



# HỌC VIỆN KỸ THUẬT MẬT MÃ

## KHOA CÔNG NGHỆ THÔNG TIN

### Bài giảng: Truy vấn dữ liệu nâng cao



“Tri thức là chìa khóa mở cánh cửa tương lai”



# Truy vấn trong cơ sở dữ liệu

**Truy vấn trên một bảng**

**Truy vấn trên nhiều bảng**



# Truy vấn trên nhiều bảng

## 1. Các phép nối

- ✓ Kết nối bằng mệnh đề WHERE
- ✓ Kết bằng mệnh đề JOIN

## 2. Truy vấn con

- ✓ Truy vấn con nằm trong mệnh đề SELECT
- ✓ Truy vấn con nằm trong mệnh đề FROM
- ✓ Truy vấn con nằm trong mệnh đề WHERE

## 3. Các phép toán tập hợp

- ✓ Hợp (UNION)
- ✓ Giao (INTERSECT)
- ✓ Trừ (EXCEPT)



# Kết nối sử dụng mệnh đề WHERE

➤ **Cú pháp** `SELECT bang1.cot1, bang2.cot2,...`  
`FROM bang1, bang2,...`  
`WHERE <điều kiện kết nối các bảng>, đk chặt lọc dữ liệu`

Bảng “*Sinh viên*”

	MaSV	HotenSV	Gioitinh	Lop	Quequan	Ngaysinh	Sdt	Email
1	AT1	Cao Thu Huyền	Nữ	AT16A	Hà Nội	1998-10-10	0987878909	abc@gmail.com
2	AT2	Nguyễn Thị Hải	Nữ	AT16K	Hà Nam	1998-11-15	0987845609	cgd@gmail.com
3	CN1	Trần Mạnh Cường	Nam	CT4A	Thái Bình	1998-12-01	0867878909	123@gmail.com
4	CN2	Lê Văn Minh	Nam	CT4A	Thanh Hóa	1998-09-20	0977878809	sutu@gmail.com
5	DT1	Nguyễn Bảo lâm	Nam	DT3A	Hà Nội	1998-05-12	0677878909	khk@gmail.com
6	DT3	Vũ Tuấn Đạt	Nam	DT3D	Hà Nam	1998-09-13	0677878959	cao@gmail.com

Bảng “*Môn học*”

	mamh	tenmh	DVHP
1	CSDL	Cơ sở dữ liệu	3
2	HTMT	Hệ thống máy tính	3
3	MMT	Mạng máy tính	4
4	TCC	Toán cao cấp	2
5	TCS	Tin học cơ sở	5

Bảng “*Kết quả*”

	masv	mamh	diem
1	AT1	CSDL	8
2	AT1	HTMT	9
3	AT1	MMT	7
4	AT1	TCS	9
5	AT2	CSDL	4
6	AT2	HTMT	4
7	AT2	MMT	5
8	AT2	TCC	6
9	AT2	TCS	5
10	CN1	CSDL	9
11	CN1	HTMT	9
12	CN1	MMT	10



# Kết nối sử dụng mệnh đề WHERE

- **Ví dụ 1** Làm thế nào để lấy ra thông tin các sinh viên đã thi ít nhất 1 môn nào đó?

```
SELECT SV.Masv, HotenSV, Lop
FROM Sinhvien Sv, Ketqua Kq
WHERE Sv.Masv = Kq.MaSV
```

```
SELECT DISTINCT SV.Masv, HotenSV, Lop
FROM Sinhvien Sv, Ketqua Kq
WHERE Sv.Masv = Kq.MaSV
```

➤ **Kết quả**

	Masv	HotenSV	Lop
1	AT1	Cao Thu Huyền	AT16A
2	AT1	Cao Thu Huyền	AT16A
3	AT1	Cao Thu Huyền	AT16A
4	AT1	Cao Thu Huyền	AT16A
5	AT2	Nguyễn Thị Hải	AT16K
6	AT2	Nguyễn Thị Hải	AT16K
7	AT2	Nguyễn Thị Hải	AT16K
8	AT2	Nguyễn Thị Hải	AT16K
9	AT2	Nguyễn Thị Hải	AT16K
10	CN1	Trần Mạnh Cường	CT4A
11	CN1	Trần Mạnh Cường	CT4A
12	CN1	Trần Mạnh Cường	CT4A
13	CN1	Trần Mạnh Cường	CT4A
14	CN1	Trần Mạnh Cường	CT4A
15	CN2	Lê Văn Minh	CT4A
16	CN2	Lê Văn Minh	CT4A
17	CN2	Lê Văn Minh	CT4A
18	CN2	Lê Văn Minh	CT4A
19	CN2	Lê Văn Minh	CT4A
20	DT1	Nguyễn Bảo Lâm	DT3A

	Masv	HotenSV	Lop
1	AT1	Cao Thu Huyền	AT16A
2	AT2	Nguyễn Thị Hải	AT16K
3	CN1	Trần Mạnh Cường	CT4A
4	CN2	Lê Văn Minh	CT4A
5	DT1	Nguyễn Bảo Lâm	DT3A



# Kết nối sử dụng mệnh đề WHERE

➤ **Ví dụ 2** Đưa ra thông tin các sinh viên đã thi môn có tên là “Cơ sở dữ liệu”

```
SELECT SV.Masv, HotenSV, Lop  
FROM Sinhvien Sv, Ketqua Kq, Monhoc MH  
WHERE Sv.Masv = Kq.MaSV and Kq.MaMH = MH.MaMH and TenMH = N'Cơ sở dữ liệu'
```

➤ **Kết quả**

	Masv	HotenSV	Lop
1	AT1	Cao Thu Huyền	AT16A
2	AT2	Nguyễn Thị Hải	AT16K
3	CN1	Trần Mạnh Cường	CT4A
4	CN2	Lê Văn Minh	CT4A
5	DT1	Nguyễn Bảo lâm	DT3A



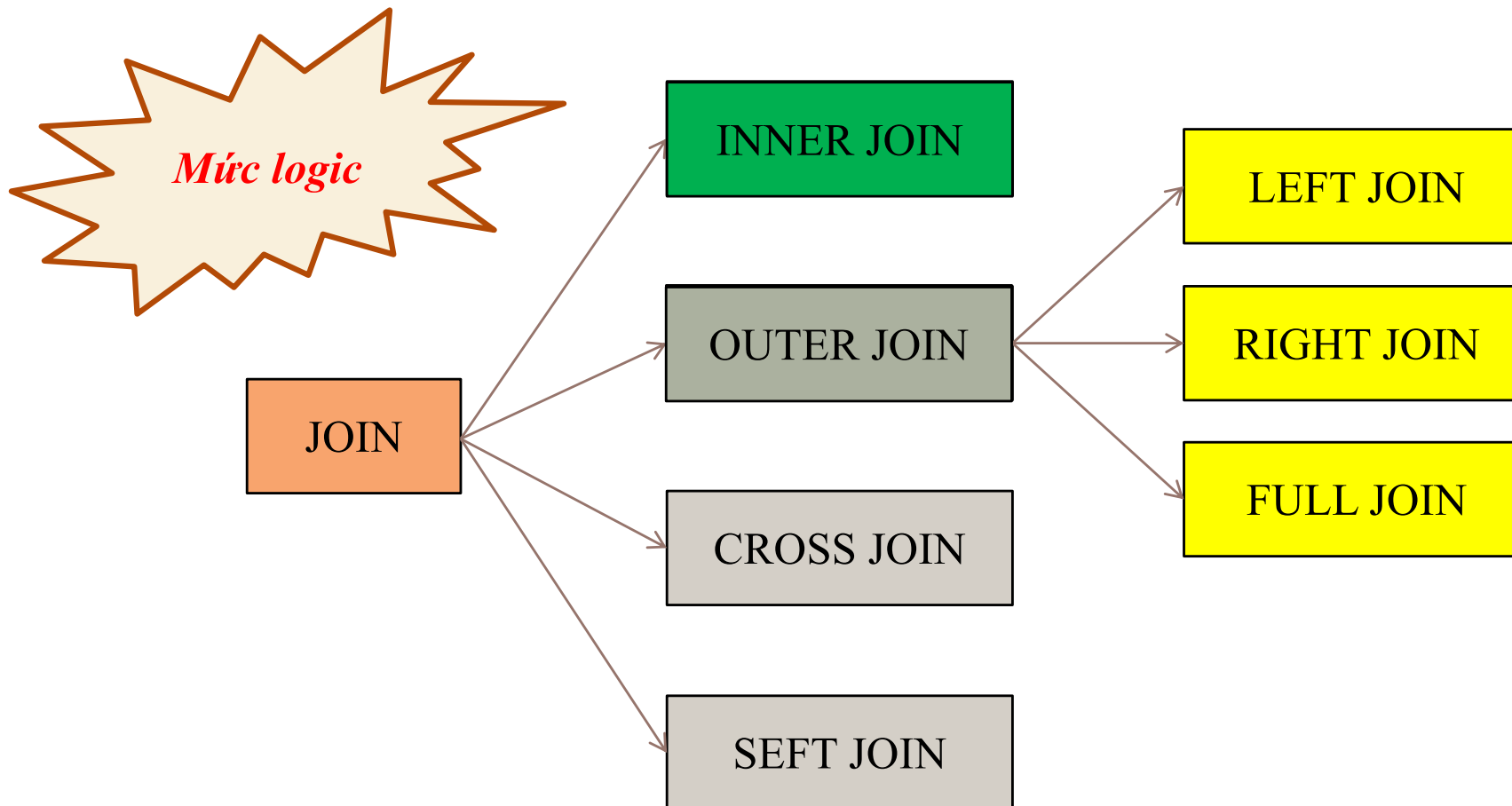
# Kết bằng mệnh đề JOIN

## ➤ *Ý nghĩa*

Mệnh đề **JOINS** trong SQL được sử dụng để kết hợp các bản ghi từ hai hoặc nhiều bảng trong một Database. Một JOIN là một phương tiện để kết hợp các trường từ hai bảng bằng việc sử dụng các giá trị chung của mỗi bảng.



# Kết bảng mệnh đề JOIN

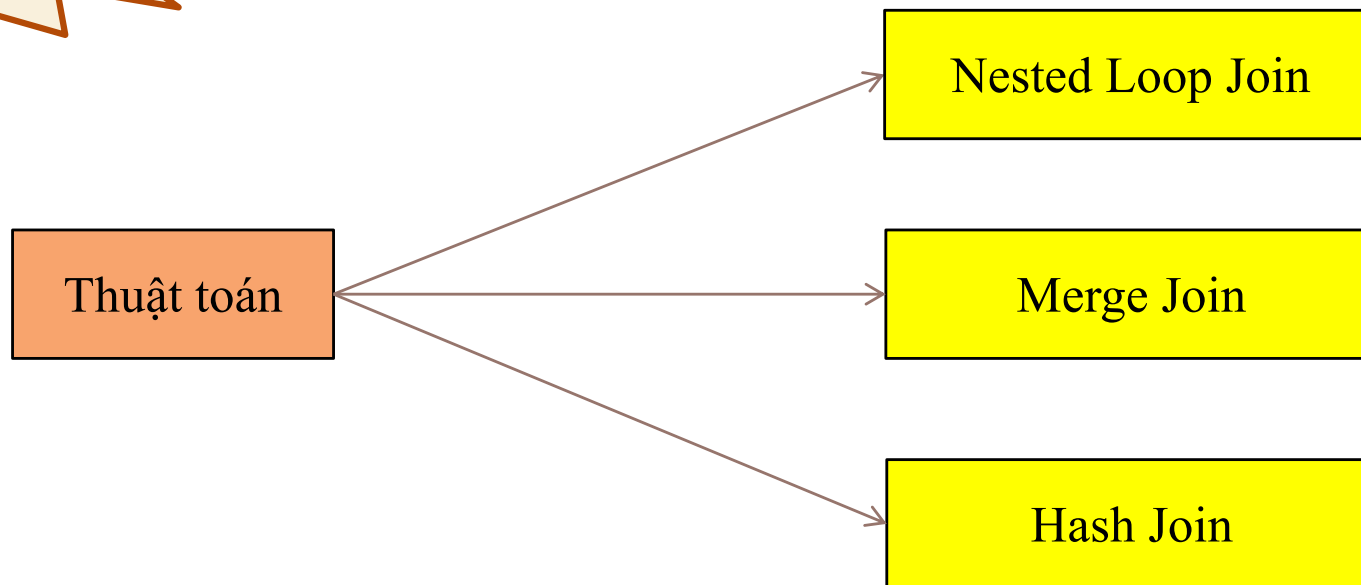






# Các cơ chế thực thi lệnh JOIN

*Mức vật lý*

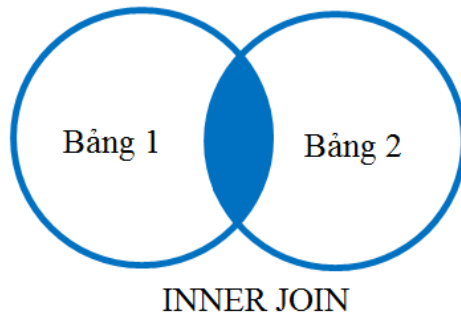




# INNER JOIN

## ➤ Ý nghĩa

**INNER JOIN** trả về kết quả là các bản ghi mà trường được **JOIN** ở hai bảng khớp nhau, các bản ghi chỉ xuất hiện ở một trong hai bảng sẽ bị loại.



## ➤ Cú pháp

```
SELECT bang1.cot1, bang2.cot2...  
FROM bang1 INNER JOIN bang2  
ON bang1.cot_chung = bang2.cot_chung...
```



# Ví dụ sử dụng INNER JOIN

➤ **Ví dụ 1** Hiển thị thông tin sinh viên (MaSV, HotenSV) đã thi

**Cách 1**

```
SELECT DISTINCT SV.MaSV, HotenSV
FROM Sinhvien SV, Ketqua KQ
WHERE SV.MaSV = KQ.MaSV
```

**Cách 2**

```
SELECT DISTINCT SV.MaSV, HotenSV
FROM Sinhvien SV INNER JOIN Ketqua KQ
ON SV.MaSV = KQ.MaSV
```

➤ **Kết quả**

	MaSV	HotenSV
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh
5	DT1	Nguyễn Bảo lâm



# Ví dụ sử dụng INNER JOIN

- **Ví dụ 2** Hiển thị thông tin sinh viên đã thi môn có tên là ‘Cơ sở dữ liệu’  
Thông tin bao gồm: MaSV, HotenSV

## Cách 1

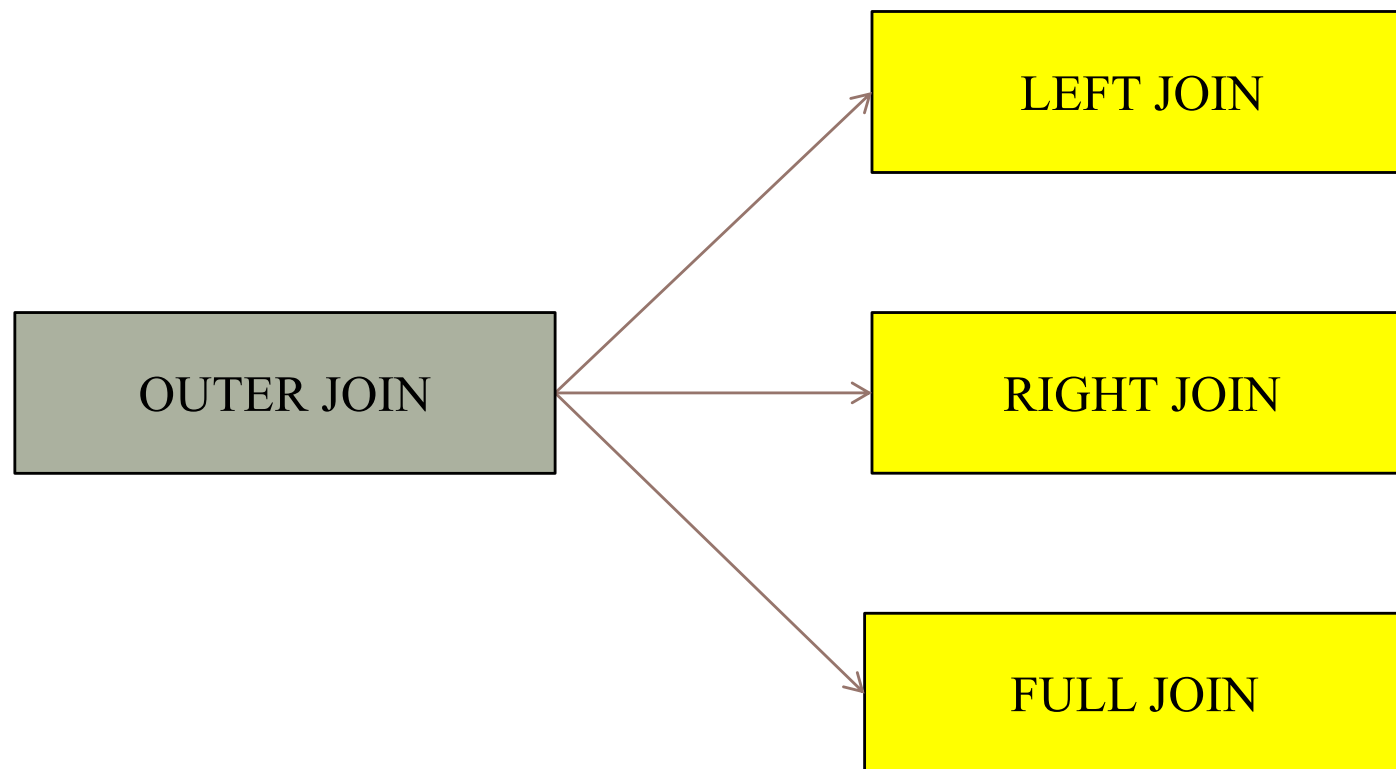
```
SELECT SV.MaSV, HotenSV
FROM Sinhvien SV, Ketqua KQ, Monhoc MH
WHERE SV.MaSV = KQ.MaSV and Mh.MaMH = KQ.MaMH and TenMH = N'Cơ sở dữ liệu'
```

## Cách 2

```
SELECT SV.MaSV, HotenSV
FROM (Sinhvien SV INNER JOIN Ketqua KQ
ON SV.MaSV = KQ.MaSV) INNER JOIN Monhoc MH ON MH.MaMH = KQ.MaMH
WHERE TenMH = N'Cơ sở dữ liệu'
```

## ➤ Kết quả

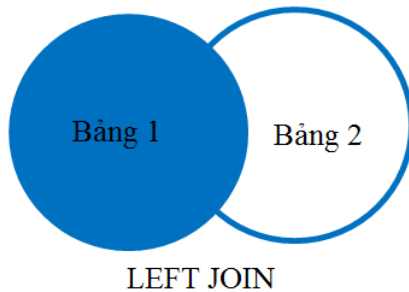
	MaSV	HotenSV
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh





# 1. LEFT JOIN

- **Ý nghĩa** Trả về tất cả giá trị từ bảng **bên trái** + các giá trị được so khớp từ bảng bên phải hoặc NULL trong trường hợp không có so khớp nào.



➤ **Cú pháp**

```
SELECT bang1.cot1, bang2.cot2...  
FROM bang1 LEFT JOIN bang2  
ON bang1.cot_chung = bang2.cot_chung
```

- **Ví dụ** Kết nối hai bảng “**Sinhvien**” và “**Ketqua**” bằng LEFT JOIN. So sánh sự khác nhau với INNER JOIN.

```
SELECT SV.MaSV, HotenSV, MaMH, Diem  
FROM Sinhvien SV LEFT JOIN Ketqua KQ  
ON SV.MaSV = KQ.MaSV
```

**Bảng chính**

**Bảng ẩn theo**

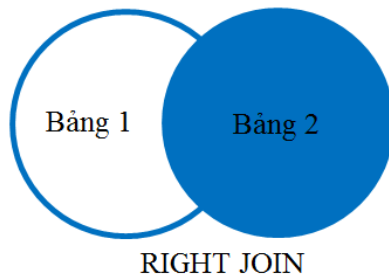
➤ **Kết quả**

	MaSV	HotenSV	MaMH	Diem
7	AT2	Nguyễn Thị Hải	MMT	5
8	AT2	Nguyễn Thị Hải	TCC	6
9	AT2	Nguyễn Thị Hải	TCS	5
10	CN1	Trần Mạnh Cường	CSDL	9
11	CN1	Trần Mạnh Cường	HTMT	9
12	CN1	Trần Mạnh Cường	MMT	10
13	CN1	Trần Mạnh Cường	TCC	7
14	CN1	Trần Mạnh Cường	TCS	9
15	CN2	Lê Văn Minh	CSDL	8
16	CN2	Lê Văn Minh	HTMT	7
17	CN2	Lê Văn Minh	MMT	7
18	CN2	Lê Văn Minh	TCC	8
19	CN2	Lê Văn Minh	TCS	8
20	DT1	Nguyễn Bảo lâm	HTMT	5
21	DT1	Nguyễn Bảo lâm	MMT	6
22	DT3	Vũ Tuấn Đạt	NULL	NU...



## 2. RIGHT JOIN

- **Ý nghĩa** Trả về tất cả giá trị từ bảng **bên phải**, cộng với các giá trị được so khớp từ bảng bên trái hoặc NULL trong trường hợp không có so khớp nào.



➤ **Cú pháp**

```
SELECT bang1.cot1, bang2.cot2...  
FROM bang1 RIGHT JOIN bang2  
ON bang1.cot_chung = bang2.cot_chung
```

- **Ví dụ** Kết nối hai bảng “**Monhoc**” và “**Ketqua**” bằng RIGHT JOIN. So sánh sự khác nhau với INNER JOIN, LEFT JOIN.

```
SELECT MaSV, Diem, MH.MaMH, TenMH  
FROM Ketqua KQ RIGHT JOIN Monhoc MH  
ON KQ.MaMH = MH.MaMH
```

**Bảng ăn theo**

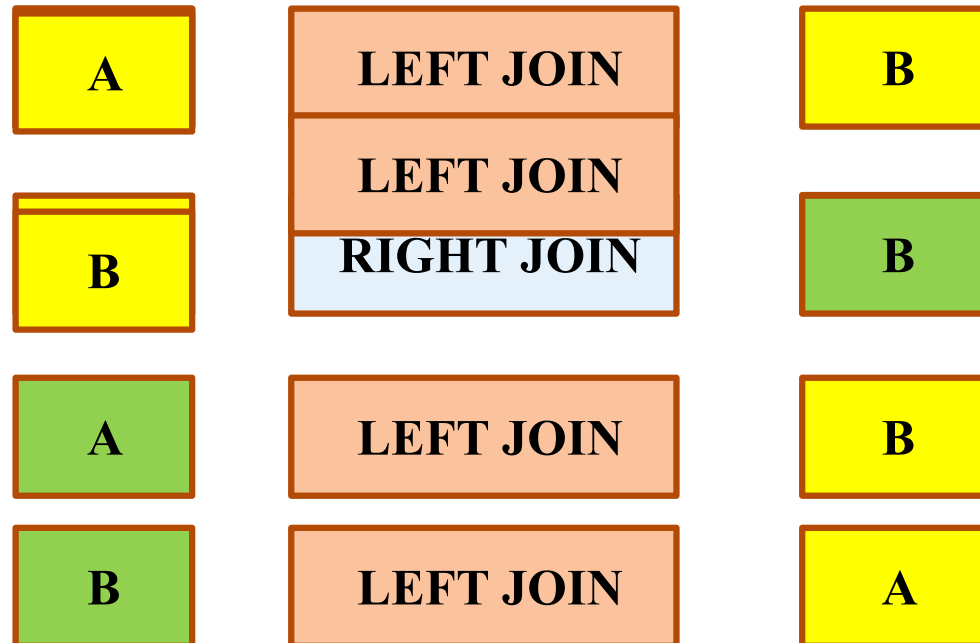
**Bảng chính**

➤ **Kết quả**

	MaSV	Diem	MaMH	TenMH
7	CN1	9	HTMT	Hệ thống máy tính
8	CN2	7	HTMT	Hệ thống máy tính
9	DT1	5	HTMT	Hệ thống máy tính
10	AT1	7	MMT	Mạng máy tính
11	AT2	5	MMT	Mạng máy tính
12	CN1	10	MMT	Mạng máy tính
13	CN2	7	MMT	Mạng máy tính
14	DT1	6	MMT	Mạng máy tính
15	NULL	NULL	TA2	Tiếng anh 2
16	AT2	8	TCC	Toán cao cấp
17	CN1	7	TCC	T...



# Nhận xét

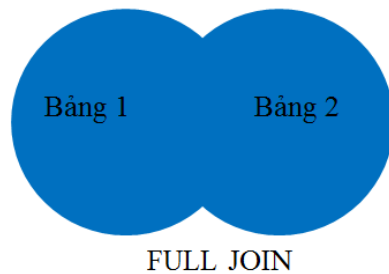






### 3. FULL JOIN

- **Ý nghĩa** Bảng được kết hợp sẽ chứa tất cả bản ghi từ **cả hai bảng**, và điền vào đó giá trị NULL cho các giá trị không so khớp nhau.



- **Cú pháp**

```
SELECT bang1.cot1, bang2.cot2...  
FROM bang1 FULL JOIN bang2  
ON bang1.cot_chung = bang2.cot_chung
```

- **Ví dụ** Kết nối hai bảng “**Sinhvien**” và “**Ketqua**” bằng FULL JOIN. So sánh!

```
SELECT SV.MaSV, HotenSV, MaMH, Diem  
FROM Sinhvien SV FULL JOIN Ketqua KQ  
ON SV.MaSV = KQ.MaSV
```

- **Kết quả**

	MaSV	HotenSV	MaMH	Diem
7	AT2	Nguyễn Thị Hải	MMT	5
8	AT2	Nguyễn Thị Hải	TCC	6
9	AT2	Nguyễn Thị Hải	TCS	5
10	CN1	Trần Mạnh Cường	CSDL	9
11	CN1	Trần Mạnh Cường	HTMT	9
12	CN1	Trần Mạnh Cường	MMT	10
13	CN1	Trần Mạnh Cường	TCC	7
14	CN1	Trần Mạnh Cường	TCS	9
15	CN2	Lê Văn Minh	CSDL	8
16	CN2	Lê Văn Minh	HTMT	7
17	CN2	Lê Văn Minh	MMT	7
18	CN2	Lê Văn Minh	TCC	8
19	CN2	Lê Văn Minh	TCS	8
20	DT1	Nguyễn Bảo lâm	HTMT	5
21	DT1	Nguyễn Bảo lâm	MMT	6
22	DT3	Vũ Tuấn Đạt	NULL	NU...



# CROSS JOIN

## ➤ Cú pháp

```
SELECT bang1.cot1, bang2.cot2  
FROM bang1, bang2
```

Không có  
*where*

## ➤ Ý nghĩa

CROSS JOIN là phép tích đề các giữa các dòng dữ liệu trong 2 bảng. Kích thước của Cross Join là  $n*m$ . Trong đó,  $n$  và  $m$  là số dòng dữ liệu của 2 bảng.



Ít dùng



# SELF JOIN

## ➤ Ý nghĩa

**SELF JOIN** là phép nối 1 bảng với chính nó, SELF JOIN thường được sử dụng khi muốn trích xuất những dòng dữ liệu có mối quan hệ hoặc tương tự với các dòng dữ liệu khác trong cùng một bảng.

## ➤ Cú pháp

```
SELECT a.ten_cot, b.ten_cot...  
FROM bang1 a, bang1 b  
WHERE a.field_chung = b.field_chung
```



# Truy vấn con

## ➤ Ví dụ

```
SELECT < danh sách các cột >  
FROM < danh sách các bảng >  
WHERE < biểu thức >....
```

<so sánh tập hợp >

```
(SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE < điều kiện>).....
```

*Truy vấn cha*

*Truy vấn con*



# Phân loại Truy vấn con

## ❖ *Các dạng truy vấn con theo kết quả trả về:*

1. Scalar Subquery - Truy vấn con trả về kết quả đơn trị
2. Multiple - Row Subqueries:
  - Truy vấn con trả về kết quả là một cột, nhiều hàng (List of values)
  - Truy vấn con trả về kết quả là nhiều cột, nhiều hàng (Table)

## ❖ *Các dạng truy vấn con theo vị trí:*

1. Truy vấn con nằm trong mệnh đề SELECT
2. Truy vấn con nằm trong mệnh đề FROM
3. Truy vấn con nằm trong mệnh đề WHERE



# Truy vấn con (Subquery)

## Một số quy tắc cần ghi nhớ khi sử dụng truy vấn con

- ✓ Câu truy vấn con phải nằm trong dấu ngoặc đơn

```
SELECT COUNT(MaSV)
```

```
FROM KETQUA
```

```
WHERE MaMH = 'CSDL' and Diem = (SELECT MAX(Diem) FROM KETQUA  
WHERE MaMH = 'CSDL')
```

Điều kiện so sánh

Truy vấn con

- ✓ Đặt truy vấn con bên phải điều kiện so sánh

- ✓ Sử dụng các toán tử một dòng với các truy vấn con trả về một dòng và sử dụng các toán tử nhiều dòng với các truy vấn con trả về nhiều dòng

```
SELECT *
```

```
FROM Sinhvien
```

```
WHERE MaSV = (SELECT MaSV FROM Ketqua)
```

IN



# Truy vấn con (Subquery)

## Một số quy tắc cần ghi nhớ khi sử dụng truy vấn con

- ✓ Một truy vấn con không thể bị bao quanh một cách trực tiếp trong một hàm tập hợp

```
SELECT COUNT(MaSV)
FROM KETQUA
WHERE MaMH = 'CSDL' and Diem = (
```

MAX

```
(SELECT Diem FROM KETQUA
WHERE MaMH = 'CSDL')
```

Truy vấn con

- ✓ Toán tử BETWEEN không thể được sử dụng với **MỘT** truy vấn con; tuy nhiên, toán tử BETWEEN có thể được sử dụng bên trong truy vấn con



# Truy vấn con nằm trong mệnh đề SELECT

## ➤ Ý nghĩa

- Kết quả của truy vấn con thường là một giá trị đơn (scalar subquery)
- Kết quả chỉ mang tính chất trung gian, không có ý nghĩa lưu trữ thực sự

## ➤ Ví dụ: Tìm ra số sinh viên chưa thi môn nào

```
SELECT COUNT(MaSV) FROM Sinhvien
```

Tổng số sinh viên

```
SELECT COUNT( DISTINCT MaSV) FROM Ketqua
```

Tổng số sinh viên  
đã thi

```
SELECT COUNT(MaSV) - ( SELECT COUNT(DISTINCT MaSV)  
FROM Ketqua ) AS 'SL SV chưa thi'  
FROM Sinhvien
```

## ➤ Kết quả

Results		Messages	
		SL SV chưa thi	
1	1		





# Truy vấn con nằm trong mệnh đề FROM

## ➤ Ý nghĩa

- ✓ Kết quả trả về của một câu truy vấn là một bảng (table subquery)
- ✓ Bảng trung gian trong quá trình truy vấn
- ✓ Không có lưu trữ thật sự

## ➤ Cú pháp

*SELECT* < danh sách các cột >

*FROM* R1, R2, (< truy vấn con >) *AS* tên\_bảng

*WHERE* < điều kiện >

- ## ➤ Ví dụ
- Đưa ra MaSV và số môn đã thi của từng sinh viên mà số môn đã thi  $\geq 2$



# Truy vấn con nằm trong mệnh đề FROM

➤ *Ví dụ* Đưa ra MaSV và số môn đã thi của từng sinh viên mà số môn đã thi  $\geq 2$

*Cách 2*

```
SELECT *  
FROM (  
    SELECT MaSV , COUNT(MaMH) as Somondathi  
    FROM Ketqua  
    GROUP BY MaSV) Bang1  
WHERE Somondathi >1
```

*Cách 1*

```
SELECT MaSV, COUNT(MaMH) AS Somondathi  
FROM Ketqua  
GROUP BY MaSV  
HAVING COUNT(MaMH) >1
```

➤ *Kết quả*

Results			Messages
	MaSV	Somondathi	
1	AT1	4	
2	AT2	5	
3	CN1	5	
4	CN2	5	
5	DT1	2	



# Truy vấn con nằm trong mệnh đề WHERE

## ➤ Ví dụ

**SELECT** < danh sách các cột >  
**FROM** < danh sách các bảng >  
**WHERE** < biểu thức >....

*Truy vấn cha*

< so sánh tập hợp >



**(SELECT** <danh sách các cột>  
**FROM** <danh sách các bảng>  
**WHERE** < điều kiện>).....

*Truy vấn con*



# So sánh tập hợp

Kiểm tra so sánh thông thường ( $=$ ,  $<>$ ,  $<$ ,  $<=$ ,  $>$ ,  $>=$ )

So sánh giá trị của một biểu thức với một giá trị đơn được tạo ra từ truy vấn con

Kiểm tra tập hợp thành viên (IN)

Kiểm tra giá trị của một biểu thức có phải là một trong các giá trị được tạo ra từ truy vấn con hay không

Kiểm tra sự tồn tại (EXISTS)

Trả về giá trị True hoặc False của kết quả trong truy vấn con

Kiểm tra so sánh lượng hóa (ANY, ALL, SOME)

So sánh một giá trị của biểu thức với từng giá trị thuộc tập hợp các giá trị được tạo ra bởi truy vấn con



# Truy vấn con nằm trong mệnh đề WHERE

## ➤ Ví dụ

1. Cho biết **số sinh viên** đạt điểm cao nhất môn có mã là CSDL
2. Cho biết **danh sách sinh viên** (MaSV, Hoten, Lop) của các sinh viên đạt điểm cao nhất môn có mã là CSDL

```
SELECT COUNT(MaSV) 'Số SV đạt điểm  
cao nhất môn CSDL'  
FROM KETQUA  
WHERE MaMH = 'CSDL' and Diem =  
(SELECT MAX(Diem) FROM KETQUA WHERE  
MaMH = 'CSDL')
```

```
SELECT SV.MaSV, HotenSV, Lop  
FROM SINHVIEN SV, KETQUA KQ  
WHERE SV.MaSV = KQ.MaSV and MaMH =  
'CSDL' and Diem = (SELECT MAX(Diem)  
FROM KETQUA WHERE MaMH = 'CSDL')
```

## ➤ Kết quả

Results	Messages
	Số SV đạt điểm cao nhất môn CSDL
1	1

Results	Messages
	MaSV HotenSV Lop
1	CN1 Trần Mạnh Cường CT4A



Truy vấn con nằm trong mệnh đề SELECT

Truy vấn con nằm trong mệnh đề FROM

Truy vấn con nằm trong mệnh đề WHERE





# Các phép toán tập hợp

*T-SQL có cài đặt các phép toán:*

**Hợp (UNION)**

**Trừ (EXCEPT)**

**Giao  
(INTERSECT)**



# Phép UNION

## ➤ Ý nghĩa

1. **UNION** để ghép nối các kết quả của hai hay nhiều câu lệnh **SELECT** lại với nhau thành một tập kết quả duy nhất.
2. Phép kết này sẽ loại bỏ những kết quả trùng lặp của hai bảng. Dữ liệu được sắp xếp theo thứ tự.

## ➤ Cú pháp



```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
UNION  
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```





# Phép UNION

## ➤ Điều kiện sử dụng *UNION*

1. Số cột của mỗi câu truy vấn *SELECT* phải bằng nhau
2. Kiểu dữ liệu của mỗi cột trong câu *SELECT* này phải giống với kiểu dữ liệu của cột tương ứng trong câu *SELECT* kia nhưng độ dài cột dữ liệu không nhất thiết phải bằng nhau.



# Phép UNION

- **Ví dụ** Đưa ra danh sách sinh viên (MaSV, HotenSV) đã thi ít nhất là 1 trong 2 môn có mã là 'CSDL' và 'TCC'

	masv	mamh	diem
1	AT1	CSDL	8
2	AT1	HTMT	9
3	AT1	MMT	7
4	AT1	TCS	9
5	AT2	CSDL	4
6	AT2	HTMT	4
7	AT2	MMT	5
8	AT2	TCC	6
9	AT2	TCS	5
10	CN1	CSDL	9
11	CN1	HTMT	9
12	CN1	MMT	10
13	CN1	TCC	7
14	CN1	TCS	9
15	CN2	CSDL	8
16	CN2	HTMT	7
17	CN2	MMT	7
18	CN2	TCC	8
19	CN2	TCS	8
20	DT1	HTMT	5
21	DT1	MMT	6

```
SELECT sv.masv, hotensv
FROM sinhvien sv, ketqua kq
WHERE sv.masv = kq.masv and mamh = 'CSDL'
UNION
```

```
SELECT sv.masv, hotensv
FROM sinhvien sv, ketqua kq
WHERE sv.masv = kq.masv and mamh = 'TCC'
```

	masv	hotensv
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh

	masv	hotensv
1	AT2	Nguyễn Thị Hải
2	CN1	Trần Mạnh Cường
3	CN2	Lê Văn Minh

- **Kết quả**

	masv	hotensv
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh



➤ **Ví dụ:** Đưa ra danh sách sinh viên (MaSV, HotenSV) đã thi ít nhất là 1 trong 2 môn có mã là 'CSDL' và 'TCC'

Results			
	masv	mamh	diem
1	AT1	CSDL	8
2	AT1	HTMT	9
3	AT1	MMT	7
4	AT1	TCS	9
5	AT2	CSDL	4
6	AT2	HTMT	4
7	AT2	MMT	5
8	AT2	TCC	6
9	AT2	TCS	5
10	CN1	CSDL	9
11	CN1	HTMT	9
12	CN1	MMT	10
13	CN1	TCC	7
14	CN1	TCS	9
15	CN2	CSDL	8
16	CN2	HTMT	7
17	CN2	MMT	7
18	CN2	TCC	8
19	CN2	TCS	8
20	DT1	HTMT	5
21	DT1	MMT	6

```
SELECT DISTINCT sv.masv, hotensv
FROM sinhvien sv, ketqua kq
WHERE sv.masv = kq.masv and (mamh = 'TCC'
or mamh = 'CSDL')
```

➤ **Kết quả**

Results		
	masv	hotensv
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	AT2	Nguyễn Thị Hải
4	CN1	Trần Mạnh Cường
5	CN1	Trần Mạnh Cường
6	CN2	Lê Văn Minh
7	CN2	Lê Văn Minh

Results		
	masv	hotensv
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh



# Phép UNION ALL

## ➤ Ý nghĩa

1. **UNION ALL** để ghép nối các kết quả của hai hay nhiều câu lệnh **SELECT** lại với nhau thành một tập kết quả duy nhất.
2. Phép kết này sẽ không loại bỏ những kết quả trùng lặp của hai bảng. Dữ liệu không được sắp xếp theo thứ tự.

## ➤ Cú pháp

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
UNION ALL  
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```



# Phép UNION ALL

➤ **Ví dụ:** Đưa ra danh sách sinh viên (MaSV, HotenSV) đã thi ít nhất là 1 trong 2 môn có mã là 'CSDL' và 'TCC'

	masv	mamh	diem
1	AT1	CSDL	8
2	AT1	HTMT	9
3	AT1	MMT	7
4	AT1	TCS	9
5	AT2	CSDL	4
6	AT2	HTMT	4
7	AT2	MMT	5
8	AT2	TCC	6
9	AT2	TCS	5
10	CN1	CSDL	9
11	CN1	HTMT	9
12	CN1	MMT	10
13	CN1	TCC	7
14	CN1	TCS	9
15	CN2	CSDL	8
16	CN2	HTMT	7
17	CN2	MMT	7
18	CN2	TCC	8
19	CN2	TCS	8
20	DT1	HTMT	5
21	DT1	MMT	6

```
SELECT sv.masv, hotensv
FROM sinhvien sv, ketqua kq
WHERE sv.masv = kq.masv and mamh = 'CSDL'
UNION ALL
```

```
SELECT sv.masv, hotensv
FROM sinhvien sv, ketqua kq
WHERE sv.masv = kq.masv and mamh = 'TCC'
```

➤ **Kết quả**

	masv	hotensv
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh
5	AT2	Nguyễn Thị Hải
6	CN1	Trần Mạnh Cường
7	CN2	Lê Văn Minh

1

2

	masv	hotensv
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh

	masv	hotensv
1	AT2	Nguyễn Thị Hải
2	CN1	Trần Mạnh Cường
3	CN2	Lê Văn Minh



# Phép EXCEPT

- **Ý nghĩa** *EXCEPT* cho kết quả là những dòng của bảng thứ nhất và loại bỏ những dòng trùng khớp giữa kết quả 1 và kết quả 2.



- **Cú pháp**

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
EXCEPT  
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```



# Phép EXCEPT

- *Ví dụ* Đưa ra danh sách sinh viên (MaSV, HotenSV) chưa thi môn nào

```
SELECT MaSV, HotenSV FROM Sinhvien  
EXCEPT
```

```
SELECT SV.MaSV, HotenSV  
FROM Sinhvien SV, Ketqua KQ  
WHERE SV.MaSV = KQ.MaSV
```

Lấy danh  
sách  
(MaSV)  
chưa thi  
môn nào

	MaSV	HotenSV
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh
5	DT1	Nguyễn Bảo lâm
6	DT3	Vũ Tuấn Đạt

	MaSV	HotenSV
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh
5	DT1	Nguyễn Bảo lâm

## ➤ Kết quả

	MaSV	HotenSV
1	DT3	Vũ Tuấn Đạt



# Phép INTERSECT

➤ **Ý nghĩa** *INTERSECT* cho ra kết quả là phần chung giữa hai kết quả

➤ **Cú pháp**

*SELECT* <ds cột> *FROM* <ds bảng> *WHERE* <điều kiện>

*INTERSECT*

*SELECT* <ds cột> *FROM* <ds bảng> *WHERE* <điều kiện>





# Phép INTERSECT

- *Ví dụ* Đưa ra danh sách sinh viên (MaSV, HotenSV) đã thi cả 2 môn có mã là 'CSDL' và 'TCC'

	masv	mamh	diem
1	AT1	CSDL	8
2	AT1	HTMT	9
3	AT1	MMT	7
4	AT1	TCS	9
5	AT2	CSDL	4
6	AT2	HTMT	4
7	AT2	MMT	5
8	AT2	TCC	6
9	AT2	TCS	5
10	CN1	CSDL	9
11	CN1	HTMT	9
12	CN1	MMT	10
13	CN1	TCC	7
14	CN1	TCS	9
15	CN2	CSDL	8
16	CN2	HTMT	7
17	CN2	MMT	7
18	CN2	TCC	8
19	CN2	TCS	8
20	DT1	HTMT	5
21	DT1	MMT	6

```
SELECT sv.masv, hotensv
FROM sinhvien sv, ketqua kq
WHERE sv.masv = kq.masv and mamh = 'CSDL'
INTERSECT
```

```
SELECT sv.masv, hotensv
FROM sinhvien sv, ketqua kq
WHERE sv.masv = kq.masv and mamh = 'TCC'
```

- *Kết quả*

	masv	hotensv
1	AT2	Nguyễn Thị Hải
2	CN1	Trần Mạnh Cường
3	CN2	Lê Văn Minh

	masv	hotensv
1	AT1	Cao Thu Huyền
2	AT2	Nguyễn Thị Hải
3	CN1	Trần Mạnh Cường
4	CN2	Lê Văn Minh

	masv	hotensv
1	AT2	Nguyễn Thị Hải
2	CN1	Trần Mạnh Cường
3	CN2	Lê Văn Minh

