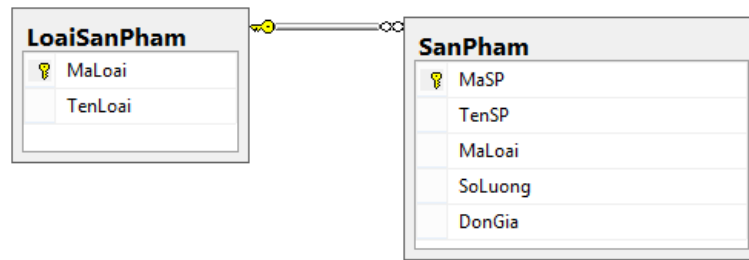


BÀI ÔN TẬP

Tạo cơ sở dữ liệu **QLBanHang** gồm 2 bảng **LoaiSanPham** và **SanPham** có lược đồ cơ sở dữ liệu như sau:



Thiết kế của 2 bảng:

	Column Name	Data Type	Allow Nulls
🔑	MaLoai	char(3)	<input type="checkbox"/>
	TenLoai	nvarchar(50)	<input type="checkbox"/>

	Column Name	Data Type	Allow Nulls
🔑	MaSP	char(4)	<input type="checkbox"/>
	TenSP	nvarchar(50)	<input type="checkbox"/>
	MaLoai	char(3)	<input checked="" type="checkbox"/>
	SoLuong	int	<input checked="" type="checkbox"/>
	DonGia	int	<input checked="" type="checkbox"/>

Câu 1: Sử dụng WPF thực hiện các yêu cầu sau:

a. Thiết kế giao diện như sau:

Cập nhật bảng sản phẩm

Mã sản phẩm:

Tên sản phẩm:

Loại sản phẩm:

Quần áo ▾

Đơn giá:

Số lượng:

Mã SP	Tên SP	Mã loại SP	Số lượng	Đơn giá	Thành tiền
01	Áo sơ mi nam	01	12	500	6000
02	Quần bò nữ	01	200	600	120000
04	Lò vi sóng	02	400	700	280000
03	Nồi cơm điện	02	100	1000	100000

Thêm

Sửa

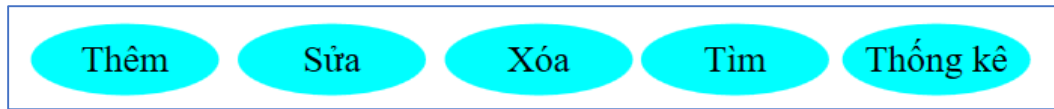
Xóa

Tìm

Thống kê

b. Tạo Style và Template thực hiện các yêu cầu sau:

- Dữ liệu nhập vào các Textbox trên Form có font chữ: **Times New Roman**, cỡ chữ: 20, kiểu chữ: *Italic*.
- Khi di chuyển chuột qua các Textbox thì dữ liệu nhập trên các Textbox hiển thị màu đỏ.
- Thiết lập các điều khiển Button có hình Ellip màu Cyan.



Câu 2: Viết chương trình có áp dụng LinQ và Entity Framework Core (EF) để thực hiện các yêu cầu sau:

- Hiện thị các **SanPham** trong DataGridView ngay khi hiển thị window. Thông tin gồm: mã sản phẩm, tên sản phẩm, mã loại sản phẩm, số lượng, đơn giá, thành tiền. Dữ liệu được sắp xếp theo đơn giá tăng dần.
- Sử dụng **combo box** hiển thị tên loại hàng từ bảng **LoaiSanPham** nhưng khi chọn lấy mã loại để cập nhật dữ liệu.
- Khi chọn 1 dòng dữ liệu trên DataGridView thì dữ liệu đó sẽ được điền vào các ô điều khiển.
- Thêm 1 bản ghi vào bảng SanPham khi nhấn nút **Thêm**, dữ liệu do người dùng nhập qua các điều khiển nhập liệu. **Xử lý ngoại lệ và kiểm tra dữ liệu người dùng nhập:**
 - Dữ liệu phải nhập vào đầy đủ, không để trống.
 - Không cho nhập trùng mã sản phẩm
 - Kiểm tra số lượng và đơn giá phải là số nguyên và > 0 .
 - Hiện thị lại dữ liệu sau khi thêm trong DataGridView.
- Sửa thông tin sản phẩm được chọn trong danh sách khi nhấn nút **Sửa**, dữ liệu do người dùng nhập qua các điều khiển nhập liệu, không sửa mã sản phẩm. Lưu thông tin đã sửa vào csdl và hiển thị lại dữ liệu trong DataGridView. **Xử lý ngoại lệ và kiểm tra dữ liệu người dùng sửa phải hợp lệ.**
- Xóa sản phẩm được chọn trong danh sách khi nhấn nút **Xóa**.
 - **Hiện thị thông báo khi không thấy sản phẩm để xóa.**
 - **Hiện thị thông báo xác nhận trước khi xóa.**
 - Hiện thị lại dữ liệu sau khi xóa trong DataGridView.
- Chọn 1 loại sản phẩm trong **ComboBox**, sau đó nhấn nút **Tìm**: Hiện thị thông tin của các sản phẩm tìm được lên DataGridView, gồm các cột: mã sản phẩm, tên sản phẩm, mã loại sản phẩm, số lượng, đơn giá, thành tiền.
- Khi nhấn nút **Thống kê**: Hiện thị thông tin từ 2 bảng tất cả các sản phẩm có **Mã loại = 11** lên DataGridView **trên cửa sổ khác**, bao gồm các cột: Mã sản phẩm, Tên sản phẩm, Tên loại sản phẩm, Số lượng, Đơn giá, Thành tiền.

Mã SP	Tên sản phẩm	Tên loại sản phẩm	Số lượng	Đơn giá	Thành tiền
sp1	Tivi SamSung 42'	Điện tử	100	4000000	400000000
sp2	Máy giặt Tosiba	Điện tử	120	5000000	600000000

* Code XAML xây dựng giao diện:

```
<Window x:Class="DemoEFCore.CapNhatSanPham"
    Title="Cập nhật bảng sản phẩm" Height="550" Width="800"
    FontFamily="Times New Roman" FontSize="20" Loaded="Window_Loaded">

<!--Khai báo Style và Template-->

<!--Sử dụng Stack Panel làm Panel chính-->
<StackPanel>
    <!--Khai báo Style và Template trong StackPanel-->
    <StackPanel.Resources>
        <!--Trong trường hợp này, có 2 Style và 1 Template-->
        <!--(1) Style quy định về kiểu phông chữ, áp dụng với Control-->
        <Style x:Key="baseStyle" TargetType="{x:Type Control}">
            <Setter Property="FontFamily" Value="Times New Roman" />
            <Setter Property="FontSize" Value="20" />
            <Setter Property="FontStyle" Value="Italic" />
        </Style>
        <!--(2) Style kế thừa từ Style trước, quy định với Control TextBox -->
        <Style BasedOn="{StaticResource baseStyle}" TargetType="{x:Type TextBox}">
            <!--Khai báo trigger-->
            <Style.Triggers>
                <!--Sự kiện khi con trỏ chuột lướt qua-->
                <Trigger Property="IsMouseOver" Value="True">
                    <Setter Property="Foreground" Value="Red" />
                </Trigger>
            </Style.Triggers>
        </Style>
        <!--(3) Thiết lập Template cho điều khiển Button-->
        <Style TargetType="Button">
            <Setter Property="OverridesDefaultStyle" Value="True"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Grid>
                            <Ellipse Fill="Cyan"/>
                            <!--Đánh dấu nơi bắt đầu đặt nội dung của Button: chính giữa-->
                            <ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>
                        </Grid>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </StackPanel.Resources>
```

<!--Khai báo phần tử trên giao diện-->

```
<Grid>
    <!--Code Xaml thiết kế các phần tử: Label, Textbox, Combobox--> ...
<DataGrid x:Name="dgvSanPham" ...
    AutoGenerateColumns="False" AlternatingRowBackground="AntiqueWhite" CanUserAddRows="False">
    <DataGrid.Columns>
        <DataGridTextColumn Header="Mã SP" Binding="{Binding MaSp}" />
        <DataGridTextColumn Header="Tên SP" Binding="{Binding TenSp}" />
        <DataGridTextColumn Header="Mã loại SP" Binding="{Binding MaLoai}" />
        <DataGridTextColumn Header="Số lượng" Binding="{Binding SoLuong}" />
        <DataGridTextColumn Header="Đơn giá" Binding="{Binding DonGia}" />
        <DataGridTextColumn Header="Thành tiền" Binding="{Binding ThanhTien}" Width="*/>
    </DataGrid.Columns>
</DataGrid>

    <!--Code Xaml thiết kế các phần tử botton: Thêm, Sửa, Xóa, Tìm, Thống kê--> ...
</Grid>
</StackPanel>
</Window>
```

* Code C# xử lý:

```
...
using EF_Core_QLBH.Models;
using System.Text.RegularExpressions; //Dùng khi Check DL
using System.Reflection; //Dùng khi chọn dòng
namespace EF_Core_QLBH
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            //Tạo đối tượng trở tới Model
            QLBanHangContext db = new QLBanHangContext();

            //Hàm Load dữ liệu lên DataGridView
            private void HienThiDuLieu()
            {
                var query = from sp in db.SanPhams
                            orderby sp.DonGia
                            select new
                            {
                                sp.MaSp,
                                sp.TenSp,
                                sp.MaLoai,
                                sp.SoLuong,
                                sp.DonGia,
                                ThanhTien = sp.SoLuong * sp.DonGia
                            };
                dgvSanPham.ItemsSource = query.ToList();
            }

            //Hàm Load dữ liệu lên DataGridView
            private void HienThiCB()
            {
                var query= from lsp in db.LoaiSanPhams
                            select lsp;
                cboLoai.ItemsSource = query.ToList();
                cboLoai.DisplayMemberPath = "TenLoai";
                cboLoai.SelectedValuePath="MaLoai";
                cboLoai.SelectedIndex = 0;
            }

            private void Window_Loaded(object sender, RoutedEventArgs e)
            {
                HienThiDuLieu();
                HienThiCB();
            }

            //Hàm Check dữ liệu
            private bool CheckDL()
            {
                string tb = "";
                if (txtMa.Text==" " || txtTen.Text==" " || txtSoLuong.Text==" " || txtDonGia.Text==" ")
                {
                    tb += "\n Ban can nhap day du du lieu!";
                }
                if (!Regex.IsMatch(txtDonGia.Text,@"\d+"))
                {
                    tb += "\n Don gia nhap vao phai la so nguyen!";
                }
                else
                {
                    int dg = int.Parse(txtDonGia.Text);
                    if (dg<0)
                    {

```

```

        tb += "\n Don gia nhap vao phai la so duong!";
    }
}
if (!Regex.IsMatch(txtSoLuong.Text, @"\d+"))
{
    tb += "\n So luong nhap vao phai la so nguyen!";
}
else
{
    int sl = int.Parse(txtSoLuong.Text);
    if (sl < 0)
    {
        tb += "\n So luong nhap vao phai la so duong!";
    }
}
if (tb != "")
{
    MessageBox.Show(tb, "Thong Bao", MessageBoxButton.OK, MessageBoxImage.Error);
    return false;
}
return true;
}

private void btnThem_Click(object sender, RoutedEventArgs e)
{
    //Kiểm tra không cho nhập trùng mã sản phẩm
    var query = db.SanPhams.SingleOrDefault(t => t.MaSp.Equals(txtMa.Text));
    if (query != null)
    {
        MessageBox.Show("Mã sản phẩm này đã tồn tại!", "Thong Bao");
        HienThiDuLieu();
    }
    else
    {
        try
        {
            if (CheckDL())
            {
                SanPham spMoi = new SanPham();
                spMoi.MaSp = txtMa.Text;
                spMoi.TenSp = txtTen.Text;
                spMoi.DonGia = int.Parse(txtDonGia.Text);
                spMoi.SoLuong = int.Parse(txtSoLuong.Text);
                LoaiSanPham itemSelected = (LoaiSanPham)cboLoai.SelectedItem;
                spMoi.MaLoai = itemSelected.MaLoai;
                db.SanPhams.Add(spMoi);
                db.SaveChanges();
                MessageBox.Show("Thêm thành công!", "Thong Bao");
                HienThiDuLieu();
            }
        }
        catch (Exception e1)
        {
            MessageBox.Show("Có lỗi khi thêm: " + e1.Message, "Thong Bao");
        }
    }
}

private void btnSua_Click(object sender, RoutedEventArgs e)
{
    try
    {
        var spSua = db.SanPhams.SingleOrDefault(t => t.MaSp.Equals(txtMa.Text));
        if (spSua != null)
        {
            if (CheckDL())
            {
                spSua.TenSp = txtTen.Text;
                LoaiSanPham itemSelected = (LoaiSanPham)cboLoai.SelectedItem;
                spSua.MaLoai = itemSelected.MaLoai;
                spSua.DonGia = int.Parse(txtDonGia.Text);
            }
        }
    }
}

```

```

        spSua.SoLuong = int.Parse(txtSoLuong.Text);
        db.SaveChanges();
        MessageBox.Show("Sửa thành công!", "Thông Báo");
        HienThiDuLieu();
    }
}
else
{
    MessageBox.Show("Không tìm thấy sản phẩm cần sửa!");
}
}
catch (Exception e1)
{
    MessageBox.Show("Có lỗi khi sửa: " + e1.Message, "Thông Báo");
}
}

private void btnXoa_Click(object sender, RoutedEventArgs e)
{
    var spXoa = db.SanPhams.SingleOrDefault(t => t.MaSp.Equals(txtMa.Text));
    if (spXoa != null)
    {
        MessageBoxResult rs = MessageBox.Show("Bạn có chắc chắn muốn xóa?", "Thông Báo",
        MessageBoxButton.YesNo);
        if (rs == MessageBoxResult.Yes)
        {
            db.SanPhams.Remove(spXoa);
            db.SaveChanges();
            HienThiDuLieu();
        }
    }
    else
    {
        MessageBox.Show("Không có sản phẩm này để xóa! ", "Thông báo");
    }
}

//Chọn dòng trong DataGridView
private void dgvSanPham_SelectedCellsChanged(object sender, SelectedCellsChangedEventArgs e)
{
    if (dgvSanPham.SelectedItem != null)
    {
        try
        {
            Type t = dgvSanPham.SelectedItem.GetType();
            PropertyInfo[] p = t.GetProperties();
            txtMa.Text = p[0].GetValue(dgvSanPham.SelectedValue).ToString();
            txtTen.Text = p[1].GetValue(dgvSanPham.SelectedValue).ToString();
            cboLoai.SelectedValue = p[2].GetValue(dgvSanPham.SelectedValue).ToString();
            txtSoLuong.Text = p[3].GetValue(dgvSanPham.SelectedValue).ToString();
            txtDonGia.Text = p[4].GetValue(dgvSanPham.SelectedValue).ToString();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Có lỗi khi chọn hàng" + ex.Message, "Thông báo");
        }
    }
}

private void btnThongke_Click(object sender, RoutedEventArgs e)
{
    Window2 myWindow = new Window2();
    myWindow.Show();
}
}

```

* Code XAML Window2:

```
<Grid>
<DataGrid Name="dgvThongke" Margin="0,66,0,86"
          AutoGenerateColumns="False"
          AlternatingRowBackground="LightBlue"
          CanUserAddRows="False" >
    <DataGrid.Columns>
        <DataGridTextColumn Header="Mã SP" Binding="{Binding MaSp}" />
        <DataGridTextColumn Header="Tên sản phẩm" Binding="{Binding TenSp}" />
        <DataGridTextColumn Header="Tên loại sản phẩm" Binding="{Binding TenLoai}" />
        <DataGridTextColumn Header="Số lượng" Binding="{Binding SoLuong}" />
        <DataGridTextColumn Header="Đơn giá" Binding="{Binding DonGia}" />
        <DataGridTextColumn Header="Thành tiền" Binding="{Binding ThanhTien}" Width="*" />
    </DataGrid.Columns>
</DataGrid>
</Grid>
```

* Code C# Window2:

using DemoEFCore.Models;

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    QLBanHangContext db = new QLBanHangContext();
    var query = from p in db.SanPhams
                join k in db.LoaiSanPhams
                on p.MaLoai equals k.MaLoai
                where p.MaLoai=="11"
                select new
                {
                    p.MaSp,
                    p.TenSp,
                    k.TenLoai,
                    p.DonGia,
                    p.SoLuong,
                    ThanhTien = p.DonGia * p.SoLuong
                };
    dgvThongke.ItemsSource = query.ToList();
}
```