



CHƯƠNG 3

NGÔN NGỮ TRUY VẤN DỮ LIỆU

(Structured Query Language - SQL)

Giới thiệu một số toán tử

✓ *Toán tử số học*

+	cộng
-	Trừ
*	Nhân
/	Chia
%	Chia lấy phần dư

✓ *Toán tử so sánh*

>	Lớn hơn.
>=	Lớn hơn hoặc bằng.
=	Bằng.
<	Nhỏ hơn.
<=	Nhỏ hơn hoặc bằng.
<>	Khác nhau
!<	không nhỏ hơn
!>	không lớn hơn

Giới thiệu một số toán tử

✓ *Toán tử logic*

NOT Phủ định.

AND Và

OR Hoặc

✓ *Toán tử tập hợp*

UNION Hợp

INTERSECT Giao

EXCEPT Hiệu

IN Thuộc về

✓ *Toán tử so khớp chuỗi:*

- LIKE <mẫu cần so khớp>
- Kí tự đại diện cho nhóm kí tự %
- Kí tự đại diện cho một kí tự _

✓ *Toán tử nối chuỗi: +*

Một số hàm thường dùng trong SQL – Toán học

- ❑ `abs(x)` : Trị tuyệt đối của x
- ❑ `sqrt(x)` : Căn bậc hai của x
- ❑ `square(x)` : x^2
- ❑ `power(y, x)` : y^x
- ❑ `log(x)` : Logarit tự nhiên của x
- ❑ `exp(x)` : Hàm mũ cơ số e của x : e^x .
- ❑ `sign(x)` : Lấy dấu của số x (-1 : $x < 0$, 0 : $x = 0$, $+1$: $x > 0$)
- ❑ `round(x,n)` : Làm tròn tới n số lẻ.
- ❑ `ceiling(x)` : Số nguyên nhỏ nhất nhưng lớn hơn x
- ❑ `floor(x)` : Số nguyên lớn nhất nhưng nhỏ hơn x

Một số hàm thường dùng trong SQL – Xử lý chuỗi

❑ Upper(str) : Chuyển sang chữ hoa

Ví dụ: Upper('hoa') → 'HOA'

❑ Lower(str) : Chuyển sang chữ thường

Ví dụ: Lower('DUNG') → 'dung'

❑ Len(str) : Trả về độ dài của chuỗi str

❑ Ltrim(str) : Trả về chuỗi sau khi bỏ kí tự trắng bên trái chuỗi str

❑ Rtrim(str) : Trả về chuỗi sau khi bỏ kí tự trắng bên phải chuỗi str

❑ Left(str,n): Trả về n kí tự bên trái chuỗi str

❑ Right(str,n): Trả về n kí tự bên phải chuỗi str

❑ Substring(str,m,n): Trả về chuỗi con, từ kí tự thứ m, lấy n kí tự trong chuỗi str

Ví dụ: Substring('abcdef',3,2) → 'cd'

Một số hàm thường dùng trong SQL – Hàm ngày tháng

- ❑ Getdate(): Trả về ngày giờ hệ thống
- ❑ Day(*date*): Trả về ngày trong tháng của ngày date
- ❑ Month(*date*): Trả về tháng của ngày date
- ❑ Year(*date*): Trả về năm của ngày date

<ul style="list-style-type: none">❑ DATEPART(datepart, date): trích ra phần tương ứng với datepart từ biểu thức ngày date.❑ DATEADD(datepart, number, date): trả về một ngày bằng biểu thức date cộng thêm với number.❑ DATEDIFF(datepart, date1, date2): trả về một số với đơn vị tính là datepart bằng date2 – date1.	Datepart	Abbreviations
	Year	yy, yyyy
	Quarter	qq, q
	Month	mm, m
	Day of year	dy, y
	Day of Month	dd, d
	Week of year	wk, ww
	Weekday	dw
	Hour	hh
	Minute	mi, n
	Second	ss, s
	Millisecond	Ms

- ❑ **Chú ý:** Để định lại thứ tự các phần của kiểu ngày cho phù hợp. Sử dụng lệnh:
Set Dateformat **mdy** | **dmy** | **ymd** | **ydm** | **myd** | **dym**

Một số hàm thường dùng trong SQL – Hàm kiểm tra dữ liệu

- ❑ ISDATE(variable | column name): Kiểm tra dạng ngày hợp lệ. Trả về 1 nếu hợp lệ và 0 nếu không hợp lệ.
- ❑ ISNUMERIC(variable | column name): Kiểm tra dạng số hợp lệ. Trả về 1 nếu hợp lệ và 0 nếu không hợp lệ.
- ❑ ISNULL(expression, value): Trả về giá trị value nếu expression có giá trị NULL, ngược lại trả về giá trị của expression. Giá trị trả về cùng kiểu với expression.
- ❑ IF(<biểu thức điều kiện>, <bt 1>, <bt 2>) : Trả về giá trị của <bt 1> nếu <biểu thức điều kiện> đúng, ngược lại trả về giá trị của <bt 2>.

Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu của Dữ Liệu

- ❑ STR(x [, len [, dec]]): Chuyển số x sang kiểu chuỗi có chiều dài len và phần lẻ dec

```
Select MaMH, Str(DonGia, 10, 2) + N' đồng'  
From MatHang
```


Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu của Dữ Liệu

❑ `Cast(expression AS data_type[(length)])`

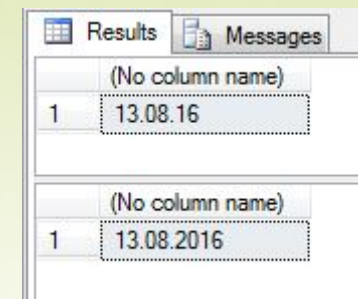
Ví dụ: `Cast(('20'+ '10') as Int) + 10 => 2020`

❑ `Convert(data_type[(length)], expression [, style])`

Style: Dạng thức kiểu ngày mà chúng ta muốn khi chuyển đổi dữ liệu kiểu datetime, datetime2 hoặc smalldatetime tới kiểu ký tự (char, varchar, nchar, or nvarchar). Hoặc dạng chuỗi mà chúng ta muốn khi chuyển dữ liệu kiểu số (float, real, money, or smallmoney) sang kiểu ký tự (char, varchar, nchar, or nvarchar).

Ví dụ:

```
Select CONVERT(varchar(10), GETDATE(), 4)
Select CONVERT(varchar(10), GETDATE(), 104)
```



(No column name)	
1	13.08.16

(No column name)	
1	13.08.2016

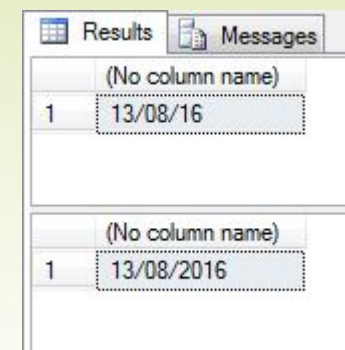
Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu của Dữ Liệu

❑ Giá trị *Style*: để chuyển kiểu ngày giờ sang kiểu chuỗi ký tự

Without century (yy)	With century (yyyy)	Standard	Input/Output
-	0 or 100	Default	mon dd yyyy hh:miAM (or PM)
1	101	USA	mm/dd/yy
2	102	ANSI	yy.mm.dd
3	103	British/French	dd/mm/yy
4	104	German	dd.mm.yy
5	105	Italian	dd-mm-yy
6	106	-	dd mon yy
7	107	-	mon dd, yy
8	108	-	hh:mm:ss
-	9 or 109	Default+milliseconds	mon dd yyyy hh:mi:ss:mmmAM (or PM)
10	110	USA	mm-dd-yy

❑ Ví dụ:

```
Select CONVERT (varchar(10), GETDATE (), 3)
Select CONVERT (varchar(10), GETDATE (), 103)
```



(No column name)	
1	13/08/16
(No column name)	
1	13/08/2016

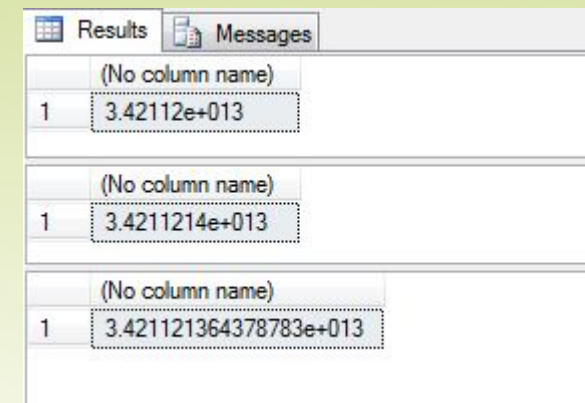
Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu của Dữ Liệu

- ❑ Giá trị *Style*: để chuyển kiểu Float, Real sang kiểu chuỗi ký tự

0 (default)	Biểu diễn số nhỏ với số ký tự tối đa là 7, biểu diễn số lớn dưới dạng khoa học với độ dài cố định 12 ký tự
1	Luôn biểu diễn số dưới dạng khoa học với độ dài cố định 14 ký tự
2	Luôn biểu diễn số dưới dạng khoa học với độ dài cố định 22 ký tự

- ❑ Ví dụ:

```
declare @n float = 34211213643787.834728
select CONVERT(varchar(50), @n, 0)
select CONVERT(varchar(50), @n, 1)
select CONVERT(varchar(50), @n, 2)
```



Results	
(No column name)	
1	3.42112e+013
(No column name)	
1	3.4211214e+013
(No column name)	
1	3.421121364378783e+013

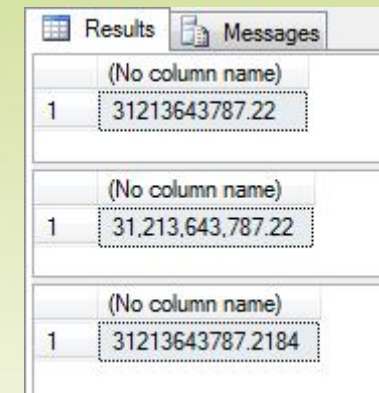
Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu của Dữ Liệu

- ❑ *Giá trị Style*: để chuyển kiểu SmallMoney, Money sang kiểu chuỗi ký tự

0 (default)	Không dấu phẩy phân cách hàng ngàn, có 2 chữ số thập phân Ví dụ: 4235.98
1	Có dấu phẩy phân cách hàng ngàn và 2 chữ số thập phân Ví dụ: 3,510.92
2	Không dấu phẩy phân cách hàng ngàn, có 4 chữ số thập phân Ví dụ: 4235.9819

- ❑ Ví dụ:

```
declare @m money = 31213643787.218389
select CONVERT(varchar(50), @m, 0)
select CONVERT(varchar(50), @m, 1)
select CONVERT(varchar(50), @m, 2)
```



(No column name)	
1	31213643787.22
(No column name)	
1	31,213,643,787.22
(No column name)	
1	31213643787.2184

Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu của Dữ Liệu

- ❑ **Hàm case:** Cú pháp 1: So sánh giá trị biểu thức với giá trị sau WHEN để xác định kết quả

CASE *biểu_thức*

WHEN *giá_trị_1* **THEN** *kết_quả_1*

WHEN *giá_trị_2* **THEN** *kết_quả_2*

[...n]

[**ELSE** *kết_quả_khác*]

END

- ❑ Ví dụ: Trả về thứ trong tuần của ngày lập hóa đơn

Select (Case Datepart(W, NgayLHD)

When 1 Then 'Chu Nhật'

When 2 Then 'Thu Hai'

When 3 Then 'Thu Ba'

When 4 Then 'Thu Tu'

When 5 Then 'Thu Nam'

When 6 Then 'Thu Sau'

Else 'Thu Bay'

End) AS Thu

From HoaDon

Một số hàm thường dùng trong SQL – Hàm chuyển đổi kiểu của Dữ Liệu

❑ **Hàm case:** Cú pháp 2: Kiểm tra từng biểu thức điều kiện để xác định kết quả
CASE

WHEN *biểu_thức_logic1* **THEN** *kết_quả_1*

WHEN *biểu_thức_logic2* **THEN** *kết_quả_2*

 [...*n*]

 [**ELSE** <*kết_quả_khác*>]

END

❑ Ví dụ: trả về số ngày trong tháng hiện tại

Select (Case

 When Month(GetDate()) In (1,3,5,7,8,10,12) Then 31

 When Month(GetDate()) In (4,6,9,11) Then 30

 When Year(GetDate()) % 4 = 0 Then 29

Else 28

End) AS SoNgay



NHÓM LỆNH TRUY VẤN DỮ LIỆU

Cú pháp tổng quát

SELECT < danh sách thuộc tính >

[INTO new_table]

[FROM < bảng nguồn >]

[WHERE < điều kiện chọn dòng >]

[GROUP BY < danh sách thuộc tính phân nhóm >]

[HAVING < điều kiện chọn nhóm >]

[ORDER BY < danh sách thuộc tính sắp xếp > [ASC | DESC]]

[UNION | INTERSECT | EXCEPT < câu truy vấn khác >]

✗ Thứ tự dịch của lệnh Select:

FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

Lệnh Select đơn giản

SELECT <danhsáchthuộctính> **FROM** <tên bảng>

Gồm 02 phần:

- ❑ Bắt đầu bởi từ khóa **Select**, theo sau là tên các thuộc tính cần truy vấn, phân cách bởi dấu phẩy. Trong <danhsáchthuộctính> có thể sử dụng các thành phần đại diện:
 - Dấu * đại diện cho tất cả các thuộc tính có ở nguồn dữ liệu.
 - TableName.* đại diện cho tất cả các thuộc tính của quan hệ có tên là TableName
- ❑ Xác định nguồn dữ liệu, bắt đầu với từ khóa **From**, theo sau là tên table hay truy vấn chứa thông tin cần tìm

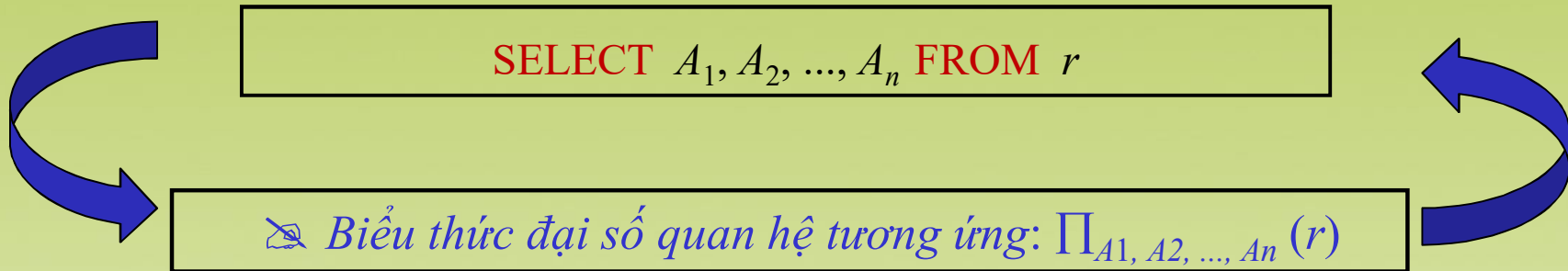
Ví dụ: Hiện danh sách sinh viên gồm: mã sinh viên, họ, tên và mã ngành học

Select MaSV, HoSV, TenSV, MaNganh **From** SinhVien

Ví dụ: Liệt kê tất cả thông tin của sinh viên

Select * **From** SinhVien **Hoặc** **Select** SinhVien.* **From** SinhVien

Lệnh Select đơn giản



r

A	B	C
a1	b1	c1
a2	b2	c2
a1	b1	c3
a3	b2	c1

SELECT A, B FROM r

Kết quả

A	B
a1	b1
a2	b2
a1	b1
a3	b2

Lệnh Select đơn giản

SELECT [**DISTINCT**] A_1, A_2, \dots, A_n **FROM** r

- **DISTINCT**: Trường hợp có nhiều bộ trùng lặp giá trị nếu chỉ muốn giữ lại một bộ trong các bộ giống nhau tìm được thì sử dụng từ khóa **DISTINCT**

r

A	B	C
a1	b1	c1
a2	b2	c2
a1	b1	c3
a3	b2	c1

SELECT A, B **FROM** r

Kết quả

A	B
a1	b1
a2	b2
a1	b1
a3	b2

r

A	B	C
a1	b1	c1
a2	b2	c2
a1	b1	c3
a3	b2	c1

SELECT **DISTINCT** A, B **FROM** r

Kết quả

A	B
a1	b1
a2	b2
a3	b2

Loại bỏ bộ trùng lặp

Lệnh Select đơn giản

SELECT **[[TOP n] PERCENT]** A_1, A_2, \dots, A_n **FROM** r

- **[[TOP n] PERCENT]**: **TOP n** nếu chỉ muốn liệt kê n bộ đầu tiên trong kết quả tìm được. **TOP n PERCENT** nếu chỉ muốn liệt kê n% bộ đầu tiên trong kết quả tìm được. (n là số nguyên dương)

r

A	B	C
a1	b1	c1
a2	b2	c2
a1	b1	c3
a3	b2	c1

SELECT A, B **FROM** r

Kết quả

A	B
a1	b1
a2	b2
a1	b1
a3	b2

r

A	B	C
a1	b1	c1
a2	b2	c2
a1	b1	c3
a3	b2	c1

SELECT **TOP 1** A, B **FROM** r

Kết quả

A	B
a1	b1

Lệnh Select đơn giản

- ❑ Thông tin cần truy vấn có thể **được dẫn xuất** (được tính toán từ các thuộc tính khác), có thể sử dụng **các toán tử** hay **hàm** có trong hệ quản trị CSDL.

Ví dụ: Xét lược đồ quan hệ **ChiTietPhieuXuat**(SoPX, MaMH, SoLg, DonGia)

Viết câu truy vấn hiển thị kết quả: số phiếu xuất, mã các mặt hàng xuất kèm số lượng, đơn giá và số tiền tương ứng của từng mặt hàng.

Select SoPX, MaMH, SoLg, DonGia, **SoLg*DonGia** **From** ChiTietPhieuXuat

SOPX	MAMH	SOLG	DONGIA
1	MB	2	1500
1	MOUSE	3	70
2	KB	1	90
3	MOUSE	5	70
3	KB	4	90
4	MB	3	1500

Kết quả

SOPX	MAMH	SOLG	DONGIA	SOLG*DONGIA
1	MB	2	1500	3000
1	MOUSE	3	70	210
2	KB	1	90	90
3	MOUSE	5	70	350
3	KB	4	90	360
4	MB	3	1500	4500

Lệnh Select đơn giản

- ❑ Có thể đặt **nhãn** cho thông tin cần truy vấn: **<biểu thức> [As] <nhãn>**

Select SoPX, MaMH, SoLg, DonGia, **SoLg*DonGia** **From** ChiTietPhieuXuat

SOPX	MAMH	SOLG	DONGIA
1	MB	2	1500
1	MOUSE	3	70
2	KB	1	90
3	MOUSE	5	70
3	KB	4	90
4	MB	3	1500

Kết quả

SOPX	MAMH	SOLG	DONGIA	SOLG*DONGIA
1	MB	2	1500	3000
1	MOUSE	3	70	210
2	KB	1	90	90
3	MOUSE	5	70	350
3	KB	4	90	360
4	MB	3	1500	4500

Select SoPX, MaMH, SoLg, DonGia, **SoLg*DonGia As TTien** **From** ChiTietPhieuXuat

SOPX	MAMH	SOLG	DONGIA
1	MB	2	1500
1	MOUSE	3	70
2	KB	1	90
3	MOUSE	5	70
3	KB	4	90
4	MB	3	1500

Kết quả

SOPX	MAMH	SOLG	DONGIA	TTIEN
1	MB	2	1500	3000
1	MOUSE	3	70	210
2	KB	1	90	90
3	MOUSE	5	70	350
3	KB	4	90	360
4	MB	3	1500	4500

MỆNH ĐỀ FROM

Mệnh đề From

- ❑ Dùng để chỉ định bảng nguồn chứa dữ liệu cần truy vấn.

From <tên bảng> [**As** <nhãn>]

- ❑ Trường hợp truy vấn thông tin từ nhiều bảng dữ liệu, sử dụng phép kết giữa các bảng trong mệnh đề **From**, các phép kết bao gồm:

1. Inner Join
2. Outer Join (Left Join, Right Join, Full Join)
3. Cross Join

Ngoài ra, trường hợp đặc biệt ta có:

1. Self Join
2. Excluding Join (Left Excluding Join, Right Excluding Join, Full Excluding Join)

Mệnh đề From - Phép kết nối INNER JOIN

✓ Inner Join

... r_1 **INNER JOIN** r_2 **ON** <đk>

✓ Biểu thức ĐSQH tương ứng: r_1  r_2

SELECT * **FROM** r_1 **INNER JOIN** r_2 **ON** $r_1.B = r_2.B$

r1

A	B
a1	b1
a2	b3
a1	b2
a3	b1

r2

B	D
b2	y
b2	z
b4	z
b3	z

r_1 **Inner Join** r_2 **On** $r_1.B = r_2.B$

A	B	B	D
a2	b3	b3	z
a1	b2	b2	y
a1	b2	b2	z

Mệnh đề From - Phép kết nối INNER JOIN

Ví dụ: SinhVien(MASV, HOSV, TENSX, MANGANH,...)


KetQua(MASV, MAMH, LANTHI, DIEM)

Viết lệnh SQL lấy thông tin gồm MASV, HOSV, TENSX của những sinh viên có điểm thi môn CSDL1, lần 1 lớn hơn hay bằng 5.


```
SELECT SinhVien.MASV, HOSV, TENSX  
FROM SinhVien INNER JOIN KetQua ON SinhVien.MASV=KetQua.MASV  
WHERE MAMH='CSDL1' and LANTHI=1 and DIEM>=5
```

Điều kiện kết

Mệnh đề From - Phép kết nối LEFT OUTER JOIN

✓ Left Outer Join

... r_1 **LEFT JOIN** r_2 **ON** <đk>

✓ Biểu thức ĐSQH tương ứng r_1  r_2

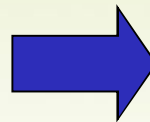
SELECT * **FROM** r_1 **LEFT JOIN** r_2 **ON** $r_1.B = r_2.B$

r_1

A	B
a1	b1
a2	b3
a1	b2
a3	b1

r_2

B	D
b2	y
b2	z
b4	z
b3	z




A	B	B	D
a1	b1	NULL	NULL
a2	b3	b3	z
a1	b2	b2	y
a1	b2	b2	z
a3	b1	NULL	NULL

Mệnh đề From - Phép kết nối RIGHT OUTER JOIN

✓ Right Outer Join

... r_1 **RIGHT JOIN** r_2 **ON** <đk>

✓ Biểu thức ĐSQH tương ứng r_1  r_2 <đk>

SELECT * FROM r_1 **RIGHT JOIN** r_2 **ON** $r_1.B = r_2.B$

r1

A	B
a1	b1
a2	b3
a1	b2
a3	b1

r2

B	D
b2	y
b2	Z
b4	z
b3	z



A	B	B	D
a1	b2	b2	y
a1	b2	b2	Z
NULL	NULL	b4	z
a2	b3	b3	z

Mệnh đề From - Phép tích Descarter

..... FROM r_1, r_2, \dots, r_n

✎ Biểu thức ĐSQH tương ứng: $r_1 \times r_2 \times \dots \times r_n$

SELECT * FROM r_1, r_2

r_1	
A	B
a1	b1
a2	b1
a3	b2

r_2	
C	D
x2	y1
x1	y2



$r_1 \times r_2$			
A	B	C	D
a1	b1	x2	y1
a1	b1	x1	y2
a2	b1	x2	y1
a2	b1	x1	y2
a3	b2	x2	y1
a3	b2	x1	y2

Mệnh đề From - Phép tích Descarter

- ✓ Ví dụ: Cho biết kết quả thi môn T001 với các thông tin MASV, HOSV, TENSX, LANTHI, DIEM

SinhVien

MASV	HOSV	TENSX	PHAINU	NGAYSINH	QQUAN	MANGANH	KHOAHOC
100	Lê Văn	An	False	060687	DN	DY	04
101	Lương Ngọc Trường	Thủy	True	140488	CM	TH	04
103	Lý Thủy	Nga	True	240288	CM	TH	05
104	Mai Nhật	An	False	251188	LA	TH	04
106	Nguyễn Thị Thủy	Anh	True	060688	BT	DY	04
108	Ngô Thị Trường	Hương	True	140488	TN	QT	04
109	Ngô Thủy	Lê	True	280588	TG	QT	05

KetQua

MASV	MAMH	LANTHI	DIEM
100	T001	1	7
103	CSDL1	1	3
103	CSDL1	2	5
103	T001	1	3
104	CSDL1	1	6
104	T001	1	4
104	T001	2	6

- ✓ Biểu thức ĐSQH:

$$\pi_{\text{MASV, HOSV, TENSX, LANTHI, DIEM}}(\sigma_{\text{MAMH}='T001'}(\text{SinhVien} \times \text{KetQua}))$$

- ✓ Lệnh SQL:

Select SinhVien.MASV, HOSV, TENSX, LANTHI, DIEM

From SinhVien, KetQua

Where SinhVien.MASV=KetQua.MASV **And** MAMH='T001'

Mệnh đề From

SinhVien

MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH	KHOAHOC
100	Lê Văn	An	False	060687	DN	DY	04
101	Lương Ngọc Trường	Thủy	True	140488	CM	TH	04
103	Lý Thủy	Nga	True	240288	CM	TH	05
104	Mai Nhật	An	False	251188	LA	TH	04
106	Nguyễn Thị Thúy	Anh	True	060688	BT	DY	04
108	Ngô Thị Trường	Hương	True	140488	TN	QT	04
109	Ngô Thuý	Lê	True	280588	TG	QT	05

KetQua

MASV	MAMH	LANTHI	DIEM
100	T001	1	7
103	CSDL1	1	3
103	CSDL1	2	5
103	T001	1	3
104	CSDL1	1	6
104	T001	1	4
104	T001	2	6

✓ *Lệnh SQL:*

Select SinhVien.MASV, HOSV, TENSVC, LANTHI, DIEM

From SinhVien, KetQua

Where SinhVien.MASV=KetQua.MASV **And** MAMH='T001'

- ❑ Một điểm **cần lưu ý** là thuộc tính mã sinh viên (MASV) có trong cả hai bảng SinhVien và KetQua, do đó khi viết lệnh phải chỉ rõ MASV của bảng nào, phải viết đầy đủ gồm: Tên bảng và dấu chấm trước tên thuộc tính (**SinhVien.MASV**)
- ❑ **Tổng quát:** Đối với các thuộc tính trùng tên trên nhiều bảng, cần chỉ rõ thuộc tính lấy của bảng nào: $\{< \text{tên_bảng} > | < \text{nhãn} > \}. < \text{tên thuộc tính} >$

Mệnh đề From

Ví dụ: Xét SINHVIEN(MASV, HOSV, TENSX, MANG,...) lấy ra danh sách các sinh viên cùng ngành với sinh viên mã số 104

- Tương ứng với biểu thức đại số quan hệ:

$$\sigma_{sv.MANGANH=SinhVien.MANGANH \wedge sv.MASV='104'}(SinhVien \times (\rho_{sv}(SinhVien)))$$

- Câu lệnh SQL tương ứng:

Select sv2.*

Đặt lại nhãn mới cho quan hệ
SinhVien

From SinhVien as sv, SinhVien sv2

Where sv.MASV='104' And sv.MANGANH=sv2.MANGANH

- ❑ Để giảm nhẹ công việc phải viết tên bảng nhiều lần trong lệnh, SQL hỗ trợ gán nhãn (Alias) cho bảng nguồn trong mệnh đề From, nhãn này có thể được sử dụng trước khi được đặt.
FROM table_name [**AS**] <nhãn>

MỆNH ĐỀ WHERE

Mệnh đề Where

❑ Mệnh đề **Where** <điều kiện chọn bộ> dùng chọn ra những bộ thỏa mãn điều kiện. <điều kiện chọn bộ> được thiết lập bao gồm:

- Các phép so sánh (comparison): <, <=, !<, >, >=, !>, =, <> hay !=
- Các phép toán logic: **AND, OR, NOT**
- Kiểm tra giá trị rỗng: <biểu thức> **IS [NOT] NULL**
- Kiểm tra thuộc tập giá trị: <biểu thức> **[NOT] IN** (<danh sách giá trị>)
- Kiểm tra thuộc miền giá trị: <biểu thức> **[NOT] BETWEEN** <Min> **AND** <Max>
- Đối sánh mẫu : <biểu thức> **[NOT] LIKE** ‘Mẫu văn bản’

Các truy vấn lồng nhau (sử dụng kết quả của một truy vấn làm điều kiện chọn bộ - **SubQuery**)

- Thuộc tập SubQuery: <biểu thức> **[NOT] IN** (SubQuery)
- So sánh hơn với tập SubQuery: <biểu thức> <comparison> **ANY** (subquery)
- So sánh nhất với tập SubQuery : <biểu thức> <comparison> **ALL** (subquery)
- Kiểm tra tồn tại: **[NOT] EXISTS**(SubQuery): Trả về **True** nếu tồn tại ít nhất 1 mẫu tin trong SubQuery, ngược lại trả về **False**

Mệnh đề Where

❑ Các phép so sánh (comparison): <, <=, !<, >, >=, !>, =, <> hay !=

..... *WHERE* <điều kiện chọn bộ>

⌵ *Biểu thức ĐSQH tương ứng: $\sigma_{\text{<điều kiện chọn>}(r)$*

A	B	C
a1	b1	c1
a2	b2	c2
a1	b1	c3
a3	b2	c1

SELECT *
FROM r
WHERE B='b2'

Kết quả

A	B	C
a2	b2	c2
a3	b2	c1

Ví dụ khác: *Select* MaSP, TenSP, SoTon *From* SanPham *WHERE* SoTon <= 10

Mệnh đề Where

❑ Các phép so sánh (comparison): <, <=, !<, >, >=, !>, =, <> hay !=

- Ví dụ: Xét lược đồ quan hệ

SINHVIEN(MASV, HOSV, TENS, PHAINU, NGAYSINH, QQUN, MANGANH)

Viết lệnh SQL lấy danh sách gồm: mã số, họ, tên, quê quán của các sinh viên ngành 'TH'

- Biểu thức đại số quan hệ:

$$\Pi_{\text{MASV, HOSV, TENS, QQUN}}(\sigma_{\text{MANGANH='TH'}}(\text{SINHVIEN}))$$

- Lệnh SQL:

SELECT MASV, HOSV, TENS, QQUN **FROM** SINHVIEN

WHERE MANGANH='TH'

SINHVIEN

MASV	HOSV	TENS	PHAINU	NGAYSINH	QQUN	MANGANH
100	Lê Văn	An	False	060687	DN	DY
101	Lương Ngọc Trường	Thủy	True	140488	CM	TH
103	Lý Thúy	Nga	True	240288	CM	TH
104	Mai Nhựt	An	False	251188	LA	TH
106	Nguyễn Thị Thúy	Anh	True	060688	BT	DY
108	Ngô Thị Trường	Hương	True	140488	TN	QT
109	Ngô Thuý	Lê	True	280588	TG	QT

Kết quả

MASV	HOSV	TENS	QQUN
101	Lương Ngọc Trường	Thủy	CM
103	Lý Thúy	Nga	CM
104	Mai Nhựt	An	LA

Mệnh đề Where

❑ Các phép toán logic: **AND, OR, NOT**

Xét lược đồ quan hệ: **NhanVien**(MANV, HO, TEN, NS, GT, LCB, MAPB)

Viết lệnh SQL lấy ra danh sách nhân viên có giới tính là “Nam” và có “LCB > 200”

SELECT *

FROM NhanVien

WHERE GT='Nam' **AND** LCB>200

NhanVien

MANV	HO	TEN	NS	GT	LCB	MAPB
01	Trần Thị	Vỹ	1970	Nữ	300	PKD
02	Võ Văn	An	1975	Nam	250	PTV
03	Lệ Thị	Bình	1982	Nữ	400	PKD
04	Võ Tấn	Viên	1970	Nam	350	PTV
05	Nguyễn Tấn	An	1985	Nam	200	PKD
06	Cao Văn	An	1968	Nam	250	PKH

Kết quả

MANV	HO	TEN	NS	GT	LCB	MAPB
02	Võ Văn	An	1975	Nam	250	PTV
04	Võ Tấn	Viên	1970	Nam	350	PTV
06	Cao Văn	An	1968	Nam	250	PKH

Mệnh đề Where

❑ Các phép toán logic: **AND, OR, NOT**

– Ví dụ: Xét lược đồ quan hệ: KETQUA(MASV, MAMH, LANTHI, DIEM)

Cho xem điểm thi lần một môn CSDL1 của các sinh viên, kết quả gồm: MASV, DIEM

– Biểu thức đại số quan hệ:

$$\Pi_{\text{MASV, DIEM}} (\sigma_{\text{MAMH}='CSDL1' \wedge \text{LANTHI}=1}(\text{KETQUA}))$$

– Lệnh SQL:

Select MaSV, Diem **From** KetQua **Where** MaMH = 'CSDL1' **AND** LanThi=1

KETQUA

MASV	MAMH	LANTHI	DIEM
100	T001	1	7
103	CSDL1	1	3
103	CSDL1	2	5
103	T001	1	3
104	CSDL1	1	6
104	T001	1	4
104	T001	2	6

Kết quả

MASV	DIEM
103	3
104	6

Mệnh đề Where

❑ Kiểm tra giá trị rỗng: <biểu thức> **IS [NOT] NULL**

- Kiểm tra giá trị rỗng:

<biểu thức> **IS NULL**

- Kiểm tra giá trị khác rỗng:

<biểu thức> **IS NOT NULL**

Ví dụ: Cho biết danh sách nhân viên chưa cung cấp số điện thoại

```
SELECT * FROM NhanVien WHERE DienThoai IS NULL
```

Chú ý: Không sử dụng phép so sánh = NULL hay <> NULL

Mệnh đề Where

❑ Kiểm tra giá trị rỗng: <biểu thức> **IS [NOT] NULL**

Cho biết thông tin về các hóa đơn có ngày thanh toán rỗng – chưa có thông tin về ngày thanh toán (NGAYTT)

SELECT *

FROM HOADON

WHERE **NGAYTT IS NULL**

SOHD	NGAYHD	MAKH	MANV	NGAYTT
HD01	05-01-2005	GDTXPY	NV3	07-01-2005
HD02	05-01-2005	CTCN02	NV3	Null
HD03	06-01-2005	KVL	NV2	08-01-2005
HD04	08-01-2005	GDTXPY	NV1	Null
HD05	08-01-2005	KVL	NV1	10-01-2005
HD06	10-01-2005	GDTXPY	NV3	Null
HD07	12-01-2005	CTCN02	NV4	Null
HD08	25-01-2005	GDTXPY	NV2	Null
HD09	03-02-2005	CTCN02	NV2	10-02-2005
HD10	05-02-2005	CTCN01	NV3	Null
HD11	10-01-2005	CTCN02	NV4	Null
HD12	23-01-2005	CTCN02	NV1	23-01-2005



SOHD	NGAYHD	MAKH	MANV	NGAYTT
HD02	05-01-2005	CTCN02	NV3	Null
HD04	08-01-2005	GDTXPY	NV1	Null
HD06	10-01-2005	GDTXPY	NV3	Null
HD07	12-01-2005	CTCN02	NV4	Null
HD08	25-01-2005	GDTXPY	NV2	Null
HD10	05-02-2005	CTCN01	NV3	Null
HD11	10-01-2005	CTCN02	NV4	Null

Mệnh đề Where

❑ Kiểm tra thuộc tập giá trị: <biểu thức> [NOT] IN (<danh sách giá trị>)

Chức năng: kiểm tra giá trị biểu thức có bằng một trong các giá trị của tập giá trị cho trước hay không.

- Kiểm tra biểu thức thuộc danh sách giá trị:
<biểu thức> IN (<danh sách giá trị>)
- Kiểm tra biểu thức không thuộc danh sách giá trị:
<biểu thức> NOT IN (<danh sách giá trị>)
- Ví dụ: Xem danh sách sinh viên có mã số '101', '102', '103'

```
SELECT * FROM SinhVien WHERE MaSV IN ('101', '102', '103')
```

Mệnh đề Where

❑ Thuộc tập giá trị: <biểu thức> [NOT] IN (<danh sách giá trị>)

Chức năng: kiểm tra giá trị biểu thức có bằng một trong các giá trị của tập giá trị cho trước hay không.

SINHVIEN

	MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH
+	100	Lê Văn	An	False	060687	DN	DY
+	101	Lương Ngọc Trường	Thúy	True	140488	CM	TH
+	103	Lý Thúy	Nga	True	240288	CM	TH
+	104	Mai Nhựt	An	False	251188	LA	TH
+	106	Nguyễn Thị Thúy	Anh	True	060688	BT	DY
+	108	Ngô Thị Trường	Hương	True	140488	TN	QT
+	109	Ngô Thuý	Lê	True	280588	TG	QT

MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH	MALOP
101	Lương Ngọc Trường	Thúy	True	140488	CM	TH	DH04KT
103	Lý Thúy	Nga	True	240288	CM	TH	DH05KT
108	Ngô Thị Trường	Hương	True	140488	TN	QT	DH04QL
109	Ngô Thuý	Lê	True	280588	TG	QT	DH05QL

Ví dụ: Cho biết danh sách sinh viên thuộc các tỉnh CM, TN, TG.

SELECT * FROM SINHVIEN WHERE QQUAN IN ('CM', 'TN', 'TG')

Mệnh đề Where

❑ Thuộc miền giá trị : <biểu thức> **[NOT] BETWEEN** <Min> **AND** <Max>

Chức năng: kiểm tra giá trị biểu thức có thuộc phạm vi miền giá trị [Min, Max] hay không.

– Biểu thức thuộc miền giá trị [Min, Max]

<biểu thức> **BETWEEN** <Min> **AND** <Max>

– Biểu thức không thuộc miền giá trị [Min, Max]

<biểu thức> **NOT BETWEEN** <Min> **AND** <Max>

Ví dụ: Liệt kê danh sách các sản phẩm có đơn giá từ 100 đến 300

SELECT * FROM SanPham **WHERE** DonGia **BETWEEN** 100 **AND** 300

Mệnh đề Where

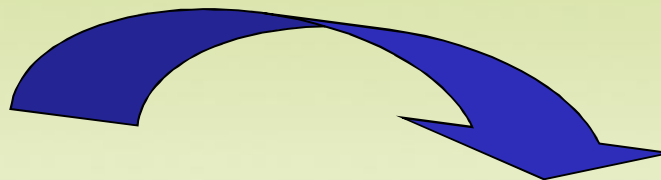
❑ Thuộc miền giá trị : <biểu thức> [NOT] BETWEEN <Min> AND <Max>

Chức năng: kiểm tra giá trị biểu thức có thuộc phạm vi miền giá trị [Min, Max] hay không.

Ví dụ: Lấy các bộ trong quan hệ KETQUA có điểm môn CSDL1 từ 5 đến 7.

KETQUA

MASV	MAMH	LAN THI	DIEM
100	T001	1	7
103	CSDL1	1	3
103	CSDL1	2	5
103	T001	1	3
104	CSDL1	1	6
104	T001	1	4
104	T001	2	6



MASV	MAMH	LAN THI	DIEM
103	CSDL1	2	5.0
104	CSDL1	1	6.0

Select * From KETQUA Where MAMH='CSDL1' And DIEM Between 5 And 7

Mệnh đề Where

❑ Đối sánh mẫu : <biểu thức> **[NOT] LIKE** 'Mẫu văn bản'

Chức năng: là phép so sánh giá trị biểu thức tương đồng với 'Mẫu văn bản'. 'Mẫu văn bản' là một giá trị kiểu *chuỗi* thường chứa các ký tự đại diện, 02 ký tự đặc biệt:

_ : Đại diện cho một ký tự bất kỳ tại vị trí đặt ký tự.

% : Đại diện cho một nhóm ký tự bất kỳ tại vị trí đó.

Ngoài ra:

[] : Ký tự đơn bất kỳ trong giới hạn được chỉ định (ví dụ [a-f] hay một tập [abcdef])

[^] : Ký tự đơn bất kỳ không nằm trong giới hạn được chỉ định (ví dụ [^a-f] hay một tập [^abcdef]).

Mệnh đề Where

❑ Đối sánh mẫu : <biểu thức> **[NOT] LIKE** 'Mẫu văn bản'

Chức năng: là phép so sánh giá trị biểu thức tương đồng với 'Mẫu văn bản'. 'Mẫu văn bản' là một giá trị kiểu *chuỗi* thường chứa các ký tự đại diện

Cho biết danh sách sinh viên họ 'Tran'

SINHVIEN				
MASV	HOSV	TENSV	MANG	NS
D0912	Tran Thi Y	Lan	CNTT	1980
D0913	Nguyen Van	Tan	KT	1981
D0801	Le Thi Bich	Ngoc	CNTT	1979
D0802	Tran Van	Bay	KT	1978
D0803	Tran Le	Khang	XD	1980
C0801	Nguyen Tuan	Canh	XD	1980



MASV	HOSV	TENSV	MANG	NS
D0912	Tran Thi Y	Lan	CNTT	1980
D0802	Tran Van	Bay	KT	1978
D0803	Tran Le	Khang	XD	1980

Select *

From SINHVIEN

Where HO **LIKE** 'Tran%'

Mệnh đề Where

❑ Đối sánh mẫu : <biểu thức> **[NOT] LIKE** 'Mẫu văn bản'

Chức năng: là phép so sánh giá trị biểu thức tương đồng với 'Mẫu văn bản'. 'Mẫu văn bản' là một giá trị kiểu *chuỗi* thường chứa các ký tự đại diện.....

Ví dụ: Xét lược đồ quan hệ: NHANVIEN(MANV, TENNV, DIACHI)

Lấy danh sách các nhân viên có kí tự thứ 3 trong mã nhân viên là 'H' và ký tự thứ 5 trong mã nhân viên là 8

```
SELECT *  
FROM NHANVIEN  
WHERE MANV LIKE '__H_8%'
```

Mệnh đề Where

- ❑ Mệnh đề **Where** <điều kiện chọn bộ> dùng chọn ra những bộ thỏa mãn điều kiện. <điều kiện chọn bộ> được thiết lập bao gồm:
 - Các phép so sánh (comparison): <, <=, !<, >, >=, !>, =, <> hay !=
 - Các phép toán logic: AND, OR, NOT
 - Kiểm tra giá trị rỗng: <biểu thức> IS [NOT] NULL
 - Kiểm tra thuộc tập giá trị: <biểu thức> [NOT] IN (<danh sách giá trị>)
 - Kiểm tra thuộc miền giá trị: <biểu thức> [NOT] BETWEEN <Min> AND <Max>
 - Đối sánh mẫu : <biểu thức> [NOT] LIKE 'Mẫu văn bản'

Các truy vấn lồng nhau (sử dụng kết quả của một truy vấn làm điều kiện chọn bộ - SubQuery)

- Thuộc tập SubQuery: <biểu thức> [NOT] IN (SubQuery)
- So sánh hơn với tập SubQuery: <biểu thức> <comparison> ANY (subquery)
- So sánh nhất với tập SubQuery : <biểu thức> <comparison> ALL (subquery)
- Kiểm tra tồn tại: [NOT] EXISTS(SubQuery): Trả về True nếu tồn tại ít nhất 1 mẫu tin trong SubQuery, ngược lại trả về False

Mệnh đề Where

❑ Thuộc tập SubQuery: <biểu thức> [NOT] IN (SubQuery)

Chức năng: kiểm tra giá trị biểu thức có bằng một trong các giá trị của tập giá trị trả về bởi SubQuery hay không. SubQuery là một câu truy vấn SQL con.

Ví dụ: Xét SINHVIEN(MASV, HOSV, TENS, MANG,...)

 KETQUA(MASV, MAMH, LANTHI, DIEM)

Viết truy vấn lấy mã số, họ, tên của các sinh viên đã có điểm thi môn ‘CSDL1’

SELECT MASV, HOSV, TENS

FROM SINHVIEN

WHERE MASV **IN** (**SELECT** MASV

FROM KETQUA

WHERE MAMH=‘CSDL1’)

Mệnh đề Where

❑ So sánh hơn với tập SubQuery: <biểu thức> <comparison> **ANY** (subquery)

Chức năng: kiểm tra giá trị biểu thức thỏa phép so sánh với ít nhất 1 giá trị tạo bởi subquery.

Ví dụ: Xét SanPham(MaSP, TenSP, DVT, DonGia, SoTon)

Cho biết danh sách các sản phẩm có đơn giá không là đơn giá nhỏ nhất.

Phân tích:

- Tìm đơn giá của tất cả các sản phẩm (SubQuery)
- Tìm những sản phẩm có đơn giá lớn hơn ít nhất đơn giá của một sản phẩm nào đó trong tập kết quả trả về bởi SubQuery.

```
Select A.* From SanPham As A Where A.DonGia > Any ( Select Distinct B.DonGia  
From SanPham As B )
```

Mệnh đề Where

❑ So sánh hơn với tập SubQuery: <biểu thức> <comparison> **ANY** (subquery)

Chức năng: kiểm tra giá trị biểu thức thỏa phép so sánh với ít nhất 1 giá trị tạo bởi subquery.

Ví dụ: Xét 2 quan hệ r và s

r	
A	B
a	5
b	3
c	9

s	
C	D
2	1
8	6
3	5
4	4

Liệt kê các bộ của quan hệ r mà có giá trị cột B nhỏ hơn bất kỳ giá trị nào của cột C trong quan hệ s

Select * **From** r **Where** B < **Any** (**Select** C **From** s)

Kết quả

A	B
a	5
b	3

Mệnh đề Where

❑ So sánh nhất với tập SubQuery : <biểu thức> <comparison> **ALL** (subquery)

Chức năng: kiểm tra giá trị biểu thức thỏa phép so sánh với tất cả các giá trị trong subquery.

Ví dụ: Xét SanPham(MaSP, TenSP, DVT, DonGia, SoTon)

Cho biết danh sách các sản phẩm có đơn giá lớn nhất

Phân tích:

- Tìm đơn giá của tất cả các sản phẩm (SubQuery)
- Tìm những sản phẩm có đơn giá lớn hơn bằng tất cả đơn giá của các sản phẩm trả về bởi SubQuery.

```
Select A.* From SanPham As A Where A.DonGia>=All ( Select Distinct B.DonGia  
From SanPham As B )
```

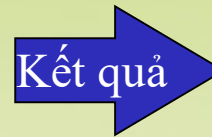
Mệnh đề Where

❑ So sánh nhất với tập SubQuery : <biểu thức> <comparison> **ALL** (subquery)

Chức năng: kiểm tra giá trị biểu thức thỏa phép so sánh với tất cả các giá trị trong subquery.

Ví dụ: Xét quan hệ: NhanVien

MaNV	TenNV	NamSinh
T1	Huong	1970
T2	Hùng	1975
T3	Hoa	1980
T4	Dũng	1973
T5	Cường	1982



MaNV	TenNV	Namsinh
T1	Huong	1970

Cho biết nhân viên lớn tuổi nhất (có năm sinh nhỏ nhất)?

Select *

From NhanVien

Where NamSinh <= **All** (**Select** NamSinh
From NhanVien)

Mệnh đề Where

❑ So sánh nhất với tập SubQuery : <biểu thức> <comparison> **ALL** (subquery)

Chức năng: kiểm tra giá trị biểu thức thỏa phép so sánh với tất cả các giá trị trong subquery.

Ví dụ: Xét quan hệ KetQua

MASV	MAMH	LANTHI	DIEM
100	T001	1	7
103	CSDL1	1	3
103	CSDL1	2	5
103	T001	1	3
104	CSDL1	1	6
104	T001	1	4
104	T001	2	6

Cho biết mã số sinh viên, điểm của các sinh viên có điểm thi lần 1 môn CSDL1 cao nhất.

Select MASV, DIEM

From KetQua

Where MAMH = 'CSDL1' **And** LANTHI = 1 **And** DIEM >= **All** (**Select** DIEM
From KetQua **Where** MAMH = 'CSDL1' **And** LANTHI = 1)

Mệnh đề Where

❑ Kiểm tra tồn tại: **[NOT] EXISTS** (SubQuery): Trả về **True** nếu tồn tại ít nhất 1 mẫu tin trong SubQuery, ngược lại trả về **False**

Chức năng: kiểm tra sự tồn tại, trả về **True** nếu có tồn tại, ngược lại **False**

Ví dụ: Xét SINHVIEN(MASV, HOSV, TENSX, MANG,...)

 KETQUA(MASV, MAMH, LANTHI, DIEM)

Lấy danh sách sinh viên đã có điểm môn CSDL1

```
SELECT *  
FROM SINHVIEN AS SV  
WHERE EXISTS ( SELECT *  
                FROM KETQUA AS KQ  
                WHERE MAMH='CSDL1' AND  
                KQ.MASV = SV.MASV )
```

Mệnh đề Where

❑ Kiểm tra tồn tại: **[NOT] EXISTS** (SubQuery): Trả về **True** nếu tồn tại ít nhất 1 mẫu tin trong SubQuery, ngược lại trả về **False**

Chức năng: kiểm tra sự tồn tại, trả về **True** nếu có tồn tại, ngược lại **False**

Ví dụ: Xét SanPham(MaSP, TenSP, DVT, DonGia, SoTon)

CTHD(MaHD, MaSP, SL, DG)

Liệt kê danh sách sản phẩm không bán được

```
Select A.* From SanPham As A Where Not Exists ( Select B.*  
From CTHD As B Where A.MaSP = B.MaSP )
```


MỆNH ĐỀ GROUP BY VÀ PHÁT BIỂU HAVING

Mệnh đề Group By và Phát biểu Having

- ❑ Mệnh đề **Group By** <các thuộc tính phân nhóm> dùng để tổng hợp, thống kê dữ liệu theo từng nhóm dựa trên giá trị của **các thuộc tính phân nhóm**. Mỗi nhóm sẽ cho ra một bộ trong quan hệ kết quả. Nếu muốn chọn ra những nhóm thỏa theo điều kiện thì sử dụng **phát biểu Having**
- ❑ Phát biểu **Having** dùng chọn ra những bộ thỏa điều kiện (điều kiện chọn nhóm)

```
Select A1, A2,...An, F(...)  
From .....  
[Where <điều kiện chọn dòng>]  
Group By A1, A2,...An  
[Having <điều kiện chọn nhóm>]
```

Trong đó:

- A₁, A₂,...A_n : Các thuộc tính phân nhóm
- F(...) : Hàm tổng hợp dữ liệu theo nhóm
- Group By A₁, A₂,...A_n : Phân nhóm theo các thuộc tính A₁, A₂,...A_n
- Having ... Chọn các nhóm kết quả thỏa theo điều kiện.

Mệnh đề Group By và Phát biểu Having

- ❑ Phân biệt *điều kiện chọn dòng* và *điều kiện chọn nhóm*

```
Select A1, A2,...An, F(...)  
From .....  
[Where <điều kiện chọn dòng>]  
Group By A1, A2,...An  
[Having <điều kiện chọn nhóm>]
```

Cần hiểu: “Chọn ra các bộ thỏa điều kiện trong mệnh đề **Where**, kết quả sau đó được phân nhóm theo mệnh đề **Group by**, chỉ những nhóm nào thỏa điều kiện trong phát biểu **Having** mới được giữ lại trong kết quả”.

Mệnh đề Group By và Phát biểu Having

❑ Các hàm tổng hợp dữ liệu theo nhóm:

- **Avg**(<thuộc tính>): Tính trung bình cộng giá trị *thuộc tính* kiểu số trên các bộ cùng nhóm.
- **Sum**(<thuộc tính>): Tính tổng giá trị *thuộc tính* kiểu số trên các bộ cùng nhóm.
- **Min**(<thuộc tính>): chọn ra giá trị *thuộc tính* nhỏ nhất trên các bộ cùng nhóm.
- **Max**(<thuộc tính>): chọn ra giá trị *thuộc tính* lớn nhất trên các bộ cùng nhóm.
- **Count**(*|<thuộc tính>|**distinct** <thuộc tính>): Đếm các bộ trong quan hệ:
 - **Count**(*): Đếm tất cả các bộ kể cả những bộ có *thuộc tính* là NULL
 - **Count**(<thuộc tính>): Chỉ đếm những bộ mà giá trị *thuộc tính* khác NULL
 - **Count**(**distinct** <thuộc tính>): Chỉ đếm những bộ mà giá trị của *thuộc tính* khác NULL và mỗi nhóm những bộ có giá trị *thuộc tính* giống nhau thì chỉ đếm 1 lần.

Mệnh đề Group By và Phát biểu Having

Ví dụ: xét HoaDon(MaHD, NgayLHD, NgayGiao, MaKH, MaNV)

CTHD(MaHD, MaSP, SL, DG)

- Đếm tổng số hóa đơn của từng khách hàng.

```
Select MaKH, Count(MaHD) As SoLuongHD From HoaDon Group By MaKH
```

- Liệt kê danh sách hóa đơn và số tiền của từng hóa đơn.

```
Select MaHD, Sum(SL*DG) As ThanhTien From CTHD Group By MaHD
```

- Liệt kê các hóa đơn có tiền hóa đơn lớn hơn 500000

```
Select MaHD, Sum(SL*DG) As ThanhTien
```

```
From CTHD Group By MaHD
```

```
Having Sum(SL*DG) > 500000
```

Mệnh đề Group By và Phát biểu Having

Ví dụ: Xét quan hệ ChiTietPhieuNhap

SOPN	MAMH	SLNHAP	DGNHAP
N001	EIT865	15	940000
N001	MAP4SX	20	950000
N003	CPUCLR12	50	600000
N003	EIT865	10	940000
N003	MAP4SX	30	970000
N004	MAP4SX	35	1000000

Cho biết tổng số lượng đã nhập của các mặt hàng

Select MAMH, **SUM**(SLNHAP) **As** TONGNHAP

From ChiTietPhieuNhap

Group By MAMH

MAMH	TONGNHAP
CPUCLR12	50
EIT865	25
MAP4SX	85

Lưu ý: Khi có **Group By**, một thuộc tính sau **Select** hoặc phải đặt sau **Group By** hoặc phải đặt trong một hàm thống kê.

Mệnh đề Group By và Phát biểu Having

Ví dụ: cho biết các ngành có số lượng sinh viên nữ lớn hơn 1

Select MaNghanh, Count(*) **As** SONU

From SinhVien

Where PHAINU = True

Group By MANGANH

Having Count(*) > 1

MANGANH	SONU
QT	2
TH	2

Select MaNghanh, Count(*) **As** SONU

From SinhVien

MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH
100	Lê Văn	An	False	060687	DN	DY
101	Lương Ngọc Trường	Thủy	True	140488	CM	QT
103	Lý Thúy	Nga	True	240288	CM	TH
104	Mai Nhut	An	False	251188	LA	TH
106	Nguyễn Thị Thúy	Anh	True	060688	BT	DY
108	Ngô Thị Trường	Hương	True	140488	TN	QT
109	Ngô Thuý	Lê	True	280588	TG	TH

Where PHAINU = True

MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH
101	Lương Ngọc Trường	Thủy	True	140488	CM	QT
103	Lý Thúy	Nga	True	240288	CM	TH
106	Nguyễn Thị Thúy	Anh	True	060688	BT	DY
108	Ngô Thị Trường	Hương	True	140488	TN	QT
109	Ngô Thuý	Lê	True	280588	TG	TH

Group By MANGANH

MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH
106	Nguyễn Thị Thúy	Anh	True	060688	BT	DY
108	Ngô Thị Trường	Hương	True	140488	TN	QT
101	Lương Ngọc Trường	Thủy	True	140488	CM	QT
109	Ngô Thuý	Lê	True	280588	TG	TH
103	Lý Thúy	Nga	True	240288	CM	TH

Having Count(*) > 1

MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH
108	Ngô Thị Trường	Hương	True	140488	TN	QT
101	Lương Ngọc Trường	Thủy	True	140488	CM	QT
109	Ngô Thuý	Lê	True	280588	TG	TH
103	Lý Thúy	Nga	True	240288	CM	TH

MỆNH ĐỀ ORDER BY ...

Mệnh đề Order By...

- ❑ Mệnh đề [**ORDER BY** <thuộc tính 1> [**ASC** | **DESC**] [...]] sắp xếp các bộ theo thứ tự tăng hay giảm dần của các giá trị trên <thuộc tính 1>,....

Trong đó:

- **ASC** : Sắp tăng trên <thuộc tính>, là kiểu sắp xếp mặc định.
- **DESC** : Sắp giảm trên <thuộc tính>, phải chỉ định.
- Thứ tự sắp xếp được ưu tiên từ trái sang phải.

Ví dụ: Xét quan hệ SINHVIEN(MASV, HOSV, TENSX, MANG,...)

Hiển thị danh sách sinh viên, sắp xếp tăng dần theo tên sinh viên, nếu cùng tên thì sắp xếp giảm dần theo họ sinh viên.

```
Select *  
From Sinhvien  
Order By TENSX, HOSX DESC
```

THỨ TỰ BIÊN DỊCH LỆNH SELECT

Thứ tự biên dịch của lệnh Select

✂ Thứ tự dịch của lệnh Select:

FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

SELECT < danh sách thuộc tính >

[INTO *new_table*]

[FROM < bảng nguồn >]

[WHERE < điều kiện chọn dòng >]

[GROUP BY < danh sách thuộc tính phân nhóm >]

[HAVING < điều kiện chọn nhóm >]

[ORDER BY < danh sách thuộc tính sắp xếp > [ASC | DESC]]

[UNION | INTERSECT | EXCEPT < câu truy vấn khác >]

Thứ tự biên dịch của lệnh Select

From

PhieuXuat

	SOPX	NGAYX	DIENGIAI
+	X001	02/01/07	Xuất bán cho Ô A
+	X002	05/01/07	Xuất bán cho Ô B
+	X003	07/01/07	Xuất bán cho Cty R
+	X004	09/01/07	Xuất bán cho Ô C

ChiTiet PX

	SOPX	MAMH	SLXUAT	DGXUAT
	X001	MAP4SX	5	980000
	X002	CPUCLR12	20	610000
	X002	EIT865	15	980000
	X002	MAP4SX	3	1000000
	X003	CPUCLR12	1	600000
	X003	MAP4SX	2	260000

Where...

SOPX	NGAYX	MAMH	SLXUAT	DGXUAT
X001	02/01/07	MAP4SX	5	980000
X002	05/01/07	MAP4SX	3	1000000
X002	05/01/07	EIT865	15	980000
X002	05/01/07	CPUCLR12	20	610000
X003	07/01/07	MAP4SX	2	260000
X003	07/01/07	CPUCLR12	1	600000

Having.....

SOPX	NGAYX	MAMH	SLXUAT	DGXUAT
X003	07/01/07	CPUCLR12	1	600000
X002	05/01/07	CPUCLR12	20	610000
X002	05/01/07	EIT865	15	980000

Group By ...

SOPX	NGAYX	MAMH	SLXUAT	DGXUAT
X003	07/01/07	CPUCLR12	1	600000
X002	05/01/07	CPUCLR12	20	610000
X002	05/01/07	EIT865	15	980000
X003	07/01/07	MAP4SX	2	260000
X002	05/01/07	MAP4SX	3	1000000
X001	02/01/07	MAP4SX	5	980000

Select ,...

MAMH	TRIGIAN
CPUCLR12	12800000
EIT865	14700000

TRUY VẤN VỚI CÁC PHÉP TOÁN TẬP HỢP

Các phép toán tập hợp

❑ Phép hợp (Union): Cấu trúc câu lệnh

(*<câu lệnh truy vấn 1>*)
Union
(*<câu lệnh truy vấn 2>*)

- Mệnh đề **Select** trong hai câu lệnh truy vấn phải cùng số cột, cùng kiểu dữ liệu.
- Bảng kết quả có tên cột được tạo từ câu truy vấn đầu tiên.

Ví dụ: (Select B,C From r1)
Union
(Select B, C From r2)


Biểu thức đại số quan hệ tương ứng
 $\Pi_{B,C}(r_1) \cup \Pi_{B,C}(r_2)$

r_1

B	C	D	E
2	4	8	7
3	3	8	5
6	1	3	11
5	2	0	8

r_2

B	C	F
12	15	21
19	16	45
22	17	11



B	C
2	4
3	3
6	1
5	2
12	15
19	16
22	17

Các phép toán tập hợp

- ❑ **Phép giao (Intersect):** Cấu trúc câu lệnh

(<câu lệnh truy vấn 1>)

Intersect

(<câu lệnh truy vấn 2>)

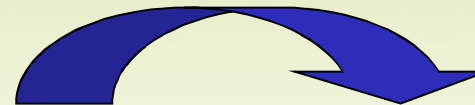
Trả về một bảng chứa các mẫu tin cùng xuất hiện trong kết quả của hai câu truy vấn 1 và 2

Ví dụ :
(Select A,B From r₁)
Intersect
(Select A, B From r₂)

Biểu thức đại số quan hệ tương ứng

$$\Pi_{A,B}(r_1) \cap \Pi_{A,B}(r_2)$$

r ₁				r ₂		
A	B	C	D	A	B	E
a	x	1	5	d	y	4
d	y	2	7	a	y	3
a	y	3	4	b	z	6
b	x	3	3			



A	B
d	y
a	y

Các phép toán tập hợp

❑ Phép trừ (Except): Cấu trúc câu lệnh

(*<câu lệnh truy vấn 1>*)
Except
(*<câu lệnh truy vấn 2>*)

Trả về một bảng chứa các mẫu tin thuộc kết quả *câu lệnh truy vấn thứ 1* mà không thuộc kết quả *câu lệnh truy vấn thứ hai*.

Ví dụ: (Select B,C From r_1)
Except
(Select B,C From r_2)


Biểu thức đại số quan hệ tương ứng
 $\Pi_{B,C}(r_1) - \Pi_{B,C}(r_2)$

r_1

A	B	C	D
a	4	3	7
a	5	1	8
c	1	4	3
b	3	1	6

r_2

B	C	E
4	3	9
1	4	7
4	3	4



B	C
5	1
3	1

Các phép toán tập hợp

Ví dụ: Cho biết danh sách mã số các sinh viên có điểm thi cả 2 môn T001 và CSDL1.

KetQua

MASV	MAMH	LAN THI	DIEM
100	T001	1	7
103	CSDL1	1	3
103	CSDL1	2	5
103	T001	1	3
104	CSDL1	1	6
104	T001	1	4
104	T001	2	6



masv
103
104

(Select Distinct MASV From KetQua Where MAMH='T001')

Intersect

(Select Distinct MASV From KetQua Where MAMH='CSDL1')

Các phép toán tập hợp

Ví dụ: Xét 2 quan hệ

SinhVien

	MASV	HOSV	TENSV	PHAINU	NGAYSINH	QQUAN	MANGANH	KHOAHOC
+	100	Lê Văn	An	False	060687	DN	DY	04
+	101	Lương Ngọc Trường	Thủy	True	140488	CM	TH	04
+	103	Lý Thúy	Nga	True	240288	CM	TH	05
+	104	Mai Nhựt	An	False	251188	LA	TH	04
+	106	Nguyễn Thị Thúy	Anh	True	060688	BT	DY	04
+	108	Ngô Thị Trường	Hương	True	140488	TN	QT	04
+	109	Ngô Thuý	Lê	True	280588	TG	QT	05

KetQua

	MASV	MAMH	LANTHI	DIEM
	100	T001	1	7
	103	CSDL1	1	3
	103	CSDL1	2	5
	103	T001	1	3
	104	CSDL1	1	6
	104	T001	1	4
	104	T001	2	6

Cho biết mã số các sinh viên ngành 'TH' khóa 04 chưa có điểm môn CSDL1

– Biểu thức đại số quan hệ:

$$\Pi_{\text{MASV}}(\sigma_{\text{MANGANH}='TH' \wedge \text{KHOAHOC}='04'}(\text{SinhVien})) - \Pi_{\text{MASV}}(\sigma_{\text{MAMH}='CSDL1'}(\text{KetQua}))$$

– Lệnh SQL:

(Select MASV From SinhVien Where MANGANH='TH' And KHOAHOC='04')

Except

(Select MASV From Ketqua Where MAMH='CSDL1')



TRUY VẤN LÒNG

Truy vấn lồng

- ❑ Với phép giao có biểu thức đại số quan hệ là:

$$\Pi_{A,B}(r_1) \cap \Pi_{A,B}(r_2)$$

Câu lệnh SQL có thể viết dưới dạng:

Select

From r_1 As t

Where ... Exists (Select *

From r_2

Where $r_2.A=t.A$ and $r_2.B=t.B$)

Truy vấn lồng

Ví dụ: Từ CSDL “QLSV” đã cho, cho biết danh sách mã số các sinh viên có điểm thi cả 2 môn T001 và CSDL1

Select MASV

From KetQua as t

Where MAMH='CSDL1' And Exists (Select *

From KetQua

Where KetQua.MAMH='T001' And

t.MASV=KetQua.MASV

)

Truy vấn lồng

- ❑ Với phép trừ có biểu thức đại số quan hệ:

$$\Pi_{B,C}(r_1) - \Pi_{B,C}(r_2)$$

Câu lệnh SQL có thể viết:

Select B, C

From r_1 as t

Where Not Exists (Select *

From r_2

Where $t.B=r_2.B$ and $t.C=r_2.C$)

Truy vấn lồng

Ví dụ: Từ CSDL “QLSV”, cho biết danh sách mã số các sinh viên *ngành TH khóa 04* chưa có điểm môn CSDL1

Select MASV

From SinhVien As sv

Where MANGANH='TH' And KHOAHOC='04' And

Not Exists (Select *

From KetQua

Where sv.MASV=KetQua.MASV And MAMH='CSDL1')

BÀI TẬP