

SQL

Bộ phận Đào tạo

2019/09

- Subquery
- Logical Operators
- Intergrated Functions
- Keywords

Sample Database

groups
id: text
name: text
capacity: integer

id	name	capacity
fresher	Fresher	20
internship	Internship	20
tester	Tester	20

employees
id: text
name: text
group_id: text
salary: integer

id	name	group_id	salary
user1	User 1	fresher	350
user2	User 2	fresher	345
user3	User 3	internship	320
user4	User 4	dev	400

【Bí Mật】 Subquery

- **Subquery** là những câu *truy vấn phụ* được sử dụng để trả về những *dữ liệu được xem như điều kiện* cho câu truy vấn chính.

- Ví dụ:

SELECT id, name, salary **FROM** employees

WHERE salary >= **ALL** (**SELECT** salary **FROM** employees)

id	salary
user1	350
user2	345
user3	320
user4	400



id	name	salary
user4	User 4	400

[Bí Mật] Logical Operators

- **Logical Operators** (*Toán tử logic*) là những toán tử dùng để thực hiện các hành động so sánh và tính toán số học.
- Thông thường được sử dụng trong các câu **SELECT**, **UPDATE** và **DELETE**.
- Một số toán tử logic thông dụng:
 - **ALL, ANY**: So sánh 1 giá trị với 1 tập giá trị.
 - **AND, OR**: Kết hợp nhiều điều kiện.
 - **BETWEEN**: Tìm các giá trị trong 1 tập giá trị có giá trị đầu và cuối.
 - **EXISTS**: Tìm 1 dòng trong bảng thỏa mãn điều kiện.
 - **IN**: So sánh 1 giá trị với 1 danh sách dữ liệu.
 - **LIKE**: So sánh 1 giá trị với những giá trị thỏa mãn 1 pattern cho trước.
 - **NOT**: Đảo ngược 1 toán tử logic.

[Bí Mật] Operator – ANY

- **ANY** được sử dụng để so sánh một dữ liệu với những dữ liệu khác trong một danh sách, mỗi lần so sánh *chỉ cần ít nhất một giá trị trong danh sách* sao cho giá trị được so sánh thỏa mãn điều kiện.
- Ví dụ:

SELECT id, name, salary **FROM** employees

WHERE salary >= **ANY** (**SELECT** salary **FROM** employees)

id	salary
user1	350
user2	345
user3	320
user4	400



id	name	salary
user1	User 1	350
user2	User 2	345
user3	User 3	320
user4	User 4	400

【Bí Mật】 Operator – ALL vs ANY

SELECT id, name, salary **FROM** employees

WHERE salary >= **ALL** / **ANY**

(**SELECT** salary **FROM** employees)

id	salary
user1	350
user2	345
user3	320
user4	400

ALL

id	name	salary
user4	User 4	400

ANY

id	name	salary
user1	User 1	350
user2	User 2	345
user3	User 3	320
user4	User 4	400

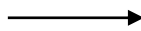
【Bí Mật】 Operator – BETWEEN

- **BETWEEN** được sử dụng để lọc ra dữ liệu cho câu truy vấn sao cho các dữ liệu được so sánh *nằm giữa hai giá trị đầu và cuối*.

- Ví dụ:

SELECT id, name, salary **FROM** employees
WHERE salary **BETWEEN** 325 **AND** 350

id	salary
user1	350
user2	345
user3	320
user4	400



id	name	salary
user1	User 1	350
user2	User 2	345

【Bí Mật】 Operator – IN

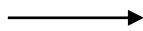
- **IN** được sử dụng để so sánh một giá trị với một danh sách cho sẵn sao cho giá trị được so sánh *trùng khớp với ít nhất một giá trị trong danh sách*.

- Ví dụ:

SELECT id, name, salary **FROM** employees

WHERE salary **IN** (300, 325, **350**, 375, **400**)

id	salary
user1	350
user2	345
user3	320
user4	400



id	name	salary
user1	User 1	350
user4	User 4	400

【Bí Mật】 Operator – LIKE

- **LIKE** được sử dụng để lọc ra các dữ liệu có giá trị *phù hợp với mẫu dữ liệu* sử dụng toán tử *Wildcard* trong câu truy vấn.
- **Wildcard** bao gồm 2 ký tự `_` và `%`,
 - `_` tương ứng với 1 ký tự
 - `%` tương ứng với nhiều ký tự
- Ví dụ:

```
SELECT id, name FROM groups  
WHERE id LIKE '%er'
```

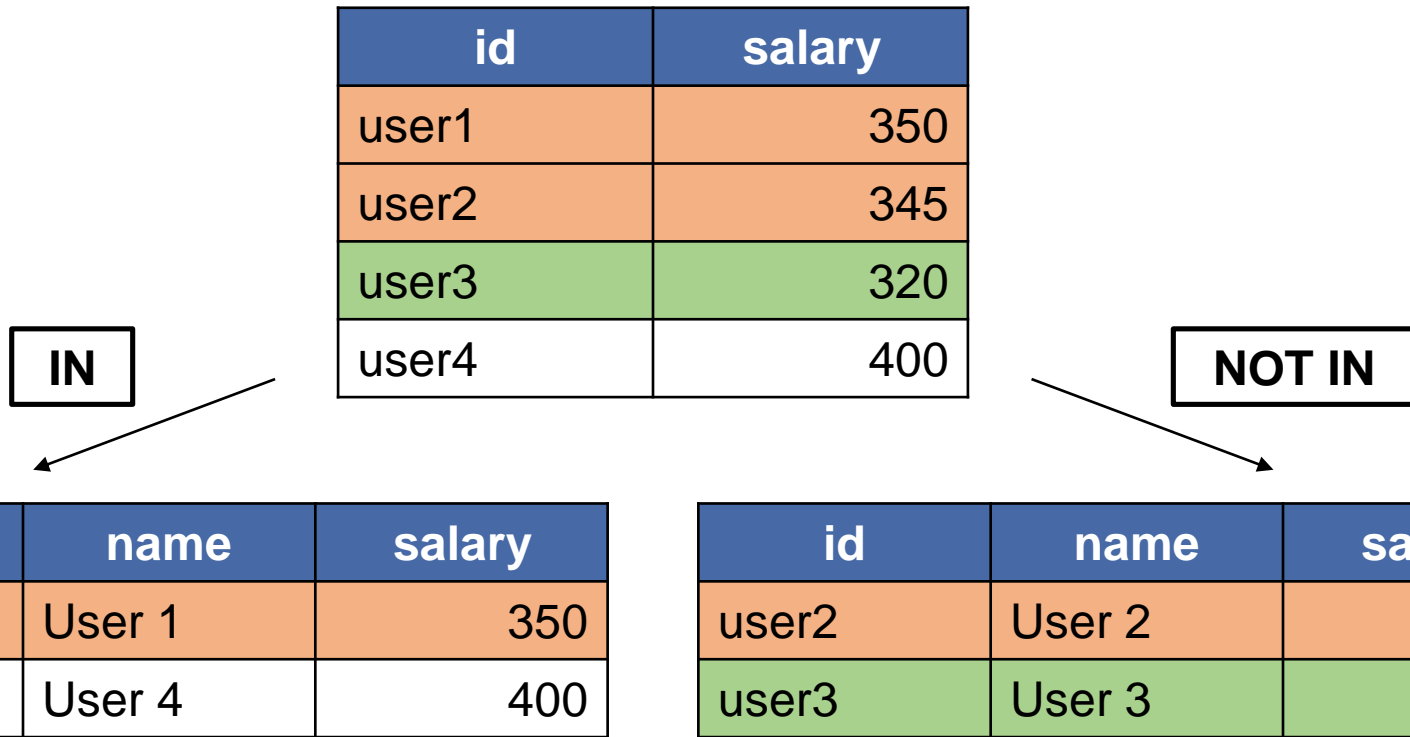
id	name
fresher	Fresher
internship	Internship
tester	Tester



id	name
fresher	Fresher
tester	Tester

【Bí Mật】 Operator – NOT

- **NOT** được sử dụng để đảo điều kiện của câu truy vấn.
- Ví dụ: **SELECT** id, name, salary **FROM** employees
WHERE salary (**NOT**) **IN** (300, 325, 350, 375, 400)



[Bí Mật] Integrated Functions

- **Integrated Functions** (*Chức năng tích hợp*) là những chức năng được tích hợp của hệ thống hỗ trợ cho việc xử lý dữ liệu cho các câu truy vấn.
- Một số chức năng phổ biến
 - **COUNT**: Đếm số dòng dữ liệu có trong bảng theo điều kiện nào đó.
 - **MAX, MIN, AVG**: Trả về 1 giá trị thỏa mãn trong 1 cột của 1 bảng nào đó.
 - **SUM**: Tính tổng của nhiều giá trị thỏa mãn điều kiện nào đó.
 - **CONCAT**: Ghép 2 chuỗi dữ liệu thành 1 chuỗi mới.
 - ...

【Bí Mật】 Function – COUNT

- **COUNT** được sử dụng để đếm số dòng dữ liệu của một cột thỏa một hay nhiều điều kiện nào đó.

- Ví dụ:

```
SELECT COUNT(id) AS total  
FROM employees  
WHERE salary < 400
```

- COUNT (*), COUNT (1)

id	salary
user1	350
user2	345
user3	320
user4	400
	250



total
3

【Bí Mật】 Function – SUM

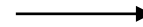
- **SUM** được sử dụng để tính tổng tất cả các dòng dữ liệu của một cột thỏa một hay nhiều điều kiện nào đó.

- Ví dụ:

SELECT SUM(salary) AS sum FROM employees

WHERE salary < 400

id	salary
user1	350
user2	345
user3	320
user4	400



sum
1015

【Bí Mật】 Function: AVG

- **AVG** được sử dụng để lấy giá trị trung bình của tất cả các dòng dữ liệu được chọn trong câu truy vấn.

- Ví dụ:

SELECT AVG(salary) AS average FROM employees

id	salary
user1	350
user2	345
user3	320
user4	400



average
353.75

$$(350+345+320+400) / 4 = 353.75 \Leftrightarrow \text{SUM} / \text{COUNT (column)} = \text{AVG}$$

【Bí Mật】 Function – CONCAT

- **CONCAT** được sử dụng để kết hai hay nhiều chuỗi text thành một chuỗi mới.

- Ví dụ:

SELECT CONCAT(name, ' - ', group_id) **AS** new_text **FROM** employees

name	group_id
User 1	fresher
User 2	fresher
User 3	internship
User 4	dev



new_text
User 1 - fresher
User 2 - fresher
User 3 - internship
User 4 - dev

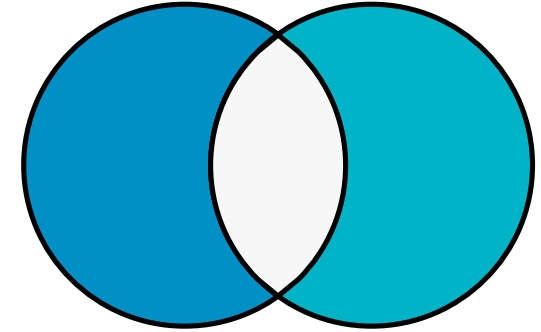
- **Keywords** (*Từ khóa*) là những từ khóa được dùng trong các câu truy vấn.
- Một số từ khóa thông dụng
 - **JOIN**: Kết các dữ liệu giữa các bảng lại với nhau thông qua một hoặc nhiều cột.
 - **ORDER BY**: Sắp xếp dữ liệu trong kết quả truy vấn theo tăng dần hoặc giảm dần.
 - **GROUP BY**: Gộp những dòng dữ liệu trong kết quả truy vấn thành các nhóm dữ liệu.
 - **DISTINCT**: Loại bỏ những dòng dữ liệu trùng trong kết quả truy vấn.
 - **UNION**: Nhóm các câu truy vấn lại với nhau.
 - ...

【Bí Mật】 Keyword – JOIN

- **JOIN** được sử dụng để kết dữ liệu giữa các bảng thông qua *một hoặc nhiều cột có dữ liệu tương ứng với nhau* để *lọc ra các dòng dữ liệu* cần thiết *thỏa mãn một hoặc nhiều điều kiện nào đó*.
- Tùy theo mục đích sử dụng mà **JOIN** có một số loại như sau:
 - **INNER JOIN**
 - **LEFT JOIN**
 - **RIGHT JOIN**
 - **FULL JOIN**
 - **CROSS JOIN**

【Bí Mật】 JOIN Type – INNER JOIN

INNER JOIN là kiểu kết dữ liệu chỉ lấy phần chung của 2 bảng dữ liệu.

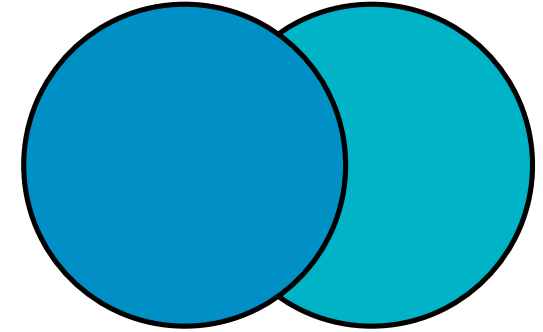


```
SELECT e.name AS emp,  
g.name AS group  
FROM groups g  
INNER JOIN employees e  
ON g.id = e.group_id
```

emp	group
User 1	Fresher
User 2	Fresher
User 3	Internship

【Bí Mật】 JOIN Type – LEFT JOIN

LEFT JOIN là kiểu kết dữ liệu lấy phần chung và toàn bộ dữ liệu còn lại của bảng bên trái (bảng đứng trước từ khóa **JOIN**).

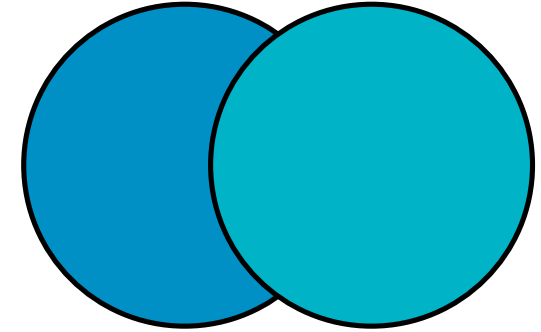


```
SELECT e.name AS emp,  
g.name AS group  
FROM groups g  
LEFT JOIN employees e  
ON g.id = e.group_id
```

emp	group
User 1	Fresher
User 2	Fresher
User 3	Internship
NULL	Tester

【Bí Mật】 JOIN Type – RIGHT JOIN

RIGHT JOIN là kiểu kết dữ liệu lấy phần chung và toàn bộ dữ liệu còn lại của bảng bên phải (bảng đứng sau từ khóa **JOIN**).

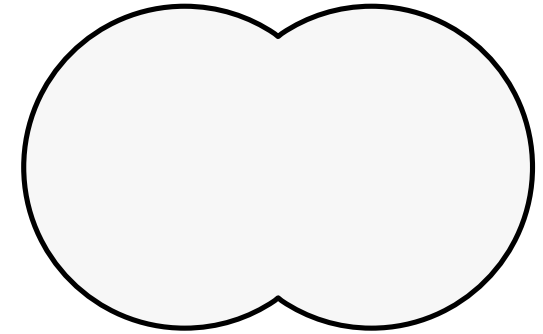


```
SELECT e.name AS emp,  
g.name AS group  
FROM groups g  
RIGHT JOIN employees e  
ON g.id = e.group_id
```

emp	group
User 1	Fresher
User 2	Fresher
User 3	Internship
User 4	NULL

【Bí Mật】 JOIN Type – FULL JOIN

FULL JOIN là kiểu kết dữ liệu lấy tất cả dữ liệu của cả 2 bảng.

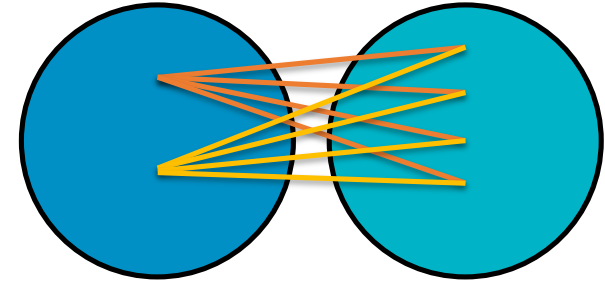


```
SELECT e.name AS emp,  
g.name AS group  
FROM groups g  
FULL JOIN employees e  
ON g.id = e.group_id
```

emp	group
User 1	Fresher
User 2	Fresher
User 3	Internship
User 4	NULL
NULL	Tester

[Bí Mật] JOIN Type – CROSS JOIN

CROSS JOIN là kiểu kết dữ liệu mà từng dòng của bảng bên trái kết với toàn bộ dòng của bảng bên phải.



```
SELECT  e1.name  AS  emp1,  
        e2.name AS  emp2  
  
FROM    employees e1  
  
CROSS JOIN employees e2
```

```
SELECT  e.name AS  emp, g.name  
        AS  group  
  
FROM    groups g, employees e
```

emp	group
User 1	Fresher
User 2	Fresher
User 3	Fresher
User 4	Fresher
User 1	Internship
User 2	Internship
User 3	Internship
User 4	Internship
...	...

[Bí Mật] Keyword – ORDER BY

- **ORDER BY** được sử dụng để sắp xếp dữ liệu từ kết quả truy vấn *theo một hay nhiều cột* nào đó với *thứ tự tăng dần hoặc giảm dần*.

- Cú pháp của **ORDER BY**:

ORDER BY Column1 [**ASC** / **DESC**, Column2...]

- Ví dụ: **SELECT** id, name **FROM** employees **ORDER BY** id **ASC**

id	name
user3	User 3
user1	User 1
user4	User 4
user2	User 2



id	name
user1	User 1
user2	User 2
user3	User 3
user4	User 4

【Bí Mật】 Keyword – GROUP BY

- **GROUP BY** được sử dụng để *nhóm lại các dòng dữ liệu* trong câu truy vấn **SELECT**.
Từ khóa này *đứng sau* **WHERE** và *đứng trước* **ORDER BY**.

- Cú pháp của **GROUP BY**:

[WHERE Conditions] GROUP BY Column1 [ORDER BY Column1...]

- Ví dụ: **SELECT** group_id **AS** position, **SUM**(salary) **AS** total **FROM** employees
GROUP BY position **ORDER BY** position **ASC**

id	salary
user1	350
user2	345
user3	320
user4	400



position	total
dev	400
fresher	695
internship	320

【Bí Mật】 Keyword – HAVING

- **HAVING** được sử dụng để *lọc dữ liệu* theo điều kiện nào đó *cho nhóm dữ liệu được nhóm lại* bởi **GROUP BY**. Và từ khóa này theo *sau* **GROUP BY** và *trước* **ORDER BY**.

- Cú pháp của **HAVING**:

[**GROUP BY** Column1...] **HAVING** Conditions [**ORDER BY** Column1...]

- Ví dụ: **SELECT** group_id **AS** position, **SUM**(salary) **AS** total **FROM** employees **GROUP BY** position **HAVING SUM**(salary) < 500 **ORDER BY** position

id	salary
user1	350
user2	345
user3	320
user4	400



position	amount
dev	400
internship	320

【Bí Mật】 Keyword – DISTINCT

- **DISTINCT** được sử dụng để loại bỏ những kết quả trùng lặp theo bộ khóa bao gồm tất cả các cột được **SELECT**.
- Cú pháp của **DISTINCT**:

SELECT DISTINCT Column1 [, Column2...]

- Ví dụ:

SELECT DISTINCT group_id **FROM** employees **ORDER BY** group_id

group_id	id	group_id
fresher	user1	dev
fresher	user2	fresher
internship	user3	internship
dev	user4	

Keyword – UNION

- **UNION** được sử dụng để kết dữ liệu kết quả của các câu truy vấn với nhau và bỏ đi những dòng dữ liệu bị trùng.
- Để sử dụng từ khóa **UNION**, các câu truy vấn phải:
 - Số cột được chọn là như nhau.
 - Kiểu dữ liệu của các cột tương ứng theo thứ tự là như nhau.
- Tương ứng với **UNION** còn có 1 số keyword như **INTERSECT**, **EXCEPT**.

```
SELECT id, name
FROM employees
UNION
SELECT id, name
FROM groups
```

id	name
fresher	Fresher
internship	Internship
tester	Tester

id	name
user1	User 1
user2	User 2
user3	User 3
user4	User 4



id	name
user1	User 1
user2	User 2
user3	User 3
user4	User 4
fresher	Fresher
internship	Internship
tester	Tester