàm:  

```
SELECT * FROM FUNC_XEM_CB('IT');
```

## Trigger

- Khái niệm
  - Trigger là một đối tượng cơ sở dữ liệu và luôn gắn với 1 bảng. Nó có thể xem là một Stored Procedure đặc biệt, không có tham số đầu vào, đầu ra, tự động thực hiện (không cần lời gọi) khi csdl có sự thay đổi về nội dung (Insert, Update, Delete)
- Ứng dụng của Trigger
  - Tính toán, cập nhật giá trị tự động
  - Tự động ghi lại những thay đổi về nội dung của một bảng
  - Cập nhật dữ liệu cho nhiều bảng bằng cách sử dụng view
  - ...

Phan Anh Phong – Vinh University

35

## Các dạng Trigger

- Trigger thông thường ( Trigger for/ After trigger)
  - Trigger thực hiện khi các lệnh SQL gây ra trigger được thực thi thành công (after/for)
  - Chỉ làm việc với bảng, không áp dụng với view
- Instead Of
  - Trigger được thực thi thay cho các câu lệnh SQL gây ra trigger (Instead of)
  - Làm việc được với cả table và view

Phan Anh Phong – Vinh University

36

## Tạo mới một Trigger

- Cách 1:  
Sử dụng SQL Server Management Studio (SSMS)
- Cách 2:  
Dùng lệnh CREATE  
Cú pháp:  
CREATE TRIGGER Tên\_Trigger ON Tên\_bảng  
{ [ INSTEAD OF ] | [ FOR | AFTER ] }  
{ [ INSERT [, UPDATE [, DELETE ] ] ] }  
AS  
[DECLARE Biến\_cục\_bộ]  
Các\_lệnh

## Xóa, Sửa một Trigger

- Xóa  
Dùng lệnh DROP TRIGGER Tên\_Trigger
- Sửa  
Dùng lệnh ALTER TRIGGER Tên\_Trigger  
ON Tên\_bảng  
{ [ INSTEAD OF ] | [ FOR | AFTER ] }  
{ [ INSERT [, UPDATE [, DELETE ] ] ] }  
AS  
[DECLARE Biến\_cục\_bộ]  
Các\_lệnh

## Ví dụ tạo mới một Trigger

- Phương pháp:
  - Tạo trigger loại gì?
  - Trigger đó áp dụng cho INSERT/UPADTE/DELETE và bảng nào?
  - Mục đích cụ thể của Triiger này là gì?
- Ví dụ 1:
  - Giả sử một CSDL có 2 sơ đồ sau:  
Department(Dept\_id, Dept\_name, Total)  
Staff(S\_id, S\_name, S\_address, Dept\_id)  
Yêu cầu 1: Mỗi khi thêm 1 nhân viên vào một phòng nào đó thì Tổng số nhân viên của phòng đó sẽ tự động tăng 1.
  - Giải pháp:
    - Giải pháp 1:
    - Giải pháp 2: Sử dụng một trigger

## Ví dụ 1 về Trigger

```
CREATE TRIGGER BoSungNV ON staff FOR INSERT
AS
```

```
UPDATE department SET total=total+1
```

```
WHERE dept_id=(select dept_id from inserted);
```

- Inserted là một bảng tạm logic khi thực hiện chèn

- Kiểm tra hoạt động của Trigger BoSungNV

```
insert into staff values(2001,'Giap',2)
```

```
select * from department;
```

```
select * from staff;
```

## Ví dụ 1 về Trigger

- Yêu cầu 2: Mỗi khi xóa 1 nhân viên vào một phòng nào đó thì Tổng số nhân viên của phòng đó sẽ tự động giảm 1.
- Khi thực hiện xóa nội dung của bảng thì có bảng tạm logic là **deleted**
- CREATE TRIGGER XoaNV ON staff FOR DELETE  
**AS <TỰ VIẾT>**
- Lưu ý: Không có bảng tạm logic updated như inserted và deleted
- Khi update được hiểu là đồng thời delete bản ghi cũ và insert bản ghi mới, do vậy khi update sẽ có 2 bảng tạm logic được tạo ra là **deleted** và **inserted**
- Yêu cầu 3: Mỗi khi thay đổi mã phòng 1 nhân viên thì Tổng số nhân viên của phòng cũ sẽ tự động giảm 1 và Tổng số nhân viên của phòng mới tự động tăng 1

Phan Anh Phong – Vinh University

41

## Ví dụ 1 về Trigger

```
CREATE TRIGGER ChuyenPhongNV ON staff
FOR UPDATE
AS
    UPDATE department SET total=total+1
    WHERE dept_id=(select dept_id from inserted);

    UPDATE department SET total=total-1
    WHERE dept_id=(select dept_id from deleted);
GO
```

Phan Anh Phong – Vinh University

42

## Ví dụ 2 về Trigger

- Bài toán:
  - Cho một bảng dữ liệu, hãy viết một trigger để lưu lại những sự thay đổi về nội dung của bảng đó vào một bảng khác, đồng thời cũng ghi lại kiểu và thời gian thay đổi vào bảng mới
  - Kiểu thay đổi nội dung: INSERT, UPDATE, DELETE
  - Thời gian
- Ví dụ 2:
  - Giả sử một CSDL có bảng Tbl\_Nhanvien như sau:  
CREATE TABLE Tbl\_NhanVien  
(manv INT,  
tennv Varchar(100),  
luong Decimal (10,2)  
)

## Ví dụ 2 về Trigger

- Ví dụ 2:
  - Bảng Tbl\_NhanVien  
CREATE TABLE Tbl\_NhanVien  
(manv INT, tennv Varchar(100), luong Decimal (10,2) )
  - Bảng mới dùng để ghi lại sự thay đổi nội dung của bảng Tbl\_NhanVien  
  
CREATE TABLE Tbl\_NhanVien\_ThayDoi  
(manv INT, tennv Varchar(100), luong Decimal (10,2),  
Kieu\_ThayDoi char(100),  
ThoiGian\_ThayDoi datetime)

## Ví dụ 2 về Trigger

- Loại after trigger / trigger for
- Viết trigger for INSERT
- Viết trigger for UPDATE
- Viết trigger for DELETE

## Ví dụ 2 về Trigger

- Viết After trigger for INSERT
- Thuật toán:
  - Tổ chức biến để lưu các giá trị của bản ghi được chèn? Mấy biến?
  - Bảng tạm logic là Inserted
  - Gán giá trị từ Inserted cho các biến
  - Gán giá trị về kiểu thay đổi, và thời gian thay đổi vào các biến
  - Chèn giá trị ở các biên vào bảng mới

## Ví dụ 2 về Trigger

```
CREATE TRIGGER GhiThayDoitKhiThemNV ON Tbl_Nhanvien
FOR INSERT
AS
declare @ma_nv int;
declare @ten_nv char(100);
declare @luong_nv decimal(10,2);
declare @kieu_td char(100);
select @ma_nv=i.manv from inserted i;
select @ten_nv=i.tennv from inserted i;
select @luong_nv=i.luong from inserted i;
set @kieu_td='Them ban ghi';
insert into Tbl_NhanVien_Thaydoi
      (manv,tennv,luong,Kieu_ThayDoi,ThoiGian_ThayDoi)
values(@ma_nv,@ten_nv,@luong_nv,@kieu_td,getdate());
```

Phan Anh Phong – Vinh University

47

## Ví dụ 2 về Trigger

- Kiểm tra lại hoạt động của Trigger GhiThayDoitKhiThemNV
- INSERT INTO Tbl\_NhanVien VALUES (1,'Giáp',1000);
- select \* from Tbl\_NhanVien;
- select \* from Tbl\_NhanVien\_Thaydoi;

Phan Anh Phong – Vinh University

48



## Ví dụ 2 về Trigger

- Viết After trigger cho UPDATE
- Thuật toán:
  - Tổ chức biến để lưu các giá trị của bản ghi bị thay đổi? Mấy biến?
  - Bảng tạm logic là Inserted hay Deleted?
  - Gán giá trị từ Inserted/Deleted cho các biến
  - Sử dụng hàm Update(Tên\_cột) để biết cột nào bị thay đổi (update)
  - Gán giá trị về kiểu thay đổi, và thời gian thay đổi vào các biến
  - Chèn giá trị ở các biến vào bảng mới

Phan Anh Phong – Vinh University

49

## Ví dụ 2 về Trigger

```
CREATE TRIGGER GhiLai_Khi_Update ON Tbl_Nhanvien FOR UPDATE AS
declare @ma_nv int;
declare @ten_nv char(100);
declare @luong_nv decimal(10,2);
declare @kieu_td char(100);
select @ma_nv=i.manv from inserted i;
select @ten_nv=i.tennv from inserted i;
select @luong_nv=i.luong from inserted i;

if update(tennv) set @kieu_td='Ten bi thay doi';
if update(luong) set @kieu_td='Luong bi thay doi';
insert into Tbl_NhanVien_Thaydoi
    (manv,tennv,luong,Kieu_ThayDoi,ThoiGian_ThayDoi)
values(@ma_nv,@ten_nv,@luong_nv,@kieu_td,getdate());
go
```

Phan Anh Phong – Vinh University

50

## Ví dụ 2 về Trigger

- Kiểm tra lại hoạt động của Trigger GhiLai\_Khi\_Update
- update Tbl\_NhanVien set luong=1400 where manv=1
- select \* from Tbl\_NhanVien;
- select \* from Tbl\_NhanVien\_Thaydoi;

## Ví dụ 2 về Trigger

```
CREATE TRIGGER GhiLai_Khi_Update_1 ON Tbl_Nhanvien FOR UPDATE AS
declare @ma_nv int;
declare @ten_nv char(100);
declare @luong_nv decimal(10,2);
declare @kieu_td char(100);
select @ma_nv=i.manv from inserted i;
select @ten_nv=i.tennv from inserted i;
select @luong_nv=i.luong from inserted i;

if update(tennv) set @kieu_td='Ten bi thay doi';
if update(luong) set @kieu_td='Luong bi thay doi';
insert into Tbl_NhanVien_Thaydoi
    (manv,tennv,luong,Kieu_ThayDoi,ThoiGian_ThayDoi)
values(@ma_nv,@ten_nv,@luong_nv,@kieu_td,getdate());
go
```



Thay inserted bằng deleted thì sao?

## Ví dụ 2 về Trigger

- Viết After trigger cho DELETE
- Thuật toán:
- Xem như bài tập về nhà

## INSTEAD OF Trigger

- Instead of trigger:
  - Trigger được thực thi thay cho các câu lệnh SQL gây ra trigger.
  - INSTEAD OF trigger dùng được cho view {[DELETE]/[INSERT]/[UPDATE]}
  - Xác định câu lệnh mà khi thực thi trên bảng hoặc view sẽ gây ra trigger
  - Deleted, inserted là các bảng tạm logic lưu trữ các giá trị của dòng cũ và mới khi thực hiện xóa, chèn tương ứng
  - Hàm Update(danh\_sách\_tên\_cột) dùng để kiểm tra sự thay đổi trên các cột ứng với Insert và Update nhưng không dùng được cho Delete

## INSTEAD OF Trigger

- Instead of trigger thường được dùng cho View:
  - Cập nhật nhiều bảng cùng một lúc qua một View
  - Kiểm tra tính đúng đắn của dữ liệu nhập
  - Instead of trigger cũng dùng các bảng logic Inserted và Deleted để lưu những thay đổi khi trigger đang thực thi

## Ví dụ về INSTEAD OF Trigger

Tạo một view:

Create view NV\_DV

AS

Select MasoNV, Hodem, Ten, ngaysinh, gioitinh, diachi,  
nv.MasoDV, TenDV, MasoNQL

From NhanVien nv, DonVi dv

Where nv.MasoDV=dv.MasoDV

Tạo một trigger:

Tạo trigger chen\_NVDV trên view NV\_DV dùng để chèn dữ liệu vào các bảng tương ứng khi chèn một bản ghi vào view.

## Ví dụ về INSTEAD OF Trigger

```
Create Trigger chen_NV DV On NV_DV
Instead of Insert
As
Begin
Insert into
    NhanVien(MasoNV,Hodem,Ten,ngaysinh,gioitinh,diachi,M
    asoDV)
Select MasoNV,Hodem,Ten,ngaysinh,gioitinh,diachi,MasoDV
From inserted;
Insert into DonVi(MasoDV,TenDV,MasoNQL)
Select MasoDV,TenDV,MasoNQL
From inserted;
End
```

Phan Anh Phong – Vinh University

57

## Bài tập về nhà

Cho một cơ sở dữ liệu (CSDL) gồm các sơ đồ quan hệ sau:  
Khoa(makhoa char(10), tenkhoa nvarchar(30), soluong int)  
và Sinhvien( mssv char(20), hoten nvarchar(50), makhoa char(10))  
Chú ý: Trong bảng Sinhvien thì Makhoa là khóa ngoài

1. Tạo một Stored Procedure để thêm thông tin của một sinh viên mới vào cơ sở dữ liệu, yêu cầu:
  - Nếu trùng mã số sinh viên thì phải thông báo là mã số sinh viên này đã tồn tại và không thực hiện thêm thông tin về sinh viên đó vào CSDL
  - Nếu dữ liệu về mã khoa của sinh viên mới mà không hợp lệ thì thông báo mã khoa này chưa có và không thực hiện thêm thông tin về sinh viên đó vào CSDL
  - Nếu không vi phạm vào 2 trường hợp trên thì bổ sung thông tin về sinh viên mới vào CSDL
2. Viết trigger để xóa một sinh viên sao cho CSDL vẫn đảm bảo tính nhất quán

Phan Anh Phong – Vinh University

58

## Thảo luận



Phan Anh Phong – Vinh University

59