**УСТАНОВКА ЗАВИСИМОСТЕЙ И СОЗДАНИЕ ПРОЕКТА**

dotnet new install Avalonia.Templates

dotnet new avalonia.mvvm -o DemoApp

cd DemoApp

dotnet add package Microsoft.EntityFrameworkCore.Tools --version 9.0.3

dotnet add package Microsoft.EntityFrameworkCore.Design --version 9.0.3

dotnet add package FirebirdSql.EntityFrameworkCore.Firebird --version 12.0.0

dotnet add package Avalonia.ReactiveUI --version 11.2.6

dotnet new tool-manifest

dotnet tool install dotnet-ef --version 9.0.3

**ГЕНЕРАЦИЯ КЛАССОВ**

dotnet tool restore

dotnet tool run dotnet-ef dbcontext scaffold "DataSource=localhost;Port=3050;Database=/db/demo.fdb;Username=sysdba;Password=password" FirebirdSql.EntityFrameworkCore.Firebird -o Entities -f

**НАПИСАНИЕ ПРОГРАММЫ**

**Partner.cs:**

using System.Linq;

using System.Collections.Generic;

namespace DemoApp.Models

{

public class Partner : Entities.Partner

{

public Entities.Partner \_partner { get; set; }

public Partner(Entities.Partner partner)

{

\_partner = partner;

}

public string? PartnerTypeDescription => \_partner.PartnerTypeNavigation?.PartnerTypeName;

public string ContactInfo => $"{\_partner.DirectorPhone} | {\_partner.DirectorMail}";

public string RatingDisplay => $"Рейтинг: {\_partner.PartnerRating ?? 0}";

public int TotalProductsSold => \_partner.Sales?.Sum(s => s.ProductCount) ?? 0;

public decimal Discount => CalculatePartnerDiscount(TotalProductsSold);

public string DiscountDisplay => $"Скидка: {Discount:P0}";

public static decimal CalculatePartnerDiscount(int totalProductsSold)

{

return totalProductsSold switch

{

< 10000 => 0,

< 50000 => 0.05m,

< 300000 => 0.10m,

\_ => 0.15m

};

}

}

}

**MainWindowViewModel:**

using Avalonia.Controls;

using CommunityToolkit.Mvvm.ComponentModel;

using DemoApp.Entities;

using DemoApp.Views;

using Microsoft.EntityFrameworkCore;

namespace DemoApp.ViewModels

{

public partial class MainWindowViewModel : ViewModelBase

{

private readonly FdbDemoFdbContext \_context;

[ObservableProperty]

private Control? content;

public void ShowContent(Control content) => Content = content;

public void ShowPartnersList()

{

Content = new PartnersView

{

DataContext = new PartnersViewModel(\_context, this)

};

}

public MainWindowViewModel()

{

\_context = new FdbDemoFdbContext();

ShowPartnersList();

}

}

}

**PartnersViewModel:**

using System.Collections.ObjectModel;

using System.Linq;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DemoApp.Entities;

using Microsoft.EntityFrameworkCore;

namespace DemoApp.ViewModels

{

public partial class PartnersViewModel : ObservableObject

{

private readonly FdbDemoFdbContext \_context;

internal readonly MainWindowViewModel \_mainViewModel;

[ObservableProperty]

private ObservableCollection<DemoApp.Models.Partner> \_partners = new();

[ObservableProperty]

private DemoApp.Models.Partner? \_selectedPartner;

public PartnersViewModel(FdbDemoFdbContext context, MainWindowViewModel mainViewModel)

{

\_context = context;

\_mainViewModel = mainViewModel;

LoadPartners();

}

public void LoadPartners()

{

var entities = \_context.Partners

.Include(p => p.Sales)

.Include(p => p.PartnerTypeNavigation)

.ToList();

Partners = new ObservableCollection<DemoApp.Models.Partner>(

entities.Select(e => new DemoApp.Models.Partner(e))

);

}

[RelayCommand]

private void EditPartner()

{

// изменение партнера

}

[RelayCommand]

private void AddPartner()

{

// добавление партнера

}

[RelayCommand]

private void ViewSalesHistory(DemoApp.Models.Partner? partner)

{

// история продаж

}

}

}

**MainWindow.axaml:**

<Window xmlns="https://github.com/avaloniaui"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:vm="using:DemoApp.ViewModels"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d" d:DesignWidth="800" d:DesignHeight="450"

x:Class="DemoApp.Views.MainWindow"

x:DataType="vm:MainWindowViewModel"

Icon="/Assets/avalonia-logo.ico"

Title="DemoApp">

<Design.DataContext>

<!-- This only sets the DataContext for the previewer in an IDE,

to set the actual DataContext for runtime, set the DataContext property in code (look at App.axaml.cs) -->

<vm:MainWindowViewModel/>

</Design.DataContext>

<ContentControl Content="{Binding Content}"/>

</Window>

**MainWindow.axaml.cs:**

using Avalonia.Controls;

namespace DemoApp.Views;

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

}

**PartnersView.axaml:**

<UserControl xmlns="https://github.com/avaloniaui"

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:vm="using:DemoApp.ViewModels"

mc:Ignorable="d"

d:DesignWidth="800"

d:DesignHeight="450"

x:Class="DemoApp.Views.PartnersView"

x:DataType="vm:PartnersViewModel">

<Grid Margin="20">

<StackPanel Spacing="10">

<TextBlock Text="Список партнеров" FontSize="24" FontWeight="Bold"/>

<StackPanel Orientation="Horizontal" Spacing="10">

<Button Content="Добавить партнера" Command="{Binding AddPartnerCommand}"/>

<Button Content="Редактировать" Command="{Binding EditPartnerCommand}"/>

</StackPanel>

<ListBox ItemsSource="{Binding Partners}" SelectedItem="{Binding SelectedPartner}" MaxHeight="800">

<ListBox.ItemTemplate>

<DataTemplate>

<StackPanel Orientation="Horizontal" Margin="5" HorizontalAlignment="Left">

<StackPanel Orientation="Vertical" Margin="5" HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Left">

<TextBlock Text="{Binding PartnerTypeDescription}" TextWrapping="Wrap"/>

<TextBlock Text=" | " TextWrapping="Wrap"/>

<TextBlock Text="{Binding \_partner.PartnerName}" TextWrapping="Wrap" Width="250"/>

</StackPanel>

<TextBlock Text="{Binding \_partner.Director}" TextWrapping="Wrap" Width="250" HorizontalAlignment="Left"/>

<TextBlock Text="{Binding \_partner.DirectorPhone}" TextWrapping="Wrap" Width="250" HorizontalAlignment="Left"/>

<TextBlock Text="{Binding RatingDisplay}" TextWrapping="Wrap" Width="250" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel Orientation="Vertical" Spacing="10">

<TextBlock Text="{Binding DiscountDisplay}" TextWrapping="Wrap" Width="150"/>

<Button Content="История продаж"/>

</StackPanel>

</StackPanel>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

</StackPanel>

</Grid>

</UserControl>

**PartnersView.axaml.cs:**

using Avalonia.Controls;

using Avalonia.Markup.Xaml;

namespace DemoApp.Views

{

public partial class PartnersView : UserControl

{

public PartnersView()

{

InitializeComponent();

}

private void InitializeComponent()

{

AvaloniaXamlLoader.Load(this);

}

}

}

**ЗАПУСК**

dotnet run