

Chuẩn bị:

- Sử dụng SQL Server, chạy script đã cho để tạo CSDL QLSach gồm 2 bảng:

TacGia(**MaTG**,TenTacGia)

Sach(**MaSach**,TenSach,NamXuatBan,SoTrang,MaTG)

- Sử dụng .NET5 để viết ứng dụng WPF

- Tạo Solution tên là: Họ tên sinh viên_3 số cuối mã sinh viên, ví dụ: TranVanA_123

Nộp bài: Kiểm tra đảm bảo file nén chứa Solution và file word câu 2a. Nộp theo hướng dẫn của giám thị.

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

a) (L3, 1đ) Thiết kế giao diện như sau:

Mã sách:	<input type="text"/>
Tên sách:	<input type="text"/>
Năm xuất bản:	<input type="text"/>
Số Trang:	<input type="text"/>
Tác giả:	<input type="text"/>

Mã sách	Tên sách	Mã tác giả	Năm xuất bản	Số trang	Thành tiền

Thêm

Sửa

Xoá

Thống kê

b) (L3, 1đ) Tạo Style và Template thực hiện các yêu cầu sau:

- Dữ liệu nhập vào các TextBox trên Form có font chữ: Time New Roman, màu chữ: Blue

- Thiết lập các điều khiển Button có hình Ellip, màu của các Button là: YellowGreen

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

a) (L3, 1đ) Tạo model cho CSDL QLSach (Sử dụng thư viện Microsoft.EntityFrameworkCore.SqlServer và Microsoft.EntityFrameworkCore.Tools phiên bản 5.0.7). Sau đó thêm mới bảng NhaXuatBan(MaNXB, TenNhaXB, SoDT) vào CSDL trên. Cập nhật các lớp trong Models để ánh xạ CSDL QLSach sau khi thêm bảng. Lưu nội dung cập nhật trong Models ra file word có tên là Cau2a.docx

b) (L3, 2đ) Hiển thị các sách có số trang ≥ 120 trong DataGrid ngay khi hiển thị cửa sổ. Thông tin gồm mã sách, tên sách, mã tác giả, năm xuất bản, số trang, thành tiền (thành tiền = số trang * 80000). Dữ liệu được sắp xếp theo số trang giảm dần.

- DataGrid được định dạng như sau:

- Tiêu đề cột: Chữ đậm, căn giữa

- Dữ liệu cột thành tiền: căn phải, chữ nghiêng, màu đỏ, có phân cách thành số dạng hàng nghìn, hàng triệu.

c) (L3, 1đ) Thêm 1 bản ghi vào bảng Sach khi ấn 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: Số trang và năm xuất bản phải là số nguyên và >0 .

- Hiển thị lại dữ liệu sau khi cập nhật vào CSDL lên DataGrid

d) (L3, 1đ) Sửa thông tin sách được chọn trong danh sách khi nhấn nút Sửa, dữ liệu do người dùng nhập qua các điều khiển nhập liệu, không sửa mã sách.

- Xử lý ngoại lệ và kiểm tra dữ liệu người dùng nhập: Số trang và năm xuất bản phải là số nguyên và >0 .

- Hiển thị lại dữ liệu sau khi cập nhật vào CSDL lên DataGrid

e) (L3, 1đ) Xóa sách được chọn trong danh sách khi ấn nút Xóa.

- Hiển thị thông báo xác nhận trước khi xóa.

- Hiển thị lại dữ liệu sau khi cập nhật vào CSDL lên DataGrid

f) (L3, 1đ) Sử dụng ComboBox hiển thị tên tác giả từ bảng TacGia, nhưng khi chọn tên thì lấy mã TG tương ứng để cập nhật dữ liệu.

g) (L3, 1đ) Khi nhấn nút Thống Kê: hiển thị tổng tiền sách của từng tác giả lên DataGrid trên cửa sổ khác, gồm các cột: Mã TG, Tên TG, Tổng tiền.

B1. Tạo CSDL QLSach: Mở SQL Server, Mở Query Analyse

- Chạy script (Hoặc click chuột phải tạo Bảng, Database, chèn dữ liệu test):

Create database QLSach

go

Use QLSach

Go

Create table TacGia(MaTG nchar(10) not null primary key,TenTacGia nvarchar(50) not null)

Create table Sach(MaSach nchar(10) not null primary key, TenSach nvarchar(50) not null, NamXuatBan int, SoTrang int, MaTG nchar(10) constraint fk_S_TG foreign key(MaTG) References TacGia(MaTG))

Go

Insert into TacGia values('TG1',N'Lê Anh Tuấn'),

('TG2',N'Nguyễn Minh Tùng')

Insert into Sach values('S01',N'Thép đã nung thề đỏ',2020,130,'TG1'),

('S02',N'Túp lều của bác Tùng',2021,150,'TG2'),

('S03',N'Ông già và ao nhỏ',2022,100,'TG1')

B2. - Sử dụng .NET5 để viết ứng dụng WPF

- Tạo Solution tên là: Họ tên sinh viên_3 số cuối mã sinh viên, ví dụ: TranVanA_123

B3. Sử dụng WPF, thiết kế giao diện như sau:

Mã sách:	<input type="text"/>
Tên sách:	<input type="text"/>
Năm xuất bản:	<input type="text"/>
Số Trang:	<input type="text"/>
Tác giả:	<input type="text"/>

Mã sách	Tên sách	Mã tác giả	Năm xuất bản	Số trang	Thành tiền

Thêm

Sửa

Xoá

Thống kê

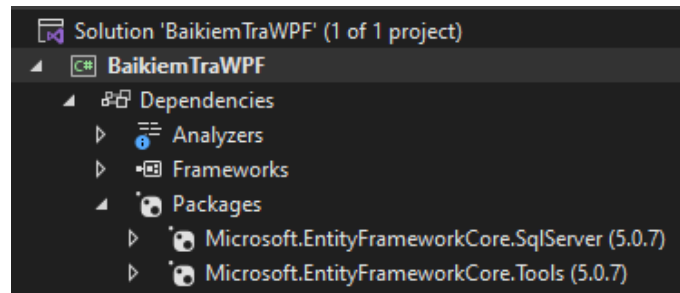
```

<Grid>
    <Label Content="Mã sách:" HorizontalAlignment="Left" Margin="66,71,0,0"
    VerticalAlignment="Top" FontSize="14"/>
    <Label Content="Tên sách:" HorizontalAlignment="Left" Margin="66,125,0,0"
    VerticalAlignment="Top" FontSize="14"/>
    <Label Content="Năm xuất bản:" HorizontalAlignment="Left"
    Margin="66,178,0,0" VerticalAlignment="Top" FontSize="14"/>
    <Label Content="Số Trang:" HorizontalAlignment="Left" Margin="66,243,0,0"
    VerticalAlignment="Top" FontSize="14"/>
    <Label Content="Tác giả:" HorizontalAlignment="Left" Margin="66,302,0,0"
    VerticalAlignment="Top" FontSize="14"/>
    <TextBox x:Name="txtMa" HorizontalAlignment="Left" Margin="187,71,0,0"
    TextWrapping="Wrap" VerticalAlignment="Top" Width="259" Height="31"/>
    <TextBox x:Name="txtName" HorizontalAlignment="Left" Margin="187,125,0,0"
    TextWrapping="Wrap" VerticalAlignment="Top" Width="259" Height="36"/>
    <TextBox x:Name="txtNamXB" HorizontalAlignment="Left" Margin="187,178,0,0"
    TextWrapping="Wrap" VerticalAlignment="Top" Width="259" Height="39"/>
    <TextBox x:Name="txtTrang" HorizontalAlignment="Left" Height="38"
    Margin="187,231,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="259"/>
    <ComboBox x:Name="cbTG" HorizontalAlignment="Left" Height="42"
    Margin="187,302,0,0" VerticalAlignment="Top" Width="259"/>
    <DataGrid x:Name="dgSach" AutoGenerateColumns="False" Margin="66,365,95,94"
    SelectedCellsChanged="dgSach_SelectedCellsChanged" FontSize="14">
        <DataGrid.Columns>
            <DataGridTextColumn Header="Mã sách" Binding="{Binding MaSach,
    Mode=OneWay}"></DataGridTextColumn>
            <DataGridTextColumn Header="Tên sách" Binding="{Binding TenSach,
    Mode=OneWay}" Width="150"></DataGridTextColumn>
            <DataGridTextColumn Header="Mã tác giả" Binding="{Binding MaTG
    ,Mode=OneWay}"></DataGridTextColumn>
            <DataGridTextColumn Header="Năm xuất bản" Binding="{Binding NamXB
    ,Mode=OneWay}"></DataGridTextColumn>
            <DataGridTextColumn Header="Số trang" Binding="{Binding SoTrang,
    Mode=OneWay}"></DataGridTextColumn>
            <DataGridTextColumn Header="Thành tiền" Binding="{Binding TongTien
    ,Mode=OneWay}" Foreground="Red" FontStyle="Italic" Width="*"></DataGridTextColumn>
        </DataGrid.Columns>
        <DataGrid.Resources>
            <Style TargetType="DataGridColumnHeader">
                <Setter Property="FontWeight" Value="Bold"></Setter>
            </Style>
        </DataGrid.Resources>
    </DataGrid>
    <Button Content="Thêm" x:Name="btnThem" HorizontalAlignment="Left"
    Height="30" Margin="86,500,0,0" VerticalAlignment="Top" Width="104"
    Click="btnThem_Click" FontSize="14"/>
    <Button Content="Sửa" x:Name="btnSua" HorizontalAlignment="Left" Height="30"
    Margin="248,500,0,0" VerticalAlignment="Top" Width="104" Click="btnSua_Click"
    FontSize="14"/>
    <Button Content="Xóa" x:Name="btnXoa" HorizontalAlignment="Left" Height="30"
    Margin="392,500,0,0" VerticalAlignment="Top" Width="104" Click="btnXoa_Click"
    FontSize="14"/>
    <Button Content="Tổng kê" x:Name="btnTK" HorizontalAlignment="Left"
    Height="30" Margin="554,500,0,0" VerticalAlignment="Top" Width="104"
    Click="btnTK_Click" FontSize="14"/>
</Grid>

```

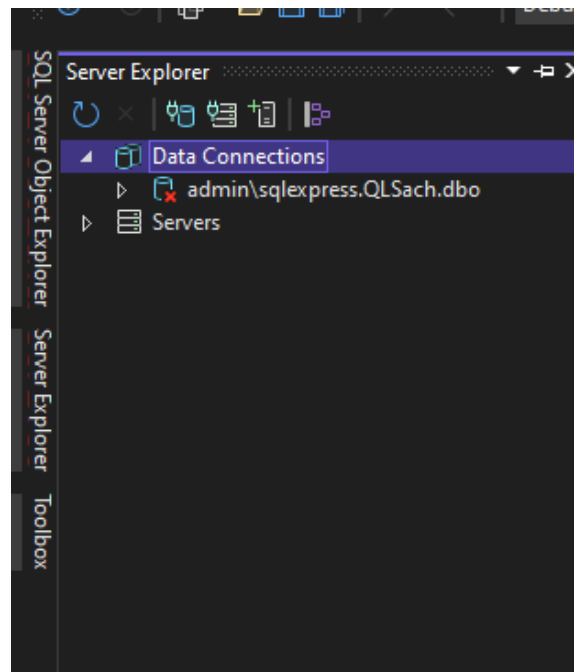
Style kiểu chữ cho
header DataGrid

B4. Cài đặt thư viện Microsoft.EntityFrameworkCore.SqlServer và Microsoft.EntityFrameworkCore.Tools phiên bản 5.0.7



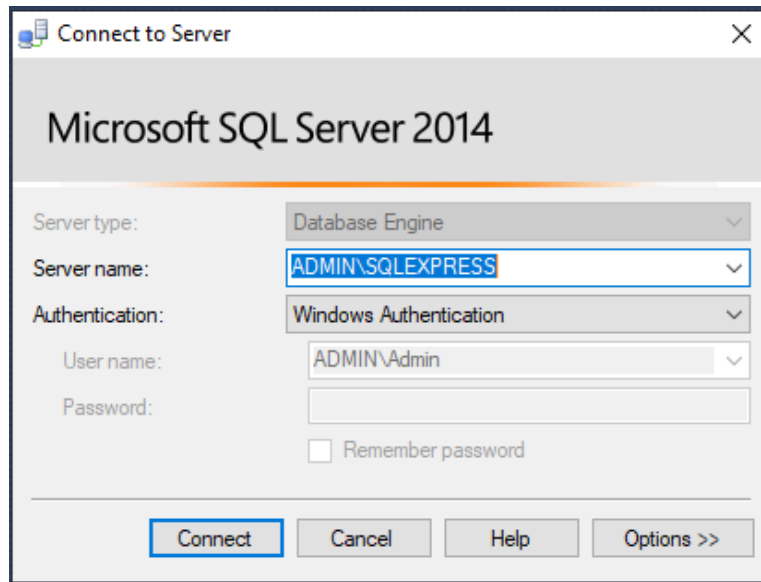
B5. Lấy đường dẫn Database QLSach trong CSDL để tạo models

- chọn Server Explorer

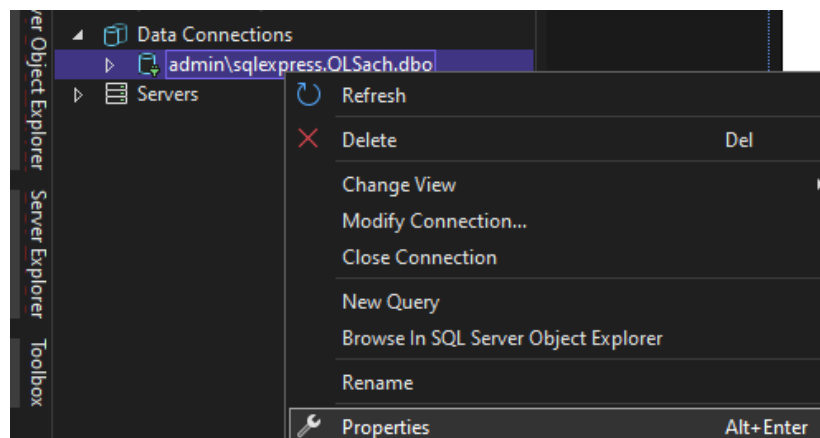


- Connect to database

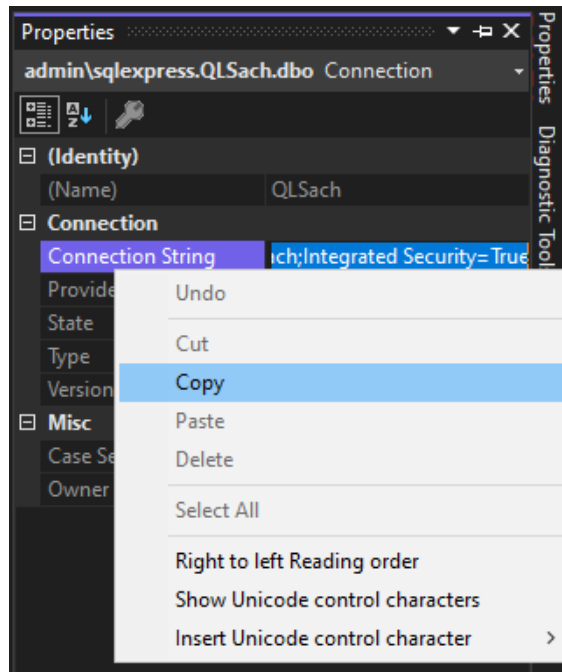
- Vào SQL Server copy Server Name: ADMIN\\SQLEXPRESS



- Paste vào dòng Server Name của VS
- Connect to Database: QLSach
- Test connection -> succeeded => OK => OK
- Lấy đường dẫn Database: Chọn Properties của kết nối tới Database vừa tạo



- Copy chuỗi Connection String:

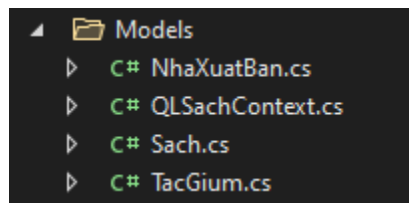


B6. Tạo Models cho CSDL QLSach:

- Tools -> NuGet Package Manager -> Package Manager Console

PM>Scaffold-DbContext "Paste chuỗi Connection string" Microsoft.EntityFrameworkCore.SqlServer - OutputDir Models

=> Sinh ra Model như sau:



B7. Thêm mới bảng NhaXuatBan(MaNXB, TenNhaXB, SoDT) vào CSDL trên.

- Mở SQL Server ra và tạo mới bảng NhaXuatBan trong CSDL QLBanHang:

Create table NhaXuatBan(MaNXB nchar(10) not null primary key, TenNhaXB nvarchar(50), SoDT nchar(10))

- Cập nhật các lớp trong Models để ánh xạ CSDL QLSach sau khi thêm bảng. Lưu nội dung cập nhật trong Models ra file word có tên là Cau2a.docx

1. Models-> Thêm 1 class có tên là NhaXuatBan.cs

- Edit lại NhaXuatBan.cs Nội dung như sau:

```
using System;
using System.Collections.Generic;

#nullable disable
```

```
namespace BaikiemTraWPF.Models
{
    public partial class NhaXuatBan
    {
        public string MaNXB { get; set; }
        public string TenNhaXB { get; set; }
        public string SoDT { get; set; }
    }
}
```

2. QLSachContext.cs

- Thêm dòng:

```
public virtual DbSet<Sach> Saches { get; set; }
public virtual DbSet<TacGium> TacGia { get; set; }
public virtual DbSet<NhaXuatBan> NhaXuatBans { get; set; }
```

- Thêm đoạn:

```
modelBuilder.Entity<NhaXuatBan>(entity =>
{
    entity.HasKey(e => e.MaNXB)
        .HasName("PK__NhaXuatB__3A19482CD75E965B");

    entity.ToTable("NhaXuatBan");

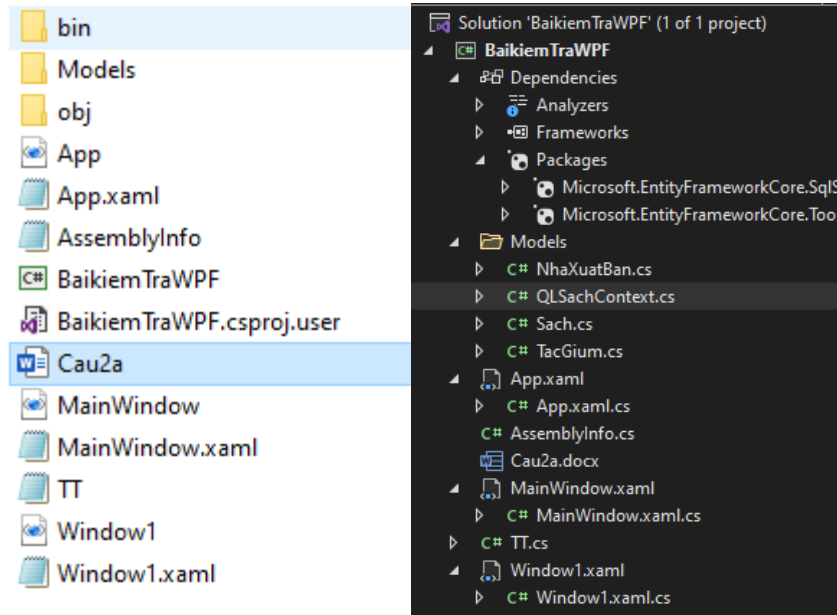
    entity.Property(e => e.MaNXB)
        .HasMaxLength(10)
        .HasColumnName("MaNXB");

    entity.Property(e => e.SoDT)
        .IsRequired()
        .HasMaxLength(11)
        .HasColumnName("SoDT");

    entity.Property(e => e.TenNhaXB)
        .IsRequired()
        .HasMaxLength(30)
        .HasColumnName("TenNhaXB");

});
```

- Toàn Bộ nội dung bước 7 này sẽ được lưu vào 1 file word có tên cau2a.docx, để ở cùng Folder với bài kiểm tra



B8. Hiển thị các sách có số trang ≥ 120 trong DataGrid ngay khi hiển thị cửa sổ. Thông tin gồm mã sách, tên sách, mã tác giả, năm xuất bản, số trang, thành tiền (thành tiền = số trang * 80000). Dữ liệu được sắp xếp theo số trang giảm dần.

- Click vào Thẻ Window trên màn hình XAML -> Click chuột phải trên đối tượng window -> Properties

-> chọn Event của Properties là: Window_Loaded -> Click double vào Ô Event này -> Sinh ra đoạn code của đối tượng này.

(Tương tự với các đối tượng khác)

- Sử dụng Models vừa tạo:

`using BaikiemTraWPF.Models;`

- Tạo đối tượng Trỏ tới Models:

```
using BaikiemTraWPF.Models;
using System.Text.RegularExpressions;
using System.Reflection;
namespace BaikiemTraWPF
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        QLSachContext db = new();
        public MainWindow()
        {

```

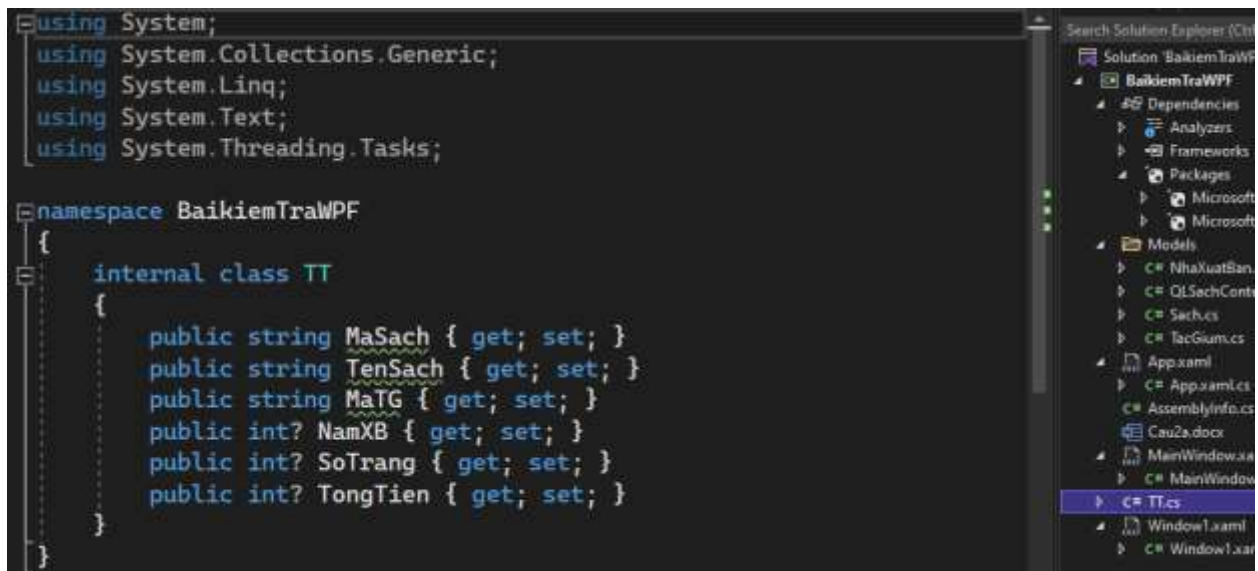
- Trên màn hình code đối với sự kiện này ta viết:

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

- Hàm HienThiDL() là Load dữ liệu từ Database -> DataGrid
- Hàm HienThiCB() là Load dữ liệu từ Database -> ComboBox

```
private void HienThiCB()
{
    var query = from t in db.TacGia
                select t;
    cbTG.ItemsSource = query.ToList();
    cbTG.DisplayMemberPath = "TenTacGia";
    cbTG.SelectedValuePath = "MaTG";
    cbTG.SelectedIndex = 0;
}
```

- Hiện thị TenTacGia, còn khi chọn TenTacGia nào -> Truy xuất MaTG tương ứng.
- Tạo 1 class có tên là TT.cs trên Project để lưu lại thông tin hiển thị trên DataGrid



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BaikiemTraWPF
{
    internal class TT
    {
        public string MaSach { get; set; }
        public string TenSach { get; set; }
        public string MaTG { get; set; }
        public int? NamXB { get; set; }
        public int? SoTrang { get; set; }
        public int? TongTien { get; set; }
    }
}
```

The Solution Explorer on the right shows the project structure for 'BaikiemTraWPF'. The 'Models' folder contains the 'TT.cs' file, which is currently selected.

- Hiện thị lên DataGrid:

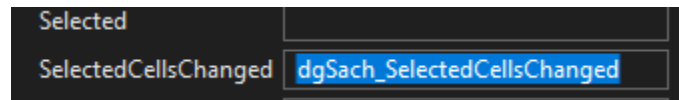
```

private List<TT> LayDL()
{
    var query = from t in db.Saches
                 where t.SoTrang >= 120
                 orderby t.SoTrang descending
                 select new TT
                 {
                     MaSach = t.MaSach,
                     TenSach = t.TenSach,
                     MaTG = t.MaTg,
                     NamXB = t.NamXuatBan,
                     SoTrang = t.SoTrang,
                     TongTien = t.SoTrang * 80000
                 };
    return query.ToList<TT>();
}
private void HienThiDL()
{
    dgSach.ItemsSource = LayDL();
}

```

B9. Click 1 dòng trên DataGrid -> Hiển thị thông tin trên các Control tương ứng:

- Chọn Event của DataGrid:



- Sử dụng 2 namespace:

```

using System.Text.RegularExpressions;
using System.Reflection;

```

```

private void dgSach_SelectedCellsChanged(object sender, SelectedCellsChangedEventArgs e)
{
    if(dgSach.SelectedItem!=null)
    {
        try
        {
            Type t= dgSach.SelectedItem.GetType();
            PropertyInfo[] p = t.GetProperties();
            txtMa.Text=p[0].GetValue(dgSach.SelectedValue).ToString();
            txtName.Text=p[1].GetValue(dgSach.SelectedValue).ToString();
            txtNamXB.Text=p[3].GetValue(dgSach.SelectedValue).ToString();
            txtTrang.Text=p[4].GetValue(dgSach.SelectedValue).ToString();
            cbTG.SelectedValue=p[2].GetValue(dgSach.SelectedValue).ToString();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Có lỗi khi chọn hàng"+ex.Message,"Thông báo",
                MessageBoxButton.OK,MessageBoxImage.Error);
        }
    }
}

```

B10. Check validation: Xử lý ngoại lệ và kiểm tra dữ liệu người dùng nhập: Số trang và năm xuất bản phải là số nguyên và >0.

- Tạo 1 hàm kiểm tra:

```
private bool CheckDL()
{
    string mess = "";
    if (txtMa.Text == "" || txtName.Text == "" || txtTrang.Text == "" || txtNamXB.Text == "")
    {
        mess += "\nBan can nhap day du du lieu";
    }
    if (!Regex.IsMatch(txtTrang.Text, @"\d+"))
    {
        mess += "\nSo trang nhap vao phai la so nguyen ";
    }
    else
    {
        int sl = int.Parse(txtTrang.Text);
        if (sl < 0)
        {
            mess += "\nSo luong nhap vao phai la so duong";
        }
    }
    if (!Regex.IsMatch(txtNamXB.Text, @"\d+"))
    {
        mess += "\nNam xuất bản nhap vao phai la so nguyen";
    }
    else
    {
        int gia = int.Parse(txtNamXB.Text);
        if (gia < 0)
        {
            mess += "\nNam xuất bản nhap vao phai la so duong";
        }
    }
    if (mess != "")
    {
        MessageBox.Show(mess, "Thong Bao", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    return true;
}
```

B11. Thêm mới dữ liệu

```

private void btnThem_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (CheckDL())
        {
            var query = db.Saches.SingleOrDefault(t=>t.MaSach.Equals(txtMa.Text));
            if (query != null)
            {
                MessageBox.Show("Da ton tai", "Thong bao", MessageBoxButton.OK);
                HienThiDL();
            }
            else
            {
                Sach a = new Sach();
                a.MaSach = txtMa.Text;
                a.TenSach = txtName.Text;
                a.SoTrang = int.Parse(txtTrang.Text);
                a.NamXuatBan = int.Parse(txtNamXB.Text);
                TacGium dm=(TacGium)cbTG.SelectedItem;
                a.MaTg = dm.MaTg;
                db.Saches.Add(a);
                db.SaveChanges();
                MessageBox.Show("Them thanh cong", "Thong bao", MessageBoxButton.OK);
                HienThiDL();
            }
        }
    }
    catch (Exception e1)
    {
        MessageBox.Show("Co loi khi them:" + e1.Message, "Thong bao", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}

```

B12. Sửa dữ liệu


```

private void btnSua_Click(object sender, RoutedEventArgs e)
{
    try
    {
        var spSua=db.Saches.SingleOrDefault(sp=>sp.MaSach==txtMa.Text);
        if (spSua != null)
        {
            if (CheckDL())
            {
                spSua.TenSach=txtName.Text;
                spSua.SoTrang = int.Parse(txtTrang.Text);
                spSua.NamXuatBan = int.Parse(txtNamXB.Text); ;
                spSua.MaTg=((TacGium)cbTG.SelectedItem).MaTg;
                db.SaveChanges();
                HienThiDL();
            }
        }
        else
        {
            MessageBox.Show("Không tìm thấy sản phẩm cần sửa");
        }
    }
    catch (Exception e1)
    {
        MessageBox.Show("Có lỗi khi sửa: " + e1.Message);
    }
}

```

B13. Xóa dữ liệu

```

private void btnXoa_Click(object sender, RoutedEventArgs e)
{
    try
    {
        var query = db.Saches.SingleOrDefault(t => t.MaSach.Equals(txtMa.Text));
        if (query != null)
        {
            MessageBoxResult rs = MessageBox.Show("Ban co chac chan muon xoa?", "Thong bao",
                MessageBoxButton.YesNoCancel);

            if (rs == MessageBoxResult.Yes)
            {
                db.Saches.Remove(query);
                db.SaveChanges();
                HienThiDL();
            }
        }
        else
        {
            MessageBox.Show("Chua ton tai thuoc can xoa", "Thong bao");
            HienThiDL();
        }
    }
    catch (Exception e1)
    {
        MessageBox.Show("Co loi khi xoa:" + e1.Message, "Thong bao",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
}

```

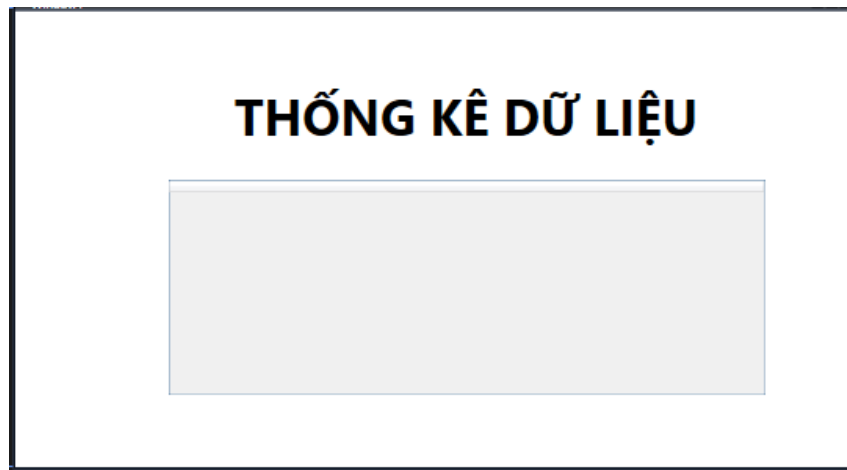
B14. Ấn vào Nút thống kê sẽ hiện ra Cửa sổ tiếp theo:

- Trên Project -> Add -> Window(WPF): Đặt tên là Window1.xaml

- Viết Code để gọi Window1 trên MainWindow.cs

```
private void btnTK_Click(object sender, RoutedEventArgs e)
{
    Window1 f = new Window1();
    f.ShowDialog();
}
```

B15. Design Window1



```
<Grid>
    <Label Content="THỐNG KÊ DỮ LIỆU" HorizontalAlignment="Left" Height="94"
Margin="191,62,0,0" VerticalAlignment="Top" Width="479"
HorizontalAlignment="Center" FontSize="48" FontWeight="Bold"/>
    <DataGrid x:Name="dgTG" HorizontalAlignment="Left" Height="205"
Margin="146,161,0,0" VerticalAlignment="Top" Width="570" />
</Grid>
```

B16. Hiển thị tổng tiền sách của từng tác giả lên DataGrid gồm các cột: Mã TG, Tên TG, Tổng tiền.

- Trên Window1.cs

```
using BaikiemTraWPF.Models;
```

- Sự kiện Window_Loaded:

```

private void Window_Loaded(Object sender, RoutedEventArgs e)
{
    QLSachContext db = new QLSachContext();
    var query = from t in db.Saches
                group t by t.MaTg into TGGR
                select new
                {
                    Matg = TGGR.Key,
                    TongTien = TGGR.Sum(x => x.SoTrang * 80000)
                };
    var query2 = from t in query
                 join s in db.TacGia on t.MaTg equals s.MaTg
                 select new
                 {
                     s.MaTg,
                     s.TenTacGia,
                     t.TongTien
                 };
    dgTG.ItemsSource = query2.ToList();
}

```

B17. Template, Style,... Các bạn về nghiên cứu thêm nhé... :D