



4. PHÉP TOÁN TẬP HỢP TRONG SQL

- SQL có cài đặt các phép toán tập hợp:
 - 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



4. PHÉP TOÁN TẬP HỢP TRONG SQL

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

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



4. PHÉP TOÁN TẬP HỢP TRONG SQL

- Ví dụ 7: Cho biết các mã đề án có nhân viên với họ là 'Nguyen' tham gia hoặc trưởng phòng chủ trì đề án đó có họ là 'Nguyen'

```
SELECT SODA
FROM NHANVIEN, PHANCONG
WHERE MANV=MA_NVIENT AND HONV='Nguyen'
UNION
SELECT MADA
FROM NHANVIEN, PHONGBAN, DEAN
WHERE MANV=TRPHG AND MAPHG=PHONG
AND HONV='Nguyen'
```



4. PHÉP TOÁN TẬP HỢP TRONG SQL

- Ví dụ 8: Tìm những nhân viên có người thân cùng tên và cùng giới tính với nhân viên đó

```
SELECT TENNV, PHAI FROM NHANVIEN  
INTERSECT  
SELECT TENTN, PHAI FROM THANNHAN
```

```
SELECT NV.*  
FROM NHANVIEN NV, THANNHAN TN  
WHERE NV.MANV=TN.MA_NVLEN  
AND NV.TENNV=TN.TENTN AND NV.PHAI=TN.PHAI
```



4. PHÉP TOÁN TẬP HỢP TRONG SQL

- Ví dụ 9: Tìm những nhân viên không có thân nhân nào

```
SELECT MANV FROM NHANVIEN
```

```
EXCEPT
```

```
SELECT MA_NVIEN AS MANV FROM THANNHAN
```

=> Không dùng Except thì sao ??



4. TRUY VẤN LÒNG

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)



4. TRUY VẤN LÒNG

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức
- Câu truy vấn con thường trả về một tập các giá trị
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- 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



4. TRUY VẤN LÒNG

- 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



4. TRUY VẤN LÒNG – PHÂN CẤP

- Ví dụ

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

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

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE PHG IN ( 1, 4, 5)
```



4. TRUY VẤN LÒNG – PHÂN CẤP

- Ví dụ 7: Cho biết các mã đề án có nhân viên với họ là 'Nguyen' tham gia hoặc trưởng phòng chủ trì đề án đó có họ là 'Nguyen'

```
SELECT DISTINCT TENDA
FROM DEAN
SELECT SODA
WHERE MADA IN (
FROM NHANVIEN NV, PHANCONG PC
SELECT SODA
WHERE NV.MANV=PC.MA_NVNEN AND NV.HONV='Nguyen'
UNION
FROM NHANVIEN, PHANCONG
WHERE MANV=MA_NVNEN AND HONV='Nguyen')
SELECT MADA
OR MADA IN (
FROM NHANVIEN NV, PHONGBAN PB, DEAN DA
SELECT MADA
WHERE NV.MANV=PB.TRPHG AND PB.MAPHG=DA.PHONG
AND NV.HONV='Nguyen'
FROM NHANVIEN, PHONGBAN DEAN
WHERE MANV=TRPHG AND MAPHG=PHONG
AND HONV='Nguyen')
```





4. TRUY VẤN LÒNG – PHÂN CẤP

- Ví dụ 9: Tìm những nhân viên không có thân nhân nào

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

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



4. TRUY VẤN LÒNG – PHÂN CẤP

- Ví dụ 10: 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 )
```



4. TRUY VẤN LÒNG – PHÂN CẤP

- Ví dụ 11: 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 )
```



4. TRUY VẤN LÒNG – PHÂN CẤP

- Ví dụ 12: Tìm những trưởng phòng có tối thiểu một thân nhân

```
SELECT *  
FROM NHANVIEN  
WHERE MANV IN (SELECT MA_NVIEN FROM THANNHAN)  
AND MANV IN (SELECT TRPHG FROM PHONGBAN)
```



4. TRUY VẤN LÒNG – TƯƠNG QUAN

- Ví dụ:

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

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE EXISTS (
```

```
    SELECT *
```

```
    FROM PHONGBAN
```

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

tìm nhân viên mà có tênPHG='Nghien cuu' and
PHG=MAPHG



4. TRUY VẤN LÒNG – TƯƠNG QUAN

- Ví dụ 8: Tìm những nhân viên có người thân cùng tên và cùng giới tính với nhân viên đó

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




4. TRUY VẤN LÒNG – TƯƠNG QUAN

- Ví dụ 9: Tìm những nhân viên không có thân nhân nào

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



4. TRUY VẤN LÒNG – TƯƠNG QUAN

- Ví dụ 10: 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 NV2.PHG=4  
    AND NV1.LUONG>NV2.LUONG)
```



4. TRUY VẤN LÒNG – TƯƠNG QUAN

- Ví dụ 12: Tìm những trưởng phòng có tối thiểu một thân nhân

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



4. TRUY VẤN LÒNG – NHẬN XÉT

■ 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



4. PHÉP CHIA

- $R \div S$ 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

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



4. PHÉP CHIA

- $R \div S$ 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

```
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 ))
```



4. PHÉP CHIA

- Ví dụ 13: 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
 - R: PHANCONG(MA_NVIEN, SODA)
 - S: DEAN(MADA)
 - R:S: KQ(MA_NVIEN)



4. PHÉP CHIA

- Ví dụ 13: Tìm tên các nhân viên được phân công làm tất cả các đề án

```
SELECT NV.TENNV
FROM PHANCONG PC1, NHANVIEN NV
WHERE PC1.MA_NVIENT = NV.MANV
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 ))
```




4. HÀM KẾT HỢP

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



4. HÀM KẾT HỢP

- Ví dụ 14: Tìm tổng lương, lương cao nhất, lương thấp nhất và lương trung bình của các nhân viên

```
SELECT SUM(LUONG), MAX(LUONG), MIN(LUONG), AVG(LUONG)
FROM NHANVIEN
```

- Ví dụ 15: Cho biết số lượng nhân viên của phòng 'Nghien cuu'

```
SELECT COUNT(*) AS SL_NV
FROM NHANVIEN, PHONGBAN
WHERE PHG=MAPHG AND TENPHG='Nghien cuu'
```



4. GOM NHÓM

- Cú pháp:

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>

GROUP BY <danh sách các thuộc tính gom nhóm>

- Ví dụ 16: Cho biết số lượng nhân viên của từng phòng ban

```
SELECT PHG, COUNT(*) AS SL_NV  
FROM NHANVIEN  
GROUP BY PHG
```

```
SELECT TENPHG, COUNT(*) AS SL_NV  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY TENPHG
```



4. GOM NHÓM

- Ví dụ 17: Với mỗi nhân viên cho biết mã số, họ tên, số lượng đề án và tổng thời gian mà họ tham gia thực hiện các đề án

```
SELECT MA_NVIEN, COUNT(*) AS SL_DA, SUM(THOIGIAN) AS TONG_TG  
FROM PHANCONG  
GROUP BY MA_NVIEN
```

```
SELECT HONV, TENNV, COUNT(*) AS SL_DA, SUM(THOIGIAN) AS TONG_TG  
FROM PHANCONG, NHANVIEN  
WHERE MA_NVIEN=MANV  
GROUP BY MA_NVIEN, HONV, TENNV
```



4. GOM NHÓM

- Ví dụ 18: Cho biết những nhân viên tham gia từ 2 đề án trở lên
 - Tính số đề án của mỗi nhân viên
 - Chọn lại nhân viên có số đề án ≥ 2

➔ Điều kiện trên nhóm

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>

GROUP BY <danh sách các thuộc tính gom nhóm>

HAVING <điều kiện trên nhóm>



4. GOM NHÓM

- Ví dụ 18: Cho biết những nhân viên tham gia từ 2 đề án trở lên
 - Tính số đề án của mỗi nhân viên
 - Chọn lại nhân viên có số đề án ≥ 2

```
SELECT MA_NVIEN  
FROM PHANCONG  
GROUP BY MA_NVIEN  
HAVING COUNT(*)  $\geq$  2
```



4. GOM NHÓM

- Ví dụ 19: Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 20000

```
SELECT PHG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN  
GROUP BY PHG  
HAVING AVG(LUONG) > 20000
```

```
SELECT TENPHG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY TENPHG  
HAVING AVG(LUONG) > 20000
```



4. GOM NHÓM – 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
 - Điều kiện trên nhóm được thực hiện sau khi gom nhóm
 - Chỉ kiểm tra điều kiện trên nhóm, không là điều kiện lọc trên từng bộ



4. GOM NHÓM

- Ví dụ 13: Tìm tên các nhân viên được phân công làm tất cả các đề án

```
SELECT MANV, TENNVN
FROM NHANVIEN, PHANCONG
WHERE MANV=MA_NVN
GROUP BY MANV, TENNV
HAVING COUNT(*) = (SELECT COUNT(*) FROM DEAN )
```



4. GOM NHÓM

- Ví dụ 20: Tìm những phòng ban có lương trung bình cao nhất

```
SELECT PHG, AVG(LUONG) AS LUONG_TB
FROM NHANVIEN
GROUP BY PHG
HAVING AVG(LUONG) >= ALL (
        SELECT AVG(LUONG)
        FROM NHANVIEN
        GROUP BY PHG)
```



MỘT SỐ DẠNG TRUY VẤN KHÁC

5



5. TRUY VẤN CON Ở MỆNH ĐỀ **FROM**

- Kết quả trả về của một câu truy vấn phụ là một 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>



5. TRUY VẤN CON Ở MỆNH ĐỀ **FROM**

- Ví dụ 19: Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 20000

```
SELECT PHG, AVG(LUONG) AS LUONG_TB
FROM NHANVIEN
GROUP BY PHG
HAVING AVG(LUONG) > 20000
```

FROM NHANVIEN TEMP

```
SELECT PHG, TENPHG,
       (SELECT AVG(LUONG) AS LUONG_TB
        FROM NHANVIEN
         WHERE MAPHG=TEMP.PHG) AS TEMP
FROM NHANVIEN PHONGBAN
WHERE MAPHG=TEMP.PHG
GROUP BY PHG, TENPHG
HAVING AVG(LUONG) > 20000
```



5. 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 [OUTER] JOIN R2 ON <biểu thức>  
WHERE <điều kiện>
```



5. ĐIỀU KIỆN KẾT Ở MỆNH ĐỀ **FROM**

- Ví dụ 21: Tìm họ tên các nhân viên và tên các đề án mà nhân viên tham gia nếu có

```
SELECT NV.HONV + ' ' + NV.TENLOT + ' ' + NV.TENNV as N'Tên nhân viên', DA.TENDA  
FROM (PHANCONG PC JOIN DEAN DA ON SODA=MADA) RIGHT JOIN NHANVIEN  
NV ON PC.MA_NVIEN=NV.MANV
```

PHANCONG join DEAN

NHANVIEN

MA_NVIEN=MANV



mở rộng



5. CẤU TRÚC CASE

- Cho phép kiểm tra điều kiện và xuất thông tin theo từng trường hợp
- Cú pháp

```
CASE <tên cột>  
    WHEN <giá trị> THEN <biểu thức>  
    WHEN <giá trị> THEN <biểu thức>  
    ...  
    [ELSE <biểu thức>]  
END
```




5. CẤU TRÚC CASE

- Ví dụ 22: Cho biết họ tên các nhân viên đã đến tuổi về hưu (nam 60 tuổi, nữ 55 tuổi)

```
SELECT HONV, TENNV  
FROM NHANVIEN  
WHERE YEAR(GETDATE()) - YEAR(NGSINH) >= ( CASE PHAI  
                                           WHEN 'Nam' THEN 60  
                                           WHEN 'Nu' THEN 55  
                                           END )
```



5. CẤU TRÚC CASE

- Ví dụ 23: Cho biết họ tên các nhân viên và năm về hưu của họ

```
SELECT HONV, TENNV,  
(CASE PHAI  
    WHEN 'Nam' THEN YEAR(NGSINH) + 60  
    WHEN 'Nu' THEN YEAR(NGSINH) + 55  
    END ) AS NAMVEHUU  
FROM NHANVIEN
```



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>





THẢO LUẬN

