HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN: CƠ SỞ DỮ LIỆU PHÂN TÁN

ĐỀ TÀI 2: HỆ THỐNG BÁN HÀNG TRỰC TUYẾN ĐA KHO

Giảng viên hướng dẫn: TS.Kim Ngọc Bách

Lớp / Học phần: INT14148 – N10

Nhóm thực hiện: Nhóm 20

Thành viên nhóm:

Đặng Thảo Nguyên	:B23DCCN609
Đỗ Trí Cương	:B23DCCN105
Nguyễn Trường Giang	:B23DCCN259

Hà Nội, tháng 6/2026

BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	Nhiệm vụ phụ trách
1	Đặng Thảo Nguyên B23DCCN6009	- Khảo sát bài toán, xác định các luồng nghiệp vụ. - Xây dựng bảng Tác nhân, biểu đồ mô tả tổng quát và đặc tả yêu cầu chức năng/phi chức năng. - Thiết kế giao diện người dùng và xây dựng logic điều hướng API từ Frontend. - Tổng hợp nội dung, chuẩn hóa định dạng và soạn thảo báo cáo Word tổng kết toàn bộ dự án
2	Đỗ Trí Cương B23DCCN105	- Thiết kế CSDL, mô hình ERD và lược đồ quan hệ toàn cục. - Xây dựng chiến lược phân mảnh. - Lập ma trận cấp phát dữ liệu cho 3 Site. - Viết Stored Procedure xử lý điều phối đơn hàng, tách đơn liên vùng. - Lập trình cơ chế khóa và giao dịch 2PC.
3	Nguyễn Trường Giang B23DCCN259	- Lập trình các câu truy vấn phân tán cốt lõi. - Thiết kế mô hình và chiến lược nhân bản cho các bảng dữ liệu dùng chung. - Cấu hình máy chủ ảo, thiết lập Linked Server để kết nối các Site. - Đo lường hiệu năng truy vấn, phân tích chi phí Semi-join và chạy kịch bản test tranh chấp dữ liệu đồng thời.

MỤC LỤC

MỤC LỤC 3
DANH MỤC HÌNH VẼ 7
DANH MỤC BẢNG BIỂU 7
CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 9
1.1. Bối cảnh và lý do chọn đề tài 9
1.1.1. Bối cảnh thị trường 9
1.1.2. Vấn đề thực tiễn 9
1.1.3. Tầm quan trọng chiến lược của giải pháp 10
1.2. Mục tiêu của đề tài 10
1.3. Phạm vi đề tài 11
1.3.1. Phạm vi hệ thống 11
1.3.2. Phạm vi nghiệp vụ 11
1.3.3. Phạm vi kỹ thuật 11
1.4. Phương pháp thực hiện 11
1.5. Bố cục báo cáo 12
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 13
2.1. Tổng quan về cơ sở dữ liệu phân tán 13
2.1.1. Định nghĩa 13
2.1.2. Các thành phần cơ bản 13
2.1.3. Đặc tính nền tảng của CSDL phân tán 13
2.1.4. Ưu điểm và thách thức 14
2.2. Phân mảnh dữ liệu 14
2.2.1. Khái niệm và mục đích 14
2.2.2. Phân mảnh ngang nguyên thủy 14
2.2.3. Phân mảnh ngang 15
2.2.4. Phân mảnh dọc 15
2.2.5. Phân mảnh hỗn hợp 16

2.3. Cấp phát và phân bổ dữ liệu 16
2.3.1. Khái niệm 16
2.3.2. Các chiến lược cấp phát 16
2.3.3. Các yếu tố ảnh hưởng đến quyết định cấp phát 16
2.4. Nhân bản dữ liệu (Replication) 17
2.4.1. Khái niệm và lợi ích 17
2.4.2. Nhân bản đồng thời và bất đồng thời 17
2.4.3. Mô hình Master-Slave và Multi-Master 17
2.5. Xử lý và tối ưu truy vấn phân tán 17
2.5.1. Tổng quan quy trình 17
2.5.2. Tối ưu hóa bằng Semi-join 18
2.5.3. Dữ liệu trung gian trong truy vấn phân tán 18
2.6. Giao dịch phân tán và nhất quán dữ liệu 18
2.6.1. Tính chất ACID trong môi trường phân tán 18
2.6.2. Giao thức Two-Phase Commit (2PC) 19
2.6.3. Kiểm soát đồng thời trong môi trường phân tán 19
2.6.4. Bài toán nhất quán tồn kho trong hệ thống đa kho 20
2.6.5. Định lý CAP và ứng dụng thực tế 20
CHƯƠNG 3. PHÂN TÍCH NGHIỆP VỤ 21
3.1. Mô tả bài toán 21
3.2. Yêu cầu chức năng 21
3.3. Yêu cầu phi chức năng 23
3.4. Tác nhân và biểu đồ mô tả tổng quát hệ thống 25
3.4.1. Tác nhân hệ thống 25
3.4.2. Biểu đồ mô tả tổng quát hệ thống 27
3.5. Đặc trưng phân tán của hệ thống 27
CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN 31
4.1. Thiết kế lược đồ toàn cục 31

4.1.1. Mô hình thực thể – liên kết (ER) 31
4.1.2. Lược đồ quan hệ toàn cục 31
4.1.3. Mô tả các thực thể 43
4.2. Phân mảnh dữ liệu 46
4.2.1. Phân mảnh ngang theo kho / khu vực 46
4.2.2. Phân mảnh dọc 52
4.2.3. Giải thích lựa chọn phân mảnh 54
4.3. Cấp phát dữ liệu 55
4.3.1. Dữ liệu lưu cục bộ tại từng vùng 55
4.3.2. Ma trận cấp phát dữ liệu 57
4.4. Nhân bản và đồng bộ dữ liệu 60
4.4.1. Dữ liệu dùng chung cần nhân bản 60
4.4.2. Cơ chế đồng bộ 62
CHƯƠNG 5. MÔ PHỎNG NGHIỆP VỤ PHÂN TÁN 64
5.1. Tình huống 1: Tra cứu tồn kho của một sản phẩm trên toàn hệ thống 64
5.2. Tình huống 2: Đặt hàng khi một kho không đủ số lượng 64
5.3. Phân tích nghiệp vụ phân tán 68
CHƯƠNG 6. CÁC TRUY VẤN PHÂN TÁN 74
6.1. Truy vấn 1: Kiểm tra sản phẩm còn hàng ở những kho nào 74
6.2. Truy vấn 2: Tổng tồn kho của một sản phẩm trên toàn hệ thống 76
6.3. Truy vấn 3: Thống kê doanh thu theo tháng của từng kho và toàn hệ thống 79
6.4. Truy vấn 4: Top sản phẩm bán chạy nhất toàn hệ thống 82
6.5. Truy vấn 5: Đơn hàng có sản phẩm được xuất từ nhiều kho khác nhau. 85
CHƯƠNG 7. TRUY XUẤT ĐỒNG THỜI VÀ NHẤT QUÁN TỒN KHO 88
7.1. Bài toán đặt ra 88
7.2. Khái niệm nhất quán trong demo 88
7.3. Mô hình SoLuongTon và SoLuongDatHang 89
7.4. Cơ chế khóa dòng khi giữ hàng 89

7.5. Chu trình giữ hàng, xác nhận xuất và hoàn tác 90
7.5.1. Giữ hàng 90
7.5.2. Xác nhận xuất hàng 91
7.5.3. Hoàn tác khi giao dịch thất bại 91
7.6. Kịch bản đồng thời trong bản demo 92
7.7. Ghi nhận thay đổi tồn kho trong hệ thống 93
7.8. Mức độ hoàn chỉnh của giải pháp 94
CHƯƠNG 8. NỘI DUNG NÂNG CAO 95
8.1 Mô phỏng cập nhật tồn kho đồng thời 95
8.2. Chiến lược replication cho thông tin sản phẩm 96
8.2.1 Các bảng áp dụng replication 96
8.2.2 Mô hình replication đề xuất 97
8.3. Phân tích ảnh hưởng của truy vấn join phân tán 98
8.4 So sánh hiệu năng giữa lưu trữ tập trung và phân tán 102
CHƯƠNG 9. ĐÁNH GIÁ HỆ THỐNG 104
9.1. Kết quả đạt được 104
9.2. Đối chiếu với tiêu chí đánh giá 105
9.3. Hạn chế 105
9.4. Hướng phát triển 106
KẾT LUẬN 107
TÀI LIỆU THAM KHẢO 108
PHỤ LỤC 108

DANH MỤC HÌNH VẼ

Hình 1 Biểu đồ mô tả tổng quát hệ thống.....27
Hình 2 Mô hình ER của hệ thống.....31
Hình 3 Sơ đồ luồng tra cứu tồn kho toàn hệ thống.....64
Hình 4 phân tích luồng đặt hàng khi kho xử lý không đủ hàng 65
Hình 5 Phân tích dữ liệu phân tán: Kết quả truy vấn sản phẩm còn hàng 76
Hình 6 Phân tích dữ liệu phân tán: Kết quả truy vấn tổng tồn kho 78
Hình 7 Phân tích dữ liệu phân tán: Kết quả truy vấn thống kê doanh thu 81
Hình 8 Phân tích dữ liệu phân tán: Kết quả truy vấn top 10 sản phẩm bán chạy 84
Hình 9 Phân tích dữ liệu phân tán: Kết quả truy vấn đơn hàng xuất từ nhiều kho 86
Hình 10 Người đặt hàng chậm không tranh chấp được dữ liệu 96
Hình 11 Người tranh chấp được dữ liệu.....96

DANH MỤC BẢNG BIỂU

Bảng 1 Tác nhân hệ thống.....25
Bảng 2 Cấu trúc bảng KhuVuc.....31
Bảng 3 Cấu trúc bảng Kho 32
Bảng 4 Cấu trúc bảng DanhMuc 32
Bảng 5 Cấu trúc bảng ThuongHieu.....33
Bảng 6 Cấu trúc bảng DanhMuc_ThuongHieu.....33
Bảng 7 Cấu trúc bảng SanPham_Core 33
Bảng 8 Cấu trúc bảng SanPham_Detail 34
Bảng 9 Cấu trúc bảng NguoiDung 34
Bảng 10 Cấu trúc bảng User_Global_Index.....35
Bảng 11 Cấu trúc bảng GioHang.....36
Bảng 12 Cấu trúc bảng ChiTietGioHang 37
Bảng 13 Cấu trúc bảng DonHang.....37
Bảng 14 Cấu trúc bảng ChiTietDonHang 38
Bảng 15 Cấu trúc bảng PhieuXuatKho 39
Bảng 16 Cấu trúc bảng ChiTietXuatKho 40
Bảng 17 Cấu trúc bảng TonKho.....41

Bảng 18 Cấu trúc bảng PhieuNhapKho.....41
Bảng 19 Cấu trúc bảng ChiTietPhieuNhap 42
Bảng 20: Mô tả các thực thể trong lược đồ toàn cục.....43
Bảng 21 Ma trận cấp phát dữ liệu 57
Bảng 22 Bảng tạm phân bổ dữ liệu 71
Bảng 23 Các nhóm xử lý phân bổ đơn hàng 71
Bảng 24 So sánh hiệu quả truy vấn phân tán trực tiếp và gọi procedure 72
Bảng 25 Ý nghĩa các thuộc tính tính toán tồn kho 74
Bảng 26 Ý nghĩa các thuộc tính kết quả truy vấn tổng tồn kho 76
Bảng 27 Ý nghĩa các điều kiện lọc tính doanh thu.....79
Bảng 28 Ý nghĩa các thuộc tính kết quả truy vấn top sản phẩm 83
Bảng 29 Tổng kết trạng thái dữ liệu khi cập nhật đồng thời.....96
Bảng 30 Chiến lược replication cho các bảng dữ liệu.....96
Bảng 31 Mô hình replication đề xuất 97
Bảng 32 Phân rã chi phí của phép join phân tán 98
Bảng 33 Ưu, nhược điểm của chiến lược chuyển toàn bộ bảng.....98
Bảng 34 Ưu, nhược điểm của chiến lược chuyển bảng nhỏ.....99
Bảng 35 Quy trình thực thi Semijoin 100
Bảng 36 Ưu, nhược điểm của chiến lược Semijoin.....100
Bảng 37 So sánh lượng dữ liệu trung gian phát sinh.....100
Bảng 38 Đề xuất áp dụng chiến lược join trong hệ thống đa kho 101
Bảng 39 So sánh hiệu năng giữa lưu trữ tập trung và phân tán.....102

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1. Bối cảnh và lý do chọn đề tài

1.1.1. Bối cảnh thị trường

Trong những năm gần đây, thương mại điện tử (TMĐT) Việt Nam đang bước vào giai đoạn tăng trưởng mạnh mẽ và liên tục. Theo các báo cáo ngành, quy mô thị trường TMĐT Việt Nam liên tục đạt mức tăng trưởng hai con số mỗi năm, với hàng chục triệu người tiêu dùng chuyển sang hình thức mua sắm trực tuyến. Sự bùng nổ này không chỉ đến từ các đô thị lớn như Hà Nội hay Thành phố Hồ Chí Minh, mà còn lan rộng ra các tỉnh thành trên khắp cả nước.

Tốc độ tăng trưởng nhanh chóng đó đặt ra áp lực ngày càng lớn lên hạ tầng logistics và quản lý kho vận - đặc biệt đối với các doanh nghiệp bán lẻ vận hành trên phạm vi toàn quốc, trải dài từ Bắc vào Nam. Để duy trì sức cạnh tranh, các doanh nghiệp buộc phải tìm cách đưa hàng hóa đến tay khách hàng nhanh hơn, rẻ hơn, và đáng tin cậy hơn - đây chính là bài toán trung tâm mà hệ thống trong đề tài này hướng tới giải quyết.

1.1.2. Vấn đề thực tiễn

Các doanh nghiệp đang vận hành kho hàng tại nhiều địa điểm địa lý khác nhau đang phải đối mặt với một loạt thách thức thực tế:

- Quản lý tồn kho thiếu đồng bộ: Dữ liệu tồn kho giữa kho Hà Nội và kho Thành phố Hồ Chí Minh thường không được cập nhật theo thời gian thực với nhau. Hệ quả là nhân viên bán hàng có thể xác nhận một đơn hàng dựa trên thông tin tồn kho đã cũ, dẫn đến tình trạng bán hàng vượt mức tồn kho thực tế hoặc ngược lại - một kho đang thiếu hàng trong khi kho kia lại dư thừa mà không được điều phối kịp thời.
- Xử lý và phân bổ đơn hàng thủ công: Khi một đơn hàng được đặt, việc quyết định kho nào sẽ xuất hàng phần lớn phụ thuộc vào thao tác thủ công hoặc quy trình rời rạc. Điều này không chỉ dễ phát sinh sai sót mà còn bỏ qua cơ hội tối ưu hóa địa lý. Chẳng hạn, khách hàng ở miền Nam lẽ ra nên được giao từ kho Thành phố Hồ Chí Minh, nhưng thực tế lại được giao từ kho Hà Nội do thiếu hệ thống định tuyến thông minh, làm tăng thời gian và chi phí giao hàng một cách không cần thiết.
- Thiếu khả năng quan sát toàn diện: Nhà quản lý không có một giao diện thống nhất để theo dõi tổng thể hiệu suất kinh doanh trên cả hệ thống. Mỗi kho vận hành một cách riêng biệt, khiến việc tổng hợp báo cáo doanh thu, tình trạng tồn kho hay hiệu quả xuất nhập trở nên chậm trễ và kém chính xác. Hệ quả là các quyết định chiến lược như bổ sung hàng, điều chuyển hàng giữa kho, hay triển khai chương trình khuyến mãi theo vùng thiếu căn cứ dữ liệu kịp thời.
- Chi phí vận hành leo thang: Khi đơn hàng không được phân bổ về đúng kho địa lý gần nhất, chi phí vận chuyển liên vùng tăng cao một cách đáng kể. Về lâu dài,

đây là gánh nặng trực tiếp làm giảm biên lợi nhuận của doanh nghiệp và làm giảm trải nghiệm của khách hàng do thời gian giao hàng kéo dài.

1.1.3. Tầm quan trọng chiến lược của giải pháp

Việc xây dựng một hệ thống sàn thương mại điện tử đa kho với cơ sở dữ liệu phân tán không chỉ đơn thuần là một bài toán kỹ thuật - đây là nền tảng cạnh tranh cốt lõi trong môi trường kinh doanh kỹ thuật số hiện nay.

Về tốc độ giao hàng, hệ thống có khả năng tự động định tuyến mỗi đơn hàng đến kho địa lý gần nhất với khách hàng, qua đó rút ngắn đáng kể thời gian vận chuyển và nâng cao sự hài lòng của người mua.

Về chi phí vận hành, bằng cách loại bỏ các chuyến giao hàng liên vùng không cần thiết và cân bằng tải hàng hóa thông minh giữa các kho, doanh nghiệp có thể tiết kiệm đáng kể chi phí logistics.

Về khả năng mở rộng, kiến trúc hệ thống được thiết kế theo hướng phân tán, cho phép tích hợp thêm kho mới trong tương lai mà không cần tái cấu trúc toàn bộ hạ tầng công nghệ.

Về năng lực quản trị, dữ liệu kinh doanh được tập trung hóa ở mức truy vấn (dù vẫn lưu trữ phân tán), cho phép ban lãnh đạo tra cứu báo cáo doanh thu, tình trạng tồn kho toàn hệ thống và hiệu suất từng kho theo thời gian thực.

1.2. Mục tiêu của đề tài

Mục tiêu tổng quát: Xây dựng một cơ sở dữ liệu phân tán phục vụ hệ thống sàn thương mại điện tử đa kho, đảm bảo các nghiệp vụ cốt lõi (tra cứu tồn kho, đặt hàng, quản lý đơn hàng, báo cáo doanh thu) hoạt động chính xác, hiệu quả và nhất quán trong môi trường có nhiều nút dữ liệu địa lý.

Các mục tiêu cụ thể của đề tài bao gồm:

- Phân tích và thiết kế lược đồ cơ sở dữ liệu toàn cục phản ánh đúng nghiệp vụ của hệ thống bán hàng đa kho.
- Thiết kế phương án phân mảnh dữ liệu hợp lý để mỗi nút kho chỉ lưu trữ phần dữ liệu thuộc phạm vi quản lý của mình.
- Xác định chiến lược cấp phát dữ liệu về các site, đảm bảo cân bằng giữa tính cục bộ của truy vấn và tính nhất quán toàn hệ thống.
- Thiết lập cơ chế nhân bản dữ liệu dùng chung giữa các site theo mô hình phù hợp.
- Xây dựng và phân tích các truy vấn phân tán phản ánh các nghiệp vụ thực tế: tra cứu tồn kho, thống kê doanh thu, tìm sản phẩm bán chạy,…
- Mô phỏng các kịch bản nghiệp vụ phân tán điển hình, trong đó có kịch bản đặt hàng liên kho khi một kho thiếu hàng.

- Phân tích và xử lý vấn đề truy xuất đồng thời và nhất quán tồn kho trong môi trường phân tán.
- Cài đặt hoặc mô phỏng hệ thống trên tối thiểu 3 site (2 kho địa phương + 1 server trung tâm) và thử nghiệm với bộ dữ liệu mẫu đủ lớn.

1.3. Phạm vi đề tài

1.3.1. Phạm vi hệ thống

Hệ thống được thiết kế và mô phỏng với 3 site (nút) cơ sở dữ liệu:

- Site 1 - Kho Bắc: Quản lý cơ sở dữ liệu cục bộ của kho hàng tại khu vực miền Bắc, phục vụ khách hàng miền Bắc. Site này lưu trữ dữ liệu tồn kho, phiếu nhập kho, đơn hàng và chi tiết đơn hàng thuộc khu vực miền Bắc.
- Site 2 - Kho Nam: Quản lý cơ sở dữ liệu cục bộ của kho hàng tại miền Nam, phục vụ khách hàng miền Trung và miền Nam. Site này lưu trữ dữ liệu tương tự nhưng thuộc phạm vi khu vực miền Nam.
- Site trung tâm: Đóng vai trò điều phối, đồng bộ dữ liệu và phục vụ các truy vấn tổng hợp liên site.

1.3.2. Phạm vi nghiệp vụ

Đề tài tập trung vào các nghiệp vụ chính sau:

- Quản lý danh mục sản phẩm, thương hiệu và khu vực.
- Quản lý kho hàng và tồn kho theo từng site.
- Quản lý người dùng (tài khoản khách hàng) và giỏ hàng.
- Tạo và xử lý đơn hàng, bao gồm trường hợp đơn hàng xuất từ nhiều kho.
- Thống kê doanh thu và báo cáo tồn kho theo kho/vùng/toàn hệ thống.

1.3.3. Phạm vi kỹ thuật

Hệ thống sử dụng một DBMS phổ biến - SQL Server để mô phỏng môi trường phân tán. Logic phân tán có thể được hiện thực hóa thông qua cơ chế native của DBMS hoặc tầng ứng dụng trung gian. Báo cáo chú trọng vào thiết kế, phân tích và lý giải các quyết định phân tán, bên cạnh phần cài đặt và thử nghiệm thực tế.

1.4. Phương pháp thực hiện

Nhóm thực hiện đề tài theo quy trình gồm các bước kế tiếp nhau:

Bước 1 - Phân tích nghiệp vụ: Tìm hiểu bài toán thực tế của hệ thống bán hàng đa kho, xác định các thực thể, mối quan hệ và luồng nghiệp vụ chính.

Bước 2 - Thiết kế lược đồ toàn cục: Xây dựng mô hình ER và lược đồ quan hệ toàn cục phản ánh đầy đủ cấu trúc dữ liệu của hệ thống.

Bước 3 - Thiết kế phân mảnh và cấp phát: Xác định cách phân mảnh từng quan hệ về các site, lý giải lựa chọn và lập ma trận cấp phát dữ liệu.

Bước 4 - Thiết kế nhân bản và đồng bộ: Xác định các quan hệ cần nhân bản, chọn chiến lược đồng bộ phù hợp (đồng thời/bất đồng thời) và mô tả cách xử lý xung đột.

Bước 5 - Xây dựng truy vấn phân tán: Viết và phân tích các truy vấn phân tán điển hình, chỉ rõ các site tham gia, dữ liệu truyền qua mạng và cách tổng hợp kết quả.

Bước 6 - Cài đặt và mô phỏng: Thiết lập môi trường thực nghiệm với các site, nạp dữ liệu mẫu, và chạy thử các kịch bản nghiệp vụ phân tán.

Bước 7 - Kiểm thử và đánh giá: Kiểm tra tính nhất quán dữ liệu, đo hiệu suất truy vấn, phân tích kết quả và đối chiếu với yêu cầu đề bài.

1.5. Bố cục báo cáo

Báo cáo được tổ chức thành các chương như sau:

- Chương 1 trình bày tổng quan về bối cảnh, lý do, mục tiêu và phạm vi của đề tài.
- Chương 2 cung cấp nền tảng lý thuyết về cơ sở dữ liệu phân tán: phân mảnh, cấp phát, nhân bản, xử lý truy vấn phân tán và quản lý giao dịch.
- Chương 3 phân tích nghiệp vụ chi tiết: yêu cầu chức năng, yêu cầu phi chức năng và đặc tính phân tán của bài toán.
- Chương 4 trình bày toàn bộ thiết kế cơ sở dữ liệu phân tán: lược đồ toàn cục, phương án phân mảnh, ma trận cấp phát và kiến trúc nhân bản.
- Chương 5 mô phỏng hai kịch bản nghiệp vụ phân tán điển hình và phân tích chi tiết quá trình thực thi.
- Chương 6 xây dựng và phân tích năm truy vấn phân tán đặc trưng của hệ thống.
- Chương 7 đề cập đến truy xuất đồng thời, quản lý giao dịch và đảm bảo nhất quán tồn kho.
- Chương 8 trình bày các nội dung nâng cao tùy chọn (so sánh hiệu năng, phân tích distributed join,...).
- Chương 9 đánh giá tổng kết hệ thống, nêu hạn chế và định hướng phát triển.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về cơ sở dữ liệu phân tán

2.1.1. Định nghĩa

Cơ sở dữ liệu phân tán (Distributed Database System - DDBMS) là một hệ thống trong đó dữ liệu được lưu trữ tại nhiều vị trí vật lý khác nhau (gọi là các site hay nút), kết nối với nhau qua mạng máy tính, nhưng vẫn được quản lý thống nhất như một cơ sở dữ liệu đơn lẻ từ góc nhìn của người dùng.

Nói cách khác: dữ liệu nằm ở nhiều nơi, nhưng người dùng làm việc như thể nó ở một chỗ.

Trong bối cảnh đề tài này, hệ thống gồm ít nhất hai site kho địa phương, mỗi site quản lý dữ liệu của khu vực mình, nhưng toàn bộ hệ thống vẫn phục vụ một nghiệp vụ bán hàng thống nhất.

2.1.2. Các thành phần cơ bản

Một hệ CSDL phân tán điển hình bao gồm ba thành phần chính:

Các site cục bộ (Local Sites): Mỗi site là một máy chủ cơ sở dữ liệu độc lập, có bộ xử lý, bộ nhớ và lưu trữ riêng. Site có khả năng xử lý các truy vấn cục bộ ngay cả khi mạng gặp sự cố.

Mạng kết nối (Communication Network): Đường truyền kết nối các site với nhau. Đặc tính của mạng (băng thông, độ trễ, độ ổn định) ảnh hưởng trực tiếp đến hiệu suất của toàn hệ thống phân tán.

Lược đồ toàn cục (Global Schema): Là cái nhìn tổng thể, thống nhất về toàn bộ dữ liệu trong hệ thống - bao gồm cả dữ liệu đang nằm phân tán ở nhiều site. Lược đồ này là cơ sở để hệ thống biết dữ liệu nào đang ở đâu và cần truy vấn những site nào khi thực thi một yêu cầu.

2.1.3. Đặc tính nền tảng của CSDL phân tán

Một hệ CSDL phân tán lý tưởng cần đảm bảo các đặc tính sau:

Trong suốt vị trí (Location Transparency): Người dùng không cần biết dữ liệu đang nằm ở site nào. Họ chỉ cần viết truy vấn dựa trên lược đồ toàn cục; hệ thống tự lo việc định vị và lấy dữ liệu.

Trong suốt nhân bản (Replication Transparency): Nếu một bảng được sao chép ra nhiều site, người dùng vẫn chỉ thấy và thao tác với một bảng duy nhất. Hệ thống tự đảm bảo các bản sao được đồng bộ.

Trong suốt phân mảnh (Fragmentation Transparency): Kể cả khi một bảng bị chia nhỏ và phân bổ ra nhiều site, người dùng vẫn truy vấn như thể đó là một bảng đầy đủ.

Tự trị cục bộ (Local Autonomy): Mỗi site có thể tự quản lý dữ liệu cục bộ của mình và tiếp tục hoạt động ngay cả khi tạm thời mất kết nối với các site khác.

2.1.4. Ưu điểm và thách thức

Ưu điểm nổi bật của CSDL phân tán so với CSDL tập trung truyền thống gồm có: tính sẵn sàng cao hơn (nếu một site gặp sự cố, các site còn lại vẫn hoạt động), hiệu suất tốt hơn nhờ xử lý cục bộ (dữ liệu của kho Hà Nội được xử lý ngay tại Hà Nội, không cần gửi lên server trung tâm), và khả năng mở rộng linh hoạt theo địa lý.

Tuy nhiên, hệ thống phân tán cũng mang lại những thách thức đáng kể: độ phức tạp thiết kế cao hơn, cần đảm bảo tính nhất quán dữ liệu giữa các site, xử lý lỗi mạng và đảm bảo giao dịch phân tán hoàn thành đúng đắn.

2.2. Phân mảnh dữ liệu

2.2.1. Khái niệm và mục đích

Phân mảnh dữ liệu là quá trình chia một quan hệ (bảng) trong lược đồ toàn cục thành các phần nhỏ hơn gọi là mảnh (fragment), sau đó phân bổ các mảnh này về các site khác nhau.

Mục đích cốt lõi của phân mảnh là đưa dữ liệu đến gần nơi sử dụng nó nhất. Ví dụ, dữ liệu tồn kho của kho Hà Nội nên nằm ngay tại site Hà Nội để các truy vấn cục bộ không cần truyền dữ liệu qua mạng.

Một phương án phân mảnh đúng đắn cần thỏa mãn ba điều kiện:

- Đầy đủ (Completeness): Mọi dòng dữ liệu trong quan hệ gốc phải xuất hiện trong ít nhất một mảnh - không được để mất dữ liệu.
- Tái tạo được (Reconstruction): Từ tập hợp các mảnh, phải có thể tái tạo lại hoàn toàn quan hệ ban đầu bằng các phép toán quan hệ (hợp, kết nối).
- Rời nhau (Disjointness): (Áp dụng cho phân mảnh ngang) Mỗi dòng dữ liệu chỉ thuộc đúng một mảnh, tránh trùng lặp không cần thiết.

2.2.2. Phân mảnh ngang nguyên thủy

Phân mảnh ngang chia một bảng theo hàng (dòng): mỗi mảnh chứa một tập con các dòng thỏa mãn một điều kiện lọc nhất định. Kết quả là mỗi mảnh có cùng cấu trúc cột với bảng gốc, nhưng chứa ít dòng hơn.

Ví dụ trong hệ thống đề tài: Bảng Kho được phân mảnh ngang theo MaKhuVuc:

```
Kho_HN  = SELECT * FROM Kho WHERE MaKhuVuc = 'KV_BAC';
```

Kho_HCM = SELECT * FROM Kho WHERE MaKhuVuc = 'KV_NAM';

Mảnh Kho_HN chỉ chứa các kho thuộc khu vực miền Bắc và được lưu tại Site Hà Nội; mảnh Kho_HCM chứa kho miền Nam và lưu tại Site TP. Hồ Chí Minh. Truy vấn liên quan đến kho Hà Nội sẽ được xử lý hoàn toàn cục bộ tại site đó.

Tương tự, các bảng TonKho, PhieuNhapKho, ChiTietPhieuNhap, GioHang, ChiTietGioHang đều được phân mảnh ngang theo MaKho - kế thừa từ phân mảnh của bảng Kho.

2.2.3. Phân mảnh ngang

Phân mảnh ngang dẫn xuất áp dụng cho các bảng phụ thuộc (con) của một bảng đã được phân mảnh trước đó (bảng cha). Thay vì định nghĩa lại điều kiện lọc, bảng con được phân mảnh bằng cách kế thừa phân mảnh từ bảng cha thông qua khóa ngoại.

Ví dụ: Bảng DonHang và ChiTietDonHang được phân mảnh dẫn xuất theo MaKhuVucGiao - kế thừa từ phân mảnh của Kho:

```
DonHang_HN  = SELECT * FROM DonHang
              WHERE MaKhuVucGiao = 'KV_BAC';
DonHang_HCM = SELECT * FROM DonHang
              WHERE MaKhuVucGiao = 'KV_NAM';
```

Cách làm này đảm bảo rằng đơn hàng giao về miền Bắc sẽ nằm cùng site với kho miền Bắc, giúp các phép kết nối (join) giữa DonHang và Kho được thực hiện hoàn toàn cục bộ mà không cần truyền dữ liệu qua mạng.

Lưu ý: bảng DonHang còn có trường MaKhoXuat để xử lý nghiệp vụ đặc biệt khi một đơn hàng được xuất từ kho của site khác.

2.2.4. Phân mảnh dọc

Phân mảnh dọc chia một bảng theo cột (thuộc tính): mỗi mảnh chứa một tập con các cột, nhưng giữ lại toàn bộ các dòng. Để có thể tái tạo lại bảng gốc, mỗi mảnh bắt buộc phải giữ lại khóa chính.

Ví dụ: Bảng SanPham có thể được tách thành hai mảnh dọc:

- Mảnh thông tin cơ bản (dùng ở mọi nơi): MaSP, TenSP, MaDanhMuc, DonGia, MoBan
- Mảnh thông tin mở rộng (ít truy cập hơn): MaSP, MoTaChiTiet, ThongSoKyThuat, HinhAnh

Phân mảnh dọc thường được áp dụng khi một bảng có nhiều cột nhưng mỗi nghiệp vụ chỉ cần một tập con các cột, nhằm giảm lượng dữ liệu truyền qua mạng và tăng tốc truy vấn.

2.2.5. Phân mảnh hỗn hợp

Phân mảnh hỗn hợp kết hợp cả phân mảnh ngang và dọc: trước tiên chia bảng theo hàng, sau đó chia tiếp mỗi mảnh theo cột (hoặc ngược lại). Kết quả là các mảnh vừa khác nhau về dòng dữ liệu, vừa khác nhau về tập thuộc tính. Đây là dạng phân mảnh linh hoạt nhất nhưng cũng phức tạp nhất khi tái tạo và tối ưu truy vấn.

2.3. Cấp phát và phân bổ dữ liệu

2.3.1. Khái niệm

Sau khi đã phân mảnh dữ liệu, bước tiếp theo là cấp phát - tức là quyết định mỗi mảnh sẽ được lưu tại site nào. Đây là một bài toán tối ưu hóa: mục tiêu là cực tiểu hóa chi phí truy vấn và truyền dữ liệu qua mạng, đồng thời đảm bảo tính sẵn sàng và hiệu năng cần thiết.

2.3.2. Các chiến lược cấp phát

Cấp phát một bản sao (Non-replicated / Fragmented-only): Mỗi mảnh chỉ được lưu tại đúng một site. Ưu điểm là không tốn chi phí đồng bộ; nhược điểm là nếu site đó gặp sự cố, mảnh dữ liệu tương ứng không thể truy cập.

Nhân bản toàn phần (Fully Replicated): Toàn bộ cơ sở dữ liệu được sao chép đầy đủ tại mọi site. Truy vấn đọc rất nhanh (luôn xử lý cục bộ), nhưng chi phí đồng bộ cho mỗi thao tác ghi tăng lên theo số lượng site, không phù hợp khi hệ thống có nhiều thao tác cập nhật.

Nhân bản một phần (Partially Replicated): Đây là chiến lược cân bằng và thực tế nhất: chỉ một số mảnh nhất định (thường là dữ liệu dùng chung, ít thay đổi) được nhân bản ra nhiều site; các mảnh còn lại chỉ nằm ở một site duy nhất. Trong đề tài này, các bảng DanhMuc, SanPham, Brand, KhuVuc được nhân bản tại tất cả các site vì chúng được đọc thường xuyên ở mọi nơi nhưng hiếm khi thay đổi.

2.3.3. Các yếu tố ảnh hưởng đến quyết định cấp phát

Khi quyết định cấp phát mỗi mảnh về site nào, cần cân nhắc:

- Tần suất truy cập: Mảnh nào được truy cập nhiều nhất từ site nào? Nên cấp phát mảnh đó về gần site đó.
- Tỉ lệ đọc/ghi: Dữ liệu đọc nhiều, ghi ít thì thích hợp nhân bản; dữ liệu ghi nhiều thì hạn chế nhân bản.
- Chi phí truyền thông: Độ trễ và băng thông mạng giữa các site ảnh hưởng trực tiếp đến chi phí truy vấn liên site.
- Yêu cầu tính sẵn sàng: Dữ liệu quan trọng (ví dụ: thông tin đơn hàng) cần được nhân bản để tránh mất mát nếu một site gặp sự cố.

2.4. Nhân bản dữ liệu (Replication)

2.4.1. Khái niệm và lợi ích

Nhân bản (Replication) là cơ chế duy trì nhiều bản sao của cùng một mảnh dữ liệu tại nhiều site khác nhau. Mục tiêu chính gồm: tăng tính sẵn sàng (nếu một site chết, bản sao ở site khác vẫn phục vụ được), tăng tốc độ đọc (đọc từ bản sao cục bộ nhanh hơn truy vấn qua mạng), và cải thiện khả năng chịu lỗi.

2.4.2. Nhân bản đồng thời và bất đồng thời

Nhân bản đồng thời (Synchronous Replication): Một thao tác ghi chỉ được coi là hoàn thành khi tất cả các bản sao tại tất cả site đã được cập nhật thành công. Ưu điểm là tính nhất quán mạnh - mọi site luôn có dữ liệu giống nhau; nhược điểm là độ trễ ghi tăng lên đáng kể vì phải chờ xác nhận từ tất cả site.

Nhân bản bất đồng thời (Asynchronous Replication): Thao tác ghi hoàn thành ngay tại site chủ; các site còn lại sẽ được cập nhật sau một khoảng thời gian nhất định. Ưu điểm là ghi nhanh hơn; nhược điểm là trong khoảng thời gian trễ đó, các site khác có thể trả về dữ liệu cũ.

Trong thực tế, các hệ thống thương mại điện tử thường kết hợp cả hai: nhân bản đồng thời cho dữ liệu tồn kho (cần nhất quán ngay lập tức để tránh overselling), nhân bản bất đồng thời cho dữ liệu ít nhạy cảm hơn như lịch sử đơn hàng hay thông tin sản phẩm.

2.4.3. Mô hình Master-Slave và Multi-Master

Master-Slave (Chủ-Tớ): Chỉ một site được phép ghi (master); các site còn lại chỉ đọc (slave) và nhận bản cập nhật từ master. Cơ chế đơn giản, không có xung đột ghi, dễ đảm bảo nhất quán. Nhược điểm là master trở thành điểm nghẽn cổ chai và single point of failure.

Multi-Master (Đa chủ): Nhiều site đều có quyền ghi. Phù hợp cho các dữ liệu cần cập nhật từ nhiều nơi đồng thời, ví dụ bảng NguoiDung - khách hàng có thể đăng nhập và cập nhật thông tin từ bất kỳ kho/site nào. Thách thức chính là cần có cơ chế phát hiện và giải quyết xung đột khi cùng một dòng dữ liệu bị sửa đồng thời ở hai site khác nhau.

2.5. Xử lý và tối ưu truy vấn phân tán

2.5.1. Tổng quan quy trình

Khi người dùng gửi một truy vấn đến hệ thống CSDL phân tán, truy vấn đó không thể được thực thi trực tiếp như trong hệ thống tập trung. Thay vào đó, nó phải trải qua một quy trình gồm bốn bước:

Bước 1 - Phân rã truy vấn (Query Decomposition): Truy vấn viết trên lược đồ toàn cục được phân tích cú pháp và kiểm tra tính hợp lệ, sau đó biến đổi thành một dạng chuẩn hóa (thường là đại số quan hệ).

Bước 2 - Cục bộ hóa (Localization): Dựa trên thông tin phân mảnh, hệ thống thay thế mỗi quan hệ toàn cục bằng tập hợp các mảnh tương ứng của nó, xác định cụ thể mảnh nào đang nằm ở site nào.

Bước 3 - Tối ưu hóa toàn cục (Global Optimization): Hệ thống sinh ra nhiều kế hoạch thực thi khác nhau và chọn kế hoạch có chi phí thấp nhất, dựa trên ước lượng kích thước dữ liệu, chi phí truyền mạng và chi phí xử lý cục bộ.

Bước 4 - Thực thi phân tán (Distributed Execution): Kế hoạch được gửi về từng site liên quan để thực thi cục bộ; kết quả từ các site được tổng hợp lại tại site điều phối (coordinator) để trả về cho người dùng.

2.5.2. Tối ưu hóa bằng Semi-join

Một kỹ thuật quan trọng trong tối ưu hóa truy vấn phân tán là semi-join (nửa kết nối). Khi cần kết nối hai bảng nằm ở hai site khác nhau, thay vì truyền toàn bộ một bảng sang site kia để thực hiện kết nối, semi-join thực hiện như sau:

- Site A gửi chỉ cột khóa kết nối (nhỏ hơn nhiều so với toàn bộ bảng) sang Site B.
- Site B dùng danh sách khóa đó để lọc ra các dòng phù hợp trong bảng của mình.
- Site B chỉ gửi lại phần kết quả đã lọc (thường nhỏ hơn nhiều bảng gốc) cho Site A.
- Site A thực hiện kết nối cuối cùng với dữ liệu đã lọc nhận được.

Kỹ thuật này đặc biệt hiệu quả khi hệ số chọn lọc (selectivity) cao - tức là chỉ một phần nhỏ dòng từ site kia thực sự cần thiết cho kết quả cuối.

2.5.3. Dữ liệu trung gian trong truy vấn phân tán

Trong quá trình thực thi truy vấn liên site, hệ thống thường phát sinh dữ liệu trung gian - là kết quả tạm thời được sinh ra tại một site rồi truyền sang site khác để xử lý tiếp. Việc tối thiểu hóa lượng dữ liệu trung gian này là một trong những mục tiêu chính của bộ tối ưu truy vấn phân tán. Các biện pháp thường dùng gồm: đẩy sớm các phép lọc, dùng semi-join, và chọn thứ tự thực hiện phép kết nối hợp lý.

2.6. Giao dịch phân tán và nhất quán dữ liệu

2.6.1. Tính chất ACID trong môi trường phân tán

Một giao dịch là một đơn vị công việc logic phải được thực hiện trọn vẹn hoặc không thực hiện gì cả. Các tính chất ACID - Nguyên tử (Atomicity), Nhất quán (Consistency),

Cô lập (Isolation), Bền vững (Durability) - là nền tảng đảm bảo độ tin cậy của mọi hệ thống cơ sở dữ liệu.

Trong môi trường phân tán, việc duy trì ACID trở nên phức tạp hơn nhiều so với hệ thống tập trung. Một giao dịch đơn giản như "đặt hàng" có thể phải đồng thời cập nhật bảng TonKho tại Site Hà Nội và bảng DonHang tại Site TP. Hồ Chí Minh - nếu một trong hai thao tác thất bại, toàn bộ giao dịch phải được hoàn tác (rollback) để đảm bảo tính nguyên tử.

2.6.2. Giao thức Two-Phase Commit (2PC)

Two-Phase Commit (2PC - Cam kết hai pha) là giao thức chuẩn để đảm bảo tính nguyên tử cho các giao dịch phân tán. Giao thức hoạt động theo hai pha rõ ràng:

Pha 1 - Chuẩn bị (Prepare Phase):

- Site điều phối gửi thông điệp PREPARE đến tất cả site tham gia.
- Mỗi site tham gia kiểm tra xem mình có thể hoàn tất phần công việc của giao dịch hay không.
- Nếu có thể, site ghi log "sẵn sàng" vào đĩa và trả về YES; nếu không, trả về NO.

Pha 2 - Cam kết (Commit Phase):

- Nếu tất cả site đều trả về YES: coordinator gửi COMMIT đến tất cả site; mỗi site chính thức áp dụng thay đổi và giải phóng khóa.
- Nếu bất kỳ site nào trả về NO (hoặc không phản hồi): coordinator gửi ABORT đến tất cả site; mỗi site hoàn tác (rollback) phần công việc của mình.

Giao thức 2PC đảm bảo mọi site hoặc cùng commit hoặc cùng rollback - không có trạng thái nửa vời. Tuy nhiên, nhược điểm của 2PC là site tham gia phải chờ (blocking) trong trường hợp coordinator gặp sự cố ở giữa quá trình, dẫn đến nguy cơ deadlock phân tán.

2.6.3. Kiểm soát đồng thời trong môi trường phân tán

Khi nhiều giao dịch cùng truy cập và sửa đổi một dữ liệu tại cùng một thời điểm, cần có cơ chế kiểm soát đồng thời (Concurrency Control) để tránh các lỗi như mất cập nhật (lost update) hay đọc dữ liệu bẩn (dirty read).

Khóa hai pha (Two-Phase Locking - 2PL) là kỹ thuật phổ biến nhất: mỗi giao dịch phải giành khóa trên dữ liệu trước khi đọc/ghi, và chỉ giải phóng khóa sau khi đã hoàn tất toàn bộ thao tác. Trong môi trường phân tán, cần có một bộ quản lý khóa phân tán (Distributed Lock Manager) hoặc cơ chế khóa tại từng site cục bộ để phối hợp với nhau.

Kiểm soát đồng thời bằng phiên bản (MVCC - Multi-Version Concurrency Control) là hướng tiếp cận hiện đại hơn, được sử dụng rộng rãi trong PostgreSQL và MySQL InnoDB. Thay vì dùng khóa cho thao tác đọc, MVCC duy trì nhiều phiên bản của cùng

một dòng dữ liệu - mỗi giao dịch đọc thấy "snapshot" nhất quán tại thời điểm nó bắt đầu, mà không cản trở các giao dịch ghi đang diễn ra song song.

2.6.4. Bài toán nhất quán tồn kho trong hệ thống đa kho

Một trong những vấn đề nhạy cảm nhất trong hệ thống bán hàng phân tán là đảm bảo tính nhất quán của tồn kho. Xét tình huống: hai khách hàng cùng đặt hàng cùng một sản phẩm tại kho Hà Nội trong cùng một giây - nếu cả hai giao dịch đồng thời đọc tồn kho là 1 và cùng nhau quyết định "còn hàng", sẽ xảy ra tình trạng bán hàng âm tồn kho.

Để phòng tránh điều này, thao tác giảm tồn kho phải được thực hiện theo dạng cập nhật có điều kiện nguyên tử:

```
UPDATE TonKho
SET SoLuong = SoLuong - :soLuongDat
WHERE MaSP = :maSP
  AND MaKho = :maKho
  AND SoLuong >= :soLuongDat;
```

Nếu câu lệnh không cập nhật được dòng nào (tức tồn kho thực tế đã không đủ), hệ thống biết ngay rằng giao dịch cần được xử lý theo hướng khác (báo lỗi hoặc kiểm tra kho khác). Đây là kỹ thuật nền tảng để tránh race condition trong môi trường nhiều giao dịch đồng thời.

2.6.5. Định lý CAP và ứng dụng thực tế

Định lý CAP (Brewer, 2000) phát biểu rằng trong một hệ thống phân tán, không thể đồng thời đảm bảo cả ba tính chất: Nhất quán (Consistency), Sẵn sàng (Availability) và Chịu phân vùng (Partition Tolerance). Khi mạng bị phân vùng (hai site không liên lạc được với nhau), hệ thống buộc phải chọn một trong hai: ưu tiên tiếp tục phục vụ với dữ liệu có thể chưa đồng bộ (chọn Availability), hoặc từ chối phục vụ cho đến khi đảm bảo được tính nhất quán (chọn Consistency).

Trong hệ thống bán hàng đa kho, đây là sự đánh đổi thực tế cần cân nhắc: với dữ liệu tồn kho, nên ưu tiên nhất quán (tránh overselling); với dữ liệu danh mục sản phẩm, có thể chấp nhận đôi chút trễ đồng bộ để duy trì tính sẵn sàng của hệ thống.

CHƯƠNG 3. PHÂN TÍCH NGHIỆP VỤ

3.1. Mô tả bài toán

Hệ thống bán hàng trực tuyến đa kho là một hệ sinh thái thương mại điện tử phục vụ khách hàng trên phạm vi toàn quốc. Để tối ưu hóa tốc độ giao hàng và chi phí vận chuyển, doanh nghiệp tổ chức lưu trữ hàng hóa tại các kho khu vực. Tuy nhiên, dữ liệu kinh doanh cần được quản lý một cách thống nhất để đảm bảo trải nghiệm mua sắm không bị gián đoạn và báo cáo tài chính chính xác.

Một doanh nghiệp kinh doanh thương mại điện tử quy mô lớn tại Việt Nam, chuyên cung cấp các sản phẩm công nghệ và đồ gia dụng thông minh. Để phục vụ nhu cầu mua sắm trực tuyến ngày càng cao, tối ưu hóa chi phí vận chuyển và rút ngắn thời gian giao hàng đến tay người tiêu dùng, doanh nghiệp đã xây dựng một hệ thống phân phối đa kho trải dài trên cả nước.

Để minh họa và triển khai hệ thống cơ sở dữ liệu phân tán, doanh nghiệp cấu hình hệ thống gồm 03 Site vật lý như sau:

- Site 1 (Site Trung tâm): Đặt tại trụ sở chính, chịu trách nhiệm quản lý danh mục dùng chung, tài khoản khách hàng toàn hệ thống và tổng hợp báo cáo doanh thu toàn quốc.
- Site 2 (Site Miền Bắc): Đặt tại Hà Nội, quản lý các kho và chịu trách nhiệm xử lý các đơn hàng của khách hàng khu vực phía Bắc.
- Site 3 (Site Miền Nam): Đặt tại TP. Hồ Chí Minh, quản lý các kho và chịu trách nhiệm xử lý các đơn hàng của khách hàng khu vực phía Nam.

3.2. Yêu cầu chức năng

Hệ thống phải đáp ứng tối thiểu các chức năng cốt lõi sau, đảm bảo vận hành trơn tru trên cả 3 Site (Trung tâm, Miền Bắc, Miền Nam):

- Quản lý kho hàng
- Quản lý thông tin kho: Hỗ trợ Site Trung tâm thêm, đọc, sửa các thông tin cơ bản của kho (Mã kho, tên kho, khu vực, địa chỉ, sức chứa, trạng thái hoạt động). Dữ liệu này dùng để định tuyến và phân bổ dữ liệu theo vùng.

- Quản lý tồn kho chi tiết: Hệ thống phải theo dõi được đồng thời hai trạng thái: Tồn thực tế (số lượng trên kệ) và Tồn khả dụng (số lượng thực tế trừ đi lượng hàng đang giữ chỗ cho các đơn chưa xuất). Nhân viên tại site nào chỉ có quyền cấu hình, cập nhật tồn kho tại site đó.
- Quản lý Nhập - Xuất kho: Cho phép các kho lập và lưu trữ phiếu nhập kho (bổ sung hàng từ nhà cung cấp/điều chuyển) và phiếu xuất kho (giao cho khách/điều chuyển đi). Toàn bộ lịch sử biến động kho được lưu trữ cục bộ tại site con để tối ưu tốc độ đọc/ghi.

- Quản lý sản phẩm và danh mục sản phẩm

- Quản lý danh mục và thương hiệu: Hỗ trợ tạo, sửa, xóa cấu trúc cây danh mục và thương hiệu để phân loại hàng hóa. Đây là dữ liệu dùng chung toàn hệ thống.
- Quản lý thông tin sản phẩm: Cho phép Site Trung tâm thêm mới hoặc cập nhật thông tin sản phẩm (Mã sản phẩm, tên, giá bán, hình ảnh, đơn vị tính, trạng thái kinh doanh). Thông tin này sau khi cập nhật tại Trung tâm phải tự động đồng bộ (nhân bản) xuống các Site chi nhánh phục vụ khách hàng tra cứu tức thời.

- Quản lý khách hàng và người dùng

- Quản lý khách hàng: Hỗ trợ đăng ký, đăng nhập, quản lý thông tin cá nhân và sổ địa chỉ giao hàng. Hệ thống tự động phân mảnh thông tin khách hàng theo khu vực (Dựa vào địa chỉ) về Site Miền Bắc hoặc Miền Nam để tối ưu hiệu năng xử lý tại chỗ.
- Quản lý nhân viên và phân quyền: Hỗ trợ Admin hệ thống tạo tài khoản và phân quyền chặt chẽ cho từng nhóm đối tượng (Nhân viên kho miền Bắc, nhân viên kho miền Nam, Admin trung tâm). Nhân viên chi nhánh nào chỉ được thấy và thao tác trên dữ liệu phát sinh tại chi nhánh đó.

- Quản lý giỏ hàng, đơn hàng và cập nhật trạng thái đơn

- Quản lý giỏ hàng: Cho phép khách hàng thêm, bớt, sửa số lượng sản phẩm trong giỏ hàng trước khi tiến hành thanh toán.
- Xử lý đơn hàng: Ghi nhận thông tin đơn đặt hàng từ khách hàng. Dữ liệu đơn hàng phát sinh ở phân vùng nào sẽ được lưu cục bộ tại phân vùng đó.
- Cập nhật trạng thái đơn: Theo dõi và cập nhật trạng thái đơn hàng theo thời gian thực (Chờ xử lý, Đã xác nhận, Đang chuẩn bị hàng, Đang giao, Đã hoàn thành,

Đã hủy). Khách hàng có quyền hủy đơn nếu đơn hàng chưa chuyển sang trạng thái "Đang chuẩn bị hàng/Xuất kho".

- Kiểm tra tồn kho theo từng kho và trên toàn hệ thống
  - Kiểm tra tồn kho cục bộ: Khách hàng hoặc nhân viên tại Site Miền Bắc có thể tra cứu nhanh số lượng hàng khả dụng tại các kho miền Bắc
  - Kiểm tra tồn kho toàn hệ thống: Hệ thống cung cấp chức năng truy vấn phân tán (kết hợp dữ liệu từ cả Site Miền Bắc và Miền Nam) để tính tổng tồn kho của một sản phẩm trên toàn quốc, phục vụ giao diện hiển thị cho khách hàng trực tuyến khi kho tại chỗ bị hết hàng.
- Xử lý đặt hàng (Kịch bản lấy hàng từ nhiều kho)
  - Định hướng kho gần nhất: Hệ thống tự động ưu tiên trừ tồn kho khả dụng tại các kho thuộc cùng khu vực địa lý của khách hàng để giảm chi phí vận chuyển.
  - Xử lý điều phối liên vùng (Multi-warehouse Fulfillment): Khi các kho tại vùng địa lý của khách không đủ hàng, hệ thống tự động kích hoạt tiến trình kiểm tra chéo sang các site miền khác. Nếu đủ, hệ thống tự động thực hiện nghiệp vụ tách đơn/chia phiếu xuất kho cho nhiều kho ở các site khác nhau cùng tham gia đáp ứng một đơn hàng của khách.
- Thống kê doanh thu và báo cáo tổng hợp
  - Báo cáo cục bộ tại chi nhánh: Hỗ trợ các site chi nhánh tự xuất báo cáo doanh thu, sản phẩm sắp hết hàng, hiệu suất đóng gói của các kho thuộc quyền quản lý của mình.
  - Báo cáo hợp nhất tại trung tâm: Site Trung tâm thực hiện truy vấn xuyên suốt, gom dữ liệu đơn hàng từ tất cả các site chi nhánh về để tính toán các chỉ số:
    - Tổng doanh thu toàn hệ thống theo tháng/quý/năm.
    - Thống kê sản phẩm bán chạy nhất (Top-selling) trên toàn quốc.
    - Thống kê số lượng khách hàng biến động theo từng khu vực địa lý.

3.3. Yêu cầu phi chức năng

Để hệ thống CSDL phân tán vận hành ổn định, an toàn và không xảy ra xung đột dữ liệu, hệ thống bắt buộc phải đáp ứng các tiêu chuẩn phi chức năng sau:

- Tính nhất quán của tồn kho

- Ràng buộc số lượng: Số lượng tồn kho khả dụng tại mọi thời điểm tuyệt đối không được phép âm (Available_Stock >= 0).
- Kiểm soát đồng thời: Khi có nhiều khách hàng cùng thực hiện đặt mua một sản phẩm cuối cùng tại một kho trong cùng một thời điểm, hệ thống phải áp dụng cơ chế khóa (locking) hoặc giao dịch phân tán (Distributed Transaction - 2PC) để đảm bảo chỉ có duy nhất một đơn hàng đặt thành công, loại bỏ hoàn toàn hiện tượng bán quá số lượng (Overbooking).

- Hiệu năng truy vấn xuyên site

- Tối ưu hóa dữ liệu dùng chung: Toàn bộ thông tin sản phẩm, danh mục, thương hiệu phải được áp dụng chiến lược nhân bản hoàn toàn (Full Replication) xuống tất cả các site chi nhánh. Điều này đảm bảo tốc độ duyệt web, tìm kiếm và xem chi tiết sản phẩm của khách hàng đạt tốc độ tức thời (< 500ms) mà không tốn chi phí truyền thông đường truyền mạng giữa các site.
- Tối ưu hóa dữ liệu giao dịch: Dữ liệu đơn hàng, phiếu nhập xuất phải được phân mảnh ngang nguyên thủy theo vùng địa lý để các thao tác ghi dữ liệu (Insert) khi khách mua hàng diễn ra cục bộ tại site chi nhánh với tốc độ cao nhất, không bị nghẽn đường truyền về trung tâm.

- Tính sẵn sàng cao

- Tính tự chủ cục bộ: Hệ thống phải đảm bảo nếu xảy ra sự cố mất kết nối mạng Internet giữa các Site (ví dụ: Mất kết nối giữa Site Miền Bắc và Site Miền Nam), thì Site Miền Bắc vẫn phải hoạt động bình thường, nhân viên kho vẫn nhập/xuất được hàng và khách hàng tại Miền Bắc vẫn mua được các sản phẩm còn tồn tại kho Miền Bắc.
- Cơ chế đồng bộ lại: Khi kết nối mạng giữa các site được khôi phục, hệ thống phải tự động đồng bộ bù các dữ liệu thay đổi (như trạng thái đơn hàng, doanh thu cập nhật) về Site Trung tâm mà không làm mất mát dữ liệu.

- Khả năng mở rộng

- Hệ thống thiết kế theo kiến trúc phân tán hướng module, cho phép doanh nghiệp dễ dàng mở rộng quy mô. Khi Cosmo Mart mở thêm chi nhánh mới (Ví dụ: Site

Miền Trung quản lý Kho Đà Nẵng, Kho Quy Nhơn), hệ thống chỉ cần cấu hình thêm 1 Site con, định tuyến phân mảnh dữ liệu theo mã khu vực mới mà không cần phải đập đi thiết kế lại toàn bộ lược đồ cơ sở dữ liệu toàn cục.

- Bảo mật và Cấp quyền

- Cô lập dữ liệu cục bộ: Hệ thống phải mã hóa đường truyền dữ liệu giữa các site (Sử dụng VPN/SSL).
- Phân quyền chặt chẽ trên CSDL: Thiết lập tài khoản kết nối CSDL (Database Users) riêng biệt cho từng site. Nhân viên kho tại Site Miền Bắc tuyệt đối không có quyền gửi lệnh UPDATE hay DELETE trực tiếp lên các bảng dữ liệu thuộc quyền quản lý của Site Miền Nam, ngăn chặn việc can thiệp hoặc làm sai lệch dữ liệu của chi nhánh khác. Khách hàng chỉ có quyền xem thông tin của chính mình thông qua các View được phân mảnh tương ứng.

3.4. Tác nhân và biểu đồ mô tả tổng quát hệ thống

3.4.1. Tác nhân hệ thống

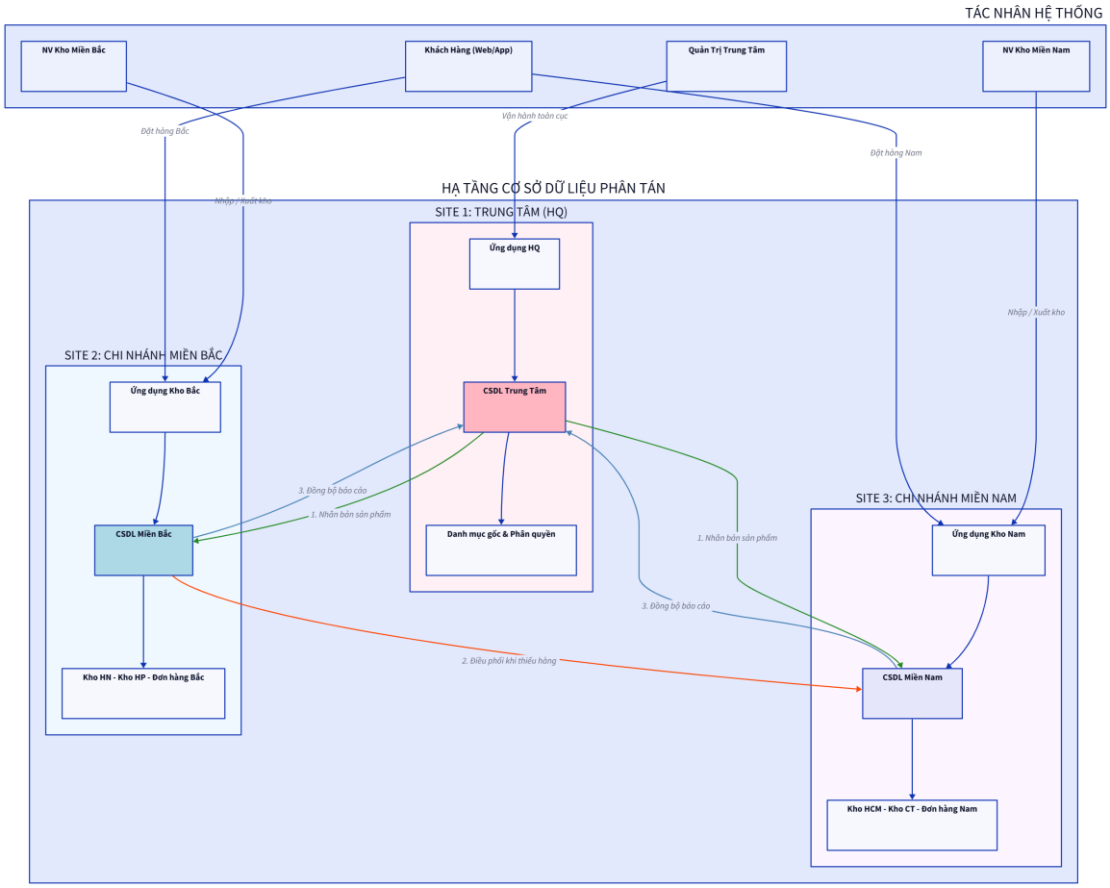
Hệ thống bao gồm các tác nhân chính sau:

Bảng 1 Tác nhân hệ thống

STT	Tác nhân (Actor)	Vị trí	Mô tả
1	Khách hàng *(Customer)*	Toàn bộ các Site (Tự động định tuyến)	- Truy cập Hệ thống để tra cứu Sản phẩm/Danh mục - Đặt hàng và cập nhật Giỏ hàng. - Hệ thống dựa vào địa chỉ để định tuyến dữ liệu đơn hàng của Khách hàng về đúng mảnh dữ liệu tương ứng (Site Miền Bắc hoặc Site Miền Nam).

STT	Tác nhân (Actor)	Vị trí	Mô tả
2	Nhân viên Kho *(Warehouse Staff)*	Các Site chi nhánh con	- Chỉ có quyền thao tác trên CSDL cục bộ của Site mình làm việc. - Thực hiện Nhập/Xuất kho, cập nhật số lượng tồn kho vật lý tại chỗ. - Cập nhật trạng thái xử lý đơn hàng do kho mình phụ trách.
3	Quản lý Trung tâm *(Central Admin)*	Site Trung tâm	- Quản lý danh mục dữ liệu dùng chung và cấu hình lệnh Nhân bản xuống các site con. - Quản lý tài khoản, phân quyền toàn hệ thống. - Phát lệnh Truy vấn phân tán xuyên Site để tổng hợp báo cáo doanh thu, tồn kho toàn quốc và điều phối các đơn hàng liên vùng.

3.4.2. Biểu đồ mô tả tổng quát hệ thống



Hình 1 Biểu đồ mô tả tổng quát hệ thống

3.5. Đặc trưng phân tán của hệ thống

Để xây dựng một kiến trúc cơ sở dữ liệu phân tán tối ưu cho hệ thống bán hàng đa kho, việc phân tích bản chất phân tán của dữ liệu là bước đi tiên quyết. Dữ liệu trong hệ thống không tồn tại tập trung mà mang tính chất phân tán tự nhiên, xuất phát trực tiếp từ dòng chảy vật chất (vận hành kho) và dòng chảy thông tin (đặt hàng, quản trị) trong mô hình kinh doanh thực tế.

Bản chất phân tán của hệ thống được minh chứng và bóc tách qua ba đặc trưng cốt lõi sau:

- Tính cục bộ cao của dữ liệu giao dịch và vận hành

Phần lớn các dữ liệu có tần suất biến động cao nhất, dung lượng tăng trưởng nhanh nhất trong hệ thống – bao gồm đơn hàng (DonHang), chi tiết đơn hàng (ChiTietDonHang),

phiếu nhập xuất và trạng thái tồn kho thực tế (TonKho) – đều có đặc điểm nghiệp vụ: Phát sinh tại đâu thì được sử dụng chủ yếu tại đó.

- Luồng vận hành kho: Các hoạt động kiểm đếm, nhập kho từ nhà cung cấp hay thực tế xuất hàng lên xe vận chuyển được thực hiện độc lập bởi các thủ kho tại từng địa bàn.
- Luồng xử lý đơn hàng: Khách hàng ở khu vực địa lý nào sẽ phát sinh đơn hàng giao về vùng đó. Đơn hàng này ngay lập tức biến thành lệnh đóng gói cục bộ cho chi nhánh phụ trách khu vực đó thực hiện.

Ý nghĩa phân tán: Nếu lưu trữ tập trung toàn bộ các bản ghi giao dịch này tại một máy chủ duy nhất (Single Point), hệ thống sẽ gánh chịu hiện tượng "nghẽn cổ chai" (Bottleneck) khi hàng ngàn lệnh ghi dữ liệu (INSERT, UPDATE) từ các kho trên toàn quốc đổ về cùng một lúc. Việc đưa các dữ liệu này về lưu trữ ngay tại Site chi nhánh phát sinh giúp tối ưu hóa tốc độ xử lý tại chỗ, giảm tải cho đường truyền mạng và đảm bảo tính tự chủ cục bộ (Local Autonomy) – nếu một site mất kết nối Internet quốc gia, các site còn lại vẫn bán hàng và đóng gói bình thường.

- Nhu cầu nhân bản dữ liệu danh mục dùng chung

Trái ngược với dữ liệu giao dịch mang tính cục bộ, nhóm dữ liệu về Danh mục ngành hàng, Thương hiệu và Thông tin sản phẩm cơ bản (SanPham, DanhMuc) lại có đặc điểm nghiệp vụ là: Ít khi thay đổi (tần suất cập nhật thấp) nhưng tần suất truy cập (Đọc - SELECT) cực kỳ lớn từ tất cả các Site trên toàn hệ thống.

- Bất kể khách hàng truy cập ứng dụng từ Hà Nội (Site Miền Bắc) hay TP.HCM (Site Miền Nam), họ đều phải được tiếp cận với một cây danh mục sản phẩm đồng nhất, chính xác về thông số, giá bán niêm yết và hình ảnh.
- Nếu thông tin sản phẩm chỉ nằm tập trung tại Site Trung tâm, mỗi khi khách hàng duyệt ứng dụng để xem chi tiết sản phẩm, hệ thống lại phải thực hiện một truy vấn xuyên mạng về Trung tâm. Điều này gây tốn kém chi phí truyền thông đường truyền và tạo ra độ trễ lớn, làm giảm trải nghiệm người dùng.

Ý nghĩa phân tán: Nhóm dữ liệu này mang đặc trưng điển hình của bài toán Nhân bản dữ liệu (Data Replication). Dữ liệu gốc sẽ được cấu hình và quản lý tại Site Trung tâm,

nhưng cần được sao chép hoàn toàn xuống tất cả các Site chi nhánh con nhằm phục vụ nhu cầu tra cứu tại chỗ với tốc độ tức thời (< 100ms).

- Nhu cầu truy vấn tổng hợp xuyên site

Mặc dù dữ liệu giao dịch được phân tán sâu rộng về các chi nhánh để tối ưu vận hành cục bộ, hệ thống Cosmo Mart vẫn phải đảm bảo tính chỉnh thể thống nhất của một doanh nghiệp thông qua các nhu cầu khai thác dữ liệu toàn cục:

- Xử lý đơn hàng liên vùng: Khi xảy ra hiện tượng lệch pha cung - cầu (ví dụ: Sản phẩm A tại Site Miền Bắc đã hết hàng tồn khả dụng, nhưng Site Miền Nam vẫn còn tồn kho lớn), hệ thống cần thực hiện truy vấn thời gian thực xuyên Site để kiểm tra khả năng đáp ứng và tách đơn, điều phối hàng hóa liên vùng để không bỏ lỡ cơ hội bán hàng.
- Tổng hợp quản trị chiến lược: Ban giám đốc tại Site Trung tâm cần các câu lệnh truy vấn phân tán phức tạp để quét qua dữ liệu của tất cả các site chi nhánh, thực hiện các phép toán gộp (SUM, COUNT, AVG) và phép hợp (UNION) để xuất ra bức tranh tài chính toàn quốc: Thống kê doanh thu theo tháng của từng kho, tìm top sản phẩm bán chạy nhất hệ thống, hay phân tích số lượng khách hàng theo khu vực.

Từ việc bóc tách và phân tích 3 đặc trưng phân tán nêu trên, hệ thống đã thiết lập được cơ sở lý luận khoa học vững chắc để tiến hành thiết kế kiến trúc dữ liệu phân tán cho giai đoạn tiếp theo, cụ thể:

- Áp dụng Phân mảnh ngang nguyên thủy: Đối với các thực thể mang tính cục bộ cao và gắn liền với yếu tố địa lý như KhachHang và DonHang. Tiêu chí phân mảnh sẽ dựa trên mã khu vực hoặc địa chỉ giao hàng để phân tách dữ liệu thành các mảnh độc lập đặt tại Site Miền Bắc và Site Miền Nam.
- Áp dụng Phân mảnh ngang dẫn xuất: Đối với thực thể ChiTietDonHang (dẫn xuất theo DonHang) và TonKho (dẫn xuất theo Kho). Điều này đảm bảo dữ liệu chi tiết của đơn hàng hay số lượng tồn kho của chi nhánh nào sẽ tự động đi theo và lưu trữ đồng bộ tại đúng Site của chi nhánh đó.
- Áp dụng Chiến lược Nhân bản hoàn toàn: Đối với các thực thể dùng chung như SanPham, DanhMuc, ThuongHieu. Các bảng dữ liệu này sẽ xuất hiện ở tất cả các

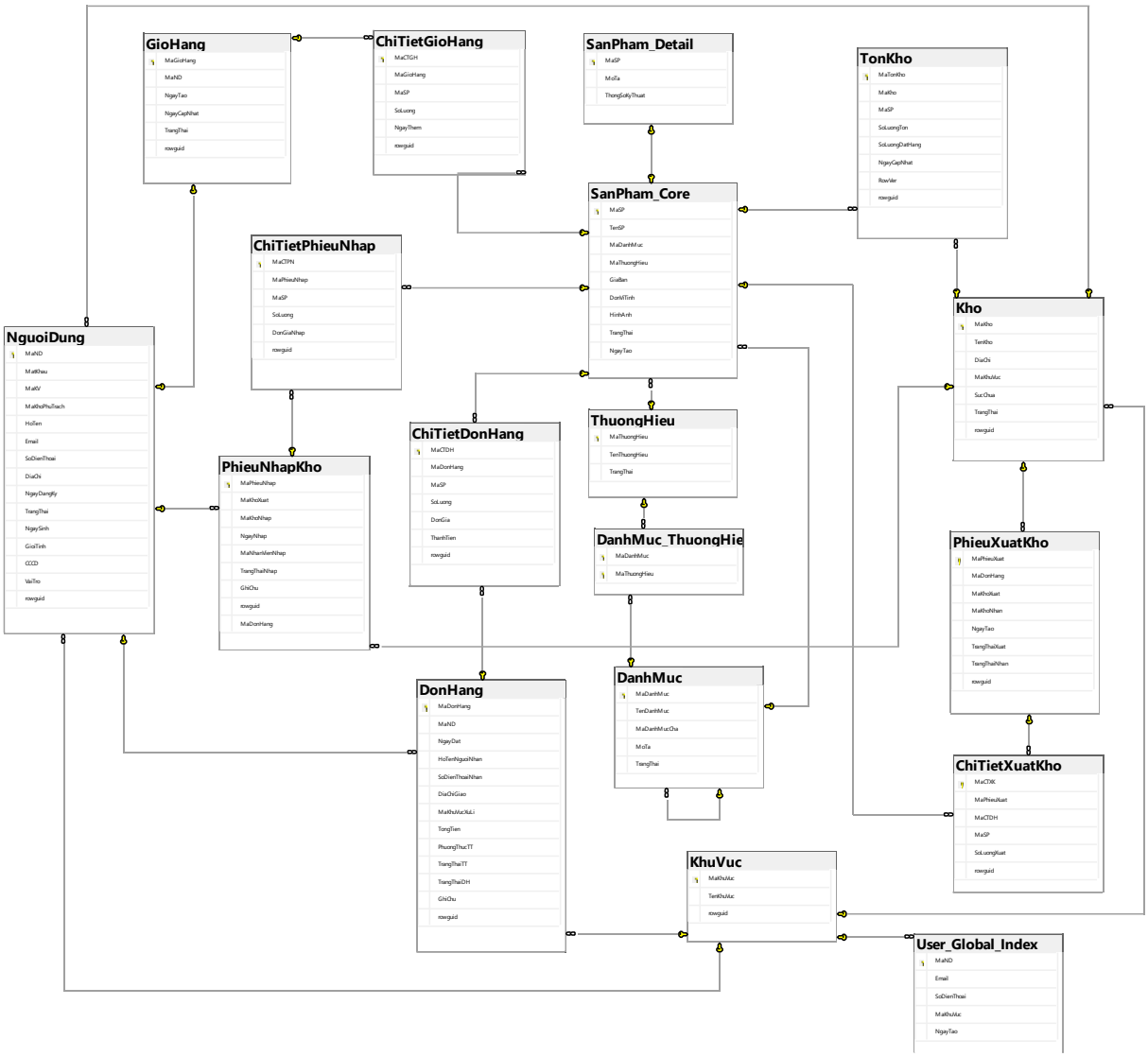
site nhằm tối ưu hóa chi phí truyền tải mạng cho các tác vụ đọc dữ liệu chiếm 80% tần suất hệ thống.

- Xây dựng Giao dịch phân tán: Thiết lập cơ chế kiểm soát đồng thời (Concurrency Control) cho luồng truy vấn xuyên site và cập nhật tồn kho đồng thời, đảm bảo tính nhất quán dữ liệu (ACID) trên toàn hệ thống đa kho.

CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

4.1. Thiết kế lược đồ toàn cục

4.1.1. Mô hình thực thể – liên kết (ER)



Hình 2 Mô hình ER của hệ thống

4.1.2. Lược đồ quan hệ toàn cục

1. KhuVuc

Bảng 2 Cấu trúc bảng KhuVuc

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaKhuVuc	VARCHAR(10)	PK	Mã khu vực

TenKhuVuc	NVARCHAR(100)		NOT NULL

2. Kho

Bảng 3 Cấu trúc bảng Kho

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaKho	VARCHAR(20)	PK	Mã kho
TenKho	NVARCHAR(100)		NOT NULL
DiaChi	NVARCHAR(300)		NOT NULL
MaKhuVuc	VARCHAR(10)	FK	NOT NULL, tham chiếu KhuVuc(MaKhuVuc)
SucChua	INT		NOT NULL, DEFAULT 0, CHECK SucChua >= 0
TrangThai	TINYINT		NOT NULL, DEFAULT 1, CHECK TrangThai IN (0,1)

3. DanhMuc

Bảng 4 Cấu trúc bảng DanhMuc

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaDanhMuc	VARCHAR(20)	PK	Mã danh mục
TenDanhMuc	NVARCHAR(100)		NOT NULL
MaDanhMucCha	VARCHAR(20)	FK	NULL, tự tham chiếu DanhMuc(MaDanhMuc)
MoTa	NVARCHAR(500)		NULL

TrangThai	TINYINT		NOT NULL, DEFAULT 1, CHECK TrangThai IN (0,1)

4. ThuongHieu

Bảng 5 Cấu trúc bảng ThuongHieu

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaThuongHieu	VARCHAR(20)	PK	Mã thương hiệu
TenThuongHieu	NVARCHAR(100)		NOT NULL, UNIQUE
TrangThai	TINYINT		NOT NULL, DEFAULT 1, CHECK TrangThai IN (0,1)

5. DanhMuc_ThuongHieu

Bảng 6 Cấu trúc bảng DanhMuc_ThuongHieu

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaDanhMuc	VARCHAR(20)	PK, FK	NOT NULL, tham chiếu DanhMuc(MaDanhMuc)
MaThuongHieu	VARCHAR(20)	PK, FK	NOT NULL, tham chiếu ThuongHieu(MaThuongHieu)

6. SanPham_Core

Bảng 7 Cấu trúc bảng SanPham_Core

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaSP	VARCHAR(20)	PK	Mã sản phẩm
TenSP	NVARCHAR(255)		NOT NULL

MaDanhMuc	VARCHAR(20)	FK	NOT NULL, tham chiếu DanhMuc(MaDanhMuc)
MaThuongHieu	VARCHAR(20)	FK	NOT NULL, tham chiếu ThuongHieu(MaThuongHieu)
GiaBan	DECIMAL(15,2)		NOT NULL, CHECK GiaBan >= 0
DonViTinh	NVARCHAR(20)		NOT NULL, DEFAULT N'Cái'
HinhAnh	VARCHAR(500)		NULL
TrangThai	TINYINT		NOT NULL, DEFAULT 1, CHECK TrangThai IN (0,1)
NgayTao	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()

7. SanPham_Detail

Bảng 8 Cấu trúc bảng SanPham_Detail

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaSP	VARCHAR(20)	PK, FK	Tham chiếu SanPham_Core(MaSP)
MoTa	NVARCHAR(MAX)		NULL
ThongSoKyThuat	NVARCHAR(MAX)		NULL, nếu có dữ liệu thì phải là JSON hợp lệ

8. NguoiDung

Bảng 9 Cấu trúc bảng NguoiDung

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú

MaND	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MatKhau	VARCHAR(255)		NOT NULL
MaKV	VARCHAR(10)	FK	NOT NULL, tham chiếu KhuVuc(MaKhuVuc)
MaKhoPhuTrach	VARCHAR(20)	FK	NULL, tham chiếu Kho(MaKho)
HoTen	NVARCHAR(100)		NOT NULL
Email	VARCHAR(100)		NOT NULL, UNIQUE
SoDienThoai	VARCHAR(15)		NULL, UNIQUE
DiaChi	NVARCHAR(300)		NULL
NgayDangKy	DATE		NOT NULL, DEFAULT CONVERT(DATE, GETDATE())
TrangThai	TINYINT		NOT NULL, DEFAULT 1, CHECK TrangThai IN (0,1)
NgaySinh	DATETIME2		NULL
GioiTinh	NVARCHAR(10)		NOT NULL, DEFAULT N'Nam'
CCCD	VARCHAR(12)		NULL, CHECK CCCD IS NULL OR LEN(CCCD) = 12
VaiTro	VARCHAR(20)		NOT NULL, DEFAULT 'USER', CHECK VaiTro IN ('ADMIN','WAREHOUSE_STAFF','USER')

9. User_Global_Index

Bảng 10 Cấu trúc bảng User_Global_Index

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú

MaND	UNIQUEIDENTIFIER	PK	Mã người dùng toàn cục
Email	VARCHAR(100)		NOT NULL, UNIQUE
SoDienThoai	VARCHAR(15)		NULL, UNIQUE
MaKhuVuc	VARCHAR(10)	FK	NOT NULL, tham chiếu KhuVuc(MaKhuVuc)
NgayTao	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()

10. GioHang

Bảng 11 Cấu trúc bảng GioHang

Thuộc tính	**Kiểu dữ liệu**	**Khóa**	**Ràng buộc / Ghi chú**
MaGioHang	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaND	UNIQUEIDENTIFIER	FK	NOT NULL, tham chiếu NguoiDung(MaND)
NgayTao	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()
NgayCapNhat	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()
TrangThai	VARCHAR(30)		NOT NULL, DEFAULT 'active', CHECK TrangThai IN ('active','ordered','cancelled','expired')

Ràng buộc bổ sung: mỗi người dùng chỉ có một giỏ hàng active.

11. ChiTietGioHang

Bảng 12 Cấu trúc bảng ChiTietGioHang

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaCTGH	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaGioHang	UNIQUEIDENTIFIER	FK	NOT NULL, tham chiếu GioHang(MaGioHang)
MaSP	VARCHAR(20)	FK	NOT NULL, tham chiếu SanPham_Core(MaSP)
SoLuong	INT		NOT NULL, CHECK SoLuong > 0
NgayThem	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()

Ràng buộc bổ sung: UNIQUE(MaGioHang, MaSP).

12. DonHang

Bảng 13 Cấu trúc bảng DonHang

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaDonHang	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaND	UNIQUEIDENTIFIER	FK	NOT NULL, tham chiếu NguoiDung(MaND)
NgayDat	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()
HoTenNguoiNhan	NVARCHAR(100)		NOT NULL

SoDienThoaiNhan	VARCHAR(15)		NOT NULL
DiaChiGiao	NVARCHAR(300)		NOT NULL
MaKhuVucXuLi	VARCHAR(10)	FK	NOT NULL, tham chiếu KhuVuc(MaKhuVuc)
TongTien	DECIMAL(15, 2)		NOT NULL, DEFAULT 0, CHECK TongTien >= 0
PhuongThucTT	VARCHAR(20)		NOT NULL, DEFAULT 'COD', CHECK PhuongThucTT = 'COD'
TrangThaiTT	VARCHAR(30)		NOT NULL, DEFAULT 'waiting_cod', CHECK TrangThaiTT IN ('waiting_cod','paid','failed','cancelled')
TrangThaiDH	VARCHAR(30)		NOT NULL, DEFAULT 'pending', CHECK TrangThaiDH IN ('pending','processing','shipping','completed','cancelled')
GhiChu	NVARCHAR(500)		NULL

13. ChiTietDonHang

Bảng 14 Cấu trúc bảng ChiTietDonHang

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú

MaCTDH	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaDonHang	UNIQUEIDENTIFIER	FK	NOT NULL, tham chiếu DonHang(MaDonHang)
MaSP	VARCHAR(20)	FK	NOT NULL, tham chiếu SanPham_Core(MaSP)
SoLuong	INT		NOT NULL, CHECK SoLuong > 0
DonGia	DECIMAL(15,2)		NOT NULL, CHECK DonGia >= 0
ThanhTien	Computed column		PERSISTED, tính bằng SoLuong * DonGia

14. PhieuXuatKho

Bảng 15 Cấu trúc bảng PhieuXuatKho

Thuộc tính	**Kiểu dữ liệu**	**Khóa**	**Ràng buộc / Ghi chú**
MaPhieuXuat	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaDonHang	UNIQUEIDENTIFIER	Logic	NOT NULL, liên kết logic tới DonHang
MaKhoXuat	VARCHAR(20)	FK	NOT NULL, tham chiếu Kho(MaKho)
MaKhoNhan	VARCHAR(20)	Logic	NULL nếu xuất trực tiếp cho khách
NgayTao	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()

TrangThaiXuat	VARCHAR(30)		NOT NULL, DEFAULT 'waiting_export', CHECK TrangThaiXuat IN ('waiting_export','exported','cancelled')
TrangThaiNhan	VARCHAR(30)		NULL, CHECK TrangThaiNhan IS NULL OR TrangThaiNhan IN ('waiting_receive','received')

Ràng buộc bổ sung: nếu MaKhoNhan NULL thì TrangThaiNhan phải NULL;

Nếu MaKhoNhan NOT NULL thì TrangThaiNhan phải NOT NULL.

Ràng buộc bổ sung: MaKhoNhan phải khác MaKhoXuat nếu MaKhoNhan khác NULL.

15. ChiTietXuatKho

Bảng 16 Cấu trúc bảng ChiTietXuatKho

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaCTXK	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaPhieuXuat	UNIQUEIDENTIFIER	FK	NOT NULL, tham chiếu PhieuXuatKho(MaPhieuXuat)
MaCTDH	UNIQUEIDENTIFIER	Logic	NULL, liên kết logic tới ChiTietDonHang
MaSP	VARCHAR(20)	FK	NOT NULL, tham chiếu SanPham_Core(MaSP)
SoLuongXuat	INT		NOT NULL, CHECK SoLuongXuat > 0

16. TonKho

Bảng 17 Cấu trúc bảng TonKho

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaTonKho	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaKho	VARCHAR(20)	FK	NOT NULL, tham chiếu Kho(MaKho)
MaSP	VARCHAR(20)	FK	NOT NULL, tham chiếu SanPham_Core(MaSP)
SoLuongTon	INT		NOT NULL, CHECK SoLuongTon >= 0
SoLuongDatHang	INT		NOT NULL, DEFAULT 0, CHECK SoLuongDatHang >= 0
NgayCapNhat	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()
RowVer	ROWVERSION		Dùng kiểm soát đồng thời

Ràng buộc bổ sung: UNIQUE(MaKho, MaSP).

Ràng buộc bổ sung: CHECK SoLuongTon >= SoLuongDatHang.

17. PhieuNhapKho

Bảng 18 Cấu trúc bảng PhieuNhapKho

Thuộc tính	Kiểu dữ liệu	Khóa	Ràng buộc / Ghi chú
MaPhieuNhap	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaKhoXuat	VARCHAR(20)	FK	NOT NULL, tham chiếu Kho(MaKho)

MaKhoNhap	VARCHAR(20)	FK	NOT NULL, tham chiếu Kho(MaKho)
NgayNhap	DATETIME2		NOT NULL, DEFAULT SYSDATETIME()
MaNhanVienNhap	UNIQUEIDENTIFIER	FK	NOT NULL, tham chiếu NguoiDung(MaND)
TrangThaiNhap	VARCHAR(30)		NOT NULL, DEFAULT 'waiting_import'
GhiChu	NVARCHAR(500)		NULL

18. ChiTietPhieuNhap

Bảng 19 Cấu trúc bảng ChiTietPhieuNhap

Thuộc tính	**Kiểu dữ liệu**	**Khóa**	**Ràng buộc / Ghi chú**
MaCTPN	UNIQUEIDENTIFIER	PK	DEFAULT NEWID()
MaPhieuNhap	UNIQUEIDENTIFIER	FK	NOT NULL, tham chiếu PhieuNhapKho(MaPhieuNhap)
MaSP	VARCHAR(20)	FK	NOT NULL, tham chiếu SanPham_Core(MaSP)
SoLuong	INT		NOT NULL, CHECK SoLuong > 0
DonGiaNhap	DECIMAL(15,2)		NOT NULL, CHECK DonGiaNhap >= 0

4.1.3. Mô tả các thực thể

Bảng 20: Mô tả các thực thể trong lược đồ toàn cục

STT	Thực thể	Ý nghĩa	Thuộc tính chính
1	KhuVuc	Lưu thông tin các khu vực vận hành trong hệ thống, ví dụ miền Bắc và miền Nam. Đây là thực thể nền tảng để phân chia dữ liệu, kho, người dùng và đơn hàng theo khu vực.	MaKhuVuc, TenKhuVuc
2	Kho	Lưu thông tin các kho hàng trong hệ thống. Mỗi kho thuộc một khu vực nhất định và là nơi quản lý tồn kho, xuất hàng, nhập hàng.	MaKho, TenKho, DiaChi, MaKhuVuc, SucChua, TrangThai
3	DanhMuc	Lưu thông tin danh mục sản phẩm. Danh mục giúp phân loại sản phẩm theo nhóm và có thể tổ chức theo dạng danh mục cha - con.	MaDanhMuc, TenDanhMuc, MaDanhMucCha, MoTa, TrangThai
4	ThuongHieu	Lưu thông tin thương hiệu sản phẩm. Thực thể này hỗ trợ quản lý, tìm kiếm và lọc sản phẩm theo thương hiệu.	MaThuongHieu, TenThuongHieu, TrangThai
5	DanhMuc_ThuongHieu	Lưu quan hệ kết hợp giữa danh mục và thương hiệu. Thực thể này cho biết một danh mục có những thương hiệu nào và một thương hiệu có thể xuất hiện trong những danh mục nào.	MaDanhMuc, MaThuongHieu
6	SanPham_Core	Lưu thông tin cơ bản của sản phẩm, phục vụ cho các nghiệp vụ bán hàng, tra cứu, đặt hàng và quản lý tồn kho. Đây là phần thông tin sản phẩm được sử dụng thường xuyên trong hệ thống.	MaSP, TenSP, MaDanhMuc, MaThuongHieu, GiaBan, DonViTinh, HinhAnh, TrangThai, NgayTao

STT	Thực thể	Ý nghĩa	Thuộc tính chính
7	SanPham_Detail	Lưu thông tin chi tiết mở rộng của sản phẩm như mô tả và thông số kỹ thuật. Thực thể này tách riêng khỏi thông tin sản phẩm cơ bản để giảm tải cho các truy vấn thông thường.	MaSP, MoTa, ThongSoKyThuat
8	NguoiDung	Lưu thông tin người dùng trong hệ thống, bao gồm khách hàng, nhân viên kho và quản trị viên. Thực thể này phục vụ đăng nhập, phân quyền và xử lý nghiệp vụ theo vai trò.	MaND, MatKhau, MaKV, MaKhoPhuTrach, HoTen, Email, SoDienThoai, DiaChi, NgayDangKy, TrangThai, NgaySinh, GioiTinh, CCCD, VaiTro
9	User_Global_Index	Lưu thông tin định danh người dùng ở phạm vi toàn hệ thống. Thực thể này dùng để kiểm soát tính duy nhất của email và số điện thoại trong môi trường phân tán.	MaND, Email, SoDienThoai, MaKhuVuc, NgayTao
10	GioHang	Lưu thông tin giỏ hàng của người dùng. Giỏ hàng thể hiện trạng thái mua sắm tạm thời trước khi người dùng tạo đơn hàng.	MaGioHang, MaND, NgayTao, NgayCapNhat, TrangThai
11	ChiTietGioHang	Lưu chi tiết các sản phẩm có trong giỏ hàng, bao gồm sản phẩm và số lượng khách muốn mua.	MaCTGH, MaGioHang, MaSP, SoLuong, NgayThem
12	DonHang	Lưu thông tin đơn hàng của khách hàng, bao gồm thông tin người nhận, địa chỉ giao hàng, khu vực xử lý, tổng tiền, phương thức thanh toán và trạng thái đơn hàng.	MaDonHang, MaND, NgayDat, HoTenNguoiNhan, SoDienThoaiNhan, DiaChiGiao, MaKhuVucXuLi,

STT	Thực thể	Ý nghĩa	Thuộc tính chính
			TongTien, PhuongThucTT, TrangThaiTT, TrangThaiDH, GhiChu
13	ChiTietDonHang	Lưu danh sách sản phẩm trong từng đơn hàng, bao gồm số lượng mua, đơn giá và thành tiền. Đây là thực thể chi tiết của đơn hàng.	MaCTDH, MaDonHang, MaSP, SoLuong, DonGia, ThanhTien
14	PhieuXuatKho	Lưu thông tin phiếu xuất kho. Phiếu xuất có thể dùng cho xuất nội bộ giữa các kho hoặc xuất trực tiếp để giao cho khách hàng.	MaPhieuXuat, MaDonHang, MaKhoXuat, MaKhoNhan, NgayTao, TrangThaiXuat, TrangThaiNhan
15	ChiTietXuatKho	Lưu chi tiết sản phẩm được xuất trong từng phiếu xuất kho, bao gồm sản phẩm, số lượng xuất và chi tiết đơn hàng liên quan nếu có.	MaCTXK, MaPhieuXuat, MaCTDH, MaSP, SoLuongXuat
16	TonKho	Lưu số lượng tồn kho của từng sản phẩm tại từng kho. Đây là thực thể quan trọng để kiểm tra tồn kho, giữ hàng và cập nhật số lượng khi xuất hoặc nhập kho.	MaTonKho, MaKho, MaSP, SoLuongTon, SoLuongDatHang, NgayCapNhat, RowVer
17	PhieuNhapKho	Lưu thông tin phiếu nhập kho, thường dùng khi hàng được nhập về kho hoặc được chuyển từ kho này sang kho khác.	MaPhieuNhap, MaKhoXuat, MaKhoNhap, NgayNhap, MaNhanVienNhap, TrangThaiNhap, GhiChu

STT	Thực thể	Ý nghĩa	Thuộc tính chính
18	ChiTietPhieuNhap	Lưu chi tiết sản phẩm trong từng phiếu nhập kho, bao gồm sản phẩm, số lượng nhập và đơn giá nhập.	MaCTPN, MaPhieuNhap, MaSP, SoLuong, DonGiaNhap

4.2. Phân mảnh dữ liệu

4.2.1. Phân mảnh ngang theo kho / khu vực

Trong hệ thống bán hàng đa kho, dữ liệu được phân tán theo khu vực vận hành. Hệ thống gồm hai site chính: SITE_BAC quản lý dữ liệu khu vực Bắc và SITE_NAM quản lý dữ liệu khu vực Nam. Phương án phân mảnh ngang được áp dụng cho các quan hệ có dữ liệu phát sinh theo khu vực, theo kho hoặc phụ thuộc trực tiếp vào các quan hệ đã được phân mảnh.

Mục tiêu của phương án này là đưa dữ liệu về gần nơi phát sinh nghiệp vụ, giúp các thao tác như kiểm tra tồn kho, đặt hàng, xuất kho và nhập kho được xử lý chủ yếu tại site cục bộ. Các truy vấn toàn hệ thống sẽ được thực hiện bằng cách hợp nhất dữ liệu từ các mảnh tại site chính.

a. Phân mảnh quan hệ Kho

Quan hệ Kho được phân mảnh theo khu vực quản lý của kho. Các kho thuộc khu vực Bắc được lưu tại SITE_BAC, các kho thuộc khu vực Nam được lưu tại SITE_NAM.

Biểu diễn toán học:

Kho_BAC = σ MaKhuVuc = 'BAC' (Kho)

Kho_NAM = σ MaKhuVuc = 'NAM' (Kho)

Quan hệ gốc được khôi phục bằng:

Kho = Kho_BAC ∪ Kho_NAM

Cách phân mảnh này phù hợp vì mỗi kho chỉ thuộc một khu vực duy nhất, đồng thời các nghiệp vụ tồn kho, xuất kho và nhập kho đều gắn với kho.

b. Phân mảnh quan hệ TonKho

Quan hệ TonKho được phân mảnh dẫn xuất theo quan hệ Kho. Tồn kho của kho nào sẽ được lưu tại site quản lý kho đó.

Biểu diễn toán học:

TonKho_BAC = σ MaKho ∈ π MaKho (Kho_BAC) (TonKho)
TonKho_NAM = σ MaKho ∈ π MaKho (Kho_NAM) (TonKho)
Quan hệ gốc được khôi phục bằng:
TonKho = TonKho_BAC ∪ TonKho_NAM

Việc phân mảnh TonKho theo MaKho giúp các thao tác cập nhật số lượng tồn, giữ hàng, xuất hàng và nhập hàng được thực hiện tại site cục bộ.

c. Phân mảnh quan hệ NguoiDung

Quan hệ NguoiDung được phân mảnh theo khu vực người dùng hoặc kho mà nhân viên phụ trách. Khách hàng được phân theo MaKV, còn nhân viên kho được phân theo MaKhoPhuTrach.

Biểu diễn toán học:

NguoiDung_BAC = σ MaKV = 'BAC' OR MaKhoPhuTrach ∈ π MaKho (Kho_BAC) (NguoiDung)
NguoiDung_NAM = σ MaKV = 'NAM' OR MaKhoPhuTrach ∈ π MaKho (Kho_NAM) (NguoiDung)

Quan hệ gốc được khôi phục bằng:

NguoiDung = NguoiDung_BAC ∪ NguoiDung_NAM

Cách phân mảnh này giúp dữ liệu người dùng được đặt tại site gần với khu vực hoặc kho mà người dùng liên quan.

d. Phân mảnh quan hệ GioHang

Quan hệ GioHang được phân mảnh dẫn xuất theo quan hệ NguoiDung. Giỏ hàng của người dùng thuộc site nào sẽ được lưu tại site đó.

Biểu diễn toán học:

GioHang_BAC = σ MaND ∈ π MaND (NguoiDung_BAC) (GioHang)

GioHang_NAM = σ MaND ∈ π MaND (NguoiDung_NAM) (GioHang)

Quan hệ gốc được khôi phục bằng:

GioHang = GioHang_BAC ∪ GioHang_NAM

Phương án này giúp các thao tác thêm sản phẩm vào giỏ hàng và chuyển giỏ hàng thành đơn hàng được xử lý cục bộ.

e. Phân mảnh quan hệ ChiTietGioHang

Quan hệ ChiTietGioHang được phân mảnh dẫn xuất theo quan hệ GioHang. Chi tiết giỏ hàng được lưu cùng site với giỏ hàng cha.

Biểu diễn toán học:

ChiTietGioHang_BAC = σ MaGioHang ∈ π MaGioHang (GioHang_BAC) (ChiTietGioHang)

ChiTietGioHang_NAM = σ MaGioHang ∈ π MaGioHang (GioHang_NAM) (ChiTietGioHang)

Quan hệ gốc được khôi phục bằng:

ChiTietGioHang = ChiTietGioHang_BAC ∪ ChiTietGioHang_NAM

Cách phân mảnh này tránh việc phải join dữ liệu giỏ hàng và chi tiết giỏ hàng giữa nhiều site.

f. Phân mảnh quan hệ DonHang

Quan hệ DonHang được phân mảnh theo khu vực xử lý đơn hàng. Đơn hàng do khu vực Bắc xử lý được lưu tại SITE_BAC, đơn hàng do khu vực Nam xử lý được lưu tại SITE_NAM.

Biểu diễn toán học:

DonHang_BAC = σ MaKhuVucXuLi = 'BAC' (DonHang)

DonHang_NAM = σ MaKhuVucXuLi = 'NAM' (DonHang)

Quan hệ gốc được khôi phục bằng:

DonHang = DonHang_BAC ∪ DonHang_NAM

Phân mảnh theo MaKhuVucXuLi phù hợp với nghiệp vụ vì đơn hàng được xử lý chủ yếu tại khu vực chịu trách nhiệm.

g. Phân mảnh quan hệ ChiTietDonHang

Quan hệ ChiTietDonHang được phân mảnh dẫn xuất theo quan hệ DonHang. Chi tiết đơn hàng được lưu cùng site với đơn hàng cha.

Biểu diễn toán học:

ChiTietDonHang_BAC = σ MaDonHang ∈ π MaDonHang (DonHang_BAC) (ChiTietDonHang)

ChiTietDonHang_NAM = σ MaDonHang ∈ π MaDonHang (DonHang_NAM) (ChiTietDonHang)

Quan hệ gốc được khôi phục bằng:

ChiTietDonHang = ChiTietDonHang_BAC ∪ ChiTietDonHang_NAM

Phương án này giúp truy vấn đơn hàng và danh sách sản phẩm trong đơn được thực hiện tại cùng một site.

h. Phân mảnh quan hệ PhieuXuatKho

Quan hệ PhieuXuatKho được phân mảnh theo kho xuất hàng. Phiếu xuất của kho Bắc được lưu tại SITE_BAC, phiếu xuất của kho Nam được lưu tại SITE_NAM.

Biểu diễn toán học:

PhieuXuatKho_BAC = σ MaKhoXuat ∈ π MaKho (Kho_BAC) (PhieuXuatKho)

PhieuXuatKho_NAM = σ MaKhoXuat ∈ π MaKho (Kho_NAM) (PhieuXuatKho)

Quan hệ gốc được khôi phục bằng:

PhieuXuatKho = PhieuXuatKho_BAC ∪ PhieuXuatKho_NAM

Phân mảnh theo MaKhoXuat là phù hợp vì khi xác nhận xuất hàng, hệ thống cần cập nhật tồn kho tại chính kho xuất. Ngay cả trường hợp xuất hàng nội bộ từ Nam ra Bắc, phiếu xuất vẫn thuộc site Nam nếu kho xuất là kho Nam.

j. Phân mảnh quan hệ ChiTietXuatKho

Quan hệ ChiTietXuatKho được phân mảnh dẫn xuất theo quan hệ PhieuXuatKho. Chi tiết xuất kho được lưu cùng site với phiếu xuất kho.

Biểu diễn toán học:

ChiTietXuatKho_BAC = σ MaPhieuXuat ∈ π MaPhieuXuat (PhieuXuatKho_BAC) (ChiTietXuatKho)
ChiTietXuatKho_NAM = σ MaPhieuXuat ∈ π MaPhieuXuat (PhieuXuatKho_NAM) (ChiTietXuatKho)

Quan hệ gốc được khôi phục bằng:

ChiTietXuatKho = ChiTietXuatKho_BAC ∪ ChiTietXuatKho_NAM

Cách phân mảnh này giúp nghiệp vụ xác nhận xuất kho được xử lý toàn bộ tại site của kho xuất.

k. Phân mảnh quan hệ PhieuNhapKho

Quan hệ PhieuNhapKho được phân mảnh theo kho nhập hàng. Phiếu nhập của kho Bắc được lưu tại SITE_BAC, phiếu nhập của kho Nam được lưu tại SITE_NAM.

Biểu diễn toán học:

PhieuNhapKho_BAC = σ MaKhoNhap ∈ π MaKho (Kho_BAC) (PhieuNhapKho)
PhieuNhapKho_NAM = σ MaKhoNhap ∈ π MaKho (Kho_NAM) (PhieuNhapKho)
Quan hệ gốc được khôi phục bằng:
PhieuNhapKho = PhieuNhapKho_BAC ∪ PhieuNhapKho_NAM

Phân mảnh theo MaKhoNhap phù hợp vì khi xác nhận nhập hàng, tồn kho của kho nhận sẽ được cập nhật tại site quản lý kho đó.

l. Phân mảnh quan hệ ChiTietPhieuNhap

Quan hệ ChiTietPhieuNhap được phân mảnh dẫn xuất theo quan hệ PhieuNhapKho. Chi tiết phiếu nhập được lưu cùng site với phiếu nhập kho.

Biểu diễn toán học:

ChiTietPhieuNhap_BAC = σ MaPhieuNhap ∈ π MaPhieuNhap (PhieuNhapKho_BAC) (ChiTietPhieuNhap)
ChiTietPhieuNhap_NAM = σ MaPhieuNhap ∈ π MaPhieuNhap (PhieuNhapKho_NAM) (ChiTietPhieuNhap)

Quan hệ gốc được khôi phục bằng:

ChiTietPhieuNhap = ChiTietPhieuNhap_BAC ∪ ChiTietPhieuNhap_NAM

Phương án này đảm bảo quá trình nhập hàng và cập nhật tồn kho được xử lý tại cùng một site.

n. Đánh giá tính đúng đắn của phương án phân mảnh

Phương án phân mảnh ngang đảm bảo tính đầy đủ vì toàn bộ dữ liệu của mỗi quan hệ gốc đều được đưa vào một trong các mảnh tại SITE_BAC hoặc SITE_NAM.

Với mỗi quan hệ R được phân mảnh thành hai mảnh R_BAC và R_NAM, ta có:

$$R = R_BAC \cup R_NAM$$

Phương án cũng đảm bảo tính không giao nhau vì mỗi bản ghi chỉ thuộc một site duy nhất dựa trên điều kiện phân mảnh. Ví dụ, một kho không thể vừa thuộc khu vực Bắc vừa thuộc khu vực Nam.

Biểu diễn toán học:

$$R_BAC \cap R_NAM = \emptyset$$

Ngoài ra, phương án đảm bảo khả năng khôi phục dữ liệu gốc. Khi cần truy vấn toàn hệ thống, quan hệ ban đầu có thể được tái tạo bằng phép hợp các mảnh dữ liệu tại hai site.

Ví dụ:

Kho = Kho_BAC ∪ Kho_NAM

TonKho = TonKho_BAC ∪ TonKho_NAM

DonHang = DonHang_BAC ∪ DonHang_NAM

PhieuXuatKho = PhieuXuatKho_BAC ∪ PhieuXuatKho_NAM

PhieuNhapKho = PhieuNhapKho_BAC ∪ PhieuNhapKho_NAM

4.2.2. Phân mảnh dọc

Bên cạnh phân mảnh ngang theo khu vực và theo kho, hệ thống còn áp dụng phân mảnh dọc đối với một số quan hệ có nhiều nhóm thuộc tính khác nhau về mục đích sử dụng. Phân mảnh dọc được sử dụng để tách các thuộc tính thường xuyên truy cập khỏi các thuộc tính ít truy cập, thuộc tính có dung lượng lớn hoặc thuộc tính nhạy cảm.

a. Phân mảnh dọc quan hệ SanPham

Trong thiết kế dữ liệu, thông tin sản phẩm được tách thành hai phần: SanPham_Core và SanPham_Detail.

SanPham_Core lưu các thuộc tính cơ bản thường xuyên được sử dụng trong tra cứu, hiển thị danh sách sản phẩm, đặt hàng và kiểm tra tồn kho. Các thuộc tính này bao gồm mã sản phẩm, tên sản phẩm, danh mục, thương hiệu, giá bán, đơn vị tính, hình ảnh, trạng thái và ngày tạo.

SanPham_Detail lưu các thuộc tính chi tiết hơn như mô tả sản phẩm và thông số kỹ thuật. Đây là nhóm dữ liệu có dung lượng lớn hơn và không phải lúc nào cũng cần truy cập trong các nghiệp vụ thông thường.

Biểu diễn phân mảnh dọc:

SanPham_Core = π MaSP, TenSP, MaDanhMuc, MaThuongHieu, GiaBan, DonViTinh, HinhAnh, TrangThai, NgayTao (SanPham)

SanPham_Detail = π MaSP, MoTa, ThongSoKyThuat (SanPham)

Quan hệ gốc được khôi phục bằng:

SanPham = SanPham_Core ⋈ SanPham_Detail

Cách tách này giúp các truy vấn phổ biến như tìm kiếm sản phẩm, hiển thị danh sách sản phẩm, kiểm tra giá bán và xử lý đơn hàng chỉ cần truy cập SanPham_Core. Các thông tin chi tiết như mô tả dài hoặc thông số kỹ thuật chỉ được truy cập khi người dùng xem chi tiết sản phẩm.

b. Phân mảnh dọc dữ liệu người dùng

Dữ liệu người dùng được tách thành hai phần: NguoiDung và User_Global_Index.

Quan hệ NguoiDung lưu thông tin đầy đủ của người dùng tại site tương ứng, bao gồm thông tin đăng nhập, thông tin cá nhân, khu vực, kho phụ trách, trạng thái và vai trò. Đây là dữ liệu phục vụ cho các nghiệp vụ cục bộ như đặt hàng, xử lý đơn hàng, phân quyền nhân viên kho và quản lý tài khoản tại từng site.

Quan hệ User_Global_Index lưu một phần thông tin định danh quan trọng của người dùng ở mức toàn hệ thống, bao gồm MaND, Email, SoDienThoai, MaKhuVuc và NgayTao. Mục đích của mảnh này là hỗ trợ kiểm tra tính duy nhất toàn cục của email và số điện thoại trong môi trường phân tán.

Biểu diễn phân mảnh dọc:

NguoiDung_Local = π MaND, MatKhau, MaKV, MaKhoPhuTrach, HoTen, DiaChi, NgayDangKy, TrangThai, NgaySinh, GioiTinh, CCCD, VaiTro (NguoiDung)
User_Global_Index = π MaND, Email, SoDienThoai, MaKhuVuc, NgayTao (NguoiDung)

Thông tin người dùng đầy đủ có thể được liên kết thông qua khóa MaND:

NguoiDung_Full = NguoiDung_Local ⋈ User_Global_Index

Trong đó, User_Global_Index đóng vai trò như một mảnh định danh toàn cục. Mảnh này giúp hệ thống kiểm soát trùng lặp email và số điện thoại giữa các site, tránh trường hợp cùng một email hoặc số điện thoại được đăng ký tại nhiều site khác nhau.

4.2.3. Giải thích lựa chọn phân mảnh

Phương án phân mảnh được lựa chọn dựa trên đặc điểm vận hành của hệ thống bán hàng đa kho. Trong hệ thống này, phần lớn nghiệp vụ phát sinh tại từng khu vực và từng kho cụ thể. Mỗi kho chủ yếu thao tác trên dữ liệu của chính mình như tồn kho, phiếu xuất, phiếu nhập và xử lý đơn hàng liên quan. Vì vậy, việc phân mảnh dữ liệu theo khu vực và theo kho là phù hợp với mẫu truy cập thực tế của hệ thống.

Đối với các bảng như Kho, TonKho, PhieuXuatKho, PhieuNhapKho, dữ liệu được phân mảnh theo kho hoặc khu vực quản lý. Kho thuộc miền Bắc được lưu tại SITE_BAC, kho thuộc miền Nam được lưu tại SITE_NAM. Cách phân mảnh này giúp các thao tác thường xuyên như kiểm tra tồn kho, giữ hàng, xác nhận xuất kho và xác nhận nhập kho được thực hiện ngay tại site quản lý kho đó, không cần truy vấn sang site khác trong hầu hết trường hợp.

Việc phân mảnh ngang cũng giúp giảm chi phí truyền thông giữa các site. Nếu toàn bộ dữ liệu tồn kho được lưu tập trung, mỗi lần kho Bắc hoặc kho Nam xử lý đơn hàng đều phải gửi yêu cầu về máy chủ trung tâm. Điều này làm tăng độ trễ và tạo áp lực lên đường truyền. Với phương án phân mảnh theo kho, các truy vấn cục bộ chỉ xử lý dữ liệu tại site tương ứng. Site chính chỉ tham gia khi cần tổng hợp báo cáo toàn hệ thống hoặc khi đơn hàng cần phối hợp giữa nhiều kho ở nhiều khu vực khác nhau.

Các bảng chi tiết như ChiTietDonHang, ChiTietXuatKho, ChiTietPhieuNhap được phân mảnh dẫn xuất theo bảng cha. Ví dụ, chi tiết đơn hàng được lưu cùng site với đơn hàng; chi tiết phiếu xuất được lưu cùng site với phiếu xuất; chi tiết phiếu nhập được lưu cùng site với phiếu nhập. Cách làm này giúp hạn chế các phép nối phân tán, vì các bảng thường xuyên được truy vấn cùng nhau đã được đặt tại cùng một site.

Đối với dữ liệu người dùng, bảng NguoiDung được phân mảnh theo khu vực người dùng hoặc kho mà nhân viên phụ trách. Điều này phù hợp vì khách hàng thường đặt hàng tại khu vực của mình, còn nhân viên kho chủ yếu thao tác với kho thuộc site mình quản lý. Tuy nhiên, để đảm bảo tính duy nhất của tài khoản trên toàn hệ thống, bảng User_Global_Index được tách riêng và lưu ở mức toàn cục. Bảng này hỗ trợ kiểm tra trùng email và số điện thoại khi tạo tài khoản mới trong môi trường phân tán.

Đối với dữ liệu sản phẩm, hệ thống tách thành SanPham_Core và SanPham_Detail. SanPham_Core chứa các thông tin cơ bản như mã sản phẩm, tên sản phẩm, danh mục, thương hiệu, giá bán và trạng thái. Đây là nhóm dữ liệu được truy cập thường xuyên nên có thể nhân bản đến các site để phục vụ tra cứu nhanh. SanPham_Detail chứa mô tả và thông số kỹ thuật, là nhóm dữ liệu ít dùng hơn và có dung lượng lớn hơn. Cách tách này giúp các truy vấn thông thường không phải đọc những thông tin chi tiết không cần thiết.

Phương án phân mảnh này cũng hỗ trợ khả năng mở rộng hệ thống. Khi phát sinh thêm khu vực hoặc thêm kho mới, hệ thống có thể bổ sung site hoặc mở rộng phân vùng dữ liệu theo kho mà không cần thay đổi toàn bộ thiết kế. Dữ liệu của kho mới sẽ được đặt tại site quản lý kho đó, còn các bảng chi tiết liên quan tiếp tục được phân mảnh dẫn xuất theo bảng cha. Nhờ vậy, hệ thống có thể mở rộng dần theo quy mô vận hành thực tế.

Nhìn chung, lựa chọn phân mảnh theo khu vực, theo kho và phân mảnh dẫn xuất theo bảng cha là phù hợp với đặc điểm của hệ thống bán hàng đa kho. Phương án này giúp tăng tính cục bộ của truy vấn, giảm chi phí truyền thông, hạn chế join phân tán và tạo nền tảng tốt để mở rộng hệ thống khi số lượng kho hoặc khu vực tăng lên.

4.3. Cấp phát dữ liệu

4.3.1. Dữ liệu lưu cục bộ tại từng vùng

a. Minh họa dữ liệu lưu tại LOCAL_SERVER

SITE_BAC lưu các dữ liệu liên quan đến kho và người dùng miền Bắc. Ví dụ:

- Các kho có MaKhuVuc = 'BAC'.
- Tồn kho của các kho Bắc như KB01, KB02.
- Người dùng thuộc khu vực Bắc.
- Giỏ hàng của người dùng miền Bắc.
- Đơn hàng có MaKhuVucXuLi = 'BAC'.
- Chi tiết của các đơn hàng thuộc SITE_BAC.
- Phiếu xuất có MaKhoXuat là kho Bắc.
- Chi tiết phiếu xuất của các phiếu xuất tại kho Bắc.
- Phiếu nhập có MaKhoNhap là kho Bắc.

- Chi tiết phiếu nhập của các phiếu nhập tại kho Bắc.

Ví dụ, nếu kho KB01 xử lý một đơn hàng, thì tồn kho của KB01, phiếu xuất của KB01 và các cập nhật liên quan đến xuất hàng sẽ được xử lý tại SITE_BAC.

SITE_NAM lưu các dữ liệu liên quan đến kho và người dùng miền Nam. Ví dụ:

- Các kho có MaKhuVuc = 'NAM'.
- Tồn kho của các kho Nam như KN01, KN02.
- Người dùng thuộc khu vực Nam.
- Giỏ hàng của người dùng miền Nam.
- Đơn hàng có MaKhuVucXuLi = 'NAM'.
- Chi tiết của các đơn hàng thuộc SITE_NAM.
- Phiếu xuất có MaKhoXuat là kho Nam.
- Chi tiết phiếu xuất của các phiếu xuất tại kho Nam.
- Phiếu nhập có MaKhoNhap là kho Nam.
- Chi tiết phiếu nhập của các phiếu nhập tại kho Nam.

Ví dụ, nếu kho KN01 xuất hàng nội bộ sang kho KB01, phiếu xuất được lưu tại SITE_NAM vì KN01 là kho xuất. Trong khi đó, phiếu nhập chờ nhận tại KB01 được lưu tại SITE_BAC vì KB01 là kho nhập.

b. Dữ liệu đặt tại MAIN_SERVER

Một số dữ liệu không nên chỉ lưu cục bộ tại từng site, mà cần đặt tại site trung tâm hoặc được quản lý tập trung để đảm bảo tính nhất quán toàn hệ thống.

Bảng quan trọng nhất đặt tại site trung tâm là User_Global_Index. Bảng này lưu các thông tin định danh toàn cục của người dùng như MaND, Email, SoDienThoai, MaKhuVuc và NgayTao. Mục đích là kiểm soát trùng lặp email và số điện thoại khi người dùng đăng ký tài khoản ở các site khác nhau.

Ngoài ra, site trung tâm có thể giữ bản chuẩn của các dữ liệu tham chiếu dùng chung như KhuVuc, DanhMuc, ThuongHieu, DanhMuc_ThuongHieu và SanPham_Core. Các

bảng này có thể được nhân bản xuống SITE_BAC và SITE_NAM để phục vụ tra cứu nhanh tại từng site.

Đối với SanPham_Detail, tùy theo quy mô hệ thống, dữ liệu này có thể đặt tại site trung tâm hoặc nhân bản xuống các site. Vì SanPham_Detail chứa mô tả và thông số kỹ thuật, dung lượng có thể lớn hơn SanPham_Core và không phải lúc nào cũng cần truy cập trong các nghiệp vụ đặt hàng thông thường.

4.3.2. Ma trận cấp phát dữ liệu

Bảng 21 Ma trận cấp phát dữ liệu

Quan hệ / Mảnh dữ liệu	**SITE_BAC**	**SITE_NAM**	**SITE_TRUNG_TAM**	**Ghi chú**
KhuVuc	R	R	L	Dữ liệu tham chiếu dùng chung, bản chính đặt tại trung tâm
DanhMuc	R	R	L	Dữ liệu đọc nhiều, ít thay đổi
ThuongHieu	R	R	L	Dữ liệu dùng chung cho lọc và tra cứu sản phẩm
DanhMuc_ThuongHieu	R	R	L	Dữ liệu liên kết danh mục - thương hiệu
SanPham_Core	R	R	L	Thông tin sản phẩm cơ bản, cần có tại các site kho để đặt hàng và kiểm tra tồn

SanPham_Detail	–	–	L	Thông tin chi tiết sản phẩm, dung lượng lớn hơn, đặt tập trung
Kho_BAC	L	–	R	Mảnh kho thuộc khu vực Bắc
Kho_NAM	–	L	R	Mảnh kho thuộc khu vực Nam
TonKho_BAC	L	–	–	Tồn kho của các kho Bắc, xử lý tại SITE_BAC
TonKho_NAM	–	L	–	Tồn kho của các kho Nam, xử lý tại SITE_NAM
NguoiDung_BAC	L	–	–	Người dùng thuộc khu vực Bắc hoặc nhân viên phụ trách kho Bắc
NguoiDung_NAM	–	L	–	Người dùng thuộc khu vực Nam hoặc nhân viên phụ trách kho Nam
User_Global_Index	–	–	L	Dùng để kiểm tra duy nhất toàn cục Email, Số điện thoại

GioHang_BAC	L	–	–	Giỏ hàng của người dùng thuộc SITE_BAC
GioHang_NAM	–	L	–	Giỏ hàng của người dùng thuộc SITE_NAM
ChiTietGioHang_BAC	L	–	–	Lưu cùng site với GioHang_BAC
ChiTietGioHang_NAM	–	L	–	Lưu cùng site với GioHang_NAM
DonHang_BAC	L	–	–	Đơn hàng do khu vực Bắc xử lý
DonHang_NAM	–	L	–	Đơn hàng do khu vực Nam xử lý
ChiTietDonHang_BAC	L	–	–	Lưu cùng site với DonHang_BAC
ChiTietDonHang_NAM	–	L	–	Lưu cùng site với DonHang_NAM
PhieuXuatKho_BAC	L	–	–	Phiếu xuất của các kho Bắc
PhieuXuatKho_NAM	–	L	–	Phiếu xuất của các kho Nam

ChiTietXuatKho_BAC	L	–	–	Lưu cùng site với PhieuXuatKho_BAC
ChiTietXuatKho_NAM	–	L	–	Lưu cùng site với PhieuXuatKho_NAM
PhieuNhapKho_BAC	L	–	–	Phiếu nhập của các kho Bắc
PhieuNhapKho_NAM	–	L	–	Phiếu nhập của các kho Nam
ChiTietPhieuNhap_BAC	L	–	–	Lưu cùng site với PhieuNhapKho_BAC
ChiTietPhieuNhap_NAM	–	L	–	Lưu cùng site với PhieuNhapKho_NAM

4.4. Nhân bản và đồng bộ dữ liệu

4.4.1. Dữ liệu dùng chung cần nhân bản

Các dữ liệu cần nhân bản

KhuVuc

- Là dữ liệu nền tảng để xác định các khu vực trong hệ thống.
- Được dùng trong các nghiệp vụ liên quan đến kho, người dùng và đơn hàng.
- Dữ liệu ít thay đổi, dung lượng nhỏ.
- Nên đặt bản chính tại site trung tâm và nhân bản sang các site kho.

DanhMuc

- Lưu thông tin danh mục sản phẩm.

- Được dùng khi hiển thị, tìm kiếm và lọc sản phẩm.
- Các site đều cần dữ liệu danh mục để xử lý nghiệp vụ bán hàng.
- Dữ liệu ít thay đổi nên phù hợp để nhân bản.

ThuongHieu

- Lưu thông tin thương hiệu sản phẩm.
- Phục vụ chức năng lọc sản phẩm theo thương hiệu.
- Được truy cập tại nhiều site khi hiển thị hoặc xử lý sản phẩm.
- Nên nhân bản để mỗi site có thể tra cứu cục bộ.

DanhMuc_ThuongHieu

- Là bảng liên kết giữa danh mục và thương hiệu.
- Hỗ trợ lọc thương hiệu theo từng danh mục sản phẩm.
- Do phụ thuộc vào DanhMuc và ThuongHieu, bảng này cũng cần được nhân bản cùng hai bảng trên.
- Giúp các site có đủ dữ liệu để lọc sản phẩm mà không cần truy vấn xuyên site.

SanPham_Core

- Lưu thông tin cơ bản của sản phẩm như mã sản phẩm, tên sản phẩm, danh mục, thương hiệu, giá bán, đơn vị tính, hình ảnh và trạng thái.
- Được dùng thường xuyên trong các nghiệp vụ:
    - Hiển thị danh sách sản phẩm.
    - Tạo đơn hàng.
    - Kiểm tra tồn kho.
    - Tạo chi tiết đơn hàng.
    - Xuất kho và nhập kho.
- Đây là dữ liệu đọc nhiều, ít thay đổi hơn dữ liệu tồn kho.
- Cần được nhân bản sang các site để xử lý nghiệp vụ nhanh tại từng khu vực.

SanPham_Detail

- Lưu thông tin chi tiết của sản phẩm như mô tả và thông số kỹ thuật.
- Được dùng khi người dùng xem chi tiết sản phẩm.
- Trong thiết kế này, SanPham_Detail cũng được nhân bản sang các site để tăng tốc độ đọc cho người dùng.
- Khi người dùng ở khu vực Bắc hoặc Nam xem chi tiết sản phẩm, hệ thống có thể đọc dữ liệu ngay tại site gần người dùng thay vì gọi về site trung tâm.
- Việc nhân bản bảng này giúp cải thiện trải nghiệm xem sản phẩm, đặc biệt với các thông tin chi tiết có thể được truy cập nhiều từ phía khách hàng.

Lợi ích của việc nhân bản dữ liệu dùng chung

- Giảm truy vấn xuyên site vì mỗi site đã có sẵn dữ liệu danh mục, thương hiệu và sản phẩm.
- Tăng tốc độ đọc dữ liệu cho người dùng tại từng khu vực.
- Giảm phụ thuộc vào site trung tâm trong các thao tác tra cứu sản phẩm.
- Tăng tính sẵn sàng của hệ thống: nếu site trung tâm tạm thời không truy cập được, các site kho vẫn có thể phục vụ các chức năng đọc sản phẩm.
- Giảm tải đường truyền giữa các site.
- Phù hợp với các bảng dữ liệu ít thay đổi nhưng được đọc thường xuyên.

4.4.2. Cơ chế đồng bộ

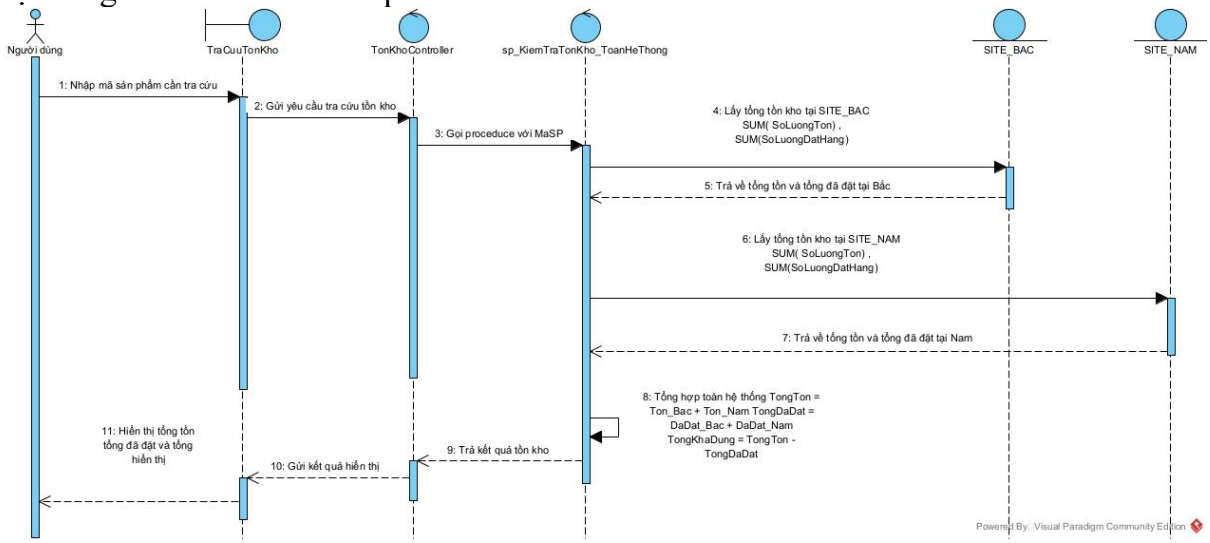
Trong dự án, cơ chế đồng bộ dữ liệu được triển khai dựa trên kỹ thuật Nhân bản Snapshot (Snapshot Replication) của SQL Server. Theo đó, toàn bộ dữ liệu từ cơ sở dữ liệu nguồn (DB) trên Server 1 - đóng vai trò Publisher - được sao chép định kỳ đến cơ sở dữ liệu đích (DB_REP) trên Server 2 - đóng vai trò Subscriber - thông qua một server trung gian gọi là Distributor. Quá trình đồng bộ được thực hiện tự động bởi hai thành phần chính: Snapshot Agent chịu trách nhiệm tạo các file snapshot chứa lược đồ và dữ liệu của các bảng cần nhân bản, sau đó lưu vào thư mục trên Distributor; Distribution Agent tiếp nhận các file này và áp dụng lên Subscriber theo mô hình Push Subscription - tức Publisher chủ động đẩy dữ liệu xuống mà không chờ yêu cầu từ phía Subscriber. Cơ chế

này đảm bảo tính nhất quán tiềm ẩn (Latent Transactional Consistency) giữa hai server, phù hợp với bài toán đồng bộ dữ liệu một chiều trong môi trường nội bộ.

CHƯƠNG 5. MÔ PHỎNG NGHIỆP VỤ PHÂN TÁN

5.1. Tình huống 1: Tra cứu tồn kho của một sản phẩm trên toàn hệ thống

Mô tả : Chức năng này dùng để kiểm tra tổng số lượng tồn kho của một sản phẩm trên toàn hệ thống. Người dùng chỉ cần nhập mã sản phẩm, ví dụ SP003. Hệ thống sẽ lấy số lượng tồn của sản phẩm này từ site Bắc và site Nam, sau đó cộng lại để biết toàn hệ thống còn bao nhiêu sản phẩm có thể bán.



Hình 3 Sơ đồ luồng tra cứu tồn kho toàn hệ thống

Kết quả truy vấn đối với sản phẩm SP003

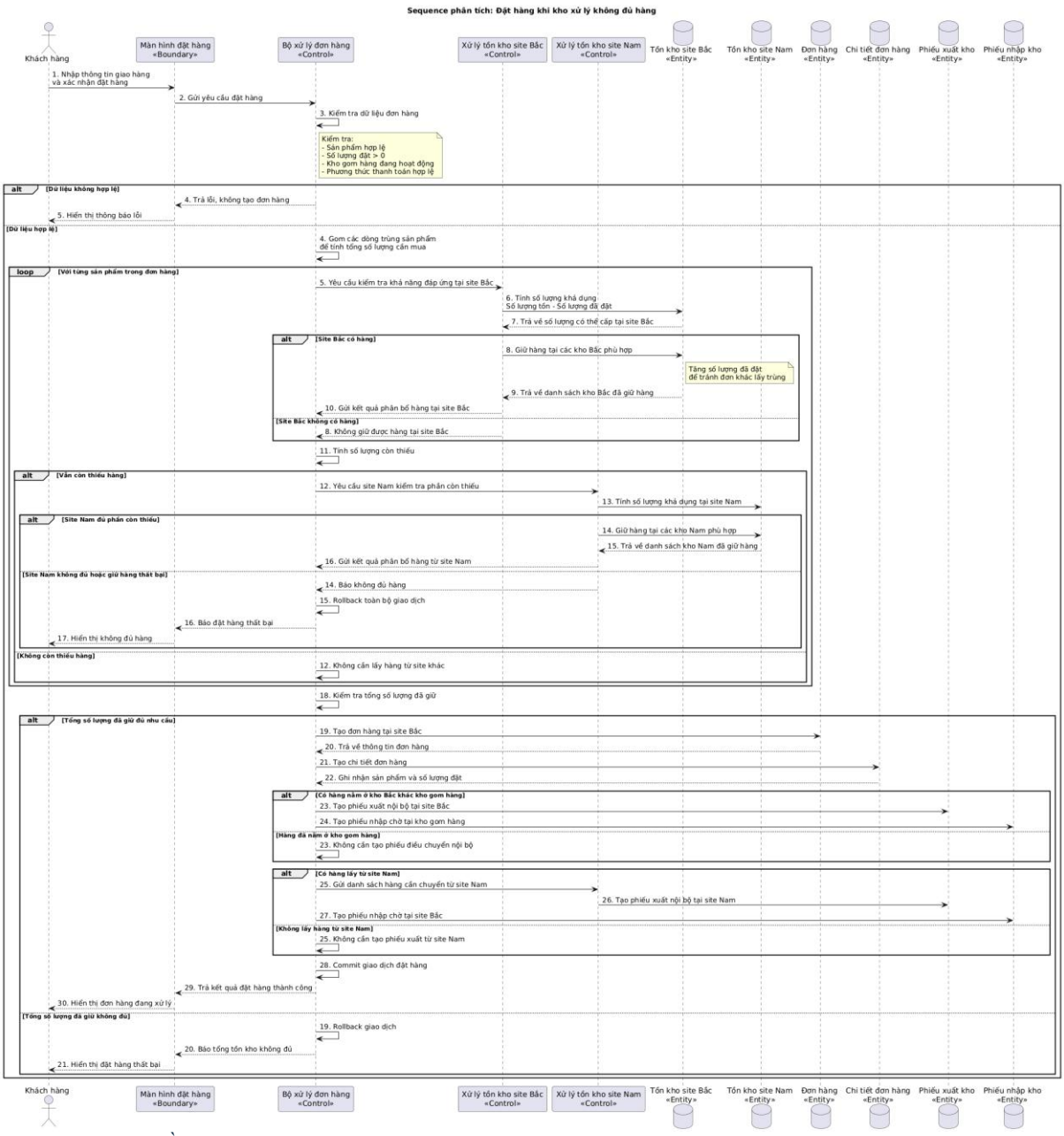
Results | Messages

	SiteNguon	MaKhuVuc	MaKho	TenKho	MaSP	TenSP	SoLuongTon	SoLuongDatHang	SoLuongKhaDung
2	BAC	Bac	KB02	Kho Bắc 02	SP003	Xiaomi Redmi Note 13	0	0	0
3	BAC	Bac	KB03	Kho Bắc 03	SP003	Xiaomi Redmi Note 13	0	0	0
4	BAC	Bac	KB04	Kho Bắc 04	SP003	Xiaomi Redmi Note 13	0	0	0
5	BAC	Bac	KB05	Kho Bắc 05	SP003	Xiaomi Redmi Note 13	0	0	0
6	NAM	Nam	KN01	Kho Nam 01	SP003	Xiaomi Redmi Note 13	26	0	26
7	NAM	Nam	KN02	Kho Nam 02	SP003	Xiaomi Redmi Note 13	0	0	0
8	NAM	Nam	KN03	Kho Nam 03	SP003	Xiaomi Redmi Note 13	1	0	1
9	NAM	Nam	KN04	Kho Nam 04	SP003	Xiaomi Redmi Note 13	0	0	0
10	NAM	Nam	KN05	Kho Nam 05	SP003	Xiaomi Redmi Note 13	0	0	0

	MaSP	TenSP	TongSoLuongTon	TongSoLuongDatHang	TongSoLuongKhaDung	SoKhoDangCoHang
1	SP003	Xiaomi Redmi Note 13	27	0	27	10

5.2. Tình huống 2: Đặt hàng khi một kho không đủ số lượng

Mô tả : Khách hàng thuộc miền Bắc đặt mua sản phẩm SP003 với số lượng 20. Do khách hàng thuộc miền Bắc nên hệ thống xác định site xử lý chính là site Bắc. Kho gom hàng của đơn là KB01. Tuy nhiên, kho KB01 không đủ số lượng để đáp ứng toàn bộ đơn hàng, vì vậy hệ thống phải kiểm tra thêm các kho khác trong site Bắc. Nếu site Bắc vẫn không đủ hàng, hệ thống tiếp tục kiểm tra khả năng đáp ứng từ site Nam.



Hình 4 phân tích luồng đặt hàng khi kho xử lý không đủ hàng

Bước 1: Tiếp nhận yêu cầu đặt hàng

Coordinator nhận request đặt hàng gồm: khách hàng thuộc khu vực BAC, sản phẩm SP003, số lượng 20, kho gom hàng KB01, phương thức thanh toán COD.

Trước khi xử lý, hệ thống kiểm tra sản phẩm có tồn tại không, còn kinh doanh không, số lượng đặt có hợp lệ không và kho KB01 có đang hoạt động không. Nếu dữ liệu không hợp lệ, hệ thống dừng và không tạo đơn.

Bước 2: Gom nhu cầu theo sản phẩm

Nếu trong đơn có nhiều dòng cùng sản phẩm, hệ thống gom lại để biết tổng số lượng cần mua. Ví dụ sau khi gom, đơn hàng cần:

- SP003: 20 sản phẩm

Bước 3: Kiểm tra tồn kho tại site Bắc

Hệ thống kiểm tra số lượng khả dụng tại các kho thuộc site Bắc theo công thức:

Số lượng khả dụng = SoLuongTon - SoLuongDatHang

Giả sử kết quả:

- KB01 khả dụng 5 sản phẩm
- KB02 khả dụng 7 sản phẩm

Tổng site Bắc đáp ứng được:

- 5 + 7 = 12 sản phẩm

Khách cần 20 sản phẩm, nên site Bắc còn thiếu:

- 20 - 12 = 8 sản phẩm

Bước 4: Kiểm tra site Nam

Do site Bắc không đủ hàng, Coordinator gửi yêu cầu sang site Nam thông qua linked server. Site Nam tự kiểm tra tồn kho của nó và trả về kết quả. Giả sử kho KN01 tại site Nam có khả dụng 8 sản phẩm.

Khi đó hệ thống tổng hợp:

- KB01 cấp 5 sản phẩm
- KB02 cấp 7 sản phẩm
- KN01 cấp 8 sản phẩm

Tổng cộng = 20 sản phẩm

Vì tổng số lượng đủ đáp ứng đơn hàng, hệ thống tiếp tục xử lý đặt hàng.

Bước 5: Giữ hàng tại từng kho

Trước khi tạo đơn hàng, hệ thống giữ hàng tại từng kho được phân bổ:

- KB01 giữ 5 sản phẩm
- KB02 giữ 7 sản phẩm
- KN01 giữ 8 sản phẩm

Ở bước này, hệ thống chưa trừ SoLuongTon, mà tăng SoLuongDatHang. Điều này có nghĩa là hàng đã được giữ cho đơn hàng này và đơn hàng khác không được dùng lại phần hàng đó.

Khi giữ hàng, hệ thống luôn kiểm tra:

- SoLuongTon - SoLuongDatHang >= Số lượng cần giữ

Nếu không đủ, giao dịch bị rollback để tránh âm tồn kho.

Bước 6: Khóa dữ liệu để tránh lỗi đồng thời

Trong lúc giữ hàng, hệ thống khóa dòng tồn kho đang xử lý. Ví dụ, khi giữ sản phẩm SP003 tại kho KB01, dòng tồn kho của SP003 - KB01 sẽ bị khóa trong transaction. Nhờ đó, hai đơn hàng không thể cùng giữ trùng một lượng hàng tại cùng thời điểm.

Bước 7: Tạo đơn hàng và chi tiết đơn hàng

Sau khi giữ đủ hàng, hệ thống tạo đơn hàng tại site Bắc với trạng thái:

- TrangThaiDH = processing

Sau đó tạo chi tiết đơn hàng:

- Đơn hàng DH001
- Sản phẩm SP003
- Số lượng 20

Trạng thái processing thể hiện đơn hàng đã được tạo, hàng đã được giữ, nhưng có thể vẫn đang chờ gom từ kho khác về KB01.

Bước 8: Tạo phiếu xuất nội bộ và phiếu nhập chờ

Vì KB01 là kho gom hàng nên những phần hàng ở kho khác phải được chuyển về KB01.

Hệ thống tạo phiếu xuất nội bộ:

- KB02 -> KB01: 7 sản phẩm
- KN01 -> KB01: 8 sản phẩm

Đồng thời, hệ thống tạo phiếu nhập chờ tại KB01 để chờ nhận hàng từ KB02 và KN01.

Bước 9: Commit giao dịch đặt hàng

Nếu các bước kiểm tra, giữ hàng, tạo đơn, tạo chi tiết đơn, tạo phiếu xuất và phiếu nhập đều thành công, hệ thống commit giao dịch. Sau bước này, đơn hàng được tạo thành công, hàng đã được giữ và đơn hàng ở trạng thái processing.

Bước 10: Xác nhận xuất nội bộ

Khi kho KB02 hoặc KN01 thực sự xuất hàng về KB01, nhân viên kho xác nhận phiếu xuất. Lúc này hệ thống mới trừ tồn kho thật tại kho xuất:

- SoLuongTon = SoLuongTon - SoLuongXuat
- SoLuongDatHang = SoLuongDatHang - SoLuongXuat

Ví dụ, KN01 xuất 8 sản phẩm thì tồn kho tại KN01 giảm 8 và số lượng đã đặt cũng giảm 8.

Bước 11: Xác nhận nhập tại KB01

Khi hàng được chuyển đến KB01, nhân viên kho KB01 xác nhận nhập hàng. Hệ thống cộng hàng vào kho KB01:

- SoLuongTon = SoLuongTon + SoLuongNhap
- SoLuongDatHang = SoLuongDatHang + SoLuongNhap

Lý do tăng cả SoLuongDatHang là vì hàng tuy đã về KB01 nhưng vẫn đang được giữ cho đơn hàng này, không được bán cho đơn khác.

Bước 12: Tạo và xác nhận phiếu xuất giao khách

Khi KB01 đã gom đủ 20 sản phẩm, hệ thống tạo phiếu xuất giao khách. Phiếu này có MaKhoNhan = NULL, thể hiện đây là xuất cho khách chứ không phải chuyển kho.

Khi kho KB01 xác nhận xuất hàng cho khách, hệ thống trừ tồn kho thật:

- SoLuongTon = SoLuongTon - 20
- SoLuongDatHang = SoLuongDatHang - 20

Sau đó đơn hàng chuyển sang trạng thái:

- TrangThaiDH = shipping

Trường hợp lỗi

Nếu trong quá trình xử lý có kho không giữ được hàng, ví dụ site Nam không còn đủ 8 sản phẩm do đơn hàng khác đã giữ trước, hệ thống rollback giao dịch. Khi rollback, đơn hàng không được tạo, phiếu xuất và phiếu nhập không được tạo sai, hàng không bị giữ treo và tồn kho không bị âm.

5.3. Phân tích nghiệp vụ phân tán

Tình huống 1 : Tra cứu tồn kho của một sản phẩm trên toàn hệ thống

- **Dữ liệu được truy xuất từ những site nào**

Trong procedure, dữ liệu được truy xuất từ hai site chính:

- Bac
- Nam

Tại mỗi site, hệ thống lấy dữ liệu từ các bảng:

- TonKho: lấy số lượng tồn và số lượng đã đặt.
- Kho: lấy thông tin kho và khu vực của kho.
- SanPham_Core: lấy tên sản phẩm.

Cụ thể:

- Từ *Bac*, hệ thống lấy tồn kho của các kho thuộc site Bắc.
- Từ *Nam*, hệ thống lấy tồn kho của các kho thuộc site Nam.

Nếu người dùng truyền mã sản phẩm, hệ thống chỉ lấy tồn kho của sản phẩm đó. Nếu không truyền mã sản phẩm, hệ thống lấy tồn kho của tất cả sản phẩm.

- **Cách tổng hợp dữ liệu**

Đầu tiên, hệ thống tạo bảng tạm *#TonKhoAll* để lưu chung dữ liệu tồn kho lấy từ các site.

Dữ liệu từ site Bắc và site Nam đều được đưa vào bảng tạm này với cùng một cấu trúc:

- SiteNguon

- MaKho
- TenKho
- MaKhuVuc
- MaSP
- TenSP
- SoLuongTon
- SoLuongDatHang
- SoLuongKhaDung

Trong đó, số lượng khả dụng được tính theo công thức:

- SoLuongKhaDung = SoLuongTon - SoLuongDatHang

Sau khi lấy dữ liệu từ cả hai site, hệ thống trả ra hai kết quả.

Kết quả thứ nhất là chi tiết tồn kho theo từng kho:

- Sản phẩm SP003 ở kho nào
- Kho đó thuộc site nào
- Số lượng tồn là bao nhiêu
- Số lượng đã đặt là bao nhiêu
- Số lượng khả dụng còn lại là bao nhiêu

Ví dụ:

- KB01 - SP003 - khả dụng 5
- KB02 - SP003 - khả dụng 7
- KN01 - SP003 - khả dụng 8

Kết quả thứ hai là tổng hợp toàn hệ thống theo sản phẩm:

- Tổng số lượng tồn
- Tổng số lượng đã đặt
- Tổng số lượng khả dụng
- Số kho có ghi nhận sản phẩm đó

Ví dụ:

SP003:

- Tổng tồn = 25
- Tổng đã đặt = 5
- Tổng khả dụng = 20

Như vậy, hệ thống vừa cho biết chi tiết từng kho, vừa cho biết tổng tồn kho của sản phẩm trên toàn hệ thống.

- **Tác động đến chi phí truyền dữ liệu**

Chức năng này có phát sinh chi phí truyền dữ liệu vì Coordinator phải truy vấn dữ liệu từ cả *Bac* và *Nam* thông qua linked server.

Chi phí truyền dữ liệu phụ thuộc vào điều kiện lọc:

- Nếu truyền MaSP cụ thể: dữ liệu truyền ít, chi phí thấp.
- Nếu không truyền MaSP: hệ thống lấy tồn kho của tất cả sản phẩm, dữ liệu truyền nhiều hơn.
- Nếu không truyền MaKho và MaKhuVuc: hệ thống phải lấy dữ liệu từ tất cả kho ở cả hai site.

Cách xử lý trong procedure giúp giảm chi phí ở mức nhất định vì điều kiện lọc được đặt ngay trong từng truy vấn đến site Bắc và site Nam. Nhờ đó, mỗi site chỉ trả về các dòng phù hợp với mã sản phẩm, mã kho hoặc khu vực cần tra cứu.

Tuy nhiên, nếu số lượng sản phẩm và số lượng kho lớn, việc lấy dữ liệu từ nhiều site vẫn có thể làm tăng thời gian truy vấn do phải truyền dữ liệu qua mạng.

Tình huống 2: Đặt hàng khi một kho không đủ số lượng

- **Dữ liệu được truy xuất từ suy nào :**

Dữ liệu được truy xuất chủ yếu từ **hai site**:

Site Bắc là site nhận và xử lý đơn hàng. Tại đây hệ thống truy xuất các bảng như SanPham_Core, Kho, TonKho, DonHang, ChiTietDonHang, PhieuXuatKho, PhieuNhapKho. Site Bắc dùng dữ liệu sản phẩm để kiểm tra sản phẩm còn bán hay không, dùng bảng tồn kho để kiểm tra số lượng khả dụng, sau đó tạo đơn hàng, chi tiết đơn hàng, phiếu xuất nội bộ, phiếu nhập chờ và phiếu xuất giao khách.

Site Nam là site được truy xuất khi site Bắc không đủ hàng. Site Bắc gọi sang site Nam qua linked server [LINK], cụ thể là gọi procedure sp_GiuHang_1SanPham_NoiBo để giữ phần hàng còn thiếu, sau đó gọi sp_TaoPhieuXuat_Batch_NoiBo_Json để tạo phiếu xuất nội bộ tại site Nam. Như vậy, site Nam không cần gửi toàn bộ bảng tồn kho về site Bắc mà chỉ thực hiện xử lý cục bộ rồi trả về kết quả phân bổ hàng.

- **Cách tổng hợp dữ liệu**

Hệ thống không xử lý từng dòng sản phẩm một cách rời rạc ngay từ đầu, mà trước tiên **gom nhu cầu theo mã sản phẩm**

Sau đó, với từng sản phẩm, hệ thống thực hiện theo luồng:

- **Bước 1**: Kiểm tra tổng khả dụng ở site Bắc
    - KhaDung = SoLuongTon - SoLuongDatHang
- **Bước 2**: Nếu Bắc đủ
    - Giữ hàng tại các kho Bắc

- **Bước 3**: Nếu Bắc thiếu
  - Giữ phần có thể lấy ở Bắc
  - Phần còn thiếu gọi sang site Nam để giữ tiếp
- **Bước 4**: Tổng hợp kết quả giữ hàng vào #PhanBo
- **Bước 5**: Kiểm tra tổng số lượng đã giữ có đủ nhu cầu không
- **Bước 6**: Tạo đơn hàng, chi tiết đơn hàng, phiếu xuất nội bộ, phiếu nhập chờ

Bảng tạm quan trọng nhất trong luồng này là #PhanBo. Bảng này lưu kết quả phân bổ hàng theo từng kho:

Bảng 22 Bảng tạm phân bổ dữ liệu

MaKho	MaSP	SoLuongGiu	Isremote
KB01	SP003	5	0
KB02	SP003	3	0
KN01	SP003	7	1

Trong đó:

- IsRemote = 0 nghĩa là hàng được giữ ở site Bắc.
- IsRemote = 1 nghĩa là hàng được giữ ở site Nam.
- SoLuongGiu là số lượng đã được "đặt giữ" bằng cách tăng SoLuongDatHang.

Sau khi có bảng phân bổ, hệ thống chia thành hai nhóm:

Bảng 23 Các nhóm xử lý phân bổ đơn hàng

Nhóm 1 :	Hàng nằm ở site Bắc Nếu kho xuất khác kho gom hàng KB01 => tạo phiếu xuất nội bộ Bắc, ví dụ KB02 -> KB01 => tạo phiếu nhập chờ tại KB01
Nhóm 2 :	Hàng nằm ở site Nam => gửi danh sách phân bổ sang Nam dưới dạng JSON => Nam tạo phiếu xuất nội bộ, ví dụ KN01 -> KB01 => Bắc tạo phiếu nhập chờ tại KB01

- **Tác động đến chi phí truyền dữ liệu**

Trong cách thiết kế này, chi phí truyền dữ liệu được giảm vì hệ thống không join trực tiếp nhiều bảng lớn giữa hai site. Thay vào đó, site Bắc gọi procedure tại site

Nam để site Nam tự xử lý dữ liệu cục bộ. Site Nam chỉ trả về kết quả cần thiết gồm MaKho, MaSP, SoLuongGiu, thay vì trả toàn bộ bảng TonKho.

Bảng 24 So sánh hiệu quả truy vấn phân tán trực tiếp và gọi procedure

Không tốt	Tốt
SELECT * FROM [LINK].[store_management].dbo.TonKho WHERE MaSP = 'SP003';	EXEC [LINK].[store_management].dbo.sp_GiuHang_1SanPham_NoiBo

Có lợi là :

- Dữ liệu xử lý chính nằm ở site nào thì xử lý tại site đó.
- Site Bắc chỉ nhận kết quả phân bổ cuối cùng.
- Giảm số dòng dữ liệu truyền qua mạng.
- Giảm số lần truy vấn phân tán trực tiếp.
- Giảm nguy cơ lock dữ liệu lâu trên bảng remote.

- **Cách đảm bảo không âm tồn kho khi có nhiều đơn hàng đồng thời**

Khi nhiều khách hàng cùng đặt một sản phẩm tại cùng thời điểm, hệ thống có thể gặp lỗi cạnh tranh tồn kho.

Ví dụ, sản phẩm SP003 tại kho KN01 đang có:

Sản phẩm	Kho	Số lượng tồn	Số lượng đã đặt
SP003	KN01	10	0

Số lượng có thể bán là:

Số lượng khả dụng = Số lượng tồn - Số lượng đã đặt

= 10 - 0

= 10

Nếu hai khách hàng cùng đặt 8 sản phẩm, cả hai có thể cùng nhìn thấy kho còn 10 sản phẩm. Nếu hệ thống không kiểm soát đồng thời, cả hai đơn đều có thể đặt thành công, tức là hệ thống bán 16 sản phẩm trong khi kho chỉ có 10 sản phẩm. Khi xuất hàng thật, tồn kho có thể bị âm.

Để tránh lỗi này, hệ thống dùng cơ chế giữ hàng trước, trừ tồn kho sau.

Khi khách đặt hàng thành công, hệ thống chưa trừ ngay số lượng tồn. Thay vào đó, hệ thống tăng số lượng đã đặt.

Ví dụ khách thứ nhất đặt 8 sản phẩm:

- Số lượng tồn = 10
- Số lượng đã đặt = 0 + 8 = 8
- Số lượng khả dụng còn lại = 10 - 8 = 2

Lúc này, kho vẫn còn 10 sản phẩm thực tế, nhưng chỉ còn 2 sản phẩm có thể bán tiếp. Phần 8 sản phẩm đã được giữ cho đơn hàng trước đó nên đơn hàng khác không được dùng lại.

Trước khi giữ hàng, hệ thống luôn kiểm tra điều kiện:

- Số lượng tồn - Số lượng đã đặt >= Số lượng cần giữ

Nếu khách thứ hai tiếp tục đặt 8 sản phẩm, hệ thống kiểm tra:

- 10 - 8 >= 8
- 2 >= 8

Điều kiện này sai, vì kho chỉ còn 2 sản phẩm khả dụng. Vì vậy, hệ thống không cho giữ thêm 8 sản phẩm nữa. Đơn hàng này sẽ bị từ chối hoặc chuyển sang kiểm tra kho khác.

Để tránh hai giao dịch cùng cập nhật một dòng tồn kho, hệ thống khóa dòng dữ liệu tồn kho trong quá trình kiểm tra và giữ hàng. Nhờ vậy, một sản phẩm trong một kho tại một thời điểm chỉ được một giao dịch giữ hàng trước. Giao dịch khác phải chờ hoặc kiểm tra lại sau khi giao dịch trước hoàn tất.

Khi nhân viên kho xác nhận xuất hàng, hệ thống mới trừ tồn kho thật:

- Số lượng tồn = Số lượng tồn - Số lượng xuất
- Số lượng đã đặt = Số lượng đã đặt - Số lượng xuất

Ví dụ kho KN01 đã giữ 8 sản phẩm. Khi xác nhận xuất 8 sản phẩm:

- Số lượng tồn = 10 - 8 = 2
- Số lượng đã đặt = 8 - 8 = 0

Trước khi xuất kho, hệ thống cũng kiểm tra:

- Số lượng tồn >= Số lượng xuất
- Số lượng đã đặt >= Số lượng xuất

Điều này đảm bảo chỉ được xuất phần hàng đã được giữ trước đó, không xuất vượt tồn kho thực tế.

CHƯƠNG 6. CÁC TRUY VẤN PHÂN TÁN

6.1. Truy vấn 1: Kiểm tra sản phẩm còn hàng ở những kho nào

Mục đích truy vấn

Truy vấn này dùng để kiểm tra một sản phẩm cụ thể còn hàng tại những kho nào trên toàn hệ thống.

Người dùng nhập mã sản phẩm, ví dụ:

DECLARE @MaSP VARCHAR(20) = 'SP001';

Hệ thống sẽ kiểm tra tồn kho của sản phẩm đó ở cả hai site:

- SITE_BAC
- SITE_NAM

Một kho được xem là còn hàng khi số lượng khả dụng > 0.

Công thức tính số lượng khả dụng:

SoLuongKhaDung = SoLuongTon - SoLuongDatHang

Trong đó:

Bảng 25 Ý nghĩa các thuộc tính tính toán tồn kho

Thuộc tính	Ý nghĩa
SoLuongTon	Tổng số lượng sản phẩm đang có trong kho
SoLuongDatHang	Số lượng đã được khách đặt hoặc đang bị giữ
SoLuongKhaDung	Số lượng thực tế còn có thể bán

Câu truy vấn (SQL / giả mã):

```
DECLARE @MaSP VARCHAR(20) = 'SP001';
SELECT
    SiteNguon, MaKhuVuc, MaKho, TenKho, MaSP, TenSP, SoLuongTon,
    SoLuongDatHang,
    SoLuongTon - SoLuongDatHang AS SoLuongKhaDung
FROM
(
```

```
    SELECT
        'SITE_BAC' AS SiteNguon, k.MaKhuVuc, tk.MaKho, k.TenKho,
        tk.MaSP, sp.TenSP, tk.SoLuongTon, tk.SoLuongDatHang
    FROM [SITE_BAC].[store_management].dbo.TonKho tk
    JOIN [SITE_BAC].[store_management].dbo.Kho k
        ON tk.MaKho = k.MaKho
    JOIN [SITE_BAC].[store_management].dbo.SanPham_Core sp
        ON tk.MaSP = sp.MaSP
    WHERE tk.MaSP = @MaSP
    UNION ALL
    SELECT
        'SITE_NAM' AS SiteNguon, k.MaKhuVuc, tk.MaKho, k.TenKho,
        tk.MaSP, sp.TenSP, tk.SoLuongTon, tk.SoLuongDatHang
    FROM [SITE_NAM].[store_management].dbo.TonKho tk
    JOIN [SITE_NAM].[store_management].dbo.Kho k
        ON tk.MaKho = k.MaKho
    JOIN [SITE_NAM].[store_management].dbo.SanPham_Core sp
        ON tk.MaSP = sp.MaSP
    WHERE tk.MaSP = @MaSP
) AS T
WHERE SoLuongTon - SoLuongDatHang > 0
ORDER BY SiteNguon, MaKho;
```

Phân tích dữ liệu phân tán

	SiteNguon	MaKhuVuc	MaKho	TenKho	MaSP	TenSP	SoLuongTon	SoLuongDatHang	SoLuongKhaDung
1	SITE_BAC	Bac	KB01	Kho Bắc 01	SP001	iPhone 15 128GB	18	0	18
2	SITE_BAC	Bac	KB02	Kho Bắc 02	SP001	iPhone 15 128GB	50	5	45
3	SITE_BAC	Bac	KB03	Kho Bắc 03	SP001	iPhone 15 128GB	50	5	45
4	SITE_BAC	Bac	KB04	Kho Bắc 04	SP001	iPhone 15 128GB	50	6	44
5	SITE_BAC	Bac	KB05	Kho Bắc 05	SP001	iPhone 15 128GB	50	5	45
6	SITE_NAM	Nam	KN01	Kho Nam 01	SP001	iPhone 15 128GB	10	2	8
7	SITE_NAM	Nam	KN02	Kho Nam 02	SP001	iPhone 15 128GB	10	2	8
8	SITE_NAM	Nam	KN03	Kho Nam 03	SP001	iPhone 15 128GB	10	2	8
9	SITE_NAM	Nam	KN04	Kho Nam 04	SP001	iPhone 15 128GB	10	2	8
10	SITE_NAM	Nam	KN05	Kho Nam 05	SP001	iPhone 15 128GB	10	2	8

Hình 5 Phân tích dữ liệu phân tán: Kết quả truy vấn sản phẩm còn hàng

Truy vấn này liên quan trực tiếp đến bảng TonKho, Kho và SanPham_Core.

Bảng TonKho là bảng có tính địa phương cao vì tồn kho của kho nào thì được quản lý tại site chứa kho đó. Vì vậy, tồn kho của các kho miền Bắc nằm ở SITE_BAC, còn tồn kho của các kho miền Nam nằm ở SITE_NAM.

Bảng Kho cũng được phân mảnh theo khu vực, vì mỗi kho thuộc một khu vực cụ thể. Khi truy vấn tồn kho, cần join TonKho với Kho để lấy thông tin kho như TenKho, MaKhuVuc.

Bảng SanPham_Core là dữ liệu sản phẩm lõi, thường được nhân bản toàn phần sang các site. Nhờ vậy, mỗi site có thể tự join với SanPham_Core để lấy tên sản phẩm mà không cần gọi về site chính.

Điểm quan trọng của truy vấn là điều kiện:

```
WHERE tk.MaSP = @MaSP
```

được đặt bên trong từng truy vấn con tại SITE_BAC và SITE_NAM. Điều này giúp giảm lượng dữ liệu truyền qua mạng, vì mỗi site chỉ trả về dòng tồn kho của sản phẩm cần kiểm tra, thay vì gửi toàn bộ bảng TonKho về site chính.

6.2. Truy vấn 2: Tổng tồn kho của một sản phẩm trên toàn hệ thống

Mục đích truy vấn

Truy vấn này dùng để tính tổng tồn kho của một sản phẩm trên toàn bộ hệ thống, không cần biết chi tiết sản phẩm đó nằm ở kho nào.

Kết quả trả về gồm:

Bảng 26 Ý nghĩa các thuộc tính kết quả truy vấn tổng tồn kho

Thuộc tính	Ý nghĩa

MaSP	Mã sản phẩm
TenSP	Tên sản phẩm
TongSoLuongTon	Tổng số lượng tồn của sản phẩm ở tất cả kho
TongSoLuongDatHang	Tổng số lượng đang được đặt/giữ
TongSoLuongKhaDung	Tổng số lượng còn có thể bán

Câu truy vấn (SQL / giả mã):

```
DECLARE @MaSP_Tong VARCHAR(20) = 'SP001';
SELECT
    MaSP, MAX(TenSP) AS TenSP,
    SUM(SoLuongTon) AS TongSoLuongTon,
    SUM(SoLuongDatHang) AS TongSoLuongDatHang,
    SUM(SoLuongTon - SoLuongDatHang) AS TongSoLuongKhaDung
FROM
(
    SELECT
        tk.MaSP,
        sp.TenSP,
        tk.SoLuongTon,
        tk.SoLuongDatHang
    FROM [SITE_BAC].[store_management].dbo.TonKho tk
    JOIN [SITE_BAC].[store_management].dbo.SanPham_Core sp
        ON tk.MaSP = sp.MaSP
    WHERE tk.MaSP = @MaSP_Tong
    UNION ALL
    SELECT
```

```
        tk.MaSP,
        sp.TenSP,
        tk.SoLuongTon,
        tk.SoLuongDatHang
    FROM [SITE_NAM].[store_management].dbo.TonKho tk
    JOIN [SITE_NAM].[store_management].dbo.SanPham_Core sp
        ON tk.MaSP = sp.MaSP
    WHERE tk.MaSP = @MaSP_Tong
) AS T
GROUP BY MaSP;
```

Phân tích dữ liệu phân tán

	MaSP	TenSP	TongSoLuongTon	TongSoLuongDatHang	TongSoLuongKhaDung
1	SP001	iPhone 15 128GB	268	31	237

Hình 6 Phân tích dữ liệu phân tán: Kết quả truy vấn tổng tồn kho

Đây là truy vấn tổng hợp toàn hệ thống trên dữ liệu tồn kho phân tán.

Ở mỗi site, bảng TonKho chỉ chứa dữ liệu tồn kho của các kho thuộc site đó. Do đó, để biết tổng tồn kho toàn hệ thống, site chính phải lấy dữ liệu từ cả SITE_BAC và SITE_NAM.

Luồng xử lý của truy vấn:

- Bước 1: SITE_BAC trả về tồn kho của sản phẩm SP001 tại các kho miền Bắc
- Bước 2: SITE_NAM trả về tồn kho của sản phẩm SP001 tại các kho miền Nam
- Bước 3: SITE chính gộp dữ liệu bằng UNION ALL
- Bước 4: SITE chính tính SUM để ra tổng toàn hệ thống

Truy vấn này minh họa rõ cách xử lý dữ liệu phân tán: dữ liệu gốc nằm rải rác ở nhiều site, nhưng kết quả cuối cùng được tổng hợp tại site chính.

Một điểm tối ưu là truy vấn chỉ lấy các cột cần thiết: MaSP, TenSP, SoLuongTon, SoLuongDatHang

Không lấy các thông tin chi tiết kho vì mục tiêu chỉ là tính tổng. Điều này giúp giảm chi phí truyền dữ liệu qua mạng.

6.3. Truy vấn 3: Thống kê doanh thu theo tháng của từng kho và toàn hệ thống

Mục đích truy vấn

Truy vấn này dùng để thống kê doanh thu theo tháng, bao gồm hai mức:

1. Doanh thu theo từng kho.
2. Doanh thu tổng toàn hệ thống.

Doanh thu được tính dựa trên số lượng sản phẩm đã xuất kho và đơn giá bán trong chi tiết đơn hàng:

DoanhThu = SoLuongXuat * DonGia

Chỉ những đơn hàng thỏa mãn điều kiện sau mới được tính vào doanh thu:

dh.TrangThaiTT = 'paid'

AND dh.TrangThaiDH = 'completed'

AND pxk.TrangThaiXuat = 'exported'

Ý nghĩa:

Bảng 27 Ý nghĩa các điều kiện lọc tính doanh thu

Điều kiện	**Ý nghĩa**
TrangThaiTT = 'paid'	Đơn hàng đã thanh toán
TrangThaiDH = 'completed'	Đơn hàng đã hoàn thành
TrangThaiXuat = 'exported'	Kho đã thực hiện xuất hàng

Câu truy vấn (SQL / giả mã):

```
WITH DoanhThuTheoKho AS
(
    SELECT
        'SITE_BAC' AS SiteNguon,
        YEAR(dh.NgayDat) AS Nam,
        MONTH(dh.NgayDat) AS Thang,
        pxk.MaKhoXuat AS MaKho,
        k.TenKho,
        SUM(ctx.SoLuongXuat * ctdh.DonGia) AS DoanhThu
```

```
    FROM [SITE_BAC].[store_management].dbo.PhieuXuatKho pxk
    JOIN [SITE_BAC].[store_management].dbo.ChiTietXuatKho ctx
        ON pxk.MaPhieuXuat = ctx.MaPhieuXuat
    JOIN [SITE_BAC].[store_management].dbo.Kho k
        ON pxk.MaKhoXuat = k.MaKho
    JOIN [SITE_BAC].[store_management].dbo.DonHang dh
        ON pxk.MaDonHang = dh.MaDonHang
    JOIN [SITE_BAC].[store_management].dbo.ChiTietDonHang ctdh
        ON ctx.MaCTDH = ctdh.MaCTDH
    WHERE
        dh.TrangThaiTT = 'paid'
        AND dh.TrangThaiDH = 'completed'
        AND pxk.TrangThaiXuat = 'exported'
    GROUP BY
        YEAR(dh.NgayDat),
        MONTH(dh.NgayDat),
        pxk.MaKhoXuat,
        k.TenKho
    UNION ALL
    SELECT
        'SITE_NAM' AS SiteNguon,
        YEAR(dh.NgayDat) AS Nam,
        MONTH(dh.NgayDat) AS Thang,
        pxk.MaKhoXuat AS MaKho,
        k.TenKho,
        SUM(ctx.SoLuongXuat * ctdh.DonGia) AS DoanhThu
    FROM [SITE_NAM].[store_management].dbo.PhieuXuatKho pxk
    JOIN [SITE_NAM].[store_management].dbo.ChiTietXuatKho ctx
        ON pxk.MaPhieuXuat = ctx.MaPhieuXuat
    JOIN [SITE_NAM].[store_management].dbo.Kho k
        ON pxk.MaKhoXuat = k.MaKho
    JOIN [SITE_NAM].[store_management].dbo.DonHang dh
        ON pxk.MaDonHang = dh.MaDonHang
    JOIN [SITE_NAM].[store_management].dbo.ChiTietDonHang ctdh
        ON ctx.MaCTDH = ctdh.MaCTDH
    WHERE
        dh.TrangThaiTT = 'paid'
        AND dh.TrangThaiDH = 'completed'
        AND pxk.TrangThaiXuat = 'exported'
    GROUP BY
        YEAR(dh.NgayDat),
        MONTH(dh.NgayDat),
        pxk.MaKhoXuat,
        k.TenKho
)
SELECT
    SiteNguon,
```

```
    Nam,
    Thang,
    MaKho,
    TenKho,
    SUM(DoanhThu) AS DoanhThu
FROM DoanhThuTheoKho
GROUP BY
    SiteNguon,
    Nam,
    Thang,
    MaKho,
    TenKho
UNION ALL
SELECT
    'TOAN_HE_THONG' AS SiteNguon,
    Nam,
    Thang,
    NULL AS MaKho,
    N'Tất cả kho' AS TenKho,
    SUM(DoanhThu) AS DoanhThu
FROM DoanhThuTheoKho
GROUP BY
    Nam,
    Thang
ORDER BY
    Nam,
    Thang,
    SiteNguon,
    MaKho;
```

Phân tích dữ liệu phân tán

	SiteNguon	Nam	Thang	MaKho	TenKho	DoanhThu
1	SITE_BAC	2026	6	KB01	Kho Bắc 01	105950000.00
2	SITE_BAC	2026	6	KB02	Kho Bắc 02	65970000.00
3	SITE_BAC	2026	6	KB04	Kho Bắc 04	19960000.00
4	TOAN_HE_THONG	2026	6	NULL	Tất cả kho	191880000.00

Hình 7 Phân tích dữ liệu phân tán: Kết quả truy vấn thống kê doanh thu

Trong thiết kế phân tán, PhieuXuatKho được phân mảnh theo MaKhoXuat. Nghĩa là phiếu xuất của kho miền Bắc nằm tại SITE_BAC, còn phiếu xuất của kho miền Nam nằm tại SITE_NAM.

Bảng ChiTietXuatKho được phân mảnh dẫn xuất theo PhieuXuatKho, tức là chi tiết xuất kho được lưu cùng site với phiếu xuất kho. Cách này giúp việc join giữa PhieuXuatKho và ChiTietXuatKho được thực hiện cục bộ tại từng site, không cần join xuyên site.

Truy vấn dùng CTE DoanhThuTheoKho để gom doanh thu cục bộ tại từng site trước. Sau đó phần ngoài của truy vấn thực hiện hai nhóm kết quả:

- Nhóm 1: Doanh thu theo từng kho
- Nhóm 2: Doanh thu toàn hệ thống theo tháng

Dòng tổng toàn hệ thống được sinh ra bằng phần:

```
UNION ALL
SELECT
    'TOAN_HE_THONG' AS SiteNguon,
    Nam,
    Thang,
    NULL AS MaKho,
    N'Tất cả kho' AS TenKho,
    SUM(DoanhThu) AS DoanhThu
FROM DoanhThuTheoKho
GROUP BY
    Nam,
    Thang
```

Điểm cần chú ý là truy vấn không tính doanh thu dựa trực tiếp vào TongTien của đơn hàng, mà tính theo lượng hàng đã xuất kho. Điều này phù hợp với hệ thống đa kho vì một đơn hàng có thể được xuất từ nhiều kho khác nhau.

6.4. Truy vấn 4: Top sản phẩm bán chạy nhất toàn hệ thống

Mục đích truy vấn

Truy vấn này dùng để tìm 10 sản phẩm bán chạy nhất trên toàn hệ thống.

Tiêu chí xếp hạng:

1. Tổng số lượng bán giảm dần.

2. Nếu số lượng bán bằng nhau, xét tiếp tổng doanh thu giảm dần.

Kết quả trả về:

Bảng 28 Ý nghĩa các thuộc tính kết quả truy vấn top sản phẩm

Cột	Ý nghĩa
MaSP	Mã sản phẩm
TenSP	Tên sản phẩm
TongSoLuongBan	Tổng số lượng đã bán
TongDoanhThu	Tổng doanh thu của sản phẩm

Câu truy vấn (SQL / giả mã):

```
SELECT TOP 10
   MaSP,
   MAX(TenSP) AS TenSP,
   SUM(SoLuongBan) AS TongSoLuongBan,
   SUM(DoanhThu) AS TongDoanhThu
FROM
(
   SELECT
      ctdh.MaSP,
      sp.TenSP,
      ctdh.SoLuong AS SoLuongBan,
      ctdh.SoLuong * ctdh.DonGia AS DoanhThu
   FROM [SITE_BAC].[store_management].dbo.DonHang dh
   JOIN [SITE_BAC].[store_management].dbo.ChiTietDonHang ctdh
      ON dh.MaDonHang = ctdh.MaDonHang
   JOIN [SITE_BAC].[store_management].dbo.SanPham_Core sp
      ON ctdh.MaSP = sp.MaSP
   WHERE
      dh.TrangThaiTT = 'paid'
      AND dh.TrangThaiDH = 'completed'
   UNION ALL
   SELECT
      ctdh.MaSP,
      sp.TenSP,
      ctdh.SoLuong AS SoLuongBan,
      ctdh.SoLuong * ctdh.DonGia AS DoanhThu
   FROM [SITE_NAM].[store_management].dbo.DonHang dh
   JOIN [SITE_NAM].[store_management].dbo.ChiTietDonHang ctdh
```

```
        ON dh.MaDonHang = ctdh.MaDonHang
    JOIN [SITE_NAM].[store_management].dbo.SanPham_Core sp
        ON ctdh.MaSP = sp.MaSP
    WHERE
        dh.TrangThaiTT = 'paid'
        AND dh.TrangThaiDH = 'completed'
) AS T
GROUP BY MaSP
ORDER BY
    TongSoLuongBan DESC,
    TongDoanhThu DESC;
```

Phân tích dữ liệu phân tán

	MaSP	TenSP	TongSoLuongBan	TongDoanhThu
1	SP003	Xiaomi Redmi Note 13	4	19960000.00
2	SP002	Samsung Galaxy S24	3	65970000.00
3	SP001	iPhone 15 128GB	3	59970000.00

Hình 8 Phân tích dữ liệu phân tán: Kết quả truy vấn top 10 sản phẩm bán chạy

Truy vấn này liên quan đến dữ liệu đơn hàng, chi tiết đơn hàng và sản phẩm.

Trong hệ thống phân tán, bảng DonHang được phân mảnh theo khu vực xử lý đơn hàng. Điều này có nghĩa là đơn hàng của khách thuộc khu vực miền Bắc được xử lý tại SITE_BAC, còn đơn hàng của khách thuộc khu vực miền Nam được xử lý tại SITE_NAM.

Bảng ChiTietDonHang là bảng chi tiết phụ thuộc vào DonHang, nên được phân mảnh dẫn xuất theo MaDonHang. Nói cách khác, chi tiết đơn hàng được lưu cùng site với đơn hàng cha. Nhờ đó, join giữa DonHang và ChiTietDonHang được thực hiện cục bộ tại từng site.

Bảng SanPham_Core được nhân bản toàn phần sang các site. Vì vậy, mỗi site đều có sẵn thông tin tên sản phẩm để join với ChiTietDonHang.

Luồng xử lý:

- Bước 1: SITE_BAC tính số lượng bán và doanh thu từng sản phẩm ở miền Bắc
- Bước 2: SITE_NAM tính số lượng bán và doanh thu từng sản phẩm ở miền Nam
- Bước 3: SITE chính gộp kết quả từ hai site
- Bước 4: SITE chính GROUP BY theo MaSP

- Bước 5: Sắp xếp giảm dần và lấy TOP 10

Đây là truy vấn thống kê toàn hệ thống. Vì dữ liệu bán hàng nằm ở nhiều site nên bắt buộc phải gộp dữ liệu từ các site trước khi xác định sản phẩm bán chạy nhất.

6.5. Truy vấn 5: Đơn hàng có sản phẩm được xuất từ nhiều kho khác nhau.

Mục đích truy vấn

Truy vấn này dùng để tìm các đơn hàng có sản phẩm được xuất từ nhiều kho khác nhau. Một đơn hàng được xem là xuất từ nhiều kho khi cùng một MaDonHang xuất hiện với từ hai MaKhoXuat khác nhau trở lên.

Ví dụ:

MaDonHang	MaKhoXuat
DH001	KHO_BAC_01
DH001	KHO_NAM_02

Trong trường hợp này, đơn hàng DH001 là đơn hàng xuất từ nhiều kho.

Câu truy vấn (SQL / giả mã):

```
WITH TatCaPhieuXuat AS
(
    SELECT
        'SITE_BAC' AS SiteNguon,
        pxk.MaDonHang,
        pxk.MaKhoXuat
    FROM [SITE_BAC].[store_management].dbo.PhieuXuatKho pxk
    WHERE pxk.TrangThaiXuat = 'exported'

    UNION ALL

    SELECT
        'SITE_NAM' AS SiteNguon,
        pxk.MaDonHang,
        pxk.MaKhoXuat
    FROM [SITE_NAM].[store_management].dbo.PhieuXuatKho pxk
    WHERE pxk.TrangThaiXuat = 'exported'
),
KhoXuatKhongTrung AS
(
    SELECT DISTINCT
        MaDonHang,
```

```
        MaKhoXuat
    FROM TatCaPhieuXuat
)
SELECT
    MaDonHang,
    COUNT(*) AS SoKhoXuat,
    STRING_AGG(MaKhoXuat, ', ') AS DanhSachKhoXuat
FROM KhoXuatKhongTrung
GROUP BY MaDonHang
HAVING COUNT(*) > 1
ORDER BY
    SoKhoXuat DESC,
    MaDonHang;
```

Phân tích dữ liệu phân tán

	MaDonHang	SoKhoXuat	DanhSachKhoXuat
1	C6B1C901-8EDF-4E2E-8837-9EB2EA7103C7	2	KB04, KN03
2	30978E78-1EFE-4EBC-A47E-A6CB41088857	2	KB01, KN01
3	53E36EB7-F07F-4B18-96B9-AA73D34EA5B6	2	KB01, KB02

Hình 9 Phân tích dữ liệu phân tán: Kết quả truy vấn đơn hàng xuất từ nhiều kho

Truy vấn này thể hiện rõ đặc điểm của bài toán phân tán trong hệ thống đa kho.

Bảng PhieuXuatKho được phân mảnh theo MaKhoXuat. Vì vậy, nếu một đơn hàng được xuất từ nhiều kho ở nhiều khu vực khác nhau, các phiếu xuất của đơn hàng đó có thể nằm ở nhiều site.

Ví dụ:

Đơn hàng DH001:

- Một phần hàng xuất từ kho Bắc tại SITE_BAC
- Một phần hàng xuất từ kho Nam tại SITE_NAM

Trong trường hợp này, nếu chỉ truy vấn một site thì không thể phát hiện đầy đủ đơn hàng được xuất từ nhiều kho. Site chính phải lấy dữ liệu từ cả hai site, sau đó gom theo MaDonHang.

CTE TatCaPhieuXuat có nhiệm vụ gom toàn bộ phiếu xuất đã xuất thành công từ hai site:

```
WHERE pxk.TrangThaiXuat = 'exported'
```

CTE KhoXuatKhongTrung dùng SELECT DISTINCT để loại bỏ trường hợp một đơn hàng có nhiều phiếu xuất từ cùng một kho.

Cuối cùng:

HAVING COUNT(*) > 1

được dùng để chỉ giữ lại các đơn hàng có nhiều hơn một kho xuất.

Truy vấn này có ý nghĩa nghiệp vụ quan trọng vì nó giúp phát hiện các đơn hàng cần phối hợp xử lý giữa nhiều kho. Trong hệ thống thực tế, loại đơn hàng này thường phức tạp hơn đơn hàng xuất từ một kho duy nhất vì cần đồng bộ trạng thái xuất hàng, vận chuyển và hoàn tất đơn hàng giữa nhiều site.

CHƯƠNG 7. TRUY XUẤT ĐỒNG THỜI VÀ NHẤT QUÁN TỒN KHO

7.1. Bài toán đặt ra

Trong hệ thống bán hàng trực tuyến đa kho, nhiều khách hàng có thể đồng thời đặt mua cùng một sản phẩm tại cùng một kho. Đây là vấn đề điển hình của truy xuất đồng thời trong cơ sở dữ liệu. Nếu hệ thống chỉ kiểm tra tồn kho bằng cách đọc dữ liệu trước rồi mới cập nhật sau, nhiều giao dịch có thể cùng nhìn thấy một lượng tồn kho khả dụng giống nhau và cùng đặt hàng thành công. Điều này có thể dẫn đến các lỗi nghiêm trọng như bán vượt tồn kho, mất cập nhật hoặc làm cho số lượng tồn kho bị âm.

Ví dụ, tại kho KB03, sản phẩm SP004 có SoLuongTon = 5 và SoLuongDatHang = 0. Khi đó số lượng khả dụng là:

KhaDung = SoLuongTon - SoLuongDatHang = 5

Nếu có hai giao dịch cùng lúc đặt mua 4 sản phẩm, cả hai giao dịch đều có thể đọc thấy khả dụng là 5 nếu không có cơ chế khóa. Khi đó cả hai cùng giữ hàng thành công, làm cho tổng số lượng đã đặt là 8 trong khi tồn thực tế chỉ có 5. Đây chính là hiện tượng bán vượt tồn kho.

Vì vậy, mục tiêu bắt buộc của hệ thống là: không để số lượng khả dụng bị âm, không cho phép giữ hàng vượt quá số lượng tồn thực tế và nếu giao dịch thất bại thì mọi thay đổi tạm thời phải được hoàn tác.

7.2. Khái niệm nhất quán trong demo

Trong phạm vi demo, nhất quán tồn kho được hiểu theo các nguyên tắc sau:

- Tại mọi thời điểm, hệ thống phải đảm bảo SoLuongTon >= SoLuongDatHang.
- Số lượng khả dụng được tính bằng SoLuongTon - SoLuongDatHang không được nhỏ hơn 0.
- Một giao dịch chỉ được giữ hàng nếu số lượng khả dụng hiện tại lớn hơn hoặc bằng số lượng cần giữ.
- Nếu giao dịch đặt hàng thất bại ở bất kỳ bước nào, các thay đổi tạm thời như tăng SoLuongDatHang phải được rollback.
- Khi xuất hàng thành công cho khách, hệ thống mới chính thức trừ SoLuongTon.

Như vậy, hệ thống không trừ tồn kho thực tế ngay khi khách hàng vừa đặt hàng. Thay vào đó, hệ thống giữ hàng trước bằng cách tăng SoLuongDatHang. Khi đơn hàng được xử lý và kho xác nhận xuất hàng, lúc đó tồn kho thực tế mới được trừ.

7.3. Mô hình SoLuongTon và SoLuongDatHang

Bảng TonKho sử dụng hai trường chính để quản lý trạng thái tồn kho:

Trường	Ý nghĩa
SoLuongTon	Số lượng tồn thực tế trong kho
SoLuongDatHang	Số lượng đã được giữ cho các đơn hàng đang xử lý nhưng chưa xuất kho
KhaDung	Số lượng còn có thể bán, được tính bằng SoLuongTon - SoLuongDatHang

Cách thiết kế này giúp tách rõ giữa hàng còn trong kho và hàng đã được giữ cho đơn hàng. Khi khách hàng đặt hàng, hệ thống chỉ tăng SoLuongDatHang, chưa trừ ngay SoLuongTon. Nhờ đó, hàng đã được giữ sẽ không bị giao dịch khác lấy tiếp, nhưng hệ thống vẫn biết hàng chưa thực sự được xuất khỏi kho.

Ví dụ, ban đầu sản phẩm SP004 tại kho KB03 có:

MaKho	MaSP	SoLuongTon	SoLuongDatHang	KhaDung
KB03	SP004	5	0	5

Khi giao dịch thứ nhất giữ 4 sản phẩm thành công, dữ liệu trở thành:

MaKho	MaSP	SoLuongTon	SoLuongDatHang	KhaDung
KB03	SP004	5	4	1

Lúc này, kho vẫn còn 5 sản phẩm thực tế, nhưng chỉ còn 1 sản phẩm khả dụng để bán cho giao dịch mới.

7.4. Cơ chế khóa dòng khi giữ hàng

Do hệ thống sử dụng SQL Server, cơ chế khóa được triển khai bằng UPDLOCK và HOLDLOCK thay cho SELECT ... FOR UPDATE. Khi một giao dịch kiểm tra và giữ hàng, dòng tồn kho tương ứng trong bảng TonKho sẽ bị khóa cho đến khi giao dịch kết thúc.

Câu lệnh cập nhật tồn kho được thực hiện theo nguyên tắc kiểm tra điều kiện và cập nhật trong cùng một thao tác:

```
UPDATE TonKho WITH (UPDLOCK, HOLDLOCK)
```

```
SET
    SoLuongDatHang = SoLuongDatHang + @SoLuongCanGiu,
    NgayCapNhat = SYSDATETIME()
WHERE MaKho = @MaKho
  AND MaSP = @MaSP
  AND (SoLuongTon - SoLuongDatHang) >= @SoLuongCanGiu;
```

Sau khi thực hiện câu lệnh này, hệ thống kiểm tra @@ROWCOUNT:

- Nếu @@ROWCOUNT = 1, nghĩa là giữ hàng thành công.
- Nếu @@ROWCOUNT = 0, nghĩa là không đủ hàng hoặc dòng tồn kho đã bị giao dịch khác cập nhật trước đó.

Cơ chế này giúp tránh tình trạng hai giao dịch cùng đọc một lượng tồn kho cũ rồi cùng cập nhật thành công. Giao dịch đến sau phải chờ giao dịch trước kết thúc, sau đó đọc lại dữ liệu mới nhất rồi mới quyết định có được giữ hàng hay không.

7.5. Chu trình giữ hàng, xác nhận xuất và hoàn tác

7.5.1. Giữ hàng

Khi khách hàng đặt hàng, procedure sp_DatHang_TuSiteBac_ModelB sẽ gom nhu cầu theo từng sản phẩm, sau đó gọi procedure sp_GiuHang_1SanPham_NoiBo để giữ hàng tại site tương ứng.

Quá trình giữ hàng gồm các bước:

- Kiểm tra sản phẩm có hợp lệ và còn kinh doanh hay không.
- Gom số lượng cần đặt theo từng mã sản phẩm.
- Kiểm tra tổng khả dụng tại site đang xử lý.
- Khóa các dòng tồn kho bằng UPDLOCK, HOLDLOCK.
- Tăng SoLuongDatHang theo số lượng cần giữ.
- Ghi kết quả phân bổ vào bảng tạm #PhanBo.

Nếu site Bắc không đủ hàng, hệ thống tiếp tục gọi sang site Nam thông qua linked server để giữ phần còn thiếu. Nếu tổng số lượng giữ được ở các site vẫn không đủ nhu cầu, hệ thống phát sinh lỗi và rollback giao dịch.

7.5.2. Xác nhận xuất hàng

Sau khi đơn hàng đã được tạo và hàng đã được gom đủ về kho xử lý, hệ thống tạo phiếu xuất giao khách. Khi nhân viên kho xác nhận xuất hàng, procedure sp_XacNhanXuat_GiaoKhach được sử dụng để trừ tồn kho thực tế.

Procedure này thực hiện các bước chính:

- Khóa phiếu xuất kho bằng UPDLOCK, HOLDLOCK.
- Kiểm tra phiếu xuất có tồn tại và đang ở trạng thái waiting_export.
- Kiểm tra đây là phiếu xuất giao khách, tức MaKhoNhan IS NULL.
- Gom số lượng xuất theo từng sản phẩm.
- Kiểm tra SoLuongTon và SoLuongDatHang có đủ để xuất hay không.
- Trừ đồng thời SoLuongTon và SoLuongDatHang.
- Cập nhật trạng thái phiếu xuất thành exported.
- Cập nhật trạng thái đơn hàng sang shipping.

Khi đó, hàng không chỉ được giữ nữa mà đã chính thức được xuất khỏi kho để giao cho khách.

7.5.3. Hoàn tác khi giao dịch thất bại

Trong hệ thống hiện tại, nhóm không xây dựng procedure release riêng để hoàn trả hàng đã giữ. Thay vào đó, toàn bộ quá trình đặt hàng được đặt trong một transaction.

Nếu lỗi xảy ra ở bất kỳ bước nào, ví dụ:

- Site Bắc giữ được hàng nhưng site Nam không đủ hàng.
- Tạo đơn hàng thất bại.
- Tạo chi tiết đơn hàng thất bại.
- Tạo phiếu xuất hoặc phiếu nhập nội bộ thất bại.

Thì khối CATCH sẽ thực hiện:

```
IF @@TRANCOUNT > 0
    ROLLBACK TRANSACTION;
```

Nhờ đó, các thay đổi tạm thời như tăng SoLuongDatHang sẽ được hoàn tác. Kết quả là nếu đơn hàng không được tạo thành công thì hàng đã giữ cũng không bị treo trong tồn kho.

Đối với thao tác liên site qua linked server, cơ chế rollback toàn bộ chỉ đảm bảo đầy đủ khi môi trường SQL Server được cấu hình giao dịch phân tán phù hợp. Trong phạm vi demo, hệ thống mô phỏng theo hướng transaction rollback để đảm bảo dữ liệu không bị để lại ở trạng thái dở dang.

7.6. Kịch bản đồng thời trong bản demo

Kịch bản demo được thực hiện với sản phẩm SP004 tại kho KB03.

Dữ liệu ban đầu:

MaKho	MaSP	SoLuongTon	SoLuongDatHang	KhaDung
KB03	SP004	5	0	5

Hai giao dịch được mở gần như đồng thời ở hai cửa sổ truy vấn khác nhau. Mỗi giao dịch đều muốn giữ 4 sản phẩm SP004 tại kho KB03.

- Trường hợp không có cơ chế khóa:

Nếu hệ thống chỉ đọc tồn kho rồi cập nhật sau, cả hai giao dịch có thể cùng đọc thấy KhaDung = 5. Khi đó cả hai đều nghĩ rằng kho đủ hàng để bán 4 sản phẩm. Nếu cả hai cùng cập nhật thành công, SoLuongDatHang có thể tăng lên 8, vượt quá SoLuongTon = 5. Đây là lỗi bán vượt tồn kho.

- Trường hợp có UPDLOCK, HOLDLOCK

Giao dịch thứ nhất bắt đầu trước và thực hiện cập nhật:

```
UPDATE dbo.TonKho WITH (UPDLOCK, HOLDLOCK)
SET
    SoLuongDatHang = SoLuongDatHang + 4,
    NgayCapNhat = SYSDATETIME()
WHERE MaKho = 'KB03'
  AND MaSP = 'SP004'
  AND (SoLuongTon - SoLuongDatHang) >= 4;
```

Vì ban đầu khả dụng là 5 nên giao dịch thứ nhất giữ hàng thành công. Lúc này:

MaKho	MaSP	SoLuongTon	SoLuongDatHang	KhaDung
KB03	SP004	5	4	1

Trong khi giao dịch thứ nhất chưa commit, giao dịch thứ hai phải chờ vì dòng tồn kho đang bị khóa. Sau khi giao dịch thứ nhất commit, giao dịch thứ hai tiếp tục chạy và kiểm tra lại điều kiện:

SoLuongTon - SoLuongDatHang >= 4

Lúc này: 5 - 4 = 1

Do 1 < 4, giao dịch thứ hai không cập nhật được dòng nào.

Kết quả @@ROWCOUNT = 0, nghĩa là giữ hàng thất bại do không đủ hàng.

Kết quả mong đợi:

Giao dịch	Số lượng muốn giữ	Kết quả	Giá trị @@ROWCOUNT
Giao dịch 1	4	Giữ hàng thành công	1
Giao dịch 2	4	Giữ hàng thất bại	0

Kết quả cuối cùng của tồn kho:

MaKho	MaSP	SoLuongTon	SoLuongDatHang	KhaDung
KB03	SP004	5	4	1

Như vậy, hệ thống chỉ cho phép một giao dịch giữ hàng thành công, giao dịch còn lại bị từ chối do không đủ khả dụng. Điều này chứng minh hệ thống không xảy ra bán vượt tồn kho.

7.7. Ghi nhận thay đổi tồn kho trong hệ thống

Trong bản demo hiện tại, hệ thống chưa xây dựng bảng audit/log tồn kho riêng. Do đó, hệ thống không lưu lịch sử chi tiết từng lần reserve, commit hoặc release giống như các hệ thống thương mại điện tử thực tế.

Việc theo dõi trạng thái tồn kho hiện tại được thực hiện thông qua bảng TonKho, gồm các trường:

Trường	Vai trò
SoLuongTon	Ghi nhận số lượng tồn thực tế

SoLuongDatHang	Ghi nhận số lượng đang được giữ cho đơn hàng
NgayCapNhat	Ghi nhận thời điểm cập nhật gần nhất
RowVer	Hỗ trợ kiểm soát phiên bản dữ liệu

Cách làm này đủ để chứng minh cơ chế chống bán vượt tồn kho trong phạm vi demo. Tuy nhiên, nếu phát triển thành hệ thống thực tế, nên bổ sung bảng log tồn kho, ví dụ TonKho_Audit, để ghi lại các hành động như giữ hàng, xác nhận xuất, hoàn tác giữ hàng. Bảng log này sẽ giúp truy vết nguyên nhân sai lệch tồn kho, kiểm toán nghiệp vụ và hỗ trợ debug trong môi trường phân tán.

7.8. Mức độ hoàn chỉnh của giải pháp

Giải pháp hiện tại chưa triển khai đầy đủ một distributed transaction manager hoàn chỉnh như Two-Phase Commit ở mức hệ thống. Tuy nhiên, trong phạm vi đồ án, cơ chế hiện tại đáp ứng được mục tiêu mô phỏng truy xuất đồng thời và đảm bảo nhất quán tồn kho vì các lý do sau:

- Hệ thống sử dụng transaction trong SQL Server để đảm bảo một nghiệp vụ đặt hàng hoặc thành công toàn bộ hoặc rollback toàn bộ.
- Hệ thống dùng UPDLOCK, HOLDLOCK để khóa dòng tồn kho khi kiểm tra và giữ hàng.
- Điều kiện đủ hàng được kiểm tra trực tiếp trong câu lệnh UPDATE, tránh tách rời bước đọc và bước ghi.
- Hệ thống dùng SoLuongDatHang để giữ hàng trước, chưa trừ ngay SoLuongTon.
- Khi xuất hàng cho khách, procedure sp_XacNhanXuat_GiaoKhach mới trừ chính thức cả SoLuongTon và SoLuongDatHang.
- Nếu lỗi xảy ra trong quá trình đặt hàng, ROLLBACK TRANSACTION giúp hoàn tác các thay đổi tạm thời, tránh hàng bị giữ treo.
- Kịch bản demo với KB03, SP004, tồn ban đầu 5 và hai giao dịch cùng đặt 4 chứng minh rằng chỉ một giao dịch được giữ hàng thành công, giao dịch còn lại bị từ chối.

Tóm lại, hệ thống đảm bảo được các bất biến quan trọng: không bán vượt tồn kho, không để tồn kho khả dụng âm và không để giao dịch thất bại làm sai lệch dữ liệu tồn kho. Đây là cơ sở để chứng minh tính đúng đắn của cơ chế kiểm soát đồng thời trong hệ thống bán hàng trực tuyến đa kho.

CHƯƠNG 8. NỘI DUNG NÂNG CAO

8.1 Mô phỏng cập nhật tồn kho đồng thời

- Luồng thực thi và Kết quả mô phỏng

Giả sử chạy kịch bản theo thứ tự thực tế như sau: Chạy khối Chuẩn bị dữ liệu -> Chạy Tab 1 -> Sang Tab 2 chạy ngay lập tức.

1. Khởi tạo dữ liệu (Chuẩn bị)

Hệ thống ghi nhận: SoLuongTon = 5, SoLuongDatHang = 0.

Số lượng khả dụng (KhaDung) = 5.

2. Tab 1 bắt đầu Transaction (Giao dịch 1)

Tab 1 chạy lệnh UPDATE và lấy được khóa UPDLOCK và HOLDLOCK trên dòng dữ liệu của sản phẩm SP004.

Điều kiện (5 - 0) >= 4 thỏa mãn. Dữ liệu được cập nhật trên bộ nhớ tạm: SoLuongDatHang tăng lên 4.

Ngay sau đó, Tab 1 gặp lệnh WAITFOR DELAY '00:00:20', nó sẽ "ngủ" trong 20 giây. Trong suốt 20 giây này, khóa dữ liệu vẫn bị Tab 1 giữ chặt.

3. Tab 2 bắt đầu Transaction (Giao dịch 2 - Cùng lúc đó)

Tab 2 cố gắng chạy lệnh UPDATE trên cùng dòng SP004.

Tuy nhiên, do Tab 1 đang giữ khóa (Locking) trên dòng này, lệnh của Tab 2 sẽ bị chặn lại (Blocked). Bạn sẽ thấy vòng quay "Executing query" trên Tab 2 xoay liên tục. Nó phải đứng xếp hàng chờ Tab 1 giải phóng khóa.

4. Tab 1 kết thúc (Sau 20 giây)

Tab 1 thức dậy, in ra SoDongCapNhat_Tab1 = 1 và chạy lệnh COMMIT TRANSACTION.

Dữ liệu chính thức được ghi vào đĩa: SoLuongDatHang = 4. KhaDung lúc này chỉ còn 1.

Tab 1 giải phóng toàn bộ khóa.

5. Tab 2 tiếp tục thực thi (Ngay sau khi Tab 1 Commit)

Khóa được mở, Tab 2 ngay lập tức lao vào thực thi lệnh UPDATE của nó.

Lúc này, cơ sở dữ liệu sẽ đánh giá lại điều kiện WHERE dựa trên dữ liệu mới nhất mà Tab 1 vừa lưu.

Hệ thống kiểm tra: SoLuongTon (5) - SoLuongDatHang (4) = 1.

Điều kiện (5 - 4) >= 4 trở thành SAI.

Kết quả: Không có dòng nào được cập nhật. Lệnh SELECT @@ROWCOUNT của Tab 2 sẽ in ra SoDongCapNhat_Tab2 = 0.

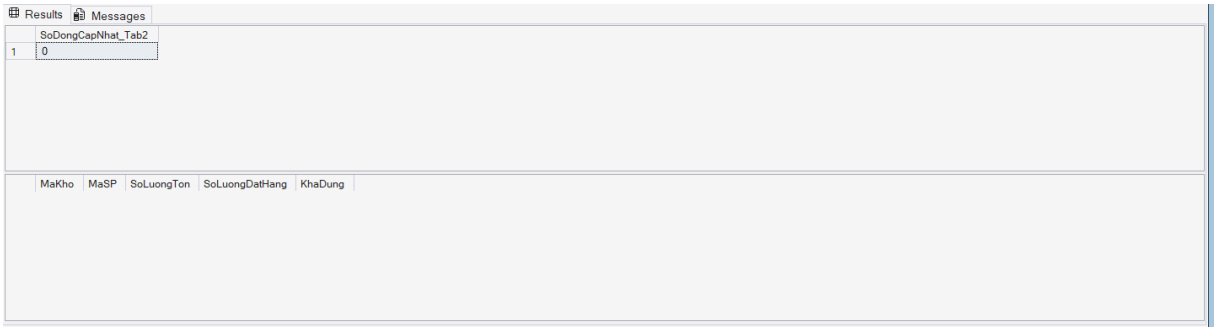
Tab 2 COMMIT.

Tổng kết trạng thái cuối cùng

Bảng 29 Tổng kết trạng thái dữ liệu khi cập nhật đồng thời

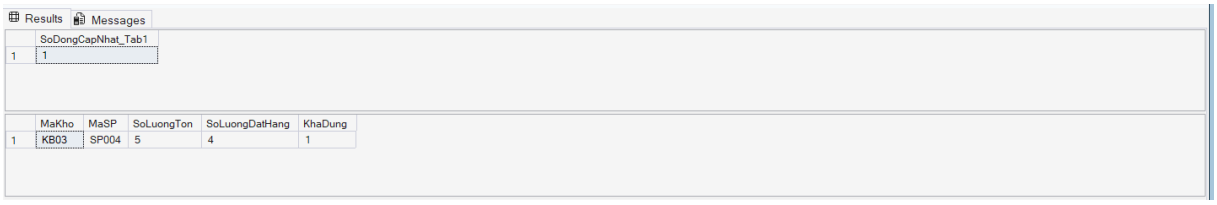
Thông số	Giá trị khởi tạo	Kết quả Tab 1	Kết quả Tab 2
SoLuongTon	5	5	5
SoLuongDatHang	0	4	4
KhaDung	5	1	1
Số dòng cập nhật	-	1	0

- Người đặt hàng chậm không tranh chấp được dữ liệu



Hình 10 Người đặt hàng chậm không tranh chấp được dữ liệu

- Người tranh chấp được dữ liệu



Hình 11 Người tranh chấp được dữ liệu

8.2. Chiến lược replication cho thông tin sản phẩm

8.2.1 Các bảng áp dụng replication

Bảng 30 Chiến lược replication cho các bảng dữ liệu

STT	Bảng dữ liệu	Chiến lược đề xuất	Lý do
1	DanhMuc	Nhân bản toàn phần	Dữ liệu dùng để phân loại và lọc sản phẩm, đọc nhiều, ít thay đổi

2	ThuongHieu	Nhân bản toàn phần	Phục vụ tìm kiếm, lọc sản phẩm theo thương hiệu
3	DanhMuc_ThuongHieu	Nhân bản toàn phần	Phục vụ chức năng lọc thương hiệu theo danh mục
4	SanPham_Core	Nhân bản toàn phần	Chứa thông tin cốt lõi của sản phẩm như mã sản phẩm, tên, giá, danh mục, thương hiệu, trạng thái
5	SanPham_Detail	Có thể nhân bản toàn phần hoặc lưu tại trung tâm	Dữ liệu mô tả chi tiết và thông số kỹ thuật có thể lớn hơn, tùy quy mô hệ thống

8.2.2 Mô hình replication đề xuất

Bảng 31 Mô hình replication đề xuất

Thành phần	**Mô tả**
Mô hình	Mô hình trung tâm - các site kho
Site trung tâm	Đóng vai trò Publisher, lưu dữ liệu gốc của sản phẩm/danh mục
Site kho	Đóng vai trò Subscriber, nhận bản sao dữ liệu từ trung tâm
Cơ chế ghi dữ liệu	Single-writer, multi-reader
Nơi được phép cập nhật sản phẩm	Site trung tâm
Các site kho	Chỉ đọc dữ liệu sản phẩm/danh mục đã được nhân bản
Dữ liệu nhân bản	DanhMuc, ThuongHieu, DanhMuc_ThuongHieu, SanPham_Core, có thể gồm SanPham_Detail

8.3. Phân tích ảnh hưởng của truy vấn join phân tán

Trong cơ sở dữ liệu phân tán, truy vấn JOIN giữa các bảng đặt ở các site khác nhau có chi phí cao hơn so với join trong một cơ sở dữ liệu tập trung. Nguyên nhân là ngoài chi phí đọc dữ liệu và thực hiện phép kết, hệ thống còn phát sinh thêm chi phí truyền dữ liệu qua mạng giữa các site.

Ví dụ, khi site Bắc cần kiểm tra đơn hàng nhưng một phần phiếu xuất kho lại nằm ở site Nam, hệ thống có thể phải join các bảng như DonHang, ChiTietDonHang, PhieuXuatKho, ChiTietXuatKho giữa hai site. Nếu xử lý không tối ưu, hệ thống có thể phải chuyển một lượng lớn dữ liệu từ site này sang site khác, làm tăng thời gian truy vấn và tải mạng.

1. Chi phí của phép join phân tán

Bảng 32 Phân rã chi phí của phép join phân tán

Thành phần chi phí	Ý nghĩa
Chi phí đọc dữ liệu	Mỗi site phải đọc các bảng liên quan đến truy vấn
Chi phí truyền dữ liệu	Dữ liệu từ site này phải gửi sang site khác để thực hiện join
Chi phí xử lý join	Site nhận dữ liệu phải thực hiện phép kết
Chi phí dữ liệu trung gian	Dữ liệu tạm sinh ra trong quá trình lọc, gửi khóa, gửi kết quả khớp

Trong đó, chi phí truyền dữ liệu qua mạng thường là phần đáng quan tâm nhất. Nếu bảng lớn bị chuyển toàn bộ qua mạng, truy vấn sẽ chậm và gây tải lớn cho hệ thống.

2. Chiến lược 1: Chuyển toàn bộ bảng sang một site để join

Cách đơn giản nhất là chuyển toàn bộ một bảng từ site này sang site khác rồi thực hiện join tại một site.

Ví dụ, nếu PhieuXuatKho nằm ở site Nam và DonHang nằm ở site Bắc, hệ thống có thể chuyển toàn bộ bảng PhieuXuatKho từ Nam sang Bắc để join với DonHang.

Bảng 33 Ưu, nhược điểm của chiến lược chuyển toàn bộ bảng

Ưu điểm	Nhược điểm
Dễ hiểu, dễ triển khai	Tốn nhiều băng thông mạng
Phù hợp khi bảng cần chuyển rất nhỏ	Không phù hợp với bảng lớn

Ưu điểm	Nhược điểm
Không cần xử lý phức tạp	Sinh nhiều dữ liệu trung gian không cần thiết

Chiến lược này chỉ nên dùng khi bảng cần chuyển có kích thước nhỏ, ví dụ như bảng danh mục hoặc thương hiệu. Tuy nhiên, với các bảng nghiệp vụ lớn như đơn hàng, chi tiết đơn hàng, phiếu xuất kho hoặc tồn kho, việc chuyển toàn bộ bảng là không tối ưu.

3. Chiến lược 2: Chuyển bảng nhỏ đến site có bảng lớn

Một cách tốt hơn là xác định bảng nào nhỏ hơn thì chuyển bảng đó sang site chứa bảng lớn hơn.

Ví dụ, nếu cần join ChiTietDonHang với SanPham_Core, mà SanPham_Core đã được nhân bản ở các site, hệ thống không cần chuyển bảng sản phẩm qua mạng. Site xử lý đơn hàng có thể join trực tiếp với bản sao SanPham_Core tại chỗ.

Bảng 34 Ưu, nhược điểm của chiến lược chuyển bảng nhỏ

Ưu điểm	Nhược điểm
Giảm lượng dữ liệu truyền qua mạng	Cần biết bảng nào nhỏ hơn
Phù hợp với bảng danh mục, sản phẩm đã nhân bản	Nếu chọn sai bảng để chuyển thì vẫn tốn chi phí
Đơn giản hơn semijoin	Chưa tối ưu bằng semijoin trong trường hợp bảng lớn

Chiến lược này phù hợp với các bảng nhỏ hoặc bảng dùng chung. Vì vậy, việc nhân bản SanPham_Core, DanhMuc, ThuongHieu đến các site giúp giảm đáng kể số phép join phân tán.

4. Chiến lược 3: Semijoin

Semijoin là chiến lược tối ưu hơn trong nhiều trường hợp. Thay vì chuyển toàn bộ bảng, hệ thống chỉ chuyển các khóa cần thiết sang site còn lại để lọc dữ liệu trước, sau đó chỉ gửi lại các dòng thật sự cần join.

Ví dụ, site Bắc cần lấy thông tin phiếu xuất liên quan đến một số đơn hàng cụ thể ở site Nam. Thay vì chuyển toàn bộ bảng PhieuXuatKho từ Nam sang Bắc, site Bắc chỉ gửi danh sách MaDonHang cần tìm sang site Nam. Site Nam dùng danh sách này để lọc PhieuXuatKho, sau đó chỉ gửi về các phiếu xuất có liên quan.

Quy trình semijoin có thể mô tả như sau:

Bảng 35 Quy trình thực thi Semijoin

Bước	Mô tả
Bước 1	Site Bắc lấy danh sách khóa cần join, ví dụ MaDonHang
Bước 2	Gửi danh sách khóa này sang site Nam
Bước 3	Site Nam lọc các bản ghi khớp trong PhieuXuatKho
Bước 4	Site Nam chỉ gửi các bản ghi khớp về site Bắc
Bước 5	Site Bắc thực hiện join với dữ liệu nhận được

Bảng 36 Ưu, nhược điểm của chiến lược Semijoin

Ưu điểm	Nhược điểm
Giảm lượng dữ liệu truyền qua mạng	Cách triển khai phức tạp hơn
Chỉ gửi dữ liệu thật sự cần thiết	Có thêm bước gửi khóa trung gian
Phù hợp khi bảng ở site xa rất lớn nhưng số dòng cần lấy ít	Không hiệu quả nếu phần lớn dữ liệu đều khớp
Giảm dữ liệu trung gian so với chuyển toàn bộ bảng	Cần có chỉ mục trên khóa join

Semijoin phù hợp với hệ thống đa kho vì nhiều truy vấn chỉ cần lấy dữ liệu liên quan đến một đơn hàng, một sản phẩm hoặc một kho cụ thể, chứ không cần lấy toàn bộ bảng ở site khác.

5. So sánh lượng dữ liệu trung gian phát sinh

Bảng 37 So sánh lượng dữ liệu trung gian phát sinh

Chiến lược	Dữ liệu truyền qua mạng	Dữ liệu trung gian	Đánh giá
Chuyển toàn bộ bảng	Rất lớn nếu bảng nhiều dòng	Nhiều	Không tối ưu với bảng nghiệp vụ lớn
Chuyển bảng nhỏ sang site bảng lớn	Trung bình hoặc nhỏ	Ít hơn chuyển toàn bộ bảng lớn	Phù hợp khi có bảng nhỏ

Chiến lược	Dữ liệu truyền qua mạng	Dữ liệu trung gian	Đánh giá
Semijoin	Chỉ gửi khóa và các dòng khớp	Ít	Tối ưu hơn khi số dòng cần join ít
Nhân bản bảng dùng chung	Gần như không cần truyền khi join	Rất ít	Phù hợp với bảng sản phẩm, danh mục, thương hiệu

Qua so sánh, có thể thấy chiến lược chuyển toàn bộ bảng tạo ra nhiều dữ liệu trung gian nhất. Semijoin giúp giảm lượng dữ liệu truyền qua mạng vì chỉ gửi các khóa cần thiết và các bản ghi thật sự khớp. Đối với các bảng dùng chung như sản phẩm, danh mục, thương hiệu, nhân bản toàn phần là cách hiệu quả nhất để tránh join phân tán.

6. Đề xuất áp dụng trong hệ thống đa kho

Trong hệ thống bán hàng trực tuyến đa kho, nên áp dụng các nguyên tắc sau:

Bảng 38 Đề xuất áp dụng chiến lược join trong hệ thống đa kho

Loại dữ liệu	Chiến lược đề xuất	Lý do
SanPham_Core, DanhMuc, ThuongHieu	Nhân bản toàn phần đến các site	Tránh join phân tán khi tra cứu sản phẩm
DonHang, ChiTietDonHang	Phân mảnh theo khu vực xử lý đơn	Đơn hàng thuộc site nào thì xử lý tại site đó
TonKho	Phân mảnh theo kho	Kho nào quản lý thì site đó cập nhật tồn kho
PhieuXuatKho, ChiTietXuatKho	Phân mảnh theo kho xuất	Phiếu xuất phát sinh tại kho nào thì lưu tại site đó
Truy vấn cần lấy dữ liệu từ site khác	Ưu tiên semijoin	Giảm lượng dữ liệu truyền qua mạng

Khi viết truy vấn phân tán, cần lọc dữ liệu càng sớm càng tốt bằng các điều kiện như MaDonHang, MaSP, MaKho, MaKhuVuc, ngày đặt hàng hoặc trạng thái đơn hàng. Không nên SELECT * hoặc join toàn bộ bảng giữa các site. Ngoài ra, các cột thường dùng để join như MaSP, MaDonHang, MaKho, MaPhieuXuat cần có chỉ mục để tăng tốc độ lọc và join.

8.4 So sánh hiệu năng giữa lưu trữ tập trung và phân tán

1. Bảng so sánh tổng quát

Bảng 39 So sánh hiệu năng giữa lưu trữ tập trung và phân tán

Tiêu chí	Phương án tập trung	Phương án phân tán
Cách lưu trữ dữ liệu	Toàn bộ dữ liệu được lưu tại một site trung tâm.	Dữ liệu được chia ra nhiều site theo khu vực, kho hoặc nghiệp vụ.
Truy vấn JOIN	Join đơn giản hơn vì các bảng nằm cùng một nơi.	Join phức tạp hơn nếu các bảng nằm ở nhiều site khác nhau.
Chi phí truyền dữ liệu	Ít phát sinh khi join nội bộ trong trung tâm.	Có thể phát sinh chi phí truyền dữ liệu qua mạng giữa các site.
Hiệu năng truy vấn cục bộ	Site kho phải truy vấn về trung tâm nên có thể chậm.	Site kho xử lý dữ liệu cục bộ nhanh hơn nếu dữ liệu nằm đúng site.
Dữ liệu trung gian	Ít dữ liệu trung gian truyền qua mạng.	Có thể sinh dữ liệu trung gian lớn nếu chuyển toàn bộ bảng để join.
Khả năng mở rộng	Khó mở rộng khi số lượng người dùng và đơn hàng tăng.	Dễ mở rộng hơn vì mỗi site xử lý một phần dữ liệu.
Khả năng chịu lỗi	Nếu site trung tâm lỗi, toàn hệ thống bị ảnh hưởng.	Nếu một site lỗi, các site khác vẫn có thể hoạt động một phần.
Quản lý dữ liệu	Dễ quản lý hơn vì dữ liệu nằm một nơi.	Khó quản lý hơn do phải xử lý phân mảnh, đồng bộ và truy vấn phân tán.
Tính nhất quán	Dễ đảm bảo nhất quán hơn.	Cần cơ chế kiểm soát nhất quán giữa các site.
Phù hợp với hệ thống đa kho	Không tối ưu vì mọi kho đều phụ thuộc vào trung tâm.	Phù hợp hơn vì kho nào xử lý dữ liệu của kho đó.

2. Phân tích

2.1 Phương án tập trung

Ở phương án tập trung, toàn bộ dữ liệu như đơn hàng, tồn kho, sản phẩm và phiếu xuất kho đều được lưu tại site trung tâm. Ưu điểm của cách này là việc quản lý dữ liệu và thực hiện truy vấn JOIN đơn giản hơn vì các bảng nằm cùng một nơi. Hệ thống không phải xử lý nhiều truy vấn phân tán và không cần truyền dữ liệu giữa các site khi join.

Tuy nhiên, phương án tập trung có nhược điểm lớn là các site kho luôn phải truy vấn về trung tâm. Khi số lượng đơn hàng tăng hoặc nhiều site cùng truy cập, site trung tâm dễ

trở thành điểm nghẽn. Ngoài ra, nếu kết nối mạng đến trung tâm bị lỗi, các site kho sẽ khó xử lý nghiệp vụ.

2.2 Phương án phân tán

Ở phương án phân tán, dữ liệu được chia theo khu vực hoặc theo kho. Ví dụ, DonHang có thể phân mảnh theo khu vực xử lý, TonKho phân mảnh theo MaKho, còn PhieuXuatKho lưu tại site của kho xuất. Cách này giúp mỗi site xử lý dữ liệu gần với nơi phát sinh nghiệp vụ, giảm tải cho trung tâm và tăng tốc các truy vấn cục bộ.

Nhược điểm của phương án phân tán là khi cần JOIN giữa các bảng nằm ở nhiều site khác nhau, hệ thống có thể phải truyền dữ liệu qua mạng. Nếu chuyển toàn bộ bảng từ site này sang site khác để join, lượng dữ liệu trung gian sẽ lớn và truy vấn bị chậm. Vì vậy, khi dùng phương án phân tán, cần tối ưu truy vấn bằng cách lọc dữ liệu sớm, nhân bản các bảng dùng chung như sản phẩm/danh mục, hoặc sử dụng semijoin để chỉ truyền các khóa và bản ghi cần thiết.

3. Kết luận

Phương án tập trung đơn giản hơn và dễ đảm bảo nhất quán, nhưng không phù hợp khi hệ thống có nhiều kho, nhiều khu vực và lượng truy vấn lớn.

Phương án phân tán phức tạp hơn nhưng phù hợp hơn với hệ thống bán hàng đa kho vì dữ liệu được xử lý gần nơi phát sinh, giảm tải cho trung tâm và tăng khả năng mở rộng.

Do đó, hệ thống nên chọn phương án phân tán cho các bảng nghiệp vụ lớn như đơn hàng, tồn kho, phiếu xuất kho; đồng thời nhân bản các bảng dùng chung như sản phẩm, danh mục, thương hiệu để hạn chế join phân tán.

CHƯƠNG 9. ĐÁNH GIÁ HỆ THỐNG

9.1. Kết quả đạt được

Đồ án đã phân tích, thiết kế và mô phỏng thành công cơ sở dữ liệu phân tán cho bài toán quản lý hệ thống bán hàng trực tuyến đa kho. Cụ thể, nhóm đã đạt được những kết quả trọng tâm sau:

- Thiết kế lược đồ toàn cục: Xây dựng mô hình ER và lược đồ quan hệ gồm 18 thực thể, bao quát toàn diện các luồng nghiệp vụ phức tạp từ quản lý sản phẩm, tồn kho đến điều phối đơn hàng.
- Chiến lược phân mảnh và cấp phát: Ứng dụng linh hoạt phân mảnh ngang nguyên thủy và dẫn xuất cho các bảng giao dịch (Kho, Tồn kho, Đơn hàng, Phiếu xuất/nhập) nhằm giữ dữ liệu ở sát nơi phát sinh. Đồng thời, áp dụng chiến lược nhân bản toàn phần cho nhóm dữ liệu tham chiếu (Danh mục, Sản phẩm) để tối ưu hóa hiệu năng tra cứu.
- Triển khai hạ tầng phân tán: Thiết lập thành công mô hình hệ thống gồm 3 Site độc lập: Site Trung Tâm (HQ), Site Miền Bắc và Site Miền Nam.
- Giải quyết bài toán truy vấn phân tán: Viết và phân tích chi phí cho 5 truy vấn phân tán điển hình, khai thác hiệu quả các phép kết nối (JOIN) và phép hợp (UNION ALL) xuyên Site.
- Kiểm soát tương tranh và nhất quán: Xử lý triệt để bài toán tranh chấp tồn kho bằng kỹ thuật khóa bi quan (UPDLOCK, HOLDLOCK) kết hợp với cơ chế giao dịch phân tán (2PC), đảm bảo tính nguyên tử (ACID) và loại bỏ hoàn toàn rủi ro âm tồn kho.

9.2. Đối chiếu với tiêu chí đánh giá

Tiêu chí	Mức đáp ứng	Ghi chú minh chứng
Thiết kế đúng đặc trưng phân tán của hệ nhiều kho	Đạt / Tốt	Tách biệt rõ ràng luồng dữ liệu mang tính cục bộ (vận hành kho, đặt hàng) và luồng dữ liệu cần truy vấn tổng hợp.
Phân mảnh dữ liệu hợp lý	Đạt / Tốt	Áp dụng đầy đủ phân mảnh ngang, phân mảnh dọc (bảng SanPham, NguoiDung) và phân mảnh dẫn xuất chặt chẽ.
Có xử lý truy vấn toàn hệ thống	Đạt / Tốt	Thiết kế 5 truy vấn phức tạp, tổng hợp dữ liệu từ cả Site Bắc và Nam lên Site Trung Tâm.
Có phân tích tồn kho và nhất quán dữ liệu	Đạt / Tốt	Giải quyết bằng thuật toán "Giữ hàng trước, trừ tồn sau" trong một Transaction hợp nhất.
Có minh họa rõ ràng và đánh giá kết quả	Đạt / Tốt	Chạy thực nghiệm song song 2 phiên giao dịch (Tab 1, Tab 2) để chứng minh cơ chế khóa (Locking) hoạt động chính xác.

9.3. Hạn chế

Dù đã đáp ứng tốt các yêu cầu cốt lõi, hệ thống mô phỏng vẫn còn một số điểm giới hạn:

- Độ trễ trong đồng bộ dữ liệu: Việc sử dụng cơ chế Snapshot Replication của SQL Server để nhân bản dữ liệu danh mục là phương thức bất đồng bộ. Điều này gây

ra một độ trễ nhất định (Latent Transactional Consistency) chứ không phải là đồng bộ thời gian thực tuyệt đối.

- Phụ thuộc vào Linked Server: Các giao dịch liên vùng (điều phối hàng từ Nam ra Bắc) đang phụ thuộc nhiều vào kết nối Linked Server. Trong hạ tầng mạng thực tế có độ trễ lớn, việc mở và chờ Commit các giao dịch qua mạng dạng này có nguy cơ gây thắt cổ chai (bottleneck) hoặc Deadlock phân tán.
- Tối ưu Join phân tán: Giải pháp Semi-join nhằm giảm thiểu lượng dữ liệu trung gian truyền qua mạng mới chỉ được phân tích ở mức độ lý thuyết và đề xuất, chưa được triển khai hoàn toàn bằng mã lệnh.

9.4. Hướng phát triển

Để hệ thống hoàn thiện và sẵn sàng áp dụng cho các doanh nghiệp thương mại điện tử thực tế, nhóm đề xuất các hướng cải tiến sau:

- Mở rộng kiến trúc Node: Tiến hành cấu hình thêm Site Miền Trung để kiểm chứng độ linh hoạt và khả năng mở rộng (Scalability) của ma trận cấp phát dữ liệu.
- Nâng cấp cơ chế Replication: Chuyển đổi từ mô hình Snapshot Replication sang Transactional Replication đối với các bảng cần tính cập nhật cao để giảm thiểu độ trễ đồng bộ giữa các Site.
- Tích hợp Message Broker: Xây dựng kiến trúc hướng sự kiện (Event-driven Architecture) kết hợp với hàng đợi tin nhắn (như RabbitMQ hoặc Apache Kafka) thay thế cho việc gọi trực tiếp qua Linked Server. Điều này giúp hệ thống điều phối đơn hàng liên vùng hoạt động mượt mà và chịu lỗi (Fault tolerance) tốt hơn ngay cả khi rớt mạng.
- Cài đặt thuật toán Semi-join: Đưa logic lọc khóa (Keys) của thuật toán Semi-join lên tầng ứng dụng (Application Layer) hoặc viết các Stored Procedure chuyên biệt để cắt giảm triệt để băng thông mạng khi thực thi các truy vấn có phép kết nối lớn.

KẾT LUẬN

Sự bùng nổ của thương mại điện tử hiện đại đòi hỏi các doanh nghiệp phải liên tục tối ưu hóa hạ tầng logistics và khả năng đáp ứng đơn hàng. Bài toán quản lý hệ thống bán hàng đa kho không chỉ đơn thuần là việc ghi chép số liệu, mà là thách thức về việc đồng bộ luồng thông tin và luồng hàng hóa theo thời gian thực trên một phạm vi địa lý rộng lớn.

Qua quá trình nghiên cứu và thực hiện đề tài "Hệ thống bán hàng trực tuyến đa kho", nhóm đã vận dụng thành công các kiến thức chuyên sâu của học phần Cơ sở dữ liệu phân tán để giải quyết trọn vẹn bài toán thực tiễn này. Đồ án đã đạt được những kết quả nổi bật sau:

Thứ nhất, về mặt kiến trúc dữ liệu, nhóm đã thiết kế thành công lược đồ toàn cục và tiến hành phân mảnh dữ liệu (phân mảnh ngang, dọc và dẫn xuất) một cách hợp lý. Việc phân bổ và nhân bản dữ liệu trên 3 Site (Trung tâm, Miền Bắc, Miền Nam) đã bám sát đúng đặc trưng "cục bộ hóa" của nghiệp vụ vận hành kho, đồng thời đảm bảo trải nghiệm tra cứu sản phẩm xuyên suốt cho người dùng.

Thứ hai, về mặt xử lý nghiệp vụ, đồ án không chỉ dừng lại ở thiết kế tĩnh mà còn xây dựng thành công các truy vấn và giao dịch phân tán phức tạp. Kịch bản điều phối đơn hàng liên vùng (khi một kho thiếu hàng) đã minh họa rõ nét khả năng giao tiếp xuyên Site của hệ thống.

Thứ ba, về tính toàn vẹn dữ liệu, bài toán tranh chấp tồn kho – rủi ro lớn nhất trong thương mại điện tử – đã được nhóm xử lý triệt để. Bằng việc áp dụng cơ chế khóa bi quan (UPDLOCK, HOLDLOCK) và quản lý giao dịch phân tán (2PC), hệ thống đảm bảo tính nhất quán tuyệt đối (ACID), loại bỏ hoàn toàn tình trạng bán vượt quá số lượng thực tế.

Nhìn chung, hệ thống đã đáp ứng đầy đủ và bám sát các mục tiêu, tiêu chí đánh giá ban đầu của đề tài. Kết quả của đồ án không chỉ giúp các thành viên củng cố vững chắc nền tảng lý thuyết về cơ sở dữ liệu phân tán, mà còn mang lại kinh nghiệm thực tiễn quý báu trong việc tư duy, thiết kế và tối ưu hóa kiến trúc hệ thống phần mềm cho các mô hình kinh doanh quy mô lớn.

TÀI LIỆU THAM KHẢO

[1] M. T. Özsu, P. Valduriez, *Principles of Distributed Database Systems (Fourth Edition)*, Springer, 2020.

[2] Microsoft Learn, *SQL Server Replication Overview*, https://learn.microsoft.com/en-us/sql/relational-databases/replication/sql-server-replication.

[3] Microsoft Learn, *Create Linked Servers (SQL Server Database Engine)*, https://learn.microsoft.com/en-us/sql/relational-databases/linked-servers/create-linked-servers-sql-server-database-engine.

[4] Hussein Nasser, *Distributed Transactions & Two Phase Commit (2PC) Explained*, YouTube, https://youtu.be/EbqjuqHhFfU.

[5] System Design Interview, *Understanding Distributed Database Architecture*, YouTube, https://youtu.be/kXhWwZ8103c.

PHỤ LỤC

Đường dẫn link Github chứa toàn bộ dự án: https://github.com/NTG259/csdlpt-n20