

# Subquery

# Chuan bi

```
CREATE TABLE departments (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  department_name VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE employees (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(100) NOT NULL,  
  salary DECIMAL(10,2) NOT NULL,  
  department_id INT,  
  FOREIGN KEY (department_id) REFERENCES departments(id)  
);  
  
INSERT INTO departments (department_name) VALUES  
(  
  'Nhân sự',  
  'Công nghệ thông tin',  
  'Tài chính',  
  'Marketing');  
  
INSERT INTO employees (name, salary, department_id) VALUES  
(  
  'Nguyễn Văn An', 12000000, 1), -- Nhân sự  
  ('Trần Thị Bình', 15000000, 2), -- Công nghệ thông tin  
  ('Lê Quốc Cường', 13000000, 2), -- Công nghệ thông tin  
  ('Phạm Hồng Đăng', 11000000, 3), -- Tài chính  
  ('Hoàng Văn Nam', 14000000, NULL); -- Không có phòng ban
```

## Subquery - Truy vấn con

- ❖ **Subquery** (truy vấn con) là một câu lệnh SELECT được lồng trong một câu lệnh SQL khác như SELECT, INSERT, UPDATE, hoặc DELETE.
  - Có thể trả về một giá trị hoặc nhiều giá trị.
  - Dùng trong WHERE, HAVING, FROM, SELECT.... của truy vấn cha
  - Có thể là phụ thuộc vào truy vấn cha hoặc độc lập

## Subquery tra về **mot gia tri duy nhât**

### ❖ Tìm nhân viên có lương cao hơn mức lương trung bình

```
SELECT name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

- Subquery này độc lập với truy vấn cha
- **(SELECT AVG(salary) FROM employees)** chỉ trả về 1 giá trị thì mới so sánh được với salary (vế trái và vế phải có cùng số lượng)

Subquery tra ve **mot gia tri duy nhât**

❖ **Tìm khách hàng có tổng giá trị đơn hàng lớn nhất**

```
SELECT customer_id, customer_name
FROM customers
WHERE customer_id = (
    SELECT customer_id
    FROM orders
    GROUP BY customer_id
    ORDER BY SUM(total_amount) DESC
    LIMIT 1
);
```

## Subquery tra ve **mot gia tri duy nhât**

- ❖ Tìm phòng ban có lương **trung bình** nhân viên lớn hơn mức trung bình của **tất cả** nhân viên trong công ty
- ❖ VD: Trung bình lương cả công ty là 10 triệu. Phòng IT có lương trung bình 12 triệu

```
SELECT department_id, AVG(salary) AS avg_department
FROM employees
GROUP BY department_id
HAVING avg_department > (SELECT AVG(salary) FROM employees);
```

- **AVG(salary)** tính lương trung bình từng phòng ban
- **SELECT AVG(salary) FROM employees** tính lương trung bình cả công ty

## Subquery tra về **mot cot với nhiều dòng**

### ❖ Subquery trả về **một cột với nhiều dòng**

VD: Một phòng ban có 0-N nhân viên. Lọc ra các phòng ban **đã có** nhân viên  
(Có những phòng chưa có nhân viên)

```
SELECT *  
FROM departments  
WHERE dept_id IN (SELECT DISTINCT dept_id FROM employees);
```

➤ `SELECT DISTINCT dept_id FROM employees` lấy thông tin phòng từ nhân viên

## Subquery tra về nhiều dòng với nhiều cột

### ❖ Subquery trả về một dòng với nhiều cột

VD: Tìm những nhân viên có lương cao nhất **của mỗi phòng ban**. Trong 1 phòng ban, có nhiều nhân viên cùng lương cao nhất thì liệt kê tất cả

```
SELECT *  
FROM employees  
WHERE (department_id, salary) IN  
      (SELECT department_id, MAX(salary) FROM employees GROUP BY department_id);
```

| # | id | name           | salary      | department_id |
|---|----|----------------|-------------|---------------|
| 1 | 1  | Nguyễn Văn An  | 12000000.00 | 1             |
| 2 | 2  | Trần Thị Bình  | 15000000.00 | 2             |
| 3 | 4  | Phạm Hồng Đăng | 11000000.00 | 3             |



## Subquery tạo bảng ao

VD: Tìm những nhân viên có lương cao nhất **của mỗi phòng ban**. Trong 1 phòng ban, có nhiều nhân viên cùng lương cao nhất thì liệt kê tất cả

```
SELECT e.name, e.department_id, e.salary
FROM employees e
JOIN (SELECT department_id, MAX(salary) AS max_salary
      FROM employees
      GROUP BY department_id) as sub
ON e.department_id = sub.department_id AND e.salary = sub.max_salary;
```

hoặc

```
SELECT e.name, e.department_id, e.salary
FROM employees
WHERE (dept_id, salary) =
      (SELECT dept_id, MAX(salary) FROM employees GROUP BY dept_id);
```

## Subquery **phụ thuộc truy vấn cha**

**Tìm các nhân viên có lương thấp hơn lương trung bình **theo phòng ban của họ**.**

VD: Nhân viên A thuộc phòng **IT**, kiểm tra xem Nhân Viên A có lương thấp hơn lương trung bình của phòng **IT** hay không?

```
SELECT A.id , A.name
FROM Employee AS A
WHERE salary <
      (SELECT avg(salary) from Employee AS B WHERE A.department = B.department);
```

Đối với mỗi nhân viên A, sẽ tìm lương trung bình của tất cả nhân viên cùng phòng với nhân viên này ⇒ Hiệu năng **tệ**

## Subquery **phụ thuộc truy vấn cha**

VD: Hiển thị thời gian đặt hàng gần nhất của mỗi khách hàng (`order_date` lớn nhất)

```
SELECT
    c.customer_id,
    c.customer_name,
    (SELECT MAX(order_date)
     FROM orders o
     WHERE o.customer_id = c.customer_id) AS last_order_date
FROM customers c;
```

## Subquery **nhưng lưu ý**

- ❖ **Giới hạn kết quả:** Subquery trong SELECT phải trả về chính xác một giá trị (một hàng, một cột). Nếu trả về nhiều dòng sẽ gây lỗi.
- ❖ **Hiệu suất:** Subquery trong SELECT thường được thực thi cho mỗi dòng của truy vấn chính, có thể gây *giảm hiệu suất* với dữ liệu lớn.
- ❖ **Thay thế bằng JOIN:** Trong nhiều trường hợp, có thể **thay thế bằng JOIN** để cải thiện hiệu suất:
  - Subquery được thực thi lặp đi lặp lại cho **mỗi dòng** của truy vấn chính
  - JOIN được thực thi **một lần duy nhất** và tạo ra một tập kết quả hoàn chỉnh.
  - query optimizer được thiết kế để tối ưu hóa JOIN operations tốt hơn subquery

-- Thay vì:

```
SELECT
    c.customer_id,
    c.customer_name,
    (SELECT COUNT(*) FROM orders o WHERE o.customer_id = c.customer_id) AS order_count
FROM customers c;
```

-- Thay thế bằng JOIN

```
SELECT
    c.customer_id,
    c.customer_name,
    COUNT(tmp.order_id) AS order_count
FROM customers c
LEFT JOIN (
    select customer_id, order_id from orders o
) as tmp ON c.customer_id = tmp.customer_id
GROUP BY c.customer_id, c.customer_name
```

[https://sql.zcode.vn/contests/CSDL\\_SQL/questions/29a2b2cc9fca4a25a105ca5a162fda70](https://sql.zcode.vn/contests/CSDL_SQL/questions/29a2b2cc9fca4a25a105ca5a162fda70)

```
SET @a = (select round(avg(freight), 2) from SalesOrder);
```

```
select orderId, freight, round(freight - @a, 2) as diff_freight  
from SalesOrder SO  
where freight > (select avg(freight) from SalesOrder);
```

## Bai tap Subquery



[https://sql.zcode.vn/contests/CSDL\\_SQL/questions/0dca6545d4ca4cb3a5db6d6529dc8b31](https://sql.zcode.vn/contests/CSDL_SQL/questions/0dca6545d4ca4cb3a5db6d6529dc8b31)

[https://sql.zcode.vn/contests/CSDL\\_SQL/questions/29a2b2cc9fca4a25a105ca5a162fda70](https://sql.zcode.vn/contests/CSDL_SQL/questions/29a2b2cc9fca4a25a105ca5a162fda70)

[https://sql.zcode.vn/contests/CSDL\\_SQL/questions/2e33a40949e04a858f7f9264d37780f0](https://sql.zcode.vn/contests/CSDL_SQL/questions/2e33a40949e04a858f7f9264d37780f0)

[https://sql.zcode.vn/contests/CSDL\\_SQL/questions/59d1e8a7e1c64d62b941aef4a92cfd43](https://sql.zcode.vn/contests/CSDL_SQL/questions/59d1e8a7e1c64d62b941aef4a92cfd43)

[https://sql.zcode.vn/contests/CSDL\\_SQL/questions/f4bcac5001114c2fb76a4584872384b5](https://sql.zcode.vn/contests/CSDL_SQL/questions/f4bcac5001114c2fb76a4584872384b5)