

CHƯƠNG 3: XỬ LÝ THÔNG TIN TRÊN MÁY TÍNH: TRUY VẤN DỮ LIỆU

Khoa Khoa học và kỹ thuật thông tin
Bộ môn Thiết bị di động và Công nghệ Web

NỘI DUNG

1. Truy vấn SQL.
2. Xpath/Xquery.
3. Các dạng truy vấn select thường gặp.

SQL – truy vấn SQL

Giới thiệu

- SQL (Structured Query Language):
 - + Là ngôn ngữ cấp cao.
 - + Dùng để truy vấn dữ liệu trong CSDL quan hệ.
 - + Được IBM phát triển (1970s).
 - + Được gọi là SEQUEL.
 - + Được ANSI công nhận và phát triển thành chuẩn SQL-86, SQL-92, SQL-99.
- Đây là ngôn ngữ chuẩn dùng để truy vấn trong các CSDL quan hệ. Các CSDL quan hệ dù khác nhau về nền tảng và hãng sản xuất nhưng luôn có điểm chung là dùng SQL làm ngôn ngữ truy vấn.

Các nhóm lệnh

- Nhóm định nghĩa dữ liệu (DDL):
 - + Gồm các lệnh tạo, thay đổi cấu trúc các bảng dữ liệu (*Create, Drop, Alter*)
- Nhóm thao tác dữ liệu (DML):
 - + Gồm các lệnh làm thay đổi dữ liệu lưu trong bảng (*Insert, Delete, Update, Select*)
- Nhóm điều khiển dữ liệu (DCL):
 - + Gồm các lệnh quản lý quyền truy cập vào dữ liệu và các bảng (*Grant, Revoke, Deny*)

Ngôn ngữ định nghĩa dữ liệu

- **Lệnh tạo bảng (CREATE)**

 - Cú pháp

 - Một số kiểu dữ liệu

- **Lệnh sửa cấu trúc bảng (ALTER)**

 - Thêm thuộc tính

 - Sửa kiểu dữ liệu của thuộc tính

 - Xoá thuộc tính

 - Thêm ràng buộc toàn vẹn

 - Xoá ràng buộc toàn vẹn

- **Lệnh xóa bảng (DROP)**

Lệnh tạo bảng

Cú pháp

```
CREATE TABLE <tên_bảng>
(
    <tên_cột1>    <kiểu_dữ_liệu> [not null],
    <tên_cột2>    <kiểu_dữ_liệu> [not null],
    ...
    <tên_cột1>    <kiểu_dữ_liệu> [not null],
    khai báo khóa chính, khóa ngoại, ràng buộc
)
```

Lệnh tạo bảng (2)

Một số kiểu dữ liệu

Kiểu dữ liệu	SQL Server
Chuỗi ký tự	varchar(n), char(n), nvarchar(n), nchar(n)
Số	tinyint, smallint, int, numeric(m,n), decimal(m,n), float, real, smallmoney, money
Ngày tháng	smalldatetime, datetime
Luận lý	bit

Lệnh tạo bảng (3)

Lược đồ CSDL quản lý bán hàng gồm có các quan hệ sau:

KHACHHANG (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANHISO, NGDK, CMND)

NHANVIEN (MANV,HOTEN, NGVL, SODT)

SANPHAM (MASP,TENSP, DVT, NUOCSX, GIA)

HOADON (SOHD, NGHD, MAKH, MANV, TRIGIA)

CTHD (SOHD,MASP,SL)

Lệnh tạo bảng (4)

Create table KHACHHANG

```
(
    MAKH          char(4) primary key,
    HOTEN         varchar(40),
    DCHI          varchar(50),
    SODT          varchar(20),
    NGSINH        smalldatetime,
    DOANHISO      money,
    NGDK          smalldatetime,
    CMND          varchar(10)
)
```

Lệnh tạo bảng (5)

Create table CTHD

```
(
    SOHD    int    foreign key
            references HOADON(SOHD),
    MASP    char(4) foreign key
            references SANPHAM(MASP),
    SL      int,
    constraint PK_CTHD primary key (SOHD,MAASP)
)
```

Sửa cấu trúc bảng(1)

- Thêm thuộc tính

ALTER TABLE tênbảng ADD tên cột kiểu dữ liệu

+ Ví dụ: thêm cột Ghi_chu vào bảng khách hàng

ALTER TABLE KHACHHANG ADD GHI_CHU varchar(20)

- Sửa kiểu dữ liệu thuộc tính

*ALTER TABLE tênbảng ALTER COLUMN tên cột
kiểu dữ liệu_mới*

– Lưu ý:

Không phải sửa bất kỳ kiểu dữ liệu nào cũng được

Sửa cấu trúc bảng(2)

+ **Ví dụ:** Sửa Cột Ghi_chu thành kiểu dữ liệu varchar(50)

ALTER TABLE KHACHHANG ALTER COLUMN GHI_CHU varchar(50)

+ Nếu sửa kiểu dữ liệu của cột Ghi_chu thành varchar(5), mà trước đó đã nhập giá trị cho cột Ghi_chu có độ dài hơn 5 ký tự thì không được phép.

+ Hoặc sửa từ kiểu chuỗi ký tự sang kiểu số, ...

- Xóa thuộc tính

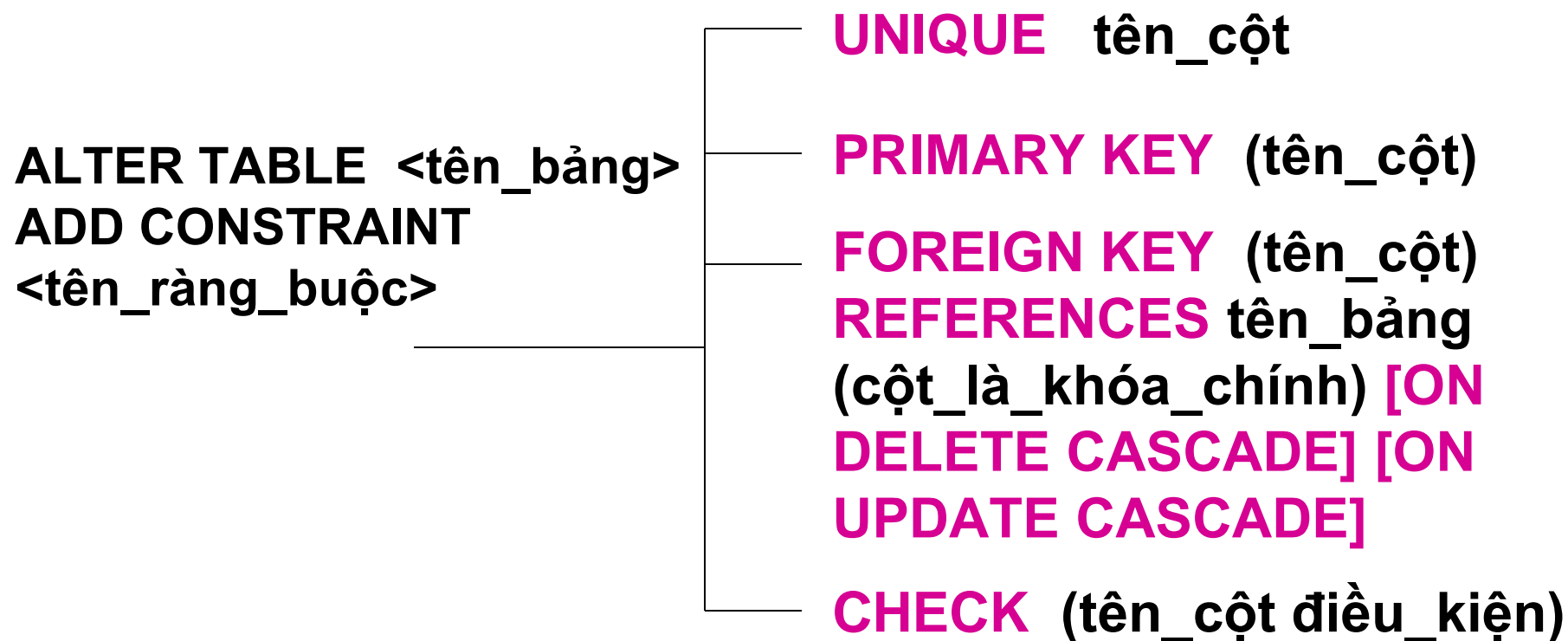
ALTER TABLE tên_bảng DROP COLUMN tên_cột

+ Ví dụ: xóa cột Ghi_chu trong bảng KHACHHANG

ALTER TABLE KHACHHANG DROP COLUMN Ghi_chu

Sửa cấu trúc bảng(3)

Thêm ràng buộc toàn vẹn



Sửa cấu trúc bảng(4)

Ví dụ

– Thêm ràng buộc khóa chính:

+ ALTER TABLE NHANVIEN ADD CONSTRAINT PK_NV PRIMARY KEY (MANV)

– Thêm ràng buộc khóa ngoại

+ ALTER TABLE CTHD ADD CONSTRAINT FK_CT_SP FOREIGN KEY (MASP)
REFERENCES SANPHAM(MASP)

Sửa cấu trúc bảng(5)

- Ví dụ
- Thêm ràng buộc check:
 - + ALTER TABLE **SANPHAM** ADD CONSTRAINT CK_GIA CHECK (GIA >=500)
- Thêm ràng buộc Unique:
 - + ALTER TABLE **KHACHHANG** ADD CONSTRAINT UQ_KH UNIQUE (CMND)

Sửa cấu trúc bảng(6)

- Xóa ràng buộc toàn vẹn

ALTER TABLE tên_bảng DROP CONSTRAINT tên_ràng_buộc

+ Ví dụ:

- Alter table CTHD drop constraint FK_CT_SP
- Alter table SANPHAM drop constraint ck_gia

— **Lưu ý:** đối với ràng buộc khóa chính, muốn xóa ràng buộc này phải xóa hết các ràng buộc khóa ngoại tham chiếu tới nó.

Lệnh xóa bảng

- **Cú pháp**

`DROP TABLE tên_bảng`

- **Ví dụ:** xóa bảng KHACHHANG.

`DROP TABLE KHACHHANG`

- **Lưu ý:** khi muốn xóa một bảng phải xóa tất cả những khóa ngoại tham chiếu tới bảng đó trước.

Ngôn ngữ thao tác dữ liệu

- Lệnh thêm dữ liệu (INSERT)
- Lệnh sửa dữ liệu (UPDATE)
- Lệnh xóa dữ liệu (DELETE)

Thêm dữ liệu

– Cú pháp

INSERT INTO tên_bảng (cột1,...,cộtn) VALUES (giá_trị_1,..., giá_trị_n)

INSERT INTO tên_bảng VALUES (giá_trị_1, giá_trị_2,..., giá_trị_n)

– Ví dụ:

+ insert into SANPHAM values('BC01','But chi', 'cay', 'Singapore', 3000)

+ insert into SANPHAM(masp,tensp,dvt,nuocsx,gia) values ('BC01','But chi','cay','Singapore',3000)

Sửa dữ liệu

– Cú pháp

```
UPDATE tên_bảng
SET cột_1 = giá_trị_1, cột_2 = giá_trị_2 ....
[WHERE điều_kiện]
```

– **Lưu ý:** cẩn thận với các lệnh xóa và sửa, nếu không có điều kiện ở WHERE nghĩa là xóa hoặc sửa tất cả.

– **Ví dụ:** Tăng giá 10% đối với những sản phẩm do “Trung Quoc” sản xuất

```
UPDATE SANPHAM
SET Gia = Gia*1.1
WHERE Nuocsx='Trung Quoc'
```

Sửa dữ liệu

NHANVIEN

MANV	HOTEN	DTHOAI	NGVL
NV01	Nguyen Nhu Nhut	0927345678	13/4/2006
NV02	Le Thi Phi Yen	0987567390	21/4/2006
NV03	Nguyen Van B	0997047382	27/4/2006
NV04	Ngo Thanh Tuan	0913758498	24/6/2006
NV05	Nguyen Thi Truc Thanh	0918590387	20/7/2006

NHANVIEN

MANV	HOTEN	DTHOAI	NGVL
NV01	Nguyen Nhu Nhut	0927345678	13/4/2006
NV02	Le Thi Phi Yen	0987567390	21/4/2006
NV03	Nguyen Van B	0989999999	27/4/2006
NV04	Ngo Thanh Tuan	0913758498	24/6/2006
NV05	Nguyen Thi Truc Thanh	0918590387	20/7/2006

UPDATE NHANVIEN
SET dthoai = '0989999999'
WHERE manv = 'NV03'

Sửa dữ liệu

NHANVIEN

MANV	HOTEN	DTHOAI
NV01	Nguyen Nhu Nhut	0927345678
NV02	Le Thi Phi Yen	0987567390
NV03	Nguyen Van B	0997047382
NV04	Ngo Thanh Tuan	0913758498
NV05	Nguyen Thi Truc Thanh	0918590387

NHANVIEN

MANV	HOTEN	DTHOAI
NV01	Nguyen Nhu Nhut	0927345678
NV02	Le Thi Phi Yen	0987567390
NV03	Nguyen Van Minh	0989999999
NV04	Ngo Thanh Tuan	0913758498
NV05	Nguyen Thi Truc Thanh	0918590387

UPDATE NHANVIEN

SET hoten='Nguyen Van Minh', dthoai = '0989999999'

WHERE manv = 'NV03'

Sửa dữ liệu

NHANVIEN

MANV	HOTEN	DTHOAI
NV01	Nguyen Nhu Nhut	0927345678
NV02	Le Thi Phi Yen	0987567390
NV03	Nguyen Van B	0997047382
NV04	Ngo Thanh Tuan	0913758498
NV05	Nguyen Thi Truc Thanh	0918590387

NHANVIEN

MANV	HOTEN	DTHOAI
NV01	Nguyen Nhu Nhut	0989999999
NV02	Le Thi Phi Yen	0989999999
NV03	Nguyen Van Minh	0989999999
NV04	Ngo Thanh Tuan	0989999999
NV05	Nguyen Thi Truc Thanh	0989999999

UPDATE NHANVIEN
SET dthoai = '0989999999'

Xóa dữ liệu

– Cú pháp

DELETE FROM tên_bảng [WHERE điều_kiện]

– Ví dụ:

- + Xóa toàn bộ nhân viên

DELETE FROM NHANVIEN

- + Xóa những sản phẩm do Trung Quốc sản xuất có giá thấp hơn 10000

DELETE FROM SANPHAM

WHERE (Gia < 10000) and (Nuocsx = 'Trung Quoc')

Xóa dữ liệu

NHANVIEN

MANV	HOTEN	DTHOAI	LUONG
NV01	Nguyen Nhu Nhut	0927345678	2.800.000
NV02	Le Thi Phi Yen	0987567390	2.000.000
NV03	Nguyen Van B	0997047382	2.300.000
NV04	Ngo Thanh Tuan	0913758498	1.800.000
NV05	Nguyen Thi Truc Thanh	0918590387	2.500.000

NHANVIEN

MANV	HOTEN	DTHOAI	LUONG
NV01	Nguyen Nhu Nhut	0927345678	2.800.000
NV03	Nguyen Van B	0997047382	2.300.000
NV04	Ngo Thanh Tuan	0913758498	1.800.000
NV05	Nguyen Thi Truc Thanh	0918590387	2.500.000

DELETE FROM nhanvien
WHERE manv = 'NV02'

Xóa dữ liệu

NHANVIEN

MANV	HOTEN	DTHOAI	LUONG
NV01	Nguyen Nhu Nhut	0927345678	2.800.000
NV02	Le Thi Phi Yen	0987567390	2.000.000
NV03	Nguyen Van B	0997047382	2.500.000
NV04	Ngo Thanh Tuan	0913758498	1.800.000
NV05	Nguyen Thi Truc Thanh	0918590387	2.500.000

NHANVIEN

DELETE FROM nhanvien
WHERE luong=2.500.000



MANV	HOTEN	DTHOAI	LUONG
NV01	Nguyen Nhu Nhut	0927345678	2.800.000
NV02	Le Thi Phi Yen	0987567390	2.000.000
NV04	Ngo Thanh Tuan	0913758498	1.800.000

Xóa dữ liệu

NHANVIEN

MANV	HOTEN	DTHOAI	LUONG
NV01	Nguyen Nhu Nhut	0927345678	2.800.000
NV02	Le Thi Phi Yen	0987567390	2.000.000
NV03	Nguyen Van B	0997047382	2.500.000
NV04	Ngo Thanh Tuan	0913758498	1.800.000
NV05	Nguyen Thi Truc Thanh	0918590387	2.500.000

NHANVIEN

MANV	HOTEN	DTHOAI	LUONG
NV02	Le Thi Phi Yen	0987567390	2.000.000
NV04	Ngo Thanh Tuan	0913758498	1.800.000

DELETE FROM nhanvien
WHERE luong>2.000.000

Một số hàm toán học và xử lý chuỗi ký tự, ngày, tháng, năm

— Hàm toán học:

+ Hàm ABS: trả về giá trị tuyệt đối của một số.

- `ABS(-1234.56)`

+ Hàm PI() trả về số pi trong toán học.

- `PI()` Kết quả: 3.14159265358979

+ Hàm POWER trả về phép tính lũy thừa

- `POWER(3,2)`. Kết quả là: 9

Một số hàm toán học và xử lý chuỗi ký tự, ngày, tháng, năm

— Hàm toán học:

+ Hàm ROUND trả về số được làm tròn lên

- $\text{ROUND}(123.4567, 2) = 123.46$

- $\text{ROUND}(123.4567, -1) = 120$

+ Hàm SQRT trả về kết quả căn bậc hai

- $\text{SQRT}(9) = 3$

Một số hàm toán học và xử lý chuỗi ký tự, ngày, tháng, năm

— Hàm toán học:

- + Hàm FLOOR trả về số nguyên nhỏ hơn hoặc bằng số được làm tròn: $\text{FLOOR}(12.3) = 12$
- + Hàm CEILING trả về số nguyên lớn hơn hoặc bằng giá trị làm tròn: $\text{CEILING}(23.45) = 24$

Một số hàm toán học và xử lý chuỗi ký tự, ngày, tháng, năm

— Hàm xử lý chuỗi ký tự:

+ Hàm UPPER: trả về chữ in hoa

- `UPPER('Hello') = 'HELLO'`

+ Hàm LOWER: trả về chữ in thường

- `LOWER ('HEllo') = 'hello'`

+ Hàm LEN: trả về số ký tự trong chuỗi

- `LEN ('HEllo') = 5`

Một số hàm toán học và xử lý chuỗi ký tự, ngày, tháng, năm

— Hàm xử lý chuỗi ký tự:

+ Hàm LEFT, RIGHT, SUBSTRING: hàm cắt chuỗi bên trái, phải, giữa

- LEFT ('Hello world', 3) = 'HEL'
- RIGHT ('Hello world', 5) = 'world'
- SUBSTRING ('Hello world', 3, 2) = 'll'

+ Hàm LTRIM, RTRIM: loại bỏ khoảng trắng bên trái, bên phải.

Một số hàm toán học và xử lý chuỗi ký tự, ngày, tháng, năm

— Hàm xử lý ngày tháng năm

+ Hàm GETDATE() trả về ngày tháng hiện tại.

+ Hàm DATEPART(datepart, date):

- date là ngày truyền vào để lấy các phần tương ứng với tham số datepart.
- datepart: đại diện một phần của tham số date:
 - yy, yyyy: năm của date
 - q, qq quý của date
 - mm, m: tháng của date

Một số hàm toán học và xử lý chuỗi ký tự, ngày, tháng, năm

- Hàm xử lý ngày tháng năm
 - + Hàm DAY(date): trả về ngày của tham số date.
 - + Hàm MONTH (date): trả về tháng của tham số date.
 - + Hàm YEAR(date): trả về năm của tham số date.

Ngôn ngữ truy vấn dữ liệu có cấu trúc

1. Toán tử truy vấn
2. Câu truy vấn tổng quát
3. Truy vấn đơn giản
4. Đặt bí danh, sử dụng *, distinct
5. Phép kết
6. Hàm tính toán, gom nhóm
7. Truy vấn lồng
8. Phép toán tập hợp

1. Toán tử truy vấn

- Toán tử so sánh: =, >, <, >=, <=, <>
- Toán tử logic: AND, OR, NOT
- Phép toán: +, -, *, /
- BETWEEN AND
- IS NULL, IS NOT NULL
- LIKE (_ %)
- IN, NOT IN
- EXISTS , NOT EXISTS
- SOME, ALL

2. Câu truy vấn tổng quát

SELECT [DISTINCT] *|tên_cột | hàm
FROM bảng
[WHERE điều_kiện]
[GROUP BY tên_cột]
[HAVING điều_kiện]
[ORDER BY tên_cột ASC | DESC]

3. Truy vấn đơn giản(1)

— SELECT

- + Tương đương phép chiếu của ĐSQH
- + Liệt kê các thuộc tính cần hiển thị trong kết quả

— WHERE

- + Tương ứng với điều kiện chọn trong ĐSQH
- + Điều kiện liên quan tới thuộc tính, sử dụng các phép nối luận lý AND, OR, NOT, các phép toán so sánh, BETWEEN

— FROM

- + Liệt kê các quan hệ cần thiết, các phép kết

3. Truy vấn đơn giản(2)

– SANPHAM (MASP,TENS, DVT, NUOCSX, GIA)

– Tìm masp, tensp do “Trung Quoc” sản xuất

Select masp,tensp

From SANPHAM

Where nuocsx=‘Trung Quoc’

3. Truy vấn đơn giản(3)

– SANPHAM (MASP,TENS, DVT, NUOCSX, GIA)

– Tìm masp, tensp do “Trung Quoc” sản xuất có giá từ 2000 đến 3000

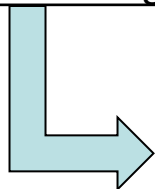
```
Select      masp,tensp
From        SANPHAM
Where       nuocsx='Trung Quoc'
           and gia >= 2000 and gia<= 3000
```

3. Truy vấn đơn giản(4)

– Sử dụng between

SANPHAM

masp	tensp	dvt	nuocsx	gia
BB03	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	2500
ST04	So tay	quyen	Thai Lan	55000
ST05	So tay mong	quyen	Thai Lan	20000



masp	tensp	dvt	nuocsx	gia
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	2500

Select *
from SANPHAM
where gia between 2000 and 3000

3. Truy vấn đơn giản(5)

Toán tử LIKE

- + So sánh chuỗi tương đối
- + Cú pháp: s LIKE p, p có thể chứa % hoặc _
- + % : thay thế một chuỗi ký tự bất kỳ
- + _ : thay thế một ký tự bất kỳ

3. Truy vấn đơn giản(6)

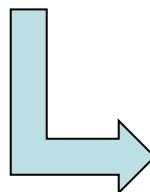
♦ Toán tử LIKE

- ♦ Hiển thị sản phẩm có masp bắt đầu là B, kết thúc là 1

SANPHAM

masp	tensp	dvt	nuocsx	gia
BB21	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	2500
ST04	So tay	quyen	Thai Lan	55000
ST01	So tay mong	quyen	Thai Lan	20000

Select masp, tensp
From SANPHAM
Where masp like 'B%1'



masp	tensp
BB21	But bi
BC01	But chi

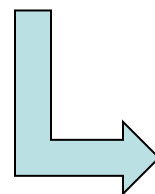
3. Truy vấn đơn giản(6)

♦ Toán tử LIKE

- ♦ Select masp,tensp from SANPHAM where masp like 'B_01'

SANPHAM

masp	tensp	dvt	nuocsx	gia
BB21	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	2500
ST04	So tay	quyen	Thai Lan	55000
ST01	So tay mong	quyen	Thai Lan	20000



masp	tensp
BC01	But chi

3. Truy vấn đơn giản(7)

– IS NULL, IS NOT NULL

- + Select * from HOADON where makh is Null
- + Select SOHD from HOADON where makh is Not Null

HOADON

SOHD	NGHD	MAKH	TRIGIA
1017	02/01/2007	KH08	35,000
1018	13/01/2007	KH08	330,000
1019	13/01/2007	KH01	30,000
1020	14/01/2007	KH09	70,000
1021	16/01/2007	KH10	67,500
1022	16/01/2007	Null	7,000
1023	17/01/2007	Null	330,000



SOHD	NGHD	MAKH	TRIGIA
1022	16/01/2007	Null	7,000
1023	17/01/2007	Null	330,000

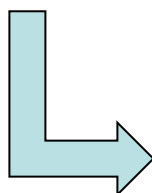
Select * from HOADON
where makh is Null

3. Truy vấn đơn giản(8)

– Toán tử IN, NOT IN

SANPHAM

masp	tensp	dvt	nuocsx	gia
BB03	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	2500
ST04	So tay	quyen	Thai Lan	55000
ST05	So tay mong	quyen	Thai Lan	20000



masp	tensp	dvt	nuocsx	gia
BC01	But chi	cay	Singapore	3000
ST04	So tay	quyen	Thai Lan	55000
ST05	So tay mong	quyen	Thai Lan	20000

Select *

From SANPHAM

Where masp **NOT IN** ('BB01','BC02','BB03')

4. Đặt bí danh (1)

- Đặt bí danh – Alias: cho thuộc tính và quan hệ: `tên_cũ AS tên_mới` (hoặc `tên_cũ tên_mới`)

```
Select manv, hoten as [ho va ten]
From NHANVIEN
```

NHANVIEN

manv	hoten	dthoai
NV01	Nguyen Nhu Nhut	0927345678
NV02	Le Thi Phi Yen	0987567390
NV03	Nguyen Van B	0997047382
NV04	Ngo Thanh Tuan	0913758498
NV05	Nguyen Thi Truc Thanh	0918590387



manv	ho va ten
NV01	Nguyen Nhu Nhut
NV02	Le Thi Phi Yen
NV03	Nguyen Van B
NV04	Ngo Thanh Tuan
NV05	Nguyen Thi Truc Thanh

4. Đặt bí danh (2)

```
Select  nv.manv,  nv.hoten  [ho va ten]
From    NHANVIEN  nv
```

NHANVIEN

manv	hoten	dthoai
NV01	Nguyen Nhu Nhut	0927345678
NV02	Le Thi Phi Yen	0987567390
NV03	Nguyen Van B	0997047382
NV04	Ngo Thanh Tuan	0913758498
NV05	Nguyen Thi Truc Thanh	0918590387



manv	ho va ten
NV01	Nguyen Nhu Nhut
NV02	Le Thi Phi Yen
NV03	Nguyen Van B
NV04	Ngo Thanh Tuan
NV05	Nguyen Thi Truc Thanh

4. Đặt bí danh (3)

Select nuocsx, gia*1.1 as [gia ban]
from SANPHAM
where nuocsx<>'Thai Lan'

SANPHAM

masp	tensp	dvt	nuocsx	gia
BB03	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	5000
ST04	So tay	quyen	Thai Lan	55000
ST05	So tay mong	quyen	Thai Lan	20000



nuocsx	gia ban
Singapore	3300
Singapore	5500

4. Sử dụng * (3)

— Liệt kê tất cả các thuộc tính của quan hệ:

- + Select * from Nhanvien where luong > 2400000
- + Select NHANVIEN.* from NHANVIEN where luong > 2400000
- + Select hv.* from NHANVIEN hv where luong > 2400000

NHANVIEN

MANV	HOTEN	LUONG
NV01	Nguyen Nhu Nhut	2.800.000
NV02	Le Thi Phi Yen	2.000.000
NV03	Nguyen Van B	2.500.000
NV04	Ngo Thanh Tuan	1.800.000
NV05	Nguyen Thi Truc Thanh	2.500.000



NHANVIEN

MANV	HOTEN	LUONG
NV01	Nguyen Nhu Nhut	2.800.000
NV03	Nguyen Van B	2.500.000
NV05	Nguyen Thi Truc Thanh	2.500.000

4. Sử dụng distinct (4)

- Distinct: trùng chỉ lấy một lần
- + `SELECT DISTINCT lương FROM nhanvien`

NHANVIEN

MANV	HOTEN	LUONG
NV01	Nguyen Nhu Nhut	2.800.000
NV02	Le Thi Phi Yen	2.000.000
NV03	Nguyen Van B	2.500.000
NV04	Ngo Thanh Tuan	2.800.000
NV05	Nguyen Thi Truc Thanh	2.500.000



LUONG
2.800.000
2.000.000
2.500.000

4. Sử dụng Order by (5)

- Order by: sắp xếp kết quả hiển thị
 - + Sắp xếp tăng dần: ASC
 - + Sắp xếp giảm dần: DESC
 - + `Select * from NHANVIEN order by LUONG DESC`

NHANVIEN

MANV	HOTEN	LUONG
NV01	Nguyen Nhu Nhut	2.800.000
NV02	Le Thi Phi Yen	2.000.000
NV03	Nguyen Van B	2.500.000
NV04	Ngo Thanh Tuan	2.800.000
NV05	Nguyen Thi Truc Thanh	2.500.000

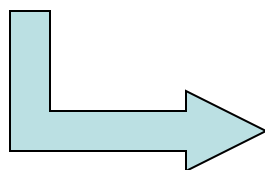


MANV	HOTEN	LUONG
NV01	Nguyen Nhu Nhut	2.800.000
NV04	Ngo Thanh Tuan	2.800.000
NV03	Nguyen Van B	2.500.000
NV05	Nguyen Thi Truc Thanh	2.500.000
NV02	Le Thi Phi Yen	2.000.000

4. Sử dụng Order by (6)

SANPHAM

masp	tensp	dvt	nuocsx	gia
BB03	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	5000
ST04	So tay	quyen	Thai Lan	55000
ST05	So tay mong	quyen	Thai Lan	20000



```
SELECT *
FROM sanpham
ORDER BY nuocsx DESC, gia ASC
```

masp	tensp	dvt	nuocsx	gia
ST05	So tay mong	quyen	Thai Lan	20000
ST04	So tay	quyen	Thai Lan	55000
BB03	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	5000

5. Phép kết

- Phép ϕ kết, kết tự nhiên
 - + Mệnh đề WHERE chỉ ra điều kiện kết giữa các thuộc tính của các bảng
 - + Hoặc dùng từ khóa **Inner Join** (hoặc Join) trong mệnh đề **FROM**.
- Phép kết trái, phải, ngoài
 - + Dùng Half Outer Join (**Left join**, **Right Join**), **Full Outer Join** trong mệnh đề FROM

5. Phép kết: Inner Join (1)

MONHOC	
Mamon	Tenmh
CSDL	Co so du lieu
CTRR	Cau truc roi rac
THDC	Tin hoc dai cuong

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0
HV01	THDC	7.0
HV02	THDC	0
HV03	THDC	7.5
HV03	CSDL	6.0

In ra mã học viên, tên môn học và điểm thi của học viên

```
SELECT mahv, tenmh, diem
FROM monhoc INNER JOIN ketquathi
ON mamon =mamh
```

Lưu ý: có thể thay inner join bằng join

Cách khác

```
SELECT mahv, tenmh, diem
FROM monhoc , ketquathi
WHERE mamon =mamh
```


5. Phép kết: Inner Join (2)

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le
HV04	Le Minh

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0

In ra mã học viên, tên học viên, mã môn học và điểm thi của học viên

```
SELECT hocvien.mahv, hoten, mamh, diem
FROM   hocvien JOIN ketquathi
      ON  hocvien.mahv=ketquathi.mahv
```

Đặt lại tên cho quan hệ hocvien và ketquathi:

```
SELECT hv.mahv, hoten, mamh, diem
FROM   hocvien hv JOIN ketquathi kq
      ON  hv.mahv=kq.mahv
```

Cách khác:

```
SELECT hv.mahv, hoten, mamh, diem
FROM   hocvien hv, ketquathi kq
WHERE  hv.mahv=kq.mahv
```

5. Phép kết: Inner Join (3)

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le
HV04	Le Minh

In ra mã học viên, tên học viên, mã môn học và điểm thi của học viên có mã số HV01

```
SELECT hv.mahv, hoten, mamh, diem
FROM   hocvien hv JOIN   ketquathi kq
      ON hv.mahv=kq.mahv
WHERE  hv.mahv = 'HV01'
```

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0

Cách khác:

```
SELECT hv.mahv, hoten, mamh, diem
FROM   hocvien hv,   ketquathi kq
WHERE  hv.mahv=kq.mahv
      AND hv.mahv = 'HV01'
```

5. Phép kết: Inner Join (4)

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le
HV04	Le Minh

Tìm những học viên có điểm thi môn CSDL nhỏ hơn 8

```
SELECT hv.mahv, hoten -- Hoặc hv.*
FROM hocvien hv JOIN ketquathi kq
ON hv.mahv=kq.mahv
WHERE kq.mamh='CSDL' AND diem<8
```

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0

Cách khác:

```
SELECT hv.mahv, hoten -- Hoặc hv.*
FROM hocvien hv, ketquathi kq
WHERE hv.mahv=kq.mahv
AND kq.mamh='CSDL' AND
diem<8
```

5. Phép kết: Inner Join (5)

— Ví dụ:

+ In ra danh sách các khách hàng (MAKH, HOTEN) đã mua hàng trong ngày 1/1/2007.

```
select KHACHHANG.makh,hoten
from   KHACHHANG inner join HOADON on
       KHACHHANG.makh=HOADON.makh
where  nghd='1/1/2007'
```

5. Phép kết: left join (6)

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le
HV04	Le Minh

In ra danh sách tất cả các học viên và mã môn học, điểm thi của học viên đó (nếu có)

```
SELECT hv.mahv, hoten, mamh, diem
FROM hocvien hv LEFT JOIN ketquathi
kq
ON hv.mahv=kq.mahv
```

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0



Mahv	hoten	Mamh	Diem
HV01	Nguyen Van Lan	CSDL	7.0
HV01	Nguyen Van Lan	CTRR	8.5
HV02	Tran Hong Son	CSDL	8.5
HV03	Nguyen Le	CTRR	9.0
HV04	Le Minh	NULL	NULL

5. Phép kết: left join (7)

— **Ví dụ:** In ra danh sách tất cả các hóa đơn và họ tên của khách hàng mua hóa đơn đó (nếu có)

```
+ Select  sohd, hoten
  From    HOADON left join KHACHHANG on
         HOADON.makh=KHACHHANG.makh

+ Select  sohd, hoten
  From    HOADON ,KHACHHANG
  where   HOADON.makh*=KHACHHANG.makh
```

5. Phép kết: right join (8)

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le
HV04	Le Minh

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0

In ra danh sách tất cả các học viên và mã môn học, điểm thi của học viên đó (nếu có)

```
SELECT  mamh,    diem, hv.mahv,    hoten
FROM    ketquathi kq RIGHT JOIN  hocvien
hv
ON      hv.mahv=kq.mahv
```



mamh	diem	mahv	hoten
CSDL	7.0	HV01	Nguyen Van Lan
CTRR	8.5	HV01	Nguyen Van Lan
CSDL	8.5	HV02	Tran Hong Son
CTRR	9.0	HV03	Nguyen Le
NULL	NULL	HV04	Le Minh

Phép kết trên nhiều bảng

NHANVIEN (HONV, TENLOT, TENNV, MANV, NGSINH, DCHI, PHAI, LUONG, MA_NQL, PHG)

PHONGBAN (TENPHG, MAPHG, TRPHG, NG_NC)

DIADIEM_PHG (MAPHG, DIADIEM)

THANNHAN (MA_NVIEN, TENTN, PHAI, NGSINH, QUANHE)

DEAN (TENDA, MADA, DDIEM_DA, PHONG)

PHANCONG (MA_NVIEN, SODA, THOIGIAN)

Phép kết trên nhiều bảng

- 1. In danh sách mã số, họ tên nhân viên và tên thân nhân của nhân viên đó.*
- 2. In danh sách mã số, họ tên của tất cả các nhân viên và tên thân nhân của nhân viên đó (nếu có).*

TRUY VẤN LỒNG

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức.
- Các câu truy vấn con thường trả về một tập các giá trị.
- Các câu truy vấn con được kết hợp bằng phép nối logic với câu truy vấn cha.

In và Not In

+ Cú pháp:

<thuộc tính> (NOT) IN (<truy vấn con>)

+ Lưu ý:

- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

Ví dụ:

3. *Tìm nhân viên chưa được phân công thực hiện đề án nào*

```
SELECT manv, honv+'`'+tenlot+'`'+tennv as hoten
FROM Nhanvien
Where manv NOT IN
                (SELECT distinct ma_nvien FROM Phan cong)
```

Any/Some và All

— Cú pháp:

<thuộc tính> <phép so sánh> Any/Some/All (<truy vấn con>)

— Lưu ý:

- + Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha
- + Any/Some: so sánh với bất kỳ giá trị nào đó trong tập hợp.
- + All: so sánh với tất cả các giá trị trong tập hợp.

VÍ DỤ

Vd 4

```
SELECT manv, honv+'`'+tenlot+'`'+tennv as hoten
FROM Nhanvien
Where luong >= ALL
                (SELECT luong FROM Nhanvien)
```

Vd 5

```
SELECT manv, honv+'`'+tenlot+'`'+tennv as hoten
FROM Nhanvien
Where phai= 'Nu' AND luong >ANY
                (SELECT luong FROM Nhanvien Where phai='Nam')
```

Vd 6

```
SELECT nv1.manv, nv1.honv+'`'+nv1.tenlot+'`'+nv1.tennv as hoten
FROM Nhanvien nv1, Nhanvien nv2
Wheres nv1.phai='Nu' AND nv1.luong>nv2.luong AND nv2.phai='Nam'
```

Exists và Not Exists

— Cú pháp:

(NOT) EXISTS (<câu truy vấn con>)

— Lưu ý:

- + Không cần có thuộc tính, hằng số hay biểu thức nào khác đứng trước
- + Không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con
- + Những câu truy vấn có điều kiện “= ANY” hay “IN” đều có thể chuyển thành câu truy vấn có EXISTS

VÍ DỤ

Vd 7

```
SELECT manv, honv+'`'+tenlot+'`'+tennv as hoten
FROM Nhanvien
Where manv NOT IN
                (SELECT distinct ma_nvien FROM Phancong)
```

Vd 8

```
SELECT manv, honv+'`'+tenlot+'`'+tennv as hoten
FROM Nhanvien
Where manv <> ALL
                (SELECT distinct ma_nvien FROM Phancong)
```

Vd 9

```
SELECT manv, honv+'`'+tenlot+'`'+tennv as hoten
FROM Nhanvien
Where NOT EXISTS
                (SELECT * FROM Phancong WHERE manv=ma_nvien )
```

Truy vấn lòng

In hoặc Exists

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le
HV04	Le Minh

KETQUATHI		
Mahvien	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0

Tìm những học viên thi được 8.5 điểm

```
SELECT *
FROM   hocvien
WHERE  mahv IN (SELECT mahvien
                FROM   ketquathi
                WHERE  diem=8.5)
```

```
SELECT *
FROM   hocvien
WHERE  EXISTS      (SELECT
mahvien
                    FROM   ketquathi
                    WHERE
mahvien=mahv
                    AND diem=8.5)
```


Truy vấn lồng

In hoặc Exists

– **Ví dụ:** Tìm các số hóa đơn (sohd) mua cùng lúc 2 sản phẩm có mã số “BB01” và “BB02”.

- + select distinct sohd
from CTHD where masp='BB01' and **sohd IN**
(select sohd from CTHD where masp='BB02')
- + select distinct A.sohd
from CTHD A where A.masp='BB01' and
EXISTS (select * from CTHD B
where B.masp='BB02' and A.sohd=B.sohd)

Truy vấn lồng

Not In hoặc Not Exists

— Ví dụ: Tìm các số hóa đơn (sohd) có mua sản phẩm mã số 'BB01' nhưng không mua sản phẩm mã số 'BB02'.

- + select distinct sohd
from CTHD where masp='BB01' and sohd **NOT IN**
(select sohd from CTHD where masp='BB02')
- + select distinct A.sohd
from CTHD A where A.masp='BB01' and
NOT EXISTS (select * from CTHD B
where B.masp='BB02' and A.sohd=B.sohd)

Nhận xét IN và EXISTS

— IN

- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

— EXISTS

- Không cần có thuộc tính, hằng số hay biểu thức nào khác đứng trước
- Không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con
- Những câu truy vấn có =ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS

Truy vấn lòng

Tìm học viên (Mahv) có điểm thi cao nhất

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.5
HV03	CTRR	9.0

```
SELECT Mahv
FROM ketquathi
WHERE diem >= ALL (SELECT diem
                   FROM ketquathi)
```

Hoặc có thể viết như sau

```
SELECT Mahv
FROM ketquathi
WHERE diem = (SELECT MAX(diem)
             FROM ketquathi)
```

Truy vấn lòng

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le
HV04	Le Minh

KETQUATHI		
Mahvien	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.5
HV01	CTRR	8.5
HV03	CTRR	9.0

Tìm những học viên thi được 8.5 điểm

```
SELECT *
FROM   hocvien
WHERE  mahv = ANY      (SELECT
                        FROM ketquathi
                        WHERE diem=8.5)
```

=ANY tương đương với toán tử IN
Cách khác: sử dụng phép kết, ...

Một số dạng truy vấn khác

- Câu truy vấn con không chỉ xuất hiện ở mệnh đề WHERE mà có thể xuất hiện ở những nơi khác (SELECT, FROM, HAVING,...)
- Kết quả trả về của câu truy vấn con
 - + Là một bảng trung gian trong quá trình truy vấn
 - + Bảng này không có lưu trữ thật sự

VÍ DỤ

- Vd: Cho biết số lượng nhân viên Nam (phai='Nam'), nhân viên Nữ (phai='Nu') trong từng phòng ban.

```
SELECT MAPHG, TENPHG, NAM.SLNV, NU.SLNV
FROM PHONGBAN,
    (SELECT PHG, COUNT(*) AS SLNV
     FROM NHANVIEN
     WHERE PHAI='Nam'
     GROUP BY PHG) AS NAM,
    (SELECT PHG, COUNT(*) AS SLNV
     FROM NHANVIEN
     WHERE PHAI='Nu'
     GROUP BY PHG) AS NU
WHERE MAPHG=NAM.PHG AND MAPHG=NU.PHG
```

VÍ DỤ

— Vd: Cho biết số lượng nhân viên Nam (phai='Nam'), nhân viên Nữ (phai='Nu') trong từng phòng ban.

```
SELECT MAPHG, TENPHG,
      (SELECT COUNT(*) FROM NHANVIEN
       WHERE PHAI='Nam' AND PHG=MAPHG) AS SLNAM,
      (SELECT COUNT(*) FROM NHANVIEN
       WHERE PHAI='Nu' AND PHG=MAPHG) AS SLNU
FROM PHONGBAN
```


Phép toán tập hợp

— SQL có cài đặt các phép toán

+ Hội (UNION).

+ Giao (INTERSECT).

+ Trừ (EXCEPT).

— Kết quả trả về là tập hợp

+ Loại bỏ các bộ trùng nhau.

+ Để giữ lại các bộ trùng nhau:

- UNION ALL
- INTERSECT ALL
- EXCEPT ALL

Phép toán tập hợp

— Cú pháp:

SELECT <các thuộc tính> FROM <các bảng> WHERE <các điều kiện>
UNION [ALL]
SELECT <các thuộc tính> FROM <các bảng> WHERE <các điều kiện>

SELECT < các thuộc tính> FROM <các bảng> WHERE <các điều kiện>
INTERSECT [ALL]
SELECT <các thuộc tính> FROM <các bảng> WHERE <các điều kiện>

SELECT <các thuộc tính> FROM <các bảng> WHERE <các điều kiện>
EXCEPT [ALL]
SELECT <các thuộc tính> FROM <các bảng> WHERE <các điều kiện>

Phép toán tập hợp

Phép hội (union).

NHANVIEN

MANV	HOTEN	LUONG
NV001	Nguyễn Ngọc Linh	2.800.000
NV002	Đinh Bá Tiến	2.000.000
NV003	Nguyễn Văn Mạnh	2.300.000
NV004	Trần Thanh Long	1.800.000
NV005	Nguyễn Thị Hồng Vân	2.500.000
NV009	Nguyễn Kim Anh	2.000.000

PHONGBAN

MAPH	TENPH	TrgPH
NC	Nghiên cứu	NV05
DH	Điều hành	NV02
TC	Tài chính	NULL

Hiển thị những nhân viên (manv) có lương lớn hơn 2.500.000 hoặc là trưởng phòng của phòng của phòng 'DH'

```
SELECT manv
FROM nhanvien
WHERE luong>2500000
UNION
SELECT trgPH
FROM phongban
WHERE maph='DH'
```

Phép toán tập hợp Phép trừ (EXCEPT).

CTHD

SOHD	MASP	SL
1001	BB01	10
1001	BC04	20
1002	BB01	20
1002	BB02	20
1003	BB03	10
1004	TV01	20
1004	BB01	5

Hiển thị những hóa đơn (sohd)
mua BB01 nhưng không mua
BB02

```
SELECT sohd
FROM cthd
WHERE masp='BB01'
EXCEPT
SELECT sohd
FROM cthd
WHERE masp='BB02'
```

8 Phép toán tập hợp (3)

Phép giao (INTERSECT).

CTHD

SOHD	MASP	SL
1001	BB01	10
1001	BC04	20
1002	BB01	20
1002	BB02	20
1003	BB03	10
1004	TV01	20
1004	BB01	5

Hiển thị những hóa đơn (sohd)
vừa mua BB01 vừa mua BB02

```
SELECT sohd
FROM cthd
WHERE masp='BB01'
INTERSECT
SELECT sohd
FROM cthd
WHERE masp='BB02'
```

Phép toán tập hợp

- Tuy nhiên, chúng ta có thể sử dụng **IN, NOT IN, EXISTS, NOT EXISTS, ...** để thực hiện các phép toán hội, giao trừ trên tập hợp.
- Đối với phép chia: sử dụng **NOT EXISTS**.
- Ví dụ:
 1. Tìm họ tên nhân viên được phân công thực hiện tất cả các đề án.
 2. Tìm tên các đề án được phân công cho tất cả các nhân viên thuộc phòng số 5 thực hiện

Các phép toán trên tập hợp

Vd 1

```
SELECT manv, honv, tenlot, tennv
FROM Nhanvien
Where NOT EXISTS
      (SELECT * FROM Dean WHERE NOT EXISTS
      (SELECT * FROM Phancong WHERE manv=ma_nvien
      AND mada=soda))
```

```
SELECT tenda
FROM Dean
Where NOT EXISTS
      (SELECT * FROM Nhanvien WHERE phg=5 AND NOT EXISTS
      (SELECT * FROM Phancong WHERE manv=ma_nvien
      AND mada=soda))
```

Vd 2

Hàm tính toán, gom nhóm

- Top N
- Hàm tính toán
- Gom nhóm

Top N

- Trả về N dòng kết quả đầu tiên của câu truy vấn
- Cú pháp: **TOP N**
với N là số nguyên dương
- Nên sử dụng ORDER BY để sắp xếp kết quả
- VD: Tìm mức lương cao nhất:

```
SELECT TOP 1 lương as CN
FROM Nhanvien
ORDER BY lương DESC
```

Top N

- SELECT TOP N [PERCENT] [WITH TIES]
FROM bang [WHERE dieu_kien]
[ORDER BY bieu_thuc [ASC | DESC]];
- Ví dụ TOP 10 sẽ chèn 10 hàng đầu tiên từ bộ kết quả.
- **PERCENT** : Ví dụ như TOP 10 PERCENT sẽ trả về 10% giá trị đầu trong bộ kết quả.
- **WITH TIES** : Các hàng có giá trị giống với hàng cuối cùng trong bộ kết quả sẽ được trả về. Điều này có thể gây ra tình huống số hàng trả về nhiều hơn biến TOP cho phép.

Hàm tính toán, kết hợp

Các hàm tính toán cơ bản

- + COUNT: Đếm số bộ dữ liệu của thuộc tính
- + MIN: Tính giá trị nhỏ nhất
- + MAX: Tính giá trị lớn nhất
- + AVG: Tính giá trị trung bình
- + SUM: Tính tổng giá trị các bộ dữ liệu

Hàm tính toán, kết hợp

- MIN, MAX, SUM, AVG: cú pháp: **TÊN HÀM(<tên thuộc tính>)**
- COUNT
 - ✓ COUNT(*) đếm số dòng
 - ✓ COUNT(<tên thuộc tính>) đếm số dòng thuộc tính có giá trị khác NULL
 - ✓ COUNT(DISTINCT <tên thuộc tính>) đếm số dòng thuộc tính có giá trị khác nhau và khác NULL

Hàm tính toán, kết hợp

Hiển thị điểm thi cao nhất

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0

```
SELECT MAX(Diem) Max_D
FROM ketquathi
```



Max_D
9.0

Hàm tính toán, kết hợp

Hiển thị điểm thi cao nhất, thấp nhất, trung bình

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0



max_D	min_D	avg_D
9.0	7	8.25

```
SELECT MAX(Diem) max_D, MIN(Diem) min_D, AVG(Diem) avg_D
FROM ketquathi
```

Hàm tính toán, kết hợp

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	8.0
HV01	CTRR	8.0
HV03	CTRR	9.0

Hiển thị điểm thi cao nhất của môn
CSDL



max_D
8.0

```
SELECT MAX(Diem) max_D
FROM ketquathi
WHERE mamh = 'CSDL'
```

Hàm tính toán, kết hợp

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0

Có bao nhiêu học viên thi môn
CSDL đạt điểm lớn hơn 5



SL
2

```
SELECT COUNT(Mahv) SL
FROM ketquathi
WHERE mamh= 'CSDL' AND diem>5
```


Gom nhóm

— Cú pháp:

SELECT <các thuộc tính>
FROM <các bảng>
[WHERE <các điều kiện>
GROUP BY <các thuộc tính gom nhóm>
[HAVING <các điều kiện>

— Trong đó:

- Điều kiện ở WHERE thực hiện **trước** khi gom nhóm
- Điều kiện ở HAVING thực hiện **sau** khi gom nhóm
- Các thuộc tính sau GROUP BY dùng để gom nhóm và phải có đầy đủ các thuộc tính sau SELECT (trừ những thuộc tính trong những hàm kết hợp)

Gom nhóm

Gom nhóm: mệnh đề GROUP BY

- Sử dụng hàm gom nhóm trên các bộ trong quan hệ.
- Mỗi nhóm bộ bao gồm tập hợp các bộ có cùng giá trị trên các thuộc tính gom nhóm
- Hàm gom nhóm áp dụng trên mỗi bộ độc lập nhau.
- SQL có mệnh đề GROUP BY để chỉ ra các thuộc tính gom nhóm, các thuộc tính này phải xuất hiện trong mệnh đề SELECT

Gom nhóm

Hiển thị điểm thi cao nhất của từng học viên

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0



mahv	max_D
HV01	8.0
HV02	9.0
HV03	9.0

```
SELECT mahv, MAX(Diem) max_D
FROM ketquathi
GROUP BY mahv
```

Gom nhóm

Hiển thị điểm thi cao nhất, thấp nhất của từng học viên

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0



mahv	max_D	min_D
HV01	8.0	7.0
HV02	9.0	9.0
HV03	9.0	3.0

```
SELECT mahv, MAX(Diem) max_D, MIN (Diem) min_D
FROM ketquathi
GROUP BY mahv
```

Gom nhóm

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le

Hiển thị điểm thi cao nhất của từng học viên (mahv, hoten, max_D)

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0



mahv	Hoten	max_D
HV01	Nguyen Van Lan	8.0
HV02	Tran Hong Son	9.0
HV03	Nguyen Le	9.0

```
SELECT hv.mahv, hoten, MAX(Diem) max_D
FROM hocvien hv, ketquathi kq
WHERE hv.mahv = kq.mahv
GROUP BY hv.mahv, hoten
```

Ghi chú: Nếu không gom nhóm theo họ tên mà select theo họ tên thì câu lệnh sẽ gây ra lỗi.

Gom nhóm

HOCVIEN	
Mahv	Hoten
HV01	Tran Hong Son
HV02	Tran Hong Son
HV03	Nguyen Le

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0

Hiển thị điểm thi cao nhất của từng học viên (mahv, hoten, max_D). (Cách khác)



mahv	Hoten	max_D
HV01	Tran Hong Son	8.0
HV02	Tran Hong Son	9.0
HV03	Nguyen Le	9.0

```
SELECT h.mahv, hoten, k.max_D
FROM hocvien h JOIN (SELECT mahv, MAX(Diem) max_D
                     FROM ketquathi
                     GROUP BY mahv) k
ON h.mahv = k.mahv
```

Gom nhóm

HOCVIEN	
Mahv	Hoten
HV01	Tran Hong Son
HV02	Tran Hong Son
HV03	Nguyen Le

Hiển thị điểm thi cao nhất của từng học viên (hoten, max_D)

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0



Hoten	max_D
Tran Hong Son	8.0
Tran Hong Son	9.0
Nguyen Le	9.0

```
SELECT hoten, MAX(Diem) max_D
FROM hocvien hv, ketquathi kq
WHERE hv.mahv = kq.mahv
GROUP BY hv.mahv, hoten
```

Điều kiện trên nhóm dữ liệu - HAVING

- Điều kiện sau gom nhóm: mệnh đề HAVING

- + Lọc kết quả theo điều kiện, sau khi đã gom nhóm
 - + Điều kiện ở HAVING được thực hiện sau khi gom nhóm, các điều kiện có liên quan đến thuộc tính Group By
- Ví dụ: tìm phòng có số lượng nhân viên “Nữ” trên 5 người

SELECT	phong
FROM	NhanVien
WHERE	phai = 'Nữ'
GROUP BY	phong
HAVING	count(manv) > 5

Điều kiện trên nhóm dữ liệu - HAVING

Hiển thị những học viên có điểm thi cao nhất lớn hơn 8.5

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	8.0
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0



mahv	MAX(Diem)
HV01	8.0
HV02	9.0
HV03	9.0



mahv	max_D
HV02	9.0
HV03	9.0

```
SELECT mahv, MAX(Diem) max_D
FROM ketquathi
GROUP BY mahv
HAVING MAX(Diem) >8.5
```

Điều kiện trên nhóm dữ liệu - HAVING

HOCVIEN	
Mahv	Hoten
HV01	Nguyen Van Lan
HV02	Tran Hong Son
HV03	Nguyen Le

Những học viên nào có điểm trung bình >6
(mahv, hoten, diemTB)



Mahv	Hoten	diemTB
HV01	Nguyen Van Lan	7.0
HV02	Tran Hong Son	9.0

KETQUATHI		
Mahv	Mamh	Diem
HV01	CSDL	7.0
HV02	CSDL	9.0
HV01	CTRR	7.5
HV03	CTRR	9.0
HV03	CSDL	3.0
HV01	THDC	8.0

```
SELECT hv.mahv, hoten, AVG(Diem) diemTB
FROM hocvien hv, ketquathi kq
WHERE hv.mahv = kq.mahv
GROUP BY hv.mahv, hoten
HAVING AVG(Diem)>6
```

Ví dụ

NHANVIEN					
MANV	HOTEN	PHAI	MANQL	PHONG	LUONG
NV001	Nguyễn Ngọc Linh	Nữ	Null	NC	2.800.000
NV002	Đinh Bá Tiến	Nam	NV002	DH	2.000.000
NV003	Nguyễn Văn Mạnh	Nam	NV001	NC	2.300.000
NV004	Trần Thanh Long	Nam	NV002	DH	1.800.000
NV005	Nguyễn Thị Hồng Vân	Nữ	NV001	NC	2.500.000
NV006	Nguyễn Minh	Nam	NV002	DH	2.000.000
NV007	Hà Duy Lập	Nam	NV003	NC	1.800.000
NV008	Trần Kim Duyên	Nữ	NV003	NC	1.800.000
NV009	Nguyễn Kim Anh	Nữ	NV003	NC	2.000.000

Ví dụ

1. Tính lương thấp nhất, cao nhất, trung bình và tổng lương của tất cả các nhân viên.
2. Có tất cả bao nhiêu nhân viên.
3. Bao nhiêu nhân viên có người quản lý.
4. Có tất cả bao nhiêu người quản lý.
5. Có bao nhiêu nhân viên không có người quản lý trực tiếp.
6. Bao nhiêu phòng ban có nhân viên trực thuộc.
7. Tính lương trung bình của tất cả các nhân viên.
8. Tính lương trung bình của các nhân viên theo từng phòng ban.

Bài tập

NHANVIEN

MANV	HOTEN	PHAI	MANQL	PHONG	LUONG
NV001	Nguyễn Ngọc Linh	Nữ	Null	NC	2.800.000
NV002	Đinh Bá Tiến	Nam	NV002	DH	2.000.000
NV003	Nguyễn Văn Mạnh	Nam	NV001	NC	2.300.000
NV004	Trần Thanh Long	Nam	NV002	DH	1.800.000
NV005	Nguyễn Thị Hồng Vân	Nữ	NV001	NC	2.500.000
NV009	Nguyễn Kim Anh	Nữ	NV003	TC	2.000.000

PHONGBAN

MAPH	TENPH	TrgPh
NC	Nghiên cứu	NV005
DH	Điều hành	NV002
TC	Tài chính	NULL

NHANVIEN-DEAN

MaNV

MaDA

ThoiGian

DEAN

MaDA

TenDA

DiaDiemDA

MaPB

Bài tập

1. Hiển thị maph, tenph, trgh, hoten (họ tên trưởng phòng).
2. Phòng ban NC (maph) có bao nhiêu nhân viên.
3. Phòng “Nghiên cứu” có bao nhiêu nhân viên.
4. Hiển thị những phòng ban (maph) có số nhân viên nhiều hơn 4.
5. Hiển thị những phòng ban (maph, tenph) có số nhân viên nữ lớn hơn 2.
6. Tìm số lượng nhân viên Nam (phai=‘Nam’) của từng phòng ban.
7. Tìm phòng ban có từ 2 nhân viên Nam (phai=‘Nam’) trở lên.
8. Tìm phòng ban có đông nhân viên nhất.
9. Tìm đề án có ít nhân viên Nữ (phai=‘Nu’) tham gia nhất.
10. Tìm 3 nhân viên thuộc phòng số 4 (phong=NC) có lương thấp nhất.

Xpath/Xquery

Giới thiệu

— Xpath và Xquery:

- + Là hai ngôn ngữ có rất nhiều mặt giống nhau, hỗ trợ tìm kiếm thông tin trong tài liệu XML.
- + Có thể xem Xpath là tập hợp con của Xquery.
- + Xquery sử dụng Xpath như là một ngôn ngữ chính để định hướng tìm kiếm thay vì dùng đệ quy để duyệt cây.

Xpath

- XML Path language (gọi tắt là Xpath) là một chuẩn để xử lý tài liệu XML (cũng như SQL là một chuẩn để làm việc với csdl).
- Dùng để xử lý nhiều kiểu truy vấn trong tài liệu XML và các biến thể của nó (như HTML).
- Là ngôn ngữ rất phổ biến.
- Tiết kiệm thời gian trích xuất dữ liệu.

Ví dụ 36: CSDL XML

```

1. <users>
2.   <user>
3.     <name>
4.       <first>Lola</first>
5.       <last>Solis</last>
6.     </name>
7.     <age>2</age>
8.   </user>
9.   <user>
10.    <name>
11.      <first>Nina</first>
12.      <last>Serafina</last>
13.    </name>
14.    <age>4</age>

```

```

15. <visits>
16.   <first>2008-01-15</first>
17.   <last>2008-02-15</last>
18. </visits>
19. </user>
20. <user>
21.   <name>
22.     <first>Tracy</first>
23.     <last>Keller</last>
24.   </name>
25.   <age>35</age>
26. </user>
27. </users>

```

Ví dụ

— Đoạn Xpath sau: tìm tên của người dùng dưới 18 tuổi

1. `/user[age lt 18]/name/last/text()`
- 2.
3. (: Result
4. Solis
5. Serafina
6. :)

— Nếu không dùng Xpath sẽ gặp chút khó khăn khi xử lý việc loại bỏ giá trị trong node visits.

Nhận xét về Xpath

- Biểu thức Xpath ngắn gọn, rõ ràng.
- Hiểu được các node phức tạp trong tài liệu XML và biết các mối quan hệ giữa chúng.
- Một hạn chế của Xpath là *không cung cấp cách chuyển đổi tập kết quả trả về*.
 - + Ở ví dụ trên không thể sắp kết quả hiện thị tăng dần theo tên.

XQuery

- Phức tạp hơn so với Xpath.
- Sử dụng cú pháp pha trộn XML và Xpath.
- Khắc phục được nhược điểm của Xpath:
 - + Sắp xếp kết quả của câu truy vấn hoặc chuyển chúng thành HTML, CSV, SQL, XML ...
 - + Cung cấp tính năng biểu thức FLWOR.
 - + Sử dụng hàm và đệ quy.
 - + Diễn tả các phép nối.

Biểu thức FLOWR

- Biểu thức FLWOR dùng để liên kết các tiêu chí rút trích dữ liệu và chuyển đổi tập kết quả trả về của câu truy vấn.
- FLWOR là viết tắt của các từ **for**, **let**, **where**, **order by** và **return**.
- Bắt đầu bằng một biểu thức *for* hoặc *let* và kết thúc bằng một biểu thức *return*.

Ví dụ 1

— Tìm tên của người dùng dưới 18 tuổi, có sắp xếp kết quả tăng dần.

1. let \$xml:= _XML from **Vidu36**
2. for \$user in \$xml//user[age lt 18]
3. order by \$user/name/last
4. return \$user/name/last/text()

5. (: Result
6. Serafina
7. Solis
8. :)

Ví dụ 2

— Kết quả truy vấn trả về 1 đoạn HTML, danh sách có đánh số thứ tự.

```

1. let $xml:= _XML from Vidu36
2. return
3.     <ol>{
4.         for $user in $xml//user[age lt 18]
5.         order by $user/name/last
6.         return <li>{$user/name/last/text()}</li>
7.     }</ol>
8. (: Result
9.     <ol><li>Serafina</li><li>Solis</li></ol>
10.:)

```


Sử dụng hàm và đệ quy

- Xquery cung cấp các hàm, các phép toán được xây dựng sẵn và cho phép định nghĩa các hàm riêng.
- Xquery cũng hỗ trợ đệ quy: tiện lợi khi làm việc với XML (có thể chứa các node lồng nhau tùy ý).

Ví dụ 4

- Định nghĩa hàm transform-names dùng để thay đổi tên các node trong bất kỳ tài liệu XML nào.

Ví dụ 4 – xquery (1)

1. (: Part 1 :)
2. **define function transform-names(\$node as node()) as node() {**
3. **element{replace(name(\$node), "_", "-")} {**
4. **\$node/text(), for \$subnode in \$node/* return transform-names(\$subnode)**
5. **}**
6. **}**

Ví dụ 4 – xquery (2)

```

7. (: Part 2 :)
8. let $xml:=
9.   <item>
10.    <item_type>book</item_type>
11.    <contributors>
12.     <author>
13.      <first_name>Charles</first_name>
14.      <last_name>Edward</last_name>
15.      <home_address>
16.       <home_street>206 S. Solomon St.</home_street>
17.       <home_city>New Orleans</home_city>
18.       <home_state>LA</home_state>
19.       <home_zip>70119</home_zip>
20.      </home_address>
21.     </author>
22.     <artist>
23.      <last_name>Salinas</last_name>
24.     </artist>
25.    </contributors>
26.  </item>
27. return transform-names($xml)

```

Ví dụ 4 – xquery (3)

```

28. (: Result
29.   <item>
30.     <item-type>book</item-type>
31.     <contributors>

32.       <author>
33.         <first-name>Charles</first-name>
34.         <last-name>Edward</last-name>
35.         <home-address>
36.           <home-street>206 S. Solomon St.</home-street>
37.           <home-city>New Orleans</home-city>
38.           <home-state>LA</home-state>
39.           <home-zip>70119</home-zip>
40.         </home-address>
41.       </author>
42.       <artist>
43.         <last-name>Salinas</last-name>
44.       </artist>
45.     </contributors>
46.   </item>
47. :)
```

MỘT SỐ DẠNG TRUY VẤN SELECT THƯỜNG GẶP

CSDL QUẢN LÝ BÁN HÀNG

KHACHHANG (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANHSONG, NGDK)

NHANVIEN (MANV,HOTEN, NGVL, SODT)

SANPHAM (MASP,TENSP, DVT, NUOCSX, GIA)

HOADON (SOHD, NGHD, MAKH, MANV, TRIGIA)

CTHD (SOHD,MASP,SL)

TRUY VẤN SELECT

— Cú pháp câu truy vấn **SELECT**:

SELECT <cột 1>, <cột 2>,

FROM <tên bảng>

WHERE <điều kiện>

ORDER BY <tên cột> **ASC** | **DESC**

GROUP BY <tên cột 1>, <tên cột 2>,

HAVING <điều kiện>

— Lưu ý:

+ Mệnh đề **HAVING** sử dụng cho các **hàm gom nhóm**.

+ **ASC** – sắp xếp tăng dần; **DESC** – sắp xếp giảm dần.

TRUY VẤN SELECT – DẠNG 1

– **Dạng 1:** Truy vấn lấy dữ liệu tất cả.

```
SELECT * FROM <tên bảng>
```

hoặc

```
SELECT <danh sách cột> FROM <tên bảng>
```

VD:

```
SELECT * FROM KHACHHANG
```

```
SELECT MAKH, HOTEN, DCHI FROM KHACHHANG
```

TRUY VẤN SELECT – DẠNG 2

– **Dạng 2:** Truy vấn dữ liệu có điều kiện:

```
SELECT <danh sách cột> FROM <tên bảng>  
WHERE <điều kiện>
```

VD: Lấy thông tin MAKH, HOTEN DCHI của khách hàng có doanh số trên 10000

```
SELECT MAKH, HOTEN, DCHI FROM KHACHHANG  
WHERE DOANHSCO > 10000
```

TRUY VẤN SELECT – DẠNG 3

– **Dạng 3:** Truy vấn dữ liệu có kết bảng.

SELECT <danh sách cột> FROM <tên bảng 1>

INNER JOIN <tên bảng 2> ON <tên bảng 1>.<mã khoá ngoại> = <tên bảng 2>.<mã khoá chính>

[WHERE <điều kiện>]

– Các phép kết:

+ **INNER JOIN:** kết bằng.

+ **LEFT OUTER JOIN:** kết mở rộng về bên trái.

+ **RIGHT OUTER JOIN:** kết mở rộng về bên phải.

VD

VD:

```
SELECT MAKH, HOTEN  
FROM HOADON INNER JOIN KHACHHANG ON KHACHHANG.MAKH  
= HOADON.MAKH  
WHERE NGHD = '16/7/2019'
```

TRUY VẤN SELECT – DẠNG 4

– Dạng 4: Truy vấn dữ liệu có sắp xếp

SELECT <danh sách tên cột> FROM <tên bảng>

[WHERE <điều kiện>]

ORDER BY <danh sách cột cần sắp xếp> ASC hoặc DESC

VD: Sắp xếp khách hàng theo ngày sinh giảm dần

SELECT MAKH, HOTEN, NGSINH FROM KHACHHANG

ORDER BY NGSINH DESC

TRUY VẤN SELECT – DẠNG 5

- **Dạng 5:** Truy vấn sử dụng các hàm gom nhóm

SELECT <các hàm gom nhóm> FROM <tên bảng>

[WHERE <điều kiện>]

GROUP BY <tên cột 1>, <tên cột 2>, ...

- Các hàm gom nhóm: *COUNT()*, *AVG()*, *MAX()*, *MIN()*, *SUM()*.

- **Lưu ý:** Các thuộc tính trong mệnh đề SELECT (trừ các hàm kết hợp), phải xuất hiện trong mệnh đề **GROUP BY**.

VD

VD: Tính giá trị trung bình doanh số theo từng MAKH đối với khách hàng có doanh số trên 10000

```
SELECT MAKH, AVG(DOANHSON) FROM KHACHHANG
WHERE DOANHSON > 10000
GROUP BY MAKH
```

TRUY VẤN SELECT – DẠNG 6

– **Dạng 6:** Truy vấn sử dụng hội - giao - trừ

```
SELECT <danhsách cột 1> FROM <tên bảng>  
[WHERE <điều kiện 1>]
```

UNION (hội) | INTERSECT (giao) | EXCEPT (trừ)

```
SELECT <danhsách cột 2> FROM <tên bảng>  
[WHERE <điều kiện 2>]
```

Lưu ý: Để sử dụng các phép hội giao trừ thì 2 quan hệ phải **khả hợp**,
tức *<danhsách cột 1> = <danhsách cột 2>*

VD

VD1: Tìm khách hàng mua hoá đơn HD01 hoặc HD02

```
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
```

```
WHERE MAHD = 'HD01'
```

UNION

```
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
```

```
WHERE MAHD = 'HD02'
```

VD

VD2: Tìm khách hàng mua cùng lúc hoá đơn HD01 và HD02

```
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
```

```
WHERE MAHD = 'HD01'
```

INTERSECT

```
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
```

```
WHERE MAHD = 'HD02'
```

VD

VD3: Tìm khách hàng không mua hoá đơn nào

```
SELECT MAKH, HOTEN FROM KHACHHANG
```

```
EXCEPT
```

```
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN  
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
```

TRUY VẤN SELECT – DẠNG 7

– **Dạng 7:** Truy vấn lồng:

SELECT <danh sách cột> FROM <tên bảng>

WHERE <so sánh tập hợp> (

 SELECT <danh sách cột> FROM <tên bảng>

 WHERE <điều kiện>

)

<so sánh tập hợp>: *ALL, IN, NOT IN, ALL, ANY, EXISTS, NOT EXISTS.*

VD

VD: Tìm thông tin mã hoá đơn có trị giá cao nhất

```
SELECT SOHD FROM HOADON
```

```
WHERE TRIGIA = (
```

```
    SELECT MAX(TRIGIA) FROM HOADON
```

```
    GROUP BY SOHD
```

```
)
```

TRUY VẤN SELECT – DẠNG 8

– **Dạng 8:** Truy vấn lồng tương quan

```
SELECT <danh sách cột> FROM <tên bảng> AS OB1
```

```
WHERE <so sánh tập hợp> (
```

```
    SELECT <danh sách cột> FROM <tên bảng> AS OB2
```

```
    WHERE OB1.<tên cột> = OB2.<tên cột>
```

```
)
```

<so sánh tập hợp>: *ALL, IN, NOT IN, ALL, ANY, EXISTS, NOT EXISTS.*

VD

VD: Tìm sản phẩm có giá cao nhất theo từng nước sản xuất.

```
SELECT NUOCSX FROM SANPHAM AS SP1
```

```
WHERE GIA = (
```

```
    SELECT MAX(GIA) FROM SANPHAM AS SP2
```

```
    WHERE SP1.MASP = SP2.MASP
```

```
)
```

TRUY VẤN SELECT – DẠNG 9

– **Dạng 9:** Truy vấn dùng bảng “con” (inner aggregate).

```
SELECT <danh sách cột 1> FROM (  
    SELECT <danh sách cột 2> FROM <tên bảng>  
    WHERE <điều kiện>  
) AS <tên bảng con>
```

Lưu ý: <danh sách cột 1> phụ thuộc vào <danh sách cột 2> trả về từ câu truy vấn con (subquery).

VD

VD: Tìm khách hàng có số lần mua hàng nhiều nhất.

```
SELECT MAKH FROM HOADON
GROUP BY MAKH
HAVING COUNT(SOHD) = (
    SELECT MAX(SL_HD) FROM (
        SELECT MAKH, COUNT(SOHD) AS SL_HD
        FROM HOADON
        GROUP BY MAKH) AS T
)
```

Lưu ý: Không thể sử dụng dạng *COUNT (MAX (SOHD))* trong SQL Server

TRUY VẤN SELECT – DẠNG 10

– Dạng 10: Phép chia

Tìm *<đối tượng 1>* đã ... tất cả *<đối tượng 2>*

Cần xác định:

Đối tượng 1 (MaDT1,).

Đối tượng 2 (MaDT2,).

Quan hệ Đối tượng 1 và đối tượng 2 (*MaDT1*, *MaDT2*,).

VD

```

SELECT <danhsách cột> FROM <tên bảng đối tượng 1>
AS OB1 WHERE NOT EXISTS (
    SELECT <danhsách cột> FROM <tên bảng đối
    tượng 2> AS OB2 WHERE AND NOT EXISTS (
        SELECT * FROM <tên bảng quan hệ đối tượng
        1 và 2> as OB3 WHERE OB2.<khoá chính> = OB3.<khoá
        ngoại> and OB3.<khoá ngoại> = OB1.<khoá chính>
    )
)

```

VD

— VD: Tìm hoá đơn đã mua tất cả sản phẩm xuất xứ Thái Lan:

Đối tượng 1: **HOADON**(SOHD, NGHD, ...)

Đối tượng 2: **SANPHAM**(MASP, TENSP, XUATXU)

Quan hệ giữa 2 đối tượng: **CTHD**(MASP, SOHD, SL).

```
SELECT SOHD FROM HOADON AS T1 WHERE NOT EXISTS (
    SELECT MASP FROM SANPHAM AS T2 WHERE XUATXU =
    "Thái Lan" AND NOT EXISTS (
        SELECT MASP, SOHD FROM CTHD AS T3 WHERE
        T2.MASP = T3.MASP AND T1.SOHD = T3.SOHD)
    )
```

Tổng kết

1. Để truy vấn dữ liệu trong CSDL quan hệ, ta dùng ngôn ngữ SQL.
2. SQL có 3 nhóm lệnh chính: DDL, DML và DCL.
3. Các truy vấn nâng cao trên SQL: kết (nhiều bảng), gom nhóm, truy vấn lồng.
4. Khung nhìn View là một bảng ảo được tạo ra dựa trên bảng vật lý, dùng để đảm bảo an ninh dữ liệu.
5. Xpath/Xquery là ngôn ngữ dùng để truy vấn CSDL XML.
6. Xquery là phiên bản mở rộng của Xpath, hỗ trợ các hàm, đệ quy, ... mà bản thân Xpath không làm được.

TÀI LIỆU THAM KHẢO

1. Nguyễn Gia Tuấn Anh, Trương Châu Long, *Bài tập và bài giải SQL Server*, NXB Thanh niên (2005).
2. Đỗ Phúc, Nguyễn Đăng Ty, *Cơ sở dữ liệu*, NXB Đại học quốc gia TP HCM (2010).
3. Nguyễn Gia Tuấn Anh, Mai Văn Cường, Bùi Danh Hương, *Cơ sở dữ liệu nâng cao*, NXB Đại học quốc gia TP HCM (2019).
4. Itzik Ben-Gan, *Microsoft SQL Server 2012- TSQL Fundamentals*.

