

## Lab 07

### View và Store PROCEDURE

#### Phần I – Hướng dẫn thực hành từng bước

##### 1. Mục tiêu

- Định nghĩa View, tạo, chỉnh sửa và xóa View.
- Hiểu khái niệm PROCEDURE, các kiểu STORE PROCEDURE, tạo, sửa, xóa và kích hoạt một PROCEDURE.
- PROCEDURE lồng nhau.
- Truy vấn SQL Server metadata.

##### 2. Thực hiện

**Views** là bảng ảo được tạo thành từ các cột đã được chọn từ một hoặc nhiều bảng.

Những bảng mà từ đó Views được tạo ra gọi là bảng cơ sở. Những bảng này có thể từ csdl khác nhau. Views có thể có tối đa là 1024 cột.

Nói một cách đơn giản thì việc tạo View giúp việc truy cập dữ liệu cần thiết đơn giản hơn, thay vì mỗi lần phải gõ một cơ sở dòng lệnh để lấy về tập hợp dữ liệu thì chỉ phải gõ một dòng SELECT tới View là xong.

**Bài thực hành 1:** Tạo một View từ bảng Product để hiển thị ID sản phẩm, số sản phẩm, tên và mức độ lưu kho an toàn của sản phẩm.

Bước 1: Viết lệnh SQL như sau:

```
USE QuanLyBanHang
GO

CREATE VIEW vwProductInfo AS
SELECT ProductId, ProductName, PriceInput, PriceOutput, Status
FROM Product;
GO
```

Tiền tố “vw” trong tên của View là quy ước để ám chỉ đây là view, phân biệt với các đối tượng khác.

Bước 2: Gõ mã lệnh truy vấn View:

```
SELECT * FROM vwProductInfo
```

	ProductID	ProductName	PriceInput	PriceOutput	Status
1	1	Giày buộc giày công sở Sanvado	350000	400000	1
2	2	Giày nam buộc dây James Blanc	500000	600000	1
3	3	Giày da mềm Asos	350000	400000	1
4	4	Giày cao gót mũi nhọn	60000	650000	1
5	5	Giày cao gót đế đinh đá mũi hờ	350000	450000	1
6	6	Giày cao gót dây buộc Asos	550000	600000	1
7	7	Đép xỏ ngón nhiều màu - DT170	750000	800000	1
8	8	Đép quai ngang	300000	350000	1
9	9	Đép đế xuồng hoa hồng quai trong	350000	400000	1

Q | 6LWMEK2EIS43HRG\SQLEXPRESS ... | sa (51) | QuanLyBanHang | 00:00:00 | 20 rows

**Bài thực hành 2:** Tạo View có tên là **vwProductDetail** với các cột được chỉ định từ bảng Product và bảng Catalog.

Bước 1: Gõ lệnh SQL:

```
-- QuanLyBanHang
CREATE VIEW vwProductDetails
AS
SELECT p.ProductId,p.ProductName, p.PriceOutput
      ,c.CatalogName ,p.Status
FROM Product p
INNER JOIN Catalog c
ON p.CatalogId = c.CatalogId
GO
```

Bước 2: Thực thi dòng lệnh trên và viết lệnh truy vấn VIEW vừa tạo:

```
SELECT * FROM vwProductDetails
```

	ProductId	ProductName	PriceOutput	CatalogName	Status
2	2	Giày nam buộc dây James Blanc	600000	Giày nam	1
3	3	Giày da mềm Asos	400000	Giày nam	1
4	4	Giày cao gót mũi nhọn	650000	Giày nữ	1
5	5	Giày cao gót đế đinh đá mũi hờ	450000	Giày nữ	1
6	6	Giày cao gót dây buộc Asos	600000	Giày nữ	1
7	7	Đép xỏ ngón nhiều màu - DT170	800000	Đép nữ	1
8	8	Đép quai ngang	350000	Đép nữ	1
9	9	Đép đế xuồng hoa hồng quai trong	400000	Đép nữ	1

**Lưu ý:**

- Một view được tạo bằng việc sử dụng câu lệnh CREATE VIEW.
- Tên view phải là duy nhất, không thể trùng với tên các bảng khác trong cùng lược đồ.
- View không thể tạo trên các bảng tạm.
- View không thể có full-text index.
- View không chứa định nghĩa DEFAULT.
- Câu lệnh CREATE VIEW chỉ có thể bao gồm mệnh đề ORDER BY nếu như có từ khóa TOP được sử dụng.
- View không thể tham chiếu hơn 1024 cột.
- Câu lệnh CREATE VIEW không thể bao gồm từ khóa INTO .
- Câu lệnh CREATE VIEW không thể kết hợp với các lệnh Transact-SQL khác trong cùng một khối (batch).

**Bài thực hành 3:** Viết câu lệnh tạo VIEW vwSortedProductDetail.**Bước 1:** Gõ lệnh SQL:

```
-- Tạo View đã sắp xếp và lấy ra 10 sản phẩm đầu tiên
CREATE VIEW vwSortedProductDetails
AS
SELECT TOP 10
        p.ProductId, p.ProductName, p.PriceOutput, c.CatelogName
FROM Product p
INNER JOIN Catelog c
ON c.CatelogId = p.CatelogId
ORDER BY p.ProductName -- Sắp xếp theo tên
GO
```

**Bước 2:** Thực thi lệnh SQL sau:

```
SELECT * FROM vwSortedProductDetails
```

Results		Messages		
	ProductId	ProductName	PriceOutput	CatalogName
1	16	Áo sơ mi nam dài tay VŨ TUẤN	450000	Áo nam
2	14	Áo sơ mi nam ngắn tay họa tiết	400000	Áo nam
3	15	Áo thun nam cổ tròn ECO JEA	300000	Áo nam
4	13	Áo thun Nike cổ tròn	400000	Áo nam
5	17	Áo voan cộc tay cao cổ	900000	Áo nữ
6	10	Đép cao su nam quai liền	680000	Đép nam
7	9	Đép đế xuồng hoa hồng quai trong	400000	Đép nữ
8	8	Đép quai ngang	350000	Đép nữ
9	12	Đép xỏ ngón kito thailand	850000	Đép nam
10	7	Đép xỏ ngón nhiều màu - DT170	800000	Đép nữ

Có thể thực hiện các thao tác sau trên **VIEW**:

- ✓ INSERT
- ✓ UPDATE
- ✓ DELETE

Khi INSERT, giá trị cho cột được cung cấp tự động nếu:

- ✓ Cột có thuộc tính IDENTITY.
- ✓ Cột có giá trị mặc định đã chỉ ra.
- ✓ Cột có kiểu dữ liệu timestamp.
- ✓ Cột có các giá trị null.
- ✓ Cột là một cột tính.

**Bài thực hành 4:** Viết câu lệnh tạo 2 bảng: **Employee\_Personal\_Details** và **Employee\_Salary\_Details** như mô tả ở bảng bên dưới. Sau đó tạo một view tên là **vwEmployee\_Details** kết nối 2 bảng này dựa trên cột **EmpID**. Viết lệnh thêm dữ liệu vào view tạo trên.

Bảng <b>Employee_Personal_Details</b>		
Cột	Kiểu dữ liệu	Mô tả
EmpID	INT NOT NULL	ID
FirstName	varchar(30) NOT NULL	Họ

LAStName	varchar(30) NOT NULL	Tên
Address	varchar(30)	Địa chỉ

Bảng Employee_Salary_Details		
Cột	Kiểu dữ liệu	Mô tả
EmpID	INT NOT NULL	ID
Designation	varchar(30)	Chỉ định
Salary	INT NOT NULL	Lương

**Bước 1: Gõ lệnh SQL tạo bảng:**

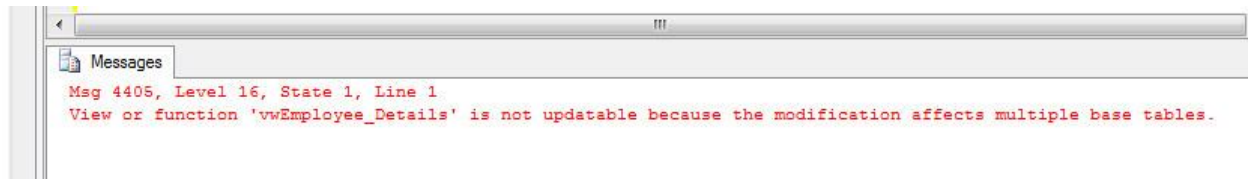
```
-- Thực hiện INSERT trên VIEW
-- Tạo bảng
CREATE TABLE Employee_Personal_Details
(
EmpID INT NOT NULL,
FirstName varchar(30) NOT NULL,
LAStName varchar(30) NOT NULL,
Address varchar(30)
)
GO
CREATE TABLE Employee_Salary_Details
(
EmpID
INT NOT NULL,
Designation varchar(30),
Salary INT NOT NULL
)
```

**Bước 2: Tạo view:**

```
-- Tạo View
CREATE VIEW vwEmployee_Details
AS
SELECT e1.EmpID, FirstName, LAStName, Designation, Salary
FROM Employee_Personal_Details e1
JOIN Employee_Salary_Details e2
ON e1.EmpID = e2.EmpID
```

**Bước 3: Thực hiện lệnh INSERT:**

```
-- Thêm dữ liệu vào view
INSERT INTO vwEmployee_Details VALUES
(2, 'Jack', 'Wilson', 'Software Developer', 16000)
```



Câu lệnh lỗi vì thêm dữ liệu vào View có ảnh hưởng tới nhiều bảng cơ sở.

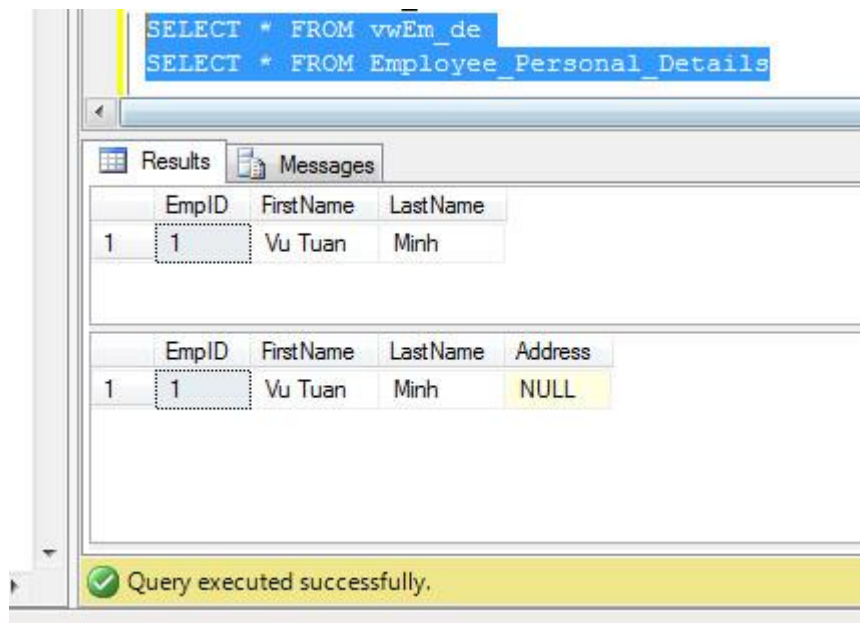
#### Bước 4: Tạo một View khác như sau:

```
-- Tạo View
CREATE VIEW vwEm_de AS
SELECT EmpID, FirstName, LastName
FROM Employee_Personal_Details
GO

-- Thêm dữ liệu
INSERT INTO vwEm_de VALUES (1, 'Vu Tuan', 'Minh')
```

#### Bước 5: Kết quả truy vấn trên View và trên bảng cơ sở:

```
SELECT * FROM vwEm_de
SELECT * FROM Employee_Personal_Details
```



Lệnh thực thi thành công và dữ liệu được chèn vào bảng cơ sở.

**Bài thực hành 5:** Tạo bảng Product\_Details theo mô tả như bảng dưới, sao đó tạo view vwProduct\_Details rồi thực hiện cập nhật cho VIEW:

Bảng Product_Details		
Cột	Kiểu dữ liệu	Mô tả
ProductID	INT	ID
ProductName	varchar(30)	Tên sản phẩm
Rate	money	Đánh giá

**Bước 1: Tạo bảng:**

```
-- UPDATE View
-- Tạo bảng
CREATE TABLE Product_Details
(
ProductID INT,
ProductName varchar(30),
Rate money
)
```

**Bước 2: Thêm dữ liệu:**

```
-- Thêm dữ liệu
INSERT INTO Product_Details VALUES
(1, 'Kem đánh răng', N'PK nhà vệ sinh', 50),
(2, 'Bàn chải', N'PK nhà vệ sinh', 33),
(3, 'Xà phòng tắm', N'HC phòng tắm', 69),
(4, 'Bong ngoáy tai', N'VV may mặc', 99),
(6, 'Dầu gội', N'HC phòng tắm', 39)
-- Truy vấn
SELECT * FROM Product_Details
```

	ProductID	ProductName	Rate	Description
	1	Kem đánh răng	3000.0000	PK nhà vệ sinh
	2	Bàn chải	33.0000	PK nhà vệ sinh
	3	Xà phòng tắm	69.0000	HC phòng tắm
	4	Bong ngoáy tai	86.0000	BT cá nhân
	5	Khan mat	99.0000	VV may mặc
	6	Dầu gội	39.0000	HC phòng tắm
»»	NULL	NULL	NULL	NULL

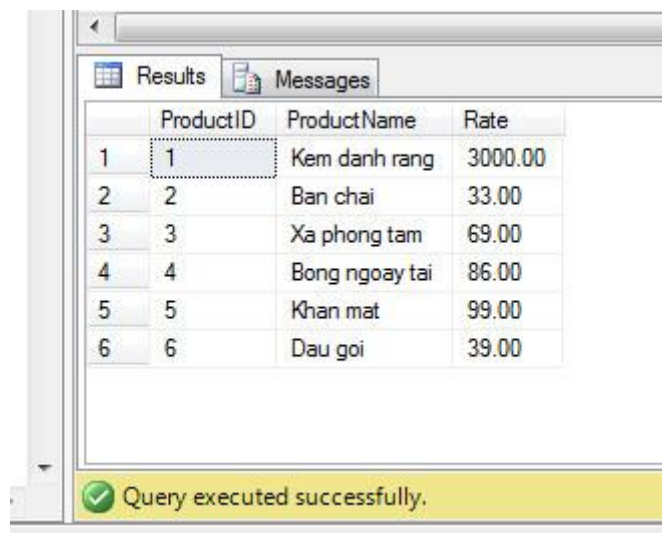
**Bước 3: Tạo View:**

```
-- Tạo view
CREATE VIEW vwProduct_Details
AS
```

```
SELECT
ProductName, Rate FROM Product_Details
```

#### Bước 4: Cập nhật:

```
-- Cập nhật View
UPDATE vwProduct_Details
SET Rate=3000
WHERE ProductName = 'Kem danh rang'
```



	ProductID	ProductName	Rate
1	1	Kem danh rang	3000.00
2	2	Ban chai	33.00
3	3	Xa phong tam	69.00
4	4	Bong ngoay tai	86.00
5	5	Khan mat	99.00
6	6	Dau goi	39.00

Query executed successfully.

Các kiểu dữ liệu giá trị lớn bao gồm varchar(max), NVARCHAR(max), và varbinary(max). Để cập nhật dữ liệu có các kiểu dữ liệu giá trị lớn, mệnh đề .WRITE được sử dụng. Mệnh đề .WRITE chỉ ra rằng một phần của giá trị trong cột sẽ được sửa đổi. Mệnh đề .WRITE không thể được sử dụng để cập nhật giá trị NULL trong một cột. Ngoài ra, không thể sử dụng mệnh đề này để đặt giá trị cột thành NULL.

Cú pháp: *column\_name*.WRITE (*expression*, @Offset, @Length)

Trong đó:

**column\_name:** chỉ ra tên của cột có kiểu dữ liệu lớn.

**expression:** chỉ ra giá trị được copy tới cột.

**@Offset:** chỉ ra điểm bắt đầu của giá trị trong cột mà Expression sẽ được ghi tại đó.



**@Length:** chỉ ra độ dài của đoạn(section) trong cột.

### Bài thực hành 6: Tạo View **vwProduct\_Details** sau đó cập nhật dữ liệu ở cột

ProductName:

#### Bước 1: Gõ lệnh SQL:

```
-- Sửa bảng và thêm cột
ALTER TABLE Product_Details
ADD [Description] NVARCHAR(max)
GO

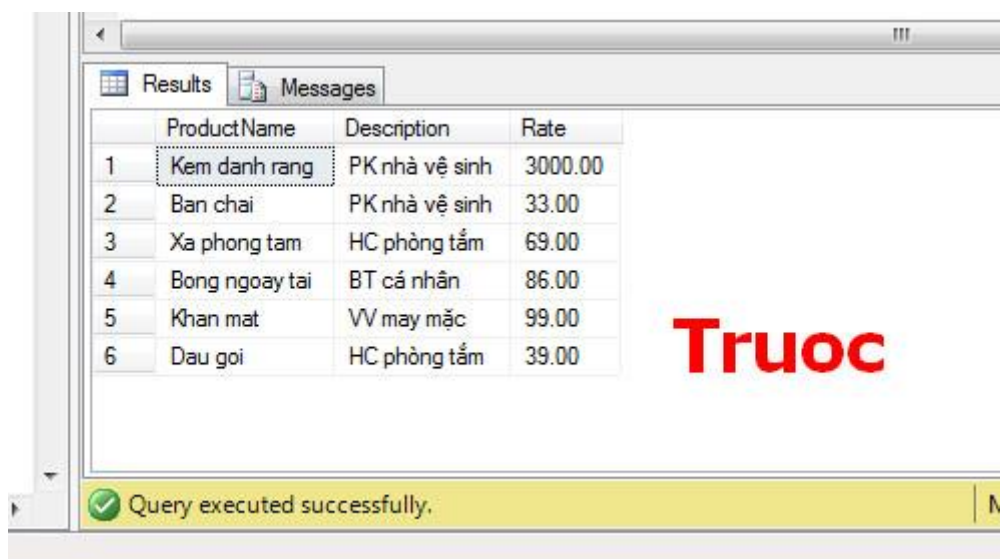
-- Tạo View
CREATE VIEW vwProduct_Full_Details
AS
    SELECT ProductName, [Description], Rate
    FROM Product_Details
```

#### Bước 2: Sửa dữ liệu “ảo” cho cột Description:



ProductID	ProductName	Rate	Description
1	Kem đánh răng	3000.0000	PK nhà vệ sinh
2	Bàn chải	33.0000	PK nhà vệ sinh
3	Xà phòng tắm	69.0000	HC phòng tắm
4	Bong ngoay tai	86.0000	BT cá nhân
5	Khăn mặt	99.0000	VV may mặc
6	Dầu gội	39.0000	HC phòng tắm
NULL	NULL	NULL	NULL

#### Bước 3: Gõ lệnh SQL cập nhật Views:



	ProductName	Description	Rate
1	Kem đánh răng	PK nhà vệ sinh	3000.00
2	Bàn chải	PK nhà vệ sinh	33.00
3	Xà phòng tắm	HC phòng tắm	69.00
4	Bong ngoay tai	BT cá nhân	86.00
5	Khan mat	VV may mặc	99.00
6	Dau goi	HC phòng tắm	39.00

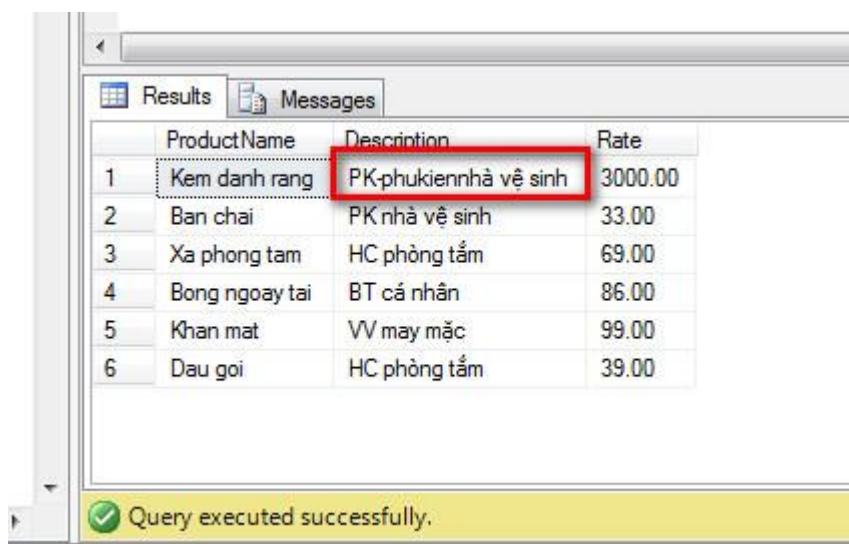
**Truoc**

Query executed successfully.

```
-- Cập nhật View với dữ liệu text
UPDATE vwProduct_Full_Details
SET [Description] .WRITE(N'-phukien',2,1)
WHERE ProductName='Kem đánh răng'
GO
```

```
SELECT * FROM vwProduct_Full_Details
```

Kết quả:



	ProductName	Description	Rate
1	Kem đánh răng	PK-phukiennhà vệ sinh	3000.00
2	Bàn chải	PK nhà vệ sinh	33.00
3	Xà phòng tắm	HC phòng tắm	69.00
4	Bong ngoay tai	BT cá nhân	86.00
5	Khan mat	VV may mặc	99.00
6	Dau goi	HC phòng tắm	39.00

Query executed successfully.

Ký tự bắt đầu từ vị trí 2 và chiều dài (offset) là 1 tương đương với khoảng trắng (dấu cách) giữa từ “PK” và “nhà” được thay thế bằng từ “-phukien”.

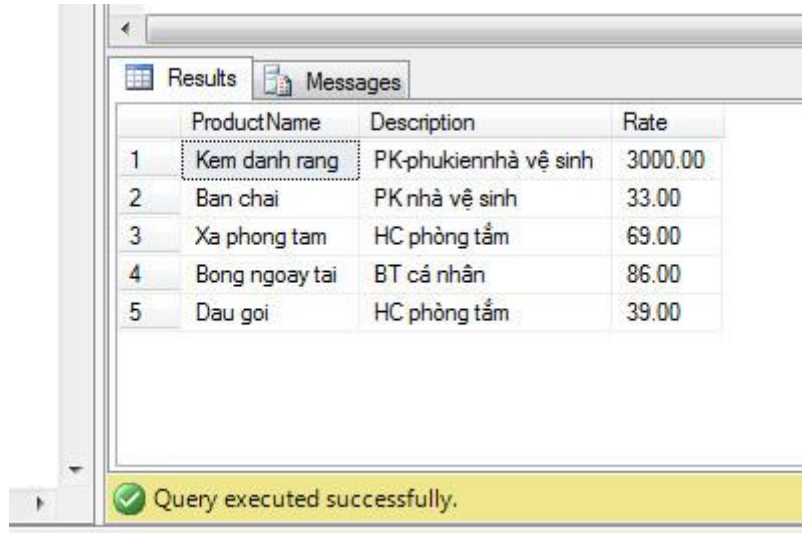
**Bài thực hành 7:** Delete dòng trong VIEW vwProduct\_Full\_Details tại tên sản phẩm là “Khăn mặt”:

Bước 1: Gõ lệnh SQL:

```
-- Xóa trong View  
DELETE FROM vwProduct_Full_Details WHERE ProductName LIKE 'Khan%'
```

**Bước 2: Truy vấn lại View:**

```
SELECT * FROM vwProduct_Full_Details
```



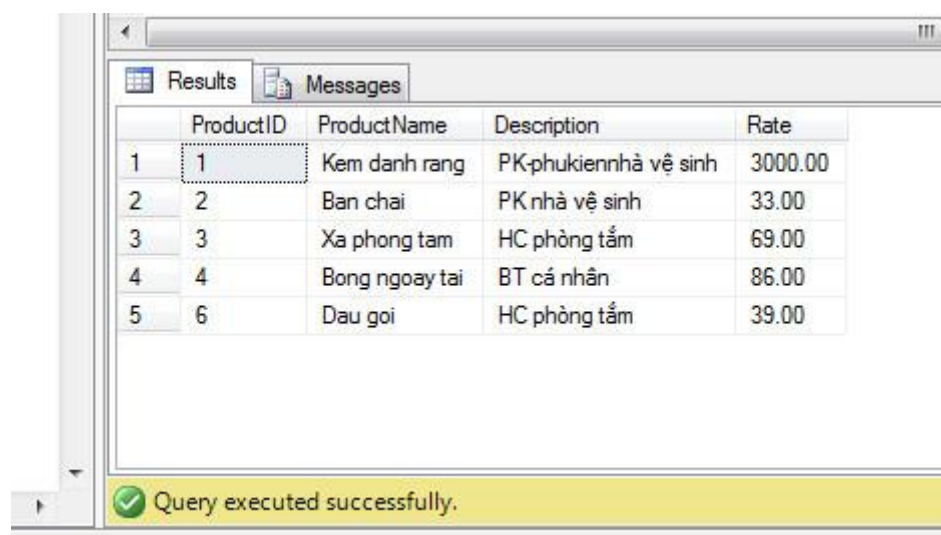
	ProductName	Description	Rate
1	Kem đánh răng	PK-phukiennhà vệ sinh	3000.00
2	Ban chải	PK nhà vệ sinh	33.00
3	Xa phòng tắm	HC phòng tắm	69.00
4	Bong ngoáy tai	BT cá nhân	86.00
5	Dầu gội	HC phòng tắm	39.00

**Bài thực hành 8: Chính sửa lại View để cập nhật thêm cột ID:****Bước 1: Gõ lệnh SQL:**

```
-- Chính sửa View  
ALTER VIEW vwProduct_Full_Details  
AS  
    SELECT ProductID, ProductName, [Description], Rate  
    FROM Product_Details
```

**Bước 2: Truy vấn View vừa sửa:**

```
-- Truy vấn  
SELECT * FROM vwProduct_Full_Details
```



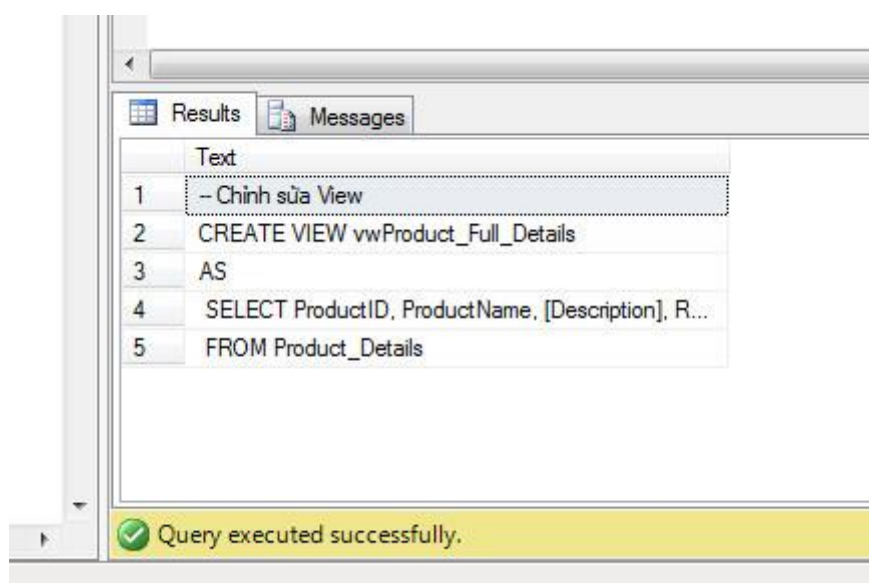
	ProductID	ProductName	Description	Rate
1	1	Kem danh rang	PK-phukiennhà vệ sinh	3000.00
2	2	Bàn chải	PK nhà vệ sinh	33.00
3	3	Xà phòng tắm	HC phòng tắm	69.00
4	4	Bong ngoay tai	BT cá nhân	86.00
5	6	Dau goi	HC phòng tắm	39.00

**Bài thực hành 9:** Xem mã nguồn viết lên View:

Bước 1: Gõ lệnh SQL:

```
-- Xem mã nguồn SQL viết lên View
EXEC sp_helptext vwProduct_Full_Details
```

Bước 2: Quét dòng mã lệnh trên rồi bấm F5:



	Text
1	-- Chỉnh sửa View
2	CREATE VIEW vwProduct_Full_Details
3	AS
4	SELECT ProductID, ProductName, [Description], R...
5	FROM Product_Details

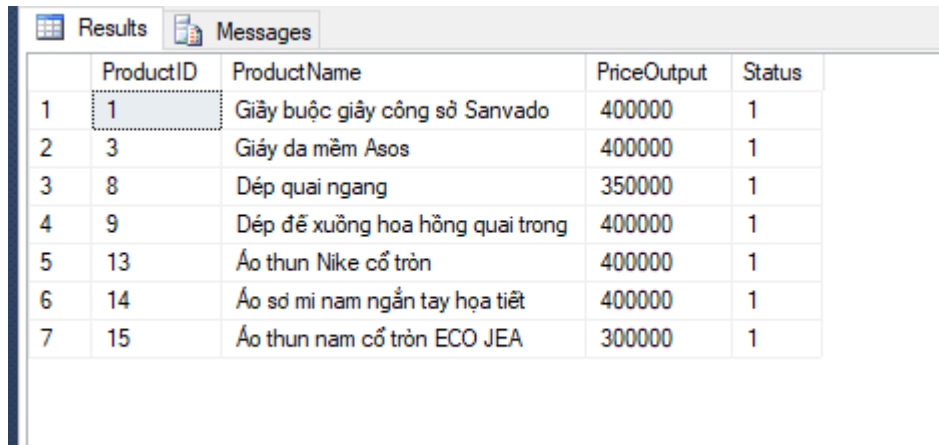
**Bài thực hành 10:** Thao tác với từ khóa WITH CHECK OPTION:

Bước 1: Gõ lệnh SQL:

```
-- View với từ khóa With Check Option
CREATE VIEW vwProductInfo AS
SELECT ProductId, ProductName, PriceOutput, Status
FROM Product
```

```
WHERE PriceOutput <= 400000
WITH CHECK OPTION;
GO
```

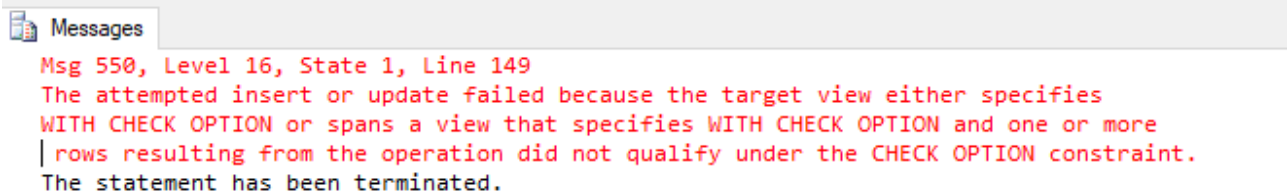
Bước 2: Viết lệnh SQL cập nhật View với PriceOutput bằng 500000 tại ProductID bằng 13:



	ProductID	ProductName	PriceOutput	Status
1	1	Giày buộc giấy công sở Sanvado	400000	1
2	3	Giày da mềm Asos	400000	1
3	8	Đép quai ngang	350000	1
4	9	Đép đế xuồng hoa hồng quai trong	400000	1
5	13	Áo thun Nike cổ tròn	400000	1
6	14	Áo sơ mi nam ngắn tay họa tiết	400000	1
7	15	Áo thun nam cổ tròn ECO JEA	300000	1

```
-- Lệnh cập nhật.
UPDATE vwProductInfo set PriceOutput = 500000
WHERE ProductId = 13
```

Và lỗi:



Msg 550, Level 16, State 1, Line 149  
The attempted insert or update failed because the target view either specifies WITH CHECK OPTION or spans a view that specifies WITH CHECK OPTION and one or more rows resulting from the operation did not qualify under the CHECK OPTION constraint. The statement has been terminated.

Nguyên nhân lỗi là do **CHECK OPTION** kiểm tra câu lệnh nhận thấy nó vi phạm định nghĩa của View trên là chỉ nhận PriceOutput <= 400000. Và tất nhiên, nếu câu chuyện ngược lại là dữ liệu trên bảng cơ sở được cập nhật thì lệnh sẽ thành công tuy nhiên dữ liệu đó sẽ không hiển thị ở trong View.

### SCHEMABINDING:

Thông thường khi VIEW được tạo thì mọi chỉnh sửa (vd: thêm cột, xóa cột...) sau đó trên bảng cơ sở sẽ không được cập nhật. Nếu muốn ràng buộc việc chỉnh sửa bảng với VIEW (không cho phép chỉnh sửa trừ khi xóa bỏ sự phụ thuộc này) có thể dùng với từ khóa SCHEMABINDING.

**Bài thực hành 11:** Viết lệnh tạo ràng buộc bảng cơ sở :

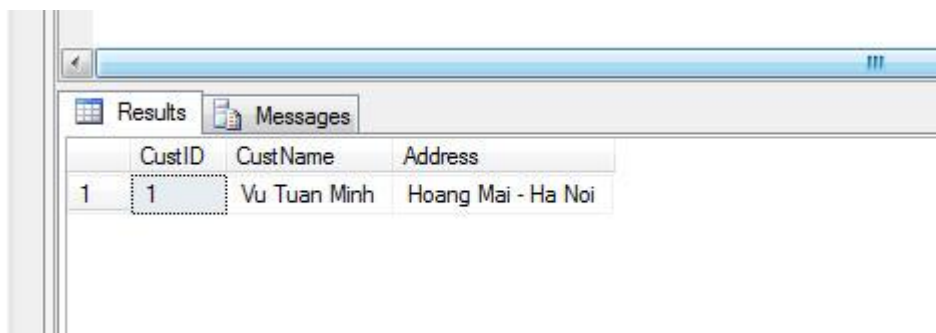
Bước 1: Tạo bảng Customers, sau đó tạo view cho bảng này như sau:

```
-- Tạo bảng
CREATE TABLE Customers
(
  CustID INT,
  CustName varchar(50),
  Address varchar(60)
)
GO

-- Tạo view
CREATE VIEW vwCustomers
AS
SELECT * FROM Customers
GO
```

Bước 2: Thêm dữ liệu vào bảng cơ sở, sau đó truy cập View:

```
-- Thêm dữ liệu
INSERT INTO Customers VALUES
(1, 'Vu Tuan Minh', 'Hoang Mai - Ha Noi')
GO
SELECT * FROM vwCustomers
GO
```

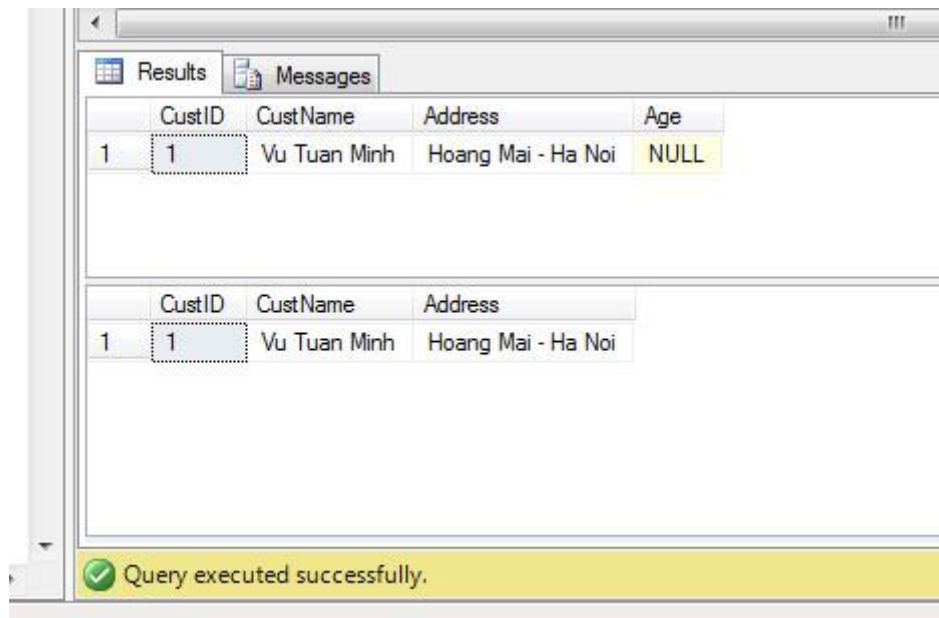


The screenshot shows the SQL Server Enterprise Manager interface. The 'Results' window is active, displaying a table with three columns: CustID, CustName, and Address. The table contains one row of data: CustID 1, CustName Vu Tuan Minh, and Address Hoang Mai - Ha Noi.

CustID	CustName	Address
1	Vu Tuan Minh	Hoang Mai - Ha Noi

Bước 3: Thêm cột vào bảng cơ sở, sau đó truy cập View:

```
-- Chính sửa bảng
ALTER TABLE Customers ADD Age INT
GO
SELECT * FROM Customers
GO
SELECT * FROM vwCustomers
GO
```



The screenshot shows a SQL Server query result window. The 'Results' tab is active, displaying a table with the following data:

CustID	CustName	Address	Age
1	Vu Tuan Minh	Hoang Mai - Ha Noi	NULL

Below the table, there is a status bar that says 'Query executed successfully.' with a green checkmark icon.

Trên View không được cập nhật thêm cột Age

Bước 4: Gõ hàm refresh, sau đó truy cập View:

```
-- Refresh view
EXEC sp_refreshview vwCustomers
GO
SELECT * FROM vwCustomers
GO
```



The screenshot shows a SQL Server query result window. The 'Results' tab is active, displaying a table with the following data:

CustID	CustName	Address	Age
1	Vu Tuan Minh	Hoang Mai - Ha Noi	NULL

Below the table, there is a status bar that says 'Query executed successfully.' with a green checkmark icon.

**Bài thực hành 12:** Tạo thủ tục và thực thi:

Bước 1: Gõ lệnh SQL:

```
-- Tạo thủ tục
```

```
CREATE PROCEDURE getProductInfor
AS
BEGIN
SELECT top 10
ProductId, ProductName, PriceOutput, CatalogName
FROM Product inner join Catalog
ON Product.CatalogId = Catalog.CatalogId
END
```

**Bước 2: Quét dòng mã lệnh trên rồi bấm F5:**

```
-- Thực thi thủ tục
EXEC getProductInfor
```

	ProductId	ProductName	PriceOutput	CatalogName
1	1	Giày buộc giấy công sở Sanvado	400000	Giày nam
2	2	Giày nam buộc dây James Blanc	600000	Giày nam
3	3	Giày da mềm Asos	400000	Giày nam
4	4	Giày cao gót mũi nhọn	650000	Giày nữ
5	5	Giày cao gót đính đá mũi hờ	450000	Giày nữ
6	6	Giày cao gót dây buộc Asos	600000	Giày nữ
7	7	Dép xỏ ngón nhiều màu - DT170	800000	Dép nữ
8	8	Dép quai ngang	350000	Dép nữ
9	9	Dép đế xuồng hoa hồng quai trong	400000	Dép nữ
10	10	Dép cao su nam quai liền	680000	Dép nam

**Bài thực hành 13: Tạo thủ tục có tham số truyền vào và thực thi:**

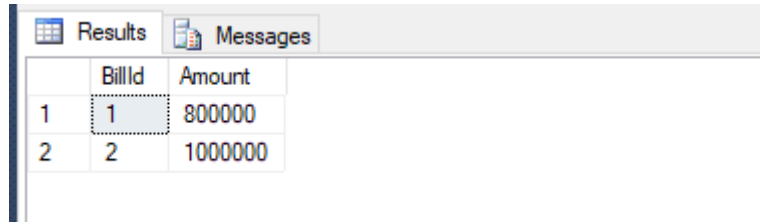
**Bước 1: Gõ lệnh SQL:**

```
-- Tạo thủ tục có tham số truyền vào
CREATE PROCEDURE getBillBuyCustomer
@id INT
AS
BEGIN
SELECT BillId, Amount FROM Bill
WHERE CustomerId = @id
END
```

**Bước 2: Quét dòng mã lệnh trên rồi bấm F5:**

```
-- Kích hoạt stored PROCEDURE
EXEC getBillBuyCustomer 1
```





	BillId	Amount
1	1	800000
2	2	1000000

**Bài thực hành 14:** Tạo thủ tục có tham số trả về và thực thi:

Bước 1: Gõ lệnh SQL:

```
-- Thủ tục có tham số trả về
CREATE PROCEDURE uspGetTotalSales
@name NVARCHAR(50),
@sum INT OUTPUT
AS
BEGIN
SELECT @sum= SUM(BillId)
FROM Bill b
JOIN Customer c
ON b.CustomerId = c.CustomerId
WHERE c.CustomerName = @name
END
GO
```

Bước 2: Quét dòng mã lệnh trên rồi bấm F5:

```
-- Thực thi thủ tục
DECLARE @sum INT;
EXEC totalBill 'Trinh Dinh Long', @sum OUTPUT;
PRINT 'So luong don hang: ' + convert(varchar(100),@sum);
GO
```



**Bài thực hành 15:** Xem mã nguồn tạo thủ tục:

Bước 1: Gõ lệnh SQL:

```
-- Xem mã nguồn tạo thủ tục
EXEC sp_helptext getBillBuyCustomer
```

	Text
1	create procedure getBillBuyCustomer
2	@id int
3	as
4	begin
5	Select BillId, Amount from Bill
6	where CustomerId = @id
7	end

**Bước 2: Sửa thủ tục:**

```
-- Sửa thủ tục
ALTER PROCEDURE getBillBuyCustomer
@name NVARCHAR(50)
AS
SELECT BillId, Amount
FROM Bill b
JOIN Customer c
ON b.CustomerId = c.CustomerId
WHERE c.CustomerName = @name;
GO
```

**Bước 3: Xóa thủ tục:**

```
-- Xóa thủ tục
DROP PROCEDURE getBillBuyCustomer
```

**Bài thực hành 16: Truy cập dữ liệu metadata:****Bước 1: Gõ lệnh SQL:**

```
-- Truy cập metadata
SELECT TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, TABLE_TYPE
FROM INFORMATION_SCHEMA.TABLES;
```

100 %

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	QuanLyBanHang	dbo	Catelog	BASE TABLE
2	QuanLyBanHang	dbo	Product	BASE TABLE
3	QuanLyBanHang	dbo	Color	BASE TABLE
4	QuanLyBanHang	dbo	Size	BASE TABLE
5	QuanLyBanHang	dbo	ProductDetail	BASE TABLE
6	QuanLyBanHang	dbo	Customer	BASE TABLE
7	QuanLyBanHang	dbo	Bill	BASE TABLE
8	QuanLyBanHang	dbo	BillDetail	BASE TABLE
9	QuanLyBanHang	dbo	vwProductDetails	VIEW
10	QuanLyBanHang	dbo	vwSortedProductDetails	VIEW
11	QuanLyBanHang	dbo	vwProductInfo	VIEW

**Phần II - Bài tập tự làm**

Bài số 1: Sử dụng câu lệnh T-SQL thực hiện các yêu cầu sau

**Question 1:**

- Tạo Cơ sở dữ liệu
- Use Cơ sở dữ liệu này

**Question 2:**

Tạo 3 bảng theo mẫu sau:

*[QuanHuyen]*

```
(  
    [MAQH] [INT] IDENTITY(1,1) NOT NULL,  
    [TenQH] [NVARCHAR](100) NOT NULL  
)
```

*[DuongPho]*

```
(  
    [DuongID] [INT] NOT NULL,  
    [MAQH] [INT] NOT NULL,  
    [TENDUONG] NVARCHAR(max) not null,  
    [NgayDuyetTen] DATETIME NULL  
)
```

*[Nha\_Tren\_Phó]*

```
(  
    [NhaID] [INT] NOT NULL,  
    [DuongID] [INT] NOT NULL,  
    [ChuHo] [NVARCHAR](50) NULL,  
    [Dientich] [money] NULL  
)
```

**Question 3:**

Sử dụng câu lệnh Alter table để thêm cột [SoNhanKhu] kiểu [INT] vào bảng [Nha\_tren\_pho].

**Question 4:**

Áp dụng Primary Key ConstraiNT cho cột [MAQH] của bảng [QuanHuyen].

Áp dụng Primary Key ConstraiNT cho cột [**DuongID**] của bảng [**DuongPho**].

Áp dụng Primary Key ConstraiNT cho cột [**NhaID**] trên bảng [**Nha\_Tren\_Ph**].

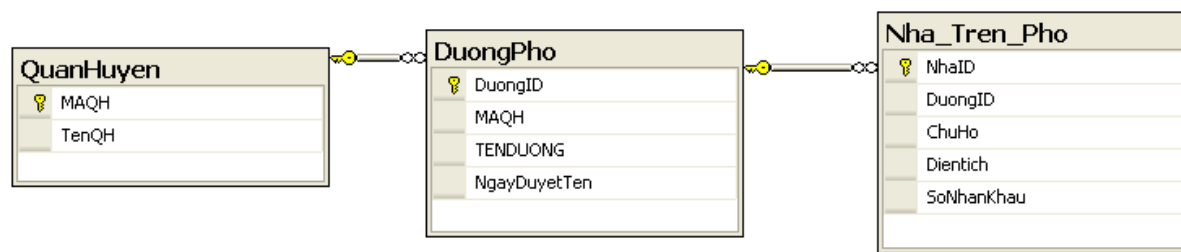
Áp dụng Foreign Key ConstraiNT cho bảng [**Nha\_Tren\_Ph**] tham chiếu tới bảng [**DuongPho**].

Áp dụng Foreign Key ConstraiNT cho bảng [**DuongPho**] tham chiếu tới bảng [**Quanhuyen**].

Áp dụng Check ConstraiNT cho cột [**NgayDuyetTen**] trong bảng [**DuongPho**] với giá trị trong khoảng “02/09/1945” và ngày hiện tại.

Áp dụng Unique ConstraiNT cho cột [**TenQH**] trong bảng [**QuanHuyen**].

Áp dụng Default ConstraiNT cho cột [**SoNhanKhau**] trong bảng [**Nha\_Tren\_Ph**] với giá trị mặc định 1.



### Question 5:

Insert dữ liệu theo mẫu:

QuanHuyen	
MAQH	TENQH
1	Pho Cau Giay
2	Hoang Mai

DuongPho			
DuongID	MaQH	TENDuong	NgayDuyetTen
1	1	Hoang Quoc Viet	1998-02-30
2	1	Pham Van Dong	1998-02-30
3	2	Tran Cung	1998-02-30

NhaTrenPho				
NhaID	DuongID	ChuHo	DienTich	SoNhanKhau
1	1	DienNQ	250	2
2	1	HienNV	69	4
3	2	HoangVH	99	6

Question 6:

Sử dụng câu lệnh update với .WRITE(expression, @offset, @Length) function để thay đổi [TENDUONG] từ “Pho Cau Giay” thành “Duong Cau Giay”.

**Question 7:**

Tạo view có tên [vw\_all\_Nha\_Tren\_Phlo] kết hợp dữ liệu từ 3 bảng: [Nha\_Tren\_Phlo], [DuongPho] và [QuanHuyen].

Thông tin gồm các cột: QuanHuyen.TenQH, DuongPho.TENDUONG, DuongPho.NgayDuyetTen, Nha\_Tren\_Phlo.ChuHo, Nha\_Tren\_Phlo.Dientich, Nha\_Tren\_Phlo.SoNhanKhau

Hiển thị view [vw\_all\_Nha\_Tren\_Phlo]

**Question 8:**

Sử dụng câu lệnh SELECT and hàm AVG (Ví dụ: AVG (Nha\_Tren\_Phlo.Dientich) và AVG (Nha\_Tren\_Phlo.SonhanKhau)) và mệnh đề group by (Ví dụ: group by DuongPho.TENDUONG) để tạo view có tên [vw\_AVG\_Nha\_Tren\_Phlo] kết hợp dữ liệu từ 2 bảng [Nha\_Tren\_Phlo] và [DuongPho].

- Kết quả view hiển thị theo mẫu:

	TENDUONG	AVG_Dientich	AVG_SoNhanKhau
1	Duong Vinh Phuc	40.00	1
2	Duong Doi Can	60.00	8

- Sắp xếp theo thứ tự tăng dần của average [Dientich] và average [SonhanKhau] (Ví dụ: order by AVG\_Dientich,AVG\_SoNhanKhau).

- Hiển thị view [vw\_AVG\_Nha\_Tren\_Phlo]

**Question 9:**

Tạo thủ tục có tên [sp\_NgayQuietTen\_DuongPho]. Thủ tục này nhận 1 tham số là

@NgayDuyet (Date time type):

- Hiển thị thông tin với điều kiện:

NgayQuietTen = @NgayDuyet.

- Kết quả thủ tục trả về :

	NgayDuyetTen	TENDUONG	TenQH
1	1998-12-30 00:00:00.000	Duong Vinh Phuc	Ba Dinh

- Thực hiện thủ tục [sp\_NgayQuietTen\_DuongPho] với tham số ['30/12/1998'] .

Gợi ý:

Bạn phải khai báo 1 biến

Sử dụng Convert (DATETIME,'dd/mm/yyyy', 103).