
Chương 3

Truy vấn dữ liệu

Nội dung trình bày

1. Thao tác cập nhật dữ liệu (DML).
2. Các toán tử trong SQL
3. Truy vấn dữ liệu
4. Truy vấn con, truy vấn lồng
5. Toán tử Union, Intersect, Except
6. Common Table Expressions
7. Ranking Function

1. Thao tác cập nhật dữ liệu (DML)

1. Thao tác cập nhật dữ liệu (DML)

DML – Data Manipulation Language: Thao tác cập nhật dữ liệu gồm lệnh:

- Insert
- Update
- Delete
- Select

1. Thao tác cập nhật dữ liệu (DML)

1.1. Insert: Là lệnh dùng để thêm 1 dòng dữ liệu vào bảng

Cú pháp: có thể dùng 2 cú pháp để thêm dòng dữ liệu

Cách 1:

```
INSERT INTO <tên bảng> VALUES (giá trị 1, giá trị 2, ..., giá trị n);
```

Cách 2:

```
INSERT INTO <tên bảng> (cột 1, cột 2, ..., cột n) VALUES (gt1, gt 2, ..., gt n);
```

Lưu ý: Với cách 1 thì các giá trị thêm vào phải đúng với thứ tự các cột trong bảng và tuân theo quy tắc các ràng buộc đặt ra

1. Thao tác cập nhật dữ liệu (DML)

GiaoVien			
	Column Name	Data Type	Allow Nulls
🔑	MaGV	int	<input type="checkbox"/>
	HoTen	nvarchar(10)	<input type="checkbox"/>
	NamSinh	date	<input type="checkbox"/>
	DiaChi	nvarchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Insert into GiaoVien **Values**(1, 'Nguyễn Văn A', 1-1-1987, 'Bien Hoa, Dong Nai')

Insert into GiaoVien (MaGV, HoTen, NamSinh, DiaChi)

Values (1, 'Nguyễn Văn A', 1-1-1987, 'Bien Hoa, Dong Nai')

1. Thao tác cập nhật dữ liệu (DML)

1.2. Update: Sử dụng sửa nội dung của 1 hay nhiều dòng dữ liệu

Cú pháp:

```
UPDATE <tên bảng>  
SET cột1 = giá trị 1, cột2 = giá trị 2, ..., cột n = giá trị n  
[WHERE <điều kiện> ]
```

Lưu ý: Trường hợp câu lệnh không có where thì tất cả giá trị của cột sẽ được sửa đổi

1. Thao tác cập nhật dữ liệu (DML)

GiaoVien			
	Column Name	Data Type	Allow Nulls
🔑	MaGV	int	<input type="checkbox"/>
	HoTen	nvarchar(10)	<input type="checkbox"/>
	NamSinh	date	<input type="checkbox"/>
	DiaChi	nvarchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Update GiaoVien

Set HoTen = '**Tran Van B**', NamSinh = '**1-1-1987**'

Where MaGV = 1

Lưu ý: Khi cập nhật các giá trị của bảng, thì các giá trị này vẫn phải tuân theo quy tắc ràng buộc thiết lập lúc tạo bảng

1. Thao tác cập nhật dữ liệu (DML)

1.3. Delete: Sử dụng xóa nội dung của 1 hay nhiều dòng dữ liệu

Cú pháp:

```
Delete from <tên bảng> [WHERE <điều kiện> ]
```

Lưu ý: Trường hợp câu lệnh không có where thì tất cả giá trị của cột sẽ được xóa

1. Thao tác cập nhật dữ liệu (DML)

GiaoVien			
	Column Name	Data Type	Allow Nulls
🔑	MaGV	int	<input type="checkbox"/>
	HoTen	nvarchar(10)	<input type="checkbox"/>
	NamSinh	date	<input type="checkbox"/>
	DiaChi	nvarchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Delete from GiaoVien Where MaGV = 1

Lưu ý: Các dữ liệu khi delete sẽ mất đi, do đó phải thật cẩn thận với lệnh này.

2. Các toán tử trong SQL

2. Các toán tử trong SQL

Toán tử trong SQL là gì?

Toán tử là **từ dành riêng hoặc ký tự** được sử dụng chủ yếu trong mệnh đề WHERE của lệnh SQL để **thực hiện các thao tác, chẳng hạn như so sánh, các phép tính số học,...**

Toán tử trong SQL bao gồm:

- Toán tử số học
- Toán tử so sánh
- Toán tử logic

2. Các toán tử trong SQL

2.1. Toán tử số học:

Toán tử	Mô tả	Ví dụ: $a = 10, b = 20$
+ (Cộng)	Cộng các giá trị ở hai bên của toán tử (phép cộng).	$a + b$ cho kết quả là 30
- (Trừ)	Lấy toán hạng bên trái trừ toán hạng bên phải (phép trừ).	$a - b$ cho kết quả là -10
* (Nhân)	Nhân giá trị của hai toán hạng ở hai bên toán tử (phép nhân).	$a * b$ cho kết quả là 200
/ (Chia)	Chia toán hạng bên trái cho toán hạng bên phải (phép chia).	b / a cho kết quả là 2
% (Chia lấy số dư)	Chia toán hạng bên trái cho toán hạng bên phải rồi lấy số dư (phép chia lấy phần dư).	$b \% a$ cho kết quả là 0

2. Các toán tử trong SQL

```
select 10 + 20 ketqua
```

% ▾

Results Messages

ketqua
30

```
select 10 - 20 ketqua
```

% ▾

Results Messages

ketqua
-10

```
select 10 * 20 ketqua
```

% ▾

Results Messages

ketqua
200

```
select 30 / 20 ketqua
```

% ▾

Results Messages

ketqua
1

```
select 30 / 20.0 ketqua
```

% ▾

Results Messages

ketqua
1.500000

```
select 30 % 20 ketqua
```

% ▾

Results Messages

ketqua
10

2. Các toán tử trong SQL

2.2. Toán tử so sánh:

Toán tử	Mô tả	Ví dụ
=	Kiểm tra xem giá trị của 2 toán hạng ở hai bên dấu bằng có bằng nhau không, nếu có bằng thì điều kiện là true.	(a = b) không true.
!=	Kiểm tra xem giá trị của 2 toán hạng có bằng nhau không, nếu khác thì điều kiện là true.	(a != b) là true.
<>	Kiểm tra xem giá trị của 2 toán hạng có bằng nhau không, nếu không bằng thì điều kiện là true.	(a <> b) là true.
>	Kiểm tra xem giá trị của toán hạng bên trái có lớn hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true.	(a > b) không true.
<	Kiểm tra xem giá trị của toán hạng bên trái có nhỏ hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true.	(a < b) là true.
>=	Kiểm tra xem giá trị của toán hạng bên trái có lớn hơn hoặc bằng giá trị của toán hạng bên phải không, nếu có thì điều kiện là true.	(a >= b) không true.
<=	Kiểm tra xem giá trị của toán hạng bên trái có nhỏ hơn hoặc bằng giá trị của toán hạng bên phải không, nếu có thì điều kiện là true.	(a <= b) là true.
!<	Kiểm tra xem giá trị của toán hạng bên trái là không nhỏ hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true.	(a !< b) là false.
!>	Kiểm tra xem giá trị của toán hạng bên trái là không lớn hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true.	(a !> b) là true.

2. Các toán tử trong SQL

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

`select * from sanpham where SoTon = 7`

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7

`select * from sanpham where SoTon != 7`

MaSP	TenSP	DVT	DonGia	SoTon
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

`select * from sanpham where SoTon >= 7`

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
8	Cá hộp	Thùng	20000	10

2. Các toán tử trong SQL

2.3. Toán tử logic:

Toán tử	Mô tả
AND	Toán tử AND cho phép nhiều điều kiện cùng tồn tại trong mệnh đề WHERE của lệnh SQL.
BETWEEN	Toán tử BETWEEN được sử dụng để tìm kiếm giá trị trong một tập hợp giá trị, với giá trị nhỏ nhất và giá trị lớn nhất được cho trước.
IN	Toán tử IN được sử dụng để so sánh giá trị với danh sách chuỗi giá trị đã được xác định trước.
LIKE	Toán tử LIKE được sử dụng để so sánh giá trị với các giá trị tương tự, sử dụng các toán tử Wildcard.
NOT	Toán tử NOT đảo ngược ý nghĩa của toán tử logic mà nó được sử dụng. Ví dụ: NOT EXISTS, NOT BETWEEN, NOT IN,... Đây là một toán tử phủ định.
OR	Toán tử OR thường được sử dụng để nối nhiều điều kiện trong mệnh đề WHERE của lệnh SQL.
IS NULL	Toán tử NULL được sử dụng để so sánh một giá trị với giá trị NULL.
EXISTS	Toán tử EXISTS được sử dụng để tìm kiếm sự có mặt của một hàng trong bảng thỏa mãn điều kiện cụ thể.
ALL	Toán tử ALL được sử dụng để so sánh một giá trị với tất cả các giá trị trong tập hợp giá trị khác.
ANY	Toán tử ANY được sử dụng để so sánh một giá trị với bất kỳ giá trị thích hợp nào trong danh sách tùy theo điều kiện.

3. Truy vấn dữ liệu

3. Truy vấn dữ liệu

Trong SQL, Câu lệnh SELECT sẽ được sử dụng để truy xuất dữ liệu từ các dòng và các cột của một hay nhiều bảng, View

3.1. Truy vấn cơ bản

```
SELECT [tính chất] <cột 1, cột 2, ... Cột n>  
FROM <table> [as Alias]
```

[tính chất] có thể là một trong các từ khóa:

- *** : Lấy tất cả dữ liệu
- DISTINCT** : Lấy dữ liệu không trùng lặp
- TOP**<n> : Lấy dữ liệu thứ n đầu tiên

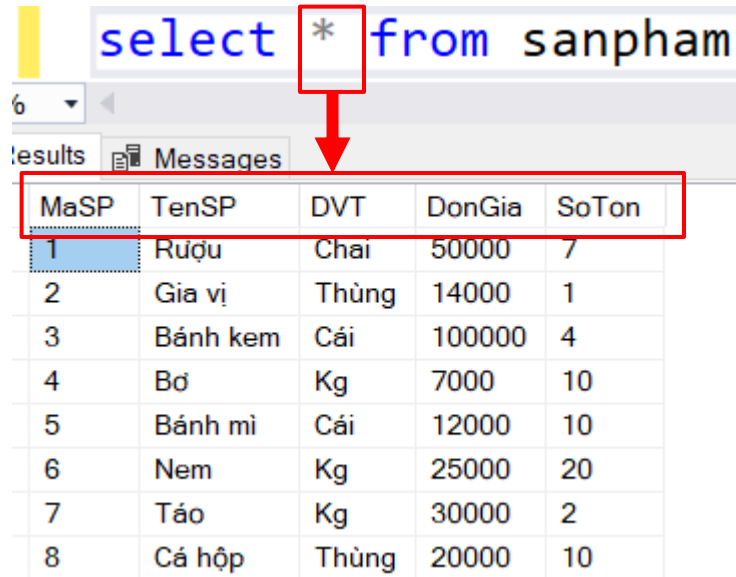
Có thể gán tên cho các cột với cú pháp: <cột> **AS** 'Tên thay thế'.

Alias là bí danh (hay còn gọi là tên gọi tắt) của table dùng cho các table có tên quá dài, hoặc một table được dùng nhiều lần trong câu truy vấn.

Có thể đặt Alias theo cú pháp: <tên table> **AS** <tên Alias>

3. Truy vấn dữ liệu

Hiển thị tất cả các dòng dữ liệu và tất cả cột dữ liệu trong bảng SanPham



```
select * from sanpham
```

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

3. Truy vấn dữ liệu

Hiển thị tất cả các dòng dữ liệu SanPham, nhưng chỉ hiển thị cột mã sản phẩm và tên sản phẩm

```
select MaSP, TenSP from dbo.SanPham
```

MaSP	TenSP
1	Rượu
2	Gia vị
3	Bánh kem
4	Bơ
5	Bánh mì
6	Nem
7	Táo
8	Cá hộp

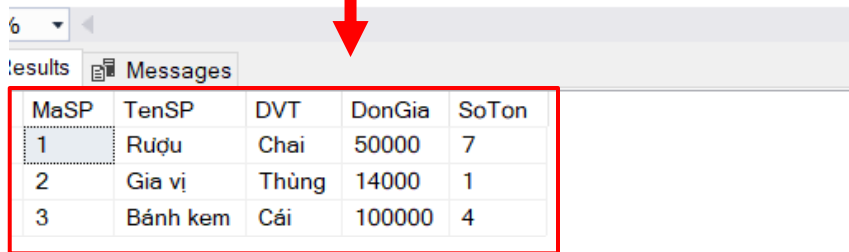
```
select MaSP as MaSanPham, TenSP as TenSanPham  
from dbo.SanPham as BangSanPham
```

MaSanPham	TenSanPham
1	Rượu
2	Gia vị
3	Bánh kem
4	Bơ
5	Bánh mì
6	Nem
7	Táo
8	Cá hộp

3. Truy vấn dữ liệu

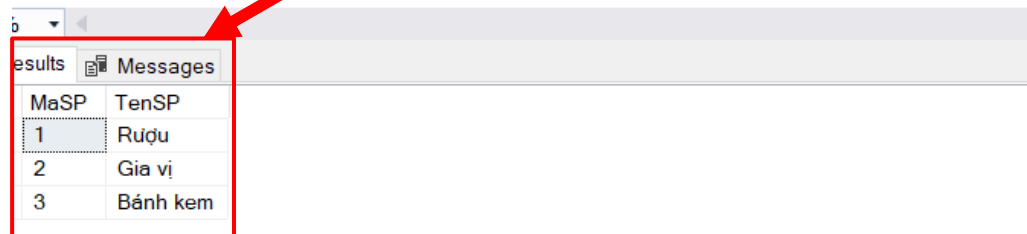
Hiển thị 3 dòng dữ liệu đầu tiên trong bảng sản phẩm

```
select top 3 * from dbo.SanPham
```



MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4

```
select top 3 MaSP, TenSP from dbo.SanPham
```



MaSP	TenSP
1	Rượu
2	Gia vị
3	Bánh kem

3. Truy vấn dữ liệu

Hiển thị các dòng dữ liệu không trùng lặp

```
select distinct * from dbo.SanPham
```

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

```
select distinct MaSP, TenSP from dbo.SanPham
```

MaSP	TenSP
1	Rượu
2	Gia vị
3	Bánh kem
4	Bơ
5	Bánh mì
6	Nem
7	Táo
8	Cá hộp

3. Truy vấn dữ liệu

Hiển thị các dòng dữ liệu không trùng lặp

```
select distinct * from dbo.SanPham
```

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

```
select distinct MaSP, TenSP from dbo.SanPham
```

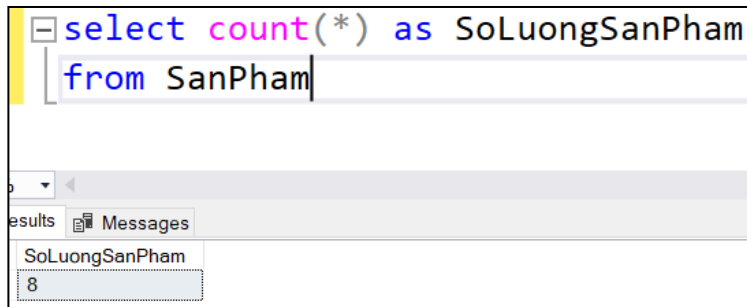
MaSP	TenSP
1	Rượu
2	Gia vị
3	Bánh kem
4	Bơ
5	Bánh mì
6	Nem
7	Táo
8	Cá hộp

3. Truy vấn dữ liệu

Khi truy vấn có 3 hàm thường xuyên sử dụng:

- COUNT()**: Đếm số lượng dòng truy vấn được:
- SUM()**: Tính tổng giá trị của cột được chỉ định
- AVG()**: Tính giá trị trung bình của cột được chỉ định
- MIN()**: Tìm giá trị nhỏ nhất của cột được chỉ định
- Max()**: Tìm giá trị lớn nhất của cột được chỉ định

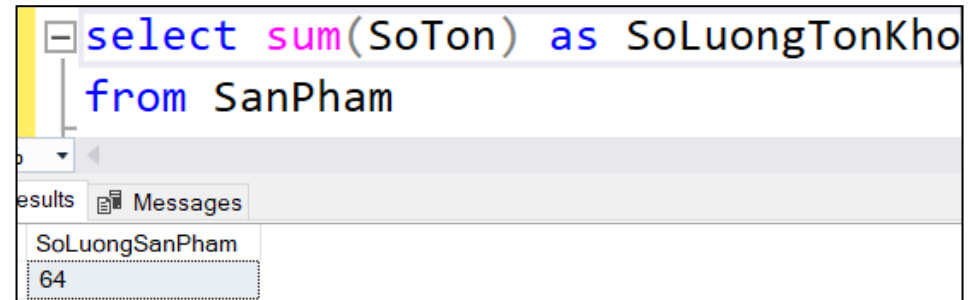
Cho biết số lượng sản phẩm



```
select count(*) as SoLuongSanPham
from SanPham
```

SoLuongSanPham
8

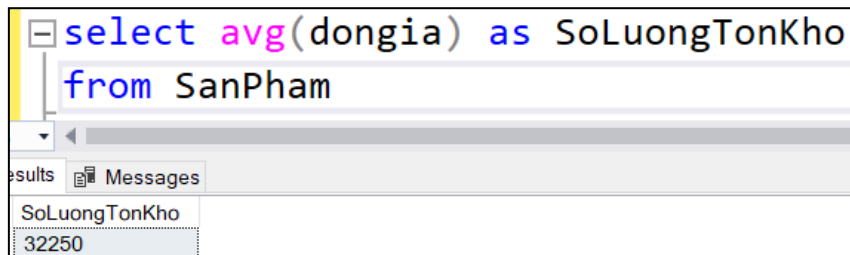
Cho biết số lượng hàng còn tồn kho



```
select sum(SoTon) as SoLuongTonKho
from SanPham
```

SoLuongTonKho
64

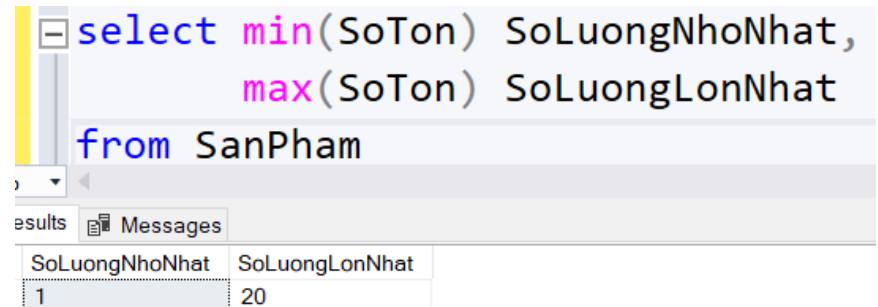
Cho biết đơn giá trung bình của các sản phẩm



```
select avg(dongia) as SoLuongTonKho
from SanPham
```

SoLuongTonKho
32250

Tìm số tồn nhỏ nhất và lớn nhất



```
select min(SoTon) SoLuongNhoNhat,
       max(SoTon) SoLuongLonNhat
from SanPham
```

SoLuongNhoNhat	SoLuongLonNhat
1	20

3. Truy vấn dữ liệu

3.2. Truy vấn có điều kiện

```
SELECT [tính chất] <cột 1, cột 2, ... Cột n>  
FROM <table> [as Alias]  
WHERE <Điều kiện>
```

<Điều kiện> là điều kiện để lọc dữ liệu, trích xuất các record chứa dữ liệu thỏa điều kiện
Có thể sử dụng toán tử logic, toán tử so sánh trong điều kiện
WHERE

3. Truy vấn dữ liệu

Bảng sản phẩm có dữ liệu như sau:

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

- Hiển thị các sản phẩm có đơn giá lớn hơn 50.000

```
SELECT *  
FROM SanPham  
WHERE DonGia > 50000;
```

- Hiển thị sản phẩm có đơn vị tính là 'Cái'

```
SELECT *  
FROM SanPham  
WHERE DVT LIKE N'Cái';
```

- Hiển thị sản phẩm có tên bắt đầu là từ 'Bánh'

```
SELECT *  
FROM SanPham  
WHERE TenSP LIKE N'Bánh%';
```

3. Truy vấn dữ liệu

Kết hợp biểu thức Logic:

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

Hiển thị thông tin sản phẩm có đơn giá lớn hơn 10.000 và nhỏ hơn 20.000

```
SELECT *  
FROM SanPham  
WHERE DonGia >10000 AND  
      DonGia < 20000;
```

Hiển thị thông tin sản phẩm không phải Rượu

```
SELECT *  
FROM SanPham  
WHERE TenSP NOT LIKE N'RƯỢU';
```

Hiển thị thông tin sản phẩm có đơn giá từ 10.000 đến 20.000

```
SELECT *  
FROM SanPham  
WHERE  
DonGia BETWEEN 10000 AND 20000;
```

3. Truy vấn dữ liệu

Kết hợp biểu thức Logic:

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	NULL
8	Cá hộp	Thùng	20000	NULL

Hiển thị thông tin sản phẩm chưa với số tồn có giá trị không xác định

```
SELECT *  
FROM SanPham  
WHERE SoTon IS NULL;
```

Hiển thị thông tin sản phẩm với Số tồn có giá trị xác định

```
SELECT *  
FROM SanPham  
WHERE SoTon IS NOT NULL;
```

3. Truy vấn dữ liệu

Kết hợp biểu thức Logic:

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

Hiển thị thông tin sản phẩm:
Rượu, Gia vị, Bánh kem

```
SELECT *  
FROM SanPham  
WHERE TenSP IN(N' Rượu',  
                N' Gia vị',  
                N' Bánh kem');
```

Hiển thị thông tin sản phẩm:
Rượu, Gia vị, Bánh kem

```
SELECT *  
FROM SanPham  
WHERE TenSP LIKE N' Rượu' OR  
       TenSP LIKE N' Gia vị' OR  
       TenSP LIKE N' Bánh kem';
```

Hiển thị thông tin sản phẩm:
Rượu, Gia vị, Bánh kem

```
SELECT *  
FROM SanPham  
WHERE TenSP NOT IN(N' Rượu',  
                    N' Gia vị',  
                    N' Bánh kem');
```

3. Truy vấn dữ liệu

3.3. Truy vấn nhiều bảng

```
SELECT [tính chất] <cột 1, cột 2, ... Cột n>  
FROM <table1, table2, ... /View1, view2,...> [as Alias]  
WHERE <Điều kiện>
```

Có thể truy xuất dữ liệu từ nhiều bảng khác nhau trong 1 câu lệnh
SELECT

3. Truy vấn dữ liệu

NhacTruong

MaNhatTruong	TenNhatTruong
1	Trịnh Công Sơn
2	Đức Huy
3	Trần Thiện Thanh

BuoiHoaNhat

MaBHN	MaNhatTruong	TenBHM
1	1	Buổi hòa nhạc 1
2	2	Buổi hòa nhạc 2
3	3	Buổi hòa nhạc 3

SELECT *

FROM NhatTruong NT, BuoiHoaNhat BHN

MaNhatTruong	TenNhatTruong	MaBHN	MaNhatTruong	TenBHM
1	Trịnh Công Sơn	1	1	Buổi hòa nhạc 1
2	Đức Huy	1	1	Buổi hòa nhạc 1
3	Trần Thiện Thanh	1	1	Buổi hòa nhạc 1
1	Trịnh Công Sơn	2	2	Buổi hòa nhạc 2
2	Đức Huy	2	2	Buổi hòa nhạc 2
3	Trần Thiện Thanh	2	2	Buổi hòa nhạc 2
1	Trịnh Công Sơn	3	3	Buổi hòa nhạc 3
2	Đức Huy	3	3	Buổi hòa nhạc 3
3	Trần Thiện Thanh	3	3	Buổi hòa nhạc 3

3. Truy vấn dữ liệu

NhacTruong

MaNhatTruong	TenNhatTruong
1	Trịnh Công Sơn
2	Đức Huy
3	Trần Thiện Thanh

BuoiHoaNhat

MaBHN	MaNhatTruong	TenBHM
1	1	Buổi hòa nhạc 1
2	2	Buổi hòa nhạc 2
3	3	Buổi hòa nhạc 3

```
SELECT BHN.MaBHN, BHN.TenBHM, NT.TenNhatTruong
FROM BuoiHoaNhat BHN, NhatTruong NT
WHERE BHN.MaNhatTruong = NT.MaNhatTruong
```

MaBHN	TenBHM	TenNhatTruong
1	Buổi hòa nhạc 1	Trịnh Công Sơn
2	Buổi hòa nhạc 2	Đức Huy
3	Buổi hòa nhạc 3	Trần Thiện Thanh

3. Truy vấn dữ liệu

3.4. Truy vấn với GROUP BY:

Mệnh đề **GROUP BY** trong SQL được sử dụng kết hợp trong câu lệnh **SELECT** để sắp xếp dữ liệu giống nhau thành các nhóm.

Thông thường, GROUP BY được sử dụng để tính tổng theo điều kiện, đếm dòng dữ liệu thỏa mãn điều kiện nào đó hoặc tìm dữ liệu MIN, MAX

```
SELECT <cột 1, cột 2, ... cột n>  
FROM <Bảng 1, bảng 2, .... bảng n>  
[WHERE <Điều kiện> ]  
[GROUP BY <cột A, cột B, ...>]
```

3. Truy vấn dữ liệu

NhacTruong

MaNhacTruong	TenNhacTruong
1	Trịnh Công Sơn
2	Đức Huy
3	Trần Thiện Thanh
4	Vũ Đình Trung

BuoiHoaNhac

MaBHN	MaNhacTruong	TenBHM
1	1	Buổi hòa nhạc 1
2	2	Buổi hòa nhạc 2
3	3	Buổi hòa nhạc 3
4	1	Buổi hòa nhạc 4
5	1	Buổi hòa nhạc 5
6	3	Buổi hòa nhạc 6

Hiển thị danh sách các nhạc trưởng cùng số lượng buổi hòa nhạc đã điều khiển.

→ Phải hiển thị được danh sách nhạc trưởng cùng buổi hòa nhạc mà người đó điều khiển trước

→ Nhóm dữ liệu bằng GROUP BY để đếm số lượng buổi hòa nhạc mà người đó điều khiển

3. Truy vấn dữ liệu

→ Phải hiển thị được danh sách nhạc trưởng cùng buổi hòa nhạc mà người đó điều khiển trước

```
select NT.MaNhacTruong, NT.TenNhacTruong, BHN.TenBHM
from NhacTruong NT, BuiHoaNhac BHN
where NT.MaNhacTruong = BHN.MaNhacTruong
```

MaNhacTruong	TenNhacTruong	TenBHM
1	Trịnh Công Sơn	Buổi hòa nhạc 1
2	Đức Huy	Buổi hòa nhạc 2
3	Trần Thiện Thanh	Buổi hòa nhạc 3
1	Trịnh Công Sơn	Buổi hòa nhạc 4
1	Trịnh Công Sơn	Buổi hòa nhạc 5
3	Trần Thiện Thanh	Buổi hòa nhạc 6

3. Truy vấn dữ liệu

→ Nhóm dữ liệu bằng **GROUP BY** để đếm số lượng buổi hòa nhạc mà người đó điều khiển

```
select NT.MaNhacTruong, NT.TenNhacTruong, count(BHN.TenBHM) SoLuongBHN
from NhacTruong NT, BuiHoaNhac BHN
where NT.MaNhacTruong = BHN.MaNhacTruong
group by NT.MaNhacTruong, NT.TenNhacTruong
```

MaNhacTruong	TenNhacTruong	SoLuongBHN
1	Trịnh Công Sơn	3
2	Đức Huy	1
3	Trần Thiện Thanh	2

3. Truy vấn dữ liệu

3.5. HAVING:

HAVING được dùng kết hợp với mệnh đề GROUP BY, để giới hạn nhóm các hàng trả về, chỉ khi điều kiện được đáp ứng là **TRUE**

```
SELECT <cột 1, cột 2, ... cột n>  
FROM <Bảng 1, bảng 2, .... bảng n>  
[WHERE <Điều kiện> ]  
[GROUP BY <cột A, cột B, ...>]  
[HAVING <Điều kiện Having> ]
```

3. Truy vấn dữ liệu

Trường hợp ta đã biết được thông tin nhạc trưởng và số buổi hòa nhạc người đó biểu diễn.

Tiếp tục muốn **hiển thị những nhạc trưởng điều khiển nhiều hơn 1 buổi hòa nhạc**

```
select NT.MaNhacTruong, NT.TenNhacTruong, count(BHN.TenBHM) SoLuongBHN
from NhacTruong NT, BuiHoiHoaNhac BHN
where NT.MaNhacTruong = BHN.MaNhacTruong
group by NT.MaNhacTruong, NT.TenNhacTruong
having count(BHN.TenBHM)>1
```

MaNhacTruong	TenNhacTruong	SoLuongBHN
1	Trịnh Công Sơn	3
3	Trần Thiện Thanh	2

3. Truy vấn dữ liệu

3.6. ORDER BY:

ORDER BY sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần hoặc giảm dần, dựa trên một hoặc nhiều cột.

Lệnh ASC được sử dụng để sắp xếp tăng dần và DESC được sử dụng để sắp xếp giảm dần. Nếu không chỉ định ASC hoặc DESC trong câu lệnh thì mặc định là ASC

```
SELECT <cột 1, cột 2, ... cột n>  
FROM <Bảng 1, bảng 2, .... bảng n>  
[WHERE <Điều kiện> ]  
[GROUP BY <cột 1, cột 2, ...>]  
[HAVING <Điều kiện Having> ]  
[ORDER BY <cột 1, cột 2, ...>] [DESC/ ASC]
```


3. Truy vấn dữ liệu

Kết hợp biểu thức Logic:

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

Hiển thị danh sách sản phẩm
với tên sản phẩm tăng dần

```
SELECT *  
FROM SanPham  
ORDER BY TenSP ASC;
```

MaSP	TenSP	DVT	DonGia	SoTon
3	Bánh kem	Cái	100000	4
5	Bánh mì	Cái	12000	10
4	Bơ	Kg	7000	10
8	Cá hộp	Thùng	20000	NULL
2	Gia vị	Thùng	14000	1
6	Nem	Kg	25000	20
1	Rượu	Chai	50000	7
7	Táo	Kg	30000	15

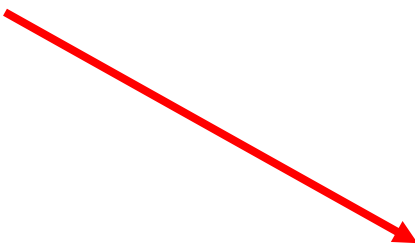
3. Truy vấn dữ liệu

Kết hợp biểu thức Logic:

MaSP	TenSP	DVT	DonGia	SoTon
1	Rượu	Chai	50000	7
2	Gia vị	Thùng	14000	1
3	Bánh kem	Cái	100000	4
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
6	Nem	Kg	25000	20
7	Táo	Kg	30000	2
8	Cá hộp	Thùng	20000	10

Hiển thị danh sách sản phẩm
với tên sản phẩm giảm dần

```
SELECT *  
FROM SanPham  
ORDER BY TenSP DESC;
```



MaSP	TenSP	DVT	DonGia	SoTon
7	Táo	Kg	30000	15
1	Rượu	Chai	50000	7
6	Nem	Kg	25000	20
2	Gia vị	Thùng	14000	1
8	Cá hộp	Thùng	20000	NULL
4	Bơ	Kg	7000	10
5	Bánh mì	Cái	12000	10
3	Bánh kem	Cái	100000	4

Câu hỏi ???