

LẬP TRÌNH CƠ SỞ DỮ LIỆU LOCAL SQL VỚI C#

1. Thiết kế cơ sở dữ liệu đặt tên BanHang, trong đó có bảng:

tblMatHang(MaSP nchar(5), TenSP nvarchar(30), NgaySX Date, NgayHH Date, DonVi nvarchar(10), DonGia Float , GhiChu nvarchar(200))

2. Thiết kế giao diện

	MaSP	TenSP	NgaySX	NgayHH	DonVi	DonGia
▶	SP00001	Bán phẩm Dell	03/01/2015	15/08/2018	Chiếc	500000
	SP00002	Chuột quang khô...	03/01/2018	15/08/2021	Cái	200000
*						

- Các phần của sổ co giãn hợp lý khi cửa sổ của Form phóng to hoặc thu nhỏ, ví dụ cửa sổ (3) luôn giữ cố định ở bên trái, phần cửa sổ (4) và (5) co giãn khi Form thay đổi kích thước, (1) và (2) luôn nằm trên đầu và phía dưới cùng của Form.
- Các nút có gắn các phím tắt ví dụ như Alt + T cho nút Tìm kiếm, Alt + H cho nút Thêm, v.v.

a) Chức năng Tìm kiếm

Hỗ trợ các tiêu chí: MaSP, TenSP, NgaySX, NgayHetHan, DonGia

Tiêu đề: TÌM KIẾM MẶT HÀNG

Khi ấn nút Tìm:

- ✓ dgvDanhSach sẽ hiển thị tất cả các thông tin về các sản phẩm thoả mãn các tiêu chí lựa chọn.
- ✓ Cấm Click vào các nút Sửa, Xoá
- ✓ Khi Click vào từng trường trên dgvDanhSach các thông tin sẽ hiển thị chi tiết lên grbChiTiet, cho phép kích hoạt nút Sửa để sửa đổi dữ liệu. Không hiển thị các nút Lưu, Huỷ trong group box ChiTiet.

b) Chức năng Thêm

Tiêu đề: THÊM MẶT HÀNG

Khi ấn nút Thêm:

- ✓ GroupBox ChiTiet sẽ xoá trắng và cho phép sửa đổi dữ liệu
- ✓ Cấm Click vào các nút Sửa, Xoá.
- ✓ Trong GroupBox ChiTiet hiển thị 2 nút lệnh Lưu, Huỷ. Khi ấn nút Huỷ, sẽ thoát khỏi chức năng thêm (Xoá trắng GroupBox ChiTiet, không cho nhập dữ liệu)
- ✓ Nếu Thêm thành công, cập nhật lại dgvDanhSach

c) Chức năng Sửa

Tiêu đề: CẬP NHẬT MẶT HÀNG

Nút Sửa chỉ cho Enable = True khi người dùng chọn một bản ghi trong dgvDanhSach.

Khi ấn nút Sửa:

- ✓ Cấm Click vào các nút Thêm, Xoá.
- ✓ GroupBox ChiTiet cho phép sửa dữ liệu. Trong GroupBox ChiTiet cho hiển thị nút lệnh Lưu, Huỷ bỏ. Khi ấn nút Huỷ, sẽ thoát khỏi chức năng Sửa (cấm Click vào nút Sửa, cho phép Click vào nút Thêm)
- ✓ Nếu Cập nhật thành công, cập nhật lại dgvDanhSach

d) Chức năng Xoá

Tiêu đề: XOÁ MẶT HÀNG

Nút Xoá chỉ cho phép Click khi người dùng chọn một bản ghi trong dgvDanhSach

Khi ấn nút Xoá:

- ✓ Cấm Click vào các nút Thêm, Sửa.
- ✓ GroupBox ChiTiet không cho phép sửa dữ liệu. Trong grbChiTiet cho hiển thị nút lệnh Lưu, Huỷ bỏ.
- ✓ Khi ấn nút Huỷ bỏ sẽ thoát khỏi chức năng Xoá (cấm Click vào nút Xoá, cho phép Click vào nút Thêm)
- ✓ Xây dựng chương trình

HƯỚNG DẪN

1. Thiết kế giao diện

Phần Giao diện	Loại Control	Các thuộc tính
Cửa sổ	Form	Name=frmMatHang
	ErrorProvider	Name = errChiTiet
(1)	Panel	Dock=Top
	Label	Name=lblTieuDe
(2)	Panel	Dock=Bottom
	Button	Name=btnTimKiem Text=&Tìm kiếm
	Button	Name=btnThem Text=T&hêm
	Button	Name=btnSua Text=&Sua
	Button	Name=btnXoa Text=&Xóa
	Button	Name=btnThoat Text=Th&oat
(3)	GroupBox	Text=Chi tiết Dock=Right
	Splitter	Dock=Left
	Label	Text=Mã SP:
	TextBox	Name=txtMaSP
	Label	Text=Tên SP:
	TextBox	Name=txtTenSP
	Label	Text=Ngày SX:
	DateTimePicker	Name=dtpNgaySX Format=Short
	Label	Text=Ngày HH:
	DateTimePicker	Name=dtpNgayHH Format=Short
	Label	Text=Đơn vị:
	TextBox	Name=txtDonVi
	Label	Text=Đơn giá:
	TextBox	Name=txtDonGia
	Label	Text=Ghi chú:
	TextBox	Name=txtGhiChu
	Button	Name=btnLuu Text=&Luu

	Button	Name=btnHuy Text=&Hủy
(4)	GroupBox	Text=Tìm kiếm Dock=Top
	Splitter	Dock=Bottom
	Label	Text=Mã SP:
	TextBox	Name=txtTKMaSP
	Label	Text=Tên SP:
	TextBox	Name=txtTKTenSP
(5)	GroupBox	Dock=Fill Text=Kết quả
	DataGridView	Name=dgvMatHang Dock=Fill

Chú ý để thiết kế được giao diện thân thiện với người dùng và các control tự co giãn

khi Form thay đổi kích thước cần thiết kế, chú ý kéo các control đúng thứ tự như sau:

- ✓ Giao diện phần (1): là phần nhãn tiêu đề, kéo control Panel vào Form và đặt thuộc tính Dock = Top để cho Panel luôn nằm ở trên đầu của Form, sau đó kéo một Label nằm trên Panel này.
- ✓ Giao diện phần (2): là phần các nút Tìm kiếm, Thêm ... cũng làm tương tự như phần (1), kéo control Panel và đặt thuộc tính Dock = Bottom để Panel này luôn nằm cuối của Form, sau đó mới đặt các Button nằm trên Panel này.
- ✓ Giao diện phần (3): là phần chi tiết, kéo control GroupBox vào Form và đặt thuộc tính Dock = Right, sau đó thiết kế các control cho nhập Mã SP, Tên SP ... Sau đó ta kéo thêm control Splitter và đặt thuộc tính Dock = Left để cho phép người dùng thay đổi độ rộng giữa phần (3) với phần (4) và (5).
- ✓ Giao diện phần (4): là phần tìm kiếm, kéo control GroupBox vào Form và đặt thuộc tính Dock = Top, lúc này nó sẽ nằm ở trên phần còn lại. Sau đó đặt các control cho phép nhập tìm kiếm như Mã SP, Tên SP ... Trong phần này ta cũng kéo control Splitter, đặt thuộc tính Dock = Top để cho phép người sử dụng thay đổi độ rộng giữa phần (4) và phần (5)
- ✓ Giao diện phần (5): là phần hiển thị dữ liệu, kéo control GroupBox và đặt thuộc tính Dock = Fill, sau đó kéo control DataGridView nằm trong GroupBox này nó sẽ chiếm toàn bộ phần không gian còn lại.
- ✓ Viết chương trình chạy trên Windows thì cần tuân theo chuẩn của Windows, ví dụ khi ấn phím Tab trên bàn phím thì con trỏ sẽ chạy từ bên trên -> sang trái
→ xuống dưới -> sang phải. Nếu thứ tự bạn kéo control không đúng thì khi ấn phím Tab sẽ nhảy lung tung. Bạn có thể đặt lại dùng thuộc tính TabIndex.

2. Viết mã lệnh

2.1. Xây dựng Class xử lý liên kết DataBase và thao tác với dữ liệu gồm các phương thức sau:

- Phương thức mở kết nối đến CSDL: OpenConnect()
- Phương thức đóng kết nối đến CSDL: CloseConnect()
- Phương thức đọc dữ liệu từ CSDL trả về một DataTable: DataReader(string sqlSelect)
- Phương thức thực thi câu lệnh SQL dạng: insert, delete, update, tên Phương thức là: DataChager(string sql)

Hướng dẫn

Tạo một thư mục chứa các lớp dùng chung cho ứng dụng đặt tên là **Classes**, trong thư mục này thêm Class **DataBaseProcess.cs** với mã lệnh gọi ý như sau:

```
class DataBaseProcess
{
    //Khai báo biến toàn cục, bạn phải thay đổi chuỗi kết nối phù hợp
    string strConnect = "Data Source=THUHUONG\\SQLEXPRESS;" +
        "DataBase=BanHang;User ID=sa;" +
        "Password=123456;Integrated Security=false";
    SqlConnection sqlConnect=null;
    //Phương thức mở kết nối
    void OpenConnect()
    {
        sqlConnect = new SqlConnection(strConnect);
        if (sqlConnect.State != ConnectionState.Open)
            sqlConnect.Open();
    }
    //Phương thức đóng kết nối
    void CloseConnect()
    {
        if(sqlConnect.State!=ConnectionState.Closed)
        {
            sqlConnect.Close();
            sqlConnect.Dispose();
        }
    }
    //Phương thức thực thi câu lệnh Select trả về một DataTable
    public DataTable DataReader(string sqlSelct)
    {
        DataTable tblData = new DataTable();
        OpenConnect();
        SqlDataAdapter sqlData = new SqlDataAdapter( sqlSelct,sqlConnect);
        sqlData.Fill(tblData);
        CloseConnect();
        return tblData;
    }
    //Phương thức thực hiện câu lệnh dạng insert,update,delete
    public void DataChange(string sql)
    {

```

```

        OpenConnect();
        SqlCommand sqlcomma = new SqlCommand();
        sqlcomma.Connection = sqlConnect;
        sqlcomma.CommandText = sql;
        sqlcomma.ExecuteNonQuery();
        CloseConnect();
    }
}

```

2.2. Viết sự kiện form Load cho frmMatHang

- Thực hiện load dữ liệu lên dgvMatHang
- Khi Form hiện lên thì ẩn các nút sửa, xóa và ẩn GroupBox Chi tiết

Hướng dẫn

*Chú ý: Để sử dụng class DataBaseProcess ở trên ta cần khai báo và khởi tạo một đối tượng của Class

```

//Khai báo và khởi tạo biến toàn cục trong class frmMatHang sử dụng class DataBaseProcess
Classes.DataBaseProcess dtbase = new Classes.DataBaseProcess();

```

```

//Phương thức ẩn hiện các control trong groupBox Chi tiết

```

```

private void HienChiTiet(bool hien)
{
    txtMaSP.Enabled = hien;
    txtTenSP.Enabled = hien;
    dtpNgayHH.Enabled = hien;
    dtpNgaySX.Enabled = hien;
    txtDonVi.Enabled = hien;
    txtDonGia.Enabled = hien;
    txtGhiChu.Enabled = hien;
    //Ẩn hiện 2 nút Lưu và Hủy
    btnLuu.Enabled = hien;
    btnHuy.Enabled = hien;
}

```

```

//Sự kiện load_Form

```

```

private void frmMatHang_Load(object sender, EventArgs e)
{
    //Load dữ liệu lên DataGridView
    dgvKetQua.DataSource = dtbase.DataReader("Select * from tblMatHang");
    //Ẩn nút Sửa,xóa
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
    //Ẩn groupBox chi tiết
    HienChiTiet(false);
}

```

2.3. Chức năng nút Tìm kiếm

- Sử dụng sự kiện Click của nút btnTimKiem

```

private void btnTimKiem_Click(object sender, EventArgs e)
{
    //Cập nhật trên nhãn tiêu đề
    lblTieuDe.Text = "TÌM KIẾM MẶT HÀNG";
    //Cấm nút Sửa và Xóa
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
    //Viết câu lệnh SQL cho tìm kiếm
    string sql = "SELECT * FROM tblMatHang where MaSP is not null ";
    //Tìm theo MaSP khác rỗng
    if (txtTKMaSP.Text.Trim() != "")
    {
        sql += " and MaSP like '%" + txtTKMaSP.Text + "%'";
    }
    //kiểm tra TenSP
    if (txtTKTenSP.Text.Trim() != "" )
    {
        sql += " AND TenSP like N'" + txtTKTenSP.Text + "%'";
    }
    //Load dữ liệu tìm được lên dataGridView
    dgvKetQua.DataSource = dtbase.DataReader(sql);
}

```

*Khi click vào từng dòng trong dataGridView thì hiển thị thông tin lên groupBoxChiTiet và cho hiện nút Sửa, xóa, ẩn nút Thêm

→ Ta viết sự kiện CellClick cho dataGridView như sau:

```

private void dgvKetQua_CellClick(object sender, DataGridViewCellEventArgs e)
{
    //Hiện thi nút sửa
    btnSua.Enabled = true;
    btnXoa.Enabled = true;
    btnThem.Enabled = false;
    //Bắt lỗi khi người sử dụng kích linh tinh lên datagrid
    try
    {
        txtMaSP.Text = dgvKetQua.CurrentRow.Cells[0].Value.ToString();
        txtTenSP.Text = dgvKetQua.CurrentRow.Cells[1].Value.ToString();
        dtpNgaySX.Value = (DateTime)dgvKetQua.CurrentRow.Cells[2].Value;
        dtpNgayHH.Value = (DateTime)dgvKetQua.CurrentRow.Cells[3].Value;
        txtDonVi.Text = dgvKetQua.CurrentRow.Cells[4].Value.ToString();
        txtDonGia.Text = dgvKetQua.CurrentRow.Cells[5].Value.ToString();
        txtGhiChu.Text = dgvKetQua.CurrentRow.Cells[6].Value.ToString();
    }
    catch (Exception ex)
    {
    }
}

```

2.4. Chức năng thêm mới mặt hàng

- Viết sự kiện Click nút Thêm
- Khi nhấn nút thêm cần xóa trắng chi tiết để nhập mới

```
private void XoaTrangChiTiet()
{
    txtMaSP.Text = "";
    txtTenSP.Text = "";
    dtpNgaySX.Value = DateTime.Today;
    dtpNgayHH.Value = DateTime.Today;
    txtDonVi.Text = "";
    txtDonGia.Text = "";
    txtGhiChu.Text = "";
}

private void btnThem_Click(object sender, EventArgs e)
{
    lblTieuDe.Text = "THÊM MẶT HÀNG";
    //Xoa trang GroupBox chi tiết sản phẩm
    XoaTrangChiTiet();
    //Cam nút sua xoa
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
    //Hiện GroupBox Chi tiết
    HienChiTiet(true);
}
```

2.5. Chức năng nút Sửa

- Viết sự kiện Click cho nút Sửa.

```
private void btnSua_Click(object sender, EventArgs e)
{
    //Cập nhật tiêu đề
    lblTieuDe.Text = "CẬP NHẬT MẶT HÀNG";
    //Ẩn hai nút Thêm và Xóa
    btnThem.Enabled = false;
    btnXoa.Enabled = false;
    //Hiện gropbox chi tiết
    HienChiTiet(true);
}
```

2.6. Chức năng nút Xóa

- Viết sự kiện Click của nút Xóa

```
private void btnXoa_Click(object sender, EventArgs e)
{
    //Bật Message Box cảnh báo người sử dụng
    if (MessageBox.Show("Bạn có chắc chắn xóa mã mặt hàng " +
        txtMaSP.Text + " không ? Nếu có ấn nút Lưu, không thì ấn nút Hủy",
        "Xóa sản phẩm", MessageBoxButtons.YesNo) == DialogResult.Yes)
    {
        lblTieuDe.Text = "XÓA MẶT HÀNG";
        btnThem.Enabled = false;
    }
}
```



```

        btnSua.Enabled = false;
        //Hiện groupbox chi tiết
        HienChiTiet(true);
    }
}

```

2.7. Chức năng nút Lưu

- Chức năng Lưu, chúng ta dựa vào trạng thái Enable của các nút Thêm, Sửa hoặc Xóa để xác nhận trạng thái thực thi câu lệnh SQL

- Viết sự kiện Click của nút Lưu

```

private void btnLuu_Click(object sender, EventArgs e)
{
    string sql="";
    //Chúng ta sử dụng control ErrorProvider để hiển thị lỗi
    //Kiểm tra tên sản phẩm có bị để trống không
    if (txtTenSP.Text.Trim() == "")
    {
        errChiTiet.SetError(txtTenSP, "Bạn không để trống tên sản phẩm!");
        return;
    }
    else
    {
        errChiTiet.Clear();
    }
    //Kiểm tra ngày sản xuất, lỗi nếu người sử dụng nhập vào ngày
    sản xuất lớn hơn ngày hiện tại
    if (dtpNgaySX.Value > DateTime.Now)
    {
        errChiTiet.SetError(dtpNgaySX, "Ngày sản xuất không hợp lệ!");
        return;
    }
    else
    {
        errChiTiet.Clear();
    }
    //Kiểm tra ngày hết hạn xem có lớn hơn ngày sản xuất không
    if (dtpNgayHH.Value < dtpNgaySX.Value)
    {
        errChiTiet.SetError(dtpNgayHH, "Ngày hết hạn nhỏ hơn ngày sản
        xuất!");
        return;
    }
    else
    {
        errChiTiet.Clear();
    }
    //Kiểm tra đơn vị xem có để trống không
    if (txtDonVi.Text.Trim() == "")
    { errChiTiet.SetError(txtDonVi, "Bạn không để trống đơn vị!");
      return;
    }
    else
    {

```

```

        errChiTiet.Clear();
    }
    //Kiểm tra đơn giá
    if (txtDonGia.Text.Trim() == "")
    {
        errChiTiet.SetError(txtDonGia, "Bạn không để trống đơn giá!");
        return;
    }
    else
    {
        errChiTiet.Clear();
    }
    //Nếu nút Thêm enable thì thực hiện thêm mới
    //Dùng ký tự N' trước mỗi giá trị kiểu text để insert giá trị có dấu
    tiếng viết vào CSDL được đúng
    if(btnThem.Enabled == true)
    { //Kiểm tra xem ô nhập MaSP có bị trống không if
        if(txtMaSP.Text.Trim() == "")
        {
            errChiTiet.SetError(txtMaSP, "Bạn không để trống mã sản phẩm
trường này!");
            return;
        }
        else { //Kiểm tra xem mã sản phẩm đã tồn tại chưa để tránh
việc insert mới bị lỗi
            sql = "Select * From tblMatHang Where MaSP =" + txtMaSP.Text +
"";

            DataTable dtSP = dtbase.DataReader(sql);
            if (dtSP.Rows.Count > 0)
            {
                errChiTiet.SetError(txtMaSP, "Mã sản phẩm trùng trong cơ sở dữ
liệu");
                return;
            }
            errChiTiet.Clear();
        }
        //Insert vào CSDL
        sql = "INSERT INTO tblMatHang(MaSP, TenSP, NgaySX, NgayHH, DonVi,
DonGia, GhiChu) VALUES(";
        sql += "N'" + txtMaSP.Text + "',N'" + txtTenSP.Text + "'," +
dtpNgaySX.Value.Date + "'," +
dtpNgayHH.Value.Date + "',N'" + txtDonVi.Text + "',N'" +
txtDonGia.Text + "',N'" + txtGhiChu.Text + "')";
    }
    //Nếu nút Sửa enable thì thực hiện cập nhật dữ liệu
    if (btnSua.Enabled == true)
    {
        sql = "Update tblMatHang SET ";
        sql += "TenSP = N'" + txtTenSP.Text + "',";
        sql += "NgaySX = '" + dtpNgaySX.Value.Date + "',";
        sql += "NgayHH = '" + dtpNgayHH.Value.Date + "',";
        sql += "DonVi = N'" + txtDonVi.Text + "',";
        sql += "DonGia = '" + txtDonGia.Text + "',";
        sql += "GhiChu = N'" + txtGhiChu.Text + "' ";
        sql += "Where MaSP = N'" + txtMaSP.Text + "'";
    }
    //Nếu nút Xóa enable thì thực hiện xóa dữ liệu
    if (btnXoa.Enabled == true)

```

```

    {
        sql = "Delete From tblMathHang Where MaSP =N'" + txtMaSP.Text + "'";
    }
    dtbase.DataChange(sql);
    //Cap nhat lai DataGrid
    sql = "Select * from tblMathHang";
    dgvKetQua.DataSource = dtbase.DataReader(sql);
    //Ẩn hiện các nút phù hợp chức năng
    HienChiTiet(false);
    btnSua.Enabled = false;
    btnXoa.Enabled = false;
}

```

2.8. Chức năng nút Hủy

```

private void btnHuy_Click(object sender, EventArgs e)
{
    //Thiết lập lại các nút như ban đầu
    btnXoa.Enabled = false;
    btnSua.Enabled = false;
    btnThem.Enabled = true;
    //xoa trang chi tiet
    XoaTrangChiTiet();
    //Cam nhap vào groupBox chi tiet
    HienChiTiet(false);
}

```

2.9. Chức năng nút Thoát

```

private void btnThoat_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Bạn có muốn thoát không?", "TB", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
        this.Close();
}

```