

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

---

# Bài 08

## Truy xuất dữ liệu

# Tóm tắt

- Lệnh SELECT
- Lệnh sắp xếp với ORDER BY



# SELECT

- **SELECT là câu lệnh cốt lõi** được sử dụng để truy xuất dữ liệu trong SQL Server 2014.
- Có thể **xem(view) dữ liệu** của một bảng bằng câu lệnh SELECT.
- Câu lệnh SELECT cho phép **lấy các dòng và các cột** từ một hoặc nhiều bảng.
- Kết quả trả về của lệnh SELECT **là một bảng** hay còn được gọi là tập kết quả (**resultset**).

# SELECT

- Câu lệnh SELECT cũng cho phép **ghép (join) hai bảng** hoặc lấy một tập nhỏ các cột từ một hoặc trong các bảng được ghép.
- Câu lệnh SELECT **xác định các cột** được sử dụng cho một truy vấn.
- Cú pháp của lệnh SELECT gồm có **một dãy các biểu thức**, được phân cách nhau bởi các dấu phẩy (commas).
- Mỗi biểu thức trong câu lệnh sẽ tạo ra một cột trong tập kết quả (**resultset**).
- Thứ tự các cột xuất hiện theo **đúng thứ tự của biểu thức** được chỉ ra trong câu lệnh SELECT.

# SELECT

## Cú pháp

```
SELECT <column_name1>...<column_nameN> FROM <table_name>
```

Trong đó:

- **<table\_name>**: là tên bảng có chứa dữ liệu sẽ được lấy để hiển thị.
- **<column\_name1>...<column\_nameN>**: là danh sách các cột được hiển thị.

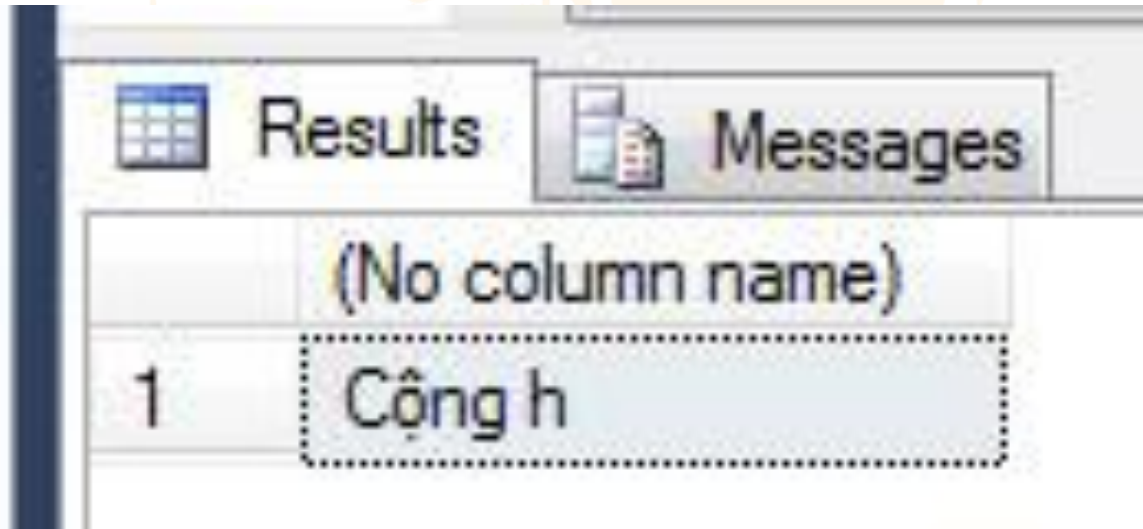
# SELECT

**SELECT không có FROM**: được hỗ trợ kể từ phiên bản SQL Server 2005.

Code:

```
SELECT LEFT(N'Cộng hòa xã hội chủ nghĩa', 6)
```

Kết quả:



	(No column name)
1	Cộng h

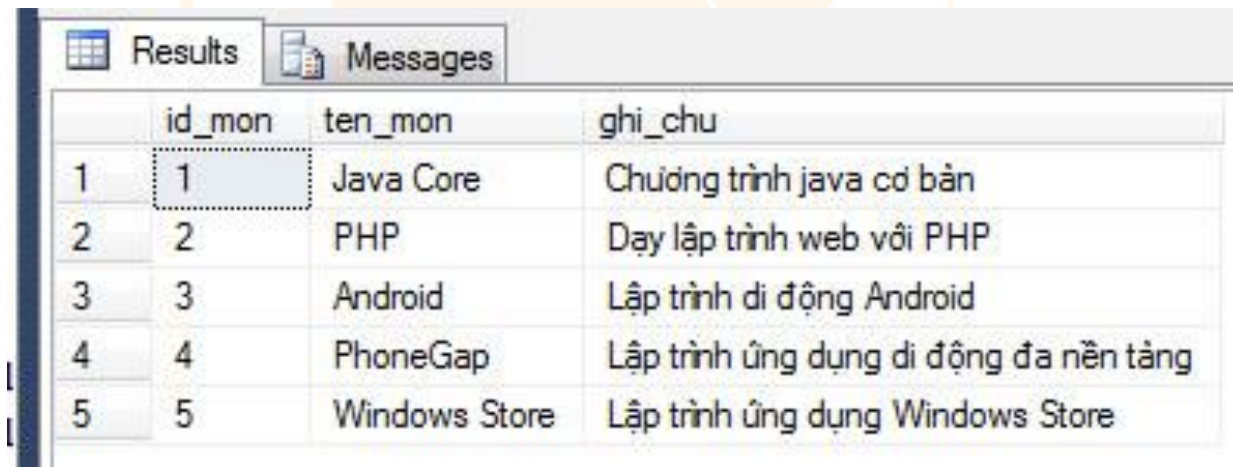
# SELECT

**Hiển thị tất cả các cột:** Dấu hoa thị (\*) thường được sử dụng trong câu lệnh SELECT khi muốn **lấy tất cả các cột** có trong bảng.

Code:

`SELECT * FROM MonHoc`

Kết quả:



	id_mon	ten_mon	ghi_chu
1	1	Java Core	Chương trình java cơ bản
2	2	PHP	Dạy lập trình web với PHP
3	3	Android	Lập trình di động Android
4	4	PhoneGap	Lập trình ứng dụng di động đa nền tảng
5	5	Windows Store	Lập trình ứng dụng Windows Store

# SELECT

## Hiển thị cột được chọn:

- Hiển thị hoặc trả về các cột **do người dùng chọn hoặc chỉ ra** trong câu lệnh.
- Để hiển thị các cột cụ thể, **cần phải biết tên các cột** có trong bảng.

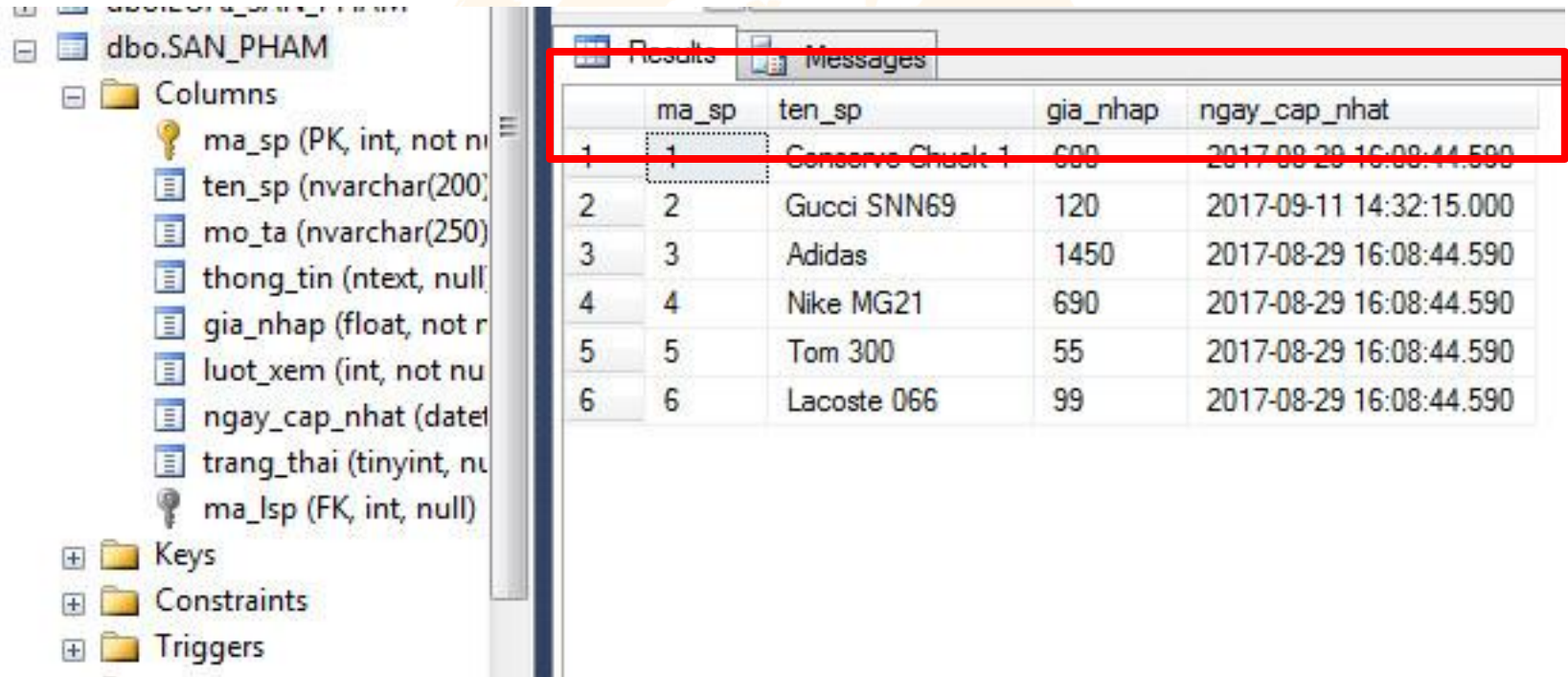
Code:

```
SELECT ma_sp, ten_sp, gia_nhap, ngay_cap_nhat  
FROM SAN_PHAM
```



# SELECT

**Kết quả:**



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Columns' folder for the 'dbo.SAN\_PHAM' table is expanded, showing the following fields:

- ma\_sp (PK, int, not null)
- ten\_sp (nvarchar(200))
- mo\_ta (nvarchar(250))
- thong\_tin (ntext, null)
- gia\_nhap (float, not null)
- luot\_xem (int, not null)
- ngay\_cap\_nhat (datetime)
- trang\_thai (tinyint, null)
- ma\_lsp (FK, int, null)

Below the columns, the 'Keys', 'Constraints', and 'Triggers' folders are also visible. On the right, the 'Results' tab shows the data returned by a SELECT query. A red rectangle highlights the first row of the results.

	ma_sp	ten_sp	gia_nhap	ngay_cap_nhat
1	1	Conserve Chuck 1	600	2017-08-29 16:08:44.590
2	2	Gucci SNN69	120	2017-09-11 14:32:15.000
3	3	Adidas	1450	2017-08-29 16:08:44.590
4	4	Nike MG21	690	2017-08-29 16:08:44.590
5	5	Tom 300	55	2017-08-29 16:08:44.590
6	6	Lacoste 066	99	2017-08-29 16:08:44.590

# SELECT

## Thêm hằng chuỗi ký tự vào kết quả:

- Các hằng chuỗi ký tự được sử dụng khi các cột ký tự được ghép (**joined**).
- Chúng giúp **định dạng đúng cách** và dễ đọc.

Code:

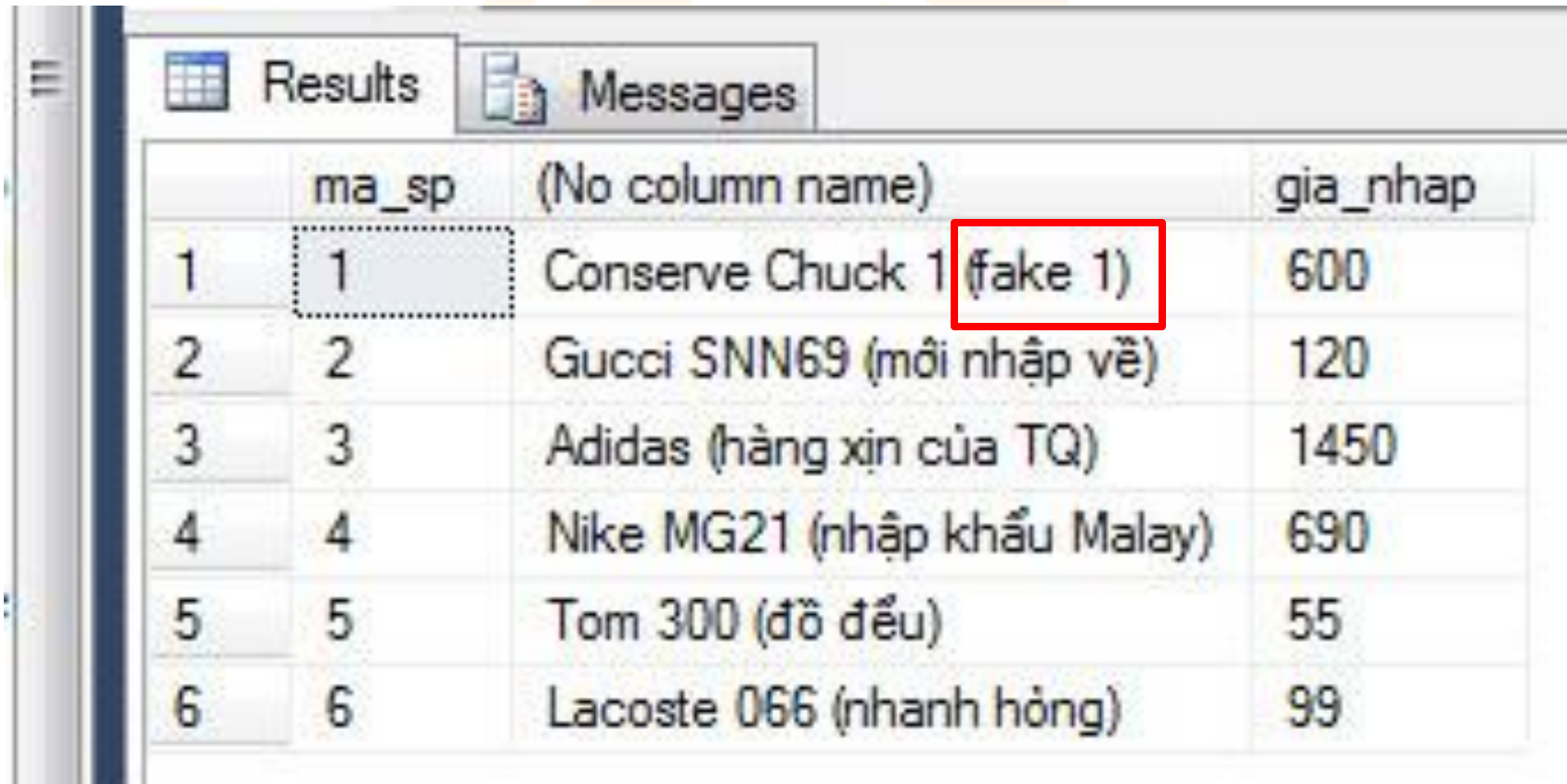
```
SELECT ma_sp, ten_sp + ' (' + mo_ta + ')',  
gia_nhap FROM SAN_PHAM
```

**Bảng gốc với đủ cột**

	ma_sp	ten_sp	mo_ta	gia_nhap
1	1	Conserve Chuck 1	fake 1	600
2	2	Gucci SNN69	môi nhập về	120
3	3	Adidas	hàng xịn của TQ	1450
4	4	Nike MG21	nhập khẩu Malay	690
5	5	Tom 300	đồ đều	55
6	6	Lacoste 066	nhanh hồng	99

# SELECT

Kết quả: ten\_sp đã ghép với mo\_ta.



	ma_sp	(No column name)	gia_nhap
1	1	Conserve Chuck 1 (fake 1)	600
2	2	Gucci SNN69 (môi nhập về)	120
3	3	Adidas (hàng xịn của TQ)	1450
4	4	Nike MG21 (nhập khẩu Malay)	690
5	5	Tom 300 (đồ đều)	55
6	6	Lacoste 066 (nhanch hồng)	99

# SELECT

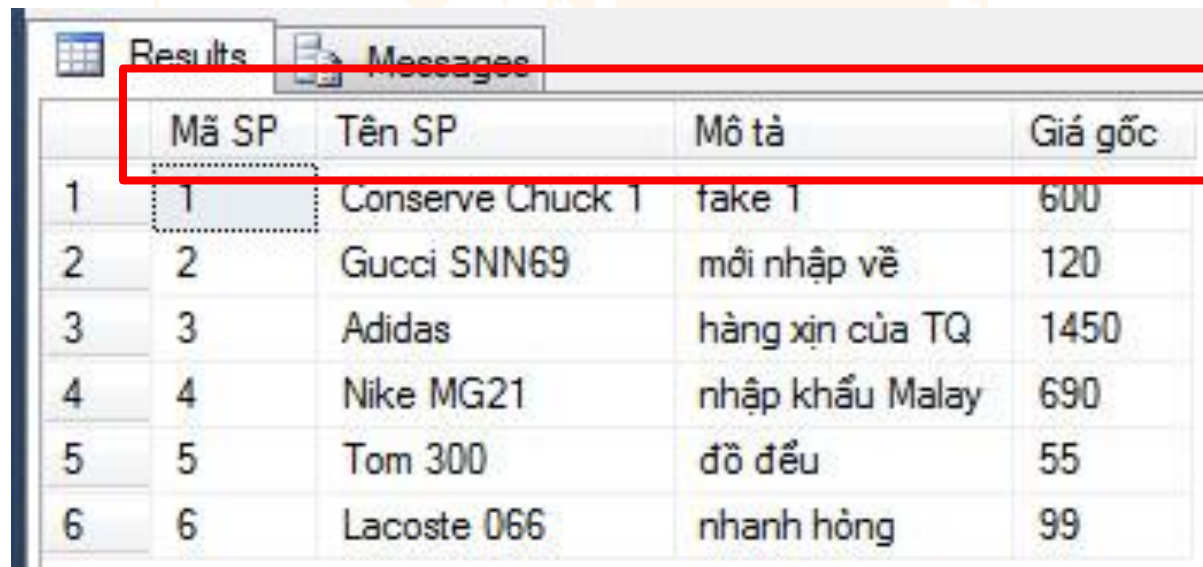
## Đổi tên cột trong tập kết quả trả về:

- Khi các cột được hiển thị trong tập kết quả, chúng đi kèm với **các tiêu đề tương ứng** đã được xác định trong bảng.
- Các tiêu đề này có thể được **thay đổi, đổi tên, đặt lại tên**, hoặc có thể gán một tên mới bằng **mệnh đề AS**.
- Do vậy, bằng cách tùy biến các tiêu đề, sẽ làm chúng trở nên **dễ hiểu hơn** và có ý nghĩa.

# SELECT

Code:

```
SELECT ma_sp AS 'Mã SP', ten_sp AS 'Tên SP',  
mo_ta AS 'Mô tả', gia_nhap AS 'Giá gốc' FROM  
SAN_PHAM
```



	Mã SP	Tên SP	Mô tả	Giá gốc
1	1	Conserve Chuck 1	fake 1	600
2	2	Gucci SNN69	mới nhập về	120
3	3	Adidas	hàng xịn của TQ	1450
4	4	Nike MG21	nhập khẩu Malay	690
5	5	Tom 300	đồ đều	55
6	6	Lacoste 066	nhanh hòng	99

# SELECT

## Tính toán trong tập kết quả trả về:

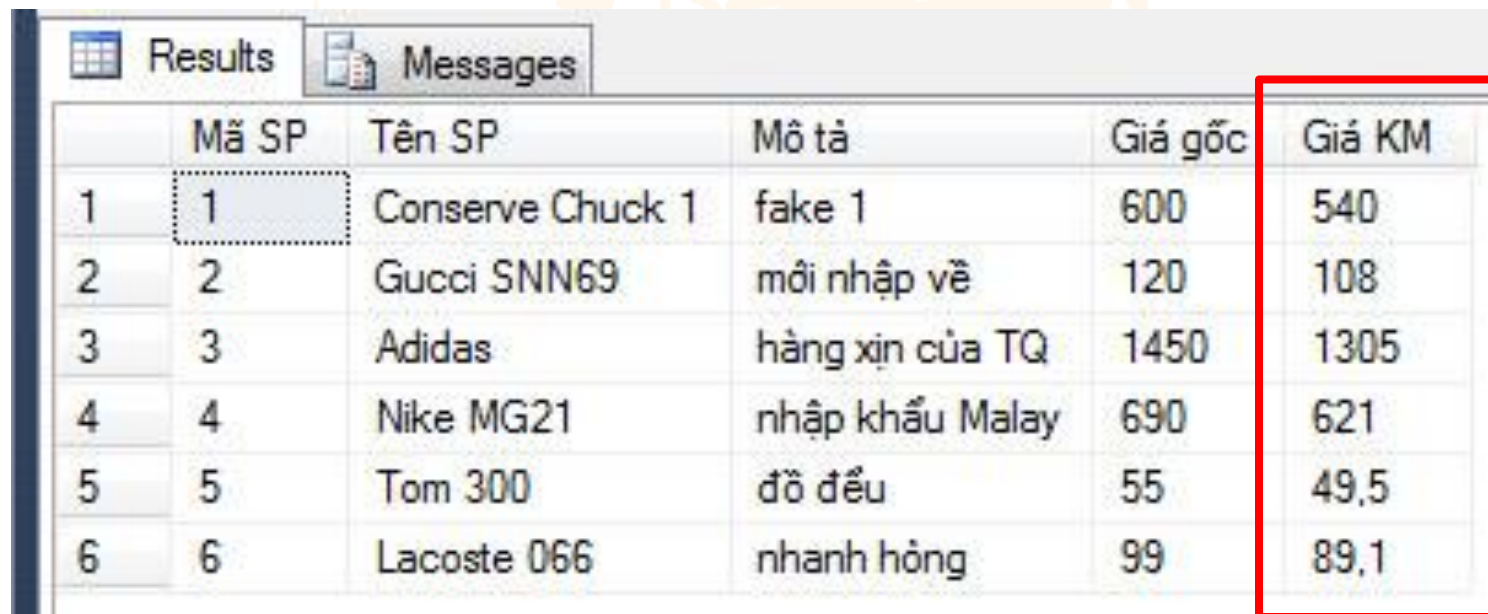
- Một câu lệnh SELECT có thể chứa **các biểu thức toán học** bằng việc áp dụng các toán tử trên một hoặc nhiều cột.
- Nó cho phép tập kết quả có thêm **những giá trị không hề tồn tại trong bảng** cơ sở (base table), mà là do được **tính toán từ các giá trị có sẵn** trong bảng cơ sở (base table).



# SELECT

Code:

```
SELECT ma_sp AS 'Mã SP', ten_sp AS 'Tên SP',  
mo_ta AS 'Mô tả', gia_nhap AS 'Giá gốc',  
gia_nhap * 0.9 AS 'Giá KM' FROM SAN_PHAM
```



	Mã SP	Tên SP	Mô tả	Giá gốc	Giá KM
1	1	Conserve Chuck 1	fake 1	600	540
2	2	Gucci SNN69	mới nhập về	120	108
3	3	Adidas	hàng xịn của TQ	1450	1305
4	4	Nike MG21	nhập khẩu Malay	690	621
5	5	Tom 300	đồ đều	55	49,5
6	6	Lacoste 066	nhanh hòng	99	89,1

# SELECT

## Từ khóa **DISTINCT**:

- Từ khóa DISTINCT dùng để ngăn việc lấy ra các **dòng dữ liệu trùng nhau**
- Nó giúp **loại bỏ** các dòng trùng lặp trong tập kết quả của câu lệnh SELECT.

Ví dụ: Việt Nam có 54 tỉnh thành, muốn lọc quê quán của sinh viên theo học ở Bachkhoa-Aptech gồm những tỉnh thành nào thì:

```
SELECT DISTINCT dia_chi FROM SinhVien
```



# SELECT

Results							
Messages							
	id_sv	ma_sv	ten_sv	gioi_tinh	ngay_sinh	sdt	dia_chi
1	1	K1571	Nguyễn Công Phượng	1	1997-08-16	19001265	Nha Trang
2	2	K1572	Ngô Bảo Châu	1	1976-10-18	123456789	Bắc Giang
3	3	K1573	Phạm Văn Mách	1	1976-08-22	46575675	Bắc Ninh
4	4	K1574	Huỳnh Công Kiệt	1	1986-12-09	76978978	Bắc Giang
5	5	K1575	Lê Huy Ngọ	1	1996-04-09	47876878	Hà Giang
6	6	K1576	Bùi Xuân Phúc	1	1990-11-19	87976324	Nam Định
7	7	K1577	Nguyễn Trung Trực	1	1982-02-28	9875734524234	Hà Nội

	dia_chi
1	Bắc Giang
2	Bắc Ninh
3	Hà Giang
4	Hà Nội
5	Nam Định
6	Nha Trang

Chỉ còn 1 kết quả là tp  
Bắc Giang

# SELECT

## TOP và PERCENT:

- Từ khóa TOP sẽ chỉ lấy ra **một vài dòng đầu tiên** trong tập kết quả.
- Có thể **giới hạn** số dòng cần lấy từ tập kết quả bằng một số hoặc **tỷ lệ phần trăm** (PERCENT).
- Từ khóa TOP cũng có thể sử dụng với những câu lệnh khác như INSERT, UPDATE, và DELETE.

### Cú pháp:

```
SELECT [ALL|DISTINCT] [TOP expression [PERCENT] [WITH TIES]]
```

# SELECT

Results		Messages					
	id_sv	ma_sv	ten_sv	gioi_tinh	ngay_sinh	sdt	dia_chi
1	1	K1571	Nguyễn Công Phượng	1	1997-08-16	19001265	Nha Trang
2	2	K1572	Ngô Bảo Châu	1	1976-10-18	123456789	Bắc Giang
3	3	K1573	Phạm Văn Mách	1	1976-08-22	46575675	Bắc Ninh
4	4	K1574	Huỳnh Công Kiệt	1	1986-12-09	76978978	Bắc Giang
5	5	K1575	Lê Huy Ngọ	1	1996-04-09	47876878	Hà Giang
6	6	K1576	Bùi Xuân Phúc	1	1990-11-19	87976324	Nam Định
7	7	K1577	Nguyễn Trung Trục	1	1982-02-28	9875734524234	Hà Nội

SELECT \* FROM SinhVien

	id_sv	ma_sv	ten_sv	gioi_tinh	ngay_sinh	sdt	dia_chi
1	1	K1571	Nguyễn Công Phượng	1	1997-08-16	19001265	Nha Trang
2	2	K1572	Ngô Bảo Châu	1	1976-10-18	123456789	Bắc Giang

SELECT TOP 2 \* FROM SinhVien

	id_sv	ma_sv	ten_sv	gioi_tinh	ngay_sinh	sdt	dia_chi
1	1	K1571	Nguyễn Công Phượng	1	1997-08-16	19001265	Nha Trang

SELECT TOP 2 PERCENT \* FROM SinhVien

# SELECT

## SELECT với INTO:

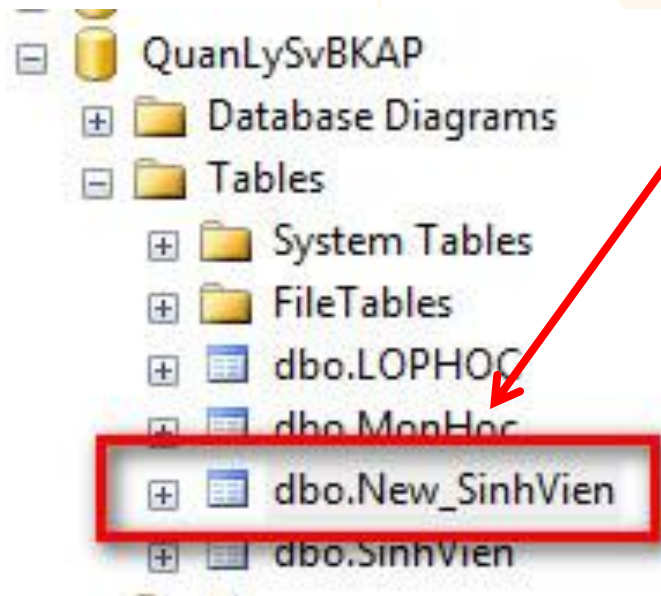
- Mệnh đề INTO được sử dụng cùng với SELECT để **tạo một bảng mới**.
- Cấu trúc cột của bảng mới là các cột được liệt kê trong câu lệnh SELECT, và toàn bộ các dòng lấy được bởi SELECT **được chèn (insert) vào bảng mới**.
- Để thực thi mệnh đề này với câu lệnh SELECT, người thực hiện **phải có quyền** sử dụng lệnh CREATE TABLE trên cơ sở dữ liệu đích.

# SELECT

```
SELECT <column_name1>..<column_nameN> [INTO new_table]  
FROM table_list
```

Ví dụ:

```
SELECT ma_sv, ten_sv, dia_chi INTO New_SinhVien FROM  
SinhVien
```



Results		Messages	
	ma_sv	ten_sv	dia_chi
1	K1571	Nguyễn Công Phượng	Nha Trang
2	K1572	Ngô Bảo Châu	Bắc Giang
3	K1573	Phạm Văn Mách	Bắc Ninh
4	K1574	Huỳnh Công Kiệt	Bắc Giang
5	K1575	Lê Huy Ngọ	Hà Giang
6	K1576	Bùi Xuân Phúc	Nam Định
7	K1577	Nguyễn Trung Trực	Hà Nội

```
SELECT * FROM New_SinhVien
```

# SELECT

## SELECT với WHERE:

- Mệnh đề WHERE được sử dụng với câu lệnh SELECT để **chọn có điều kiện** hoặc để **giới hạn** các bản ghi được lấy về bởi truy vấn.
- Mệnh đề WHERE chỉ ra một **biểu thức Boolean** để kiểm tra (test) các dòng được trả về bởi truy vấn.
- Dòng được trả về nếu biểu thức là true và **bị loại bỏ nếu là false.**



# SELECT

Ví dụ:

```
SELECT * FROM SinhVien WHERE ngay_sinh > '1990-01-01'
```

	id_sv	ma_sv	ten_sv	gioi_tinh	ngay_sinh	sdt	dia_chi
1	1	K1571	Nguyễn Công Phượng	1	1997-08-16	19001265	Nha Trang
2	5	K1575	Lê Huy Ngộ	1	1996-04-09	47876878	Hà Giang
3	6	K1576	Bùi Xuân Phúc	1	1990-11-19	87976324	Nam Định

# SELECT

Các toán tử sử dụng trong mệnh đề WHERE:

Toán tử	Giải thích
=	Bằng
<>	Khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
!	Không (phủ định)
BETWEEN	Giữa một khoảng
LIKE	Tìm với các ký tự đại diện
IN	Bên trong một khoảng



# SELECT

- Mệnh đề WHERE cũng được sử dụng với các toán tử logic như **AND, OR, và NOT**. Những toán tử này được dùng ghép nhiều điều kiện trong mệnh đề WHERE.
- Toán tử AND **ghép hai hay nhiều điều kiện** và trả về TRUE nếu tất cả các điều kiện cùng là TRUE.

Logical Operators	
Operator	Meaning
&&	and
	or
!	not

# SELECT

Ví dụ sau xem xét lọc sinh viên với ngày sinh **sau ngày 01/01/1990** và có địa chỉ **bắt đầu là 'N'**.

Code:

```
SELECT * FROM SinhVien WHERE ngay_sinh >
'1990-01-01' AND dia_chi LIKE 'N%'
```

Results		Messages					
	id_sv	ma_sv	ten_sv	gioi_tinh	ngay_sinh	sdt	dia_chi
1	1	K1571	Nguyễn Công Phượng	1	1997-08-16	19001265	Nha Trang
2	6	K1576	Bùi Xuân Phúc	1	1990-11-19	87976324	Nam Định

# SELECT

## GROUP BY:

- Mệnh đề GROUP chia (partition) tập kết quả thành **một hoặc nhiều nhóm** (subset) con. Mỗi nhóm đều có các giá trị và các biểu thức chung.
- Nếu sử dụng **hàm thống kê (Sum, Count,...)** trong mệnh đề GROUP BY, tập kết quả sẽ sinh ra giá trị tổng hợp duy nhất cho mỗi nhóm tập hợp (aggregate).
- Mỗi cột được nhóm sẽ làm giới hạn số dòng của tập kết quả . **Mỗi cột được nhóm sẽ chỉ có một dòng** cho mỗi nhóm dòng.

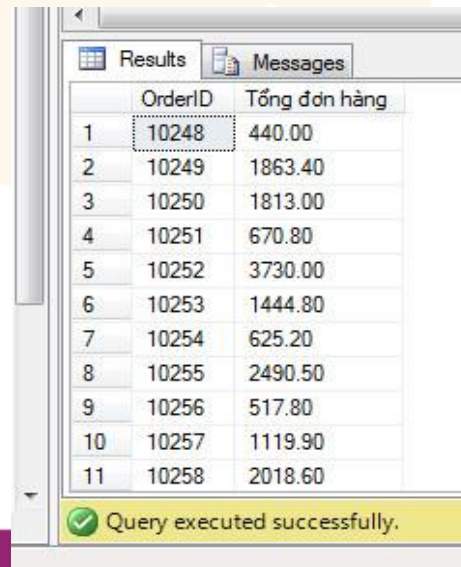
# SELECT

Cú pháp của mệnh đề GROUP BY:

```
SELECT <column_name1>..<column_nameN> FROM <table_name>  
GROUP BY <column_name>
```

Trong đó:

- column\_name1,...: là tên các cột mà theo đó tập kết quả được nhóm lại.



The screenshot shows a database query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'OrderID' and 'Tổng đơn hàng'. The table contains 11 rows of data. The first row is highlighted with a dashed border. Below the table, a status bar indicates 'Query executed successfully.' with a green checkmark icon.

	OrderID	Tổng đơn hàng
1	10248	440.00
2	10249	1863.40
3	10250	1813.00
4	10251	670.80
5	10252	3730.00
6	10253	1444.80
7	10254	625.20
8	10255	2490.50
9	10256	517.80
10	10257	1119.90
11	10258	2018.60

Tổng giá trị đơn hàng  
nhóm theo cột mã hóa  
đơn (OrderID)

# ORDER BY

- Dùng cho việc **sắp xếp thứ tự** các dòng trong tập kết quả theo giá trị của cột được chỉ ra.
- Nó **sắp xếp kết quả truy vấn** theo một hoặc nhiều cột. Sắp xếp có thể theo thứ tự tăng [ascending (**ASC**)] hoặc giảm [descending (**DESC**)].
- Mặc định, các bản ghi được **sắp xếp theo chiều tăng dần** ASC. Để sắp xếp theo chiều giảm sử dụng từ khóa tùy chọn DESC.
- Khi có nhiều cột được chỉ ra, SQL Server sẽ **ưu tiên thực hiện sắp xếp dữ liệu trong cột từ phía bên trái** trước, sau đó lần lượt đến cột kế tiếp.

# ORDER BY

## Cú pháp

```
SELECT <column_name>  
FROM <table_name>  
ORDER BY column_name> {ASC|DESC}
```

```
SELECT * FROM SinhVien ORDER BY ten_sv DESC
```

	id_sv	ma_sv	ten_sv	gioi_tinh	ngay_sinh	sdt	dia_chi
1	3	K1573	Phạm Văn Mách	1	1976-08-22	46575675	Bắc Ninh
2	7	K1577	Nguyễn Trung Trục	1	1982-02-28	9875734524234	Hà Nội
3	1	K1571	Nguyễn Công Phượng	1	1997-08-16	19001265	Nha Trang
4	2	K1572	Ngô Bảo Châu	1	1976-10-18	123456789	Bắc Giang
5	5	K1575	Lê Huy Ngọ	1	1996-04-09	47876878	Hà Giang
6	4	K1574	Huỳnh Công Kiệt	1	1986-12-09	76978978	Bắc Giang
7	6	K1576	Bùi Xuân Phúc	1	1990-11-19	87976324	Nam Định

# Tóm tắt bài học

- Câu lệnh SELECT **lấy hàng và cột từ bảng**.
- Câu lệnh SELECT cho phép người sử dụng để xác định **biểu thức** khác nhau để **xem các tập kết quả** một cách có trật tự.
- Một câu lệnh SELECT có thể chứa **biểu thức toán học** bằng cách áp dụng các nhà khai thác một hoặc nhiều cột.
- Từ khóa DISTINCT ngăn lấy các **bản ghi bị trùng lặp**.



## TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

---

# Thank for watching!

