Лабораторна робота №5

з дисципліни «Декларативне програмування графічних інтерфейсів»

«Модель "сутність-зв'язок" та її використання для організації роботи з базою даних»

Варіант №19

Виконав: Студент групи ТР-12

Ковальов Олександр

Перевірив: доцент, к.ф.-м.н.

Тарнавський Ю. А.

КИЇВ 2024

Мета роботи. Розробити програму для керування даними, що зберігаються в SQL-базі даних.

Завдання 1: Створення моделі даних

Скрипт для створення бази даних та таблиць:

```sql
CREATE DATABASE DPGI
GO;

USE DPGI
GO;

CREATE TABLE dbo.Publishers
(
    id   INT IDENTITY (1, 1),
    name VARCHAR(255) NOT NULL,

    PRIMARY KEY (id)
)
GO;

CREATE TABLE dbo.Books
(
    isbn             VARCHAR(30)  NOT NULL,
    title            VARCHAR(255) NOT NULL,
    authors          VARCHAR(255) NOT NULL,
    publisher_code   INT          NOT NULL,
    publication_year SMALLINT,

    PRIMARY KEY (isbn),

    CONSTRAINT FK_PublishersCode
        FOREIGN KEY (publisher_code) REFERENCES dbo.Publishers (id)
)
GO;
```

Скрипт для вставки даних:

```sql
USE DPGI
GO;

DROP TABLE IF EXISTS #BulkData
GO;

CREATE TABLE #BulkData
(
    isbn             VARCHAR(30),
    title            VARCHAR(255),
    authors          VARCHAR(255),
    publisher        VARCHAR(255),
    publication_year SMALLINT
)
GO;

BULK INSERT #BulkData
    FROM 'D:\Programming\KPI-DPGI\Lab5\Database\Data\Data.csv'
    WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2
    );
INSERT INTO dbo.Publishers (name)
SELECT DISTINCT publisher
FROM #BulkData
GO;

INSERT INTO dbo.Books (isbn, title, authors, publication_year, publisher_code)
SELECT bd.isbn,
       bd.title,
       bd.authors,
       bd.publication_year,
       pub.id AS publisher_code
FROM #BulkData AS bd
INNER JOIN dbo.Publishers AS pub
ON bd.publisher = pub.name;
```
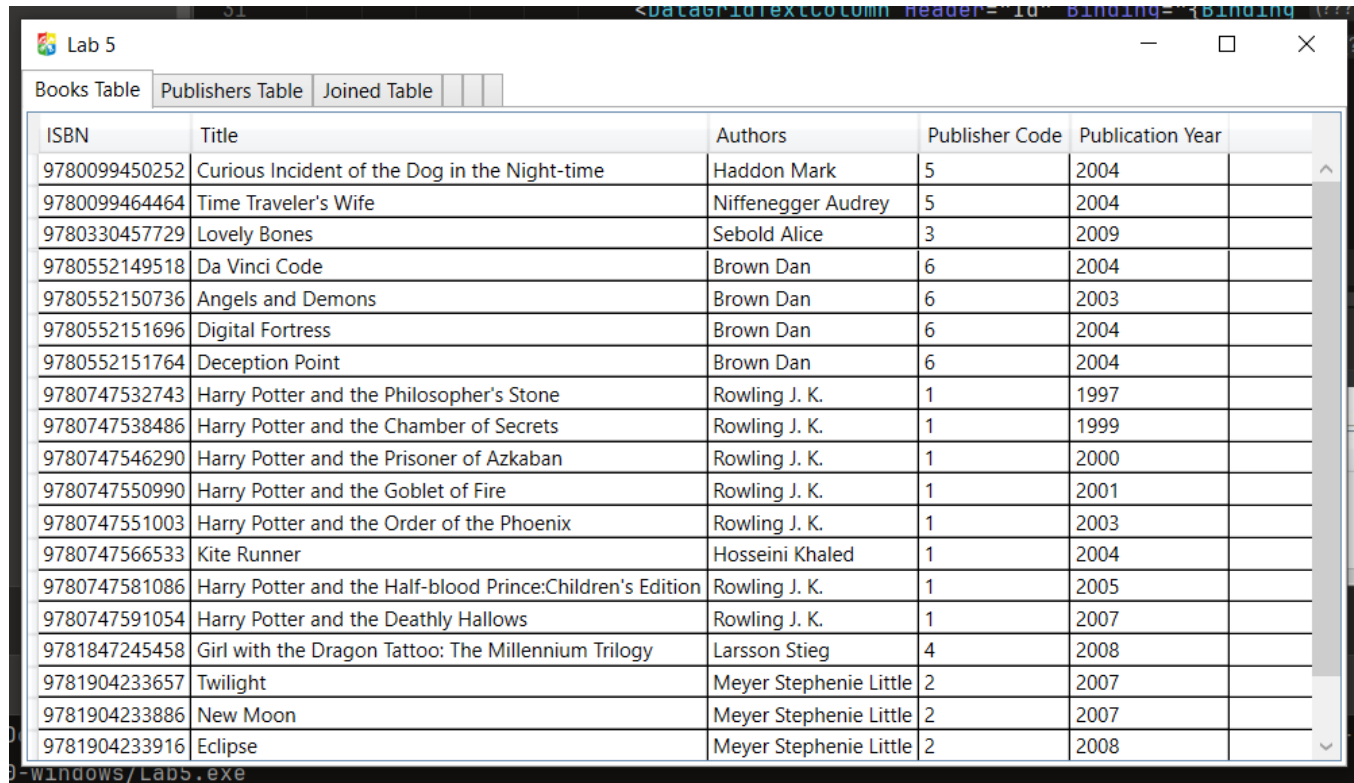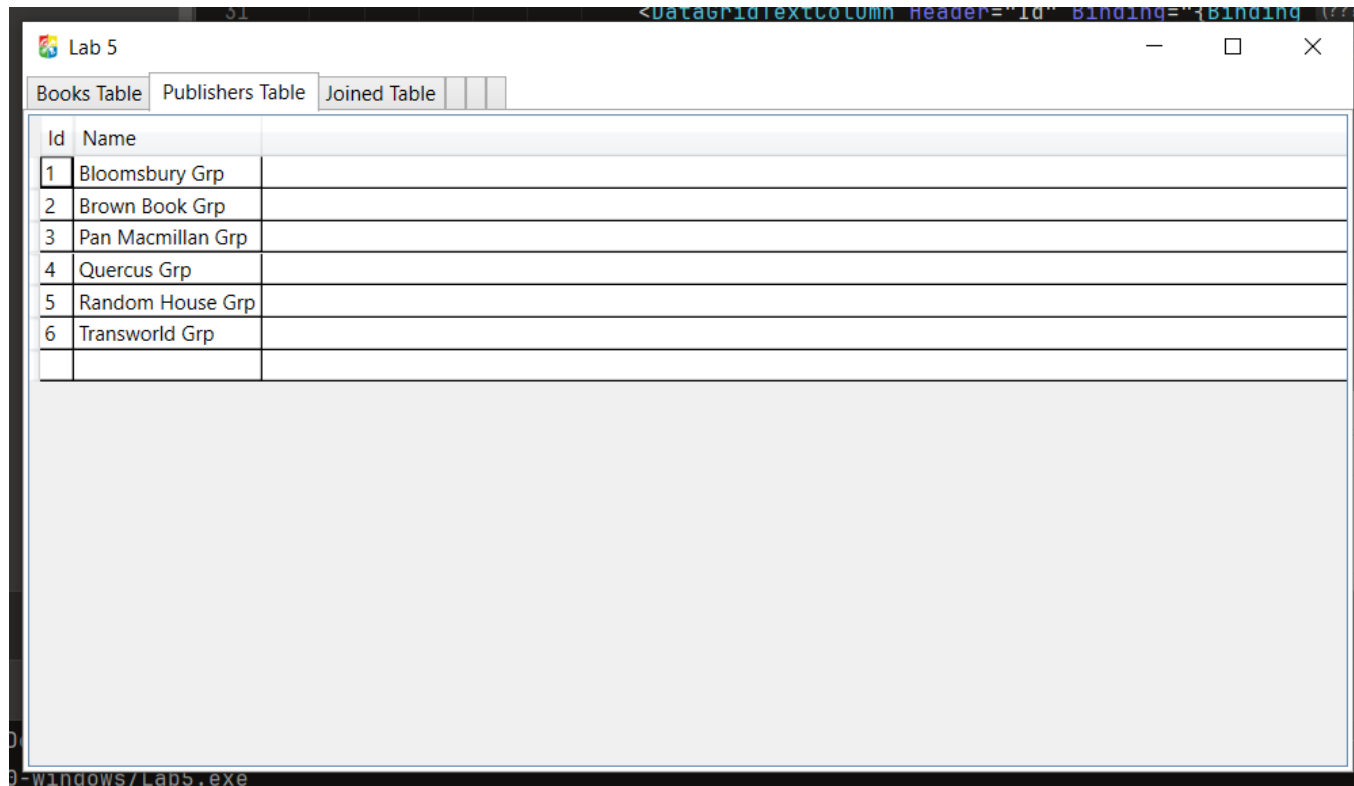
```
DROP TABLE IF EXISTS #BulkData
GO;
```

Завдання 2: Створення моделі
Скріншоти:

Завдання 3: Створення запитів до Entity Framework
Об'єднана таблиця:

Пошук книжок за всіма полями:

Пошук книжок в об'єднаній таблиці за видавництвом:

Висновок: за результатами виконання цієї лабораторної роботи були набуті практичні навички роботи з Entity Framework.

Код програми.

MainWindow.xaml:

```xml
<Window x:Class="Lab5.Views.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Lab5.Views"
        mc:Ignorable="d"
        WindowStartupLocation="CenterScreen"
        Title="Lab 5" MinHeight="450" Height="450" MinWidth="800" Width="800">
    <Grid>
        <TabControl>
            <TabItem Header="Books Table">
                <DataGrid x:Name="BooksGrid"
                          AutoGenerateColumns="False"
                          ItemsSource="{Binding}">
                    <DataGrid.Columns>
                        <DataGridTextColumn Header="ISBN" Binding="{Binding Isbn}" />
                        <DataGridTextColumn Header="Title" Binding="{Binding Title}" />
                        <DataGridTextColumn Header="Authors" Binding="{Binding Authors}" />
                        <DataGridTextColumn Header="Publisher Code" Binding="{Binding PublisherCode}"
/>
                        <DataGridTextColumn    Header="Publication    Year"    Binding="{Binding
PublicationYear}" />
                    </DataGrid.Columns>
                </DataGrid>
            </TabItem>

            <TabItem Header="Publishers Table">
                <DataGrid x:Name="PublishersGrid"
                          AutoGenerateColumns="False"
                          ItemsSource="{Binding}">
                    <DataGrid.Columns>
                        <DataGridTextColumn Header="Id" Binding="{Binding Id}" />
                        <DataGridTextColumn Header="Name" Binding="{Binding Name}" />
                    </DataGrid.Columns>
                </DataGrid>
            </TabItem>

            <TabItem Header="Joined Table">
                <DataGrid x:Name="JoinedGrid"
                          AutoGenerateColumns="True"
                          ItemsSource="{Binding}">
                </DataGrid>
            </TabItem>

            <TabItem Header="Search | Books">
                <Grid>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="Auto" />
                        <ColumnDefinition />
                    </Grid.ColumnDefinitions>
                    <Grid.RowDefinitions>
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="*" />
                    </Grid.RowDefinitions>

                    <TextBlock Grid.Row="0" Grid.Column="0"  Margin="5">ISBN:</TextBlock>
                    <TextBox x:Name="IsbnBox"
                             TextChanged="SearchBooksTextChanged"
                             Grid.Row="0" Grid.Column="1"
                             Focusable="True" />

                    <TextBlock Grid.Row="1" Grid.Column="0" Margin="5">Title:</TextBlock>
                    <TextBox x:Name="TitleBox"
                             TextChanged="SearchBooksTextChanged"
                             Grid.Row="1" Grid.Column="1"
                             Focusable="True" />
```

```xml
                        <TextBlock Grid.Row="2" Grid.Column="0" Margin="5">Authors:</TextBlock>
                        <TextBox x:Name="AuthorsBox"
                                 TextChanged="SearchBooksTextChanged"
                                 Grid.Row="2" Grid.Column="1"
                                 Focusable="True" />

                        <TextBlock Grid.Row="3" Grid.Column="0" Margin="5">Publisher:</TextBlock>
                        <TextBox x:Name="PublisherCodeBox"
                                 TextChanged="SearchBooksTextChanged"
                                 PreviewTextInput="NumberValidationTextBox"
                                 Grid.Row="3" Grid.Column="1"
                                 Focusable="True" />

                        <TextBlock Grid.Row="4" Grid.Column="0" Margin="5">Publication Year:</TextBlock>
                        <TextBox x:Name="PublicationYearBox"
                                 TextChanged="SearchBooksTextChanged"
                                 PreviewTextInput="NumberValidationTextBox"
                                 Grid.Row="4" Grid.Column="1"
                                 Focusable="True" />

                        <DataGrid Grid.Row="5" Grid.Column="0" Grid.ColumnSpan="2"
                                  x:Name="SearchBooksGrid"
                                  AutoGenerateColumns="False"
                                  ItemsSource="{Binding}">
                            <DataGrid.Columns>
                                <DataGridTextColumn Header="ISBN" Binding="{Binding Isbn}" />
                                <DataGridTextColumn Header="Title" Binding="{Binding Title}" />
                                <DataGridTextColumn Header="Authors" Binding="{Binding Authors}" />
                                <DataGridTextColumn    Header="Publisher    Code"    Binding="{Binding
PublisherCode}" />
                                <DataGridTextColumn    Header="Publication    Year"    Binding="{Binding
PublicationYear}" />
                            </DataGrid.Columns>
                        </DataGrid>
                    </Grid>
                </TabItem>

                <TabItem Header="Search | By Publisher">
                    <Grid>
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="Auto" />
                            <ColumnDefinition />
                        </Grid.ColumnDefinitions>
                        <Grid.RowDefinitions>
                            <RowDefinition Height="Auto" />
                            <RowDefinition Height="*" />
                        </Grid.RowDefinitions>

                        <TextBlock Grid.Row="0" Grid.Column="0" Margin="5">Publisher:</TextBlock>
                        <TextBox x:Name="PublisherSearchBox"
                                 TextChanged="SearchByPublisher"
                                 Grid.Row="0" Grid.Column="1"
                                 Focusable="True" />

                        <DataGrid Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2"
                                  x:Name="SearchByPublisherGrid"
                                  AutoGenerateColumns="True"
                                  ItemsSource="{Binding}">
                        </DataGrid>
                    </Grid>
                </TabItem>
            </TabControl>
        </Grid>
</Window>
```

MainWindow.xaml.cs:

```csharp
using System.Text.RegularExpressions;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using Lab5.Context;
using Lab5.Entities;
using LinqKit;
using Microsoft.IdentityModel.Tokens;

namespace Lab5.Views;

/// <summary>
/// Interaction logic for MainWindow.xaml
```

```csharp
        /// </summary>
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();

        using var dbContext = new LibraryDbContext();
        BooksGrid.DataContext = dbContext.Books.ToList();
        PublishersGrid.DataContext = dbContext.Publishers.ToList();

        InitializeJoinedTable(dbContext);
    }

    private void InitializeJoinedTable(LibraryDbContext dbContext)
    {
        var joined = dbContext.Books.Join(dbContext.Publishers,
            x => x.PublisherCode,
            y => y.Id,
            (x, y) => new
            {
                x.Isbn,
                x.Title,
                x.Authors,
                Publisher = y.Name,
                x.PublicationYear
            });

        JoinedGrid.DataContext = joined.ToList();
    }

    private void SearchBooksTextChanged(object sender, TextChangedEventArgs e)
    {
        using var dbContext = new LibraryDbContext();

        var isbn = IsbnBox.Text.Trim();
        var title = TitleBox.Text.Trim();
        var authors = AuthorsBox.Text.Trim();
        var code = Convert.ToDecimal(ZeroIfEmpty(PublisherCodeBox.Text.Trim()));
        var year = Convert.ToDecimal(ZeroIfEmpty(PublicationYearBox.Text.Trim()));

        var predicate = PredicateBuilder.New<Book>();

        if (!isbn.IsNullOrEmpty()) predicate.And(b => b.Isbn.Contains(isbn));
        if (!title.IsNullOrEmpty()) predicate.And(b => b.Title.Contains(title));
        if (!authors.IsNullOrEmpty()) predicate.And(b => b.Authors.Contains(authors));
        if (code != 0) predicate.And(b => b.PublisherCode == code);
        if (year != 0) predicate.And(b => b.PublicationYear == year);

        SearchBooksGrid.DataContext = dbContext.Books.Where(predicate).ToList();
    }

    private void SearchByPublisher(object sender, TextChangedEventArgs e)
    {
        using var dbContext = new LibraryDbContext();

        var publisher = PublisherSearchBox.Text.Trim();

        var joined = dbContext.Books.Join(dbContext.Publishers,
            x => x.PublisherCode,
            y => y.Id,
            (x, y) => new
            {
                x.Isbn,
                x.Title,
                x.Authors,
                Publisher = y.Name,
                x.PublicationYear
            });

        SearchByPublisherGrid.DataContext = joined
            .Where(b => b.Publisher.Contains(publisher))
            .ToList();
    }

    private static string ZeroIfEmpty(string s)
    {
        return string.IsNullOrEmpty(s) ? "0" : s;
    }

    private void NumberValidationTextBox(object sender, TextCompositionEventArgs e)
```

```
        {
            var regex = new Regex("[^0-9]+");
            e.Handled = regex.IsMatch(e.Text);
        }
}
```

CreateTable.sql:

```sql
CREATE DATABASE DPGI
GO;

USE DPGI
GO;

CREATE TABLE dbo.Publishers
(
    id   INT IDENTITY (1, 1),
    name VARCHAR(255) NOT NULL,

    PRIMARY KEY (id)
)
GO;


CREATE TABLE dbo.Books
(
    isbn             VARCHAR(30)  NOT NULL,
    title            VARCHAR(255) NOT NULL,
    authors          VARCHAR(255) NOT NULL,
    publisher_code   INT          NOT NULL,
    publication_year SMALLINT,

    PRIMARY KEY (isbn),

    CONSTRAINT FK_PublishersCode
        FOREIGN KEY (publisher_code) REFERENCES dbo.Publishers (id)
)
GO;
```

InsertData.sql:

```sql
USE DPGI
GO;

DROP TABLE IF EXISTS #BulkData
GO;

CREATE TABLE #BulkData
(
    isbn             VARCHAR(30),
    title            VARCHAR(255),
    authors          VARCHAR(255),
    publisher        VARCHAR(255),
    publication_year SMALLINT
)
GO;

BULK INSERT #BulkData
    FROM 'D:\Programming\KPI-DPGI\Lab5\Database\Data\Data.csv'
    WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2
    );
INSERT INTO dbo.Publishers (name)
SELECT DISTINCT publisher
FROM #BulkData
GO;

INSERT INTO dbo.Books (isbn, title, authors, publication_year, publisher_code)
SELECT bd.isbn,
       bd.title,
       bd.authors,
       bd.publication_year,
       pub.id AS publisher_code
FROM #BulkData AS bd
INNER JOIN dbo.Publishers AS pub
ON bd.publisher = pub.name;

DROP TABLE IF EXISTS #BulkData
GO;
```