

# SELECT

 Cover



## ▼ CÔNG DỤNG

Được sử dụng để lấy dữ liệu từ một hoặc nhiều bảng trong cơ sở dữ liệu

## ▼ CÚ PHÁP

```
SELECT danh_sach_cot
FROM "bảng"
WHERE "điều kiện";
```

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

**ALL:** Trả lại tất cả các hàng phù hợp.

**DISTINCT:** Loại bỏ tất cả các giá trị trùng lặp khỏi bộ kết quả.

**TOP (gia\_tri\_dau):** Nếu chỉ định cụ thể sẽ trả về những giá trị đầu trong bộ kết quả dựa trên *gia\_tri\_dau* 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 (chỉ định bằng *gia\_tri\_dau*). Ví dụ TOP(10) PERCENT sẽ trả về top 10% kết quả đầu tiên trong bộ kết quả.

**WITH TIES:** Nếu chỉ định cụ thể thì các hàng cố định ở cuối trong kết quả có giới hạn sẽ được trả về. Điều này có thể khiến cho số hàng trả về nhiều hơn biến TOP cho phép.

**Biểu thức:** Cột hay giá trị tính toán mà bạn muốn lấy về. **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”**: Điều kiện mà kết quả trả về phải đáp ứng được.

**GROUP BY “biểu thức”**: Thu thập dữ liệu từ nhiều bản ghi và nhóm kết quả theo một hoặc nhiều cột

**HAVING “điều kiện”**: Dùng kết hợp với **GROUP BY** để giới hạn nhóm các hàng trả về khi điều kiện được đáp ứng là TRUE.

**ORDER BY “biểu thức”**: Được dùng để lọc bộ kết quả. **ASC** sẽ lọc theo thứ tự tăng dần và **DESC** sẽ lọc theo thứ tự giảm dần.

- **SELECT [danh\_sách\_cột]** :
  - Chỉ định các cột mà bạn muốn lấy dữ liệu.
  - Bạn có thể sử dụng tên cột, biểu thức hoặc hàm để lấy dữ liệu.
  - Nếu bạn muốn lấy tất cả các cột, hãy sử dụng `*`.
- **FROM [danh\_sách\_bảng]** :
  - Chỉ định bảng mà bạn muốn lấy dữ liệu.
  - Bạn có thể chỉ định một hoặc nhiều bảng.
- **[WHERE điều\_kiện]** :
  - (Tùy chọn) Lọc dữ liệu dựa trên điều kiện cụ thể.
  - Điều kiện được viết bằng biểu thức Boolean.
- **[ORDER BY danh\_sách\_cột\_sắp\_xếp]** :
  - (Tùy chọn) Sắp xếp kết quả truy vấn theo một hoặc nhiều cột.
  - Bạn có thể chỉ định thứ tự sắp xếp (ASC cho tăng dần, DESC cho giảm dần).
- **[GROUP BY danh\_sách\_cột\_nhóm]** :
  - (Tùy chọn) Nhóm kết quả truy vấn theo một hoặc nhiều cột.
  - Giá trị nhóm được tóm tắt bằng các hàm tổng hợp (SUM, COUNT, AVG, MIN, MAX).

```
+-----+-----+-----+-----+
|IDChuyenmuc  |Muccon          |Chuyenmuc1on  |Sobai  |
+-----+-----+-----+-----+
|      1      |SQL Server      |Lap trinh      |101     |
```

	2	Facebook	Mang xa hoi	152	
	3	Python	Lap trinh	111	
	4	JavaScript	Lap trinh	122	
	5	Chrome	Web	94	
	6	Instagram	Mang xa hoi	165	
+-----+-----+-----+-----+					

+-----+-----+-----+				
	ID	Ten muc	Trang thai	
+-----+-----+-----+				
	1	Lap trinh	Hien	
	2	Mang xa hoi	An	
	3	Web	An	
+-----+-----+-----+				

### ▼ Ví dụ về SELECT

Chọn tất cả các trường có **Sobai** lớn hơn 111 trong bảng **Quantrimang.com**. Bộ kết quả được sắp xếp theo thứ tự tăng dần của **Sobai**.

```
SELECT *
FROM Quantrimang.com
WHERE Sobai > 111
ORDER BY Sobai ASC;
```

+-----+-----+-----+-----+					
	IDChuyenmuc	Muccon	Chuyenmuclon	Sobai	
+-----+-----+-----+-----+					
	4	JavaScript	Lap trinh	122	
	2	Facebook	Mang xa hoi	152	
	6	Instagram	Mang xa hoi	165	
+-----+-----+-----+-----+					

Lấy ra **IDChuyenmuc**, **Chuyenmuclon**, **Sobai** với **IDChuyenmuc** lớn hơn 2 và **Chuyenmuclon** là 'Lap trinh'. Kết quả trả về được sắp xếp theo **Sobai** giảm dần.

```
SELECT IDChuyenmuc, Chuyenmuclon, Sobai
FROM Quangtrimang.com
WHERE IDChuyenmuc > 2 AND Chuyenlucomon = 'Laptrinh'
ORDER BY Sobai DESC;
```

IDChuyenmuc	Chuyenmuclon	Sobai
4	Lap trinh	122
3	Lap trinh	111

#### Giải thích:

- **Chọn các cột cần lấy từ các bảng:**
  - `[Quangtrimang].Muccon`
  - `[Quangtrimang].Chuyenmuclon`
  - `Muc.Trangthai`
- **Kết hợp hai bảng `Quangtrimang` và `Muc` bằng phép nối nội (INNER JOIN):**
  - Điều kiện nối là giá trị của cột `Chuyenmuclon` trong bảng `Quangtrimang` phải bằng giá trị của cột `Tenmuc` trong bảng `Muc`.
  - Và cột `Trangthai` trong bảng `Muc` phải có giá trị là `'Hien'`.
- **Sắp xếp kết quả theo giá trị của cột `Muccon` trong bảng `Quangtrimang` theo thứ tự tăng dần (ASC).**

Lấy những **Muccon** có **Trangthai** là *Hien*, và những bản ghi phải thỏa mãn **Chuyenmuclon** của bảng **Quangtrimang.com** và **Tenmuc** của bảng **Muc** phải trùng nhau và kết quả trả về sắp xếp theo thứ tự ASC của cột **Muccon**.

```
SELECT [Quangtrimang].Muccon,
       [Quangtrimang].Chuyenmuclon,
       [Muc].Trangthai
FROM [Quangtrimang]
INNER JOIN [Muc]
```

```
ON [Quantrimang].Chuyenmuclon = [Muc].Tenmuc AND [Muc].Trangt
ORDER BY [Quantrimang].Muccon ASC;
```

```
+-----+-----+-----+
|Muccon   |Chuyenmuclon|Trangthai |
+-----+-----+-----+
|JavaScript|Lap trinh   |Hien       |
|Python    |Lap trinh   |Hien       |
|SQL Server|Lap trinh   |Hien       |
+-----+-----+-----+
```

Lấy những **Muccon** có **Trangthai** là An, và những bản ghi phải thỏa mãn **Chuyenmuclon** của bảng **Quantrimang.com** và **Tenmuc** của bảng **Muc** phải trùng nhau và kết quả trả về sắp xếp theo thứ tự ASC của cột **Muccon**.

```
SELECT [Quantrimang].Muccon,
       [Quantrimang].Chuyenmuclon,
       [Muc].Trangthai
FROM [Quantrimang]
INNER JOIN [Muc]
ON [Quantrimang].Chuyenmuclon = [Muc].Tenmuc AND [Muc].Trangt
ORDER BY [Quantrimang].Muccon ASC;
```

```
+-----+-----+-----+
|Muccon   |Chuyenmuclon|Trangthai |
+-----+-----+-----+
|Chrom     |Web          |An         |
|Facebook  |Mang xa hoi  |An         |
|Instagram |Mang xa hoi  |An         |
+-----+-----+-----+
```

### ▼ Ví dụ về SELECT TOP

Lấy 2 giá trị đầu tiên từ bảng **Quantrimang** với **Chuyemuclon** là 'Lap trinh'. Kết quả được sắp xếp tăng dần theo **IDChuyenmuc**. Dù còn các giá trị khác đáp ứng yêu cầu thì chúng cũng sẽ không được trả về

```
SELECT TOP(2) *
FROM Quantrimang.com
WHERE Chuyenmuclon = 'Lap trinh'
ORDER BY IDChuyenmuc ASC;
```

IDChuyenmuc	Muccon	Chuyenmuclon	Sobai
1	SQL Server	Lap trinh	101
3	Python	Lap trinh	111

### Giải thích:

- **SELECT TOP(2) \*** : Lựa chọn hai hàng đầu tiên từ tập kết quả trả về và chọn tất cả các cột.
- **FROM [Quantrimang]** : Chỉ định bảng **Quantrimang** làm nguồn dữ liệu.
- **WHERE Chuyenmuclon = 'Lap trinh'** : Điều kiện lọc các hàng sao cho chỉ những hàng có giá trị của cột **Chuyenmuclon** là 'Lap trinh' được chọn.
- **ORDER BY IDChuyenmuc ASC** : Sắp xếp các kết quả theo thứ tự tăng dần của **IDChuyenmuc**.

### ▼ Ví dụ về SELECT TOP PERCENT

Lấy gồm 10% giá trị đầu tiên trong toàn bộ kết quả **Chuyenmuclon** là 'Lap trinh' trong bảng **Quantrimang.com**. 90% kết quả còn lại sẽ không được trả về.

```
SELECT TOP(10) PERCENT *
FROM Quantrimang.com
WHERE Chuyenmuclon = 'Laptrinh'
ORDER BY IDchuyenmuc ASC;
```

IDChuyenmuc	Muccon	Chuyenmuclon	Sobai
-------------	--------	--------------	-------

1	SQL Server	Lap trinh	101
+	+	+	+

### Giải thích:

- `SELECT TOP(10) PERCENT *` : Lựa chọn 10% hàng đầu tiên từ tập kết quả trả về và chọn tất cả các cột.
- `FROM [Quantrimang]` : Chỉ định bảng `Quantrimang` làm nguồn dữ liệu.
- `WHERE Chuyenmuclo = 'Lap trinh'` : Điều kiện lọc các hàng sao cho chỉ những hàng có giá trị của cột `Chuyenmuclo` là 'Lap trinh' được chọn.
- `ORDER BY IDChuyenmuc ASC` : Sắp xếp các kết quả theo thứ tự tăng dần của `IDChuyenmuc`.

### ▼ Chọn tất cả các cột từ một bảng

Ví dụ này sẽ lấy tất cả các cột từ bảng `employees`

```
SELECT *
FROM employees;
```

### ▼ Chọn các cột cụ thể từ một bảng

lấy dữ liệu của các cột `first_name`, `last_name`, và `salary` từ bảng `employees`.

```
SELECT first_name, last_name, salary
FROM employees
WHERE department_id = 10;
```

### ▼ Sử dụng điều kiện lọc dữ liệu

lấy dữ liệu của các cột `first_name`, `last_name`, và `salary` từ bảng `employees` nhưng chỉ với những hàng có `department_id` bằng 10

```
SELECT first_name, last_name, salary
FROM employees
WHERE department_id = 10;
```

### ▼ Sử dụng các hàm tổng hợp (aggregate functions)

Nhóm các hàng theo `department_id` và tính toán số lượng nhân viên và mức lương trung bình cho mỗi phòng ban

```
SELECT department_id, COUNT(*) AS num_employees, AVG(salary)
FROM employees
GROUP BY department_id;
```

Giả sử chúng ta có một bảng `employees` với dữ liệu như sau:

employee_id	first_name	last_name	department_id	salary
1	John	Doe	10	60000
2	Jane	Smith	20	75000
3	Larry	Page	10	65000
4	Sergey	Brin	30	80000
5	Alan	Turing	20	70000
6	Ada	Lovelace	30	90000

## Câu lệnh SQL

Câu lệnh SQL để nhóm các hàng theo `department_id` và tính toán số lượng nhân viên và mức lương trung bình cho mỗi phòng ban là:

```
SELECT department_id, COUNT(*) AS num_employees, AVG(salary)
FROM employees
GROUP BY department_id;
```

Kết quả của câu lệnh trên sẽ tạo ra bảng tổng hợp như sau:

department_id	num_employees	avg_salary
10	2	62500
20	2	72500
30	2	85000

## Giải thích

- **department\_id:** Cột `department_id` là khóa dùng để nhóm các hàng.
- **num\_employees:** Sử dụng hàm tổng hợp `COUNT(*)` để đếm số lượng nhân viên trong mỗi nhóm `department_id`.



- **avg\_salary**: Sử dụng hàm tổng hợp `AVG(salary)` để tính lương trung bình của nhân viên trong mỗi nhóm `department_id`.

## Các bước thực hiện

1. **GROUP BY**: Nhóm các hàng lại với nhau dựa trên giá trị của cột `department_id`.
2. **COUNT(\*)**: Đếm số lượng hàng (nhân viên) trong mỗi nhóm.
3. **AVG(salary)**: Tính lương trung bình cho mỗi nhóm nhân viên dựa trên cột `salary`.

## Kết luận

Câu lệnh `SELECT department_id, COUNT(*) AS num_employees, AVG(salary) AS avg_salary FROM employees GROUP BY department_id;` giúp bạn tổng hợp và phân tích dữ liệu nhân viên theo từng phòng ban, cho phép bạn dễ dàng biết được số lượng nhân viên và mức lương trung bình của từng phòng ban một cách hiệu quả.

### ▼ Sử dụng các mệnh đề khác như `JOIN`

Kết hợp dữ liệu từ bảng `employees` và `departments` dựa trên cột `department_id`.

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id;
```

## Bảng `employees`

employee_id	first_name	last_name	department_id	salary
1	John	Doe	10	60000
2	Jane	Smith	20	75000
3	Larry	Page	10	65000
4	Sergey	Brin	30	80000
5	Alan	Turing	20	70000
6	Ada	Lovelace	30	90000

## Bảng `departments`

department_id	department_name
10	Engineering

20	Marketing
30	Research

## Câu lệnh SQL

Câu lệnh SQL để kết hợp dữ liệu từ hai bảng `employees` và `departments` dựa trên cột `department_id`:

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id;
```

Kết quả của câu lệnh trên sẽ tạo ra bảng kết hợp như sau:

first_name	last_name	department_name
John	Doe	Engineering
Larry	Page	Engineering
Jane	Smith	Marketing
Alan	Turing	Marketing
Sergey	Brin	Research
Ada	Lovelace	Research

## Giải thích

- **e.first\_name** và **e.last\_name**: Lấy cột `first_name` và `last_name` từ bảng `employees` với bí danh `e`.
- **d.department\_name**: Lấy cột `department_name` từ bảng `departments` với bí danh `d`.
- **JOIN departments d ON e.department\_id = d.department\_id**: Thực hiện phép nối (join) giữa bảng `employees` và bảng `departments` dựa trên giá trị chung của cột `department_id`.

## Các bước thực hiện

1. **JOIN**: Kết hợp các hàng từ bảng `employees` và bảng `departments` nơi giá trị `department_id` bằng nhau.
2. **SELECT**: Chọn các cột cần thiết từ các bảng đã kết hợp.

## Kết luận

Câu lệnh `SELECT e.first_name, e.last_name, d.department_name FROM employees e JOIN departments d ON e.department_id = d.department_id;` giúp bạn kết hợp và trích xuất thông tin liên quan từ hai bảng dựa trên cột chung `department_id`. Đây là cách hiệu quả để kết hợp dữ liệu từ nhiều bảng và tạo ra các báo cáo hoặc phân tích chi tiết về nhân viên và phòng ban

### ▼ Sử dụng alias (bí danh) để đặt tên tạm thời cho cột hoặc bảng

```
SELECT first_name AS fname, last_name AS lname, salary AS sal
FROM employees AS e
WHERE e.department_id = 10;
```

i	MaSP	TenSP	LoaiSP	DonGia
1		Điện thoại A	DienThoai	1000.00
2		Điện thoại B	DienThoai	1200.00
3		Máy tính xách tay	Laptop	1500.00
4		Tai nghe không dây	PhuKien	200.00

i	MaDH	TenNV	MaSP	SoLuong
1		Nguyen A	1	5
2		Nguyen B	1	1
3		Nguyen A	2	3
4		Nguyen C	3	1
5		Nguyen B	4	2
6		Nguyen A	2	1

1. Lấy các cột `MaSP`, `TenSP` và `DonGia` từ bảng `SanPham` cho các sản phẩm thuộc loại `DienThoai`:

```
SELECT MaSP, TenSP, DonGia
FROM SanPham
WHERE LoaiSP = 'DienThoai';
```

MaSP	TenSP	DonGia
1	Điện thoại A	1000.00
2	Điện thoại B	1200.00

2. Lấy tên nhân viên ( `TenNV` ) và đếm số lượng đơn hàng ( `SoLuongDonHang` ) mà mỗi nhân viên đã xử lý từ bảng `DonHang` , nhóm theo tên nhân viên và sắp xếp theo số lượng đơn hàng giảm dần:

```
SELECT TenNV, COUNT(*) AS SoLuongDonHangDaXuLy
FROM DonHang
GROUP BY TenNV
ORDER BY SoLuongDonHang DESC;
```

TenNV	SoLuongDonHangDaXuLy
Nguyen A	3
Nguyen B	2
Nguyen C	1

⇒ Điều này có nghĩa là `Nguyen A` đã xử lý 3 đơn hàng, `Nguyen B` đã xử lý 2 đơn hàng, và `Nguyen C` đã xử lý 1 đơn hàng.

#### Giải thích:

- **GROUP BY:** Được sử dụng để nhóm các dòng có cùng giá trị trong một hoặc nhiều cột lại với nhau. Trong trường hợp này, **GROUP BY TenNV** có ý nghĩa là nhóm các dòng trong bảng `DonHang` dựa trên giá trị của cột `TenNV` (Tên nhân viên).
- **COUNT(\*):** Thực hiện đếm số lượng dòng của mỗi nhóm được tạo bởi `GROUP BY TenNV` . Trong trường hợp này, **COUNT(\*)** được sử dụng để đếm số lượng đơn hàng mà mỗi nhân viên đã xử lý.
- `AS SoLuongDonHang` : Đoạn này đặt tên cho cột kết quả mà hàm **COUNT(\*)** sẽ trả về. Bằng cách đặt tên cho cột, bạn có thể sử dụng tên này để tham chiếu đến cột trong kết quả của câu truy vấn.

3. Đếm tổng số lượng các đơn hàng mà mỗi nhân viên đã xử lý

```
SELECT TenNV, SUM(SoLuong) AS TongSoLuongDonHang
FROM DonHang
GROUP BY TenNV
ORDER BY TongSoLuongDonHang DESC;
```

TenNV	TongSoLuongDonHang
Nguyen A	9
Nguyen B	3
Nguyen C	1

4. Nếu bạn chỉ muốn đếm số lượng dòng của một cột cụ thể, bạn có thể thay `*` bằng tên của cột mà bạn muốn đếm. Ví dụ, nếu bạn muốn đếm số lượng dòng của cột `SoLuong` trong bảng `DonHang`, bạn có thể sử dụng `COUNT(SoLuong)`.

```
SELECT TenNV, COUNT(SoLuong) AS SoLuongDonHang
FROM DonHang
GROUP BY TenNV
ORDER BY SoLuongDonHang DESC;
```

TenNV	SoLuongDonHang
Nguyen A	3
Nguyen B	2
Nguyen C	1

⇒ Kết quả này không khác biệt so với sử dụng `COUNT(*)` trong trường hợp này, vì cột `SoLuong` không chứa giá trị NULL trong dữ liệu mẫu. Tuy nhiên, sử dụng `COUNT(SoLuong)` sẽ giúp nếu bạn chỉ quan tâm đến số lượng các giá trị không NULL trong cột `SoLuong`.

5. Nếu bạn muốn đếm tổng số lượng dòng trong cột `SoLuong` mà không quan tâm đến giá trị của nó, thì bạn nên sử dụng `COUNT(*)` thay vì `COUNT(SoLuong)`.

```
SELECT COUNT(*) AS SoLuongDong
FROM DonHang;
```

```
| SoLuongDong |
| - - - - - |
|          6  |
```

6. Bằng cách đặt tên cho cột, bạn có thể sử dụng tên này để tham chiếu đến cột trong kết quả của câu truy vấn.

Ví dụ, nếu bạn muốn lọc ra các hàng mà số lượng đơn hàng lớn hơn 1, bạn có thể sử dụng tên `SoLuongDonHang` như sau:

```

SELECT TenNV, COUNT(*) AS SoLuongDonHang
FROM DonHang
GROUP BY TenNV
ORDER BY SoLuongDonHang DESC;
-----
SELECT TenNV, SoLuongDonHang
FROM (
    SELECT TenNV, COUNT(*) AS SoLuongDonHang
    FROM DonHang
    GROUP BY TenNV
) AS TongSoLuongDonHang
WHERE SoLuongDonHang > 1;

```

TenNV	SoLuongDonHang
Nguyen A	3
Nguyen B	2
Nguyen C	1

TenNV	SoLuongDonHang
Nguyen A	3
Nguyen B	2