

Bài 1: TRUY VẤN DỮ LIỆU ĐƠN GIẢN

Nội dung

- Cú pháp chung của câu lệnh SELECT
- Truy vấn dữ liệu đơn giản
- Sắp xếp dữ liệu trong kết quả truy vấn

Cú pháp câu lệnh SELECT

```
SELECT      [ALL|DISTINCT] [TOP (n) [PERCENT]]
            select_list_items
[INTO       name_of_new_table]
FROM        table(s)
[WHERE      search_conditions]
[GROUP BY  list_of_columns]
[HAVING     search_conditions]
[ORDER BY  list_of_sorted_columns]
```

- Các mệnh đề xuất hiện trong câu lệnh SELECT phải đúng theo thứ tự qui định của cú pháp.
- Mệnh đề SELECT và FROM là 2 mệnh đề bắt buộc.
- Các mệnh đề còn lại là tùy chọn.

Kết quả của truy vấn [1]

- Câu lệnh SELECT khi thực thi sẽ trả về kết quả là dữ liệu được hiển thị dưới dạng một bảng, bao gồm:
 - Một hoặc nhiều cột (column),
 - Không hoặc nhiều dòng (row)

Kết quả của truy vấn [2]

```
SELECT  MASV,
        HODEM,
        TEN,
        NGAYSINH
FROM    SINHVIEN
```

MASV	HODEM	TEN	NGAYSINH
0241010001	Ngô Thị Nhật	Anh	1982-11-27
0241010002	Nguyễn Thị Ngọc	Anh	1983-03-21
0241010003	Ngô Việt	Bắc	1982-05-11
0241010004	Nguyễn Đình	Bình	1982-10-06
0241010005	Hồ Đăng	Chiến	1982-01-20
0241020001	Nguyễn Tuấn	Anh	1979-07-15
0241020002	Trần Thị Kim	Anh	1982-11-04
0241020003	Võ Đức	Ấn	1982-05-24
0241020004	Nguyễn Công	Bình	1979-06-06

Câu lệnh SELECT đơn giản [1]

- Ở dạng đơn giản nhất, một câu lệnh SELECT được sử dụng để biểu diễn yêu cầu truy vấn dữ liệu từ một bảng (table) trong cơ sở dữ liệu. Thông thường, bạn sử dụng câu lệnh SELECT để mô tả:
 - Bạn cần biết những thông tin gì?
 - Thông tin mà bạn cần biết được lấy từ bảng nào?
 - Có điều kiện gì không?
- Trong trường hợp này, câu lệnh SELECT thường bao gồm các mệnh đề:
 - SELECT:** Danh sách các cột cần được hiển thị trong kết quả truy vấn. Các cột được phân cách nhau bởi dấu phẩy.
 - FROM:** Tên của bảng chứa dữ liệu cần truy vấn.
 - WHERE:** Điều kiện được sử dụng để xác định những dòng dữ liệu nào sẽ được hiển thị trong kết quả truy vấn.

Câu lệnh SELECT đơn giản [2]

- Ví dụ:** Hiển thị mã sinh viên, họ, tên, ngày sinh của các sinh viên sinh ở Huế.

```
SELECT  MASV,
        HODEM,
        TEN,
        NGAYSINH
FROM    SINHVIEN
WHERE   NOISINH = N'Huế'
```

Danh sách các cột cần hiển thị trong kết quả truy vấn

Tên của bảng cần truy vấn dữ liệu

Điều kiện truy vấn dữ liệu

Mệnh đề FROM

- Mệnh đề FROM được sử dụng để chỉ định bảng dữ liệu cần truy vấn dữ liệu. Ngay sau mệnh đề FROM là tên của bảng mà câu lệnh SELECT sử dụng để truy vấn dữ liệu từ đó
- Thay vì sử dụng tên của bảng, bạn có thể sử dụng một tên khác để đặt bí danh (alias) cho bảng. Khi một bảng đã được đặt bí danh, trong câu truy vấn bạn phải sử dụng bí danh thay vì sử dụng tên của bảng.

Ví dụ: Câu lệnh dưới đây đặt bí danh cho bảng SINHVIEN là sv

```
SELECT  sv.MASV, sv.HODEM, sv.TEN
FROM    SINHVIEN AS sv
WHERE   NOISINH = N'Huế'
```

Vì bảng SINHVIEN đã được đặt bí danh là sv, nên phải sử dụng bí danh thay vì tên bảng

Danh sách chọn trong câu lệnh SELECT [1]

- Sau mệnh đề SELECT là danh sách các biểu thức cần được hiển thị trong kết quả truy vấn (gọi là danh sách chọn), phân cách nhau bởi dấu phẩy.
- Một biểu thức trong danh sách chọn có thể là:
 - Tên của một cột trong bảng (column name)
 - Một giá trị hằng (constant)
 - Một biểu thức tính toán (expression)

Ví dụ:

```
SELECT  HODEM, TEN,
        'có tuổi là',
        YEAR(GETDATE()) - YEAR(NGAYSINH)
FROM    SINHVIEN
```

Tên cột

Hằng

Biểu thức tính toán

Danh sách chọn trong câu lệnh SELECT [2]

- Dấu sao (*) được sử dụng trong trường hợp bạn cần hiển thị tất cả các cột có trong bảng.

```
SELECT * FROM SINHVIEN
```

- Bạn có thể thay đổi tiêu đề của các cột được hiển thị trong kết quả truy vấn bằng cách đặt tiêu đề cho cột, như ở ví dụ sau:

```
SELECT    MASV AS MaSinhVien,
          HODEM + ' ' + TEN AS [Họ và tên],
          YEAR(NGAYSINH) AS 'Năm sinh'
FROM      SINHVIEN
```

MaSinhVien	Họ và tên	Năm sinh
0241010001	Ngô Thị Nhật Anh	1982
0241010002	Nguyễn Thị Ngọc Anh	1983
0241010003	Ngô Việt Bắc	1982

Tiêu đề cột phải đặt trong cặp dấu nháy đơn hoặc ngoặc vuông nếu có khoảng trắng hoặc ký tự đặc biệt.

Danh sách chọn trong câu lệnh SELECT [3]

- Trong câu lệnh SELECT, bạn có thể sử dụng các biểu thức tính toán để thực hiện những phép tính trên các cột.
- Một biểu thức tính toán sử dụng các phép toán (+, -, *, /,...) và có thể kết hợp với các hàm để thực hiện được những phép tính theo yêu cầu truy vấn.

```
SELECT MASV, HODEM, TEN,
       YEAR(GETDATE()) - YEAR(NGAYSINH) AS TUOI
FROM SINHVIEN
```

MASV	HODEM	TEN	TUOI
0241010001	Ngô Thị Nhật	Anh	21
0241010002	Nguyễn Thị Ngọc	Anh	20
0241010003	Ngô Việt	Bắc	21

Biểu thức SQL

Danh sách chọn trong câu lệnh SELECT [4]

- Cấu trúc CASE trong câu lệnh SELECT được sử dụng để xử lý và đưa ra những lựa chọn khác nhau cho giá trị đầu ra dựa trên giá trị của biểu thức đầu vào.
- Cú pháp của cấu trúc CASE như sau:

```
CASE biểu_thức_đầu_vào
WHEN giá_trị_biểu_thức THEN giá_trị_đầu_ra
[...n]
[ELSE giá_trị_đầu_ra_của_ELSE]
END
```

Ví dụ:

```
SELECT    MASV, HODEM, TEN,
          CASE GIOITINH
            WHEN 0 THEN 'Nữ'
            WHEN 1 THEN 'Nam'
            ELSE 'Không xác định'
          END AS GIOITINHSV
FROM      SINHVIEN
```

Danh sách chọn trong câu lệnh SELECT [5]

- Cấu trúc CASE còn có thể sử dụng dưới dạng cú pháp khác như sau:

```
CASE
  WHEN điều_kiện THEN giá_trị_đầu_ra
  [...]
  [ELSE giá_trị_đầu_ra_của_ELSE]
END
```

- Ví dụ:

```
SELECT    MASV, HODEM, TEN,
          CASE
            WHEN GIOITINH = 0 THEN 'Nữ'
            WHEN GIOITINH = 1 THEN 'Nam'
            ELSE 'Không xác định'
          END AS GIOITINHSV
FROM      SINHVIEN
```

Loại bỏ các dòng lặp trong kết quả truy vấn

- Trong kết quả truy vấn, có thể sẽ xuất hiện những dòng dữ liệu trùng nhau.
- Từ khóa DISTINCT được sử dụng ngay sau mệnh đề SELECT để chỉ giữ lại một dòng trong số các dòng dữ liệu trùng nhau.

Ví dụ:

```
SELECT KHOA, NAMNHAPHOC FROM LOP
SELECT DISTINCT KHOA, NAMNHAPHOC FROM LOP
```

KHOA	NAMNHAPHOC
24	2000
24	2000
24	2000
24	2000
25	2001
25	2001
25	2001
25	2001
26	2002
26	2002

KHOA	NAMNHAPHOC
24	2000
25	2001
26	2002

Hạn chế số lượng dòng trong kết quả truy vấn

- Để hạn chế số lượng dòng dữ liệu được hiển thị trong kết quả truy vấn, bạn có thể sử dụng mệnh đề TOP ngay sau SELECT.
- Số lượng dòng được hạn chế có thể là một con số hoặc cũng có thể theo phần trăm của tổng số lượng dòng.

Ví dụ:

```
SELECT TOP (5) MASV, HODEM, TEN
FROM      SINHVIEN
```

Hiển thị mã, họ tên của 5% sinh viên đầu tiên trong tổng số sinh viên hiện có

```
SELECT TOP (5) PERCENT
MASV, HODEM, TEN
FROM      SINHVIEN
```

- Thông thường, việc hạn chế số lượng dòng với mệnh đề TOP thường được kết hợp với việc sắp xếp kết quả truy vấn bằng cách sử dụng mệnh đề ORDER BY.

Điều kiện của câu truy vấn [1]

- Mệnh đề WHERE được sử dụng trong câu lệnh SELECT để chỉ định điều kiện đối với các dòng dữ liệu cần truy vấn. Trong trường hợp này, chỉ những dòng dữ liệu thỏa mãn điều kiện được chỉ định sau WHERE mới được hiển thị trong kết quả truy vấn.
- Việc xác định dòng dữ liệu được hiển thị trong kết quả truy vấn dựa trên điều kiện sau WHERE như sau:
 - Nếu điều kiện trên dòng dữ liệu là đúng (TRUE), dòng dữ liệu sẽ được giữ lại trong kết quả truy vấn.
 - Nếu điều kiện trên dòng dữ liệu là sai (FALSE), dòng dữ liệu sẽ bị loại ra khỏi kết quả truy vấn.
 - Nếu điều kiện trên dòng dữ liệu trả về kết quả NULL, dòng dữ liệu cũng sẽ bị loại ra khỏi kết quả truy vấn.

Điều kiện của câu truy vấn [2]

- Ví dụ: Hiển thị mã, họ, tên và ngày sinh của các sinh viên sinh vào năm 1983:

```
SELECT  MASV , HODEM , TEN , NGAYSINH
FROM    SINHVIEN
WHERE   YEAR (NGAYSINH) = 1983
```

Điều kiện truy vấn

MASV	HODEM	TEN	NGAYSINH
0241010002	Nguyễn Thị Ngọc	Anh	1983-03-21
0251010001	Nguyễn Ngọc	Chinh	1983-11-05
0251010002	Huỳnh Thị Yến	Giang	1983-05-20
0251020001	Dương Văn	Anh	1983-01-24
0251020002	Nguyễn Hoàng	Anh	1983-04-27
0251020003	Nguyễn Thế	Anh	1983-01-26
0251020004	Nguyễn Thị Lan	Anh	1983-08-01
0251020005	Vô Thị Kim	Anh	1983-07-10

Điều kiện của câu truy vấn [3]

- SQL hỗ trợ các loại điều kiện truy vấn dữ liệu như sau:
 - Các phép so sánh (comparison test):** Sử dụng các toán tử so sánh để so sánh của hai biểu thức với nhau.
 - So sánh trong một đoạn giá trị (range test):** Kiểm tra xem giá trị của một biểu thức có (hoặc không) nằm trong một đoạn giá trị hay không.
 - So sánh trong tập giá trị (set membership test):** Kiểm tra xem giá trị của một biểu thức có (hoặc không) nằm trong một tập hợp các giá trị hay không.
 - Đối sánh mẫu (pattern matching test):** Kiểm tra xem giá trị của một biểu thức (kiểu chuỗi) có thỏa mãn một khuôn dạng dữ liệu nào đó hay không.
 - Kiểm tra giá trị NULL (NULL value test):** Kiểm tra xem giá trị của một biểu thức bằng (hoặc không bằng) NULL hay không?

Phép so sánh trong điều kiện truy vấn [1]

- Phép so sánh được sử dụng để SQL tính toán và so sánh giá trị của hai biểu thức với nhau.
- SQL cung cấp các phép toán so sánh số học sau đây:

=
<>
>
>=
<
<=

Phép so sánh trong điều kiện truy vấn [2]

- **Ví dụ:** Hiển thị mã sinh viên, họ, tên và ngày sinh của ccs sinh viên có tuổi lớn hơn 22.

```
SELECT MASV, HODEM, TEN, NGAYSINH
FROM SINHVIEN
WHERE YEAR(GETDATE()) - YEAR(NGAYSINH) > 22
```

Phép so sánh

Kết hợp các biểu thức điều kiện

- SQL cho phép kết hợp các biểu thức điều kiện với nhau để tạo thành các biểu thức phức tạp, sử dụng các toán tử logic sau:

- AND
- OR
- NOT

- **Ví dụ:** Hiển thị mã, họ, tên, ngày sinh của sinh viên Nữ sinh ở Huế.

```
SELECT MASV, HODEM, TEN, NGAYSINH
FROM SINHVIEN
WHERE (GIOITINH = 0)
AND (NOISINH = N'Huế')
```

Phép so sánh trong đoạn giá trị [1]

- SQL cung cấp phép toán BETWEEN để thực hiện phép so sánh giá trị của một biểu thức với một đoạn giá trị.
- Phép toán BETWEEN có thể sử dụng với cú pháp như sau:

Cú pháp sử dụng	Biểu thức điều kiện tương đương
x BETWEEN a AND b	(x >= a) AND (x <= b)
x NOT BETWEEN a AND b	(x < a) OR (x > b)

Phép so sánh trong đoạn giá trị [2]

- Ví dụ:** Hiển thị danh sách các môn học có số đơn vị học trình lớn hơn hoặc bằng 3 và nhỏ hơn hoặc bằng 5

```
SELECT *
FROM MONHOC
WHERE SODVHT BETWEEN 3 AND 5
```

MAMONHOC	TENMONHOC	SODVHT
HO-001	Hoá đại cương	3
TI-001	Tin học đại cương	4
TI-002	Ngôn ngữ C	5
TI-003	Lý thuyết hệ điều hành	4
TI-004	Cấu trúc dữ liệu và giải thuật	4
TO-001	Đại số tuyến tính	4
TO-002	Giải tích 1	4
VL-001	Vật lý đại cương	3

Phép so sánh trong tập hợp [1]

- Phép toán IN (hoặc NOT IN) trong SQL được sử dụng để kiểm tra xem giá trị của một biểu thức có nằm trong (hoặc không nằm trong) một tập các giá trị được chỉ định hay không.
- Cú pháp sử dụng của phép toán này được minh họa như bảng sau:

Cú pháp	Biểu thức tương đương
x IN (a, b, c)	(x = a) OR (x = b) OR (x = c)
x NOT IN (a, b, c)	(x <> a) AND (x <> b) AND (x <> c)

- Chú ý: Tập hợp các giá trị được viết phân cách nhau bởi dấu phẩy và được đặt trong cặp dấu ngoặc tròn.

Phép so sánh trong tập hợp [2]

- Ví dụ:** Hiển thị mã, họ, tên và ngày sinh của các sinh viên sinh ở Huế, Quảng Trị hoặc Quảng Bình

```
SELECT MASV, HODEM, TEN, NGAYSINH
FROM SINHVIEN
WHERE (NOISINH = N'Huế')
OR (NOISINH = N'Quảng Trị')
OR (NOISINH = N'Quảng Bình')
```



```
SELECT MASV, HODEM, TEN, NGAYSINH
FROM SINHVIEN
WHERE NOISINH IN (N'Huế', 'Quảng Trị', 'Quảng Bình')
```

Phép so sánh mẫu [1]

- Phép so sánh mẫu sử dụng (LIKE) được sử dụng để kiểm tra một chuỗi giá trị có tuân theo một khuôn dạng/mẫu nhất định hay không?
- Mẫu/khuôn dạng là một chuỗi ký tự, kết hợp với các ký tự đại diện (wildcard) để mô tả khuôn dạng của dữ liệu.

Ví dụ: Một số điều kiện dưới đây minh họa cho điều kiện sử dụng phép so sánh mẫu.

- Họ của sinh viên bắt đầu với 2 ký tự *Ng* (ví dụ: *Nguyễn, Ngô,...*)
- Số điện thoại bắt đầu với 0234 và kết thúc bởi số 72 (ví dụ: *02343865272, 02343826272,...*)

Phép so sánh mẫu [2]

- Các ký tự đại diện:

Ký tự đại diện	Ý nghĩa	Ví dụ
%	Chuỗi ký tự bất kỳ có thể rỗng.	Tên sinh viên bắt đầu bởi chữ B (chẳng hạn: Ba, Bình, Bảy, B,...) TEN LIKE 'B%'
_ (dấu gạch chân)	Một ký tự đơn	Tên sinh viên gồm 2 ký tự, trong đó ký tự thứ 2 là chữ y (chẳng hạn: Ly, Ty, My,...) TEN LIKE '_y'

Phép so sánh mẫu [3]

- Các ký tự đại diện:

Ký tự đại diện	Ý nghĩa	Ví dụ
[]	Một ký tự đơn bất kỳ trong danh sách các ký tự được chỉ định.	Tên có 2 ký tự bắt đầu bởi A, B, C hoặc D: TEN LIKE '[ABCD]_' Hoặc: TEN LIKE '[A-D]_'
[^]	Một ký tự đơn bất kỳ không nằm trong danh sách ký tự được chỉ định	Tên bắt đầu bởi chữ Ch và tiếp theo đó không phải là a, u hay o: TEN LIKE 'Ch[^auo]%'

Phép so sánh mẫu [4]

- Ví dụ:** Hiển thị danh sách các sinh viên có họ là **Nguyễn** hoặc **Lê**

```
SELECT *
FROM SINHVIEN
WHERE (HODEM LIKE N'Nguyễn%')
OR (HODEM LIKE N'Lê%')
```

- Ví dụ:** Hiển thị danh sách các Khoa có số điện thoại bắt đầu bởi **054** và kết thúc bởi số **2, 4** hoặc **9**.

```
SELECT *
FROM KHOA
WHERE (DIENTHOAI LIKE '054%[249]')
```

Phép so sánh với giá trị NULL

- Bạn không thể sử dụng các phép toán so sánh số học (=, >, <, ...) để so sánh với giá trị NULL.
- SQL bắt buộc bạn sử dụng phép toán IS NULL (hoặc IS NOT NULL) để thực hiện việc so sánh một biểu thức với giá trị NULL, theo cú pháp:

```
WHERE biểu_thức IS NULL
hoặc:
WHERE biểu_thức IS NOT NULL
```

- Ví dụ:** Hiển thị danh sách các sinh viên chưa được phân lớp (lớp sinh viên học chưa được xác định)

```
SELECT *
FROM SINHVIEN
WHERE MALOP IS NULL
```

Sắp xếp kết quả truy vấn [1]

- Mệnh đề ORDER BY được sử dụng trong câu lệnh SELECT để yêu cầu SQL sắp xếp dữ liệu trong kết quả truy vấn, theo cú pháp

ORDER BY ds_cột_sử_dụng_để_sắp_xếp

- Mặc định, việc sắp xếp được thực hiện theo thứ tự tăng (ASC). Trong trường hợp cần sắp xếp theo thứ tự giảm, bạn phải chỉ định thêm từ khóa DESC sau cột được sử dụng để sắp xếp.

Sắp xếp kết quả truy vấn [2]

- Ví dụ:** Hiển thị danh sách sinh viên, sắp xếp theo thứ tự tăng dần của tên, tiếp đến là họ.

```
SELECT *
FROM SINHVIEN
ORDER BY TEN, HODEM
```

MASV	HODEM	TEN	NGAYSINH	GIOTINH	NOISINH	MALOP
0251020006	Hoàng Nhân	Ái	1981-05-11 00:00:00	1	Tuyên Hoá, Quảng Bình	C25102
0241020003	Võ Đức	Án	1982-05-24 00:00:00	1	Huế	C24102
0261020001	Bùi Tuấn	Ánh	1983-03-01 00:00:00	1	Bà nằng	C26102
0251020001	Dương Văn	Ánh	1983-01-24 00:00:00	1	Hương Phú - TTHuế	C25102
0261010001	Lê Hoàng Phương	Ánh	1984-03-04 00:00:00	0	Huế	C26101
0243010001	Lê Thị	Ánh	1980-10-12 00:00:00	0	Hương Trà, TTHuế	C24301
0261010002	Lê Thị Văn	Ánh	1984-10-14 00:00:00	0	Thanh Hoá	C26101
0261020002	Lê Thúc Quốc	Ánh	1984-12-04 00:00:00	1	TTHuế	C26102

Sắp xếp kết quả truy vấn [3]

- Có thể sử dụng biểu thức hoặc tiêu đề cột trong ORDER BY để chỉ định việc sắp xếp dữ liệu.

Ví dụ: Hiển thị mã, họ tên và tuổi sinh viên. Sắp xếp theo chiều giảm dần của tuổi.

```
SELECT MASV, HODEM, TEN,
       YEAR(GETDATE()) - YEAR(NGAYSINH) AS TUOI
FROM SINHVIEN
ORDER BY YEAR(GETDATE()) - YEAR(NGAYSINH) DESC
```

Hoặc:

```
SELECT MASV, HODEM, TEN,
       YEAR(GETDATE()) - YEAR(NGAYSINH) AS TUOI
FROM SINHVIEN
ORDER BY TUOI DESC
```


