

The image features a dark blue background with a lighter blue curved line starting from the top left and curving towards the center. A bright blue spotlight effect is centered on the text, radiating outwards. The text 'WPF' is written in a bold, yellow, sans-serif font.

**WPF**



# Полезные ссылки

---

<https://metanit.com/sharp/wpf>

<https://nationalteam.worldskills.ru/skills/>

[sozdanie-karkasa-prilozheniya-sozdanie-i-ispolzovanie-stiley/](https://nationalteam.worldskills.ru/skills/sozdanie-karkasa-prilozheniya-sozdanie-i-ispolzovanie-stiley/)

# WPF (Windows Presentation

## Foundation)



---

Windows Presentation Foundation. Аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

WPF предоставляет средства **для создания визуального интерфейса**, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление. Производительность WPF выше за счёт использования аппаратного ускорения графики через DirectX.

# Использование разметки



## XAML

---

XAML представляет собой язык декларативного описания интерфейса, основанный на XML.

Также реализована **модель разделения кода и дизайна**, позволяющая кооперироваться программисту и дизайнеру.

Кроме того, есть встроенная поддержка стилей элементов, а сами элементы легко разделить на элементы управления второго уровня, которые, в свою очередь, разделяются до уровня векторных фигур и свойств/действий. Это позволяет легко задать стиль для любого элемента, например, Button (кнопка).



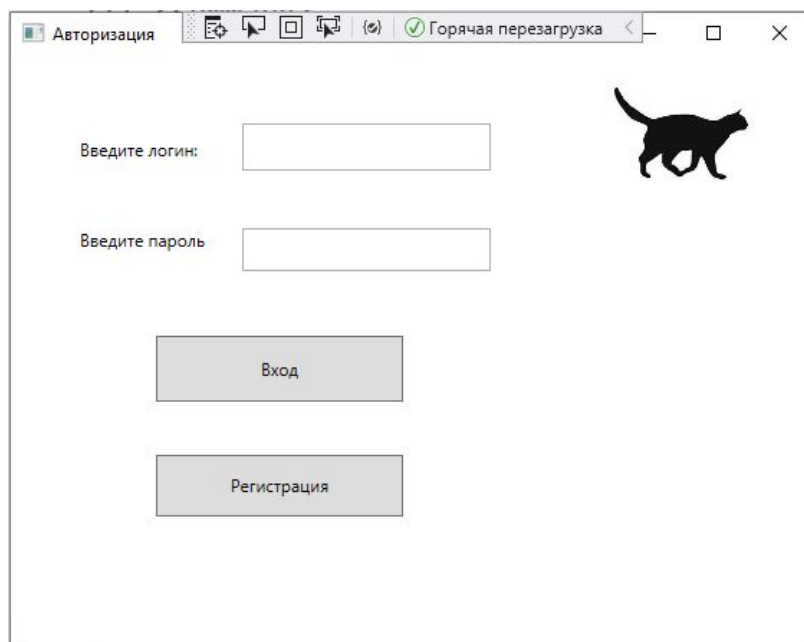
# EF (Entity Framework)

---

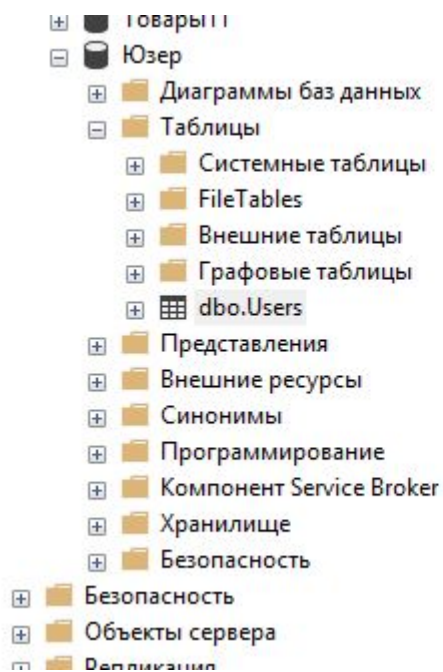
**Entity Framework позволяет работать с базой данных через объектно-ориентированный подход.**

Он предоставляет ряд существенных преимуществ: вам не нужно беспокоиться о коде доступа к данным, а также знать детали работы СУБД SQL Server и синтаксиса языка структурированных запросов SQL. Вместо этого вы работаете с **таблицами базы данных как с классами C#**, а с полями этих таблиц, как со свойствами классов, используя вместо SQL запросов более удобный подход — LINQ. Entity Framework берет на себя обязанности по преобразованию кода C# в SQL инструкции.

# Создаем простой проект



# Создаем таблицу в базе

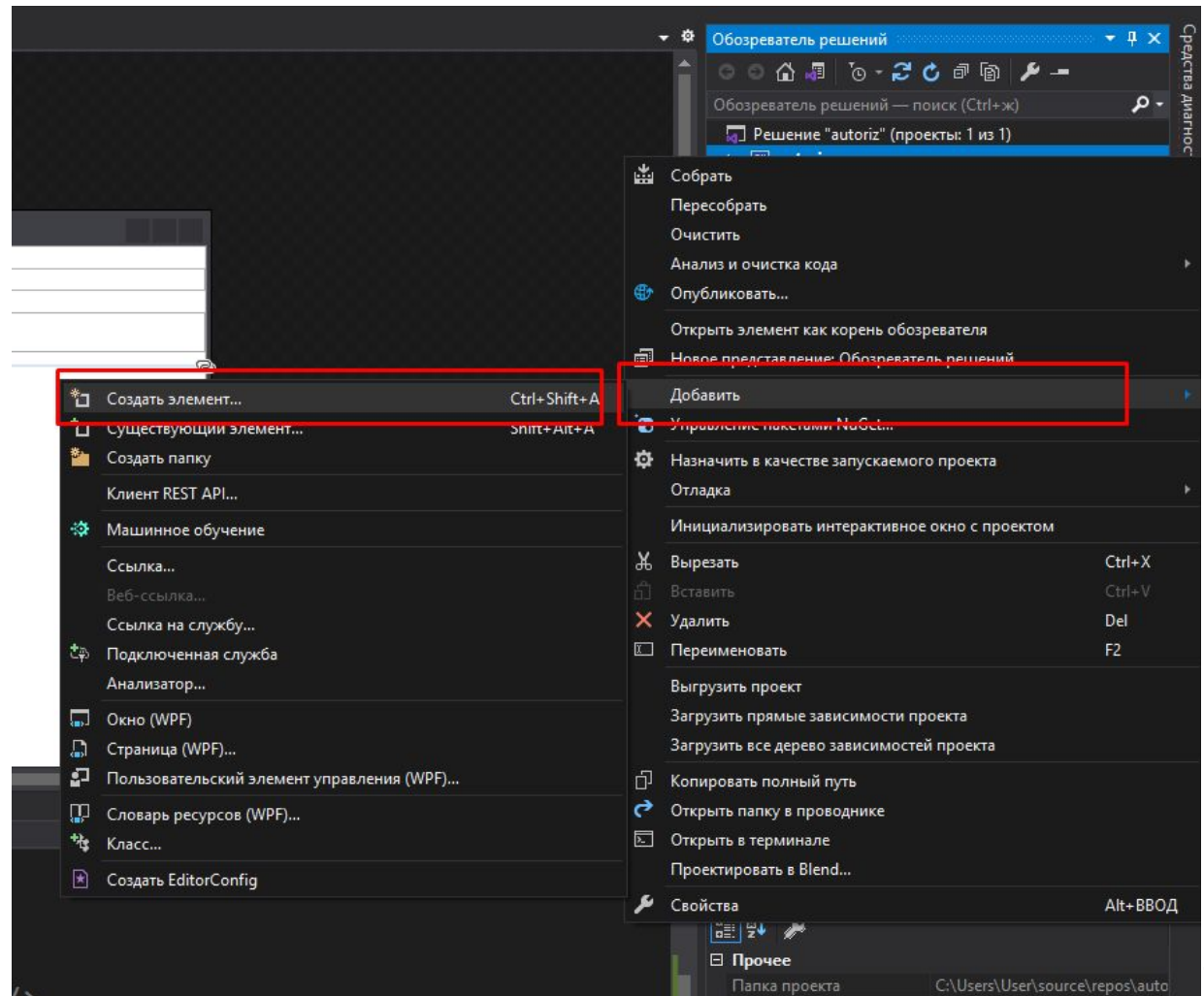


Скриншот окна результатов запроса в SQL Server. Таблица содержит 5 записей. В таблице выделена запись с id=1.

	id	login	password	role
1	1	mamon	12345	manager
2	2	papa	fdsterhsd	admin
3	3	rerec	dasdfr123	manager
4	4	guest	234234	klient
5	5	mixon	sdf34fd	admin

# Подключение фреймворка

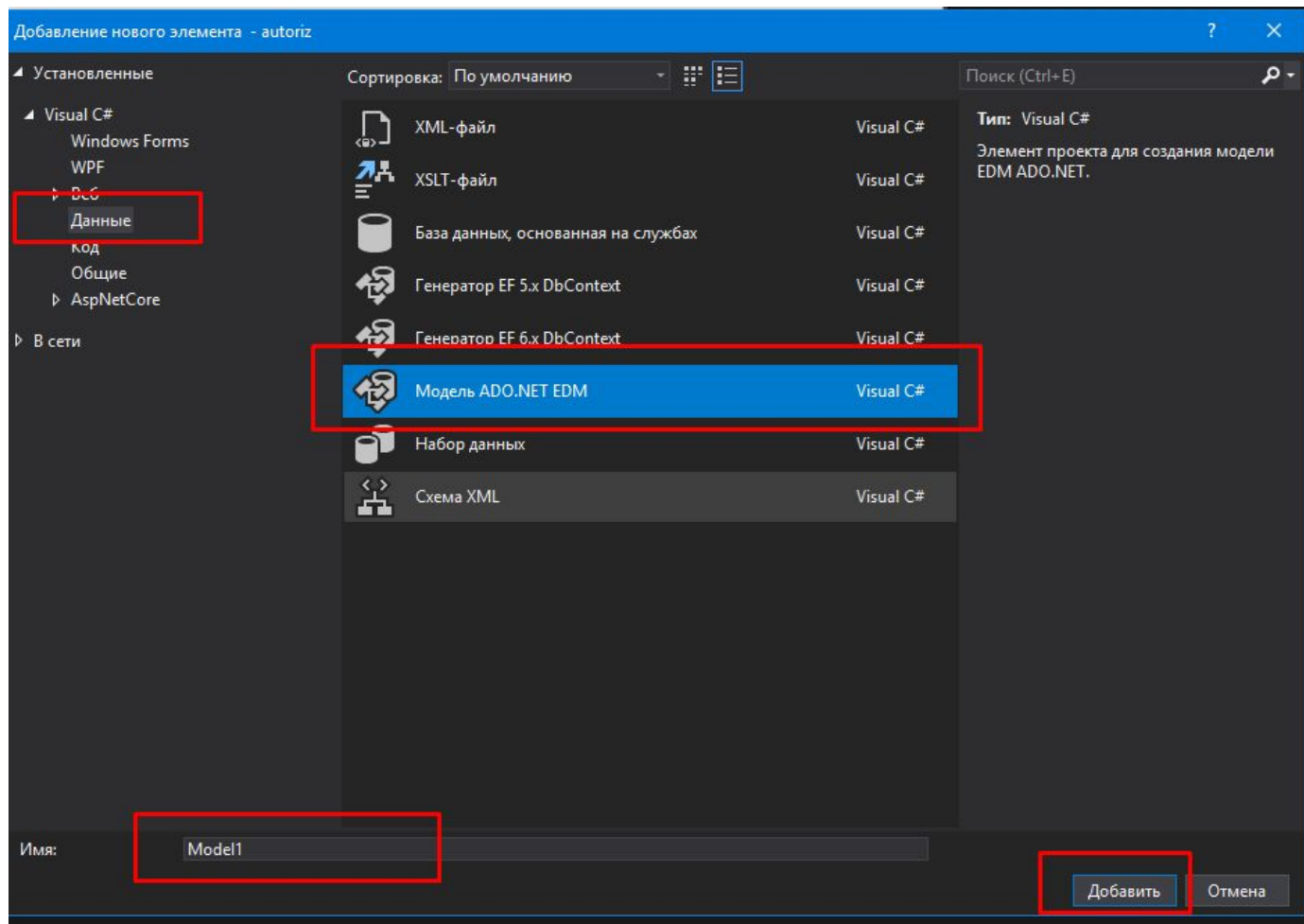
К проекту  
добавляем  
элемент



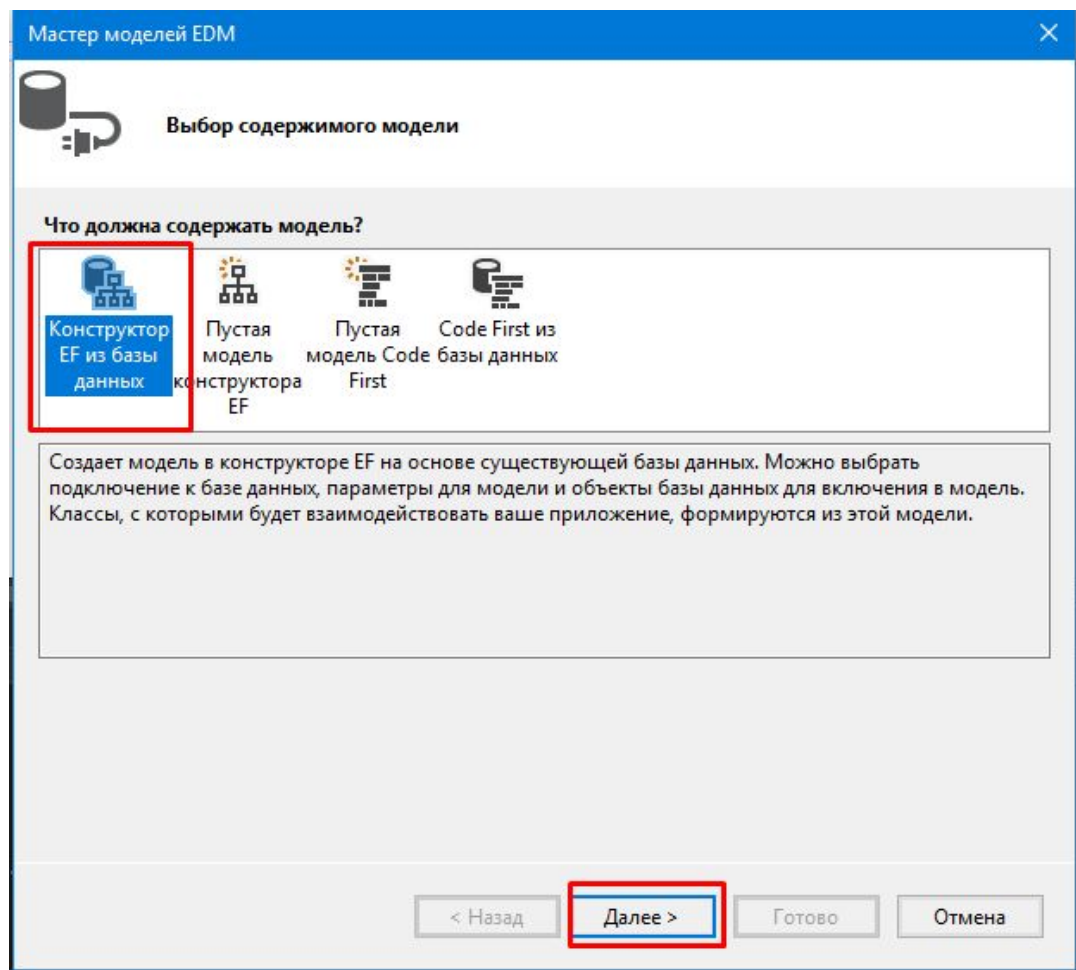


# Подключение фреймворка

Выбираем  
ADO.NET.  
Даем имя  
модели данных.

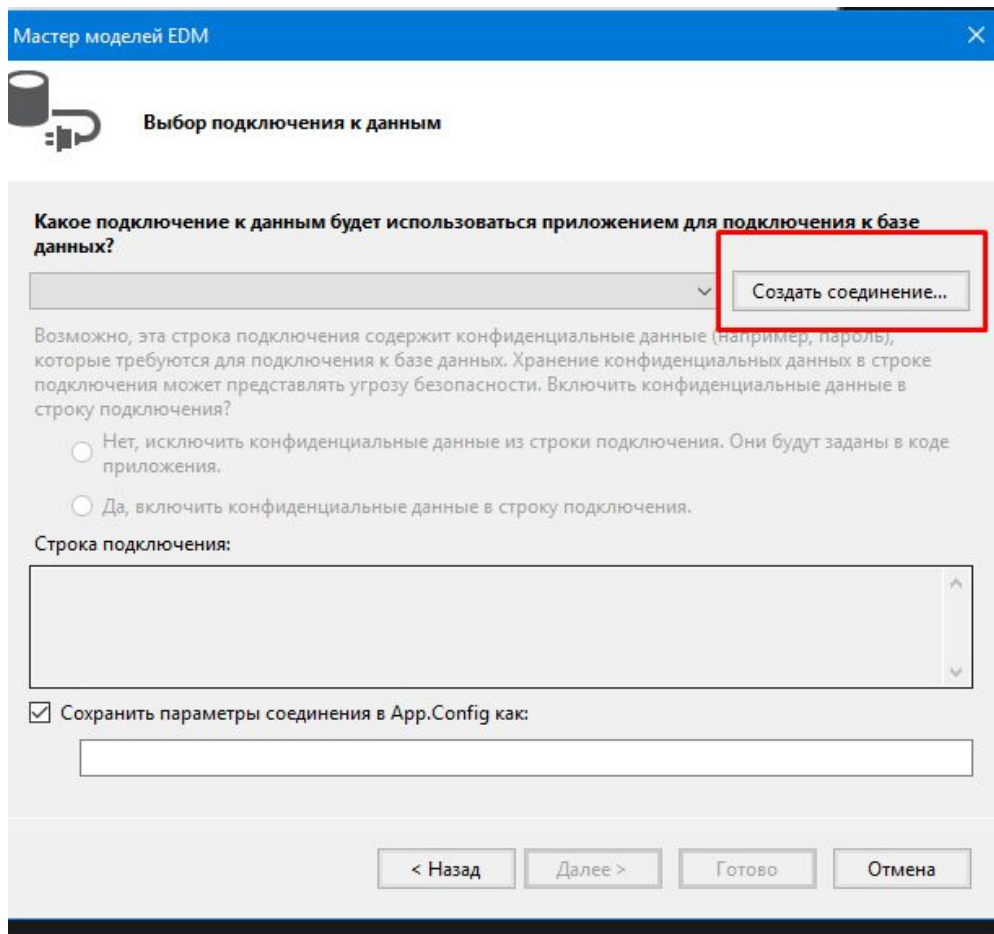


# Подключение фреймворка



# Подключение фреймворка

Создаем  
соединение к MS  
SQL Server



Мастер моделей EDM

Выбор подключения к данным

Какое подключение к данным будет использоваться приложением для подключения к базе данных?

Создать соединение...

Возможно, эта строка подключения содержит конфиденциальные данные (например, пароль), которые требуются для подключения к базе данных. Хранение конфиденциальных данных в строке подключения может представлять угрозу безопасности. Включить конфиденциальные данные в строку подключения?

Нет, исключить конфиденциальные данные из строки подключения. Они будут заданы в коде приложения.

Да, включить конфиденциальные данные в строку подключения.

Строка подключения:

Сохранить параметры соединения в App.Config как:

< Назад    Далее >    Готово    Отмена

# Подключение фреймворка

Свойства подключения

Введите данные для подключения к выбранному источнику данных или нажмите кнопку "Изменить", чтобы выбрать другой источник данных и (или) поставщик.

Источник данных: Microsoft SQL Server (SqlClient) Изменить...

Имя сервера: Обновить

Вход на сервер

Проверка подлинности: Проверка подлинности Windows

Имя пользователя:

Пароль:

Сохранить пароль

Подключение к базе данных

Выберите или введите имя базы данных:

Прикрепить файл базы данных:  Обзор...

Логическое имя:

Дополнительно...

Проверить подключение OK Отмена

Мастер моделей EDM

Выбор подключения к данным

Какое подключение к данным будет использоваться приложением для подключения к базе данных?

computer\sqlexpress.Юзеп.dbo Создать соединение...

Возможно, эта строка подключения содержит конфиденциальные данные (например, пароль), которые требуются для подключения к базе данных. Хранение конфиденциальных данных в строке подключения может представлять угрозу безопасности. Включить конфиденциальные данные в строку подключения?

Нет, исключить конфиденциальные данные из строки подключения. Они будут заданы в коде приложения.

Да, включить конфиденциальные данные в строку подключения.

Строка подключения:

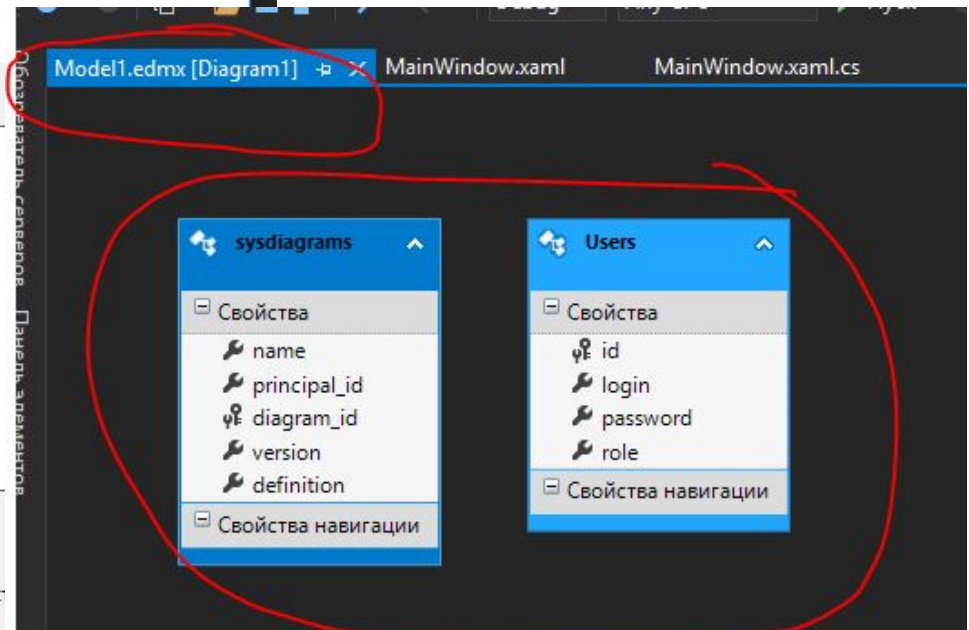
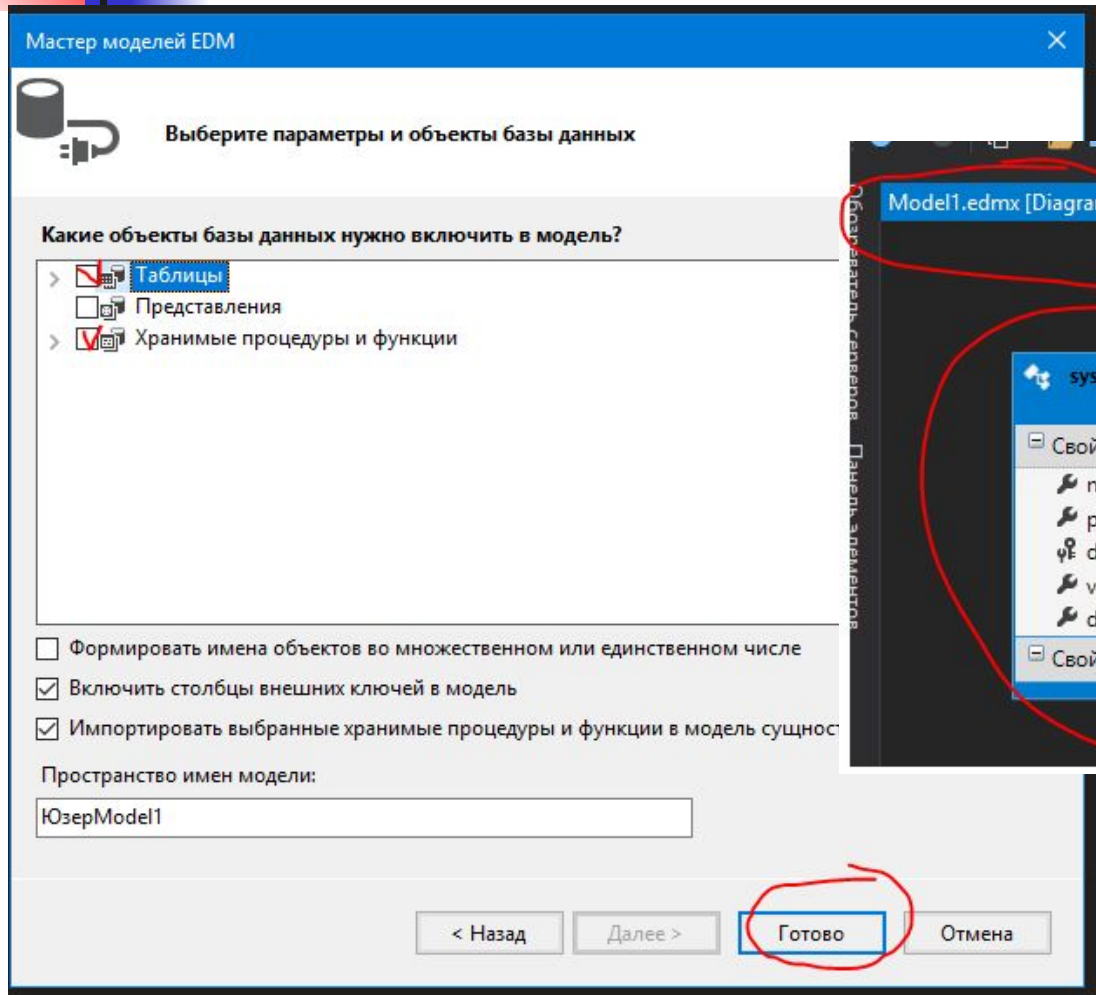
```
metadata=res://*/Model.csdl|res://*/Model.ssdl|res://*/Model.ms;provider=System.Data.SqlClient;provider connection string="data source=COMPUTER\SQLSERVER;initial catalog=Юзеп;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

Сохранить параметры соединения в App.Config как:

ЮзепEntities1

< Назад Далее > Готово Отмена

# Подключение фреймворка



# Создание объекта – БД – можно прямо в метод

В коде класса главного окна (или отдельной страницы авторизации) в метод обработки кнопки Вход добавим подключение:

```
using (var db = new ЮзерEntities())
```



# Код авторизации

---

в коде кнопки авторизации нужно:

- Проверить поля на пустоту
- Присвоить объекту результат запроса
- Если пользователь не найден, вывести сообщение

Иначе выведем его логин и роль



# Код авторизации

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }
    private void BtnIn_Click(object sender, RoutedEventArgs e)
    {
        if (string.IsNullOrEmpty(TextLogin.Text) || string.IsNullOrEmpty(passw.Password))
        {
            MessageBox.Show("Не введен логин или пароль");
            return;
        }
        using (var db = new ЮзерEntities())
        {
            var user = db.Users.AsNoTracking().FirstOrDefault(u => u.Логин == TextLogin.Text
                && u.Пароль == passw.Password);

            if (user == null)
            {
                MessageBox.Show("Таких данных не найдено");
                return;
            }
            else
            {
                textRes.Text = TextLogin.Text + " вошел как " + user.Роль;
            }
        }
    }
}
```



# Проверка

