

# Xây dựng chương trình quản lý bán hàng bằng C#

## Nội dung [Ẩn]

- [Giới thiệu](#)
- [1. Yêu cầu](#)
- [2. Thiết kế cơ sở dữ liệu](#)
  - [a\) Tạo ứng dụng mới](#)
  - [b\) Tạo cơ sở dữ liệu](#)
    - [Bảng tblChatLieu \(chất liệu\)](#)
    - [Bảng tblKhach \(khách\)](#)
    - [Bảng tblHang \(hàng\)](#)
    - [Bảng tblNhanVien \(nhân viên\)](#)
    - [Bảng tblHDBan \(hoá đơn bán\)](#)
    - [Bảng tblChiTietHDBan \(chi tiết hoá đơn bán\)](#)
    - [Quan hệ giữa các bảng \(Relationship\)](#)
    - [Một số chú ý](#)
    - [Xem video hướng dẫn](#)
- [3. Thiết kế giao diện](#)
  - [a\) Form chính](#)
    - [Xem Video hướng dẫn](#)
  - [b\) Form Danh mục chất liệu](#)
    - [Xem Video hướng dẫn](#)
  - [c\) Form Danh mục nhân viên](#)
    - [Xem Video hướng dẫn](#)
  - [d\) Form Danh mục Khách Hàng](#)
    - [Xem Video hướng dẫn](#)
  - [e\) Form Danh mục hàng hoá](#)
    - [Xem Video hướng dẫn](#)
  - [f\) Form Hoá đơn bán hàng](#)
    - [Xem Video hướng dẫn](#)
  - [g\) Form Tìm kiếm hoá đơn](#)
    - [Xem Video hướng dẫn](#)
- [4. Xử lý sự kiện](#)
  - [4.1. Lớp Functions](#)
  - [4.2. Form frmMain](#)

- a) Sự kiện frmMain Load
- b) Sự kiện mnuThoat Click
- c) Hiển thị các form khác
- 4.3. Form Chat lieu
  - a) Khai báo
  - b) Sự kiện frmDMChatLieu Load
  - c) Phương thức LoadDataGridView
  - d) Phương thức GetDataToTable
  - Xem Video hướng dẫn
  - e) Phương thức dgvChatLieu Click
  - f) Phương thức btnThem Click
  - g) Phương thức ResetValues
  - h) Phương thức btnLuu Click
  - i) Hàm CheckKey
  - j) Phương thức RunSQL
  - k) Phương thức btnSua Click
  - l) Phương thức btnXoa Click
  - m) Phương thức RunSQLDel
  - n) Phương thức btnBoQua Click
  - o) Phương thức dùng phím Enter thay cho phím Tab
  - Xem video hướng dẫn phần trên
- 4.4. Form Danh mục Nhân viên
  - a) Khai báo
  - b) Phương thức frmDMNhanvien Load
  - c) Phương thức LoadDataGridView – Hiển thị dữ liệu lên lưới
  - d) Phương thức dgvNhanVien Click
  - e) Phương thức btnThem Click
  - f) Phương thức ResetValues
  - g) Phương thức btnLuu Click
  - h) Hàm IsDate
  - i) Hàm ConvertDateTime
  - j) Phương thức btnSua Click
  - k) Phương thức btnXoa Click
  - l) Phương thức btnBoQua Click
  - n) Phương thức btnDong Click
  - Xem video hướng dẫn chi tiết
- 4.5. Form Danh mục Khách hàng
  - a) Khai báo
  - b) Phương thức frmDMKhachHang Load
  - c) Phương thức LoadDataGridView

- d) Phương thức dgvKhachHang Click
- e) Phương thức btnThem Click
- f) Phương thức ResetValues
- g) Phương thức btnLuu Click
- h) Phương thức btnSua Click
- i) Phương thức btnXoa Click
- j) Phương thức btnBoQua Click
- k) Phương thức dùng phím Enter thay cho phím Tab
- Xem video hướng dẫn Quản lý Khách Hàng

○ 4.6. Form Danh mục Hàng hóa

- a) Khai báo
- b) Phương thức frmDMHang Load
- c) Phương thức FillCombo
- d) Phương thức ResetValues
- e) Phương thức LoadDataGridView
- f) Phương thức dgvHang Click
- h) Phương thức btnThem Click
- i) Phương thức btnLuu Click
- j) Phương thức btnSua Click
- k) Phương thức btnXoa Click
- l) Phương thức btnBoQua Click
- m) Phương thức btnOpen Click
- n) Phương thức btnTimKiem Click
- o) Phương thức btnHienThi Click
- Xem Video hướng dẫn Quản lý Hàng Hoá

○ 4.7. Form Hóa đơn bán

- a) Khai báo
- b) Phương thức frmHoaDonBan Load
- c) Phương thức LoadDataGridView
- d) Phương thức LoadInfoHoaDon()
- e) Phương thức btnThem Click
- f) Hàm CreateKey
- g) Hàm ConvertTimeTo24
- h) Phương thức ResetValues
- i) Phương thức btnLuu Click
- j) Hàm ChuyenSoSangChu
- k) Phương thức ResetValuesHang
- l) Phương thức dgvHDBanHang DoubleClick
- m) Phương thức btnXoa Click
- n) Phương thức cboMaNhanVien TextChanged
- o) Phương thức cboMaKhach TextChanged

- [p\) Phương thức cboMaHang\\_TextChanged](#)
- [q\) Phương thức txtSoLuong\\_TextChanged](#)
- [r\) Phương thức txtGiamGia\\_TextChanged](#)
- [s\) Phương thức btnInHoaDon\\_Click](#)
- [t\) Phương thức btnTimKiem\\_Click](#)
- [u\) Phương thức txtSoLuong\\_KeyPress](#)
- [v\) Phương thức cboMaHDBan\\_DropDown](#)
- [Xem Video hướng dẫn Quản lý Hoá Đơn](#)
- [4.8. Form tìm kiếm Hóa đơn bán](#)
  - [a\) Khai báo](#)
  - [b\) Phương thức frmTimHDBan\\_Load](#)
  - [c\) Phương thức ResetValues](#)
  - [d\) Phương thức btnTimKiem\\_Click](#)
  - [e\) Phương thức LoadDataGridView](#)
  - [f\) Phương thức btnTimLai\\_Click](#)
  - [g\) Phương thức txtTongTien\\_KeyPress](#)
  - [h\) Phương thức dgvTKHoaDon\\_DoubleClick](#)
  - [i\) Phương thức btnDong\\_Click](#)

## Giới thiệu

Với sự ra đời của .Net và Visual Studio của Microsoft, việc xây dựng một chương trình quản lý với cơ sở dữ liệu đã đơn giản hơn rất nhiều. Bài này hướng dẫn bạn cách tạo ứng dụng quản lý bán hàng lưu niệm sử dụng ngôn ngữ lập trình C# và hệ quản trị cơ sở dữ liệu SQL Server. Tham khảo thêm [khóa học đầy đủ lập trình .NET](#) và [khóa thành thạo với lập trình C#](#).

## 1. Yêu cầu

Xây dựng chương trình quản lý cửa hàng Bán hàng lưu niệm sử dụng ngôn ngữ C# và hệ quản trị cơ sở dữ liệu SQL Server. Hệ thống có các chức năng cơ bản như quản lý mặt hàng, quản lý khách hàng, quản lý hoá đơn bán. Cửa hàng có thể có một hoặc nhiều người bán, với mỗi hoá đơn bán hàng phải có thông tin của người bán cho khách hàng cụ thể.

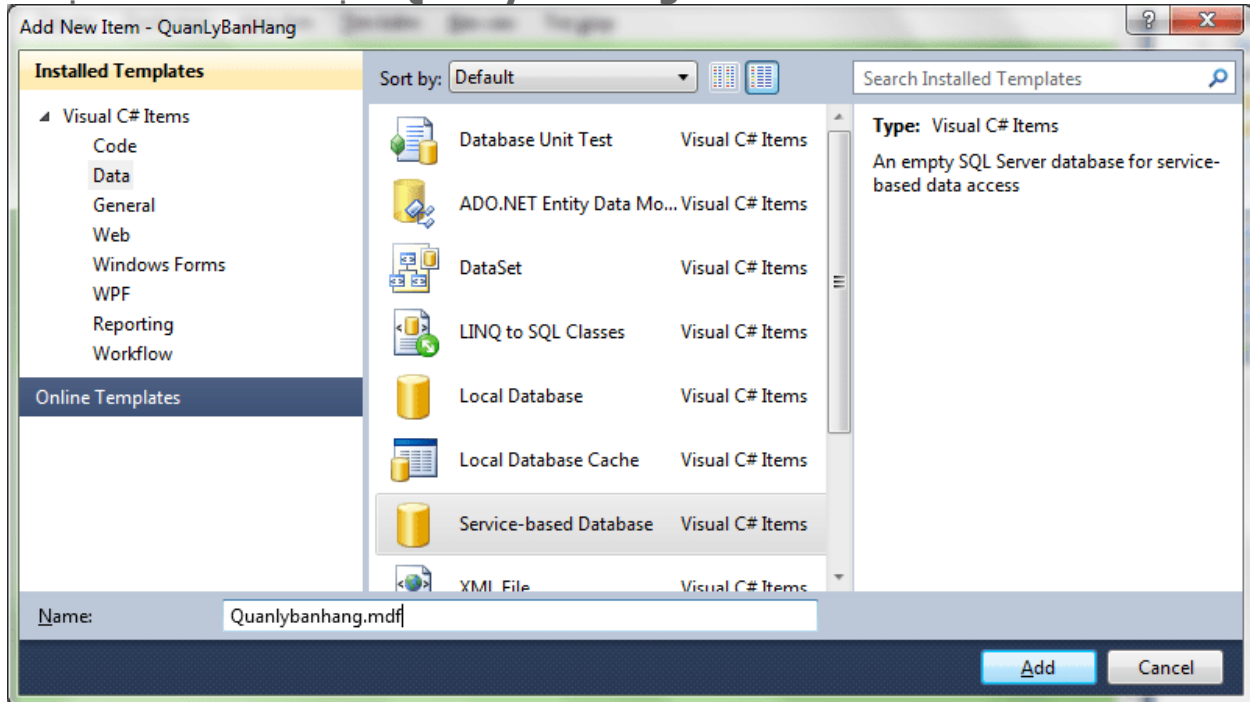
## 2. Thiết kế cơ sở dữ liệu

### *a) Tạo ứng dụng mới*

– Tên project: QuanLyBanHang

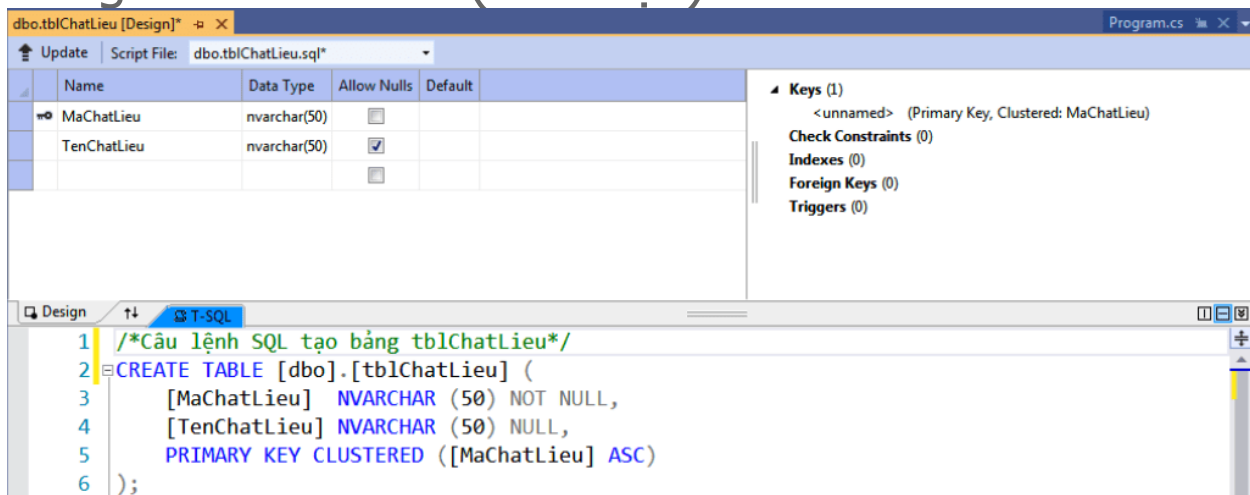
## b) Tạo cơ sở dữ liệu

- Trong cửa sổ **Solution Explorer**, nháy phải chuột lên tên ứng dụng, chọn **Add -> New Item...** Chọn **Data -> Service-based Database** (hoặc **SQL Database** trong Visual Studio Net 2005).
- Đặt tên cơ sở dữ liệu: **Quanlybanhang.mdf**

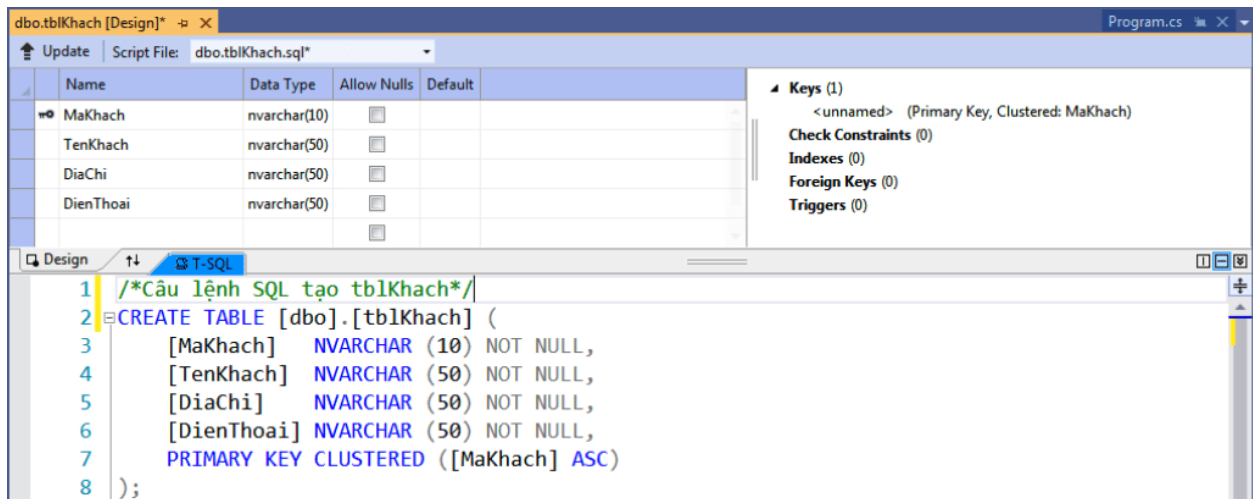


- Tạo các bảng: Trong cửa sổ **Server Explorer**, chọn cơ sở dữ liệu, nháy phải chuột lên **Tables**, chọn **Add New Table...**

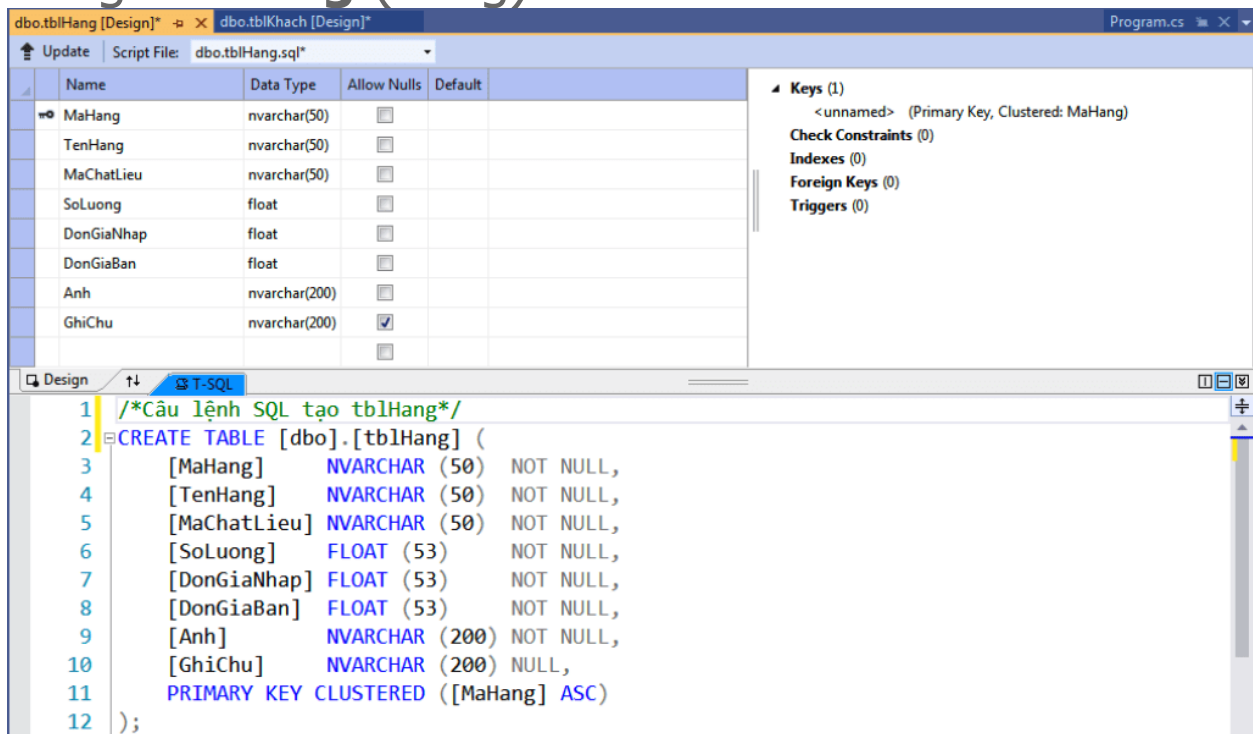
## Bảng **tblChatLieu** (chất liệu)



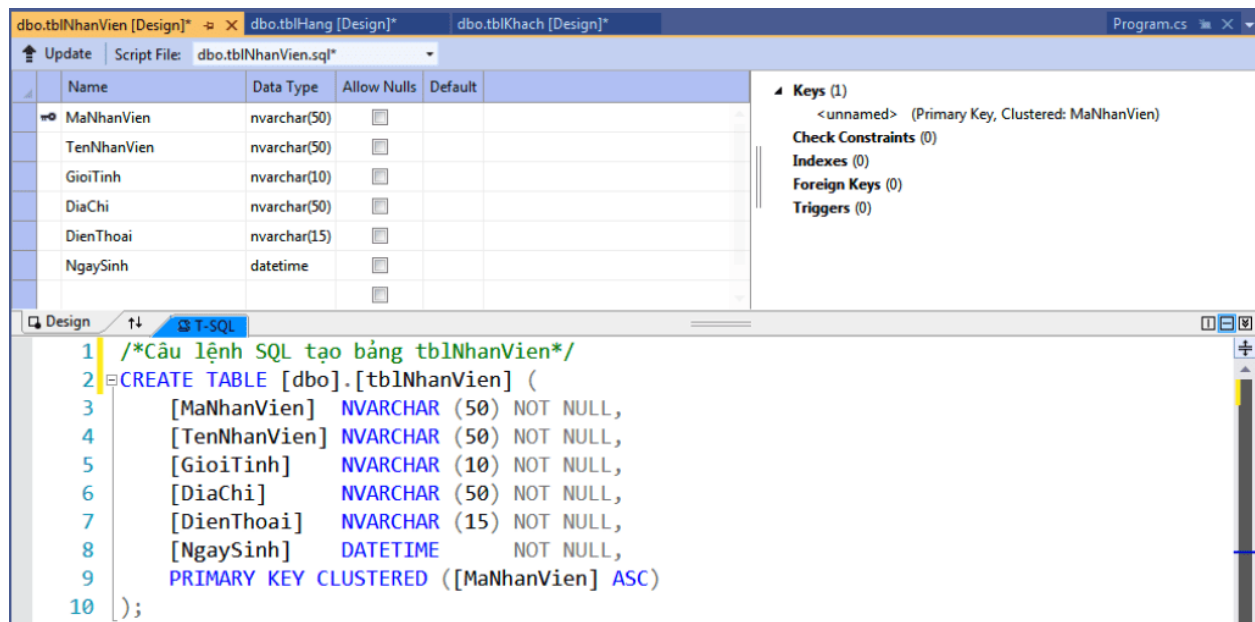
## Bảng **tblKhach** (khách)



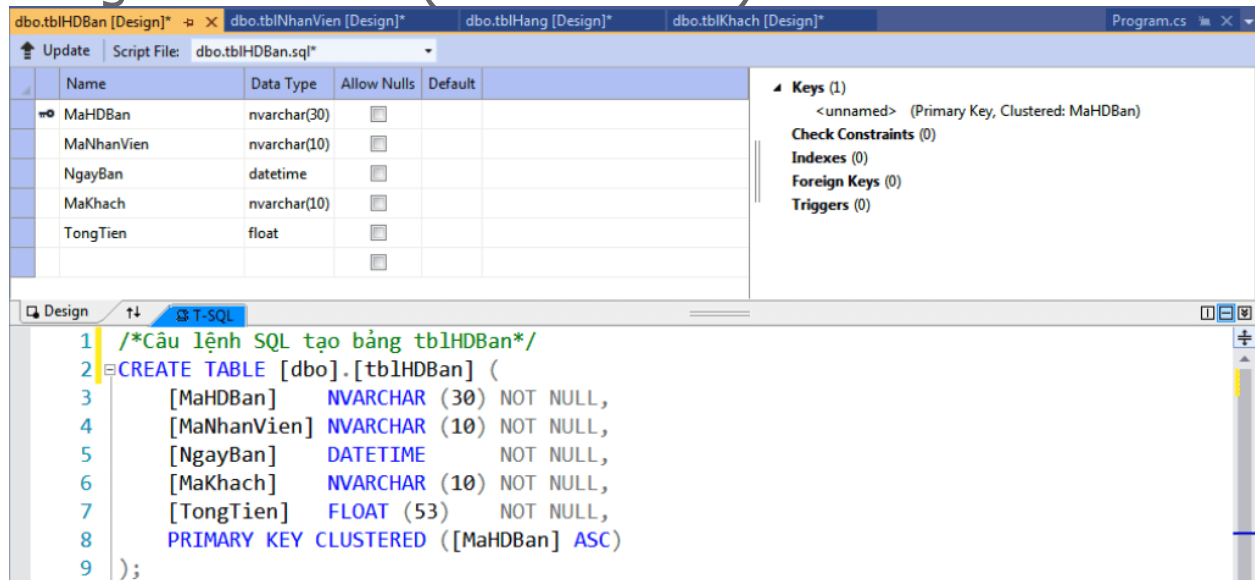
## Bảng **tblHang** (hàng)



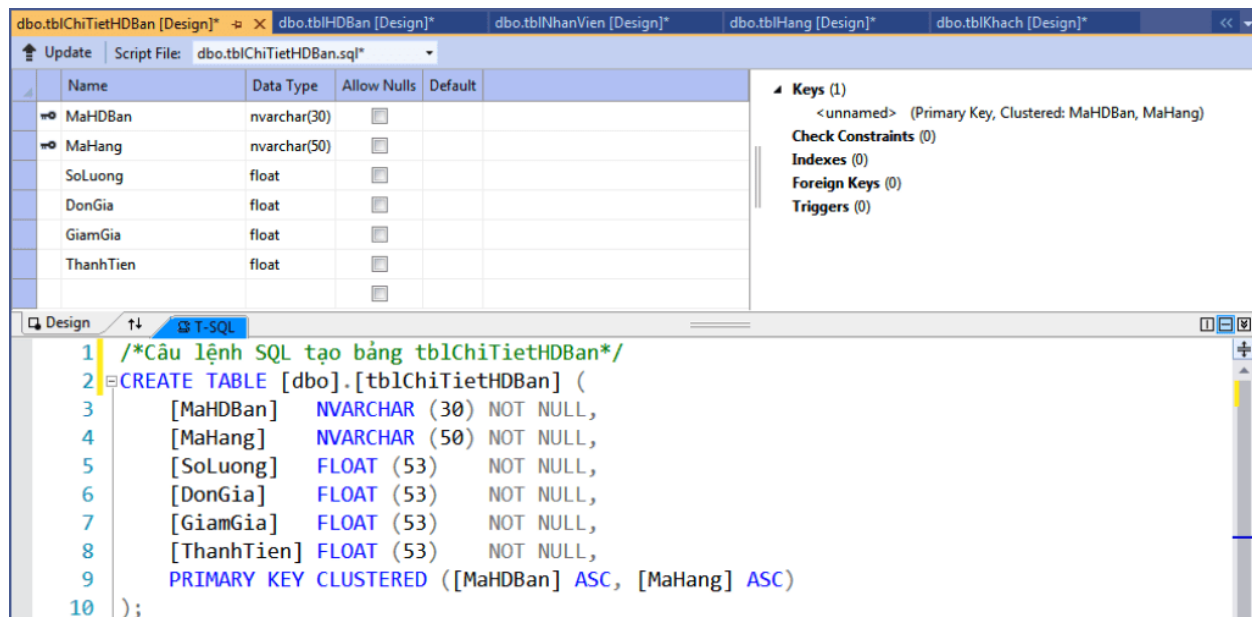
## Bảng **tblNhanVien** (nhân viên)



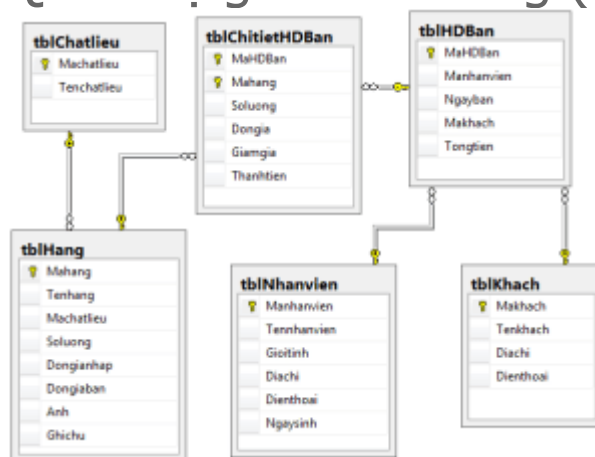
## Bảng **tblHDBan** (hoá đơn bán)



## Bảng **tblChiTietHDBan** (chi tiết hoá đơn bán)



## Quan hệ giữa các bảng (Relationship)



## Một số chú ý

– Để cho phép người dùng sửa đổi thông tin của các bảng, thực hiện như sau:

Vào **Tools -> Options**, chọn **Database Tools -> Table and Database Designers**, bỏ dấu chọn ở mục **Prevent saving changes that require table re-creation**.

– Xóa liên kết tới bản sao cơ sở dữ liệu: trong cửa sổ **Solution Explorer**, nháy phải lên file dữ liệu (*Quanlybanhang.mdf*), chọn **Exclude From Project**

– Thực hiện kết nối dữ liệu:



Trong **Server Explorer**, nháy phải **Data Connections**, chọn **Add Connection**, chọn **Microsoft SQL Server Database File**, nhấn **Continue**. Trong hộp thoại **Add Connection**, nhấn **Browse** tìm file dữ liệu (*Quanlybanhang.mdf*) (tìm đến thư mục Project của bạn). Nhấn **Test Connection**, nếu thành công sẽ xuất hiện thông báo "**Test connection succeeded**".

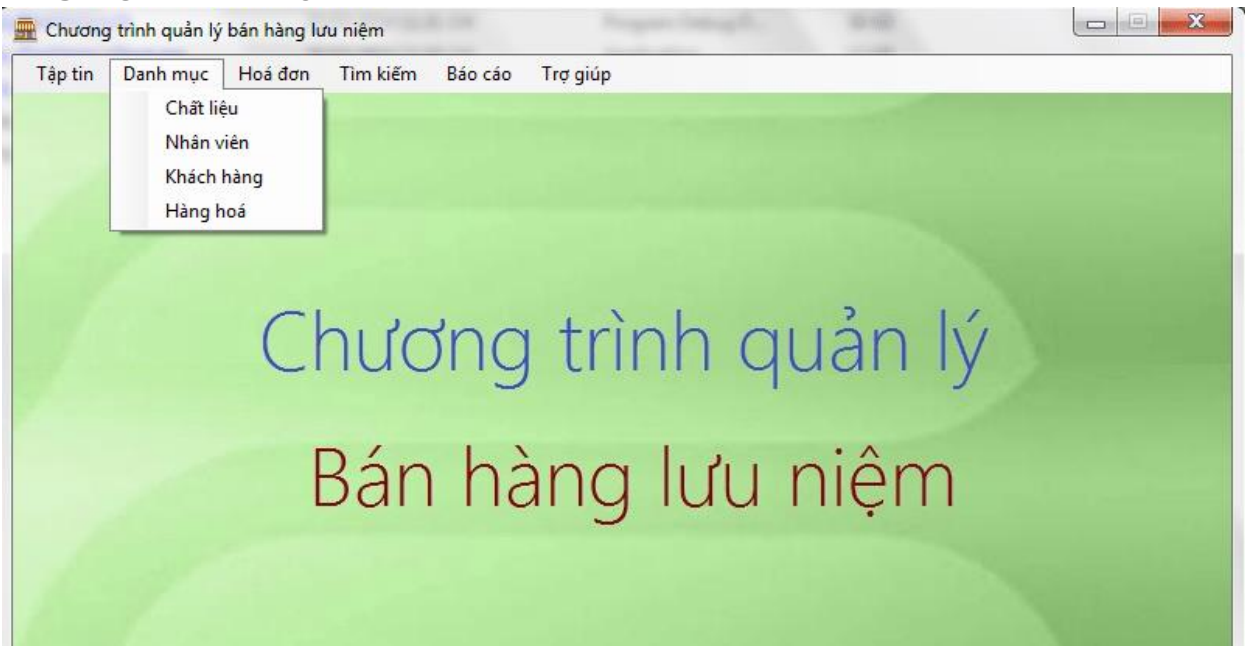
– Nên tạo thư mục chứa ảnh: nháy phải lên tên project **QuanLyBanHang**, chọn **Add**, chọn **New Folder**, đặt tên thư mục là *Images*. Nháy phải lên thư mục *Images*, chọn **Add**, chọn **Existing Item** và duyệt chọn các file ảnh, nhấn **Add** để thêm vào thư mục *Images*.

Xem video hướng dẫn

### 3. Thiết kế giao diện

#### a) Form chính

– Tên form: **frmMain**



Các thành phần của menu

Name	Text
mnuFile	Tập tin
mnuThoat	Thoát

mnuDanhMuc	Danh mục
mnuChatLieu	Chất liệu
mnuNhanVien	Nhân viên
mnuKhachHang	Khách hàng
mnuHangHoa	Hàng hoá
mnuHoaDon	Hoá đơn
mnuHoaDonBan	Hoá đơn bán
mnuTimKiem	Tìm kiếm
mnuFindHoaDon	Hoá đơn
mnuFindHang	Hàng
mnuFindKhachHang	Khách hàng
mnuBaoCao	Báo cáo
mnuBCHangTon	Hàng tồn
mnuBCDoanhThu	Doanh thu
mnuTroGiup	Trợ giúp
mnuHienTroGiup	Trợ giúp
mnuVaiNet	Vài nét

Xem Video hướng dẫn

## ***b) Form Danh mục chất liệu***

Tên form: **frmDMChatLieu**

Các thành phần trên form:

Điều khiển	Name	Text
------------	------	------

TextBox	txtMaChatLieu	
	txtTenChatLieu	
Button	btnThem	Thêm
	btnXoa	Xoá
	btnSua	Sửa
	btnLuu	Lưu
	btnBoqua	Bỏ qua
	btnDong	Đóng
DataGridView	dgvChatLieu	

Xem Video hướng dẫn

## c) Form Danh mục nhân viên

Tên form: **frmDMNhanvien**

Các thành phần trên form:

Điều khiển	Name	Text
TextBox	txtMaNhanVien	
	txtTenNhanVien	
	txtDiaChi	
Button	btnThem, btnXoa, btnSua, btnLuu, btnBoQua, btnDong	
CheckBox	chkGioiTinh	Nam
MaskedTextBox	mtbDienThoai	Mask: Phone Number
	mskNgaySinh	Mask: Short Date
DataGridView	dgvNhanVien	

Xem Video hướng dẫn

## ***d) Form Danh mục Khách Hàng***

Tên form: **frmDMKhachHang**

Các thành phần trên form:

Điều khiển	Name	Text
TextBox	txtMaKhach	
	txtTenKhach	
	txtDiaChi	
Button	btnThem, btnXoa, btnSua, btnLuu, btnBoQua, btnDong	
MaskedTextBox	mtbDienThoai	Mask: Phone Number
DataGridView	dgvKhachHang	

Xem Video hướng dẫn

## ***e) Form Danh mục hàng hoá***

Tên form: **frmDMHang**

Các thành phần trên form

Điều khiển	Name
TextBox	txtMaHang, txtTenHang, txtSoLuong, txtDonGiaNhap, txtDonGiaBan, txtAnh, txtGhiChu
ComboBox	cboMaChatLieu
PictureBox	picAnh (thuộc tính SizeMode = Zoom)
DataGridView	dgvHang
Button	btnThem, btnXoa, btnSua, btnLuu, btnBoQua, btnTimKiem, btnHienThi, btnDong, btnOpen

Xem Video hướng dẫn

## ***f) Form Hoá đơn bán hàng***

Tên form: **frmHoaDonBan**

Các thành phần trên form

Điều khiển	Name	Text
Label	lblBangChu	Bằng chữ:
TextBox	txtMaHDBan, txtNgayBan, txtTenNhanVien, txtTenKhach, txtDiaChi, txtDienThoai, txtTongTien, txtTenHang, txtDonGiaBan, txtSoLuong, txtGiamGia, txtThanhTien.	
ComboBox	cboMaNhanVien, cboMaKhach, cboMaHang, cboMaHDBan.	
DataGridView	dgvHDBanHang	
Button	btnNgay, btnThem, btnLuu, btnXoa, btnInHoaDon, btnDong, btnTimKiem	

Xem Video hướng dẫn

## g) Form Tìm kiếm hoá đơn

Tên form: **frmTimHDBan**

Các thành phần trên form

Điều khiển	Name
TextBox	txtMaHDBan, txtThang, txtNam, txtMaNhanVien, txtMaKhach, txtTongTien.
DataGridView	dgvTKHoaDon
Button	btnTimKiem, btnTimLai, btnDong

## Xem Video hướng dẫn

### Chú ý:

- Tất cả các form (trừ frmMain), thuộc tính StartPostion = CenterParent, ShowInTaskbar = False

## 4. Xử lý sự kiện

### 4.1. Lớp Functions

- Chứa các phương thức dùng chung
- Trong khung **Solution Explorer**, nháy phải lên tên project, chọn **Add -> New Folder**, đặt tên thư mục là **Class**
- Nháy phải thư mục **Class**, nháy phải chọn **Add -> Class**, đặt tên **Functions.cs**
- Trong class **Functions**:

## + Khai báo bổ sung các thư viện:

```
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;    // Sử dụng đối tượng MessageBox
```

## + Viết 2 phương thức: *Connect()* và *Disconnect()*

```
namespace QuanLyBanHang.Class
{
    class Functions
    {
        public static SqlConnection Con;    //Khai báo đối tượng kết nối

        public static void Connect()
        {
            Con = new SqlConnection();    //Khởi tạo đối tượng
            Con.ConnectionString = @"Data
Source=.\SQLEXPRESS;AttachDbFilename=" + Application.StartupPath +
@"\Quanlybanhang.mdf;Integrated Security=True;Connect Timeout=30;User
Instance=True";
            Con.Open();    //Mở kết nối
            //Kiểm tra kết nối
            if (Con.State == ConnectionState.Open)
                MessageBox.Show("Kết nối thành công");
            else MessageBox.Show("Không thể kết nối với dữ liệu");
        }

        public static void Disconnect()
        {
            if (Con.State == ConnectionState.Open)
            {
                Con.Close();    //Đóng kết nối
                Con.Dispose();    //Giải phóng tài nguyên
                Con = null;
            }
        }
    }
}
```

## 4.2. Form frmMain

### ***a) Sự kiện frmMain\_Load***

```
private void frmMain_Load(object sender, EventArgs e)
{
    Class.Functions.Connect(); //Mở kết nối
}
```

### ***b) Sự kiện mnuThoat\_Click***

```
private void mnuThoat_Click(object sender, EventArgs e)
{
    Class.Functions.Disconnect(); //Đóng kết nối
    Application.Exit(); //Thoát
}
```

### ***c) Hiển thị các form khác***

Cú pháp:

<Tên lớp form> <Tên đối tượng> = **new** <Tên lớp form>(); //Khởi tạo đối tượng

<Tên đối tượng>.ShowDialog(); //Hiển thị dưới dạng hộp thoại

Hoặc <Tên đối tượng>.Show(); //Hiện thị dạng thông thường

+ *Hiển thị form frmChatLieu*

```
private void mnuChatLieu_Click(object sender, EventArgs e)
{
    frmDMChatLieu frmChatLieu = new frmDMChatLieu(); //Khởi tạo
    đối tượng
    frmChatLieu.ShowDialog(); //Hiển thị
}
```

## 4.3. Form Chât liệu



## *a) Khai báo*

– Khai báo

```
using System.Data.SqlClient; //Sử dụng thư viện để làm việc SQL server
using QuanLyBanHang.Class; //Sử dụng class Functions.cs
```

– Khai báo biến toàn cục

```
DataTable tblCL; //Chứa dữ liệu bảng Chất liệu
```

## *b) Sự kiện **frmDMChatLieu\_Load***

```
private void frmDMChatLieu_Load(object sender, EventArgs e)
{
    txtMaChatLieu.Enabled = false;
    btnLuu.Enabled = false;
    btnBoQua.Enabled = false;
    LoadDataGridView(); //Hiển thị bảng tblChatLieu
}
```

Trong đó, phương thức LoadDataGridView có tác dụng lấy dữ liệu từ bảng tblChatLieu đổ vào DataGridView

## *c) Phương thức LoadDataGridView*

```
private void LoadDataGridView()
{
    string sql;
    sql = "SELECT MaChatLieu, TenChatLieu FROM tblChatLieu";
    tblCL = Class.Functions.GetDataToTable(sql); //Đọc dữ liệu từ
bảng

    dgvChatLieu.DataSource = tblCL; //Nguồn dữ liệu
    dgvChatLieu.Columns[0].HeaderText = "Mã chất liệu";
    dgvChatLieu.Columns[1].HeaderText = "Mã chất liệu";
}
```

```

        dgvChatLieu.Columns[0].Width = 100;
        dgvChatLieu.Columns[1].Width = 300;
        dgvChatLieu.AllowUserToAddRows = false; //Không cho người dùng
thêm dữ liệu trực tiếp
        dgvChatLieu.EditMode =
DataGridViewEditMode.EditProgrammatically; //Không cho sửa dữ liệu trực
tiếp
    }

```

Với *GetDataToTable* được viết trong lớp Functions có tác dụng thực hiện câu lệnh SQL truy vấn dữ liệu từ CSDL đổ vào đối tượng bảng.

### *d) Phương thức GetDataToTable*

Mở cửa sổ lớp Class Functions viết mã lệnh như sau:

```

//Lấy dữ liệu vào bảng
public static DataTable GetDataToTable(string sql)
{
    SqlDataAdapter dap = new SqlDataAdapter(); //Định nghĩa đối
tượng thuộc lớp SqlDataAdapter
    //Tạo đối tượng thuộc lớp SqlCommand
    dap.SelectCommand = new SqlCommand();
    dap.SelectCommand.Connection = Functions.Con; //Kết nối cơ sở
dữ liệu

    dap.SelectCommand.CommandText = sql; //Lệnh SQL
    //Khai báo đối tượng table thuộc lớp DataTable
    DataTable table = new DataTable();
    dap.Fill(table);
    return table;
}

```

Hoặc có thể thực hiện ngắn gọn bằng cách gán tham số khi khai báo đối tượng như sau:

```

//Lấy dữ liệu vào bảng
public static DataTable GetDataToTable(string sql)
{

```

```

        SqlDataAdapter dap = new SqlDataAdapter(sql, Con); //Định
nghĩa đối tượng thuộc lớp SqlDataAdapter
        //Khai báo đối tượng table thuộc lớp DataTable
        DataTable table = new DataTable();
        dap.Fill(table); //Đổ kết quả từ câu lệnh sql vào table
        return table;
    }

```

## Xem Video hướng dẫn

### *e) Phương thức dgvChatLieu\_Click*

Phương thức này có tác dụng lấy nội dung dòng dữ liệu người dùng chọn trong lưới DataGridView và hiển thị lên các điều khiển trên Form.

```

private void dgvChatLieu_Click(object sender, EventArgs e)
{
    if (btnThem.Enabled == false)
    {
        MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaChatLieu.Focus();
        return;
    }
    if (tblCL.Rows.Count == 0) //Nếu không có dữ liệu
    {
        MessageBox.Show("Không có dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    txtMaChatLieu.Text =
dgvChatLieu.CurrentRow.Cells["MaChatLieu"].Value.ToString();
    txtTenChatLieu.Text =
dgvChatLieu.CurrentRow.Cells["TenChatLieu"].Value.ToString();
    btnSua.Enabled = true;
    btnXoa.Enabled = true;
    btnBoQua.Enabled = true;
}

```

### *f) Phương thức btnThem\_Click*

```

private void btnThem_Click(object sender, EventArgs e)
{
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
    btnBoQua.Enabled = true;
    btnLuu.Enabled = true;
    btnThem.Enabled = false;
    ResetValue(); //Xoá trắng các textbox
    txtMaChatLieu.Enabled = true; //cho phép nhập mới
    txtMaChatLieu.Focus();
}

```

Với *ResetValues* là phương thức của form *frmDMChatLieu* có tác dụng xóa hết dữ liệu trong các điều khiển trên Form.

### *g) Phương thức ResetValues*

```

private void ResetValue()
{
    txtMaChatLieu.Text = "";
    txtTenChatLieu.Text = "";
}

```

### *h) Phương thức btnLuu\_Click*

Phương thức này có tác dụng kiểm tra thông tin người dùng nhập vào các điều khiển trên Form trong trường hợp thêm mới và lưu các thông tin đó vào CSDL.

```

private void btnLuu_Click(object sender, EventArgs e)
{
    string sql; //Lưu lệnh sql
    if (txtMaChatLieu.Text.Trim().Length == 0) //Nếu chưa nhập mã
    chất liệu
    {
        MessageBox.Show("Bạn phải nhập mã chất liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

```

        txtMaChatLieu.Focus();
        return;
    }
    if(txtTenChatLieu.Text.Trim().Length==0) //Nếu chưa nhập tên
chất liệu
    {
        MessageBox.Show("Bạn phải nhập tên chất liệu", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtTenChatLieu.Focus();
        return;
    }
    sql = "Select MaChatLieu From tblChatLieu where MaChatLieu=N'"
+ txtMaChatLieu.Text.Trim() + "'";
    if (Class.Functions.CheckKey(sql))
    {
        MessageBox.Show("Mã chất liệu này đã có, bạn phải nhập mã
khác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtMaChatLieu.Focus();
        return;
    }

    sql = "INSERT INTO tblChatLieu VALUES(N'" +
        txtMaChatLieu.Text + "',N'" + txtTenChatLieu.Text + "')";
    Class.Functions.RunSQL(sql); //Thực hiện câu lệnh sql
    LoadDataGridView(); //Nạp lại DataGridView
    ResetValue();
    btnXoa.Enabled = true;
    btnThem.Enabled = true;
    btnSua.Enabled = true;
    btnBoQua.Enabled = false;
    btnLuu.Enabled = false;
    txtMaChatLieu.Enabled = false;
}

```

Với *CheckKey* và *RunSQL* là các phương thức được viết trong lớp Functions. *CheckKey* có tác dụng kiểm tra khóa trùng, *RunSQL* có tác dụng thực thi các câu lệnh SQL.

### *i) Hàm CheckKey*

Mở cửa sổ lớp Class Functions viết mã lệnh:

```
//Hàm kiểm tra khoá trùng
public static bool CheckKey(string sql)
{
    SqlDataAdapter dap = new SqlDataAdapter(sql, Con);
    DataTable table = new DataTable();
    dap.Fill(table);
    if (table.Rows.Count > 0)
        return true;
    else return false;
}
```

## *j) Phương thức RunSQL*

Mở cửa sổ lớp Class Functions viết mã lệnh như sau:

```
//Hàm thực hiện câu lệnh SQL
public static void RunSQL(string sql)
{
    SqlCommand cmd; //Đối tượng thuộc lớp SqlCommand
    cmd = new SqlCommand();
    cmd.Connection = Con; //Gán kết nối
    cmd.CommandText = sql; //Gán lệnh SQL
    try
    {
        cmd.ExecuteNonQuery(); //Thực hiện câu lệnh SQL
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
    cmd.Dispose(); //Giải phóng bộ nhớ
    cmd = null;
}
```

**Chú ý:** Đối tượng SqlCommand có hai phương thức để thực thi câu lệnh SQL, trong đó:

- **ExecuteReader:** thực thi câu lệnh SQL có dữ liệu trả về, ví dụ SELECT.

- **ExecuteNoneQuery:** thực thi các câu lệnh SQL không yêu cầu trả về tập dữ liệu, ví dụ: INSERT, UPDATE, DELETE.

### *k) Phương thức btnSua\_Click*

Khi người dùng nháy chuột vào một dòng bản ghi bất kỳ trên lưới để hiển thị dữ liệu của bản ghi đó lên Form thì người dùng có thể chỉnh sửa các thông tin đó.

Phương thức *btnSua\_Click* có tác dụng lưu các thông tin người dùng đã sửa vào CSDL.

```
private void btnSua_Click(object sender, EventArgs e)
{
    string sql; //Lưu câu lệnh sql
    if (tblCL.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtMaChatLieu.Text == "") //nếu chưa chọn bản ghi nào
    {
        MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtTenChatLieu.Text.Trim().Length==0) //nếu chưa nhập tên
    chất liệu
    {
        MessageBox.Show("Bạn chưa nhập tên chất liệu", "Thông
        báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    sql = "UPDATE tblChatLieu SET TenChatLieu=N'" +
        txtTenChatLieu.Text.ToString() +
        "' WHERE MaChatLieu=N'" + txtMaChatLieu.Text + "'";
    Class.Functions.RunSQL(sql);
    LoadDataGridView();
    ResetValue();

    btnBoQua.Enabled = false;
}
```

## *l) Phương thức btnXoa\_Click*

```
private void btnXoa_Click(object sender, EventArgs e)
{
    string sql;
    if (tblCL.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtMaChatLieu.Text == "") //nếu chưa chọn bản ghi nào
    {
        MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (MessageBox.Show("Bạn có muốn xóa không?", "Thông báo",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        sql = "DELETE tblChatLieu WHERE MaChatLieu=N'" +
        txtMaChatLieu.Text + "'";
        Class.Functions.RunSqlDel(sql);
        LoadDataGridView();
        ResetValue();
    }
}
```

## *m) Phương thức RunSQLDel*

Phương thức RunSQLDel tương tự như RunSQL nhưng trong trường hợp xóa dữ liệu nếu dữ liệu đang được dùng bởi một đối tượng khác thì không được phép xóa.

Mở cửa sổ lớp Class Functions viết mã lệnh như sau:

```
public static void RunSqlDel(string sql)
```



```

{
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = Functions.Con;
    cmd.CommandText = sql;
    try
    {
        cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        //MessageBox.Show("Dữ liệu đang được dùng, không thể
        xoá...", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);
        MessageBox.Show(ex.ToString());
    }
    cmd.Dispose();
    cmd = null;
}

```

### *n) Phương thức btnBoQua\_Click*

Phương thức này được gọi khi người dùng muốn hủy bỏ các chức năng Thêm mới hoặc Sửa dữ liệu.

```

private void btnBoQua_Click(object sender, EventArgs e)
{
    ResetValue();
    btnBoQua.Enabled = false;
    btnThem.Enabled = true;
    btnXoa.Enabled = true;
    btnSua.Enabled = true;
    btnLuu.Enabled = false;
    txtMaChatLieu.Enabled = false;
}

```

### *o) Phương thức dùng phím Enter thay cho phím Tab*

```

private void txtMaChatLieu_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)

```

```
        SendKeys.Send("{TAB}");  
    }
```

Thực hiện tương tự cho txtTenChatLieu\_KeyUp

#### ***p) Phương thức btnDong\_Click***

```
private void btnDong_Click(object sender, EventArgs e)  
{  
    this.Close();  
}
```

Xem video hướng dẫn phần trên  
**4.4. Form Danh mục Nhân viên**

#### ***a) Khai báo***

– Thư viện:

```
using System.Data;  
using System.Data.SqlClient;  
using QuanLyBanHang.Class;
```

– ***Biến:***

```
DataTable tblNV; //Lưu dữ liệu bảng nhân viên
```

#### ***b) Phương thức frmDMNhanvien\_Load***

```
private void frmDMNhanvien_Load(object sender, EventArgs e)  
{  
    txtMaNhanVien.Enabled = false;  
    btnLuu.Enabled = false;  
    btnBoQua.Enabled = false;  
    LoadDataGridView();  
}
```

### *c) Phương thức LoadDataGridView – Hiển thị dữ liệu lên lưới*

```
public void LoadDataGridView()
{
    string sql;
    sql = "SELECT
MaNhanVien,TenNhanVien,GioiTinh,DiaChi,DienThoai,NgaySinh FROM
tblNhanVien";
    tblNV = Functions.GetDataToTable(sql); //lấy dữ liệu
    dgvNhanVien.DataSource = tblNV;
    dgvNhanVien.Columns[0].HeaderText = "Mã nhân viên";
    dgvNhanVien.Columns[1].HeaderText = "Tên nhân viên";
    dgvNhanVien.Columns[2].HeaderText = "Giới tính";
    dgvNhanVien.Columns[3].HeaderText = "Địa chỉ";
    dgvNhanVien.Columns[4].HeaderText = "Điện thoại";
    dgvNhanVien.Columns[5].HeaderText = "Ngày sinh";
    dgvNhanVien.Columns[0].Width = 100;
    dgvNhanVien.Columns[1].Width = 150;
    dgvNhanVien.Columns[2].Width = 100;
    dgvNhanVien.Columns[3].Width = 150;
    dgvNhanVien.Columns[4].Width = 100;
    dgvNhanVien.Columns[5].Width = 100;
    dgvNhanVien.AllowUserToAddRows = false;
    dgvNhanVien.EditMode =
DataGridViewEditMode.EditProgrammatically;
}
```

### *d) Phương thức dgvNhanVien\_Click*

```
private void dgvNhanVien_Click(object sender, EventArgs e)
{
    if (btnThem.Enabled == false)
    {
        MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaNhanVien.Focus();
        return;
    }
}
```

```

    }
    if (tblNV.Rows.Count == 0)
    {
        MessageBox.Show("Không có dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    txtMaNhanVien.Text =
    dgvNhanVien.CurrentRow.Cells["MaNhanVien"].Value.ToString();
    txtTenNhanVien.Text =
    dgvNhanVien.CurrentRow.Cells["TenNhanVien"].Value.ToString();
    if (dgvNhanVien.CurrentRow.Cells["GioiTinh"].Value.ToString()
    == "Nam") chkGioiTinh.Checked = true;
    else chkGioiTinh.Checked = false;
    txtDiaChi.Text =
    dgvNhanVien.CurrentRow.Cells["DiaChi"].Value.ToString();
    mtbDienThoai.Text =
    dgvNhanVien.CurrentRow.Cells["DienThoai"].Value.ToString();
    mskNgaySinh.Text =
    dgvNhanVien.CurrentRow.Cells["NgaySinh"].Value.ToString();
    btnSua.Enabled = true;
    btnXoa.Enabled = true;
    btnXoa.Enabled = true;
}

```

### *e) Phương thức btnThem\_Click*

```

private void btnThem_Click(object sender, EventArgs e)
{
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
    btnBoQua.Enabled = true;
    btnLuu.Enabled = true;
    btnThem.Enabled = false;
    ResetValues();
    txtMaNhanVien.Enabled = true;
    txtMaNhanVien.Focus();
}

```

### *f) Phương thức ResetValues*

```

private void ResetValues()
{
    txtMaNhanVien.Text = "";
    txtTenNhanVien.Text = "";
    chkGioiTinh.Checked = false;
    txtDiaChi.Text = "";
    mskNgaySinh.Text = "";
    mtbDienThoai.Text = "";
}

```

### *g) Phương thức btnLuu\_Click*

```

private void btnLuu_Click(object sender, EventArgs e)
{
    string sql,gt;
    if (txtMaNhanVien.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập mã nhân viên", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtMaNhanVien.Focus();
        return;
    }
    if (txtTenNhanVien.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập tên nhân viên", "Thông
        báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtTenNhanVien.Focus();
        return ;
    }
    if (txtDiaChi.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtDiaChi.Focus();
        return ;
    }
    if (mtbDienThoai.Text == "( ) -")
    {
        MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

```

        mtbDienThoai.Focus();
        return ;
    }
    if (mskNgaySinh.Text == " / ")
    {
        MessageBox.Show("Bạn phải nhập ngày sinh", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        mskNgaySinh.Focus();
        return ;
    }
    if (!Functions.IsDate(mskNgaySinh.Text))
    {
        MessageBox.Show("Bạn phải nhập lại ngày sinh", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        // mskNgaySinh.Text = "";
        mskNgaySinh.Focus();
        return ;
    }
    if (chkGioiTinh.Checked == true)
        gt = "Nam";
    else
        gt = "Nữ";
    sql = "SELECT MaNhanVien FROM tblNhanVien WHERE MaNhanVien=N'"
+ txtMaNhanVien.Text.Trim() + "'";
    if (Functions.CheckKey(sql))
    {
        MessageBox.Show("Mã nhân viên này đã có, bạn phải nhập mã
khác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtMaNhanVien.Focus();
        txtMaNhanVien.Text = "";
        return ;
    }
    sql = "INSERT INTO
tblNhanVien(MaNhanVien,TenNhanVien,GioiTinh, DiaChi,DienThoai, NgaySinh)
VALUES (N'" + txtMaNhanVien.Text.Trim() + "',N'" +
txtTenNhanVien.Text.Trim() + "',N'" + gt + "',N'" + txtDiaChi.Text.Trim()
+ "',''" + mtbDienThoai.Text + "',''" +
Functions.ConvertDateTime(mskNgaySinh.Text) + "')";
    Functions.RunSQL(sql);
    LoadDataGridView();
    ResetValues();
    btnXoa.Enabled = true;
    btnThem.Enabled = true;
    btnSua.Enabled = true;
    btnBoQua.Enabled = false;

```

```
        btnLuu.Enabled = false;  
        txtMaNhanVien.Enabled = false;  
    }
```

Với *IsDate* và *ConvertDateTime* là các hàm được viết trong lớp Functions  
*IsDate* có tác dụng kiểm tra một biến có ở dạng ngày tháng không,  
*ConvertDateTime* có tác dụng đổi một chuỗi ngày tháng do người dùng nhập  
có dạng dd/mm/yyyy thành chuỗi ngày tháng có dạng mm/dd/yyyy để lưu  
vào CSDL.

### *h) Hàm IsDate*

Soạn thảo trong lớp Functions:

```
public static bool IsDate(string date)  
{  
    string[] elements = date.Split('/');  
    if ((Convert.ToInt32(elements[0]) >= 1) &&  
        (Convert.ToInt32(elements[0]) <= 31) && (Convert.ToInt32(elements[1]) >=  
1) && (Convert.ToInt32(elements[1]) <= 12) &&  
        (Convert.ToInt32(elements[2]) >= 1900))  
        return true;  
    else return false;  
}
```

### *i) Hàm ConvertDateTime*

Soạn thảo hàm trong lớp Functions:

```
public static string ConvertDateTime(string date)  
{  
    string[] elements = date.Split('/');  
    string dt = string.Format("{0}/{1}/{2}", elements[0],  
elements[1], elements[2]);  
    return dt;  
}
```

## *j) Phương thức btnSua\_Click*

```
private void btnSua_Click(object sender, EventArgs e)
{
    string sql, gt;
    if (tblNV.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtMaNhanVien.Text == "")
    {
        MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông
        báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtTenNhanVien.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập tên nhân viên", "Thông
        báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtTenNhanVien.Focus();
        return;
    }
    if (txtDiaChi.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtDiaChi.Focus();
        return;
    }
    if (mtbDienThoai.Text == "(   )    -")
    {
        MessageBox.Show("Bạn phải nhập điện thoại", "Thông
        báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        mtbDienThoai.Focus();
        return;
    }
    if (mskNgaySinh.Text == " / /")
    {

```



```

        MessageBox.Show("Bạn phải nhập ngày sinh", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        mskNgaySinh.Focus();
        return;
    }
    if (!Functions.IsDate(mskNgaySinh.Text))
    {
        MessageBox.Show("Bạn phải nhập lại ngày sinh", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        mskNgaySinh.Text = "";
        mskNgaySinh.Focus();
        return;
    }
    if (chkGioiTinh.Checked == true)
        gt = "Nam";
    else
        gt = "Nữ";
    sql = "UPDATE tblNhanVien SET TenNhanVien=N'" + txtTenNhanVien.Text.Trim().ToString() +
        "',DiaChi=N'" + txtDiaChi.Text.Trim().ToString() +
        "',DienThoai='" + mtbDienThoai.Text.ToString() +
        "',GioiTinh=N'" + gt +
        "',NgaySinh='" +
        Functions.ConvertDateTime(mskNgaySinh.Text) +
        "' WHERE MaNhanVien=N'" + txtMaNhanVien.Text + "'";
    Functions.RunSQL(sql);
    LoadDataGridView();
    ResetValues();
    btnBoQua.Enabled = false;
}

```

### *k) Phương thức btnXoa\_Click*

```

private void btnXoa_Click(object sender, EventArgs e)
{
    string sql;
    if (tblNV.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
}

```

```

        if (txtMaNhanVien.Text == "")
        {
            MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
            return;
        }
        if (MessageBox.Show("Bạn có muốn xóa không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)
        {
            sql = "DELETE tblNhanVien WHERE MaNhanVien=N'" + txtMaNhanVien.Text + "'";
            Functions.RunSqlDel(sql);
            LoadDataGridView();
            ResetValues();
        }
    }
}

```

### *l) Phương thức btnBoQua\_Click*

```

private void btnBoQua_Click(object sender, EventArgs e)
{
    ResetValues();
    btnBoQua.Enabled = false;
    btnThem.Enabled = true;
    btnXoa.Enabled = true;
    btnSua.Enabled = true;
    btnLuu.Enabled = false;
    txtMaNhanVien.Enabled = false;
}

```

### *m) Phương thức dùng phím Enter thay cho phím Tab*

```

private void txtMaNhanVien_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        SendKeys.Send("{TAB}");
}

```

Thực hiện tương tự cho txtTenNhanVien.KeyUp, txtDiaChi.KeyUp, mtbDienThoai.KeyUp, mskNgaySinh.KeyUp.

### *n) Phương thức btnDong\_Click*

```
private void btnDong_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Xem video hướng dẫn chi tiết  
4.5. Form Danh mục Khách hàng

### *a) Khai báo*

– Thư viện

```
using System.Data;
using System.Data.SqlClient;
using Quanlybanhang.Class;
```

– Biến

```
DataTable tblKH; //Bảng khách hàng
```

### *b) Phương thức frmDMKhachHang\_Load*

```
private void frmDMKhachHang_Load(object sender, EventArgs e)
{
    txtMaKhach.Enabled = false;
    btnLuu.Enabled = false;
    btnBoQua.Enabled = false;
    LoadDataGridView();
}
```

### *c) Phương thức LoadDataGridView*

```
private void LoadDataGridView()
{
    string sql;
    sql = "SELECT * from tblKhach";
    tblKH = Functions.GetDataToTable(sql); //Lấy dữ liệu từ bảng
    dgvKhachHang.DataSource = tblKH; //Hiển thị vào dataGridView
    dgvKhachHang.Columns[0].HeaderText = "Mã khách";
    dgvKhachHang.Columns[1].HeaderText = "Tên khách";
    dgvKhachHang.Columns[2].HeaderText = "Địa chỉ";
    dgvKhachHang.Columns[3].HeaderText = "Điện thoại";
    dgvKhachHang.Columns[0].Width = 100;
    dgvKhachHang.Columns[1].Width = 150;
    dgvKhachHang.Columns[2].Width = 150;
    dgvKhachHang.Columns[3].Width = 150;
    dgvKhachHang.AllowUserToAddRows = false;
    dgvKhachHang.EditMode =
DataGridViewEditMode.EditProgrammatically;
}
```

### *d) Phương thức dgvKhachHang\_Click*

```
private void dgvKhachHang_Click(object sender, EventArgs e)
{
    if (btnThem.Enabled == false)
    {
        MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaKhach.Focus();
        return;
    }
    if (tblKH.Rows.Count == 0)
    {
        MessageBox.Show("Không có dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    txtMaKhach.Text =
dgvKhachHang.CurrentRow.Cells["MaKhach"].Value.ToString();
}
```

```

        txtTenKhach.Text =
dgvKhachHang.CurrentRow.Cells["TenKhach"].Value.ToString();
        txtDiaChi.Text =
dgvKhachHang.CurrentRow.Cells["DiaChi"].Value.ToString();
        mtbDienThoai.Text =
dgvKhachHang.CurrentRow.Cells["DienThoai"].Value.ToString();
        btnSua.Enabled = true;
        btnXoa.Enabled = true;
        btnBoQua.Enabled = true;
    }

```

### *e) Phương thức btnThem\_Click*

```

private void btnThem_Click(object sender, EventArgs e)
{
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
    btnBoQua.Enabled = true;
    btnLuu.Enabled = true;
    btnThem.Enabled = false;
    ResetValues();
    txtMaKhach.Enabled = true;
    txtMaKhach.Focus();
}

```

### *f) Phương thức ResetValues*

```

private void ResetValues()
{
    txtMaKhach.Text = "";
    txtTenKhach.Text = "";
    txtDiaChi.Text = "";
    mtbDienThoai.Text = "";
}

```

### *g) Phương thức btnLuu\_Click*

```

private void btnLuu_Click(object sender, EventArgs e)
{
    string sql;
    if (txtMaKhach.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập mã khách", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaKhach.Focus();
        return;
    }
    if (txtTenKhach.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập tên khách", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtTenKhach.Focus();
        return;
    }
    if (txtDiaChi.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtDiaChi.Focus();
        return;
    }
    if (mtbDienThoai.Text == "( ) -")
    {
        MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        mtbDienThoai.Focus();
        return;
    }
    //Kiểm tra đã tồn tại mã khách chưa
    sql = "SELECT MaKhach FROM tblKhach WHERE MaKhach=N'" +
    txtMaKhach.Text.Trim() + "'";
    if (Functions.CheckKey(sql))
    {
        MessageBox.Show("Mã khách này đã tồn tại", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaKhach.Focus();
        return;
    }
    //Chèn thêm
    sql = "INSERT INTO tblKhach VALUES (N'" +
    txtMaKhach.Text.Trim() +

```

```

            "','" + txtTenKhach.Text.Trim() + "','" +
txtDiaChi.Text.Trim() + "','" + mtbDienThoai.Text + "')";
        Functions.RunSQL(sql);
        LoadDataGridView();
        ResetValues();

        btnXoa.Enabled = true;
        btnThem.Enabled = true;
        btnSua.Enabled = true;
        btnBoQua.Enabled = false;
        btnLuu.Enabled = false;
        txtMaKhach.Enabled = false;
    }

```

## *h) Phương thức btnSua\_Click*

```

private void btnSua_Click(object sender, EventArgs e)
{
    string sql;
    if (tblKH.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtMaKhach.Text == "")
    {
        MessageBox.Show("Bạn phải chọn bản ghi cần sửa", "Thông
        báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtTenKhach.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập tên khách", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtTenKhach.Focus();
        return;
    }
    if (txtDiaChi.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

```

        txtDiaChi.Focus();
        return;
    }
    if (mtbDienThoai.Text=="( ) -")
    {
        MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        mtbDienThoai.Focus();
        return;
    }
    sql = "UPDATE tblKhach SET TenKhach=N'" +
    txtTenKhach.Text.Trim().ToString() + "',DiaChi=N'" +
        txtDiaChi.Text.Trim().ToString() + "',DienThoai='" +
    mtbDienThoai.Text.ToString() +
        "' WHERE MaKhach=N'" + txtMaKhach.Text + "'";
    Functions.RunSQL(sql);
    LoadDataGridView();
    ResetValues();
    btnBoQua.Enabled = false;
}

```

### *i) Phương thức btnXoa\_Click*

```

private void btnXoa_Click(object sender, EventArgs e)
{
    string sql;
    if (tblKH.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtMaKhach.Text.Trim() == "")
    {
        MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (MessageBox.Show("Bạn có muốn xóa bản ghi này không?",
    "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
    DialogResult.Yes)
    {

```



```

        sql = "DELETE tblKhach WHERE MaKhach=N'" + txtMaKhach.Text
+ "'";

        Functions.RunSqlDel(sql);
        LoadDataGridView();
        ResetValues();
    }
}

```

### *j) Phương thức btnBoQua\_Click*

```

private void btnBoQua_Click(object sender, EventArgs e)
{
    ResetValues();
    btnBoQua.Enabled = false;
    btnThem.Enabled = true;
    btnXoa.Enabled = true;
    btnSua.Enabled = true;
    btnLuu.Enabled = false;
    txtMaKhach.Enabled = false;
}

```

### *k) Phương thức dùng phím Enter thay cho phím Tab*

```

private void txtMaKhach_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        SendKeys.Send("{TAB}");
}

```

### *l) Phương thức btnDong\_Click*

```

private void btnDong_Click(object sender, EventArgs e)
{
    this.Close();
}

```

# Xem video hướng dẫn Quản lý Khách Hàng

## 4.6. Form Danh mục Hàng hóa

### *a) Khai báo*

– Thư viện

```
using System.Data;  
using System.Data.SqlClient;  
using Quanlybanhang.Class;
```

– Biến

```
DataTable tblH; //Bảng hàng
```

### *b) Phương thức frmDMHang\_Load*

```
private void frmDMHang_Load(object sender, EventArgs e)  
{  
    string sql;  
    sql = "SELECT * from tblChatLieu";  
    txtMaHang.Enabled = false;  
    btnLuu.Enabled = false;  
    btnBoQua.Enabled = false;  
    LoadDataGridView();  
    Functions.FillCombo(sql, cboMaChatLieu, "MaChatLieu",  
"TenChatLieu");  
    cboMaChatLieu.SelectedIndex = -1;  
    ResetValues();  
}
```

Với *FillCombo* là một phương thức được viết trong Class Functions, có tác dụng lấy dữ liệu từ một câu lệnh SQL đổ vào một ComboBox.

### *c) Phương thức FillCombo*

Soạn thảo trong Class Functions :

```

public static void FillCombo(string sql, ComboBox cbo, string ma, string
ten)
{
    SqlDataAdapter dap = new SqlDataAdapter(sql, Con);
    DataTable table = new DataTable();
    dap.Fill(table);
    cbo.DataSource = table;
    cbo.ValueMember = ma; //Trường giá trị
    cbo.DisplayMember = ten; //Trường hiển thị
}

```

### *d) Phương thức ResetValues*

```

private void ResetValues()
{
    txtMaHang.Text = "";
    txtTenHang.Text = "";
    cboMaChatLieu.Text = "";
    txtSoLuong.Text = "0";
    txtDonGiaNhap.Text = "0";
    txtDonGiaBan.Text = "0";
    txtSoLuong.Enabled = true ;
    txtDonGiaNhap.Enabled = false;
    txtDonGiaBan.Enabled = false;
    txtAnh.Text = "";
    picAnh.Image = null;
    txtGhichu.Text = "";
}

```

### *e) Phương thức LoadDataGridView*

```

private void LoadDataGridView()
{
    string sql;
    sql = "SELECT * from tblHang";
    tblH = Functions.GetDataToTable(sql);
    dgvHang.DataSource = tblH;
}

```

```

dgvHang.Columns[0].HeaderText = "Mã hàng";
dgvHang.Columns[1].HeaderText = "Tên hàng";
dgvHang.Columns[2].HeaderText = "Chất liệu";
dgvHang.Columns[3].HeaderText = "Số lượng";
dgvHang.Columns[4].HeaderText = "Đơn giá nhập";
dgvHang.Columns[5].HeaderText = "Đơn giá bán";
dgvHang.Columns[6].HeaderText = "Ảnh";
dgvHang.Columns[7].HeaderText = "Ghi chú";
dgvHang.Columns[0].Width = 80;
dgvHang.Columns[1].Width = 140;
dgvHang.Columns[2].Width = 80;
dgvHang.Columns[3].Width = 80;
dgvHang.Columns[4].Width = 100;
dgvHang.Columns[5].Width = 100;
dgvHang.Columns[6].Width = 200;
dgvHang.Columns[7].Width = 300;
dgvHang.AllowUserToAddRows = false;
dgvHang.EditMode = DataGridViewEditMode.EditProgrammatically;
}

```

## *f) Phương thức dgvHang\_Click*

```

private void dgvHang_Click(object sender, EventArgs e)
{
    string MaChatLieu;
    string sql;
    if (btnThem.Enabled == false)
    {
        MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaHang.Focus();
        return;
    }
    if (tblH.Rows.Count == 0)
    {
        MessageBox.Show("Không có dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    txtMaHang.Text =
dgvHang.CurrentRow.Cells["MaHang"].Value.ToString();
}

```

```

        txtTenHang.Text =
dgvHang.CurrentRow.Cells["TenHang"].Value.ToString();
        MaChatLieu =
dgvHang.CurrentRow.Cells["MaChatLieu"].Value.ToString();
        sql = "SELECT TenChatLieu FROM tblChatLieu WHERE
MaChatLieu=N'" + MaChatLieu + "'";
        cboMaChatLieu.Text = Functions.GetFieldValues(sql);
        txtSoLuong.Text =
dgvHang.CurrentRow.Cells["SoLuong"].Value.ToString();
        txtDonGiaNhap.Text =
dgvHang.CurrentRow.Cells["DonGiaNhap"].Value.ToString();
        txtDonGiaBan.Text =
dgvHang.CurrentRow.Cells["DonGiaBan"].Value.ToString();
        sql = "SELECT Anh FROM tblHang WHERE MaHang=N'" +
txtMaHang.Text + "'";
        txtAnh.Text = Functions.GetFieldValues(sql);
        picAnh.Image = Image.FromFile(txtAnh.Text);
        sql = "SELECT Ghichu FROM tblHang WHERE MaHang = N'" +
txtMaHang.Text + "'";
        txtGhichu.Text = Functions.GetFieldValues(sql);
        btnSua.Enabled = true;
        btnXoa.Enabled = true;
        btnBoQua.Enabled = true;
    }

```

Với *GetFieldValues* là một hàm được viết trong Class Functions, có tác dụng lấy dữ liệu từ một câu lệnh SQL.

### ***g) Hàm GetFieldValues***

Soạn thảo trong Class Functions

```

public static string GetFieldValues(string sql)
{
    string ma = "";
    SqlCommand cmd = new SqlCommand(sql, Con);
    SqlDataReader reader;
    reader = cmd.ExecuteReader();
    while (reader.Read())
        ma = reader.GetValue(0).ToString();
    reader.Close();
    return ma;
}

```

## *h) Phương thức btnThem\_Click*

```
private void btnThem_Click(object sender, EventArgs e)
{
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
    btnBoQua.Enabled = true;
    btnLuu.Enabled = true;
    btnThem.Enabled = false;
    ResetValues();
    txtMaHang.Enabled = true;
    txtMaHang.Focus();
    txtSoLuong.Enabled = true;
    txtDonGiaNhap.Enabled = true;
    txtDonGiaBan.Enabled = true;
}
```

## *i) Phương thức btnLuu\_Click*

```
private void btnLuu_Click(object sender, EventArgs e)
{
    string sql;
    if (txtMaHang.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập mã hàng", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaHang.Focus();
        return;
    }
    if (txtTenHang.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập tên hàng", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtTenHang.Focus();
        return;
    }
    if (cboMaChatLieu.Text.Trim().Length == 0)
    {

```

```

        MessageBox.Show("Bạn phải nhập chất liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        cboMaChatLieu.Focus();
        return;
    }
    if (txtAnh.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải chọn ảnh minh hoạ cho hàng",
        "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        btnOpen.Focus();
        return;
    }
    sql = "SELECT MaHang FROM tblHang WHERE MaHang=N'" +
    txtMaHang.Text.Trim() + "'";
    if (Functions.CheckKey(sql))
    {
        MessageBox.Show("Mã hàng này đã tồn tại, bạn phải chọn mã
        hàng khác", "Thông báo", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        txtMaHang.Focus();
        return;
    }
    sql = "INSERT INTO
    tblHang (MaHang, TenHang, MaChatLieu, SoLuong, DonGiaNhap,
    DonGiaBan, Anh, Ghichu) VALUES (N'"
    + txtMaHang.Text.Trim() + "',N'" + txtTenHang.Text.Trim()
    +
    "',N'" + cboMaChatLieu.SelectedValue.ToString() +
    "'," + txtSoLuong.Text.Trim() + "," + txtDonGiaNhap.Text +
    "," + txtDonGiaBan.Text + ",'" + txtAnh.Text + "',N'" +
    txtGhichu.Text.Trim() + "')";

    Functions.RunSQL(sql);
    LoadDataGridView();
    //ResetValues();
    btnXoa.Enabled=true;
    btnThem.Enabled = true;
    btnSua.Enabled = true;
    btnBoQua.Enabled = false;
    btnLuu.Enabled = false;
    txtMaHang.Enabled = false;
}

```

## *j) Phương thức btnSua\_Click*

```
private void btnSua_Click(object sender, EventArgs e)
{
    string sql;
    if (tblH.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtMaHang.Text == "")
    {
        MessageBox.Show("Bạn chưa chọn bàn ghi nào", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtMaHang.Focus();
        return;
    }
    if (txtTenHang.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập tên hàng", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtTenHang.Focus();
        return;
    }
    if (cboMaChatLieu.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải nhập chất liệu", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        cboMaChatLieu.Focus();
        return;
    }
    if (txtAnh.Text.Trim().Length == 0)
    {
        MessageBox.Show("Bạn phải ảnh minh họa cho hàng", "Thông
        báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        txtAnh.Focus();
        return;
    }
    sql = "UPDATE tblHang SET TenHang=N'" +
    txtTenHang.Text.Trim().ToString() +
    "',MaChatLieu=N'" + cboMaChatLieu.SelectedValue.ToString()
    +
```



```

        "',SoLuong="+txtSoLuong.Text+
        "',Anh='" + txtAnh.Text + "',Ghichu=N'" + txtGhichu.Text
+ "' WHERE MaHang=N'" + txtMaHang.Text + "'";
        Functions.RunSQL(sql);
        LoadDataGridView();
        ResetValues();
        btnBoQua.Enabled = false;
    }

```

### *k) Phương thức btnXoa\_Click*

```

private void btnXoa_Click(object sender, EventArgs e)
{
    string sql;
    if (tblH.Rows.Count == 0)
    {
        MessageBox.Show("Không còn dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (txtMaHang.Text == "")
    {
        MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (MessageBox.Show("Bạn có muốn xóa bản ghi này không?",
    "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
    DialogResult.Yes)
    {
        sql = "DELETE tblHang WHERE MaHang=N'" + txtMaHang.Text +
        "'";

        Functions.RunSqlDel(sql);
        LoadDataGridView();
        ResetValues();
    }
}

```

### *l) Phương thức btnBoQua\_Click*

```
private void btnBoQua_Click(object sender, EventArgs e)
{
    ResetValues();
    btnXoa.Enabled = true;
    btnSua.Enabled = true;
    btnThem.Enabled = true;
    btnBoQua.Enabled = false;
    btnLuu.Enabled = false;
    txtMaHang.Enabled = false;
}
```

### *m) Phương thức btnOpen\_Click*

```
private void btnOpen_Click(object sender, EventArgs e)
{
    OpenFileDialog dlgOpen = new OpenFileDialog();
    dlgOpen.Filter =
    "Bitmap(*.bmp)|*.bmp|JPEG(*.jpg)|*.jpg|GIF(*.gif)|*.gif|All
    files(*.*)|*.*";
    dlgOpen.FilterIndex = 2;
    dlgOpen.Title = "Chọn ảnh minh họa cho sản phẩm";
    if (dlgOpen.ShowDialog() == DialogResult.OK)
    {
        picAnh.Image = Image.FromFile(dlgOpen.FileName);
        txtAnh.Text = dlgOpen.FileName;
    }
}
```

### *n) Phương thức btnTimKiem\_Click*

Phương thức tìm kiếm cho phép tìm các bản ghi thỏa mãn một số điều kiện nào đấy và hiển thị kết quả tìm được vào lưới DataGridView.

```
private void btnTimKiem_Click(object sender, EventArgs e)
{
    string sql;
```

```

        if ((txtMaHang.Text == "") && (txtTenHang.Text == "") &&
(cboMaChatLieu.Text == ""))
        {
            MessageBox.Show("Bạn hãy nhập điều kiện tìm kiếm", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        sql = "SELECT * from tblHang WHERE 1=1";
        if (txtMaHang.Text != "")
            sql += " AND MaHang LIKE N'" + txtMaHang.Text + "'";
        if (txtTenHang.Text != "")
            sql += " AND TenHang LIKE N'" + txtTenHang.Text + "'";
        if (cboMaChatLieu.Text!="")
            sql += " AND MaChatLieu LIKE N'" +
cboMaChatLieu.SelectedValue + "'";
        tblH = Functions.GetDataToTable(sql);
        if (tblH.Rows.Count == 0)
            MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm
kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        else MessageBox.Show("Có " + tblH.Rows.Count + " bản ghi thoả
mãn điều kiện!", "Thông báo", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        dgvHang.DataSource = tblH;
        ResetValues();
    }

```

### *o) Phương thức btnHienThi\_Click*

```

private void btnHienThi_Click(object sender, EventArgs e)
{
    string sql;
    sql = "SELECT
MaHang,TenHang,MaChatLieu,SoLuong,DonGiaNhap,DonGiaBan,Anh,Ghichu FROM
tblHang";

    tblH = Functions.GetDataToTable(sql);
    dgvHang.DataSource = tblH;
}

```

### *p) Phương thức btnDong\_Click*

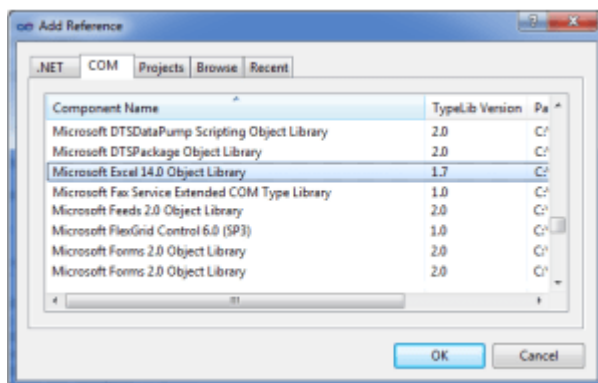
```
private void btnDong_Click(object sender, EventArgs e)
{
    this.Close();
}
```

## Xem Video hướng dẫn Quản lý Hàng Hoá

### 4.7. Form Hóa đơn bán

#### *a) Khai báo*

Chọn menu Project/Add Reference, chọn thẻ COM hộp thoại Add Reference, chọn Microsoft Excel 14.0 Object Library, nhấn OK.



– Thư viện

```
using System.Data.SqlClient;
using QuanLyBanHang.Class;
using COMExcel = Microsoft.Office.Interop.Excel;
```

– Biến

```
DataTable tblCTHDB; //Bảng chi tiết hoá đơn bán
```

#### *b) Phương thức frmHoaDonBan\_Load*

```
private void frmHoaDonBan_Load(object sender, EventArgs e)
{
```

```

        btnThem.Enabled = true;
        btnLuu.Enabled = false;
        btnXoa.Enabled = false;
        btnInHoaDon.Enabled = false;
        txtMaHDBan.ReadOnly = true;
        txtTenNhanVien.ReadOnly = true;
        txtTenKhach.ReadOnly = true;
        txtDiaChi.ReadOnly = true;
        txtDienThoai.ReadOnly = true;
        txtTenHang.ReadOnly = true;
        txtDonGiaBan.ReadOnly = true;
        txtThanhTien.ReadOnly = true;
        txtTongTien.ReadOnly = true;
        txtGiamGia.Text = "0";
        txtTongTien.Text = "0";
        Functions.FillCombo("SELECT MaKhach, TenKhach FROM tblKhach",
cboMaKhach, "MaKhach", "MaKhach");
        cboMaKhach.SelectedIndex = -1;
        Functions.FillCombo("SELECT MaNhanVien, TenNhanVien FROM
tblNhanVien",cboMaNhanVien, "MaNhanVien", "TenKhach");
        cboMaNhanVien.SelectedIndex = -1;
        Functions.FillCombo("SELECT MaHang, TenHang FROM tblHang",
cboMaHang, "MaHang", "MaHang");
        cboMaHang.SelectedIndex = -1;
        //Hiển thị thông tin của một hóa đơn được gọi từ form tìm kiếm
        if (txtMaHDBan.Text != "")
        {
            LoadInfoHoaDon();
            btnXoa.Enabled = true;
            btnInHoaDon.Enabled = true;
        }
        LoadDataGridView();
    }

```

### *c) Phương thức LoadDataGridView*

```

private void LoadDataGridView()
{
    string sql;
    sql = "SELECT a.MaHang, b.TenHang, a.SoLuong, b.DonGiaBan,
a.GiamGia,a.ThanhTien FROM tblChiTietHDBan AS a, tblHang AS b WHERE
a.MaHDBan = N'" + txtMaHDBan.Text + "' AND a.MaHang=b.MaHang";

```

```

tblCTHDB = Functions.GetDataToTable(sql);
dgvHDBanHang.DataSource = tblCTHDB;
dgvHDBanHang.Columns[0].HeaderText = "Mã hàng";
dgvHDBanHang.Columns[1].HeaderText = "Tên hàng";
dgvHDBanHang.Columns[2].HeaderText = "Số lượng";
dgvHDBanHang.Columns[3].HeaderText = "Đơn giá";
dgvHDBanHang.Columns[4].HeaderText = "Giảm giá %";
dgvHDBanHang.Columns[5].HeaderText = "Thành tiền";
dgvHDBanHang.Columns[0].Width = 80;
dgvHDBanHang.Columns[1].Width = 130;
dgvHDBanHang.Columns[2].Width = 80;
dgvHDBanHang.Columns[3].Width = 90;
dgvHDBanHang.Columns[4].Width = 90;
dgvHDBanHang.Columns[5].Width = 90;
dgvHDBanHang.AllowUserToAddRows = false;
dgvHDBanHang.EditMode =
DataGridViewEditMode.EditProgrammatically;
}

```

### *d) Phương thức LoadInfoHoaDon()*

```

private void LoadInfoHoaDon()
{
    string str;
    str = "SELECT NgayBan FROM tblHDBan WHERE MaHDBan = N'" +
txtMaHDBan.Text + "'";
    txtNgayBan.Text =
Functions.ConvertDateTime(Functions.GetFieldValues(str));
    str = "SELECT MaNhanVien FROM tblHDBan WHERE MaHDBan = N'" +
txtMaHDBan.Text + "'";
    cboMaNhanVien.Text = Functions.GetFieldValues(str);
    str = "SELECT MaKhach FROM tblHDBan WHERE MaHDBan = N'" +
txtMaHDBan.Text + "'";
    cboMaKhach.Text = Functions.GetFieldValues(str);
    str = "SELECT TongTien FROM tblHDBan WHERE MaHDBan = N'" +
txtMaHDBan.Text + "'";
    txtTongTien.Text = Functions.GetFieldValues(str);
    lblBangChu.Text = "Bằng chữ: " +
Functions.ChuyenSoSangChu(txtTongTien.Text);
}

```

## *e) Phương thức btnThem\_Click*

```
private void btnThem_Click(object sender, EventArgs e)
{
    btnXoa.Enabled = false;
    btnLuu.Enabled = true;
    btnInHoaDon.Enabled = false;
    btnThem.Enabled = false;
    ResetValues();
    txtMaHDBan.Text = Functions.CreateKey("HDB");
    LoadDataGridView();
}
```

Với *CreateKey* là một hàm được viết trong Class Functions, có tác dụng sinh khóa tự động cho Mã hóa đơn bán.

## *f) Hàm CreateKey*

Soạn thảo trong Class Functions:

```
//Hàm tạo khóa có dạng: TientoNgaythangnam_giophutgiay
public static string CreateKey(string tiento)
{
    string key = tiento;
    string[] partsDay;
    partsDay = DateTime.Now.ToShortDateString().Split('/');
    //Ví dụ 07/08/2009
    string d = String.Format("{0}{1}{2}", partsDay[0],
partsDay[1], partsDay[2]);
    key = key + d;
    string[] partsTime;
    partsTime = DateTime.Now.ToLongTimeString().Split(':');
    //Ví dụ 7:08:03 PM hoặc 7:08:03 AM
    if (partsTime[2].Substring(3, 2) == "PM")
        partsTime[0] = ConvertTimeTo24(partsTime[0]);
    if (partsTime[2].Substring(3, 2) == "AM")
        if (partsTime[0].Length == 1)
            partsTime[0] = "0" + partsTime[0];
    //Xóa ký tự trắng và PM hoặc AM
    partsTime[2] = partsTime[2].Remove(2, 3);
    string t;
```

```
        t = String.Format("_{0}{1}{2}", partsTime[0], partsTime[1],  
partsTime[2]);  
        key = key + t;  
        return key;  
    }  
}
```

Với *ConvertTimeTo24* là một hàm toàn cục được viết trong Class Functions, có tác dụng chuyển đổi giờ từ dạng PM sang dạng 24h.

### ***g) Hàm ConvertTimeTo24***

Soạn thảo trong Class Functions:

```
//Chuyển đổi từ PM sang dạng 24h  
public static string ConvertTimeTo24(string hour)  
{  
    string h = "";  
    switch (hour)  
    {  
        case "1":  
            h = "13";  
            break;  
        case "2":  
            h = "14";  
            break;  
        case "3":  
            h = "15";  
            break;  
        case "4":  
            h = "16";  
            break;  
        case "5":  
            h = "17";  
            break;  
        case "6":  
            h = "18";  
            break;  
        case "7":  
            h = "19";  
            break;  
        case "8":  
            h = "20";  
    }  
}
```



```

        break;
    case "9":
        h = "21";
        break;
    case "10":
        h = "22";
        break;
    case "11":
        h = "23";
        break;
    case "12":
        h = "0";
        break;
    }
    return h;
}

```

## *h) Phương thức ResetValues*

```

private void ResetValues()
{
    txtMaHDBan.Text = "";
    txtNgayBan.Text = DateTime.Now.ToShortDateString();
    cboMaNhanVien.Text = "";
    cboMaKhach.Text = "";
    txtTongTien.Text = "0";
    lblBangChu.Text = "Bằng chữ: ";
    cboMaHang.Text = "";
    txtSoLuong.Text = "";
    txtGiamGia.Text = "0";
    txtThanhTien.Text = "0";
}

```

## *i) Phương thức btnLuu\_Click*

```

private void btnLuu_Click(object sender, EventArgs e)
{
    string sql;
    double sl, SLcon, tong, Tongmoi;
}

```

```

        sql = "SELECT MaHDBan FROM tblHDBan WHERE MaHDBan=N'" +
txtMaHDBan.Text + "'";
        if( ! Functions.CheckKey(sql) )
        {
            // Mã hóa đơn chưa có, tiến hành lưu các thông tin chung
            // Mã HDBan được sinh tự động do đó không có trường hợp
trùng khóa
            if( txtNgayBan.Text.Length == 0 )
            {
                MessageBox.Show("Bạn phải nhập ngày bán", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtNgayBan.Focus();
                return ;
            }
            if( cboMaNhanVien.Text.Length == 0 )
            {
                MessageBox.Show("Bạn phải nhập nhân viên", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
                cboMaNhanVien.Focus();
                return;
            }
            if( cboMaKhach.Text.Length == 0 )
            {
                MessageBox.Show("Bạn phải nhập khách hàng", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Information );
                cboMaKhach.Focus();
                return;
            }
            sql = "INSERT INTO tblHDBan(MaHDBan, NgayBan, MaNhanVien,
MaKhach, TongTien) VALUES (N'" + txtMaHDBan.Text.Trim() + "', '" +
                Functions.ConvertDateTime(txtNgayBan.Text.Trim())
+ "',N'" + cboMaNhanVien.SelectedValue + "',N'" +
                cboMaKhach.SelectedValue + "'," + txtTongTien.Text
+ ")";

            Functions.RunSQL(sql);
        }
        // Lưu thông tin của các mặt hàng
        if( cboMaHang.Text.Trim().Length == 0 )
        {
            MessageBox.Show("Bạn phải nhập mã hàng", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            cboMaHang.Focus();
            return;
        }

```

```

        if( (txtSoLuong.Text.Trim().Length == 0) || (txtSoLuong.Text
== "0" ))
        {
            MessageBox.Show("Bạn phải nhập số lượng", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtSoLuong.Text = "";
            txtSoLuong.Focus();
            return;
        }
        if( txtGiamGia.Text.Trim().Length == 0 )
        {
            MessageBox.Show("Bạn phải nhập giảm giá", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtGiamGia.Focus();
            return;
        }
        sql = "SELECT MaHang FROM tblChiTietHDBan WHERE MaHang=N'" +
cboMaHang.SelectedValue + "' AND MaHDBan = N'" + txtMaHDBan.Text.Trim() +
"'";

        if(Functions.CheckKey(sql))
        {
            MessageBox.Show("Mã hàng này đã có, bạn phải nhập mã
khác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
            ResetValuesHang();
            cboMaHang.Focus();
            return;
        }
        // Kiểm tra xem số lượng hàng trong kho còn đủ để cung cấp
không?
        sl = Convert.ToDouble(Functions.GetFieldValues("SELECT SoLuong
FROM tblHang WHERE MaHang = N'" + cboMaHang.SelectedValue + "'"));
        if( Convert.ToDouble(txtSoLuong.Text) > sl )
        {
            MessageBox.Show("Số lượng mặt hàng này chỉ còn " + sl,
"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtSoLuong.Text = "";
            txtSoLuong.Focus();
            return;
        }
        sql = "INSERT INTO
tblChiTietHDBan(MaHDBan,MaHang,SoLuong,DonGia, GiamGia,ThanhTien)
VALUES(N'" + txtMaHDBan.Text.Trim() + "',N'" + cboMaHang.SelectedValue +
"', " + txtSoLuong.Text + ", " + txtDonGiaBan.Text+ ", "+ txtGiamGia.Text +
", " + txtThanhTien.Text + ")";
        Functions.RunSQL(sql);

```

```

        LoadDataGridView();
        // Cập nhật lại số lượng của mặt hàng vào bảng tblHang
        SLcon = sl - Convert.ToDouble(txtSoLuong.Text);
        sql = "UPDATE tblHang SET SoLuong =" + SLcon + " WHERE MaHang="
N'" + cboMaHang.SelectedValue + "'";
        Functions.RunSQL(sql);
        // Cập nhật lại tổng tiền cho hóa đơn bán
        tong = Convert.ToDouble(Functions.GetFieldValues("SELECT
TongTien FROM tblHDBan WHERE MaHDBan = N'" + txtMaHDBan.Text + "'"));
        Tongmoi = tong + Convert.ToDouble(txtThanhTien.Text);
        sql = "UPDATE tblHDBan SET TongTien =" + Tongmoi + " WHERE
MaHDBan = N'" + txtMaHDBan.Text + "'";
        Functions.RunSQL(sql);
        txtTongTien.Text = Tongmoi.ToString();
        lblBangChu.Text = "Bảng chữ: " +
Functions.ChuyenSoSangChu(Tongmoi.ToString());
        ResetValuesHang();
        btnXoa.Enabled = true;
        btnThem.Enabled = true;
        btnInHoaDon.Enabled = true;
    }

```

Với *ChuyenSoSangChu* là một hàm toàn cục được viết trong Class Functions, có tác dụng đọc từ dạng số sang dạng chữ.

## *j) Hàm ChuyenSoSangChu*

Soạn thảo trong Class Functions:

```

public static string ChuyenSoSangChu(string sNumber)
{
    int mLen, mDigit;
    string mTemp = "";
    string[] mNumText;
    //Xóa các dấu ",", " nếu có
    sNumber = sNumber.Replace(",", "");
    mNumText =
"không;một;hai;ba;bốn;năm;sáu;bảy;tám;chín".Split(';');
    mLen = sNumber.Length - 1; // trừ 1 vì thứ tự đi từ 0
    for (int i = 0; i <= mLen; i++) {
        mDigit = Convert.ToInt32(sNumber.Substring(i, 1));
        mTemp = mTemp + " " + mNumText[mDigit];
        if (mLen == i) // Chữ số cuối cùng không cần xét tiếp break;
    }
}

```

```

switch ((mLen - i) % 9) {
case 0:
    mTemp = mTemp + " tỷ";
    if (sNumber.Substring(i + 1, 3) == "000") i = i + 3;
    if (sNumber.Substring(i + 1, 3) == "000") i = i + 3;
    if (sNumber.Substring(i + 1, 3) == "000") i = i + 3;
    break;
case 6:
    mTemp = mTemp + " triệu";
    if (sNumber.Substring(i + 1, 3) == "000") i = i + 3;
    if (sNumber.Substring(i + 1, 3) == "000") i = i + 3;
    break;
case 3:
    mTemp = mTemp + " nghìn";
    if (sNumber.Substring(i + 1, 3) == "000") i = i + 3;
    break;
default:
    switch ((mLen - i) % 3)
    {
    case 2:
        mTemp = mTemp + " trăm";
        break;
    case 1:
        mTemp = mTemp + " mươi";
        break;
    }
    break;
}
}
//Loại bỏ trường hợp x00
mTemp = mTemp.Replace("không mươi không ", "");
mTemp = mTemp.Replace("không mươi không", ""); //Loại bỏ trường
hợp 00x
mTemp = mTemp.Replace("không mươi ", "linh "); //Loại bỏ trường
hợp x0, x>=2
mTemp = mTemp.Replace("mười không", "mười");
//Fix trường hợp 10
mTemp = mTemp.Replace("một mươi", "mười");
//Fix trường hợp x4, x>=2
mTemp = mTemp.Replace("mười bốn", "mười tư");
//Fix trường hợp x04
mTemp = mTemp.Replace("linh bốn", "linh tư");
//Fix trường hợp x5, x>=2
mTemp = mTemp.Replace("mười năm", "mười lăm");
//Fix trường hợp x1, x>=2

```

```

        mTemp = mTemp.Replace("mười một", "mười mốt");
        //Fix trường hợp x15
        mTemp = mTemp.Replace("mười năm", "mười lăm");
        //Bỏ ký tự space
        mTemp = mTemp.Trim();
        //Viết hoa ký tự đầu tiên
        mTemp = mTemp.Substring(0, 1).ToUpper() + mTemp.Substring(1) + "
đồng";

        return mTemp;
    }

```

## *k) Phương thức ResetValuesHang*

```

private void ResetValuesHang()
{
    cboMaHang.Text = "";
    txtSoLuong.Text = "";
    txtGiamGia.Text = "0";
    txtThanhTien.Text = "0";
}

```

## *l) Phương thức dgvHDBanHang\_DoubleClick*

Phương thức này cho phép người dùng nhấp đúp chuột vào một mặt hàng trong lưới để xóa.

```

private void dgvHDBanHang_DoubleClick(object sender, EventArgs e)
{
    string MaHangxoa, sql;
    Double ThanhTienxoa, SoLuongxoa, sl, slcon, tong, tongmoi;
    if (tblCTHDB.Rows.Count == 0)
    {
        MessageBox.Show("Không có dữ liệu!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if ((MessageBox.Show("Bạn có chắc chắn muốn xóa không?",
    "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
    DialogResult.Yes))
    {

```

```

        //Xóa hàng và cập nhật lại số lượng hàng
        MaHangxoa =
dgvHDBanHang.CurrentRow.Cells["MaHang"].Value.ToString();
        SoLuongxoa =
Convert.ToDouble(dgvHDBanHang.CurrentRow.Cells["SoLuong"].Value.ToString()
);
        ThanhTienxoa =
Convert.ToDouble(dgvHDBanHang.CurrentRow.Cells["ThanhTien"].Value.ToString()
());
        sql = "DELETE tblChiTietHDBan WHERE MaHDBan=N'" +
txtMaHDBan.Text + "' AND MaHang = N'" + MaHangxoa + "'";
        Functions.RunSQL(sql);
        // Cập nhật lại số lượng cho các mặt hàng
        sl = Convert.ToDouble(Functions.GetFieldValues("SELECT
SoLuong FROM tblHang WHERE MaHang = N'" + MaHangxoa + "'"));
        slcon = sl + SoLuongxoa;
        sql = "UPDATE tblHang SET SoLuong =" + slcon + " WHERE
MaHang= N'" + MaHangxoa + "'";
        Functions.RunSQL(sql);
        // Cập nhật lại tổng tiền cho hóa đơn bán
        tong = Convert.ToDouble(Functions.GetFieldValues("SELECT
TongTien FROM tblHDBan WHERE MaHDBan = N'" + txtMaHDBan.Text + "'"));
        tongmoi = tong - ThanhTienxoa;
        sql = "UPDATE tblHDBan SET TongTien =" + tongmoi + " WHERE
MaHDBan = N'" + txtMaHDBan.Text + "'";
        Functions.RunSQL(sql);
        txtTongTien.Text = tongmoi.ToString();
        lblBangChu.Text = "Bằng chữ: " +
Functions.ChuyenSoSangChu(tongmoi.ToString());
        LoadDataGridView();
    }
}

```

### *m) Phương thức btnXoa\_Click*

Phương thức này cho phép xóa toàn bộ thông tin của một hóa đơn

```

private void btnXoa_Click(object sender, EventArgs e)
{
    double sl, slcon, slxoa;

```

```

        if (MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            string sql = "SELECT MaHang, SoLuong FROM tblChiTietHDBan WHERE MaHDBan = N'" + txtMaHDBan.Text + "'";
            DataTable tblHang = Functions.GetDataToTable(sql);
            for (int hang = 0; hang <= tblHang.Rows.Count - 1; hang++)
            {
                // Cập nhật lại số lượng cho các mặt hàng
                sl = Convert.ToDouble(Functions.GetFieldValues("SELECT SoLuong FROM tblHang WHERE MaHang = N'" + tblHang.Rows[hang][0].ToString() + "'"));
                slxoa = Convert.ToDouble(tblHang.Rows[hang][1].ToString());
                slcon = sl + slxoa;
                sql = "UPDATE tblHang SET SoLuong =" + slcon + " WHERE MaHang= N'" + tblHang.Rows[hang][0].ToString() + "'";
                Functions.RunSQL(sql);
            }

            //Xóa chi tiết hóa đơn
            sql = "DELETE tblChiTietHDBan WHERE MaHDBan=N'" + txtMaHDBan.Text + "'";
            Functions.RunSqlDel(sql);

            //Xóa hóa đơn
            sql = "DELETE tblHDBan WHERE MaHDBan=N'" + txtMaHDBan.Text + "'";
            Functions.RunSqlDel(sql);
            ResetValues();
            LoadDataGridview();
            btnXoa.Enabled = false;
            btnInHoaDon.Enabled = false;
        }
    }

```

## *n) Phương thức cboMaNhanVien\_TextChanged*

```

private void cboMaNhanVien_TextChanged(object sender, EventArgs e)
{
    string str;

```



```

        if (cboMaNhanVien.Text == "")
            txtTenNhanVien.Text = "";
        // Khi chọn Mã nhân viên thì tên nhân viên tự động hiện ra
        str = "Select TenNhanVien from tblNhanVien where MaNhanVien"
        =N'" + cboMaNhanVien.SelectedValue + "'";
        txtTenNhanVien.Text = Functions.GetFieldValues(str);
    }

```

### *o) Phương thức cboMaKhach\_TextChanged*

```

private void cboMaKhach_TextChanged(object sender, EventArgs e)
{
    string str;
    if (cboMaKhach.Text == "")
    {
        txtTenKhach.Text = "";
        txtDiaChi.Text = "";
        txtDienThoai.Text = "";
    }
    //Khi chọn Mã khách hàng thì các thông tin của khách hàng sẽ
    hiện ra
    str = "Select TenKhach from tblKhach where MaKhach = N'" +
    cboMaKhach.SelectedValue+ "'";
    txtTenKhach.Text = Functions.GetFieldValues(str);
    str = "Select DiaChi from tblKhach where MaKhach = N'" +
    cboMaKhach.SelectedValue + "'";
    txtDiaChi.Text = Functions.GetFieldValues(str);
    str = "Select DienThoai from tblKhach where MaKhach= N'" +
    cboMaKhach.SelectedValue + "'";
    txtDienThoai.Text = Functions.GetFieldValues(str);
}

```

### *p) Phương thức cboMaHang\_TextChanged*

```

private void cboMaHang_TextChanged(object sender, EventArgs e)
{
    string str;
    if (cboMaHang.Text == "")

```

```

    {
        txtTenHang.Text = "";
        txtDonGiaBan.Text = "";
    }
    // Khi chọn mã hàng thì các thông tin về hàng hiện ra
    str = "SELECT TenHang FROM tblHang WHERE MaHang =N'" +
cboMaHang.SelectedValue+ "'";
    txtTenHang.Text = Functions.GetFieldValues(str);
    str = "SELECT DonGiaBan FROM tblHang WHERE MaHang =N'" +
cboMaHang.SelectedValue+ "'";
    txtDonGiaBan.Text = Functions.GetFieldValues(str);
}

```

### *q) Phương thức txtSoLuong\_TextChanged*

```

private void txtSoLuong_TextChanged(object sender, EventArgs e)
{
    //Khi thay đổi số lượng thì thực hiện tính lại thành tiền
    double tt, sl, dg, gg;
    if (txtSoLuong.Text == "")
        sl = 0;
    else
        sl = Convert.ToDouble(txtSoLuong.Text);
    if (txtGiamGia.Text == "")
        gg = 0;
    else
        gg = Convert.ToDouble(txtGiamGia.Text);
    if (txtDonGiaBan.Text == "")
        dg = 0;
    else
        dg = Convert.ToDouble(txtDonGiaBan.Text);
    tt = sl * dg - sl * dg * gg / 100;
    txtThanhTien.Text = tt.ToString();
}

```

### *r) Phương thức txtGiamGia\_TextChanged*

```

private void txtGiamGia_TextChanged(object sender, EventArgs e)
{

```

```

//Khi thay đổi giảm giá thì tính lại thành tiền
double tt, sl, dg, gg;
if (txtSoLuong.Text == "")
    sl = 0;
else
    sl = Convert.ToDouble(txtSoLuong.Text);
if (txtGiamGia.Text == "")
    gg = 0;
else
    gg = Convert.ToDouble(txtGiamGia.Text);
if (txtDonGiaBan.Text == "")
    dg = 0;
else
    dg = Convert.ToDouble(txtDonGiaBan.Text);
tt = sl * dg - sl * dg * gg / 100;
txtThanhTien.Text = tt.ToString();
}

```

## *s) Phương thức btnInHoaDon\_Click*

```

private void btnInHoaDon_Click(object sender, EventArgs e)
{
    // Khởi động chương trình Excel
    COMExcel.Application exApp = new COMExcel.Application();
    COMExcel.Workbook exBook; //Trong 1 chương trình Excel có
nhiều Workbook
    COMExcel.Worksheet exSheet; //Trong 1 Workbook có nhiều
Worksheet
    COMExcel.Range exRange;
    string sql;
    int hang=0, cot=0;
    DataTable tblThongtinHD, tblThongtinHang;
    exBook =
exApp.Workbooks.Add(COMExcel.XlWBATemplate.xlWBATWorksheet);
    exSheet = exBook.Worksheets[1];
    // Định dạng chung
    exRange = exSheet.Cells[1, 1];
    exRange.Range["A1:Z300"].Font.Name = "Times new roman"; //Font
chữ
    exRange.Range["A1:B3"].Font.Size = 10;
    exRange.Range["A1:B3"].Font.Bold = true;
    exRange.Range["A1:B3"].Font.ColorIndex = 5; //Màu xanh da trời

```

```

exRange.Range["A1:A1"].ColumnWidth = 7;
exRange.Range["B1:B1"].ColumnWidth = 15;
exRange.Range["A1:B1"].MergeCells = true;
exRange.Range["A1:B1"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
exRange.Range["A1:B1"].Value = "Shop B.A.";
exRange.Range["A2:B2"].MergeCells = true;
exRange.Range["A2:B2"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
exRange.Range["A2:B2"].Value = "Chùa Bộc - Hà Nội";
exRange.Range["A3:B3"].MergeCells = true;
exRange.Range["A3:B3"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
exRange.Range["A3:B3"].Value = "Điện thoại: (04)38526419";
exRange.Range["C2:E2"].Font.Size = 16;
exRange.Range["C2:E2"].Font.Bold = true;
exRange.Range["C2:E2"].Font.ColorIndex = 3; //Màu đỏ
exRange.Range["C2:E2"].MergeCells = true;
exRange.Range["C2:E2"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
exRange.Range["C2:E2"].Value = "HÓA ĐƠN BÁN";
// Biểu diễn thông tin chung của hóa đơn bán
sql = "SELECT a.MaHDBan, a.NgayBan, a.TongTien, b.TenKhach,
b.DiaChi, b.DienThoai, c.TenNhanVien FROM tblHDBan AS a, tblKhach AS b,
tblNhanVien AS c WHERE a.MaHDBan = N'" + txtMaHDBan.Text + "' AND
a.MaKhach = b.MaKhach AND a.MaNhanVien = c.MaNhanVien";
tblThongtinHD = Functions.GetDataToTable(sql);
exRange.Range["B6:C9"].Font.Size = 12;
exRange.Range["B6:B6"].Value = "Mã hóa đơn:";
exRange.Range["C6:E6"].MergeCells = true;
exRange.Range["C6:E6"].Value =
tblThongtinHD.Rows[0][0].ToString();
exRange.Range["B7:B7"].Value = "Khách hàng:";
exRange.Range["C7:E7"].MergeCells = true;
exRange.Range["C7:E7"].Value =
tblThongtinHD.Rows[0][3].ToString();
exRange.Range["B8:B8"].Value = "Địa chỉ:";
exRange.Range["C8:E8"].MergeCells = true;
exRange.Range["C8:E8"].Value =
tblThongtinHD.Rows[0][4].ToString();
exRange.Range["B9:B9"].Value = "Điện thoại:";
exRange.Range["C9:E9"].MergeCells = true;
exRange.Range["C9:E9"].Value =
tblThongtinHD.Rows[0][5].ToString();
//Lấy thông tin các mặt hàng

```

```

        sql = "SELECT b.TenHang, a.SoLuong, b.DonGiaBan, a.GiamGia,
a.ThanhTien " +
        "FROM tblChiTietHDBan AS a , tblHang AS b WHERE
a.MaHDBan = N'" +
        txtMaHDBan.Text + "' AND a.MaHang = b.MaHang";
tblThongtinHang = Functions.GetDataToTable(sql);
//Tạo dòng tiêu đề bảng
exRange.Range["A11:F11"].Font.Bold = true;
exRange.Range["A11:F11"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
exRange.Range["C11:F11"].ColumnWidth = 12;
exRange.Range["A11:A11"].Value = "STT";
exRange.Range["B11:B11"].Value = "Tên hàng";
exRange.Range["C11:C11"].Value = "Số lượng";
exRange.Range["D11:D11"].Value = "Đơn giá";
exRange.Range["E11:E11"].Value = "Giảm giá";
exRange.Range["F11:F11"].Value = "Thành tiền";
for (hang = 0 ; hang < tblThongtinHang.Rows.Count; hang++)
{
    //Điền số thứ tự vào cột 1 từ dòng 12
    exSheet.Cells[1][hang + 12] = hang + 1;
    for (cot = 0; cot < tblThongtinHang.Columns.Count; cot++)
        //Điền thông tin hàng từ cột thứ 2, dòng 12
        {
            exSheet.Cells[cot + 2][hang + 12] =
tblThongtinHang.Rows[hang][cot].ToString();
            if (cot == 3) exSheet.Cells[cot + 2][hang + 12] =
tblThongtinHang.Rows[hang][cot].ToString() + "%";
        }
    }
    exRange = exSheet.Cells[cot][hang + 14];
    exRange.Font.Bold = true;
    exRange.Value2 = "Tổng tiền:";
    exRange = exSheet.Cells[cot + 1][hang + 14];
    exRange.Font.Bold = true;
    exRange.Value2 = tblThongtinHD.Rows[0][2].ToString();
    exRange = exSheet.Cells[1][hang + 15]; //Ô A1
    exRange.Range["A1:F1"].MergeCells = true;
    exRange.Range["A1:F1"].Font.Bold = true;
    exRange.Range["A1:F1"].Font.Italic = true;
    exRange.Range["A1:F1"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignRight;
    exRange.Range["A1:F1"].Value = "Bảng chữ: " +
Functions.ChuyenSoSangChu(tblThongtinHD.Rows[0][2].ToString());
    exRange = exSheet.Cells[4][hang + 17]; //Ô A1

```

```

exRange.Range["A1:C1"].MergeCells = true;
exRange.Range["A1:C1"].Font.Italic = true;
exRange.Range["A1:C1"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
DateTime d = Convert.ToDateTime(tblThongtinHD.Rows[0][1]);
exRange.Range["A1:C1"].Value = "Hà Nội, ngày " + d.Day + "
tháng " + d.Month + " năm " + d.Year;
exRange.Range["A2:C2"].MergeCells = true;
exRange.Range["A2:C2"].Font.Italic = true;
exRange.Range["A2:C2"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
exRange.Range["A2:C2"].Value = "Nhân viên bán hàng";
exRange.Range["A6:C6"].MergeCells = true;
exRange.Range["A6:C6"].Font.Italic = true;
exRange.Range["A6:C6"].HorizontalAlignment =
COMExcel.XlHAlign.xlHAlignCenter;
exRange.Range["A6:C6"].Value = tblThongtinHD.Rows[0][6];
exSheet.Name = "Hóa đơn nhập";
exApp.Visible = true;
}

```

	A	B	C	D	E	F
1		Shop B.A.				
2		Chùa Bộc - Hà Nội				
3		Điện thoại: (04)38526419				
4						
5						
6		Mã hóa đơn:	HDB04062014_211856			
7		Khách hàng:	Nguyễn Văn Dương			
8		Địa chỉ:	256 Chùa Bộc, Hà Nội			
9		Điện thoại:	(043) 986-4869			
10						
11	STT	Tên hàng	Số lượng	Đơn giá	Giảm giá	Thành tiền
12	1	Cối xay giò	1	15000	0%	15000
13	2	Gỏi búng	2	150000	3%	291000
14	3	Chân giò	1	150000	0%	150000
15	4	Cốc dĩa	3	20000	10%	54000
16						
17						
18					Tổng tiền:	510000
19					Bằng chữ: Năm trăm mười nghìn đồng	
20						
21					Hà Nội, ngày 6 tháng 4 năm 2014	
22					Nhân viên bán hàng	
23						
24						
25						
26					Nguyễn Mai Hoa	

**Kết quả ta có tệp Excel như sau:**

**t) Phương thức btnTimKiem\_Click**

```

private void btnTimKiem_Click(object sender, EventArgs e)
{
    if (cboMaHDBan.Text == "")
    {
        MessageBox.Show("Bạn phải chọn một mã hóa đơn để tìm",
        "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

```

        cboMaHDBan.Focus();
        return;
    }
    txtMaHDBan.Text = cboMaHDBan.Text;
    LoadInfoHoadon();
    LoadDataGridView();
    btnXoa.Enabled = true;
    btnLuu.Enabled = true;
    btnInHoadon.Enabled = true;
    cboMaHDBan.SelectedIndex = -1;
}

```

### *u) Phương thức txtSoLuong\_KeyPress*

```

private void txtSoLuong_KeyPress(object sender, KeyPressEventArgs e)
{
    if (((e.KeyChar >= '0') && (e.KeyChar <= '9')) ||
    (Convert.ToInt32(e.KeyChar) == 8))
        e.Handled = false;
    else
        e.Handled = true;
}

```

Thực hiện tương tự cho ***txtGiamGia\_KeyPress***.

### *v) Phương thức cboMaHDBan\_DropDown*

Phương thức này cập nhật lại danh sách các mã hóa đơn bán và lưu vào cboMaHDBan mỗi khi người dùng nhấp chuột vào nút xổ xuống của cbo.

```

private void cboMaHDBan_DropDown(object sender, EventArgs e)
{
    Functions.FillCombo("SELECT MaHDBan FROM tblHDBan",
    cboMaHDBan, "MaHDBan", "MaHDBan");
    cboMaHDBan.SelectedIndex = -1;
}

```

### *w) Phương thức frmHoadonBan\_FormClosing*

```
private void frmHoadonBan_FormClosing(object sender, FormClosingEventArgs e)
{
    //Xóa dữ liệu trong các điều khiển trước khi đóng Form
    ResetValues();
}
```

### ***x) Phương thức btnDong\_Click***

```
private void btnDong_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Xem Video hướng dẫn Quản lý Hoá Đơn  
**4.8. Form tìm kiếm Hóa đơn bán**

### ***a) Khai báo***

– Thư viện

```
using System.Data.SqlClient;
using QuanLyBanHang.Class;
```

– Biến

```
DataTable tblHDB; //Hoá đơn bán
```

### ***b) Phương thức frmTimHDBan\_Load***

```
private void frmTimHDBan_Load(object sender, EventArgs e)
{
    ResetValues();
}
```



```

        dgvTKHoaDon.DataSource = null;
    }

```

### *c) Phương thức ResetValues*

```

private void ResetValues()
{
    foreach (Control Ctl in this.Controls)
        if (Ctl is TextBox)
            Ctl.Text = "";
    txtMaHDBan.Focus();
}

```

### *d) Phương thức btnTimKiem\_Click*

```

private void btnTimKiem_Click(object sender, EventArgs e)
{
    string sql;
    if ((txtMaHDBan.Text == "") && (txtThang.Text == "") &&
(txtNam.Text == "") &&
        (txtMaNhanVien.Text == "") && (txtMaKhach.Text == "") &&
        (txtTongTien.Text == ""))
    {
        MessageBox.Show("Hãy nhập một điều kiện tìm kiếm!!!", "Yêu
cầu ...",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    sql = "SELECT * FROM tblHDBan WHERE 1=1";
    if (txtMaHDBan.Text != "")
        sql = sql + " AND MaHDBan Like N'" + txtMaHDBan.Text +
        "'";
    if (txtThang.Text != "")
        sql = sql + " AND MONTH(NgayBan) =" + txtThang.Text;
    if (txtNam.Text != "")
        sql = sql + " AND YEAR(NgayBan) =" + txtNam.Text;
    if (txtMaNhanVien.Text != "")
        sql = sql + " AND MaNhanVien Like N'" +
txtMaNhanVien.Text + "'";
    if (txtMaKhach.Text != "")

```

```

        sql = sql + " AND MaKhach Like N'" + txtMaKhach.Text +
        "%'";

        if (txtTongTien.Text != "")
            sql = sql + " AND TongTien <=" + txtTongTien.Text;
        tblHDB = Functions.GetDataToTable(sql);
        if (tblHDB.Rows.Count == 0)
        {
            MessageBox.Show("Không có bản ghi thỏa mãn điều kiện!!",
            "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
            MessageBox.Show("Có " + tblHDB.Rows.Count + " bản ghi thỏa
            mãn điều kiện!", "Thông báo", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        dgvTKHoaDon.DataSource = tblHDB;
        LoadDataGridView();
    }

```

### *e) Phương thức LoadDataGridView*

```

private void LoadDataGridView()
{
    dgvTKHoaDon.Columns[0].HeaderText = "Mã HDB";
    dgvTKHoaDon.Columns[1].HeaderText = "Mã nhân viên";
    dgvTKHoaDon.Columns[2].HeaderText = "Ngày bán";
    dgvTKHoaDon.Columns[3].HeaderText = "Mã khách";
    dgvTKHoaDon.Columns[4].HeaderText = "Tổng tiền";
    dgvTKHoaDon.Columns[0].Width = 150;
    dgvTKHoaDon.Columns[1].Width = 100;
    dgvTKHoaDon.Columns[2].Width = 80;
    dgvTKHoaDon.Columns[3].Width = 80;
    dgvTKHoaDon.Columns[4].Width = 80;
    dgvTKHoaDon.AllowUserToAddRows = false;
    dgvTKHoaDon.EditMode =
    DataGridViewEditMode.EditProgrammatically;
}

```

### *f) Phương thức btnTimLai\_Click*

```
private void btnTimLai_Click(object sender, EventArgs e)
{
    ResetValues();
    dgvTKHoaDon.DataSource = null;
}
```

### *g) Phương thức txtTongTien\_KeyPress*

```
private void txtTongTien_KeyPress(object sender, KeyPressEventArgs e)
{
    if ((e.KeyChar >= '0') && (e.KeyChar <= '9')) ||
    (Convert.ToInt32(e.KeyChar) == 8))
        e.Handled = false;
    else
        e.Handled = true;
}
```

### *h) Phương thức dgvTKHoaDon\_DoubleClick*

Phương thức này cho phép người dùng nhấp đúp chuột chọn một hóa đơn trên lưới, rồi gọi và hiển thị thông tin của hóa đơn đó lên form frmHoadonBan.

*Chú ý:* Thay đổi phạm vi truy cập của điều khiển *txtMaHDBan* thành *Public* trong form frmHoadonBan (trong file frmHoadonBan.Designer.cs)

```
private void dgvTKHoaDon_DoubleClick(object sender, EventArgs e)
{
    string mahd;
    if (MessageBox.Show("Bạn có muốn hiển thị thông tin chi tiết?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        mahd =
        dgvTKHoaDon.CurrentRow.Cells["MaHDBan"].Value.ToString();
    }
}
```

```
        frmHoadonBan frm = new frmHoadonBan();  
        frm.txtMaHDBan.Text = mahd;  
        frm.StartPosition = FormStartPosition.CenterParent;  
        frm.ShowDialog();  
    }  
}
```

### *i) Phương thức btnDong\_Click*

```
private void btnDong_Click(object sender, EventArgs e)  
{  
    this.Close();  
}
```