



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

QUẢN TRỊ CƠ SỞ DỮ LIỆU VỚI SQL SERVER

BÀI 7: HÀM NGƯỜI DÙNG ĐỊNH NGHĨA & VIEW

- ⦿ Hàm người dùng định nghĩa
- ⦿ Khung nhìn View



Hàm người dùng định nghĩa

- ❖ Tạo hàm giá trị vô hướng
- ❖ Tạo hàm giá trị bảng

Khung nhìn View

- ❖ Định nghĩa
- ❖ Các loại View





PHẦN 1

HÀM NGƯỜI DÙNG TỰ ĐỊNH NGHĨA

- ❑ Là một đối tượng CSDL chứa các câu lệnh SQL, được biên dịch sẵn và lưu trữ trong CSDL.
- ❑ Thực hiện một hành động như các tính toán phức tạp và trả về kết quả là một giá trị.
- ❑ Giá trị trả về có thể là:
 - ❖ Giá trị vô hướng
 - ❖ Một bảng

❑ Tương tự như Stored Procedure

- ❖ Là một đối tượng CSDL chứa các câu lệnh SQL, được biên dịch sẵn và lưu trữ trong CSDL.

❑ Khác với Stored Procedure

- Các hàm luôn phải trả về một giá trị, sử dụng câu lệnh RETURN
- Hàm không có tham số đầu ra
- Không được chứa các câu lệnh INSERT, UPDATE, DELETE một bảng hoặc view đang tồn tại trong CSDL
- Có thể tạo bảng, bảng tạm, biến bảng và thực hiện các câu lệnh INSERT, UPDATE, DELETE trên các bảng, bảng tạm, biến bảng vừa tạo trong thân hàm

VÍ DỤ VỀ HÀM NGƯỜI DÙNG ĐỊNH NGHĨA

❑ Viết hàm tính tuổi của người có năm sinh là @ns:

```
--Xóa hàm nếu đã có
if object_id('fTuoi','FN') is not null
    drop function fTuoi
go

--Tạo hàm fTuoi
Create function fTuoi (@ns int)
Returns int
As
Begin
    return year(getdate()) - @ns
end
go

--Biên dịch hàm với F5
--Kiểm tra thử hàm
print dbo.fTuoi(1982) --phải có dbo.
```

Kiểu hàm	Mô tả
Hàm giá trị vô hướng	Trả về giá trị đơn của mọi kiểu dữ liệu T-SQL.
Hàm giá trị bảng đơn giản	Trả về bảng, là kết quả của một câu lệnh SELECT đơn.
Hàm giá trị bảng nhiều câu lệnh	Trả về bảng, là kết quả của nhiều câu lệnh.

TẠO HÀM GIÁ TRỊ VÔ HƯỚNG

```

CREATE FUNCTION [<tên schema>.] <Tên hàm>
([@<tên tham số> <kiểu dữ liệu> [= <Giá trị mặc định>]] [, ...])
RETURNS <kiểu dữ liệu>
[WITH [ENCRYPTION] [, SCHEMABINDING] [, <Mệnh đề EXECUTE AS>]]
[AS]
BEGIN
    [<Câu lệnh SQL>]
    RETURN <Biểu thức vô hướng>
END
  
```

- ☐ Không thể truyền tham số theo tên
- ☐ Truyền đầy đủ các tham số theo vị trí. Kể cả tham số tùy chọn, nếu muốn sử dụng giá trị mặc định, phải đặt từ khóa DEFAULT tại đúng vị trí tham số tùy chọn đó.

- ❑ Câu lệnh tạo hàm giá trị vô hướng trả về tổng số lượng nhân viên:

```
--Câu lệnh tạo hàm giá trị vô hướng trả về tổng số lượng nhân viên  
CREATE FUNCTION fDemNv()  
RETURNS int  
BEGIN  
    RETURN (SELECT count(MANV)  
            FROM NHANVIEN)  
END
```

```
--Gọi hàm cách 1  
PRINT 'Tong so luong nhan vien:' + CONVERT(varchar, dbo.fDemNv())  
--Gọi hàm cách 2  
declare @tong int  
select @tong=dbo.fDemNv()
```

125 %

Messages

Tong so luong nhan vien:10

- ❑ Câu lệnh tạo hàm giá trị vô hướng trả về tổng số lượng nhân viên theo giới tính:

```
--Câu lệnh tạo hàm giá trị vô hướng trả về  
--tổng số lượng nhân viên theo giới tính  
CREATE FUNCTION fDemNv_gioitinh(@phai nvarchar(3))  
RETURNS int  
BEGIN  
    RETURN (SELECT count(MANV)  
            FROM NHANVIEN  
            where PHAI like @phai)  
END
```

```
PRINT 'Tong so luong nhan vien:' + CONVERT(varchar, dbo.fDemNv_gioitinh(N'Nam'))
```

□ Cú pháp

```
CREATE FUNCTION [<tên schema>.] <tên hàm>
  ([@<tên tham số> <kiểu dữ liệu> [= <Giá trị mặc định>]] [, ...])
  RETURNS TABLE
  [WITH {ENCRYPTION|SCHEMABINDING|ENCRYPTION,
  SCHEMABINDING}]
  [AS]
  RETURN [(] <Câu lệnh SELECT> [)]
```

- Hàm giá trị bảng đơn giản còn gọi là hàm giá trị bảng nội tuyến
- Hàm giá trị bảng đơn giản có thể được dùng trong câu lệnh truy vấn thay thế cho tên bảng hoặc tên view

- ❑ Viết hàm trả về bảng các nhân viên làm việc ở phòng số 5

```
--Viết hàm trả về bảng các nhân viên làm việc ở phòng số 5
CREATE FUNCTION FNhanVien_PB(@Maphg int)
RETURNS Table
AS
    RETURN
    (
        SELECT MANV,HONV,TENNV FROM NHANVIEN
        WHERE PHG = @Maphg
    )
GO

-- Gọi hàm
SELECT * FROM FNhanVien_PB(005)
```

Results		Messages	
	MANV	HONV	TENNV
1	003	Trần	Tâm
2	004	Nguyễn	Hùng
3	005	Nguyễn	Tùng
4	009	Đinh	Tiên
5	017	Đinh	Tiên

□ Cú pháp

```
CREATE FUNCTION [<tên schema>] <tên hàm>
([@<tên tham số> <tên kiểu dữ liệu> [= <Giá trị mặc định>]]
[,...])
RETURNS @<tên biến trả về> TABLE
(<tên cột 1> <kiểu dữ liệu> [<Các thuộc tính cột>]
[, <tên cột 2> <kiểu dữ liệu> [<Các thuộc tính cột>]]...)
[WITH [ENCRYPTION] [, SCHEMABINDING] [, <mệnh đề  
EXECUTE AS>]]
[AS]
BEGIN
<Các câu lệnh SQL>
RETURN
END
```

❑ Ví dụ hàm giá trị bảng đa câu lệnh

--Hàm nhận dữ liệu từ bảng phòng ban

```
CREATE FUNCTION [dbo].[fListPhong] (@phong Int)
```

```
RETURNS @ProdList Table
```

```
( ten nvarchar(15) , ma int, trphg nvarchar(9), ngay date
)
```

```
AS
```

```
BEGIN
```

```
IF @phong IS NULL
```

```
BEGIN
```

```
INSERT INTO @ProdList (ten,ma,trphg,ngay)
```

```
SELECT TENPHG, MAPHG,TRPHG,NG_NHANCHUC
```

```
FROM PhongBan
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
INSERT INTO @ProdList (ten,ma,trphg,ngay)
```

```
SELECT TENPHG, MAPHG,TRPHG,NG_NHANCHUC
```

```
FROM PHONGBAN
```

```
WHERE MAPHG=@phong
```

```
END
```

```
RETURN
```

```
END
```

```
select * from dbo.fListPhong(NULL)
```

	ten	ma	trphg	ngay
1	Quản Lý	1	006	1971-06-19
2	Điều Hành	4	008	1985-01-01
3	Nghiên Cứu	5	005	0197-05-22
4	CNTT	6	008	1985-01-01

❑ Cú pháp câu lệnh DROP FUNCTION

DROP FUNCTION [<tên schema>.] <tên hàm> [, ...]

❑ Cú pháp lệnh Alter FUNCTION cho hàm giá trị vô hướng



- ❖ Tạo hàm nhận tham số đầu vào là giới tính nam hoặc nữ và đếm số lượng nhân viên theo giới tính.
- ❖ Tạo hàm nhận tham số đầu vào là mã phòng, trả về bảng các nhân viên thuộc mã phòng đó.



PHẦN 2

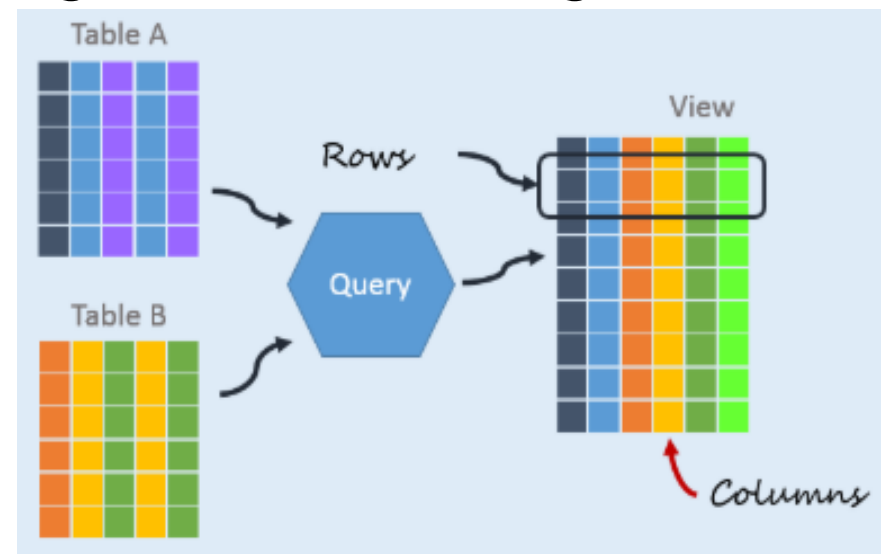
❑ Che dấu và bảo mật dữ liệu

- ❖ Không cho phép người dùng xem toàn bộ dữ liệu chứa trong các bảng.
- ❖ Bằng cách chỉ định các cột trong View, các dữ liệu quan trọng chứa trong một số cột của bảng có thể được che dấu

❑ Hiển thị dữ liệu một cách tùy biến

- ❖ Với mỗi người dùng khác nhau, có thể tạo các View khác nhau phù hợp với nhu cầu xem thông tin của từng người dùng

- ❑ Lưu trữ câu lệnh truy vấn phức tạp và thường xuyên sử dụng.
- ❑ Thực thi nhanh hơn các câu lệnh truy vấn do đã được biên dịch sẵn
- ❑ Đảm bảo tính toàn vẹn dữ liệu
 - ❖ Khi sử dụng View để cập nhật dữ liệu trong các bảng cơ sở, SQL Server sẽ tự động kiểm tra các ràng buộc toàn vẹn trên các bảng



```
CREATE VIEW <tên view> [( <tên cột 1> [, <tên cột 2> ]...)]  
[WITH  
{ENCRYPTION|SCHEMABINDING|ENCRYPTION,SCHEMABINDING}]  
AS  
<Câu lệnh SELECT>  
[WITH CHECK OPTION]
```

- ❑ Tên view không được trùng với tên bảng hoặc view đã tồn tại
- ❑ Câu lệnh SELECT tạo VIEW
 - ❖ Không được chứa mệnh đề INTO, hoặc ORDER BY trừ khi chứa từ khóa TOP
- ❑ Đặt tên cột
 - ❖ Cột chứa giá trị được tính toán từ nhiều cột khác phải được đặt tên
 - ❖ Nếu cột không được đặt tên, tên cột sẽ được mặc định giống tên cột của bảng cơ sở

- ❑ Ví dụ tạo view chứa 2 thông tin là tên nhân viên và tên phòng ban

```
--Tạo View chứa 2 thông tin là tên nhân viên và tên phòng ban
```

```
Create View NV_PB
```

```
as
```

```
Select TENNV,TENPHG
```

```
From NHANVIEN inner join PhongBan on NHANVIEN.PHG=PHONGBAN.MAPHG
```

Xem dữ liệu

```
SELECT * from NV_PB
```

	TENNV	TENPHG
1	Như	Điều Hành
2	The	Điều Hành
3	Tâm	Nghiên Cứu
4	Hùng	Nghiên Cứu
5	Tùng	Nghiên Cứu

❑ VIEW chỉ đọc (read-only view)

- ❖ View này chỉ dùng để xem dữ liệu

❑ VIEW có thể cập nhật (updatable view)

- ❖ Xem dữ liệu
- ❖ Có thể sử dụng câu lệnh INSERT, UPDATE, DELETE để cập nhật dữ liệu trong các bảng cơ sở qua View

- ❑ Yêu cầu: Câu lệnh select không được chứa
 - ❖ Mệnh đề DISTINCT hoặc TOP
 - ❖ Một hàm kết tập (Aggregate function)
 - ❖ Một giá trị được tính toán
 - ❖ Mệnh đề GROUP BY và HAVING
 - ❖ Toán tử UNION
 - ❖ Nếu câu lệnh tạo View vi phạm một trong số điều kiện trên. VIEW được tạo ra là VIEW chỉ đọc

- ❑ Ví dụ tạo View có thể cập nhật được, hiển thị tên phòng có mã phòng 6

```
CREATE VIEW ThongTin_Phong
AS
SELECT TENPHG,MAPHG
FROM PHONGBAN
WHERE MAPHG =6
```

```
select *from ThongTin_Phong
```

	TENPHG	MAPHG
1	CNTT	6

- ❑ Câu lệnh UPDATE cập nhật view đổi tên phòng thành 'IT'

```
UPDATE ThongTin_Phong
SET TENPHG = 'IT'
```

```
select *from ThongTin_Phong
```


	TENPHG	MAPHG
1	IT	6

❑ Ví dụ tạo view chỉ đọc

```
Create view ThongtinNV  
as
```

```
Select TenNV, LUONG, YEAR(GetDate()) - YEAR(NHANVIEN.NGSINH) as Tuổi  
from NHANVIEN  
where YEAR(GetDate()) - YEAR(NHANVIEN.NGSINH) < 57
```

Biểu thức tính toán



```
select * from ThongtinNV
```

	TenNV	LUONG	Tuoi
1	Tâm	25000	62
2	Hành	25000	65
3	Tiên	30000	59

❑ Cú pháp câu lệnh xóa View

DROP VIEW <tên View>

❑ Cú pháp câu lệnh chỉnh sửa View

```
ALTER VIEW <tên View> [(<tên cột 1> [, <tên cột 2>]...)]
```

```
[WITH
```

```
{ENCRYPTION|SCHEMABINDING|ENCRYPTION,SCHEMABINDING}]
```

```
AS <câu lệnh SELECT>
```

```
[WITH CHECK OPTION]
```

XÓA VÀ CHỈNH SỬA VIEW

```

CREATE VIEW ThongTin_Phong
AS
  SELECT TENPHG,MAPHG
  FROM PHONGBAN
  WHERE MAPHG =6
  
```

```

Alter VIEW ThongTin_Phong
AS
  SELECT TENPHG,MAPHG
  FROM PHONGBAN
  WHERE MAPHG =5
  
```

```

drop view Thongtin_Phong
  
```



- ❖ Tạo View hiển thị Danh sách những trưởng phòng (HONV, TENLOT, TENNV) có tối thiểu một thân nhân

☑ Hàm người dùng định nghĩa

- ❖ Tạo hàm giá trị vô hướng
- ❖ Tạo hàm giá trị bảng

☑ Khung nhìn View

- ❖ Định nghĩa
- ❖ Các loại View



thank
you!