

Danh sách Stored Procedures và Functions đúng với nguyên tắc Clean Architecture

Clean Architecture yêu cầu **logic nghiệp vụ (business logic)** phải nằm trong lớp ứng dụng (use cases/entities), không phải trong cơ sở dữ liệu. Các stored procedures (SP) và functions chỉ nên xử lý **thao tác dữ liệu đơn thuần** hoặc tối ưu truy vấn. Dưới đây là danh sách các SP và functions **phù hợp** với nguyên tắc này:

1. Stored Procedures (SP) Hợp Lệ

Tên SP	Mục Đích	Lý Do Phù Hợp
<code>sp_update_order_status</code>	Cập nhật trạng thái đơn hàng (ví dụ: "đang giao", "đã giao").	Chỉ thay đổi giá trị trạng thái, không chứa logic nghiệp vụ.
<code>sp_remove_from_cart</code>	Xóa sản phẩm khỏi giỏ hàng.	Thao tác xóa dữ liệu đơn giản.
<code>sp_add_address</code>	Thêm địa chỉ mới cho người dùng.	Lưu dữ liệu địa chỉ, không xử lý logic (ví dụ: validation).
<code>sp_set_default_address</code>	Đặt một địa chỉ làm mặc định.	Thao tác cập nhật trường <code>is_default</code> , không logic phức tạp.
<code>sp_create_payment</code>	Tạo bản ghi thanh toán cho đơn hàng.	Lưu thông tin thanh toán, không xử lý logic (ví dụ: xác thực phương thức).
<code>sp_update_payment_status</code>	Cập nhật trạng thái thanh toán (ví dụ: "thành công", "thất bại").	Chỉ thay đổi trạng thái, không kiểm tra điều kiện nghiệp vụ.
<code>sp_assign_shipper</code>	Gán shipper cho đơn hàng.	Thao tác gán ID shipper vào đơn hàng.
<code>sp_update_shipping_status</code>	Cập nhật trạng thái giao hàng.	Thay đổi trạng thái, không xử lý logic (ví dụ: tính toán thời gian giao).
<code>sp_register_user</code>	Đăng ký người dùng mới.	Lưu thông tin người dùng, mã hóa mật khẩu (xử lý kỹ thuật, không logic nghiệp vụ).
<code>sp_update_user_profile</code>	Cập nhật thông tin hồ sơ người dùng.	Thao tác cập nhật dữ liệu cơ bản.

2. Functions Hợp Lệ

Tên Function	Mục Đích	Lý Do Phù Hợp
<code>fn_check_stock</code>	Kiểm tra số lượng tồn kho của sản phẩm.	Trả về giá trị boolean (đủ/không đủ hàng), không xử lý logic nghiệp vụ.
<code>fn_search_products</code>	Tìm kiếm sản phẩm theo tên, danh mục, giá.	Truy vấn dữ liệu dựa trên điều kiện, không chứa logic kinh doanh.
<code>fn_verify_user_credentials</code>	Xác thực thông tin đăng nhập.	Kiểm tra mật khẩu và trả về kết quả, không xử lý logic nghiệp vụ.

3. Stored Procedures và Functions Cần Điều Chỉnh

Những SP và functions sau **vi phạm Clean Architecture** vì chứa logic nghiệp vụ hoặc quy tắc kinh doanh:

Tên SP/Function	Lý Do Vi Phạm
<code>sp_create_order</code>	Chứa logic kiểm tra tồn kho, áp dụng mã giảm giá, tính toán giá.
<code>sp_apply_discount_code</code>	Kiểm tra điều kiện mã giảm giá (thời hạn, số lần dùng), cập nhật <code>uses_count</code> .
<code>sp_apply_promotion</code>	Áp dụng khuyến mãi dựa trên quy tắc kinh doanh (ví dụ: giảm giá theo danh mục).
<code>fn_calculate_total_amount</code>	Tính tổng tiền sau khi áp dụng khuyến mãi và mã giảm giá (logic nghiệp vụ).
<code>sp_add_review</code>	Kiểm tra điều kiện đánh giá (ví dụ: người dùng đã mua sản phẩm).

Giải Pháp Điều Chỉnh để Tuân Thủ Clean Architecture

1. Di chuyển logic nghiệp vụ vào Use Cases:
- Ví dụ: Logic kiểm tra mã giảm giá trong `sp_apply_discount_code` nên được xử lý trong lớp ứng dụng (NestJS Service).

```
// Trong OrderService (NestJS)
async applyDiscount(orderId: string, discountCode: string) {
  const isValid = await this.discountService.validate(discountCode);
```

```
if (!isValid) throw new Error("Mã giảm giá không hợp lệ");
await this.orderRepository.applyDiscount(orderId, discountCode);
}
```

1. Sử dụng SP/Functions chỉ cho thao tác dữ liệu:

- Ví dụ: `sp_create_order` chỉ nên chèn dữ liệu vào bảng `Order` và `OrderDetail`, không xử lý logic tồn kho hoặc mã giảm giá.

2. Tách biệt lớp ứng dụng và cơ sở dữ liệu:

- Sử dụng **Repository Pattern** để gọi SP/Functions từ lớp cơ sở dữ liệu, không để logic nghiệp vụ phụ thuộc vào chúng.

Kết Luận

Các stored procedures và functions **phù hợp với Clean Architecture** phải: * Chỉ thực hiện **thao tác dữ liệu cơ bản** (CRUD). * Không chứa **quy tắc kinh doanh** (ví dụ: validation, tính toán giá). * Được gọi từ lớp cơ sở dữ liệu (Repository) thông qua interface trừu tượng.

Những SP/Functions chứa logic nghiệp vụ cần được tái cấu trúc để logic đó nằm trong lớp ứng dụng, đảm bảo tính linh hoạt và dễ bảo trì.