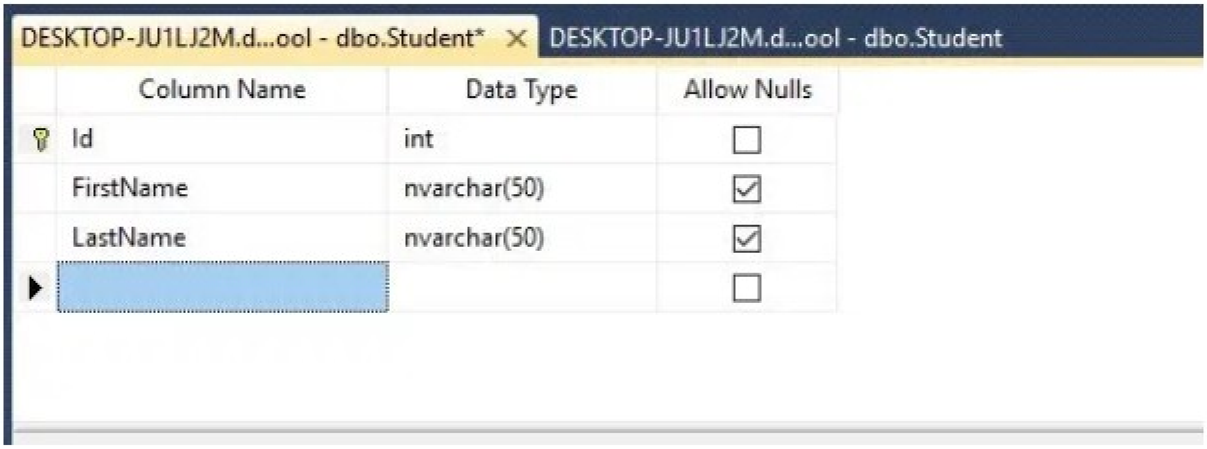
WPF + Entity Framework: выбор, вывод, обновление и удаление данных

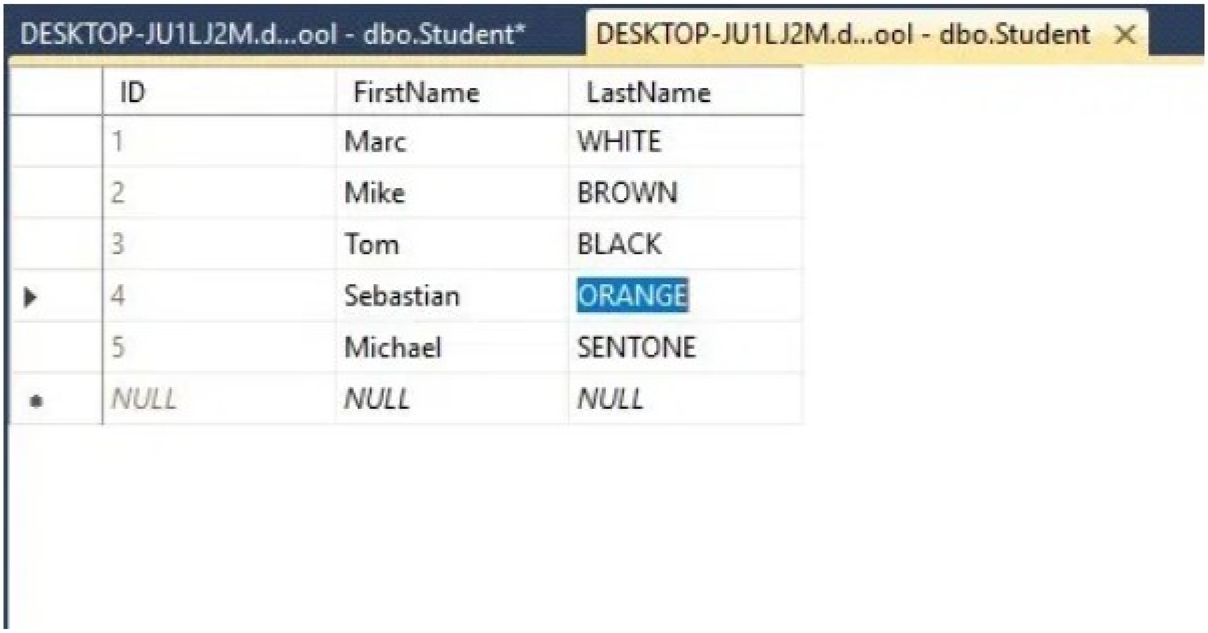
Шаг 1.

Создадим таблицу Student с помощью MSSQL. Таблица должна содержать следующие поля: Id, FirstName, LastName.


DESKTOP-JU1LJ2M.d...ool - dbo.Student* ×
DESKTOP-JU1LJ2M.d...ool - dbo.Student

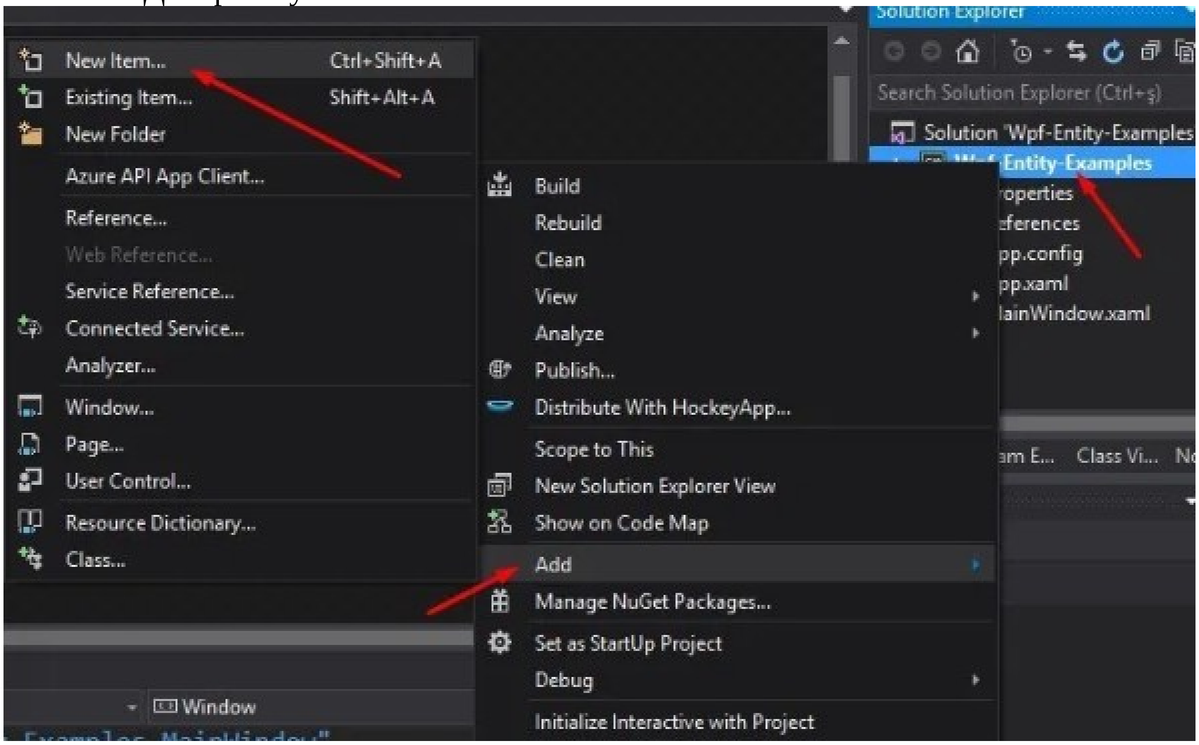
Column Name	Data Type	Allow Nulls
Id	int	□
FirstName	nvarchar(50)	☑
LastName	nvarchar(50)	☑
		□

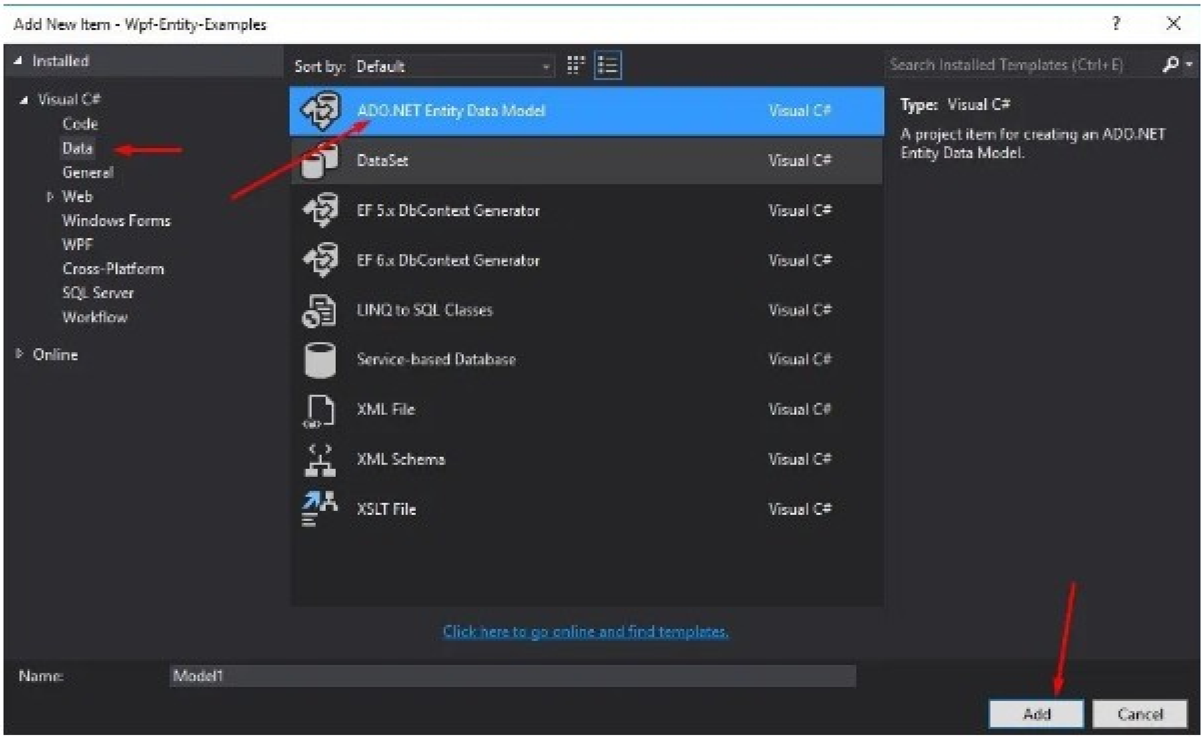
Заполним таблицу произвольными данными:


DESKTOP-JU1LJ2M.d...ool - dbo.Student*
DESKTOP-JU1LJ2M.d...ool - dbo.Student ×

ID	FirstName	LastName
1	Marc	WHITE
2	Mike	BROWN
3	Tom	BLACK
4	Sebastian	ORANGE
5	Michael	SENTONE
NULL	NULL	NULL

Шаг 2.
Подключение БД к проекту WPF.

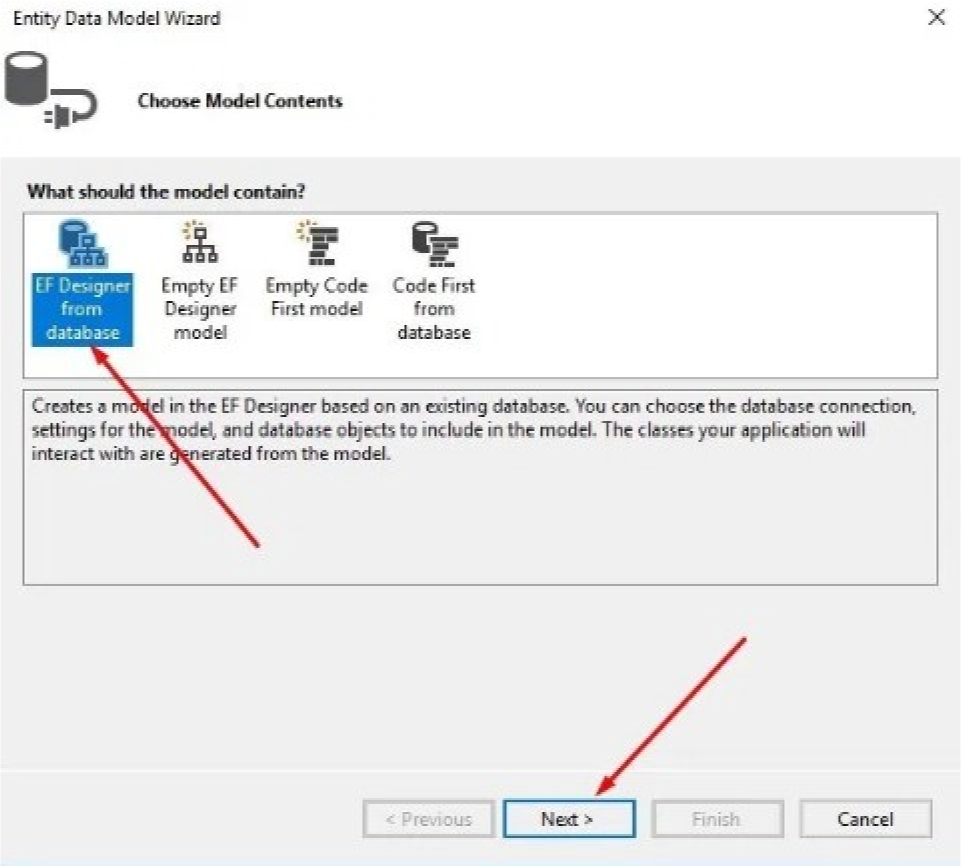



Add New Item - Wpf-Entity-Examples
?
Installed
Visual C#
Code
Data
General
Web
Windows Forms
WPF
Cross-Platform
SQL Server
Workflow
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
ADO.NET Entity Data Model Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator Visual C#
LINQ to SQL Classes Visual C#
Service-based Database Visual C#
XML File Visual C#
XML Schema Visual C#
XSLT File Visual C#
Click here to go online and find templates.
Add
Cancel

Type: Visual C#

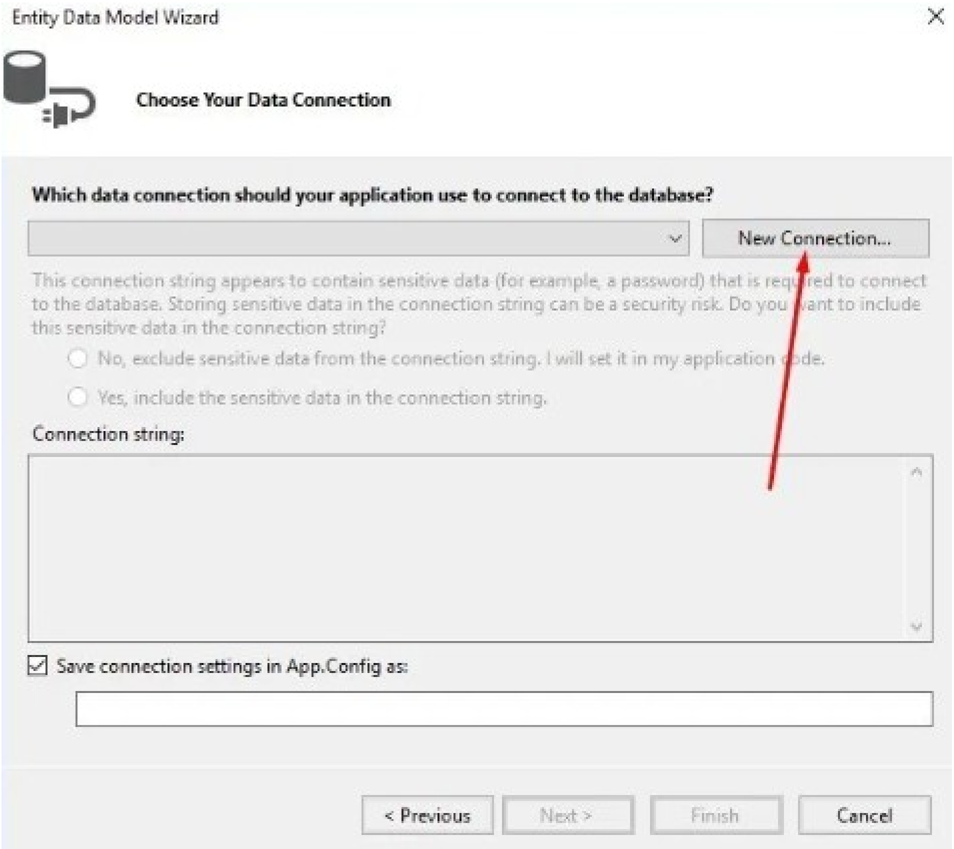
A project item for creating an ADO.NET Entity Data Model.

Name: Model1



What should the model contain?

Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.



Which data connection should your application use to connect to the database?

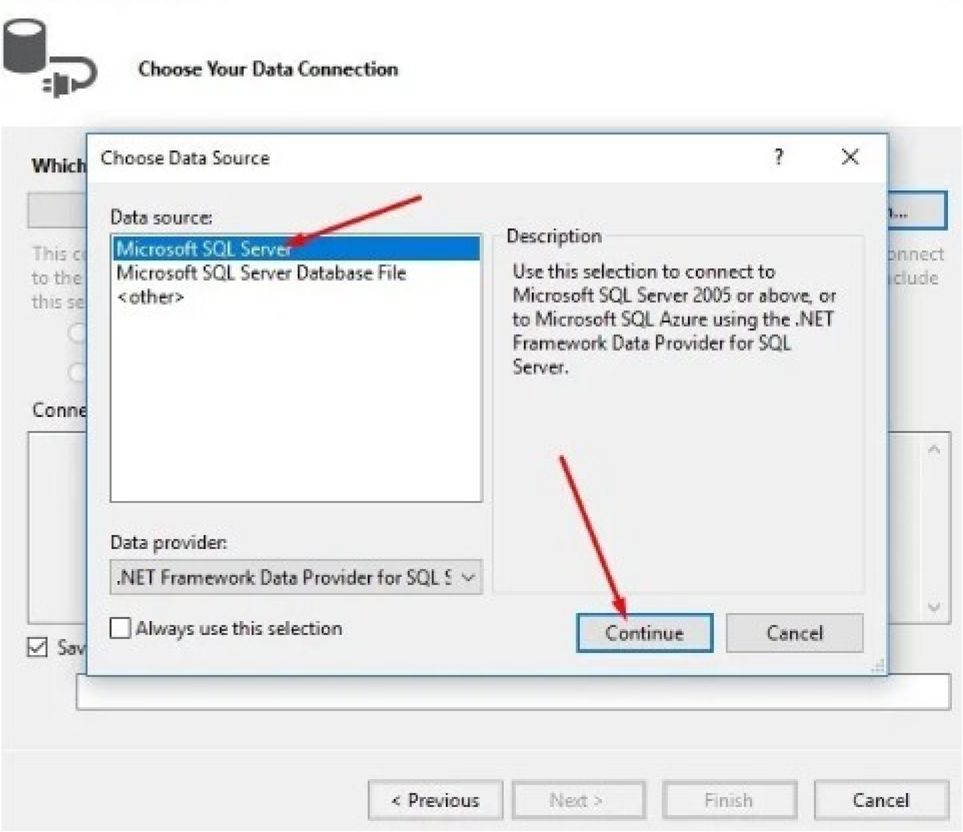
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

○ Yes, include the sensitive data in the connection string.

Connection string:

☑ Save connection settings in App.Config as:



Choose Your Data Connection

Choose Data Source

Data source:

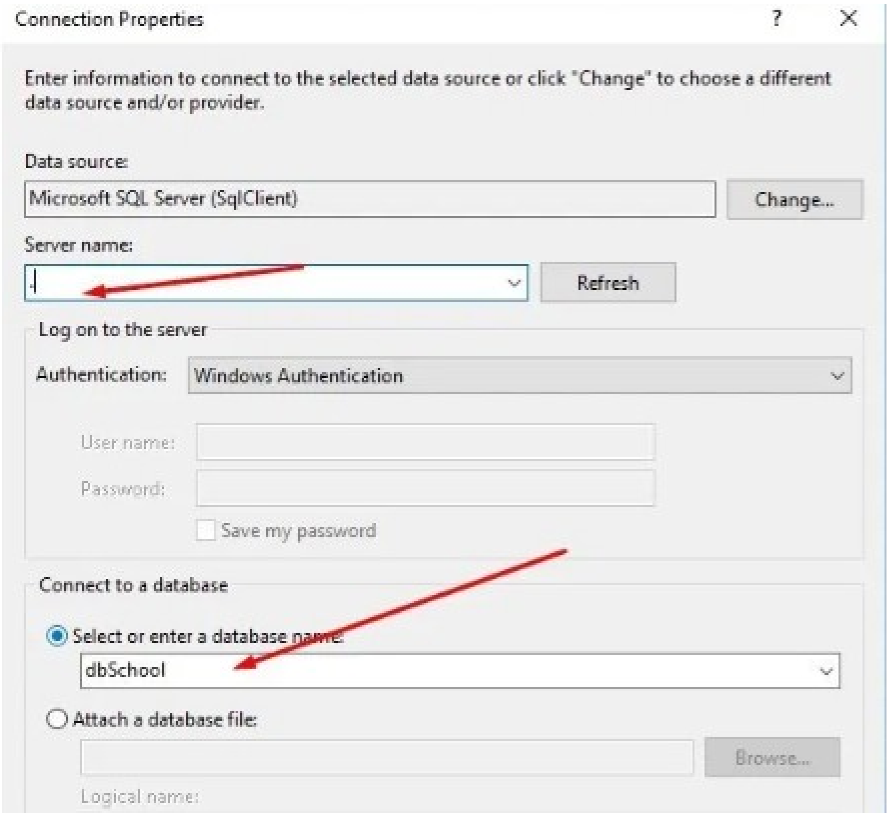
- Microsoft SQL Server
- Microsoft SQL Server Database File
- <other>

Description

Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.

Data provider: .NET Framework Data Provider for SQL S

□ Always use this selection



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: .

Log on to the server

Authentication: Windows Authentication

User name:

Password:

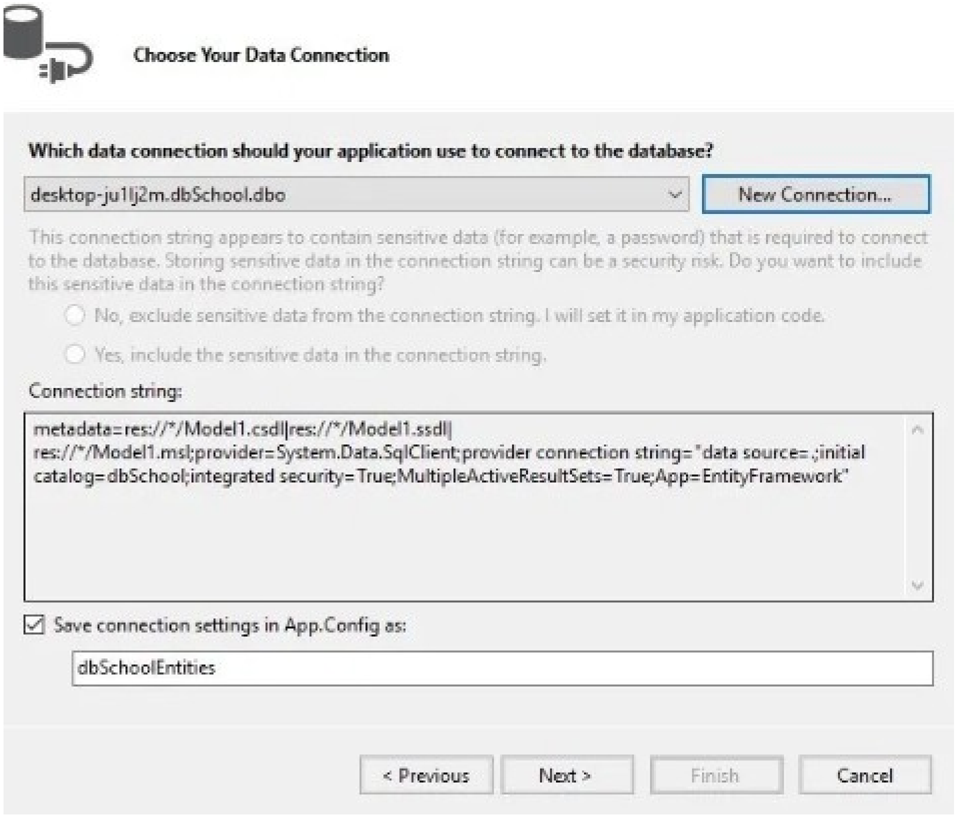
□ Save my password

Connect to a database

● Select or enter a database name: dbSchool

○ Attach a database file:

Logical name:



Choose Your Data Connection

Which data connection should your application use to connect to the database?

desktop-ju1lj2m.dbSchool.dbo

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

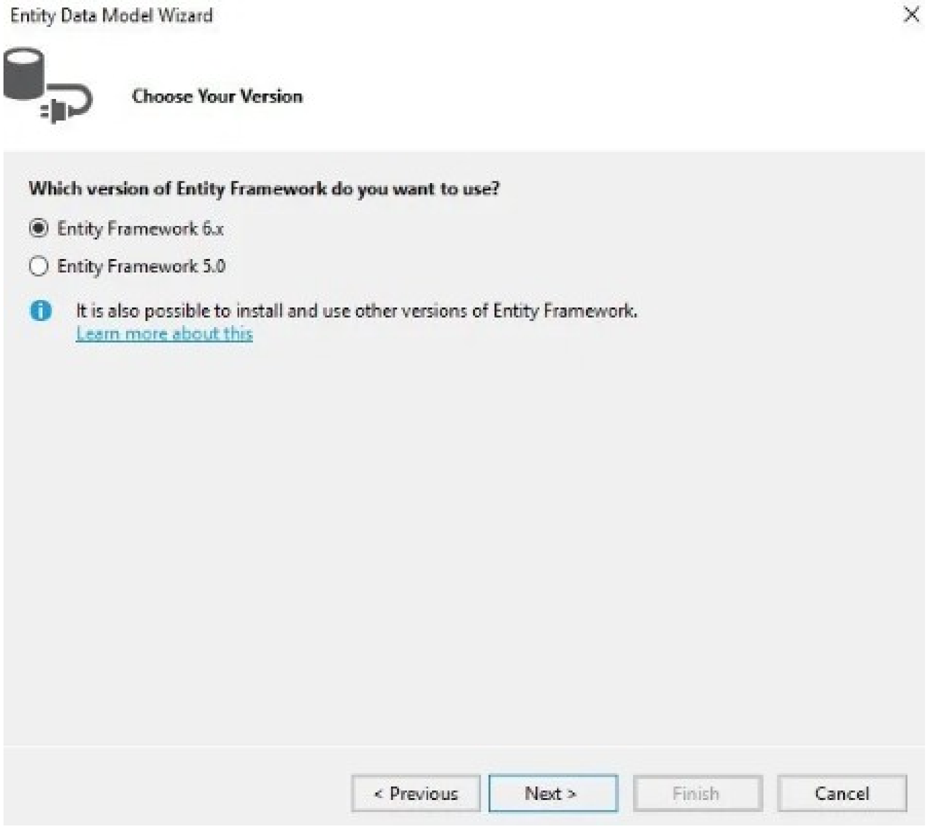
○ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/Model1.csdl|res://*/Model1.ssdl|
res://*/Model1.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial
catalog=dbSchool;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☑ Save connection settings in App.Config as:

dbSchoolEntities



Entity Data Model Wizard

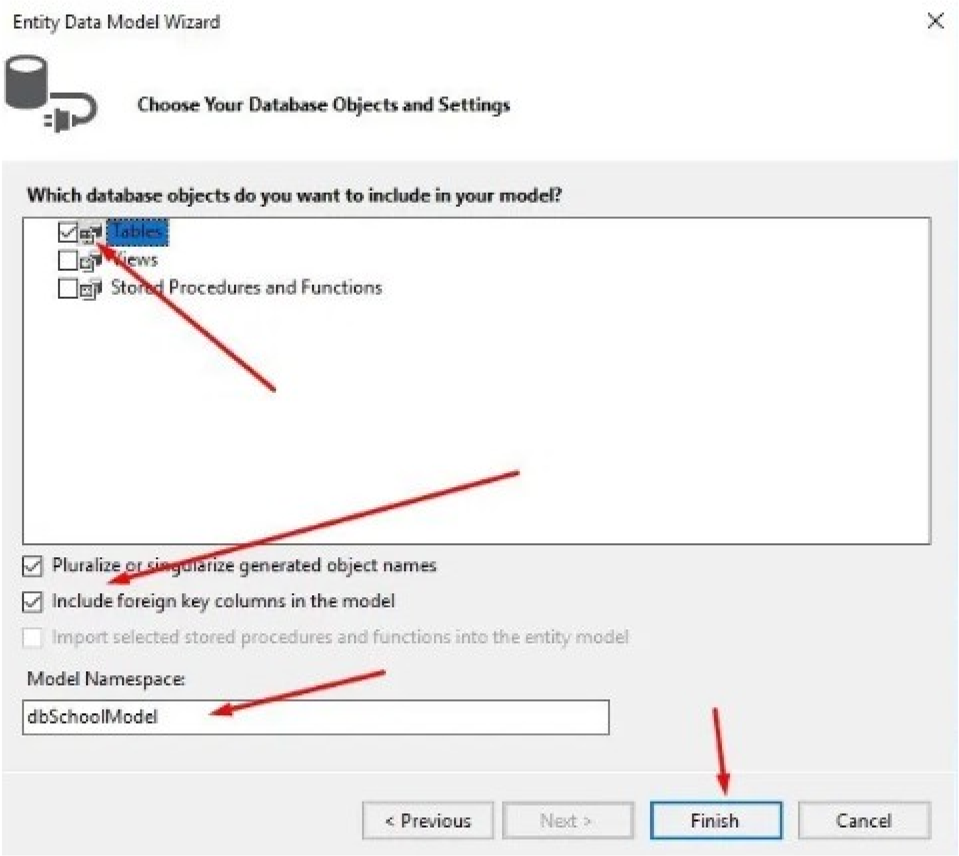
Choose Your Version

Which version of Entity Framework do you want to use?

● Entity Framework 6.x

○ Entity Framework 5.0

It is also possible to install and use other versions of Entity Framework.
Learn more about this



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

- ☑ Tables
- □ Views
- □ Stored Procedures and Functions

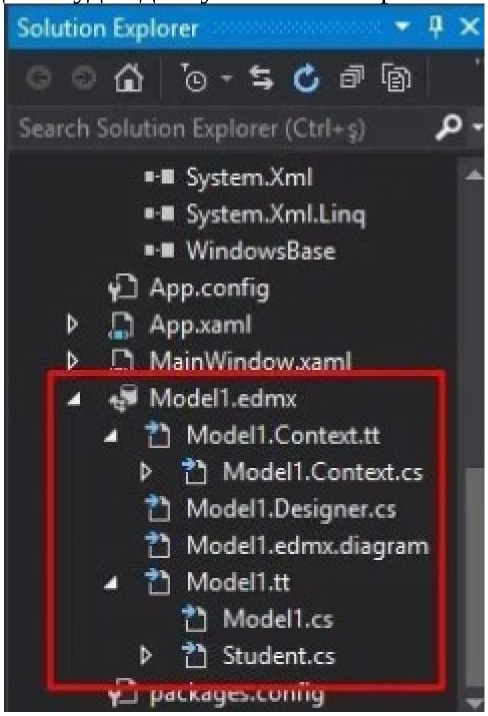
☑ Pluralize or singularize generated object names

☑ Include foreign key columns in the model

□ Import selected stored procedures and functions into the entity model

Model Namespace: dbSchoolModel

Ждём добавления модели. Модель будет доступна в «Обозревателе решений».



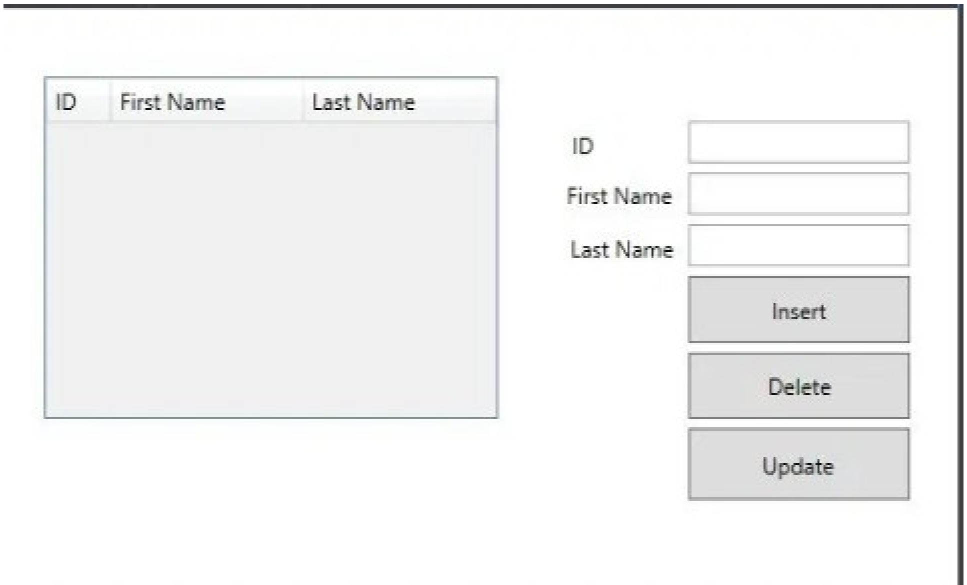
Solution Explorer

```
Search Solution Explorer (Ctrl+ş)
        System.Xml
        System.Xml.Linq
        WindowsBase
      App.config
    ▷ App.xaml
    ▷ MainWindow.xaml
    ◢ Model1.edmx
      ◢ Model1.Context.tt
        ▷ Model1.Context.cs
        Model1.Designer.cs
        Model1.edmx.diagram
      ◢ Model1.tt
          Model1.cs
        ▷ Student.cs
      packages.config
```

Добавляем элементы на окно WPF через разметку XAML:

```
<Window x:Class="Wpf_Entity_Examples.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Wpf_Entity_Examples"
        mc:Ignorable="d"
        Title="www.csharp-console-examples.com" Height="350" Width="525" Loaded="Window_Loaded">
    <Grid>
        <DataGrid Name="dgogrenci" HorizontalAlignment="Left" AutoGenerateColumns="False"
Height="185" Margin="22,37,0,0" VerticalAlignment="Top" Width="246">
            <DataGrid.Columns>
                <DataGridTextColumn Binding="{Binding ID}" Header="ID" Width="*"/>
                <DataGridTextColumn Binding="{Binding FirstName}" Header="First Name" Width="3*"/>
                <DataGridTextColumn Binding="{Binding LastName}" Header="Last Name" Width="3*" />
            </DataGrid.Columns>
        </DataGrid>
        <TextBox Name="txtId"  HorizontalAlignment="Left" Height="23" Margin="371,61,0,0"
TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>
        <TextBox Name="txtFirst" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text=""
VerticalAlignment="Top" Width="120" Margin="371,89,0,0"/>
        <TextBox Name="txtLast" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text=""
VerticalAlignment="Top" Width="120" Margin="371,117,0,0"/>
        <Button Content="Insert" HorizontalAlignment="Left" Margin="371,145,0,0" VerticalAlignment="Top"
Width="120" Height="36" Click="Button_Click"/>
        <Button Content="Delete" HorizontalAlignment="Left" Height="36" Margin="371,186,0,0"
VerticalAlignment="Top" Width="120" Click="Button_Click_1"/>
        <Button Content="Update" HorizontalAlignment="Left" Height="39" Margin="371,227,0,0"
VerticalAlignment="Top" Width="120" Click="Button_Click_2"/>
        <Label x:Name="label" Content="ID" HorizontalAlignment="Left" Margin="303,61,0,0"
VerticalAlignment="Top"/>
        <Label x:Name="label1" Content="First Name" HorizontalAlignment="Left" Margin="300,88,0,0"
VerticalAlignment="Top"/>
        <Label x:Name="label2" Content="Last Name" HorizontalAlignment="Left" Margin="302,117,0,0"
VerticalAlignment="Top"/>
    </Grid>
</Window>
```

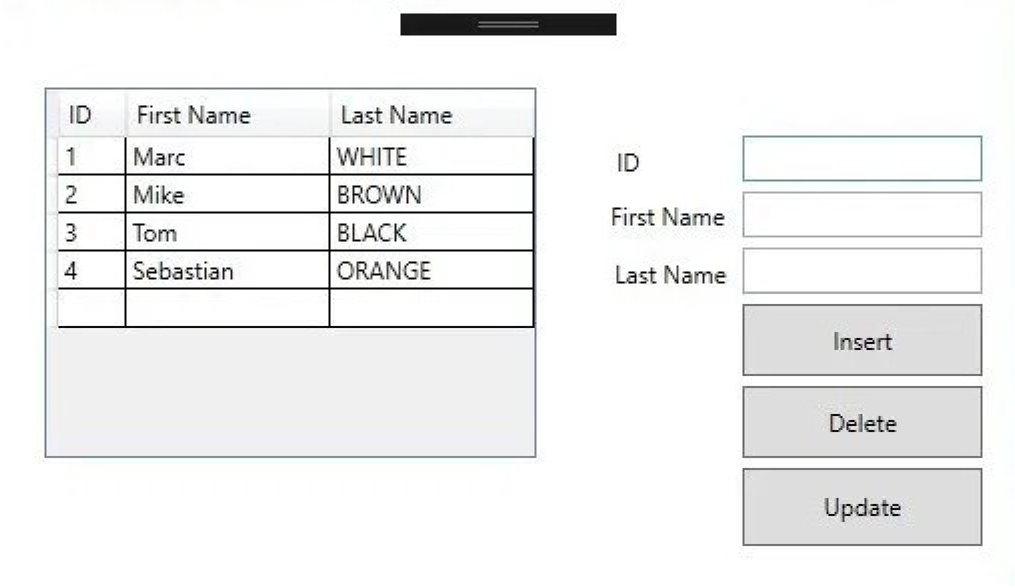
В результате наше окно приобретает следующий вид:



Переходим к написанию кода для нашего окна:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
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;

namespace Wpf_Entity_Examples
{

    public partial class MainWindow : Window
    {
        dbSchoolEntities db;
        public MainWindow()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            Student student = new Student();
            student.FirstName = txtFirst.Text;
            student.LastName = txtLast.Text;
            db.Students.Add(student);
            db.SaveChanges();
            dgogrenci.ItemsSource = db.Students.ToList();
        }

        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
            int num = Convert.ToInt32(txtId.Text);
            var dRow = db.Students.Where(w => w.ID == num).FirstOrDefault();
            db.Students.Remove(dRow);
            db.SaveChanges();
            dgogrenci.ItemsSource = db.Students.ToList();
        }

        private void Button_Click_2(object sender, RoutedEventArgs e)
        {
            int num = Convert.ToInt32(txtId.Text);
            var uRow = db.Students.Where(w => w.ID == num).FirstOrDefault();
            uRow.FirstName = txtFirst.Text;
            uRow.LastName = txtLast.Text;
            db.SaveChanges();
            dgogrenci.ItemsSource = db.Students.ToList();
        }

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            db = new dbSchoolEntities();
            dgogrenci.ItemsSource = db.Students.ToList();
        }
    }
}
```

В результате получаем рабочее приложение:



ID	First Name	Last Name
1	Marc	WHITE
2	Mike	BROWN
3	Tom	BLACK
4	Sebastian	ORANGE