

BÁO CÁO PHÂN TÍCH CỘT DỮ LIỆU TIỀM NĂNG CHO VIỆC TẠO INDEX TRONG POSTGRESQL

Dự án: TheShoe - Website Thương mại điện tử Bán Giày

Ngày: 01 tháng 04 năm 2025

1. Giới thiệu

Tài liệu này trình bày kết quả phân tích các cột trong cơ sở dữ liệu PostgreSQL của dự án TheShoe, nhằm xác định những cột thường xuyên được sử dụng trong các điều kiện lọc (WHERE), nối (JOIN), hoặc sắp xếp (ORDER BY). Mục tiêu của việc phân tích này là đề xuất danh sách các cột nên được tạo index để tối ưu hóa hiệu suất truy vấn, giảm thời gian phản hồi của hệ thống, đặc biệt là các chức năng quan trọng như tìm kiếm sản phẩm, xem đơn hàng, và áp dụng khuyến mãi.

Phân tích này dựa trên các tài liệu sau:

- Software Requirement Specification (SRS) for Shoe E-commerce Website (TheShoe.pdf)
- Phân tích ERD hệ thống web bán giày (Phân Tích ERD Hệ Thống Web Bán Giày (Report).pdf)
- Danh sách Stored Procedures và Functions (StoreProcedure_Function.pdf)
- Quy trình mua hàng tổng hợp (QuyTrinhMuaHang.pdf)

2. Phương pháp Phân tích

Quá trình xác định các cột tiềm năng cho việc tạo index được thực hiện thông qua các bước sau:

- Phân tích Yêu cầu Chức năng (SRS & Quy trình mua hàng):** Xác định các hoạt động của người dùng (khách hàng, admin, shipper) và hệ thống liên quan đến việc truy xuất, lọc, và tìm kiếm dữ liệu (ví dụ: tìm kiếm sản phẩm, xem lịch sử đơn hàng, kiểm tra trạng thái, áp dụng mã giảm giá, thống kê).
- Phân tích Sơ đồ Quan hệ Thực thể (ERD):** Đối chiếu các hoạt động đã xác định với cấu trúc cơ sở dữ liệu, xác định các bảng và cột liên quan, đặc biệt là các khóa chính (PK), khóa ngoại (FK), và các cột dùng để lưu trữ trạng thái, ngày tháng, hoặc thông tin tìm kiếm.
- Phân tích Stored Procedures/Functions:** Xem xét mục đích và các tham số đầu vào/ra của các SP và Function hợp lệ (và cả các SP/Function cần điều chỉnh chứa logic nghiệp vụ) để nhận diện các cột được dùng trong điều kiện lọc hoặc cập nhật.
- Tổng hợp và Đánh giá:** Tập hợp danh sách các cột thường xuyên xuất hiện trong các điều kiện truy vấn (WHERE, JOIN ON, ORDER BY) và đánh giá mức độ ưu tiên dựa trên tần suất sử dụng và tầm quan trọng của chức năng liên quan.

3. Kết quả Phân tích: Các Cột Tiềm năng cho Indexing

Dưới đây là danh sách các cột được xác định là ứng viên tiềm năng cho việc tạo index, được nhóm theo từng bảng:

- Bảng User:**
 - id (PK): Index tự động, truy xuất người dùng cụ thể.

- Email, sodienthoai(UK): Dùng cho đăng nhập, tìm kiếm duy nhất, kiểm tra tồn tại.
- role / role_id (nếu dùng bảng UserRole): Lọc/phân loại người dùng theo vai trò.
- **Bảng Product:**
 - id (PK): Index tự động, truy xuất sản phẩm cụ thể.
 - name: Tìm kiếm sản phẩm (Cân nhắc Full-Text Index).
 - category_id (FK): Lọc/Join với bảng Category.
 - price: Lọc sản phẩm theo khoảng giá.
 - stock_quantity: kiểm tra tồn kho
- **Bảng Category:**
 - id (PK): Index tự động.
 - Name: dùng để tìm kiếm hoặc lọc theo danh mục
- **Bảng Order:**
 - id (PK): Index tự động, truy xuất đơn hàng cụ thể.
 - user_id (FK, Nullable): Lọc đơn hàng theo người dùng (quan trọng cho lịch sử đơn hàng).
 - status: Lọc đơn hàng theo trạng thái (rất thường xuyên).
 - created_at: Lọc đơn hàng theo khoảng thời gian (thống kê, xem đơn hàng gần đây).
 - discount_code_id (JOIN với bảng DiscountCode)
- **Bảng OrderDetail (Trung gian):**
 - order_id (PK, FK): Index tự động, Join với Order.
 - product_id (PK, FK): Index tự động, Join với Product, thống kê sản phẩm.
- **Bảng Cart:**
 - user_id (FK): Truy xuất giỏ hàng của người dùng.
- **Bảng CartItem (Trung gian):**
 - cart_id (PK, FK): Index tự động, Join với Cart.
 - product_id (PK, FK): Index tự động, Join với Product.
- **Bảng Address:**
 - user_id (FK): Truy xuất địa chỉ của người dùng.
 - is_default: Lọc địa chỉ mặc định.
- **Bảng Payment:**
 - order_id (FK): Join với Order.
 - status: Kiểm tra/lọc trạng thái thanh toán.
- **Bảng Shipping:**
 - order_id (FK): Join với Order.
 - shipper_id (FK, Nullable): Lọc đơn hàng theo shipper.
 - status: Lọc theo trạng thái giao hàng.
 - address_id (FK): Join với Address.
- **Bảng DiscountCode:**
 - code (UK): Kiểm tra/áp dụng mã (rất quan trọng).
 - start_date, end_date: Kiểm tra hiệu lực mã theo thời gian.
- **Bảng Promotion:**
 - start_date, end_date: Kiểm tra hiệu lực khuyến mãi.
- **Bảng PromotionProduct (Trung gian):**
 - promotion_id (PK, FK): Index tự động.
 - product_id (PK, FK): Index tự động.

- **Bảng Review:**
 - product_id (FK): Lấy đánh giá theo sản phẩm.
 - user_id (FK): Lấy đánh giá theo người dùng.
- **Bảng Wishlist:**
 - user_id (FK): Lấy danh sách mong muốn của người dùng.
- **Bảng Role, Permission, UserRole, RolePermission:**
 - Các khóa chính (id) và khóa ngoại (user_id, role_id, permission_id) trong các bảng này và bảng trung gian: Quan trọng cho việc Join kiểm tra quyền.

4. Đề xuất và Ưu tiên

Dựa trên phân tích, đề xuất các hành động sau:

1. **Ưu tiên hàng đầu: Tạo Index cho tất cả các cột Khóa Ngoại (FK).** PostgreSQL không tự động tạo index cho FK, và việc này cực kỳ quan trọng để tăng tốc các thao tác JOIN và các truy vấn lọc dựa trên mối quan hệ (ví dụ: lấy tất cả đơn hàng của một người dùng).
2. **Tạo Index cho các cột lọc thường xuyên:**
 - Các cột trạng thái: Order.status, Payment.status, Shipping.status.
 - Cột định danh duy nhất nghiệp vụ: DiscountCode.code, User.email.
 - Cột dùng cho tìm kiếm/phân loại chính: Product.name (cân nhắc Full-Text), Product.category_id.
 - Cột ngày tháng dùng cho lọc phạm vi: Order.created_at, DiscountCode.start_date, DiscountCode.end_date.
3. **Xem xét Composite Index:** Nếu có các truy vấn thường xuyên lọc trên nhiều cột cùng lúc (ví dụ: WHERE user_id = ? AND status = ? trong bảng Order), cân nhắc tạo index kết hợp trên các cột đó (INDEX ON orders (user_id, status)).

5. Lưu ý Quan trọng

- **PK và UK:** PostgreSQL tự động tạo index cho các ràng buộc Khóa chính và Duy nhất.
- **Kiểm tra thực tế:** Danh sách trên là dựa vào phân tích tài liệu. Bước quan trọng tiếp theo là sử dụng EXPLAIN ANALYZE trên các truy vấn thực tế của ứng dụng để xác nhận các điểm nghẽn và đo lường hiệu quả của index sau khi tạo.
- **Chi phí Index:** Việc tạo index không miễn phí. Index chiếm dụng không gian đĩa và làm chậm các thao tác ghi (INSERT, UPDATE, DELETE) vì cần cập nhật cả index. Do đó, chỉ nên tạo index cho các cột thực sự cần thiết và mang lại lợi ích rõ ràng cho các truy vấn đọc.
- **Cardinality:** Index trên cột có ít giá trị riêng biệt (low cardinality) như cột status có thể hiệu quả nếu truy vấn thường lọc lấy một phần nhỏ dữ liệu (ví dụ: lấy các đơn hàng "chờ xử lý").

6. Kết luận

Việc tạo index một cách chiến lược dựa trên các cột được xác định trong báo cáo này có tiềm năng cải thiện đáng kể hiệu suất của cơ sở dữ liệu và trải nghiệm người dùng trên trang web TheShoe. Tuy nhiên, cần thực hiện việc tạo index một cách cẩn thận và luôn kiểm tra, đo lường hiệu quả thực tế trên môi trường thử nghiệm trước khi áp dụng lên môi trường production.