Với Câu lệnh EXPLAIN ANALYZE: phân tích chiến lược thực thi của một câu lệnh sql:

Vd:

Seq Scan on questions (cost=0.00..45.00 rows=10 width=64) (actual time=0.050..0.150 rows=10 loops=1)

* Trong đó:
  + **Tên node**:
    - Seq Scan: Quét tuần tự qua tất cả các hàng trong bảng.
    - Index Scan: Sử dụng chỉ mục để tìm các hàng phù hợp.
    - Nested Loop, Hash Join, Merge Join: Các loại JOIN.
    - Sort: Sắp xếp dữ liệu.
    - Aggregate: Tính toán tổng hợp (SUM, COUNT, AVG...).
  + **Cost**, Chi phí ước tính thực thi của node này bao gồm:
    - Startup cost: Chi phí để bắt đầu thực hiện bước này.
    - Total cost: Chi phí để hoàn thành toàn bộ bước này, bao gồm Startup cost.
    - Ví dụ:
      * (cost=0.00..45.00 rows=10 width=64)
      * Trong đó:
        + 0.00: Chi phí khởi tạo.
        + 45.00: Chi phí ước tính hoàn tất.
        + rows=10: Số hàng PostgreSQL ước tính sẽ trả về.
        + width=64: Kích thước trung bình (byte) của mỗi hàng.
  + **Actual Time**, thời gian thực tế thực hiện mỗi node
    - actual time=start\_time..end\_time: Thời gian tính bằng milliseconds (ms) từ lúc bắt đầu đến lúc kết thúc nút.
    - rows: Số hàng thực tế được trả về tại nút này.
    - loops: Số lần lặp lại nút này (quan trọng nếu truy vấn có JOIN hoặc lặp qua nhiều tập dữ liệu).
    - Ví dụ:
      * (actual time=0.050..0.150 rows=10 loops=1)
        + 0.050: Thời gian bắt đầu.
        + 0.150: Thời gian kết thúc.
        + rows=10: Số hàng trả về.
        + loops=1: Nút này được thực thi 1 lần.
  + **Node Relationship:**
    - **Parent Node**: Nút phía trên phụ thuộc vào dữ liệu từ nút bên dưới.
    - **Child Node**: Cung cấp dữ liệu cho nút phía trên.
    - **Ví dụ**:

|  |
| --- |
| * Hash Join (cost=50.00..75.00 rows=100 width=128) (actual time=0.500..0.750 rows=100 loops=1) * Hash Cond: (q.subject\_id = s.id) * -> Seq Scan on questions q (cost=0.00..25.00 rows=1000 width=64) (actual time=0.050..0.200 rows=1000 loops=1) * -> Hash (cost=25.00..25.00 rows=100 width=64) (actual time=0.300..0.300 rows=100 loops=1) * Buckets: 1024 Batches: 1 Memory Usage: 48kB |

Trong đó:

Hash join là cha

Seq Scan và Hash là con

* **Filter/Conditions,** Nếu có lọc dữ liệu, EXPLAIN ANALYZE sẽ chỉ ra các điều kiện áp dụng trong bước đó:
  + Vd: Filter: (content ~~ '%xanh%')
* **Các Node Phổ Biến**:

|  |  |
| --- | --- |
| **Node** | **Mô tả** |
| Seq Scan | Quét tuần tự toàn bộ bảng. Hiệu suất kém nếu bảng lớn và không có chỉ mục phù hợp. |
| Index Scan | Quét các hàng phù hợp dựa trên chỉ mục. Hiệu quả hơn khi dữ liệu được lọc tốt. |
| Index Only Scan | Giống Index Scan, nhưng chỉ đọc chỉ mục, không cần truy cập bảng. |
| Bitmap Heap Scan | Sử dụng chỉ mục để tạo bitmap (bản đồ bit) và truy cập nhiều hàng từ bảng theo lô. |
| Sort | Sắp xếp dữ liệu. Có thể chậm nếu không đủ bộ nhớ (work\_mem). |
| Hash Join | Thực hiện JOIN bằng cách tạo bảng băm trên một bảng nhỏ hơn. Hiệu quả cho bảng không sắp xếp. |
| Merge Join | Thực hiện JOIN với hai bảng đã được sắp xếp. |
| Nested Loop | Lặp qua từng hàng của một bảng, kết hợp với bảng khác. Chậm nếu số hàng lớn. |

**Nhận biết điểm nghẽn:**

* Seq Scan trên bảng lớn: Có thể cần tạo chỉ mục.
* **High Rows Removed by Filter**: Xem xét di chuyển điều kiện lọc sớm hơn hoặc dùng chỉ mục.
* **Sort hoặc Hash Join chậm**: Tăng work\_mem nếu bộ nhớ không đủ.
* **High Loops**: Nếu số vòng lặp lớn, cải thiện logic JOIN hoặc sử dụng chỉ mục phù hợp.

**Ví dụ chi tiết:**

Truy vấn:

|  |
| --- |
| EXPLAIN ANALYZE  SELECT q.id, q.content  FROM questions q  WHERE q.content LIKE '%xanh%'  ORDER BY q.content DESC  LIMIT 10; |

Kết quả:

|  |
| --- |
| Limit (cost=125.00..125.25 rows=10 width=64) (actual time=1.230..1.250 rows=10 loops=1) -> Sort (cost=125.00..150.00 rows=1000 width=64) (actual time=1.220..1.230 rows=10 loops=1) Sort Key: q.content DESC Sort Method: quicksort Memory: 25kB -> Seq Scan on questions q (cost=0.00..100.00 rows=1000 width=64) (actual time=0.010..0.900 rows=1200 loops=1) Filter: (content ~~ '%xanh%') Rows Removed by Filter: 1800 |