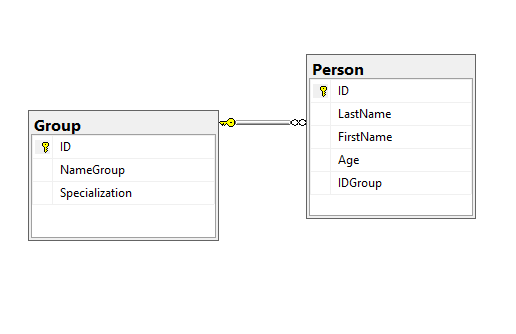
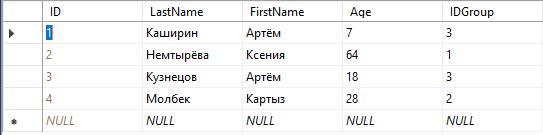
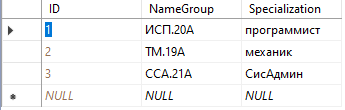
Отчёт.

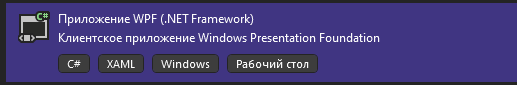
1. Создать БД.



1. Заполняем таблицы.



1. Создаём WPF (.NET Framework)



1. В нём размечаем главный экран, создаём Frame.

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="2\*"/>

<RowDefinition Height="5\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<!--<Image Source="/Resourses/Восьмерка.png"/>-->

<TextBlock Margin="10"

FontSize="20pt"

FontFamily="TimesNewRoman"

VerticalAlignment="Center"

Text="Пример проекта">

</TextBlock>

</StackPanel>

<Frame Grid.Row="1"

Name="frmMain"

NavigationUIVisibility="Hidden">

</Frame>

</Grid>

1. Настраиваем работу интерфейса. Создаём папку класса, в которой создадим класс «ClassFrame» и в нём подключим нашу страницу.



internal class ClassFrame

{

public static Frame frmObj;

}

1. Подгружаем страницу PageUser.



1. На странице PageUser создаём таблицу, даём ей имя и убираем автоматическую генерацию столбцов.

<Grid>

<DataGrid Name="DGridUsers"

AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Width="\*" Header="Фамилия"

Binding="{Binding LastName}"/>

<DataGridTextColumn Width="\*" Header="Имя"

Binding="{Binding FirstName}"/>

<DataGridTextColumn Width="\*" Header="Возраст"

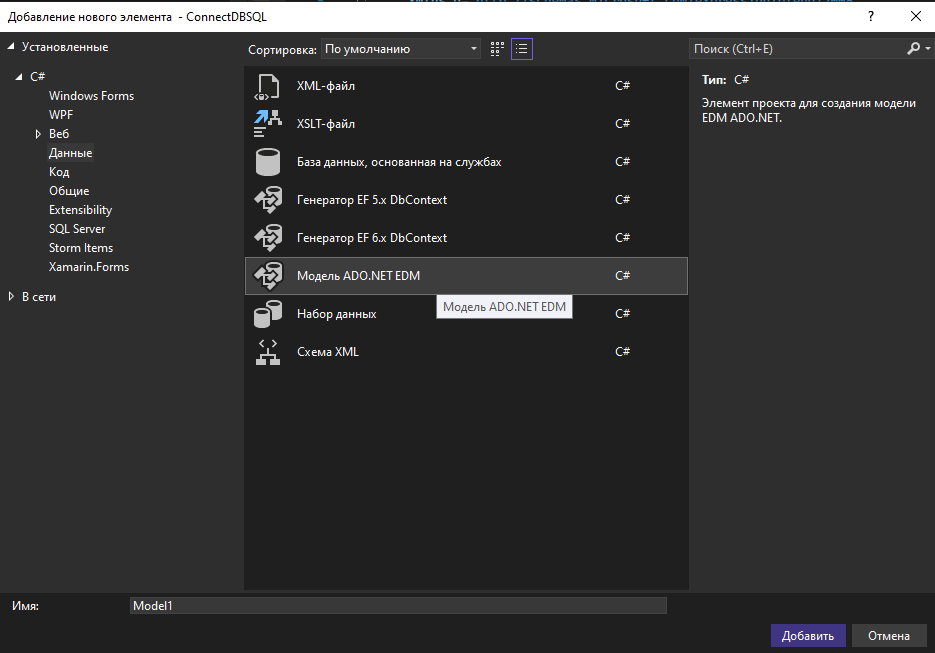
Binding="{Binding Age}"/>

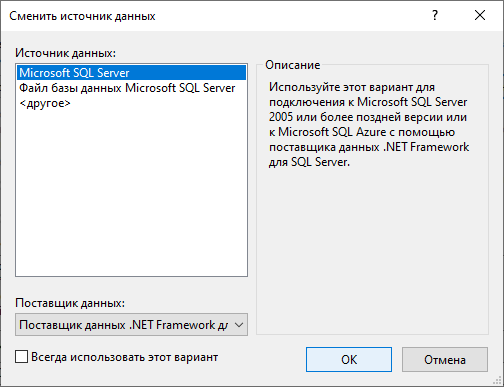
</DataGrid.Columns>

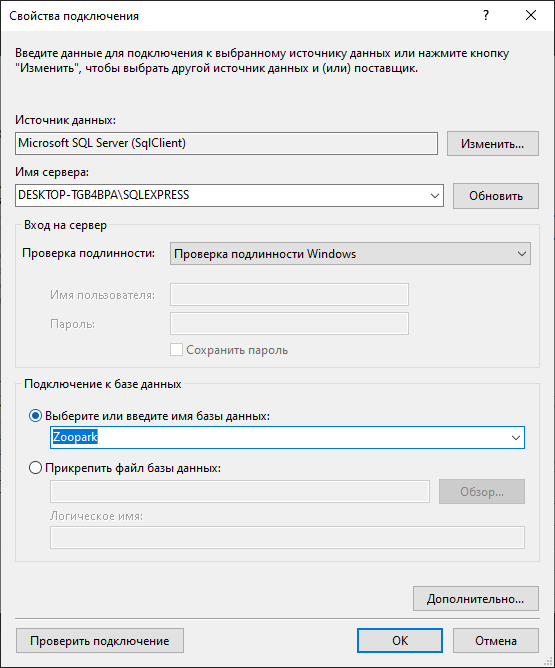
</DataGrid>

</Grid>

1. Класс – «добавить новый элемент» – «данные» – «модель ADO.NET EDM» – задаём имя – выполняем добавление по шагам.







1. В результате появиться модель, которая соответствует нашей бд. В этой модели находим ModelIDB.Context.cs и в неё дописываем.

public partial class GokkeeEntities : DbContext

{

private static GokkeeEntities \_context;

public GokkeeEntities()

: base("name=GokkeeEntities")

{

}

public static GokkeeEntities GetGokkee()

{

if (\_context == null)

\_context = new GokkeeEntities();

return \_context;

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<Group> Group { get; set; }

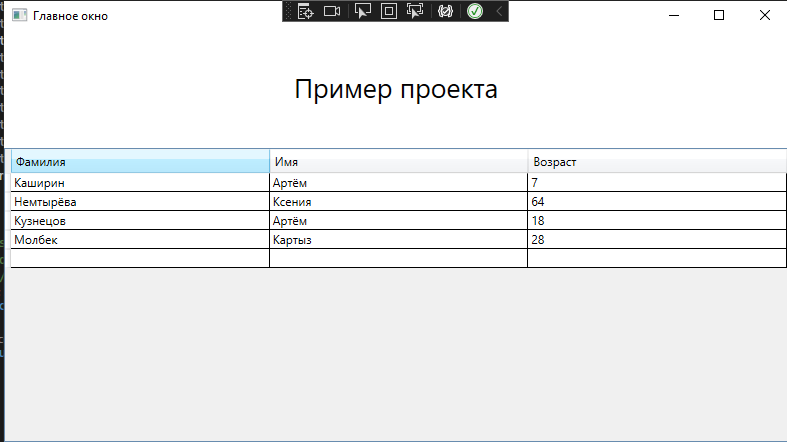
public virtual DbSet<Person> Person { get; set; }

}

1. Далее идём в логику страницы PageUser и привязываем таблицу через метод GetGokkee(). Этот метод передаст данные в нашу таблицу в проекте.

DGridUsers.ItemsSource = GokkeeEntities.GetGokkee().Person.ToList();

1. Готово.



Кнопки добавление удаление

Кнопка редактировать (в плане интефейса)

Событие извизибл

Программные коды всех кнопок

Кнопки.

Кнопки добавить и удалить (Внешний вид)

<Button Name="BtnAdd" Content="Добавить" Click="BtnAdd\_Click"/>

<Button Name="BtnDelete" Content="Удалить" Click="BtnDelete\_Click" />



Логика кнопок

Добавить: В PageUser:

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new AddEditPage(null));

}

Удалить:

private void BtnDelete\_Click(object sender, RoutedEventArgs e)

{

var usersForRemoving = DGridUsers.SelectedItems.Cast<Person>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} пользователей?", "Внимание!",

MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes);

try

{

GokkeeEntities.GetGokkee().Person.RemoveRange(usersForRemoving);

GokkeeEntities.GetGokkee().SaveChanges();

MessageBox.Show("Данные удалены");

}

catch(Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

Кнопка Назад. Создать обработчик

IsVisibleChanged="Page\_IsVisibleChanged">

В PageUser

Прописываем логику:

if(Visibility == Visibility.Visible)

{

GokkeeEntities.GetGokkee().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridUsers.ItemsSource = GokkeeEntities.GetGokkee().Person.ToList();

}

В PageUser создаём шаблон кнопки редактирования

<DataGrid Name="DGridUsers"

AutoGenerateColumns="False"

IsReadOnly="True"

Grid.Row="0">

<DataGrid.Columns>

<DataGridTextColumn Width="\*" Header="Фамилия"

Binding="{Binding LastName}"/>

<DataGridTextColumn Width="\*" Header="Имя"

Binding="{Binding FirstName}"/>

<DataGridTextColumn Width="\*" Header="Возраст"

Binding="{Binding Age}"/>

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit"

Click="BtnEdit\_Click"

Content="Редактирование"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

Прописываем логику

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new AddEditPage((sender as Button).DataContext as Person));

}

Кнопка сохранения: В AddEditPage:

private void BtnSave\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();//Объект ошибка

if (string.IsNullOrWhiteSpace(\_person.FirstName))

error.AppendLine("Укажите имя");

if(string.IsNullOrWhiteSpace(\_person.LastName))

error.AppendLine("Укажите фамилию");

if (string.IsNullOrWhiteSpace(\_person.Age.ToString()))

error.AppendLine("Укажите возраст");

if(error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//Если пользователь новый

if(\_person.ID == 0)

GokkeeEntities.GetGokkee().Person.Add(\_person);//Добавить его

try

{

GokkeeEntities.GetGokkee().SaveChanges(); //Сохранить изменения

MessageBox.Show("Данные сохранены");

}

catch(Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

Блок AddEditPage:

private Person \_person = new Person();//новое поле, которое будет хранить в себе экземпляр пользователя

public AddEditPage(Person selectedPerson)// объект selectedPerson

{

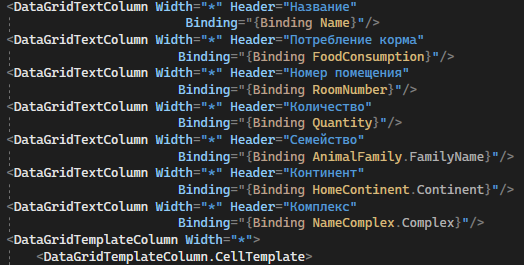
InitializeComponent();

if(selectedPerson != null)

\_person = selectedPerson;

DataContext = \_person; // Созданный контекст

}



<DataGridTextColumn Width="\*" Header="Название"

Binding="{Binding Name}"/>

<DataGridTextColumn Width="\*" Header="Потребление корма"

Binding="{Binding FoodConsumption}"/>

<DataGridTextColumn Width="\*" Header="Номер помещения"

Binding="{Binding RoomNumber}"/>

<DataGridTextColumn Width="\*" Header="Количество"

Binding="{Binding Quantity}"/>

<DataGridTextColumn Width="\*" Header="Семейство"

Binding="{Binding AnimalFamily.FamilyName}"/>

<DataGridTextColumn Width="\*" Header="Континент"

Binding="{Binding HomeContinent.Continent}"/>

<DataGridTextColumn Width="\*" Header="Комплекс"

Binding="{Binding NameComplex.Complex}"/>