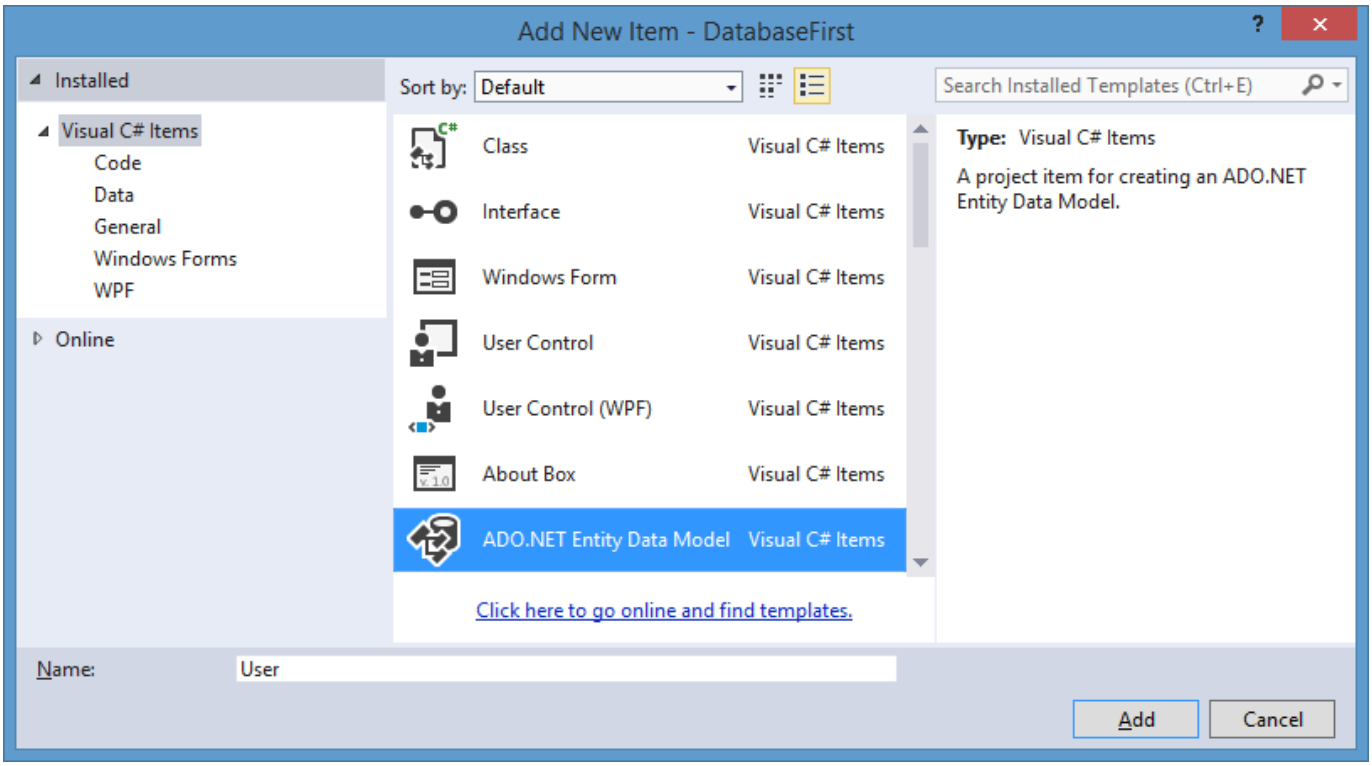
Model First

Данное руководство устарело. Актуальное руководство: Руководство по Entity Framework Core

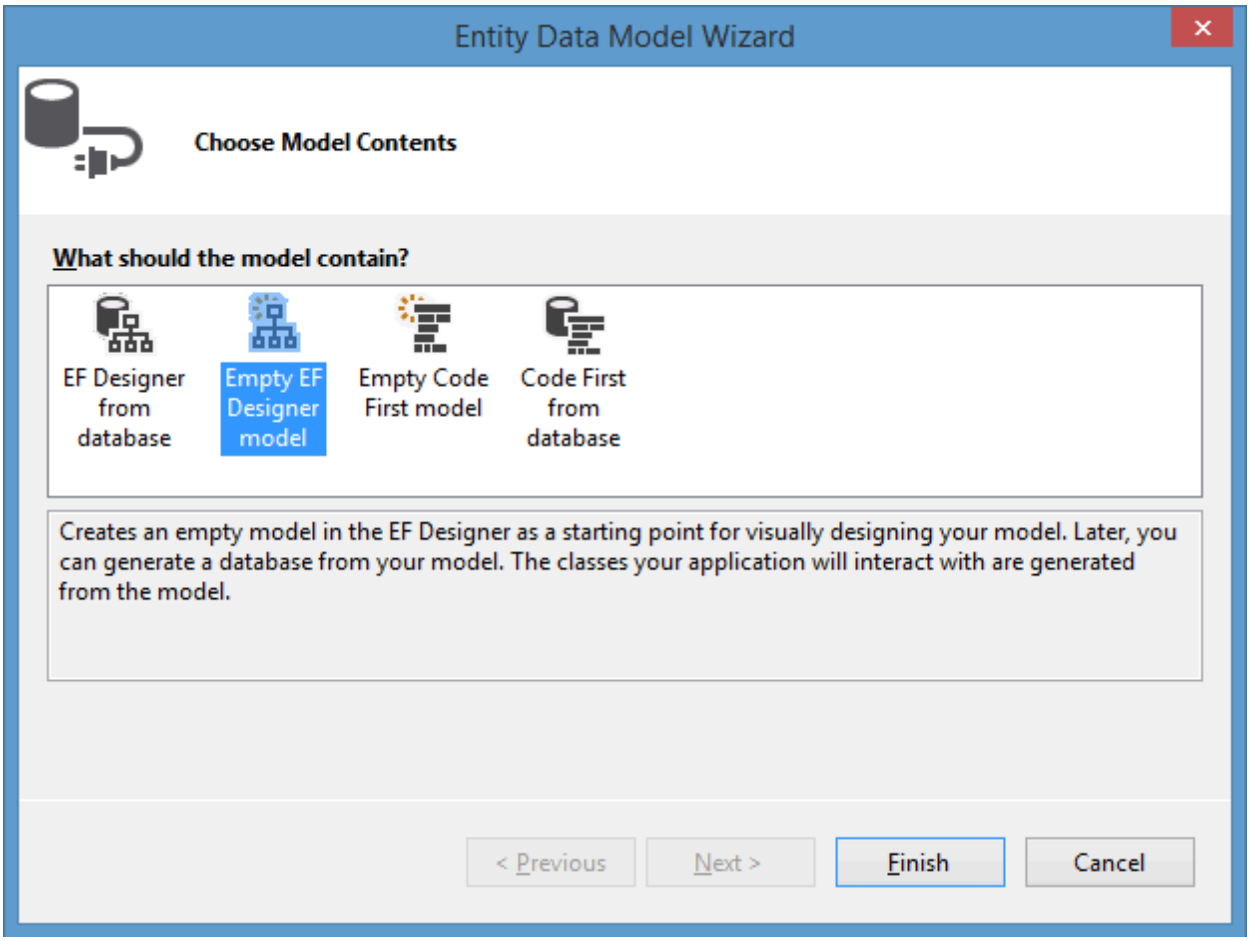
Последнее обновление: 31.10.2015

Model First представляет еще один подход к работе с Entity Framework. Суть данного подхода состоит в том, что сначала делается модель, а потом по ней создается база данных.

Итак, создадим новый проект по типу Console Application. И затем добавим в проект новый элемент. Нажмем правой кнопкой мыши на проект в окне Solution Explorer и в появившемся списке выберем **Add -> New Item**. И затем в окне добавления нового элемента выберем **ADO.NET Entity Data Model**:

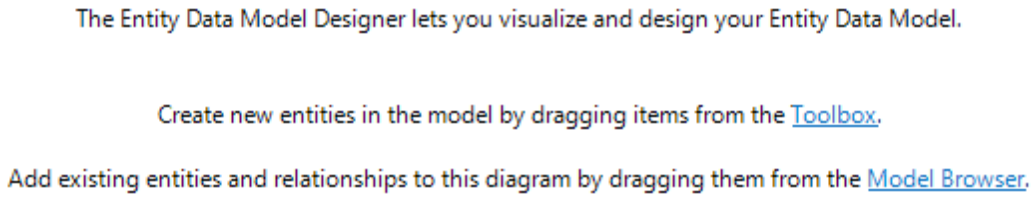


Поскольку модель будет описывать человека, то назовем ее User. Нажмем OK и нам откроется мастер создания модели. Если у нас Visual Studio с пакетами обновления SP2, SP3, то мастер создания модели выглядит следующим образом:

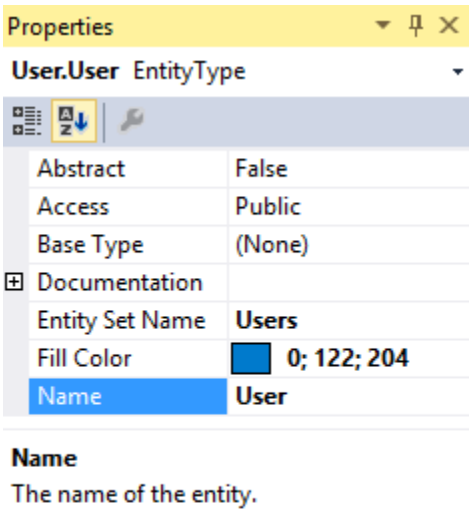


Окно нам предлагает четыре варианта создания модели, из которых нам надо выбрать **Empty EF Designer Model**.

Нажмем кнопку Finish, и перед нами откроется пустое окно создания модели.



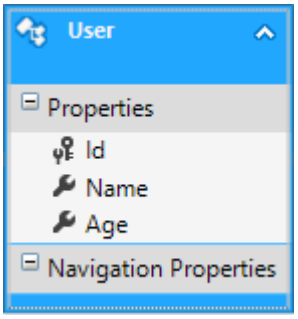
Перетащим на это поле с панели Toolbox (Панель Инструментов) в левой части элемент Entity. Теперь у нас на поле создания модели имеется небольшая схема будущей модели, в которой сейчас по умолчанию указано лишь одно поле - Id. Во-первых, переименуем сущность. По умолчанию она называется Entity1. Выделим схему и перейдем к окну свойств в правом нижнем углу:



Здесь изменим значение свойства Name на User. Это у нас будет имя сущности. И также изменим значение свойства Entity Set Name на Users. Это у нас будет название набора объектов User.

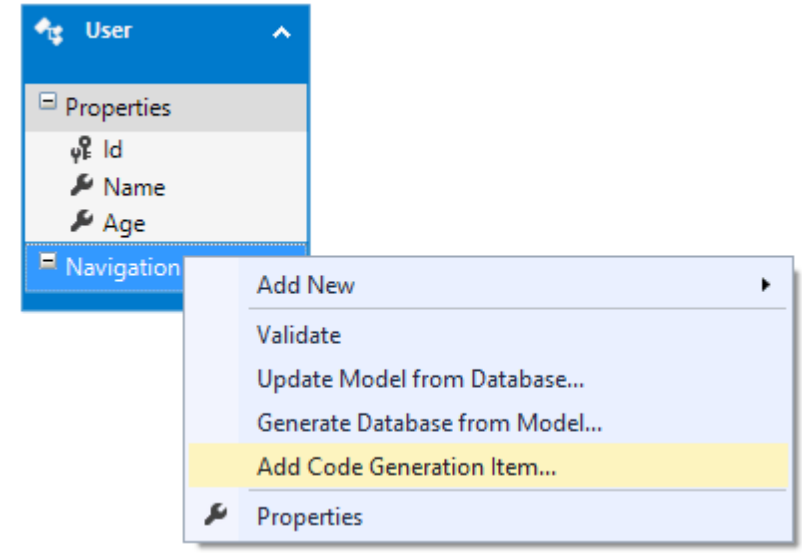
Далее создадим несколько свойств. Сущность у нас будет простая и будет содержать всего два свойства для имени и возраста. Итак, выделим схему сущности и нажмем на правую кнопку мыши. В выпадающем списке выберем **Add New -> Scalar Property**. После этого будет добавлено новое свойство. Scalar Property подразумевают свойства на основе простейших типов int, float, string и т.д. Добавим два свойства - Name и Age. По умолчанию все добавляемые свойства имеют тип string. Однако мы можем изменить тип в окне свойств.

Таким образом, у нас должна получиться следующая схема сущности:

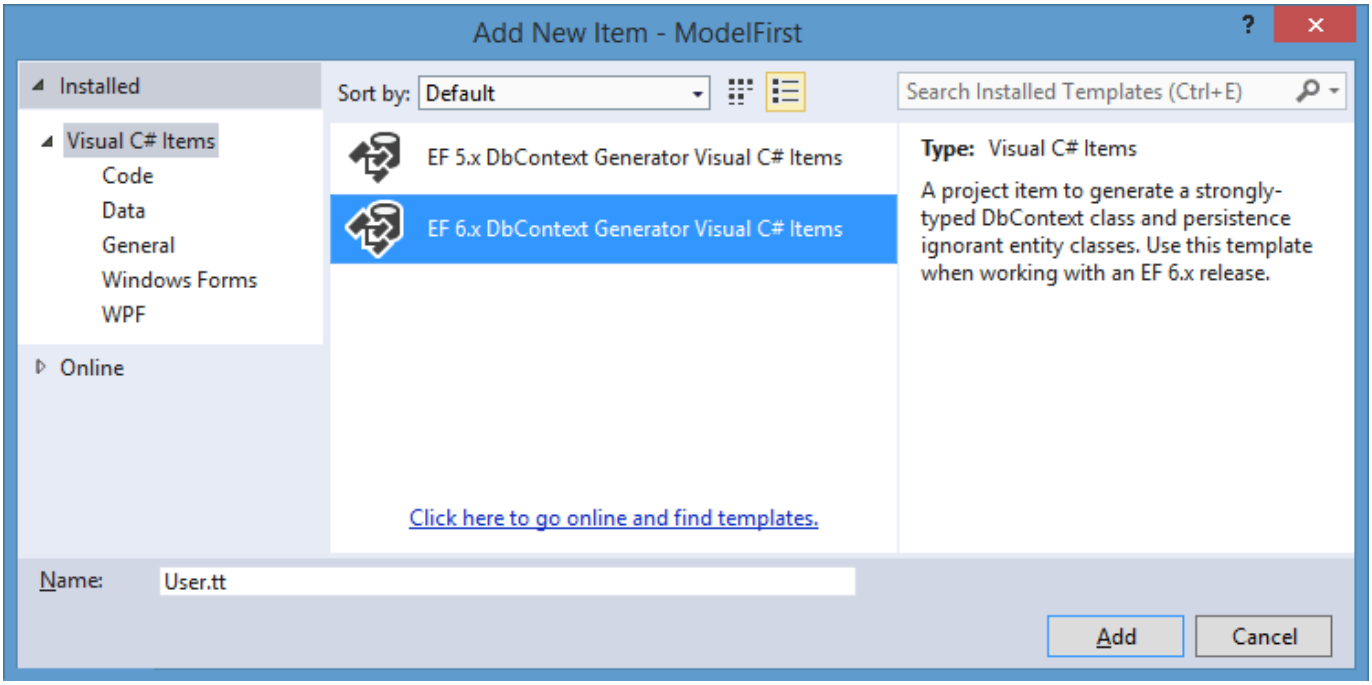


После создания диаграммы модели перестроим проект с помощью опции Rebuild.

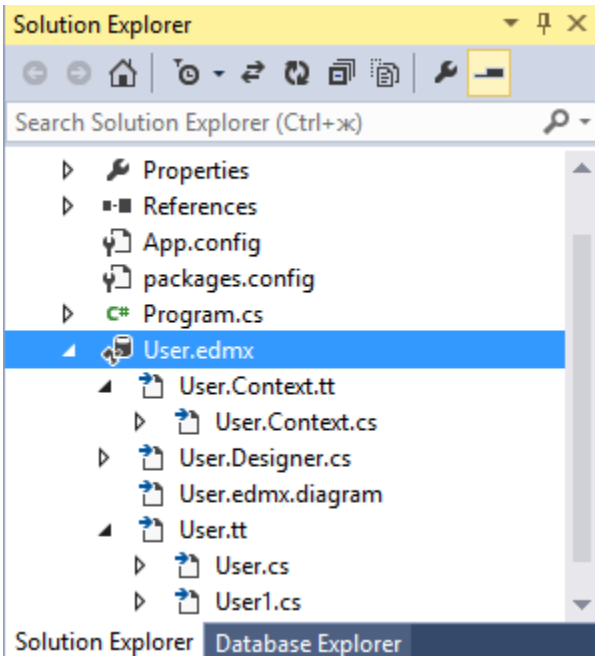
Теперь по модели мы можем сгенерировать код и базу данных. Вначале сгенерируем код модели. Для этого нажмем на диаграмму модели правой кнопкой мыши и выберем пункт **Add Code Generation Item**:



Далее нам будет предложено выбрать версию EF. Выберем шестую версию:



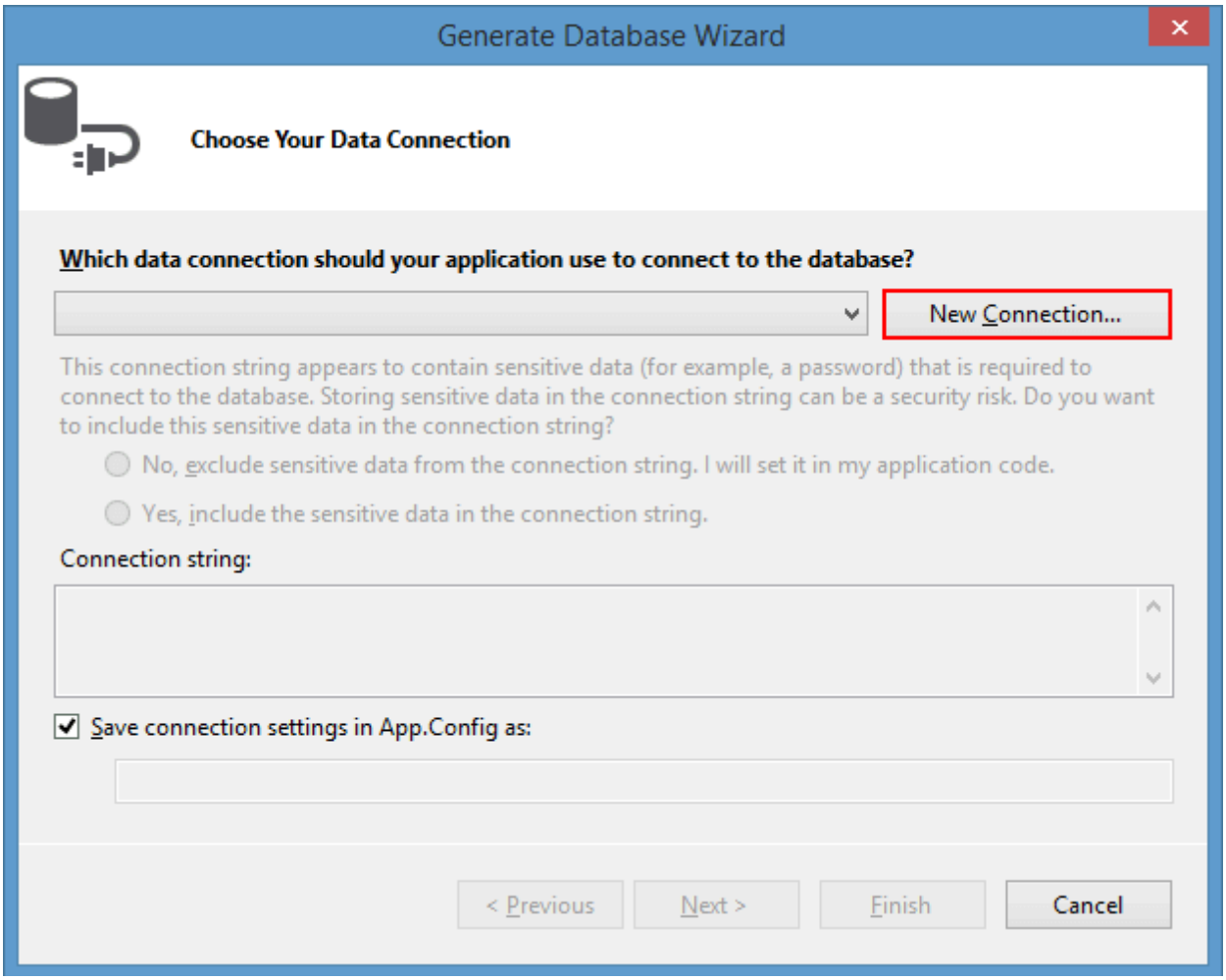
После этого в структуре проекта мы можем увидеть узел *User.tt*, который в качестве подузла будет содержать класс модели в файле User.cs:



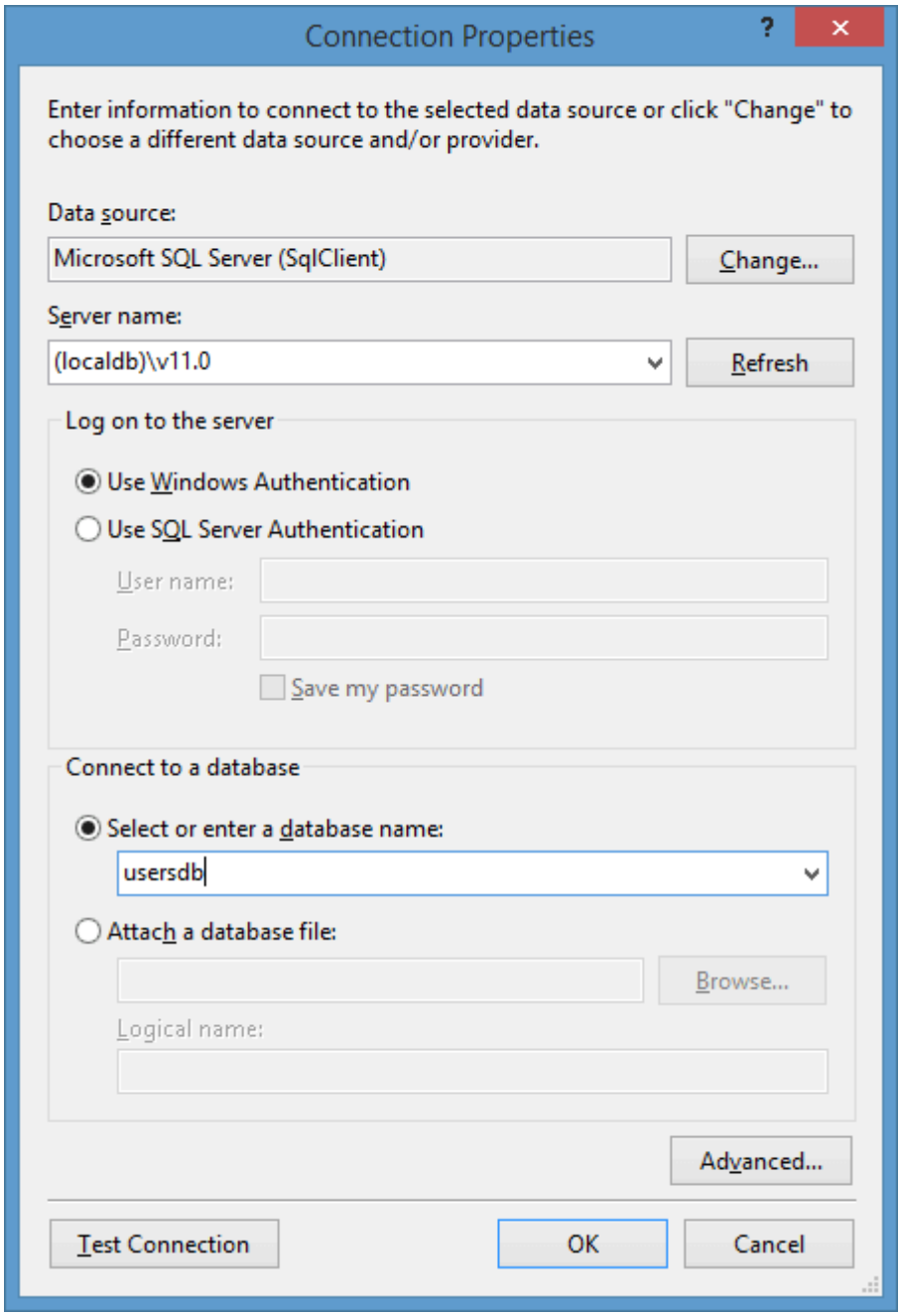
Также здесь мы можем найти файл контекста данных *User.Context.cs*, который выглядит в моем случае следующим образом:

```
namespace ModelFirst
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class UserContainer : DbContext
    {
        public UserContainer()
            : base("name=UserContainer")
        {
        }

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

        public virtual DbSet<User> Users { get; set; }
    }
}
```

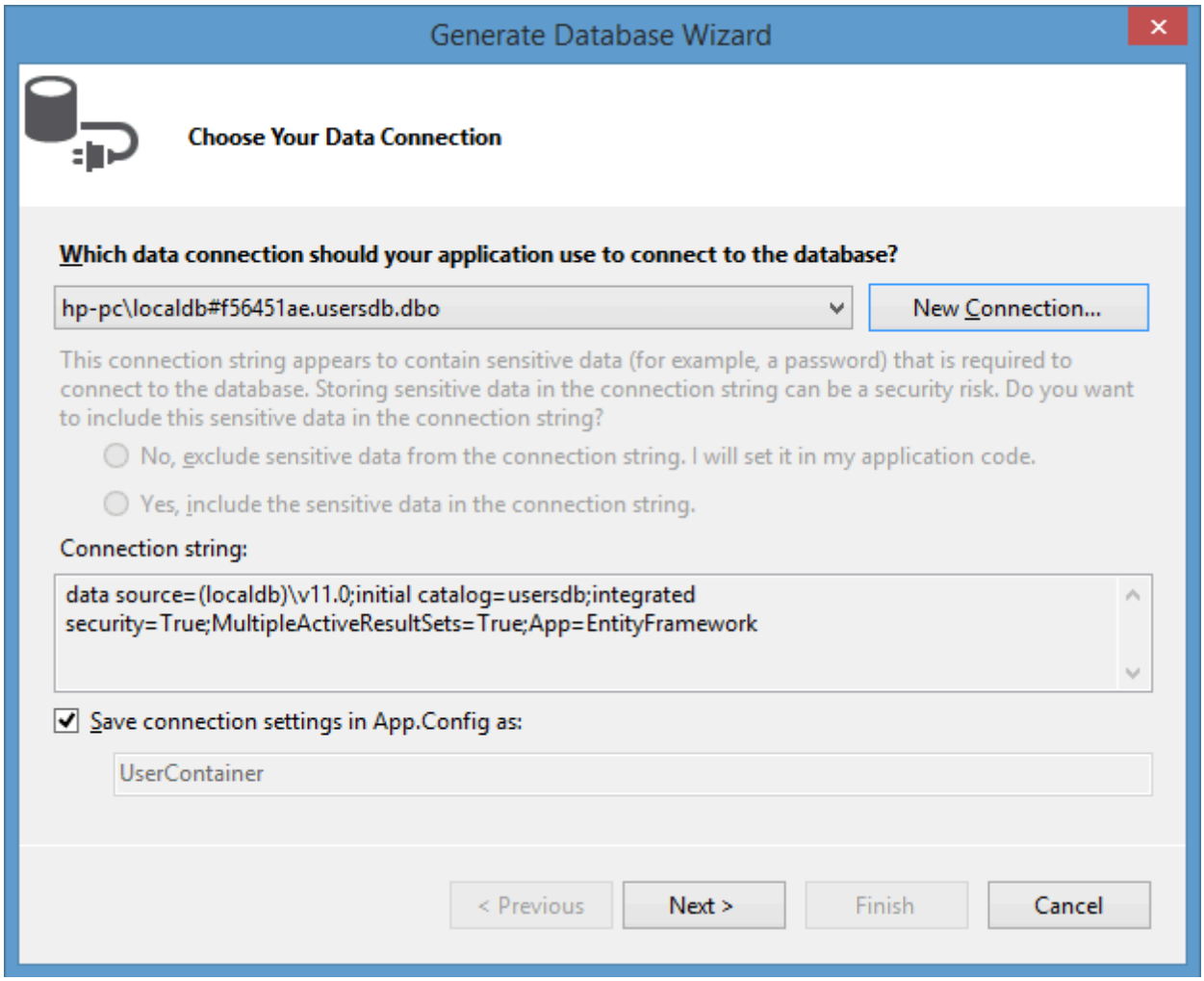
Теперь сгенерируем базу данных по нашей модели. Итак, нажмем на диаграмму модели правой кнопкой мыши и в выпадающем списке выберем **Generate Database from Model** (Сгенерировать базу данных по модели). Перед нами откроется мастер создания подключения.



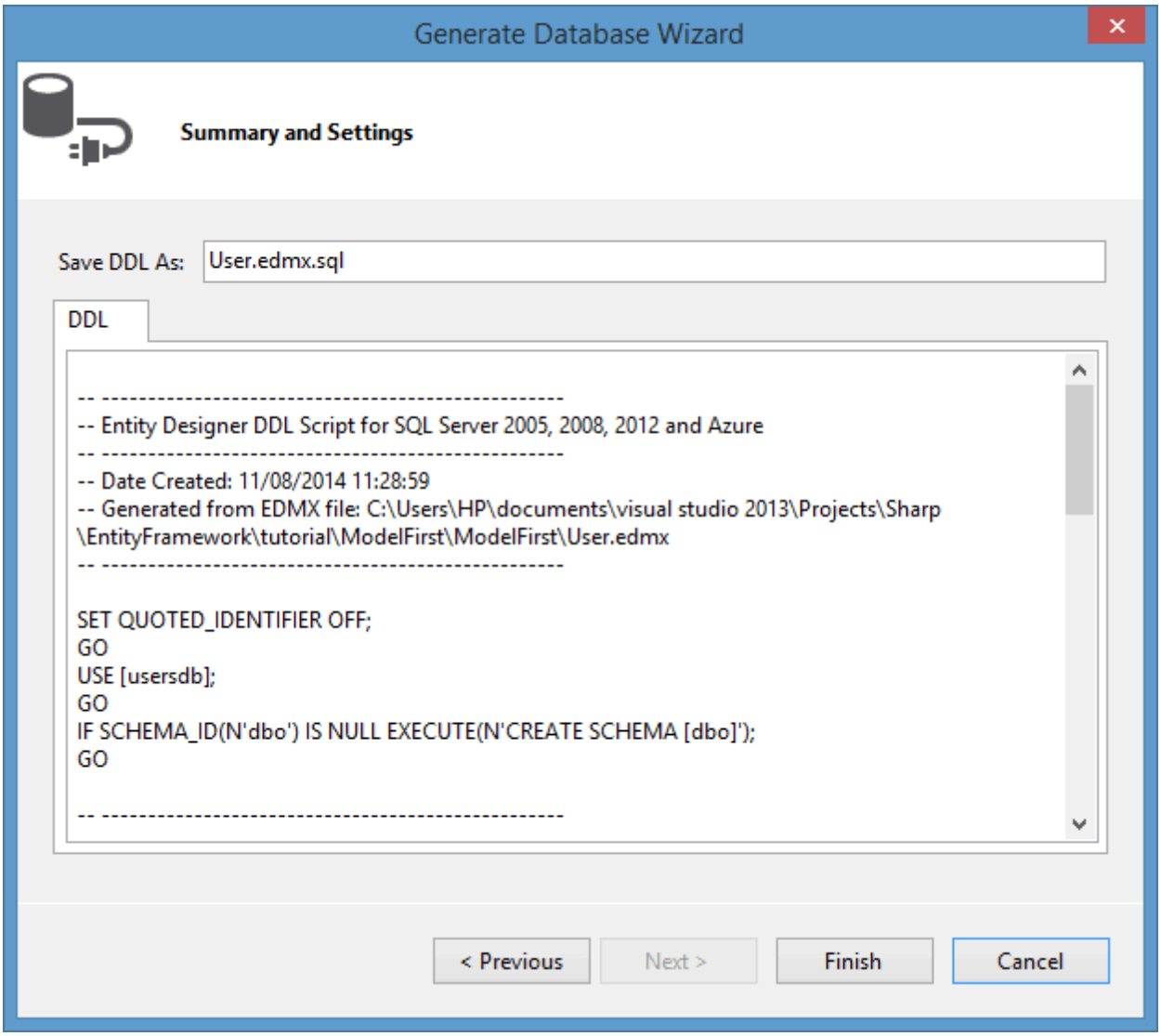
Нажмем на кнопку **New Connection** (Новое подключение). Далее открывается новое окно, где будет предложено настроить подключение и создать базу данных:



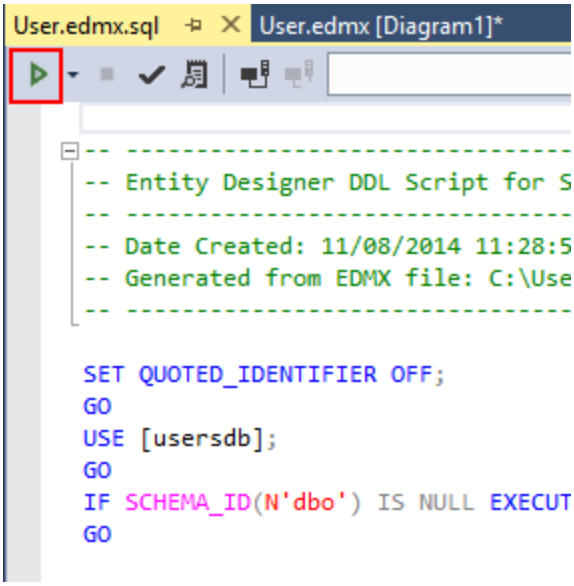
Здесь нам надо ввести имя сервера. А также название бд. В качестве имени базы данных введем usersdb, а в качестве сервера: **(localdb)\v11.0**. Нажмем OK и затем Visual Studio установит в качестве подключения модели только что созданную базу данных:



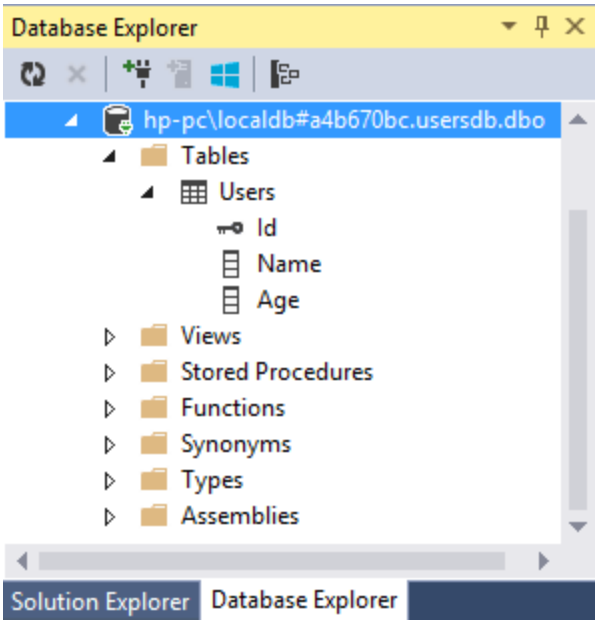
После этого будет сгенерирован скрипт базы данных:



Нажмем Finish (Готово). У нас автоматически откроется в Visual Studio файл скрипта User.edmx.sql. И в завершении нам надо будет запустить этот скрипт. Для этого нажмем в верхнем левом углу на зеленую кнопку Execute (Выполнить):



После этого в нижнем окне Visual Studio нам сообщит об успешном (или неуспешном) создании базы данных. Если мы откроем окно Database Explorer (его можно открыть, выбрав в меню View->Other Windows), то мы увидим нашу базу данных. А раскрыв узел, также увидим, что она содержит всю ту схему, которую мы определили в модели:



Это все была работа по созданию модели и базы данных. И в конце определим минимальный код для работы с базой данных:

```
static void Main(string[] args)
{
    using(UserContainer db = new UserContainer())
    {
        // добавление элементов
        db.Users.Add(new User {Name="Tom", Age=45});
        db.Users.Add(new User {Name="John", Age=22});
        db.SaveChanges();
        // получение элементов
        var users = db.Users;
        foreach (User u in users)
            Console.WriteLine("{0}.{1} - {2}", u.Id, u.Name, u.Age);
    }
    Console.Read();
}
```

Назад **Содержание** **Вперед**

Юмани:
410011174743222

Номер карты:
4048415020898850

Телеграмм

Вконтакте | Телеграм | Донаты/Помощь сайту

Contacts: metanit22@mail.ru