



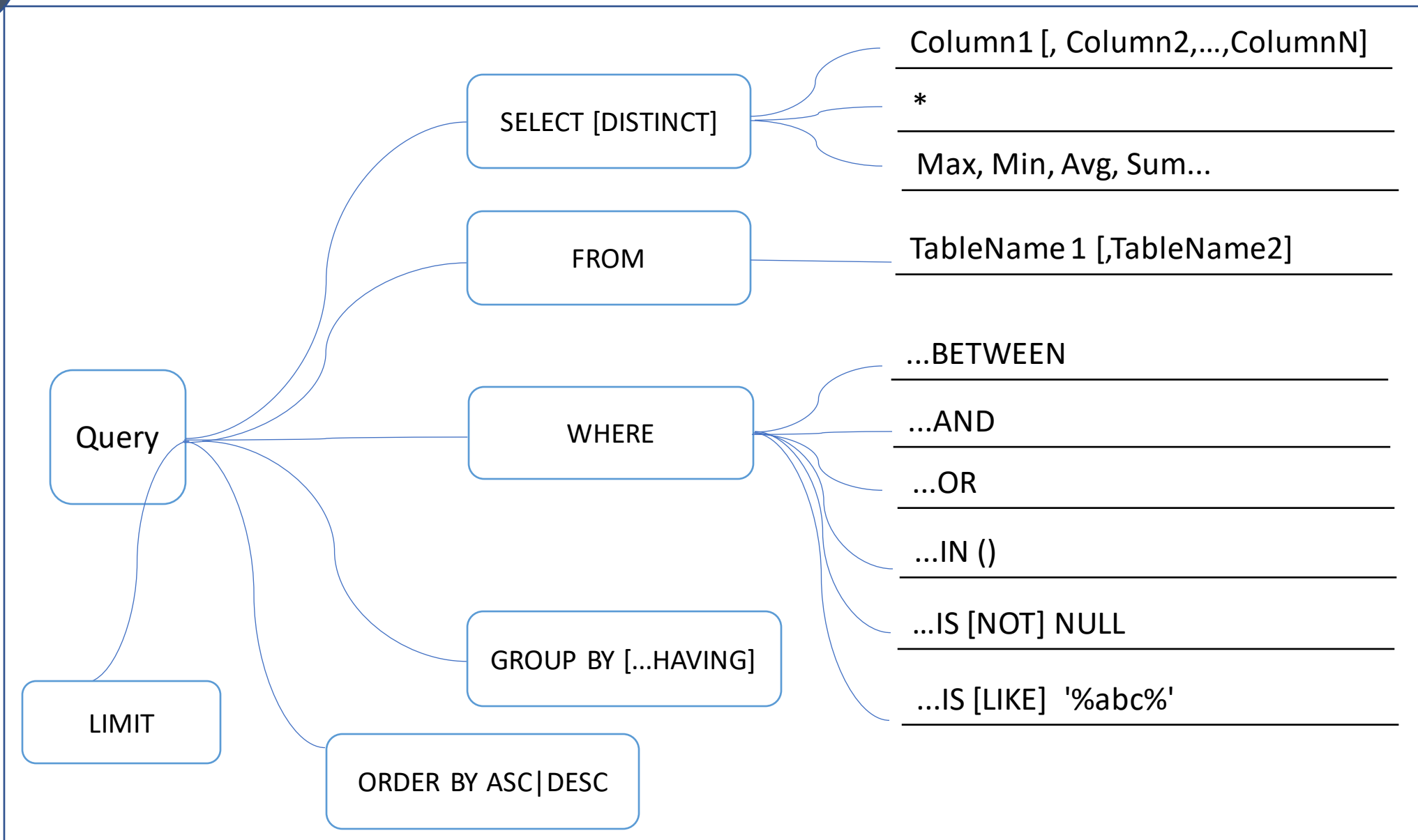
**PANTECH**

The foundation of tomorrow

**BUỔI 2** CÁC LỆNH TRUY VẤN DỮ LIỆU

## 1

# GIỚI THIỆU SƠ ĐỒ QUERY CƠ BẢN



## Truy vấn dữ liệu cơ bản với toán tử SELECT

Câu lệnh **SELECT** trong SQL được sử dụng để lấy dữ liệu từ một bảng cơ sở dữ liệu trả về dữ liệu này dưới dạng một bảng kết quả. Các bảng kết quả này được gọi là tập kết quả (result-sets).

Cú pháp cơ bản của câu lệnh SELECT trong SQL như sau:

```
Select <Tên Cột> <Bí Danh> <Biểu Thức>  
From <Tên Bảng>;
```

- Select : Xác định những cột cần hiển thị
- From: Xác định bảng dữ liệu nguồn.

**SELECT <Tên cột> FROM <Tên bảng>**

### 1. Hiển thị tất cả các cột trong bảng

Ví dụ: Hiển thị tất cả các cột trong bảng Profiles

```
| SELECT * FROM Profiles
```

### 2. Hiển thị các cột theo yêu cầu

Ví dụ: Hiển thị các cột Province, Region, Manager trong bảng Profiles

```
| SELECT Province, Region, Manager FROM Profiles
```



## Truy vấn dữ liệu cơ bản với toán tử SELECT

```
Select <Tên Cột> <Bí Danh> <Biểu Thức>  
From <Tên Bảng>;
```

- Select : Xác định những cột cần hiển thị
- From: Xác định bảng dữ liệu nguồn.

## SELECT <Bí danh> FROM <Tên bảng>

### 1. Sử dụng các cột bí danh (AS)

**Ví dụ:** Hiển thị bảng gồm cột Tỉnh (tương ứng cột Province), Khu\_vuc (tương ứng cột Region) từ bảng Profiles

| **SELECT Province AS Tỉnh, Region as Khu\_vuc from Profiles**

**Thực hành:** Hiển thị bảng gồm cột Ma\_dat\_hang (tương ứng cột order\_id), Ngay\_tra\_hang (tương ứng cột returned\_date) từ bảng Returns



## Truy vấn dữ liệu cơ bản với toán tử SELECT

```
Select <Tên Cột> <Bí Danh> <Biểu Thức>  
From <Tên Bảng>;
```

- Select : Xác định những cột cần hiển thị
- From: Xác định bảng dữ liệu nguồn.

**SELECT <Bí danh> FROM <Tên bảng>**

### 2. Sử dụng toán tử nối chuỗi

**Ví dụ:** Hiển thị bảng gồm cột Quan\_ly (tương ứng cột Manager), Dia\_chi (Nối cột Province và cột Region ngăn cách bởi dấu ",") từ bảng Profiles

```
| SELECT Manager as Quan_ly,  
Province + ',' + Region as Dia_chi FROM Profiles
```

**Thực hành:** Hiển thị bảng gồm cột Ma\_dat\_hang (tương ứng cột order\_id), Ngay\_tra\_hang (tương ứng cột returned\_date) từ bảng Returns



### 3

# SELECT DISTINCT STATEMENT

## LOẠI BỎ NHỮNG DÒNG TRÙNG LẶP

```
SELECT DISTINCT column1, column2, column3 FROM table_name
```

- Distinct: Loại bỏ những dòng trùng lặp

**Ví dụ:** Hiển thị danh sách không trùng lặp của cột Manager trong bảng Profiles

| **SELECT DISTINCT Manager FROM Profiles**

**Hãy so sánh với kết quả câu truy vấn sau:**

| **SELECT Manager FROM Profiles**

**Thực hành:** Hãy viết câu truy vấn trả về danh sách không trùng lặp của cột Province trong bảng Managers



# 4

## SELECT LIMIT STATEMENT

### GIỚI HẠN SỐ DÒNG KẾT QUẢ

```
SELECT TOP N [percent] [columns] FROM table_name
```

- TOP N: Trả về kết quả N dòng đầu tiên
- TOP N Percent: Trả về kết quả N% dòng đầu tiên

#### Ví dụ:

1. Hiển thị 10 dòng đầu tiên của bảng Orders

| **SELECT TOP 10 \* FROM Orders**

2. Hiển thị bảng gồm các cột order\_id, order\_date, order\_priority, order\_quantity, value từ bảng Orders. Giới hạn 20% dòng đầu tiên.

| **SELECT TOP 20 PERCENT order\_id, order\_date, order\_priority, order\_quantity, value  
FROM Orders**

**Thực hành:** Hãy viết câu truy vấn trả về 20 dòng đầu tiên của bảng gồm các cột order\_id, order\_date, order\_priority từ bảng Orders.



# 5

## TẠO CỘT SỬ DỤNG TOÀN TỬ SỐ HỌC

### SỬ DỤNG CÁC PHÉP TOÁN SỐ HỌC

Sử dụng các phép tính : + , - , \* , / , %

**Ví dụ:** Tạo bảng chứa các cột order\_id, order\_date, order\_quantity, value, profit , revenue từ bảng Orders. Trong đó:

Revenue được tính bằng công thức:  $\text{order\_quantity} * \text{unit\_price} * (1 - \text{discount})$

```
| SELECT order_id, order_date, order_quantity, value, profit,  
  order_quantity* unit_price*(1- discount) as revenue  
FROM Orders
```

**Thực hành:** Tạo thêm cột mới total\_cost vào ví dụ trên. Trong đó:

Total\_cost được tính bằng công thức:  $\text{product\_base\_margin} * \text{unit\_price} + \text{shipping\_cost}$





# LỘC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

## LỘC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name = value1;
```

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (value1, value2, ...);
```

### 1. Trả về kết quả theo 1 điều kiện xác định trong một cột

=> Sử dụng dấu "=" cho mệnh đề WHERE

Ví dụ: Viết câu truy vấn trả về các bản ghi từ bảng Orders với điều kiện cột order\_priority='Medium'

```
| SELECT * FROM Orders WHERE order_priority='Medium'
```

### 2. Trả về kết quả theo nhiều điều kiện xác định trong một cột

=> Sử dụng "in()" cho mệnh đề WHERE

Ví dụ: Viết câu truy vấn trả về các bản ghi từ bảng Orders với điều kiện cột order\_priority có các giá trị là 'Medium', 'Low', 'High'

```
| SELECT * FROM Orders WHERE order_priority in ('Medium', 'Low','High')
```



# LỘC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

## LỘC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

```
SELECT column_name(s)
```

```
FROM table_name
```

```
WHERE column_name (not) like parttern;
```

*Trong đó: 'a%' => Bắt đầu với ký tự a | '%a' => Kết thúc với ký tự a | '%a%' => Chứa ký tự a*

**3. Trả về kết quả theo điều kiện chưa xác định ( điều kiện là chứa kí tự nào đó) trong một cột**

**=> Sử dụng "like" trong mệnh đề WHERE**

**VD1:** Từ bảng Orders trả về 100 dòng kết quả đầu tiên với điều kiện cột shipping\_mode chứa ký tự 'Air'

```
| SELECT TOP 100 *
```

```
FROM Orders
```

```
WHERE shipping_mode like '%Air%'
```

**VD2:** Từ bảng Orders trả về các kết quả với điều kiện cột product\_subcategory bắt đầu với ký tự "Co"

```
| Select * from Orders where product_subcategory like 'Co%'
```



# LỌC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

## LỌC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name is (not) null;
```

### 3. Lọc các kết quả theo điều kiện null

**VD1:** Từ bảng Returns, trả về các kết quả điều kiện cột status null (trống)

```
| SELECT *  
  
FROM Returns  
  
WHERE status is null
```

**VD2:** Từ bảng Returns, trả về các kết quả điều kiện cột status không null (trống)

```
| SELECT *  
  
FROM Returns  
  
WHERE status is not null
```



## 6

# LỌC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

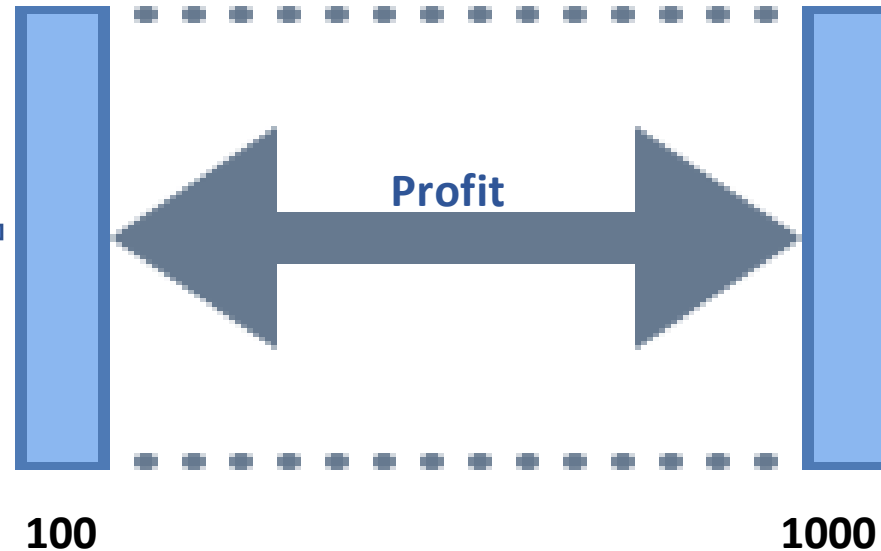
## LỌC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```

### 3. Lọc kết quả trong một khoảng nào đó

VD1: Từ bảng Orders, trả về các kết quả điều kiện cột profit nằm trong khoảng (100,1000)

```
| SELECT *  
FROM Orders  
WHERE profit between 100 and 1000
```



## 7

# HÀM MAX() AND MIN()

## SỬ DỤNG HÀM MIN() VÀ MAX()

```
SELECT MIN(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT MAX(column_name)  
FROM table_name  
WHERE condition;
```

**VD1:** Lấy giá trị nhỏ nhất của cột profit từ bảng Orders

```
| SELECT MIN(profit)  
FROM Orders
```

**VD2:** Lấy giá trị lớn nhất của cột Order\_quantity từ bảng Orders

```
| SELECT MAX(Order_quantity)  
FROM Orders
```



## SỬ DỤNG HÀM COUNT, AVG , SUM

```
SELECT COUNT ( [DISTINCT] column_name)
FROM table_name
WHERE condition;
```

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

**VD1:** Đếm số dòng cột order\_priority từ bảng Orders

```
| SELECT COUNT(Order_priority)
FROM Orders
```

**VD2:** Tính giá trị trung bình profit từ bảng Orders

```
| SELECT AVG(Profit)
FROM Orders
```

**VD3:** Tính tổng profit từ bảng Orders

```
| SELECT SUM(Profit)
FROM Orders
```



# TẠO BẢNG TỔNG HỢP VỚI GROUP BY

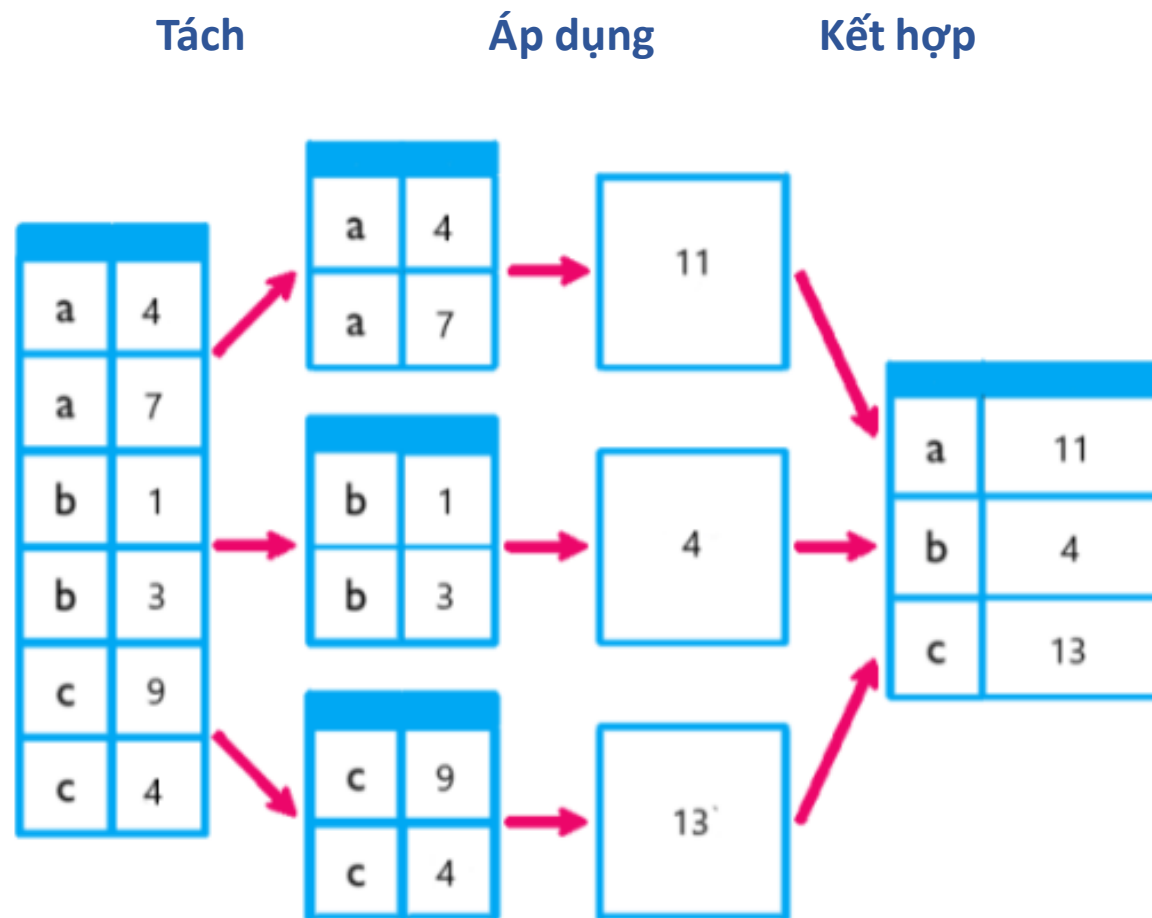
## MÔ TẢ MỤC ĐÍCH SỬ DỤNG GROUP BY

Câu lệnh SQL Group By sử dụng chiến lược tách-áp-dụng-kết hợp.

**1->Tách:** Các nhóm khác nhau được chia theo giá trị của chúng.

**2->Áp dụng:** Hàm tổng hợp được áp dụng cho các giá trị của các nhóm này.

**3->Kết hợp:** Các giá trị được kết hợp trong một hàng.



# TẠO BẢNG TỔNG HỢP VỚI GROUP BY

## NHÓM KẾT QUẢ THEO MỘT HOẶC NHIỀU CỘT SỬ DỤNG GROUP BY

```
SELECT column_1, function_name(column_2)--bao gồm cột tạo mới bằng hàm tổng hợp  
FROM table_name  
GROUP BY column_name--nhóm các hàng để áp dụng hàm tổng hợp  
ORDER BY column_name;--sắp xếp tăng/giảm nếu có
```

- Câu lệnh GROUP BY **nhóm các hàng** có **cùng giá trị** thành các hàng tóm tắt, như "tìm số lượng khách hàng ở mỗi quốc gia".
- Câu lệnh GROUP BY **thường được sử dụng với các hàm tổng hợp** (COUNT (), MAX (), MIN (), SUM (), AVG ()) để nhóm tập hợp kết quả theo một hoặc nhiều cột.

**VD1:** Trả về bảng tính tổng lợi nhuận (profit) theo các tỉnh (Province)

```
SELECT Province AS Tỉnh, SUM(Profit) as Tong_loi_nhuan  
FROM Orders  
GROUP BY Province
```

Hãy mang tất cả các cột không áp dụng hàm tổng hợp ở SELECT để đặt vào GROUP BY

**Bí quyết nè**





## SẮP XẾP KẾT QUẢ TĂNG DẦN/GIẢM DẦN SỬ DỤNG ORDER BY

```
SELECT column_name(s) --Các cột muốn hiển thị
FROM table_name
GROUP BY column_name(s) --Nhóm các hàng để áp dụng hàm tổng hợp (nếu có)
ORDER BY column_name(s) ASC/DESC --Sắp xếp theo thứ tự tăng/giảm tùy chọn
```

- ASC: Tăng dần
- DESC: Giảm dần

**VD1:** Trả về bảng tính tổng lợi nhuận (profit) theo các tỉnh (Province) và sắp xếp theo thứ tự giảm dần

```
| SELECT Province AS Tinh, SUM(Profit) as Tong_loi_nhuan
FROM Orders
GROUP BY Province
ORDER BY SUM(Profit) DESC
```



## MỆNH ĐỀ HAVING

```

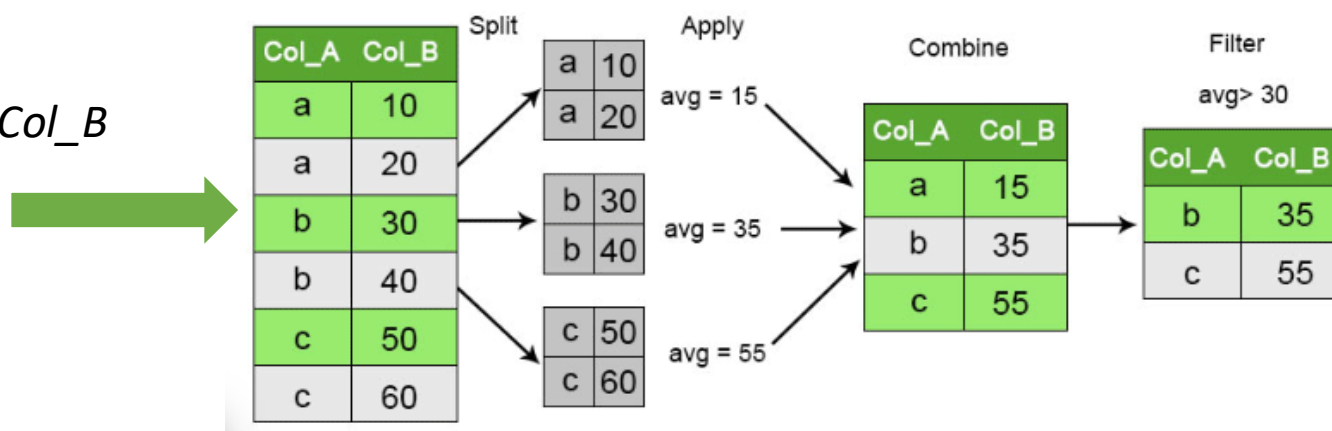
SELECT <column_1, function_name(column_2),...> -- Trong select chứa hàm tổng hợp
FROM <table>
WHERE <condition> -- Điều kiện áp dụng với cột có sẵn
GROUP BY <column(s)> -- Nhóm các hàng để áp dụng hàm tổng hợp
HAVING <condition> -- Hàm tổng hợp trong điều kiện
ORDER BY <column(s)>; -- Xác định thứ tự tăng giảm (Tùy chọn)

```

```

SELECT Col_A, avg(Col_B) as Col_B
FROM MyTable
GROUP BY Col_A
HAVING avg(Col_B) > 30

```



(\*) Mệnh đề HAVING đã được thêm vào SQL vì từ khóa WHERE không thể được sử dụng với các hàm tổng hợp

## MỆNH ĐỀ HAVING

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
```

- Mệnh đề HAVING đã được thêm vào SQL vì từ khóa WHERE không thể được sử dụng với các hàm tổng hợp.

**VD1:** Trả về bảng tính tổng lợi nhuận (profit) theo các tỉnh (Province). Điều kiện là lấy các tỉnh có tổng lợi nhuận lớn hơn 10000

```
| SELECT Province AS Tinh, SUM(Profit) as Tong_loi_nhuan
FROM Orders
GROUP BY Province
HAVING SUM(profit) > 10000
```



## SỰ KHÁC NHAU GIỮA 2 MỆNH ĐỀ WHERE VÀ HAVING

**WHERE***Sử dụng để lọc điều kiện trên **cột có sẵn*****HAVING***Sử dụng để lọc điều kiện của **hàm tổng hợp***

Bài tập thực hành:

1. Từ bảng Orders, lọc ra các kết quả với điều kiện cột Order\_priority là "Low"

Output:

	ID	order_id	order_date	order_priority	order_quantity	value	discou
1	1	3	2010-10-13 00:00:00.000	Low	6	224.2944	0.04
2	11	66	2009-01-19 00:00:00.000	Low	41	107.4528	0.09
3	14	70	2010-12-17 00:00:00.000	Low	48	86.6016	0.03
4	15	70	2010-12-17 00:00:00.000	Low	46	9001.763	0.05
5	18	129	2012-11-18 00:00:00.000	Low	4	23.5872	0.09



2. Từ bảng Orders, tạo bảng tính tổng số lượng sản phẩm (total\_quantity) theo từng product\_name. Lọc các kết quả có tổng số lượng sản phẩm lớn hơn 10

Output:

	product_name	total_quantity
1	Avery 510	382
2	Advantus Push Pins, Aluminum Head	344
3	Acco D-Ring Binder w/DublLock®	105
4	Staples Standard Envelopes	251
5	Cross DC1000F Demand Center	100

Ở dạng đầy đủ, cú pháp của lệnh SELECT trong SQL Server sẽ là:

**Cú pháp:**

```
SELECT [ ALL | DISTINCT ]  
[ TOP (n) [ PERCENT ] ]  
"biểu thức"  
FROM "bảng"  
[WHERE "điều kiện"]  
[GROUP BY "biểu thức"]  
[HAVING "điều kiện"]  
[ORDER BY "biểu thức" [ ASC | DESC ]];
```



### Truy vấn dữ liệu cơ bản với toán tử SELECT

Giải thích biến:

Biến	Ý nghĩa
ALL	Trả lại tất cả các hàng phù hợp.
DISTINCT	Loại bỏ tất cả các giá trị trùng lặp khỏi bộ kết quả.
TOP	Nếu chỉ định cụ thể sẽ trả về những giá trị đầu trong bộ kết quả dựa trên <i>gia_tri_dau</i> đã chọn. Ví dụ TOP(10) sẽ trả về 10 hàng đầu tiên trong bộ kết quả.
PERCENT	Nếu chỉ định cụ thể thì các hàng đầu tiên dựa theo phần trăm trên bộ kết quả. Ví dụ TOP(10) PERCENT sẽ trả về top 10% kết quả đầu tiên trong bộ kết quả.
Biểu thức	Bao gồm các cột có sẵn, các cột tạo mới bằng hàm tổng hợp... Dùng * nếu muốn lấy tất cả các cột.
Bảng	Bảng mà bạn muốn lấy kết quả về. Phải có ít nhất 1 bảng được liệt kê trong lệnh FROM.
WHERE	Điều kiện mà kết quả trả về phải đáp ứng được.
GROUP BY	Thu thập dữ liệu từ nhiều bản ghi và nhóm kết quả theo một hoặc nhiều cột.
HAVING	Dùng kết hợp với <b>GROUP BY</b> để giới hạn nhóm các hàng trả về khi điều kiện được đáp ứng là TRUE.
ORDER BY	Được dùng để lọc bộ kết quả. <b>ASC</b> sẽ lọc theo thứ tự tăng dần và <b>DESC</b> sẽ lọc theo thứ tự giảm dần.



# 13

## BÀI TẬP VỀ NHÀ

Bài 1: Viết câu lệnh Select các trường từ bảng Orders: order\_id, order\_date, order\_quantity, value, profit

- Và tạo thêm các cột mới:
  - Đặt tên cho cột mới là “revenue” trong đó công thức tính revenue là:  
 $\text{order\_quantity} * \text{unit\_price} * (1 - \text{discount})$
  - Đặt tên cho cột mới là “total\_cost” trong đó công thức tính total\_cost là:  
 $\text{product\_base\_margin} * \text{unit\_price} + \text{shipping\_cost}$
  - Đặt tên cho cột mới là “net\_profit” trong đó công thức tính net\_profit là:  
 $\text{revenue} - \text{total\_cost}$

## BÀI TẬP VỀ NHÀ

Bài 2: Từ bảng Orders. Viết các câu lệnh theo các điều kiện sau:

**2.1** region là “West”

**2.2** order\_priority không bao gồm “Critical”

**2.3** order\_priority là “High” hoặc order\_priority là “Low” hoặc order\_priority là “Medium” hoặc order\_priority là “Not Specified”

**2.4** province chứa từ “New”

**2.5** shipping\_mode không chứa từ “Air” và value nhỏ hơn 500

**2.6** product\_subcategory bắt đầu với từ “Co”

**2.7** customer\_segment kết thúc là “e” và order\_quantity lớn hơn 10



## 13

# BÀI TẬP VỀ NHÀ

Bài 3: Từ bảng Orders:

Viết lệnh trả về kết quả danh sách 10 khách hàng của tỉnh "Nunavut" đem lại tổng lợi nhuận cao nhất.  
Kết quả trả về như bảng bên.

	province	customer_name	total_profit
1	Nunavut	Carl Ludwig	7128.7904
2	Nunavut	Grant Carroll	6246.2908
3	Nunavut	Edward Hooks	4633.114
4	Nunavut	Barry French	3639.712
5	Nunavut	Monica Federle	2662.8742
6	Nunavut	Clay Rozendal	2633.8028
7	Nunavut	Annie Cyprus	2607.6585
8	Nunavut	Sylvia Foulston	1976.0961
9	Nunavut	Eugene Barchas	1951.7983
10	Nunavut	Barry Weirich	1894.492

