

CƠ SỞ DỮ LIỆU



GIÁO VIÊN: ĐỖ THỊ MAI HƯỜNG
BỘ MÔN: CÁC HỆ THỐNG THÔNG TIN
KHOA: CÔNG NGHỆ THÔNG TIN
Email: dohuong@gmail.com



Truy vấn SQL

Nội dung chi tiết



- Truy vấn dữ liệu
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Nội dung chi tiết



- **Truy vấn dữ liệu**
 - Truy vấn cơ bản
 - Truy vấn lồng
 - Truy vấn tập hợp
 - Hàm kết hợp và gom nhóm
 - Một số kiểu truy vấn khác
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Truy vấn dữ liệu



- Là ngôn ngữ rút trích dữ liệu thỏa một số điều kiện nào đó
- Dựa trên

Phép toán ĐSQH

+

Một số bổ sung

- Cho phép 1 bảng có nhiều dòng trùng nhau

Truy vấn cơ bản



- Gồm 3 mệnh đề

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

- <danh sách các cột>
 - Tên các cột cần được hiển thị trong kết quả truy vấn
- <danh sách các bảng>
 - Tên các bảng liên quan đến câu truy vấn
- <điều kiện>
 - Biểu thức boolean xác định dòng nào sẽ được rút trích
 - Nối các biểu thức: AND, OR, và NOT
 - Phép toán: < , > , ≤ , ≥ , ≠ , =, LIKE và BETWEEN

Truy vấn cơ bản (tt)



- SQL và ĐSQH

~~SELECT~~ <danh sách các cột>
~~FROM~~ <danh sách các bảng>
~~WHERE~~ <điều kiện>

SELECT L
FROM $\pi_L(\sigma_C(R))$ →
WHERE C

Ví dụ



Lấy tất cả các cột
của quan hệ kết quả

SELECT *

FROM NHANVIEN

WHERE PHONG=5

MANV	HONV	TENDEM	TENNV	NS	DCHI	GT	LUONG	MA_NQL	PHO
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	NG 5
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	5

Mệnh đề SELECT



```
SELECT MANV, HONV, TENDEM, TENNV  
FROM NHANVIEN  
WHERE PHONG=5 AND GT='Nam'
```

MANV	HONV	TENDEM	TENNV
333445555	Nguyen	Thanh	Tung
987987987	Nguyen	Manh	Hung

Mệnh đề SELECT (tt)



Tên bí danh

```
SELECT MANV, HONV AS HO, TENDEM AS 'TEN DEM', TENNV AS TEN  
FROM NHANVIEN  
WHERE PHONG=5 AND GT='Nam'
```

MANV	HO	TEN DEM	TEN
333445555	Nguyen	Thanh	Tung
987987987	Nguyen	Manh	Hung

Mệnh đề SELECT (tt)



Mở rộng

```
SELECT MANV, HONV + ' ' + TENDEM + ' ' + TENNV AS 'HO TEN'  
FROM NHANVIEN  
WHERE PHONG=5 AND GT='Nam'
```

MANV	HO TEN
333445555	Nguyen Thanh Tung
987987987	Nguyen Manh Hung

Mệnh đề SELECT (tt)



Mở rộng

```
SELECT MANV, LUONG*1.1 AS 'LUONG10%'  
FROM NHANVIEN  
WHERE PHONG=5 AND GT='Nam'
```

MANV	LUONG10%
333445555	33000
987987987	27500

Mệnh đề SELECT (tt)



Loại bỏ các dòng trùng nhau

```
SELECT SELECT SELECT LUONG  
FROM FROM FROM NAMVINEN  
WHERE WHERE WHERE PHONG = 5 AND LGT = 'Nam'
```

LUONG

30000

25000

28000

38000

- Tốn chi phí

- Người dùng muốn thấy

Ví dụ



- Cho biết MANV và TENNV làm việc ở phòng ‘Nghiên cứu’

Mệnh đề WHERE



```
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE TENPHONG='Nghien cuu' AND PHONG=MAPHONG
```

Biểu thức logic

↓
TRUE

↓
TRUE

Mệnh đề WHERE (tt)



Độ ưu tiên

```
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE PHONG=MAPHONG AND  
      (TENPHONG='Nghien cuu' OR TENPHONG='Quan ly') AND GT='Nam'
```


Mệnh đề WHERE (tt)



BETWEEN

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE LUONG>20000 AND LUONG<30000
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE LUONG BETWEEN 20000 AND 30000
```

Mệnh đề WHERE (tt)



NOT BETWEEN

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE LUONG NOT BETWEEN 20000 AND 30000
```

Mệnh đề WHERE (tt)



LIKE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyen _ _ _ _ '
```

Ký tự bất kỳ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyen %'
```

Chuỗi bất kỳ

Mệnh đề WHERE (tt)



NOT LIKE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HONV LIKE 'Nguyen'
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HONV NOT LIKE 'Nguyen'
```

Mệnh đề WHERE (tt)



ESCAPE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE '% Nguyens_%' ESCAPE 's'
```



'Nguyen_'

```
WHERE ColumnA LIKE '%5/%%' ESCAPE '/'
```

Mệnh đề WHERE (tt)



Ngày giờ

SELECT MANV, TENNV

FROM NHANVIEN

WHERE NGSINH BETWEEN '1955-12-08' AND '1966-07-19'

Mệnh đề WHERE (tt)



NULL

- Sử dụng trong trường hợp
 - Không biết (value unknown)
 - Không thể áp dụng (value inapplicable)
 - Không tồn tại (value withheld)

Mệnh đề WHERE (tt)



NULL

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE MA_NQL IS NULL
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE MA_NQL IS NOT NULL
```


Mệnh đề FROM



Không sử dụng mệnh đề WHERE

SELECT MANV, MAPHONG
FROM NHANVIEN, PHONGBAN



MANV	MAPHONG
333445555	1
333445555	4
333445555	5
987987987	1
987987987	4
987987987	5
...	...

Mệnh đề FROM (tt)



Tên bí danh

```
SELECT TENPHONG, DIADIEM
FROM PHONGBAN, AS DIEM DD PHG AS DD
WHERE MA_PHONG=MA_PHONG
```

```
SELECT TENNV, NOSSINH, TEN, TNGSINH
FROM NHANVIEN, TV, THANHAN TN
WHERE MANV=MA_NVIENT
```

Ví dụ:



- Với những dự án ở ‘Ha Noi’, cho biết mã dự án, mã phòng, tên phòng chủ trì dự án, họ tên trưởng phòng cùng với ngày sinh và địa chỉ của người ấy

PhongBan(MaPB, TenPB, DC, DT, MaTP, NgayNC)

DuAn(MaDA, TenDA, DC, DT, MaPB)

NhanVien(MaNV, Hoten, NS, GT, DC, MaNGS, Luong, MaPB)

PhanCong(MaNV, MaDA, sogio)

ThanNhan(MaNV, Hoten, NS, GT, DC, Quanhe)

Ví dụ:



- Tìm họ tên của nhân viên phòng số 5 có tham gia vào dự án “Sản phẩm X” với số giờ làm việc trên 10 giờ

Ví dụ



- Tìm họ tên của từng nhân viên và người phụ trách trực tiếp nhân viên đó

Ví dụ



- Tìm họ tên của những nhân viên được “Nguyen Thanh Tung” phụ trách trực tiếp

Mệnh đề ORDER BY



- Dùng để hiển thị kết quả câu truy vấn theo một thứ tự nào đó
- Cú pháp

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
ORDER BY <danh sách các cột>
```

- ASC: tăng (mặc định)
- DESC: giảm

Mệnh đề ORDER BY (tt)



- Ví dụ

```
SELECT MA_NVIEN, SODA  
FROM PHANCONG  
ORDER BY MA_NVIEN DESC, SODA
```

MA_NVIEN	SODA
99988777	10
99988777	30
98798798	10
98798798	30
98765432	10
98765432	20
98765432	30

1

Nội dung chi tiết



- Truy vấn dữ liệu
 - Truy vấn cơ bản
 - **Truy vấn lồng**
 - Truy vấn tập hợp
 - Hàm kết hợp và gom nhóm
 - Một số dạng truy vấn khác
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Truy vấn lồng



- Là dạng truy vấn mà câu lệnh Select lồng trong câu lệnh Select
- Câu lệnh Select bên ngoài gọi là Truy vấn ngoài
- Câu lệnh Select bên trong gọi là Truy vấn trong hay Truy vấn con
- Có 2 loại:
 - Loại 1: Truy vấn con nằm trong mệnh đề Where
 - Loại 2: Truy vấn con nằm trong mệnh đề From

Truy vấn lồng (tt)



Loại 1: Truy vấn con nằm trong mệnh đề Where

Câu truy vấn ngoài
(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 trong
(Subquery)

Truy vấn lồng (tt)



- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức
- Câu truy vấn con thường trả về một tập các giá trị
- Mệnh đề WHERE của câu truy vấn ngoài
 - <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

Truy vấn lồng (tt)



- Câu lệnh truy vấn con nằm trong mệnh đề Where: có 2 loại
 - Lồng phân cấp:
 - Mệnh đề WHERE của truy vấn trong không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn ngoài.
 - Khi thực hiện, câu truy vấn trong sẽ được thực hiện trước.
 - Lồng tương quan:
 - Mệnh đề WHERE của truy vấn trong tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn ngoài.
 - Khi thực hiện, câu truy vấn trong sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn ngoài.

Truy vấn lồng (tt)



Ví dụ - Lồng phân cấp: Đưa ra mã, họ tên của những nhân viên thuộc phòng ban có địa điểm tại TP HCM

```
SELECT MANV, HOTEN
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.MAPB=PHONGBAN.MAPB AND DIADIEM='TP HCM'
```

```
SELECT MANV, HOTEN
```

```
FROM NHANVIEN
```

```
WHERE MAPB IN (1, 5)
```

```
SELECT MAPB
```

```
FROM PHONGBAN
```

```
WHERE DIADIEM='TP HCM' )
```

Truy vấn lồng (tt)



Ví dụ: Đưa ra mã, họ tên của những nhân viên là trưởng phòng

PhongBan						
	MaPB	TenPB	MaTP	NgayNC	DiaDiem	Dienthoai
1	PB01	Hành chính	NV01	2001-10-20	Hà nội	NULL
2	PB02	Kế hoạch	NV02	2012-12-01	TP HCM	NULL
3	PB03	Nghiên cứu	NV03	2011-06-02	Hà nội	NULL
4	PB04	Quan hệ quốc tế	NV04	2009-12-22	TP HCM	NULL

NhanVien								
	MaNV	HoTen	MaPB	NgaySinh	Luong	MaNGS	Gioitinh	Diachi
1	NV01	Nguyễn Văn Tuấn	PB01	1970-10-06	15000000	NULL	Nam	Hà nội
2	NV02	Nguyễn Hải Lâm	PB02	1972-07-12	15000000	NULL	Nam	TP HCM
3	NV03	Nguyễn Quang Hà	PB03	1978-10-04	12000000	NULL	Nam	Hà nội
4	NV04	Trần Mỹ Linh	PB04	1976-04-27	12000000	NULL	Nữ	TP HCM
5	NV05	Hoàng Anh Tuấn	PB01	1992-09-25	8000000	NV07	Nam	Hà nội
6	NV06	Thái Thùy Trang	PB02	1992-08-19	7000000	NV08	Nữ	TP HCM
7	NV07	Phạm Thị Tuyết	PB01	1960-12-22	7500000	NULL	Nữ	Hà nội
8	NV08	Phan Sơn	PB02	1970-12-03	9000000	NULL	Nam	TP HCM
9	NV09	Nguyễn Hoàng ...	PB04	1992-04-05	6000000	NV04	Nam	TP HCM
10	NV10	Nguyễn Thảo My	PB03	1987-05-07	8000000	NULL	Nữ	Hà nội

Trưởng phòng

Truy vấn lồng (tt)



Ví dụ - Lồng tương quan: Đưa ra mã, họ tên của những nhân viên thuộc phòng Nghiên cứu

```
SELECT MANV, HOTEN
```

```
FROM NHANVIEN N, PHONGBAN P
```

```
WHERE N.MAPB=P.MAPB AND TENPB='Nghien cuu'
```

```
SELECT MANV, HOTEN
```

```
FROM NHANVIEN
```

```
WHERE EXISTS (
```

```
SELECT *
```

```
FROM PHONGBAN
```

```
WHERE MAPB=NHANVIEN.MAPB AND TENPB='Nghien cuu' )
```


Truy vấn lòng (tt)



Ví dụ: Đưa ra mã, họ tên của những nhân viên là trưởng phòng

Truy vấn lồng (tt)



- Nhận xét IN và EXISTS
- IN
 - <tên cột> IN <câu truy vấn trong>
 - Thuộc tính ở mệnh đề SELECT của truy vấn trong phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn ngoài
- 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 trong
 - Những câu truy vấn có = ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS

Truy vấn lồng (tt)



- Loại 2: Truy vấn con nằm trong mệnh đề FROM
 - Kết quả trả về của một câu truy vấn là một bảng
 - Bảng trung gian trong quá trình truy vấn
 - Không có lưu trữ thật sự
- Cú pháp

```
SELECT <danh sách các cột>  
FROM R1, R2, (<truy vấn con>) AS tên_bảng  
WHERE <điều kiện>
```

Truy vấn lồng (tt)



- Ví dụ: Đưa ra mã nhân viên, họ tên và tên phòng ban tương ứng của nhân viên

```
select manv,hoten,tenpb  
from nhanvien, (select mapb,tenpb from phongban) PB  
where nhanvien.mapb=pb.mapb
```

Truy vấn lòng (tt)



- Ví dụ: Đưa ra danh sách nhân viên là trưởng phòng.

Truy vấn lồng (tt)



Bài tập: Sử dụng các loại truy vấn thực hiện:

1. Đưa ra nhân viên là người giám sát

Gợi ý: Nhân viên giám sát: là những nhân viên có giá trị trong trường MaNGS

NhanVien

	MaNV	HoTen	MaPB	NgaySinh	Luong	MaNGS	Gioitinh	Diachi
1	NV01	Nguyễn Văn Tuấn	PB01	1970-10-06	15000000	NULL	Nam	Hà nội
2	NV02	Nguyễn Hải Lâm	PB02	1972-07-12	15000000	NULL	Nam	TP HCM
3	NV03	Nguyễn Quang Hà	PB03	1978-10-04	12000000	NULL	Nam	Hà nội
4	NV04	Trần Mỹ Linh	PB04	1976-04-27	12000000	NULL	Nữ	TP HCM
5	NV05	Hoàng Anh Tuấn	PB01	1992-09-25	8000000	NV07	Nam	Hà nội
6	NV06	Thái Thùy Trang	PB02	1992-08-19	7000000	NV08	Nữ	TP HCM
7	NV07	Phạm Thị Tuyết	PB01	1960-12-22	7500000	NULL	Nữ	Hà nội
8	NV08	Phan Sơn	PB02	1970-12-03	9000000	NULL	Nam	TP HCM
9	NV09	Nguyễn Hoàng ...	PB04	1992-04-05	6000000	NV04	Nam	TP HCM
10	NV10	Nguyễn Thảo My	PB03	1987-05-07	8000000	NULL	Nữ	Hà nội

Truy vấn lồng (tt)



Bài tập: Sử dụng các loại truy vấn thực hiện:

2. Tìm những nhân viên có lương bằng lương của ít nhất một nhân viên thuộc phòng 'PB01'.

NhanVien

	MaNV	HoTen	MaPB	NgaySinh	Luong	MaNGS	Gioitinh	Diachi
1	NV01	Nguyễn Văn Tuấn	PB01	1970-10-06	15000000	NULL	Nam	Hà nội
2	NV02	Nguyễn Hải Lâm	PB02	1972-07-12	15000000	NULL	Nam	TP HCM
3	NV03	Nguyễn Quang Hà	PB03	1978-10-04	12000000	NULL	Nam	Hà nội
4	NV04	Trần Mỹ Linh	PB04	1976-04-27	12000000	NULL	Nữ	TP HCM
5	NV05	Hoàng Anh Tuấn	PB01	1992-09-25	8000000	NV07	Nam	Hà nội
6	NV06	Thái Thùy Trang	PB02	1992-08-19	7000000	NV08	Nữ	TP HCM
7	NV07	Phạm Thị Tuyết	PB01	1960-12-22	7500000	NULL	Nữ	Hà nội
8	NV08	Phan Sơn	PB02	1970-12-03	9000000	NULL	Nam	TP HCM
9	NV09	Nguyễn Hoàng ...	PB04	1992-04-05	6000000	NV04	Nam	TP HCM
10	NV10	Nguyễn Thảo My	PB03	1987-05-07	8000000	NULL	Nữ	Hà nội

Truy vấn lòng (tt)



- Bài tập: Thực hiện truy vấn
 1. Đưa ra phòng ban có phụ trách dự án. Thông tin đưa ra gồm mã phòng ban, tên phòng ban.
 2. Đưa ra phòng ban chưa có trưởng phòng và không phụ trách dự án nào. Thông tin đưa ra gồm mã phòng ban, tên phòng ban.
 3. Đưa ra nhân viên không có người thân. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, địa chỉ.
 4. Đưa ra nhân viên không tham gia dự án. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, địa chỉ.
 5. Đưa ra dự án không có nhân viên nào tham gia. Thông tin đưa ra gồm mã dự án, tên dự án.

Truy vấn lòng (tt)



- Bài tập: Thực hiện truy vấn
 6. Đưa ra nhân viên là trưởng phòng có tham gia dự án. Thông tin đưa ra gồm mã nhân viên, họ tên
 7. Đưa ra nhân viên là người giám sát có tham gia dự án.
 8. Đưa ra những nhân viên có lương lớn hơn lương của tất cả các nhân viên của phòng 'Hành chính'
 9. Đưa ra những nhân viên có ít nhất một người thân

Nội dung chi tiết



- Truy vấn dữ liệu
 - Truy vấn cơ bản
 - Truy vấn lồng
 - Truy vấn tập hợp
 - Hàm kết hợp và gom nhóm
 - Một số dạng truy vấn khác
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Phép toán tập hợp trong SQL



- SQL có cài đặt các phép toán
 - Hợp (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

Phép toán tập hợp trong SQL (tt)



- Cú pháp

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
UNION [ALL]
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
INTERSECT
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
EXCEPT
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

Ví dụ



- Cho biết các mã đề án có
 - Nhân viên với họ là ‘Nguyen’ tham gia hoặc,
 - Trưởng phòng chủ trì đề án đó với họ là ‘Nguyen’

Ví dụ



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

Ví dụ

- Tìm những nhân viên không có thân nhân nào



Phép chia trong SQL



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

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 ))
```

Ví dụ



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

Ví dụ (tt)



- Tìm tên các nhân viên được phân công làm tất cả các đề án

Select *

From nhanvien

where manv in(

SELECT r1.manv

FROM PHANCONG R1

WHERE not exists(

SELECT MADA

FROM DEAN S

WHERE not EXISTS (

SELECT *

FROM PHANCONG R2

WHERE R2.SODA=S.MADA

and r1.manv=r2.manv)

))

Nội dung chi tiết



- Truy vấn dữ liệu
 - Truy vấn cơ bản
 - Truy vấn lồng
 - Truy vấn tập hợp
 - **Hàm tính toán, thống kê và gom nhóm**
 - Một số dạng truy vấn khác
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Hàm tính toán, thống kê



- COUNT
 - COUNT(*) đếm số dòng
 - COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
 - COUNT(DISTINCT <tên thuộc tính>) đếm số giá trị khác nhau và khác NULL của thuộc tính
- MIN
- MAX
- SUM
- AVG
- Các hàm tính được đặt ở mệnh đề SELECT

Ví dụ



- Tìm tổng lương, lương cao nhất, lương thấp nhất và lương trung bình của các nhân viên

```
select tongluong=sum(luong) ,max(luong) as  
luong_max,min(luong) luong_min ,avg(luong) luongTB  
from nhanvien
```

Ví dụ



- Cho biết số lượng nhân viên của phòng ‘Nghien cuu’

Ví dụ



- Cho biết số lượng nhân viên của từng phòng ban

PHONG	SL_NV
5	3
4	3
1	1

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHO
33344555	Nguyen	Thanh	Tung	12/08/1955	633 NVC	Nam	40000	88866555	5
98798798	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	33344555	5
45345345	Tran	Thanh	Tam	07/31/1972	543 MTL	Nu	25000	33344555	5
99988777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	98765432	4
98765432	Le	Quynh	Nhu	07/20/195	219 TD Q3	Nu	43000	88866555	4
98798798	Tran	Hong	Quang	04/08/1969	980 LHP	Nam	25000	98765432	4
88866555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

Gom nhóm



- Cú pháp

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
GROUP BY <danh sách các cột gom nhóm>
```

- Sau khi gom nhóm

- Mỗi nhóm các bộ sẽ có cùng giá trị tại các thuộc tính gom nhóm

Ví dụ

- Cho biết số lượng nhân viên của từng phòng ban



Ví dụ



- Với mỗi nhân viên cho biết mã số, họ tên, số lượng đề án và tổng thời gian mà họ tham gia

MA_NVIEN	SODA	THOIGIAN
123456789	1	32.5
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
453453453	1	20.0
453453453	2	20.0

Ví dụ



- Cho biết những nhân viên tham gia từ 2 đề án trở lên

MA_NVIENT	SODA	THOIGIAN
12345678	1	32.5
12345678	2	7.5
33344555	2	10.0
33344555	3	10.0
33344555	10	10.0
88866555	20	20.0
98798798	10	35.0
98798798	30	5.0
98765432	30	20.0
98765432	20	15.0
45345345	1	20.0
45345345	2	20.0

bị loại ra

Điều kiện trên nhóm



- Cú pháp

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
GROUP BY <danh sách các cột gom nhóm>  
HAVING <điều kiện trên nhóm>
```

Ví dụ



- Cho biết những nhân viên tham gia từ 2 đề án trở lên

Ví dụ



- Cho biết những phòng ban (TENPHONG) có lương trung bình của các nhân viên lớn hơn 2tr

Nhận xét



- Mệnh đề GROUP BY
 - Các thuộc tính trong mệnh đề SELECT (trừ những thuộc tính trong các hàm kết hợp) phải xuất hiện trong mệnh đề GROUP BY
- Mệnh đề HAVING
 - Sử dụng các hàm kết hợp trong mệnh đề SELECT để kiểm tra một số điều kiện nào đó
 - Chỉ kiểm tra điều kiện trên nhóm, không là điều kiện lọc trên từng bộ
 - Sau khi gom nhóm điều kiện trên nhóm mới được thực hiện

Nhận xét (tt)



- Thứ tự thực hiện câu truy vấn có mệnh đề GROUP BY và HAVING
 - (1) Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
 - (2) Những dòng này sẽ được gom thành nhiều nhóm tương ứng với mệnh đề GROUP BY
 - (3) Áp dụng các hàm kết hợp cho mỗi nhóm
 - (4) Bỏ qua những nhóm không thỏa điều kiện trong mệnh đề HAVING
 - (5) Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT

Ví dụ



- Tìm các nhân viên được phân công làm tất cả các đề án

Ví dụ



- Tìm 3 nhân viên có lương cao nhất

```
select top 3 *  
from nhanvien  
order by lương desc
```

Ví dụ



- Đưa ra các nhân viên có lương cao nhất trong từng phòng ban

Ví dụ



- Đưa ra phòng ban có đông nhân viên nhất

Nội dung chi tiết



- Giới thiệu
- Định nghĩa dữ liệu
- **Truy vấn dữ liệu**
 - Truy vấn cơ bản
 - Tập hợp, so sánh tập hợp và truy vấn lồng
 - Hàm kết hợp và gom nhóm
 - **Một số dạng truy vấn khác**
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Một số dạng truy vấn khác



- Điều kiện kết ở mệnh đề FROM
 - Phép kết tự nhiên
 - Phép kết ngoài

Điều kiện kết ở mệnh đề FROM



- Kết bằng

```
SELECT <danh sách các cột>  
FROM R1 [INNER] JOIN R2 ON <biểu thức>  
WHERE <điều kiện>
```

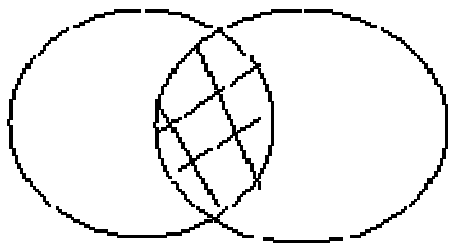
- Kết ngoài

```
SELECT <danh sách các cột>  
FROM R1 LEFT|RIGHT [FULL] JOIN R2 ON <biểu thức>  
WHERE <điều kiện>
```

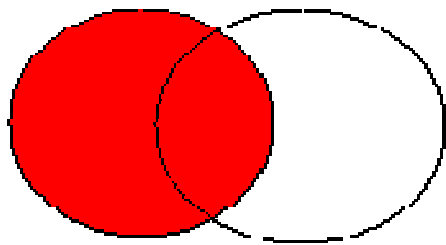

Lệnh Select với Join



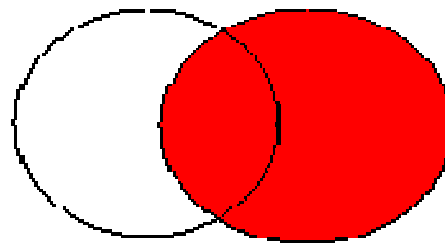
JOIN - Kết hợp dữ liệu từ 2 hoặc nhiều bảng



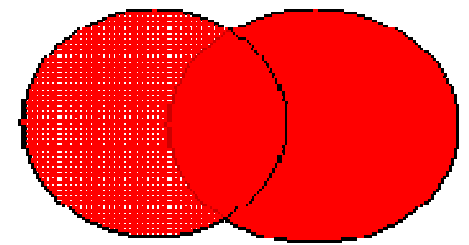
Join



Left Join



Right Join



Full Join

Ví dụ



- Tìm mã và tên các nhân viên làm việc tại phòng ‘Nghien cuu’

```
select manv,tennv
```

```
from nhanvien join phongban on phong=maphong
```

```
where tenphong = ‘Nghien cuu’
```

- Tìm mã, tên nhân viên, và tên đề án mà nhân viên tham gia

```
Select nv.manv,tenda
```

```
From nhanvien nv join phancong pc on
```

```
nv.manv=pc.manv
```

```
Join dean da on da.mada=pc.mada
```

Ví dụ



- Tìm họ tên các nhân viên và tên các đề án nhân viên tham gia nếu có

`select` tennv,mada,tenda

`From` (`select` tennv,soda

`from` nhanvien nv `left join` phancong pc `on`
nv.manv=pc.manv) `as` a `left join` dean `on`
soda=mada

Truy vấn với Computer-by, Rollup và Cube



Cho biết họ tên, tên đơn vị hệ số lương của nhân viên đồng thời cho biết lương trung bình của các nhân viên trong mỗi đơn vị

```
SELECT hoten,tendonvi,hsluong  
FROM nhanvien n,donvi d  
WHERE n.madonvi=d.madonvi  
ORDER BY n.madonvi  
COMPUTE AVG(hsluong) BY n.madonvi
```

Hoten	tendonvi	hsluong
Nguyễn Thị Hoa	Phòng kế toán	2.11
		avg
		2.11
Lê Hoài Nam	Phòng Tổ chức	1.86
Hoàng Nam Phong	Phòng Tổ chức	3.21
		avg
		2.535

compute
by

Truy vấn với Computer-by, Rollup và Cube



Rollup

	Student name	Subject name	Average mark
1	Hung	C	7.0
2	Hung	JAVA	5.0
3	Hung	SQL	5.0
4	Hung	NULL	7.25
5	Tuan	C	7.0
6	Tuan	JAVA	4.0
7	Tuan	SQL	5.5
8	Tuan	NULL	6.5
9	NULL	NULL	6.875

Tùy chọn ROLLUP
thêm vào

```
SELECT CASE When GROUPING(name) = 1 THEN 'All students'
        ELSE name END AS [Student name],
       CASE when GROUPING(subject) = 1 THEN 'All subjects'
        ELSE subject END AS [Subject name],
       CAST(AVG(mark) AS DECIMAL(9, 2)) AS [Average mark]
FROM Student
GROUP BY name, subject WITH ROLLUP
```

Truy vấn với Computer-by, Rollup và Cube



	Student name	Subject name	Average mark
1	Hung	C	8.00
2	Hung	JAVA	8.00
3	Hung	SQL	5.00
4	Hung	All subjects	7.25
5	Tuan	C	7.00
6	Tuan	JAVA	4.00
7	Tuan	SQL	7.50
8	Tuan	All subjects	6.50
9	All students	All subjects	6.88
10	All students	C	7.67
11	All students	JAVA	6.00
12	All students	SQL	6.67

Các bản ghi do
tùy chọn CUBE
thêm vào

```
SELECT CASE When GROUPING(name) = 1 THEN 'All students'
        ELSE name END AS [Student name],
       CASE When GROUPING(subject) = 1 THEN 'All subjects'
        ELSE subject END AS [Subject name],
       CAST(AVG(mark) AS DECIMAL(9, 2)) AS [Average mark]
FROM Student
GROUP BY name, subject WITH CUBE
```

Kết luận



SELECT <danh sách các cột>

FROM <danh sách các bảng>

[**WHERE** <điều kiện>]

[**GROUP BY** <các thuộc tính gom nhóm>]

[**HAVING** <điều kiện trên nhóm>]

[**ORDER BY** <các thuộc tính sắp thứ tự>]

Nội dung chi tiết



- Truy vấn dữ liệu
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Cập nhật dữ liệu từ truy vấn



- Thêm dữ liệu:

Insert into tên_bảng

Câu_lệnh_Select

- Sửa dữ liệu

Update tên_bảng

Set tên_trường = giá_trị[,

tên_trường = giá_trị]

Where điều_kiện

// Câu lệnh Select nằm trong mệnh đề Where

- Xóa dữ liệu:

Delete tên_bảng

Where điều_kiện

Cập nhật dữ liệu từ truy vấn (tt)



- Ví dụ: Thêm một nhân viên, cho nhân viên này tham gia tất cả các dự án mà phòng ban của mình phụ trách

Cập nhật dữ liệu từ truy vấn (tt)



- Ví dụ: Thêm trường Tongsogio int vào bảng Duan.

Cập nhật giá trị cho trường Tongsogio của mỗi dự án sao cho giá trị của trường Tongsogio bằng tổng số giờ của dự án tương ứng trong bảng PhanCong .

Cập nhật dữ liệu từ truy vấn (tt)



- Ví dụ: Xóa một nhân viên. Xóa hết các thông tin về nhân viên này trong các bảng liên quan

Nội dung chi tiết



- Truy vấn dữ liệu
- Cập nhật dữ liệu từ truy vấn
- Tính đầy đủ của SQL
- Khung nhìn (view)

Tính đầy đủ của SQL



- Xem xét khả năng của SQL có thực hiện được tất cả các biểu thức của các phép toán đại số quan hệ ta cần xét lần lượt cho các phép toán cơ bản của đại số quan hệ.
- Trong chương Đại số quan hệ, ta đã có 5 phép toán cơ bản là hợp, hiệu, tích Decac, chọn, chiếu của đại số quan hệ độc lập với nhau.
- Các phép toán khác của đại số quan hệ như nối tự nhiên, giao, nối nửa, nối theta, chia đều có thể nhận được từ phép toán cơ bản trên.
- Vậy để SQL thực hiện được các phép toán đại số quan hệ, ta chỉ cần cài đặt cho SQL thực hiện 5 phép toán cơ bản hợp, hiệu, tích Decac, chọn, chiếu .

Tính đầy đủ của SQL (tt)



- Phép hợp:

Giả sử ta có 2 quan hệ r và s có cùng lược đồ

$R=\{A_1, A_2, \dots, A_n\}$. Khi đó để tính $T=r+s$ ta viết:

```
Insert into T
```

```
Select *
```

```
From r;
```

Tiếp theo là:

```
Insert into T
```

```
Select *
```

```
From s;
```

Tính đầy đủ của SQL (tt)



- Phép trừ:

Để tính $T=r-s$ trước tiên chúng ta chèn r vào T

```
Insert into T
```

```
Select *
```

```
From r;
```

sau đó ta dùng câu lệnh xóa như sau:

```
Delete from T
```

```
Where ( $A_1, \dots, A_n$ ) in
```

```
(Select * from s)
```


Tính đầy đủ của SQL (tt)



- Tích Decac

Insert into T

Select r.*,s.*

From r,s

Tính đầy đủ của SQL (tt)



- Phép chọn $T=r(E)$

Insert into T

Select *

From r

Where E;

- Phép chiếu:

Giả sử X là tập con của R, $X=\{A_1, \dots, A_k\}$

$T=r.X$, ta có câu lệnh

Insert into T

Select A_1, \dots, A_k

From r;

Nội dung chi tiết



- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu
- Tính đầy đủ của SQL
- **Khung nhìn (view)**
 - Định nghĩa
 - Truy vấn
 - Cập nhật

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

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

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**

Ví dụ



```
CREATE VIEW NV_P5 AS
    SELECT MANV, HODEM, TENVN
    FROM NHANVIEN
    WHERE PHONG=5
```

```
CREATE VIEW TONGLNG_SLVN_PB AS
    SELECT MAPHONG, TENPHONG, COUNT(*) AS
SLNV,          SUM(LUONG) AS TONGLNG
    FROM NHANVIEN, PHONGBAN
    WHERE PHONG=MAPHONG
    GROUP BY MAPHONG ,TENPHONG
```

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 HODEM LIKE 'Nguyen'
```

$$NV_P5 \leftarrow \pi_{MANV, HODEM, TENNV} (\sigma_{Phong=5} (NHANVIEN))$$
$$\pi_{TENNV} (\sigma_{HODEM='Nguyen'} (NV_P5))$$

Truy vấn trên khung nhìn (tt)



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

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

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

Ví dụ



- Tìm những phòng ban có lương trung bình cao nhất

```
CREATE VIEW LUONGTB_PHONG AS  
SELECT TENPHONG,MAPHONG,AVG(LUONG) AS LUONGTB  
FROM NHANVIEN,PHONGBAN  
WHERE PHONG=MAPHONG  
GROUP BY TENPHONG,MAPHONG
```

```
SELECT *  
FROM LUONGTB_PHONG  
WHERE LUONGTB =(  
SELECT MAX(LUONGTB)  
FROM LUONGTB_PHONG)
```

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

Cập nhật trên khung nhìn (tt)



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

```
UPDATE NV_P5
```

```
SET HODEM='Pham'
```

```
WHERE MANV= '123456789'
```

