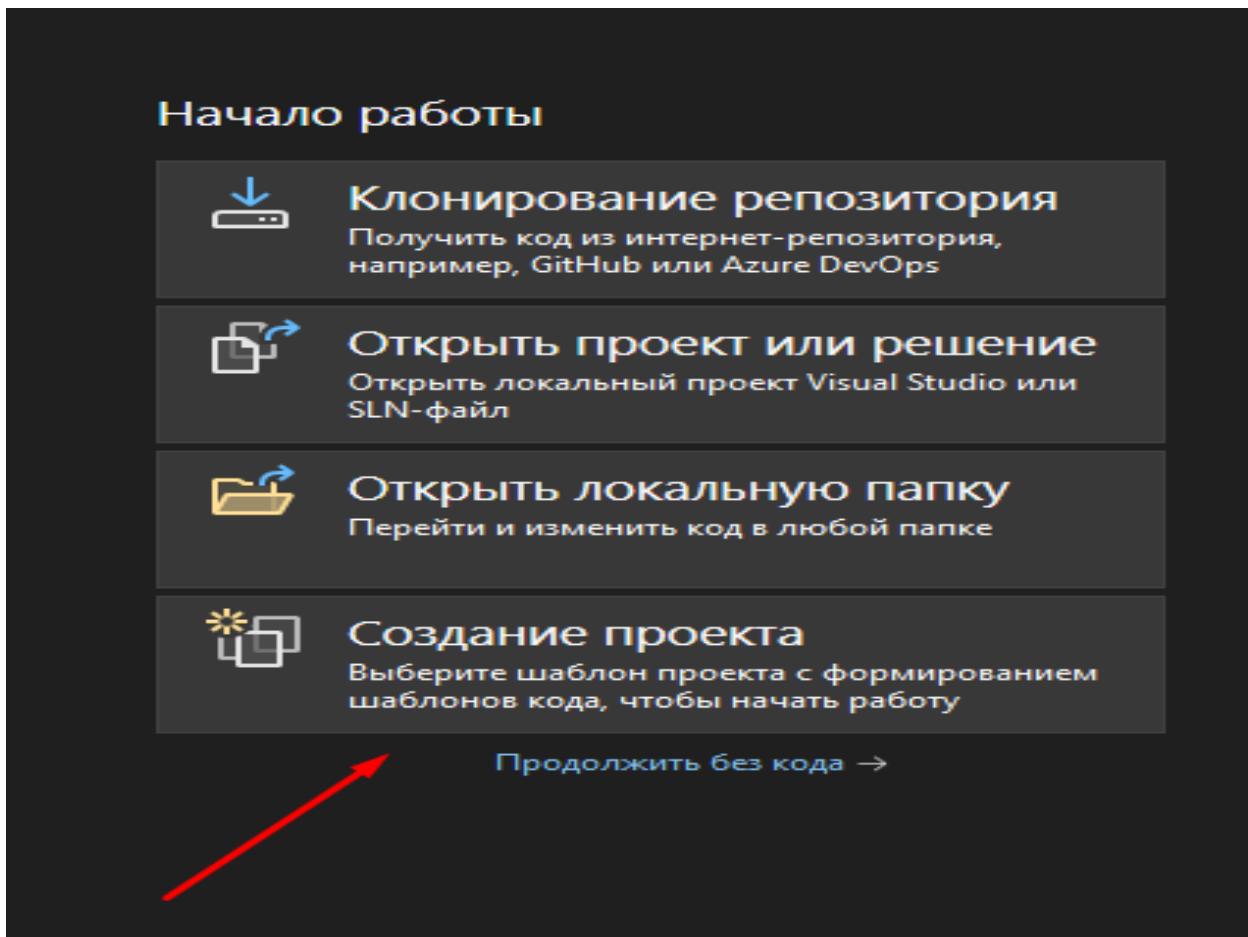
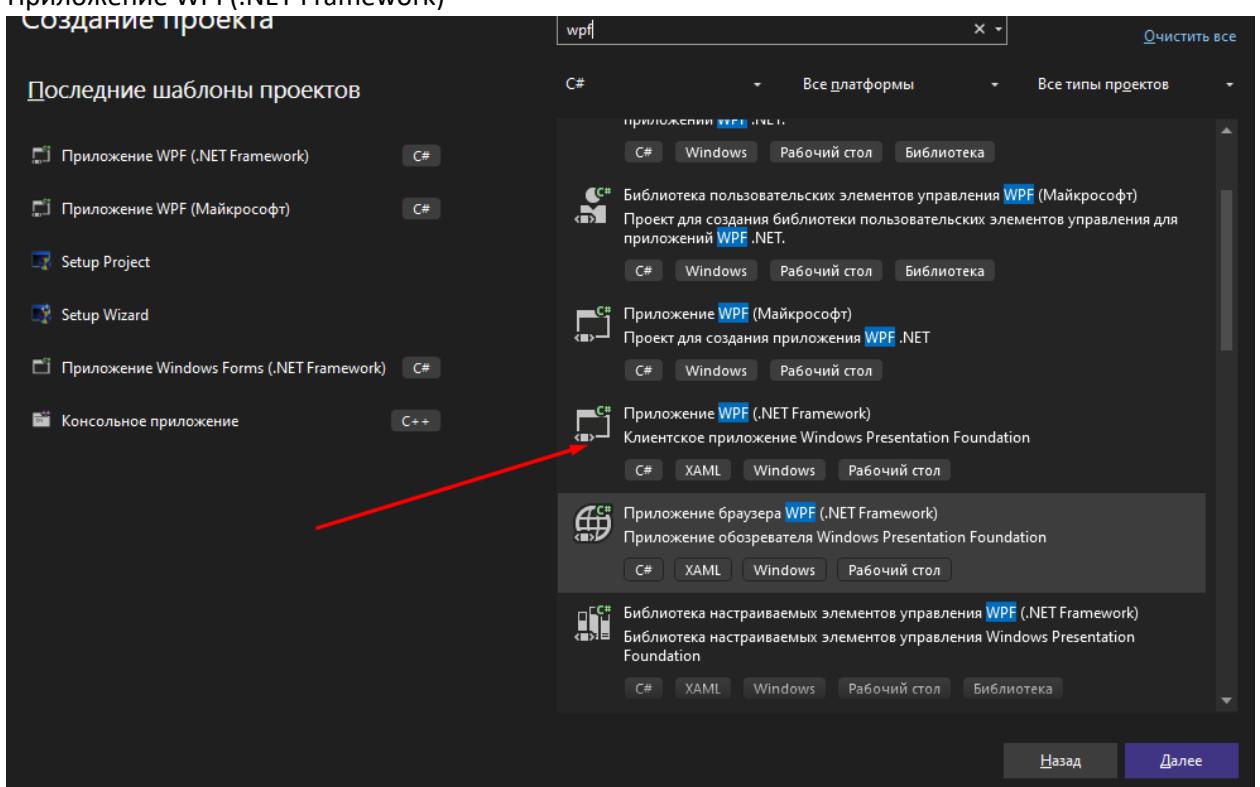


## Разработка wpf приложения

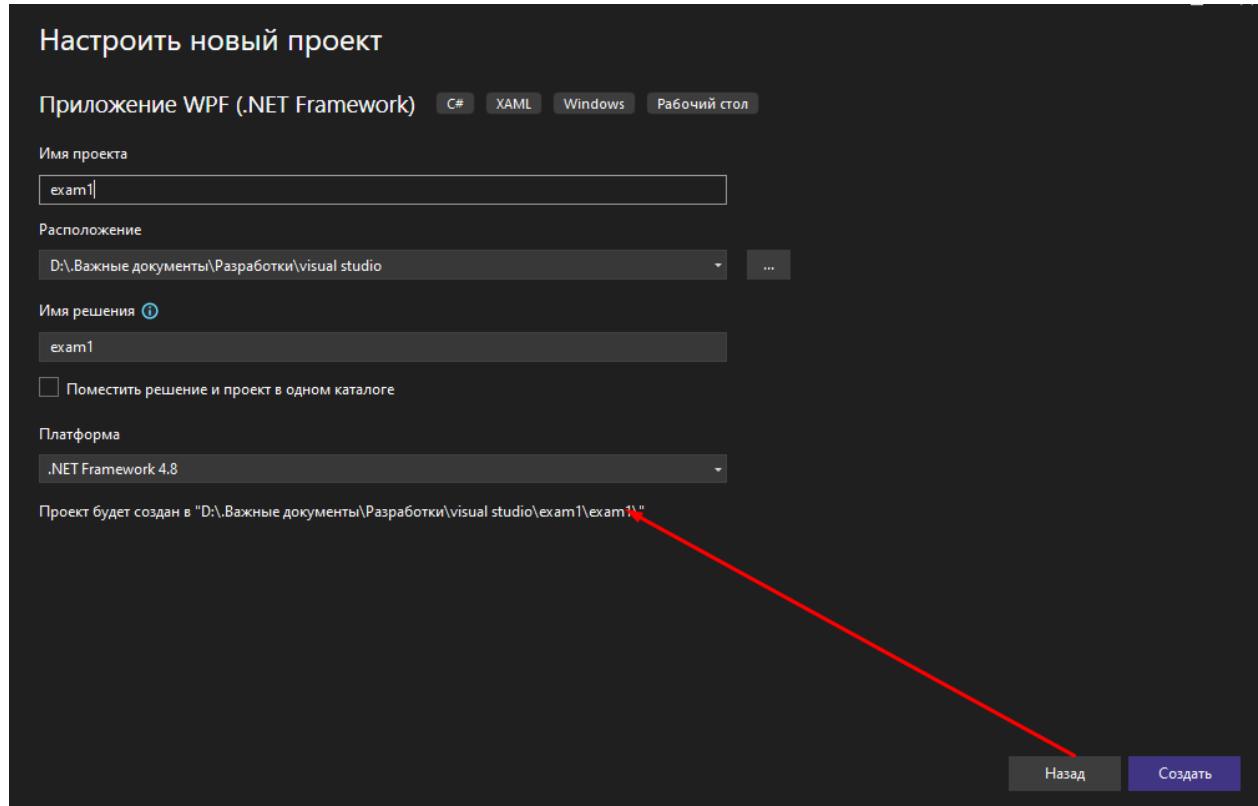
### 1. Создание проекта.



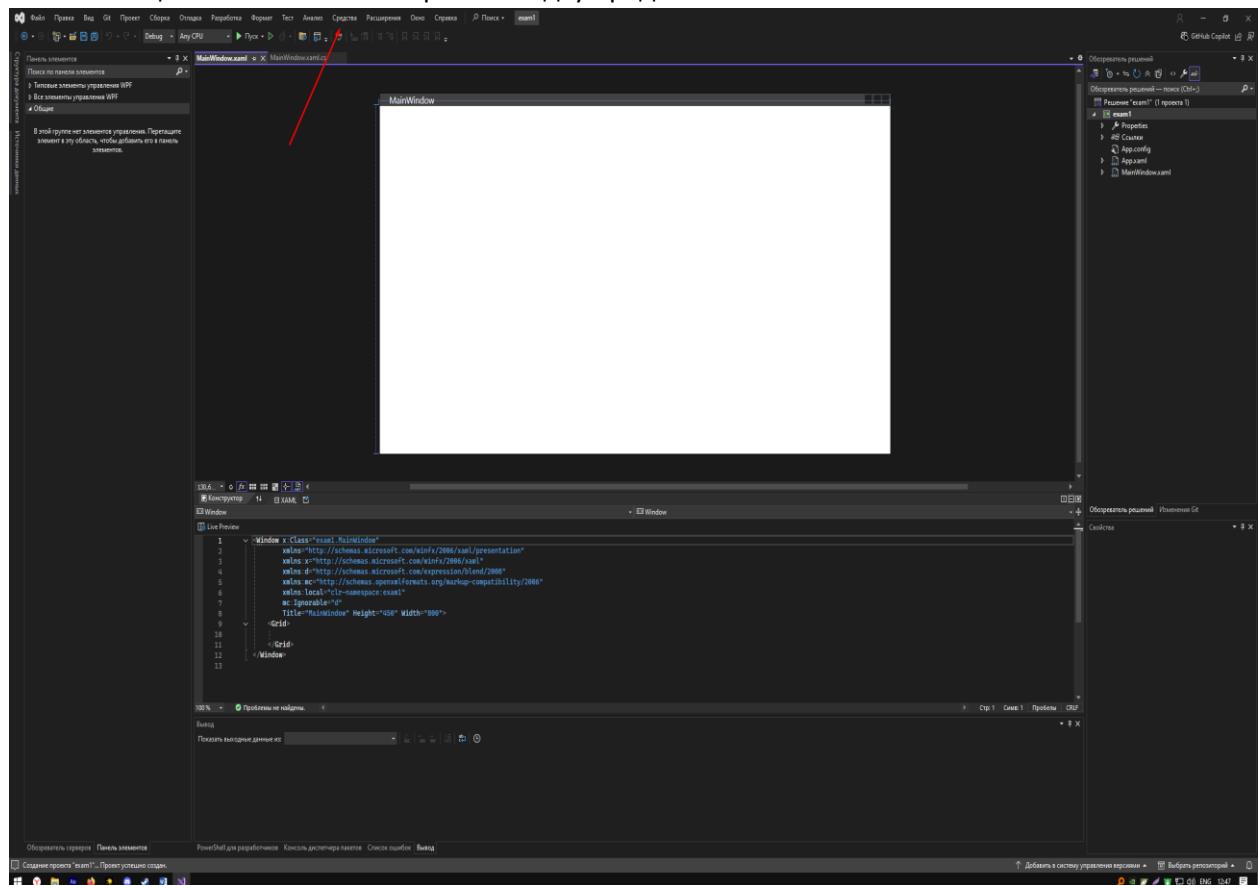
2. Выбор шаблона. При создании приложение впишите в поисковую строку WPF и выберите Приложение WPF(.NET Framework)



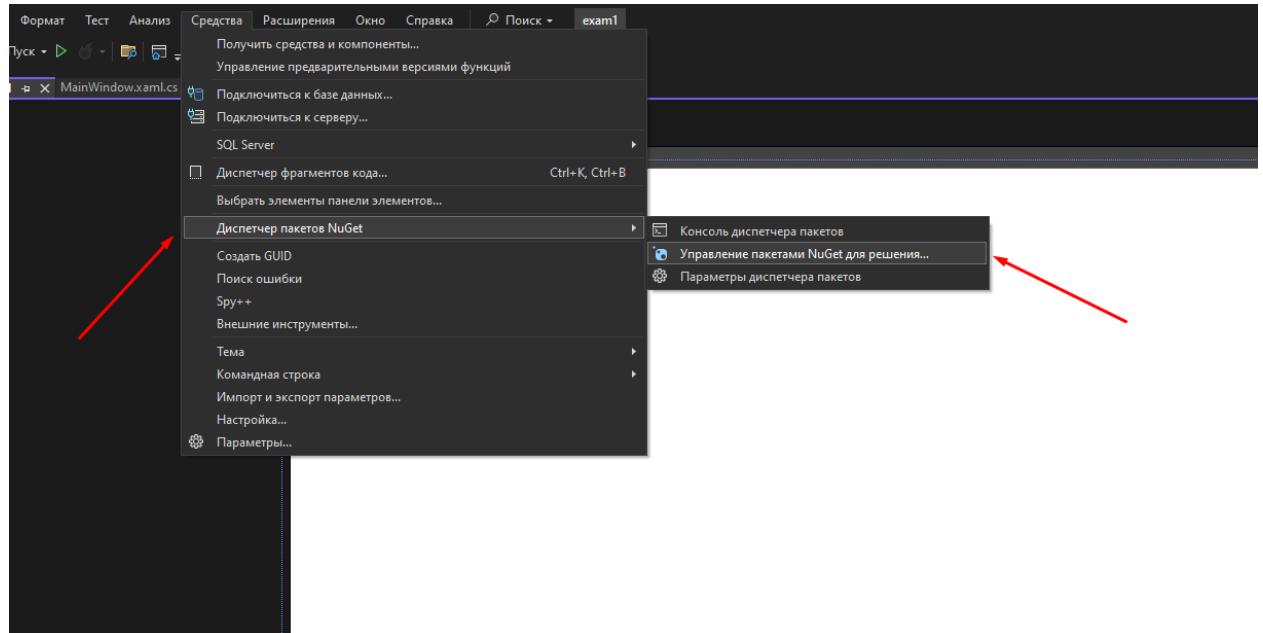
3. Назовите свой проект и ВАЖНО, ВЫБЕРИТЕ ПЛАТФОРМУ .NET Framework 4.8!!!!!!! И нажмите кнопку создать



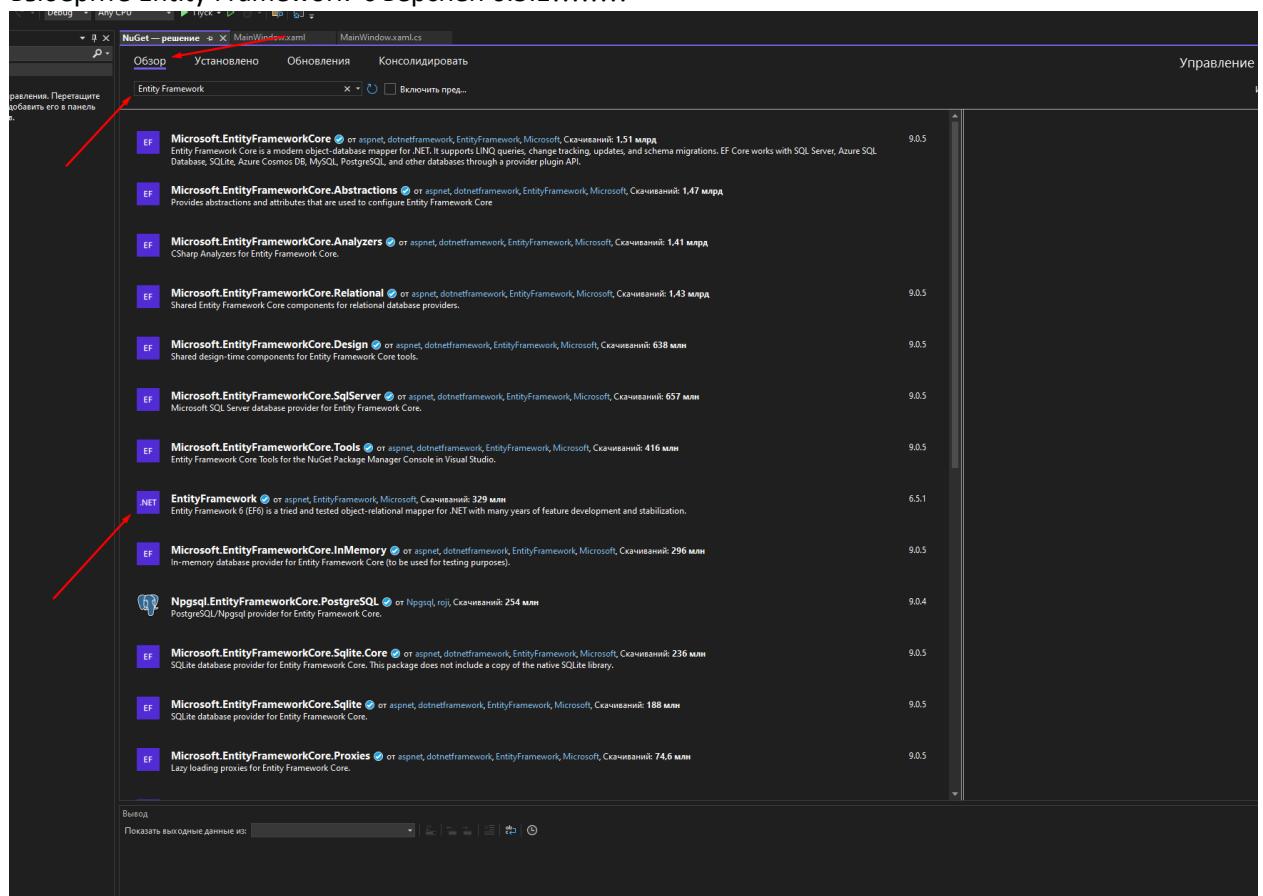
4. В навигационной панели выберите вкладку Средства



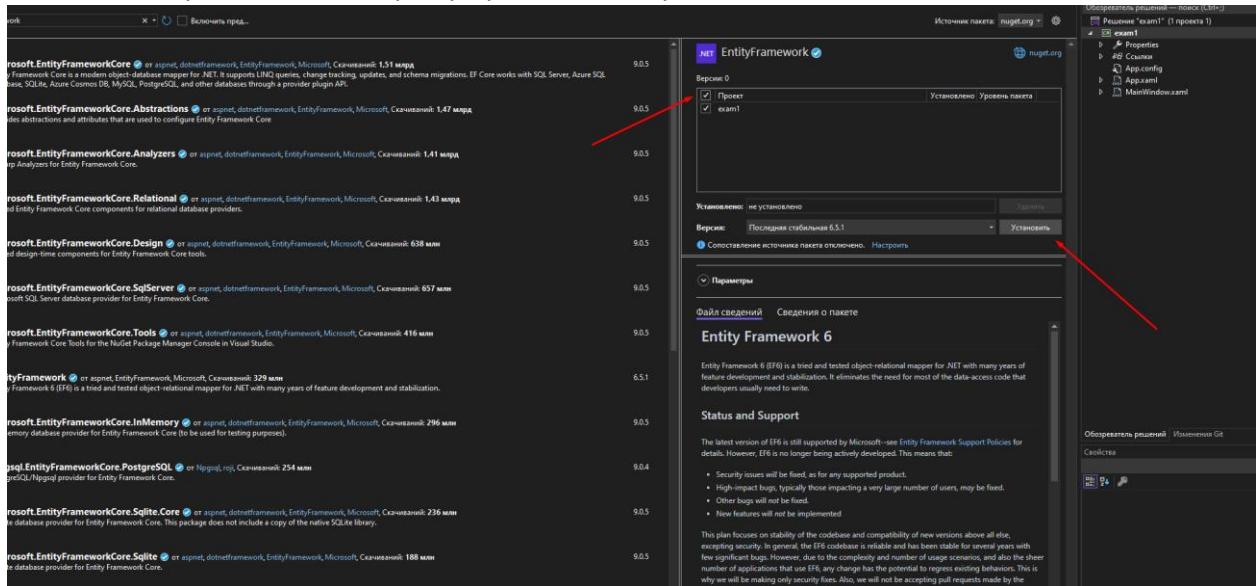
5. Во вкладке Средства выберите пункт Диспетчер пакетов NuGet, а в самом пункте, подпункт Управление пакетами NuGet для решения...



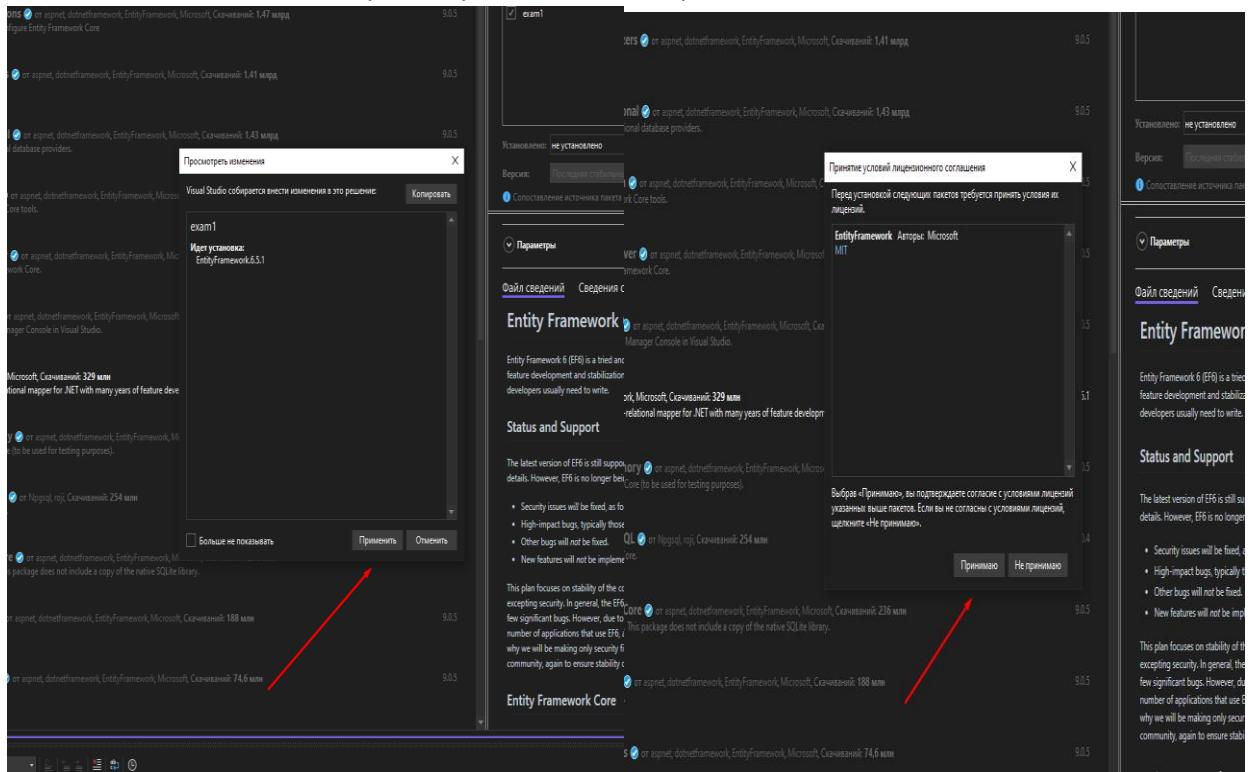
6. В панели NuGet перейдите на вкладку Обзор и в панели поиск впишите Entity Framework, после выберите Entity Framework с версией 6.5.1!!!!!!!



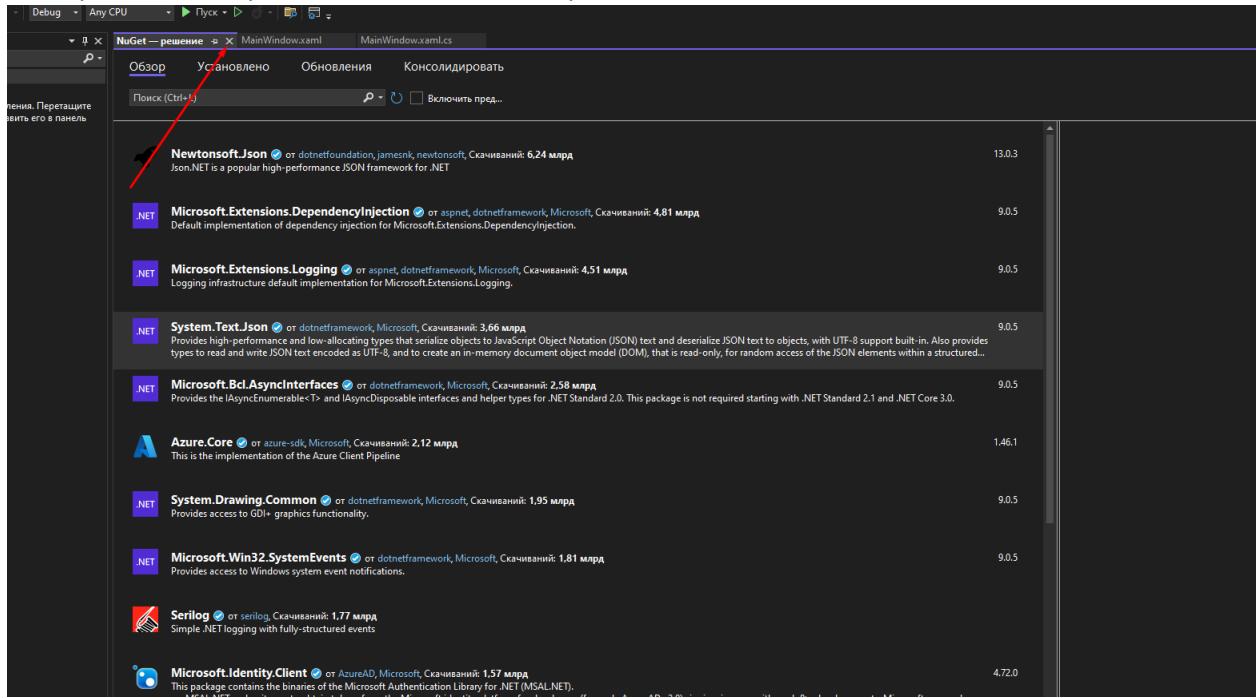
## 7. После выбора нажмите по пункту Проект и кнопку Установить



## 8. Далее появятся окна, где просто нужно нажимать Применить



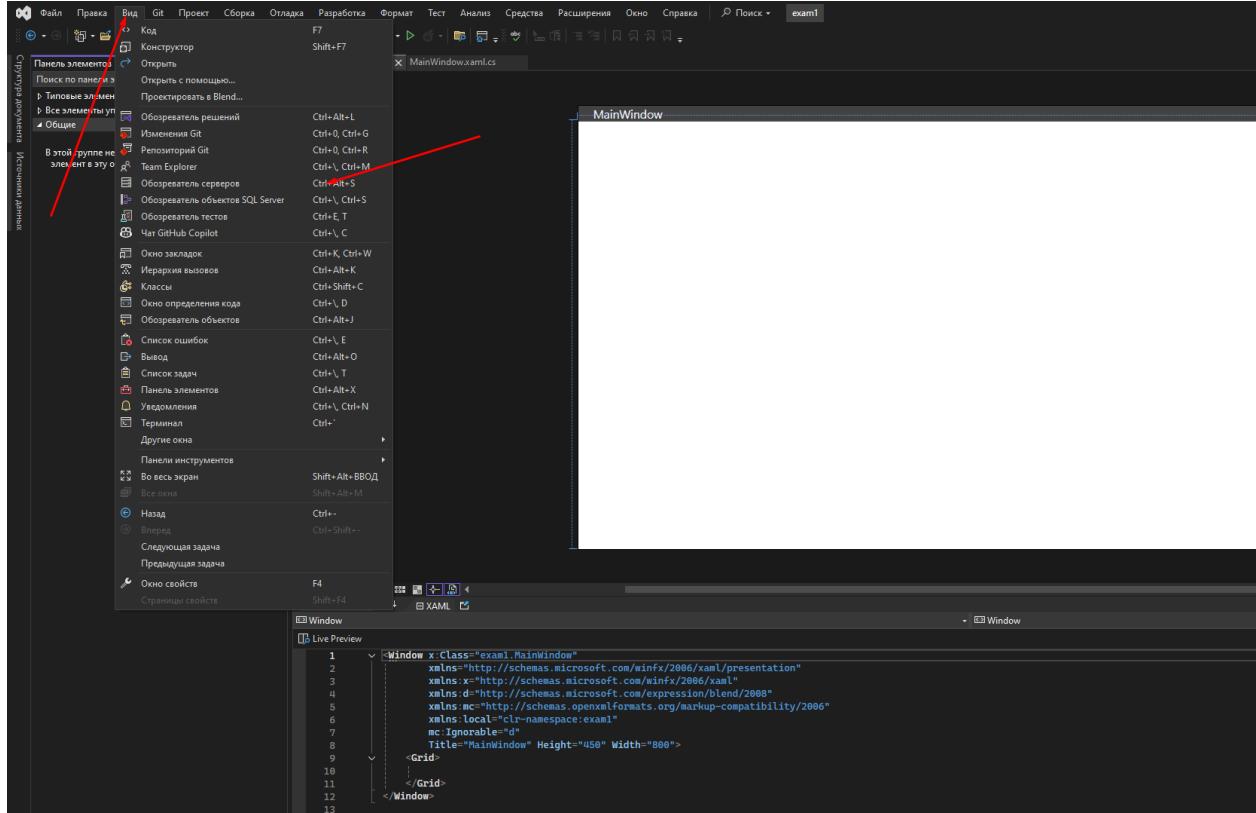
## 9. Закройте вкладку NuGet, она больше не нужна



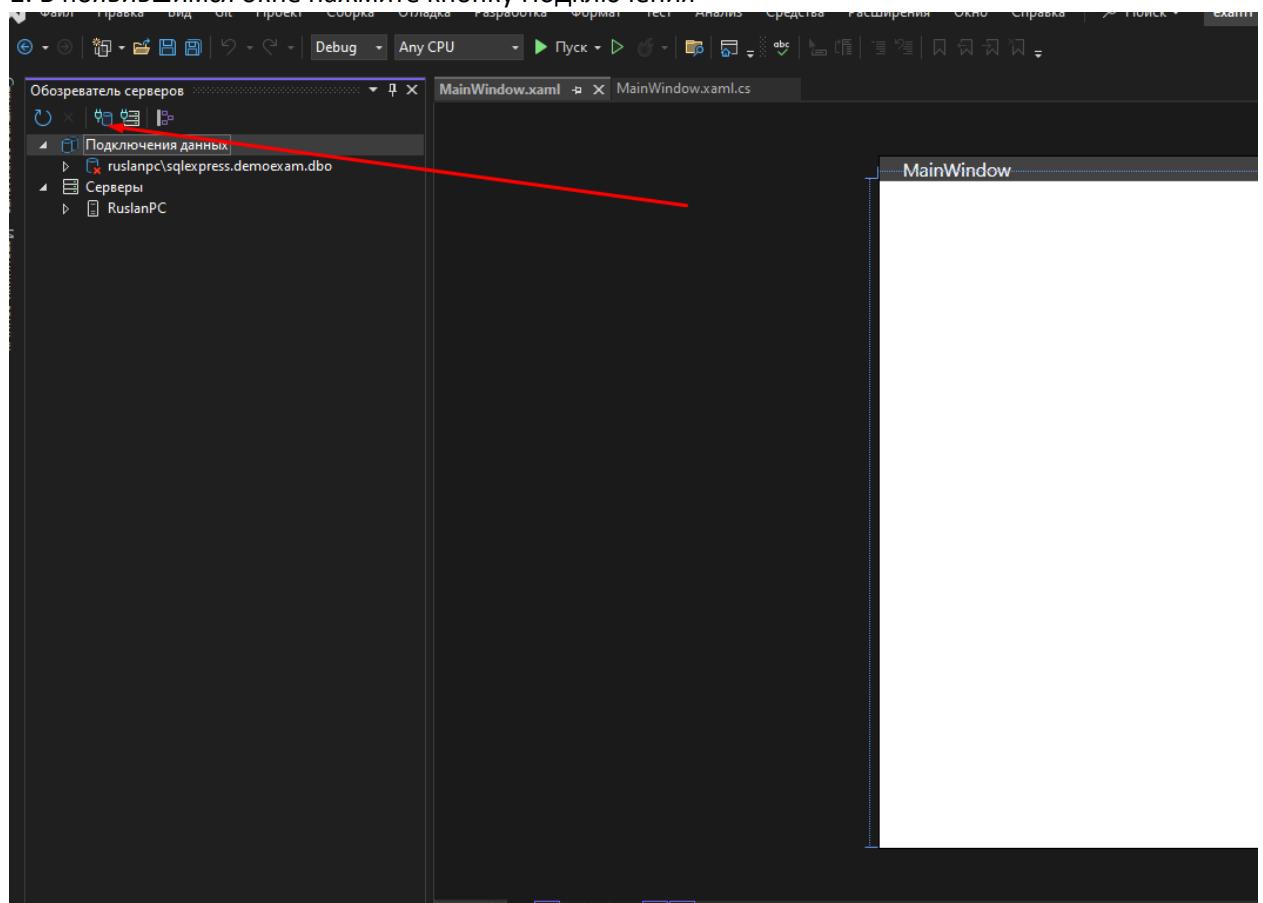
Поздравляю вы установили Entity Framework

## Подключение сервера к приложению WPF

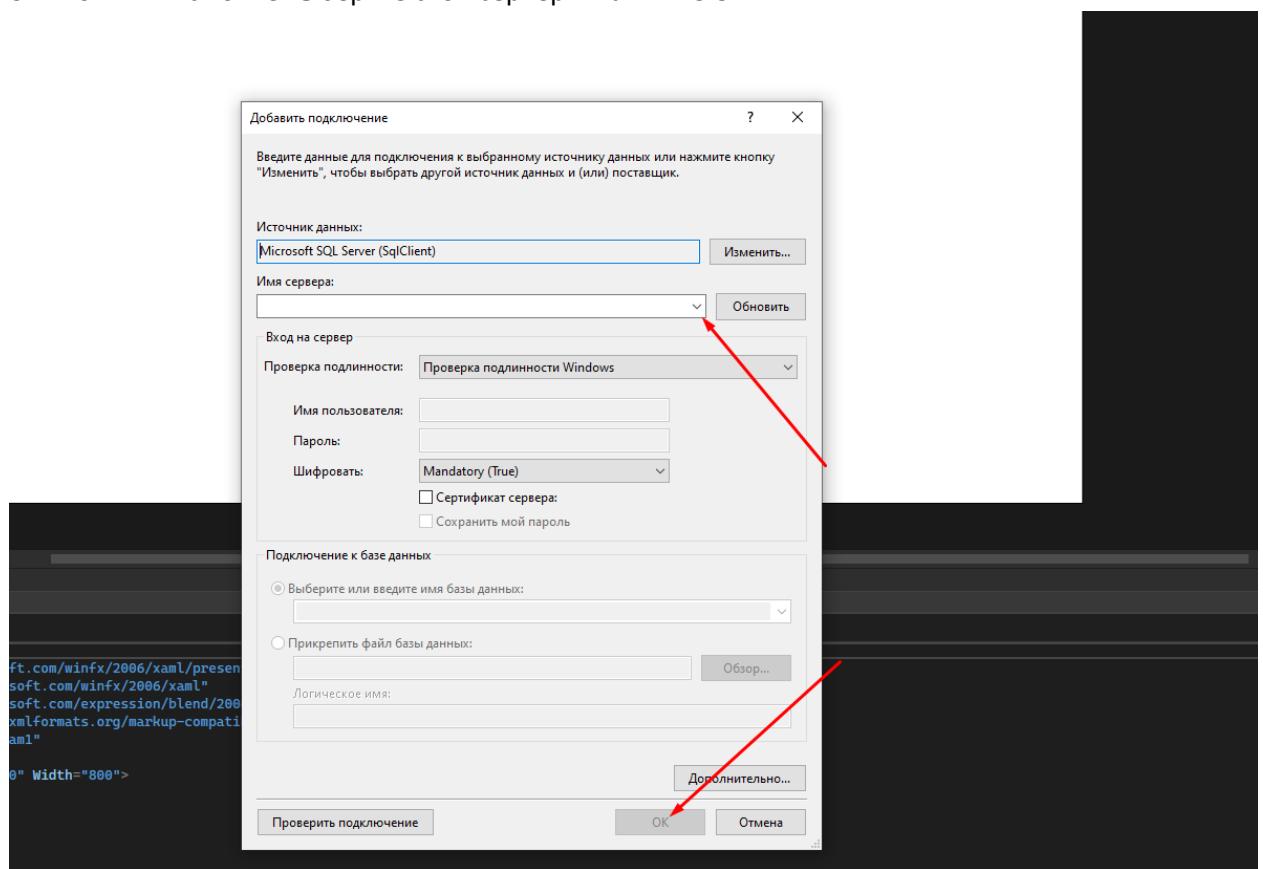
### 1. Нажмите кнопку Вид на навигационной панели и пункт Обозреватель серверов



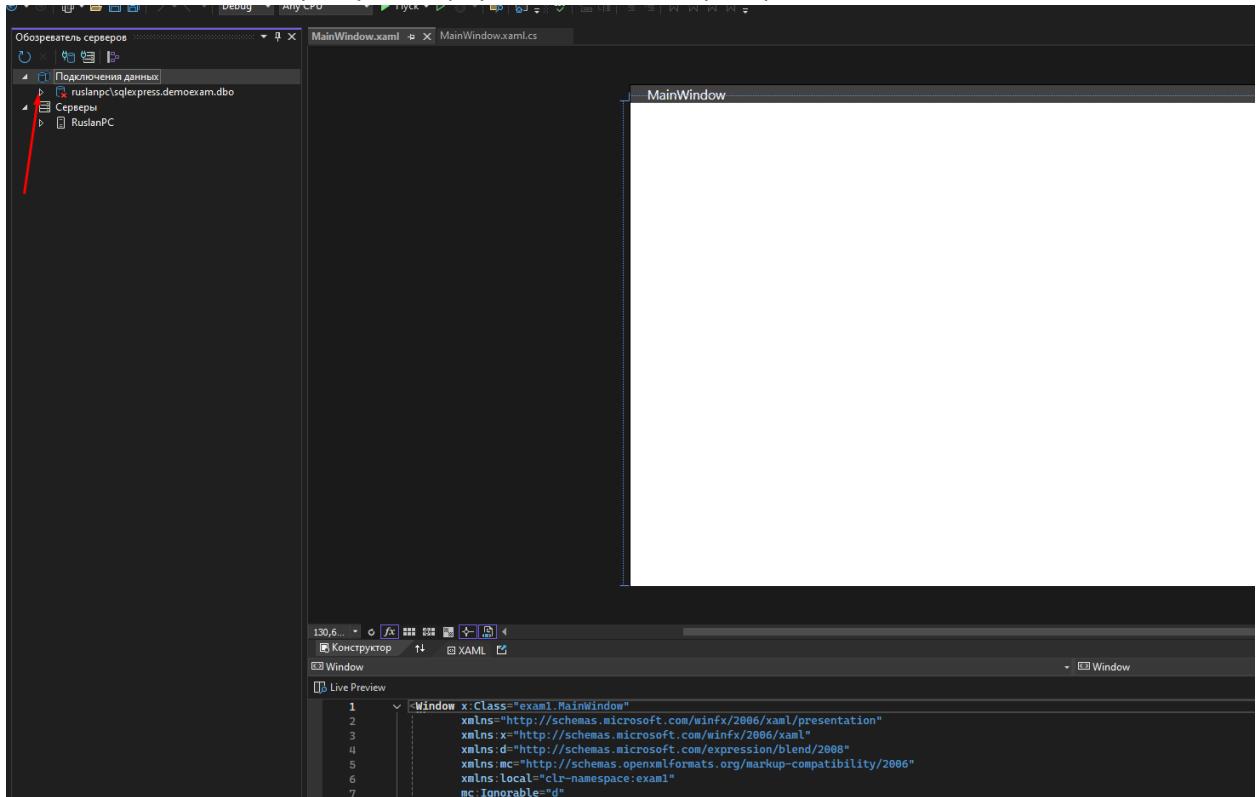
2. В появившимся окне нажмите кнопку Подключения



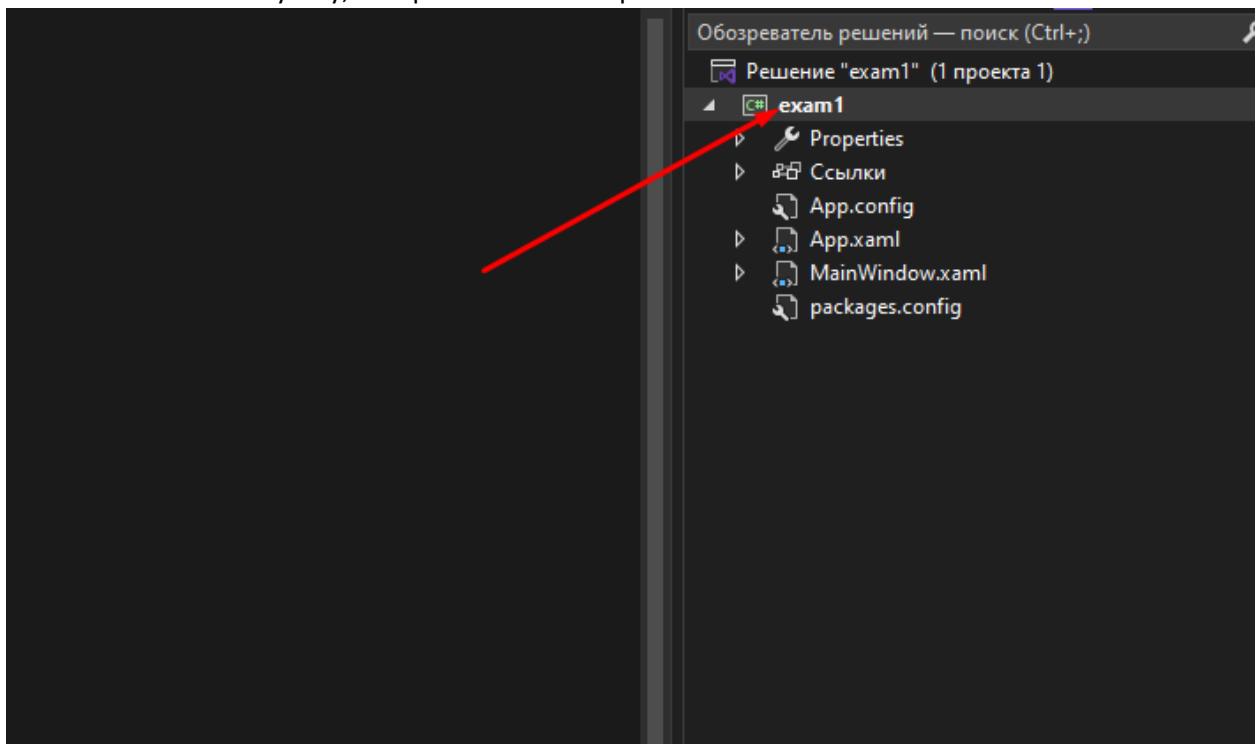
3. В появившися окне выберите свой сервер и нажмите Ok



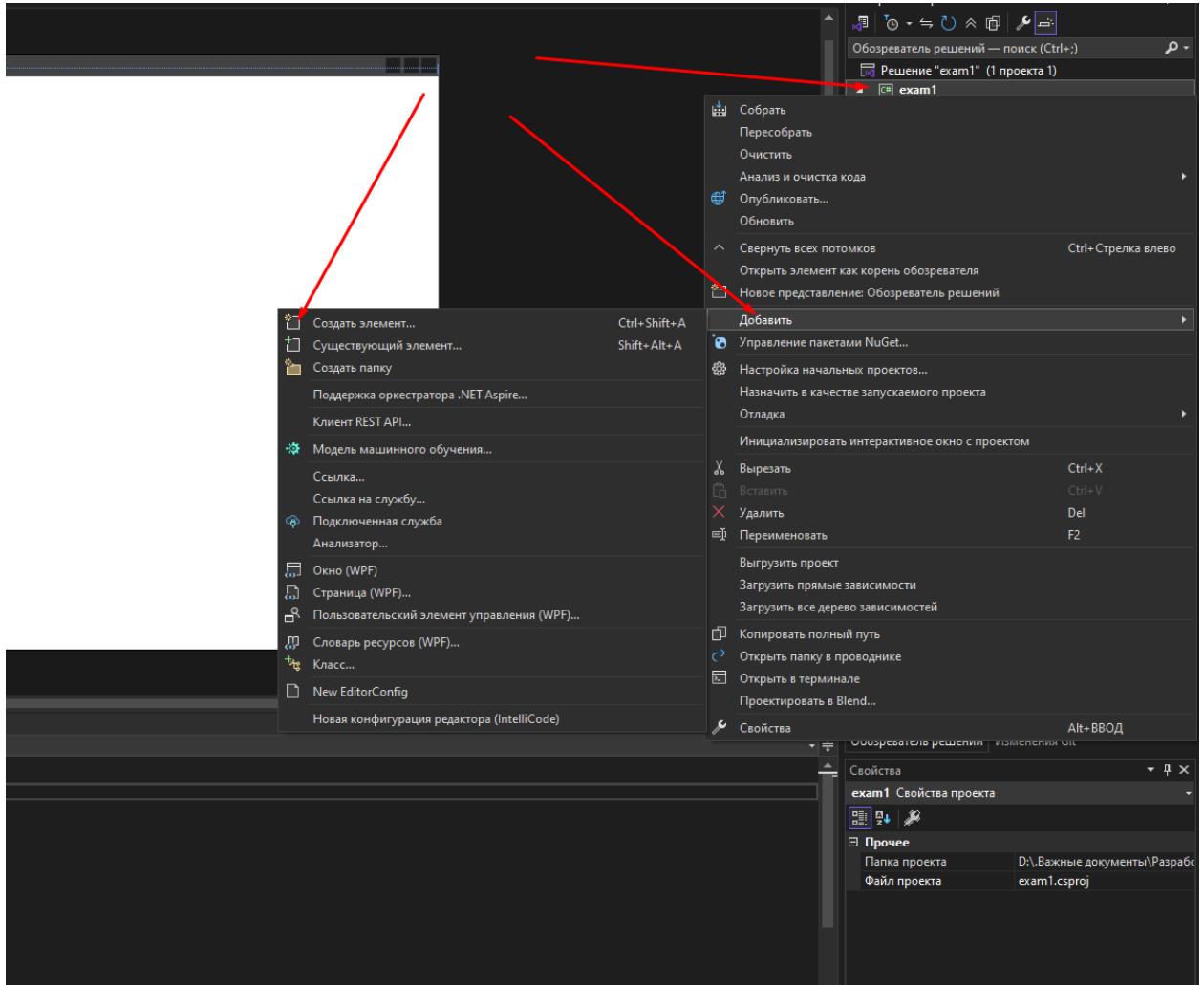
4. В списке баз данных раскройте треугольник, чтобы проверить соединение



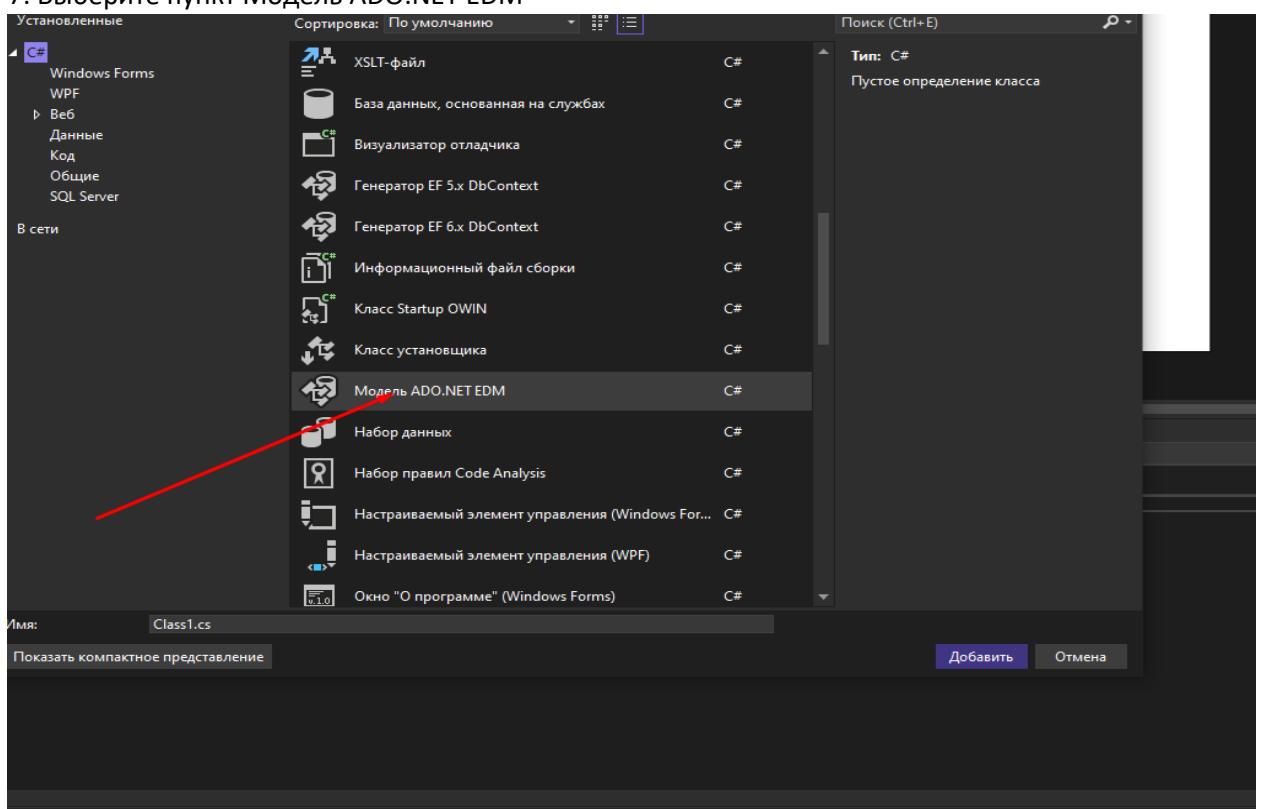
5. Нажмите ПКМ по пункту, который показан на фото



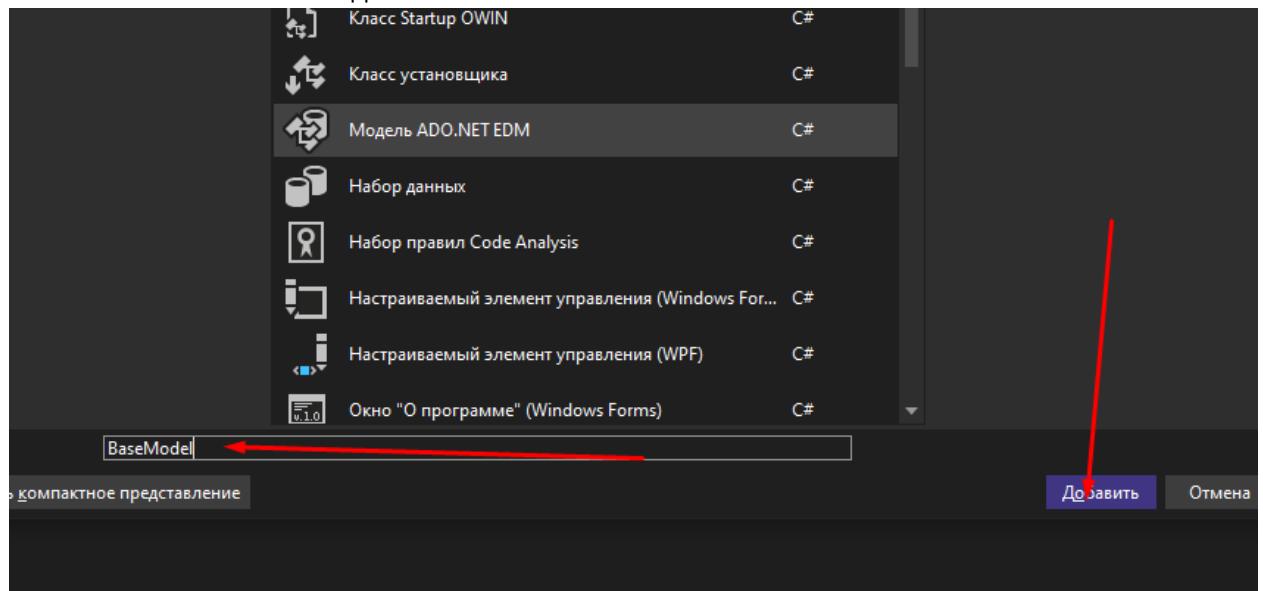
6. После выберите пункт Добавить и подпункт в нём Создать элемент...



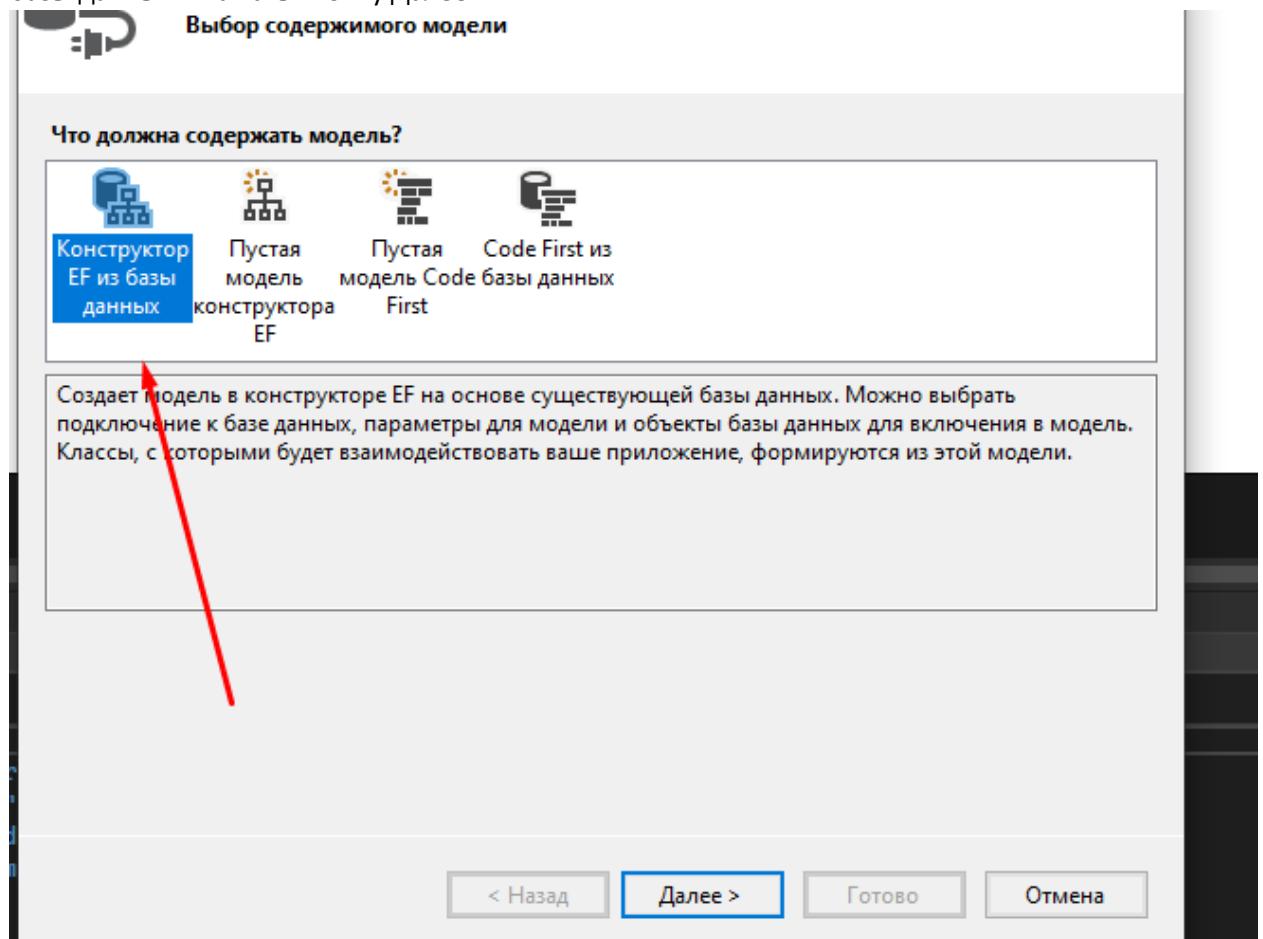
7. Выберите пункт Модель ADO.NET EDM



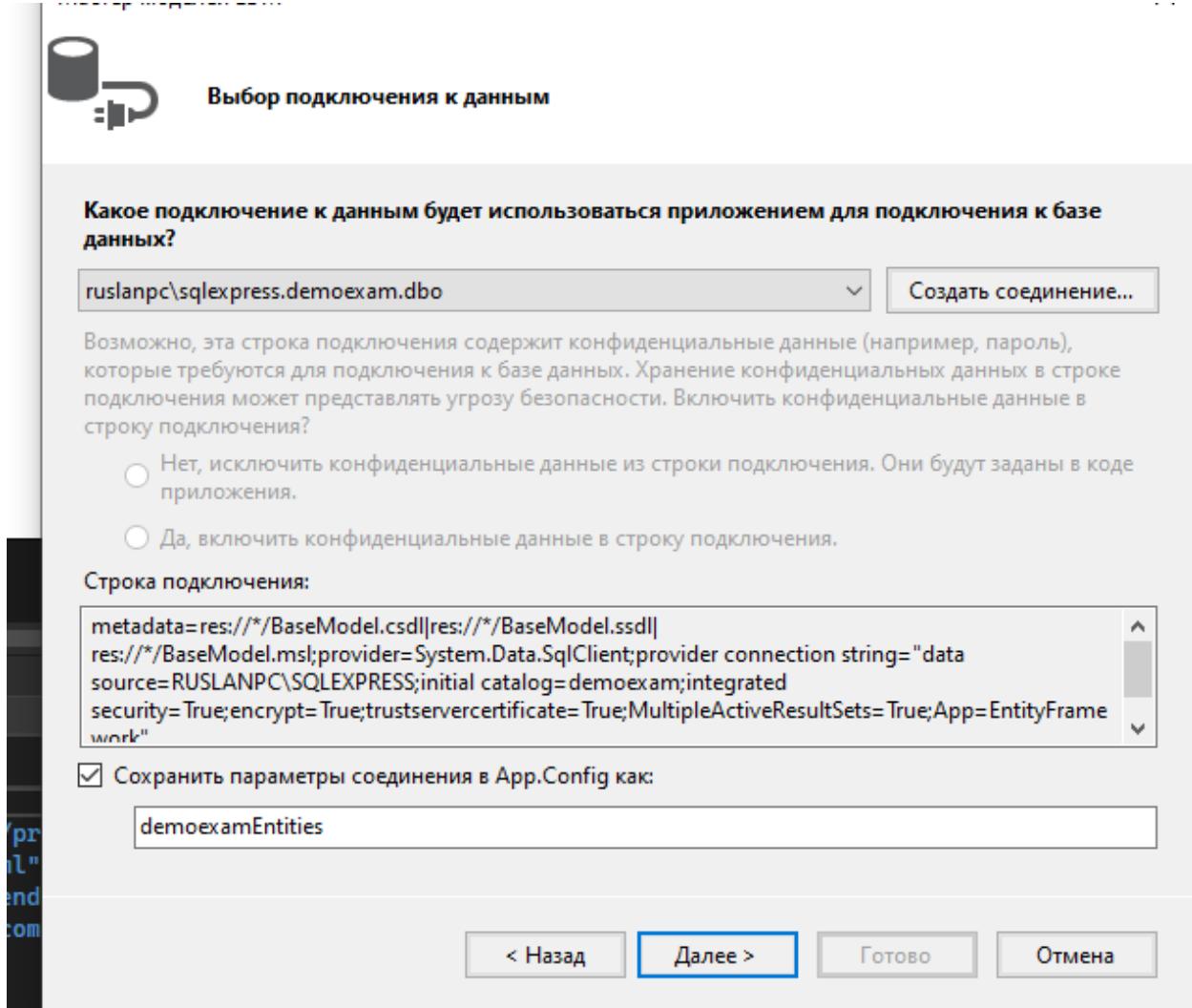
8. Назовите его и нажмите добавить



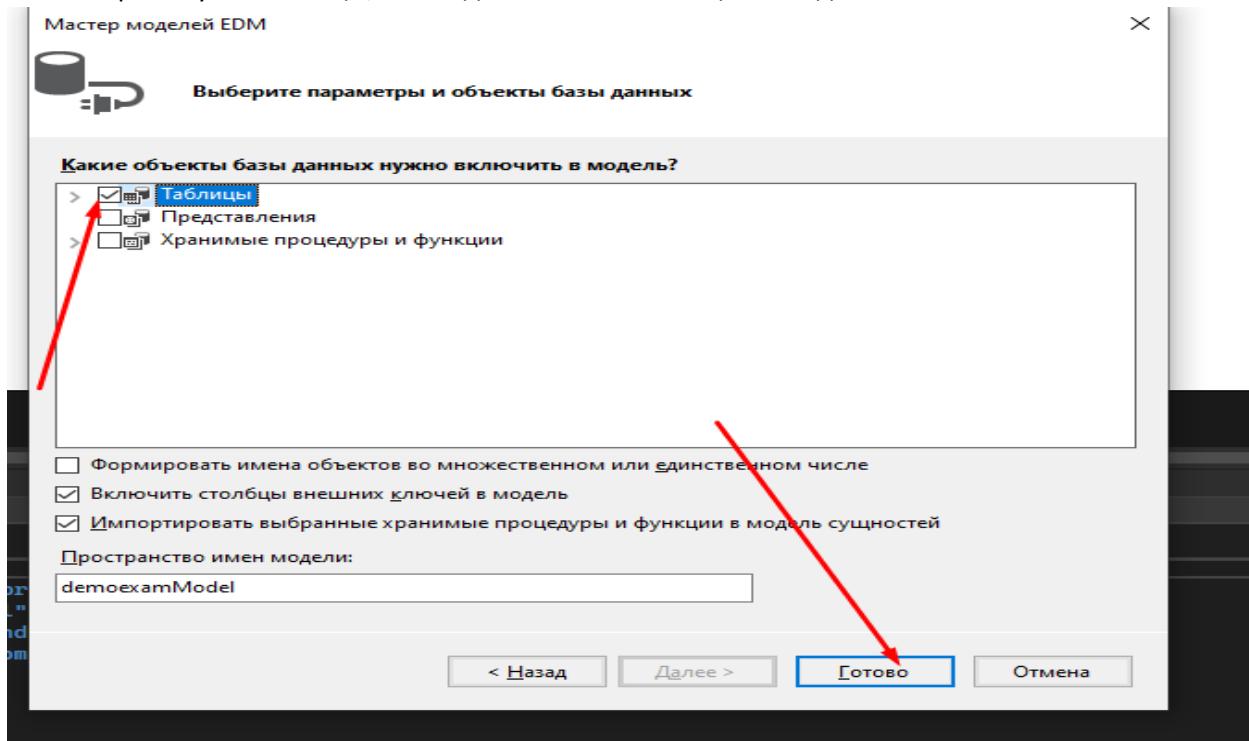
9. После кнопки добавить высветиться это окно, в нём нужно выбрать пункт Конструктор EF из базы данных и нажать кнопку Далее



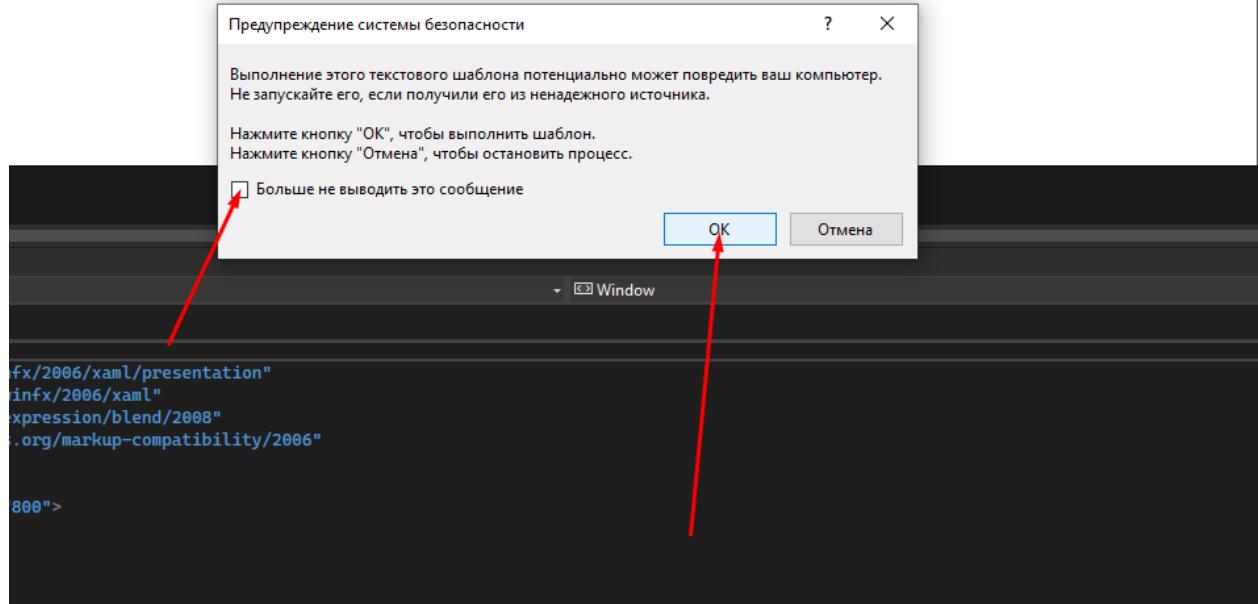
10. Выберите свою базу данных



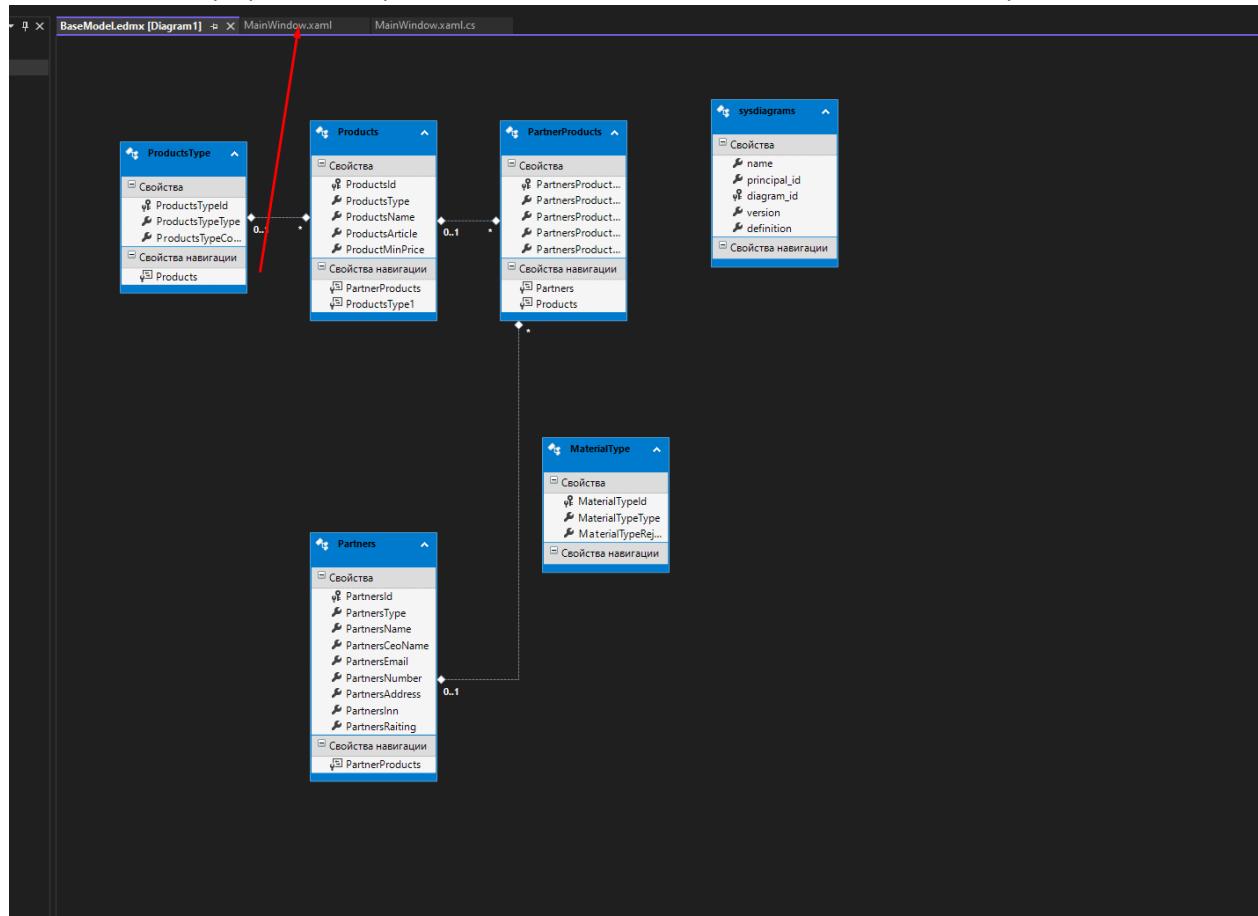
11. Выберите пункт Таблицы, чтобы добавить все таблицы базы данных



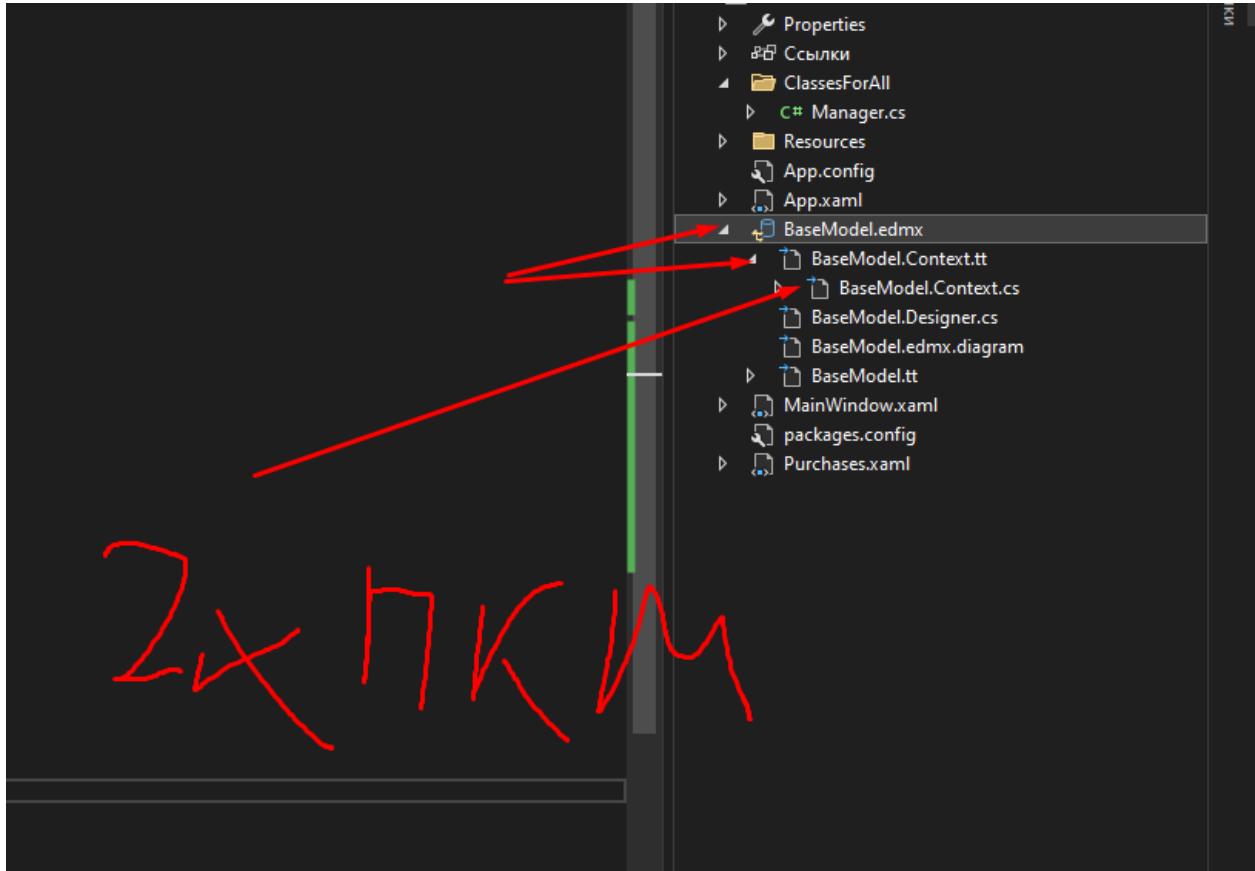
12. После нажатия кнопки Готово высветится это предупреждение. Нажмите пункт Больше не выводить это сообщение и нажмите кнопку Ok



13. Высветится графическое представление вашей базы данных, его можно закрыть



14. Раскрываем файл BaseModel.edmx и выбираем файл BaseModel.Context.cs



15. Создаём переменную \_context

```
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
```

```
using System;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;

Ссылка: 5
public partial class demoexamEntities : DbContext
{
    private static demoexamEntities _context;
    Ссылка: 1
    public demoexamEntities()
        : base("name=demoexamEntities")
    {
    }

    Ссылка: 1
    public static demoexamEntities GetContext()
    {
        if (_context == null)
            _context = new demoexamEntities();
        return _context;
    }

    Ссылка: 0
    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        throw new UnintentionalCodeFirstException();
    }

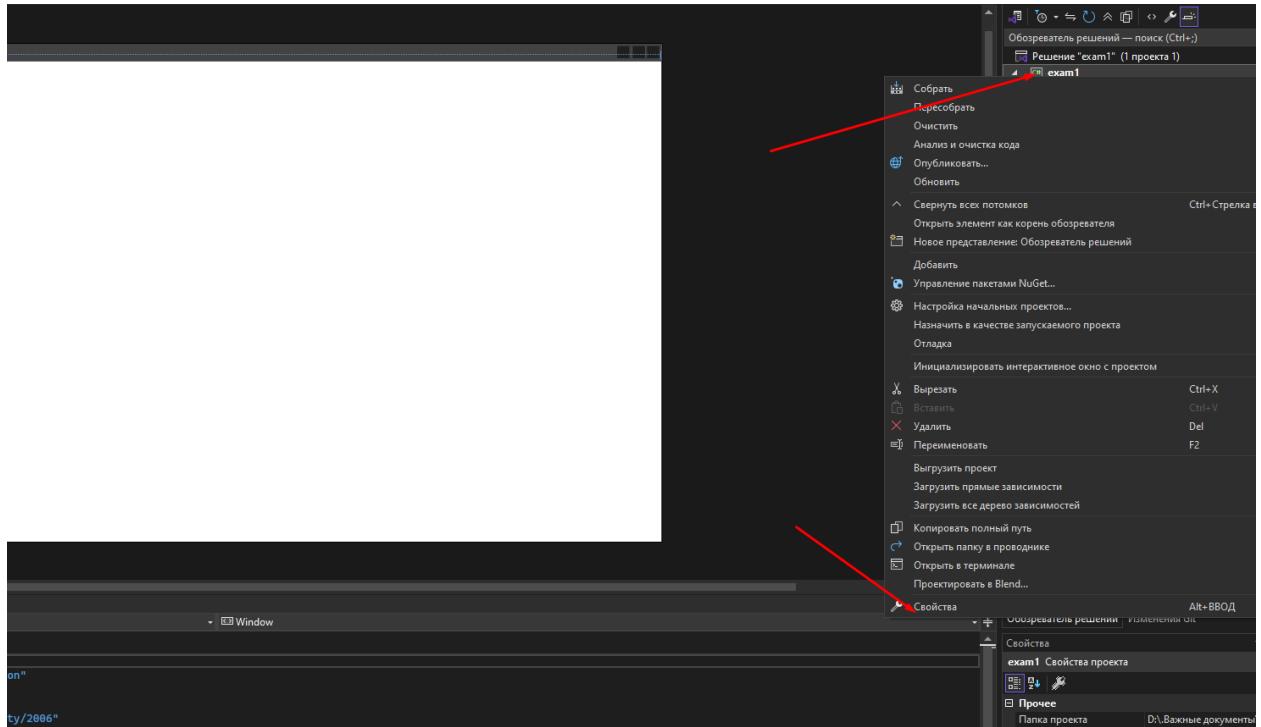
    Ссылка: 0
    public virtual DbSet<MaterialType> MaterialType { get; set; }
    Ссылка: 1
    public virtual DbSet<PartnerProducts> PartnerProducts { get; set; }
    Ссылка: 0
    public virtual DbSet<Partners> Partners { get; set; }
    Ссылка: 0
    public virtual DbSet<Products> Products { get; set; }
    Ссылка: 0
    public virtual DbSet<ProductsType> ProductsType { get; set; }
    Ссылка: 0
    public virtual DbSet<sysdiagrams> sysdiagrams { get; set; }
}
```

The screenshot shows the 'demoexamEntities.cs' code editor in Visual Studio. A red box highlights the declaration of the '\_context' variable and the 'GetContext()' method. Handwritten red text '2x наж' is written over the bottom left of the window.

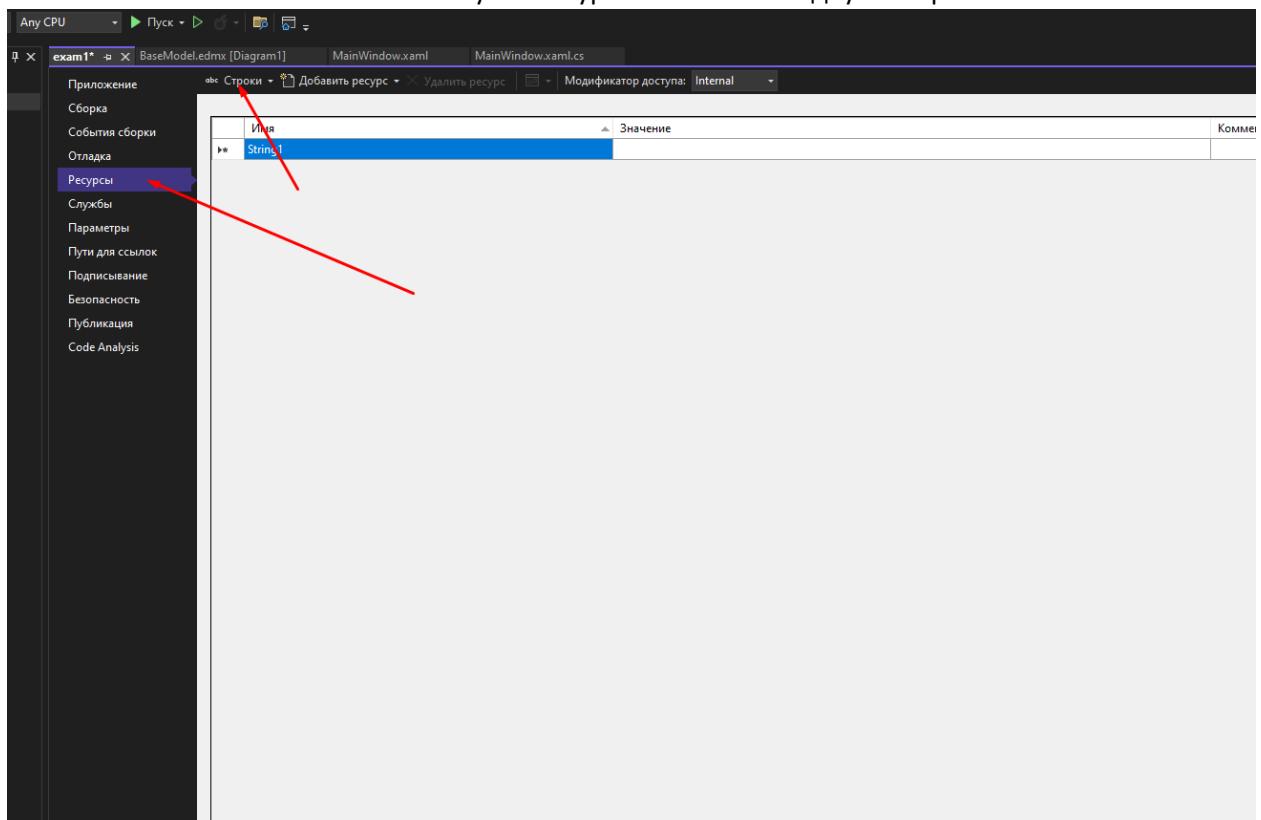
Поздравляю вы подключили базу данных к вашему приложению WPF!

## Добавление иконки приложения

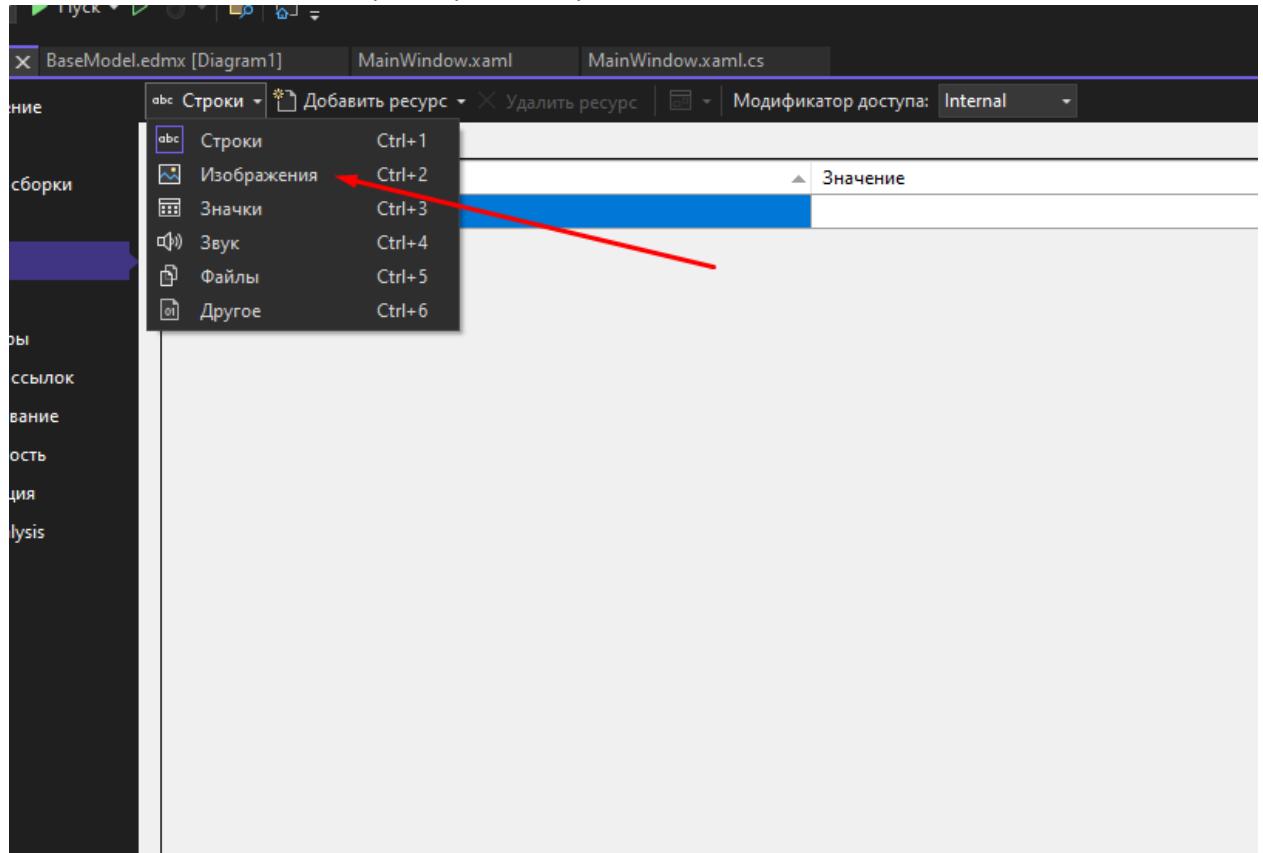
1. Нажмите ПКМ по пункту на фото и нажмите подпункт Свойства



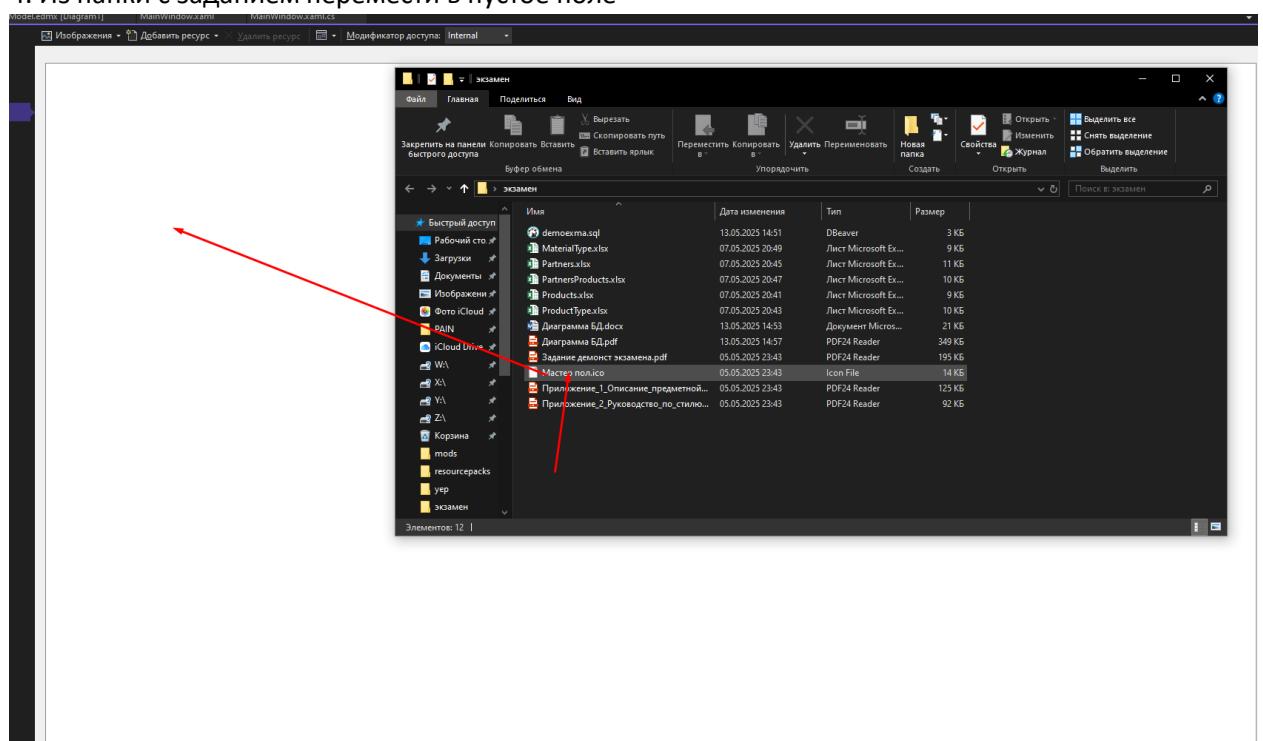
2. В появившемся окне нажмите на пункт Ресурсы и в нём на подпункт Строки



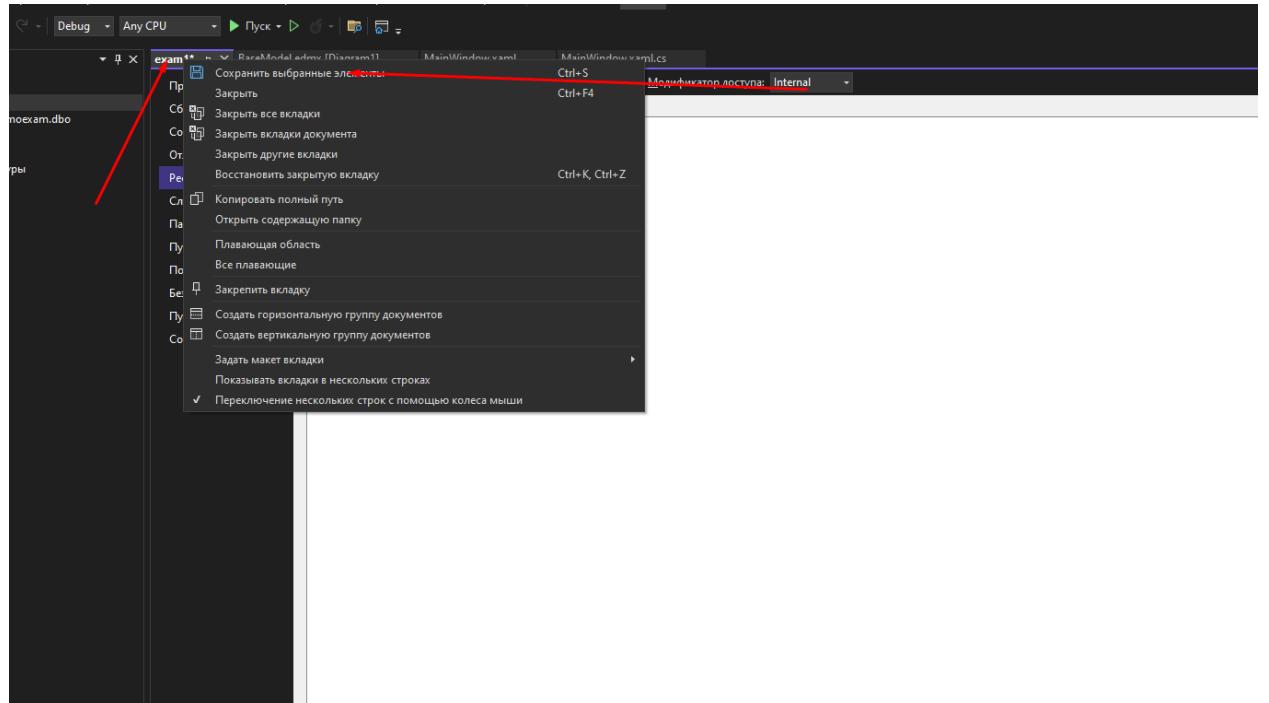
3. В появившемся окне выберите пункт Изображения



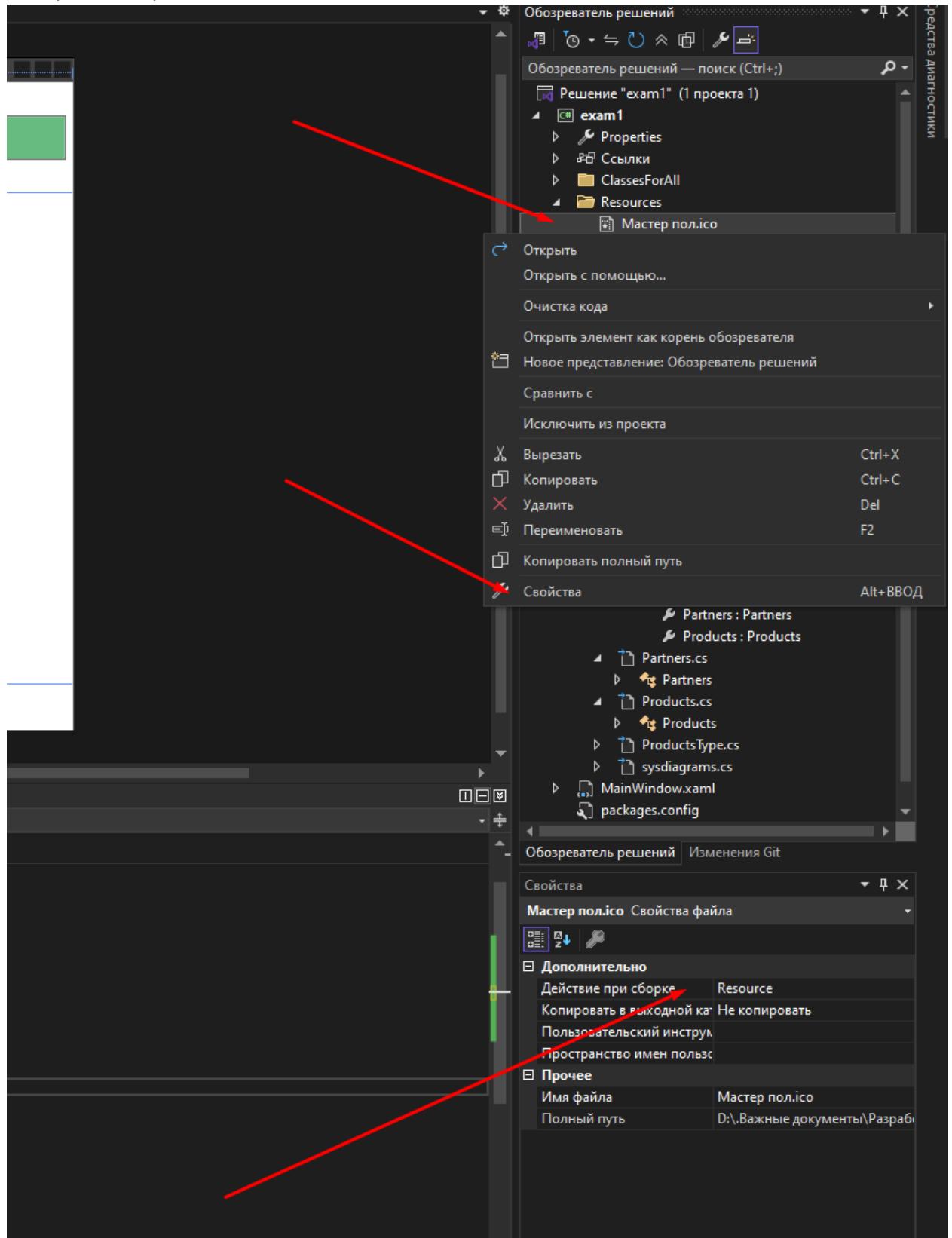
4. Из папки с заданием перемести в пустое поле



5. После иконки появится и можно нажимать по вкладке ПКМ и пункт Сохранить выбранные элементы

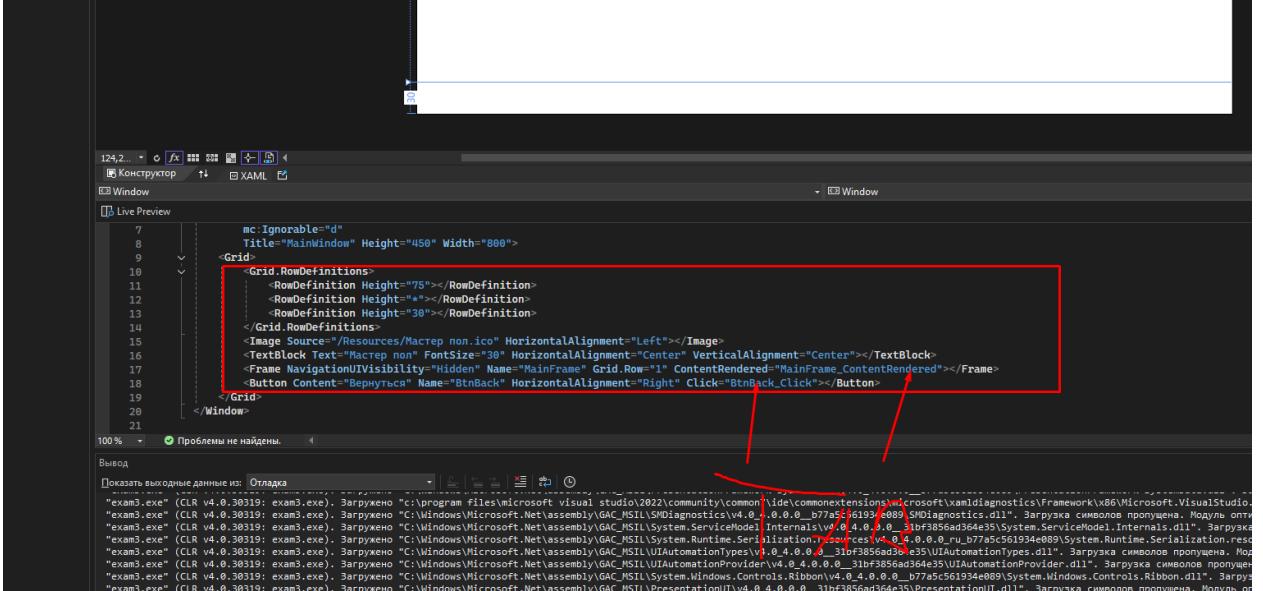


6. Нажмите ПКМ по иконке, нажмите пункт свойства и в нижнем окне в выпадающем списке выберите подпункт Resource



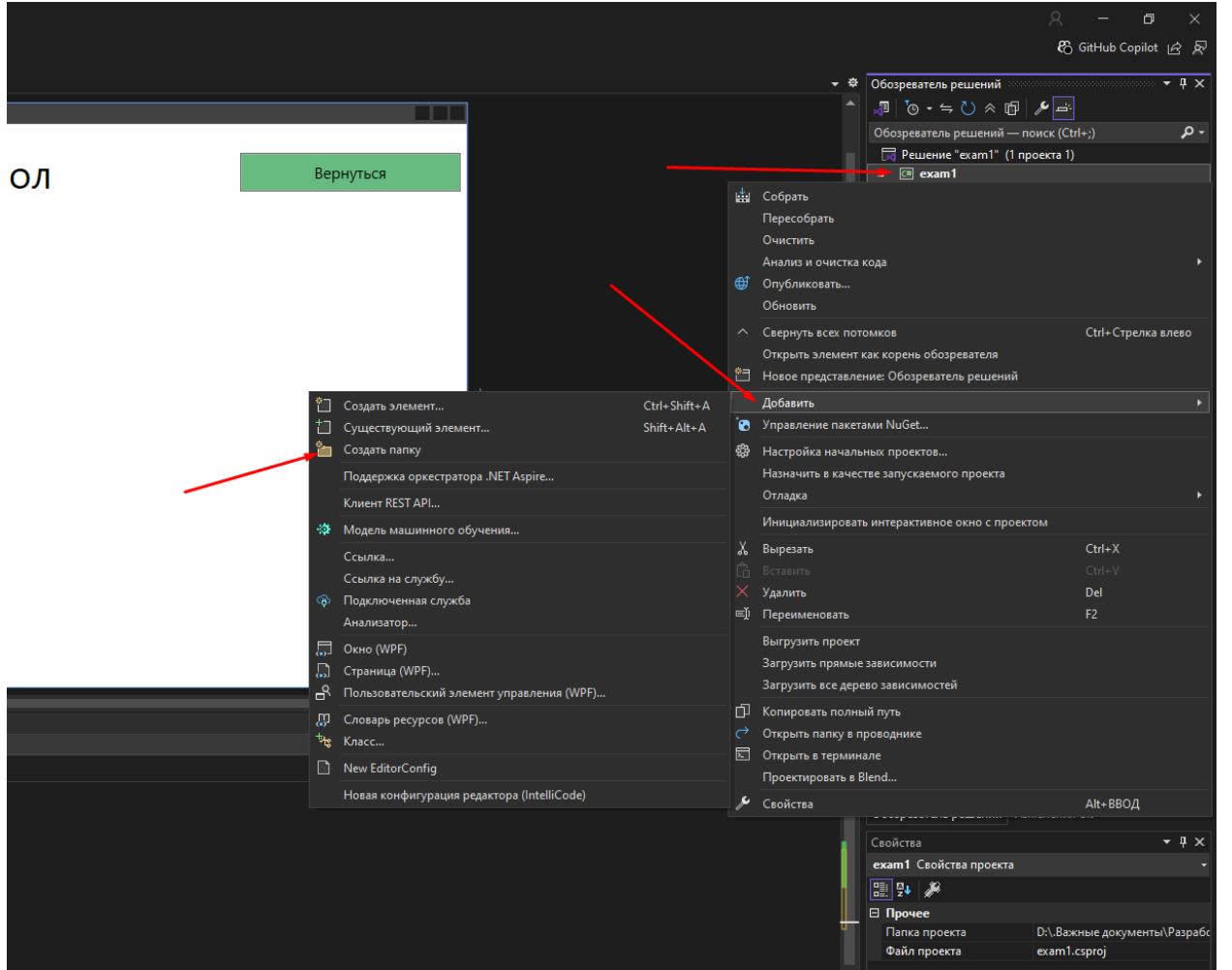
## Разработка приложения

### 1. В открывшемся окне в тэге Grid запишите данный код

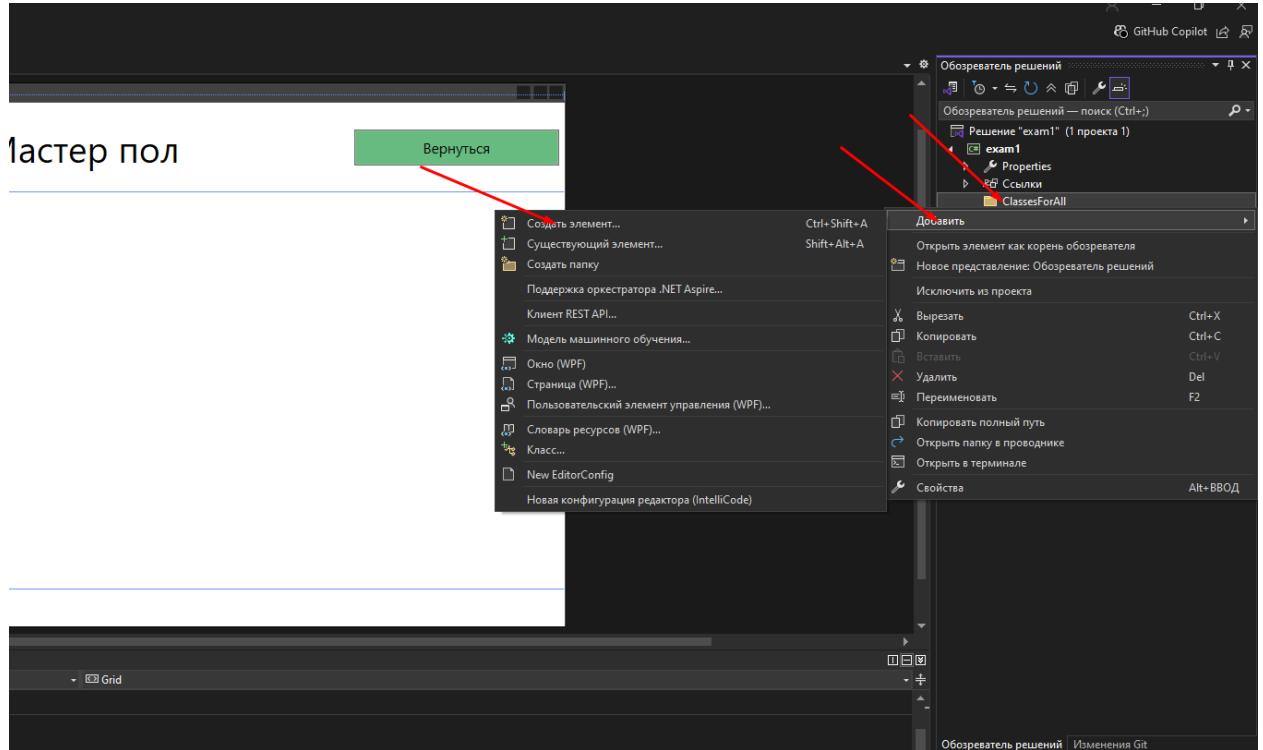


```
mc:Ignorable="d"
Title="MainWindow" Height="450" Width="800">
<Grid>
<Grid.RowDefinitions>
<RowDefinition Height="75"/>
<RowDefinition Height="*"/>
<RowDefinition Height="30"/>
</Grid.RowDefinitions>
<Image Source="/Resources/Master non.ico" HorizontalAlignment="Left"/>
<TextBlock Text="Мастер non" FontSize="30" HorizontalAlignment="Center" VerticalAlignment="Center"/>
<Frame NavigationUIVisibility="Hidden" Name="MainFrame" Grid.Row="1" ContentRendered="MainFrame_ContentRendered"/>
<Button Content="Вернуться" Name="BtnBack" HorizontalAlignment="Right" Click="BtnBack_Click"/>
</Grid>
</Window>
```

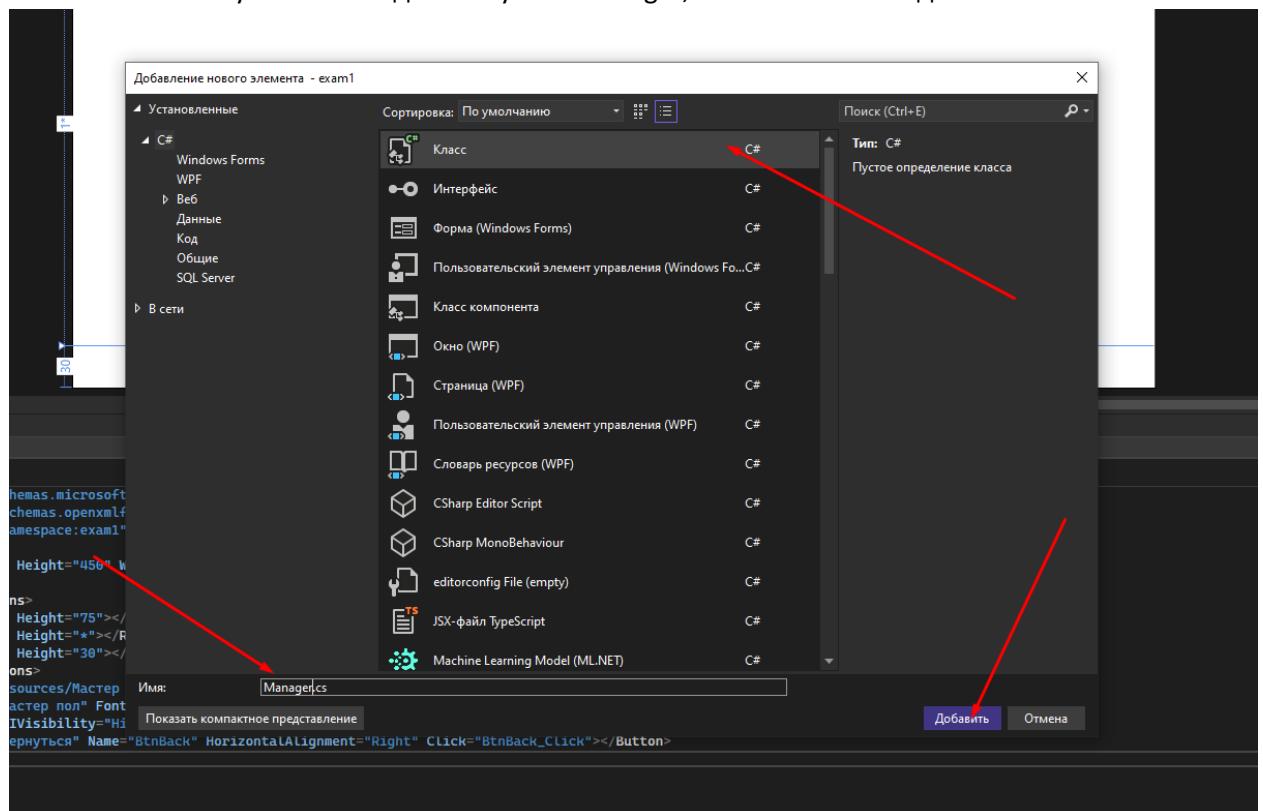
### 2. Далее пкм по exam1 и создайте новую папку с названием ClassesForAll



3. Далее пкм по папке и нажимаем на пункт Добавить и в нём подпункт Создать элемент...



4. Нажимаем на пункт Класс и даём ему имя Manager, после нажимаем добавить

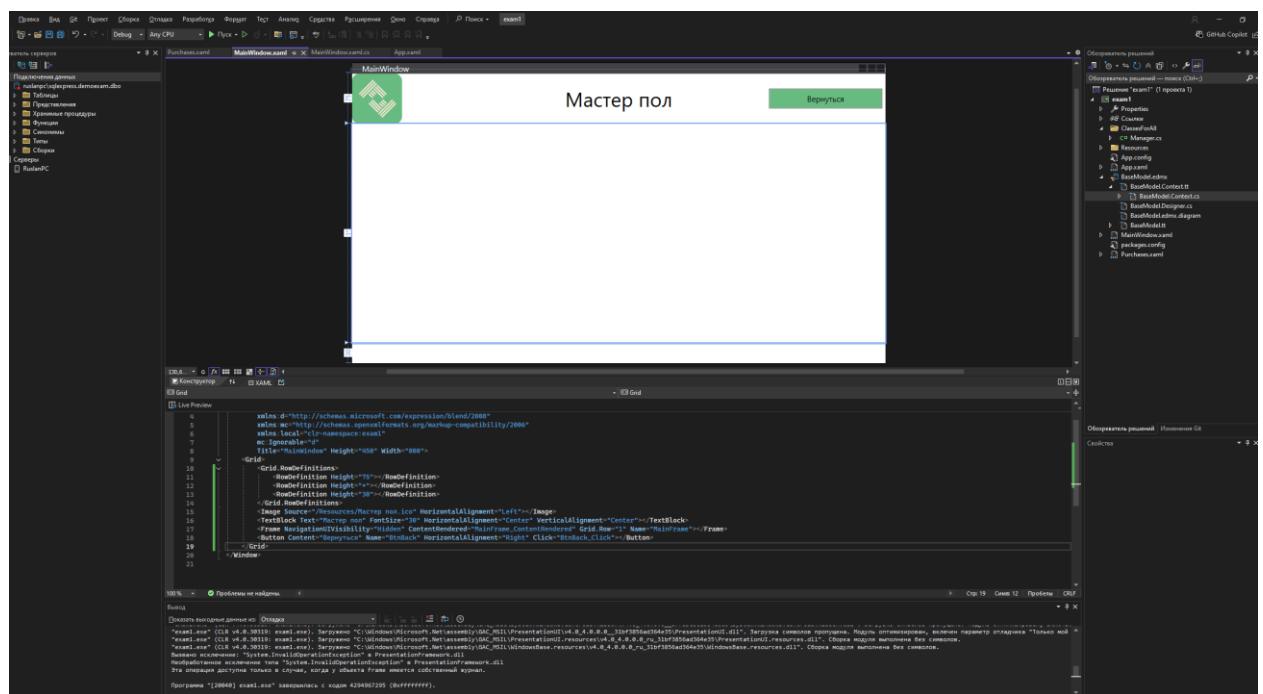


5. В самом классе пишем такой код, при ошибке в слове Frame в самом верху дописываем using System.Windows.Controls;

```
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Windows.Controls;
7
8  namespace exam1.ClassesForAll
9  {
10    Ссылка 0
11    internal class Manager
12    {
13      Ссылка 0
14      public static Frame MainFrame { get; set; }
15    }
16 }
```

TAB

6. Возвращаемся на главный экран



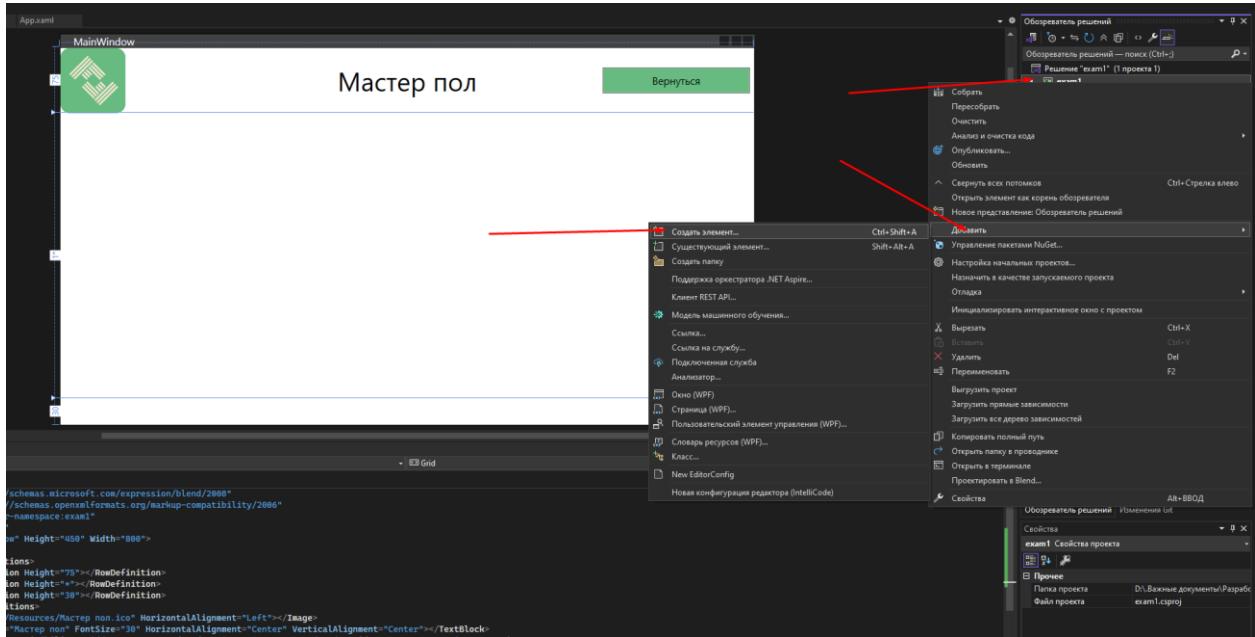
7. Нажимаем F7 и в этом файле прописываем данный код, будет ошибка пока что

```
5  using System; using System.Threading.Tasks; using System.Windows; using System.Windows.Controls; using System.Windows.Data; using System.Windows.Documents; using System.Windows.Input; using System.Windows.Media; using System.Windows.Media.Imaging; using System.Windows.Navigation; using System.Windows.Shapes; 16  namespace exam3 17  { 18      /// <summary> 19      /// Логика взаимодействия для MainWindow.xaml 20      /// </summary> 21      public partial class MainWindow : Window 22      { 23          //Ссылка на окно 24          public MainWindow() 25          { 26              InitializeComponent(); 27              Manager.MainFrame = MainFrame; 28              Manager.MainFrame.Navigate(new PurchasesPage()); 29          } 30          //Событие 31          private void MainFrame_ContentRendered(object sender, EventArgs e) 32          { 33              if(Manager.MainFrame.CanGoBack) 34              { 35                  BtnBack.Visibility = Visibility.Visible; 36              } 37              else 38              { 39                  BtnBack.Visibility = Visibility.Hidden; 40              } 41          } 42          //Событие 43          private void BtnBack_Click(object sender, RoutedEventArgs e) 44          { 45              Manager.MainFrame.GoBack(); 46          } 47      } 48  
```

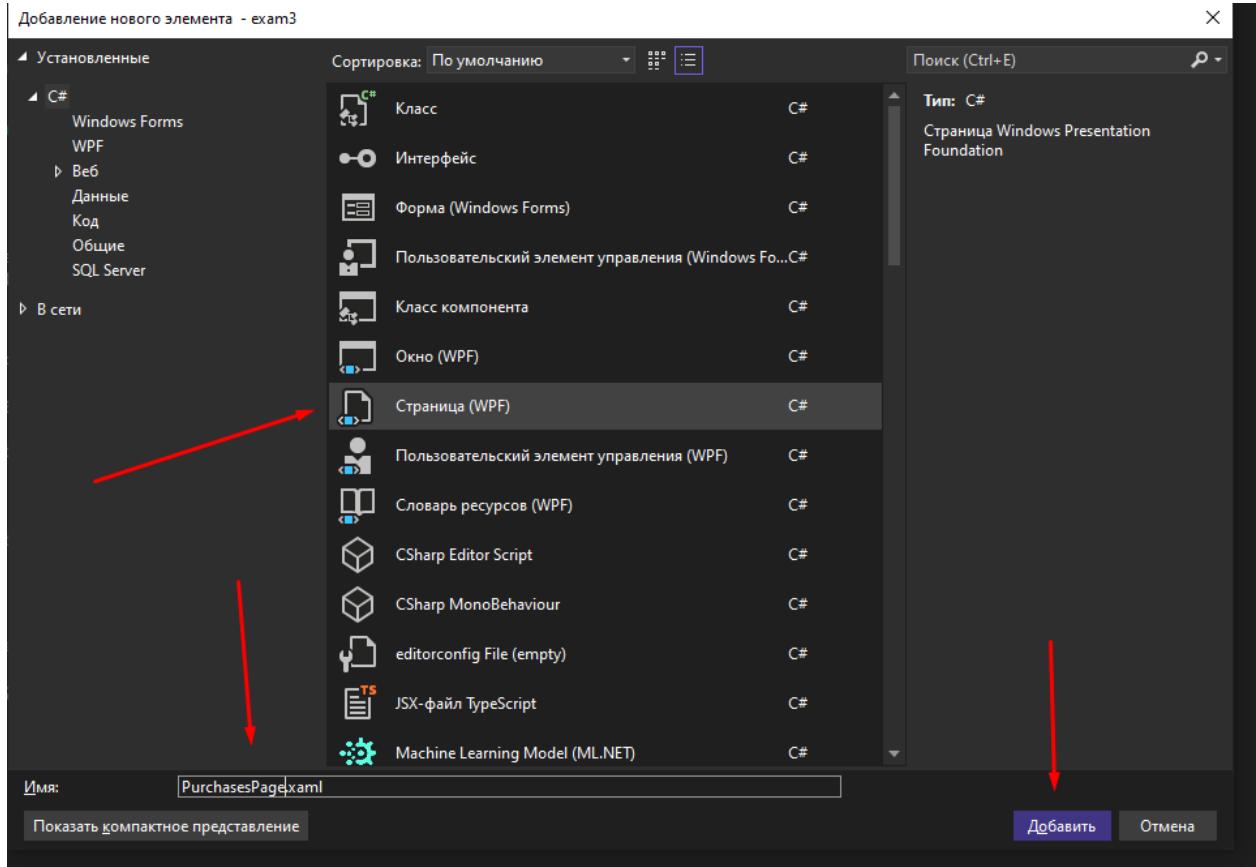
Поздравляю вы создали главную страницу!

Создание страницы с заказами

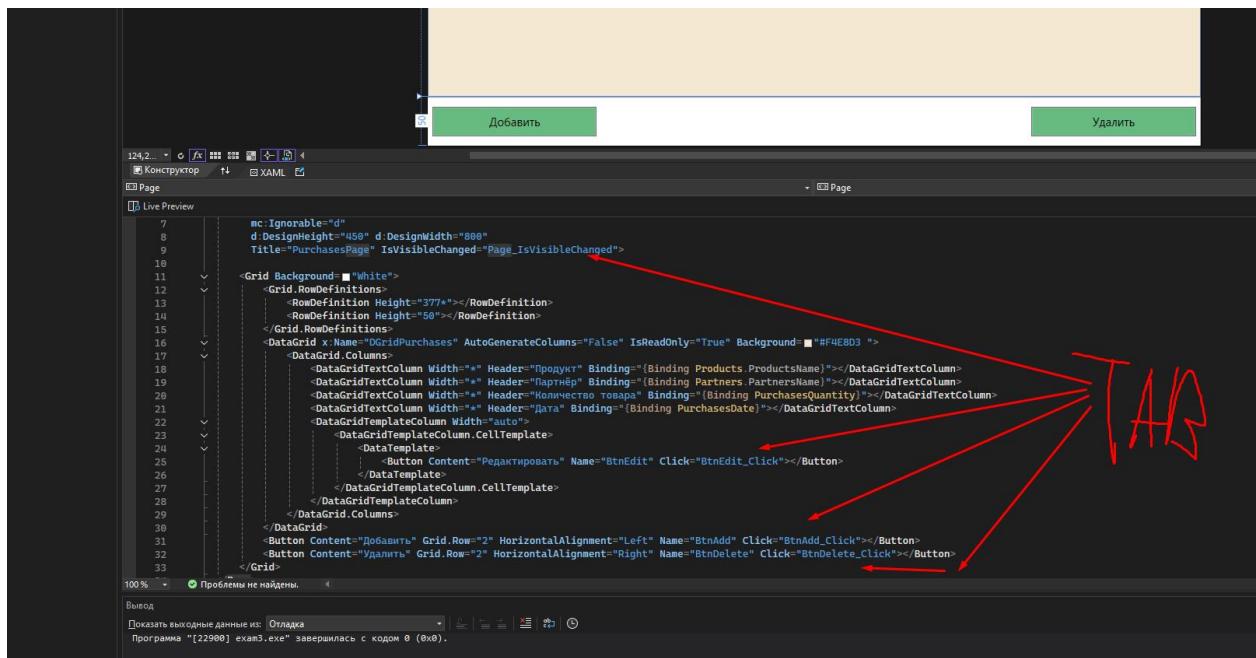
1. Нажмите пкм по exam1 и выберите подпункт Создать элемент... в пункте Добавить



2. Выберите пункт Страница(WPF) и назовите её Purchases.xaml, после нажмите добавить



3. Заполняем контейнер Grid вёрсткой



#### 4. Нажимаем F7 и заполняем кодом файл

```

using System.Data.Entity;
namespace exam3
{
    // ...
    public partial class PurchasesPage : Page
    {
        // ...
        public PurchasesPage()
        {
            InitializeComponent();
            DGridPurchases.ItemsSource = demoexamEntities.GetContext().Purchases.Include(p => p.Products).Include(p => p.Partners).ToList();
        }
    }
}

private void BtnEdit_Click(object sender, RoutedEventArgs e)
{
    var selecteditem = (sender as Button).DataContext as Purchases;
    var context = demoexamEntities.GetContext();
    var loadedItem = context.Purchases.Include(p => p.Products).FirstOrDefault(p => p.PurchasesId == selecteditem.PurchasesId);
    Manager.MainFrame.Navigate(new AddEditPage(loadedItem));
}

private void BtnAdd_Click(object sender, RoutedEventArgs e)
{
    Manager.MainFrame.Navigate(new AddEditPage(null));
}

private void BtnDelete_Click(object sender, RoutedEventArgs e)
{
    var purchasesForRemoving = DGridPurchases.SelectedItems.Cast<Purchases>().ToList();
    if (MessageBox.Show($"Вы точно хотите удалить {purchasesForRemoving.Count()} элементов?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
    {
        try
        {
            demoexamEntities.GetContext().Purchases.RemoveRange(purchasesForRemoving);
            demoexamEntities.GetContext().SaveChanges();
            MessageBox.Show("Данные удалены!");
            DGridPurchases.ItemsSource = demoexamEntities.GetContext().Purchases.ToList();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message.ToString());
        }
    }
}

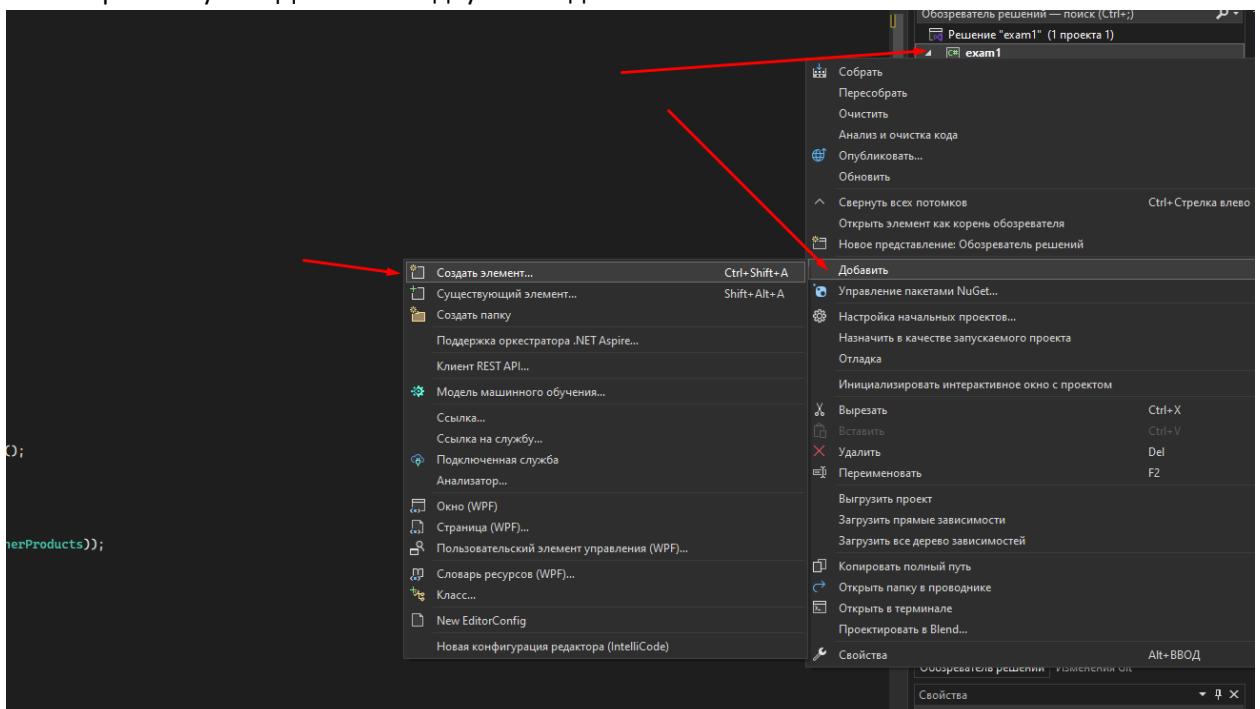
private void Page_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)
{
    if (Visibility == Visibility.Visible)
    {
        demoexamEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());
        DGridPurchases.ItemsSource = demoexamEntities.GetContext().Purchases.Include(p => p.Products).Include(p => p.Partners).ToList();
    }
}

```

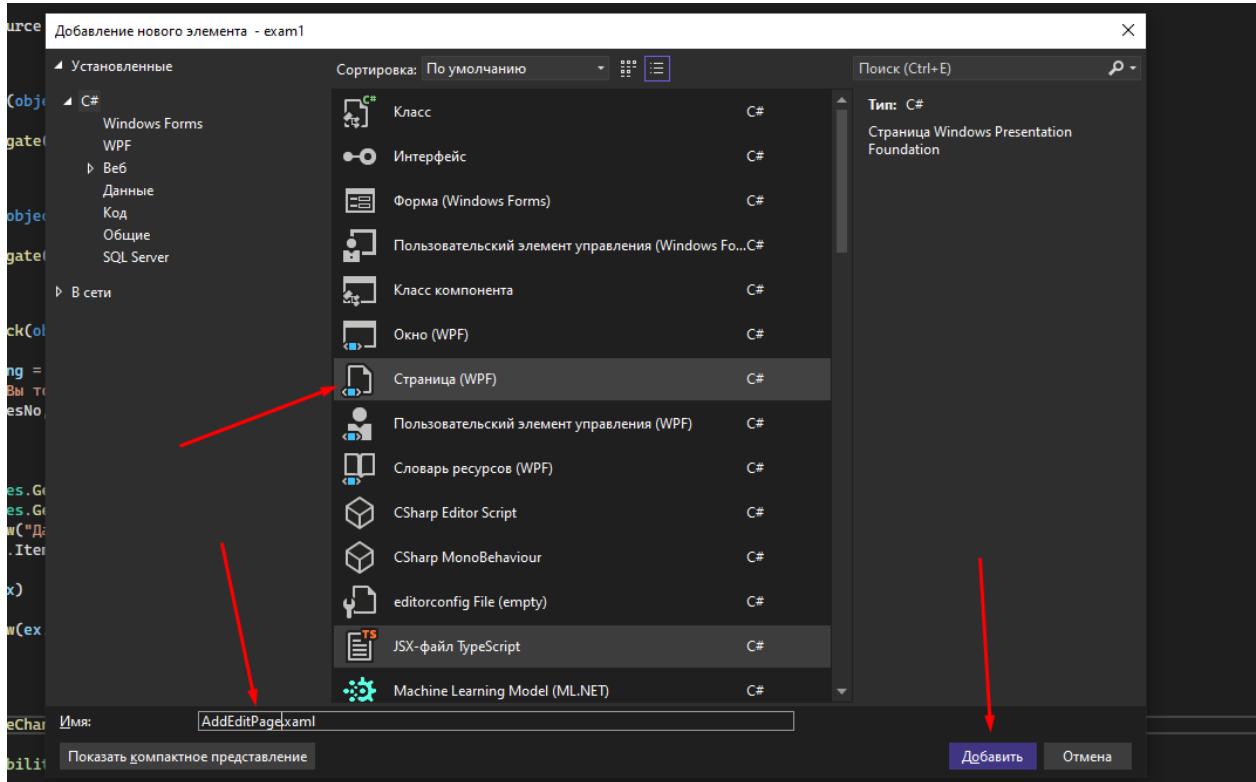
Поздравляю, вторая страница готова!

#### Создание третьей страницы

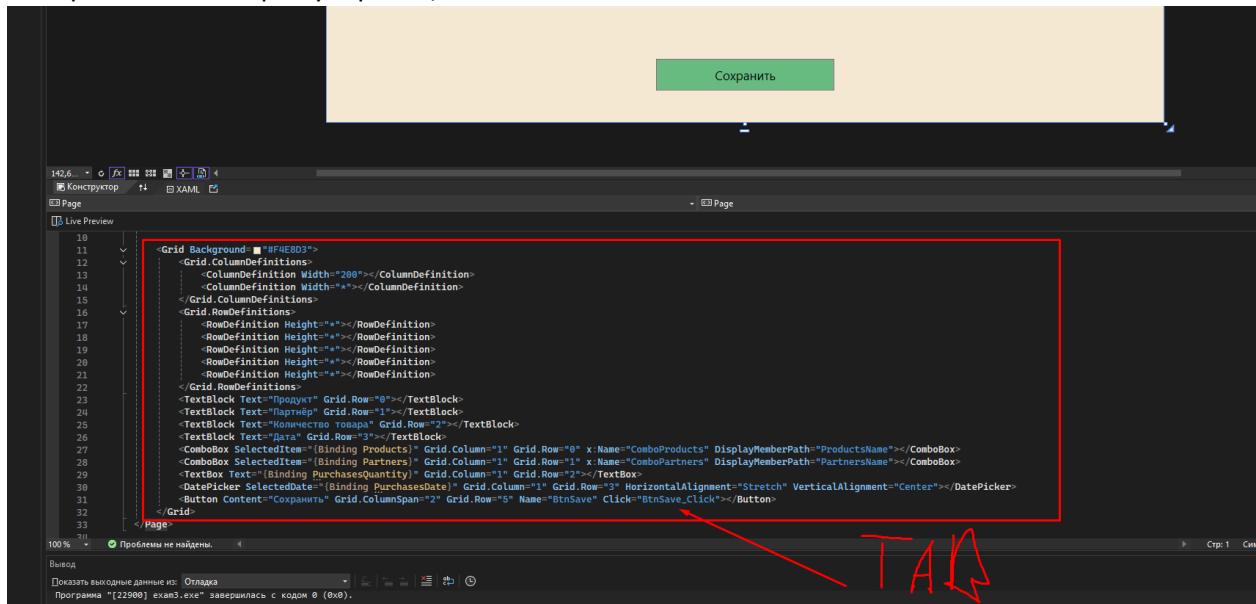
##### 1. Выберите в пункте Добавить подпункт Создать элемент...



2. Выберите пункт Страница(WPF) и назовите ее, после нажмите Добавить



3. Переписываем вёрстку страницы и нажимаем F7



#### 4. Переписываем код в этом файле

```

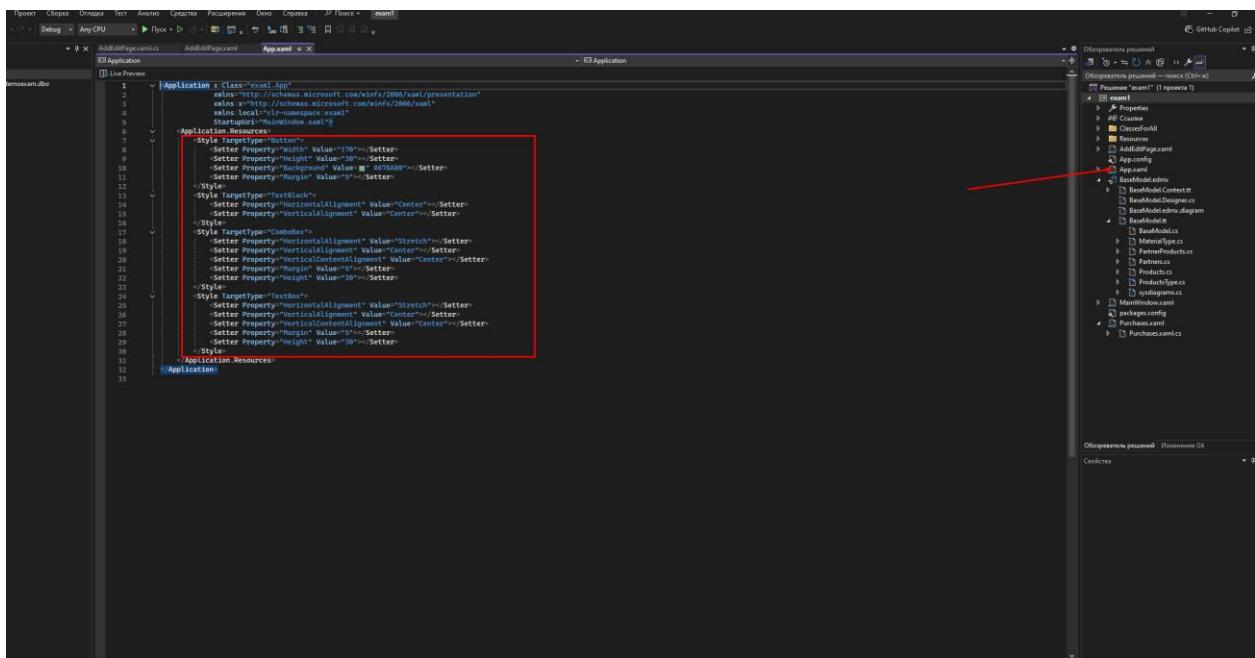
16  namespace exam3
17  {
18      /// <summary>
19      /// Логика взаимодействия для AddEditPage.xaml
20      /// </summary>
21      [Codeless]
22      public partial class AddEditPage : Page
23      {
24          private Purchases _currentPurchase = new Purchases();
25          public AddEditPage(Purchases selectedPurchase)
26          {
27              InitializeComponent();
28              if (selectedPurchase != null)
29                  _currentPurchase = selectedPurchase;
30              DataContext = _currentPurchase;
31              ComboProducts.ItemsSource = demoexamEntities.GetContext().Products.ToList();
32              ComboPartners.ItemsSource = demoexamEntities.GetContext().Partners.ToList();
33          }
34
35          private void BtnSave_Click(object sender, RoutedEventArgs e)
36          {
37              StringBuilder errors = new StringBuilder();
38              if (_currentPurchase.Products == null)
39                  errors.AppendLine("Выберите продукт");
40              if (_currentPurchase.Partner == null)
41                  errors.AppendLine("Выберите партнера");
42              if (_currentPurchase.PurchasesDate == null)
43                  errors.AppendLine("Установите дату > 0");
44              if (_currentPurchase.PurchasesDate < 0)
45                  errors.AppendLine("Установите дату > 0");
46              if (_currentPurchase.PurchasesDate == null)
47                  errors.AppendLine("Установите дату");
48              if (errors.Length > 0)
49              {
50                  MessageBox.Show(errors.ToString());
51                  return;
52              }
53
54              if (demoexamEntities.GetContext().Entry(_currentPurchase).State == EntityState.Detached)
55              {
56                  demoexamEntities.GetContext().Purchases.Add(_currentPurchase);
57              }
58
59              try
60              {
61                  demoexamEntities.GetContext().SaveChanges();
62                  MessageBox.Show("Изменения сохранены");
63                  Manager.MainFrame.GoBack();
64              }
65              catch (Exception ex)
66              {
67                  string errorMessage = "Ошибка: " + ex.Message;
68                  while (ex.InnerException != null)
69                  {
70                      ex = ex.InnerException;
71                      errorMessage += "\nInnerException: " + ex.Message;
72                  }
73                  MessageBox.Show(errorMessage);
74              }
75          }
76      }
77  }
78

```

Более простой метод  
MessageBox.Show(ex.Message.ToString());

#### Стили

Открываем файл App.xaml и переписываем данное содержимое



#### Руководство по стилю

Все экранные формы пользовательского интерфейса должны иметь заголовок. Кроме того, на главной форме должен быть установлен логотип (представлен в ресурсах). Логотип не искажать: не менять изображение, пропорции изображения, цвет.

Для приложения должна быть установлена иконка (дана в ресурсах), если это реализуемо в рамках платформы.

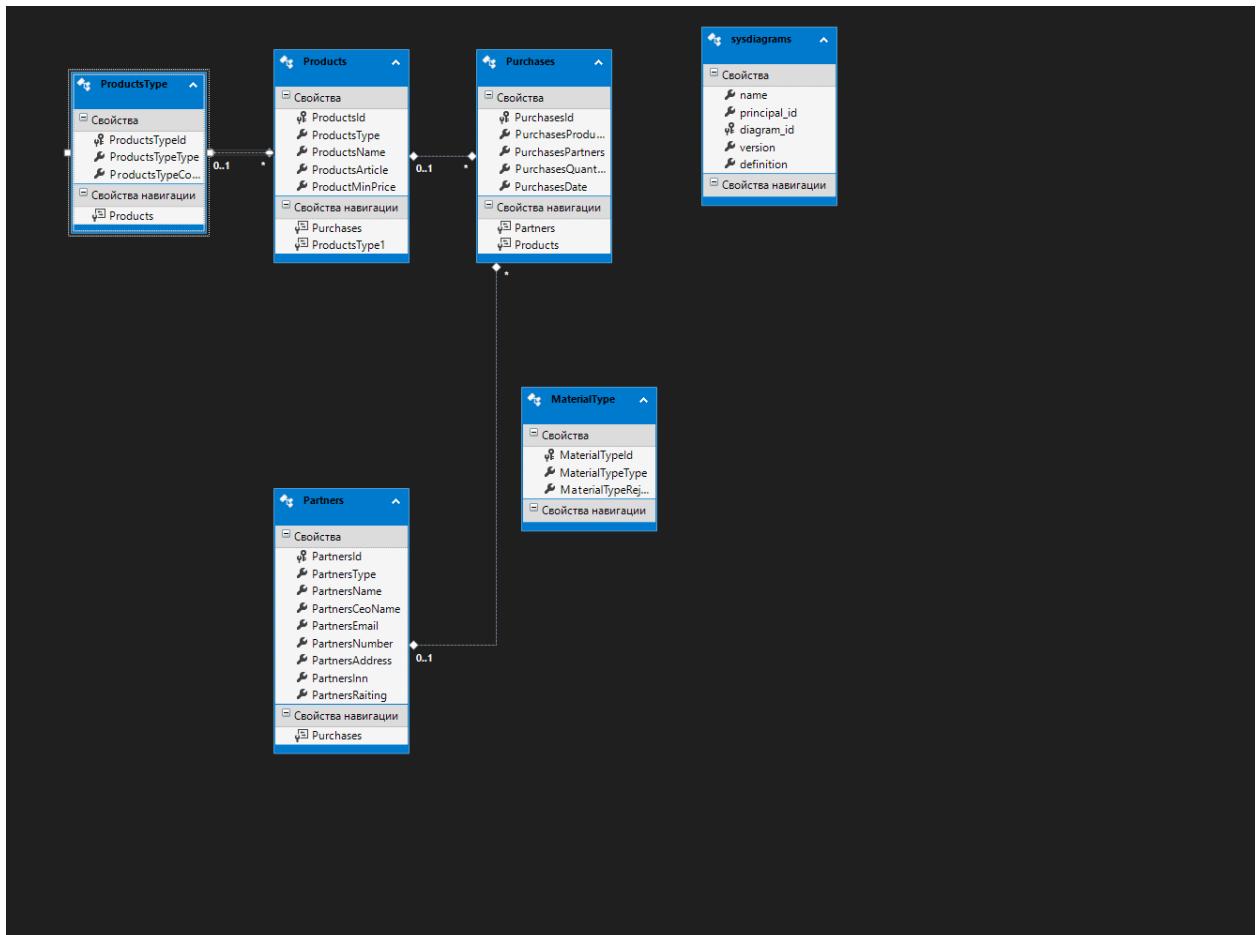
Использовать шрифт Segoe UI.

В качестве основного фона используется белый цвет, дополнительного фона используется цвет #F4E8D3. Для акцентирования внимания пользователя на целевом действии интерфейса используется цвет #67BA80.

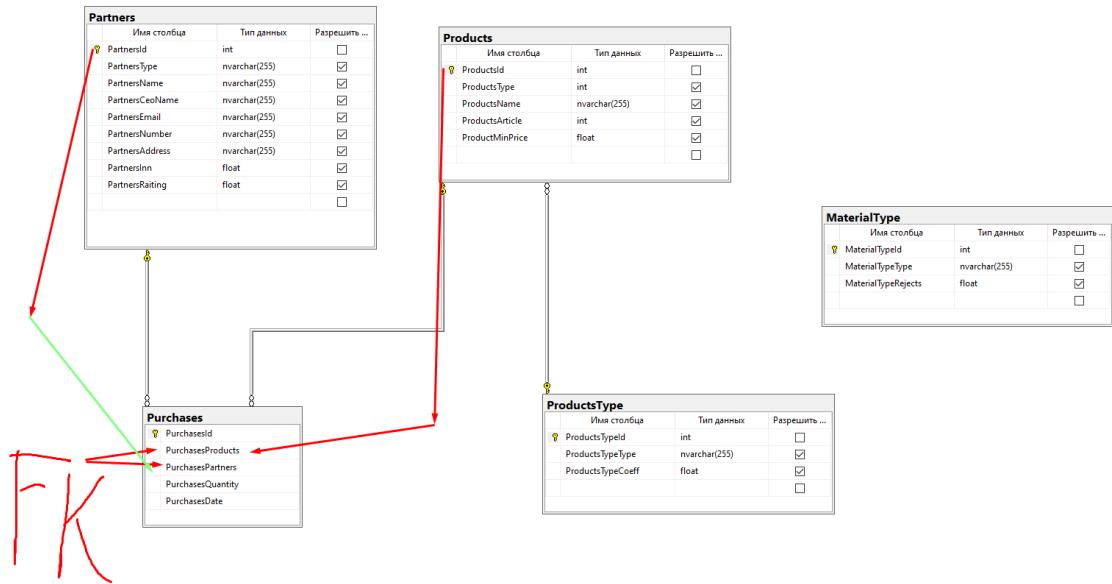
Основной фон	Дополнительный фон	Акцентирование внимания
#FFFFFF	#F4E8D3	#67BA80

## Схема базы данных

1.



2.



Можно смело запускать проект, всё готово!