

Chương 5

Ngôn ngữ truy vấn SQL

Nội dung trình bày

- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Giới thiệu

- SQL (Structured Query Language)
 - Ngôn ngữ cấp cao.
 - 1970, phát triển bởi IBM.
 - Được chuẩn hóa bởi ANSI và ISO
 - SQL-86.
 - SQL-92.
 - SQL-99.
- Gồm
 - Ngôn ngữ định nghĩa dữ liệu.
 - Ngôn ngữ thao tác dữ liệu.
 - Ngôn ngữ định nghĩa khung nhìn.
 - Ngôn ngữ phân quyền và bảo mật.

Định nghĩa dữ liệu

- Ngôn ngữ
 - Mô tả lược đồ cho các quan hệ.
 - Mô tả miền giá trị cho các thuộc tính.
 - Mô tả ràng buộc toàn vẹn.
 - Chỉ mục trên mỗi quan hệ.
- Gồm các lệnh
 - CREATE / DROP DATABASE
 - CREATE / DROP / ALTER TABLE
 - CREATE / DROP DOMAIN

Kiểu dữ liệu (1)

- Số
 - Số nguyên
 - INTEGER (INT)
 - SMALL INTEGER (SMALLINT)
 - Số thực
 - FLOAT[n]
 - REAL
 - DOUBLE PRECISION
 - Số thập phân
 - DECIMAL(p,n)
-

Kiểu dữ liệu (2)

- Chuỗi ký tự
 - Chuỗi có độ dài cố định
 - CHARACTER(n) (CHAR(n))
 - Chuỗi có độ dài thay đổi
 - CHARACTER VARYING(n) (VARCHAR(n))
 - Chuỗi ký tự unicode
 - NATIONAL CHARACTER(n)
 - NATIONAL CHARACTER VARYING (n)
 - Giá trị mặc định n = 1.
 - Chuỗi Bit
 - BIT(n)
 - BIT VARYING(n)
 - Ngày, giờ
 - DATE
 - TIME
 - TIMESTAMP
-

Tạo và hủy CSDL

- CREATE DATABASE <tên csdl>
 - create database CsdICongtty
 - DROP DATABASE <tên csdl> [RESTRICT | CASCADE]
 - drop database CsdICongtty
-

Tạo bảng (1)

- CREATE TABLE <tên bảng> (
 - <tên thuộc tính> <kiểu dữ liệu> [<RBTV>],
 - <tên thuộc tính> <kiểu dữ liệu> [<RBTV>],
 - ...
 - [<RBTV>])
 - RBTV
 - NOT NULL
 - NULL
 - UNIQUE
 - DEFAULT
 - PRIMARY KEY
 - FOREIGN KEY / REFERENCES
 - CHECK
-

Tạo bảng (2)

- Ví dụ

```
create table NHANVIEN (  
    Ho          varchar(15)    not null,  
    Dem          varchar(20),  
    Ten          varchar(15)    not null,  
    MaNV         char(9)        not null,  
    Ngsinh       date,  
    Dchi         varchar(30),  
    GTinh        char,  
    Luong        decimal(10,2),  
    MaGSat       char(9),  
    MaPhong      int            not null  
)
```

Tạo bảng (3)

- Đặt tên cho RBTV

- CONSTRAINT <tên RBTV> <RBTV>

- Ví dụ

```
create table PHONGBAN (  
    Ten          varchar(15)    unique,  
    MaPB         int            not null,  
    TrPhong      varchar(9)     not null,  
    NgNhanChuc   date,  
    constraint PB_PK primary key (MaPB),  
    constraint PB_TrPh foreign key (TrPhong) references  
        NHANVIEN (MaNV) on delete set null on update cascade  
)
```

Xóa bảng

- DROP TABLE <tên bảng> [RESTRICT | CASCADE]

- DROP TABLE <tên bảng>

- Ví dụ

```
drop table PHONGBAN
```

Sửa bảng (1)

- ALTER TABLE <tên bảng> ADD COLUMN <tên thuộc tính> <kiểu dữ liệu> [<RBTV>]

- Ví dụ

```
alter table NHANVIEN add column DThoi char(10)
```

- ALTER TABLE <tên bảng> DROP COLUMN <tên thuộc tính>

- Ví dụ

```
alter table NHANVIEN drop column DThoi
```

- ALTER TABLE <tên bảng> ALTER COLUMN <tên thuộc tính> <kiểu dữ liệu mới>

- Ví dụ

```
alter table NHANVIEN alter column GTinh int
```

Sửa bảng (2)

- ALTER TABLE <tên bảng> ADD CONSTRAINT <tên RBTV> <RBTV>
 - Ví dụ
alter table NHANVIEN add constraint NV_Luong check (Luong > 0)
- ALTER TABLE <tên bảng> DROP CONSTRAINT <tên RBTV>
 - Ví dụ
alter table NHANVIEN drop constraint NV_Luong

Tạo và xóa miền giá trị

- CREATE DOMAIN <tên kiểu dữ liệu mới> AS <kiểu dữ liệu cơ bản>
 - Ví dụ
create domain Diachi as varchar(100)
- DROP DOMAIN <tên kiểu dữ liệu>

Cập nhật dữ liệu (1)

- Chèn dữ liệu
 - Chèn từng dòng
 - INSERT INTO <tên bảng> [(<danh sách thuộc tính>)] VALUES (<danh sách giá trị>)
 - Ví dụ
+ insert into DUAN (TenDA, MaDA, Diadiem, Phong) values ('San pham X',1,'Tan Binh',5)
+ insert into DUAN values ('San pham X',1,'Tan Binh',5)
 - Chèn nhiều dòng
 - INSERT INTO <tên bảng> [(<danh sách thuộc tính>)] <lệnh truy vấn>

Cập nhật dữ liệu (2)

- Xóa dữ liệu
 - DELETE FROM <tên bảng> [WHERE <điều kiện>]
 - Ví dụ
 - delete from NHANVIEN where MaPhong = 5
 - delete from NHANVIEN

Cập nhật dữ liệu (3)

PHONGBAN	TenPB	MaPB	TrPhong	NgNhanChuc
	Nghiên cứu	5	888665555	19/06/1981
	Điều hành	4	987987987	01/01/1995

DIADIEM_PHG	MaPB	Diadiem
	5	Tân Bình
	5	Thủ Đức
	5	Phú Nhuận
	4	Gò Vấp

DUAN	TenDA	MaDA	Diadiem	Phong
	Sản phẩm X	1	Tân Bình	ngll
	Sản phẩm Y	2	Thủ Đức	ngll

Cập nhật dữ liệu (4)

■ Sửa dữ liệu

- UPDATE <tên bảng>
SET <tên thuộc tính> = <giá trị mới>,
<tên thuộc tính> = <giá trị mới>,

...

[WHERE <điều kiện>]

• Ví dụ

- update DUAN
set Diadiem = 'Hanoi', Phong = 4
where Maso=10
- update NHANVIEN
set Luong = 1.1 * Luong

Truy vấn dữ liệu

■ Cú pháp

- SELECT <dstt>
FROM <dsb>
[WHERE <đk>]
[GROUP BY <dstt nhóm>]
[HAVING <đk nhóm>]
[ORDER BY <dstt sắp xếp>]

■ Kết quả của lệnh truy vấn là một bảng.

■ Bảng trong SQL có thể chứa các bộ trùng nhau.

■ Trong đó

- <dstt>: danh sách thuộc tính của quan hệ kết quả.
- <dsb>: danh sách bảng liên quan đến câu truy vấn.
- <đk>: điều kiện để chọn hoặc ghép các bộ.
- <dstt nhóm>: danh sách thuộc tính để nhóm các bộ.
- <đk nhóm>: điều kiện chọn các nhóm.
- <dstt sắp xếp>: danh sách các thuộc tính và thứ tự sắp xếp tương ứng.

Phép toán quan hệ

- $\pi_{\langle \text{dstt} \rangle}(R)$
select <dstt>
from R

- $\sigma_{\langle \text{đkc} \rangle}(R)$
select *
from R
where <đkc>

- $R \times S$
select *
from R, S

- $R \bowtie_{\langle \text{đkk} \rangle} S$
select *
from R, S
where <đkk>

- $\pi_{\langle \text{dstt} \rangle}(\sigma_{\langle \text{đkc} \rangle}(R))$
select <dstt>
from R
where <đkc>

Ví dụ

- Tìm các nhân viên làm việc trong phòng số 4.
 - $\sigma_{MaPB=4}(NHANVIEN)$
 - `select *`
from NHANVIEN
where MaPB = 4
- Cho biết họ, tên, giới tính và mức lương của các nhân viên.
 - $\pi_{Ho, Ten, GTinh, Luong}(NHANVIEN)$
 - `select Ho, Ten, GTinh, Luong`
from NHANVIEN
- Cho biết tên các trưởng phòng
 - $\pi_{Ho, Ten}(PHONGBAN \bowtie_{TrPhg=MaNV} NHANVIEN)$
 - `select Ho, Ten`
from NHANVIEN, PHONGBAN
where TrPhg = MaNV

Phép toán tập hợp

- SQL hỗ trợ các phép toán
 - UNION (Hội)
 - EXCEPT (Hiệu).
 - INTERSECT (Giao).
 - Đặc điểm
 - Các dòng giống nhau bị loại trong bảng kết quả.
 - Các bảng tham gia phép toán phải có tính khả hợp
- Giữ lại các dòng giống nhau
 - UNION ALL
 - EXCEPT ALL
 - INTERSECT ALL
- `(SELECT <dstt> FROM <dsb> WHERE <đk>)`
UNION [ALL]
`(SELECT <dstt> FROM <dsb> WHERE <đk>)`
(SELECT <dstt> FROM <dsb> WHERE <đk>)
EXCEPT [ALL]
`(SELECT <dstt> FROM <dsb> WHERE <đk>)`
(SELECT <dstt> FROM <dsb> WHERE <đk>)
INTERSECT [ALL]
`(SELECT <dstt> FROM <dsb> WHERE <đk>)`

Đặt tên

- Đặt tên trong mệnh đề SELECT:
`select count(*) as 'Tong so nhan vien'`
from NHANVIEN
- Đặt tên trong mệnh đề FROM:
 - Đặt tên cho bảng
`select NV.Ho, NV.Ten, PB.Ten`
from NHANVIEN as NV, PHONGBAN as PB
where NV.MaPB = PB.MaPB
 - Đặt tên cho bảng và thuộc tính
`select *`
from PHONGBAN AS PB(Ten, Maso, MaTrPhg, Ngay)

Phép toán số học

- +, -, *, / có thể áp dụng cho các *giá trị số* trong mệnh đề SELECT
`select 1.1 * SUM(Luong) as 'Luong moi'`
from NHANVIEN

NHANVIEN	...	Luong	...	Luong moi
		40000		44000
		30000		33000
		38000		41800
		25000		27500

- +, - có thể áp dụng cho các *giá trị kiểu ngày giờ*
`select NgNhanChuc + 150 as 'Cong Ngay'`
from PHONGBAN

Phép toán so sánh và luận lý

- Dùng trong mệnh đề WHERE hoặc HAVING để xây dựng các điều kiện chọn và điều kiện kết.
 - =, <, ≤, >, ≥, <>
 - BETWEEN <giá trị> AND <giá trị>
 - AND, OR, NOT
- Tìm các nhân viên phòng số 5 có lương giữa 30.000 và 40.000
 - ```
select *
from NHANVIEN
where (Luong >= 30000) and (Luong <= 40000) and Phg = 5
```
  - ```
select *  
from NHANVIEN  
where (Luong between 30000 and 40000) and Phg = 5
```

Phép toán so sánh chuỗi

- LIKE <mẫu đối sánh>
[ESCAPE <kí tự thoát>]
- Mẫu đối sánh
 - Chuỗi ký tự để so sánh.
 - % - thay cho một đoạn ký tự tùy ý.
 - _ - thay cho một ký tự tùy ý.
- Ký tự thoát
 - Ký tự để loại bỏ chức năng đặc biệt của % và _.
 - Có thể dùng ký tự bất kỳ không xuất hiện trong mẫu đối sánh.
- Tìm nhân viên họ Nguyen
 - ```
select MaNV, Ho, Ten from
NHANVIEN
where Ho LIKE 'Nguyen%'
```
- Tìm nhân viên họ Nguyen\_
  - ```
select MaNV, Ho, Ten from  
NHANVIEN  
where Ho LIKE 'Nguyen\_%' escape '\'
```

Khử các dòng giống nhau

- ```
select Luong
from NHANVIEN
```
- SQL không tự động loại các bộ trùng nhau
  - Tốn thời gian so sánh và sắp xếp.
  - Sử dụng cho các truy vấn thống kê
- ```
select distinct Luong  
from NHANVIEN
```

Luong
40000
30000
38000
25000
43000
25000
25000

Luong
40000
30000
38000
25000
43000

Các hàm tập hợp

- SQL cung cấp 5 hàm tập hợp:
 - SUM(<tên thuộc tính>) - tính tổng các giá trị của thuộc tính
 - MAX(<tên thuộc tính>) - tìm giá trị lớn nhất của thuộc tính
 - MIN(<tên thuộc tính>) - tìm giá trị nhỏ nhất của thuộc tính
 - AVG(<tên thuộc tính>) - tính giá trị trung bình của thuộc tính
 - COUNT(*) - đếm số dòng của bảng
 - COUNT(<tên thuộc tính>) - đếm các giá trị khác null của thuộc tính
- Ví dụ
 - ```
select sum(Luong) AS TongLuong,
max(Luong) AS LuongCaonhat,
min(Luong) AS LuongThapnhat,
avg(Luong) AS LuongTB
from NHANVIEN
```
- Chỉ được xuất hiện trong mệnh đề SELECT hoặc HAVING

## Gom nhóm các bộ

- GROUP BY - HAVING
 

```
SELECT <dstt nhóm> [, <dshth>]
FROM <dsb>
[WHERE <đk>]
GROUP BY <dstt nhóm>
[HAVING <đk nhóm>]
```
- Trong đó
  - <dstt nhóm>: danh sách thuộc tính gom nhóm
  - <dshth>: danh sách các hàm tập hợp.
  - <đk>: điều kiện chọn hoặc điều kiện kết.
  - <đk nhóm>: điều kiện lựa chọn các nhóm.
- Chú ý
  - WHERE được thực hiện trước GROUP BY.
  - HAVING chỉ xuất hiện khi có GROUP BY

## Ví dụ

- Với mỗi phòng, cho biết số dự án phòng đó điều phối.  

```
select Phong, count(MaDA) as 'Số dự án'
from DUAN
group by Phong
```

| DUAN | TenDA       | MaDA | Diadiem   | Phong |
|------|-------------|------|-----------|-------|
|      | Sản phẩm X  | 1    | Tân Bình  | 5     |
|      | Sản phẩm Y  | 2    | Thủ Đức   | 5     |
|      | Sản phẩm Z  | 3    | Phú Nhuận | 5     |
|      | Tin học hóa | 10   | Gò Vấp    | 4     |
|      | Phúc lợi    | 30   | Gò Vấp    | 4     |
|      | Tái tổ chức | 20   | Phú Nhuận | 1     |

| Phong | Số dự án |
|-------|----------|
| 5     | 3        |
| 4     | 2        |
| 1     | 1        |

## Ví dụ

- Cho biết mã số, tên dự án và số nhân viên tham gia đối với những dự án có nhiều hơn 2 nhân viên tham gia.  

```
select DA.MaDA, DA.Ten, count(*) as 'Số nhân viên'
from DUAN as DA, THAMGIA as TG
WHERE DA.MaDA=TG.MaDA
group by DA.MaDA, DA.Ten
having count(*) > 2
```

## Gom nhóm các bộ (4)

| DA.TenDA   | DA.MaDA | ... | TG.MaNV   | TG.MaDA | ... |
|------------|---------|-----|-----------|---------|-----|
| Sản phẩm X | 1       |     | 123456789 | 1       |     |
| Sản phẩm X | 1       |     | 453453453 | 1       |     |
| Sản phẩm Y | 2       |     | 123456789 | 2       |     |
| Sản phẩm Y | 2       |     | 453453453 | 2       |     |
| Sản phẩm Y | 2       |     | 333445555 | 2       |     |
| ...        |         |     |           |         |     |

| DA.MaDA | DA.TenDA   | Số nhân viên |
|---------|------------|--------------|
| 2       | Sản phẩm Y | 3            |



## Sắp xếp kết quả

- ORDER BY
  - SELECT <dstt>  
FROM <dsb>  
[WHERE <đk>]  
ORDER BY <dstt sắp xếp>
  - <dstt sắp xếp>: danh sách các cặp (tên thuộc tính, thứ tự sắp xếp).
  - Thứ tự:
    - ASC - tăng dần.
    - DESC - giảm dần.
    - Mặc định là ASC.

## Ví dụ

- Với mỗi nhân viên, cho biết mã nhân viên và mã dự án mà nhân viên đó tham gia. Sắp xếp kết quả theo thứ tự tăng dần của mã nhân viên và giảm dần của mã dự án.

```
select MaNV, MaDA
from THAMGIA
order by MaNV, MaDA desc
```

| MaNV      | MaDA |
|-----------|------|
| 123456789 | 2    |
| 123456789 | 1    |
| 333445555 | 20   |
| 333445555 | 10   |
| 333445555 | 3    |
| 333445555 | 2    |
| ...       |      |

## So sánh với NULL

- NULL
  - Không biết.
  - Không sẵn sàng.
  - Không thể áp dụng.
- Tính toán và so sánh với NULL
  - $null + 3 \rightarrow null$ .
  - $null > 3 \rightarrow unknown$ .
- SQL cung cấp 2 phép toán
  - IS NULL.
  - IS NOT NULL.
- Tìm các nhân viên không có người giám sát

```
select MaNV, Ho, Ten
from NHANVIEN
where MaNQL is null
```
- Tìm các nhân viên có người giám sát

```
select MaNV, Ho, Ten
from NHANVIEN
where MaNQL is not null
```

## Logic 3 chân trị

| AND     | TRUE    | FALSE | UNKNOWN |
|---------|---------|-------|---------|
| TRUE    | TRUE    | FALSE | UNKNOWN |
| FALSE   | FALSE   | FALSE | FALSE   |
| UNKNOWN | UNKNOWN | FALSE | UNKNOWN |

| OR      | TRUE | FALSE   | UNKNOWN |
|---------|------|---------|---------|
| TRUE    | TRUE | TRUE    | TRUE    |
| FALSE   | TRUE | FALSE   | UNKNOWN |
| UNKNOWN | TRUE | UNKNOWN | UNKNOWN |

| NOT     |         |
|---------|---------|
| TRUE    | FALSE   |
| FALSE   | TRUE    |
| UNKNOWN | UNKNOWN |

## Truy vấn lồng

- Truy vấn sử dụng các giá trị của truy vấn khác trong điều kiện so sánh.

Truy vấn cha {  
SELECT <dstt>  
FROM <dsb>  
WHERE <so sánh tập hợp> (  
SELECT <dstt>  
FROM <dsb>  
WHERE <đk>)  
}

Truy vấn con

- Chỉ xuất hiện trong mệnh đề WHERE.

## So sánh tập hợp (1)

- Phép toán
  - IN - kiểm tra sự tồn tại của một giá trị trong một tập hợp.
  - ALL - so sánh một giá trị với tất cả các giá trị của tập hợp.
  - ANY - so sánh một giá trị với một giá trị nào đó của tập hợp.
  - ALL, ANY được kết hợp với các phép toán so sánh {=, <, ≤, >, ≥, <>}.
  - EXISTS - kiểm tra sự tồn tại của kết quả của một câu truy vấn.
- Cú pháp
  - <tên thuộc tính> IN <truy vấn con>
  - <tên thuộc tính> <phép toán so sánh> ALL <truy vấn con>
  - <tên thuộc tính> <phép toán so sánh> ANY <truy vấn con>
  - EXISTS <truy vấn con>

## Ví dụ

- Tìm các nhân viên của phòng số 2 và 5.
- Tìm các nhân viên của phòng 'Nghiên cứu'.

```
select *
from NHANVIEN
where MaPB in (2, 5)
```

Tập hợp tường minh

```
select *
from NHANVIEN
where MaPB = (
select MaPB
from PHONGBAN
where TenPB =
'Nghiên cứu')
```

Truy vấn con vô hướng

## Ví dụ

- Cho biết tên nhân viên có mức lương lớn hơn lương của các nhân viên phòng số 5.
- Cho biết các nhân viên không tham gia các dự án mà phòng số 5 quản lý.

```
select *
from NHANVIEN
where Luong > ALL (
select Luong
from NHANVIEN
where MaPB = 5)
```

```
select MaNV
from THAMGIA
where MaDA not in (
select MaNV
from DUAN
where Phg = 5)
```

## Truy vấn lồng phân cấp (1)

- Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các bảng trong mệnh đề FROM của truy vấn cha.
- Truy vấn con được thực hiện trước truy vấn cha.

### Ví dụ

```
select *
from NHANVIEN
where Luong > ALL (
 select Luong
 from NHANVIEN
 where MaPB = 5)
```

## Truy vấn lồng phân cấp (2)

| NHANVIEN | MaNV      | ... | Luong |       | Luong |
|----------|-----------|-----|-------|-------|-------|
|          | 888665555 |     | 55000 |       | 40000 |
|          | 333445555 |     | 40000 | > ALL | 30000 |
|          | 123456789 |     | 30000 |       | 38000 |
|          | 666884444 |     | 38000 |       | 25000 |
|          | 453453453 |     | 25000 |       |       |
|          | 987654321 |     | 43000 |       |       |
|          | 999887777 |     | 25000 |       |       |
|          | 987987987 |     | 25000 |       |       |

  

| MaNV      | ... | Luong |
|-----------|-----|-------|
| 888665555 |     | 55000 |
| 987654321 |     | 43000 |

## Truy vấn lồng tương quan (1)

- Mệnh đề WHERE của truy vấn con tham chiếu đến thuộc tính của các bảng trong mệnh đề FROM của truy vấn cha.
- Truy vấn con được thực hiện nhiều lần, mỗi lần ứng với một bộ của truy vấn cha.
- Tìm các nhân viên không có thân nhân nào

```
select *
from NHANVIEN as NV
where not exists (
 select *
 from THANNHAN as TN
 where TN.MaNV = NV.MaNV)
```

## Truy vấn lồng tương quan (2)

| NHANVIEN | MaNV      | ... | Luong | THANNHAN | MaNV      | Ten  | ... |
|----------|-----------|-----|-------|----------|-----------|------|-----|
|          | 888665555 |     | 55000 |          | 333445555 | Anh  |     |
|          | 333445555 |     | 40000 |          | 333445555 | The  |     |
|          | 123456789 |     | 30000 |          | 987654321 | An   |     |
|          |           |     |       |          | 123456789 | Minh |     |
|          |           |     |       |          | 123456789 | Anh  |     |

  

| MaNV      | Ten  | ... |
|-----------|------|-----|
| 323456789 | Minh |     |
| 323456789 | Anh  |     |

  

| MaNV      | ... | Luong |
|-----------|-----|-------|
| 888665555 |     | 55000 |

### Truy vấn lồng tương quan (3)

| A        | B        | C  |
|----------|----------|----|
| $\alpha$ | $\alpha$ | 1  |
| $\alpha$ | $\beta$  | 12 |
| $\beta$  | $\alpha$ | 23 |
| $\beta$  | $\beta$  | 3  |
| $\beta$  | $\beta$  | 23 |
| $\beta$  | $\beta$  | 12 |

$\pi_{A,B,C}(R)$

| S | D | E |
|---|---|---|
| 7 | 2 |   |
| 2 | 5 |   |

| R | A        | B        | C  | D  | E  |
|---|----------|----------|----|----|----|
|   | $\alpha$ | $\alpha$ | 1  | 7  | 2  |
|   | $\alpha$ | $\alpha$ | 1  | 2  | 5  |
|   | $\alpha$ | $\beta$  | 12 | 7  | 2  |
|   | $\beta$  | $\alpha$ | 23 | 7  | 2  |
|   | $\beta$  | $\beta$  | 3  | 1  | 10 |
|   | $\beta$  | $\alpha$ | 23 | 2  | 5  |
|   | $\beta$  | $\beta$  | 23 | 10 | 10 |
|   | $\beta$  | $\beta$  | 12 | 1  | 2  |

| R ÷ S | A        | B        | C  |
|-------|----------|----------|----|
|       | $\alpha$ | $\alpha$ | 1  |
|       | $\beta$  | $\alpha$ | 23 |

### Truy vấn lồng tương quan (4)

- Biểu diễn bằng EXISTS.  

```
SELECT R1.A, R1.B, R1.C
FROM R as R1
WHERE NOT EXISTS (
 SELECT *
 FROM S
 WHERE NOT EXISTS (
 SELECT *
 FROM R as R2
 WHERE R2.D = S.D AND
 R2.E = S.E AND
 R2.A = R1.A AND
 R2.B = R1.B AND
 R2.C = R1.C))
```
- Tìm các nhân viên *tham gia tất cả* các dự án do phòng số 5 quản lý.  

```
select *
from NHANVIEN as NV
where not exists (
 select *
 from DUAN as DA
 where DA.Phong = 5 and
 not exists (
 select *
 from THAMGIA as TG
 where TG.MaNV = NV.MaNV
 and TG.MaDA = DA.MaDA))
```

### Phép kết trong SQL

- JOIN, INNER JOIN
  - Dùng kết nối hai bảng trong mệnh đề FROM.
  - SELECT <dstt>  
FROM (<bảng 1> JOIN <bảng 2> ON <đkk>)
- Các phép kết mở rộng:
  - LEFT OUTER JOIN, LEFT JOIN.
  - RIGHT OUTER JOIN, RIGHT JOIN.
  - FULL OUTER JOIN, FULL JOIN.

### Ví dụ

- Cho biết tên các nhân viên của phòng 'Nghiên cứu'.  

```
select NV.Ho, NV.Dem, NV.Ten
from (NHANVIEN as NV join
 PHONGBAN as PB on
 NV.MaPB = PB.MaPB)
where PB.Ten = 'Nghiên cứu'
```
- Các phép kết có thể lồng nhau  

```
select DA.MaDA, PB.MaPB, NV.Ten
from ((DUAN as DA join
 PHONGBAN as PB on
 DA.Phong = PB.Maso)
join NHANVIEN on
 MaNQL = NV.MaNV)
where Diadiem = 'TpHCM'
```