**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ ĐẶT VÀ GIAO HÀNG**

**Giáo viên hướng dẫn: Nguyễn Ngọc Ân**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
| 1 | 1771020534 | ĐÀO ĐỨC PHONG | 17-02 |
| 2 | 1771020358 | NGUYỄN QUỐC HUY | 17-02 |
| 3 | 1771020319 | ĐỖ KHÁNH HÙNG | 17-02 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ ĐẶT VÀ GIAO HÀNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020534 | ĐÀO ĐỨC PHONG | 05/11/2005 |  |  |
| 2 | 1771020358 | NGUYỄN QUỐC HUY | 18/03/2005 |  |  |
| 3 | 1771020319 | ĐỖ KHÁNH HÙNG | 24/11/2005 |  |  |

CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

# LỜI NÓI ĐẦU

Trong thời đại công nghệ số hóa ngày càng phát triển, cơ sở dữ liệu đã trở thành trái tim của mọi hệ thống thông tin, nó đóng vai trò quan trọng trong việc lưu trữ, quản lý và xử lý dữ liệu một cách hiệu quả. Có thể nói, hầu hết mọi mặt từ các ứng dụng thương mại điện tử, hay quản lý doanh nghiệp, đến các hệ thống hỗ trợ ra quyết định, cơ sở dữ liệu không chỉ là nền tảng kỹ thuật mà còn là yếu tố quyết định sự thành công của các tổ chức trong việc khai thác, quản lý thông tin. Và ở đây, môn học "Hệ quản trị cơ sở dữ liệu" không chỉ mang đến những kiến thức lý thuyết sâu rộng mà còn trang bị kỹ năng thực hành cần thiết, giúp chúng ta nắm bắt cách thiết kế, triển khai và quản trị các hệ thống cơ sở dữ liệu hiện đại.

Trong môn học này, chúng tôi được tiếp cận với các khái niệm cốt lõi như ngôn ngữ T-SQL, khung nhìn (views), chỉ mục (indexes), cũng như các kỹ thuật lập trình nâng cao như thủ tục (stored procedures), hàm (functions), trigger… Mỗi thành phần đều đóng vai trò quan trọng trong việc tối ưu hóa hiệu suất, đảm bảo tính toàn vẹn và bảo mật của dữ liệu. Chẳng hạn, trigger giúp tự động kiểm soát các thay đổi dữ liệu, trong khi chỉ mục thì lại cải thiện tốc độ truy vấn, mang lại trải nghiệm mượt mà cho người dùng cuối…

Hơn nữa, môn học còn khuyến khích chúng tôi áp dụng kiến thức vào thực tiễn thông qua việc giải quyết các bài toán cụ thể, từ thiết kế cơ sở dữ liệu cho các hệ thống quản lý đến triển khai các chức năng phức tạp như bảo mật và sao lưu dữ liệu. Đây là cơ hội để ta củng cố nền tảng lý thuyết và còn phát triển thêm cả tư duy logic và sáng tạo, sẵn sàng đối mặt với những thách thức trong lĩnh vực công nghệ thông tin.

Nhìn chung, "Hệ quản trị cơ sở dữ liệu" không chỉ là một môn học mà còn là chìa khóa mở ra cánh cửa khám phá thế giới dữ liệu số. Nó cung cấp kiến thức vững chắc, khơi dậy niềm đam mê học hỏi và ứng dụng công nghệ, đồng thời định hướng cho chúng tôi trên con đường sự nghiệp trong ngành quản trị hệ thống thông tin thời đại 4.0. Và trong bài tập lớn này, chúng tôi sẽ cùng nhau tìm hiểu và triển khai một hệ thống cơ sở dữ liệu thực tế với đề tài “Quản lý đặt và giao hàng”.

# MỤC LỤC

[LỜI NÓI ĐẦU 3](#_Toc192998417)

[MỤC LỤC 4](#_Toc192998418)

[CHƯƠNG I. MỞ ĐẦU 6](#_Toc192998419)

[1.1. Giới thiệu tổng quan về đề tài 6](#_Toc192998420)

[1.2. Mục đích và lý do chọn đề tài 6](#_Toc192998421)

[CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 8](#_Toc192998422)

[2.1. Xác định yêu cầu của hệ thống 8](#_Toc192998423)

[2.1.1. Yêu cầu chức năng 8](#_Toc192998424)

[2.1.2. Yêu cầu phi chức năng 9](#_Toc192998425)

[2.2 Thiết kế mô hình dữ liệu (ERD) 9](#_Toc192998426)

[2.2.1 Các thực thể và thuộc tính 9](#_Toc192998427)

[2.2.2 Mối quan hệ giữa các thực thể 11](#_Toc192998428)

[2.3 Mô hình ERD 12](#_Toc192998429)

[2.4. Chuyển đổi mô hình dữ liệu sang mô hình vật lý (bảng, khóa, ràng buộc) 12](#_Toc192998430)

[CHƯƠNG III. TRIỂN KHAI CƠ SỞ DỮ LIỆU 17](#_Toc192998431)

[3.1 Tạo cơ sở dữ liệu và các bảng với các định nghĩa khóa, ràng buộc và các mối quan hệ 17](#_Toc192998432)

[3.2 Nhập dữ liệu mẫu cho các bảng 19](#_Toc192998433)

[3.3 Truy vấn cơ bản (SELECT, INSERT, UPDATE, DELETE) 21](#_Toc192998434)

[3.4 Truy vấn nâng cao (INNER JOIN, GROUP BY, HAVING, SUBQUERY) 24](#_Toc192998435)

[CHƯƠNG IV. TRIỂN KHAI CƠ SỞ DỮ LIỆU 26](#_Toc192998436)

[4.1. Tạo 7-10 View từ cơ bản đến nâng cao 26](#_Toc192998437)

[4.2. Tạo 7-10 Index cần thiết cho các bảng 28](#_Toc192998438)

[4.3. Xây dựng 10 Stored Procedure (không tham số, có tham số, có OUTPUT) 29](#_Toc192998439)

[4.4. Tạo 10 Function (trả về kiểu vô hướng, bảng, biến bảng) 33](#_Toc192998440)

[4.5. Tạo 3-5 Trigger để kiểm soát dữ liệu 38](#_Toc192998441)

[CHƯƠNG V. BẢO MẬT VÀ QUẢN TRỊ 42](#_Toc192998442)

[5.1. Tạo người dùng 42](#_Toc192998443)

[5.2. Thiết lập các quyền truy cập và phân quyền người dùng 42](#_Toc192998444)

[5.3. Quản lý sao lưu và phục hồi dữ liệu 44](#_Toc192998445)

[CHƯƠNG VI. KẾT LUẬN 46](#_Toc192998446)

[CHƯƠNG VII. TÀI LIỆU THAM KHẢO 47](#_Toc192998447)

# CHƯƠNG I. MỞ ĐẦU

## 1.1. Giới thiệu tổng quan về đề tài

Ngày nay, thế giới càng ngày càng phát triển, và khi công nghệ thông tin len lỏi vào từng ngóc ngách của cuộc sống, chúng tôi nhận thấy rằng việc quản lý dữ liệu không còn là chuyện đơn giản trong thực tế bởi lượng thông tin đã quá nhiều mà nó đã trở thành yếu tố tối quan trọng trong doanh nghiệp. Từ những cửa hàng nhỏ lẻ cho đến các tập đoàn lớn, tất cả đều cần một hệ thống thông tin hiệu quả để theo dõi các hoạt động từ đó phục vụ khách hàng và cũng để cạnh tranh trên thị trường. Đặc biệt, trong lĩnh vực kinh doanh trực tuyến, nơi mà mọi thứ diễn ra nhanh chóng và liên tục, cơ sở dữ liệu chính xác là một công cụ giúp mọi thứ vận hành trơn tru hơn từ quản lý hàng hóa, đơn hàng, cho đến chăm sóc khách hàng.

Chính trong bối cảnh đó, hệ thống “Quản lý đặt và giao hàng” đã nổi lên như một giải pháp thiết yếu, đặc biệt khi mua sắm trực tuyến ngày càng trở thành thói quen của nhiều người. Chúng ta có thể bắt gặp những ví dụ thực tế rất gần gũi như các nền tảng thương mại điện tử lớn – Shopee, Lazada, hay thậm chí là Tiktok Shop – nơi mà việc đặt hàng và giao nhận diễn ra liên tục, từ đó đòi hỏi một cơ chế quản lý dữ liệu chặt chẽ và hiệu quả. Đề tài này chính là nỗ lực của nhóm 2 chúng tôi trong việc tái hiện một hệ thống như vậy.

## 1.2. Mục đích và lý do chọn đề tài

Mục đích chính của bài tập lớn này là giúp chúng tôi vận dụng kiến thức từ môn “Hệ quản trị cơ sở dữ liệu” để thiết kế và triển khai một hệ thống hoàn chỉnh. Chúng tôi muốn tạo ra một cơ sở dữ liệu không chỉ đáp ứng được các yêu cầu cơ bản như lưu trữ và truy xuất thông tin, mà còn tích hợp các kỹ thuật lập trình T-SQL như khung nhìn, chỉ mục, thủ tục, hàm và trigger để tối ưu hóa hiệu suất và đảm bảo tính toàn vẹn dữ liệu. Qua đó, chúng tôi hy vọng hiểu rõ hơn cách một hệ thống cơ sở dữ liệu thực sự hoạt động trong môi trường kinh doanh, đồng thời rèn luyện kỹ năng làm việc nhóm và giải quyết vấn đề thực tế.

Lý do chúng tôi chọn đề tài “Quản lý đặt và giao hàng” xuất phát từ cả yếu tố thực tiễn lẫn học thuật. Thứ nhất, đây là một bài toán quen thuộc trong đời sống, và nó gắn liền với nhu cầu mua sắm trực tuyến ngày càng tăng của xã hội. Việc tìm hiểu và xây dựng hệ thống này giúp chúng tôi hình dung rõ hơn cách các doanh nghiệp thương mại điện tử vận hành, từ đó có cái nhìn thực tế về ngành công nghệ thông tin. Thứ hai, đề tài này đủ phức tạp để chúng tôi áp dụng toàn bộ kiến thức đã học, từ thiết kế mô hình dữ liệu, viết truy vấn, lập trình T-SQL, cho đến quản trị và bảo mật hệ thống.

Cuối cùng, việc hoàn thành bài tập lớn này sẽ là một phần, một bước đệm quan trọng giúp chúng tôi tự tin hơn khi tiếp cận các dự án thực tế trong tương lai, đặc biệt là trong lĩnh vực quản trị cơ sở dữ liệu và phát triển hệ thống thông tin.

# ****CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU****

## 2.1. Xác định yêu cầu của hệ thống

Để xây dựng một hệ thống cơ sở dữ liệu hiệu quả cho bài toán “Quản lý đặt và giao hàng” của cửa hàng bán sĩ và lẻ, chúng tôi đã tiến hành phân tích kỹ lưỡng các thông tin được cung cấp. Dựa trên bài toán, chúng tôi chia các yêu cầu thành hai nhóm chính: yêu cầu chức năng (liên quan đến các tính năng cụ thể mà hệ thống cần thực hiện) và yêu cầu phi chức năng (liên quan đến chất lượng vận hành của hệ thống).

### ****2.1.1. Yêu cầu chức năng****

**- Quản lý thông tin nhà cung cấp**: Hệ thống cần lưu trữ đầy đủ các thông tin cơ bản của nhà cung cấp, bao gồm mã nhà cung cấp (để phân biệt từng đơn vị), tên nhà cung cấp, địa chỉ liên lạc, và số điện thoại. Những thông tin này rất quan trọng vì cửa hàng cần biết rõ mình đang giao dịch với ai, đặc biệt khi có vấn đề phát sinh trong quá trình đặt hàng hay giao hàng.

**- Quản lý thông tin mặt hàng**: Chúng tôi cần ghi nhận chi tiết mọi mặt hàng mà cửa hàng kinh doanh, bao gồm mã mặt hàng (duy nhất cho mỗi sản phẩm), tên hàng, đơn vị tính (như cái, hộp, kg), quy cách (mô tả đặc điểm như kích thước, màu sắc), và số lượng tồn kho hiện tại. Đây là những thông tin cốt lõi để cửa hàng nắm được tình trạng hàng hóa của mình.

**- Quản lý đơn đặt hàng**: Khi cửa hàng hết hàng và cần đặt thêm, hệ thống phải cho phép tạo đơn đặt hàng với các thông tin quan trọng: số đơn đặt hàng (mã định danh duy nhất), ngày đặt hàng, mã nhà cung cấp liên quan, ghi chú (nếu có yêu cầu đặc biệt như giao gấp), và danh sách các mặt hàng cần đặt.

**- Theo dõi và cập nhật dữ liệu**: Hệ thống cần cung cấp khả năng tra cứu trạng thái của đơn đặt hàng và phiếu giao hàng, chẳng hạn đơn nào đã hoàn tất giao, đơn nào còn thiếu bao nhiêu hàng. Điều này giúp cửa hàng kiểm soát được tiến độ nhập hàng từ nhà cung cấp.

**- Hỗ trợ truy vấn và báo cáo**: Hệ thống cần hỗ trợ các truy vấn cơ bản để quản lý dữ liệu hàng ngày, bao gồm: thêm thông tin mới (INSERT), tìm kiếm thông tin (SELECT), cập nhật dữ liệu (UPDATE), và xóa dữ liệu không cần thiết (DELETE). Chẳng hạn, nếu một nhà cung cấp ngừng hợp tác, hệ thống phải cho phép xóa thông tin của nhà cung cấp đó.

### ****2.1.2. Yêu cầu phi chức năng****

**- Hiệu suất**: Hệ thống phải xử lý nhanh các thao tác, đặc biệt khi cửa hàng có nhiều đơn đặt hàng hoặc phiếu giao hàng cần ghi nhận cùng lúc. Chẳng hạn, trong giờ cao điểm, nếu có 10 đơn đặt hàng được tạo, hệ thống không được bị chậm hay treo.

**- Tính chính xác và toàn vẹn dữ liệu**: Dữ liệu phải được lưu trữ chính xác, không để xảy ra trùng lặp hoặc sai lệch. Ví dụ, mã nhà cung cấp và mã mặt hàng cần duy nhất để tránh nhầm lẫn giữa các bản ghi. Khi cập nhật số lượng tồn kho hay trạng thái giao hàng, hệ thống phải kiểm soát chặt chẽ. Chẳng hạn, nếu đơn đặt hàng đã giao đủ 3 lần, không được phép thêm phiếu giao hàng mới cho đơn đó.

**- Dễ sử dụng và mở rộng:** Các câu lệnh truy vấn hoặc giao diện (nếu áp dụng) cần được thiết kế sao cho nhân viên cửa hàng dễ thao tác, không đòi hỏi quá nhiều kiến thức kỹ thuật. Ví dụ, để thêm một đơn đặt hàng mới, quy trình phải đơn giản và rõ ràng.

**- Bảo mật**: Thông tin nhà cung cấp, đơn đặt hàng, và phiếu giao hàng cần được bảo vệ khỏi truy cập trái phép. Chẳng hạn, một nhân viên bình thường chỉ được xem dữ liệu, còn việc sửa đổi đơn giá hay xóa đơn hàng phải do quản lý cấp cao thực hiện.

## ****2.2 Thiết kế mô hình dữ liệu (ERD)****

Sau khi xác định các yêu cầu của hệ thống ở phần trước, chúng tôi tiến hành thiết kế mô hình dữ liệu dạng ERD (Entity-Relationship Diagram) để thể hiện cấu trúc của cơ sở dữ liệu “Quản lý đặt và giao hàng”.

### ****2.2.1 Các thực thể và thuộc tính****

Chúng tôi đã xác định 6 thực thể chính từ bài toán, mỗi thực thể đại diện cho một đối tượng quan trọng trong hệ thống.

#### **a) NhaCungCap (Nhà cung cấp)**

Thuộc tính:

**+** MaNhaCungCap: Mã nhà cung cấp (kiểu số hoặc chuỗi, duy nhất).

+ TenNhaCungCap: Tên nhà cung cấp (chuỗi ký tự).

+ DiaChi: Địa chỉ nhà cung cấp (chuỗi ký tự).

+ DienThoai: Số điện thoại (chuỗi ký tự).

#### **b) MatHang (Mặt hàng)**

Thuộc tính:

**+ MaMatHang**: Mã mặt hàng (duy nhất).

+ TenMatHang: Tên mặt hàng (chuỗi ký tự).

+ DonViTinh: Đơn vị tính (ví dụ: cái, kg, hộp).

+ SoLuongTon: Số lượng tồn kho (số nguyên).

#### **c) DonDatHang (Đơn đặt hàng)**

Thuộc tính:

**+ SoDonDatHang:** Số đơn đặt hàng (duy nhất).

+ NgayDatHang: Ngày đặt hàng (kiểu ngày tháng).

+ MaNhaCungCap: Mã nhà cung cấp (liên kết với NhaCungCap).

+ GhiChu: Ghi chú (chuỗi ký tự, có thể để trống).

#### **d) ChiTietDonDatHang (Chi tiết đơn đặt hàng)**

Thuộc tính:

**+ SoDonDatHang:** Số đơn đặt hàng (liên kết với DonDatHang).

**+ MaMatHang:** Mã mặt hàng (liên kết với MatHang).

+ SoLuongDat: Số lượng đặt (số nguyên).

+ DonGiaDat: Đơn giá đặt (số tiền).

#### **e) PhieuGiaoHang (Phiếu giao hàng)**

Thuộc tính:

**+ SoPhieuGiaoHang:** Số phiếu giao hàng (duy nhất).

+ NgayGiao: Ngày giao hàng (kiểu ngày tháng).

+ SoDonDatHang: Số đơn đặt hàng (liên kết với DonDatHang).

#### **f) CungUng (Cung ứng)**

Thuộc tính:

**+ MaNhaCungCap:** Mã nhà cung cấp (liên kết với NhaCungCap).

**+ MaMatHang:** Mã mặt hàng (liên kết với MatHang).

### ****2.2.2 Mối quan hệ giữa các thực thể****

Chúng tôi xác định các mối quan hệ giữa các thực thể để đảm bảo dữ liệu được liên kết logic và phản ánh đúng quy trình thực tế của cửa hàng. Dưới đây là các mối quan hệ.

#### **a) NhaCungCap - DonDatHang**

- Loại: Một-nhiều (1:N).

- Mô tả: Một nhà cung cấp có thể nhận nhiều đơn đặt hàng từ cửa hàng, nhưng mỗi đơn đặt hàng chỉ liên quan đến một nhà cung cấp duy nhất.

- Liên kết: Thuộc tính MaNhaCungCap trong DonDatHang là khóa ngoại (foreign key) trỏ về MaNhaCungCap của NhaCungCap.

#### **b) DonDatHang - ChiTietDonDatHang**

- Loại: Một-nhiều (1:N).

- Mô tả: Một đơn đặt hàng có thể chứa nhiều mặt hàng (chi tiết), nhưng mỗi chi tiết chỉ thuộc về một đơn đặt hàng.

- Liên kết: SoDonDatHang trong ChiTietDonDatHang trỏ về SoDonDatHang của DonDatHang.

#### **c) MatHang - ChiTietDonDatHang**

- Loại: Một-nhiều (1:N).

- Mô tả: Một mặt hàng có thể xuất hiện trong nhiều chi tiết đơn đặt hàng, nhưng mỗi chi tiết chỉ liên quan đến một mặt hàng.

- Liên kết: MaMatHang trong ChiTietDonDatHang trỏ về MaMatHang của MatHang.

#### **e) DonDatHang - PhieuGiaoHang**

- Loại: Một-nhiều (1:N).

- Mô tả: Một đơn đặt hàng có thể có nhiều phiếu giao hàng (tối đa 3 lần), nhưng mỗi phiếu giao hàng chỉ thuộc về một đơn đặt hàng.

- Liên kết: SoDonDatHang trong PhieuGiaoHang trỏ về SoDonDatHang của DonDatHang.

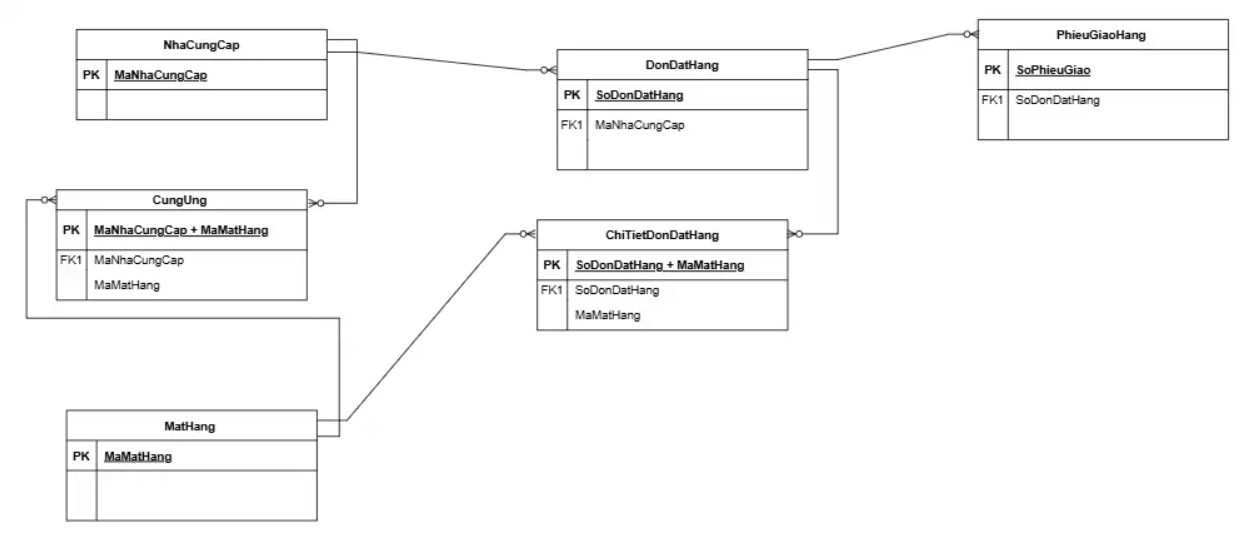
#### **f) NhaCungCap - MatHang (qua CungUng)**

- Loại: Nhiều-nhiều (N:N).

- Mô tả: Một nhà cung cấp có thể cung cấp nhiều mặt hàng, và một mặt hàng có thể được cung cấp bởi nhiều nhà cung cấp. Để thể hiện mối quan hệ này, chúng tôi tạo thực thể trung gian CungUng.

- Liên kết: CungUng chứa cặp MaNhaCungCap và MaMatHang làm khóa chính ghép, đồng thời là khóa ngoại trỏ về hai thực thể tương ứng.

## ****2.3 Mô hình ERD****



## 2.4. Chuyển đổi mô hình dữ liệu sang mô hình vật lý (bảng, khóa, ràng buộc)

Chúng tôi tạo 6 bảng tương ứng với 6 thực thể trong ERD. Mỗi bảng được định nghĩa với các cột (columns), kiểu dữ liệu phù hợp, và các ràng buộc (constraints) để phản ánh đúng yêu cầu của hệ thống “Quản lý đặt và giao hàng”:

### ****2.4.1. Bảng NhaCungCap****

- Mô tả: Lưu thông tin nhà cung cấp.

- Cấu trúc:

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- |
| MaNhaCungCap | INT | PRIMARY KEY, NOT NULL |
| TenNhaCungCap | NVARCHAR(100) | NOT NULL |
| DiaChi | NVARCHAR(200) | NULL |
| DienThoai | VARCHAR(15) | UNIQUE, NULL |

- Giải thích:

+ MaNhaCungCap là khóa chính, dùng kiểu INT để dễ quản lý và tăng tự động (identity).

+ TenNhaCungCap bắt buộc nhập (NOT NULL) vì không thể thiếu tên nhà cung cấp, dùng NVARCHAR để hỗ trợ tiếng Việt.

+ DiaChi có thể để trống (NULL) nếu không cần thiết.

+ DienThoai dùng VARCHAR với độ dài 15 (đủ cho số điện thoại), ràng buộc UNIQUE để tránh trùng lặp.

### 2.4.2. Bảng MatHang

- Mô tả: Lưu thông tin mặt hàng.

- Cấu trúc:

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- |
| MaMatHang | INT | PRIMARY KEY, NOT NULL |
| TenMatHang | NVARCHAR(100) | NOT NULL |
| DonViTinh | NVARCHAR(20) | NOT NULL |
| QuyCach | NVARCHAR(50) | NULL |
| SoLuongTon | INT | NOT NULL, DEFAULT 0 |

- Giải thích:

+ MaMatHang là khóa chính, kiểu INT, tự tăng.

TenMatHang và DonViTinh bắt buộc nhập, kiểu NVARCHAR hỗ trợ tiếng Việt.

+ QuyCach có thể để trống nếu không có mô tả đặc biệt.

+ SoLuongTon kiểu INT, mặc định là 0 nếu chưa nhập hàng, không cho phép NULL.

### 2.4.3. Bảng DonDatHang

- Mô tả: Lưu thông tin đơn đặt hàng.

- Cấu trúc:

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- |
| SoDonDatHang | INT | PRIMARY KEY, NOT NULL |
| NgayDatHang | DATE | NOT NULL |
| MaNhaCungCap | INT | FOREIGN KEY, NOT NULL |
| GhiChu | NVARCHAR(200) | NULL |

- Giải thích:

+ SoDonDatHang là khóa chính, kiểu INT, tự tăng.

+ NgayDatHang kiểu DATE để ghi ngày tháng, bắt buộc nhập.

+ MaNhaCungCap là khóa ngoại trỏ về bảng NhaCungCap, kiểu INT, không NULL vì mỗi đơn phải có nhà cung cấp.

+ GhiChu kiểu NVARCHAR, để trống nếu không cần.

### 2.4.4. Bảng ChiTietDonDatHang

- Mô tả: Lưu chi tiết mặt hàng trong đơn đặt hàng.

- Cấu trúc:

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- |
| SoDonDatHang | INT | PRIMARY KEY, FOREIGN KEY, NOT NULL |
| MaMatHang | INT | PRIMARY KEY, FOREIGN KEY, NOT NULL |
| SoLuongDat | INT | NOT NULL, CHECK (SoLuongDat > 0) |
| DonGiaDat | DECIMAL(15,2) | NOT NULL, CHECK (DonGiaDat >= 0) |

- Giải thích:

+ SoDonDatHang và MaMatHang ghép thành khóa chính ghép, đồng thời là khóa ngoại trỏ về DonDatHang và MatHang.

+ SoLuongDat kiểu INT, ràng buộc CHECK để đảm bảo số lượng lớn hơn 0.

+ DonGiaDat kiểu DECIMAL (15 số, 2 chữ số thập phân) để lưu giá tiền, CHECK không âm.

### 2.4.5. Bảng PhieuGiaoHang

- Mô tả: Lưu thông tin phiếu giao hàng.

- Cấu trúc:

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- |
| SoPhieuGiaoHang | INT | PRIMARY KEY, NOT NULL |
| NgayGiao | DATE | NOT NULL |
| SoDonDatHang | INT | FOREIGN KEY, NOT NULL |

- Giải thích:

+ SoPhieuGiaoHang là khóa chính, kiểu INT, tự tăng.

+ NgayGiao kiểu DATE, bắt buộc nhập.

+ SoDonDatHang là khóa ngoại trỏ về DonDatHang, không NULL vì mỗi phiếu phải liên quan đến một đơn.

### 2.4.6. Bảng CungUng

- Mô tả: Lưu mối quan hệ cung ứng giữa nhà cung cấp và mặt hàng.

- Cấu trúc:

| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- |
| MaNhaCungCap | INT | PRIMARY KEY, FOREIGN KEY, NOT NULL |
| MaMatHang | INT | PRIMARY KEY, FOREIGN KEY, NOT NULL |

- Giải thích:

+ MaNhaCungCap và MaMatHang ghép thành khóa chính ghép, đồng thời là khóa ngoại trỏ về NhaCungCap và MatHang.

+ Cả hai đều không NULL để đảm bảo mối quan hệ cung ứng luôn rõ ràng.

# CHƯƠNG III. TRIỂN KHAI CƠ SỞ DỮ LIỆU

## 3.1 Tạo cơ sở dữ liệu và các bảng với các định nghĩa khóa, ràng buộc và các mối quan hệ

--Tạo cơ sở dữ liệu

CREATE DATABASE QLDatGiaoHang;

GO

USE QLDatGiaoHang;

GO

-- Tạo bảng Nhà cung cấp

CREATE TABLE Supplier (

SupplierID INT PRIMARY KEY,

Name NVARCHAR(100),

Address NVARCHAR(200),

Phone VARCHAR(20)

);

-- Tạo bảng Mặt hàng

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

ProductName NVARCHAR(100),

Unit NVARCHAR(50),

Specification NVARCHAR(200),

Inventory INT

);

-- Tạo bảng Đơn đặt hàng

CREATE TABLE PurchaseOrder (

OrderID INT PRIMARY KEY,

OrderDate DATE,

SupplierID INT,

Note NVARCHAR(500),

FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)

);

-- Chi tiết đơn hàng

CREATE TABLE OrderDetail (

OrderID INT,

ProductID INT,

OrderQuantity INT,

UnitPrice DECIMAL(10,2),

PRIMARY KEY (OrderID, ProductID),

FOREIGN KEY (OrderID) REFERENCES PurchaseOrder(OrderID),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID)

);

-- Phiếu giao hàng

CREATE TABLE DeliverySlip (

DeliveryID INT PRIMARY KEY,

DeliveryDate DATE,

OrderID INT,

FOREIGN KEY (OrderID) REFERENCES PurchaseOrder(OrderID)

);

-- Bảng trung gian NCC cung cấp MH

CREATE TABLE Supply (

SupplierID INT,

ProductID INT,

PRIMARY KEY (SupplierID, ProductID),

FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID)

);

-- Chi tiết giao hàng

CREATE TABLE DeliveryDetail (

DeliveryID INT,

ProductID INT,

DeliveredQuantity INT,

PRIMARY KEY (DeliveryID, ProductID),

FOREIGN KEY (DeliveryID) REFERENCES DeliverySlip(DeliveryID),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID)

);

## ****3.2 Nhập dữ liệu mẫu cho các bảng****

### 3.2.1. Nhập dữ liệu cho Supplier

INSERT INTO Supplier VALUES

(1, 'NCC A', 'Hà Nội', '0987654321'),

(2, 'NCC B', 'TP.HCM', '0912345678'),

(3, 'NCC C', 'Đà Nẵng', '0978123456'),

(4, 'NCC D', 'Cần Thơ', '0933123456'),

(5, 'NCC E', 'Hải Phòng', '0988999888'),

(6, 'NCC F', 'Bình Dương', '0911111111'),

(7, 'NCC G', 'Nha Trang', '0922222222'),

(8, 'NCC H', 'Vũng Tàu', '0933333333'),

(9, 'NCC I', 'Huế', '0944444444'),

(10, 'NCC J', 'Quảng Ninh', '0955555555'),

(11, 'NCC K', 'Long An', '0966666666'),

(12, 'NCC L', 'Bắc Ninh', '0977777777'),

(13, 'NCC M', 'Hải Dương', '0988888888'),

(14, 'NCC N', 'Thái Nguyên', '0999999999'),

(15, 'NCC O', 'Lâm Đồng', '0900000000');

### 3.2.2. Nhập dữ liệu cho Product

INSERT INTO Product VALUES

(101, 'Gạo', 'Bao', 'Loại 1', 100),

(102, 'Đường', 'Kg', 'Hạt trắng', 200),

(103, 'Sữa', 'Thùng', 'Nguyên kem', 50),

(104, 'Dầu ăn', 'Chai', '1000ml', 150),

(105, 'Dầu ăn', 'Chai', '500ml', 80),

(106, 'Nước ngọt', 'Thùng', '24 lon', 200),

(107, 'Bánh mì', 'Cái', 'Nhân thịt', 150),

(108, 'Trứng gà', 'Chục', 'Tươi', 200),

(109, 'Cà phê', 'Túi', 'Hạt rang', 150),

(110, 'Bia', 'Thùng', '330ml', 300),

(111, 'Nước mắm', 'Chai', 'Đặc biệt', 100),

(112, 'Mì tôm', 'Thùng', 'Vị tôm chua cay', 500),

(113, 'Kẹo', 'Gói', 'Vị trái cây', 400),

(114, 'Tã giấy', 'Hộp', 'Size M', 250),

(115, 'Khẩu trang', 'Hộp', '3 lớp', 1000),

(116, 'Bút bi', 'Cây', 'Mực xanh', 600),

(117, 'Vở', 'Quyển', '96 trang', 800);

### 3.2.3. Nhập dữ liệu cho OrderDetail

INSERT INTO OrderDetail VALUES

(1001, 101, 50, 200000), (1001, 102, 100, 15000),

(1002, 103, 30, 250000), (1003, 101, 20, 210000),

(1004, 105, 100, 45000), (1004, 106, 50, 120000),

(1005, 107, 200, 10000), (1005, 101, 30, 220000),

(1006, 102, 80, 16000), (1007, 108, 50, 30000),

(1007, 109, 100, 50000), (1008, 110, 30, 120000),

(1008, 111, 80, 25000), (1009, 112, 200, 15000),

(1009, 113, 150, 10000), (1010, 114, 100, 45000),

(1010, 115, 300, 5000), (1011, 116, 500, 8000),

(1011, 117, 400, 10000), (1012, 108, 60, 32000),

(1012, 110, 40, 130000), (1013, 109, 120, 48000),

(1013, 112, 180, 16000), (1014, 111, 90, 27000),

(1014, 113, 130, 11000), (1015, 114, 110, 46000),

(1015, 116, 450, 8200), (1016, 115, 320, 5200),

(1016, 117, 420, 10500);

### 3.2.4. Nhập dữ liệu cho DeliverySlip

INSERT INTO DeliverySlip VALUES

(501, '2023-10-05', 1001), (502, '2023-10-06', 1001),

(503, '2023-10-07', 1002), (504, '2023-10-12', 1004),

(505, '2023-10-13', 1004), (506, '2023-10-14', 1005),

(507, '2023-10-15', 1005), (508, '2023-10-17', 1004),

(509, '2023-10-18', 1007), (510, '2023-10-19', 1007),

(511, '2023-10-20', 1008), (512, '2023-10-21', 1009),

(513, '2023-10-22', 1010), (514, '2023-10-23', 1011),

(515, '2023-10-24', 1012), (516, '2023-10-25', 1013),

(517, '2023-10-26', 1014), (518, '2023-10-27', 1015);

### 3.2.5. Chi tiết giao hàng (DeliveryDetail)

INSERT INTO DeliveryDetail VALUES

(501, 101, 30), (501, 102, 50),

(502, 101, 20), (502, 102, 30),

(503, 103, 10), (504, 105, 50),

(504, 106, 20), (505, 105, 30),

(505, 106, 10), (506, 107, 100),

(506, 101, 15), (507, 107, 50),

(508, 105, 10), (509, 108, 20),

(509, 109, 50), (510, 108, 30),

(510, 109, 50), (511, 110, 15),

(511, 111, 40), (512, 112, 100),

(512, 113, 80), (513, 114, 50),

(513, 115, 150), (514, 116, 250),

(514, 117, 200), (515, 108, 30),

(515, 110, 20), (516, 109, 60),

(516, 112, 90), (517, 111, 45),

(517, 113, 65), (518, 114, 60),

(518, 116, 200);

### 3.2.6. Đơn đặt hàng (PurchaseOrder)

INSERT INTO PurchaseOrder VALUES

(1001, '2023-10-01', 1, 'Gấp'),

(1002, '2023-10-02', 2, 'Kiểm tra kỹ'),

(1003, '2023-10-03', 3, 'Giao vào buổi sáng'),

(1004, '2023-10-10', 4, 'Giao trước 15/10'),

(1005, '2023-10-11', 5, 'Kiểm tra hạn sử dụng'),

(1006, '2023-10-16', 2, 'Giao nhanh'),

(1007, '2023-10-17', 6, 'Giao sáng'),

(1008, '2023-10-18', 7, 'Kiểm tra kỹ'),

(1009, '2023-10-19', 8, 'Hàng dễ vỡ'),

(1010, '2023-10-20', 9, 'Giao trước 22/10'),

(1011, '2023-10-21', 10, 'Yêu cầu hóa đơn'),

(1012, '2023-10-22', 11, 'Không giao trễ'),

(1013, '2023-10-23', 12, 'Liên hệ trước'),

(1014, '2023-10-24', 13, 'Giao buổi chiều'),

(1015, '2023-10-25', 14, 'Đóng gói cẩn thận'),

(1016, '2023-10-26', 15, 'Hàng khuyến mãi');

### 3.2.7. NCC cung cấp MH (Supply)

INSERT INTO Supply VALUES

(1, 101), (1, 102), (2, 102), (2, 103), (3, 101), (3, 103),

(4, 105), (4, 106), (5, 107), (5, 101), (6, 108), (6, 109),

(7, 110), (7, 111), (8, 112), (8, 113), (9, 114), (9, 115),

(10, 116), (10, 117), (11, 108), (11, 110), (12, 109), (12, 112),

(13, 111), (13, 113), (14, 114), (14, 116), (15, 115), (15, 117);

## ****3.3 Truy vấn cơ bản (SELECT, INSERT, UPDATE, DELETE)****

**- SELECT:** Lấy tất cả đơn hàng nhà cung cấp A.

SELECT \* FROM PurchaseOrder WHERE SupplierID = 1;

**- INSERT**: Thêm một mặt hàng mới.

INSERT INTO Product VALUES (118, 'Giấy vệ sinh', 'Cuộn', '2 lớp', 300);

**- UPDATE**: Cập nhật số lượng tồn.

UPDATE Product SET Inventory = 200 WHERE ProductID = 102;

**- DELETE**: Xóa phiếu giao hàng.

DELETE FROM DeliverySlip WHERE DeliveryID = 503;

## ****3.4 Truy vấn nâng cao (INNER JOIN, GROUP BY, HAVING, SUBQUERY)****

**- INNER JOIN**: Liệt kê đơn hàng chưa giao đủ

SELECT

od.OrderID,

od.ProductID,

od.OrderQuantity - SUM(COALESCE(dd.DeliveredQuantity, 0)) AS Remaining

FROM OrderDetail od

LEFT JOIN DeliveryDetail dd ON od.ProductID = dd.ProductID

GROUP BY od.OrderID, od.ProductID, od.OrderQuantity

HAVING Remaining > 0;

**- GROUP BY**: Tổng giá trị đơn hàng theo nhà cung cấp.

SELECT

s.Name,

SUM(od.OrderQuantity \* od.UnitPrice) AS TotalValue

FROM PurchaseOrder po

JOIN Supplier s ON po.SupplierID = s.SupplierID

JOIN OrderDetail od ON po.OrderID = od.OrderID

GROUP BY s.Name;

- **HAVING**: Sản phẩm được giao nhiều nhất

SELECT TOP 1

p.ProductName,

SUM(dd.DeliveredQuantity) AS TotalDelivered

FROM Product p

JOIN DeliveryDetail dd ON p.ProductID = dd.ProductID

GROUP BY p.ProductName

ORDER BY TotalDelivered DESC;

**- SUBQUERY**: Tìm mặt hàng được đặt nhiều nhất.

SELECT TOP 1

p.ProductName,

SUM(od.OrderQuantity) AS TotalOrdered

FROM Product p

JOIN OrderDetail od ON p.ProductID = od.ProductID

GROUP BY p.ProductName

ORDER BY TotalOrdered DESC;

# CHƯƠNG IV. TRIỂN KHAI CƠ SỞ DỮ LIỆU

## ****4.1. Tạo 7-10 View từ cơ bản đến nâng cao****

### 4.1.1. View cơ bản: Hiển thị thông tin đơn hàng và nhà cung cấp

CREATE VIEW vw\_OrderSupplierInfo AS

SELECT

po.OrderID,

po.OrderDate,

s.Name AS SupplierName,

s.Phone

FROM PurchaseOrder po

JOIN Supplier s ON po.SupplierID = s.SupplierID;

### 4.1.2. View cơ bản: Hiển thị thông tin chi tiết đơn hàng và sản phẩm

CREATE VIEW vw\_OrderProductDetails AS

SELECT

od.OrderID,

p.ProductName,

od.OrderQuantity,

od.UnitPrice

FROM OrderDetail od

JOIN Product p ON od.ProductID = p.ProductID;

### 4.1.3. View nâng cao: Tổng sô lượng sản phẩm theo nhà cung cấp

CREATE VIEW vw\_TotalOrderBySupplier AS

SELECT

s.SupplierID,

s.Name,

SUM(od.OrderQuantity) AS TotalOrdered

FROM Supplier s

JOIN PurchaseOrder po ON s.SupplierID = po.SupplierID

JOIN OrderDetail od ON po.OrderID = od.OrderID

GROUP BY s.SupplierID, s.Name;

### 4.1.4. View nâng cao: Số lần giao hàng của mỗi đơn hàng

CREATE VIEW vw\_DeliveryCountPerOrder AS

SELECT

OrderID,

COUNT(DeliveryID) AS DeliveryCount

FROM DeliverySlip

GROUP BY OrderID;

### 4.1.5. View nâng cao: Thông tin giao hàng chi tiết

CREATE VIEW vw\_DeliveryDetails AS

SELECT

ds.DeliveryID,

ds.DeliveryDate,

po.OrderDate,

s.Name AS SupplierName,

dd.ProductID,

p.ProductName,

dd.DeliveredQuantity

FROM DeliverySlip ds

JOIN PurchaseOrder po ON ds.OrderID = po.OrderID

JOIN Supplier s ON po.SupplierID = s.SupplierID

JOIN DeliveryDetail dd ON ds.DeliveryID = dd.DeliveryID

JOIN Product p ON dd.ProductID = p.ProductID;

### 4.1.6. View nâng cao: Sản phẩm còn tồn kho dưới mức trung bình

CREATE VIEW vw\_LowInventoryProducts AS

SELECT

ProductID,

ProductName,

Inventory

FROM Product

WHERE Inventory < (SELECT AVG(Inventory) FROM Product);

### 4.1.7. View nâng cao: Đơn hàng chưa được giao đủ

CREATE VIEW vw\_IncompleteOrders AS

SELECT

od.OrderID,

p.ProductName,

od.OrderQuantity - ISNULL(SUM(dd.DeliveredQuantity), 0) AS Remaining

FROM OrderDetail od

LEFT JOIN DeliveryDetail dd ON od.ProductID = dd.ProductID AND od.OrderID = (SELECT OrderID FROM DeliverySlip WHERE DeliveryID = dd.DeliveryID)

GROUP BY od.OrderID, od.ProductID, od.OrderQuantity, p.ProductName

HAVING od.OrderQuantity - ISNULL(SUM(dd.DeliveredQuantity), 0) > 0;

## 4.2. Tạo 7-10 Index cần thiết cho các bảng

### 4.2.1. Index trên cột OrderID trong bảng PurchaseOrder

CREATE INDEX idx\_PurchaseOrder\_OrderDate ON PurchaseOrder(OrderDate);

### 4.2.2. Index trên cột OrderID và ProductID của bảng OrderDetail

CREATE INDEX idx\_OrderDetail\_OrderID\_ProductID ON OrderDetail(OrderID, ProductID);

### 4.2.3. Index trên cột OrderID trong bảng DeliverySlip

CREATE INDEX idx\_DeliverySlip\_OrderID ON DeliverySlip(OrderID);

### 4.2.4. Index trên cột ProductName của bảng Product

CREATE INDEX idx\_Product\_ProductName ON Product(ProductName);

### 4.2.5. Index trên cột Name của bảng Supplier

CREATE INDEX idx\_Supplier\_Name ON Supplier(Name);

### 4.2.6. Index trên cột DeliveryID và ProductID của bảng DeliveryDetail

CREATE INDEX idx\_DeliveryDetail\_DeliveryID\_ProductID ON DeliveryDetail(DeliveryID, ProductID);

### 4.2.7. Index trên cột Inventory của bảng Product

CREATE INDEX idx\_Product\_Inventory ON Product(Inventory);

## 4.3. Xây dựng 10 Stored Procedure (không tham số, có tham số, có OUTPUT)

### 4.3.1. Stored Procedure: Lấy tất cả đơn hàng

CREATE PROCEDURE sp\_GetAllOrders

AS

BEGIN

SELECT \* FROM PurchaseOrder;

END;

### 4.3.2. Stored Procedure: Lấy đơn hàng theo nhà SupplierID

CREATE PROCEDURE sp\_GetOrdersBySupplier

@SupplierID INT

AS

BEGIN

SELECT \* FROM PurchaseOrder WHERE SupplierID = @SupplierID;

END;

;

### 4.3.3. Stored Procedure: Thêm đơn hàng mới

CREATE PROCEDURE sp\_AddOrder

@OrderDate DATE,

@SupplierID INT,

@Note NVARCHAR(500)

AS

BEGIN

INSERT INTO PurchaseOrder (OrderDate, SupplierID, Note)

VALUES (@OrderDate, @SupplierID, @Note);

END;

### 4.3.4. Stored Procedure: Cập nhật số lượng tồn kho

CREATE PROCEDURE sp\_UpdateInventory

@ProductID INT,

@Quantity INT

AS

BEGIN

UPDATE Product

SET Inventory = Inventory + @Quantity

WHERE ProductID = @ProductID;

END;

### 4.3.5. Stored Procedure: Xoá đơn hàng (KT ràng buộc)

CREATE PROCEDURE sp\_DeleteOrder

@OrderID INT

AS

BEGIN

IF NOT EXISTS (SELECT 1 FROM DeliverySlip WHERE OrderID = @OrderID)

BEGIN

DELETE FROM PurchaseOrder WHERE OrderID = @OrderID;

END

ELSE

BEGIN

RAISERROR('Không thể xóa đơn hàng đã có phiếu giao!', 16, 1);

END

END;

### 4.3.6. Stored Procedure: tính tổng giá trị đơn hàng (có OUTPUT)

CREATE PROCEDURE sp\_CalculateOrderTotal

@OrderID INT,

@Total DECIMAL(10,2) OUTPUT

AS

BEGIN

SELECT @Total = SUM(OrderQuantity \* UnitPrice)

FROM OrderDetail

WHERE OrderID = @OrderID;

END;

### 4.3.7. Stored Procedure có: lấy sản phẩm theo khoảng tồn kho

CREATE PROCEDURE sp\_GetProductsByInventoryRange

@MinInventory INT,

@MaxInventory INT

AS

BEGIN

SELECT \*

FROM Product

WHERE Inventory BETWEEN @MinInventory AND @MaxInventory;

END;

### 4.3.8. Stored Procedure: thêm chi tiết giao hàng (KT số lượng)

CREATE PROCEDURE sp\_AddDeliveryDetail

@DeliveryID INT,

@ProductID INT,

@DeliveredQuantity INT

AS

BEGIN

DECLARE @OrderQuantity INT;

SELECT @OrderQuantity = OrderQuantity

FROM OrderDetail

WHERE ProductID = @ProductID

AND OrderID = (SELECT OrderID FROM DeliverySlip WHERE DeliveryID = @DeliveryID);

IF @DeliveredQuantity <= @OrderQuantity

BEGIN

INSERT INTO DeliveryDetail (DeliveryID, ProductID, DeliveredQuantity)

VALUES (@DeliveryID, @ProductID, @DeliveredQuantity);

END

ELSE

BEGIN

RAISERROR('Số lượng giao vượt quá số lượng đặt!', 16, 1);

END

END;

### 4.3.9. Stored Procedure: thống kê đơn hàng theo tháng (có OUTPUT)

CREATE PROCEDURE sp\_GetMonthlyOrderSummary

@Year INT,

@Month INT,

@TotalOrders INT OUTPUT,

@TotalAmount DECIMAL(10,2) OUTPUT

AS

BEGIN

SELECT

@TotalOrders = COUNT(DISTINCT po.OrderID),

@TotalAmount = SUM(od.OrderQuantity \* od.UnitPrice)

FROM PurchaseOrder po

JOIN OrderDetail od ON po.OrderID = od.OrderID

WHERE YEAR(po.OrderDate) = @Year

AND MONTH(po.OrderDate) = @Month;

END;

### 4.3.10. Stored Procedure: sao lưu dữ liệu nhà cung cấp

CREATE PROCEDURE sp\_BackupSuppliers

AS

BEGIN

SELECT \* INTO SupplierBackup FROM Supplier;

END;

## 4.4. Tạo 10 Function (trả về kiểu vô hướng, bảng, biến bảng)

### 4.4.1. Function: Tính tổng số lượng đặt hàng của sản phẩm

CREATE FUNCTION fn\_GetTotalOrderedQuantity(@ProductID INT)

RETURNS INT

AS

BEGIN

DECLARE @Total INT;

SELECT @Total = SUM(OrderQuantity)

FROM OrderDetail

WHERE ProductID = @ProductID;

RETURN @Total;

END;

### 4.4.2. Function vô hướng: Kiểm tra nhà CC tồn tại

CREATE FUNCTION fn\_SupplierExists(@SupplierID INT)

RETURNS BIT

AS

BEGIN

RETURN CASE

WHEN EXISTS (SELECT 1 FROM Supplier WHERE SupplierID = @SupplierID)

THEN 1

ELSE 0

END;

END;

### 4.4.3. Function trả về bảng: Lấy danh sách sản phẩm theo nhà CC

CREATE FUNCTION fn\_GetProductsBySupplier(@SupplierID INT)

RETURNS TABLE

AS

RETURN (

SELECT p.\*

FROM Product p

JOIN Supply s ON p.ProductID = s.ProductID

WHERE s.SupplierID = @SupplierID

);

### 4.4.4. Function: Tính tổng giá trị đơn hàng

CREATE FUNCTION fn\_CalculateOrderTotal(@OrderID INT)

RETURNS DECIMAL(10,2)

AS

BEGIN

DECLARE @Total DECIMAL(10,2);

SELECT @Total = SUM(OrderQuantity \* UnitPrice)

FROM OrderDetail

WHERE OrderID = @OrderID;

RETURN @Total;

END;

### 4.4.5. Function trả về bảng: Lấy đơn hàng trong khoảng thời gian

CREATE FUNCTION fn\_GetOrdersByDateRange(@StartDate DATE, @EndDate DATE)

RETURNS TABLE

AS

RETURN (

SELECT \*

FROM PurchaseOrder

WHERE OrderDate BETWEEN @StartDate AND @EndDate

);

### 4.4.6. Function đếm số lần giao hàng của đơn hàng

CREATE FUNCTION fn\_CountDeliveries(@OrderID INT)

RETURNS INT

AS

BEGIN

DECLARE @Count INT;

SELECT @Count = COUNT(\*)

FROM DeliverySlip

WHERE OrderID = @OrderID;

RETURN @Count;

END;

### 4.4.7. Function lấy thông tin chi tiết giao hàng (biến bảng)

CREATE FUNCTION fn\_GetDeliveryDetails(@DeliveryID INT)

RETURNS @Details TABLE (

ProductName NVARCHAR(100),

DeliveredQuantity INT,

UnitPrice DECIMAL(10,2))

AS

BEGIN

INSERT INTO @Details

SELECT

p.ProductName,

dd.DeliveredQuantity,

od.UnitPrice

FROM DeliveryDetail dd

JOIN Product p ON dd.ProductID = p.ProductID

JOIN OrderDetail od ON dd.ProductID = od.ProductID

AND od.OrderID = (SELECT OrderID FROM DeliverySlip WHERE DeliveryID = @DeliveryID);

RETURN;

END;

### 4.4.8. Function kiểm tra số lượng tồn kho

CREATE FUNCTION fn\_CheckInventory(@ProductID INT)

RETURNS INT

AS

BEGIN

DECLARE @Inventory INT;

SELECT @Inventory = Inventory

FROM Product

WHERE ProductID = @ProductID;

RETURN @Inventory;

END;

### 4.4.9. Function lấy top sản phẩm bán chạy

CREATE FUNCTION fn\_GetTopProducts(@TopN INT)

RETURNS TABLE

AS

RETURN (

SELECT TOP (@TopN)

p.ProductName,

SUM(od.OrderQuantity) AS TotalOrdered

FROM OrderDetail od

JOIN Product p ON od.ProductID = p.ProductID

GROUP BY p.ProductName

ORDER BY TotalOrdered DESC

);

### 4.4.10. Function tính doanh thu theo nhà CC

CREATE FUNCTION fn\_GetSupplierRevenue()

RETURNS @Revenue TABLE (

SupplierName NVARCHAR(100),

TotalRevenue DECIMAL(10,2))

AS

BEGIN

INSERT INTO @Revenue

SELECT

s.Name,

SUM(od.OrderQuantity \* od.UnitPrice)

FROM Supplier s

JOIN PurchaseOrder po ON s.SupplierID = po.SupplierID

JOIN OrderDetail od ON po.OrderID = od.OrderID

GROUP BY s.Name;

RETURN;

END;

## 4.5. Tạo 3-5 Trigger để kiểm soát dữ liệu

### 4.5.1. Trigger: Kiểm soát số lần giao hàng tối đa 3 lần / đơn

CREATE TRIGGER trg\_CheckMaxDelivery

ON DeliverySlip

AFTER INSERT

AS

BEGIN

DECLARE @OrderID INT, @Count INT;

SELECT @OrderID = OrderID FROM inserted;

SELECT @Count = COUNT(\*) FROM DeliverySlip WHERE OrderID = @OrderID;

IF @Count > 3

BEGIN

RAISERROR('Mỗi đơn chỉ được giao tối đa 3 lần!', 16, 1);

ROLLBACK TRANSACTION;

END

END;

### 4.5.2. Trigger: Cập nhật tồn khi giao hàng

CREATE TRIGGER trg\_UpdateInventoryOnDelivery

ON DeliveryDetail

AFTER INSERT

AS

BEGIN

UPDATE p

SET Inventory = Inventory + i.DeliveredQuantity

FROM Product p

JOIN inserted i ON p.ProductID = i.ProductID;

END;

### 4.5.3. Trigger: Ngăn chặn xóa nhà cung cấp có đơn hàng

CREATE TRIGGER trg\_PreventSupplierDelete

ON Supplier

INSTEAD OF DELETE

AS

BEGIN

IF EXISTS (SELECT 1 FROM deleted d JOIN PurchaseOrder po ON d.SupplierID = po.SupplierID)

BEGIN

RAISERROR('Không thể xóa nhà cung cấp đã có đơn hàng!', 16, 1);

ROLLBACK TRANSACTION;

END

ELSE

BEGIN

DELETE FROM Supplier WHERE SupplierID IN (SELECT SupplierID FROM deleted);

END

END;

### 4.5.4. Trigger: Kiểm tra số lượng giao không vượt quá số lượng đặt

CREATE TRIGGER trg\_CheckDeliveryQuantity

ON DeliveryDetail

AFTER INSERT

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted i

JOIN OrderDetail od ON i.ProductID = od.ProductID

AND od.OrderID = (SELECT OrderID FROM DeliverySlip WHERE DeliveryID = i.DeliveryID)

WHERE i.DeliveredQuantity > od.OrderQuantity

)

BEGIN

RAISERROR('Số lượng giao vượt quá số lượng đặt!', 16, 1);

ROLLBACK TRANSACTION;

END

END;

### 4.5.5. Trigger: Ghi log khi xoá đơn hàng

CREATE TABLE OrderDeleteLog (

LogID INT IDENTITY PRIMARY KEY,

OrderID INT,

DeletedDate DATETIME

);

CREATE TRIGGER trg\_LogOrderDeletion

ON PurchaseOrder

AFTER DELETE

AS

BEGIN

INSERT INTO OrderDeleteLog (OrderID, DeletedDate)

SELECT OrderID, GETDATE() FROM deleted;

END;

# CHƯƠNG V. BẢO MẬT VÀ QUẢN TRỊ

## 5.1. Tạo người dùng

### 5.1.1 Tạo Login (TK đăng nhập)

USE master;

GO

-- Tạo Login cho Quản trị viên

CREATE LOGIN AdminLogin WITH PASSWORD = 'Admin@123';

-- Tạo Login cho Nhân viên bán hàng

CREATE LOGIN SalesLogin WITH PASSWORD = 'Sales@123';

-- Tạo Login cho Nhân viên kho

CREATE LOGIN WarehouseLogin WITH PASSWORD = 'Warehouse@123';

### 5.1.2. Tạo User (TK trong Database)

USE QLDatGiaoHang; -- Thay tên database của bạn

GO

-- User cho Quản trị viên

CREATE USER AdminUser FOR LOGIN AdminLogin;

-- User cho Nhân viên bán hàng

CREATE USER SalesUser FOR LOGIN SalesLogin;

-- User cho Nhân viên kho

CREATE USER WarehouseUser FOR LOGIN WarehouseLogin;

## 5.2. Thiết lập các quyền truy cập và phân quyền người dùng

### 5.2.1. Phân quyền cho AdminUser (Toàn quyền)

-- Cấp quyền db\_owner (toàn quyền trên database)

ALTER ROLE db\_owner ADD MEMBER AdminUser;

### 5.2.2. Phân quyền cho SalesUser (Truy cập đơn hàng và sản phẩm)

-- Quyền SELECT, INSERT, UPDATE trên bảng PurchaseOrder và OrderDetail

GRANT SELECT, INSERT, UPDATE ON PurchaseOrder TO SalesUser;

GRANT SELECT, INSERT, UPDATE ON OrderDetail TO SalesUser;

-- Quyền SELECT trên bảng Product và Supplier

GRANT SELECT ON Product TO SalesUser;

GRANT SELECT ON Supplier TO SalesUser;

-- Từ chối quyền DELETE

DENY DELETE ON PurchaseOrder TO SalesUser;

### 5.2.3. Phân quyền cho WarehouseUser (Quản lý giao hàng và tồn kho)

-- Quyền SELECT, INSERT, UPDATE trên DeliverySlip và DeliveryDetail

GRANT SELECT, INSERT, UPDATE ON DeliverySlip TO WarehouseUser;

GRANT SELECT, INSERT, UPDATE ON DeliveryDetail TO WarehouseUser;

-- Quyền UPDATE trên cột Inventory của bảng Product

GRANT UPDATE ON Product(Inventory) TO WarehouseUser;

-- Từ chối quyền DELETE

DENY DELETE ON DeliverySlip TO WarehouseUser;

DENY DELETE ON DeliveryDetail TO WarehouseUser;

## 5.3. Quản lý sao lưu và phục hồi dữ liệu

### 5.3.1. Sao lưu cơ sở dữ liệu

--Sao lưu toàn bộ cơ sở dữ liệu

USE master;

GO

BACKUP DATABASE QLDatGiaoHang

TO DISK = 'C:\Backup\QLDatGiaoHang\_Full.bak'

--Sao lưu chênh lệch

BACKUP DATABASE QLDatGiaoHang

TO DISK = 'C:\Backup\QLDatGiaoHang\_Diff.bak'

WITH DIFFERENTIAL, NAME = 'Differential Backup of QLDatGiaoHang';

--Sao lưu nhật kí giao dịch

BACKUP LOG QLDatGiaoHang

TO DISK = 'C:\Backup\QLDatGiaoHang\_Log.trn'

WITH NAME = 'Transaction Log Backup of QLDatGiaoHang';

### 5.3.2. Phục hồi cơ sở dữ liệu

--Phục hồi từ full backup

RESTORE DATABASE QLDatGiaoHang

FROM DISK = 'C:\Backup\QLDatGiaoHang\_Full.bak'

WITH REPLACE;

--Phục hồi đến thời điểm cụ thể

RESTORE DATABASE QLDatGiaoHang

FROM DISK = 'C:\Backup\QLDatGiaoHang\_Full.bak'

WITH NORECOVERY;

RESTORE LOG QLDatGiaoHang

FROM DISK = 'C:\Backup\QLDatGiaoHang\_Log.trn'

WITH STOPAT = '2023-10-30T12:00:00', RECOVERY;

### 5.3.3. Kiểm tra phân quyền

- KT quyền của SalesUser

-- Xem quyền trên bảng PurchaseOrder

EXEC sp\_helprotect @username = 'SalesUser', @name = 'PurchaseOrder';

- KT quyền của WareHouseUser

-- Xem quyền trên bảng DeliverySlip

EXEC sp\_helprotect @username = 'WarehouseUser', @name = 'DeliverySlip';

d) Lưu ý

- Thay đổi đường dẫn sao lưu (C:\Backup\...) phù hợp với hệ thống của bạn.

- Đặt mật khẩu mạnh cho các Login để đảm bảo bảo mật.

- Kiểm tra định kỳ quyền truy cập và sao lưu để tránh mất dữ liệu.

- Test kỹ lưỡng các quyền và sao lưu trên môi trường không ảnh hưởng đến dữ liệu thật.

# CHƯƠNG VI. KẾT LUẬN

Quá trình thực hiện bài tập lớn về hệ quản trị cơ sở dữ liệu ,chúng tôi đã có cái nhìn hiểu rõ hơn về cách thiết kế, xây dựng và quản lý một cơ sở dữ liệu hoàn chỉnh. Từ việc tạo các bảng cơ bản như Supplier, Product, PurchaseOrder, đến việc thiết lập các mối quan hệ phức tạp thông qua các bảng trung gian như Supply và OrderDetail, chúng tôi đã học được cách tổ chức dữ liệu một cách khoa học và hiệu quả. Việc thêm dữ liệu mẫu và thực hiện các truy vấn từ cơ bản đến nâng cao đã giúp tôi nắm vững kiến thức về SQL, đặc biệt là cách sử dụng các câu lệnh SELECT, INSERT, UPDATE, DELETE, và các hàm tổng hợp như SUM, COUNT. Ngoài ra, việc viết các trigger để kiểm soát ràng buộc dữ liệu và tự động cập nhật thông tin còn giúp chúng tôi hiểu sâu hơn về cách đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong hệ thống.

Hệ thống của chúng tôi hiện tại đang có nhiều ưu điểm nổi bật, chẳng hạn như thiết kế rõ ràng, các bảng và mối quan hệ giữa chúng được xác định cụ thể, giúp dễ dàng quản lý và truy vấn dữ liệu. Các trigger được triển khai hiệu quả, đảm bảo tính toàn vẹn dữ liệu, chẳng hạn như giới hạn số lần giao hàng, tự động cập nhật tồn kho, và ngăn chặn việc xóa nhà cung cấp có liên quan đến dữ liệu khác. Tuy nhiên, hệ thống này cũng có một số nhược điểm, như chưa xử lý các trường hợp ngoại lệ phức tạp (ví dụ: khi dữ liệu đầu vào không hợp lệ) và chưa tối ưu hóa hiệu suất truy vấn khi dữ liệu phát triển lớn hơn.

Để hệ thống trở nên hoàn thiện và hiệu quả hơn, ta có thể phát triển theo một số hướng về sau. Đầu tiên, tối ưu hóa hiệu suất bằng cách áp dụng các kỹ thuật như indexing và partitioning là một việc cần thiết để cải thiện tốc độ truy vấn, đặc biệt khi dữ liệu ngày càng lớn. Thứ hai là hệ thống cần bổ sung cơ chế xử lý ngoại lệ chi tiết hơn, đảm bảo rằng các trường hợp dữ liệu đầu vào không hợp lệ được kiểm tra và xử lý kịp thời. Cuối cùng, hệ thống có thể mở rộng thêm các tính năng như tích hợp báo cáo tự động để đáp ứng nhu cầu đa dạng trong thực tế và nâng cao tính linh hoạt, dự đoán đây cũng sẽ trở thành một đề tài lớn từ nay về sau.

# CHƯƠNG VII. TÀI LIỆU THAM KHẢO

Trong quá trình tham khảo và nghiên cứu cũng như thu thập thông tin từ các nguồn tài liệu sau đây:

1. <https://www.ods.vn/tai-lieu/>
2. <https://cystack.net/blog/>
3. <https://securitybox.vn>
4. <https://vinsep.com/>
5. <https://securitybox.vn/>
6. <https://viblo.asia/>