**Лабораторная работа**

**Фильтрация и сортировка табличных данных**

**Рассмотрим задание.**

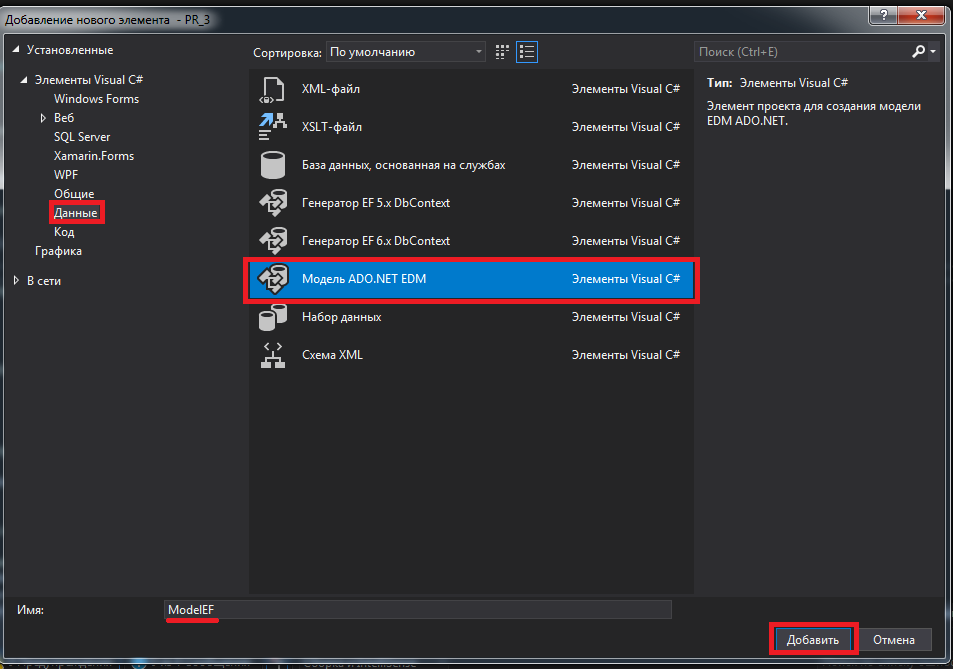
|  |
| --- |
| **Требуется**: создать программу, в которой будет модель работы с БД User\_Data и будет включать 2 таблицы Users, Roles. Осуществить авторизацию пользователей и вход на определённую форму в зависимости от роли. |

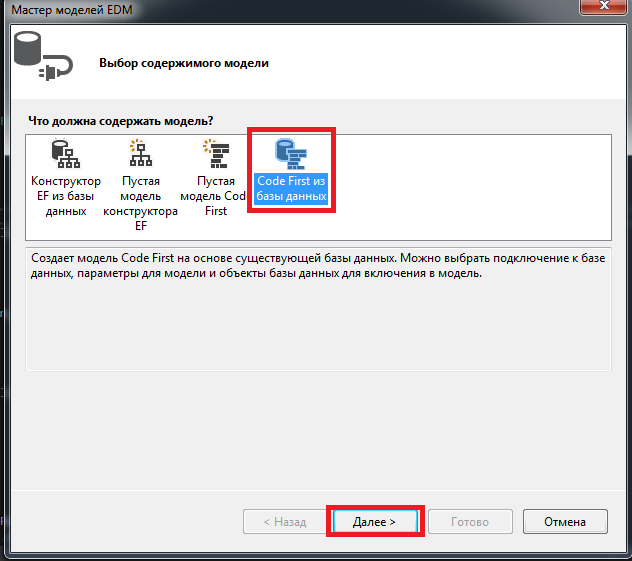
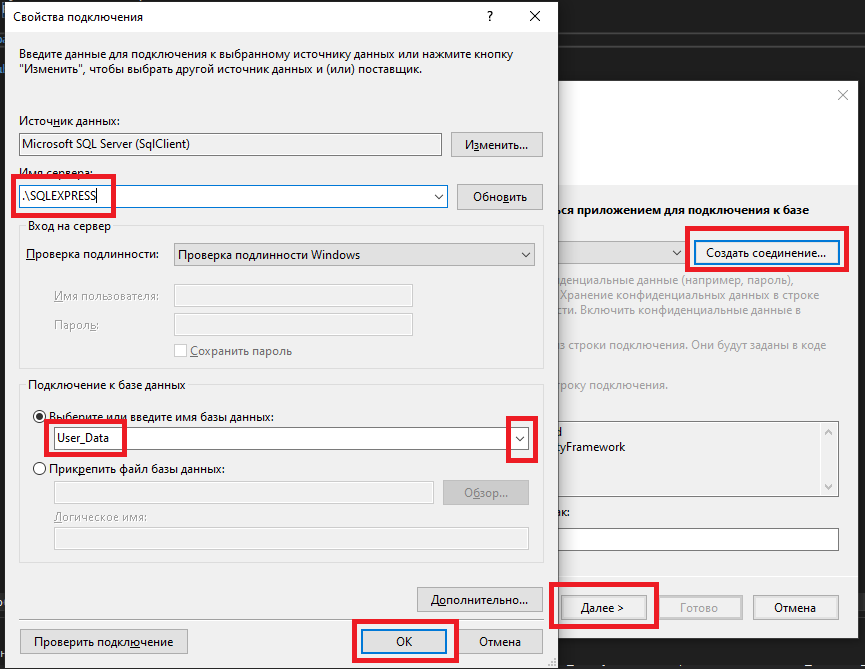
## 1) Создаем в MS SQL базу данных User\_Data, с помощью скрипта (User\_Data.sql)

## 2) Создаем в Visual Studio проект по шаблону «Приложение Windows Forms (.Net Framework)».

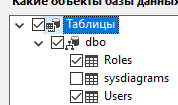
## 3) Добавляем в созданный проект модель EF

Для начала создадим новую папку для правильной архитектуры проекта. В новой папке создаём (добавляем) **Модель ADO .NET EMD**

Зададим имя модели Model1. Создадим соединение с БД User\_Data. И включим в модель таблицы Users, Roles.

Добавим в эту папку модель EF. Нажимаем правой кнопкой мыши на папку **«Добавить» 🡪»Создать Элемент» 🡪»Данные» 🡪 «Модель ADO .NET EMD» 🡪 «CodeFirst из базы данных» 🡪»Создать соединение» 🡪**

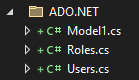
## В открывшемся окне в строку имя сервера ставим .\SQLEXPRESS А в строке базы данных выберите User\_Data 🡪 «OK»



**Далее выберем все пункты 🡪 «Готово»**

В результате к проекту добавятся семь файлов:

* класс контекста Model1.cs
* класс сущностей Roles.cs таблицы Mall.
* класс сущностей Users.cs таблицы Pavilion.

**Теперь нужно ОБЯЗАТЕЛЬНО выполнить хотя бы одну успешную сборку проекта!!! (Ctrl+Shift+B)**

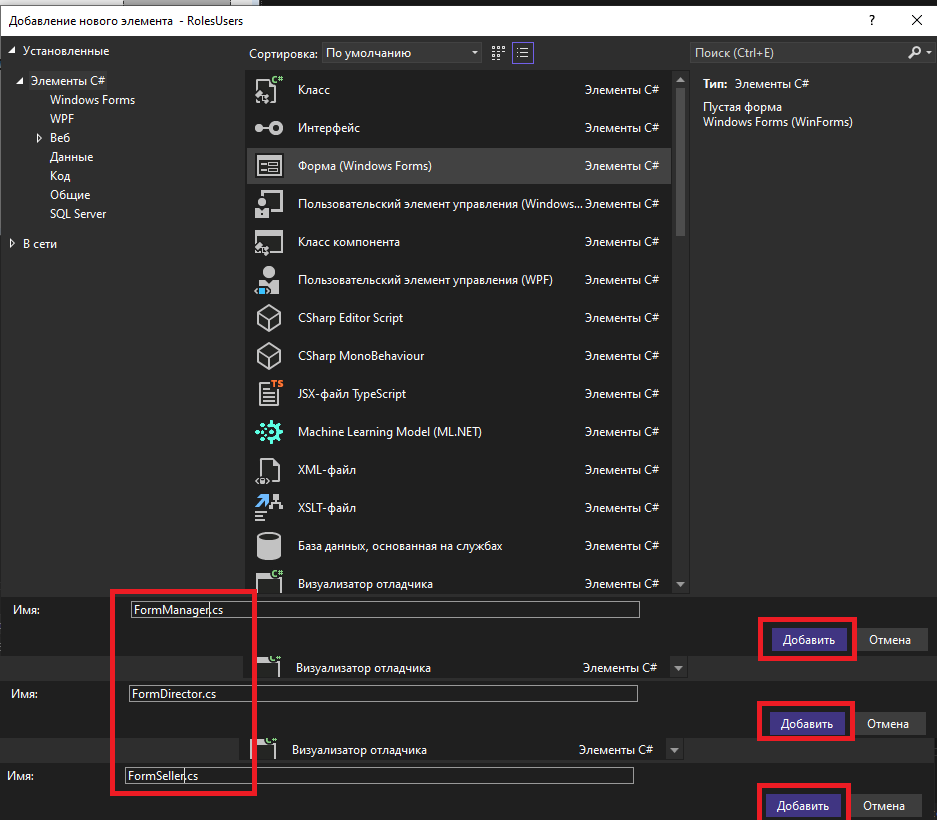
## 4) Добавляем новых форм

## Переименуйте стартовую Form1 в FormAutorization и добавьте дополнительно 3 новых формы с именами:

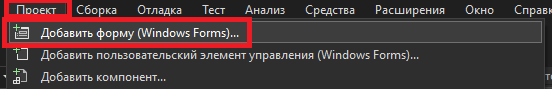
## FormDirector

## FormManager

## FormSeller



## Добавить новую форму можно через вкладку вкладку Проект 🡪 и нажимаем «Добавить форму».



## 5)Добавление элементов на формы

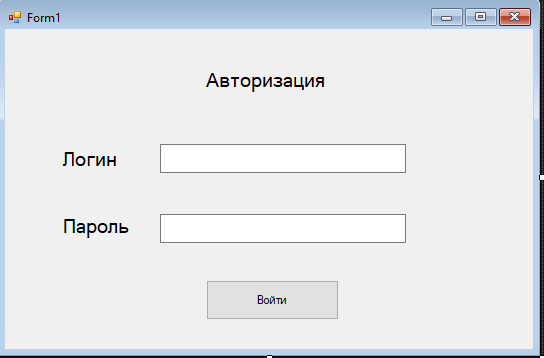
После чего добавим элементов на форму **FormAutorization**:

Два **TextBox – textBoxLogin, textBoxPassword**

Один **Button -buttonEnter**

Три **Label**

И соответственно поменять этим элементам свойство Name и Text как показано на картинке ниже.

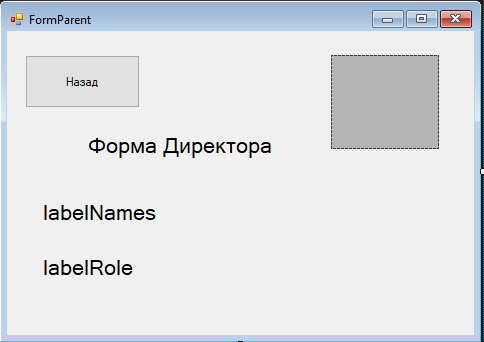


TextBox

Label

Button

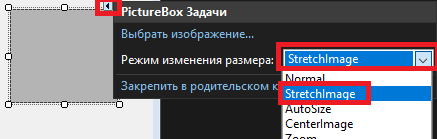
Дизайн остальных трёх форм (**FormDirector**, **FormManager**, **FormSeller**) могут выглядеть одинаково, смысл в самом переходе на разные формы в зависимости от роли пользователя. Главное условие чтобы менялась надпись на форме. Также можно создать основную форму родителя (Parent) и унаследовать от неё остальные. Пример можно увидеть ниже



Label

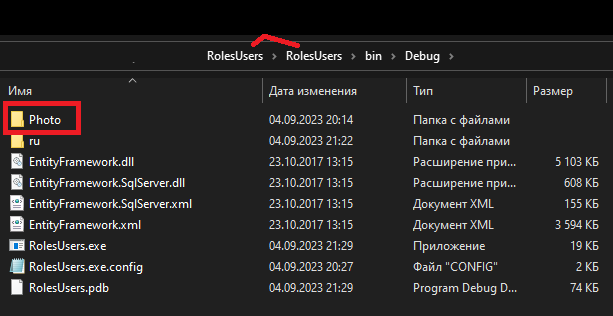
Button

PictureBox

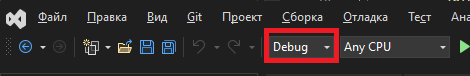
В **pictureBox1** следует изменить свойство Режим изменения размера (**SizeMode**) на **StretchImage,** как это можно сделать показано на картинке ниже.

## 6)Добавление папки Photo в папку проекта

Добавьте Photo к исполняемому файлу вашей программы. Либо в Debug либо в Release.

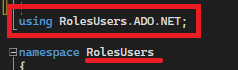
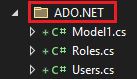


Название проекта



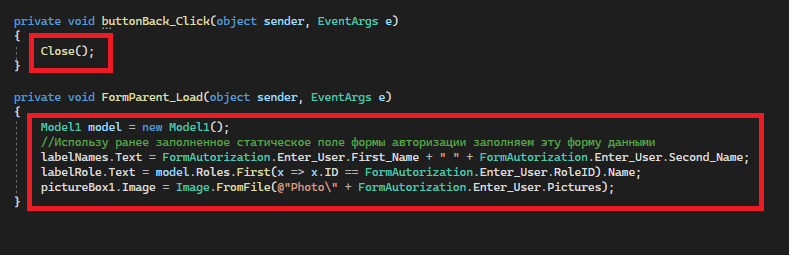
## 7) Написание кода программы

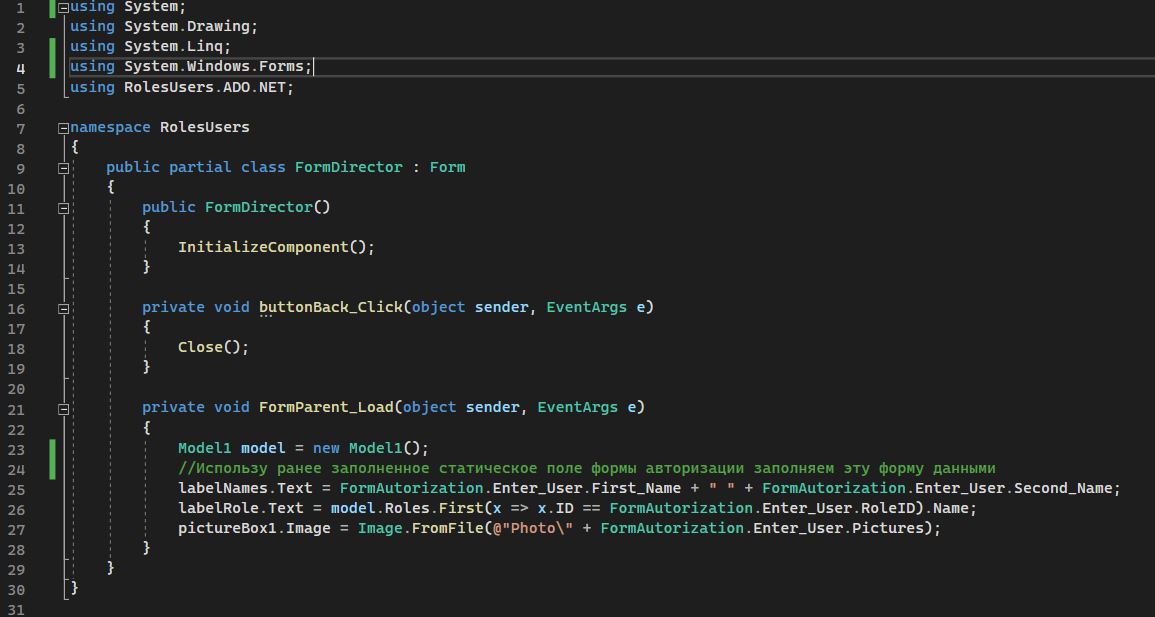
**Обязательно на всех формах добавляем библиотеку** !!!  
using ***Название\_Проекта.Название*\_*Папки\_с\_Моделью*;**



**Комментарии к коду переписывать не нужно!!!**

Ниже указан код FormAutorization. В пространстве класса необходимо создать статическую переменную класса Users (который мы ранее добавили через Code First), после добавить для кнопки «***Входа***» событие *Click*() и внутри написать код проверки правильности логина и пароля и перехода по формам в зависимости от роли.

Ниже показан код, который необходимо написать в формах (**FormDirector**, **FormManager**, **FormSeller**), как я уже говорил до того не обязательно всё копировать можно использовать наследование форм и написать код только на «Родительской» форме.

***Полный код классов форм*:**

