



# HỌC VIỆN KỸ THUẬT MẬT MÃ

## KHOA CÔNG NGHỆ THÔNG TIN

### Bài giảng: Truy vấn dữ liệu đơn giản



“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**



# Câu lệnh SELECT

**DDL**

**DML**

**DCL**

- **Ý nghĩa** Lệnh **SELECT** là lệnh cho phép truy vấn dữ liệu mà không làm thay đổi dữ liệu hoặc các đối tượng trong CSDL



# Câu lệnh SELECT

## ➤ Cú pháp

*SELECT* [*DISTINCT* | *Top n* | *\** ] <biểu thức/ cột [*AS* <tên mới>],...>

[*INTO* <tên bảng mới>]

*FROM* <tên bảng> [<bí danh>],...

[*WHERE* <điều kiện chọn>]

[*GROUP BY* < ds tên cột gom nhóm>]

[*HAVING* <điều kiện lọc nhóm>]

[*ORDER BY* <tên cột>[*ASC*/*DESC*],...];



# Truy vấn đơn giản

## ➤ Cú pháp

*SELECT* <danh sách các cột>

*FROM* <tên bảng>

*[WHERE* <biểu thức điều kiện>]

## **Trong đó**

<danh sách các cột>

Tên các thuộc tính (cột) sẽ được hiển thị trong kết quả truy vấn.

<tên bảng>

Tên bảng liên quan để lấy kết quả

<biểu thức điều kiện>

là điều kiện đưa vào để chọn lọc dữ liệu, thường gồm:

- ✓ Các phép toán so sánh:  $=$ ,  $<$ ,  $>$ ,  $\leq$ ,  $\geq$ ,  $\neq$ ,  $\square$
- ✓ Các phép toán logic: AND, OR, và NOT
- ✓ Các từ khóa: BETWEEN ... AND, IN, EXISTS, LIKE...



# Câu lệnh SELECT

1. Dùng **SELECT** để chọn tất cả các đối tượng trong bảng

➤ **Cú pháp**      **SELECT \* FROM <tenbang>**

➤ **Ví dụ**          **SELECT \* FROM Sinhvien**

➤ **Kết quả**

	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



# Câu lệnh SELECT

## 2. Dùng **SELECT** để chọn một số cột trong bảng

### ➤ *Cú pháp*

**SELECT** <tencot1>, <tencot2> ...

**FROM** <tenbang>

### ➤ *Ví dụ*

**SELECT** HotenSV **FROM** Sinhvien

### ➤ *Kết quả*

Results Messages	
	hotenSV
1	Cao Thu Huyền
2	Nguyễn Thị Hải
3	Trần Mạnh Cường
4	Lê Văn Minh
5	Nguyễn Bảo lâm
6	Vũ Tuấn Đạt



# Liên kết các cột dữ liệu trong kết quả truy vấn

## ➤ Cú pháp

`SELECT <tencot> + 'string' + <tencot> + 'string' + <tencot>... FROM <tenbang>`

## ➤ Ví dụ

`SELECT HotenSV + (N' học lớp ' ) + Lop + (N'quê ở ' ) + Quequan FROM Sinhvien`

`SELECT HotenSV + (N' học lớp ' ) + Lop + (N'quê ở ' ) + Quequan AS 'Thông tin sinh viên' FROM Sinhvien`

## ➤ Kết quả

Results		Messages
(No column name)		
1	Cao Thu Huyền học lớp AT16A quê ở Hà Nội	
2	Nguyễn Thị Hải học lớp AT16K quê ở Hà Nam	
3	Trần Mạnh Cường học lớp CT4A quê ở Thái Bình	
4	Lê Văn Minh học lớp CT4A quê ở Thanh Hóa	
5	Nguyễn Bảo lâm học lớp DT3A quê ở Hà Nội	
6	Vũ Tuấn Đạt học lớp DT3D quê ở Hà Nam	

Results		Messages
Thông tin sinh viên		
1	Cao Thu Huyền học lớp AT16A quê ở Hà Nội	
2	Nguyễn Thị Hải học lớp AT16K quê ở Hà Nam	
3	Trần Mạnh Cường học lớp CT4A quê ở Thái Bình	
4	Lê Văn Minh học lớp CT4A quê ở Thanh Hóa	
5	Nguyễn Bảo lâm học lớp DT3A quê ở Hà Nội	
6	Vũ Tuấn Đạt học lớp DT3D quê ở Hà Nam	





# Hạn chế tập kết quả bằng việc sử dụng TOP và PERCENT

## 1. Lấy n hàng trong bản ghi với TOP

➤ **Cú pháp** `SELECT TOP n <tencot> FROM <tenbang>`

➤ **Ví dụ** Lấy thông tin 3 sinh viên đầu tiên trong bảng dữ liệu

`SELECT TOP 3 * FROM Sinhvien`

➤ **Kết quả**

	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

## 2. Lấy n hàng trong bản ghi với PERCENT

➤ **Cú pháp** `SELECT TOP n PERCENT <tencot> FROM <tenbang>`

➤ **Ví dụ** Lấy 20% số sinh viên đầu tiên trong bảng dữ liệu

`SELECT TOP 20 PERCENT * FROM Sinhvien`

➤ **Kết quả**

	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



# Loại bỏ dữ liệu lặp với DISTINCT

## ➤ Cú pháp

*SELECT DISTINCT <tencot> FROM <tenbang>*

## ➤ Ví dụ

	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

*SELECT Lop FROM Sinhvien*

*SELECT DISTINCT Lop FROM Sinhvien*

## ➤ Kết quả

	lop
1	AT16A
2	AT16K
3	CT4A
4	CT4A
5	DT3A
6	DT3D

## ➤ Kết quả

	lop
1	AT16A
2	AT16K
3	CT4A
4	DT3A
5	DT3D



# Sử dụng mệnh đề WHERE

➤ **Cú pháp** *SELECT* <tencot>  
*FROM* <tenbang>  
*WHERE* <dieukien>

➤ **Ví dụ** Đưa ra thông tin các sinh viên quê ở Hà Nội

	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

*SELECT* \* *FROM* *Sinhvien* *WHERE* *Quequan* = *N'Hà Nội'*

➤ **Kết quả**

	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	DT1	Nguyễn Bảo lâm	Nam	DT3A	Hà Nội	1998-05-12	0677878909	khk@gmail.com



# Các toán tử so sánh, logic

## 1. Toán tử so sánh

Toán tử	Ý nghĩa
=	Bằng nhau
>	Lớn hơn
<	Bé hơn
>=	Lớn hơn hoặc bằng
<=	Bé hơn hoặc bằng
<>	Khác nhau
!	Phủ định

**Ví dụ** Đưa ra danh sách các môn học có DVHP >2

```
SELECT * FROM Monhoc WHERE DVHP >2
```

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

## 2. Toán tử logic

Các toán tử logic **AND**, **OR** và **NOT**

- ✓ **AND**, **OR** được sử dụng để kết nối điều kiện tìm kiếm trong mệnh **WHERE**
- ✓ **NOT** phủ định điều kiện tìm kiếm.



# Ví dụ về các toán tử so sánh, logic

1. Liệt kê danh sách các môn học có  $DVHP > 2$  và  $DVHP < 5$

`SELECT * FROM Monhoc WHERE DVHP > 2 and DVHP < 5`

	mamh	tenmh	DVHP
1	CSDL	Cơ sở dữ liệu	3
2	HTMT	Hệ thống máy tính	3
3	MMT	Mạng máy tính	4

2. Liệt kê danh sách các môn học có  $DVHP < 3$  hoặc  $DVHP > 4$

`SELECT * FROM Monhoc WHERE DVHP < 3 or DVHP > 4`

	mamh	tenmh	DVHP
1	TCC	Toán cao cấp	2
2	TCS	Tin học cơ sở	5

3. Liệt kê danh sách môn học có  $DVHP$  không bé hơn 4

`SELECT * FROM Monhoc WHERE NOT DVHP < 4`

	mamh	tenmh	DVHP
1	MMT	Mạng máy tính	4
2	TCS	Tin học cơ sở	5



# Lấy dữ liệu trong một khoảng với BETWEEN

## ➤ Cú pháp

*SELECT* <col1>, <col2> ...

*FROM* <ten bang>

*WHERE* <ten\_col> *BETWEEN* *giatri1* *AND* *giatri2*

## ➤ Ví dụ

*SELECT* \*

*FROM* Monhoc

*WHERE* DVHP BETWEEN 1 AND 3



# Tìm kiếm gần đúng với LIKE

## ➤ Cú pháp

```
SELECT <tencot1>, <tencot2> ...  
FROM <ten bang>  
WHERE <tencot> LIKE <du_lieu_mau>
```

## Trong đó

<du\_lieu\_mau> đặt sau từ khóa LIKE dùng để đại diện gần chính xác cho một dữ liệu mẫu, thông qua hai ký tự % và \_

% Biểu thị một hoặc nhiều ký tự, hoặc thể hiện ký tự bằng 0.

\_ Biểu thị một ký tự đơn



# Một số dạng tìm kiếm gần đúng

Dạng tìm kiếm	Mệnh đề WHERE
Tìm kiếm dữ liệu bắt đầu bằng ký tự <b>K</b>	...WHERE <column> LIKE 'K%'
Tìm kiếm dữ liệu kết thúc bằng ký tự <b>K</b>	...WHERE <column> LIKE '%K'
Tìm kiếm dữ liệu có chứa ký tự <b>Kt</b> ở vị trí bất kỳ	...WHERE <column> LIKE '%Kt%'
Tìm kiếm dữ liệu có ký tự <b>K</b> ở vị trí thứ hai	...WHERE <column> LIKE '_K%'
Tìm kiếm dữ liệu bắt đầu bằng ký tự <b>K</b> , và có ít nhất có chiều dài là 3 ký tự	...WHERE <column> LIKE 'K_%_'
Tìm kiếm dữ liệu bắt đầu bằng ký tự <b>K</b> , kết thúc bằng ký tự <b>m</b>	...WHERE <column> LIKE 'K%m'





# Một số dạng tìm kiếm gần đúng

- **Ví dụ 1** Xuất ra thông tin sinh viên có tên là Minh

```
SELECT * FROM Sinhvien WHERE HotenSV like '%Minh'
```

Results		Messages						
	masv	hotensv	gioitinh	lop	quequan	ngaysinh	sdt	email
1	CN2	Lê Văn Minh	Nam	CT4A	Thanh Hóa	1998-09-20	0977878809	sutu@gmail.com

- **Ví dụ 2** Xuất ra thông tin sinh viên có chữ ‘Minh’ trong họ và tên

```
SELECT * FROM Sinhvien WHERE Hotensv LIKE '%Minh%'
```

Results		Messages						
	masv	hotensv	gioitinh	lop	quequan	ngaysinh	sdt	email
1	AT3	Hoàng Minh Chiến	Nam	AT16A	Hà Nội	1998-11-10	0967878909	ktm@gmail.com
2	CN2	Lê Văn Minh	Nam	CT4A	Thanh Hóa	1998-09-20	0977878809	sutu@gmail.com



# Mệnh đề ORDER BY

## ➤ Ý nghĩa

Mệnh đề **ORDER BY** trong SQL được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần (**ASC**) hoặc theo thứ tự giảm dần (**DESC**), trên một hoặc nhiều cột. Một số CSDL sắp xếp kết quả truy vấn theo thứ tự tăng dần theo mặc định.

## ➤ Cú pháp

```
SELECT <tencot>  
FROM <tenbang>  
[WHERE <dieukien>]  
ORDER BY <tencot> [ASC | DESC]
```



# Mệnh đề ORDER BY

➤ *Ví dụ* Sắp xếp danh sách môn học theo thứ tự tăng dần, giảm dần của DVHP

`SELECT * FROM Monhoc ORDER BY DVHP`

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

`SELECT * FROM Monhoc ORDER BY DVHP DESC`

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



# Mệnh đề GROUP BY và HAVING

## ➤ Ý nghĩa

Mệnh đề **GROUP BY** được sử dụng kết hợp với lệnh **SELECT** để sắp xếp dữ liệu đồng nhất vào trong các nhóm

## ➤ Cú pháp

```
SELECT <tencot>  
FROM <tenbang>  
[WHERE <dieukien>]  
GROUP BY <tencot>
```



### Một số toán tử trong T-SQL

- DISTINCT
- IN
- LIKE
- IS [NOT] NULL
- BETWEEN
- EXISTS,...

### Một số hàm trong T-SQL

✓ Hàm toán học:

- ROUND
- POWER
- CEILING
- SQRT,...
- FLOOR

✓ Hàm tập hợp:

- SUM(),
- AVG()
- COUNT()
- MAX(),...
- MIN()



# Mệnh đề GROUP BY và HAVING

## ➤ Ví dụ 1 Đếm tổng số sinh viên

```
SELECT COUNT(MaSV) 'Số lượng sinh viên'  
FROM Sinhvien
```

## ➤ Kết quả

Results Messages	
	Số lượng sinh viên
1	6

## ➤ Ví dụ 2 Đếm tổng sinh viên của mỗi tỉnh thành

```
SELECT Quequan 'Quê quán', count(MaSV) 'Số sinh  
viên'  
FROM Sinhvien  
GROUP BY Quequan
```

## ➤ Kết quả

Results Messages		
	Quê quán	Số sinh viên
1	Hà Nam	2
2	Hà Nội	2
3	Thái Bình	1
4	Thanh Hóa	1

Results Messages								
	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



# Mệnh đề GROUP BY và HAVING

## ➤ Ý nghĩa

Mệnh đề **HAVING** được thêm vào SQL vì mệnh đề WHERE không áp dụng được đối với các hàm tập hợp (như SUM). Nếu không có HAVING, ta không thể nào kiểm tra được điều kiện với các hàm tập hợp. Nên ta có thể hiểu như HAVING dùng để thay thế WHERE khi sử dụng các hàm tập hợp.

## ➤ Cú pháp

```
SELECT <tencot>, ham_tap_hop(<tencot>)  
FROM <tenbang>  
WHERE <dieu_kien>  
GROUP BY <tencot>  
HAVING Ham_tap_hop(<tencot>) dieukien giatri
```



# Mệnh đề GROUP BY và HAVING

- *Ví dụ* Đưa ra danh sách các tỉnh thành có số sinh viên >1

```
SELECT Quequan 'Quê quán', count(MaSV) 'Số sinh viên'  
FROM Sinhvien  
GROUP BY Quequan  
HAVING count(MaSV) >1
```

- *Kết quả*

	Quê quán	Số sinh viên
1	Hà Nam	2
2	Hà Nội	2





# So sánh mệnh đề WHERE và HAVING

## ➤ *Giống nhau*

**WHERE** và **HAVING** đều là câu lệnh dùng để lọc ra các bản ghi dựa trên một hoặc nhiều điều kiện

## ➤ *Khác nhau*

- ✓ **WHERE** là câu lệnh dùng để đặt điều kiện lọc trên từng bộ (từng dòng)
- ✓ **HAVING** cũng là câu lệnh đặt điều kiện nhưng là ở trên 1 nhóm xác định, thường đi kèm với câu lệnh GROUP BY.



## ➤ *Cú pháp tổng quát*

*SELECT* <danh sách các cột>  
*FROM* <danh sách các bảng>  
*WHERE* <điều kiện>  
*GROUP BY* <danh sách các cột gom nhóm>  
*HAVING* <điều kiện trên nhóm>  
*ORDER BY* <tên cột> [*ASC* | *DESC*]

