
Chương 03

Truy vấn CSDL (tt)

GV: Ha My Trinh

Nội dung chi tiết

1. Truy vấn lồng

1. *Truy vấn lồng phân cấp*
2. *Truy vấn lồng tương quan*

2. Phép chia

1. *Biểu diễn ĐSQH*
2. *Ngôn ngữ SQL*

3. Khung nhìn

4. Ví dụ minh họa và bài tập áp dụng

1. Truy vấn lồng

```
SELECT MANV, TENNV
```

```
FROM NHANVIEN, PHONGBAN
```

```
WHERE TENPHG='Nghien cuu' AND PHG=MAPHG
```

Câu truy vấn cha
(Outer query)

```
SELECT <danh sách các cột>
```

```
FROM <danh sách các bảng>
```

```
WHERE <so sánh tập hợp> (
```

```
SELECT <danh sách các cột>
```

```
FROM <danh sách các bảng>
```

```
WHERE <điều kiện>)
```

Câu truy vấn con
(Subquery)

1. Truy vấn lồng (tt)

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- Câu truy vấn con thường trả về một tập các giá trị
- Mệnh đề WHERE của câu truy vấn cha
 - <biểu thức> <so sánh tập hợp> <truy vấn con>
 - So sánh tập hợp thường đi cùng với một số toán tử
 - IN, NOT IN
 - ALL
 - ANY hoặc SOME
 - Kiểm tra sự tồn tại
 - EXISTS
 - NOT EXISTS

1. Truy vấn lồng (tt)

■ Có 2 loại truy vấn lồng

- Lồng phân cấp

- Mệnh đề WHERE của **truy vấn con không tham chiếu** đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, câu **truy vấn con sẽ được thực hiện trước**

- Lồng tương quan

- Mệnh đề WHERE của **truy vấn con tham chiếu** ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, câu truy **vấn con sẽ được thực hiện nhiều lần**, mỗi lần tương ứng với một bộ của truy vấn cha

1. Ví dụ - Lồng phân cấp

1. Cho biết mã và tên nhân viên làm việc tại 'TP HCM'
2. Tìm những đề án có nhân viên họ 'Nguyễn' tham gia hoặc những đề án có trưởng phòng là họ 'Nguyễn'
3. Tìm những nhân viên không có thân nhân nào làm trong cùng công ty
4. Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên trong phòng 4
5. Tìm những trưởng phòng có tối thiểu một thân nhân

1. Ví dụ - Lông phân cấp

```
SELECT MANV, TENNV  
FROM NHANVIEN, DIADIEM_PHG  
WHERE DIADIEM='TP HCM' AND PHG=MAPHG
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE PHG IN (1, 5)
```

```
SELECT MAPHG  
FROM DIADIEM_PHG  
WHERE DIADIEM='TP HCM' )
```

1. Ví dụ - Lồng phân cấp (tt)

```
SELECT DISTINCT TENDA
```

```
FROM DEAN
```

```
WHERE MADA IN (
```

```
SELECT SODA
```

```
FROM NHANVIEN, PHANCONG
```

```
WHERE MANV=MA_NVNIEN AND HONV='Nguyen' )
```

```
OR MADA IN (
```

```
SELECT MADA
```

```
FROM NHANVIEN, PHONGBAN, DEAN
```

```
WHERE MANV=TRPHG AND MAPHG=PHONG
```

```
AND HONV='Nguyen' )
```

1. Ví dụ - Lồng phân cấp (tt)

- Tìm những nhân viên không có thân nhân nào làm việc trong cùng công ty

```
SELECT *  
FROM NHANVIEN  
WHERE MANV NOT IN (  
    SELECT MA_NVIEN  
    FROM THANNHAN )
```

```
SELECT *  
FROM NHANVIEN  
WHERE MANV <> ALL (  
    SELECT MA_NVIEN  
    FROM THANNHAN )
```

1. Ví dụ - Lông phân cấp (tt)

- Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG > ANY (  
    SELECT LUONG  
    FROM NHANVIEN  
    WHERE PHG=4 )
```

```
SELECT NV1.*  
FROM NHANVIEN NV1, NHANVIEN NV2  
WHERE NV1.LUONG > NV2.LUONG AND NV2.PHG=4
```

1. Ví dụ - Lồng phân cấp (tt)

- Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên phòng 4

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG > ALL (  
    SELECT LUONG  
    FROM NHANVIEN  
    WHERE PHG=4 )
```

1. Ví dụ - Lông phân cấp (tt)

- Tìm những trưởng phòng có tối thiểu một thân nhân

```
SELECT *
```

```
FROM NHANVIEN
```

```
WHERE MANV IN (SELECT MA_NVIAN FROM THANNHAN)
```

```
AND MANV IN (SELECT TRPHG FROM PHONGBAN)
```

1. Ví dụ - Lòng tương quan

1. Cho biết mã và tên nhân viên làm việc tại phòng 'Nghien cuu'
2. Tìm nhân viên có người thân cùng tên và cùng giới tính
3. Tìm những nhân viên không có thân nhân nào làm trong cùng công ty
4. Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên trong phòng 4
5. Tìm những trưởng phòng có tối thiểu một thân nhân

1. Ví dụ - Lồng tương quan

Cho biết mã và tên nhân viên làm việc tại phòng 'Nghien cuu'

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE EXISTS (
    SELECT *
    FROM PHONGBAN
    WHERE TENPHG='Nghien cuu' AND PHG=MAPHG )
```

1. Ví dụ - Lồng tương quan (tt)

Tìm nhân viên có người thân cùng tên và cùng giới tính

```
SELECT *  
FROM NHANVIEN NV  
WHERE EXISTS (  
    SELECT *  
    FROM THANNHAN TN  
    WHERE NV.MANV=TN.MA_NVIAN  
    AND NV.TENNV=TN.TENTN  
    AND NV.PHAI=TN.PHAI )
```

1. Ví dụ - Lồng tương quan (tt)

- Tìm những nhân viên không có thân nhân nào

```
SELECT *  
FROM NHANVIEN  
WHERE NOT EXISTS (  
    SELECT *  
    FROM THANNHAN  
    WHERE MANV=MA_NVIEEN)
```

1. Ví dụ - Lồng tương quan (tt)

- Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4

```
SELECT *
```

```
FROM NHANVIEN NV1
```

```
WHERE EXISTS (
```

```
    SELECT *
```

```
    FROM NHANVIEN NV2
```

```
    WHERE NV2PHG=4
```

```
    AND NV1.LUONG>NV2.LUONG)
```

1. Ví dụ - Lồng tương quan (tt)

- Tìm những trưởng phòng có tối thiểu một thân nhân

```
SELECT *  
FROM NHANVIEN  
WHERE EXISTS (  
    SELECT *  
    FROM THANNHAN  
    WHERE MANV=MA_NVNIEN )  
AND EXISTS (  
    SELECT *  
    FROM PHONGBAN  
    WHERE MANV=TRPHG )
```

Bài tập truy vấn lồng

1. Cho biết những nhân viên có địa chỉ ở Tp. HCM và thực hiện đề án cũng ở Tp. HCM
2. Cho biết những nhân viên chưa được phân công thực hiện đề án (dùng ***NOT in*** và ***NOT exists***)
3. Cho biết những đề án có thời gian thực hiện lớn hơn tất cả thời gian thực hiện của những đề án do phòng số 5 quản lý

Nhận xét IN và EXISTS

■ IN

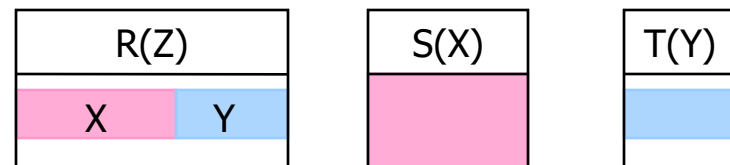
- <tên cột> IN <câu truy vấn con>
- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

■ 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

Phép chia

- Gọi R là quan hệ n ngôi và S là quan hệ m ngôi ($n > m, s \neq \emptyset$). Phép chia $R : S$ là tập hợp tất cả $(n-m)$ – bộ t sao cho với mọi bộ $u \in S$ thì bộ $t \cup u \in R$.
- **Ký hiệu : $R \div S$**
- Sử dụng định nghĩa phép tích **Đề - Các**, có thể định nghĩa phép chia hình thức hơn **$R \div S = Q$ sao cho $Q \times S \subseteq R$**
 - $R(Z)$ và $S(X)$
 - Z là tập thuộc tính của R , X là tập thuộc tính của S
 - $X \subseteq Z$



2. Phép chia (tt)

■ Ví dụ

R	A	B	C	D	E
	α	a	α	a	1
	α	a	γ	a	1
	α	a	γ	b	1
	β	a	γ	a	1
	β	a	γ	b	3
	γ	a	γ	a	1
	γ	a	γ	b	1
	γ	a	β	b	1

S	D	E
	a	1
	b	1

$R \div S$

A	B	C
α	a	γ
γ	a	γ

2. Phép chia (tt)

- Biểu diễn phép chia thông qua tập đầy đủ các phép toán ĐSQH

$$Q1 \leftarrow \pi_Y(R)$$

$$Q2 \leftarrow Q1 \times S$$

$$Q3 \leftarrow \pi_Y(Q2 - R)$$

$$T \leftarrow Q1 - Q3$$

2. Phép chia (tt)

B1. $Q1 \leftarrow \pi_Y (R)$

R	A	B	C	D	E
	α	a	α	a	1
	α	a	γ	a	1
	α	a	γ	b	1
	β	a	γ	a	1
	β	a	γ	b	3
	γ	a	γ	a	1
	γ	a	γ	b	1
	γ	a	β	b	1

Q1	A	B	C
	α	a	α
	α	a	γ
	β	a	γ
	γ	a	γ
	γ	a	β

2. Phép chia (tt)

$$B2. Q2 \leftarrow Q1 \times S$$

Q1	A	B	C
	α	a	α
	α	a	γ
	β	a	γ
	γ	a	γ
	γ	a	β

S	D	E
	a	1
	b	1

Q2	A	B	C	D	E
	α	a	α	a	1
	α	a	γ	a	1
	β	a	γ	a	1
	γ	a	γ	a	1
	γ	a	β	a	1
	α	a	α	b	1
	α	a	γ	b	1
	β	a	γ	b	1
	γ	a	γ	b	1
	γ	a	β	b	1

2. Phép chia (tt)

B3. $Q3 \leftarrow \pi_Y(Q2 - R)$

Q2	A	B	C	D	E
α	a	α	a	1	
α	a	γ	a	1	
β	a	γ	a	1	
γ	a	γ	a	1	
γ	a	β	a	1	
α	a	α	b	1	
α	a	γ	b	1	
β	a	γ	b	1	
γ	a	γ	b	1	
γ	a	β	b	1	

R	A	B	C	D	E
α	a	α	a	1	
α	a	γ	a	1	
α	a	γ	b	1	
β	a	γ	a	1	
β	a	γ	b	3	
γ	a	γ	a	1	
γ	a	γ	b	1	
γ	a	β	b	1	

Q3	A	B	C
γ	a	β	
α	a	α	
β	a	γ	

2. Phép chia (tt)

B4. $T \leftarrow Q1 - Q3$

Q1	A	B	C
	α	a	α
	α	a	γ
	β	a	γ
	γ	a	γ
	γ	a	β

Q3	A	B	C
	γ	a	β
	α	a	α
	β	a	γ

T	A	B	C
	α	a	γ
	γ	a	γ

2. Phép chia trong SQL $R \div S \Leftrightarrow R - (S - R)$

R	A	B	C	D	E
	α	a	α	a	1
	α	a	γ	a	1
	α	a	γ	b	1
	β	a	γ	a	1
	β	a	γ	b	3
	γ	a	γ	a	1
	γ	a	γ	b	1
	γ	a	β	b	1

S	D	E
b_i	a	1
	b	1

$R \div S$	A	B	C
a_i	α	a	γ
	γ	a	γ

- $R \div S$ là tập các giá trị a_i trong R sao cho không có giá trị b_i nào trong S làm cho bộ (a_i, b_i) không tồn tại trong R

2. Phép chia trong SQL (tt)

- Sử dụng NOT EXISTS để biểu diễn

```
SELECT R1.A, R1.B, R1.C
```

```
FROM R R1
```

```
WHERE NOT EXISTS (
```

```
    SELECT *
```

```
    FROM S
```

```
    WHERE NOT EXISTS (
```

```
        SELECT *
```

```
        FROM R R2
```

```
        WHERE R2.D=S.D AND R2.E=S.E
```

```
        AND R1.A=R2.A AND R1.B=R2.B AND R1.C=R2.C ))
```

2. Ví dụ

Cho biết Mã và họ tên các nhân viên được phân công làm tất cả các đề án

1. Sử dụng hàm kết hợp
2. Sử dụng phép chia

2. Ví dụ - Phép chia (tt)

- Cho biết mã số và họ tên các nhân viên được phân công làm tất cả các đề án
 - Tìm các nhân viên mà không có đề án nào là không được phân công làm
 - Tập bị chia: PHANCONG(MA_NVNIEN, SODA)
 - Tập chia: DEAN(MADA)
 - Tập kết quả: KQ(MA_NVNIEN)
 - Kết KQ với NHANVIEN để lấy ra TENNV

2. Ví dụ - Phép chia (tt)

```
SELECT NV.TENNV
FROM NHANVIEN NV, PHANCONG PC1
WHERE NV.MANV=PC1.MA_NVIENT
AND NOT EXISTS (
    SELECT *
    FROM DEAN DA
    WHERE NOT EXISTS (
        SELECT *
        FROM PHANCONG PC2
        WHERE PC2.SODA=DA.MADA
        AND PC1.MA_NVIENT=PC2.MA_NVIENT ))
```

Bài tập

1. Cho biết nhân viên làm việc cho **tất cả** các đề án mà phòng số 5 chủ trì
2. Cho biết những nhân viên tham gia **tất cả** đề án mà có nhân viên '001' tham gia
3. Cho biết phòng ban chủ trì tất cả các đề án ở Tp. HCM
4. Cho biết những nhân được phân công cho tất cả các công việc trong đề án 'Sản phẩm X'
5. Cho biết danh sách nhân viên tham gia vào tất cả các đề án ở Tp. HCM

3. Khung nhìn

- Bảng là một quan hệ được tổ chức lưu trữ vật lý trong CSDL
- Khung nhìn cũng là một quan hệ
 - Không được lưu trữ vật lý (bảng ảo)
 - Không chứa dữ liệu
 - Được định nghĩa từ những bảng khác
 - Có thể truy vấn hay cập nhật thông qua khung nhìn

3. Khung nhìn (tt)

- Tại sao phải sử dụng khung nhìn?
 - Che dấu tính phức tạp của dữ liệu
 - Đơn giản hóa các câu truy vấn
 - Hiển thị dữ liệu dưới dạng tiện dụng nhất
 - An toàn dữ liệu

3. Định nghĩa khung nhìn

■ Cú pháp

```
CREATE VIEW <tên khung nhìn> AS  
    <câu truy vấn>
```

```
DROP VIEW <tên khung nhìn>
```

■ Bảng ảo này có

- Danh sách thuộc tính trùng với các thuộc tính trong mệnh đề SELECT
- Số dòng phụ thuộc vào điều kiện ở mệnh đề WHERE
- Dữ liệu được lấy từ các bảng ở mệnh đề FROM

3. Ví dụ

```
CREATE VIEW NV_P5 AS
    SELECT MANV, HONV, TENLOT, TENVN
    FROM NHANVIEN
    WHERE PHG=5
```

```
CREATE VIEW TONGLNG_SLVN_PB AS
    SELECT MAPHG, TENPHG, COUNT(*) AS SLNV,
           SUM(LUONG) AS TONGLNG
    FROM NHANVIEN, PHONGBAN
    WHERE PHG=MAPHG
    GROUP BY TENPHG, MAPHG
```

3. Truy vấn trên khung nhìn

- Tuy không chứa dữ liệu nhưng có thể thực hiện các câu truy vấn trên khung nhìn

```
SELECT TENNV  
FROM NV_P5  
WHERE HONV LIKE 'Nguyen'
```

$$NV_P5 \leftarrow \pi_{MANV, HONV, TENLOT, TENNV} (\sigma_{PHG=5} (NHANVIEN))$$
$$\pi_{TENV} (\sigma_{HONV='Nguyen'} (NV_P5))$$

3. Truy vấn trên khung nhìn (tt)

- Có thể viết câu truy vấn dữ liệu từ khung nhìn và bảng

```
SELECT HONV, TENVN, TENDA, THOIGIAN  
FROM NV_P5, PHANCONG, DEAN  
WHERE MANV=MA_NVIEN AND SODA=MADA
```

$$NV_P5 \leftarrow \pi_{MANV, HONV, TENLOT, TENNV} (\sigma_{PHG=5} (NHANVIEN))$$
$$TMP \leftarrow NV_P5 \bowtie_{MANV=MA_NVIEN} PHANCONG \bowtie_{SODA=MADA} DEAN$$
$$\pi_{TENN, TENDA, THOIGIAN} (TMP)$$

3. Cập nhật trên khung nhìn

- Có thể dùng các câu lệnh INSERT, DELETE và UPDATE cho các khung nhìn đơn giản
 - Khung nhìn được xây dựng trên 1 bảng và có khóa chính của bảng
- Không thể cập nhật dữ liệu nếu
 - Khung nhìn có dùng từ khóa DISTINCT
 - Khung nhìn có sử dụng các hàm kết hợp
 - Khung nhìn có mệnh đề SELECT mở rộng
 - Khung nhìn được xây dựng từ bảng có RB trên cột
 - Khung nhìn được xây dựng từ nhiều bảng

3. Cập nhật trên khung nhìn (tt)

- Sửa lại họ cho nhân viên mã '123456789' ở phòng 5 là 'Pham'

```
UPDATE NV_P5  
SET HONV='Pham'  
WHERE MANV= '123456789'
```

4. Chỉ mục

- Chỉ mục trên thuộc tính A là một cấu trúc dữ liệu làm cho việc tìm kiếm mẫu tin có chứa A hiệu quả hơn

SELECT *

FROM NHANVIEN

WHERE PHG=5 AND PHAI='Nu'

Đọc 10.000 bộ

Đọc 200 bộ

Bảng NHANVIEN có 10.000 bộ

Có 200 nhân viên làm việc cho phòng 5

Đọc 70 bộ

4. Chỉ mục (tt)

- Cú pháp

```
CREATE INDEX <tên chỉ mục> ON <tên bảng>( <tên cột>)
```

```
DROP INDEX <tên chỉ mục>
```

- Ví dụ

```
CREATE INDEX PHG_IND ON NHANVIEN(PHG)
```

```
CREATE INDEX PHG_PHAU_IND ON NHANVIEN(PHG, PHAU)
```

4. Chỉ mục (tt)

■ Nhận xét

- Tìm kiếm nhanh trong trường hợp so sánh với hằng số và phép kết
- Làm chậm đi các thao tác thêm, xóa và sửa
- Tốn chi phí
 - Lưu trữ chỉ mục
 - Truy xuất đĩa nhiều

■ Chọn lựa cài đặt chỉ mục hợp lý???