

## 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:

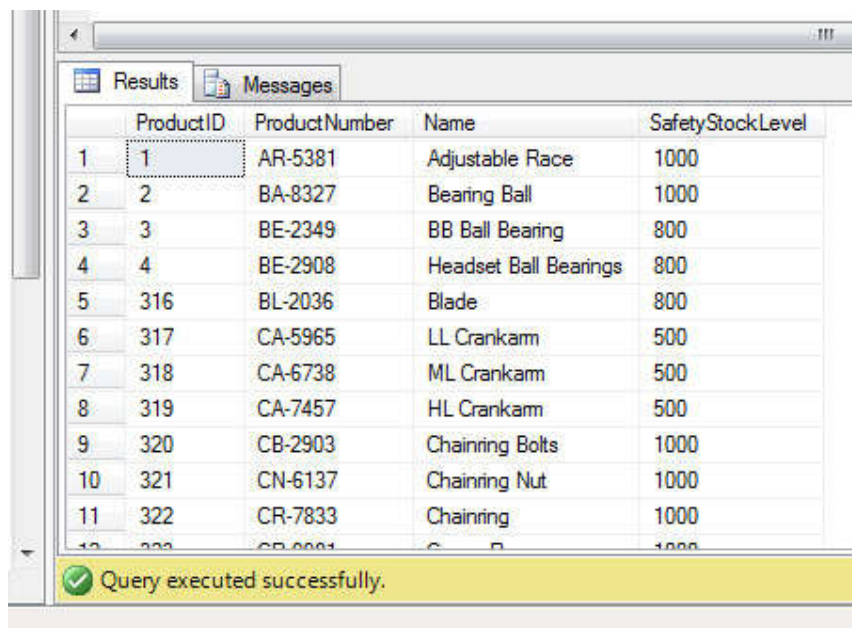
```
USE AdventureWorks
Go

CREATE VIEW vwProductInfo AS
SELECT ProductID, ProductNumber, Name, SafetyStockLevel
FROM Production.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	ProductNumber	Name	SafetyStockLevel
1	1	AR-5381	Adjustable Race	1000
2	2	BA-8327	Bearing Ball	1000
3	3	BE-2349	BB Ball Bearing	800
4	4	BE-2908	Headset Ball Bearings	800
5	316	BL-2036	Blade	800
6	317	CA-5965	LL Crankarm	500
7	318	CA-6738	ML Crankarm	500
8	319	CA-7457	HL Crankarm	500
9	320	CB-2903	Chaining Bolts	1000
10	321	CN-6137	Chaining Nut	1000
11	322	CR-7833	Chaining	1000

**Bài thực hành 2:** Tạo View có tên là **vwPersonDetail** với các cột được chỉ định từ bảng Person và bảng Employee.

**Bước 1:** Gõ lệnh SQL:

```
-- AdventureWork 2008
CREATE VIEW vwPersonDetails
AS
SELECT p.Title,p.[FirstName],p.[MiddleName]
      ,p.[LastName] ,e.[Title] AS 'JobTitle'
FROM [HumanResources].[Employee] e
INNER JOIN Person.Contact p
ON p.[ContactID] = e.[ContactID]
GO

/*
-- AdventureWork 2012
CREATE VIEW vwPersonDetails
AS
SELECT p.Title,p.[FirstName],p.[MiddleName]
      ,p.[LastName] ,e.[JobTitle]
FROM [HumanResources].[Employee] e
INNER JOIN [Person].[Person] p
ON p.[BusinessEntityID] = e.[BusinessEntityID]
GO

*/
```

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

```
SELECT * FROM vwPersonDetails
```

	Title	FirstName	MiddleName	LastName	JobTitle
1	NULL	Guy	R	Gilbert	Production Technician - WC60
2	NULL	Kevin	F	Brown	Marketing Assistant
3	NULL	Roberto	NULL	Tamburello	Engineering Manager
4	NULL	Rob	NULL	Walters	Senior Tool Designer
5	NULL	Thierry	B	D'Hers	Tool Designer
6	NULL	David	M	Bradley	Marketing Manager
7	NULL	JoLynn	M	Dobney	Production Supervisor - WC60

Query executed successfully. MINHVI

Cột Title (Mr, Mrs...) có giá trị NULL nên sẽ gây khó hiểu. Hàm COALESCE() được thêm vào sẽ có tác dụng loại bỏ giá trị NULL.

#### Bước 2: Viết lại câu lệnh tạo Views:

Lưu ý cần xóa VIEWS trước để viết lại

```
-- Xóa VIEW vwPersonDetails
DROP VIEW vwPersonDetails
GO

-- Tạo lại VIEW với hàm COALESCE()
CREATE VIEW vwPersonDetails
AS
SELECT COALESCE(p.Title, 'GAY/LES') AS 'Title',
       p.[FirstName],p.[MiddleName]
       ,p.[LastName] ,e.[Title] AS 'JobTitle'
FROM [HumanResources].[Employee] e
INNER JOIN Person.Contact p
ON p.[ContactID] = e.[ContactID]
GO
```

Results		Messages			
	Title	FirstName	MiddleName	LastName	JobTitle
1	GAY/LES	Guy	R	Gilbert	Production Technician - WC60
2	GAY/LES	Kevin	F	Brown	Marketing Assistant
3	GAY/LES	Roberto	NULL	Tamburello	Engineering Manager
4	GAY/LES	Rob	NULL	Walters	Senior Tool Designer
5	GAY/LES	Thierry	B	D'Hers	Tool Designer
6	GAY/LES	David	M	Bradley	Marketing Manager
7	GAY/LES	JoLynn	M	Dobney	Production Supervisor - WC60
8	GAY/LES	Ruth	Ann	Ellerbrock	Production Technician - WC10
9	Ms.	Gail	A	Erickson	Design Engineer
10	GAY/LES	Bary	K	Johnson	Production Technician - WC10
11	Mr.	Jossef	H	Goldberg	Design Engineer
12	GAY/LES	Teri	Lee	Duffy	Vice President of Engineering
13	GAY/LES	Sidney	M	Higa	Production Technician - WC10
14	GAY/LES	Taylor	R	Maxwell	Production Supervisor - WC50
15	GAY/LES	Jeffrey	L	Ford	Production Technician - WC10
16	GAY/LES	Jo	A	Brown	Production Supervisor - WC60
17	GAY/LES	Doris	M	Hartwig	Production Technician - WC10

Query executed successfully. MINHVUF

**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).

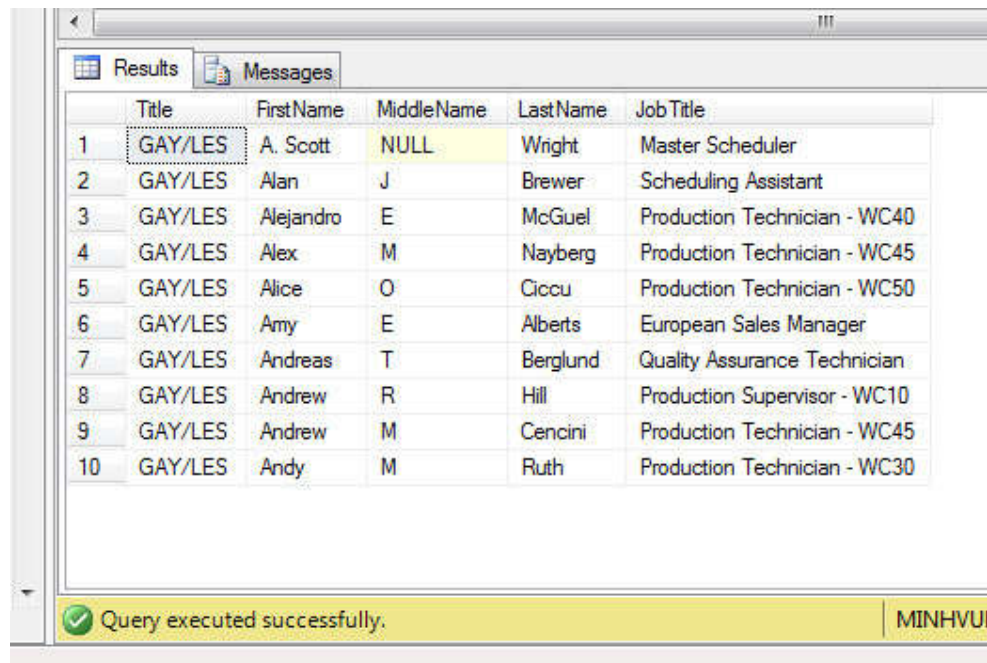
**Bài thực hành 3:** Viết câu lệnh tạo VIEW vwSortedPersonDetail.

**Bước 1:** Gõ lệnh SQL:

```
-- Tạo View đã sắp xếp và lấy ra 10 người đầu tiên
CREATE VIEW vwSortedPersonDetails
AS
SELECT TOP 10
    COALESCE(p.Title, 'GAY/LES') AS 'Title',
    p.[FirstName],p.[MiddleName]
    ,p.[LastName] ,e.[Title] AS 'JobTitle'
FROM [HumanResources].[Employee] e
INNER JOIN Person.Contact p
ON p.[ContactID] = e.[ContactID]
ORDER BY p.FirstName -- Sắp xếp theo tên
GO
```

**Bước 2:** Thực thi lệnh SQL sau:

```
SELECT * FROM vwSortedPersonDetails
```



	Title	FirstName	MiddleName	LastName	Job Title
1	GAY/LES	A. Scott	NULL	Wright	Master Scheduler
2	GAY/LES	Alan	J	Brewer	Scheduling Assistant
3	GAY/LES	Alejandro	E	McGuel	Production Technician - WC40
4	GAY/LES	Alex	M	Nayberg	Production Technician - WC45
5	GAY/LES	Alice	O	Ciccu	Production Technician - WC50
6	GAY/LES	Amy	E	Alberts	European Sales Manager
7	GAY/LES	Andreas	T	Berglund	Quality Assurance Technician
8	GAY/LES	Andrew	R	Hill	Production Supervisor - WC10
9	GAY/LES	Andrew	M	Cencini	Production Technician - WC45
10	GAY/LES	Andy	M	Ruth	Production Technician - WC30

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 Employee_Personal_Details		
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
```

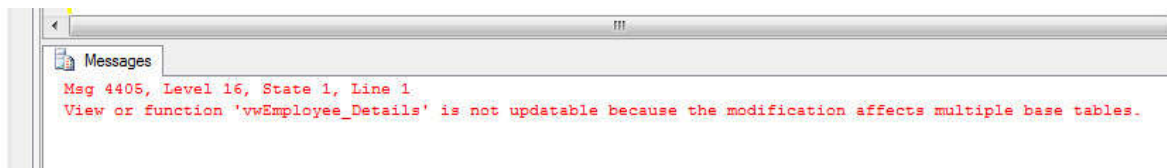
```
CREATE TABLE Employee_Personal_Details
(
EmpID int NOT NULL,
FirstName varchar(30) NOT NULL,
LastName varchar(30) NOT NULL,
Address varchar(30)
)
GO
CREATE TABLE Employee_Salary_Details
(
EmpID
int NOT NULL,
Designation varchar(30),
Salary int NOT NULL
)
```

**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:**

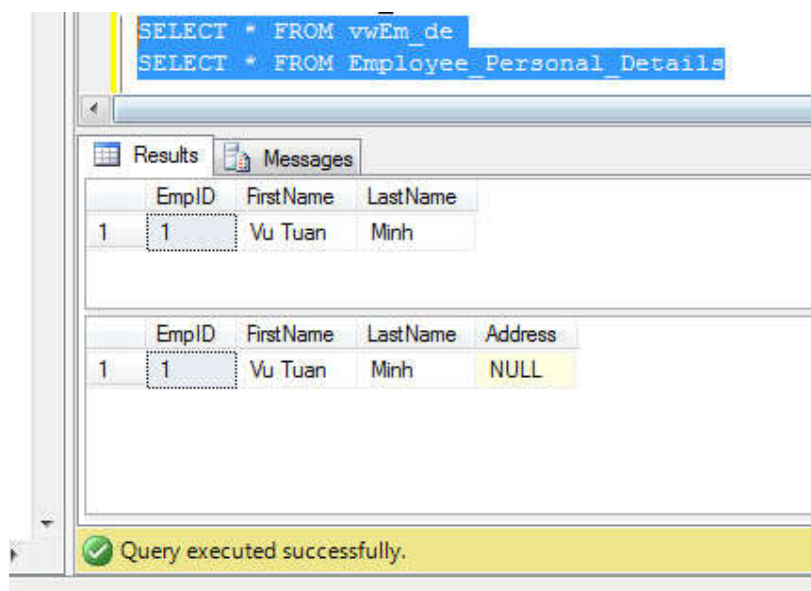
```
-- Tạo View
CREATE VIEW vwEm_de AS
SELECT EmpID, FirstName, LastName
FROM Employee_Personal_Details
GO

-- 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, sau 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:**

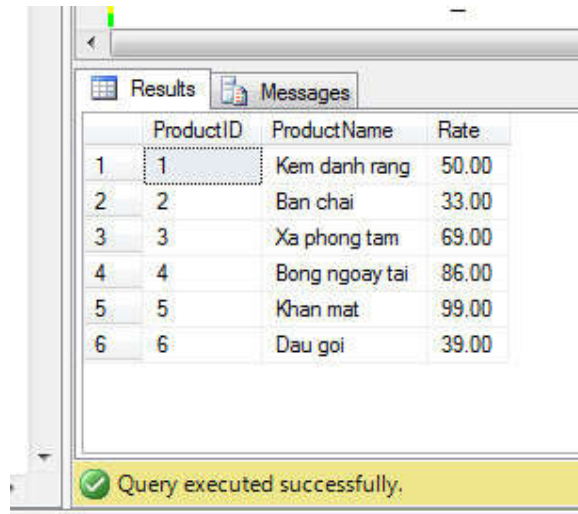
```
-- UPDATE View
-- Tạo bảng
CREATE TABLE Product_Details
(
ProductID int,
ProductName varchar(30),
Rate money
)
```



**Bước 2: Thêm dữ liệu:**

```
-- Thêm dữ liệu
INSERT INTO Product_Details VALUES
(1, 'Kem danh rang', 50),
(2, 'Ban chai', 33),
(3, 'Xa phong tam', 69),
(4, 'Bong ngoay tai', 86),
(5, 'Khan mat', 99),
(6, 'Dau goi', 39)

-- Truy vấn
SELECT * FROM Product_Details
```



	ProductID	ProductName	Rate
1	1	Kem danh rang	50.00
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

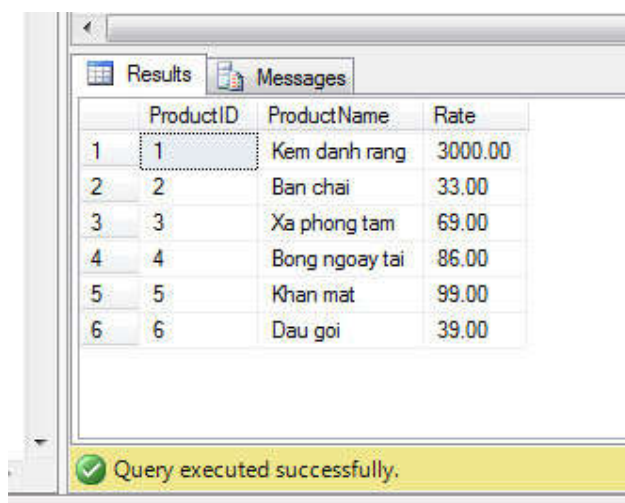
Query executed successfully.

**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 đánh răng	3000.00
2	2	Ban chải	33.00
3	3	Xà phòng tắm	69.00
4	4	Bong ngoáy tai	86.00
5	5	Khăn mặt	99.00
6	6	Dầu gội	39.00

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 danh rang	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	Khan mat	99.0000	VV may mặc
6	Dau goi	39.0000	HC phòng tắm
NULL	NULL	NULL	NULL

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

ProductID	ProductName	Description	Rate
1	Kem danh rang	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

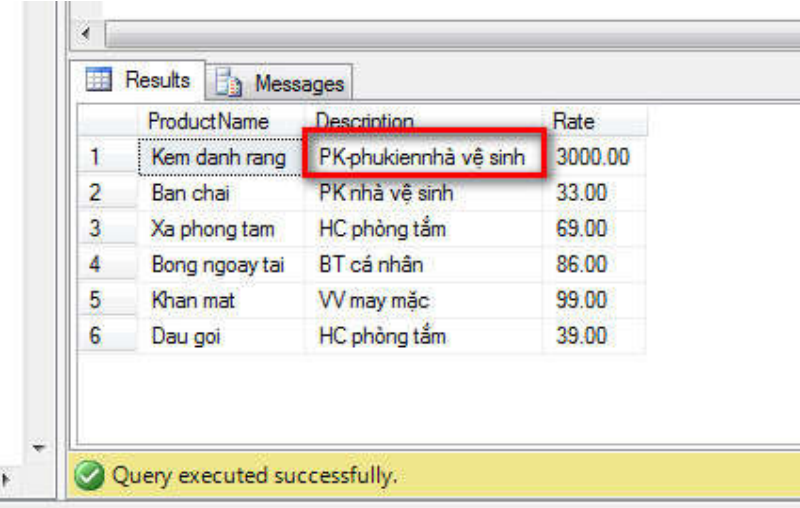
**Trước**

Query executed successfully.

```
-- Cập nhật View với dữ liệu text
UPDATE vwProduct_Full_Details
SET [Description] .WRITE(N'-phukien',2,1)
WHERE ProductName='Kem danh rang'
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	Khăn mặt	VV may mặc	99.00
6	Dầu gội	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”.

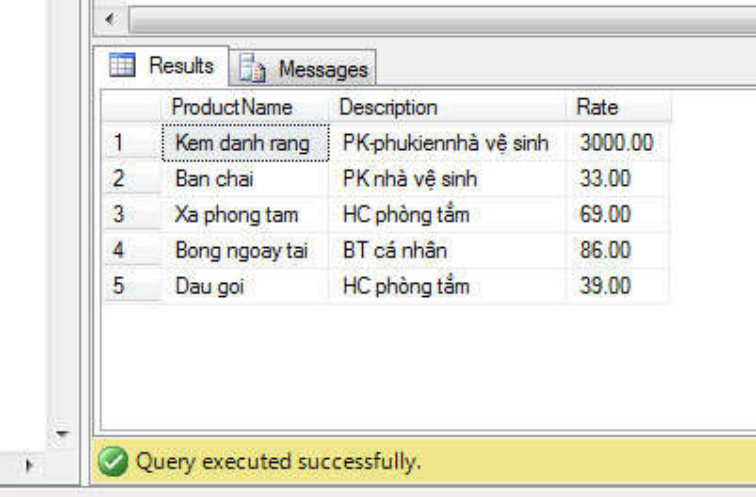
**Bài thực hành 7:** Delete dòng trong VIEW vwProduct\_Full\_Details tại tên sản phẩm là “Khăn mặt”:

**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	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	Dầu gội	HC phòng tắm	39.00

Query executed successfully.

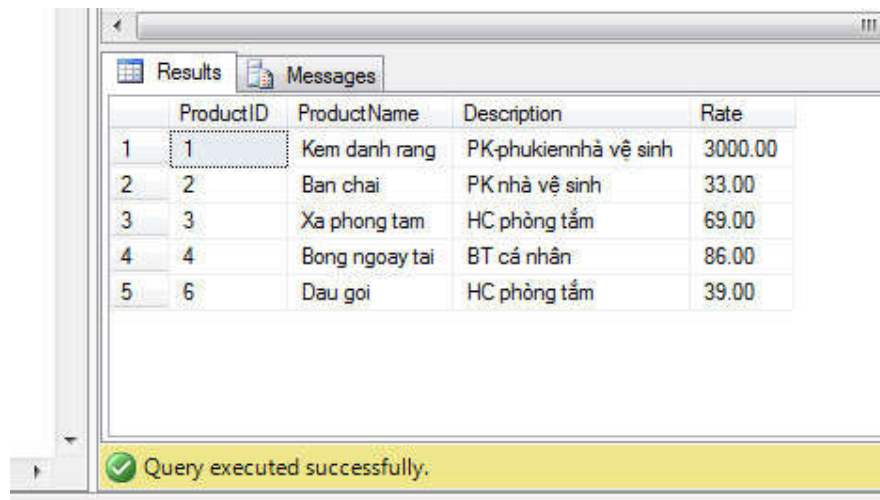
**Bài thực hành 8:** Chỉnh sửa lại View để cập nhật thêm cột ID:

**Bước 1: Gõ lệnh SQL:**

```
-- Chỉnh 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
```

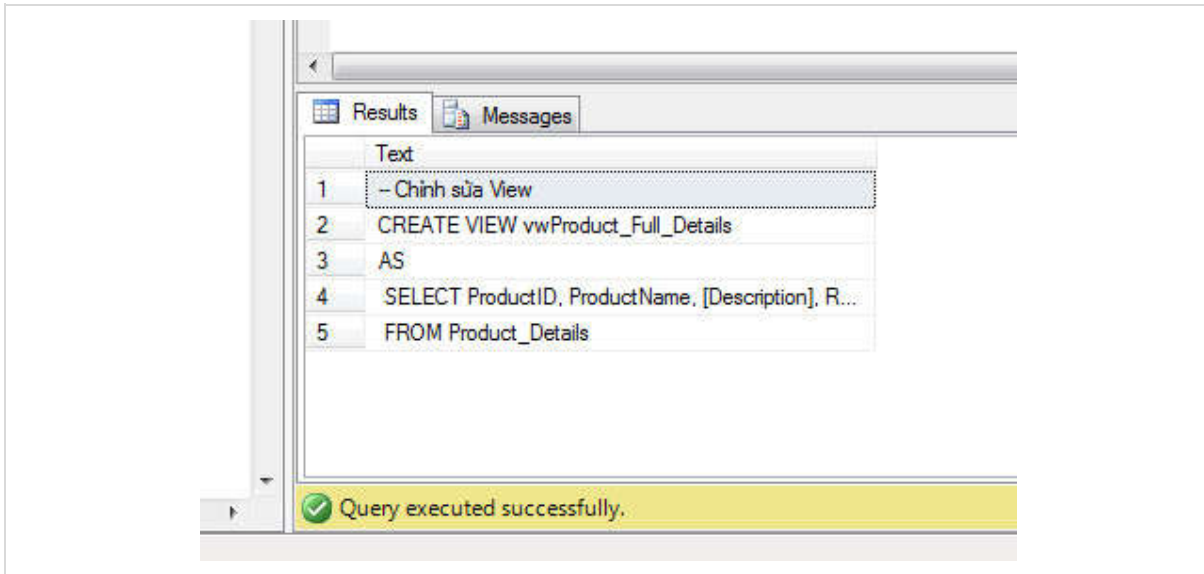


	ProductID	ProductName	Description	Rate
1	1	Kem đánh răng	PK-phukiennhà vệ sinh	3000.00
2	2	Bàn chải	PK nhà vệ sinh	33.00
3	3	Xà phòng tắm	HC phòng tắm	69.00
4	4	Bong ngoay tai	BT cá nhân	86.00
5	6	Dầu gội	HC phòng tắm	39.00

**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:**



**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, ProductNumber, Name, SafetyStockLevel,
ReOrderPoint
FROM Production.Product
WHERE SafetyStockLevel <=1000
WITH CHECK OPTION;
GO
```

Bước 2: Viết lệnh SQL cập nhật View với SafetyStockLevel bằng 2500 tại ProductID bằng 321:

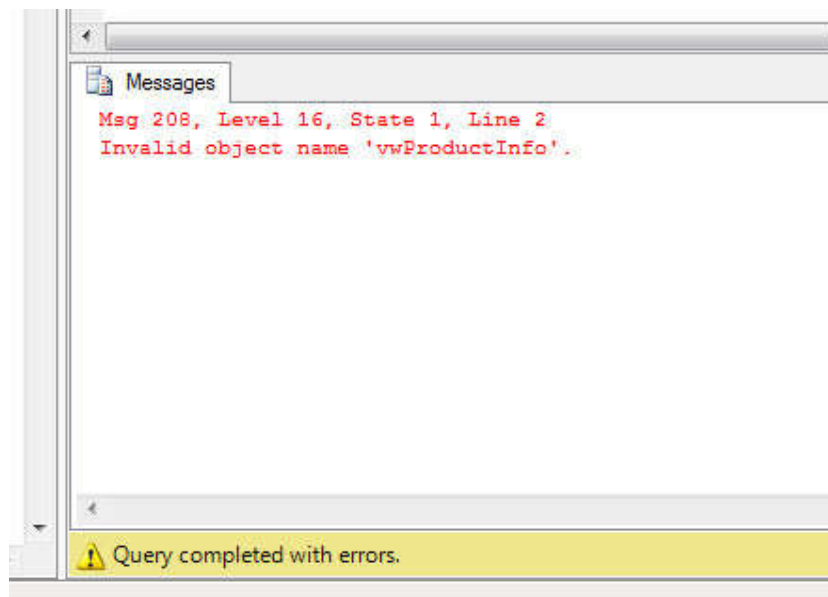
	ProductID	ProductNumber	Name	SafetyStockLevel	ReOrderPoint
4	4	BE-2908	Headset Ball Bearings	800	600
5	316	BL-2036	Blade	800	600
6	317	CA-5965	LL Crankarm	500	375
7	318	CA-6738	ML Crankarm	500	375
8	319	CA-7457	HL Crankarm	500	375
9	320	CB-2903	Chaining Bolts	1000	750
10	321	CN-6137	Chaining Nut	1000	750
11	322	CR-7833	Chaining	1000	750
12	323	CR-9981	Crown Race	1000	750
13	324	CS-2812	Chain Stays	1000	750
14	325	DC-8732	Decal 1	1000	750

Query executed successfully. MINHVUFC-PC

-- Lệnh cập nhật.

```
UPDATE vwProductInfo SET SafetyStockLevel= 2500
WHERE ProductID=321
```

Và lỗi:



Nguyên nhân lỗi là do **CHECK OPTION** kiểm tra câu lệnh nhận thấy nó vi phạm định nghĩa của View trên là chỉ nhận SafetyStockLevel ≤ 1000. Và tất nhiên, nếu câu chuyển 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 **Production.Product** với VIEW

**vwNewProductInfo:**

Bước 1: Gõ lệnh SQL:

```
-- Lệnh ràng buộc SCHEMABINDING
CREATE VIEW vwNewProductInfo
WITH SCHEMABINDING AS
SELECT ProductID, ProductNumber, Name, SafetyStockLevel
FROM Production.Product;
GO
```

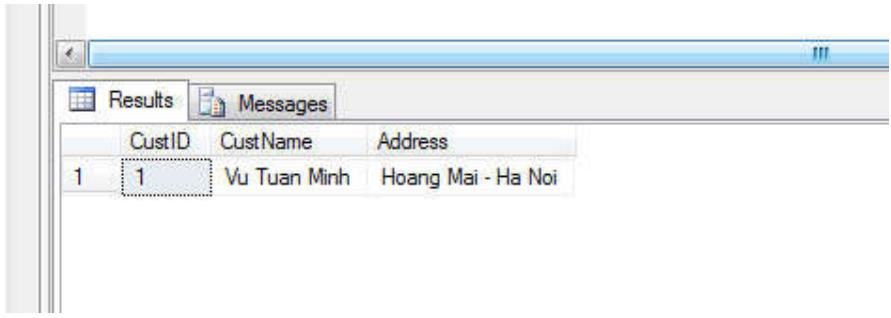
Bước 2: Tạo bảng Customers, sau đó tạo view cho bảng này như sau:

```
-- 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 3: 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
```



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

**Bước 4: Thêm cột vào bảng cơ sở, sau đó truy cập View:**

```
-- Chỉnh sửa bảng
ALTER TABLE Customers ADD Age int
GO
SELECT * FROM Customers
GO
SELECT * FROM vwCustomers
GO
```



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

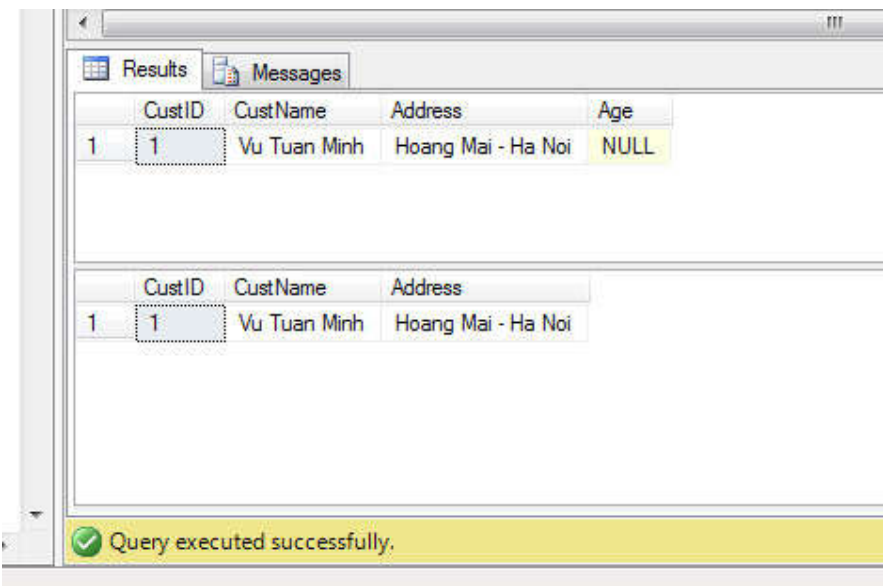
	CustID	CustName	Address
1	1	Vu Tuan Minh	Hoang Mai - Ha Noi

Query executed successfully.

Trên View không được cập nhật thêm cột Age

**Bước 5: Gọi hàm refresh, sau đó truy cập View:**

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



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

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

Query executed successfully.

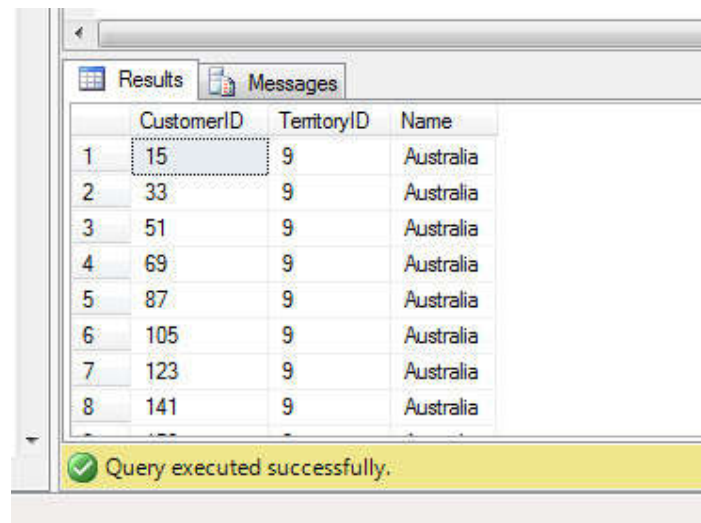
**Bài thực hành 12:** Tạo thủ tục và thực thi:

Bước 1: Gõ lệnh SQL:

```
-- Tạo thủ tục
CREATE PROCEDURE uspGetCustTerritory
AS
SELECT TOP 10
    CustomerID, Customer.TerritoryID, Sales.SalesTerritory.Name
FROM Sales.Customer JOIN Sales.SalesTerritory
ON Sales.Customer.TerritoryID = Sales.SalesTerritory.TerritoryID
```

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

```
-- Thực thi thủ tục
EXEC uspGetCustTerritory
```



	CustomerID	TerritoryID	Name
1	15	9	Australia
2	33	9	Australia
3	51	9	Australia
4	69	9	Australia
5	87	9	Australia
6	105	9	Australia
7	123	9	Australia
8	141	9	Australia

Query executed successfully.

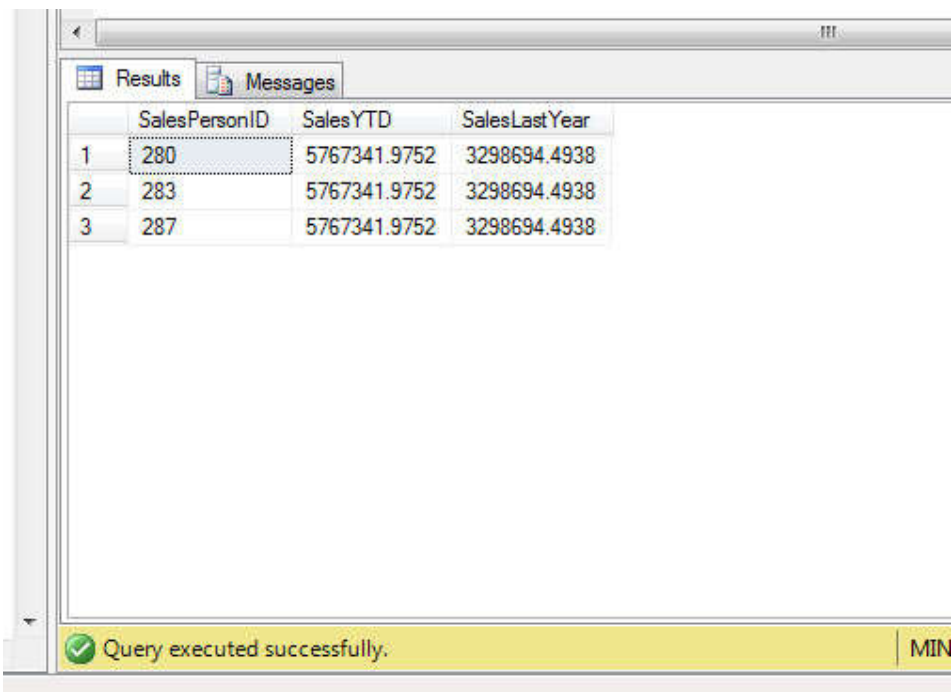
**Bài thực hành 13:** Tạo thủ tục có tham số truyền vào và thực thi:

Bước 1: Gõ lệnh SQL:

```
-- Tạo thủ tục có tham số truyền vào
CREATE PROCEDURE uspGetSales
@territory varchar(40)
AS
SELECT SalesPersonID, B.SalesYTD, B.SalesLastYear
FROM Sales.SalesPerson A
JOIN Sales.SalesTerritory B
ON A.TerritoryID = B.TerritoryID
WHERE B.Name = @territory;
```

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

```
-- Kích hoạt stored procedure
EXEC uspGetSales 'Northwest'
```



The screenshot shows a SQL Server Enterprise Manager window with a 'Results' tab. It displays a table with three columns: SalesPersonID, SalesYTD, and SalesLastYear. There are three rows of data. Below the table, a status bar indicates 'Query executed successfully.' with a green checkmark icon.

	SalesPersonID	SalesYTD	SalesLastYear
1	280	5767341.9752	3298694.4938
2	283	5767341.9752	3298694.4938
3	287	5767341.9752	3298694.4938

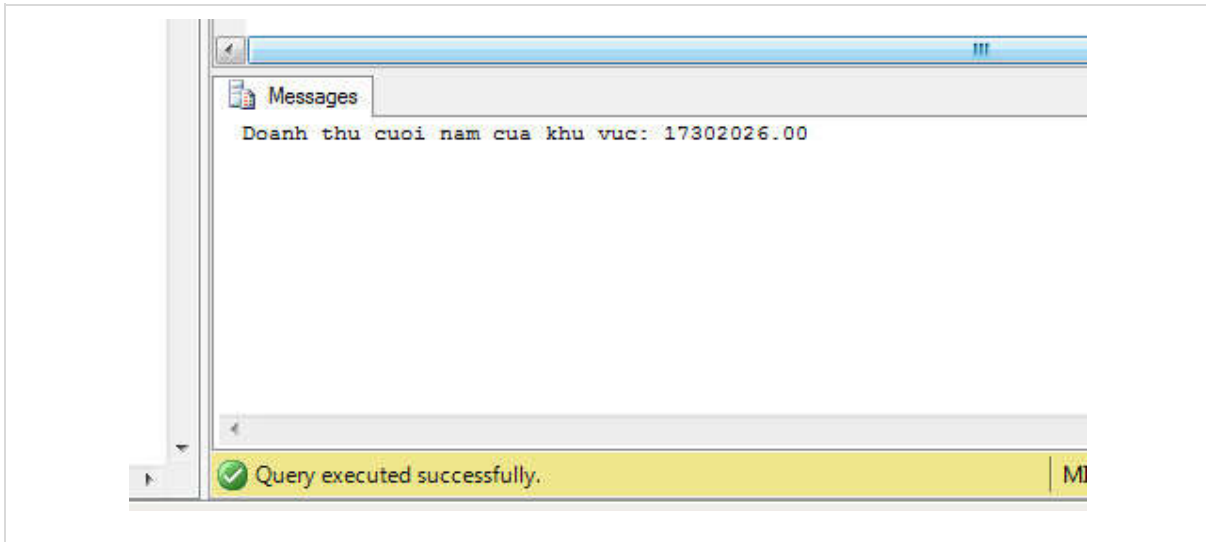
**Bài thực hành 14:** Tạo thủ tục có tham số trả về và thực thi:

**Bước 1:** Gõ lệnh SQL:

```
-- Thủ tục có tham số trả về
CREATE PROCEDURE uspGetTotalSales
@territory varchar(40), @sum int OUTPUT
AS
SELECT @sum= SUM(B.SalesYTD)
FROM Sales.SalesPerson A
JOIN Sales.SalesTerritory B
ON A.TerritoryID = B.TerritoryID
WHERE B.Name = @territory
GO
```

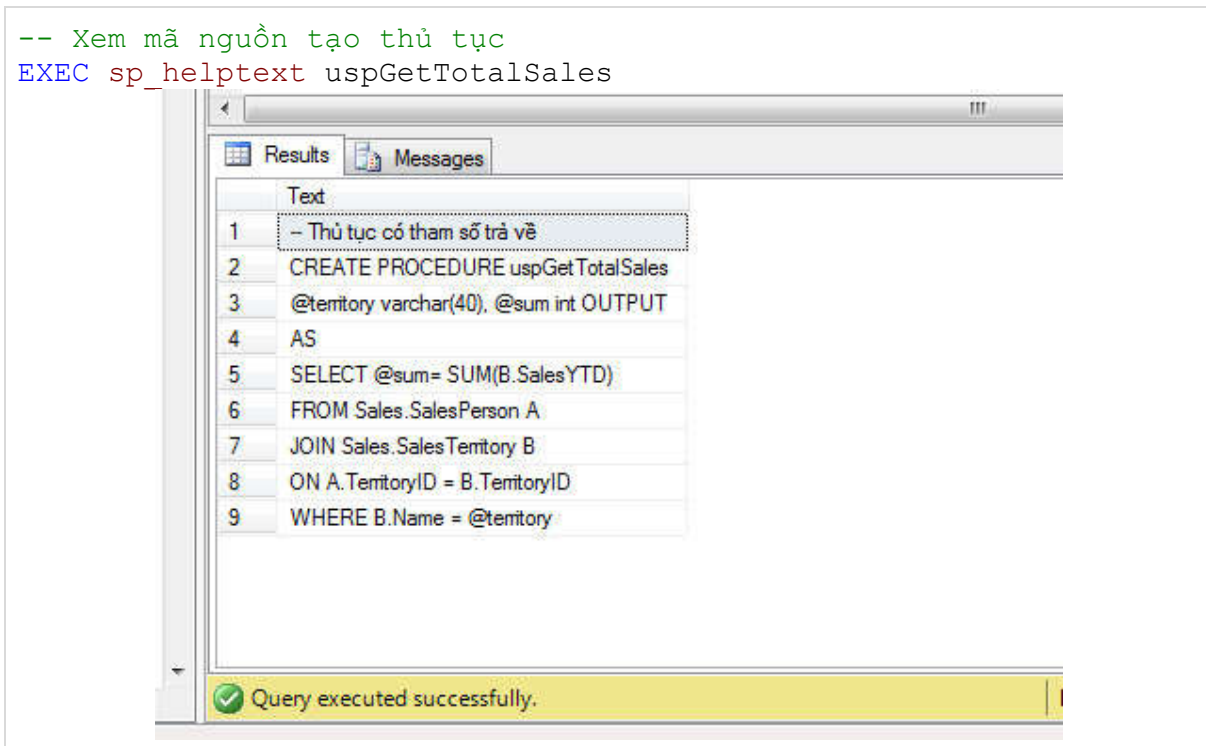
**Bước 2:** Quét dòng mã lệnh trên rồi bấm F5:

```
-- Thực thi thủ tục
DECLARE @sumsales money;
EXEC uspGetTotalSales 'Northwest', @sumsales OUTPUT;
PRINT 'Doanh thu cuối năm của khu vực: ' +
convert(varchar(100),@sumsales);
GO
```

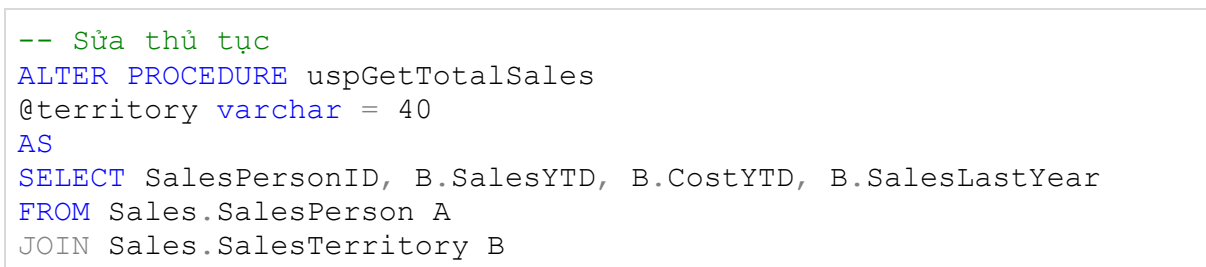


**Bài thực hành 15:** Xem mã nguồn tạo thủ tục:

**Bước 1:** Gõ lệnh SQL:



**Bước 2:** Sửa thủ tục:



```
ON A.TerritoryID = B.TerritoryID
WHERE B.Name = @territory;
GO
```

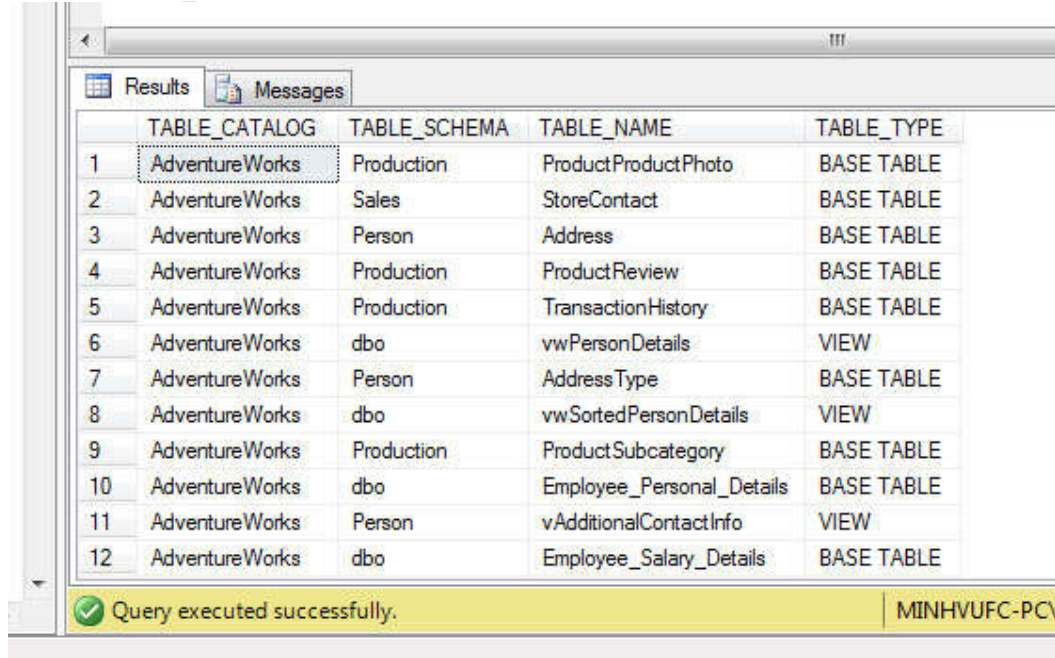
### Bước 3: Xóa thủ tục:

```
-- Xóa thủ tục
DROP PROCEDURE uspGetTotalSales
```

### 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;
```



	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	AdventureWorks	Production	ProductProductPhoto	BASE TABLE
2	AdventureWorks	Sales	StoreContact	BASE TABLE
3	AdventureWorks	Person	Address	BASE TABLE
4	AdventureWorks	Production	ProductReview	BASE TABLE
5	AdventureWorks	Production	TransactionHistory	BASE TABLE
6	AdventureWorks	dbo	vwPersonDetails	VIEW
7	AdventureWorks	Person	AddressType	BASE TABLE
8	AdventureWorks	dbo	vwSortedPersonDetails	VIEW
9	AdventureWorks	Production	ProductSubcategory	BASE TABLE
10	AdventureWorks	dbo	Employee_Personal_Details	BASE TABLE
11	AdventureWorks	Person	vAdditionalContactInfo	VIEW
12	AdventureWorks	dbo	Employee_Salary_Details	BASE TABLE

## 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\_Phlo]**

```
(  
    [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 [SoNhanKhau] kiểu [int] vào bảng [Nha\_tren\_phlo].

### 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\_Phlo].

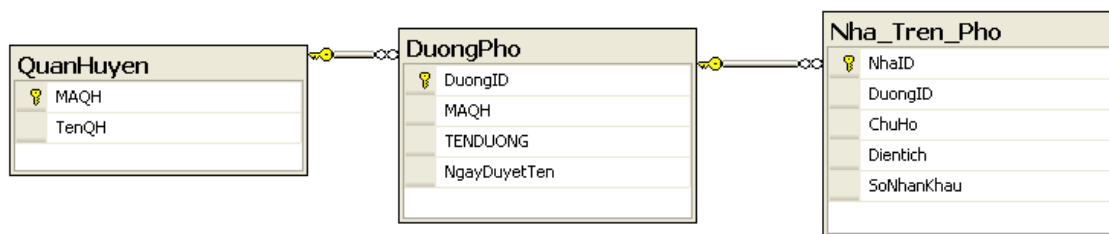
Áp dụng Foreign Key Constraint cho bảng [Nha\_Tren\_Phlo] tham chiếu tới bảng [DuongPho].

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\_Phó] 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.SoNhanKhu

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.SonhanKhu)) 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_SonhanKhu
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 [SonhanKhu] (Ví dụ: order by AVG\_Dientich,AVG\_SonhanKhu).

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

**Question 9:**

Tạo thủ tục có tên [sp\_NgayQuyếtTen\_DuongPho]. Thủ tục này nhận 1 tham số là @NgayDuyet (Date time type):

- Hiển thị thông tin với điều kiện:

NgayQuyếtTen = @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\_NgayQuyếtTen\_DuongPho] với tham số ['30/12/1998'] .

Gợi ý:

Bạn phải khai báo 1 biến

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