

Database First

Данное руководство устарело. Актуальное руководство: [Руководство по Entity Framework Core](#)

Последнее обновление: 31.10.2015



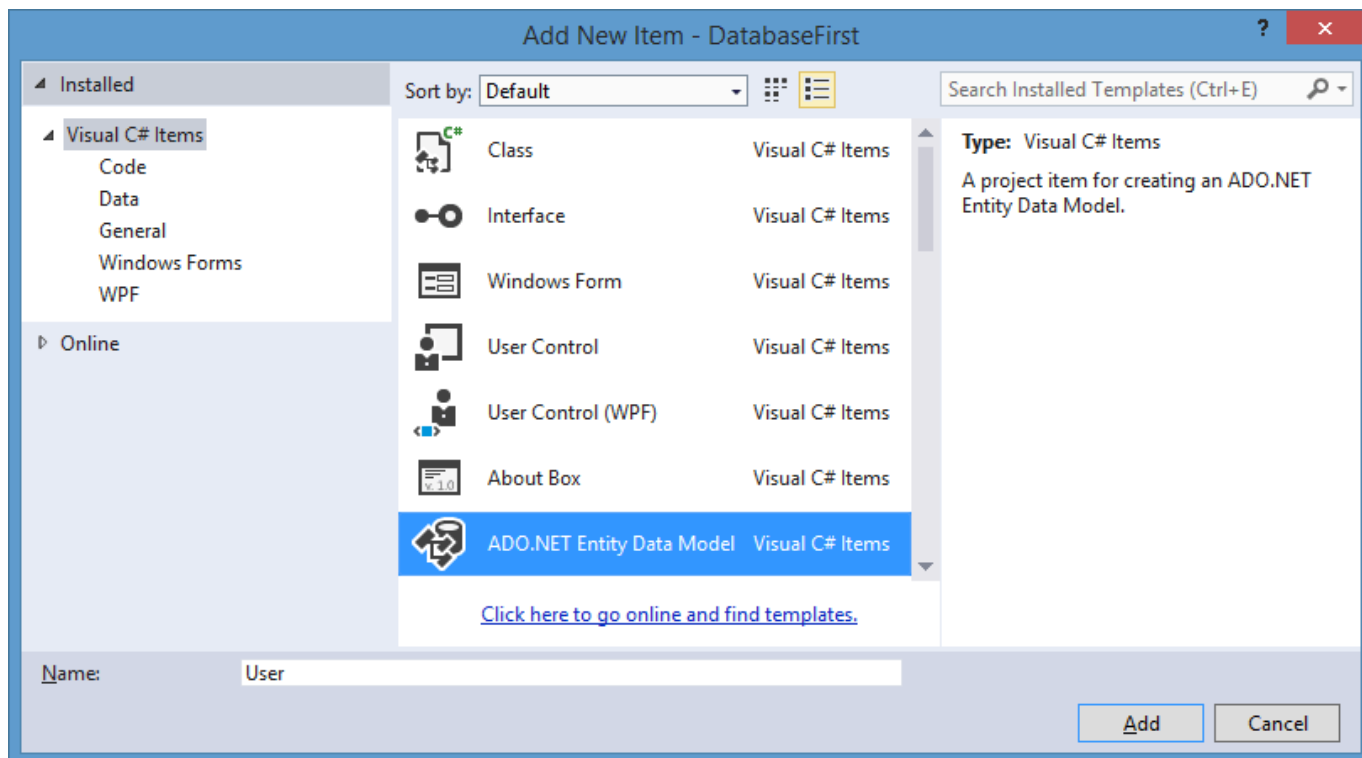
Database First был первым подходом, который появился в Entity Framework. Данный подход во многом похож на Model First и подходит для тех случаев, когда разработчик уже имеет готовую базу данных.

Чтобы Entity Framework мог получить доступ к базе данных, в системе должен быть установлен соответствующий провайдер. Так, Visual Studio уже поддерживает соответствующую инфраструктуру для СУБД MS SQL Server. Для остальных СУБД, например, MySQL, Oracle и других надо устанавливать соответствующие провайдеры. Список провайдеров для наиболее распространенных СУБД можно найти на странице [ADO.NET Data Providers](#).

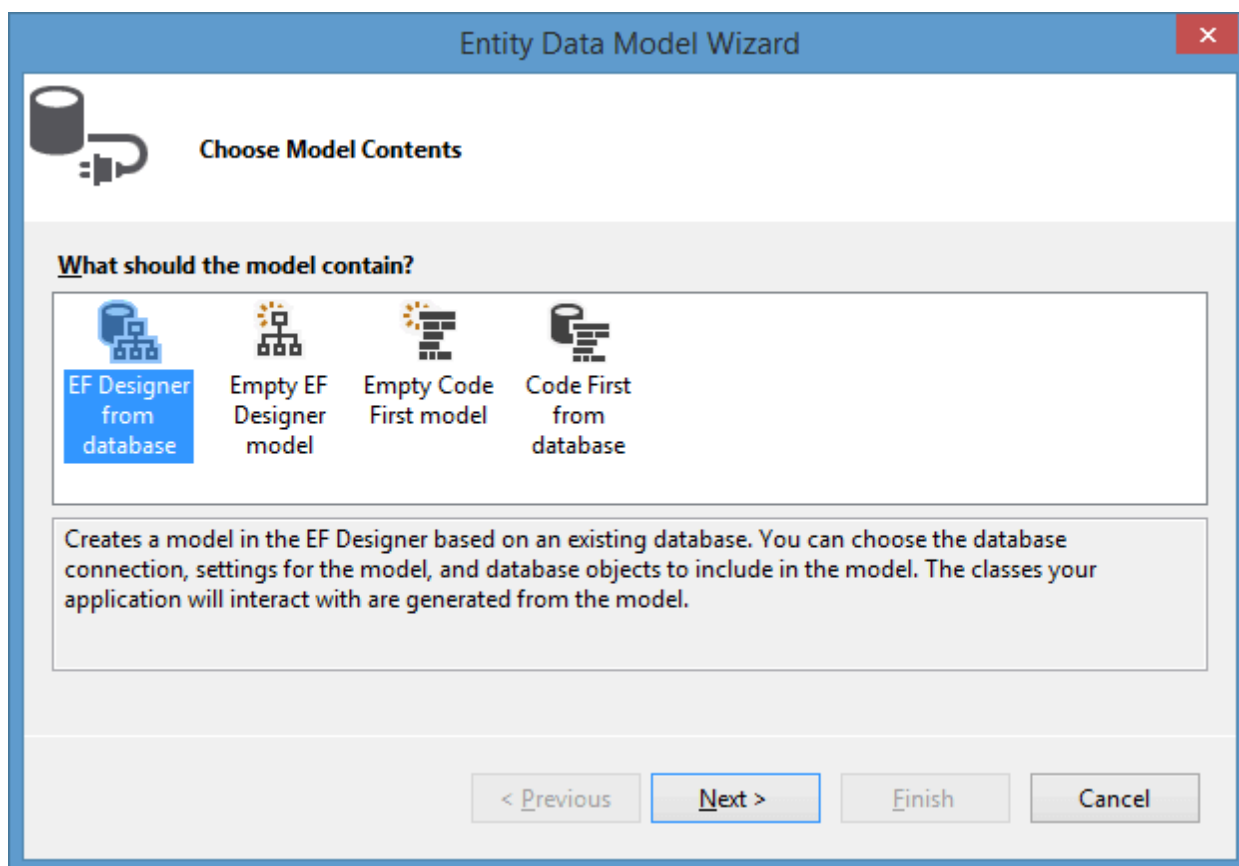
Итак, создадим новый проект по типу Console Application. Его функциональность будет той же самой, что и у предыдущих проектов, только подход к использованию Entity Framework будет отличаться.

После создания нового проекта, чтобы задействовать базу данных, нам надо ее иметь. Создадим новую бд или возьмем уже имеющуюся.

В Visual Studio в окне Solution Explorer нажмем на проект правой кнопкой мыши и выберем в **Add - > New Item**. Далее в появившемся окне добавления нового элемента выберем **ADO.NET Entity Data Model**. Дадим новому компоненту какое-либо название, например, **User**:

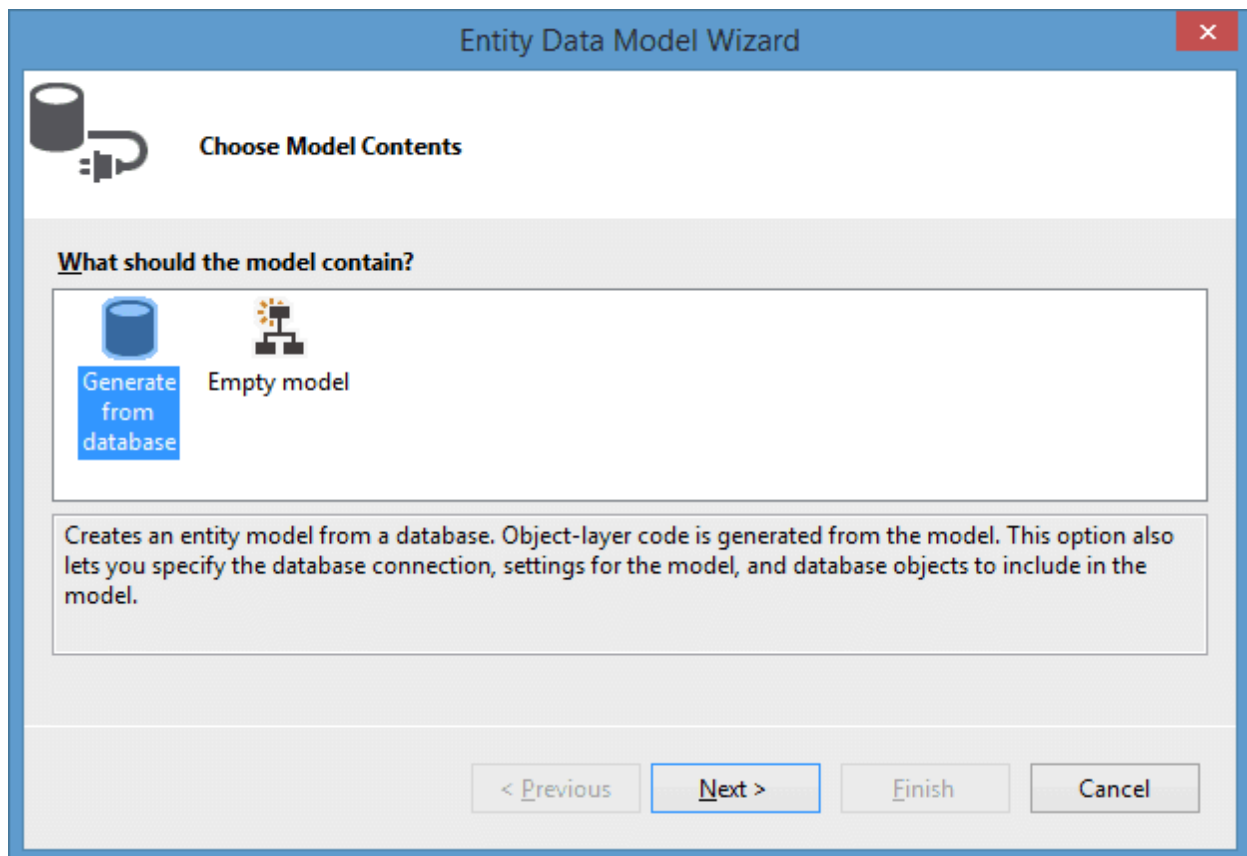


После этого нам откроется окно мастера создания модели. Если вы работаете с Visual Studio 2013 с пакетом обновления SP2, SP3, то откроется следующее окно мастера модели:



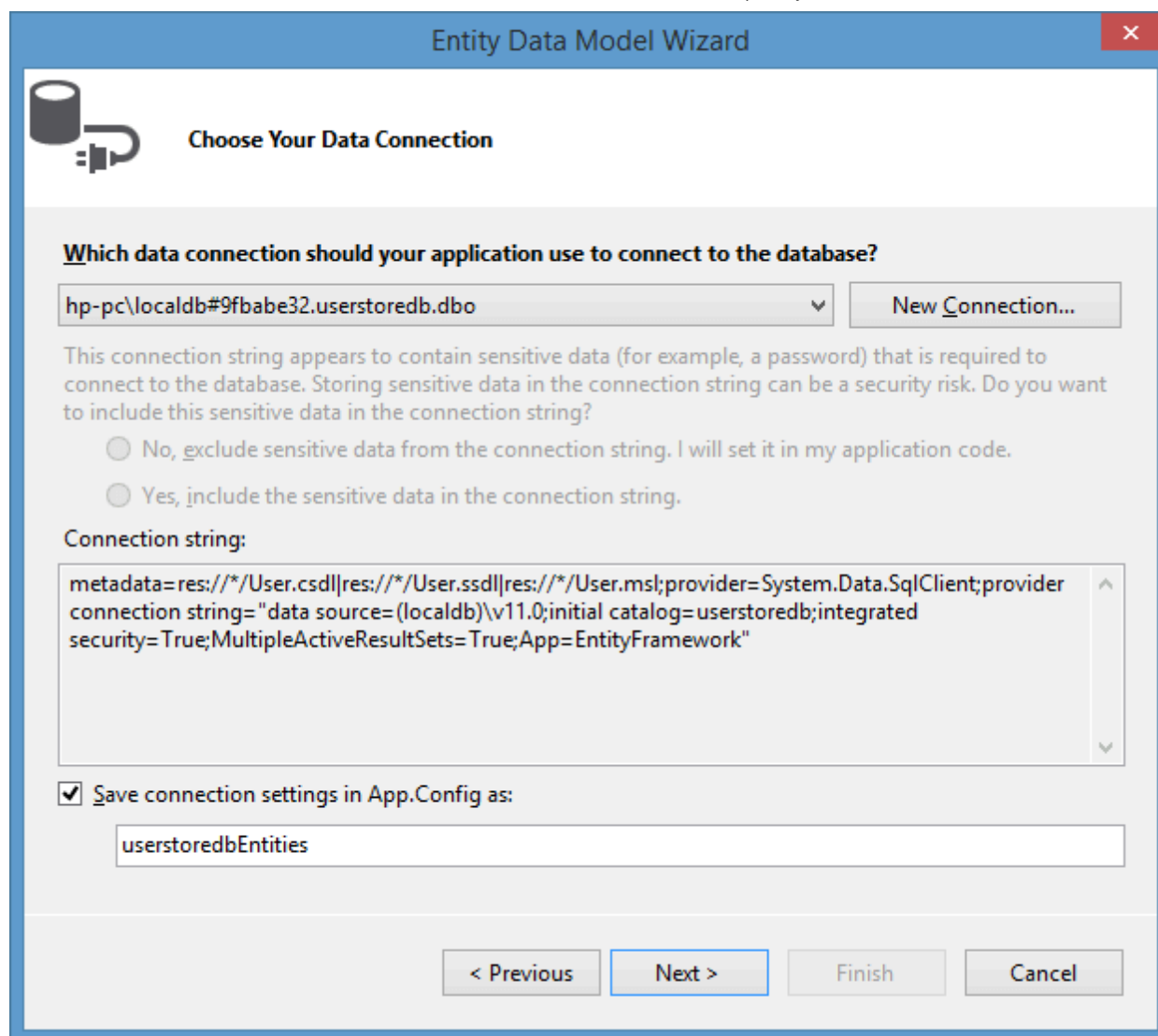
В нем нам надо выбрать опцию **EF Designer from database**.

Если нашей целевой средой является Visual Studio 2013 без пакетов обновлений, то окно мастера будет выглядеть следующим образом:



В этом случае надо выбрать пункт **Generate from database** (Создание модели по имеющейся базе данных).

Затем откроется окно следующего шага по созданию модели, на котором надо будет установить подключение к базе данных:

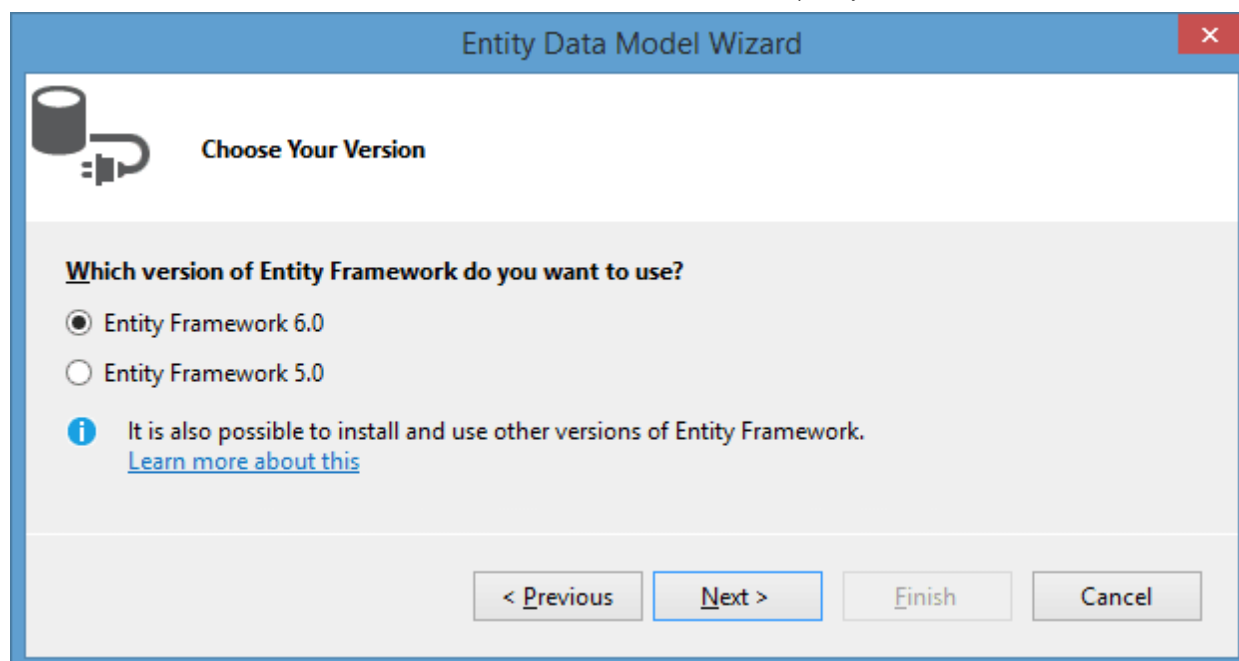


The image shows a screenshot of the 'Entity Data Model Wizard' dialog box, specifically the 'Choose Your Data Connection' step. The title bar reads 'Entity Data Model Wizard' with a close button. The main heading is 'Choose Your Data Connection'. Below this, a question asks: 'Which data connection should your application use to connect to the database?'. A dropdown menu shows the selected connection: 'hp-pc\localdb#9fbabe32.userstoredb.dbo'. To the right of the dropdown is a 'New Connection...' button. Below the dropdown, a warning message states: '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?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' (which is selected) and 'Yes, include the sensitive data in the connection string.'. Below this, the 'Connection string:' is displayed in a text box with the following content: 'metadata=res://*/User.csdl|res://*/User.ssdl|res://*/User.msl;provider=System.Data.SqlClient;provider connection string="data source=(localdb)\v11.0;initial catalog=userstoredb;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"'. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked. Below the checkbox is a text box containing 'userstoredbEntities'. At the very bottom, there are four buttons: '< Previous', 'Next >' (highlighted with a blue border), 'Finish', and 'Cancel'.

В выпадающем списке выберем одно из доступных подключений. Если в списке нет предпочтительных подключений, то можно нажать на кнопку **New Connection** и установить новое подключение.

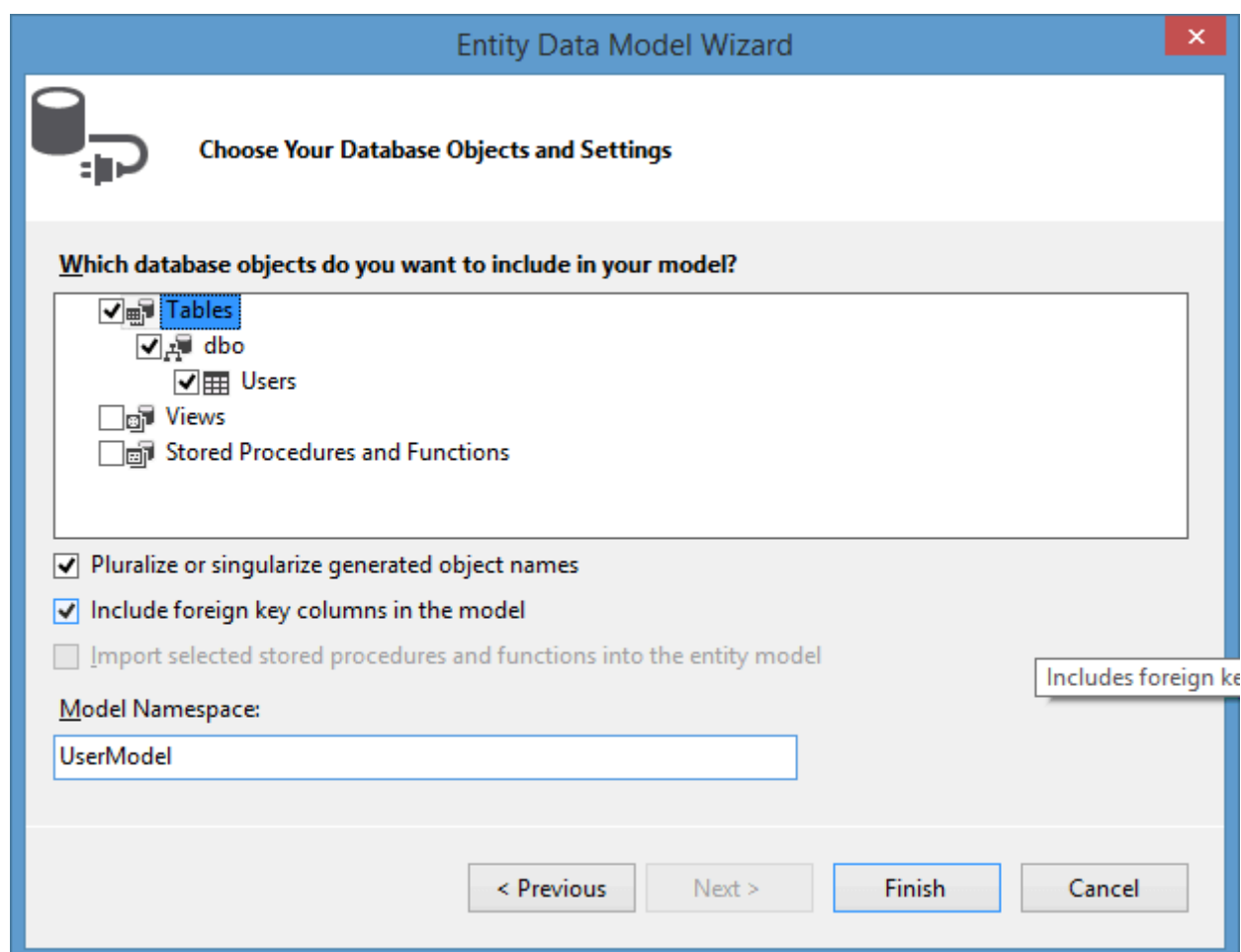
Также внизу указывается название **контекста данных**, который будет использоваться для доступа к данным. По умолчанию у меня контекст имеет название **userstoredbEntities**. Можно изменить, а можно и оставить.

Выбрав подключение, переходим к следующему шагу. Если у нас Visual Studio 2013 без пакетов обновления, то будет предложено также выбрать версию Entity Framework. Выберем шестую версию:



В версиях Visual Studio 2013 SP2, SP3 по умолчанию используется EF 6, поэтому этот шаг пропускается.

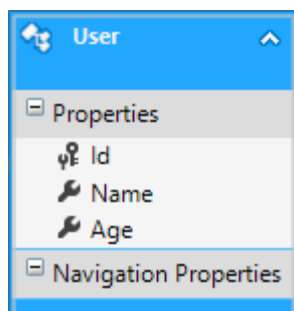
Далее Visual Studio извлекает всю информацию о базе данных:



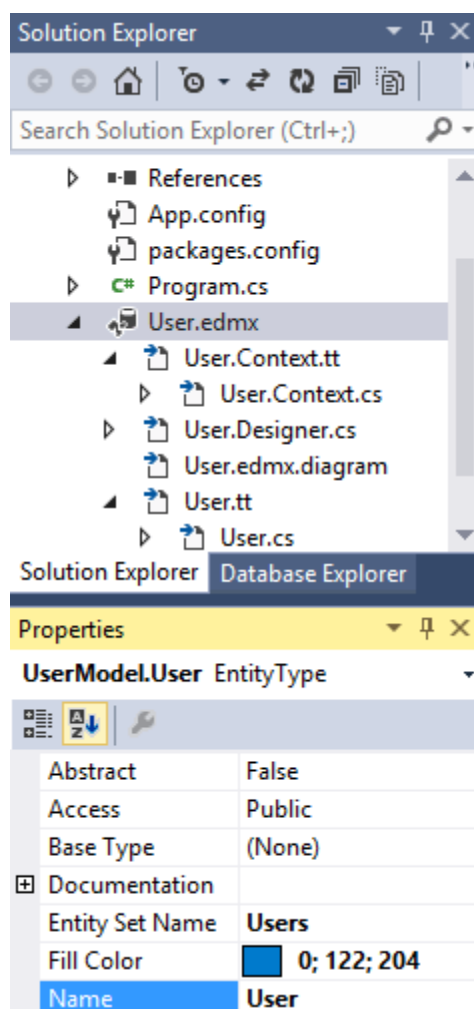
Раскроем узел Tables. Он отображает все таблицы, имеющиеся в базе данных. В моем случае имеется только одна таблица Users. Отметим все подузлы в ветке Tables.

В поле **Model Namespace** установим предпочтительное имя модели и нажмем Finish. После этого Entity Framework сгенерирует модель по базе данных и добавит ее в проект.

Visual Studio отобразит нам схему модели. В моем случае в бд есть только одна таблица Users, поэтому на схеме отображается только одна сущность User:



После выделения сущности в правом нижнем углу Visual Studio мы увидим свойства для этой сущности:



Свойство Name в окне свойств указывает на класс, которым будет представлена данная сущность (то есть классом User). А свойство **Entity Set Name** указывает на имя набора объектов (то есть свойство DbSet контекста данных) - в данном случае Users.

И теперь определим минимальный код для получения данных в коде приложения:

```
1 using(userstoredbEntities db = new userstoredbEntities())
2 {
3     var users = db.Users;
4     foreach (User u in users)
5         Console.WriteLine("{0}.{1} - {2}", u.Id, u.Name, u.Age);
6 }
```

[Назад](#) [Содержание](#) [Вперед](#)



Помощь сайту

[Помощь сайту](#)

Юмани:

410011174743222

Номер карты:

4048415020898850

[Телеграмм](#)

[Вконтакте](#) | [Телеграм](#) | [Донаты/Помощь сайту](#)

Contacts: metanit22@mail.ru

Copyright © Евгений Попов, metanit.com, 2025. Все права защищены.