**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO BÀI TẬP LỚN**

**THIẾT KẾ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỬA HÀNG**

**SIÊU THỊ CO.OP MART**

Giảng viên hướng dẫn: ThS. Phạm Thị Miên

Nhóm sinh viên thực hiện: MSSV:

- Bùi Văn Tân 6051071104

- Nguyễn Công Chí 6051071009

- Võ Văn Trí 6051071126

- Hồ Ngọc Thống 6051071118

- Huỳnh Xuân Lãm 6051071062

TP. Hồ Chí Minh, năm 2021

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO BÀI TẬP LỚN**

**THIẾT KẾ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỬA HÀNG**

**SIÊU THỊ CO.OP MART**

Giảng viên hướng dẫn: ThS. Phạm Thị Miên

Nhóm sinh viên thực hiện: MSSV:

- Bùi Văn Tân 6051071104

- Nguyễn Công Chí 6051071009

- Võ Văn Trí 6051071126

- Hồ Ngọc Thống 6051071118

- Huỳnh Xuân Lãm 6051071062

TP. Hồ Chí Minh, năm 2021

**LỜI CẢM ƠN**

Lời đầu tiên chúng em xin gửi lời cảm ơn chân thành đến quý thầy, cô giáo trong Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông vận tải.Những người đã truyền dạy, đã trang bị cho chúng em kho tàng kiến thức về bầu trời công nghệ thông tin rộng lớn. Em cám ơn chân thành đến cô Phạm Thị Miên đã tạo điều kiện để em có thể hoàn thành bài đồ án môn học. Nhờ đó mà em có cơ hội học tập, trau dồi những kiến thức, kĩ năng và làm quen với việc tìm hiểu các tài liệu. Em chân thành cảm ơn cô đã tận tâm chỉ bảo, lắng nghe những ý tưởng, tạo điều kiện tốt nhất có thể để giúp chúng em hoàn thành đề tài.

Ở đây, chúng em không chỉ học được kiến thức về sách vở mà chúng em còn học được các bài học, kỷ năng sống trước khi tạm biệt mái trường đại học thân yêu này và tiến ra biển đời mênh mông rộng lớn.

Trong quá trình học tập và tìm hiểu nhóm chúng em đã nỗ lực rất nhiều với mong muốn hoàn thành bài tập lớn một cách tốt nhất, nhưng sẽ không tránh khỏi những thiếu sót. Nhóm chúng em mong thầy, cô có thể thông cảm và cho nhóm em những ý kiến, đóng góp để nhóm em có thể hoàn thành bài tập lớn của nhóm một cách trọn vẹn.

Sau cùng, chúng em xin kính chúc Quý Thầy Cô lời chúc sức khoẻ, luôn hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Nhóm chúng em xin chân thành cảm ơn!

# 

**NHẬN XÉT CỦA GIÁO VIÊN**

*Tp. Hồ Chí Minh, ngày … tháng … năm …*

Giáo viên hướng dẫn

Phạm Thị Miên

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc72505628)

[NHẬN XÉT CỦA GIÁO VIÊN iv](#_Toc59718232)

[1. Lý do chọn đề tài. 1](#_Toc72505629)

[2. Hướng tiếp cận của đề tài. 1](#_Toc72505630)

[3. Mục tiêu nghiên cứu. 1](#_Toc72505631)

[4. Đối tượng và phạm vi nghiên cứu. 1](#_Toc72505632)

[5. Phương pháp nghiên cứu. 1](#_Toc72505633)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 2](#_Toc72505634)

[1.1 Hệ quản trị cơ sở dữ liệu SQL Server. 2](#_Toc72505635)

[1.1.1 Giới thiệu. 2](#_Toc72505636)

[1.1.2 Khái niệm. 2](#_Toc72505637)

[1.1.3 Kiến trúc của hệ quản trị cơ sở dữ liệu SQL Server. 3](#_Toc72505638)

[1.1.4 Ưu và nhược điểm. 5](#_Toc72505639)

[1.2 Ngôn ngữ lập trình C#. 5](#_Toc72505640)

[1.2.1 Giới thiệu. 5](#_Toc72505641)

[1.2.2 Khái niệm. 6](#_Toc72505642)

[1.2.3 Các đặc điểm của ngôn ngữ lập trình C#. 6](#_Toc72505643)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ 8](#_Toc72505644)

[2.1 Mô tả chi tiết bài toán. 8](#_Toc72505645)

[2.2 Lược đồ quan hệ 8](#_Toc72505646)

[2.3 Sơ đồ thực thể các mối liên kết 10](#_Toc72505655)

[2.4 Sơ đồ ER 12](#_Toc72505661)

[2.5 Mô hình quan hệ 12](#_Toc72505662)

[2.6 Lược đồ cơ sở dữ liệu 13](#_Toc72505663)

[2.7 Các loại bảng 13](#_Toc72505664)

[2.8 Danh sách các ràng buộc 16](#_Toc72505673)

[2.9 Danh sách phụ thuộc hàm 19](#_Toc72505677)

[3. Chuẩn hóa quan hệ 26](#_Toc72505701)

CHƯƠNG 3: THỦ TỤC HÀM VÀ PHÂN QUYỀN

[3.1 USER FUNCTION VÀ TRIGGER 9](#_Toc58247296)

[3.2: Phân Quyền: 15](#_Toc58247299)

[KẾT LUẬN VÀ KHUYẾN NGHỊ 27](#_Toc72505702)

[TÀI LIỆU THAM KHẢO 30](#_Toc59718259)

[PHÂN CHIA CÔNG VIỆC 30](#_Toc59718259)

# **LỜI MỞ ĐẦU**

1. **Lý do chọn đề tài.**

Tính tới thời điểm hiện tại trên thế giới cũng như ở Việt Nam công nghệ thông tin đang trở thành một ngành công nghệ mũi nhọn với nhiều ứng dụng trong đời sống. Bất kỳ một ngàng nghề nào, lĩnh vực nào cũng đều cần đến sự góp sức của công nghệ thông tin, chẳng hạn như xây dựng hệ thống quản lý danh sách nhân viên, quản lý tiền điện nước hay hệ thống quản lý sách trong thư viện, …...

Và trong sự phát triển ấy thì nhu cầu về ăn uống của con người ngày càng cao, từ có mặc thành mặc đẹp. Chính vì vậy các cửa hàng quần áo xuất hiện ngày càng nhiều. Đây là một cơ hội để nhiều người sử dụng phát triển kinh tế. Tuy nhiên nhiều cửa hàng quần áo vẫn quản lý cửa hàng theo cách thủ công, tính toán hóa đơn, doanh thu, các chi tiêu đều phụ thuộc vào con người điều đó dẫn tới những sai sót cũng như tốn thời gian.

Vậy làm sao để có thể khắc phục được những khó khăn trên, xuất phát từ nhu cầu thực tế em đã lựa chọn sẽ xây dựng một ứng dụng quản lý cửa hàng quần áo. Ứng dụng được tạo ra sẽ hỗ trợ các cửa hàng quần áo, quản lý cửa hàng theo quy trình, quản lý được nhân viên, sản phẩm, khách hàng, sản phẩm, cũng như các nghiệp vụ tính toán.

**Hướng tiếp cận của đề tài.**

* Nghiên cứu và hiểu được cách thức hoạt động của SQL Server.
* Thiết kế cơ sở dữ liệu sao cho phù hợp với ứng dụng đang phát triển.
* Nắm vững kiến thức nền tảng ngôn ngữ lập trình C#, để phát triển ứng dụng.
* Tiến hành kiểm tra và chạy thử ứng dụng.

1. **Mục tiêu nghiên cứu.**

* Hiểu được cách thức hoạt động của cơ sở dữ liệu.
* Khiến ứng dụng trở nên có ích hơn cho xã hội.

1. **Phạm vi nghiên cứu.**

- Về phần dữ liệu

Sử dụng hệ quản trị cơ sở dữ liệu SQL SERVER 2019

- Về phần xử lý

+ Thực hiện các thao tác thêm xóa sửa đối với khách hàng, nhân viên, hóa đơn, sản phẩm, danh mục sản phẩm, nhà cung cấp, phiếu nhập.

+ xử lý các ràng buộc toàn vẹn

+ Các tính toán về thành tiền trong hóa đơn, phiếu nhập được tính toán tự động

+ Tự động cập nhật số lượng tồn của sản phẩm. Theo dõi số lượng tồn của sản phẩm.

+ Thống kê doanh thu của cửa hàng, Thể hiện doanh thu qua biểu đồ.

+ Gửi mail cho khách hàng, gửi tin nhắn sms cho khách hàng.

+ Report hóa đơn

1. **Phương pháp nghiên cứu.**

* Tài liệu: Tìm hiểu và khai thác các tài liệu liên quan đến cơ sở dữ liệu, các nền tảng kiến thức về C#.
* Thực hành: Tiến hành kết nối cơ sở dữ liệu vớI C#. Tiến hành thao tác với dữ liệu.

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

1. **Hệ quản trị cơ sở dữ liệu SQL Server.**
2. **Giới thiệu.**

SQL server chính là cụm từ viết tắt của Structure Query Language được sử dụng nhiều trong các lĩnh vực bởi chức năng quản lý dữ liệu. Các ngôn ngữ cấp cao như: [Visual C](https://www.visualstudio.com/vs/cplusplus/), [Oracle](https://www.oracle.com/database/index.html), Visual Basic,… đều có trình hỗ trợ là SQL. Những ứng dụng khi chạy phải sử dụng SQL khi người dùng truy cập tới cơ sở dữ liệu thì không cần sử dụng trực tiếp SQL.

1. **Khái niệm.**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (Transact-SQL**)**để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….

Một vài ấn bản SQL Server:

**Enterprise** : chứa tất cả cá đặc điểm nổi bật của SQL Server, bao gồm nhân bộ máy cơ sở dữ liệu và các dịch vụ đi kèm cùng với các công cụ cho tạo và quản lý phân cụm SQL Server. Nó có thể quản lý các CSDL lớn tới 524 [petabytes](https://en.wikipedia.org/wiki/Petabyte) và đánh địa chỉ 12 [terabytes](https://en.wikipedia.org/wiki/Terabyte) bộ nhớ và hỗ trợ tới 640 bộ vi xử lý(các core của cpu)

**Standard** : Rất thích hợp cho các công ty vừa và nhỏ vì giá thành rẻ hơn nhiều so với Enterprise Edition, nhưng lại bị giới hạn một số chức năng cao cấp (advanced features) khác, edition này có thể chạy tốt trên hệ thống lên đến 4 CPU và 2 GB RAM.

**Developer** : Có đầy đủ các tính năng của Enterprise Edition nhưng được chế tạo đặc biệt như giới hạn số lượng người kết nối vào Server cùng một lúc…. Ðây là phiên bản sử dụng cho phát triển và kiểm tra ứng dụng. Phiên bản này phù hợp cho các cá nhân, tổ chức xây dựng và kiểm tra ứng dụng

**Workgroup**: ấn bản SQL Server Workgroup bao gồm chức năng lõi cơ sở dữ liệu nhưng không có các dịch vụ đi kèm. Chú ý phiên bản này không còn tồn tại ở SQL Server 2012.

**Express**: SQL Server Express dễ sử dụng và quản trị cơ sở dữ liệu đơn giản. Được tích hợp với Microsoft Visual Studio, nên dễ dàng để phát triển các ứng dụng dữ liệu, an toàn trong lưu trữ, và nhanh chóng triển khai. SQL Server Express là phiên bản miễn phí, không giới hạn về số cơ ở dữ liệu hoặc người sử dụng, nhưng nó chỉ dùng cho 1 bộ vi xử lý với 1 GB bộ nhớ và 10 GB file cơ sở dữ liệu. SQL Server Express là lựa chọn tốt cho những người dùng chỉ cần một phiên bản SQL Server 2005 nhỏ gọn, dùng trên máy chủ có cấu hình thấp, những nhà phát triển ứng dụng không chuyên hay những người yêu thích xây dựng các ứng dụng nhỏ.

1. **Kiến trúc của hệ quản trị cơ sở dữ liệu SQL Server.**



*Hình 1. Kiến trúc của hệ quản trị cơ sở dữ liệu SQL Server*

**Database Engine:** Cái lõi của SQL Server:

Ðây là một engine có khả năng chứa data ở các quy mô khác nhau dưới dạng table và support tất cả các kiểu kết nối (data connection) thông dụng của Microsoft như

ActiveX Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC).

Ngoài ra nó còn có khả năng tự điều chỉnh (tune up) ví dụ như sử dụng thêm các tài nguyên (resource) của máy khi cần và trả lại tài nguyên cho hệ điều hành khi một user log off.

**Replication:** Cơ chế tạo bản sao (Replica):

Giả sử bạn có một database dùng để chứa dữ liệu được các ứng dụng thường xuyên cập nhật. Một ngày đẹp trời bạn muốn có một cái database giống y hệt như thế trên một server khác để chạy báo cáo (report database) (cách làm này thường dùng để tránh ảnh hưởng đến performance của server chính). Vấn đề là report server của bạn cũng cần phải được cập nhật thường xuyên để đảm bảo tính chính xác của các báo cáo. Bạn không thể dùng cơ chế back up and restore trong trường hợp này.

**Integration Services (DTS):**Là một tập hợp các công cụ đồ họa và các đối tượng lập trình cho việc di chuyển, sao chép và chuyển đổi dữ liệu.

Nếu bạn làm việc trong một công ty lớn trong đó data được chứa trong nhiều nơi khác nhau và ở các dạng khác nhau cụ thể như chứa trong Oracle, DB2 (của IBM), SQL Server, Microsoft Access…. Bạn chắc chắn sẽ có nhu cầu di chuyển data giữa các server này (migrate hay transfer) và không chỉ di chuyển bạn còn muốn định dạng (format) nó trước khi lưu vào database khác, khi đó bạn sẽ thấy DTS giúp bạn giải quyết công việc trên dễ dàng.

**Analysis Services:** Một dịch vụ phân tích dữ liệu rất hay của Microsoft

Dữ liệu (Data) chứa trong database sẽ chẳng có ý nghĩa gì nhiều nếu như bạn không thể lấy được những thông tin (Information) bổ ích từ đó. Do đó Microsoft cung cấp cho bạn một công cụ rất mạnh giúp cho việc phân tích dữ liệu trở nên dễ dàng và hiệu quả bằng cách dùng khái niệm hình khối nhiều chiều (multi-dimension cubes) và kỹ thuật “khai phá dữ liệu” (data mining).

**Notification Services:** Dịch vụ thông báo Notification Services là nền tảng cho sự phát triển và triển khai các ứng dụng tạo và gửi thông báo. Notification Services có thể gửi thông báo theo địch thời đến hàng ngàn người đăng ký sử dụng nhiều loại thiết bị khác nhau.

**Reporting Services:** Reporting Services bao gồm các thành phần server và client cho việc tạo, quản lý và triển khai các báo cáo. Reporting Services cũng là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

**Full Text Search Service:** Dịch vụ SQL Server Full Text Search là một dịch vụ đặc biệt cho đánh chỉ mục và truy vấn cho dữ liệu văn bản không cấu trúc được lưu trữ trong các CSDL SQL Server. Đánh chỉ mục với Full Text Search có thể dduwowcj tạo trên bất kỳ cột dựa trên dữ liệu văn bản. Nó sẽ rất hiệu quả cho việc tìm các sử dụng toán tử LIKE trong SQL với trường hợp tìm văn bản.

**Service Broker :** Được sử dụng bên trong mỗi Instance, là môi trường lập trình cho việc các ứng dụng nhảy qua các Instance. Service Broker giao tiếp qua giao thức TCP/IP và cho phép các component khác nhau có thể được đồng bộ cùng nhau theo hướng trao đổi các message. Service Broker chạy như một phần của bộ máy cơ sở dữ liệu, cung cấp một nền tảng truyền message tin cậy và theo hàng đợi cho các ứng dụng SQL Server.

1. **Ưu và nhược điểm.**

* **Ưu điểm:**
* Không cần code: Rất dễ dàng để quản lý các hệ thống cơ sở dữ liệu bằng việc sử dụng SQL chuẩn mà không cần phải viết bất cứ dòng code nào.
* Tiêu chuẩn được quy định rõ ràng: SQL sử dụng hai tiêu chuẩn ISO và ANSI, trong khi với các non-SQL database không có tiêu chuẩn nào được tuân thủ.
* Tính di động: SQL có thể được sử dụng trong chương trình trong PCs, servers, laptops, và thậm chí cả mobile phones.
* Ngôn ngữ tương tác: Language này có thể được sử dụng để giao tiếp với cơ sở dữ liệu và nhận câu trả lời cho các câu hỏi phức tạp trong vài giây.
* Multiple data views: Với sự trợ giúp của ngôn ngữ SQL, người dùng có thể tạo các hiển thị khác nhau về cấu trúc cơ sở dữ liệu và cơ sở dữ liệu cho những người dùng khác nhau.
* **Nhược điểm:**
* Giao diện khó dùng: SQL có giao diện phức tạp khiến một số người dùng khó truy cập.
* Không được toàn quyền kiểm soát: Các lập trình viên sử dụng SQL không có toàn quyền kiểm soát cơ sở dữ liệu do các quy tắc nghiệp vụ ẩn.
* Thực thi: Hầu hết các chương trình cơ sở dữ liệu SQL đều có phần mở rộng độc quyền riêng của nhà cung cấp bên cạnh các tiêu chuẩn SQL.
* Giá cả: Chi phí vận hành của một số phiên bản SQL khiến một số lập trình viên gặp khó khăn khi tiếp cận.

1. **Ngôn ngữ lập trình C#.**
2. **Giới thiệu.**

Ngôn ngữ C# là một ngôn ngữ lập trình hướng đối tượng được được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. Ngôn ngữ này được xem là ngôn ngữ lập trình hướng đối tượng trong sáng và thuần nhất. Nó hiện thực hầu hết các tính chất tốt của mô hình hướng đối tượng giống như ngôn ngữ lập trình Java. Làm việc trên framework .NET, khả năng tạo ra nhiều ứng dụng mạnh mẽ và an toàn cho nền tảng Windows. Các thành phần máy chủ, dịch vụ web, ứng dụng di động, web và nhiều khả năng khác nữa.

1. **Khái niệm.**

C# là một loại ngôn ngữ lập trình hướng đối tượng đơn giản được phát triển trên nền tảng của C++ và Java. Ngôn ngữ cho phép người dùng sử dụng nhiều ngôn ngữ bậc cao trên những nền tảng và cấu trúc máy tính. Đây cũng là một trong số ít các ngôn ngữ lập trình có sự cân bằng giữa Java, C++, Visual Basic và cả Delphi.

1. **Các đặc điểm của ngôn ngữ lập trình C#**.

* **Đơn giản** – Đặc trưng đầu tiên của C# là loại bỏ những vấn đề phức tạo đã có trong Java và C++ như macro, template, tính đa kế thừa, lớp cơ sở ảo (hay còn gọi virtual base class). Các cú pháp, toán tử, biểu thức và cả tính năng của C# khá tương đương Java và C++ song đã qua cải tiến nên đơn giản hơn nhiều.
* **Hiện đại** – C# sở hữu nhiều khả năng như xử lý ngoại lệ, tự động trong thu gom bộ nhớ, bảo mật mã nguồn, dữ liệu mở rộng,... Đây là tất cả những đặc điểm được mong chờ ở một ngôn ngữ lập trình hiện đại.
* **Hướng đối tượng** – C# là một trong những ngôn ngữ được đánh giá là thuần hướng đối tượng. Nó sở hữu cả 4 tính chất quan trọng, đặc trưng là tính kế thừa, tính đóng gói, tính trừu tượng và tính đa hình.
* **Ít từ khóa** – Một trong những đặc trưng cơ bản của C# là ít từ khóa. Từ khóa được dùng trong ngôn ngữ này chỉ nhằm mục đích mô tả thông tin. Tuy ít từ khóa song C# vẫn rất mạnh mẽ. Lập trình viên có thể sử dụng nó để thực hiện mọi nhiệm vụ.
* **Mã nguồn mở** – C# là một trong những ngôn ngữ lập trình mã nguồn mở, được phát triển, điều hành một cách độc lập với Microsoft. Đây là một trong những nét độc đáo khiến ngôn ngữ này được biết đến và ưa chuộng rộng rãi.
* **Đa nền tảng** – C# là ngôn ngữ được sử dụng trong lập trình trên nhiều nền tảng. Các ứng dụng hoặc website được xây dựng bằng ngôn ngữ này có thể hoạt động tốt trên nhiều nền tảng như Windows, Linux và Mac.
* **Tiến hóa** – C# vẫn đang được nâng cấp và cho ra mắt các phiên bản mới với nhiều tính năng vượt trội và khả năng làm việc mạnh mẽ hơn. Hiện C# có thể làm việc với console, điện toán đám mây, phần mềm học máy,...
* Ngoài những đặc điểm trên thì còn một số ưu điểm nổi bật của C#:
  + Có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, nên cũng khá dể dàng tiếp cận và học nhanh với C#.
  + Có thể biên dịch trên nhiều nền tảng máy tính khác nhau.
  + Được xây dựng trên nền tảng của C++ và Java nên nó được thừa hưởng những ưu điểm của ngôn ngữ đó.
  + Là một phần của .NET Framework nên được sự chống lưng khá lớn đến từ bộ phận này.
  + Có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ**

1. **Mô tả chi tiết bài toán.**

Để thực hiện chương trình cũng như cụ thể hóa một cách dễ hiểu nhất thì cần dựng bài toán, cụ thể như sau: Một ngân hàng sẽ có nhiều chi nhánh để quản lý. Thuộc tính của **chi nhánh** là mã chi nhánh, tên chi nhánh, địa chỉ. Một chi nhánh có nhiều nhân viên làm việc và nhân viên thì chỉ làm việc tại một chi nhánh. Thuộc tính xác định của **nhân viên** là mã nhân viên, tên nhân viên, ngày vào làm. Mỗi khách hàng đến giao dịch tại ngân hàng có thể được một hoặc nhiều nhân viên phục vụ giao dịch. Thông tin **khách hàng** là chứng minh nhân dân, tên khách hàng, ngày sinh, giới tính, số điện thoại, địa chỉ. Nhân viên trong ngày có thể xử lý nhu cầu của một hoặc nhiều khách hàng. **Xử lý nhu cầu** gồm nhu cầu và ngày làm việc. Để sử dụng dịch vụ hoặc giao dịch tại ngân hàng, khách hàng bắt buộc sẽ phải tạo một tài khoản. Một khách hàng có thể tạo nhiều tài khoản cùng lúc. Tài khoản của mỗi khách hàng là duy nhất. Thông tin **tài khoản** là số tài khoản, tên đăng nhập, mật khẩu, số thẻ, số dư và ngày đăng kí. Tài khoản sẽ được quản lý bởi nhân viên của ngân hàng. Một nhân viên có thể quản lý một hoặc nhiều tài khoản ngân hàng. Thuộc tính của **quản lý** gồm ngày kiểm tra và ghi chú. Sau khi tạo tài khoản, khách hàng có thể thực hiện một hoặc nhiều giao dịch tại ngân hàng. Thuộc tính **giao dịch** gồm mã giao dịch và tên giao dịch. Sau khi giao dịch xong, khách hàng sẻ được nhận một phiếu **chi tiết giao dịch** trong đó có số tài khoản nhận, ngày giao dịch, số tiền giao dịch ghi chú.

1. **Lược đồ quan hệ**
   * 1. **Thực thể NHÂN VIÊN**

NHANVIEN(MaNV, TenNV, GioiTinh, DiaChi, ChucVu, SoDienThoai)

*Diễn giải*: Mỗi nhân viên sẽ có mã nhân viên để phân biệt với nhân viên khác (MaNV), ngoài ra còn có tên nhân viên (TenNV), giới tính (GioiTinh), địa chỉ (DiaChi), chức vụ (ChucVu) và số điện thoại (SoDienThoai).

* + 1. **Thực thể KHÁCH HÀNG**

KHACHHANG(MaKH, TenKH, NgaySinh, GioiTinh, DiaChi, SoDienThoai)

*Diễn giải*: Mỗi khách hàng sẽ có một mã khách hàng riêng để phân biệt khách hàng với nhau (MaKH). Một số thông tin của khách hàng như: tên khách hàng(TenKH), ngày sinh (NgaySinh), giới tính (GioiTinh), địa chỉ (DiaChi) và số điện thoại (SoDienThoai).

* + 1. **Thực thể NHÀ CUNG CẤP**

**N**HACUNGCAP(MaNCC, TenNCC, DiaChi, SoDienThoai)

*Diễn giải*: Một cửa hàng sẽ hợp tác với nhiều nhà cung. Mỗi nhà cung cấp bao gồm mã nhà cung cấp (MaNCC) để phân biệt với các nhà cung cấp khác. Mỗi nhà cung cấp còn được xác định bởi tên nhà cung cấp (TenNCC), địa chỉ (DiaChi) và số điện thoại (SoDienThoai).

* + 1. **Thực thể PHIẾU NHẬP**

PHIEUNHAP(MaPN, MaNCC, NgayNhap)

*Diễn giải*: Mỗi nhà cung cấp sẽ có mã phiếu nhập (MaPN) và mã nhà cung cấp (MaNCC) để biết phiếu nhập đó là của ai. Ngoài ra còn có thuộc tính là ngày nhập (NgayNhap).

* + 1. **Thực thể MẶT HÀNG**

MATHANG(MaMH, TenMH, LoaiMH, DonViTinh, SoLuong, GiaBan)

*Diễn giải*: Thực thể mặt hàng sẽ có mã mặt hàng (MaMH) Ngoài ra còn có thuộc tính tên quầy hàng (TenMH), loại mặt hàng (LoaiMH), đơn vị tính (DonViTinh), số lượng (SoLuong) và giá bán (GiaBan).

* + 1. **Thực thể HÓA ĐƠN**

HOADON(MaHD , TongTien, NgayTao**, MaNV, MaKH)**

*Diễn giải*: Mỗi hóa đơn sẽ có mã hóa đơn riêng (MaGD), mã nhân viên (MaNV) để dễ dàng xác định ai là nhân viên thanh toán hóa đơn này và mã khách hàng (MaKH) của hóa đơn. Ngoài ra còn có các thuộc tính tổng tiền (TongTien) và ngày tạo hóa đơn (NgayTao).

* + 1. **Thực thể CHI TIẾT PHIẾU NHẬP**

CHITIETPHIEUNHAP(MaPN, MaMH, DonGia, SoLuong)

*Diễn giải*: Mỗi phiếu nhập sẽ có mã phiếu nhập riêng (MaPN), mã mặt hàng (MaMH) để nhân viên dễ dàng xác định mặt hàng này thuộc phiếu nhập nào. Ngoài ra còn có các thuộc tính đơn giá (DonGia) và số lượng (SoLuong).

* + 1. **Thực thể CHI TIẾT HÓA ĐƠN**

CHITIETHOADON(MaKH, MaHD, DonGia, SoLuong)

*Diễn giải*: Mỗi khách hàng sẽ có mã khách hàng riêng (MaKH) và khách hàng sẽ có mã hóa đơn riêng (MaHD) để nhân viên dễ dàng xác định hóa đơn này là thuộc khách hàng nào. Ngoài ra còn có các thuộc tính đơn giá (DonGia) và số lượng (SoLuong).

1. **Sơ đồ thực thể các mối liên kết**
2. **Xét thực thể NHANVIEN và HOADON**

Ta thấy rằng một ngày nhân viên làm việc sẽ in được rất nhiều hóa đơn thanh toán và một hóa đơn thanh toán chỉ được duy nhất một nhân viên in ra. Do đó mối quan hệ giữa NHANVIEN và HOADON sẽ là một – nhiều.

(1, n) (1, 1)

Lập

HOADON

NHANVIEN

HOSOBENHAN

1. **Xét thực thể KHACHHANG và HOADON**

Ta thấy rằng một khách hàng có thể tạo ra được một hoặc nhiều hóa đơn thanh toán trong một ngày và ngược lại một hóa đơn đó chỉ thuộc duy nhất một khách hàng thanh toán. Do đó mối quan hệ giữa khách hàng và hóa đơn sẽ là một – nhiều.

(1, n) (1, 1)

HOADON

Có

KHACHHANG

1. **Xét thực thể HOADON và MATHANG**

Ta thấy rằng một hóa đơn sẽ được thanh toán cho nhiều mặt hàng và cũng như vậy một mặt hàng sẽ được thanh toán trên rất nhiều hóa đơn. Do đó mối quan hệ giữa HOADON và MATHANG là nhiều – nhiều.

Chi Tiết Phiếu Nhập

(1, n) (1, n)

MATHANG

HOADON

1. **Xét thực thể PHIEUNHAP và MATHANG**

Ta xét rằng tại một phiếu nhập thì sẽ có ít nhất một hoặc nhiều mặt hàng trong đó và cũng như vậy một mặt hàng sẽ được xuất ra từ rất nhiều phiếu nhập vì từ những nhà cung cấp khác nhau. Do đó mối quan hệ giữa PHIEUNHAP và MATHANG là nhiều – nhiều.

Chi Tiết Phiếu Nhập

(1, n) (1, n)

Mặt Hàng

Phiếu Nhập

1. **Xét thực thể NHACUNGCAP và PHIEUNHAP**

Ta thấy rằng tại một nhà cung cấp thì sẽ có rất nhiều lần xuất trên nhiều phiếu nhập và ngược lại một phiếu nhập chỉ duy nhất của một nhà cung cấp. Do đó mối quan hệ giữa NHACUNGCAP và PHIEUNHAP là một – nhiều

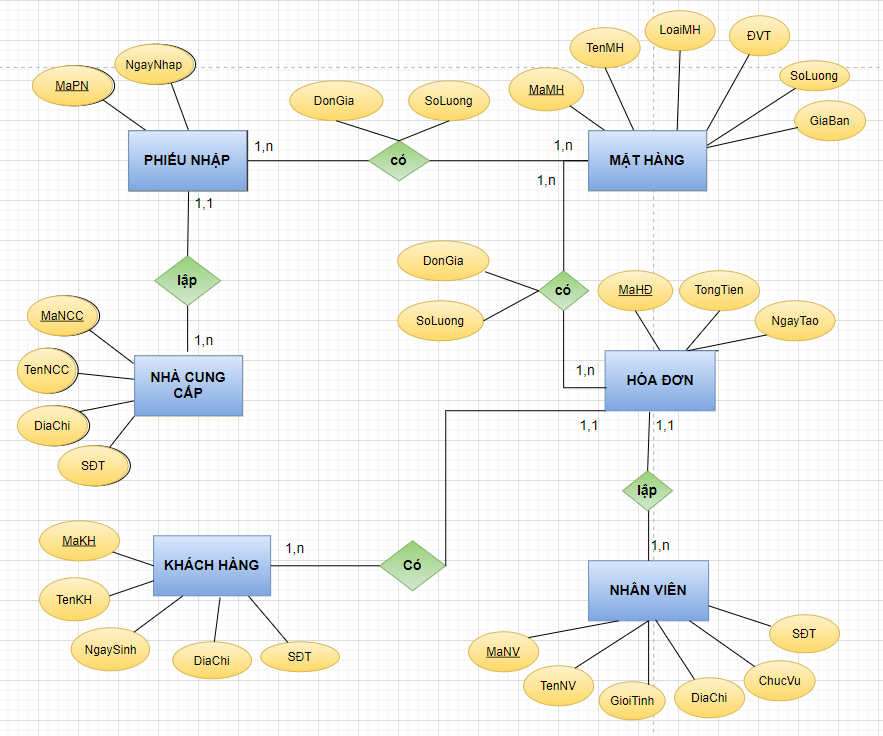
(1, n) (1, 1)

Lập

PHIEUNHAP

NHACUNGCAP

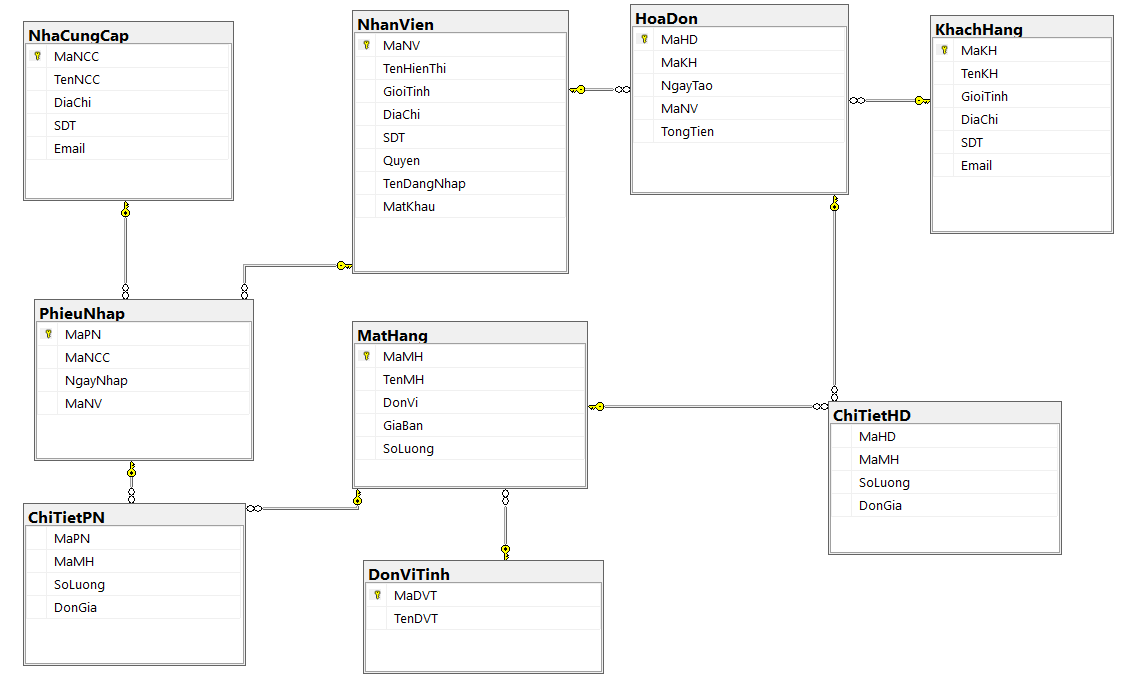
1. **Sơ đồ ER**



1. **Mô hình quan hệ**

* NHANVIEN(MaNV, TenNV, GioiTinh, DiaChi, ChucVu, SoDienThoai)
* KHACHHANG(MaKH, TenKH, GioiTinh, DiaChi, SoDienThoai)
* NHACUNGCAP(MaNCC, TenNCC, DiaChi, SoDienThoai)
* PHIEUNHAP(MaPN, NgayNhap, **MaNCC**)
* MATHANG(MaMH, TenMH, DonViTinh, SoLuong, GiaBan)
* HOADON(MaHD, TongTien, NgayTao**, MaNV, MaKH**)
* CHITIETPHIEUNHAP(MaPN, MaMH, DonGia, SoLuong)
* CHITIETHOADON(MaHD, MaNV, DonGia, SoLuong)

2.6. **Lược đồ cơ sở dữ liệu**



## 2.7 Các loại bảng

* + 1. **Bảng NHANVIEN**

Bảng chứa thông tin nhân viên

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaNV(Khóa chính) | varchar[20] | Mã nhân viên |
| TenNV  GioiTinh | nvarchar[30]  nvarchar[30] | Tên nhân viên  Giới tính nhân viên |
| DiaChi  ChucVu  SDT | nvarchar[50]  nvarchar[50]  varchar[20] | Địa chỉ nhân viên  Chức vụ nhân viên  Số điện thoại nhân viên |

* + 1. **Bảng KHACHHANG**

Bảng chứa thông tin khách hàng

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaKH(Khóa chính) | varchar[20] | Mã khách hàng |
| TenKH  NgaySinh  GioiTinh | nvarchar[30]  date  nvarchar[30] | Tên khách hàng  Ngày sinh khách hàng  Giới tính khách hàng |
| Diachi  SDT | nvarchcar[50]  varchar[10] | Địa chỉ khách hàng  Số điện thoại khách hàng |

* + 1. **Bảng NHACUNGCAP**

Bảng chứa thông tin nhà cung cấp

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaNCC(Khóa chính) | varchar[20] | Mã nhà cung cấp |
| TenNCC | nvarchar[30] | Tên nhà cung cấp |
| DiaChi | nvarchar[50] | Địa chỉ nhà cung cấp |
| SDT | varchar[10] | Số điện thoại nhà cung cấp |
|  |  |  |

* + 1. **Bảng PHIEUNHAP**

Bảng chứa thông tin phiếu nhập

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaPN(Khóa chính) | varchar[20] | Mã phiếu nhập |
| NgayNhap | date | Ngày nhập hàng |
| MaNCC(Khóa ngoại) | varchar[20] | Mã nhà cung cấp |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

* + 1. **Bảng MATHANG**

Bảng chứa thông tin mặt hàng

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaMH(Khóa chính) | varchar[20] | Mã mặt hàng |
| TenMH | nvarchar[50] | Tên mặt hàng |
| DonViTinh | nvarchar[20] | Đơn vị tính |
| SoLuong | varchar[20] | Số lượng |
| GiaBan | varchar[50] | Giá bán |
|  |  |  |
|  |  |  |

* + 1. **Bảng thông tin HOADON**

Bảng chứa thông tin hóa đơn

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaHĐ(Khóa chính) | varchar[20] | Mã hóa đơn |
| TongTien  NgayTao  MaNV(Khóa ngoại)  MaKH(Khóa ngoại) | varchar[50]  date  varchar[20]  varchar[20] | Tổng tiền hóa đơn  Ngày tạo hóa đơn  Mã nhân viên  Mã khách hàng |

* + 1. **Bảng thông tin CHITIETPHIEUNHAP**

Bảng chứa thông tin chi tiết phiếu nhập

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaPN(Khóa ngoại)  MaMH(Khóa ngoại) | varchar[20]  varchar[20] | Mã phiếu nhập  Mã mặt hàng |
| DonGia  SoLuong | float  varchar[100] | Đơn giá  Số lượng |

* + 1. **Bảng thông tin CHITIETHOADON**

Bảng chứa thông tin chi tiết háo đơn

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Chú thích** |
| MaHD(Khóa ngoại)  MaKH(Khóa ngoại) | varchar[20]  varchar[20] | Mã hóa đơn  Mã khách hàng |
| DonGia  SoLuong | float  varchar[100] | Đơn giá  Số lượng |

## Danh sách các ràng buộc

*ý hiệu:*

RB : Ràng buộc

+ : Có thể gây ra vi phạm ràng buộc

– : Không thể gây ra vi phạm ràng buộc

– (\*) : Nếu không bị vi phạm vì không được phép sửa đổi

* + 1. **Ràng buộc miền giá trị**
* **Ràng buộc 1: Trong quan hệ KHACHHANG thuộc tính giới tính chỉ có thể là Nam hoặc Nữ, thuộc tính số điện thoại không có chữ cái**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Khách Hàng | +(GioiTinh) | – | – (\*) |
| Khách Hàng | +(SoDienThoai) | – | +(SoDienThoai) |

* **Ràng buộc 2: Trong quan hệ NHANVIEN thuộc tính ngày sinh chỉ có thể là Nam hoặc Nữ, thuộc tính số điện thoại không có chữ cái**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Nhân Viên | +(GioiTinh) | – | – (\*) |
| Nhân Viên | +(SoDienThoai) | – | +(SoDienThoai) |

* **Ràng buộc 3: Trong quan hệ NHACUNGCAP thì thuộc tính số điện thoại không có chữ cái**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Nhà cung cấp | +(SoDienThoai) | – | +(SoDienThoai) |

* **Ràng buộc 4: Trong quan hệ PHIEUNHAP thì ngày nhập phải nhỏ hơn hoặc bằng ngày hiện tại.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Phiếu nhập | +(NgayNhap) | – | –(\*) |

* **Ràng buộc 5: Trong quan hệ HOADON thì ngày tạo hóa đơn phải nhỏ hơn hoặc bằng ngày hiện tại.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Hóa đơn | +(NgayTao) | – | –(\*) |

* + 1. **Ràng buộc liên thuộc tính**
    2. **Ràng buộc liên bộ**
* **Ràng buộc 1: Trong quan hệ NHANVIEN mỗi nhân viên sẽ phải có mã nhân viên MaNV khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Nhân Viên | +(MaNV) | – | –(\*) |

* **Ràng buộc 2: Trong quan hệ KHACHHANG mỗi khách hàng sẽ phải có mã khách hàng MaKH khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Khách Hàng | +(MaKH) | – | –(\*) |

* **Ràng buộc 3: Trong quan hệ NHACUNGCAP mỗi nhà cung cấp sẽ phải có mã mã nhà cung cấp MaNCC khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Nhà cung cấp | +(MaNCC) | – | –(\*) |

* **Ràng buộc 4: Trong quan hệ PHIEUNHAP mỗi phiếu nhập sẽ phải có mã phiếu nhập MaPN khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Phiếu nhập | +(MaPN) | – | -(\*) |

* **Ràng buộc 6: Trong quan hệ MATHANG mỗi mặt hàng sẽ phải có một mã mặt hàng MaMH khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Mặt Hàng | +(MaMH) | – | -(\*) |

* **Ràng buộc 7: Trong quan hệ HOADON mỗi hóa đơn sẽ phải có một mã hóa đơn MaHD khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Hóa đơn | +(MaHD) | – | -(\*) |

* **Ràng buộc 8: Trong quan hệ CHITIETPHIEUNHAP mỗi chi tiết phiếu nhập sẽ phải có một mã phiếu nhập MaPN và một mã mặt hàng MaMH khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Chi tiết phiếu nhập | +(MaPN, MaMH) | – | -(\*) |

* **Ràng buộc 8: Trong quan hệ CHITIETHOADON mỗi chi tiết hóa đơn sẽ phải có một mã hóa đơn MaHD và một mã mặt hàng MaMH khác nhau.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Chi tiết hóa đơn | +(MaHD, MaMH) | – | -(\*) |

* + 1. **Ràng buộc liên quan hệ**
* **Ràng buộc 1: Trong quan hệ PHIEUNHAP mỗi phiếu nhập sẽ phải có mã nhà cung cấp MaNCC tồn tại trong quan hệ NHACUNGCAP .**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Phiếu nhập | +(MaNV, MaNCC) | – | –(\*) |
| Nhân viên | - | +(MaNV) | +(MaNV) |
| Nhà cung cấp | - | +(MaNCC) | +(MaNCC) |

* **Ràng buộc 2: Trong quan hệ HOADON mỗi phiếu nhập sẽ phải có mã nhân viên MaNV tồn tại trong quan hệ NHANVIEN và mã khách hàng MaKH tồn tại trong quan hệ KHACHAHNG.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Hóa đơn | +(MaNV, MaKH) | – | –(\*) |
| Nhân viên | - | +(MaNV) | +(MaNV) |
| Khách hàng | - | +(MaKH) | +(MaKH) |

* **Ràng buộc 3: Trong quan hệ CHITIETPHIEUNHAP mỗi phiếu nhập sẽ phải có mã nhân viên MaNV tồn tại trong quan hệ NHANVIEN và mã khách hàng MaKH tồn tại trong quan hệ KHACHANG.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Chi tiết phiếu nhập | +(MaNV, MaKH) | – | –(\*) |
| Nhân viên | - | +(MaNV) | +(MaNV) |
| Khách hàng | - | +(MaKH) | +(MaKH) |

* **Ràng buộc 4: Trong quan hệ CHITIETHOADON mỗi hóa đơn sẽ phải có mã hóa đơn MaHD tồn tại trong quan hệ HOADON và mã mặt hàng tồn tại trong quan hệ MATHANG.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quan hệ** | **Thêm** | **Xóa** | **Sửa** |
| Chi tiết hóa đơn | +(MaHD, MaMH) | – | –(\*) |
| Hóa đơn | - | +(MaHD) | +(MaHD) |
| Mặt hàng | - | +(MaMH) | +(MaMH) |

* 1. **Danh sách phụ thuộc hàm**
     1. NHANVIEN(MaNV, TenNV, GioiTinh, DiaChi, ChucVu, SoDienThoai)

F1: MaNV TenNV

F2: MaNV 🡪 GioiTinh

F3: MaCN DiaChi

F4: MaNV 🡪 ChucVu

F5: MaNV SoDienThoai

* + 1. KHACHHANG(MaKH, TenKH, NgaySinh, GioiTinh, DiaChi, SoDienThoai)

F1: MaKH TenKH

F2: MaKH 🡪 NgaySinh

F3: MaKH 🡪 GioiTinh

F4: MaKH DiaChi

F5:MaKH 🡪 SoDienThoai

* + 1. NHACUNGCAP(MaNCC, TenNCC, DiaChi, SoDienThoai)

F1: MaNCC TenNCC

F2: MaNCC 🡪 DiaChi

F3: MaNCC SoDienThoai

* + 1. PHIEUNHAP(MaPN, NgayNhap, MaNCC)

F1: MaPN, MaNCC NgayNhap

* + 1. QUAYHANG(MaQH, TenQH)

F1: MaQH TenQH

* + 1. QUAYTHUNGAN(MaQTN, TenQTN)

F1: MaQTN TenQTN

* + 1. MATHANG(MaMH, TenMH, LoaiMH, DonViTinh, SoLuong, GiaBan, MaQH)

F1: MaMH, MaQH 🡪 TenMH

F2: MaMH, MaQH 🡪 LoaiMH

F3: MaMH, MaQH 🡪 DonViTinh

F4: MaMH, MaQH 🡪 SoLuong

F5: MaMH, MaQH 🡪 GiaBan

* + 1. HOADON(MaHD, TongTien, NgayTao, MaQTN, MaNV, MaKH)

F1: MaHD, MaQTN, MaNV, MaKH 🡪 TongTien

F2: MaHD, MaQTN, MaNV, MaKH 🡪 NgayTao

* + 1. CHITIETPHIEUNHAP(MaPN, MaMH, DonGia, SoLuong)

F1: MaPN, MaMH 🡪 DonGia

F2: MaPN, MaMH 🡪 SoLuong

* + 1. CHITIETHOADON(MaKH, MaHD, DonGia, SoLuong)

F1: MaKH, MaHD 🡪 DonGia

F2: MaKH, MaHD 🡪 SoLuong

## 3. Chuẩn hóa quan hệ

**Bảng NHANVIEN**

Đặt:

* A = MaNV
* B = TenNV
* C = NgaySinh
* D = GioiTinh
* E = DiaChi
* G = SoDienThoai

Q = { A, B, C, D, E, G }

F = { A 🡺 BCEDG }

Khóa chính là A

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì chỉ có 1 khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đề phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

**Bảng KHACHHANG**

Đặt:

* A = MaKH
* B = TenKH
* C = NgaySinh
* D = GioiTinh
* E = DiaChi
* G = SoDienThoai

Q = { A, B, C, D, E, G }

F = { A 🡺 BCEDG }

Khóa chính là A

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì chỉ có 1 khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đề phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

**Bảng NHACUNGCAP**

Đặt:

* A = MaNCC
* B = TenNCC
* C = DiaChi
* D = SoDienThoai

Q = { A, B, C, D }

F = { A 🡺 BCD }

Khóa chính là A

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì chỉ có 1 khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đề phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

**Bảng PHIEUNHAP**

Đặt:

* A = MaPN
* B = NgayNhap
* C = MaNCC

Q = { A, B, C}

F = { A 🡺 BC }

Khóa chính là A

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì chỉ có 1 khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đề phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

**Bảng MAHANG**

Đặt:

* A = MaMH
* B = TenMH
* C = LoaiMH
* D = DonViTinh
* E = SoLuong
* G = GiaBan

Q = { A, B, C, D, E, G}

F = { A 🡺 BCDEG }

Khóa chính là A

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì chỉ có 1 khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đề phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

**Bảng HOADON**

Đặt:

* A = MaHD
* B = TongTien
* C = NgayTao
* D = MaNV
* E = MaKH

Q = { A, B, C, D, E}

F = { A 🡺 BCDE }

Khóa chính là A

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì chỉ có 1 khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đề phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

**Bảng CHITIETPHIEUNHAP**

Đặt:

* A = MaPN
* B = MaMH
* C = DonGia
* D = SoLuong

Q = { A, B, C, D }

F = { AB 🡺 CD }

Khóa chính là AB

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì thuộc tính không khóa phụ thuộc hoàn toàn vào khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đều phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

**Bảng CHITIETHOADON**

Đặt:

* A = MaHD
* B = MaMH
* C = DonGia
* D = SoLuong

Q = { A, B, C, D }

F = { AB 🡺 CD }

Khóa chính là AB

* Đã đạt dạng chuẩn 1NF vì các thuộc tính đã là nguyên tố
* Đạt dạng chuẩn 2NF vì thuộc tính không khóa phụ thuộc hoàn toàn vào khóa
* Đạt dạng chuẩn 3NF vì tất cả thuộc tính không khóa đều phụ thuộc hàm trực tiếp vào khóa
* Đạt chuẩn BC vì thỏa mãn điều kiện: X 🡺 A ∈ F+ thì X là siêu khóa

## Phần 7: USER FUNCTION VÀ TRIGGER

### 7.1 User function:

a. Proc nhận vào tên một thân nhân, xuất ra tên của những nhân viên có quan hệ với thân nhân đó:  
create proc sp\_THANHNHAN\_NHANVIEN1(@tenThanNhan nvarchar(50))

as

begin

-- proc nhận vào tên một thân nhân, xuất ra tên của những nhân viên có quan hệ với thân nhân đó

if @tenThanNhan is null

begin

select \* from CONGNHAN

end

else

begin

select \*

from CONGNHAN

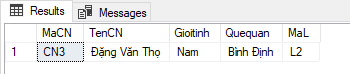
where MaCN IN

(select MaCN

from THANNHAN

where Ten = @tenThanNhan)

end

end  
  
test:  
exec sp\_THANHNHAN\_NHANVIEN1 N'Đặng Văn Cha'  
kết quả:  
  
  
b. Proc nhận vào tên Kho, xuất ra tên sản phẩm đang ở trong kho đó:  
create proc sp\_SANPHAM\_XUONG1(@tenKho nvarchar(50))

as

begin

-- nhận vào tên Kho, xuất ra tên sản phẩm đang ở trong kho đó

if @tenKho is null

begin

select \* from SANPHAM

end

else

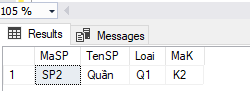
begin

select SP.\*

from KHO K inner join SANPHAM SP ON SP.MaK = K.MaK

where K.TenK = @tenKho

end

end  
  
test:  
exec sp\_SANPHAM\_XUONG1 N'Kho B'  
kết quả:  


c. Proc nhận vào tên 1 công nhân, cho biết công nhân đó có thân nhân trên 40 tuổi hay không:

create proc sp\_THANNHAN\_TUOI(@tenCN nvarchar(50))

as

begin

-- nhận vào tên 1 công nhân, cho biết công nhân đó có thân nhân trên 40 tuổi hay không

IF EXISTS(select \*

from CONGNHAN CN inner join THANNHAN TN on CN.MaCN = TN.MaCN

where CN.TenCN = @tenCN and DATEDIFF(year, TN.Ngaysinh, GETDATE()) > 40)

begin

print N'Công nhân ' + @tenCN + N' có thân nhân trên 40 tuổi'

end

else

begin

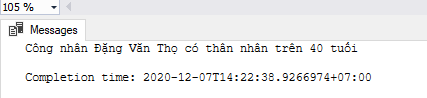
print N'Công nhân ' + @tenCN + N' không có thân nhân trên 40 tuổi'

end

end

test:

exec sp\_THANNHAN\_TUOI N'Đặng Văn Thọ'

kết quả:  


### 7.2 Triggers:

a. Giới tính mỗi bản ghi trong công nhân phải là nam hoặc nữ:  
create trigger tr\_CONGNHAN\_GIOITINH

on CONGNHAN

for insert, update

as

begin

-- Giới tính mỗi bản ghi trong công nhân phải là nam hoặc nữ

declare @gioitinh nvarchar(50)

declare pointer cursor for select Gioitinh from inserted

open pointer

fetch next from pointer into @gioitinh

while @@FETCH\_STATUS = 0

begin

if @gioitinh NOT IN (N'Nam', N'Nữ')

begin

print N'Giới tính không hợp lệ'

rollback tran

end

else

begin

print N'Giới tính thân nhân hợp lệ'

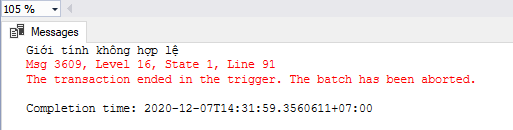
end

fetch next from pointer into @gioitinh

end

close pointer

deallocate pointer

end  
test:  
insert into CONGNHAN values('CN10', N'Lơ Minh Thái', N'BD', N'Quảng Bình', 'L1')  
kết quả:  
  
b. Giới tính mỗi bản ghi trong thân nhân phải là nam hoặc nữ:  
create trigger tr\_THANHNHAN\_GIOITINH

on THANNHAN

for insert, update

as

begin

-- Giới tính mỗi bản ghi trong thân nhân phải là nam hoặc nữ

declare @gioitinh nvarchar(50)

declare pointer cursor for select Gioitinh from inserted

open pointer

fetch next from pointer into @gioitinh

while @@FETCH\_STATUS = 0

begin

if @gioitinh NOT IN (N'Nam', N'Nữ')

begin

print N'Giới tính không hợp lệ'

rollback tran

end

else

begin

print N'Giới tính thân nhân hợp lệ'

end

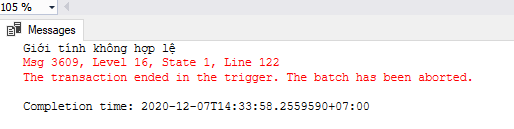
fetch next from pointer into @gioitinh

end

close pointer

deallocate pointer

end

test:  
insert into THANNHAN values ('CN9', N'Đặng Văn Cha', N'Bê', '1988-01-02', N'Ông Nội')  
kết quả:  
  
c. Thân nhân của công nhân phải trên 18 tuổi:  
create trigger tr\_THANHNHAN\_NGAYSINH

on THANNHAN

for insert, update

as

begin

declare @ngaysinh date

declare pointer cursor for select Ngaysinh from inserted

open pointer

fetch next from pointer into @ngaysinh

while @@FETCH\_STATUS = 0

begin

IF DATEDIFF(year, @ngaysinh, GETDATE()) < 18

begin

print N'Thân nhân nhập vào phải trên 18 tuổi'

rollback tran

end

else

begin

print N'Ngày sinh thân nhân hợp lệ'

end

fetch next from pointer into @ngaysinh

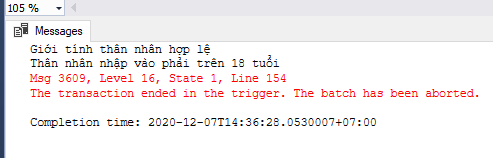
end

close pointer

deallocate pointer

end

test:  
insert into THANNHAN values ('CN5', N'Đặng Văn Cha', N'Nam', '2004-01-02', N'Ông C')

kết quả:  


d. Không được xoá công nhân có mã lương L1:  
create trigger tr\_CONGNHAN\_DELETE

ON CONGNHAN

for delete

as

begin

declare deletedCursor CURSOR FOR SELECT MaL from deleted

open deletedCursor

declare @MaLuongOnEachCursor char(10)

fetch next from deletedCursor INTO @MaLuongOnEachCursor

while @@FETCH\_STATUS = 0

begin

IF @MaLuongOnEachCursor = 'L1'

begin

print N'Không được xoá công nhân có mã lương L1'

rollback tran

end

fetch next from deletedCursor INTO @MaLuongOnEachCursor

end

close deletedCursor

deallocate deletedCursor

end

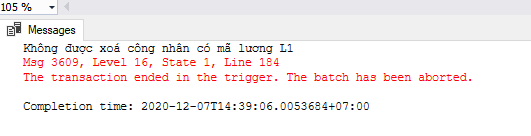
test:  
insert into CONGNHAN values('CN12', N'Lơ Minh Thái', N'Nam', N'Quảng Bình', 'L2')

insert into CONGNHAN values('CN13', N'Lơ Minh Thái', N'Nam', N'Quảng Bình', 'L3')

insert into CONGNHAN values('CN14', N'Lơ Minh Thái', N'Nam', N'Quảng Bình', 'L1')

insert into CONGNHAN values('CN15', N'Lơ Minh Thái', N'Nam', N'Quảng Bình', 'L5')

delete from CONGNHAN where MaCN IN ('CN12', 'CN13', 'CN14', 'CN15')

kết quả:  
  
  
e. Lương cơ bản không được nhỏ hơn 2000:  
create trigger tr\_LUONG\_LUONGCB

on LUONG

for insert, update

as

begin

-- Lương cơ bản không được nhỏ hơn 2000

declare pointer cursor for select LuongCB from inserted

open pointer

declare @luongcb int

fetch next from pointer into @luongcb

while @@fetch\_status = 0

begin

if @luongcb < 2000

begin

print N'Lương cơ bản không được dưới 2000'

rollback tran

end

fetch next from pointer into @luongcb

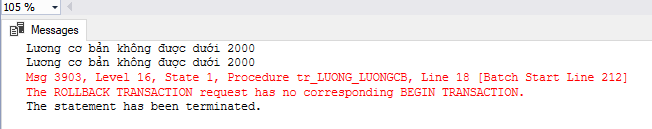
end

close pointer

deallocate pointer

end

test:  
update LUONG

set LuongCB = 1500  
kết quả:  


f. Một số điện thoại không thể dùng cho nhiều xưởng:  
create trigger tr\_XUONG\_SDT

on XUONG\_SDT

for insert, update

as

begin

-- một số điện thoại không thể dùng cho nhiều xưởng

declare pointer cursor for select SDT from inserted

open pointer

declare @SDT varchar(12)

fetch next from pointer into @SDT

while @@fetch\_status = 0

begin

if ( select count(\*) from XUONG\_SDT where SDT = @SDT ) >= 2

begin

print N'Một số điện thoại không thể dùng cho nhiều xưởng'

rollback tran

end

fetch next from pointer into @SDT

end

close pointer

deallocate pointer

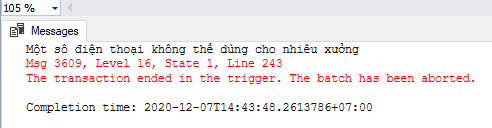
end

test:  
  
update XUONG\_SDT

set SDT = '09050508'

where MaXuong IN ('X1','X2')

kết quả:



## Phần 8: DATABASE USER VÀ PHÂN QUYỀN:

### 8.1 User role và database user:

a. User role:  
**- Tổ trưởng: Được quyền xem dữ liệu của table CONGNHAN và LUONG để cuối tháng dựa vào đó để phát lương  
- Quản lí: Được quyền xem và sửa dữ liệu của table CONGNHAN, LUONG và THANHNHAN để phục vụ cho việc xoá công nhân và thân nhân của họ khi họ nghỉ việc, thêm thông tin khi tuyển thêm công nhân vào.  
- Giám Đốc: Có tất cả quyền cơ bản trên tất cả bảng.**

b. Database user:

|  |  |
| --- | --- |
| **Tên người dùng** | **Vai trò** |
| **Nguyễn Văn A** | **Tổ trưởng** |
| **Nguyễn Thị B** | **Quản lí** |
| **Nguyễn Văn C** | **Giám Đốc** |

8.2 Phân quyền trong SQL SERVER:  
a. Tạo logins:create login AVanNguyen

with password = '123',

default\_database = QUANLYCONGNHAN

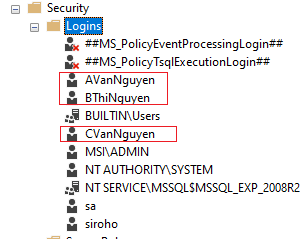
create login BThiNguyen

with password = '123',

default\_database = QUANLYCONGNHAN

create login CVanNguyen

with password = '123',

default\_database = QUANLYCONGNHAN  
  
kết quả:  
  
b. Map database user từ login:  
create user AVanNguyen

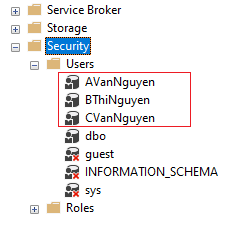
from login AVanNguyen

create user BThiNguyen

from login BThiNguyen

create user CVanNguyen

from login CvanNguyen

kết quả:  
****  
c. Tạo user role:  
create role Totruong

create role Quanli

create role Giamdoc  
  
d. Gán quyền cho role:  
grant select on CONGNHAN to Totruong

grant select on LUONG to Totruong

grant select, insert, update on CONGNHAN to Quanli

grant select, insert, update on LUONG to Quanli

grant select, insert, update on THANNHAN to Quanli

grant all on CONGNHAN to Giamdoc

grant all on LUONG to Giamdoc

grant all on XUONG to Giamdoc

grant all on SANPHAM to Giamdoc

grant all on KHO to Giamdoc

grant all on THANNHAN to Giamdoc  
e. Thêm user vào role:

alter role Totruong add member AVanNguyen

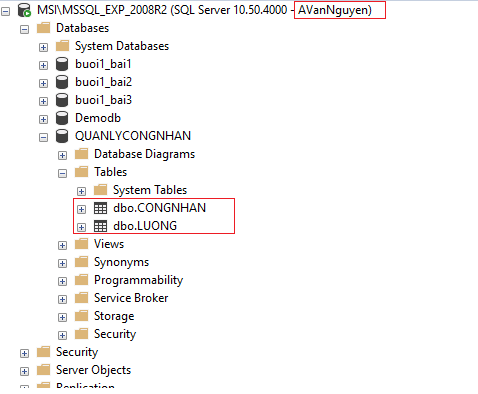
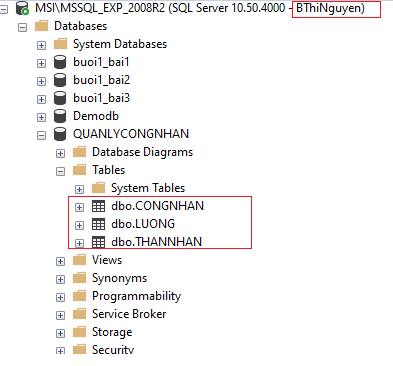
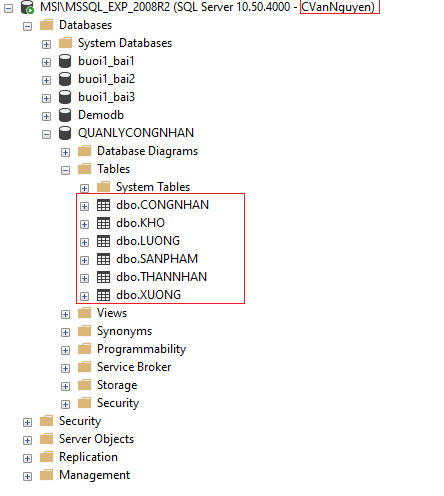
alter role Quanli add member BThiNguyen

alter role Giamdoc add member CVanNguyen

hoặc

EXEC sp\_addrolemember 'Totruong', 'AVanNguyen';

EXEC sp\_addrolemember 'Quanli', 'BThiNguyen';

EXEC sp\_addrolemember 'Giamdoc', 'CVanNguyen';   
  
Kết quả:  
  
  
  
  


# 

# **KẾT LUẬN VÀ KHUYẾN NGHỊ**

## Kết quả đạt được

- Trong suốt khoảng thời gian làm bài tập lớn, chúng em đã học thêm được nhiều kiến thức nền tảng. Thực hiện thành công ứng dụng của C# winform để quản lý cửa hàng bán quần áo. Có thêm kiến thức về cơ sở dữ liệu và ngôn ngữ lập trình C#

- Hoàn thành tất cả các chức năng đề ra so với ban đầu

* Phần mềm thân thiện với người dùng.

## Hiện tại

- Tuy nhiên Phần mềm quản lý còn gặp nhiều thiếu sót trong việc phân tích dữ liệu từ đó dẫn tới cản trở các thao tác trong việc xây dựng ứng dụng.

* Chưa tạo ra được một chương trình cho khách hàng có thể đặt hàng online.

## Hướng phát triển

Nếu có thời gian và điều kiện để tiếp tục thực hiện em sẽ cố gắng để hoàn thành ứng dụng của em một cách thật trọn vẹn. Em sẽ phát triển thêm nhiều tính năng ứng dụng, mở rộng các chức năng của hệ thống phần mềm, đảm bảo độ tin cậy, và chính xác của hệ thống.

Hướng tới việc phát triển mở rộng toàn phần như quản lý các chi nhánh, quản lý nhân sự, lương nhân viên, quản lý thu chi và đặc biệt là quản lý đặt hàng online ...

Mong muốn rằng một ngày nào ứng dụng sẽ ngày một phát triển mở rộng. Và em tin rằng ứng dụng của mình sẽ là một ứng dụng có ích cho xã hội được nhiều người biết đến và tin tưởng khi sử dụng.

## Tài liệu tham khảo

-[1]<http://thuvienso.utc2.edu.vn/doc/bai-giang-co-so-du-lieu-chuong-2-le-nhi-lam-thuy-565484.html> “Bài giảng môn cơ sở dữ liệu trường Đại học Giao Thông Vận Tải phân hiệu tại TPHCM.”

[2] <http://thuvienso.vlute.edu.vn/doc/bai-giang-ve-ngon-ngu-c-269423.html> “ Bài giảng ngôn ngữ lập trình C# trường đại học Sư Phạm Kỹ Thuật Vĩnh

[3] <https://www.howkteam.vn/course/lap-trinh-winform-co-ban-27>, “Lập trình C# winform cơ bản”

# [4] <https://timoday.edu.vn/lap-trinh-co-so-du-lieu-voi-c/> “Lập trình Cơ sở dữ liệu với C#

”