TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

Text, company name

Description automatically generated

**BÁO CÁO BÀI TẬP NHÓM**

**TÊN ĐỀ TÀI**

**QUẢN LÝ TRÀ CHANH T-MORE**

**Nhóm 8:**

**Sinh viên thực hiện:**

1. Trưởng nhóm: Trần Hoàng Liêm Lớp : 45K14
2. Thành viên: Trần Quốc Hoàn Lớp : 45K14
3. Thành viên: Nguyễn Thị Yến Phượng Lớp : 45K14
4. Thành viên: Vũ Thị Việt Trinh Lớp : 45K14

**Giảng viên hướng dẫn:** Cao Thị Nhâm

**Đà Nẵng, 10/2021**

**DANH SÁCH THÀNH VIÊN THAM GIA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Họ và tên** | **Lớp** | **Nhiệm vụ** | **Đánh giá tham gia (100%)** |
| 1 | Trần Hoàng Liêm | 45K14 | Trưởng nhóm | 25% |
| 2 | Trần Quốc Hoàn | 45K14 | Thành viên | 25% |
| 3 | Nguyễn Thị Yến Phượng | 45K14 | Thành viên | 25% |
| 4 | Vũ Thị Việt Trinh | 45K14 | Thành viên | 25% |

# MỤC LỤC

[MỤC LỤC ii](#_Toc89376006)

**DANH MỤC HÌNH ẢNH....................................................................................iii**

[LỜI MỞ ĐẦU 1](#_Toc89376007)

[CHƯƠNG 1. TỔNG QUAN quán trà chanh T-More 2](#_Toc89376008)

[1.1. Giới thiệu về quán trà chanh T-more 2](#_Toc89376009)

[1.2. Mô tả hệ thống 2](#_Toc89376010)

[1.3. Chức năng chính của hệ thống 3](#_Toc89376011)

[CHƯƠNG 2. YÊU CẦU CƠ BẢN 4](#_Toc89376012)

[2.1. Sưu tầm hóa đơn của quán 4](#_Toc89376013)

[2.2. Thiết kế cơ sở dữ liệu 5](#_Toc89376014)

[2.2.1. Thiết kế cơ sở dữ liệu ở mức khái niệm ( Xây dựng sơ đồ thực thể liên kết ERD ) 5](#_Toc89376015)

[2.2.2. Thiết kế cơ sở dữ liệu ở mức logic 9](#_Toc89376016)

[2.2.3. Thiết kế cơ sở dữ liệu ở mức vật lý 10](#_Toc89376017)

[2.3. Xây dựng cơ sở dữ liệu 13](#_Toc89376018)

[2.4. Xác định và tạo các index cho các thuộc tính cần thiết 16](#_Toc89376019)

[2.5. Xây dựng cơ chế bảo mật tương ứng cho CSDL 17](#_Toc89376020)

[2.5.1. Authentication 17](#_Toc89376021)

[2.5.2. Authority 18](#_Toc89376022)

[2.6. Xây dựng cơ chế backup dữ liệu phù hợp cho CSDL 21](#_Toc89376023)

[2.6.1. Thực hiện backup bằng tay 21](#_Toc89376024)

[2.6.2. Thực hiện backup tự động: 22](#_Toc89376025)

[2.7. Phát triển ứng dụng cho CSDL 24](#_Toc89376026)

[2.7.1. Form Login 24](#_Toc89376027)

[2.7.2. Form Main 26](#_Toc89376028)

[2.7.3. Form quản lý bán hàng 29](#_Toc89376029)

[2.7.4. Form quản lý nhập hàng 39](#_Toc89376030)

[2.7.5. Form quản lý 54](#_Toc89376031)

[CHƯƠNG 3. YÊU CẦU NÂNG CAO 84](#_Toc89376032)

[3.1. Phương án tốt nhất để đề phòng hệ thống bị tấn công bằng SQL Injection: 84](#_Toc89376033)

[CHƯƠNG 4. KẾT Luận 85](#_Toc89376034)

DANH MỤC HÌNH ẢNH

[*Hình 1. Quán trà chanh T-more* 2](#_Toc89374961)

[*Hình 2. Hóa đơn bán hàng* 4](#_Toc89374962)

[*Hình 3. Hóa đơn nhập hàng* 4](#_Toc89374963)

[*Hình 4. Mô hình ERD cho hóa đơn bán hàng* 6](#_Toc89374964)

[*Hình 5. Mô hình ERD cho hóa đơn nhập hàng* 8](#_Toc89374965)

[*Hình 6. Tích hợp hai mô hình ERD của hóa đơn nhập và bán hàng* 9](#_Toc89374966)

[*Hình 7. Tạo index cho cột MaTP* 16](#_Toc89374967)

[*Hình 8. Tạo tài khoản QUANLYBAN* 17](#_Toc89374968)

[*Hình 9. Tạo tài khoản NHANVIENBAN* 18](#_Toc89374969)

[*Hình 10. Phân quyền cho Quản lý* 18](#_Toc89374970)

[*Hình 11. Phân quyền cho quản lý bán với các bảng CSDL* 19](#_Toc89374971)

[*Hình 12. Phân quyền cho nhân viên* 20](#_Toc89374972)

[*Hình 13. Phân quyền cho nhân viên bán với các bảng CSDL* 20](#_Toc89374973)

[*Hình 14. Tạo file backup cho database Quản lý trà chanh* 21](#_Toc89374974)

[*Hình 15. Nơi lưu thư mục backup* 21](#_Toc89374975)

[*Hình 16. Create a basic task* 22](#_Toc89374976)

[*Hình 17. Task Trigger* 23](#_Toc89374977)

[*Hình 18. Thời gian thực hiện việc backup hàng ngày* 23](#_Toc89374978)

[*Hình 19. Giao diện form đăng nhập* 24](#_Toc89374979)

[*Hình 20. Giao diện form main* 26](#_Toc89374980)

[*Hình 21. Giao diện form quản lý bán hàng* 29](#_Toc89374981)

[*Hình 22. Giao diện form danh sách nhà cung cấp* 39](#_Toc89374982)

[*Hình 23. Giao diện form quản lý Nguyên liệu* 46](#_Toc89374983)

[*Hình 24. Giao diện form quản lý Nhân viên* 55](#_Toc89374984)

[*Hình 25. Giao diện form quản lý Khách hàng* 62](#_Toc89374985)

[*Hình 26. Giao diện form quản lý hàng hóa* 69](#_Toc89374986)

[*Hình 27. Giao diện quản lý hóa đơn thanh toán* 77](#_Toc89374987)

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

Đề tài này nghiên cứu xây dựng ứng dụng quản lý quán trà chanh T-More nhằm hỗ trợ công tác quản lý trong việc bán hàng. Ứng dụng có các chức năng cơ bản như: hiển thị thông tin, thống kê, thêm mới, sửa, xóa, tìm kiếm thông tin…

1. **Nhiệm vụ của đề tài**

Nghiên cứu xây dựng ứng dụng quản lý thông tin bán hàng và nhập hàng của quán trà chanh T-more.

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, 4 chương nội dung và phần kết luận

* Mở đầu
* **Chương 1**: Tổng quan quán trà chanh T-More
* **Chương 2**: Yêu cầu cơ bản
* **Chương 3**: Yêu cầu nâng cao
* **Chương 4**: Kết luận
* Tài liệu tham khảo

# TỔNG QUAN quán trà chanh T-More

## Giới thiệu về quán trà chanh T-more



*Hình 1. Quán trà chanh T-more*

Tmore đã là nhà dẫn đầu trong phong trào nâng tầm ẩm thực đường phố, đưa trà chanh vỉa hè trở thành cơn sốt đối với giới trẻ, tạo nên một "nét văn hóa" rất đặc trưng.

Hiện tại Tmore đã có 90 cửa hàng trong Hệ thống toàn quốc.

Với sự đồng hành từ nhiều phía đối tác cùng những ý tưởng táo bạo, chúng tôi hi vọng Tiệm trà chanh Tmore sẽ luôn luôn mang một màu sắc mới mẻ, gần gũi mà không bị pha lẫn với bất kì một cái tên nào khác

## Mô tả hệ thống

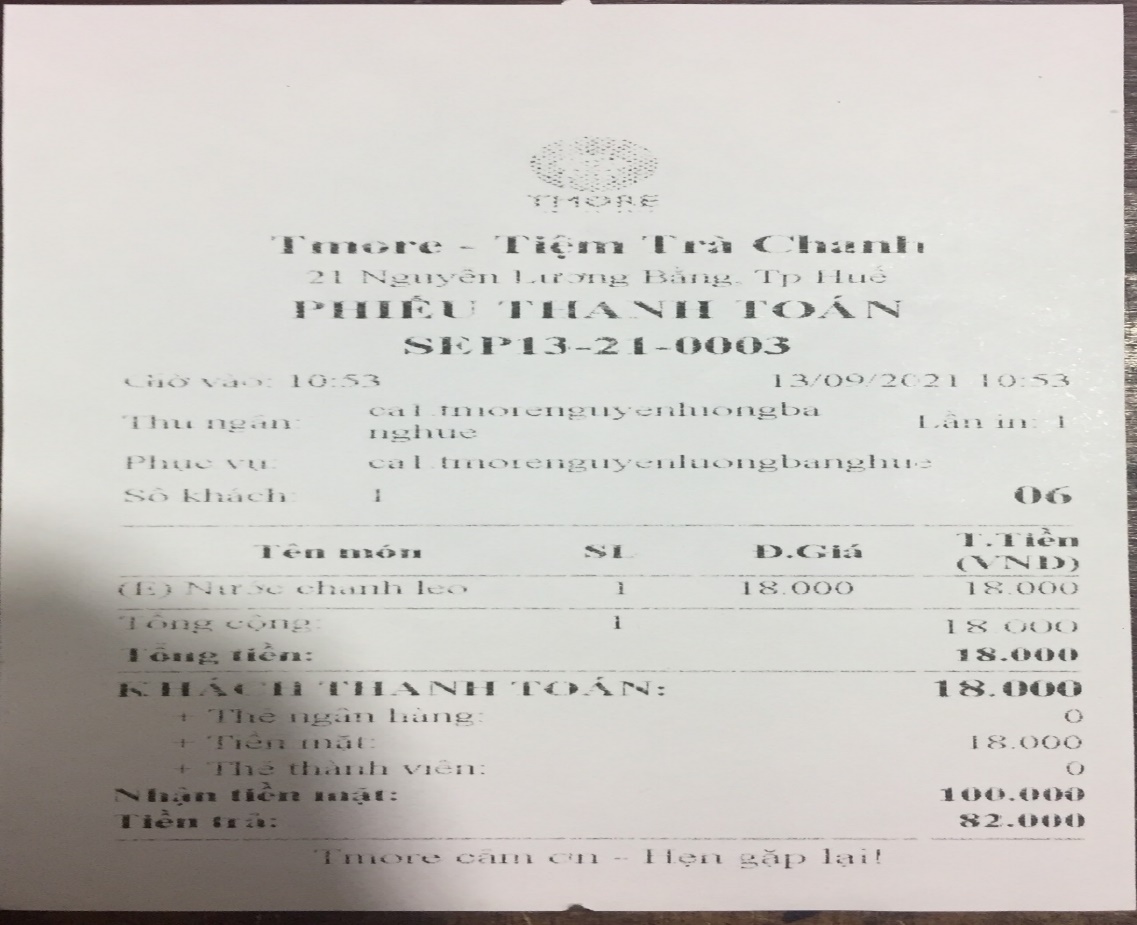
* Hệ thống quản lý quán trà chanh T-more giúp nhà quản lý có thể dễ dàng theo dõi và kiểm soát tốt hơn trong việc bán hàng cũng như nhập hàng, tối ưu việc quản lý các sản phẩm trong từng loại hàng.
* Tên hệ thống: Phần mềm quản lý quán trà chanh T-more

## Chức năng chính của hệ thống

* Chức năng của phần mềm quản lý quán trà chanh cực kỳ đa dạng. Dưới đây là một số tính năng nổi bật và phổ biến nhất:
* Đăng nhập
* Thanh toán
* Chỉnh sửa, bổ sung hóa đơn, in hóa đơn
* Báo cáo hàng ngày
* Thêm, sửa, xóa và tìm kiếm thông tin
* Môi trường triển khai hệ thống bao gồm:
* Visual studio 2013
* Microsoft SQL Server
* Ngôn ngữ lập trình C#
* Lập trình Windows Form

# YÊU CẦU CƠ BẢN

## Sưu tầm hóa đơn của quán

****

*Hình 2. Hóa đơn bán hàng*

A piece of paper with writing

Description automatically generated with medium confidence

*Hình 3. Hóa đơn nhập hàng*

## Thiết kế cơ sở dữ liệu

### Thiết kế cơ sở dữ liệu ở mức khái niệm ( Xây dựng sơ đồ thực thể liên kết ERD )

**Xây dựng ER cho hóa đơn bán hàng:**

**Bước 1: Chọn lọc thông tin**

|  |  |  |
| --- | --- | --- |
| XXHSDL | Từ rõ nghĩa | Viết tắt |
| Số | Mã hóa đơn bán hàng | MaHDB |
| Thời gian | Thời gian khách thanh toán | ThoiGian |
| Ngày bán | Ngày bán | NgayBan |
| Tên nhân viên | Tên nhân viên | TenNV |
| Số khách | Số lượng khách hàng mua | SoKhach |
| Tên hàng | Tên thành phẩm | TenTP |
| SL | Số lượng món | SoLuong |
| Đ.Giá | Đơn giá món | DonGia |
| Thành tiền | Thành tiền | ThanhTien |
| Tổng tiền | Tổng tiền thu | TongTien |
| Khách thanh toán | Tổng tiền thanh toán của khách | TongTT |
| Đơn vị | Đơn vị tính tiền | DonVi |
| Mã khách hàng | Mã khách hàng | MaKH |
| Tên khách hàng | Tên khách hàng | TenKH |
| Mã nhân viên | Mã nhân viên | MaNV |
| Mã hàng | Mã thành phẩm | MaTP |
| Số điện thoại nhân viên | Số điện thoại nhân viên | SoDienThoaiNV |
| Số điện thoại khách hàng | Số điện thoại khách hàng | SoDienThoaiKH |
| Nhận tiền mặt | Tiền khách hàng đưa | NhanTienMat |
| Tiền trả | Tiền trả lại cho khách | TienTra |

**Bước 2: Xác định thực thể, thuộc tính**

**KHÁCH HÀNG (MaKH, TenKH, SoDienThoaiKH )**

**NHÂN VIÊN (MaNV, TenNV, SoDienThoaiNV )**

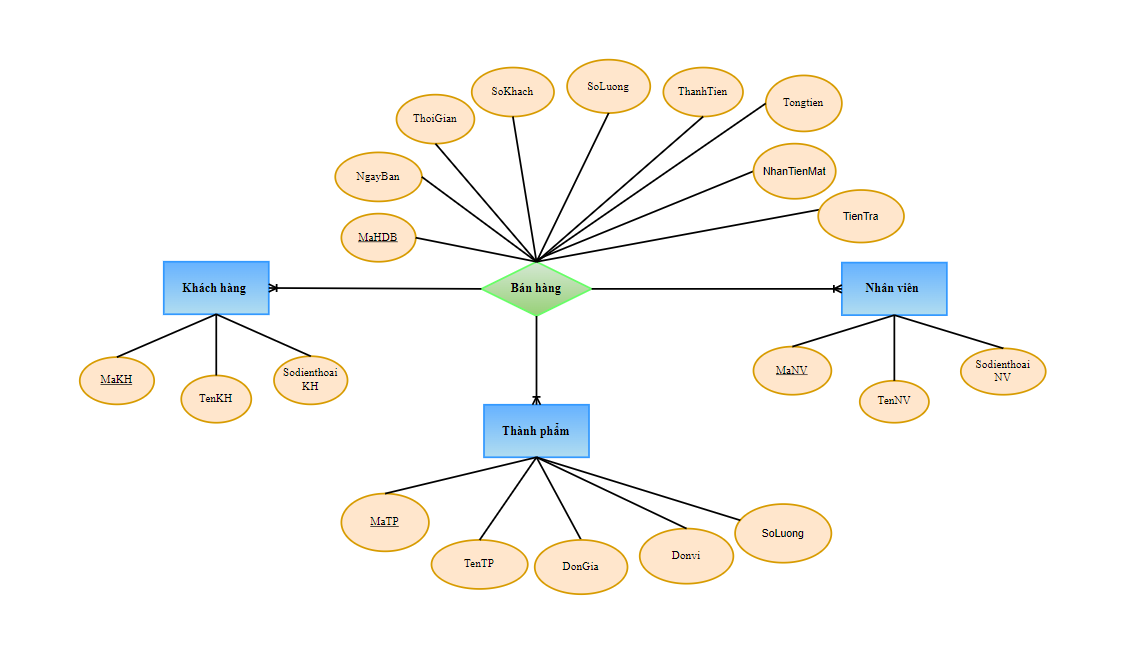
**THÀNH PHẨM (MaTP, TenTP, DonVi, DonGia )**

***Ghi chú:*** MaKH, TenKH, MaNV, MaTP, SoDienThoaiNV, SoDienThoaiKH được bổ sung thêm để dễ dàng quản lí.

**Bước 3: Xác định quan hệ**

**BÁN: MaHDB, SoKhach, SoLuong, NgayBan, ThoiGian,** **ThanhTien, TongTien, NhanTienMat, TienTra**

**Bước 4: Vẽ**



*Hình 4. Mô hình ERD cho hóa đơn bán hàng*

**Xây dựng ER cho phiếu nhập kho**

**Bước 1: Chọn lọc thông tin**

|  |  |  |
| --- | --- | --- |
| HSDL | Từ rõ nghĩa | Viết tắt |
| Số | Mã Hóa Đơn Nhập Kho | MaHDN |
| Mã người bán | Mã người bán | MaNB |
| Người bán | Người bán | TenNB |
| Địa chỉ | Địa chỉ người bán | DiaChiNB |
| Số điện thoại | Số điện thoại người bán | SoDienThoaiNB |
| Ngày nhập | Ngày nhập kho | NgayNhap |
| Mã nhân viên | Mã nhân viên nhận hàng | MaNV |
| Tên nhân viên | Tên nhân viên nhận hàng | TenNV |
| Số điện thoại | Số điện thoại nhân viên nhận hàng | SoDienThoaiNV |
| Mã hàng | Mã hàng hóa | MaH |
| Tên Hàng | Tên hàng hóa | TenH |
| Đơn vị | Đơn vị tính | DonVi |
| Số lượng | Số lượng hàng hóa | SoLuong |
| Đơn giá | Đơn giá | DonGia |
| Thành Tiền | Thành tiền | ThanhTien |
| Cộng tiền hàng | Tổng tiền hàng | TongTien |
| Tiền thuế GTGT | Tiền thuế GTGT | ThueGTGT |
| Tổng tiền thanh toán | Tổng tiền thanh toán | TongTT |

**Bước 2: Xác định thực thể, thuộc tính**

**NGƯỜI BÁN (MaNB, TenNB, DiaChiNB, SoDienThoaiNB)**

**NHÂN VIÊN (MaNV, TenNV, SoDienThoaiNV)**

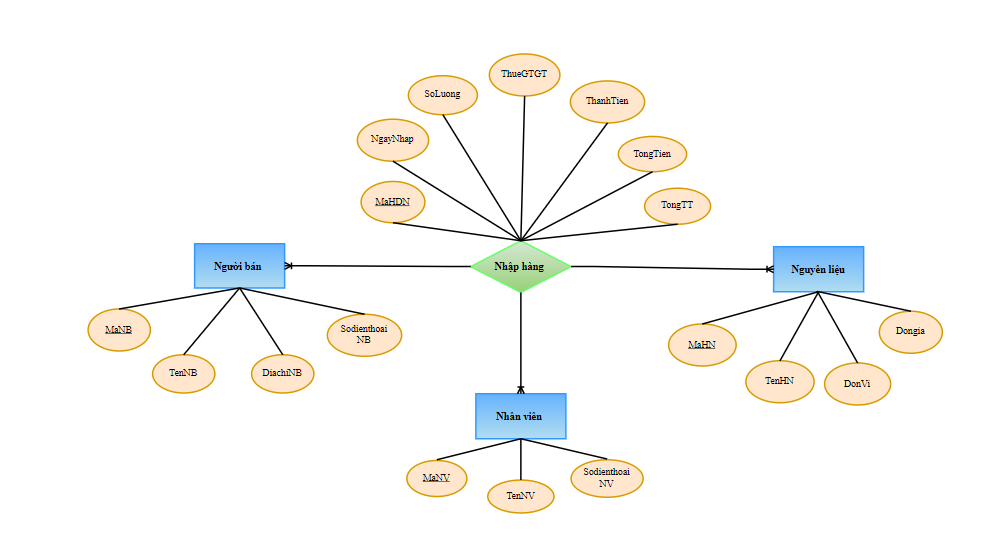
**HÀNG (MaH, TenH, DonVi, DonGia)**

***Ghi chú****:* MaNV, MaNB, TenNV, SoDienThoaiNB, SoDienThoaiNV được bổ sung thêm để dễ dàng quản lí.

**Bước 3: Xác định quan hệ**

**NHẬP: MaHDN, SoLuong, NgayNhap,** **ThanhTien, TongTien, ThueGTGT, TongTT**

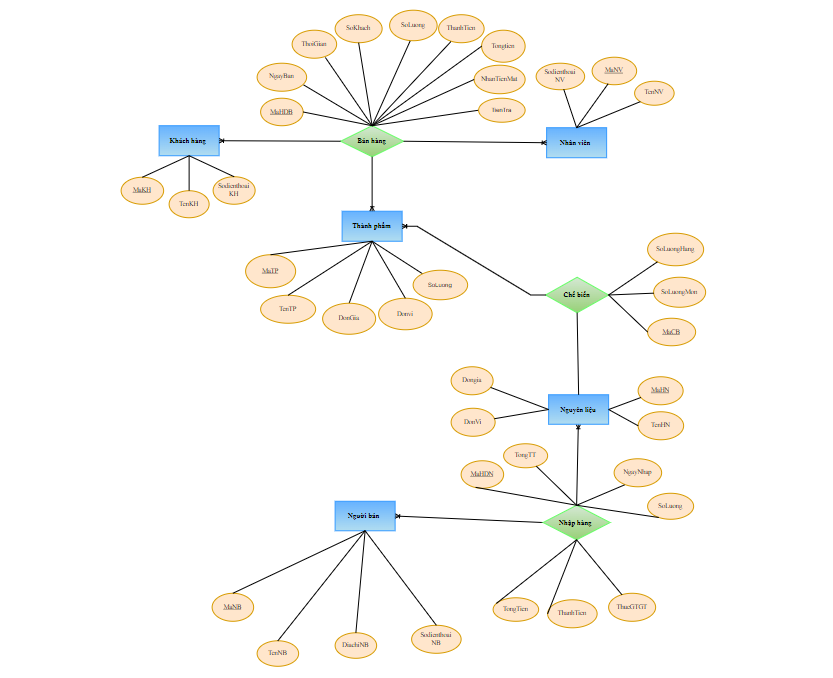
**Bước 4: Vẽ**

****

*Hình 5. Mô hình ERD cho hóa đơn nhập hàng*

**Tích hợp ER hóa đơn bán hàng và ER hóa đơn nhập hàng:**

**Thêm quan hệ CHEBIEN (MaCB, SoLuongMon, SoLuongHang) để đảm bảo tính toàn vẹn của dữ liệu**

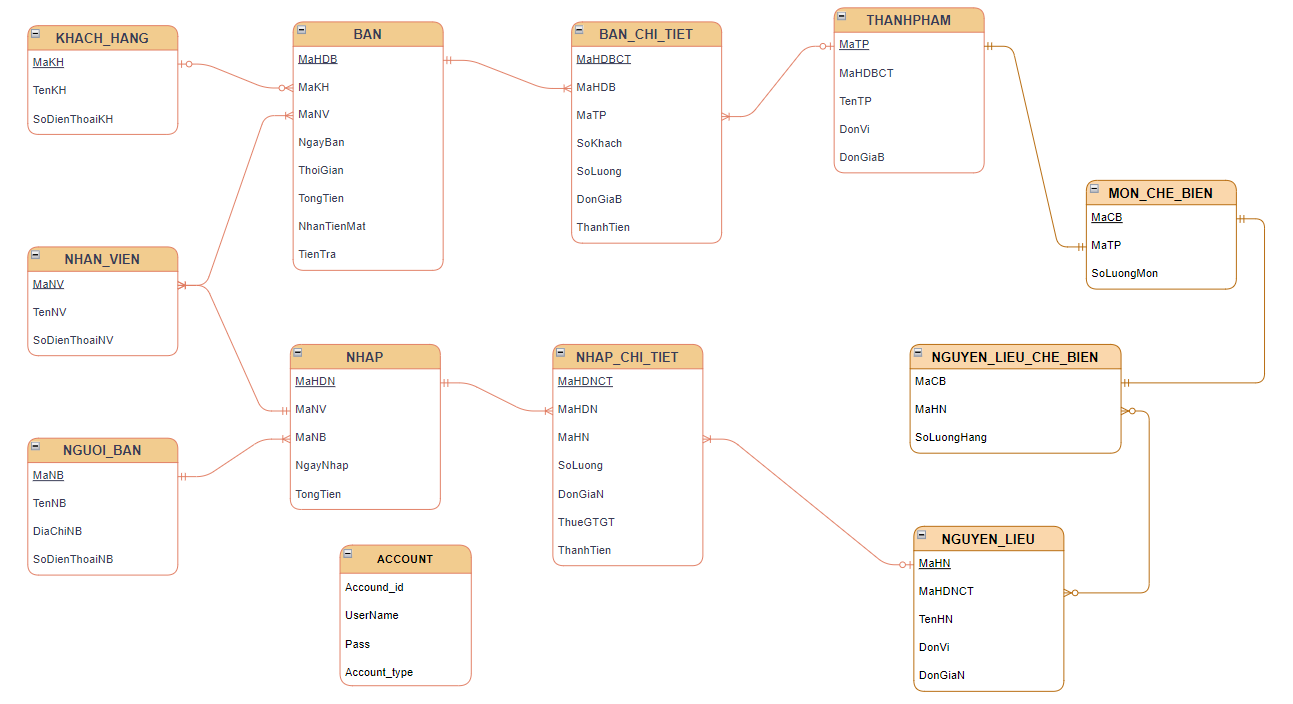
****

*Hình 6. Tích hợp hai mô hình ERD của hóa đơn nhập và bán hàng*

### Thiết kế cơ sở dữ liệu ở mức logic

* **Bước 1: Chuyển thực thể, thu được:**
* **THÀNH PHẨM (MaTP, MaHDBCT, TenTP, DonVi, DonGiaN)**
* **NHÂN VIÊN (MaNV, TenNV, SoDienThoaiNV)**
* **KHÁCH HÀNG (MaKH, TenKH, SoDienThoaiKH)**
* **NGƯỜI BÁN (MaNB, TenNB, DiaChiNB, SoDienThoaiNB)**
* **NGUYÊN LIỆU (MaHN, MaHDNCT, TenHN, DonVi, DonGiaN)**
* **Bước 2: Chuyển quan hệ, thu được:**
* **BÁN (MaHDB, SoKhach, SoLuong, NgayBan, ThoiGian, DonGiaB, ThanhTien, TongTien, MaTP, MaNV, MaKH)**
* **NHẬP (MaHDN, SoLuong, NgayNhap, DonGiaN,** **ThanhTien, TongTien, ThueGTGT, MaNV, MaNB, MaHN)**
* **CHEBIEN(MaCB, SoLuongMon, SoLuongHang, MaTP, MaHN)**
* **Bước 3: Chuẩn hóa**
* **Chuẩn hóa quan hệ ĐẶT, thu được:**
* **BÁN (MaHDB, MaNV, MaKH, NgayBan, ThoiGian, TongTien)**
* **BÁN CHI TIẾT (MaHDBCT, MaHDB, MaTP, SoKhach, SoLuong, DonGiaB, ThanhTien)**
* **Chuẩn hóa quan hệ NHẬP, thu được:**
* **NHẬP (MaHDN, MaNV,** **MaNB, NgayNhap, TongTien)**
* **NHẬP CHI TIẾT (MaHDNCT, MaHDN, MaHN, SoLuong, DonGiaN, ThueGTGT, ThanhTien)**
* **Chuẩn hoá quan hệ CHEBIEN thu được:**
* **MÓN CHẾ BIẾN (MaCB, MaTP, SoLuongMon)**
* **NGUYÊN LIỆU CHẾ BIẾN (MaCB, MaHN, SoLuongHang)**

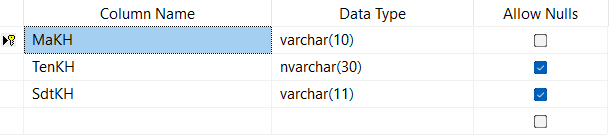
**Vẽ sơ đồ quan hệ**

****

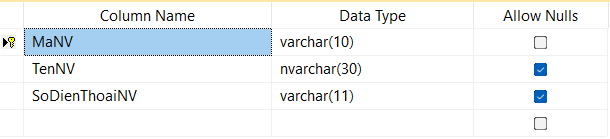
*Hình 7: Sơ đồ quan hệ*

### Thiết kế cơ sở dữ liệu ở mức vật lý

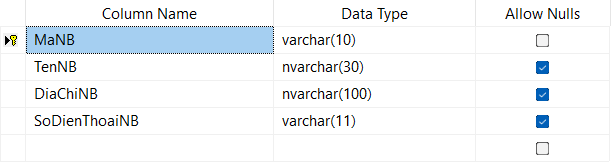
* Bảng Khách Hàng:



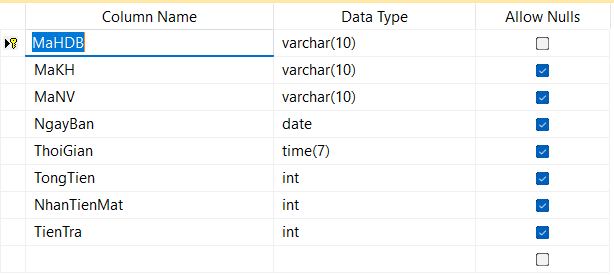
* Bảng Nhân Viên:



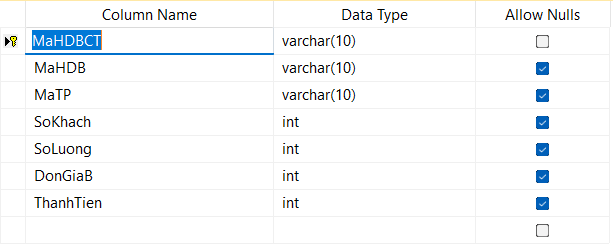
* Bảng Người Bán:



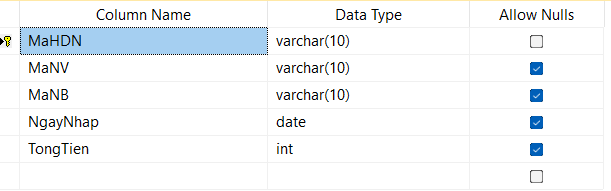
* Bảng Bán:



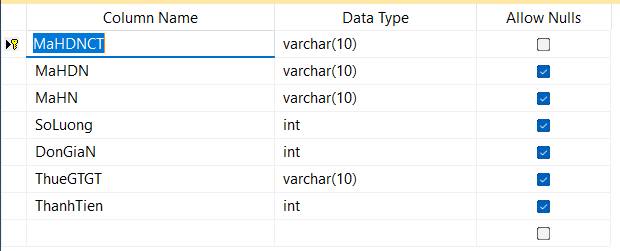
* Bảng Bán Chi Tiết:



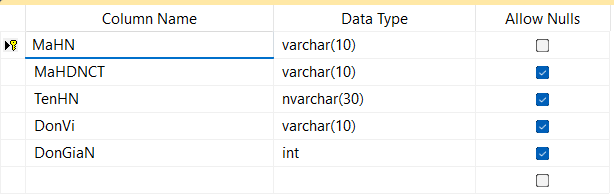
* Bảng Nhập:



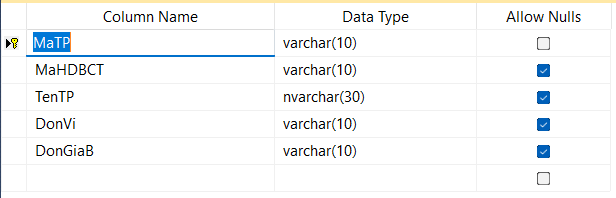
* Bảng Nhập Chi Tiết:



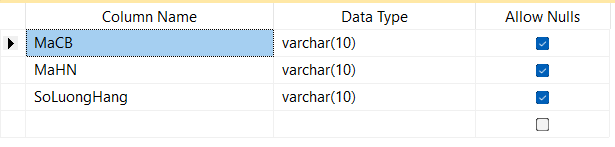
* Bảng Nguyên Liệu:



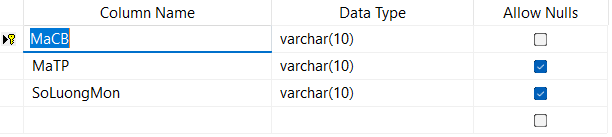
* Bảng Thành Phẩm:



* Bảng Nguyên Liệu Chế Biến:

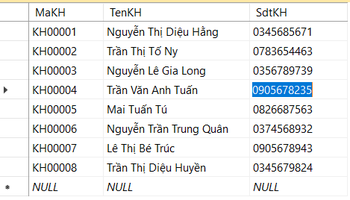


* Bảng Món Chế Biến:

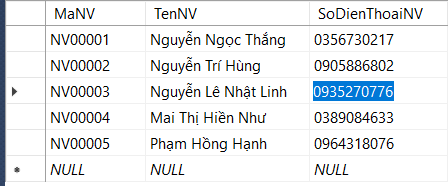


## Xây dựng cơ sở dữ liệu

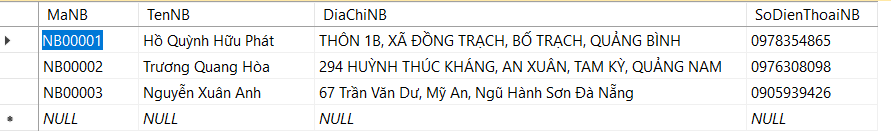
* Bảng Khách Hàng:



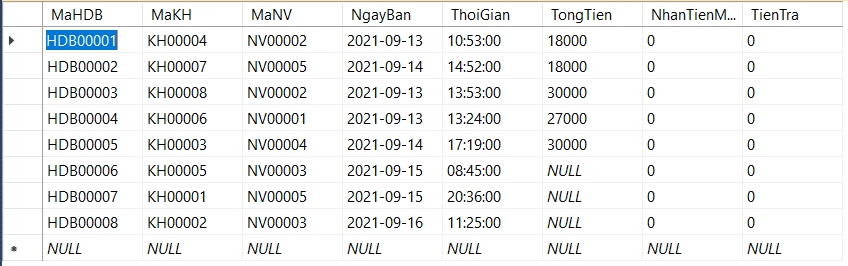
* Bảng Nhân Viên:



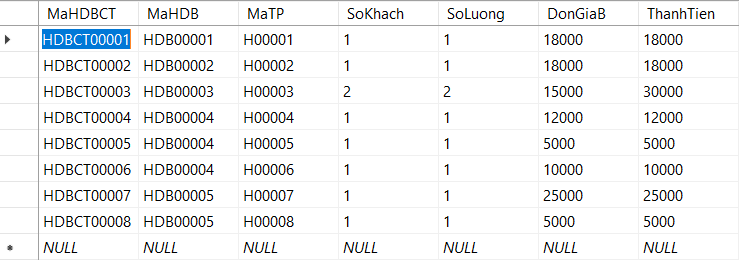
* Bảng Người Bán:



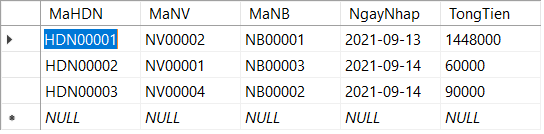
* Bảng Bán:



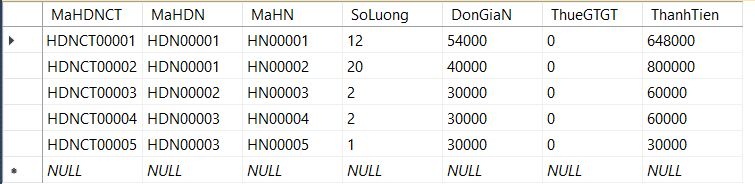
* Bảng Bán Chi Tiết:



* Bảng Nhập:



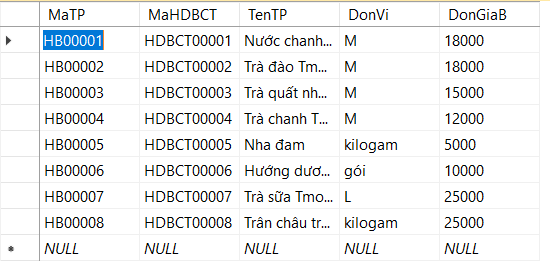
* Bảng Nhập Chi Tiết:



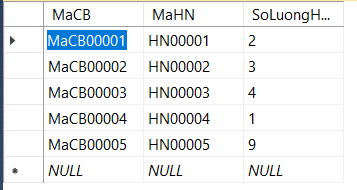
* Bảng Nguyên Liệu:



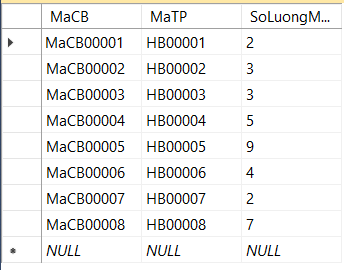
* Bảng Thành Phẩm:



* Bảng Nguyên Liệu Chế Biến:

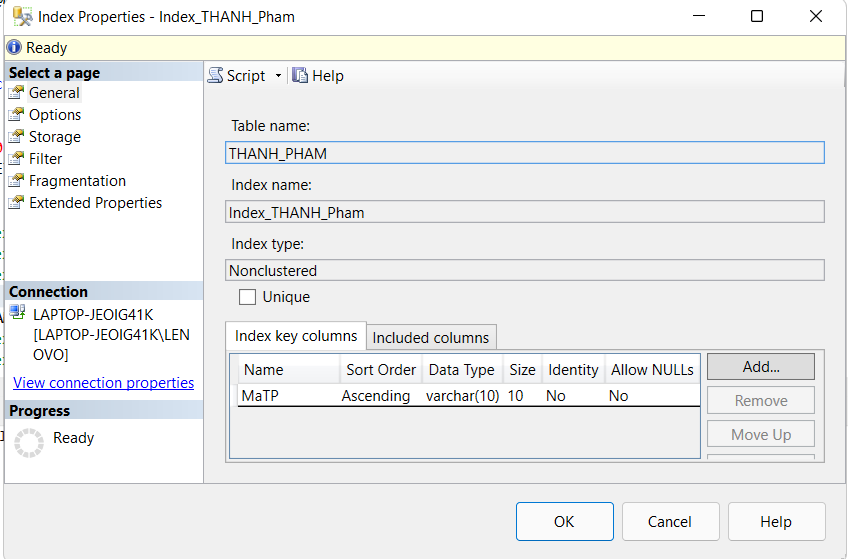


* Bảng Món Chế Biến:



## Xác định và tạo các index cho các thuộc tính cần thiết

Tạo index cho cột MaTP trong bảng THANHPHAM



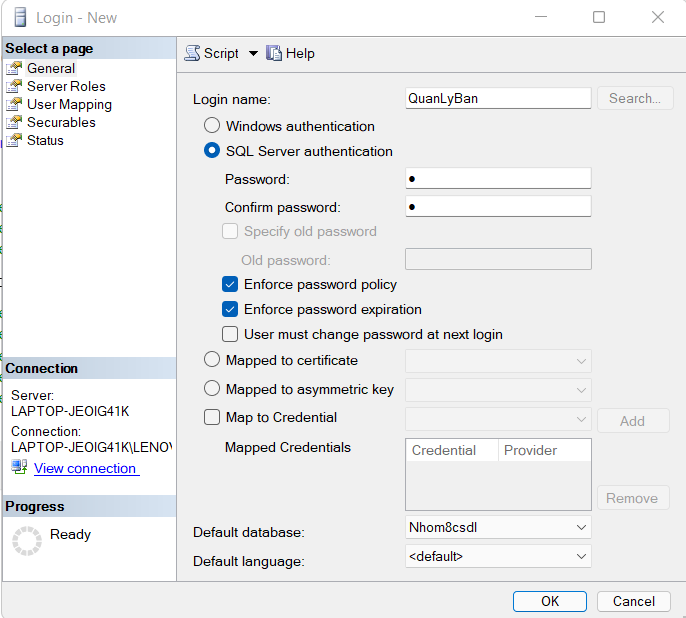
*Hình 7. Tạo index cho cột MaTP*

Nhóm chọn MaTP làm non-clustered index vì trong khi order, thường xuyên truy xuất mã món, tạo index cho MaTP giúp các thao tác diễn ra nhanh hơn.

## Xây dựng cơ chế bảo mật tương ứng cho CSDL

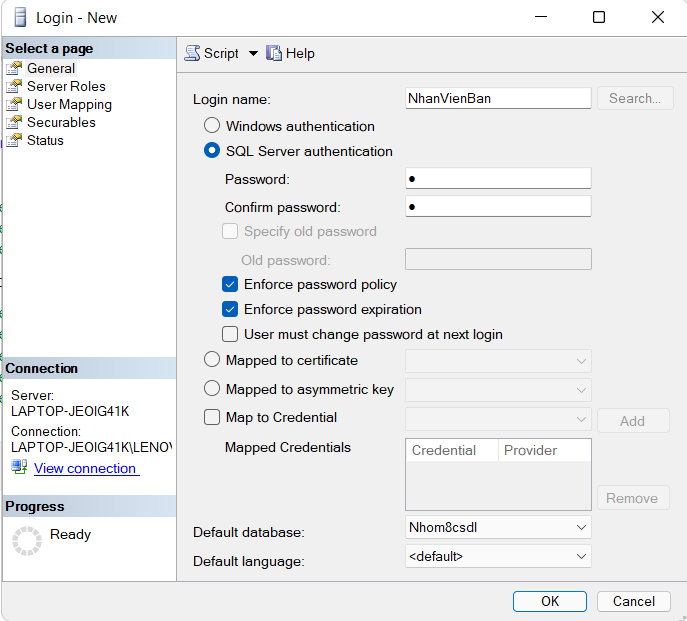
### Authentication

Tạo tài khoản QUANLYBAN với Database mặc định là Nhom8csdl



*Hình 8. Tạo tài khoản QUANLYBAN*

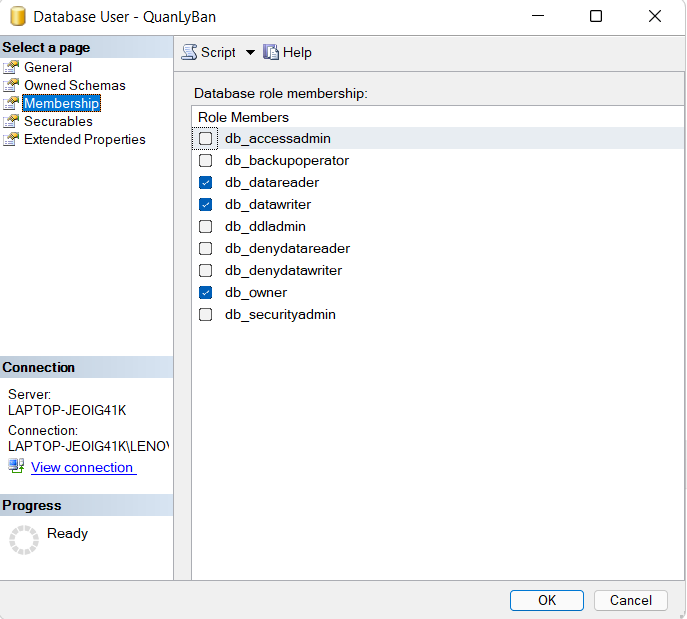
Tạo tài khoản NHANVIENBAN với database mặc định là Nhom8csdl



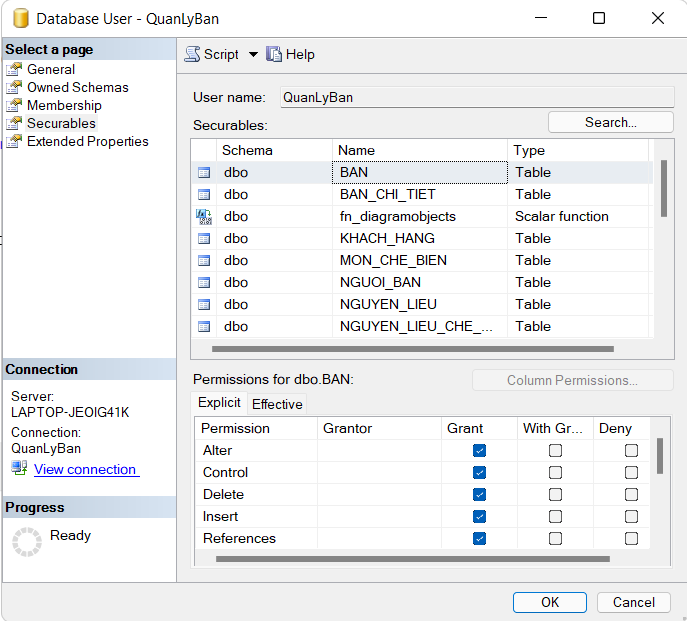
*Hình 9. Tạo tài khoản NHANVIENBAN*

### Authority

* Quản lý là người sở hữu và có mọi quyền sử dụng database Nhom8csdl



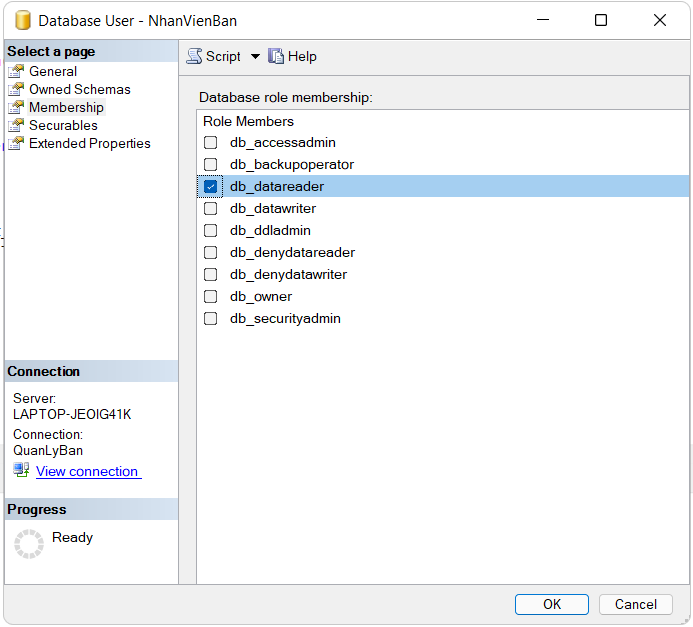
*Hình 10. Phân quyền cho Quản lý*



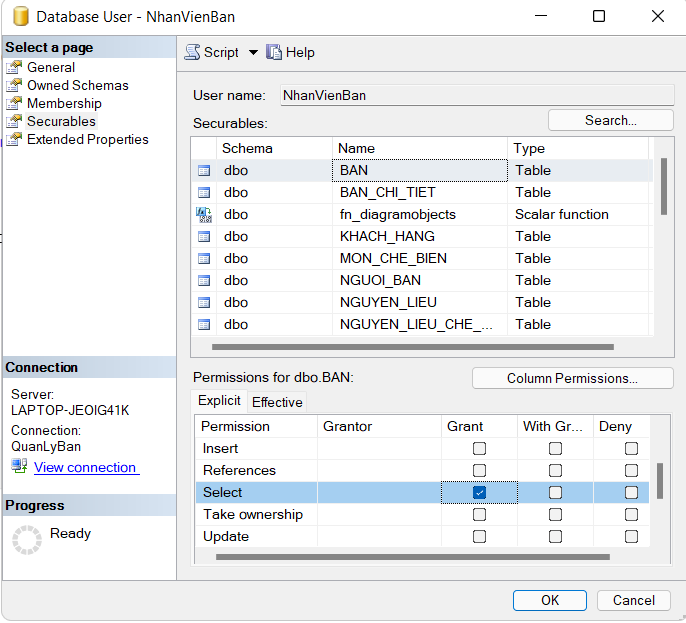
*Hình 11. Phân quyền cho quản lý bán với các bảng CSDL*

Tương tự với các bảng còn lại, quản lý cũng có quyền sở hữu và sử dụng trong database

* Phân quyền cho nhân viên chỉ có quyền xem



*Hình 12. Phân quyền cho nhân viên*



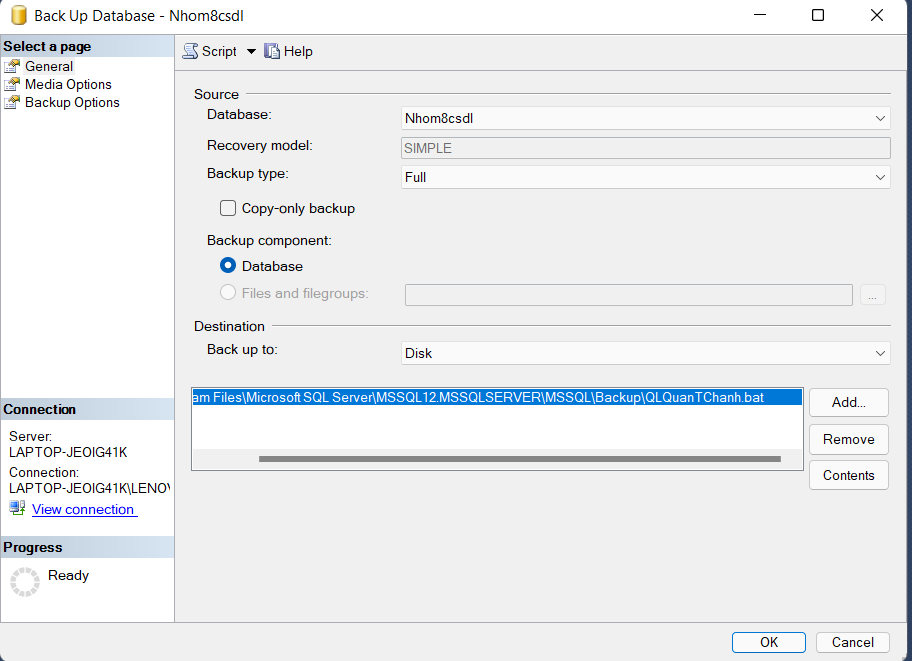
*Hình 13. Phân quyền cho nhân viên bán với các bảng CSDL*

Tương tự như vậy đối với các bảng còn lại.

## Xây dựng cơ chế backup dữ liệu phù hợp cho CSDL

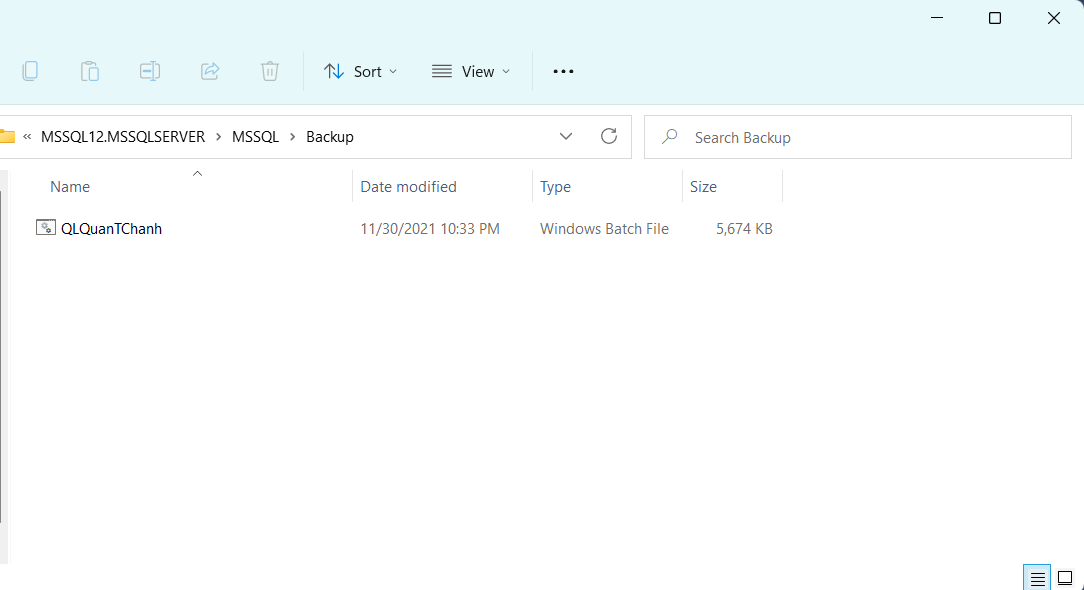
### Thực hiện backup bằng tay

Tạo file backup cho database Nhom8csdl



*Hình 14. Tạo file backup cho database Quản lý trà chanh*

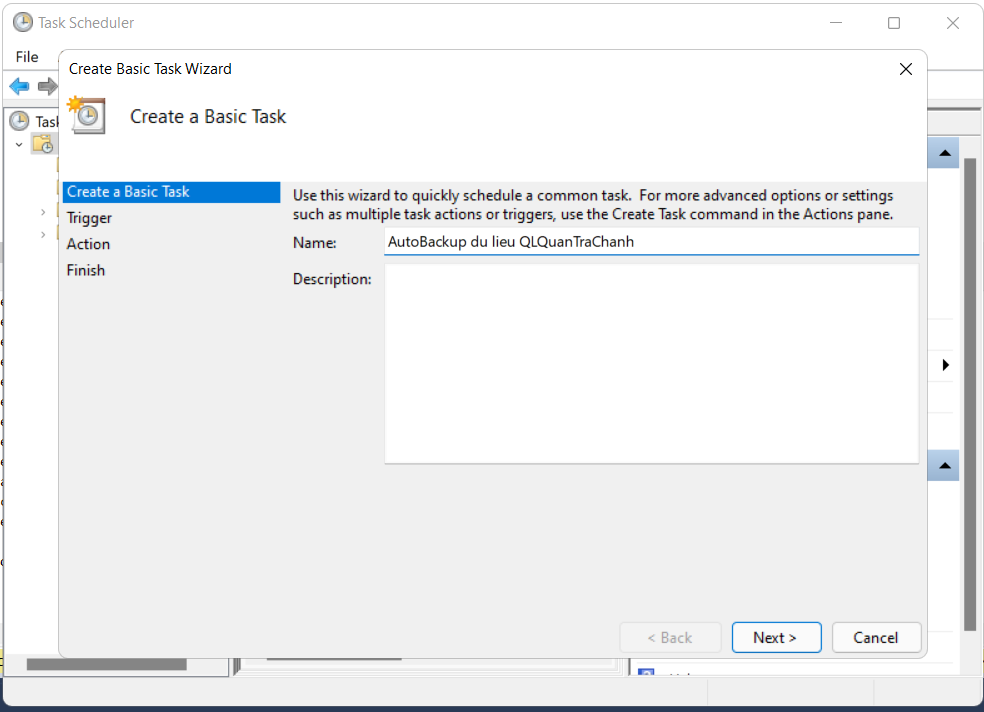
Thư mục lưu file backup



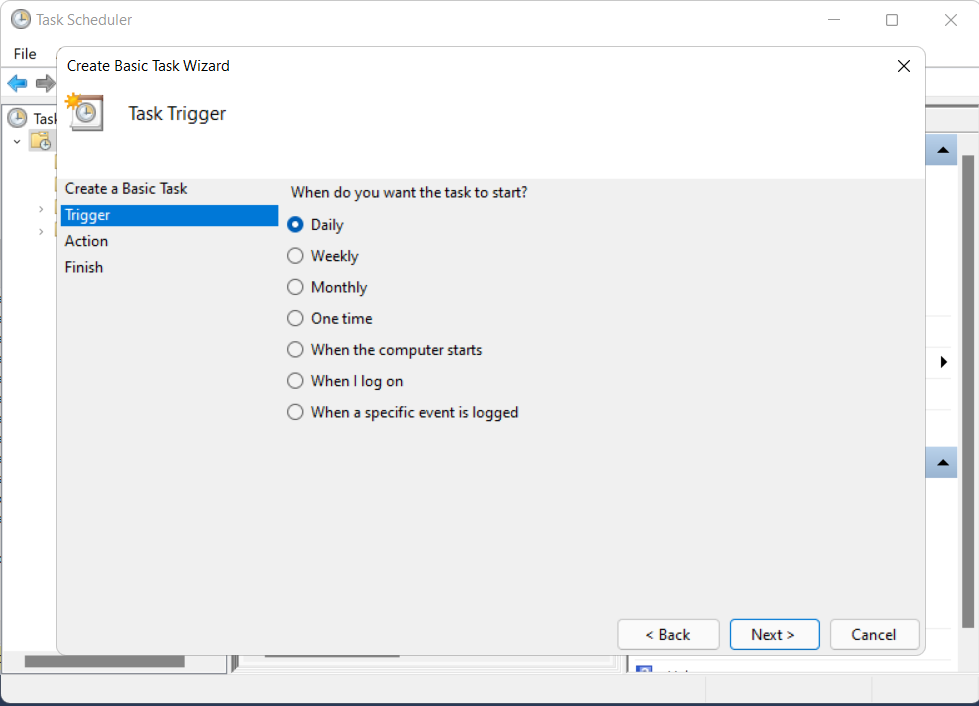
*Hình 15. Nơi lưu thư mục backup*

### Thực hiện backup tự động:

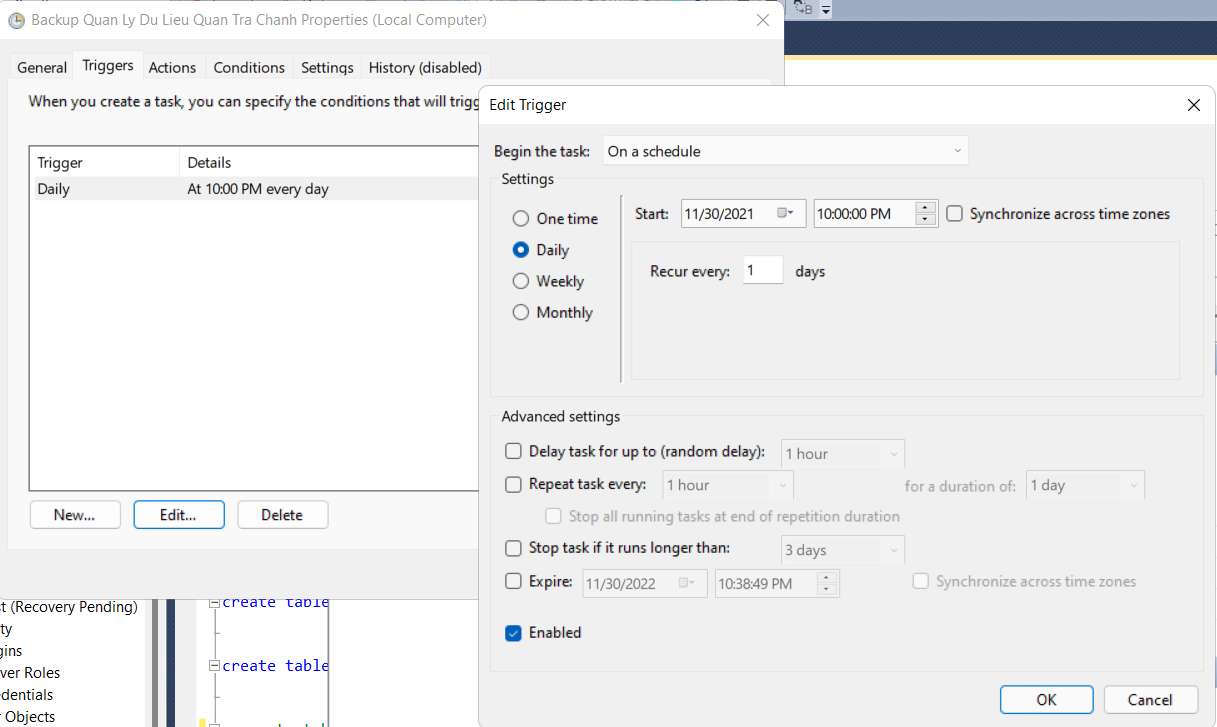
Thường xuyên sao lưu dữ liệu sẽ giúp bạn tránh được tình trạng mất mát dữ liệu do những tác động không mong muốn. Do phương án backup bằng tay có một số nhược điểm, trong đó phải kể đến là phải cần người backup nên nhóm đã triển khai phương án backup tự động và thiết lập thời gian mỗi ngày để tự động backup nhằm tiết kiệm sức người và thời gian.



*Hình 16. Create a basic task*



*Hình 17. Task Trigger*



*Hình 18. Thời gian thực hiện việc backup hàng ngày*

## Phát triển ứng dụng cho CSDL

### Form Login



*Hình 19. Giao diện form đăng nhập*

* Các thành phần của form:

+ Textbox:  Tên đăng nhập, Mật khẩu.

+ Button: Đăng nhập, Thoát

* Chức năng:

+ Nút Đăng nhập: Kiểm tra thông tin tài khoản và mật khẩu. Nếu tài khoản và mật khấu đúng, sẽ nhận được thông báo: “Đăng nhập thành công.”. Nếu sai, thông báo “Đăng nhập thất bại.”, còn lại báo “Lỗi kết nối”.

+ Nút Thoát: Thoát khỏi form đăng nhập.

* Code Form Đăng nhập:

private void btnDangNhap\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-SMKPHOFO;Initial Catalog=Nhom8csdl;Integrated Security=True");

try

{

conn.Open();

string tk = txtUsername.Text;

string mk = txtPassword.Text;

string sql = "select\*from ACCOUNT where UserName='" + tk + "'and Pass='" + mk + "'";

SqlCommand cmd = new SqlCommand(sql, conn);

SqlDataReader dta = cmd.ExecuteReader();

if (dta.Read() == true)

{

MessageBox.Show("Đăng nhập thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

frmmain f = new frmmain();

this.Hide();

f.ShowDialog();

this.Show();

}

else

{

MessageBox.Show("Đăng nhập thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi kết nối");

}

}

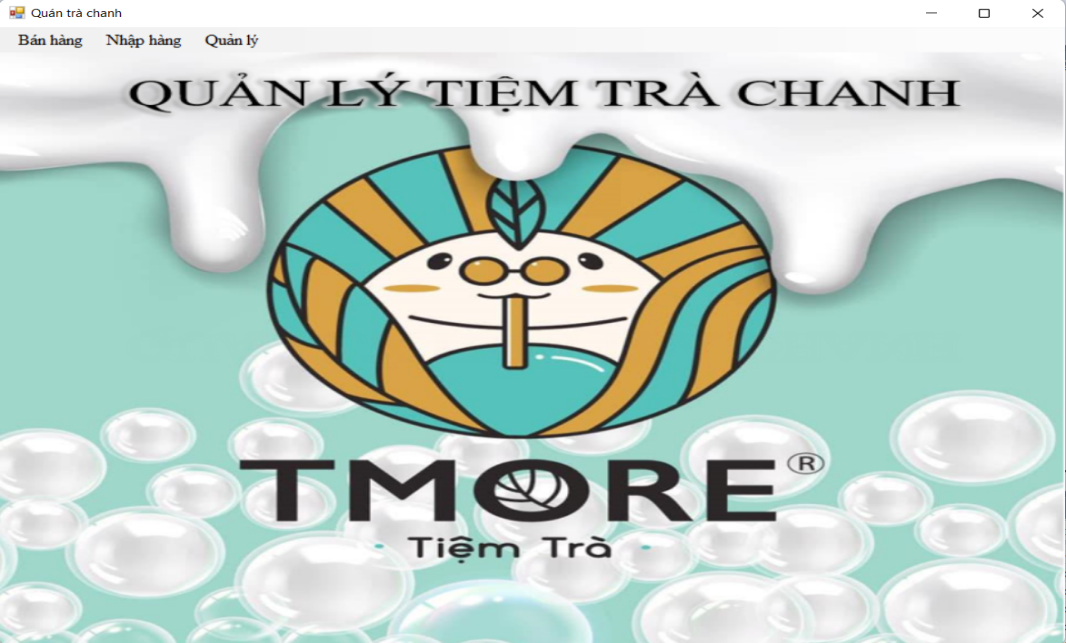
private void btnThoat\_Click(object sender, EventArgs e)

{

Application.Exit();

}

### Form Main



*Hình 20. Giao diện form main*

* Chức năng:  Người dùng sau khi đăng nhập thành công sẽ vào được giao diện Form Main của hệ thống, tại đây có hai chức năng Quản lý bán hàng, nhập hàng dành cho nhân viên và Quản trị hệ thống dành cho admin.
* Code Form Main:

public frmmain()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

nhanvien f = new nhanvien();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

khachhang f = new khachhang();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

hanghoa f = new hanghoa();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button4\_Click\_1(object sender, EventArgs e)

{

nguoiban f = new nguoiban();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

nguyenlieu f = new nguyenlieu();

this.Hide();

f.ShowDialog();

this.Show();

}

private void danhSáchCácNhàCungCấpToolStripMenuItem\_Click(object sender, EventArgs e)

{

nguoiban f = new nguoiban();

this.Hide();

f.ShowDialog();

this.Show();

}

private void khoHàngToolStripMenuItem\_Click(object sender, EventArgs e)

{

nguyenlieu f = new nguyenlieu();

this.Hide();

f.ShowDialog();

this.Show();

}

private void danhSáchNhânViênToolStripMenuItem\_Click(object sender, EventArgs e)

{

nhanvien f = new nhanvien();

this.Hide();

f.ShowDialog();

this.Show();

}

private void danhSáchKháchHàngToolStripMenuItem\_Click(object sender, EventArgs e)

{

khachhang f = new khachhang();

this.Hide();

f.ShowDialog();

this.Show();

}

private void menuSảnPhẩmToolStripMenuItem\_Click(object sender, EventArgs e)

{

hanghoa f = new hanghoa();

this.Hide();

f.ShowDialog();

this.Show();

}

private void bánHàngToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form1 f = new Form1();

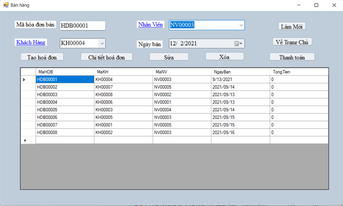
this.Hide();

f.ShowDialog();

this.Show();

}

### Form quản lý bán hàng



*Hình 21. Giao diện form quản lý bán hàng*

* Các thành phần của form bán hàng:

+ Combo box: khách hàng, nhân viên, ngày bán

+ Text box: mã hóa đơn

+ Button: Tạo hóa đơn, chi tiết hóa đơn, sửa, xóa, thanh toán, làm mới, về trang chủ

* Chức năng:

+ Nhân viên thực hiện tạo hóa đơn, thêm, sửa, xóa và thanh toán tại đây, và chọn Về trang chủ để về form chính.

* Code form bán hàng:

public Form1()

{

InitializeComponent();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

DataTable table = loadData().Tables[0];

dataGridView1.DataSource = table;

LoadMaKH();

LoadMaNV();

LoadData();

}

private void LoadData()

{

if (dataGridView1.Rows.Count > 0)

{

txtmakh.Text = dataGridView1.Rows[0].Cells["MaKH"].Value.ToString();

txtmanv.Text = dataGridView1.Rows[0].Cells["MaNV"].Value.ToString();

txtmhd.Text = dataGridView1.Rows[0].Cells["MaHDB"].Value.ToString();

}

}

void LoadMaKH()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select MaKH from KHACH\_HANG";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

DataTable dataTable = data.Tables[0];

foreach (DataRow row in dataTable.Rows)

{

txtmakh.Items.Add(row.Field<string>("MaKH"));

}

if (txtmakh.Items.Count > 0)

{

txtmakh.SelectedIndex = 0;

}

}

void LoadMaNV()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select MaNV from NHAN\_VIEN";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

DataTable dataTable = data.Tables[0];

foreach (DataRow row in dataTable.Rows)

{

txtmanv.Items.Add(row.Field<string>("MaNV"));

}

if (txtmanv.Items.Count > 0)

{

txtmanv.SelectedIndex = 0;

}

}

DataSet loadData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from BAN";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

return data;

}

private void btnthem\_Click(object sender, EventArgs e)

{

try

{

if (txtmhd.Text == "")

{

MessageBox.Show("Hãy Thêm 1 số hóa đơn chưa tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

bool contains = IsContainsData(txtmhd.Text);

if (contains == true)

{

MessageBox.Show("Số Hoá đơn đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "insert into BAN values('" + txtmhd.Text + "','" + txtmakh.Text + "','" + txtmanv.Text + "','" + txttime.Text + "','0')";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm hoá đơn thành công");

dataGridView1.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException ex)

{

MessageBox.Show("Không thể thêm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

bool IsContainsData(string value)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from BAN where MaHDB = '" + value + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int index = dataGridView1.CurrentCell.RowIndex;

txtmhd.Text = dataGridView1.Rows[index].Cells["MaHDB"].Value.ToString();

txtmakh.Text = dataGridView1.Rows[index].Cells["MaKH"].Value.ToString();

txtmanv.Text = dataGridView1.Rows[index].Cells["MaNV"].Value.ToString();

txttime.Text = dataGridView1.Rows[index].Cells["NgayBan"].Value.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

thanhtoan f = new thanhtoan();

f.ma\_hoa\_don = txtmhd.Text;

this.Hide();

f.ShowDialog();

this.Show();

dataGridView1.DataSource = loadData().Tables[0];

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

if (txtmakh.Text == "" || txtmanv.Text == "" || txtmhd.Text == "" || txttime.Text == "")

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

int index = dataGridView1.CurrentCell.RowIndex;

string mahd = dataGridView1.Rows[index].Cells["MaHDB"].Value.ToString();

bool iscontain = IsContainsData(txtmhd.Text);

if (iscontain && txtmhd.Text != mahd)

{

MessageBox.Show("Mã hóa đơn " + txtmhd.Text + " đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "update BAN set MaHDB = '" + txtmhd.Text + "',MaKH='" + txtmakh.Text + "',MaNV = N'" + txtmanv.Text + "',NgayBan = N'" + txttime.Text + "' where MaHDB='" + mahd + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Đã Sửa");

dataGridView1.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

int index = dataGridView1.CurrentCell.RowIndex;

string mahd = dataGridView1.Rows[index].Cells["MAHDB"].Value.ToString();

string sql = "delete from BAN where MAHDB= '" + mahd + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Đã Xóa");

dataGridView1.Rows.RemoveAt(index);

}

catch (System.Data.SqlClient.SqlException ex)

{

MessageBox.Show("Không thể xoá!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button4\_Click(object sender, EventArgs e)

{

int index = dataGridView1.CurrentCell.RowIndex;

int money = Convert.ToInt32(dataGridView1.Rows[index].Cells["TongTien"].Value);

ThanhToan(money);

DataTable table = loadData().Tables[0];

dataGridView1.DataSource = table;

}

private void ThanhToan(int money)

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "update Ban set TongTien = '0' where MaHDB = '" + txtmhd.Text + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

ClearData();

MessageBox.Show("Thanh toán thành công!\nTổng hoá đơn là " + money.ToString("N0") + " đồng!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void ClearData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "delete from Ban\_Chi\_Tiet where MaHDB = '" + txtmhd.Text + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

nhanvien f = new nhanvien();

this.Hide();

f.ShowDialog();

this.Show();

}

private void linkLabel2\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

khachhang f = new khachhang();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 f = new Form1();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

frmmain f = new frmmain();

this.Hide();

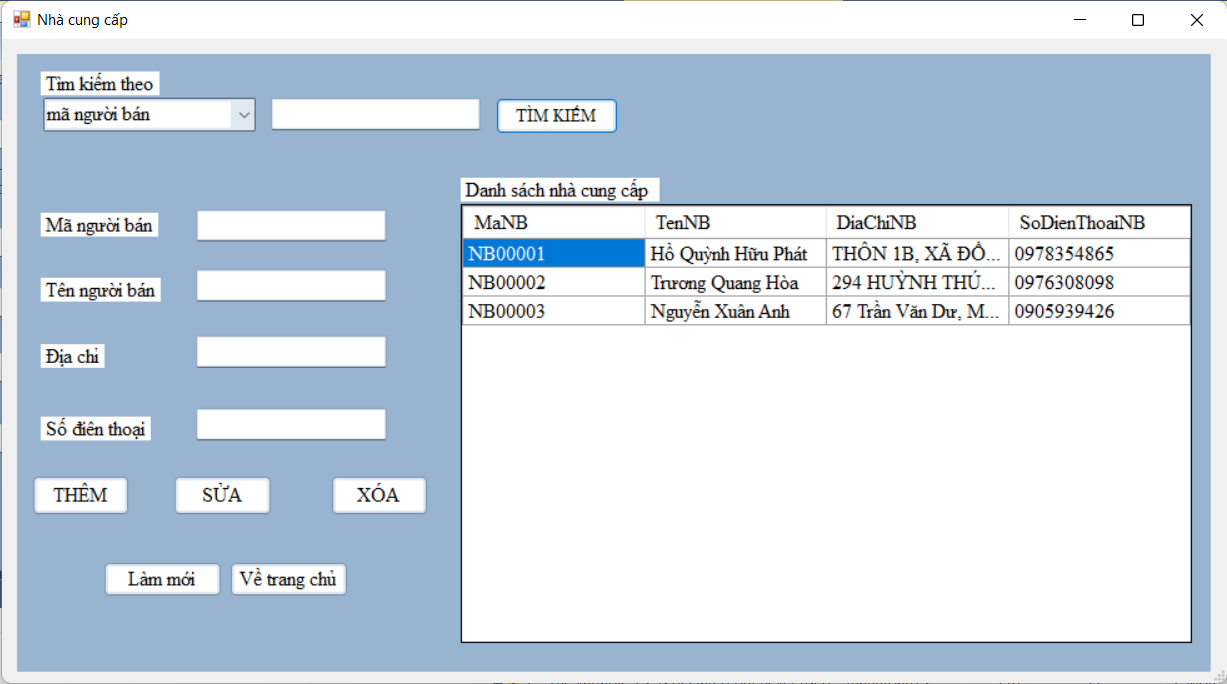
f.ShowDialog();

this.Show();

}

### Form quản lý nhập hàng

* **Form danh sách các nhà cung cấp**



*Hình 22. Giao diện form danh sách nhà cung cấp*

* Thành phần:

+ Text box: tìm kiếm, mã người bán, tên người bán, địa chỉ, Số điện thoaik

+ Combo box : Tìm kiếm theo

+ Button: Thêm, sửa xóa, tìm kiếm, làm mới và về trang chủ

* Chức năng: Nhân viên có thêm sửa xóa cho những thông tin đã chỉnh sửa, nhân viên có thể tìm kiếm người bán theo mã người bán, tên người bán, sđt và địa chỉ.
* Code form nhà cung cấp:

public partial class nguoiban : Form

{

string ketnoi = @"Data Source=LAPTOP-SMKPHOFO;Initial Catalog=Nhom8csdl;Integrated Security=True";

public nguoiban()

{

InitializeComponent();

}

private void nguoiban\_Load(object sender, EventArgs e)

{

dtgvnb.DataSource = loadData().Tables[0];

comboBox2.SelectedIndex = 0;

}

DataSet loadData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from NGUOI\_BAN";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

return data;

}

private void btnthemnv\_Click(object sender, EventArgs e)

{

try

{

if (txtnamenb.Text == "" || txtmanb.Text == "" || txtdiachi.Text == "" || txtsdtnb.Text == "" )

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

bool iscontain = IsContainsData(txtmanb.Text);

if (iscontain == true)

{

MessageBox.Show("Mã người bán đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "insert NGUOI\_BAN values('" + txtmanb.Text + "',N'" + txtnamenb.Text + "',N'" + txtdiachi.Text + "','" + txtsdtnb.Text + "')";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm người bán thành công");

dtgvnb.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể thêm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

bool IsContainsData(string manb)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from NGUOI\_BAN where MANB = '" + manb + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void dtgvnb\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int index = dtgvnb.CurrentCell.RowIndex;

txtmanb.Text = dtgvnb.Rows[index].Cells["MANB"].Value.ToString();

txtnamenb.Text = dtgvnb.Rows[index].Cells["tennb"].Value.ToString();

txtdiachi.Text = dtgvnb.Rows[index].Cells["DIACHINB"].Value.ToString();

txtsdtnb.Text = dtgvnb.Rows[index].Cells["SODIENTHOAINB"].Value.ToString();

}

private void btnsuanv\_Click(object sender, EventArgs e)

{

try

{

int index = dtgvnb.CurrentCell.RowIndex;

string manb = dtgvnb.Rows[index].Cells["MANB"].Value.ToString();

bool iscontain = IsContainsData(txtmanb.Text);

if (iscontain == true && manb != txtmanb.Text)

{

MessageBox.Show("Mã Người bán " + txtmanb.Text + " đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "update NGUOI\_BAN set MANB = '" + txtmanb.Text + "',tennb = N'" + txtnamenb.Text + "',DIACHINB = N'" + txtdiachi.Text + "',SODIENTHOAINB='" + txtsdtnb.Text + "' where MANB='" + manb + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Đã Sửa");

dtgvnb.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnxoanv\_Click(object sender, EventArgs e)

{

try

{

int index = dtgvnb.CurrentCell.RowIndex;

string makh = dtgvnb.Rows[index].Cells["MANB"].Value.ToString();

string sql = "delete from NGUOI\_BAN where MANB= '" + makh + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Đã Xóa");

dtgvnb.Rows.RemoveAt(index);

}

catch (System.Data.SqlClient.SqlException ex)

{

MessageBox.Show("Không thể xoá!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btntimnv\_Click(object sender, EventArgs e)

{

string sql = "";

if (comboBox2.SelectedIndex == 0)

{

sql = "select \* from NGUOI\_BAN where MANB like N'%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 1)

{

sql = "select \* from NGUOI\_BAN where TENNB like N'%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 2)

{

sql = "select \* from NGUOI\_BAN where diachinb like N'%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 3)

{

sql = "select \* from NGUOI\_BAN where sodienthoainb like '%" + textBox1.Text + "%'";

}

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

dtgvnb.DataSource = data.Tables[0];

}

private void button2\_Click(object sender, EventArgs e)

{

nguoiban f = new nguoiban();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

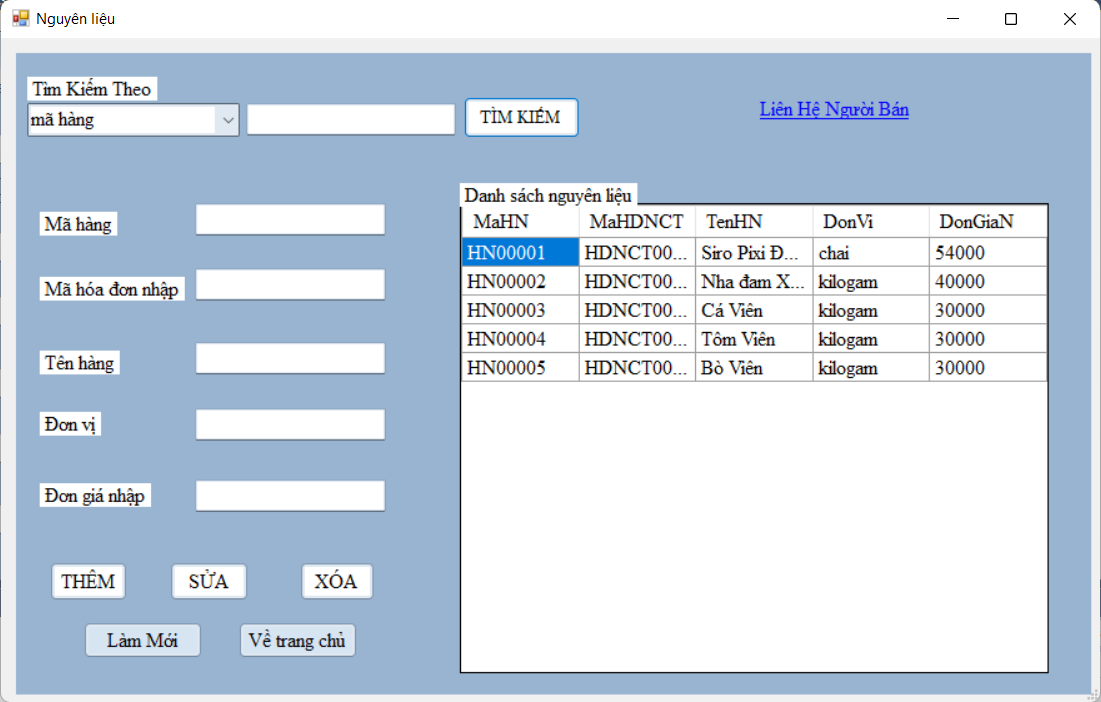
frmmain f = new frmmain();

this.Hide();

f.ShowDialog();

this.Show();

* **Form quản lý nguyên liệu nhập vào**



*Hình 23. Giao diện form quản lý Nguyên liệu*

* Thành phần:

+ Text box: tìm kiếm, mã hàng, mã hóa đơn nhập , tên hàng , đơn vị, đơn giá nhập

+ Combo box : Tìm kiếm theo

+ các button: Thêm, sửa xóa, tìm kiếm , làm mới và về trang chủ

* Chức năng: Nhân viên có thêm sửa xóa cho những thông tin đã chỉnh sửa, nhân viên có thể tìm kiếm nguyên liệu theo mã hàng, mã hóa đơn, tên hàng, đơn vị, đơn giá nhập của nguyên liệu.
* Code form nguyên liệu:

public partial class nguyenlieu : Form

{

string ketnoi = @"Data Source=LAPTOP-SMKPHOFO;Initial Catalog=Nhom8csdl;Integrated Security=True";

public nguyenlieu()

{

InitializeComponent();

}

private void nguyenlieu\_Load(object sender, EventArgs e)

{

dtgvnl.DataSource = loadData().Tables[0];

comboBox2.SelectedIndex = 0;

}

DataSet loadData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from NGUYEN\_LIEU";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

return data;

}

private void btnthemnv\_Click(object sender, EventArgs e)

{

try

{

if (txtdongianhap.Text == "" || txtdonvi.Text == "" || txtmanl.Text == "" || txtmahdn.Text == "" || txttennl.Text == "")

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

bool iscontain = IsContainsDatas(txtmanl.Text, txtmahdn.Text);

if (iscontain == true)

{

MessageBox.Show("nguyên liệu này có rồi!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "insert into NGUYEN\_LIEU values('" + txtmanl.Text + "',N'" + txtmahdn.Text + "',N'" + txttennl.Text + "','" + txtdonvi.Text+ "','" + txtdongianhap.Text + "')";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm nguyên liệu thành công");

dtgvnl.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể thêm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

bool IsContainsDatas(string manl, string mahdnct)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from NGUYEN\_LIEU where MaHN = '" + manl + "' or MaHDNCT = N'" + mahdnct + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void btnsuanv\_Click(object sender, EventArgs e)

{

try

{

if (txtdongianhap.Text == "" || txtdonvi.Text == "" || txtmahdn.Text == "" || txtmanl.Text == "" || txttennl.Text == "")

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

int index = dtgvnl.CurrentCell.RowIndex;

string manl = dtgvnl.Rows[index].Cells["MAHN"].Value.ToString();

bool iscontain = IsContainsData(txtmanl.Text);

if (iscontain && txtmanl.Text != manl)

{

MessageBox.Show("Mã nguyên liệu " + txtmanl.Text + " đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "update NGUYEN\_LIEU set MAHN = '" + txtmanl.Text + "',MaHDNCT = N'" + txtmahdn.Text + "',TenHN = N'" + txttennl.Text + "',DonVi='" + txtdonvi.Text + "',DonGiaN='" + txtdongianhap.Text + "' where MAHN='" + manl + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Đã Sửa");

dtgvnl.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

bool IsContainsData(string manl)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from NGUYEN\_LIEU where MAHN = '" + manl + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void dtgvnl\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int index = dtgvnl.CurrentCell.RowIndex;

txtmanl.Text = dtgvnl.Rows[index].Cells["MAHN"].Value.ToString();

txtmahdn.Text = dtgvnl.Rows[index].Cells["MaHDNCT"].Value.ToString();

txttennl.Text = dtgvnl.Rows[index].Cells["TenHN"].Value.ToString();

txtdonvi.Text = dtgvnl.Rows[index].Cells["DonVi"].Value.ToString();

txtdongianhap.Text = dtgvnl.Rows[index].Cells["DonGiaN"].Value.ToString();

}

private void btnxoanv\_Click(object sender, EventArgs e)

{

try

{

int index = dtgvnl.CurrentCell.RowIndex;

string manl = dtgvnl.Rows[index].Cells["MAHN"].Value.ToString();

string sql = "delete from NGUYEN\_LIEU where MAHN= '" + manl + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Đã Xóa");

dtgvnl.Rows.RemoveAt(index);

}

catch (System.Data.SqlClient.SqlException ex)

{

MessageBox.Show("Không thể xoá!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btntimnv\_Click(object sender, EventArgs e)

{

string sql = "";

if (comboBox2.SelectedIndex == 0)

{

sql = "select \* from NGUYEN\_LIEU where MAHN like '%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 1)

{

sql = "select \* from NGUYEN\_LIEU where MaHDNCT like '%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 2)

{

sql = "select \* from NGUYEN\_LIEU where TenHN like N'%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 3)

{

sql = "select \* from NGUYEN\_LIEU where DonVi like N'%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 4)

{

sql = "select \* from NGUYEN\_LIEU where DonGiaN like N'%" + textBox1.Text + "%'";

}

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

dtgvnl.DataSource = data.Tables[0];

}

private void button2\_Click(object sender, EventArgs e)

{

nguyenlieu f = new nguyenlieu();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

frmmain f = new frmmain();

this.Hide();

f.ShowDialog();

this.Show();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

nguoiban f = new nguoiban();

this.Hide();

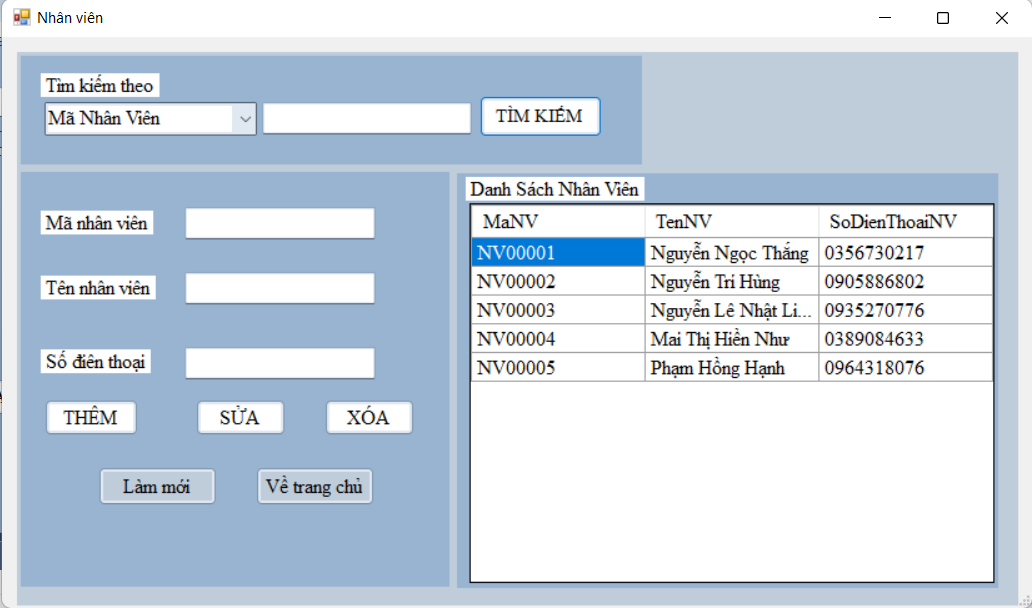
f.ShowDialog();

this.Show();

}

### Form quản lý

* **Form quản lý nhân viên**



*Hình 24. Giao diện form quản lý Nhân viên*

* Thành phần :

+ Combo box: tìm kiếm theo

+ Text box: tìm kiếm, mã nhân viên , tên nhân viên, số điện thoại

+ button: thêm, sửa, xóa, tìm kiếm, làm mới và về trang chủ

* Chức năng: Nhân viên có thể Thêm sửa xóa một nhân viên mới, và tìm kiếm theo mã nhân viên, tên nhân viên, số điện thoại.
* Code form Nhân viên:

public partial class nhanvien : Form

{

string ketnoi = @"Data Source=LAPTOP-SMKPHOFO;Initial Catalog=Nhom8csdl;Integrated Security=True";

public nhanvien()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

nhanvien f = new nhanvien();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

frmmain f = new frmmain();

this.Hide();

f.ShowDialog();

this.Show();

}

private void btnthemnv\_Click(object sender, EventArgs e)

{

try

{

if (txtmanv.Text == "" || txtnamenv.Text == "" || txtsdtnv.Text == "" )

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

bool iscontain = IsContainsData(txtmanv.Text);

if (iscontain == true)

{

MessageBox.Show("Mã nhân viên đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "insert NHAN\_VIEN values('" + txtmanv.Text + "',N'" + txtnamenv.Text + "',N'" + txtsdtnv.Text + "')";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm nhân viên thành công");

dtgvnv.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể thêm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

DataSet loadData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from NHAN\_VIEN";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

return data;

}

bool IsContainsData(string MaNV)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from NHAN\_VIEN where MANV = '" + MaNV + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void nhanvien\_Load(object sender, EventArgs e)

{

dtgvnv.DataSource = loadData().Tables[0];

comboBox2.SelectedIndex = 0;

}

private void btnsuanv\_Click(object sender, EventArgs e)

{

try

{

int index = dtgvnv.CurrentCell.RowIndex;

string manv = dtgvnv.Rows[index].Cells["MANV"].Value.ToString();

bool iscontain = IsContainsData(txtmanv.Text);

if (iscontain == true && manv != txtmanv.Text)

{

MessageBox.Show("Mã nhân viên đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "update NHAN\_VIEN set MANV = '" + txtmanv.Text + "',TenNV = N'" + txtnamenv.Text + "',SoDienThoaiNV = '" + txtsdtnv.Text + "' where MANV='" + manv + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Đã Sửa");

dtgvnv.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnxoanv\_Click(object sender, EventArgs e)

{

try

{

int index = dtgvnv.CurrentCell.RowIndex;

string manv = dtgvnv.Rows[index].Cells["MANV"].Value.ToString();

string sql = "delete from NHAN\_VIEN where MANV= '" + manv + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Đã Xóa");

dtgvnv.Rows.RemoveAt(index);

}

catch (SqlException ex)

{

MessageBox.Show("Không thể xoá!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btntimnv\_Click(object sender, EventArgs e)

{

string sql = "";

if (comboBox2.SelectedIndex == 0)

{

sql = "select \* from NHAN\_VIEN where MANV like '%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 1)

{

sql = "select \* from NHAN\_VIEN where TenNV like N'%" + textBox1.Text + "%'";

}

if (comboBox2.SelectedIndex == 2)

{

sql = "select \* from NHAN\_VIEN where SoDienThoaiNV like '%" + textBox1.Text + "%'";

}

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

dtgvnv.DataSource = data.Tables[0];

}

private void dtgvnv\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int index = dtgvnv.CurrentCell.RowIndex;

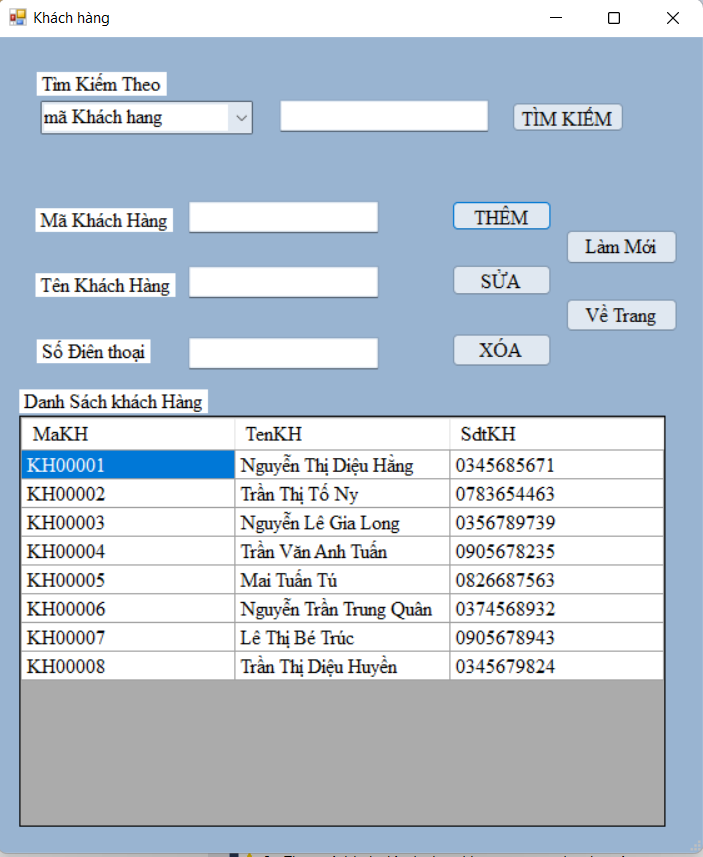
txtmanv.Text = dtgvnv.Rows[index].Cells["MANV"].Value.ToString();

txtnamenv.Text = dtgvnv.Rows[index].Cells["TenNV"].Value.ToString();

txtsdtnv.Text = dtgvnv.Rows[index].Cells["SoDienThoaiNV"].Value.ToString();

}

* **Form quản lý danh sách khách hàng**



*Hình 25. Giao diện form quản lý Khách hàng*

* Thành phần :

+ Combo box: tìm kiếm theo

+ Text box: tìm kiếm, mã khách hàng , tên khách hàng, số điện thoại

+ button: thêm, sửa, xóa, tìm kiếm, làm mới và về trang chủ

* Chức năng: Nhân viên có thể Thêm sửa xóa một khách hàng mới, và tìm kiếm theo mã khách hàng, tên khách hàng, số điện thoại.
* Code form khách hàng:

public partial class khachhang : Form

{

string ketnoi = @"Data Source=LAPTOP-SMKPHOFO;Initial Catalog=Nhom8csdl;Integrated Security=True";

public khachhang()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

khachhang f = new khachhang();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

frmmain f = new frmmain();

this.Hide();

f.ShowDialog();

this.Show();

}

private void khachhang\_Load(object sender, EventArgs e)

{

dtgvkh.DataSource = loadData().Tables[0];

comboBox1.SelectedIndex = 0;

}

DataSet loadData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from KHACH\_HANG";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

return data;

}

private void btnthem\_Click(object sender, EventArgs e)

{

try

{

if (txtname.Text == "" || txtma.Text == "" || txtsdt.Text == "" )

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

bool iscontain = IsContainsData(txtma.Text);

if (iscontain == true)

{

MessageBox.Show("Mã Khách hàng đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "insert KHACH\_HANG values('" + txtma.Text + "',N'" + txtname.Text + "','" + txtsdt.Text + "')";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm khách hàng thành công");

dtgvkh.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể thêm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

bool IsContainsData(string makh)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from KHACH\_HANG where MAKH = '" + makh + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void btnsua\_Click(object sender, EventArgs e)

{

try

{

int index = dtgvkh.CurrentCell.RowIndex;

string makh = dtgvkh.Rows[index].Cells["MAKH"].Value.ToString();

bool iscontain = IsContainsData(txtma.Text);

if (iscontain == true && makh != txtma.Text)

{

MessageBox.Show("Mã Khách hàng " + txtma.Text + " đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "update KHACH\_HANG set MAKH = '" + txtma.Text + "',tenkh = N'" + txtname.Text + "',SDTKH='" + txtsdt.Text + "' where MAKH='" + makh + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Đã Sửa");

dtgvkh.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnxoa\_Click(object sender, EventArgs e)

{

try

{

int index = dtgvkh.CurrentCell.RowIndex;

string makh = dtgvkh.Rows[index].Cells["MAKH"].Value.ToString();

string sql = "delete from KHACH\_HANG where MAKH= '" + makh + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Đã Xóa");

dtgvkh.Rows.RemoveAt(index);

}

catch (System.Data.SqlClient.SqlException ex)

{

MessageBox.Show("Không thể xoá!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void dtgvkh\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int index = dtgvkh.CurrentCell.RowIndex;

txtma.Text = dtgvkh.Rows[index].Cells["MAKH"].Value.ToString();

txtname.Text = dtgvkh.Rows[index].Cells["tenkh"].Value.ToString();

txtsdt.Text = dtgvkh.Rows[index].Cells["SDTKH"].Value.ToString();

}

private void btntimkiem\_Click(object sender, EventArgs e)

{

string sql = "";

if (comboBox1.SelectedIndex == 0)

{

sql = "select \* from KHACH\_HANG where MAKH like '%" + textBox6.Text + "%'";

}

if (comboBox1.SelectedIndex == 1)

{

sql = "select \* from KHACH\_HANG where TENKH like N'%" + textBox6.Text + "%'";

}

if (comboBox1.SelectedIndex == 2)

{

sql = "select \* from KHACH\_HANG where sdtkh like '%" + textBox6.Text + "%'";

}

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

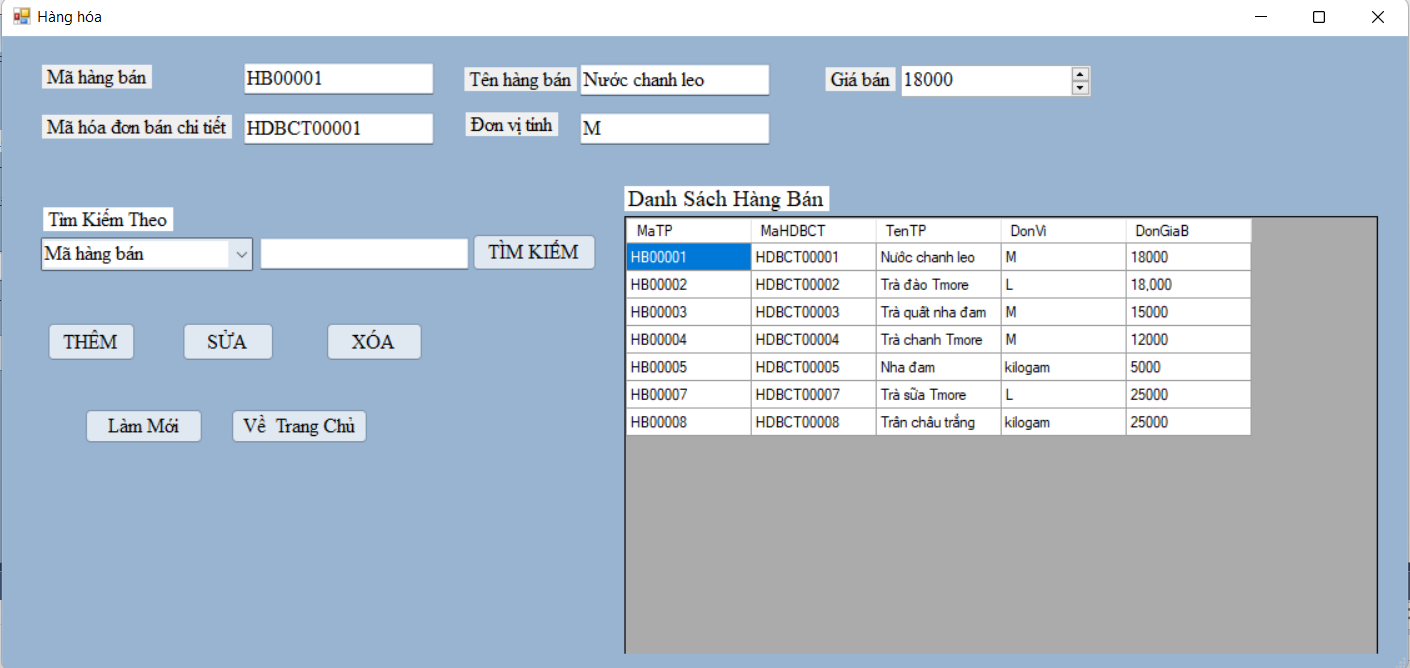
app.Fill(data);

con.Close();

dtgvkh.DataSource = data.Tables[0];

}

**Form quản lý hàng hóa**



*Hình 26. Giao diện form quản lý hàng hóa*

* Thành phần:

+ Combo box: Tìm kiếm theo

+ Text box: Mã hàng bán, Mã hóa đơn chi tiết, tên hàng bán, đơn vị tính, giá bán, tìm kiếm

+ Button: Thêm, sửa, xóa, tìm kiếm, làm mới và về trang chủ

* Chức năng: Nhân viên có thể thêm sửa xóa và tìm kiếm một sản phẩm.
* Code form quản lý hàng hóa:

public partial class hanghoa : Form

{

string ketnoi = @"Data Source=LAPTOP-SMKPHOFO;Initial Catalog=Nhom8csdl;Integrated Security=True";

public hanghoa()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

hanghoa f = new hanghoa();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

frmmain f = new frmmain();

this.Hide();

f.ShowDialog();

this.Show();

}

private void hanghoa\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = loadData().Tables[0];

comboBox3.SelectedIndex = 0;

}

DataSet loadData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from THANH\_PHAM";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

return data;

}

private void btnthemhh\_Click(object sender, EventArgs e)

{

try

{

if (txtmahh.Text == "" || txttenhh.Text == "" || txtmahhct.Text == "" || txtdvt.Text == "" || txtgia.Text == "" )

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

bool iscontain = IsContainsDatas(txtmahh.Text, txttenhh.Text);

if (iscontain == true)

{

MessageBox.Show("Sản phẩm đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "insert into THANH\_PHAM values('" + txtmahh.Text + "','"+ txtmahhct.Text+"',N'" + txttenhh.Text + "',N'" + txtdvt.Text + "','" + Convert.ToInt32(txtgia.Text).ToString("N0") + "')";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm Sản phẩm thành công");

dataGridView1.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể thêm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

bool IsContainsDatas(string mahh, string tenhh)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from THANH\_PHAM where MaTP = '" + mahh + "' or TenTP = N'" + tenhh + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void btnsuahh\_Click(object sender, EventArgs e)

{

try

{

if (txtmahh.Text == "" || txttenhh.Text == "" || txtmahhct.Text == "" || txtdvt.Text == "" || txtgia.Text == "")

{

MessageBox.Show("Hãy Thêm Thông Tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

int index = dataGridView1.CurrentCell.RowIndex;

string matp = dataGridView1.Rows[index].Cells["MaTP"].Value.ToString();

bool iscontain = IsContainsData(txtmahh.Text);

if (iscontain && txtmahh.Text != matp)

{

MessageBox.Show("Mã sản phẩm " + txtmahh.Text + " đã tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sql = "update THANH\_PHAM set MaTP = '" + txtmahh.Text + "',MaHDBCT='"+ txtmahhct.Text+ "',TenTP = N'" + txttenhh.Text + "',DonVi = N'" + txtdvt.Text + "',DonGiaB='" + Convert.ToInt32(txtgia.Text).ToString("N0") + "'where MaTP='" + matp + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Đã Sửa");

dataGridView1.DataSource = loadData().Tables[0];

con.Close();

}

catch (SqlException)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception)

{

MessageBox.Show("Không thể sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

bool IsContainsData(string matp)

{

try

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from THANH\_PHAM where MaTP = '" + matp + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

if (data.Tables[0].Rows.Count == 0)

{

return false;

}

else

{

return true;

}

}

catch (SqlException)

{

return false;

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int index = dataGridView1.CurrentCell.RowIndex;

txtmahh.Text = dataGridView1.Rows[index].Cells["MATP"].Value.ToString();

txtmahhct.Text = dataGridView1.Rows[index].Cells["MaHDBCT"].Value.ToString();

txttenhh.Text = dataGridView1.Rows[index].Cells["TENTP"].Value.ToString();

txtdvt.Text = dataGridView1.Rows[index].Cells["donvi"].Value.ToString();

txtgia.Text = dataGridView1.Rows[index].Cells["dongiaB"].Value.ToString();

}

private void btnxoahh\_Click(object sender, EventArgs e)

{

try

{

int index = dataGridView1.CurrentCell.RowIndex;

string matp = dataGridView1.Rows[index].Cells["MAtp"].Value.ToString();

string sql = "delete from THANH\_PHAM where MAtP= '" + matp + "'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Đã Xóa");

dataGridView1.Rows.RemoveAt(index);

}

catch (System.Data.SqlClient.SqlException ex)

{

MessageBox.Show("Không thể xoá!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btntimhh\_Click(object sender, EventArgs e)

{

string sql = "";

if (comboBox3.SelectedIndex == 0)

{

sql = "select \* from THANH\_PHAM where MaTP like '%" + textBox9.Text + "%'";

}

if (comboBox3.SelectedIndex == 1)

{

sql = "select \* from THANH\_PHAM where MaHDBCT like '%" + textBox9.Text + "%'";

}

if (comboBox3.SelectedIndex == 2)

{

sql = "select \* from THANH\_PHAM where TenTP like N'%" + textBox9.Text + "%'";

}

if (comboBox3.SelectedIndex == 3)

{

sql = "select \* from THANH\_PHAM where donvi like N'%" + textBox9.Text + "%'";

}

if (comboBox3.SelectedIndex == 4)

{

sql = "select \* from THANH\_PHAM where DonGiaB like N'%" + textBox9.Text + "%'";

}

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

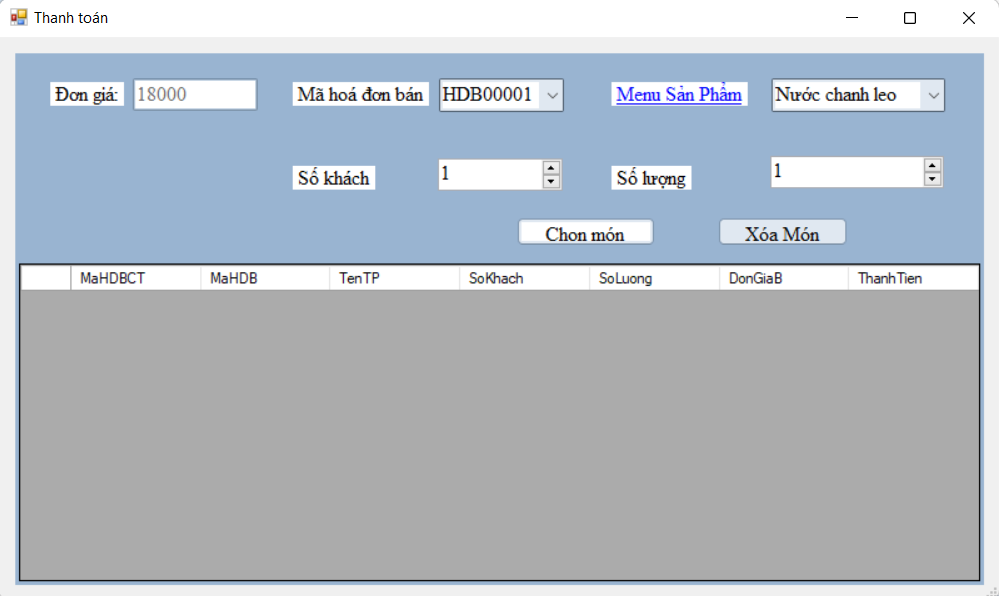
app.Fill(data);

con.Close();

dataGridView1.DataSource = data.Tables[0];

}

* **Form quản lý hóa đơn chi tiết**



*Hình 27. Giao diện quản lý hóa đơn thanh toán*

* Các thành phần

+ Combo box: Mã hóa đơn bán, menu sản phẩm

+ Text box: Đơn giá, số khách, số lượng

+ Button: Xóa và chọn món

* Chức năng: Nhân viên chọn món cho một hóa đơn và có thể xóa một món trong hóa đơn này.
* Code form quản lý hóa đơn thanh toán

public partial class thanhtoan : Form

{

string ketnoi = @"Data Source=LAPTOP-SMKPHOFO;Initial Catalog=Nhom8csdl;Integrated Security=True";

public string ma\_hoa\_don = "";

int ma\_hoa\_don\_ban\_ct = 0;

public thanhtoan()

{

InitializeComponent();

}

private void thanhtoan\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = loadData().Tables[0];

txtmahd.Text = ma\_hoa\_don;

LoadMaHDB();

LoadTenTP();

LoadDonGia();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

LoadDonGia();

}

void LoadTenTP()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select TENTP from THANH\_PHAM";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

DataTable dataTable = data.Tables[0];

foreach (DataRow row in dataTable.Rows)

{

txttentp.Items.Add(row.Field<string>("TENTP"));

}

if (txttentp.Items.Count > 0)

{

txttentp.SelectedIndex = 0;

}

}

void LoadMaHDB()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select MaHDB from BAN\_CHI\_TIET where MaHDB ='" + ma\_hoa\_don + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

DataTable dataTable = data.Tables[0];

foreach (DataRow row in dataTable.Rows)

{

txtmahd.Items.Add(row.Field<string>("MaHDB"));

}

if (txtmahd.Items.Count > 0)

{

txtmahd.SelectedIndex = 0;

}

}

void LoadDonGia()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select DonGiaB from THANH\_PHAM where TenTP = N'" + txttentp.Text + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

DataTable dataTable = data.Tables[0];

foreach (DataRow row in dataTable.Rows)

{

txtdongia.Text = row.Field<string>("DonGiaB");

}

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

int tien = Convert.ToInt32(txtdongia.Text.Replace(",","")) \* Convert.ToInt32(txtsoluong.Value);

string sql = "insert BAN\_CHI\_TIET values('" + ma\_hoa\_don\_ban\_ct + "','" + txtmahd.Text + "',N'" + txttentp.Text + "','" + txtsokhach.Text + "','" + txtsoluong.Text + "','" + txtdongia.Text + "','" + tien +"')";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm hoá đơn thành công");

dataGridView1.DataSource = loadData().Tables[0];

con.Close();

ma\_hoa\_don\_ban\_ct++;

}

catch (SqlException ex)

{

MessageBox.Show("Không thể thêm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ThanhToan()

{

int money = 0;

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select ThanhTien from Ban\_Chi\_Tiet where MaHDB = N'" + ma\_hoa\_don + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

DataTable dataTable = data.Tables[0];

foreach (DataRow row in dataTable.Rows)

{

money = Convert.ToInt32(row.Field<string>("ThanhTien")) + money;

}

UpdateStatusThanhToan(money);

}

private void UpdateStatusThanhToan(int money)

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "update Ban set TongTien = '" + money + "' where MaHDB = '" + ma\_hoa\_don + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

}

DataSet loadData()

{

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

string sql = "select \* from BAN\_CHI\_TIET where MaHDB = '" + ma\_hoa\_don + "'";

SqlDataAdapter app = new SqlDataAdapter(sql, con);

DataSet data = new DataSet();

app.Fill(data);

con.Close();

return data;

}

private void thanhtoan\_FormClosing(object sender, FormClosingEventArgs e)

{

ThanhToan();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

hanghoa f = new hanghoa();

this.Hide();

f.ShowDialog();

this.Show();

}

private void button3\_Click\_1(object sender, EventArgs e)

{

try

{

int index = dataGridView1.CurrentCell.RowIndex;

string mahdbct = dataGridView1.Rows[index].Cells["MAHDBCT"].Value.ToString();

string sql = "delete from BAN\_CHI\_TIET where MAHDB= '" + ma\_hoa\_don + "'and MaHDBCT = '"+mahdbct+"'";

SqlConnection con = new SqlConnection(ketnoi);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = sql;

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Đã Xóa");

dataGridView1.Rows.RemoveAt(index);

}

catch (System.Data.SqlClient.SqlException ex)

{

MessageBox.Show("Không thể xoá!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

# YÊU CẦU NÂNG CAO

## Phương án tốt nhất để đề phòng hệ thống bị tấn công bằng SQL Injection:

* SQL Injection được biết đến như một cách tấn công kho dữ liệu SQL vô cùng nguy hiểm. Những hậu quả nó để lại là vô cùng to lớn. Với SQL Injection, hacker có thể truy cập một phần hoặc toàn bộ dữ liệu trong hệ thống. Hậu quả lớn nhất mà SQL Injection gây ra là làm lộ dữ liệu trong database. Tuỳ vào tầm quan trọng của dữ liệu mà hậu quả dao động ở mức nhẹ cho đến vô cùng nghiêm trọng. Để đề phòng hệ thống bị tấn công trước SQL Injection, ta có thể thực hiện một số biện pháp sau:
* **Phân quyền rõ ràng trong database:** Nếu chỉ truy cập dữ liệu từ một số bảng, hãy tạo một account trong database, gán quyền truy cập cho account đó. Lúc này, dù hacker có inject được sql cũng không thể đọc dữ liệu từ các bảng chính, sửa hay xoá dữ liệu.
* Giới hạn quyền truy cập của người dùng đối với cơ sở dữ liệu: Chỉ những tài khoản có quyền truy cập theo yêu cầu mới được kết nối với cơ sở dữ liệu. Điều này có thể giúp giảm thiểu những lệnh SQL được thực thi tự động trên server.
* Thường xuyên backup dữ liệu: Dữ liệu phải được backup thường xuyên, như vậy nếu bị hacker xóa thì vẫn có thể khôi phục được.

# KẾT Luận

Hiện nay, trong thời đại kinh tế ngày càng phát triển, không chỉ những công ty, doanh nghiệp lớn mà kể cả những cá nhân đều phải xử lý một khối lượng lớn công việc mỗi ngày. Điều này đôi khi gặp rất nhiều khó khăn và cản trở. Do đó, để thoát khỏi áp lực công việc người ta bắt đầu áp dụng những **ứng dụng quản lý công việc.**

**Việc áp dụng các ứng dụng vào công việc đem lại cho người sử dụng rất nhiều lợi ích,** ứng dụng sẽ tạo ra môi trường làm việc dễ dàng hơn cho cả người quản lý và các nhân sự. Khi sử dụng ứng dụng, có thể dễ dàng theo dõi mọi thay đổi trong kế hoạch công việc hàng ngày. Điều này giúp tiết kiệm thời gian và chi phí hơn.

Trên đây là những kiến thức nhóm đã tìm hiểu và nắm được trong quá trình nghiên cứu. Nhóm đã tạo được ứng dụng quản lý quán trà chanh T-More với các chức năng cơ bản như: hiển thị thông tin, thống kê, thêm mới, sửa, xóa, tìm kiếm thông tin… nhằm hỗ trợ công tác quản lý quán. Do vốn kiến thức còn chưa rộng, hơn nữa điều kiện tiếp xúc thực tiễn và thời gian nghiên cứu còn hạn chế nên nhóm chưa trình bày thực sự tốt và kỹ lưỡng các vấn đề liên quan.