

SQL Nâng Cao: Subqueries và Temporary Tables

Hoàng-Nguyên Vũ

1 Lý thuyết

1.1 Subqueries (Câu truy vấn lồng)

Định nghĩa

Subquery là một truy vấn SQL được đặt bên trong một truy vấn khác. Truy vấn con được thực thi trước và kết quả của nó được sử dụng trong truy vấn cha.

Các loại Subquery

1. **Scalar Subquery:** Trả về đúng một giá trị (1 dòng, 1 cột)
2. **Row Subquery:** Trả về một dòng có nhiều cột
3. **Table Subquery:** Trả về nhiều dòng và nhiều cột, thường sử dụng trong FROM

Vị trí sử dụng phổ biến

- Trong SELECT để tính toán động
- Trong WHERE hoặc HAVING để lọc dữ liệu
- Trong FROM như một bảng phụ (table subquery)
- Trong EXISTS, IN, ANY, ALL để so sánh tập dữ liệu

Ưu điểm

- Cho phép viết truy vấn một cách trực quan, dễ hiểu
- Giúp tái sử dụng logic xử lý phức tạp mà không cần viết nhiều truy vấn riêng biệt

Ví dụ minh họa

```
1 SELECT name
2 FROM customers
3 WHERE id IN (
4     SELECT customer_id
5     FROM orders
6     WHERE total > 1000
7 );
```

```
1 SELECT name ,
2       (SELECT SUM(total)
3        FROM orders
4        WHERE orders.customer_id = customers.id) AS total_spent
5 FROM customers;
```

```
1 SELECT *
2 FROM (
3       SELECT customer_id, SUM(total) AS total_spent
4       FROM orders
5       GROUP BY customer_id
6 ) AS spending
7 WHERE total_spent > 1000;
```

```
1 SELECT name
2 FROM customers c
3 WHERE EXISTS (
4       SELECT 1
5       FROM orders o
6       WHERE o.customer_id = c.id AND o.total > 1000
7 );
```

1.2 Temporary Tables (Bảng tạm thời)

Định nghĩa

Bảng tạm là bảng tồn tại trong thời gian phiên làm việc hiện tại (session). Sau khi kết thúc session hoặc đóng kết nối, bảng sẽ bị xóa tự động (tùy vào hệ quản trị cơ sở dữ liệu).

Đặc điểm

- Không tồn tại lâu dài trong cơ sở dữ liệu
- Dữ liệu được lưu trên ổ đĩa hoặc bộ nhớ RAM tùy vào hệ quản trị
- Có thể truy vấn, cập nhật, và xóa như bảng thường
- Thường được dùng để xử lý dữ liệu tạm trong ETL hoặc khi cần chia bước xử lý phức tạp

Cú pháp tạo bảng tạm

```
1 CREATE TEMPORARY TABLE temp_table_name AS
2 SELECT ...
3 FROM ...
4 WHERE ...;
```

Ưu điểm của Temporary Table

- Tăng hiệu suất khi xử lý dữ liệu nhiều bước
- Giảm độ phức tạp của truy vấn lồng
- Có thể tái sử dụng cho nhiều truy vấn trong cùng session

Ví dụ sử dụng TEMP TABLE

```
1 -- Bước 1: Tạo bảng tạm top_customers
2 CREATE TEMPORARY TABLE top_customers AS
3 SELECT customer_id, SUM(total) AS total_spent
4 FROM orders
5 GROUP BY customer_id
6 HAVING SUM(total) > 1000;
7
8 -- Bước 2: Truy vấn từ bảng tạm
9 SELECT c.name, t.total_spent
10 FROM customers c
11 JOIN top_customers t ON c.id = t.customer_id;
12
13 -- (Tùy chọn) Xóa bảng tạm nếu muốn
14 DROP TABLE top_customers;
```

Lưu ý

- Trong một số hệ quản trị như MySQL, bạn cần có quyền để tạo bảng tạm
- Với PostgreSQL, TEMP TABLE bị xóa sau khi session kết thúc
- TEMP TABLE không nên dùng khi cần lưu dữ liệu lâu dài hoặc chia sẻ giữa nhiều phiên

2 Dữ liệu mẫu

Bảng customers

id	name	city
1	Alice	Hanoi
2	Bob	Ho Chi Minh
3	Charlie	Hanoi

Bảng orders

id	customer_id	order_date	total
1	1	2024-01-10	500
2	1	2024-03-01	800
3	2	2024-02-20	1200
4	3	2024-01-15	200

Bảng products

id	name	price
1	Laptop	1500
2	Mouse	50
3	Keyboard	100
4	Monitor	300

Bảng order_items

id	order_id	product_id	quantity
1	1	2	2
2	1	3	1
3	2	1	1
4	2	4	1
5	3	1	1

Bảng employees

id	name	department
1	David	Sales
2	Emma	Support
3	Frank	Sales

Bảng order_assignments

id	order_id	employee_id
1	1	1
2	2	3
3	3	2

Bài tập: Subqueries và Temporary Tables**1. Subquery trong WHERE**

Tìm tất cả khách hàng đã từng mua đơn hàng chứa sản phẩm có giá lớn hơn 1000.

2. Subquery trong SELECT

Liệt kê tên từng khách hàng và tổng giá trị tất cả đơn hàng mà họ đã đặt.

3. Subquery trong FROM

Hiển thị **top 2 khách hàng** có tổng chi tiêu cao nhất (dựa trên cột **total** của đơn hàng).

4. Subquery kết hợp EXISTS

Tìm tên nhân viên thuộc phòng Sales đã từng xử lý đơn hàng có chứa sản phẩm tên "Monitor".

5. Subquery trong HAVING

Tìm các khách hàng có tổng chi tiêu cao hơn **mức trung bình** của tất cả khách hàng.

6. Tạo bảng tạm khách hàng VIP

Tạo bảng tạm **vip_customers** gồm những khách hàng có tổng giá trị đơn hàng lớn hơn 1000.

Sau đó, hiển thị tên và tổng chi tiêu của các khách hàng trong bảng tạm này.

7. Tạo bảng tạm sản phẩm phổ biến

Tạo bảng tạm **popular_products** gồm những sản phẩm được bán ra với tổng số lượng từ 2 trở lên.

Hiển thị tên và tổng số lượng bán ra của các sản phẩm này.

8. Subquery lồng nhau nhiều cấp

Liệt kê tên các khách hàng có đơn hàng được xử lý bởi nhân viên có tổng doanh số lớn hơn 1000.

9. Kết hợp Temp Table và Subquery

- a. Tạo bảng tạm **high_value_orders** gồm những đơn hàng có tổng tiền > 800.

b. Hiển thị tên khách hàng của các đơn hàng đó.

10. **Subquery tính tổng giá trị đơn hàng**

Tính tổng giá trị thực tế của từng đơn hàng bằng cách lấy `price × quantity` theo từng dòng sản phẩm.