

Lab 06

Truy vấn CSDL nâng cao

Phần I – Hướng dẫn thực hành từng bước

1. Mục tiêu

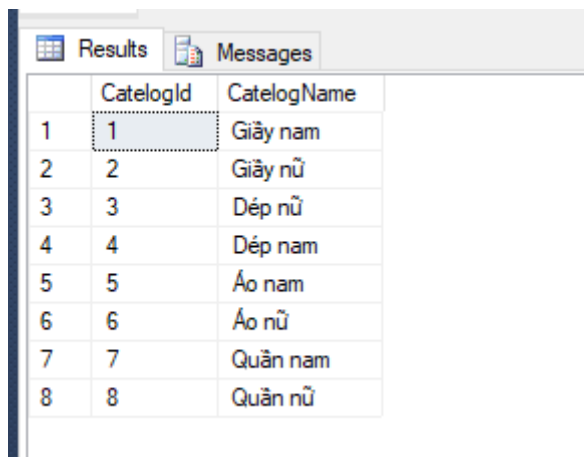
- Hiểu và sử dụng cú pháp truy vấn GROUP BY với mệnh đề WHERE, GROUP BY với mệnh đề WHERE và HAVING.
- Hiểu và sử dụng hàm tổng hợp: AVG, COUNT, MAX, MIN, SUM.
- Hiểu và sử dụng truy vấn con (lệnh SELECT lồng SELECT).
- Thực hiện truy vấn nối giữa các bảng.

2. Thực hiện

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

Bài thực hành 1: Viết câu lệnh truy vấn bảng Catalog để lấy dữ liệu Catalog và tổng số sản phẩm theo Catalog (theo mã ID). Trong cú pháp truy vấn sử dụng hàm COUNT để đếm số lượng sản phẩm. Bước 1: Mở cửa sổ gõ lệnh query truy vấn bảng lấy số giờ nguyên công:

```
-- Truy vấn tới bảng Catalog lấy thông tin Catalog  
SELECT CatalogId, CatalogName FROM Catalog
```



	CatalogId	CatalogName
1	1	Giầy nam
2	2	Giầy nữ
3	3	Dép nữ
4	4	Dép nam
5	5	Áo nam
6	6	Áo nữ
7	7	Quần nam
8	8	Quần nữ

Bước 2: Gõ mã lệnh truy vấn bảng Product lấy về tổng sản phẩm của mỗi danh mục:

```
-- Lấy về id của danh mục và số lượng sản phẩm  
SELECT  
    c.CatalogId, c.CatalogName,
```

```

COUNT(ProductId) AS 'Tổng sản phẩm'
FROM Catelog c
join Product p
on c.CatelogId = p.CatelogId
GROUP BY c.CatelogId, CatelogName -- Lệnh nhóm

```

	CatelogId	CatelogName	Tổng sản phẩm
1	1	Giày nam	3
2	2	Giày nữ	3
3	3	Dép nữ	3
4	4	Dép nam	3
5	5	Áo nam	4
6	6	Áo nữ	1
7	7	Quần nam	1
8	8	Quần nữ	2

Bài thực hành 2: Viết câu lệnh truy vấn bảng Product lấy về giá trị trung bình của PriceOutput theo danh mục sản phẩm nhóm theo CatelogId

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 Product

```

	ProductId	CatelogId	ProductName	Content	ContentDetail
1	1	1	Giày buộc giày công sở Sanvado	Giày buộc dây công sở Sanvado mũi bóng màu đen (...)	Giày buộc dây công sở Sanvado mũi bóng màu đen (...)
2	2	1	Giày nam buộc dây James Blanc	Giày nam buộc dây James Blanc da trơn màu đen (JB-...	Giày nam buộc dây James Blanc da trơn màu đen (JB-...
3	3	1	Giày da mềm Asos	Giày da mềm Asos	Giày da mềm Asos
4	4	2	Giày cao gót mũi nhọn	Giày cao gót mũi nhọn	Giày cao gót mũi nhọn
5	5	2	Giày cao gót đế lều đá mũi hờ	Giày cao gót đế lều đá mũi hờ	Giày cao gót đế lều đá mũi hờ
6	6	2	Giày cao gót dây buộc Asos	Giày cao gót dây buộc Asos	Giày cao gót dây buộc Asos
7	7	3	Dép xô ngón nhiều màu - DT170	Dép xô ngón nhiều màu - DT170	Dép xô ngón nhiều màu - DT170
8	8	3	Dép quai ngang	Dép quai ngang	Dép quai ngang
9	9	3	Dép đế xuồng hoa hồng quai trong	Dép đế xuồng hoa hồng quai trong	Dép đế xuồng hoa hồng quai trong
10	10	4	Dép cao su nam quai liền	Dép cao su nam quai liền	Dép cao su nam quai liền

Bước 2: Sử dụng lệnh GROUP BY để lấy dữ liệu,

```

-- Nhóm các sản phẩm
SELECT
    CatelogId,
    AVG(PriceOutput) AS 'Giá trung bình'
FROM Product
GROUP BY CatelogId

```

Results Messages		
	CatalogId	Giá trung bình
1	1	466666.666666667
2	2	566666.666666667
3	3	516666.666666667
4	4	760000
5	5	288125
6	6	900000
7	7	700000
8	8	975000

Bài thực hành 3: Viết câu lệnh truy vấn bảng BillDetail lấy về tổng doanh số của mỗi sản phẩm chi tiết, 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 BillDetail - lấy về tổng doanh số theo sản phẩm
SELECT * FROM BillDetail
```

Results Messages				
	BillId	ProductDetailId	don_gia	so_luong
1	1	1	400000	2
2	2	2	400000	1
3	2	3	600000	1
4	3	6	400000	1
5	3	7	300000	1
6	4	1	400000	1
7	4	7	300000	2

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
    ProductDetailId,
    SUM(PriceUnit * Quantity) AS 'Tổng doanh số'
FROM BillDetail
WHERE [ProductDetailId] <= 5 -- Điều kiện lọc
GROUP BY ALL ProductDetailId -- Lệnh nhóm chấp nhận cả khi điều kiện lọc ko thỏa và gán giá trị NULL
```

	ProductDetailId	Tổng doanh số
1	1	1200000
2	2	400000
3	3	600000
4	6	NULL
5	7	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
)
```

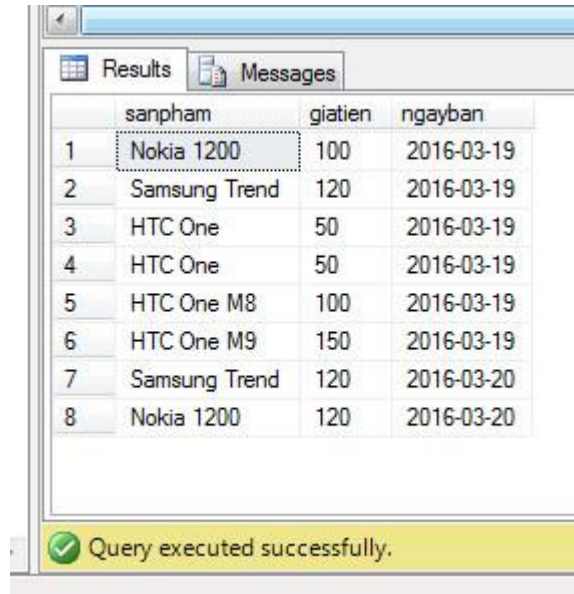
Bước 2: Thêm dữ liệu vào bảng:

```
-- 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 ThôngKeBanHang VALUES ('HTC One', 50, '2016-03-19')
INSERT INTO ThôngKeBanHang VALUES ('HTC One', 50, '2016-03-19')
INSERT INTO ThôngKeBanHang VALUES ('HTC One M8', 100, '2016-03-19')
INSERT INTO ThôngKeBanHang VALUES ('HTC One M9', 150, '2016-03-19')
```

```
INSERT INTO ThôngKeBanHang VALUES ('Samsung Trend', 120, '2016-03-20')
INSERT INTO ThôngKeBanHang VALUES ('Nokia 1200', 120, '2016-03-20')
```

Bước 3: Truy cập dữ liệu bảng:

```
-- Truy vấn
SELECT * FROM ThôngKeBanHang
```

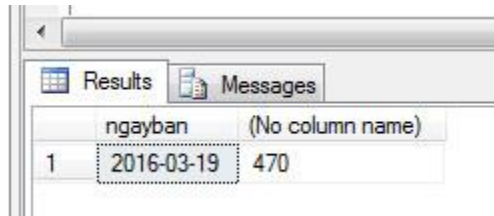


	sanpham	giatien	ngayban
1	Nokia 1200	100	2016-03-19
2	Samsung Trend	120	2016-03-19
3	HTC One	50	2016-03-19
4	HTC One	50	2016-03-19
5	HTC One M8	100	2016-03-19
6	HTC One M9	150	2016-03-19
7	Samsung Trend	120	2016-03-20
8	Nokia 1200	120	2016-03-20

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 ThôngKeBanHang
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:



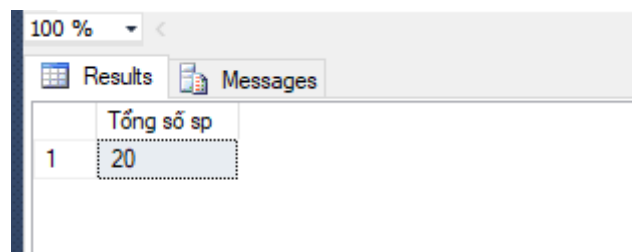
	ngayban	(No column name)
1	2016-03-19	470

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 QuanLyBanHang như sau:

Bước 1: Hàm COUNT:

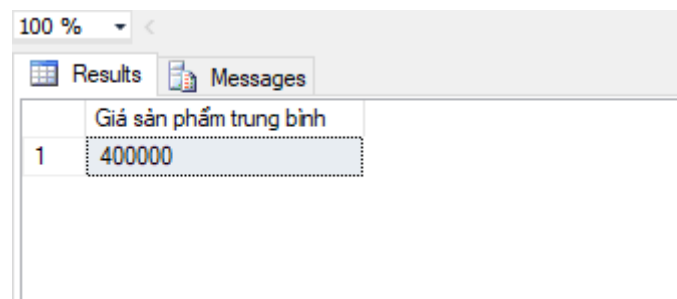
```
USE QuanLyBanHang
GO
-- Lấy về tổng số sản phẩm
SELECT COUNT(*) AS 'Tổng số sp' FROM Product
```



	Tổng số sp
1	20

Bước 2: Hàm AVG:

```
-- Tính trung bình giá sản phẩm trong tất cả hóa đơn
SELECT AVG(PriceUnit) AS 'Giá sản phẩm trung bình' FROM
BillDetail
```

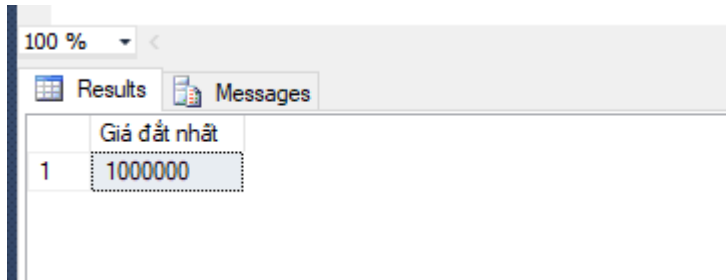


	Giá sản phẩm trung bình
1	400000

Bước 3: Hàm Max - Min:

```
USE QuanLyBanHang
Go
```

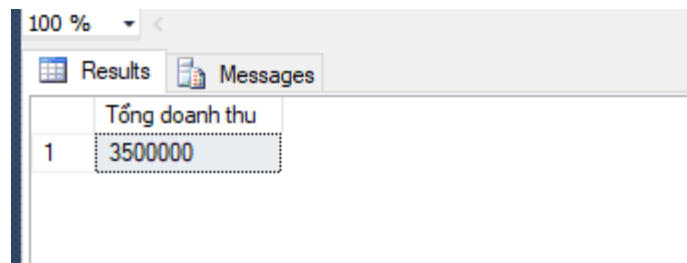
```
-- Lấy về giá sản phẩm đắt nhất  
SELECT MAX(PriceOutput) AS 'Giá đắt nhất' FROM Product
```



	Giá đắt nhất
1	1000000

Bước 4: Hàm SUM:

```
-- Tính tổng doanh thu bán hàng  
SELECT SUM(Amount) AS 'Tổng doanh thu' FROM Bill
```



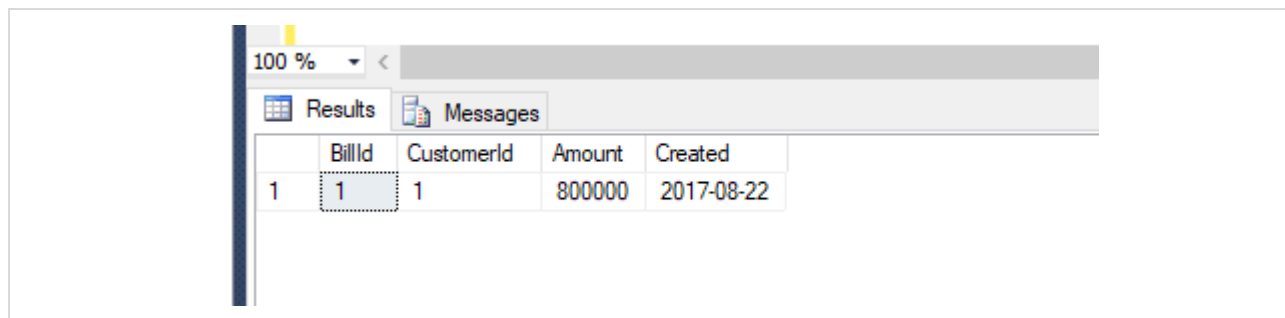
	Tổng doanh thu
1	3500000

Bài thực hành 6: Viết câu lệnh truy vấn bảng Bill lấy về tất cả thông tin đơn hàng 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  
    BillId,  
    CustomerId,  
    Amount,  
    Created  
FROM Bill  
WHERE  
-- Đặt điều kiện tìm kiếm là ngày Created là ngày cuối cùng trong csdl  
Created = (  
    -- Lấy về ngày cuối cùng trong bảng SalesOrderHeader  
    SELECT MAX(Created) FROM Bill  
)
```

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



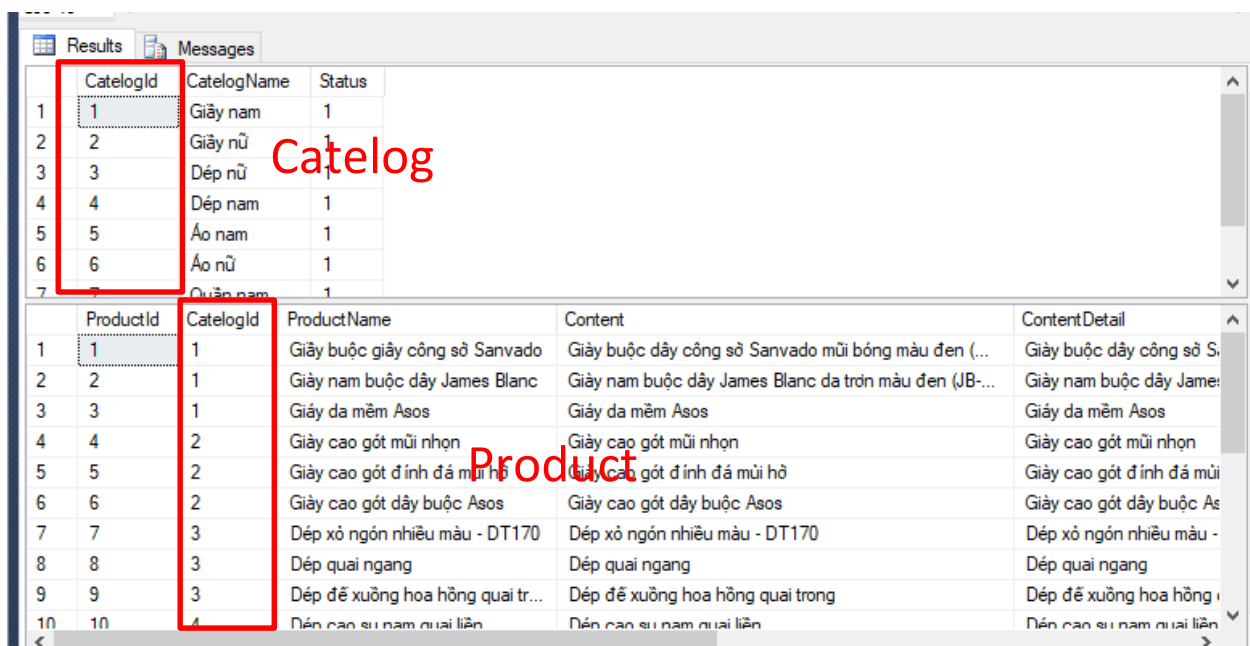
	BillId	CustomerId	Amount	Created
1	1	1	800000	2017-08-22

Bài thực hành 7: Viết câu lệnh truy vấn bảng Product lấy về những sản phẩm mà danh mục sản phẩm của nó là: “Áo nam”

Bước 1: Gõ lệnh SQL:

```
SELECT * FROM Product
GO
```

```
SELECT * FROM Catalog
GO
```



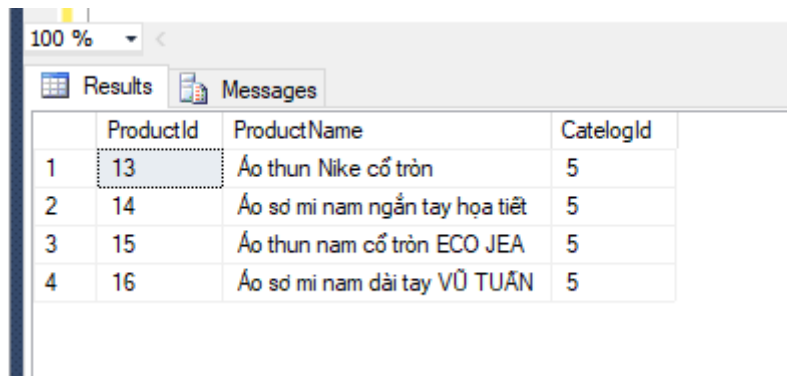
	CatalogId	CatalogName	Status
1	1	Giày nam	1
2	2	Giày nữ	1
3	3	Đép nữ	1
4	4	Đép nam	1
5	5	Áo nam	1
6	6	Áo nữ	1
7	7	Quần nam	1

	ProductId	CatalogId	ProductName	Content	ContentDetail
1	1	1	Giày buộc giày công sở Sanvado	Giày buộc dây công sở Sanvado mũi bóng màu đen (...)	Giày buộc dây công sở S...
2	2	1	Giày nam buộc dây James Blanc	Giày nam buộc dây James Blanc da trơn màu đen (JB-...	Giày nam buộc dây Jame...
3	3	1	Giày da mềm Asos	Giày da mềm Asos	Giày da mềm Asos
4	4	2	Giày cao gót mũi nhọn	Giày cao gót mũi nhọn	Giày cao gót mũi nhọn
5	5	2	Giày cao gót đế đinh đá mùi hờ	Giày cao gót đế đinh đá mùi hờ	Giày cao gót đế đinh đá mùi
6	6	2	Giày cao gót dây buộc Asos	Giày cao gót dây buộc Asos	Giày cao gót dây buộc As
7	7	3	Đép xỏ ngón nhiều màu - DT170	Đép xỏ ngón nhiều màu - DT170	Đép xỏ ngón nhiều màu -
8	8	3	Đép quai ngang	Đép quai ngang	Đép quai ngang
9	9	3	Đép đế xuồng hoa hồng quai tr...	Đép đế xuồng hoa hồng quai trong	Đép đế xuồng hoa hồng
10	10	4	Đép cao su nam quai liền	Đép cao su nam quai liền	Đép cao su nam quai liền

Hình trên dễ dàng nhận thấy giữa bảng Product và Catalog có mối liên hệ là CatalogId.

Bước 2: Yêu cầu lấy ra sản phẩm có danh mục là “Áo nam” :

```
SELECT ProductId, ProductName, CatalogId
FROM Product
WHERE Product.CatalogId IN
( SELECT CatalogId
  FROM Catalog
  WHERE CatalogName = N'Áo nam' )
```


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

The screenshot shows the SQL Server Enterprise Manager interface. The 'Results' pane displays a table with the following data:

	ProductId	ProductName	CatalogId
1	13	Áo thun Nike cổ tròn	5
2	14	Áo sơ mi nam ngắn tay họa tiết	5
3	15	Áo thun nam cổ tròn ECO JEA	5
4	16	Áo sơ mi nam dài tay VŨ TUẤN	5

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 CatalogId thỏa mãn yêu cầu có tên là “Áo nam”.
2. Lệnh WHERE bên ngoài liền kề sẽ thực hiện lọc CatalogId trên bảng Product để chỉ lấy về bản ghi mà thỏa mã CatalogId 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 bảng Product lấy về những sản phẩm có màu xanh:

Bước 1: Gõ lệnh SQL:

```
SELECT * FROM Product
GO
SELECT * FROM ProductDetail
GO
SELECT * FROM Color
```

Results Messages

	ProductId	CatalogId	ProductName	Content	ContentDetail
1	1	1	Giày buộc dây công sở Sanvado	Giày buộc dây công sở Sanvado mũi bóng màu đen (...)	Giày buộc dây công s
2	2	1	Giày nam buộc dây James Blanc	Giày nam buộc dây James Blanc da trơn màu đen (JB-...	Giày nam buộc dây Ja
3	3	1	Giày da mềm Asos	Giày da mềm Asos	Giày da mềm Asos
4	4	2	Giày cao gót mũi nhọn	Giày cao gót mũi nhọn	Giày cao gót mũi nhọn
5	5	2	Giày cao gót đế đinh đá mũi hở	Giày cao gót đế đinh đá mũi hở	Giày cao gót đế đinh đá
6	6	2	Giày cao gót dây buộc Asos	Giày cao gót dây buộc Asos	Giày cao gót dây buộc

	ProductDetailId	ProductId	ColorId	SizeId	quantity	Status
1	1	1	1	1	4	1
2	2	1	2	1	6	1
3	3	2	2	2	5	1
4	4	2	1	3	4	1
5	5	9	2	4	5	1

	ColorId	ColorName	Status
1	1	Màu đỏ	1
2	2	Màu xanh	1
3	3	Màu tím	1
4	4	Màu trắng	1
5	5	Màu đen	1
6	6	Màu hồng	1
7	7	Màu vàng	1

Query executed successfully. | DESKTOP-D10T47U\SQLEXPRESS ... | DESKTOP-D10T47U\nguye ... | QuanLyBanHang | 00:00:00 | 35

Bảng **ProductDetail** có cột **ProductId** là **ProductId** ở bảng **Product** Trong **ProductDetail** có **ColorId** là mã màu sắc quy định ở bảng **Color**. Như vậy để lấy thông tin sản phẩm có màu xanh ta phải tiến hành các bước truy vấn:

Bước 2: Lấy về sản phẩm có màu xanh

```

/*
Truy vấn lồng
*/
SELECT ProductId, ProductName, PriceInput, PriceOutput
FROM Product
WHERE ProductId IN (
    SELECT ProductId
    FROM ProductDetail
    WHERE ColorId IN (
        SELECT ColorId
        FROM Color
        WHERE ColorName = N'Màu xanh'
    )
)

```

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

100 %

Results Messages

	ProductId	ProductName	PriceInput	PriceOutput
1	1	Giày buộc giấy công sở Sanvado	350000	400000
2	2	Giày nam buộc dây James Blanc	500000	600000
3	9	Đép để xuống hoa hồng quai trong	350000	400000

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 QuanLyBanHang):

Bước 1: Gõ lệnh SQL:

```
USE QuanLyHangHoa
GO
-- Truy vấn bảng hóa đơn và lấy về cả thông tin khách hàng
SELECT
    b.BillId,
    b.Name,
    b.Address,
    b.Phone,
    b.Email
FROM Bill b, Customer c
WHERE c.CustomerId = b.CustomerId
```

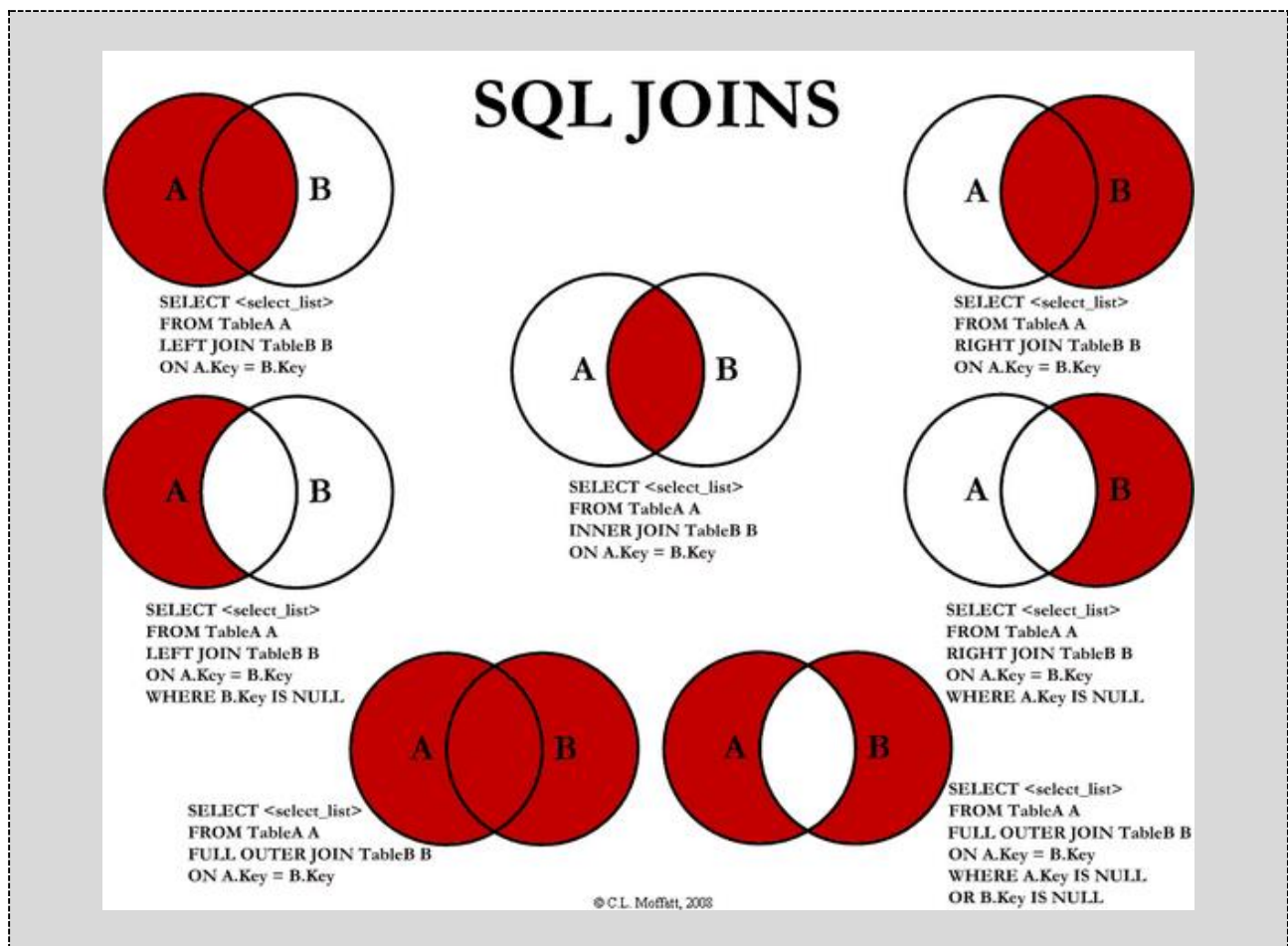
Bước 2: Quét dòng mã lệnh trên rồi bấm F5:

100 %

Results Messages

	BillId	Name	Address	Phone	Email
1	1	Dinh Long	Hai Duong	0956783453	long@gmail.com
2	2	Dinh Long	Hai Duong	0956783453	long@gmail.com
3	3	Hoang Tuan Anh	Nam Dinh	087543178	anh@gmail.com
4	4	Hoang Tuan Anh	Nam Dinh	087543178	anh@gmail.com

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 QuanLyBanHang
GO
-- Truy vấn bảng hóa đơn và lấy về cả thông tin khách hàng
SELECT
    b.BillId,
    b.Name,
    b.Address,
    b.Phone,
    b.Email
FROM Bill b INNER JOIN Customer c
ON b.CustomerId = c.CustomerId
```

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

100 %

Results Messages

	BillId	Name	Address	Phone	Email
1	1	Dinh Long	Hai Duong	0956783453	long@gmail.com
2	2	Dinh Long	Hai Duong	0956783453	long@gmail.com
3	3	Hoang Tuan Anh	Nam Dinh	087543178	anh@gmail.com
4	4	Hoang Tuan Anh	Nam Dinh	087543178	anh@gmail.com

Bước 3: Gõ lệnh SQL – LEFT JOIN:

```
-- Left join
SELECT CatelogName, ProductName
FROM Catelog LEFT JOIN Product
ON Catelog.CatelogId = Product.CatelogId
```

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

Results Messages

	CatelogName	ProductName
7	Dép nữ	Dép xỏ ngón nhiều màu - DT170
8	Dép nữ	Dép quai ngang
9	Dép nữ	Dép đế xuồng hoa hồng quai trong
10	Dép nam	Dép cao su nam quai liền
11	Dép nam	xăng đan nam
12	Dép nam	Dép xỏ ngón kito thailand
13	Áo nam	Áo thun Nike cổ tròn
14	Áo nam	Áo sơ mi nam ngắn tay họa tiết
15	Áo nam	Áo thun nam cổ tròn ECO JEA
16	Áo nam	Áo sơ mi nam dài tay VŨ TUẤN
17	Áo nữ	Áo voan cộc tay cao cổ
18	Quần nam	Quần Jean rách gối bó
19	Quần nữ	Quần jean bóng
20	Quần nữ	null

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

Bước 1: UNION - gõ lệnh SQL:

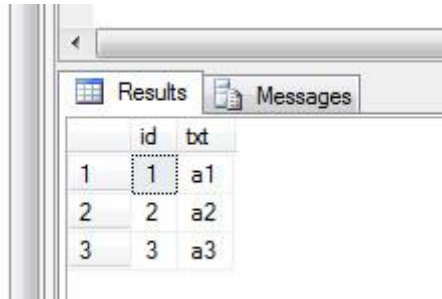
```
-- Tạo bảng
CREATE TABLE #test1(id INT, txt VARCHAR(10) )
CREATE TABLE #test2(id INT, txt VARCHAR(10) )

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
```

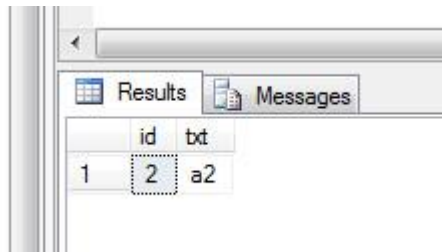


	id	txt
1	1	a1
2	2	a2
3	3	a3

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	2	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 câu trúc giống hệt nhau).

Bước 3: INTERSECT – gõ lệnh SQL:

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

	id	txt
1	1	a1
2	2	a2

	id	txt
1	1	a1
2	3	a3

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:

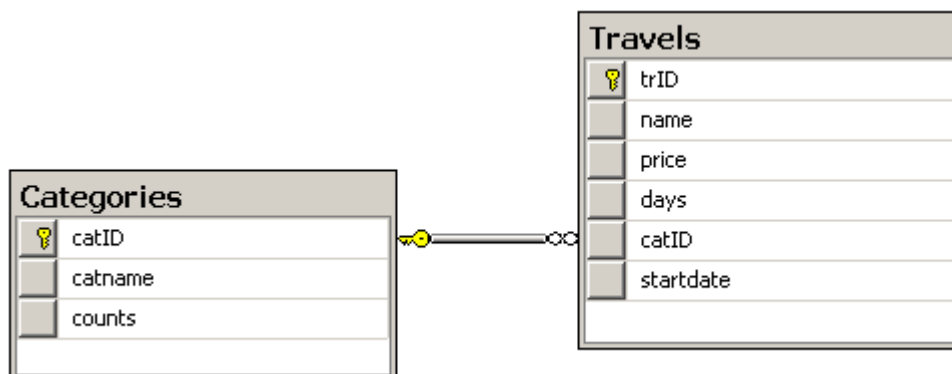
Hãy tạo các bảng với cấu trúc như sau:

Bảng <u>Travels</u>		
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 <u>Categories</u>		
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 :

	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)