

---

# Chương 4

## TRUY VẤN DỮ LIỆU SQL

---

---

# Mục tiêu

- Sau khi học xong chương này, SV có thể:
  - Nêu định nghĩa và trình bày các nhóm lệnh SQL.
  - Nắm vững cú pháp và viết được các lệnh để truy vấn và thao tác dữ liệu.
  - Tạo, cập nhật và xóa view.

---

# Nội dung chi tiết

- Giới thiệu ngôn ngữ SQL
- Câu truy vấn Select
- Câu truy vấn Insert
- Câu truy vấn Update
- Câu truy vấn Delete
- Câu truy vấn Con
- Tạo, cập nhật và xóa View
- Bài tập ứng dụng

# Giới thiệu ngôn ngữ SQL

- SQL (Structured Query Language) là ngôn ngữ truy vấn có cấu trúc.
- SQL cho phép định nghĩa, truy vấn, thao tác và điều khiển dữ liệu.
- Đa số các Hệ quản trị CSDL hiện nay sử dụng SQL (Microsoft SQL Server, Microsoft Access, Oracle, MySQL, ...)

# Giới thiệu ngôn ngữ SQL

- SQL được chia thành 4 nhóm:
  - Nhóm định nghĩa dữ liệu (DDL: Data Definition Language): gồm các lệnh tạo, thay đổi các bảng dữ liệu (create, drop, alter, ...)
  - Nhóm truy vấn dữ liệu (DQL: Data Query Language): gồm các lệnh truy vấn lựa chọn (select) để lấy thông tin nhưng không làm thay đổi dữ liệu trong các bảng.
  - Nhóm thao tác dữ liệu (DML: Data Manipulation Language): gồm các lệnh làm thay đổi dữ liệu lưu trong các bảng (insert, delete, update,...)
  - Nhóm điều khiển dữ liệu (DCL: Data Control Language): gồm các lệnh quản lý quyền truy nhập vào dữ liệu và các bảng (grant, revoke, ...)

# Câu truy vấn Select

## ■ Cú pháp câu lệnh SELECT:

**SELECT [DISTINCT] Column(s)**

**FROM TableName, Views**

**[WHERE Conditions]**

**[GROUP BY Row(s)]**

**[HAVING Conditions]**

**[ORDER BY Column(s) [asc|desc]]**

- Các mệnh đề trong cặp dấu **[ ]** không bắt buộc
- **DISTINCT** có thể là: **Distinct**: trả về các bản ghi không trùng lặp nhau hoặc **Top**: trả về n (hay %) bản ghi tìm thấy từ trên xuống.
- Mệnh đề **WHERE** cho phép truy vấn lựa chọn theo hàng
- Mệnh đề **GROUP BY** cho phép nhóm dữ liệu theo hàng
- Mệnh đề **HAVING** cho phép truy vấn lựa chọn theo nhóm
- Mệnh đề **ORDER BY** cho phép sắp xếp dữ liệu theo cột

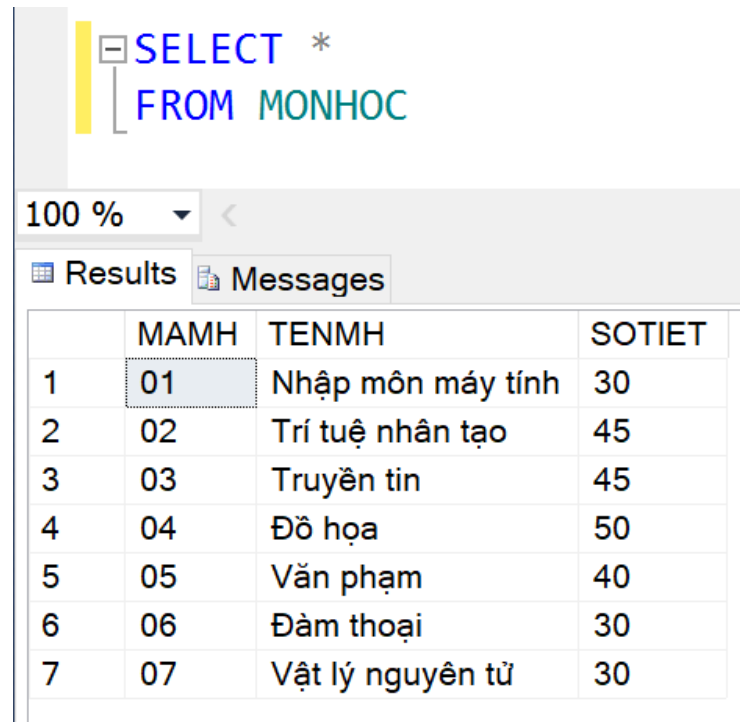
# Câu truy vấn Select

- Truy vấn lựa chọn tất cả các hàng và cột:

**SELECT \***

**FROM TableName**

**VD:** Hiển thị thông tin tất cả các môn học.



The screenshot shows a SQL query editor with a yellow vertical bar on the left. The query text is: `SELECT *` on the first line and `FROM MONHOC` on the second line. Below the query editor, there is a toolbar with a dropdown menu set to "100 %". Below the toolbar, there are two tabs: "Results" (active) and "Messages". The "Results" tab displays a table with 4 columns: an index, "MAMH", "TENMH", and "SOTIET". The table contains 7 rows of data.

	MAMH	TENMH	SOTIET
1	01	Nhập môn máy tính	30
2	02	Trí tuệ nhân tạo	45
3	03	Truyền tin	45
4	04	Đồ họa	50
5	05	Văn phạm	40
6	06	Đàm thoại	30
7	07	Vật lý nguyên tử	30

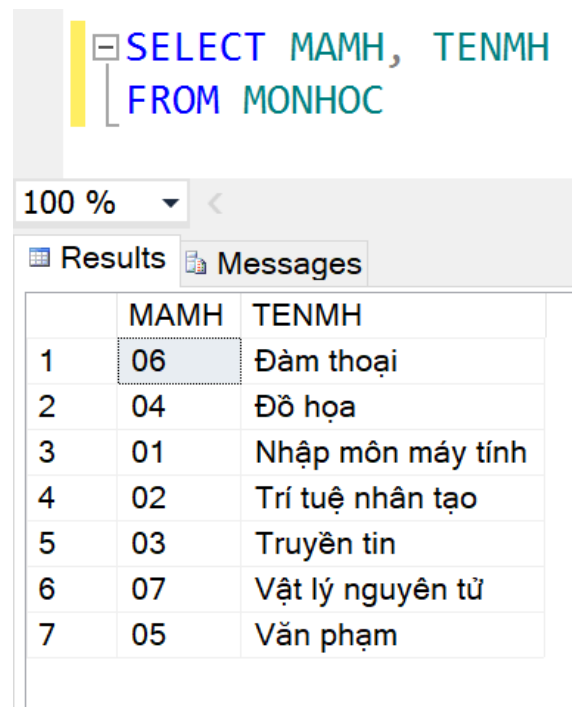
# Câu truy vấn Select

## ■ Truy vấn lựa chọn một số cột:

**SELECT** Column1, Column2 ...

**FROM** TableName

**VD:** Hiển thị thông tin tất cả các môn học bao gồm: mã môn học và tên môn học.



The screenshot shows a SQL query editor with the following text:

```
SELECT MAMH, TENMH  
FROM MONHOC
```

Below the editor, the 'Results' tab is active, displaying a table with 7 rows and 2 columns: MAMH and TENMH. The first row is highlighted.

	MAMH	TENMH
1	06	Đàm thoại
2	04	Đồ họa
3	01	Nhập môn máy tính
4	02	Trí tuệ nhân tạo
5	03	Truyền tin
6	07	Vật lý nguyên tử
7	05	Văn phạm



# Câu truy vấn Select

## ■ Mệnh đề WHERE:

**SELECT** [DISTINCT] Column(s)

**FROM** TableName

**WHERE** Conditions

- Một số toán tử (Operator) thường sử dụng trong biểu thức Conditions:

=	So sánh bằng
<> hoặc !=	Khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
BETWEEN value1 AND value2	So sánh nằm trong khoảng value1 và value2
LIKE	So sánh chuỗi tương đối

# Câu truy vấn Select

## ■ Mệnh đề WHERE:

- VD1: Hiển thị thông tin các sinh viên có học bổng là 100000.

```
SELECT *  
FROM SINHVIEN  
WHERE HOCBONG=100000
```

100 %

Results Messages

	MASV	HOSV	TENSV	PHAI	NGAYSINH	NOISINH	MAKH	HOCBONG
1	A01	Nguyễn Thu	Hải	0	1980-02-23 00:00:00.000	TP.HCM	AV	100000
2	A02	Trần Văn	Chính	1	1982-12-24 00:00:00.000	TP.HCM	TH	100000

- VD2: Hiển thị thông tin các sinh viên có học bổng từ 50000 đến 100000.

```
SELECT *  
FROM SINHVIEN  
WHERE HOCBONG BETWEEN 50000 AND 100000
```

100 %

Results Messages

	MASV	HOSV	TENSV	PHAI	NGAYSINH	NOISINH	MAKH	HOCBONG
1	A01	Nguyễn Thu	Hải	0	1980-02-23 00:00:00.000	TP.HCM	AV	100000
2	A02	Trần Văn	Chính	1	1982-12-24 00:00:00.000	TP.HCM	TH	100000
3	A04	Trần Anh	Tuấn	1	1984-12-08 00:00:00.000	Long An	LS	80000
4	A05	Trần Thanh	Triều	1	1980-02-01 00:00:00.000	Hà Nội	VL	80000
5	B03	Trần Thị	Thanh	0	1982-12-31 00:00:00.000	TP.HCM	TH	50000

# Câu truy vấn Select

## ■ Toán tử LIKE:

- Cho phép so sánh một chuỗi với chuỗi khác (có chứa các ký tự đại diện).
- Các ký tự đại diện (Wildcard) bao gồm:
  - % : đại diện 1 chuỗi các ký tự
  - \_ (Gạch dưới): đại diện 1 ký tự
- VD: Hiển thị thông tin các sinh viên họ Trần.

```
SELECT *  
FROM SINHVIEN  
WHERE HOSV LIKE N'Trần%'
```

100 % <

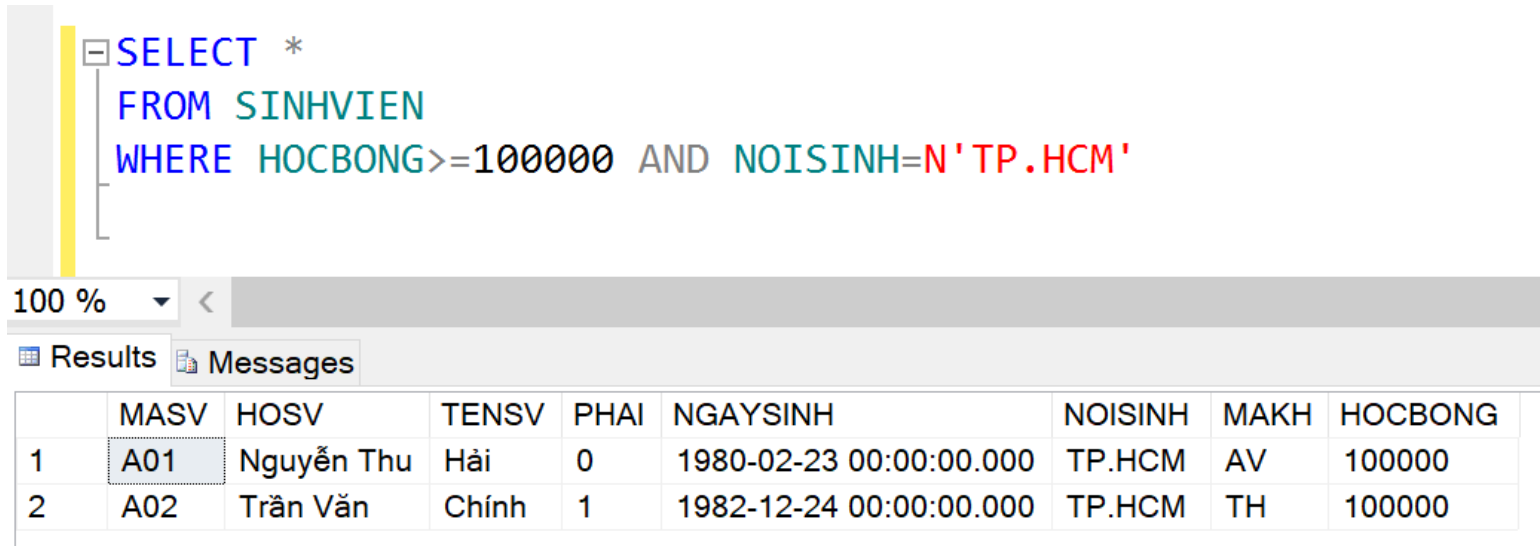
Results Messages

	MASV	HOSV	TENSV	PHAI	NGAYSINH	NOISINH	MAKH	HOCBONG
1	A02	Trần Văn	Chính	1	1982-12-24 00:00:00.000	TP.HCM	TH	100000
2	A04	Trần Anh	Tuấn	1	1984-12-08 00:00:00.000	Long An	LS	80000
3	A05	Trần Thanh	Triều	1	1980-02-01 00:00:00.000	Hà Nội	VL	80000
4	B01	Trần Thanh	Mai	0	1981-12-20 00:00:00.000	Bến Tre	TH	200000
5	B02	Trần Thị Thu	Thùy	0	1982-02-13 00:00:00.000	TP.HCM	TH	30000
6	B03	Trần Thị	Thanh	0	1982-12-31 00:00:00.000	TP.HCM	TH	50000

# Câu truy vấn Select

## ■ Toán tử logic:

- Các toán tử logic gồm: AND, OR và NOT.
- AND và OR được sử dụng để kết nối các điều kiện tìm kiếm trong mệnh đề WHERE.
- NOT phủ định kết quả tìm kiếm.
- **VD:** Hiển thị thông tin các sinh viên có học bổng từ 100000 trở lên và nơi sinh là TP.HCM.



```
SELECT *  
FROM SINHVIEN  
WHERE HOCBONG >= 100000 AND NOISINH = N'TP.HCM'
```

100 %

Results Messages

	MASV	HOSV	TENSV	PHAI	NGAYSINH	NOISINH	MAKH	HOCBONG
1	A01	Nguyễn Thu	Hải	0	1980-02-23 00:00:00.000	TP.HCM	AV	100000
2	A02	Trần Văn	Chính	1	1982-12-24 00:00:00.000	TP.HCM	TH	100000

# Câu truy vấn Select

## ■ Mệnh đề ORDER BY:

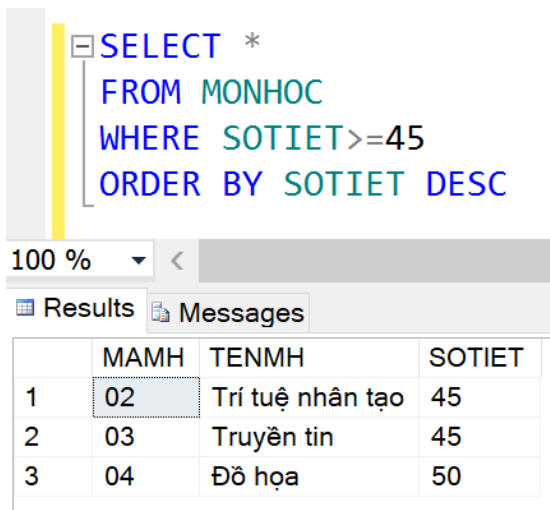
**SELECT [DISTINCT] Column(s)**

**FROM TableName**

**[WHERE Conditions ]**

**ORDER BY Column(s) [asc|desc]**

- Cho phép sắp xếp kết quả truy vấn tăng dần (asc) hoặc giảm dần (desc) theo cột.
- **VD:** Hiển thị thông tin các môn học có số tiết từ 45 trở lên, sắp xếp giảm dần theo cột số tiết.



The screenshot shows a SQL query editor with the following text:

```
SELECT *  
FROM MONHOC  
WHERE SOTIET >= 45  
ORDER BY SOTIET DESC
```

Below the query editor, there is a toolbar with a zoom dropdown set to 100% and a back arrow. Below the toolbar are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a table with the following data:

	MAMH	TENMH	SOTIET
1	02	Trí tuệ nhân tạo	45
2	03	Truyền tin	45
3	04	Đồ họa	50

# Câu truy vấn Select

## ■ Mệnh đề GROUP BY:

**SELECT [DISTINCT] Column(s)**

**FROM TableName, Views**

**[WHERE Conditions]**

**[GROUP BY Row(s)]**

**[ORDER BY Column(s) [asc|desc]]**

- Cho phép nhóm dữ liệu theo hàng.
- Mệnh đề này phải có khi câu truy vấn có sử dụng hàm thống kê và có lấy thêm các cột không sử dụng hàm thống kê trên mệnh đề Select.
- Các Hàm thống kê:
  - **Sum (Column)**: tính tổng giá trị tại cột Column
  - **Count (Column / \*)**: đếm số dòng
  - **Avg (Column)**: tìm giá trị trung bình cộng trong cột Column
  - **Max (Column)**: tìm giá trị lớn nhất trong cột Column
  - **Min (Column)**: tìm giá trị nhỏ nhất trong cột Column

# Câu truy vấn Select

## ■ Mệnh đề GROUP BY:

- **VD1:** Tìm học bổng nhỏ nhất, lớn nhất, học bổng trung bình và tổng học bổng của sinh viên

```
SELECT MIN(HOCBONG) AS HB_MIN, MAX(HOCBONG) AS HB_MAX, AVG(HOCBONG) AS HB_TB, SUM(HOCBONG) AS TONG_HB  
FROM SINHVIEN
```

100 % <

Results Messages

	HB_MIN	HB_MAX	HB_TB	TONG_HB
1	30000	200000	97500	780000

- **VD2:** Tìm học bổng nhỏ nhất, lớn nhất, học bổng trung bình và tổng học bổng của sinh viên từng khoa

```
SELECT MAKH, MIN(HOCBONG) AS HB_MIN, MAX(HOCBONG) AS HB_MAX, AVG(HOCBONG) AS HB_TB, SUM(HOCBONG) AS TONG_HB  
FROM SINHVIEN  
GROUP BY MAKH
```

100 % <

Results Messages

	MAKH	HB_MIN	HB_MAX	HB_TB	TONG_HB
1	AV	100000	140000	120000	240000
2	LS	80000	80000	80000	80000
3	TH	30000	200000	95000	380000
4	VL	80000	80000	80000	80000

# Câu truy vấn Select

## ■ Mệnh đề HAVING:

**SELECT** [DISTINCT] Column(s)

**FROM** TableName, Views

[**WHERE** Conditions]

[**GROUP BY** Row(s)]

[**HAVING** Conditions]

[**ORDER BY** Column(s) [asc|desc]]

- Được sử dụng làm điều kiện nhóm và phải kết hợp với GROUP BY.
- **VD:** Tìm học bổng nhỏ nhất của sinh viên từng khoa, với điều kiện học bổng nhỏ nhất trên 50000.

```
SELECT MAKH, MIN(HOCBONG) AS HB_MIN
FROM SINHVIEN
GROUP BY MAKH
HAVING MIN(HOCBONG) > 50000
```

100 % <

Results Messages

	MAKH	HB_MIN
1	AV	100000
2	LS	80000
3	VL	80000



# Câu truy vấn Select

## ■ Truy vấn dữ liệu từ nhiều bảng:

- Khi truy vấn trên nhiều bảng, phải kết nối các bảng.
- Có hai kiểu kết nối:
  - **Kết nối trong**: mệnh đề WHERE chỉ ra các trường khóa của các bảng cần kết nối phải như nhau hoặc dùng từ khóa **(INNER) JOIN** trong mệnh đề FROM.
  - **Kết nối ngoài**: sử dụng từ khóa **LEFT/RIGHT/FULL JOIN** trong mệnh đề FROM.

**Bảng\_1 inner join Bảng\_2 on Bảng\_1.Tên\_cột = Bảng\_2.Tên\_cột**

**Bảng\_1 left join Bảng\_2 on Bảng\_1.Tên\_cột = Bảng\_2.Tên\_cột**

**Bảng\_1 right join Bảng\_2 on Bảng\_1.Tên\_cột = Bảng\_2.Tên\_cột**

**Bảng\_1 full join Bảng\_2 on Bảng\_1.Tên\_cột = Bảng\_2.Tên\_cột**

# Câu truy vấn Select

## ■ Truy vấn dữ liệu từ nhiều bảng:

- Nếu hai bảng liên kết với nhau qua mệnh đề **inner join** thì kết quả trả về là những dòng của hai bảng có giá trị bằng nhau tại cột liên kết.
- Nếu hai bảng liên kết với nhau qua mệnh đề **left join** (**right join**) thì kết quả trả về là những dòng của hai bảng có giá trị bằng nhau tại cột liên kết và các dòng của bảng bên trái (bên phải) có dữ liệu không bằng dữ liệu của cột liên kết trong bảng kia.
- Nếu hai bảng liên kết với nhau qua mệnh đề **full join** thì kết quả trả về là những dòng của hai bảng có giá trị bằng nhau tại cột liên kết và các dòng dữ liệu của cả hai bảng không bằng nhau tại cột liên kết.

# Câu truy vấn Select

## ■ Truy vấn dữ liệu từ nhiều bảng:

- **VD1:** Hiển thị thông tin Mã SV, Họ và tên, Ngày sinh và Tên khoa tương ứng.

```
SELECT MASV, HOSV+ ' '+TENSX AS HO_TEN_SV, NGAYSINH, TENKH  
FROM SINHVIEN, KHOA  
WHERE SINHVIEN.MAKH=KHOA.MAKH
```

100 % <

Results Messages

	MASV	HO_TEN_SV	NGAYSINH	TENKH
1	A01	Nguyễn Thu Hải	1980-02-23 00:00:00.000	Anh Văn
2	A02	Trần Văn Chính	1982-12-24 00:00:00.000	Tin học
3	A03	Lê Thu Bạch Yến	1982-12-12 00:00:00.000	Anh Văn
4	A04	Trần Anh Tuấn	1984-12-08 00:00:00.000	Lịch sử
5	A05	Trần Thanh Triều	1980-02-01 00:00:00.000	Vật lý
6	B01	Trần Thanh Mai	1981-12-20 00:00:00.000	Tin học
7	B02	Trần Thị Thu Thủy	1982-02-13 00:00:00.000	Tin học
8	B03	Trần Thị Thanh	1982-12-31 00:00:00.000	Tin học

Hoặc:

```
SELECT MASV, HOSV+ ' '+TENSX AS HO_TEN_SV, NGAYSINH, TENKH  
FROM SINHVIEN INNER JOIN KHOA ON SINHVIEN.MAKH=KHOA.MAKH
```

100 % <

Results Messages

	MASV	HO_TEN_SV	NGAYSINH	TENKH
1	A01	Nguyễn Thu Hải	1980-02-23 00:00:00.000	Anh Văn
2	A02	Trần Văn Chính	1982-12-24 00:00:00.000	Tin học
3	A03	Lê Thu Bạch Yến	1982-12-12 00:00:00.000	Anh Văn
4	A04	Trần Anh Tuấn	1984-12-08 00:00:00.000	Lịch sử
5	A05	Trần Thanh Triều	1980-02-01 00:00:00.000	Vật lý
6	B01	Trần Thanh Mai	1981-12-20 00:00:00.000	Tin học
7	B02	Trần Thị Thu Thủy	1982-02-13 00:00:00.000	Tin học
8	B03	Trần Thị Thanh	1982-12-31 00:00:00.000	Tin học

# Câu truy vấn Select

## ■ Truy vấn dữ liệu từ nhiều bảng:

- VD2: Thống kê tổng số SV từng khoa.

```
SELECT KHOA.MAKH, TENKH, COUNT(MASV) AS TONG_SO_SV  
FROM KHOA INNER JOIN SINHVIEN ON KHOA.MAKH=SINHVIEN.MAKH  
GROUP BY KHOA.MAKH, TENKH
```

100 % <

Results Messages

	MAKH	TENKH	TONG_SO_SV
1	AV	Anh Văn	2
2	LS	Lịch sử	1
3	TH	Tin học	4
4	VL	Vật lý	1

- VD3: Thống kê tổng số SV từng khoa, kể cả những khoa chưa có SV.

Hoặc:

```
SELECT KHOA.MAKH, TENKH, COUNT(MASV) AS TONG_SO_SV  
FROM KHOA LEFT JOIN SINHVIEN ON KHOA.MAKH=SINHVIEN.MAKH  
GROUP BY KHOA.MAKH, TENKH
```

100 % <

Results Messages

	MAKH	TENKH	TONG_SO_SV
1	AV	Anh Văn	2
2	LS	Lịch sử	1
3	SH	Sinh học	0
4	TH	Tin học	4
5	TR	Triết	0
6	VL	Vật lý	1

```
SELECT KHOA.MAKH, TENKH, COUNT(MASV) AS TONG_SO_SV  
FROM SINHVIEN RIGHT JOIN KHOA ON SINHVIEN.MAKH=KHOA.MAKH  
GROUP BY KHOA.MAKH, TENKH
```

100 % <

Results Messages

	MAKH	TENKH	TONG_SO_SV
1	AV	Anh Văn	2
2	LS	Lịch sử	1
3	SH	Sinh học	0
4	TH	Tin học	4
5	TR	Triết	0
6	VL	Vật lý	1

# Câu truy vấn Select

## ■ Truy vấn dữ liệu từ nhiều bảng:

- **VD4:** Hiển thị kết quả thi của các SV, kể cả các SV chưa thi môn nào.

```
SELECT SINHVIEN.MASV, HOSV+' '+TENSX AS HO_TEN_SV, TENMH, DIEM
FROM SINHVIEN LEFT JOIN KETQUA ON SINHVIEN.MASV=KETQUA.MASV LEFT JOIN MONHOC ON KETQUA.MAMH=MONHOC.MAMH
```

100 %

Results Messages

	MASV	HO_TEN_SV	TENMH	DIEM
1	A01	Nguyễn Thu Hải	Nhập môn máy tính	10
2	A01	Nguyễn Thu Hải	Trí tuệ nhân tạo	4
3	A01	Nguyễn Thu Hải	Văn phạm	9
4	A01	Nguyễn Thu Hải	Đàm thoại	3
5	A02	Trần Văn Chính	Nhập môn máy tính	5
6	A03	Lê Thu Bạch Yến	Trí tuệ nhân tạo	5
7	A03	Lê Thu Bạch Yến	Đồ họa	10
8	A03	Lê Thu Bạch Yến	Đàm thoại	10
9	A04	Trần Anh Tuấn	Trí tuệ nhân tạo	4
10	A04	Trần Anh Tuấn	Đồ họa	6
11	A05	Trần Thanh Triều	NULL	NULL
12	B01	Trần Thanh Mai	Nhập môn máy tính	0
13	B01	Trần Thanh Mai	Đồ họa	8
14	B02	Trần Thị Thu Thủy	Truyền tin	6
15	B02	Trần Thị Thu Thủy	Đồ họa	8
16	B03	Trần Thị Thanh	Trí tuệ nhân tạo	10
17	B03	Trần Thị Thanh	Truyền tin	9

# Câu truy vấn Select

## ■ Truy vấn dữ liệu từ nhiều bảng:

- **VD5:** Hiển thị kết quả thi của các SV, kể cả các môn học chưa có SV thi.

```
SELECT SINHVIEN.MASV, HOSV+' '+TENSX AS HO_TEN_SV, TENMH, DIEM  
FROM SINHVIEN INNER JOIN KETQUA ON SINHVIEN.MASV=KETQUA.MASV RIGHT JOIN MONHOC ON KETQUA.MAMH=MONHOC.MAMH
```

100 %

Results Messages

	MASV	HO_TEN_SV	TENMH	DIEM
1	A01	Nguyễn Thu Hải	Đàm thoại	3
2	A03	Lê Thu Bạch Yến	Đàm thoại	10
3	A03	Lê Thu Bạch Yến	Đồ họa	10
4	A04	Trần Anh Tuấn	Đồ họa	6
5	B01	Trần Thanh Mai	Đồ họa	8
6	B02	Trần Thị Thu Thủy	Đồ họa	8
7	A01	Nguyễn Thu Hải	Nhập môn máy tính	10
8	A02	Trần Văn Chính	Nhập môn máy tính	5
9	B01	Trần Thanh Mai	Nhập môn máy tính	0
10	A01	Nguyễn Thu Hải	Trí tuệ nhân tạo	4
11	A03	Lê Thu Bạch Yến	Trí tuệ nhân tạo	5
12	A04	Trần Anh Tuấn	Trí tuệ nhân tạo	4
13	B03	Trần Thị Thanh	Trí tuệ nhân tạo	10
14	B02	Trần Thị Thu Thủy	Truyền tin	6
15	B03	Trần Thị Thanh	Truyền tin	9
16	NULL	NULL	Vật lý nguyên tử	NULL
17	A01	Nguyễn Thu Hải	Văn phạm	9

# Câu truy vấn Select

## ■ Truy vấn dữ liệu từ nhiều bảng:

- **VD6:** Hiển thị kết quả thi của các SV, kể cả các SV chưa thi môn nào và các môn học chưa có SV thi.

```
SELECT SINHVIEN.MASV, HOSV+' '+TENSX AS HO_TEN_SX, TENMH, DIEM
FROM SINHVIEN LEFT JOIN KETQUA ON SINHVIEN.MASV=KETQUA.MASV FULL JOIN MONHOC ON KETQUA.MAMH=MONHOC.MAMH
```

100 % <

Results Messages

	MASV	HO_TEN_SX	TENMH	DIEM
1	A01	Nguyễn Thu Hải	Nhập môn máy tính	10
2	A01	Nguyễn Thu Hải	Trí tuệ nhân tạo	4
3	A01	Nguyễn Thu Hải	Văn phạm	9
4	A01	Nguyễn Thu Hải	Đàm thoại	3
5	A02	Trần Văn Chính	Nhập môn máy tính	5
6	A03	Lê Thu Bạch Yến	Trí tuệ nhân tạo	5
7	A03	Lê Thu Bạch Yến	Đồ họa	10
8	A03	Lê Thu Bạch Yến	Đàm thoại	10
9	A04	Trần Anh Tuấn	Trí tuệ nhân tạo	4
10	A04	Trần Anh Tuấn	Đồ họa	6
11	A05	Trần Thanh Triều	NULL	NULL
12	B01	Trần Thanh Mai	Nhập môn máy tính	0
13	B01	Trần Thanh Mai	Đồ họa	8
14	B02	Trần Thị Thu Thủy	Truyền tin	6
15	B02	Trần Thị Thu Thủy	Đồ họa	8
16	B03	Trần Thị Thanh	Trí tuệ nhân tạo	10
17	B03	Trần Thị Thanh	Truyền tin	9
18	NULL	NULL	Vật lý nguyên tử	NULL

# Câu truy vấn Select

- Tạo bảng có cấu trúc và dữ liệu từ câu truy vấn:
- Cú pháp:

**select** [Tên\_bảng.]Tên\_cột [, ...] **into** Tên\_bảng\_mới  
**from** ...

- VD1: Tạo bảng có tên MONHOC\_DELETE, gồm các thông tin sau: Mã môn, Tên môn, Số tiết

```
select * into MONHOC_DELETE  
from MONHOC
```

- VD2: Xóa tất cả những môn học có số tiết dưới 30 trong bảng DeleteTable.

```
delete MONHOC_DELETE  
where SOTIET < 30
```



# Câu truy vấn Insert

## ■ Insert 1 dòng dữ liệu vào bảng:

**insert into Tên\_bảng (Cột\_1, Cột\_2, ...)**

**values (Giá\_trị\_1, Giá\_trị\_2, ...)**

- Có thể lược bỏ (Tên\_cột\_1, Tên\_cột\_2, ...) nếu nhập liệu đúng với thứ tự và cấu trúc các cột đã khai báo trong lúc tạo bảng.
  - **VD1:** insert into KHOA (TENKH, MAKH) values (N'Anh Văn', 'AV')
  - **VD2:** insert into KHOA values ('AV', N'Anh Văn')
  - **VD3:** insert into MONHOC values('01', N'Nhập môn máy tính', 30)
  - **VD4:** insert into SINHVIEN values ('A01', N'Nguyễn Thu', N'Hải', 0, '1980-02-23', N'TP.HCM', 'AV', 100000)

# Câu truy vấn Insert

## ■ Insert nhiều dòng dữ liệu vào bảng:

```
insert into Tên_bảng_đích (Tên_cột_1, Tên_cột_2 [...])  
select Tên_bảng_nguồn.Tên_cột, Giá_trị, Biểu_thức, ...  
from ...
```

### - VD: Thêm vào bảng kết quả gồm các thông tin sau:

- Mã sinh viên: lấy tất cả những sinh viên của khoa Tin học
- Mã môn học: 06
- Điểm: 7

```
insert into KETQUA(MASV, MAMH, DIEM)  
select MASV, '06', 7  
from SINHVIEN  
where MAKH = 'TH'
```

Hoặc:

```
insert into KETQUA  
select MASV, '06', 7  
from SINHVIEN  
where MAKH = 'TH'
```

# Câu truy vấn Update

## ■ Cập nhật dữ liệu bảng:

update Tên\_bảng

set Tên\_cột = giá\_trị\_hoặc\_biểu\_thức

[from ...]

[where điều\_kiện\_cập\_nhật]

- VD1: Cập nhật số tiết của môn Văn phạm thành 45 tiết:

```
update MONHOC  
set SOTIET=45  
where TENMH=N'Văn phạm'
```

- VD2: Cộng thêm 5 điểm môn Trí Tuệ Nhân Tạo cho các sinh viên thuộc khoa Anh văn (điểm tối đa là 10 điểm):

```
update KETQUA  
set DIEM = case when DIEM + 5 > 10 then 10 else DIEM + 5 end  
from SINHVIEN inner join KETQUA on SINHVIEN.MASV=KETQUA.MASV inner join MONHOC on KETQUA.MAMH=MONHOC.MAMH  
where MAKH = 'AV' and TENMH = N'Trí tuệ nhân tạo'
```

# Câu truy vấn Delete

## ■ Xóa dữ liệu bảng:

`delete Tên_bảng`

`[from ...]`

`[where điều_kiện_xóa]`

- VD1: Xóa tất cả các sinh viên không có học bổng:

```
delete SINHVIEN  
where HOCBONG = 0
```

- VD2: Xóa kết quả thi của sinh viên khoa Anh Văn:

```
delete KETQUA  
from SINHVIEN inner join KETQUA on SINHVIEN.MASV=KETQUA.MASV  
where MAKH = 'AV'
```

# Câu truy vấn Con

## ■ SubQuery:

- Là một truy vấn nằm trong một truy vấn khác, có kết quả trả về là điều kiện để so sánh, gán giá trị cho câu truy vấn khác.
- Chỉ trả về 1 cột dữ liệu. Có thể là:
  - 1 dòng: dùng các toán tử so sánh giá trị: =, <, >, <=, >=, <>
  - Nhiều dòng: dùng các toán tử so sánh tập hợp: in (nằm trong), not in (không nằm trong), >=ALL (lớn nhất).
- VD1: Tìm các SV chưa thi môn nào:

```
select *  
from SINHVIEN  
where MASV not in (select MASV from KETQUA)
```

- VD2: Xóa môn học chưa có SV thi :

```
delete MONHOC  
where MAMH not in (select MAMH from KETQUA)
```

# Tạo, cập nhật và xóa View

- View là một bảng ảo dựa trên tập kết quả của một câu lệnh SQL, chứa các hàng và các cột như một bảng thực, các trường trong View là các trường của một hoặc nhiều bảng trong database.
- Cú pháp tạo View:  
**CREATE VIEW view\_name AS**  
**SELECT column1, column2, ...**  
**FROM table\_name**  
**WHERE condition**
- Mỗi lần truy vấn một View, hệ thống sẽ sử dụng câu lệnh SQL của View để tạo lại dữ liệu, do đó dữ liệu của View luôn được cập nhật.

# Tạo, cập nhật và xóa View

- **VD1:** Tạo view hiển thị thông tin các SV khoa Tin Học

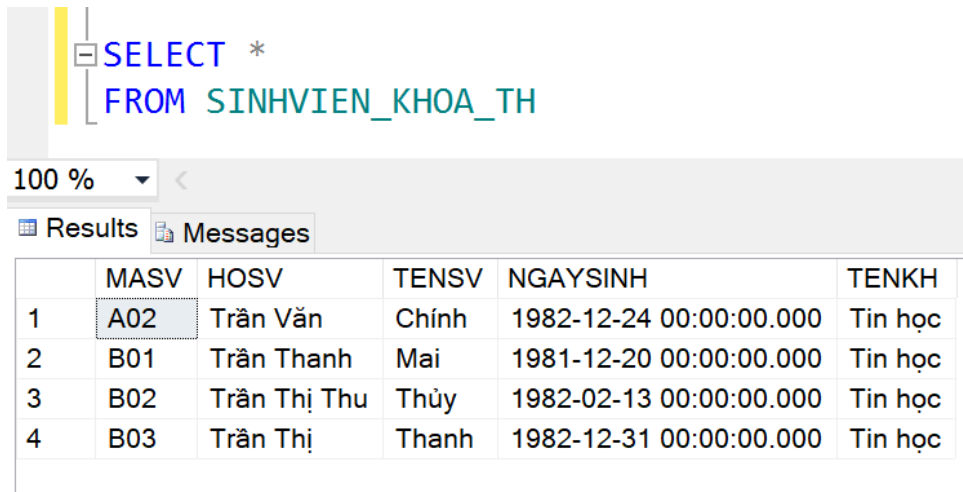
```
CREATE VIEW SINHVIEN_KHOA_TH AS
```

```
SELECT MASV, HOSV, TENS, NGAYSINH, TENKH
```

```
FROM SINHVIEN INNER JOIN KHOA ON SINHVIEN.MAKH=KHOA.MAKH
```

```
WHERE SINHVIEN.MAKH = 'TH'
```

- **VD2:** Hiển thị thông tin của View SINHVIEN\_KHOA\_TH vừa tạo ở trên:



The screenshot shows a database query tool interface. At the top, the SQL query is entered: `SELECT * FROM SINHVIEN_KHOA_TH`. Below the query editor, there is a dropdown menu set to "100 %". Underneath, there are two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with 6 columns: MASV, HOSV, TENS, NGAYSINH, and TENKH. The table contains 4 rows of data.

	MASV	HOSV	TENS	NGAYSINH	TENKH
1	A02	Trần Văn	Chính	1982-12-24 00:00:00.000	Tin học
2	B01	Trần Thanh	Mai	1981-12-20 00:00:00.000	Tin học
3	B02	Trần Thị Thu	Thủy	1982-02-13 00:00:00.000	Tin học
4	B03	Trần Thị	Thanh	1982-12-31 00:00:00.000	Tin học

# Tạo, cập nhật và xóa View

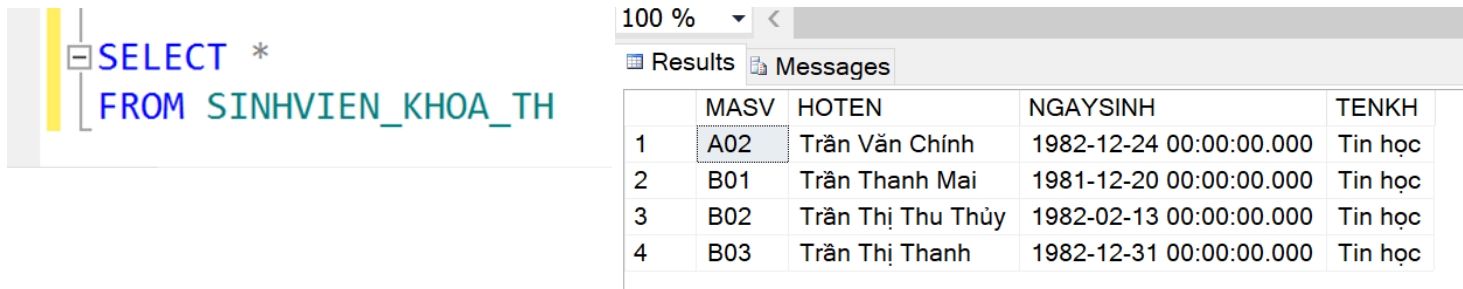
- Cú pháp cập nhật View đã có:

```
ALTER VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition
```

- VD1: Cập nhật View SINHVIEN\_KHOA\_TH đã tạo ở trên:

```
ALTER VIEW SINHVIEN_KHOA_TH AS  
SELECT MASV, HOSV+' '+ TENSU AS HOTEN, NGAYSINH, TENKH  
FROM SINHVIEN INNER JOIN KHOA ON SINHVIEN.MAKH=KHOA.MAKH  
WHERE SINHVIEN.MAKH = 'TH'
```

- VD2: Hiển thị View SINHVIEN\_KHOA\_TH vừa cập nhật:



```
SELECT *  
FROM SINHVIEN_KHOA_TH
```

	MASV	HOTEN	NGAYSINH	TENKH
1	A02	Trần Văn Chính	1982-12-24 00:00:00.000	Tin học
2	B01	Trần Thanh Mai	1981-12-20 00:00:00.000	Tin học
3	B02	Trần Thị Thu Thủy	1982-02-13 00:00:00.000	Tin học
4	B03	Trần Thị Thanh	1982-12-31 00:00:00.000	Tin học



# Tạo, cập nhật và xóa View

- Cú pháp xóa View đã có:

**DROP VIEW** *view\_name*

- VD: Xóa View SINHVIEN\_KHOA\_TH:

**DROP VIEW** *SINHVIEN\_KHOA\_TH*

# Bài tập ứng dụng

- Cho cơ sở dữ liệu QuanLyNhanSu như sau:

NHANVIEN	HONV	TENLOT	TENNV	PHAI	LUONG	MANV	NGSINH	DIACHI	PHG
	Đinh	Lê	Tiên	Nam	4000	123456789	01/09/1965	Nguyễn Trãi, Q5	1
	Nguyễn	Thị	Loan	Nữ	2500	333445555	12/08/1955	Nguyễn Huệ, Q1	5
	Nguyễn	Lan	Anh	Nữ	4300	666884444	15/09/1962	Lê Lợi, Q1	5
	Trần	Thanh	Tâm	Nam	3800	453453453	31/07/1972	Trần Nãi, Q2	2

DEAN	TENDA	MADA	DDIEM_DA	PHG
	Sản phẩm X	1	Quy Nhơn	5
	Sản phẩm Y	2	Nha Trang	5
	Sản phẩm Z	3	TP HCM	5
	Tin học hoá	10	Bình Dương	4

PHONGBAN	PHG	TENPHG
	1	Nhân sự
	2	Kế hoạch
	3	Kinh doanh
	4	Marketing
	5	Kế toán

PHANCONG	MANV	MADA	SOGIO
	123456789	1	32.0
	123456789	2	8.0
	666884444	3	40.0
	453453453	1	20.0

# Bài tập ứng dụng

- Tạo cơ sở dữ liệu trên và viết các câu truy vấn sau:
  1. Hiển thị tất cả thông tin của bảng NHANVIEN.
  2. Hiển thị thông tin của những nhân viên ở phòng số 5, sắp thứ tự theo tên nhân viên giảm dần.
  3. Hiển thị mã nhân viên, họ nhân viên, tên lót và tên nhân viên của những nhân viên ở phòng số 5 và có lương từ 3000 trở lên.
  4. Hiển thị mã nhân viên, tên nhân viên của những nhân viên có lương từ 3000 đến 5000.
  5. Hiển thị thông tin của những nhân viên ở địa chỉ có tên đường là Nguyễn.
  6. Cho biết số lượng nhân viên.
  7. Cho biết số lượng nhân viên trong từng phòng ban.

# Bài tập ứng dụng

- Tạo cơ sở dữ liệu trên và viết các câu truy vấn sau:
  - 9. Cho biết số lượng nhân viên trong từng phòng ban, chỉ hiển thị những phòng ban có từ 2 nhân viên trở lên.
  - 10. Hiển thị thông tin về mã nhân viên, tên nhân viên và tên phòng ban ở phòng kế toán.
  - 11. Thêm nhân viên mới với thông tin như sau: Lê, Ngọc, Yến, Nữ, 4500, 987654321, 07/11/2000, CMT8 Q3, 3.
  - 11. Sửa thông tin của nhân viên mới Lê Ngọc Yến là thuộc Phòng 4.
  - 12. Xóa các phòng ban không có nhân viên.
  - 13. Xóa các đề án chưa phân công nhân viên.
  - 14. Tạo view hiển thị thông tin các nhân viên phòng Kế toán, bao gồm: Mã nhân viên, Họ và tên, Phái, Lương và Tên phòng.

# Câu hỏi trắc nghiệm

## 1. Ngôn ngữ SQL được chia thành 4 nhóm:

- a. Định nghĩa, truy vấn, thao tác và tạo khung nhìn.
- b. Định nghĩa, truy vấn, ràng buộc và điều khiển dữ liệu.
- c. Định nghĩa, truy vấn, cập nhật và điều khiển dữ liệu.
- d. Định nghĩa, truy vấn, ràng buộc và điều khiển dữ liệu.

# Câu hỏi trắc nghiệm

2. Câu lệnh SQL nào cho phép chọn tất cả các cột dữ liệu trong bảng ROOMS?

- a. Select [all] from ROOMS
- b. Select all ROOMS
- c. Select \*.ROOMS
- d. **Select \* from ROOMS**

# Câu hỏi trắc nghiệm

3. Làm thế nào để chọn tất cả các bản ghi từ bảng ROOMS với giá trị trong cột RoomName bắt đầu là 'a'?

- a. **Select \* from ROOMS where RoomName like 'a%'**
- b. Select \* from ROOMS where RoomName like '%a'
- c. Select \* from ROOMS where RoomName = '%a'
- d. Select \* from ROOMS where RoomName = 'a%'

# Câu hỏi trắc nghiệm

4. Muốn chèn thêm bản ghi mới vào bảng ROOMS, ta phải gõ câu lệnh SQL nào?

- a. Insert into ROOMS (RoomName, RoomID) values ('A100', 'Room 100')
- b. **Insert into ROOMS values ('A100', 'Room 100')**
- c. Insert ('A100', 'Room 100') into ROOMS
- d. Insert values ('A100, 'Room 100') into ROOMS



# Câu hỏi trắc nghiệm

5. Câu lệnh SQL nào chp phép sửa 'Room 1' thành 'VIP Room' trong cột RoomName của bảng ROOMS?

- a. Alter ROOMS set RoomName = 'VIP Room' into RoomName = 'Room 1'
- b. Modify ROOMS set RoomName = 'VIP Room' where RoomName = 'Room 1'
- c. Update ROOMS set RoomName = 'VIP Room' into RoomName = 'Room 1'
- d. Update ROOMS set RoomName = 'VIP Room' where RoomName = 'Room 1'