

## Лабораторная работа № 6

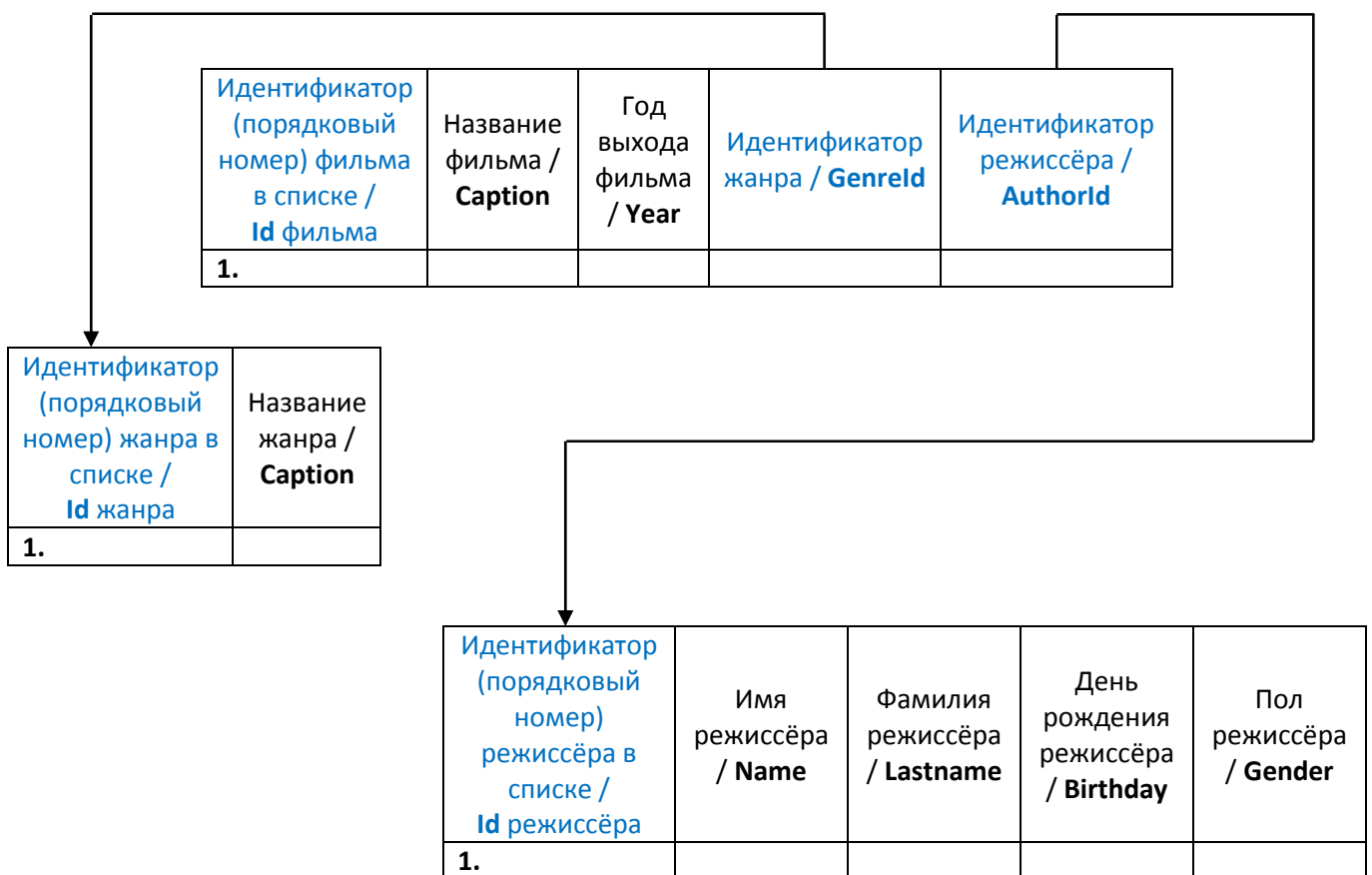
### «База данных»

**Задание:** необходимо создать приложение для работы с базой данных фильмов. Сама база данных содержит следующую информацию о фильме:

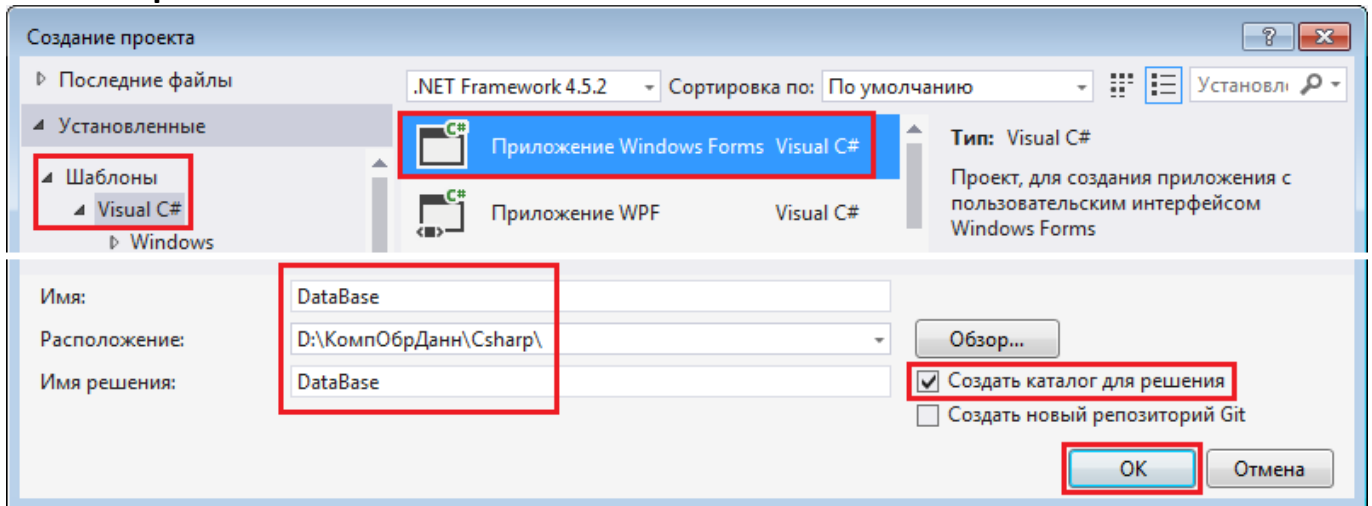
Название фильма / <b>Caption</b>	Год выхода фильма / <b>Year</b>	Жанр фильма / <b>Genre</b>	Имя режиссёра / <b>Name</b>	Фамилия режиссёра / <b>Lastname</b>	День рождения режиссёра / <b>Birthday</b>	Пол режиссёра / <b>Gender</b>

Данная база содержит избыточную информацию, т.к. жанры фильмов и данные о режиссёрах будут многократно повторяться, что ведёт к увеличению занимаемого базой данных места, и снижению скорости доступа к требуемым данным.

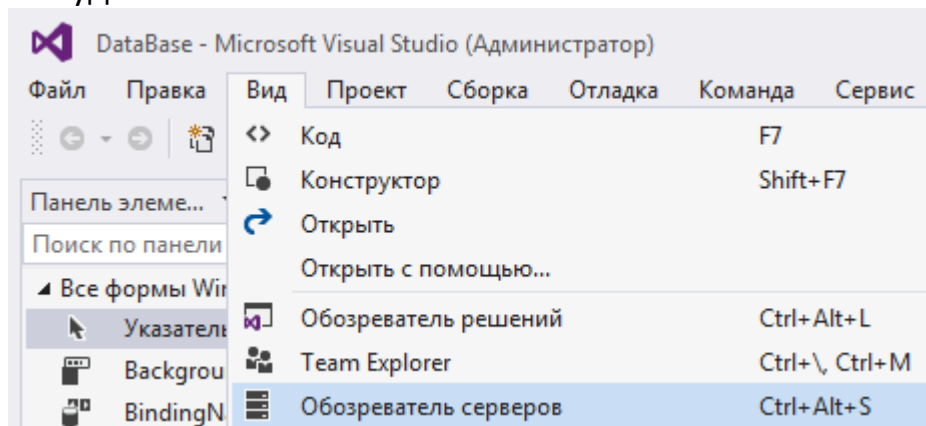
Поэтому предлагается вынести данные о жанрах и режиссёрах в две отдельные таблицы соответственно. Каждому жанру и режиссёру в пределах соответствующих таблиц базы данных присвоить уникальный идентификатор и связать таблицы между собой посредством этих идентификаторов.



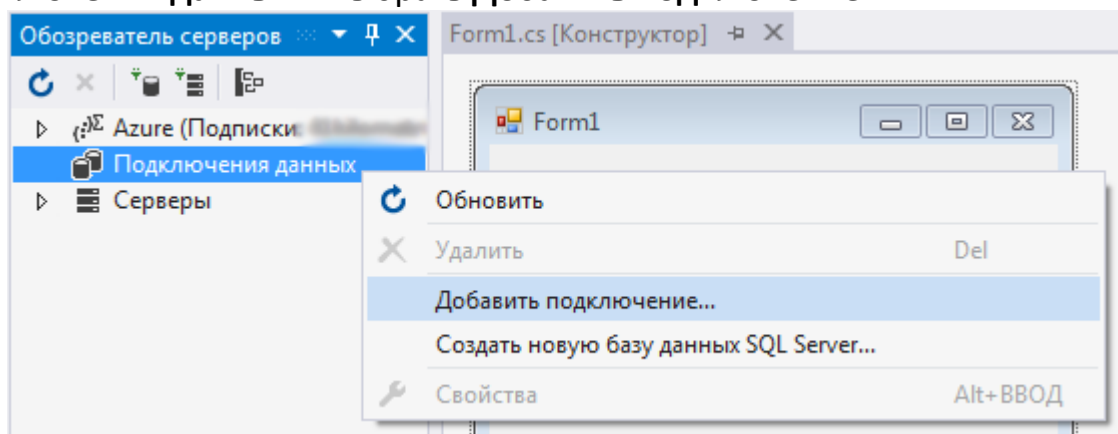
1. Создаём проект с именем **DataBase** в папке **d:\КомпОбрДанн\Csharp**. Шаблон – **Приложение Windows Forms**.



2. Перейти в меню **Вид – Обзорщик серверов**, и расположить появившееся окно в любом удобном месте.



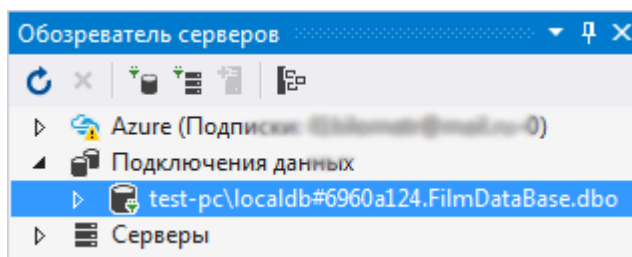
3. В окне **Обзорщика серверов** кликнуть правой кнопкой на строке **Подключения данных** и выбрать **Добавить подключение**.



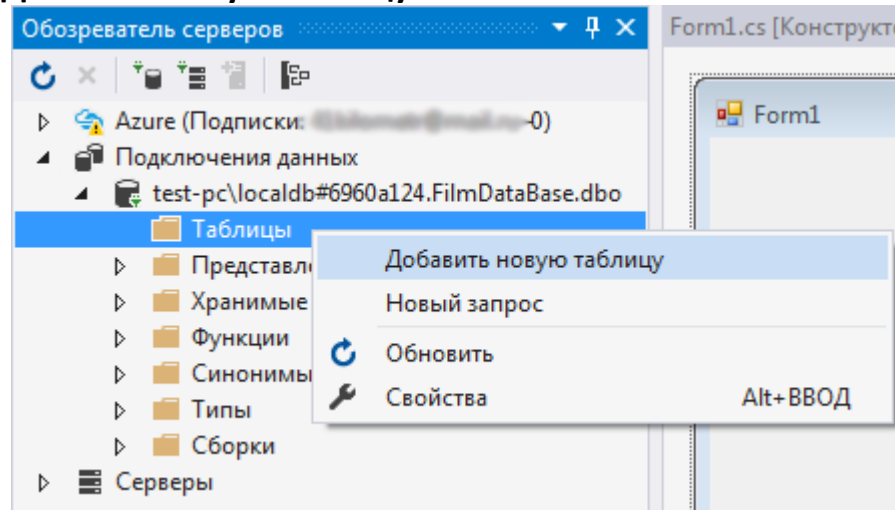
Далее будет производиться подключение к локальному экземпляру SQL-сервера с именем **(localdb)\v11.0**. Этот экземпляр SQL-сервера устанавливается вместе с Visual Studio 2013. Для более поздних версий Visual Studio узнать имя локального экземпляра SQL-сервера можно из командной строки с помощью команды **cmd> SqlLocalDB.exe info**. Результатом команды может быть имя локального экземпляра SQL-сервера **MSSQLLocalDB**.

4. В появившемся окне в строке **Имя сервера** ввести **(localdb)\v11.0** . В строке **Выберите или введите имя базы данных** ввести **FilmDataBase**. Нажать **ОК**.

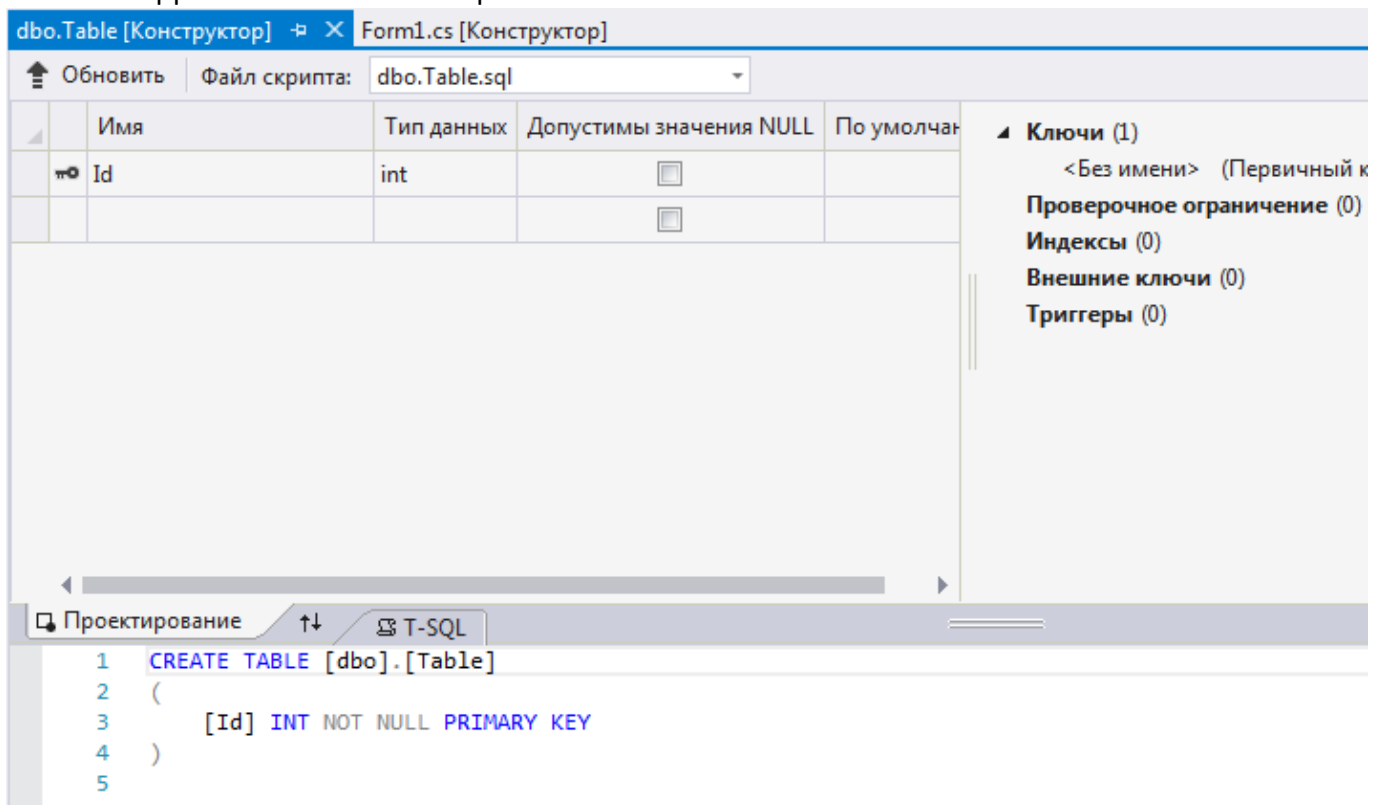
5. Появится окно, в котором написано что такая база данных не существует и хотим ли мы её создать? Согласиться и создать новую локальную базу данных. Она появится в **Обзревателе серверов** в виде **Имя компьютера\Имя базы данных**.



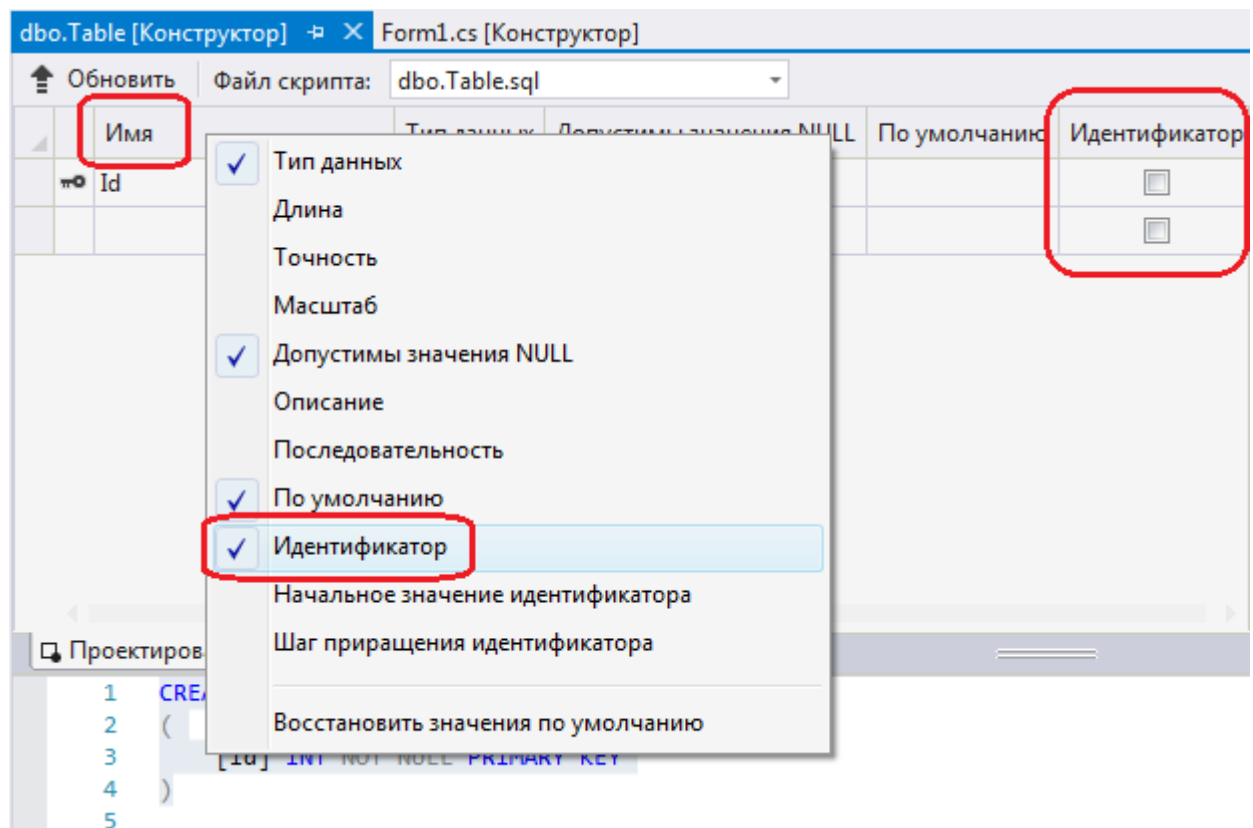
6. Раскрыв созданную базу данных, кликнуть правой кнопкой мыши на папке **Таблицы** – **Добавить новую таблицу**.



7. Создастся новая таблица



8. Кликнув правой кнопкой мыши на **Имя** столбцов, в появившемся меню установить галочку на строке **Идентификатор**. Появится дополнительный столбец с аналогичным именем.



9. Заполнить структуру новой таблицы. Она будет состоять из двух столбцов с именами **Id** и **Caption**.

У столбца **Id**:

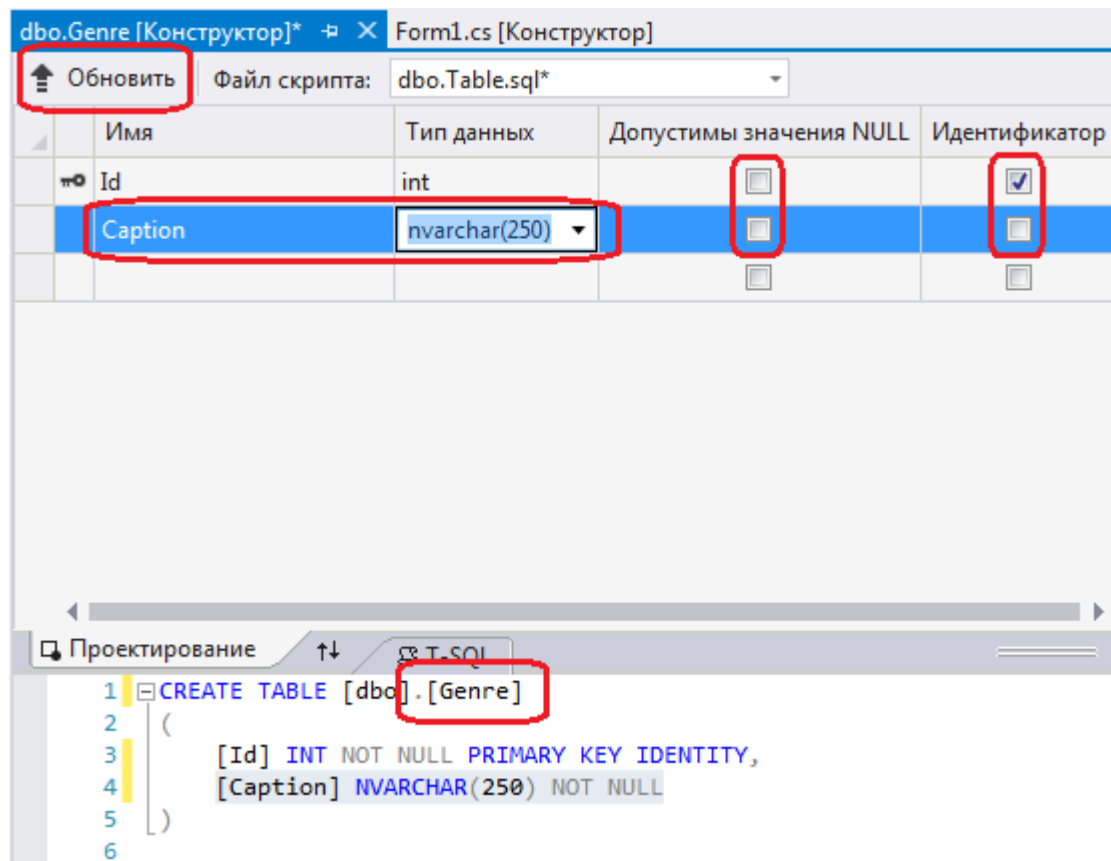
- 1) **тип данных** оставляем по умолчанию **int** (целочисленные значения)
- 2) **допустимые значения NULL** – **убираем** галочку (т.е. любая ячейка столбца не может быть пустой)
- 3) **идентификатор** – **ставим** галочку (т.е. значение в каждой новой строке данного столбца будет автоматически увеличиваться на 1 при добавлении новой записи)

У столбца **Caption**:

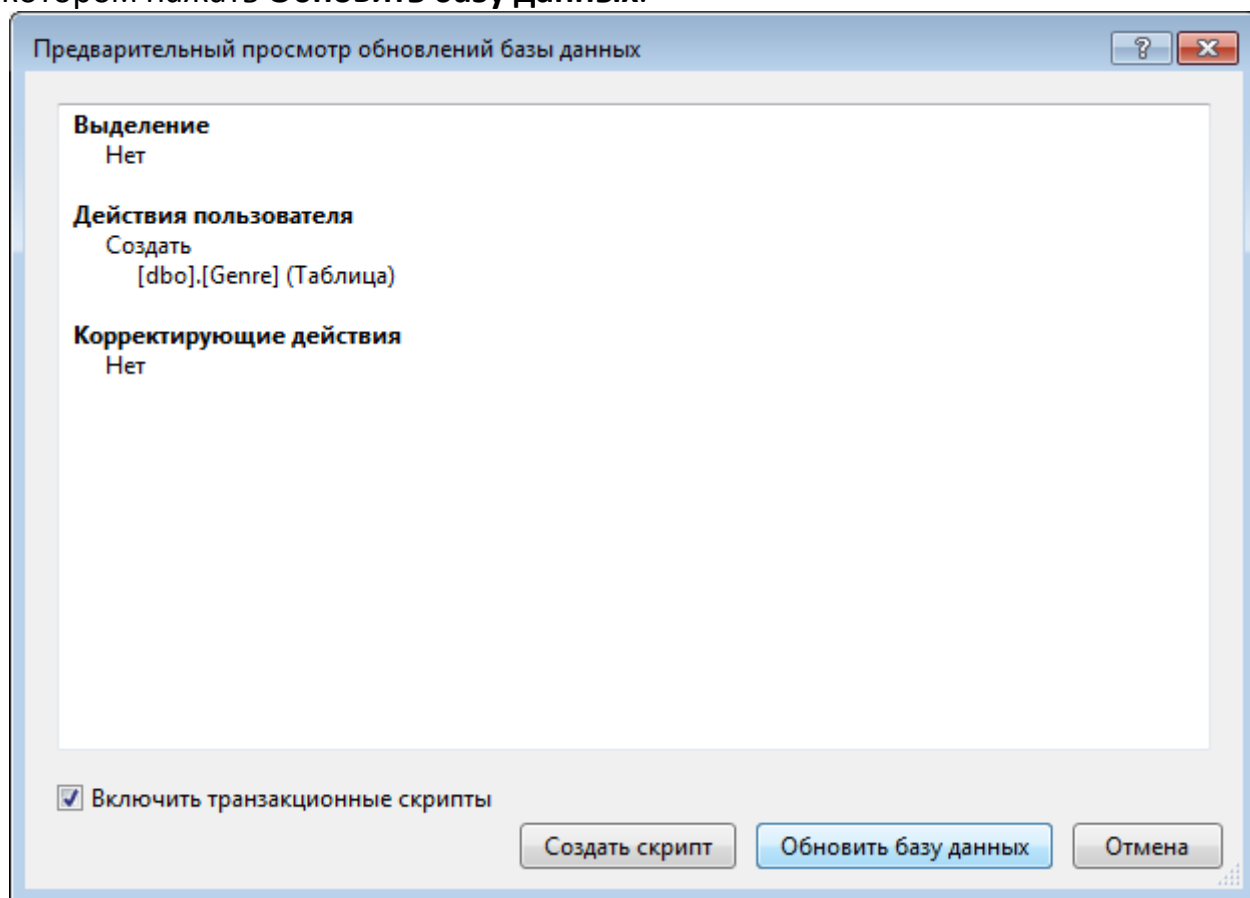
- 1) **тип данных** – выбираем **nvarchar(50)** и меняем на **nvarchar(250)** (nvarchar – строковые данные переменной длины в Юникоде)
- 2) **допустимые значения NULL** – **убираем** галочку (т.е. любая ячейка столбца не может быть пустой)
- 3) **идентификатор** – **убираем** галочку

Под таблицей находится так называемый SQL-запрос на создание таблицы. Меняем **[Table]** на **[Genre]**, так чтобы первая строка была **CREATE TABLE [dbo].[Genre]**

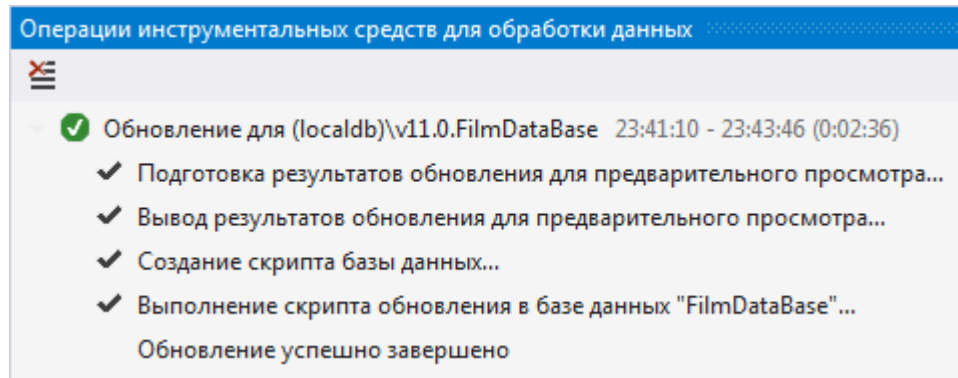
В завершение нажимаем кнопку **Обновить** в левом верхнем углу **Конструктора** таблицы.



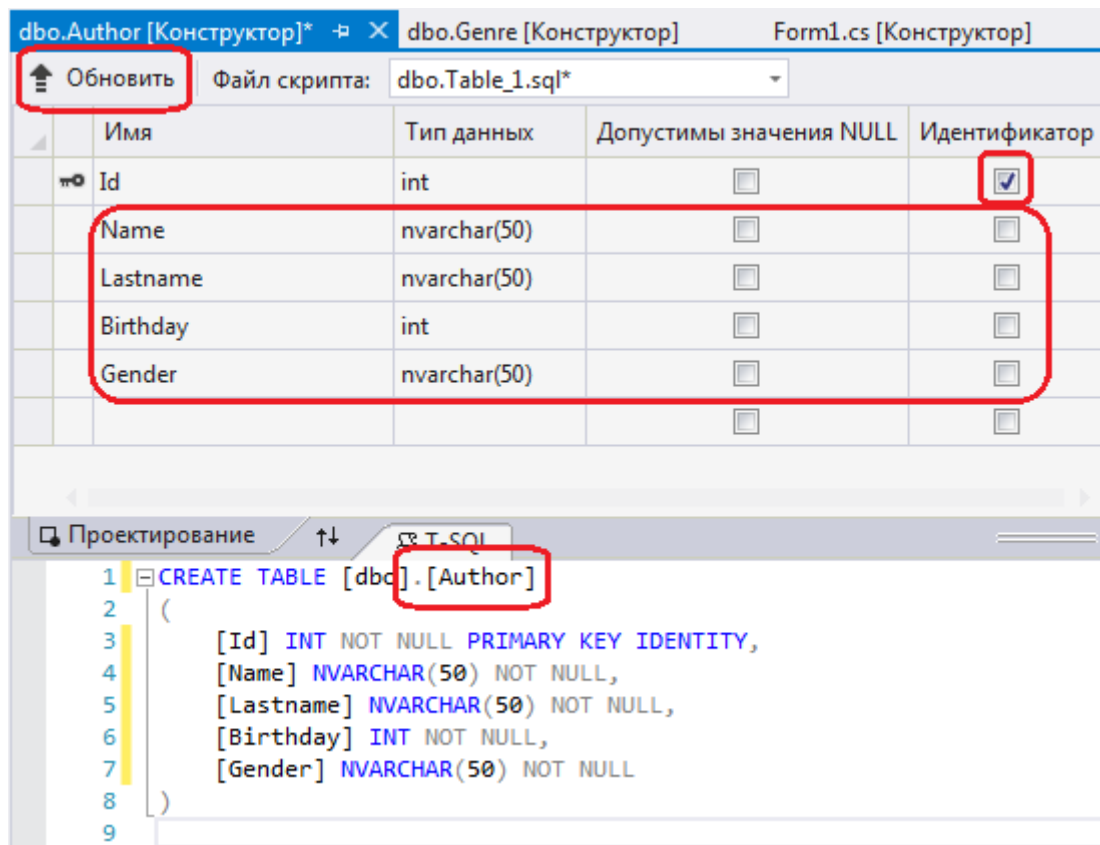
10. Появится окно предварительного просмотра обновлений базы данных, в котором нажать **Обновить базу данных**.



## 11. Дождаться успешного обновления базы данных



12. Перейти в **Обозреватель серверов**, и в папке **Таблицы** добавить новую (вторую) таблицу. Она будет состоять из пяти столбцов с именами **Id** (тип **int**), **Name** (тип **nvarchar(50)**), **Lastname** (тип **nvarchar(50)**), **Birthday** (тип **int**), **Gender** (тип **nvarchar(50)**). У всех столбцов убираем галочки **Допустимые значения NULL**. У всех столбцов, кроме **Id**, убираем галочки **Идентификатор**. Под таблицей в SQL-запросе меняем **[Table]** на **[Author]**, так чтобы первая строка была **CREATE TABLE [dbo].[Author]**
- В завершение нажимаем кнопку **Обновить** в левом верхнем углу **Конструктора** таблицы. Подтвердить обновление базы данных и дождаться операции создания новой таблицы.



13. Перейти в **Обозреватель серверов**, и в папке **Таблицы** добавить новую (третью) таблицу. Она будет состоять из пяти столбцов с именами **Id** (тип **int**), **Caption** (тип **nvarchar(150)**), **Year** (тип **int**), **GenreId** (тип **int**), **AuthorId** (тип **int**). У всех столбцов убираем галочки **Допустимые значения NULL**. У всех столбцов, кроме **Id**, убираем галочки **Идентификатор**.

Под таблицей в SQL-запросе меняем **[Table]** на **[Film]**, так чтобы первая строка была **CREATE TABLE [dbo].[Film]**

The screenshot shows the 'dbo.Film [Конструктор]\*' window. The table structure is as follows:

Имя	Тип данных	Допустимы значения NULL	Идентификатор
Id	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Caption	nvarchar(150)	<input type="checkbox"/>	<input type="checkbox"/>
Year	int	<input type="checkbox"/>	<input type="checkbox"/>
GenreId	int	<input type="checkbox"/>	<input type="checkbox"/>
AuthorId	int	<input type="checkbox"/>	<input type="checkbox"/>

On the right, the 'Ключи (1)' pane shows: <Без имени> (Первичный), Проверочное ограничение, Индексы (0), **Внешние ключи (0)** (highlighted with a red box), and Триггеры (0).

At the bottom, the T-SQL script is shown:

```

1 CREATE TABLE [dbo].[Film]
2 (
3     [Id] INT NOT NULL PRIMARY KEY IDENTITY,
4     [Caption] NVARCHAR(150) NOT NULL,
5     [Year] INT NOT NULL,
6     [GenreId] INT NOT NULL,
7     [AuthorId] INT NOT NULL
8 )
  
```

В поле **Ключи** справа от структуры таблицы кликаем правой кнопкой мыши на поле **Внешние ключи** – **Добавить новый внешний ключ**.

The screenshot shows the 'Ключи (1)' pane with the 'Внешние ключи (0)' item selected. A right-click context menu is open with the following options:

- Добавить новый внешний ключ
- Переключиться на панель T-SQL



И меняем имя внешнего ключа с **FK\_Film\_ToTable** на **FK\_Film\_ToGenre**.

Внешние ключи (1)  
FK\_Film\_ToGenre (ToTableColumn)  
Триггеры (0)

Проектирование T-SQL

```
1 CREATE TABLE [dbo].[Film]
2 (
3     [Id] INT NOT NULL PRIMARY KEY IDENTITY,
4     [Caption] NVARCHAR(150) NOT NULL,
5     [Year] INT NOT NULL,
6     [GenreId] INT NOT NULL,
7     [AuthorId] INT NOT NULL,
8     CONSTRAINT [FK_Film_ToGenre] FOREIGN KEY ([Column]) REFERENCES [ToTable]([ToTableColumn])
9 )
10
```

В SQL-запросе под структурой создаваемой таблицы автоматически добавится строка **CONSTRAINT [FK\_Film\_ToGenre] FOREIGN KEY ([Column]) REFERENCES [ToTable]([ToTableColumn])**, в которой заменить **[Column]** на **[GenreId]**, **[ToTable]** на **[Genre]**, **[ToTableColumn]** на **[Id]**.

Аналогично добавить второй внешний ключ, который назвать **FK\_Film\_ToAuthor**. В SQL-запросе в новой автоматически созданной строке **CONSTRAINT [FK\_Film\_ToAuthor] FOREIGN KEY ([Column]) REFERENCES [ToTable]([ToTableColumn])** заменить **[Column]** на **[AuthorId]**, **[ToTable]** на **[Author]**, **[ToTableColumn]** на **[Id]**.

В завершение нажимаем кнопку **Обновить** в левом верхнем углу **Конструктора** таблицы. Подтвердить обновление базы данных и дождаться операции создания новой таблицы.

dbo.Film [Конструктор]\* | dbo.Author [Конструктор] | dbo.Genre [Конструктор] | Form1.cs [Конструктор]

Обновить | Файл скрипта: dbo.Table\_2.sql\*

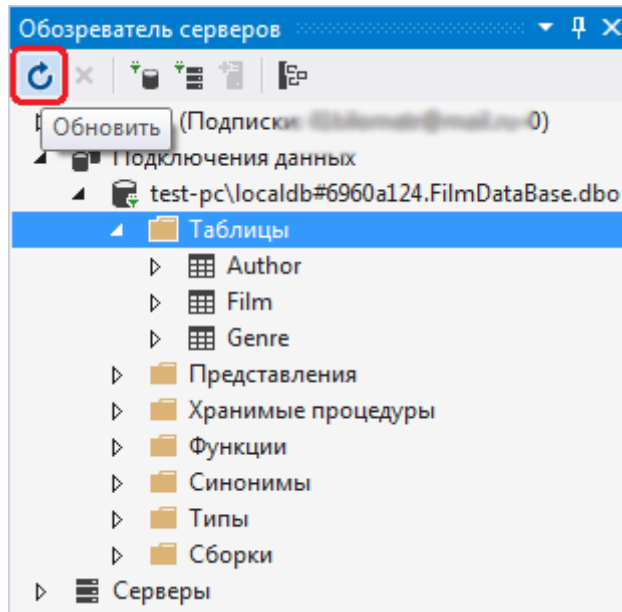
Имя	Тип данных	Допустимы значения NULL	Идентификатор
Id	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Caption	nvarchar(150)	<input type="checkbox"/>	<input type="checkbox"/>
Year	int	<input type="checkbox"/>	<input type="checkbox"/>
GenreId	int	<input type="checkbox"/>	<input type="checkbox"/>
AuthorId	int	<input type="checkbox"/>	<input type="checkbox"/>

Ключи (1)  
<Без имени> (Первич)  
Проверочное ограничени  
Индексы (0)  
Внешние ключи (2)  
FK\_Film\_ToGenre (Id)  
FK\_Film\_ToAuthor (Id)  
Триггеры (0)

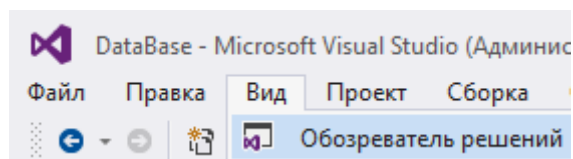
Проектирование T-SQL

```
1 CREATE TABLE [dbo].[Film]
2 (
3     [Id] INT NOT NULL PRIMARY KEY IDENTITY,
4     [Caption] NVARCHAR(150) NOT NULL,
5     [Year] INT NOT NULL,
6     [GenreId] INT NOT NULL,
7     [AuthorId] INT NOT NULL,
8     CONSTRAINT [FK_Film_ToGenre] FOREIGN KEY ([GenreId]) REFERENCES [Genre]([Id]),
9     CONSTRAINT [FK_Film_ToAuthor] FOREIGN KEY ([AuthorId]) REFERENCES [Author]([Id])
10 )
```

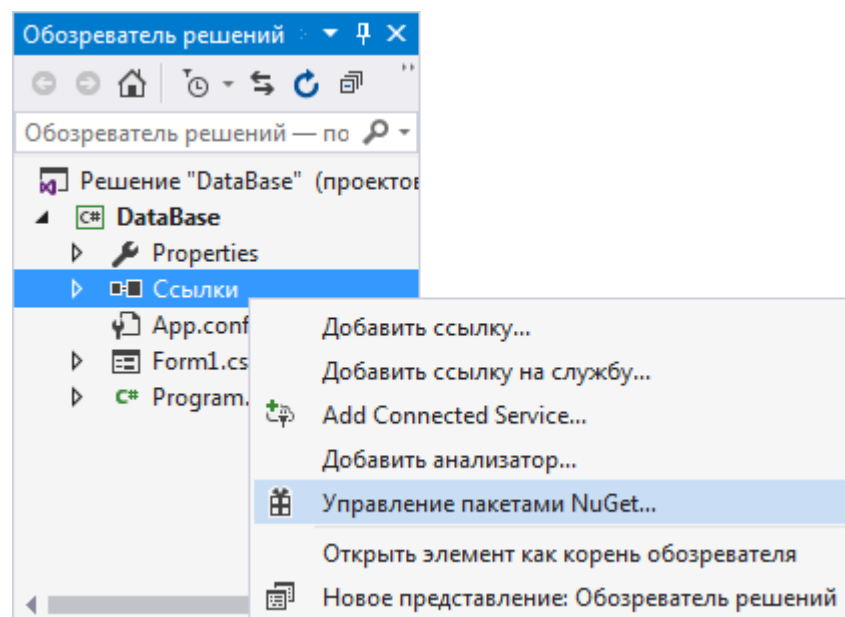
14. В **Обозревателе серверов** нажать на кнопку **Обновить**, чтобы таблицы стали видны.



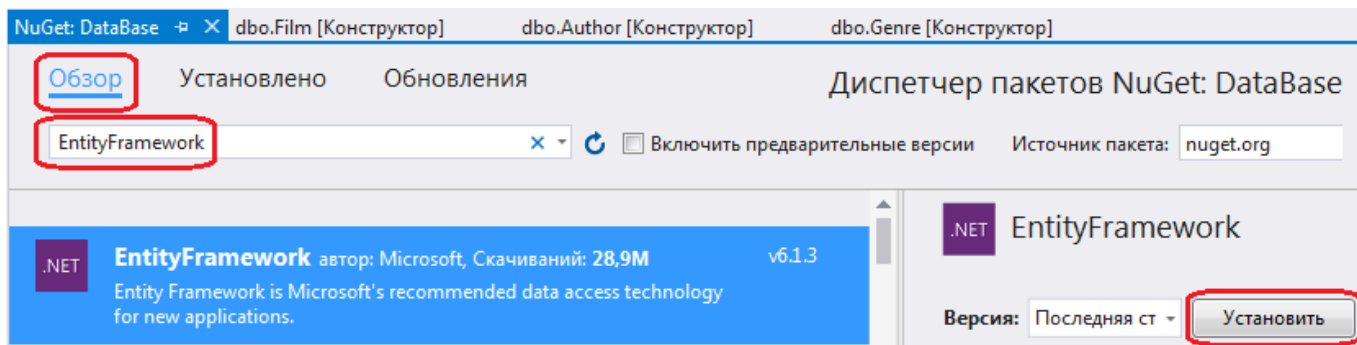
15. Перейти в **Обозреватель решений** проекта (в главном меню **Вид – Обозреватель решений**).



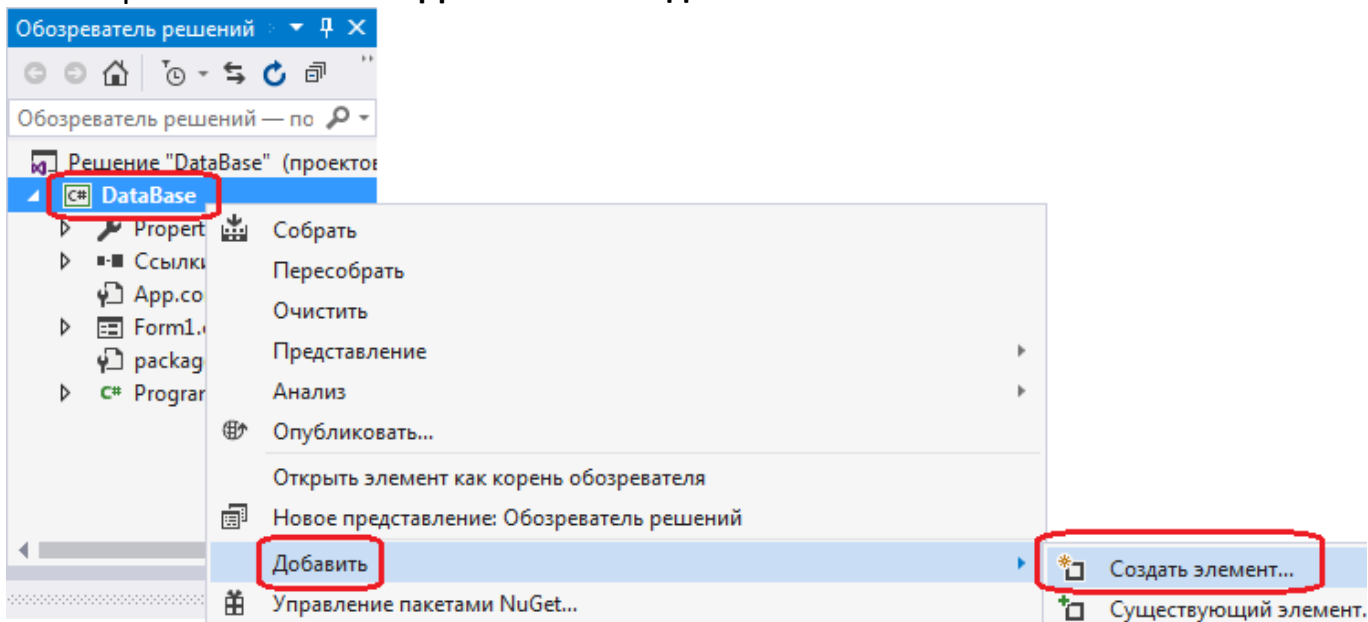
В структуре проекта кликнуть правой кнопкой мыши на пункт **Ссылки**, и выбрать **Управление пакетами NuGet**.



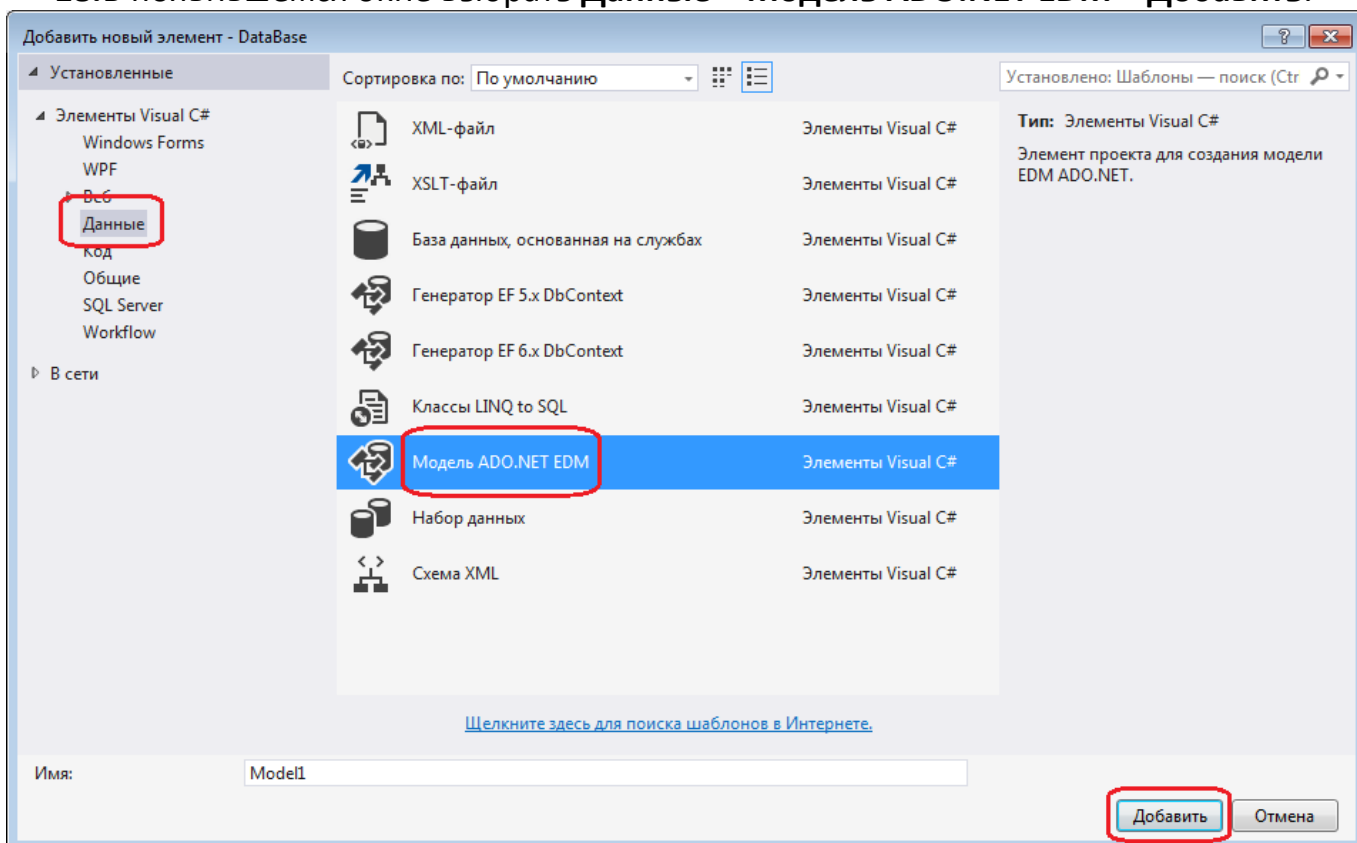
16. В появившейся вкладке **NuGet: DataBase** перейти в раздел **Обзор**, в строке поиска ввести **EntityFramework**, и из списка доступных пакетов установить пакет **EntityFramework** (если он не установлен).



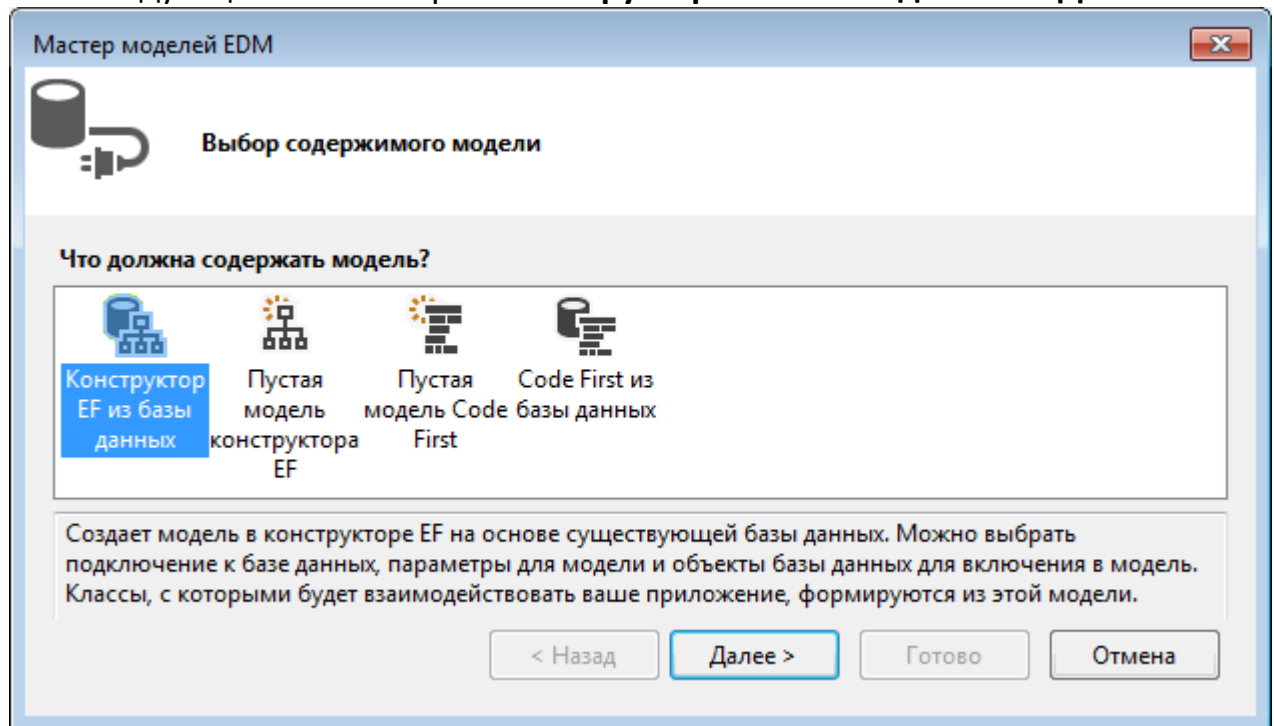
17. Перейти в **Обозреватель решений**, кликнуть правой кнопкой мыши на имени проекта **DataBase** – **Добавить** – **Создать элемент**.



18. В появившемся окне выбрать **Данные** – **Модель ADO.NET EDM** – **Добавить**.



## 19. В следующем окне выбрать **Конструктор EF из базы данных** – **Далее**



Мастер моделей EDM

Выбор содержимого модели

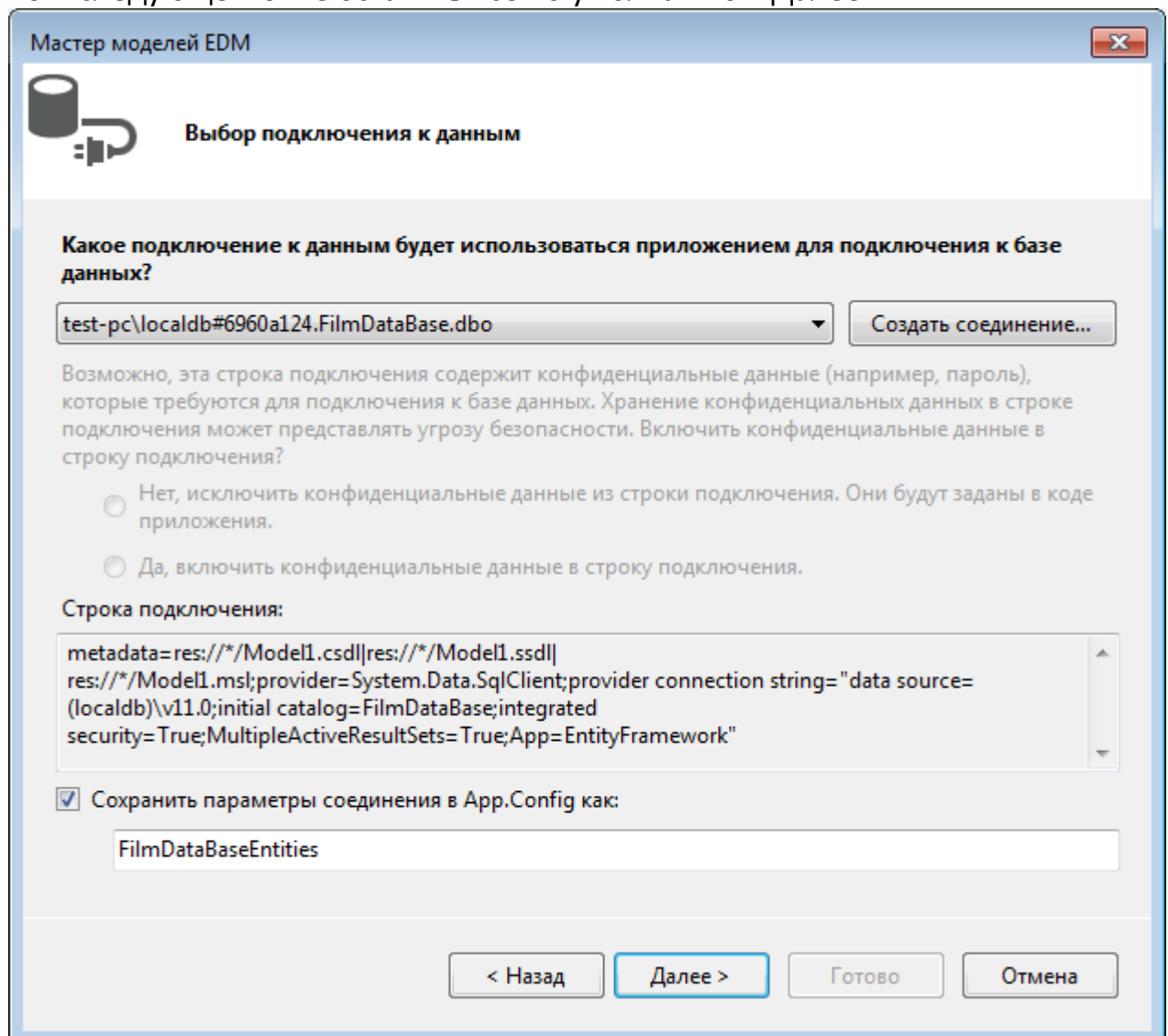
Что должна содержать модель?

Конструктор EF из базы данных | Пустая модель конструктора EF | Пустая модель Code First | Code First из базы данных

Создает модель в конструкторе EF на основе существующей базы данных. Можно выбрать подключение к базе данных, параметры для модели и объекты базы данных для включения в модель. Классы, с которыми будет взаимодействовать ваше приложение, формируются из этой модели.

< Назад | **Далее >** | Готово | Отмена

## 20. В следующем окне оставить всё по умолчанию – **Далее**



Мастер моделей EDM

Выбор подключения к данным

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

test-pc\localdb#6960a124.FilmDataBase.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:

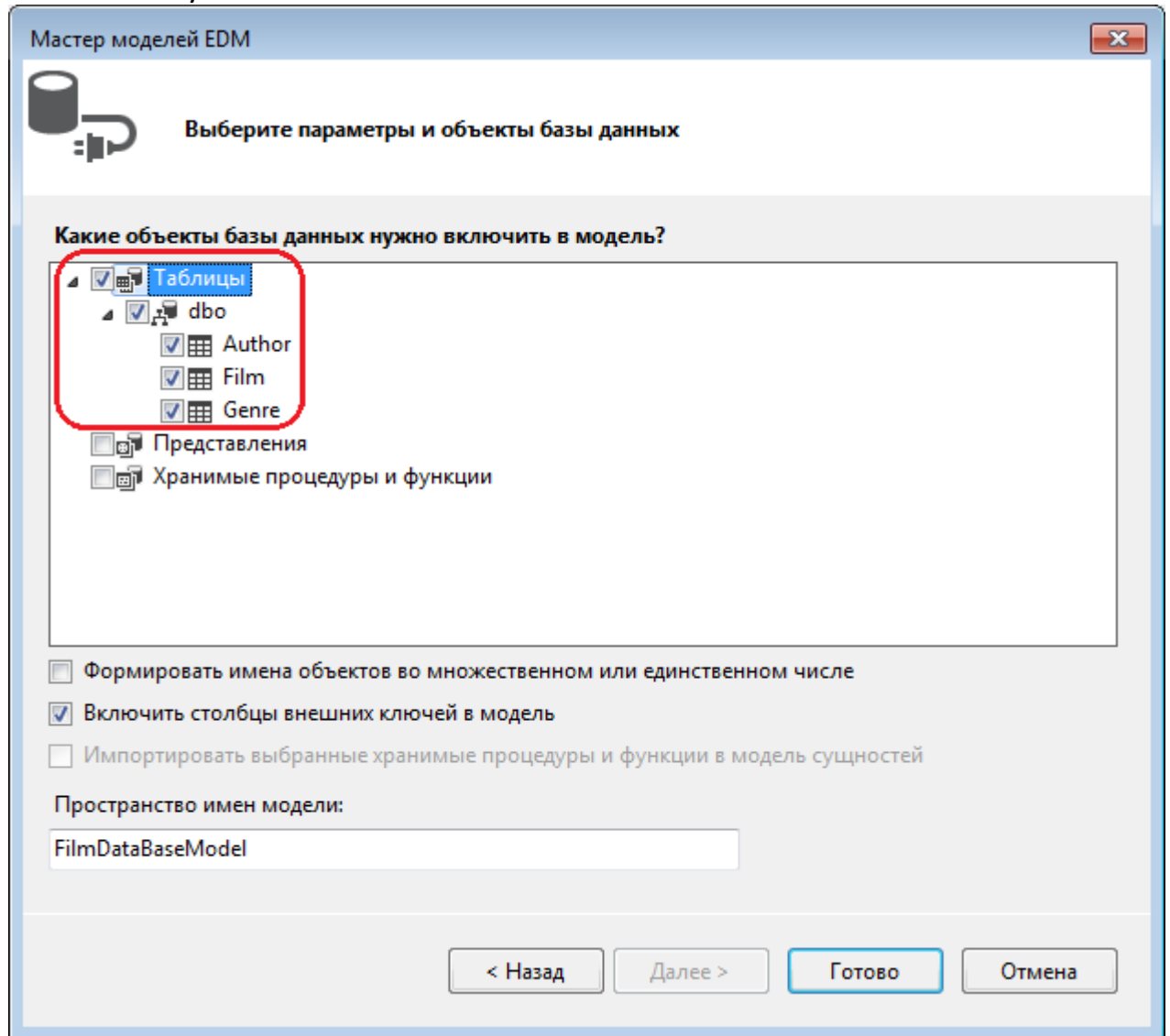
metadata=res://\*/Model1.csdl|res://\*/Model1.ssdl|res://\*/Model1.msl;provider=System.Data.SqlClient;provider connection string="data source=(localdb)\v11.0;initial catalog=FilmDataBase;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

☒ Сохранить параметры соединения в App.Config как:

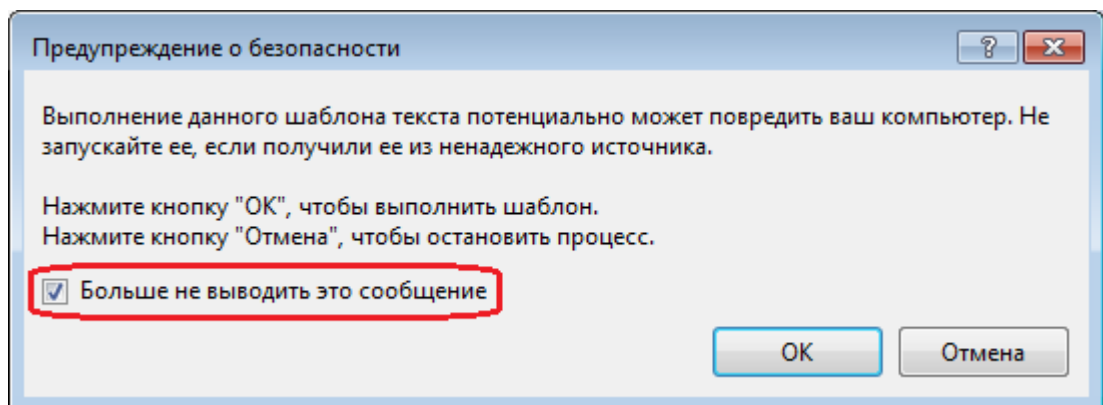
FilmDataBaseEntities

< Назад | **Далее >** | Готово | Отмена

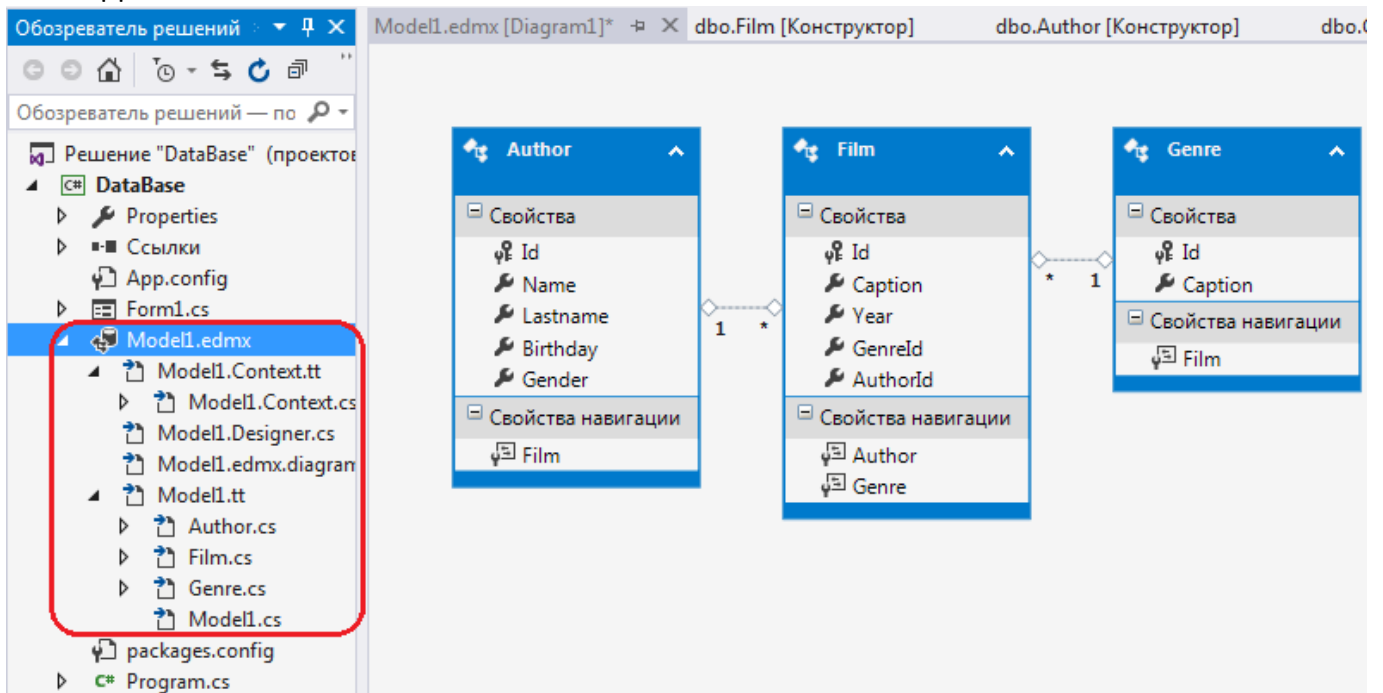
21. В следующем окне поставить галочку напротив поля **Таблицы**. Остальное оставить по умолчанию. Нажать **Готово**.



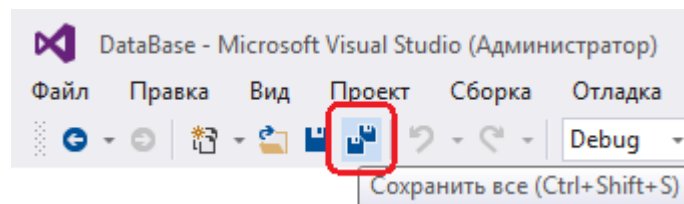
При появлении **Предупреждения о безопасности** установить галочку **Больше не выводить это сообщение** и нажать **ОК**.



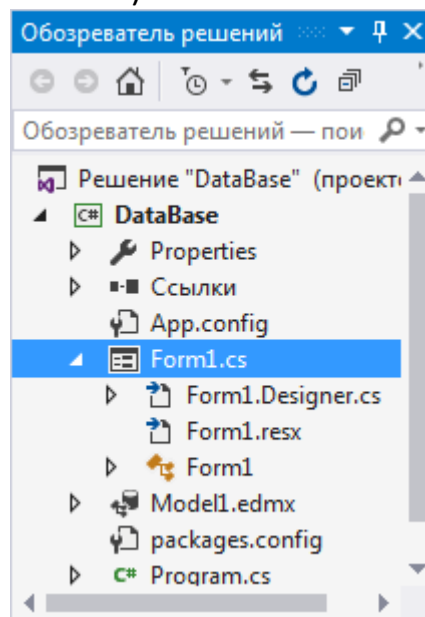
22. В Обозревателе решений появится новая ветка **Model1.edmx**. На диаграмме отобразится структура и отношения (один-ко-многим) таблиц созданной базы данных.



23. После этого обязательно сохранить весь проект.



24. Перейти на **Конструктор формы** (в окне **Обозревателя решений** двойной клик левой кнопкой мыши на **Form1.cs**)



25. На форму поместить 4 компонента **Button**, 3 компонента **TextBox**, 5 компонентов **Label**, 2 компонента **ComboBox**, 1 компонент **DataGridView**.

Заполнить свойства компонентов:

Компонент	Свойство (Name)	Свойство Text
button1	btnAddFilm	Добавить фильм
button2	btnDelete	Удалить
button3	btnAuthor	Добавить автора
button4	btnGenre	Добавить жанр
label1	label1	Жанр
label2	label2	Режиссёр
label3	label3	Название фильма
label4	label4	Год
label5	label5	Поиск фильма
textBox1	tbName	
textBox2	tbYear	
textBox3	tbFindFilm	
comboBox1	cbGenre	
comboBox2	cbAuthor	
dataGridView1	dgFilm	

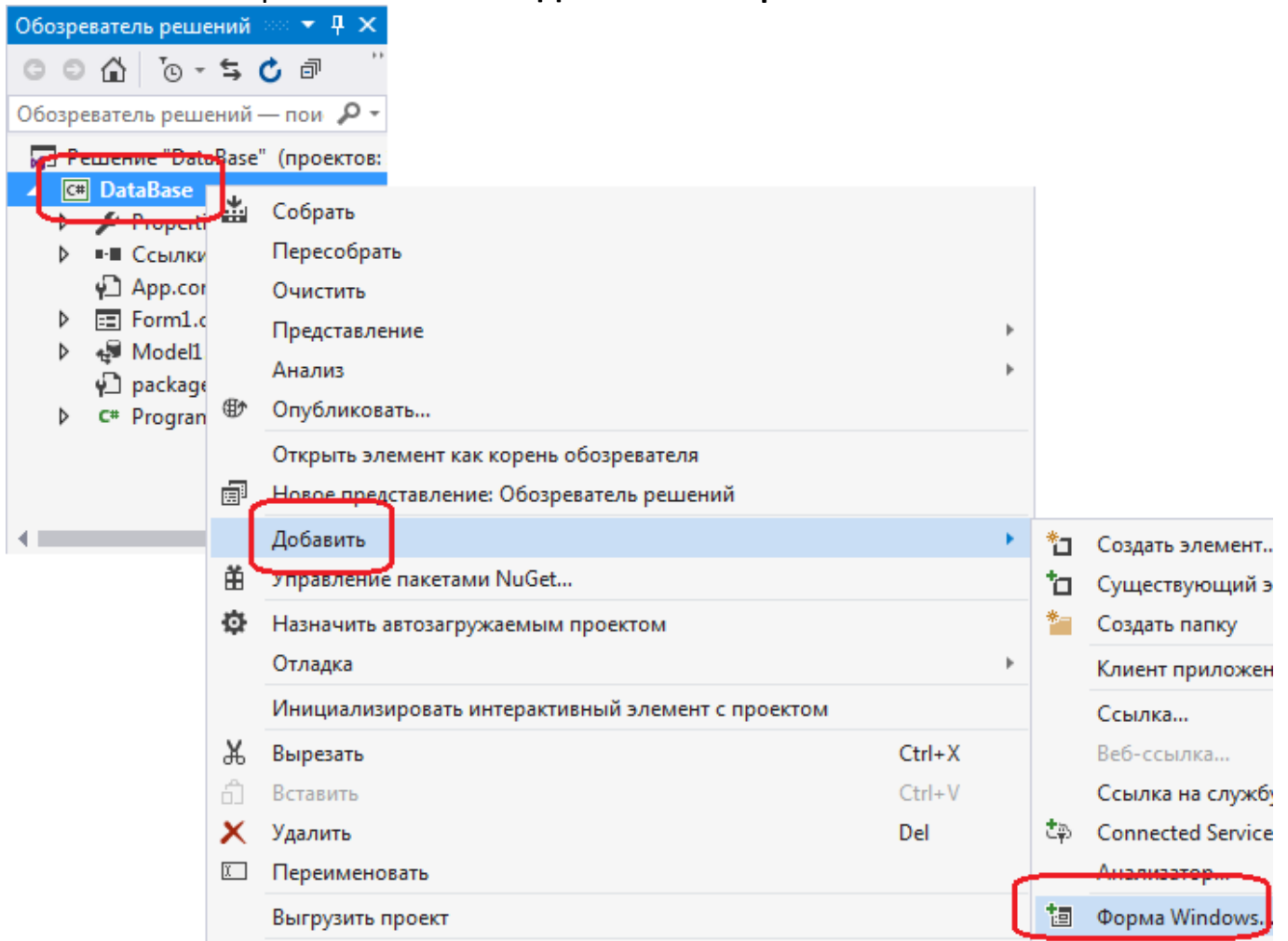
The screenshot shows a Windows Form titled "Form1" with a light gray background. At the top, there are two dropdown menus labeled "Жанр" (Genre) and "Режиссёр" (Director). Below them are three text boxes: "Название фильма" (Movie Name), "Год" (Year), and "Поиск фильма" (Search Movie). To the right of the "Название фильма" and "Год" text boxes is a button labeled "Добавить фильм" (Add Movie). To the right of the "Режиссёр" dropdown is a button labeled "Добавить автора" (Add Author). To the right of the "Поиск фильма" text box is a button labeled "Добавить жанр" (Add Genre). Below the "Поиск фильма" text box is a button labeled "Удалить" (Delete). At the bottom of the form is a large, empty gray rectangular area representing a DataGridView control.

26. Дважды кликнуть по каждой из четырёх кнопок, чтобы автоматически создались в коде методы обработки нажатия кнопок.
27. У компонента **tbFindFilm** (бывший **textBox3**) в **Окне свойств** перейти на вкладку **События**, и дважды кликнуть мышкой справа от события **KeyUp**.
28. У компонента **dgFilm** (бывший **dataGridView1**) в **Окне свойств** на вкладке **Свойства** назначить свойствам следующие значения:

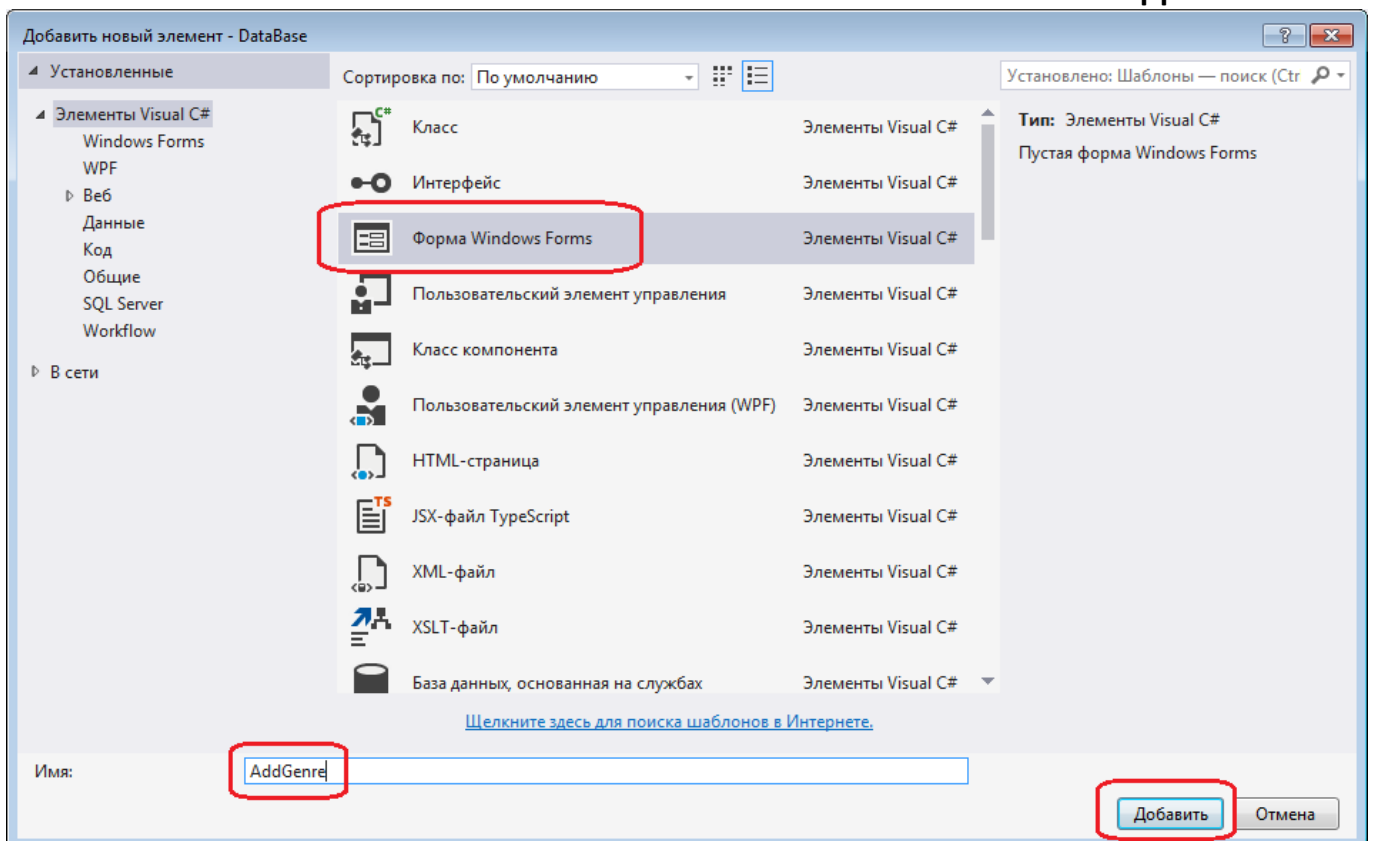
Свойство	Значение свойства	Описание
AllowUserToAddRows	False	Запретить вручную добавлять строки
AllowUserToDeleteRows	False	Запретить вручную удалять строки
AutoSizeColumnsMode	Fill	Выравнивание таблицы на всю ширину компонента
MultiSelect	False	Запрет выделения нескольких строк одновременно
RowHeadersVisible	False	Скрыть заголовочную строку
SelectionMode	FullRowSelect	При выделении выделяется вся строка целиком



29.Перейти в **Обозреватель решений**, кликнуть правой кнопкой мыши на названии проекта **DataBase – Добавить – Форма Windows**.

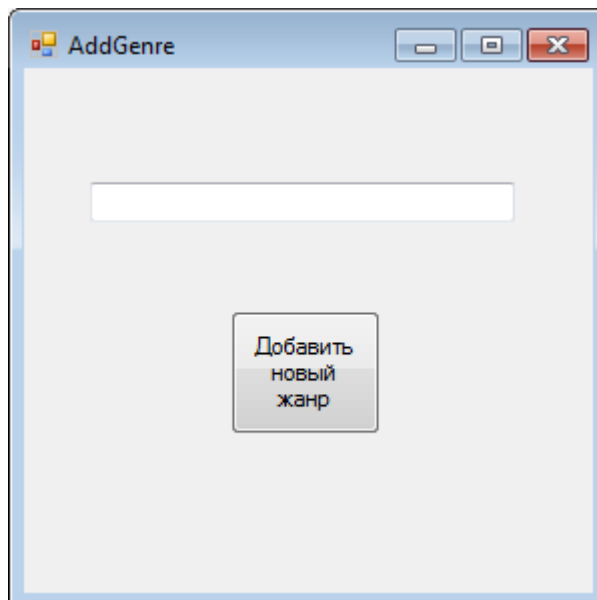


30.В появившемся окне заменить имя **Form2** на **AddGenre** и нажать **Добавить**.



**31.** На появившейся второй форме разместить 1 компонент **Button**, 1 компонент **TextBox**.

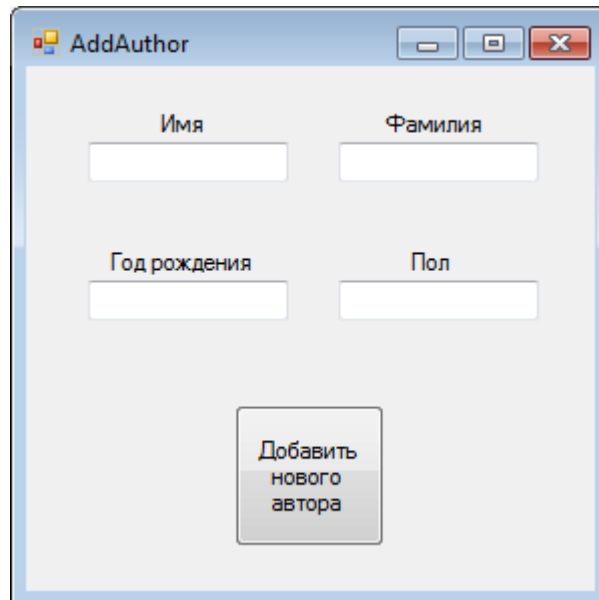
Компонент	Свойство (Name)	Свойство Text
button1	btnAddNewGenre	Добавить жанр
textBox1	tbAddNewGenre	



**32.** Дважды кликнуть по кнопке **Добавить новый жанр**, чтобы автоматически создался в коде метод обработки нажатия кнопки.

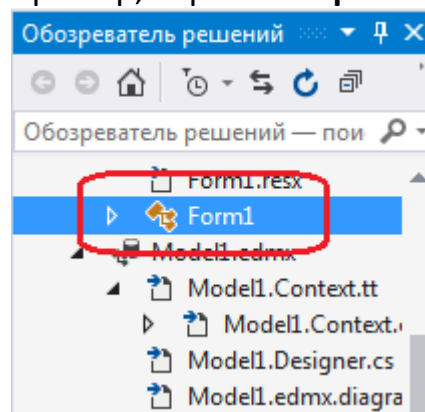
**33.** Аналогично добавить третье окно формы и назвать его **AddAuthor**. На ней разместить 4 компонента **Label**, 4 компонента **TextBox**, 1 компонент **Button**.

Компонент	Свойство (Name)	Свойство Text
button1	btnAddNewAuthor	Добавить нового автора
label1	label1	Имя
label2	label2	Фамилия
label3	label3	Год рождения
label4	label4	Пол
textBox1	tbAddNewAuthorName	
textBox2	tbAddNewAuthorLastName	
textBox3	tbAddNewAuthorBirthday	
textBox4	tbAddNewAuthorGender	



34. Дважды кликнуть по кнопке **Добавить нового автора**, чтобы автоматически создался в коде метод обработки нажатия кнопки.

35. Перейти в класс **Form1** (например, через **Обозреватель решений**)



36. В классе **Form1** написать следующий код:

```
1  using System;
2  using System.Linq;
3  using System.Windows.Forms;
4
5  namespace DataBase
6  {
7      public partial class Form1 : Form
8      {
9          // Связываем переменную entities с созданной базой данных
10         private readonly FilmDataBaseEntities entities = new FilmDataBaseEntities();
11         public Form1()
12         {
13             InitializeComponent();
14             // записываем в переменные списки авторов и фильмов
15             var authors = entities.Author.ToList();
16             var films = entities.Film.ToList();
17             // очищаем выпадающий список жанров и формируем его заново
18             ReloadGenres();
19             // формируем выпадающий список режиссёров построчно
20             foreach (var author in authors)
21                 cbAuthor.Items.Add(author.Lastname);
22             // формируем столбцы вида имя в программе/отображение
23             dgFilm.Columns.Add("Id", "Id");
24             dgFilm.Columns.Add("Caption", "Название");
25             dgFilm.Columns.Add("Year", "Год");
26             dgFilm.Columns.Add("Autor", "Автор");
27             dgFilm.Columns.Add("Genre", "Жанр");
28             // добавляем отображение сводной таблицы построчно
29             foreach (var film in films)
30                 dgFilm.Rows.Add(film.Id, film.Caption, film.Year,
31                                 film.Author.Name + " " + film.Author.Lastname,
32                                 film.Genre.Caption);
33         }
34     }
35 }
```

```

35 // метод перезагрузки списка в выпадающем окне жанров
36 private void ReloadGenres()
37 {
38     cbGenre.Items.Clear();
39     var genres = entities.Genre.ToList();
40     foreach (var genre in genres)
41         cbGenre.Items.Add(genre.Caption);
42 }
43
44 // метод перезагрузки списка в выпадающем окне авторов
45 private void ReloadAuthors()
46 {
47     cbAuthor.Items.Clear();
48     var authors = entities.Author.ToList();
49     foreach (var author in authors)
50         cbAuthor.Items.Add(author.Lastname);
51 }
52
53 // при закрытии формы закрываем соединение с базой данных
54 protected override void OnFormClosing(FormClosingEventArgs e)
55 {
56     entities.Dispose();
57     base.OnFormClosing(e);
58 }
59
60 // метод добавления фильма в базу данных
61 private void btnAddFilm_Click(object sender, EventArgs e)
62 {
63     var film = new Film();
64     film.Caption = tbName.Text;
65     film.Year = Convert.ToInt32(tbYear.Text);
66     film.AuthorId = entities.Author.FirstOrDefault(a =>
67         a.Lastname == cbAuthor.SelectedItem.ToString()).Id;
68     film.GenreId = entities.Genre.FirstOrDefault(g =>
69         g.Caption == cbGenre.SelectedItem.ToString()).Id;

```

```

70         entities.Film.Add(film);
71         entities.SaveChanges();
72         dgFilm.Rows.Add(film.Id, film.Caption, film.Year,
73             film.Author.Name + " " + film.Author.Lastname,
74             film.Genre.Caption);
75     }
76
77     // метод вызова окна добавления режиссёра
78     private void btnAuthor_Click(object sender, EventArgs e)
79     {
80         AddAuthor add = new AddAuthor();
81         add.ShowDialog();
82         ReloadAuthors();
83     }
84
85     // метод вызова окна добавления жанра
86     private void btnGenre_Click(object sender, EventArgs e)
87     {
88         AddGenre add = new AddGenre();
89         add.ShowDialog();
90         ReloadGenres();
91     }
92
93     // метод поиска названий фильмов "налету"
94     private void tbFindFilm_KeyUp(object sender, KeyEventArgs e)
95     {
96         foreach (DataGridViewRow dgFilmRow in dgFilm.Rows)
97             dgFilmRow.Visible = dgFilmRow.Cells[1].Value.ToString()
98                 .ToUpper().Contains(tbFindFilm.Text.ToUpper());
99     }
100
101     // метод удаления выделенной строки с фильмом
102     private void btnDelete_Click(object sender, EventArgs e)
103     {
104         var selectedrow = dgFilm.SelectedRows[0];
105         string name = selectedrow.Cells["Caption"].Value.ToString();
106         Film film = entities.Film.FirstOrDefault(f => f.Caption == name);
107         entities.Film.Remove(film);
108         entities.SaveChanges();
109         dgFilm.Rows.Remove(selectedrow);
110     }
111 }
112

```

37.Перейти в класс **AddAuthor** и записать код:

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace DataBase
5  {
6      public partial class AddAuthor : Form
7      {
8          // создать новую сущность и связать её с экземпляром базы данных
9          private readonly FilmDataBaseEntities entities = new FilmDataBaseEntities();
10
11         public AddAuthor()
12         {
13             InitializeComponent();
14         }
15
16         // метод добавления нового режиссера в базу данных
17         private void btnAddNewAuthor_Click(object sender, EventArgs e)
18         {
19             entities.Author.Add(new Author() { Name = tbAddNewAuthorName.Text,
20                 Lastname = tbAddNewAuthorLastName.Text,
21                 Birthday = Convert.ToInt32(tbAddNewAuthorBirthday.Text),
22                 Gender = tbAddNewAuthorGender.Text });
23             entities.SaveChanges();
24             // закрываем форму после добавления нового режиссёра
25             Close();
26         }
27     }
28 }
```

38.Перейти в класс **AddGenre** и записать код:

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace DataBase
5  {
6      public partial class AddGenre : Form
7      {
8          // создать новую сущность и связать её с экземпляром базы данных
9          private readonly FilmDataBaseEntities entities = new FilmDataBaseEntities();
10
11         public AddGenre()
12         {
13             InitializeComponent();
14         }
15
16         // метод добавления нового жанра в базу данных
17         private void btnAddNewGenre_Click(object sender, EventArgs e)
18         {
19             entities.Genre.Add(new Genre() { Caption = tbAddNewGenre.Text });
20             entities.SaveChanges();
21             // закрываем форму после добавления нового жанра
22             Close();
23         }
24     }
25 }
```

**39.** Запустить программу на компиляцию и выполнение:

- 1) Добавить несколько авторов
- 2) Добавить несколько жанров
- 3) Добавить несколько фильмов
- 4) Выполнить поиск по фильмам

The screenshot shows a Windows form titled "Form1" with a light gray background. It contains several input fields and buttons. At the top, there are two dropdown menus labeled "Жанр" (Genre) and "Режиссёр" (Director). Below them are two text boxes labeled "Название фильма" (Movie Name) and "Год" (Year). To the right of the "Год" box is a button labeled "Добавить фильм" (Add movie). Below the "Название фильма" box is a text box labeled "Поиск фильма" (Search movie). To the right of the "Поиск фильма" box is a button labeled "Найти фильм" (Find movie). On the far right, there are two buttons: "Добавить автора" (Add author) and "Добавить жанр" (Add genre). Red numbered arrows indicate the sequence of actions: arrow 1 points to the "Добавить автора" button; arrow 2 points to the "Добавить жанр" button; arrow 3 points to the "Добавить фильм" button; and arrow 4 points to the "Найти фильм" button. At the bottom of the form is a large gray rectangular area, likely a list box or a placeholder for search results.