

Ngôn ngữ SQL – Truy vấn đơn giản

Tổng quát

Một cách tổng quát, khối select gồm có 3 mệnh đề chính:

Select: Xác định các cột cần đưa ra kết quả.

From: Xác định các bảng cần lấy thông tin ra.

Where: Xác định các mẫu tin thỏa yêu cầu chọn lọc để đưa ra kết quả.

Ngoài ra, để mở rộng khả năng của ngôn ngữ, khối select-from-where còn được bổ sung thêm các mệnh đề **group by**, **having**, **order by**, các hàm hỗ trợ tính toán: **max**, **min**, **count**, **sum**, **avg**.

Sau đây là cú pháp tổng quát của câu truy vấn dữ liệu:

```

SELECT [tính chất] <danhsach các thuộc tính_1>

FROM <danhsach các table hoặc query/view [as alias] >

[WHERE <điều kiện_1>]

[GROUP BY <danhsach các thuộc tính_2>]

[HAVING <điều kiện_2>]

[ORDER BY <danhsach các thuộc tính_3 [ASC | DESC]>]

```

Diễn giải :

1. Tính chất : Một trong các từ khóa: ALL (chọn ra tất cả các dòng trong bảng), DISTINCT (loại bỏ các cột trùng lặp thông tin), DISTINCTROW (loại bỏ các dòng trùng lặp thông tin), TOP <n> (chọn n dòng đầu tiên thỏa mãn điều kiện).
2. Danh sách các thuộc tính_1: tên các thuộc tính cho biết thông tin cần lấy.

Chú ý: Các thuộc tính cách nhau bởi dấu ‘,’

Nếu lấy tất cả các thuộc tính của 1 bảng tbl thì dùng: tbl.*

Nếu sau FROM chỉ có 1 table và lấy tất cả các field của table đó thì dùng select *

Nếu tồn tại 1 thuộc tính sau select xuất hiện ở 2 table sau FROM thì phải chỉ định rõ thuộc tính đó thuộc table nào.

3. Danh sách các table: các table chứa thông tin cần lấy. Khi tìm kiếm thông tin trên nhiều hơn 2 table thì phải kết các table lại với nhau (điều kiện kết đặt sau where)
4. Alias: bí danh (tên tắt) của bảng dùng cho các bảng có tên quá dài.
5. Điều kiện_1: là điều kiện để lọc dữ liệu.

6. Danh sách các thuộc tính_2: dữ liệu sẽ được gom nhóm theo các cột này, ưu tiên từ trái sang.
7. Điều kiện_2: điều kiện lọc lại dữ liệu sau khi đã thực hiện tính toán trên dữ liệu. Điều kiện này được áp dụng trên dữ liệu thỏa mãn điều kiện_1.
8. Danh sách các thuộc tính_3: sắp xếp dữ liệu theo cột nào, thứ tự là tăng (ASC) hoặc giảm (DESC). Mặc định là dữ liệu được sắp theo thứ tự tăng dần. Việc sắp xếp được thực hiện theo thứ tự ưu tiên từ trái qua phải.

Truy vấn đơn giản

SELECT <danh sách thuộc tính>

FROM tên_bảng

Sau select, * được dùng với ý nghĩa lấy toàn bộ các cột của bảng.

Dùng từ khoá **distinct** để loại bỏ các bộ trùng nhau và **all** để lấy tất cả các bộ dữ liệu. Mặc định không để gì cả chính là có dùng từ khóa all.

Sau select có thể dùng các biểu thức số học như: +, -, *, /, và có thể thực hiện các toán tử trên thuộc tính.

VD:

- Cho biết danh sách tất cả các nhân viên với tất cả các thông tin

```
SELECT *
FROM NHANVIEN
```

A. Tìm kiếm có sắp xếp

Để sắp xếp thứ tự dữ liệu, ta sử dụng mệnh đề ORDER BY:

SELECT...

FROM...

ORDER BY thuộc_tính_1[ASC|DESC], thuộc_tính_2[ASC|DESC], ...

Tập_thuộc_tính gồm 1 thuộc tính hoặc nhiều thuộc tính và độ ưu tiên tính từ trái sang phải.

VD:

Với câu lệnh: *select * from Table1 order by B desc, A asc* trên bảng dưới đây:

A	B
An	8
Binh	8

Chi	9
Hung	10

Ta sẽ được kết quả sau:

A	B
Hung	10
Chi	9
An	8
Binh	8

Đầu tiên là xếp thứ tự theo B trước, sau đó, với những giá trị B ngang nhau thì sẽ xếp theo A.

VD:

- Cho biết danh sách các nhân viên sắp tên theo thứ tự Alphabet

```
SELECT MANV, TENNV, PHAI, LUONG
FROM NHANVIEN
ORDER BY TENNV
|
```

- Cho biết danh sách các nhân viên theo từng phòng ban, trong từng phòng ban tên nhân viên sắp theo thứ tự

```
SELECT PHG, MANV, TENNV, PHAI, LUONG
FROM NHANVIEN
ORDER BY PHG, TENNV
```

B. Tìm kiếm với điều kiện đơn giản

Để hỗ trợ tìm kiếm có điều kiện, sử dụng mệnh đề WHERE trong câu lệnh SELECT với vị trí như sau:

1. AND và OR

```
SELECT...
FROM...
WHERE (điều_kiện_1) AND/OR ....(điều_kiện_n)
```

VD:

SINHVIEN (MASV, HOTEN, NGSINH, LOP)

- Cho danh sách các sinh viên của lớp TH01.

```
SELECT *
FROM SINHVIEN
WHERE LOP = 'TH01'
```

Lưu ý: Khi thuộc tính có thể nhận giá trị null, cần cẩn thận khi sử dụng để so sánh với nhiều điều kiện liên tiếp.

2. BETWEEN...AND , NOT BETWEEN ... AND

- Cho biết các nhân viên sinh trong khoảng năm 1955 đến 1960

```
SELECT *
FROM NHANVIEN
WHERE NGSINH between '1/1/1955' and '12/31/1960'
```

Hoặc

```
SELECT *
FROM NHANVIEN
WHERE year(NGSINH) between 1955 and 1960
```

Hoặc

```
SELECT *
FROM NHANVIEN
WHERE year(NGSINH) >= 1955 and year(NGSINH) <= 1960
```

3. IS NULL và IS NOT NULL

IS NULL và IS NOT NULL : Để kiểm tra một giá trị có phải là NULL | NOT NULL hay không

- Cho biết các nhân viên không có người quản lý trực tiếp

```
SELECT HONV, TENLOT, TENNV
FROM NHANVIEN
WHERE MA_NQL IS NULL
```

- Cho biết các nhân viên có người quản lý trực tiếp

```
SELECT HONV, TENLOT, TENNV
FROM NHANVIEN
WHERE MA_NQL IS NOT NULL
```

4. IN và NOT IN

IN và NOT IN dùng để kiểm tra một giá trị nằm trong hay không nằm trong một tập hợp nào đó hay không.

- Cho biết các đơn đặt hàng có đặt mặt hàng H1, H2, H3.

```
SELECT MADH
FROM DONHANG
WHERE MAMH IN ('H1', 'H2', 'H3')
```

C. Tìm kiếm có xử lý chuỗi ký tự

Để xử lý với các dữ liệu thuộc dạng chuỗi ký tự, ngôn ngữ SQL có hỗ trợ phép LIKE. Thông thường khi so sánh thuộc tính có kiểu dữ liệu thuộc dạng chuỗi ký tự thì người ta thường dùng LIKE chứ không dùng phép bằng =

VD:

- Hiện ra các sinh viên tên Trang

```
SELECT *
FROM SINHVIEN
WHERE HOTEN LIKE '%Trang'
```

% : dùng để đại diện cho nhiều ký tự đứng trước từ 'Trang'

Ngoài ra còn có các ký tự sau để mô tả mẫu cần tìm:

_ thay thế cho ký tự bất kỳ.

Chú ý:

Like "ab%cd%" cho ra những chuỗi bắt đầu với "ab%cd"

Like "ab\\cd%" cho ra những chuỗi bắt đầu với "ab\\cd"

D. Tìm kiếm có điều kiện liên quan đến ngày tháng

VD:

DH(MADH, NGÀYDH, MAKH)

CTDH(MADH, MAHH, SOLUONG, DONGIA)

- Cho biết những đơn đặt hàng đặt trước ngày 01/01/2001

```
SELECT MADH, NGÀYDH
FROM DDH
WHERE DATEDIFF(D, NGÀYDH, '01/01/2001') > 0
```

- Cho biết những đơn đặt hàng đặt trước ngày 01/01/2001 là 1 tuần

```
SELECT MADH, NGÀYDH
FROM DDH
WHERE DATEDIFF(D, NGÀYDH, '01/01/2001') > 7
```

Lưu ý:

- Cho biết các nhân viên sinh ngày 30/4/1975

Cách 1 :

```
select *
from NHANVIEN
where NGSINH = '1/1/1965'
```

Cách 2 :

```
select *
from NHANVIEN
where DATEDIFF(d, NGSINH, '1/1/1965') = 0
```

→ Cách 2 : Chính xác hơn

E. Sử dụng các hàm trong khi tìm kiếm

- Sử dụng hàm trong mệnh đề where
- Sử dụng hàm trong mệnh đề select : Trong mệnh đề select ngoài việc được sử dụng các toán tử như +, -, *, / ta còn có thể sử dụng hàm đối với các thuộc tính.
 - Các hàm về ngày tháng :
 - DateDiff
 - DatePart
 - GetDate
 - Year
 - Month
 - Day
 - DateAdd
 - Các hàm về chuỗi
 - Các hàm chuyển đổi kiểu dữ liệu
 - Các hàm toán học
 - ...

Để xem thông tin chi tiết về các hàm có thể sử dụng Book Onlines

- Cho biết họ tên nhân viên và tuổi của nhân viên

```
select HONV, TENLOT, TENNV, datediff(yyyy, NGSINH, getdate()) as TUOI
from NHANVIEN
```

- Cho biết năm sinh của nhân viên

```
select HONV, TENLOT, TENNV, year(NGSINH) as NAMSINH  
from NHANVIEN
```

- Cho biết họ và tên đầy đủ của nhân viên

```
select HONV + ' ' + TENLOT + ' ' + TENNV as HoVaTen  
from NHANVIEN
```

F. Dùng toán tử θ some, θ all, exists, not exists

Lưu ý: $\langle \rangle$ some và not in, $\langle \rangle$ all = not in.