# Báo cáo: Đề xuất Prepared Statements để Tối ưu Hiệu suất Truy vấn cho Hệ thống TheShoe

**Dựa trên:** Tài liệu Yêu cầu Hệ thống (SRS) TheShoe và Phân tích Sơ đồ Quan hệ Thực thể (ERD).
**Mục đích:** Tài liệu này liệt kê các truy vấn cơ sở dữ liệu thường xuyên được thực thi trong hệ thống TheShoe và đề xuất sử dụng **Prepared Statements** (Truy vấn Chuẩn bị Sẵn) để tối ưu hóa hiệu suất, giảm thời gian phân tích cú pháp và lập kế hoạch truy vấn, đặc biệt hữu ích trong môi trường có lưu lượng truy cập cao.

## Danh sách Truy vấn Thường Xuyên và Prepared Statements Đề xuất

Dưới đây là danh sách các truy vấn được xác định là thường xuyên lặp lại, cùng với các Prepared Statements tương ứng được đề xuất:

### 1. Truy vấn thông tin Người Dùng

- **Truy vấn thông tin từ id, email, sodienthoai**
    - **Mục đích:** Xác thực thông tin (email/số điện thoại/ID) của người dùng.
    - **Tần suất:** Rất cao (mỗi lần người dùng thực hiện đăng nhập).
    - **Tham số:** id, email, sodienthoai
- **Lấy danh sách địa chỉ:**
    - **Mô tả**: Lấy các địa chỉ đã lưu của người dùng.
    - **Tần suất:** Rất cao (mỗi lần người dùng mua hàng)
    - **Tham số**: user_id

### 2. Truy vấn lấy thông tin Sản Phẩm

- **Truy vấn tìm kiếm sản phẩm**
    - **Mục đích:** Cho phép người dùng tìm kiếm sản phẩm dựa trên tên, mô tả, danh mục hoặc khoảng giá.
    - **Tần suất:** Cao (người dùng thường xuyên sử dụng chức năng tìm kiếm).
    - **Prepared Statement:**
      PREPARE search_products (TEXT, UUID, DECIMAL, DECIMAL) AS
      SELECT *
      FROM "Product"
      WHERE

```
    (name ILIKE '%' || $1 || '%' OR description ILIKE '%' || $1 || '%')
    AND ($2 IS NULL OR category_id = $2)
    AND ($3 IS NULL OR price >= $3)
    AND ($4 IS NULL OR price <= $4)
  ORDER BY created_at DESC;
```

- **Cập nhật tồn kho:**
  - Mô tả: Thay đổi số lượng tồn kho.
  - **Tần suất: Cao(mỗi khi người dùng mua hàng)**
  - Tham số: quantity_change, product_id
- **Lấy đánh giá sản phẩm:**
  - Mô tả: Hiển thị đánh giá của sản phẩm.
  - Tần suất: Cao (mỗi khi người dùng xem chi tiết sản phẩm)
  - Tham số: product_id

## 3. Truy vấn Lấy Đơn Hàng Theo Trạng Thái

- **Mục đích:** Lấy danh sách đơn hàng, có thể lọc theo người dùng cụ thể hoặc theo trạng thái đơn hàng (ví dụ: "đang giao hàng", "đã hoàn thành").
- **Tần suất:** Trung bình (sử dụng bởi cả người dùng xem lịch sử và quản trị viên quản lý đơn hàng).
- **Prepared Statement:**

```
PREPARE get_orders_by_status (UUID, VARCHAR) AS
 SELECT *
 FROM "Order"
 WHERE
  (user_id = $1 OR $1 IS NULL) -- $1 là NULL nếu truy vấn bởi admin
  AND status = $2;
```

## 3.1. Xử lý Đơn hàng (Order Processing)

- **Tạo đơn hàng:**
  - Mô tả: Lưu thông tin đơn hàng mới.
  - SQL Template: INSERT INTO "Order" (user_id, status, total_amount, ...) VALUES ($1, $2, $3, ...) RETURNING id;
  - Tần suất: Cao (Khi người dùng tạo đơn hàng)
  - Tham số: user_id, status, total_amount, ...
- **Thêm chi tiết đơn hàng:**
  - Mô tả: Lưu từng sản phẩm trong đơn hàng (lặp lại nhiều lần).
  - SQL Template: INSERT INTO OrderDetail (order_id, product_id, quantity, price_at_purchase) VALUES ($1, $2, $3, $4);
  - Tần suất: Cao (khi người dùng tạo đơn hàng)
  - Tham số: order_id, product_id, quantity, price_at_purchase
- **Cập nhật trạng thái đơn hàng:**
  - Mô tả: Thay đổi trạng thái đơn hàng.

    - SQL Template: UPDATE "Order" SET status = $1 WHERE id = $2;
    - Tần suất: Cao (Khi người dùng tạo đơn hàng)
    - Tham số: new_status, order_id
- **Lấy đơn hàng của người dùng:**
    - Mô tả: Xem lịch sử mua hàng.
    - SQL Template: SELECT id, status, total_amount, created_at FROM "Order" WHERE user_id = $1 ORDER BY created_at DESC;
    - Tần suất: Cao (Khi người dùng xem lại đơn hàng)
    - Tham số: user_id
- **Lấy chi tiết một đơn hàng:**
    - Mô tả: Xem thông tin đầy đủ của đơn hàng.
    - SQL Template (Order Info): SELECT * FROM "Order" WHERE id = $1;
    - SQL Template (Order Details): SELECT od.product_id, od.quantity, od.price_at_purchase, p.name FROM OrderDetail od JOIN Product p ON od.product_id = p.id WHERE od.order_id = $1;
    - Tần suất: Cao (Khi người dùng tạo xem chi tiết đơn hàng)
    - Tham số: order_id

## 3.2. Thanh toán và Giao hàng (Payment & Shipping)

- **Tạo bản ghi thanh toán:**
    - Mô tả: Lưu thông tin giao dịch.
    - SQL Template: INSERT INTO Payment (order_id, amount, method, status) VALUES ($1, $2, $3, $4);
    - Tham số: order_id, amount, method, status
- **Cập nhật trạng thái thanh toán:**
    - Mô tả: Đánh dấu thành công/thất bại.
    - SQL Template: UPDATE Payment SET status = $1 WHERE order_id = $2;
    - Tham số: new_status, order_id
- **Tạo bản ghi giao hàng:**
    - Mô tả: Lưu thông tin vận chuyển.
    - SQL Template: INSERT INTO Shipping (order_id, address_id, status) VALUES ($1, $2, $3);
    - Tham số: order_id, address_id, initial_status
- **Cập nhật trạng thái giao hàng:**
    - Mô tả: Thay đổi trạng thái hoặc gán shipper.
    - SQL Template (Assign Shipper): UPDATE Shipping SET shipper_id = $1, status = $2 WHERE order_id = $3;
    - SQL Template (Update Status): UPDATE Shipping SET status = $1 WHERE id = $2;
    - Tham số: shipper_id, new_status, order_id, shipping_id

## 4. Truy vấn Thao Tác Giỏ Hàng (Thêm/Xóa Sản Phẩm)

- **Mục đích:** Cho phép người dùng thêm sản phẩm vào giỏ hàng hoặc xóa sản phẩm khỏi giỏ hàng.
- **Tần suất:** Cao (người dùng tương tác thường xuyên với giỏ hàng).
- **Prepared Statements:**

```
-- Thêm sản phẩm vào giỏ (hoặc cập nhật số lượng nếu đã tồn tại)
PREPARE add_to_cart (UUID, UUID, INT) AS
  INSERT INTO "CartItem" (cart_id, product_id, quantity)
  VALUES ($1, $2, $3)
  ON CONFLICT (cart_id, product_id) DO UPDATE
  SET quantity = "CartItem".quantity + EXCLUDED.quantity;

-- Xóa sản phẩm khỏi giỏ
PREPARE remove_from_cart (UUID, UUID) AS
  DELETE FROM "CartItem"
  WHERE cart_id = $1 AND product_id = $2;
```

- **Cập nhật sản phẩm trong giỏ:**
    - Mô tả: Thêm mới hoặc tăng số lượng
    - Tần suất: Cao (khi người dùng chỉnh sửa số lượng sản phẩm trong giỏ hàng)
    - Tham số: cart_id, product_id, quantity
- **Lấy các mặt hàng trong giỏ:**
    - Mô tả: Hiển thị giỏ hàng.
    - Tần suất: Cao (Khi người dùng yêu cầu xem giỏ hàng)
    - Tham số: cart_id

## 5. Truy vấn Kiểm Tra Tồn Kho

- **Mục đích:** Kiểm tra xem số lượng sản phẩm mong muốn có còn đủ trong kho hay không, thường thực hiện trước khi thêm vào giỏ hoặc tiến hành thanh toán.
- **Tần suất:** Cao.
- **Prepared Statement:**

```
PREPARE check_stock (UUID, INT) AS
  SELECT stock_quantity >= $2 AS is_available
  FROM "Product"
  WHERE id = $1;
```

## 6. Truy vấn Cập Nhật Trạng Thái Đơn Hàng

- **Mục đích:** Cho phép quản trị viên hoặc hệ thống tự động cập nhật trạng thái của một đơn hàng (ví dụ: từ "chờ xử lý" sang "đang giao").
- **Tần suất:** Trung bình.
- **Prepared Statement:**

```
PREPARE update_order_status (UUID, VARCHAR) AS
  UPDATE "Order"
  SET status = $2, updated_at = CURRENT_TIMESTAMP
  WHERE id = $1;
```

## 7. Truy vấn Áp Dụng Mã Giảm Giá

- **Mục đích:** Kiểm tra tính hợp lệ của mã giảm giá (còn hạn, đủ điều kiện giá trị đơn hàng tối thiểu, chưa hết lượt sử dụng) và cập nhật số lần đã sử dụng nếu hợp lệ.
- **Tần suất:** Trung bình (khi người dùng nhập mã giảm giá lúc thanh toán).
- **Prepared Statement:**

```
PREPARE apply_discount_code (VARCHAR, DECIMAL) AS
  UPDATE "DiscountCode"
  SET uses_count = uses_count + 1
  WHERE
    code = $1
    AND CURRENT_DATE BETWEEN start_date AND end_date
    AND (max_uses IS NULL OR uses_count < max_uses)
    AND (min_order_value IS NULL OR $2 >= min_order_value)
  RETURNING discount_percentage; -- Trả về % giảm giá nếu thành công
```

## 8. Truy vấn Lấy Địa Chỉ Mặc Định Của Người Dùng

- **Mục đích:** Lấy thông tin địa chỉ giao hàng mặc định của người dùng để hiển thị sẵn trong quá trình thanh toán.
- **Tần suất:** Cao (mỗi khi người dùng vào trang thanh toán).
- **Prepared Statement:**

```
PREPARE get_default_address (UUID) AS
  SELECT *
  FROM "Address"
  WHERE user_id = $1 AND is_default = TRUE;
```

## 9. Truy vấn Thống Kê Sản Phẩm Bán Chạy

- **Mục đích:** Cung cấp thông tin về các sản phẩm bán chạy nhất, có thể lọc theo danh mục hoặc khoảng thời gian, phục vụ cho việc báo cáo của quản trị viên.
- **Tần suất:** Trung bình.
- **Prepared Statement:**

```
PREPARE get_top_products (UUID, DATE, DATE) AS
  SELECT p.id, p.name, SUM(od.quantity) AS total_sold
  FROM "OrderDetail" od
  JOIN "Product" p ON od.product_id = p.id
```

```
JOIN "Order" o ON od.order_id = o.id
WHERE
  ($1 IS NULL OR p.category_id = $1) -- $1 là category_id, NULL nếu không lọc
  AND o.created_at BETWEEN $2 AND $3 -- $2 là ngày bắt đầu, $3 là ngày kết thúc
GROUP BY p.id, p.name -- Thêm p.name vào GROUP BY
ORDER BY total_sold DESC
LIMIT 10; -- Lấy top 10 sản phẩm
```

### 10. Truy vấn Đăng Ký Người Dùng Mới

- **Mục đích:** Tạo một tài khoản người dùng mới trong hệ thống, đồng thời tự động tạo giỏ hàng liên kết với người dùng đó.
- **Tần suất:** Trung bình (khi có người dùng mới đăng ký).
- **Prepared Statement:**

```
PREPARE register_user (VARCHAR, VARCHAR, VARCHAR, VARCHAR) AS
  WITH new_user AS (
    INSERT INTO "User" (name, email, sodienthoai, password)
    VALUES ($1, $2, $3, $4) -- $1: name, $2: email, $3: sodienthoai, $4: password hash
    RETURNING id
  )
  INSERT INTO "Cart" (user_id)
  SELECT id FROM new_user;
  -- Lưu ý: Việc tạo Wishlist có thể cần một bước riêng hoặc trigger
```

## Lưu Ý Khi Triển Khai Prepared Statements

1. **Phạm vi Phiên (Session Scope):** Prepared statements thường chỉ tồn tại trong một phiên kết nối cơ sở dữ liệu. Cần có cơ chế để tạo lại chúng khi ứng dụng khởi tạo kết nối mới (ví dụ: sử dụng connection pool).
2. **Tham Số Hóa An Toàn:** Luôn sử dụng các tham số được đánh dấu ($1, $2, ...) để truyền dữ liệu vào truy vấn. **Tuyệt đối không** nối chuỗi trực tiếp vào câu lệnh SQL để tránh lỗ hổng SQL Injection.
3. **Kiểm Tra Kế Hoạch Thực Thi:** Sử dụng lệnh EXPLAIN ANALYZE trên các EXECUTE để đảm bảo rằng cơ sở dữ liệu đang chọn kế hoạch thực thi tối ưu và sử dụng các chỉ mục (indexes) hiệu quả.
4. **Giải Phóng Tài Nguyên:** Khi một prepared statement không còn cần thiết trong phiên làm việc, nên sử dụng lệnh DEALLOCATE ten_prepared_statement để giải phóng tài nguyên bộ nhớ trên máy chủ cơ sở dữ liệu.

## Ví dụ Thực Thi

Sau khi đã chuẩn bị các câu lệnh, việc thực thi chúng sẽ nhanh hơn:

```
-- Tìm kiếm sản phẩm giày chạy bộ trong khoảng giá 500k - 2 triệu
```

```
EXECUTE search_products('giày chạy', NULL, 500000, 2000000);

-- Cập nhật trạng thái đơn hàng có UUID là 'order-uuid' thành 'shipped'
EXECUTE update_order_status('order-uuid', 'shipped');

-- Kiểm tra xem sản phẩm có UUID 'product-uuid' còn đủ 5 sản phẩm không
EXECUTE check_stock('product-uuid', 5);
```

# Kết Luận

Việc áp dụng Prepared Statements cho các truy vấn thường xuyên lặp lại như đã liệt kê ở trên là một phương pháp hiệu quả để cải thiện đáng kể hiệu suất của hệ thống TheShoe. Bằng cách giảm tải cho cơ sở dữ liệu trong việc phân tích và lập kế hoạch truy vấn, hệ thống sẽ phản hồi nhanh hơn, đặc biệt là dưới tải trọng cao, mang lại trải nghiệm tốt hơn cho người dùng và quản trị viên. Nên ưu tiên triển khai các prepared statements này trong quá trình phát triển và bảo trì hệ thống.