

Lab 06

Truy vấn CSDL nâng cao

Phần I – Hướng dẫn thực hành từng bước

1. Mục tiêu

- Hiểu và sử dụng cú pháp truy vấn GROUP BY với mệnh đề WHERE, GROUP BY với mệnh đề WHERE và HAVING.
- Hiểu và sử dụng hàm tổng hợp: AVG, COUNT, MAX, MIN, SUM.
- Hiểu và sử dụng truy vấn con (lệnh SELECT lồng SELECT).
- Thực hiện truy vấn nối giữa các bảng.

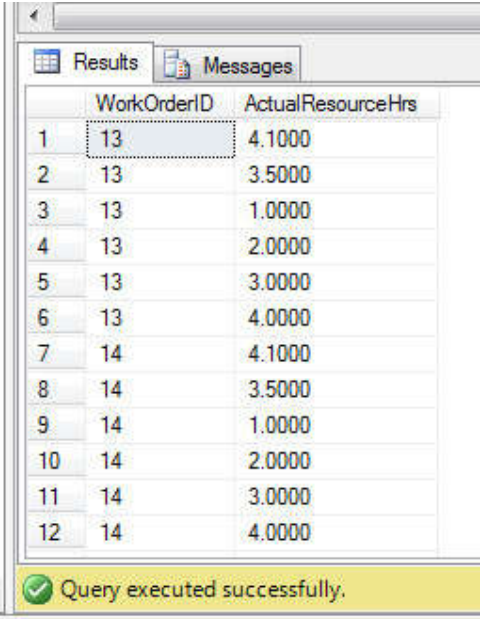
2. Thực hiện

CSDL sử dụng trong bài thực hành này nếu không đề cập gì thì mặc định sử dụng CSDL AdventureWork.

Bài thực hành 1: Viết câu lệnh truy vấn bảng WorkOrderRouting để lấy dữ liệu tổng số giờ cần thiết cho mỗi nguyên công (theo mã ID). Trong cú pháp truy vấn sử dụng hàm SUM để tính tổng. Sau khi lấy được dữ liệu, tiếp tục bổ sung lệnh SQL để lọc (dùng WHERE) nhằm lấy về các nguyên công có ID nhỏ hơn 50.

Bước 1: Mở cửa sổ gõ lệnh query truy vấn bảng lấy số giờ nguyên công:

```
-- Truy vấn tới bảng WorkOrderRouting lấy về số giờ nguyên công
SELECT WorkOrderID, ActualResourceHrs FROM
Production.WorkOrderRouting
```

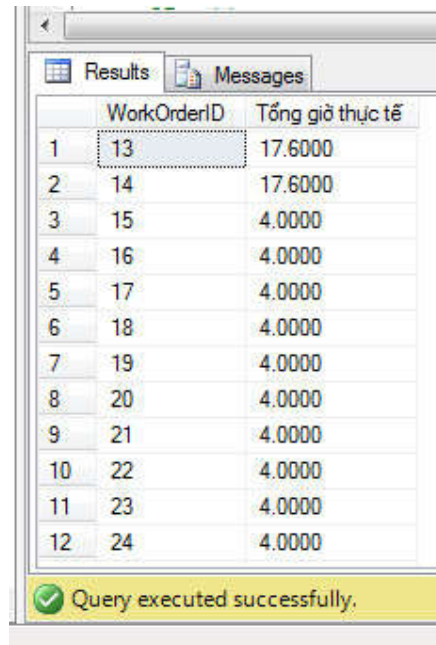


	WorkOrderID	ActualResourceHrs
1	13	4.1000
2	13	3.5000
3	13	1.0000
4	13	2.0000
5	13	3.0000
6	13	4.0000
7	14	4.1000
8	14	3.5000
9	14	1.0000
10	14	2.0000
11	14	3.0000
12	14	4.0000

Query executed successfully.

Bước 2: Gõ mã lệnh truy vấn bảng WorkOrderRouting lấy về tổng thời gian của mỗi nguyên công:

```
-- Lấy về id của nguyên công và tổng thời gian
SELECT
    Production.WorkOrderRouting.WorkOrderID,
    SUM(Production.WorkOrderRouting.ActualResourceHrs) AS 'Tổng
giờ thực tế'
FROM Production.WorkOrderRouting
GROUP BY WorkOrderID -- Lệnh nhóm
/*
    AND: &&
    OR: ||
*/
```

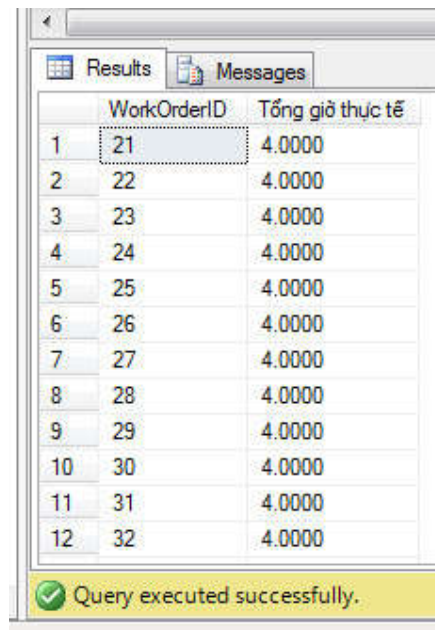


	WorkOrderID	Tổng giờ thực tế
1	13	17.6000
2	14	17.6000
3	15	4.0000
4	16	4.0000
5	17	4.0000
6	18	4.0000
7	19	4.0000
8	20	4.0000
9	21	4.0000
10	22	4.0000
11	23	4.0000
12	24	4.0000

Query executed successfully.

Bước 3: Thêm điều kiện lọc WHERE để lấy bản ghi có WorkOrderID từ có giá trị 20 đến 50:

```
-- Lấy về id của nguyên công và tổng thời gian
SELECT
    Production.WorkOrderRouting.WorkOrderID,
    SUM(Production.WorkOrderRouting.ActualResourceHrs) AS 'Tổng
giờ thực tế'
FROM Production.WorkOrderRouting
WHERE WorkOrderID > 20 AND WorkOrderID <= 50 -- Lệnh lọc
GROUP BY WorkOrderID -- Lệnh nhóm
```



	WorkOrderID	Tổng giờ thực tế
1	21	4.0000
2	22	4.0000
3	23	4.0000
4	24	4.0000
5	25	4.0000
6	26	4.0000
7	27	4.0000
8	28	4.0000
9	29	4.0000
10	30	4.0000
11	31	4.0000
12	32	4.0000

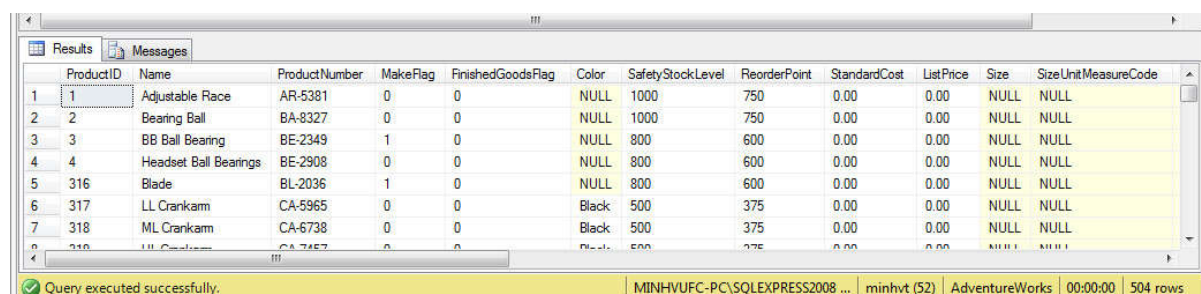
Query executed successfully.

Lệnh lọc WHERE có tác dụng lọc các bản ghi trước khi nhóm chúng lại theo lệnh GROUP BY

Bài thực hành 2: Viết câu lệnh truy vấn bảng Product lấy về giá trị trung bình của ListPrice và nhóm kết quả tìm kiếm theo cột Class, vì trong bảng này có cả dữ liệu NULL trong cột Class nên cần GROUP BY xem xét cả giá trị NULL.

Bước 1: Gõ lệnh SQL truy vấn bảng Product:

```
-- Truy vấn tới bảng Product lấy về giá trung bình của sản phẩm
theo nhóm
SELECT * FROM Production.Product
```



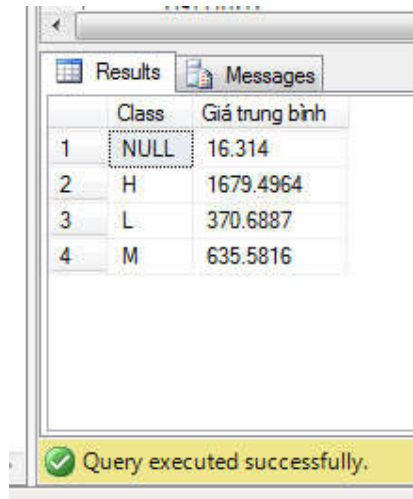
ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint	StandardCost	ListPrice	Size	SizeUnitMeasureCode
1	Adjustable Race	AR-5381	0	0	NULL	1000	750	0.00	0.00	NULL	NULL
2	Bearing Ball	BA-8327	0	0	NULL	1000	750	0.00	0.00	NULL	NULL
3	BB Ball Bearing	BE-2349	1	0	NULL	800	600	0.00	0.00	NULL	NULL
4	Headset Ball Bearings	BE-2908	0	0	NULL	800	600	0.00	0.00	NULL	NULL
5	Blade	BL-2036	1	0	NULL	800	600	0.00	0.00	NULL	NULL
6	LL Crankarm	CA-5965	0	0	Black	500	375	0.00	0.00	NULL	NULL
7	ML Crankarm	CA-6738	0	0	Black	500	375	0.00	0.00	NULL	NULL

Query executed successfully.

Bước 2: Sử dụng lệnh GROUP BY để lấy dữ liệu, chấp nhận thống kê ngay cả khi Class có giá trị là NULL:

```
-- Nhóm các sản phẩm - chấp nhận cả giá trị NULL
SELECT
```

```
Class,
AVG(ListPrice) AS 'Giá trung bình'
FROM Production.Product
GROUP BY Class
```

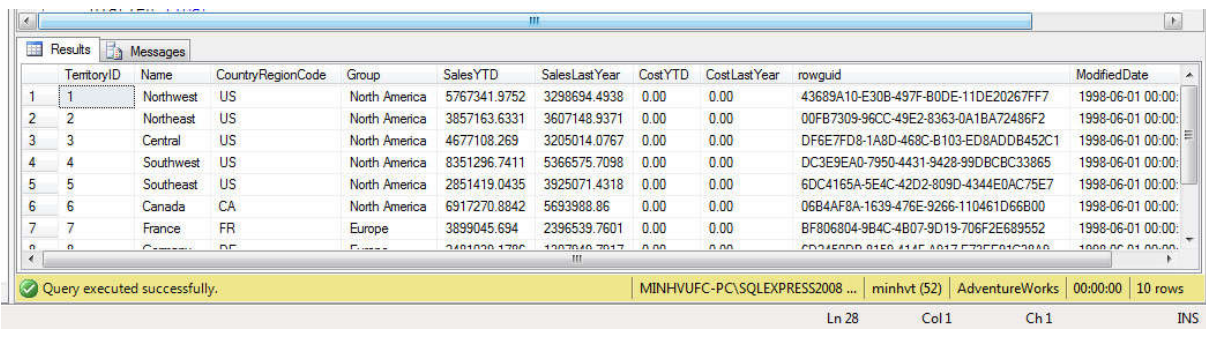


	Class	Giá trung bình
1	NULL	16.314
2	H	1679.4964
3	L	370.6887
4	M	635.5816

Bài thực hành 3: Viết câu lệnh truy vấn bảng SalesTerritory lấy về tổng doanh số của mỗi khu vực bán hàng, thậm chí lấy cả khu vực không THỎA MÃN ĐIỀU KIỆN lọc quy định trong lệnh WHERE – trong tình huống này sử dụng GROUP BY với ALL.

Bước 1: Gõ lệnh SQL:

```
-- Truy vấn tới bảng SalesTerritory - lấy về tổng doanh số theo khu vực địa lý
SELECT * FROM Sales.SalesTerritory
```

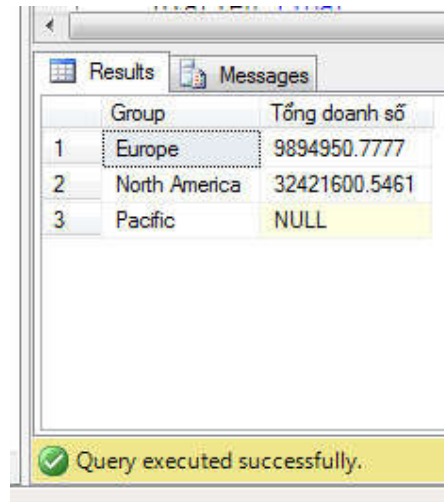


TerritoryID	Name	CountryRegionCode	Group	SalesYTD	SalesLastYear	CostYTD	CostLastYear	rowguid	ModifiedDate
1	Northwest	US	North America	5767341.9752	3298694.4938	0.00	0.00	43689A10-E30B-497F-80DE-11DE20267FF7	1998-06-01 00:00:00
2	Northeast	US	North America	3857163.6331	3607148.9371	0.00	0.00	00FB7309-96CC-49E2-8363-0A1BA72486F2	1998-06-01 00:00:00
3	Central	US	North America	4677108.269	3205014.0767	0.00	0.00	DF6E7FD8-1A8D-468C-B103-ED8ADD8452C1	1998-06-01 00:00:00
4	Southwest	US	North America	8351296.7411	5366575.7098	0.00	0.00	DC3E9EA0-7950-4431-9428-99DBCBC33865	1998-06-01 00:00:00
5	Southeast	US	North America	2851419.0435	3925071.4318	0.00	0.00	6DC4165A-5E4C-42D2-809D-4344E0AC75E7	1998-06-01 00:00:00
6	Canada	CA	North America	6917270.8842	5693988.86	0.00	0.00	06B4AF8A-1639-476E-9266-110461D66B00	1998-06-01 00:00:00
7	France	FR	Europe	3899045.694	2396539.7601	0.00	0.00	BF806804-9B4C-4B07-9D19-706F2E689552	1998-06-01 00:00:00
8	Germany	DE	Europe	3461000.1786	1367040.7617	0.00	0.00	DD44E000-8150-414E-8017-E73FC04C38A0	1998-06-01 00:00:00

Bước 2: Sử dụng lệnh GROUP BY với từ khóa ALL để lấy về cả dữ liệu không thỏa mãn điều kiện lọc WHERE:

```
SELECT
    [Group],
    SUM(SalesYTD) AS 'Tổng doanh số'
FROM Sales.SalesTerritory
WHERE [Group] LIKE 'N%' OR [Group] LIKE 'E%' -- Điều kiện lọc
```

`GROUP BY ALL [Group] --` Lệnh nhóm chấp nhận cả khi điều kiện lọc ko thỏa và gán giá trị NULL



	Group	Tổng doanh số
1	Europe	9894950.7777
2	North America	32421600.5461
3	Pacific	NULL

Bài thực hành 4: Viết câu lệnh truy vấn bảng dữ liệu bán hàng của cửa hàng điện thoại, lấy về các sản phẩm có giá bán lớn hơn \$100 và có tổng doanh thu lớn hơn \$10.000. Cho dữ liệu như sau:

Bảng ThôngKeBanHang		
Cột	Kiểu dữ liệu	Mô tả
sanpham	Nvarchar(128)	Tên sản phẩm
giatien	float	Giá tiền
ngayban	date	Ngày bán

Bước 1: Gõ lệnh SQL tạo bảng:

```
CREATE TABLE ThôngKeBanHang (
    sanpham nvarchar(128),
    giatien float,
    ngayban date
)
```

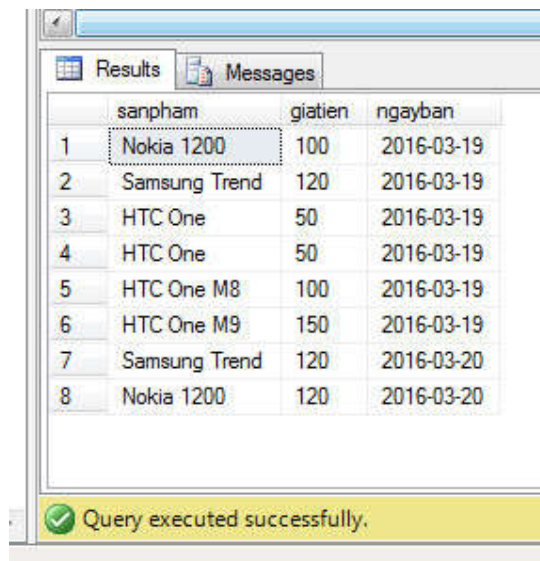
Bước 2: Thêm dữ liệu vào bảng:

```
-- Thêm dữ liệu
INSERT INTO ThôngKeBanHang VALUES ('Nokia 1200', 100, '2016-03-19')
INSERT INTO ThôngKeBanHang VALUES ('Samsung Trend', 120, '2016-
```

```
03-19')
INSERT INTO ThongKeBanHang VALUES ('HTC One', 50, '2016-03-19')
INSERT INTO ThongKeBanHang VALUES ('HTC One', 50, '2016-03-19')
INSERT INTO ThongKeBanHang VALUES ('HTC One M8', 100, '2016-03-19')
INSERT INTO ThongKeBanHang VALUES ('HTC One M9', 150, '2016-03-19')
INSERT INTO ThongKeBanHang VALUES ('Samsung Trend', 120, '2016-03-20')
INSERT INTO ThongKeBanHang VALUES ('Nokia 1200', 120, '2016-03-20')
```

Bước 3: Truy cập dữ liệu bảng:

```
-- Truy vấn
SELECT * FROM ThongKeBanHang
```

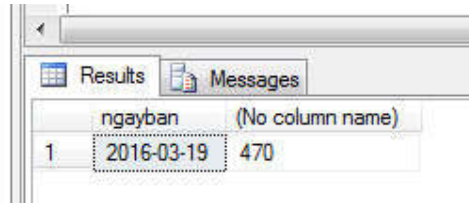


	sanpham	giatien	ngayban
1	Nokia 1200	100	2016-03-19
2	Samsung Trend	120	2016-03-19
3	HTC One	50	2016-03-19
4	HTC One	50	2016-03-19
5	HTC One M8	100	2016-03-19
6	HTC One M9	150	2016-03-19
7	Samsung Trend	120	2016-03-20
8	Nokia 1200	120	2016-03-20

Bước 4: Truy vấn bảng với yêu cầu tính tổng doanh thu trong ngày với những sản phẩm có giá lớn hơn 50, chỉ lọc lấy bản ghi có tổng doanh thu lớn hơn 300:

```
-- Xem tổng doanh thu theo ngày bán hàng
SELECT
    ngayban,
    SUM(giatien)
FROM ThongKeBanHang
WHERE giatien > 50 -- Lệnh lọc bản ghi
GROUP BY ngayban
HAVING SUM(giatien) >= 300 -- Lệnh chỉ lấy về tổng doanh số trong ngày lớn hơn 300$
GO
```

Bước 5: Kết quả truy vấn:



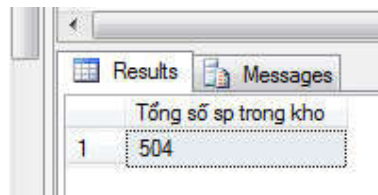
	ngayban	(No column name)
1	2016-03-19	470

Lệnh WHERE có tác dụng lọc từng bản ghi trước khi thực hiện tính toán, lệnh GROUP BY có tác dụng nhóm các bản ghi theo cột nhất định và lệnh HAVING thì có tác dụng lọc dữ liệu cuối cùng sau khi tính toán và nhóm kết quả theo cột.

Bài thực hành 5: Viết câu lệnh ứng dụng các hàm tổng hợp trên CSDL AdventureWork như sau:

Bước 1: Hàm COUNT:

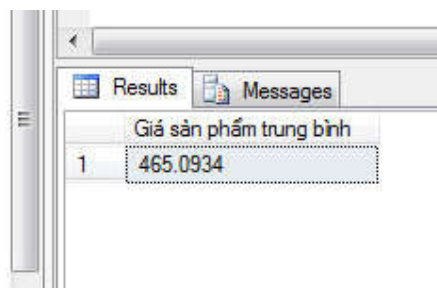
```
USE AdventureWorks
GO
-- Lấy về tổng số sản phẩm có trong kho
SELECT COUNT(*) AS 'Tổng số sp trong kho' FROM Production.Product
```



	Tổng số sp trong kho
1	504

Bước 2: Hàm AVG:

```
-- Tính trung bình giá sản phẩm trong tất cả hóa đơn
SELECT AVG(UnitPrice) AS 'Giá sản phẩm trung bình' FROM
Sales.SalesOrderDetail
```



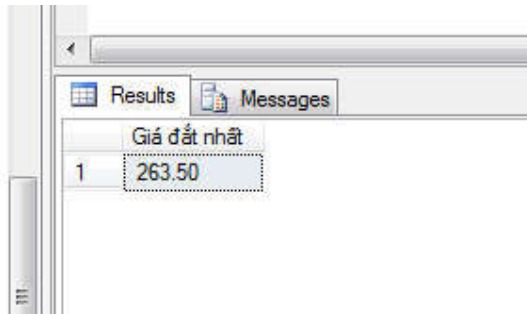
	Giá sản phẩm trung bình
1	465.0934

Bước 3: Hàm Max - Min:

```
USE Northwind
Go
-- Lấy về giá sản phẩm đắt nhất
```



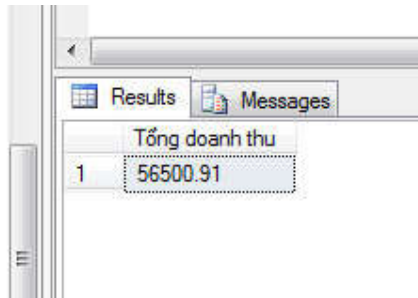
```
SELECT MAX(UnitPrice) AS 'Giá đắt nhất' FROM dbo.Products
```



	Giá đắt nhất
1	263.50

Bước 4: Hàm SUM:

```
-- Tính tổng doanh thu bán hàng
SELECT SUM(UnitPrice) AS 'Tổng doanh thu' FROM [Order Details]
```



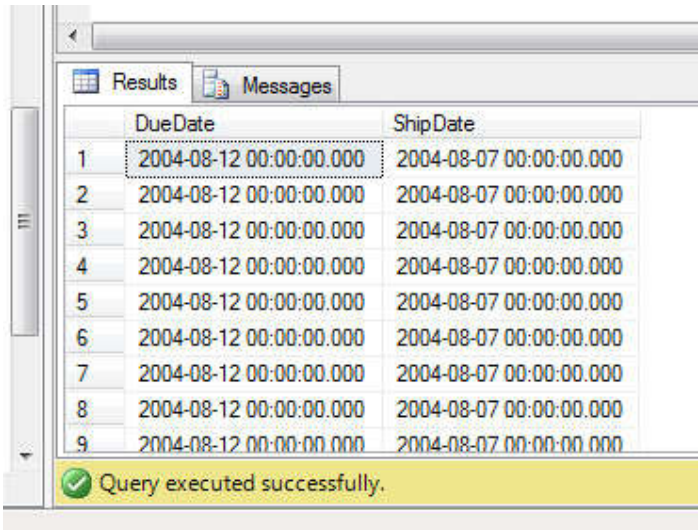
	Tổng doanh thu
1	56500.91

Bài thực hành 6: Viết câu lệnh truy vấn bảng SalesOrderHeader lấy về tất cả thông tin với điều kiện lọc là ngày đặt hàng phải là ngày gần ngày hiện tại nhất – vận dụng khả năng truy vấn con thực hiện như sau:

Bước 1: Gõ lệnh SQL:

```
-- Truy vấn lấy về ngày đặt hàng là ngày cuối cùng trong CSDL
SELECT
    DueDate,
    ShipDate
FROM Sales.SalesOrderHeader
WHERE
-- Đặt điều kiện tìm kiếm là ngày OrderDate là ngày cuối cùng trong csdl
Sales.SalesOrderHeader.OrderDate = (
    -- Lấy về ngày cuối cùng trong bảng SalesOrderHeader
    SELECT MAX(OrderDate) FROM Sales.SalesOrderHeader
)
```

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



	DueDate	ShipDate
1	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
2	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
3	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
4	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
5	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
6	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
7	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
8	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000
9	2004-08-12 00:00:00.000	2004-08-07 00:00:00.000

Query executed successfully.

Bài thực hành 7: Viết câu lệnh truy vấn bảng Person lấy về những nhân viên mà chức danh của họ quy định trong bảng Employee là “Research and Development Manager”:

Lưu ý: mã lệnh này thực hiện trên csdl AdventureWork phiên bản 2008, trên csdl phiên bản 2012 có một chút thay đổi tên bảng nhưng về nguyên lý thì hoàn toàn giống nhau.

Mã lệnh trên 2012:

```
-- Mã trên 2012
SELECT FirstName, LastName FROM Person.Person
WHERE Person.Person.BusinessEntityID IN (SELECT BusinessEntityID
FROM HumanResources.Employee WHERE JobTitle = 'Research and
Development
Manager');
```

Bước 1: Gõ lệnh SQL:

```
SELECT * FROM Person.Contact
GO

SELECT * FROM HumanResources.Employee
WHERE Title LIKE '%Manager%'
GO
```

Person.Contact

ContactID	NameStyle	Title	FirstName	LastName	Initials	PhoneNumber	EmailAddress	EmailPromotion	Phone	PasswordHash
1	0	Mr.	Gustavo	Abel	NULL	398-555-0132	gustavo0@adventure-works.com	2	747-555-0132	F295DC2A87C7F
2	0	Ms.	Catherine	R.	Abel	747-555-0171	catherine0@adventure-works.com	1	747-555-0171	19712A42FC40F
3	0	Ms.	Kim	Abercrombie	NULL	334-555-0137	kim2@adventure-works.com	0	334-555-0137	CBCE65769648C
4	0	Sr.	Humberto	Acevedo	NULL	599-555-0127	humberto0@adventure-works.com	2	599-555-0127	69CF91B5628FA
5	0	Sra.	Pilar	Ackerman	NULL	1 (11) 500 555-0132	pilar1@adventure-works.com	0	1 (11) 500 555-0132	F57E03FEA2FD0
6	0	Ms.	Frances	B.	Adams	991-555-0183	frances0@adventure-works.com	1	991-555-0183	CB11BA26C919
7	0	Ms.	Margaret	J.	Smith	959-555-0151	margaret0@adventure-works.com	0	959-555-0151	628463129E609
8	0	Ms.	Carla	J.	Adams	107-555-0138	carla0@adventure-works.com	0	107-555-0138	E1F4078A186BC

HumanResources.Employee

EmployeeID	NationalIDNumber	ContactID	LoginID	ManagerID	Title	BirthDate	MaritalStatus	Gender	HireDate
1	509647174	1002	adventure-works\roberto0	12	Engineering Manager	1964-12-13 00:00:00.000	M	M	1997-
2	24756624	1028	adventure-works\peter0	109	Marketing Manager	1965-04-19 00:00:00.000	S	M	1998-
3	277173473	1231	adventure-works\paula0	148	Production Control Manager	1972-12-04 00:00:00.000	M	M	1999-
4	535145551	1242	adventure-works\jean0	140	Human Resources Manager	1966-03-14 00:00:00.000	M	F	1999-
5	441044382	1253	adventure-works\wendy0	109	Information Services Manager	1966-01-13 00:00:00.000	S	F	1999-
6	121491555	1243	adventure-works\zainal0	140	Finance Manager	1974-11-12 00:00:00.000	S	F	1999-
7	345106466	1265	adventure-works\david6	200	Accounts Manager	1973-08-08 00:00:00.000	M	M	1999-
8	30845	1245	adventure-works\david6	140	Accounts Manager	1973-08-08 00:00:00.000	M	M	1999-

Hình trên dễ dàng nhận thấy giữa bảng Contact và Employee có mối liên hệ là ContactID.

Bước 2: Yêu cầu truy vấn là lấy ra tên của người mà trong bảng Employee giữ chức danh

Research and Development Manager:

```

/* Viết lệnh truy vấn bảng Person lấy về nhân viên mà chức danh
   quy định
   trong bảng Employee là Research and Development Manager */
SELECT Person.Contact.FirstName, Person.Contact.LastName
FROM Person.Contact
WHERE Person.Contact.ContactID IN (
    SELECT ContactID
    FROM HumanResources.Employee
    WHERE Title = 'Research and Development Manager'
)

```

Bước 3: Quét dòng mã lệnh trên rồi bấm F5:

	FirstName	LastName
1	Dylan	Miller
2	Michael	Raheem

Kết quả trả về tìm được Miller và Raheem.

Nguyên lý hoạt động của câu truy vấn trên là:

1. Lệnh SELECT ở trong dấu ngoặc sẽ thực hiện trước để lấy về danh sách ContactID thỏa mãn yêu cầu là “Research and Development Manager”.

2. Lệnh WHERE bên ngoài liên kết sẽ thực hiện lọc ContactID trên bảng Contact để chỉ lấy về bản ghi mà thỏa mã ContactID nằm trong tập hợp mà lệnh SELECT trong dấu ngoặc trả về (lệnh IN).
3. Lệnh SELECT ngoài cùng có trách nhiệm lấy dữ liệu và hiển thị ra màn hình.

Bài thực hành 8: Viết câu lệnh truy vấn bằng Person lấy về những người đến từ Canada:

Bước 1: Gõ lệnh SQL:

```
SELECT * FROM Person.Contact
GO
SELECT * FROM Sales.SalesPerson
GO
SELECT * FROM Sales.SalesTerritory
```

The screenshot shows the results of three SQL queries. The first query returns data from Person.Contact, the second from Sales.SalesPerson, and the third from Sales.SalesTerritory. Red boxes highlight the ContactID, SalesPersonID, and TerritoryID columns respectively. Arrows point from the SalesPersonID column in the second table to the ContactID column in the first table, and from the TerritoryID column in the third table to the SalesPersonID column in the second table, illustrating the relationships between the tables.

Bảng Sales.SalesPerson có cột SalesPersonID là ContactID ở bảng Person.Contact.

Trong SalesPerson có TerritoryID là mã vùng lãnh thổ quy định ở bảng SaleTerritory.

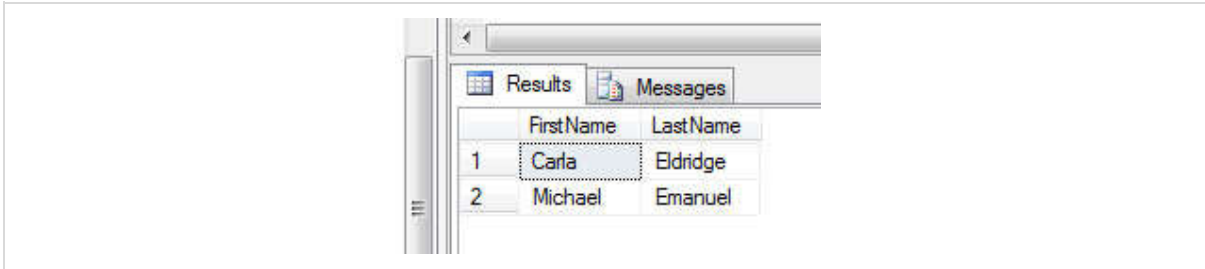
Như vậy để lấy về người bán hàng thuộc vùng lãnh thổ là Canada thì thực hiện truy vấn ở bước 2.

Bước 2: Lấy về người bán ở Canada:

```
/*
Truy vấn lồng
*/
SELECT Person.Contact.FirstName, Person.Contact.LastName
FROM Person.Contact
WHERE Person.Contact.ContactID IN (
```

```
SELECT Sales.SalesPerson.SalesPersonID
FROM Sales.SalesPerson
WHERE TerritoryID IN (
    SELECT TerritoryID
    FROM Sales.SalesTerritory
    WHERE Sales.SalesTerritory.Name = 'Canada'
)
```

Bước 3: Quét dòng mã lệnh trên rồi bấm F5:

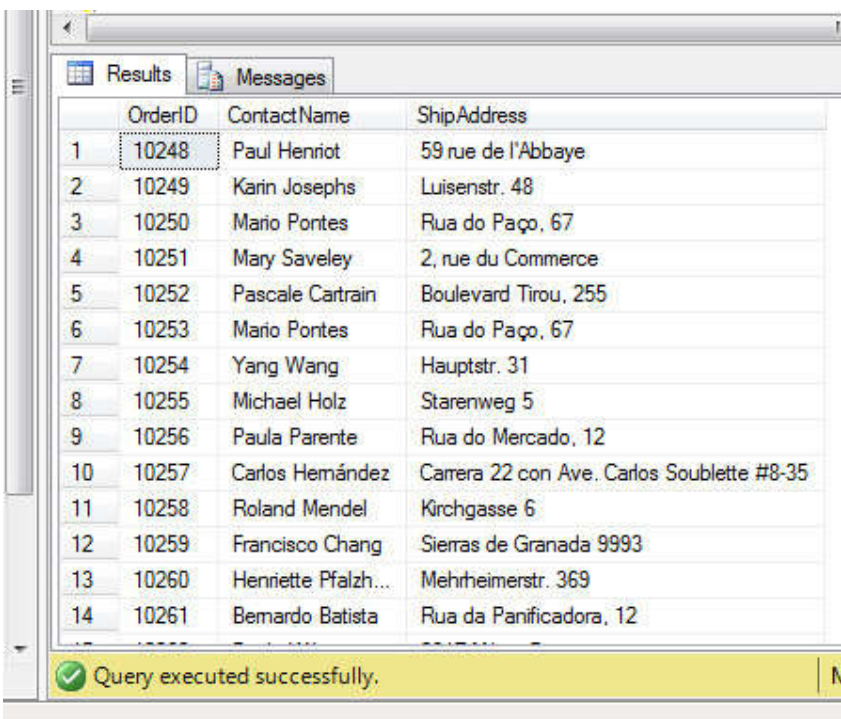


Bài thực hành 9: Viết câu lệnh hiển thị tập hợp dữ liệu mã hóa đơn và tên khách hàng đã thực hiện hóa đơn (trong bảng csdl Northwind):

Bước 1: Gõ lệnh SQL:

```
USE Northwind
GO
-- Truy vấn bảng hóa đơn và lấy về cả thông tin khách hàng
SELECT
    Orders.OrderID,
    Customers.ContactName,
    Orders.ShipAddress
FROM Orders, Customers
WHERE Orders.CustomerID = Customers.CustomerID
```

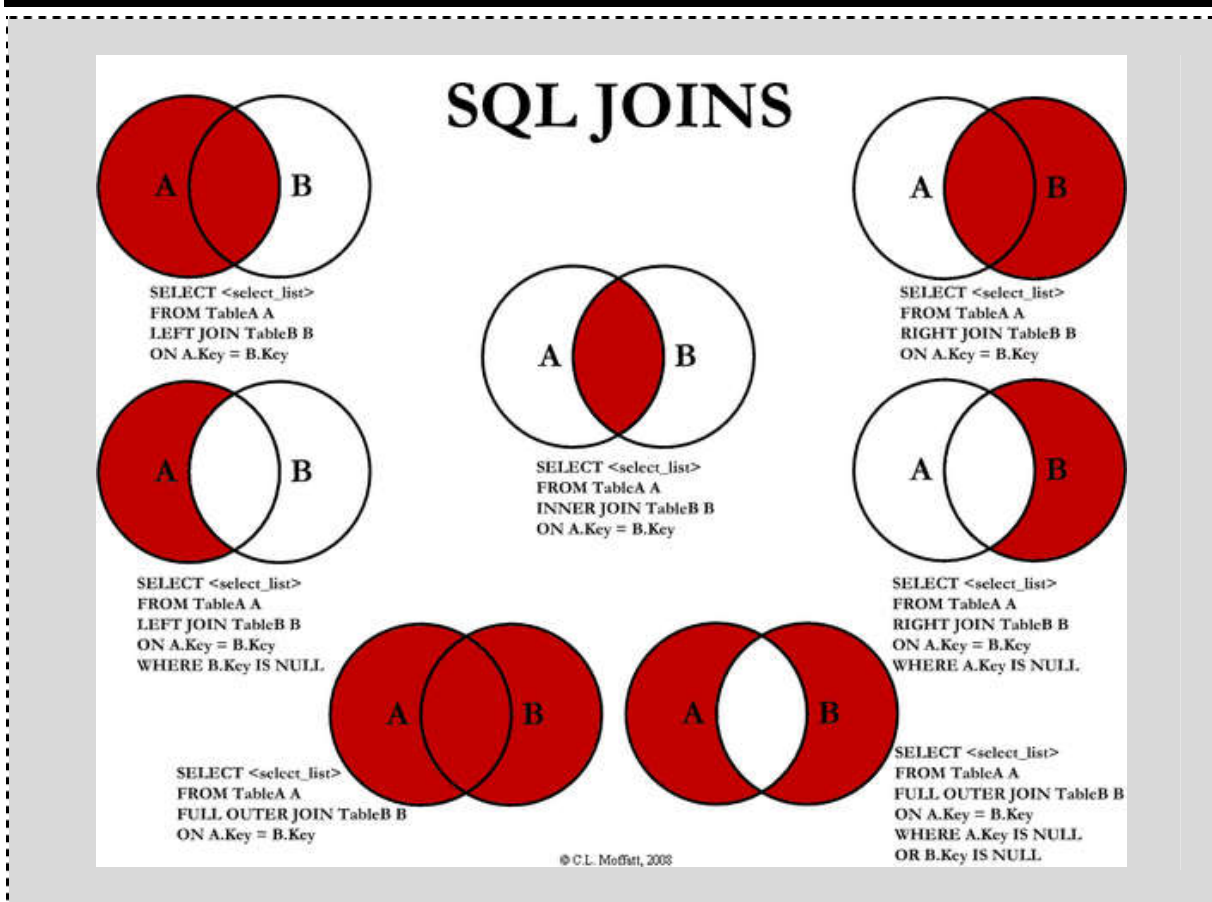
Bước 2: Quét dòng mã lệnh trên rồi bấm F5:



	OrderID	ContactName	ShipAddress
1	10248	Paul Henriot	59 rue de l'Abbaye
2	10249	Karin Josephs	Luisenstr. 48
3	10250	Mario Pontes	Rua do Paço, 67
4	10251	Mary Saveley	2, rue du Commerce
5	10252	Pascale Cartrain	Boulevard Tirou, 255
6	10253	Mario Pontes	Rua do Paço, 67
7	10254	Yang Wang	Hauptstr. 31
8	10255	Michael Holz	Starenweg 5
9	10256	Paula Parente	Rua do Mercado, 12
10	10257	Carlos Hernández	Carrera 22 con Ave. Carlos Soublette #8-35
11	10258	Roland Mendel	Kirchgasse 6
12	10259	Francisco Chang	Sierras de Granada 9993
13	10260	Henriette Pfalzheim	Mehrheimerstr. 369
14	10261	Bernardo Batista	Rua da Panificadora, 12

Query executed successfully.

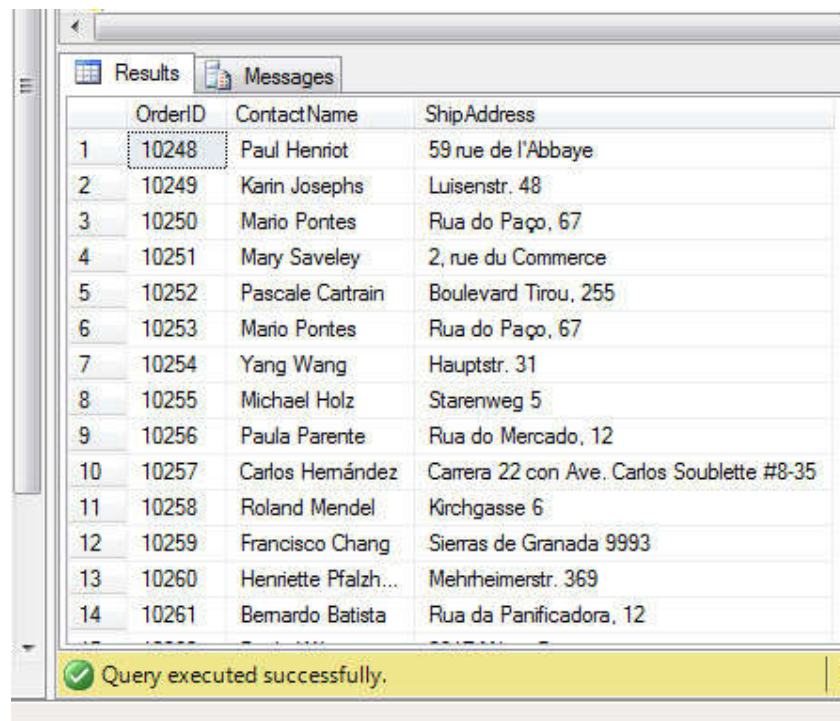
Bài thực hành 10: Viết câu lệnh ứng dụng truy vấn nối INNER, LEFT (tương tự với OUTER và RIGHT nên học viên có thể tự làm – xem chi tiết hình ảnh bên dưới):



Bước 1: Gõ lệnh SQL – INNER JOIN:

```
USE Northwind
GO
-- Truy vấn bảng hóa đơn và lấy về cả thông tin khách hàng
SELECT
    Orders.OrderID,
    Customers.ContactName,
    Orders.ShipAddress
FROM Orders INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID
```

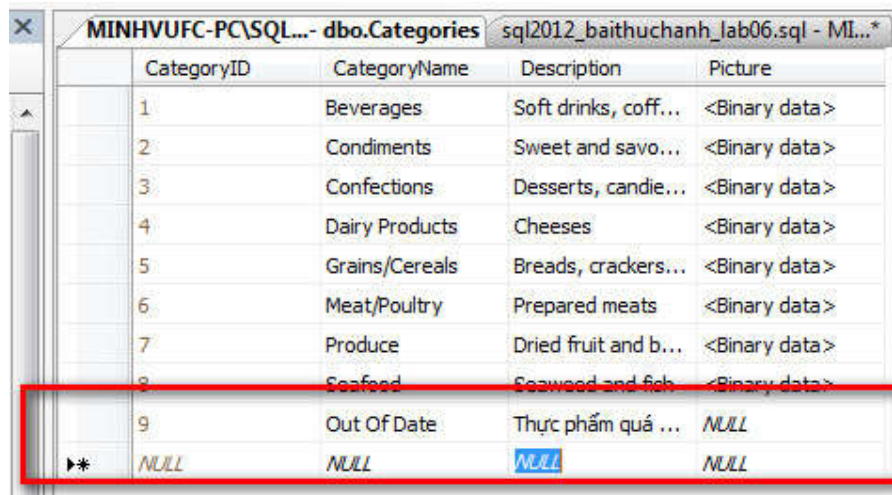
Bước 2: Quét dòng mã lệnh trên rồi bấm F5:



	OrderID	ContactName	ShipAddress
1	10248	Paul Henriot	59 rue de l'Abbaye
2	10249	Karin Josephs	Luisenstr. 48
3	10250	Mario Pontes	Rua do Paço, 67
4	10251	Mary Saveley	2, rue du Commerce
5	10252	Pascale Cartrain	Boulevard Tirou, 255
6	10253	Mario Pontes	Rua do Paço, 67
7	10254	Yang Wang	Hauptstr. 31
8	10255	Michael Holz	Starenweg 5
9	10256	Paula Parente	Rua do Mercado, 12
10	10257	Carlos Hernández	Carrera 22 con Ave. Carlos Soublette #8-35
11	10258	Roland Mendel	Kirchgasse 6
12	10259	Francisco Chang	Sierras de Granada 9993
13	10260	Henriette Pfalzheim	Mehrheimerstr. 369
14	10261	Bernardo Batista	Rua da Panificadora, 12

Query executed successfully.

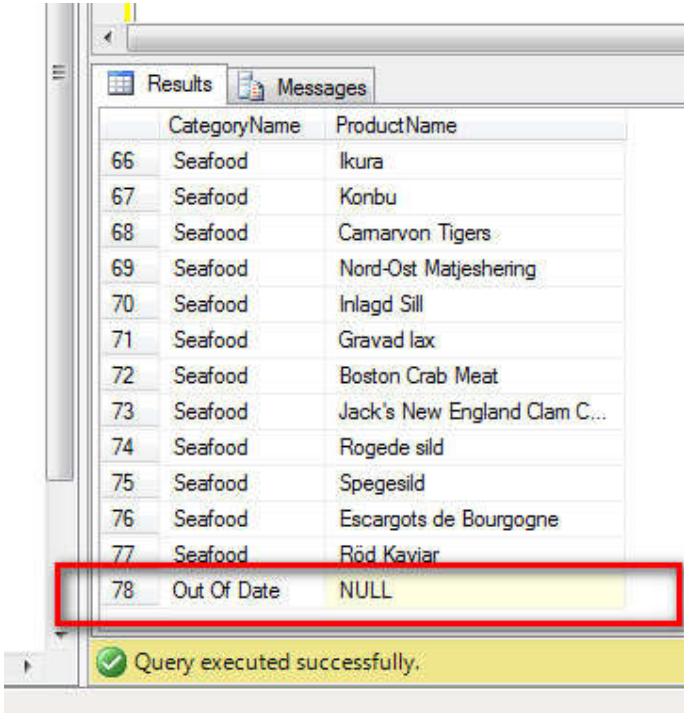
Bước 3: Gõ lệnh SQL – LEFT JOIN:



CategoryID	CategoryName	Description	Picture
1	Beverages	Soft drinks, coff...	<Binary data>
2	Condiments	Sweet and savo...	<Binary data>
3	Confections	Desserts, candie...	<Binary data>
4	Dairy Products	Cheeses	<Binary data>
5	Grains/Cereals	Breads, crackers...	<Binary data>
6	Meat/Poultry	Prepared meats	<Binary data>
7	Produce	Dried fruit and b...	<Binary data>
8	Seafood	Seaweed and fish...	<Binary data>
9	Out Of Date	Thực phẩm quá ...	NULL
NULL	NULL	NULL	NULL

```
-- Left join
SELECT CategoryName, ProductName
FROM Categories LEFT JOIN Products
ON Categories.CategoryID = Products.CategoryID
```

Bước 4: Quét dòng mã lệnh trên rồi bấm F5:



	CategoryName	ProductName
66	Seafood	Ikura
67	Seafood	Konbu
68	Seafood	Camaron Tigers
69	Seafood	Nord-Ost Matjeshering
70	Seafood	Inlagd Sill
71	Seafood	Gravad lax
72	Seafood	Boston Crab Meat
73	Seafood	Jack's New England Clam C...
74	Seafood	Rogede sild
75	Seafood	Spegesild
76	Seafood	Escargots de Bourgogne
77	Seafood	Röd Kaviar
78	Out Of Date	NULL

Query executed successfully.

Bài thực hành 11: Viết câu lệnh kết hợp dữ liệu với UNION, INTERSECT và EXCEPT:

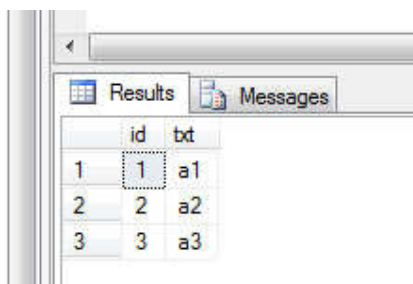
Bước 1: UNION - gõ lệnh SQL:

```
-- Tạo bảng
CREATE TABLE #test1(id INT, txt VARCHAR(10) )
CREATE TABLE #test2(id INT, txt VARCHAR(10) )

GO
INSERT #test1 VALUES(1, 'a1')
INSERT #test1 VALUES(2, 'a2')

INSERT #test2 VALUES(1, 'a1') -- trùng với một bản ghi của #test1
INSERT #test2 VALUES(3, 'a3')

-- UNION
SELECT * FROM #test1
UNION
SELECT * FROM #test2
```

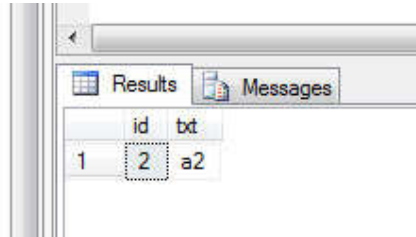


	id	txt
1	1	a1
2	2	a2
3	3	a3

Lệnh UNION có tác dụng hợp 2 kết quả truy vấn thành một

Bước 2: EXCEPT – gõ lệnh SQL:

```
-- EXCEPT
SELECT * FROM #test1
EXCEPT
SELECT * FROM #test2
```

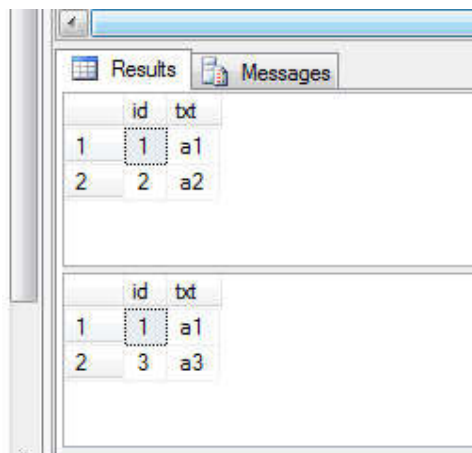


id	txt
1	a2

Lệnh EXCEPT có tác dụng lấy bản ghi có ở truy vấn thứ nhất mà không tồn tại ở truy vấn thứ 2 (điều kiện cấu trúc giống hệt nhau).

Bước 3: INTERSECT – gõ lệnh SQL:

```
-- INTERSECT
SELECT * FROM #test1
INTERSECT
SELECT * FROM #test2
```



id	txt
1	a1
2	a2

id	txt
1	a1
2	a3

Bài thực hành 12: Trong csdl Northwind, bảng Order có chứa thông tin nhân viên giao hàng và mã hình thức giao (chỉ gồm giá trị: 1, 2, hoặc 3). Viết câu lệnh lấy về thông tin của nhân viên giao hàng và tính xem mỗi nhân viên đã thực hiện hình thức giao hàng nào bao nhiêu lần, tổng số lần giao hàng. Áp dụng lệnh PIVOT:

Bước 1: Gõ lệnh SQL:

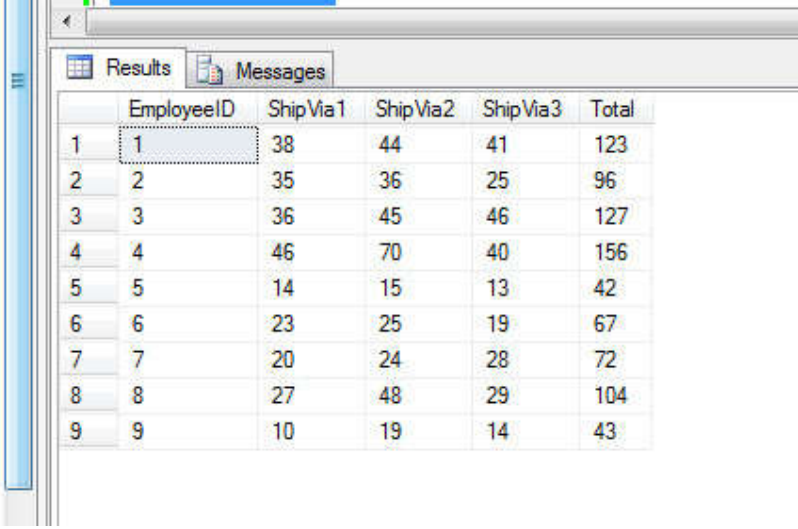
```
-- Lệnh PIVOT để đảo hàng thành cột
SELECT EmployeeID,
```

```

[1] ShipVia1,
[2] ShipVia2,
[3] ShipVia3,
[1] + [2] + [3] Total
FROM (
SELECT EmployeeID,
       ShipVia,
       OrderID
FROM   dbo.Orders
) SourceTable
PIVOT(COUNT(OrderID) FOR ShipVia IN ([1], [2], [3])) PivotTable
ORDER BY EmployeeID

```

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



	EmployeeID	ShipVia1	ShipVia2	ShipVia3	Total
1	1	38	44	41	123
2	2	35	36	25	96
3	3	36	45	46	127
4	4	46	70	40	156
5	5	14	15	13	42
6	6	23	25	19	67
7	7	20	24	28	72
8	8	27	48	29	104
9	9	10	19	14	43

Phần II - Bài tập tự làm

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

Question 1:

- Hãy tạo database có tên là Travel_Management
- Chọn database vừa tạo để làm việc

Question 2:

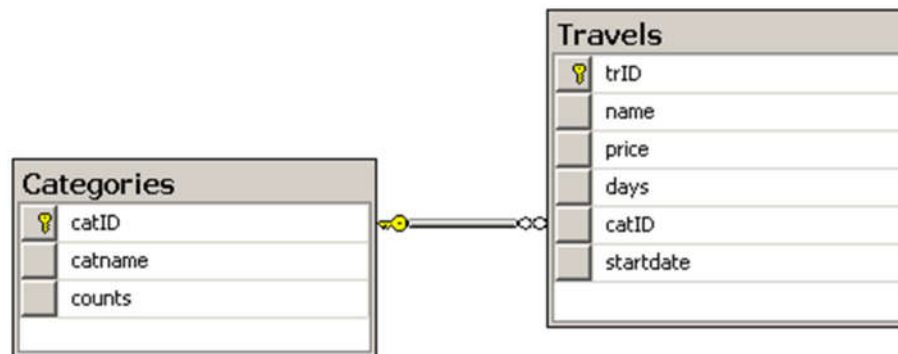
Hãy tạo các bảng với cấu trúc như sau:

Bảng Travels		
Cột	trID INT	NOT NULL
	name VARCHAR(100)	NOT NULL
	price FLOAT	NULL

	days INT	NOT NULL
	catID INT	NOT NULL
	startdate DATETIME	NULL
Bảng Categories		
Cột	catID INT	NOT NULL
	catname VARCHAR(100)	NOT NULL
	counts INT	NULL

Question 3:

1. Hãy tạo **Primary Key** trên cột “**trID**” cho bảng **Travels** .
2. Hãy tạo **Primary Key** trên cột “**catID**” cho bảng **Categories**.
3. Hãy tạo khóa ngoài **Foreign Key Constraint** cho bảng **Travels** tham chiếu đến bảng **Categories** với cột chung là “**catID**”.



Question 4:

1. Hãy tạo một **Check Constraint** trên cột **days** cho bảng **Travels** nằm trong khoảng từ **0** và **15**.
2. Hãy tạo một **Unique Constraint** trên cột **name** cho bảng **Travels**.
3. Hãy tạo một **Default Constraint** trên cột **startdate** cho bảng **Travels** với giá trị default bằng ngày giờ hiện tại của hệ thống khi thêm dữ liệu vào bảng (*gợi ý: sử dụng hàm GETDATE() để lấy ngày giờ hiện tại của hệ thống*).

Question 5:

Hãy chèn dữ liệu cho 2 bảng sau:

Bảng Categories			
	catID	catname	counts
1	100	Beaches	NULL
2	200	Family Travel	NULL
3	300	Food and Drink	NULL
4	400	Skiing	NULL

Bảng Travels						
	trID	name	price	days	catID	startdate
1	10	Manele Bay, Hawaii	200	2	100	2011-04-30 00:00:00.000
2	11	Hilton Waikoloa Village	250	4	200	2011-04-16 00:00:00.000
3	12	Clearwater Beach, Florida	300	7	100	2011-02-11 00:00:00.000
4	13	Sandwich Paradise	180	2	300	2011-01-10 00:00:00.000
5	14	Cape May, New Jersey	380	4	100	2011-01-18 00:00:00.000

Question 6:

Hãy sử dụng lệnh **Update** để tăng giá (price) 10% cho tất cả các dòng trong bảng **Travels** mà số ngày (days) > 5 ngày và thuộc loại (Categories) “*Food and Drink*”.

Question 7:

Hãy sử dụng lệnh **Select** , mệnh đề **GROUP**, **JOIN** để lấy ra thông tin thống kê số lượng (Quantity) chuyến du lịch (travels) theo từng phân loại (Categories), kết quả phải được sắp xếp theo số lượng giảm dần.

Kết quả trả về như sau khi thực thi :

Results			
	catID	Category	Quantity
1	100	Beaches	3
2	200	Family Travel	1
3	300	Food and Drink	1

Question 8:

Căn cứ trên kết quả của câu hỏi 7 ở trên hãy viết lệnh update dữ liệu cột **counts** của bảng **Categories** bằng tổng số lượng chuyến du lịch (travels) trong bảng **Travels** cho từng mục **catname** (Yêu cầu học viên phải sử dụng lệnh **INSERT** kết hợp **GROUP** và **JOIN** và hàm **COUNT** để thực thi)