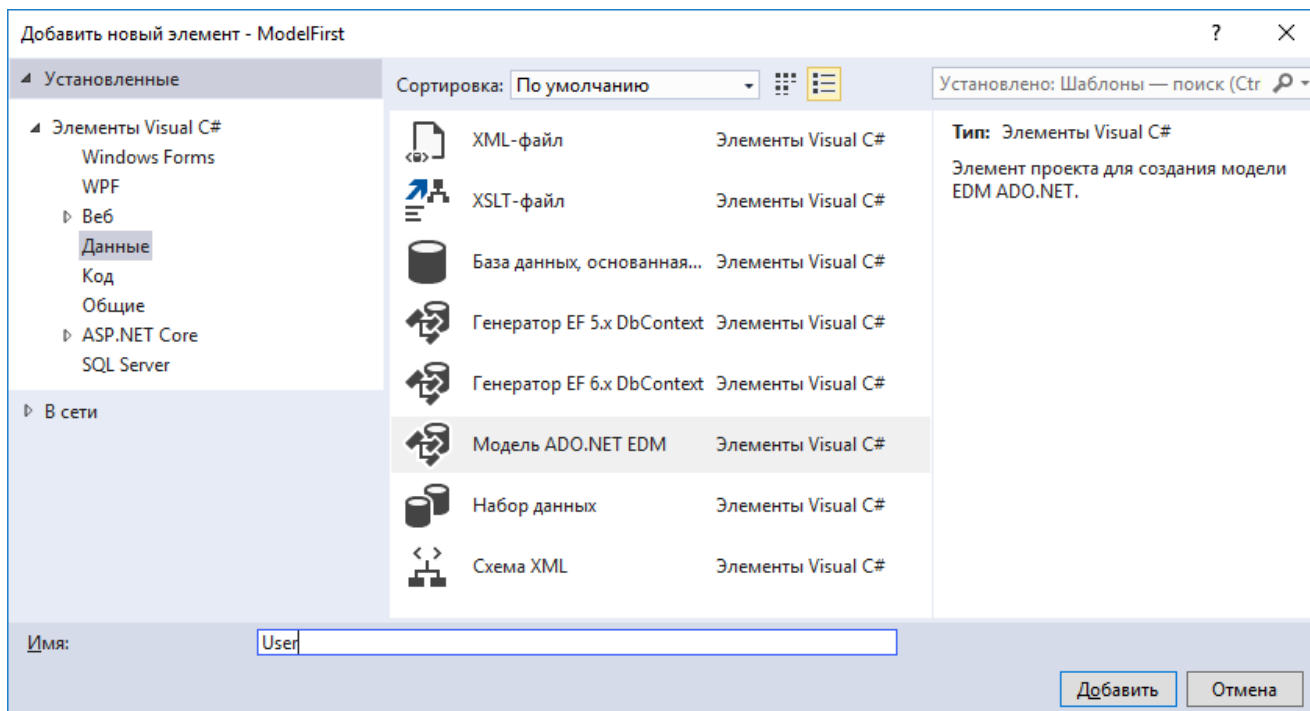


# Подход Model First

- 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.

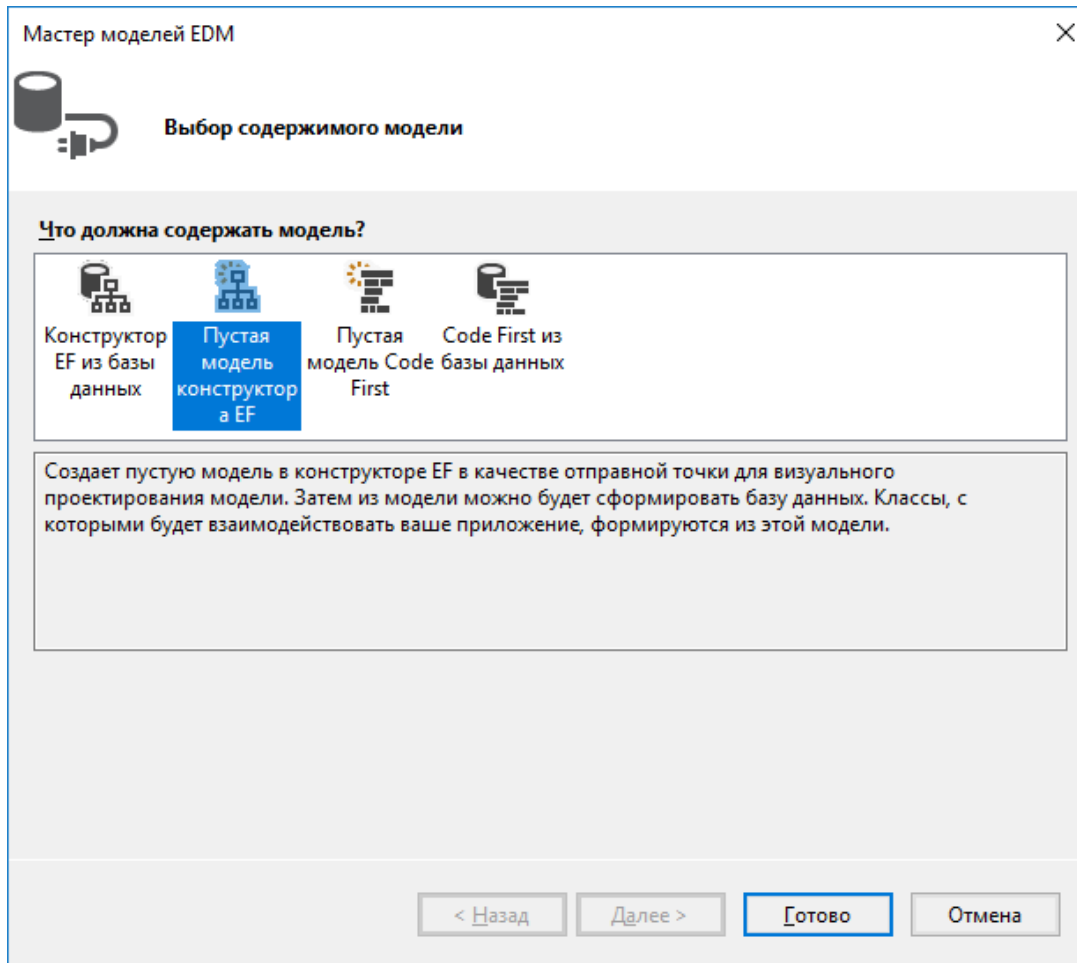
Создадим новый проект по типу Console Application.

После создания проекта добавим в него новый элемент ADO.NET EDM.



# Подход Model First

В мастере создания модели выберем Empty EF Designer Model



# Подход Model First

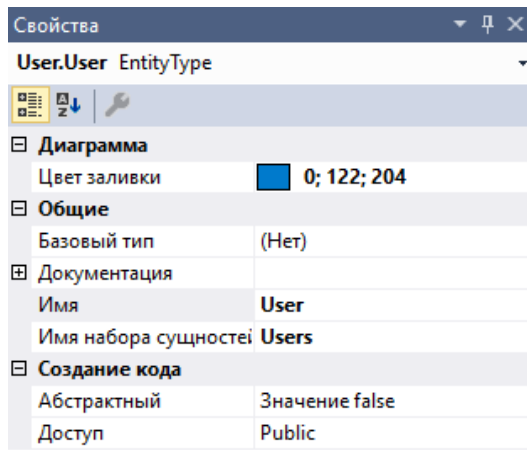
После нажатия кнопки Finish перед нами откроется пустое окно создания модели.

Конструктор Entity Data Model Designer позволяет визуализировать и проектировать модели EDM.

Создавайте новые сущности в модели, перетаскивая элементы из [Панель элементов](#).

Добавляйте существующие сущности и связи в эту диаграмму, перетаскивая их из [Браузер моделей](#).

Перетащим на это поле с панели Toolbox элемент Entity.

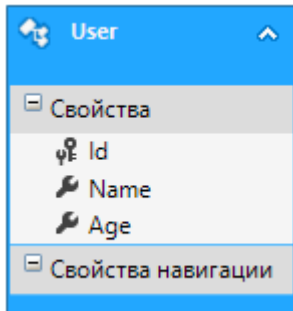


Теперь у нас на поле создания модели имеется небольшая схема будущей модели Entity1, в которой сейчас по умолчанию указано лишь одно поле - Id.

Изменим имя сущности на User и изменим значение свойства «Имя набора сущностей»

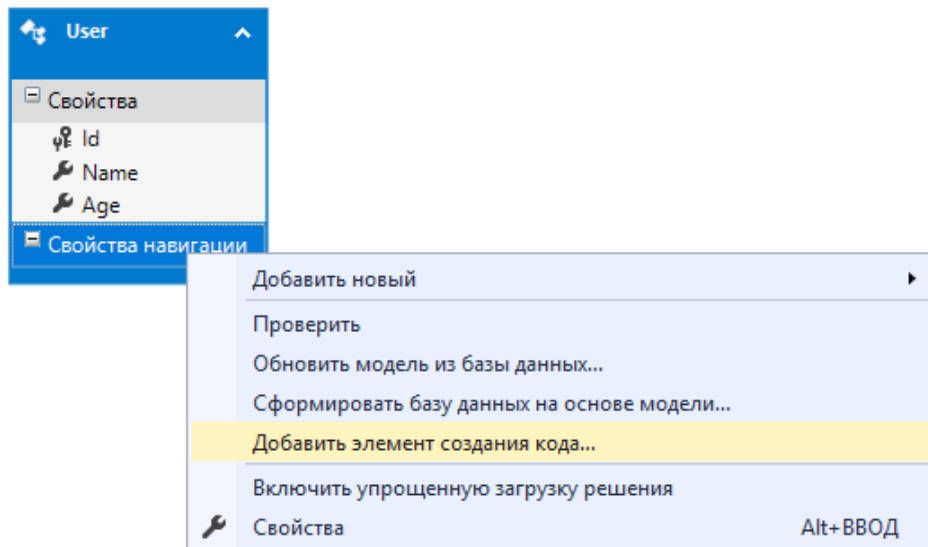
# Подход Model First

Далее создадим несколько свойств.



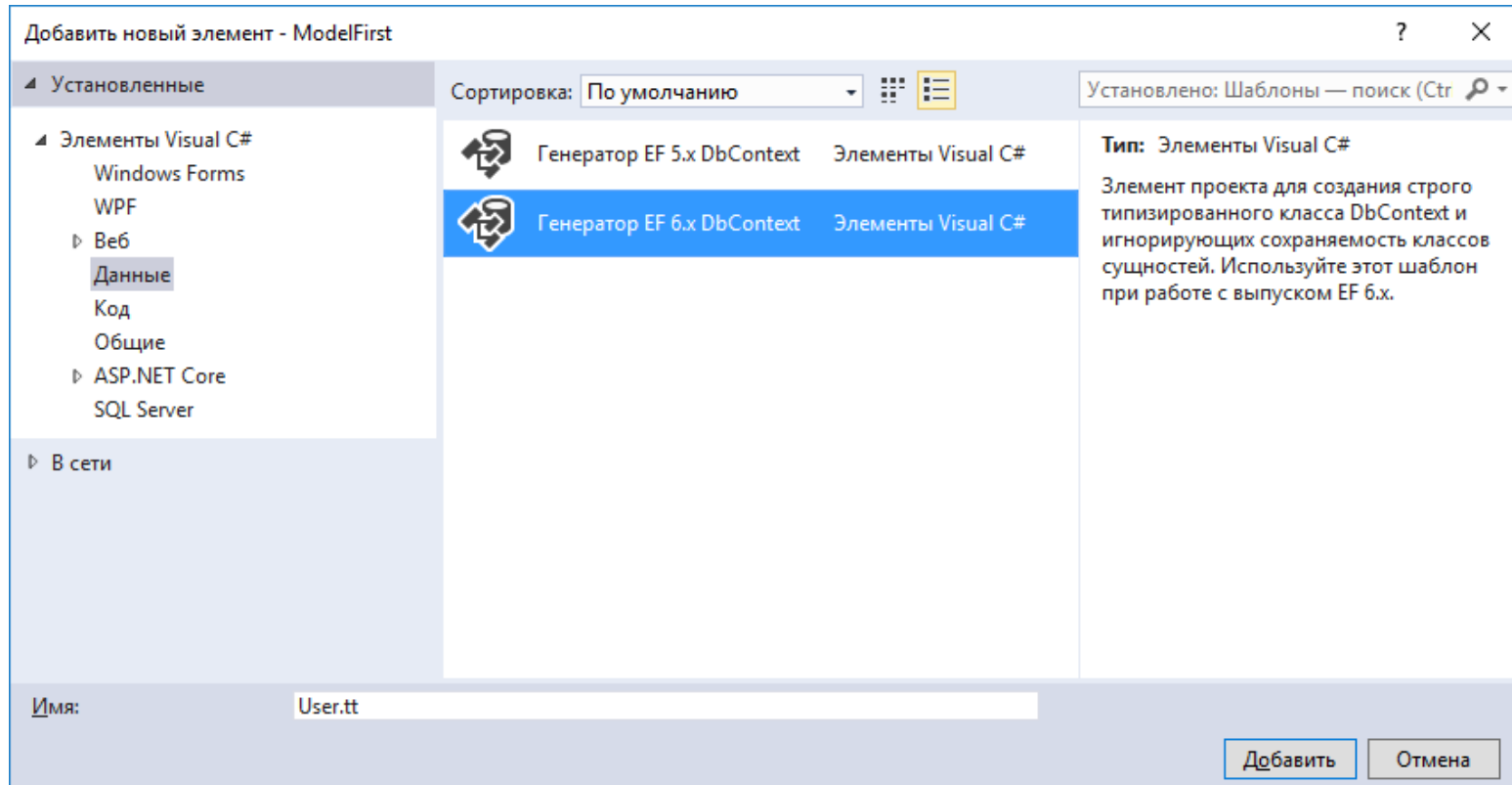
После создания диаграммы модели перестроим проект с помощью опции Rebuild.  
Теперь по модели можно сгенерировать код и базу данных.

Вначале сгенерируем код модели. Для этого нажмем на диаграмму модели правой кнопкой мыши и выберем пункт Add Code Generation Item.



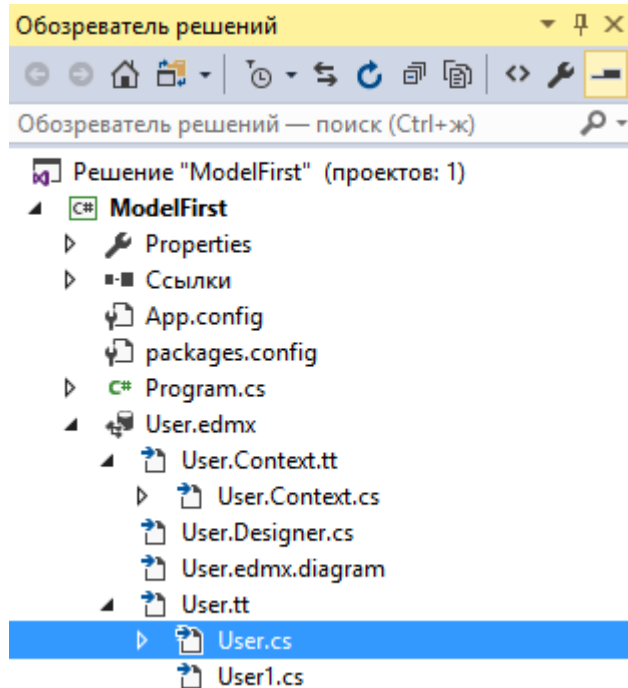
# Подход Model First

Будет предложено выбрать версию EF



# Подход Model First

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



```
namespace ModelFirst
{
    using System;
    using System.Collections.Generic;

    public partial class User
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int Age { get; set; }
    }
}
```

# Подход Model First

Также мы можем найти файл контекста данных *User.Context.cs*

```
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; }
    }
}
```

# Подход Model First

Теперь сгенерируем базу данных по нашей модели.

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.

Мастер формирования базы данных

Выбор подключения к данным

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

desktop-e4vsqbj\sqlexp2016.usersdb.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://*/User.csdl|res://*/User.ssdl|res://*/User.msl;provider=System.Data.SqlClient;provider connection string="data source=DESKTOP-E4VSQBJ\SQLEXP2016;initial catalog=usersdb;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

Сохранить параметры соединения в App.Config как:

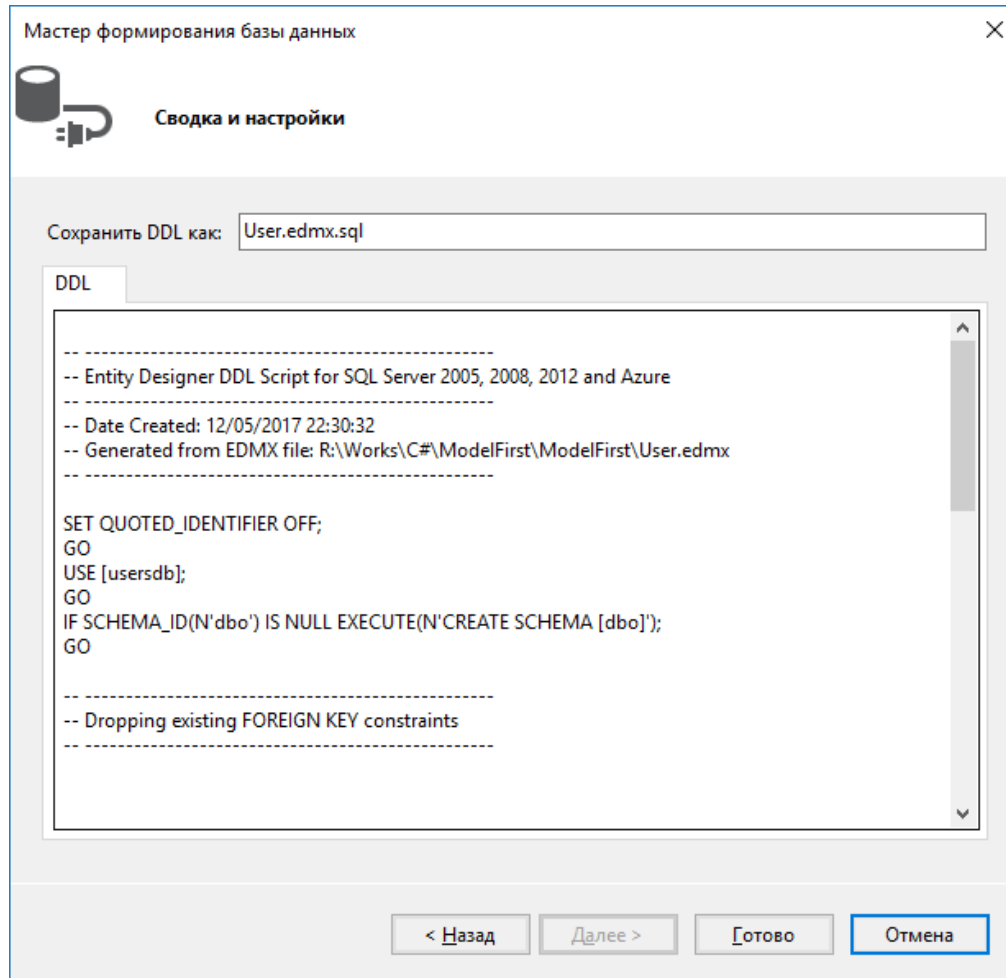
UserContainer

< Назад    Далее >    Готово    Отмена



# Подход Model First

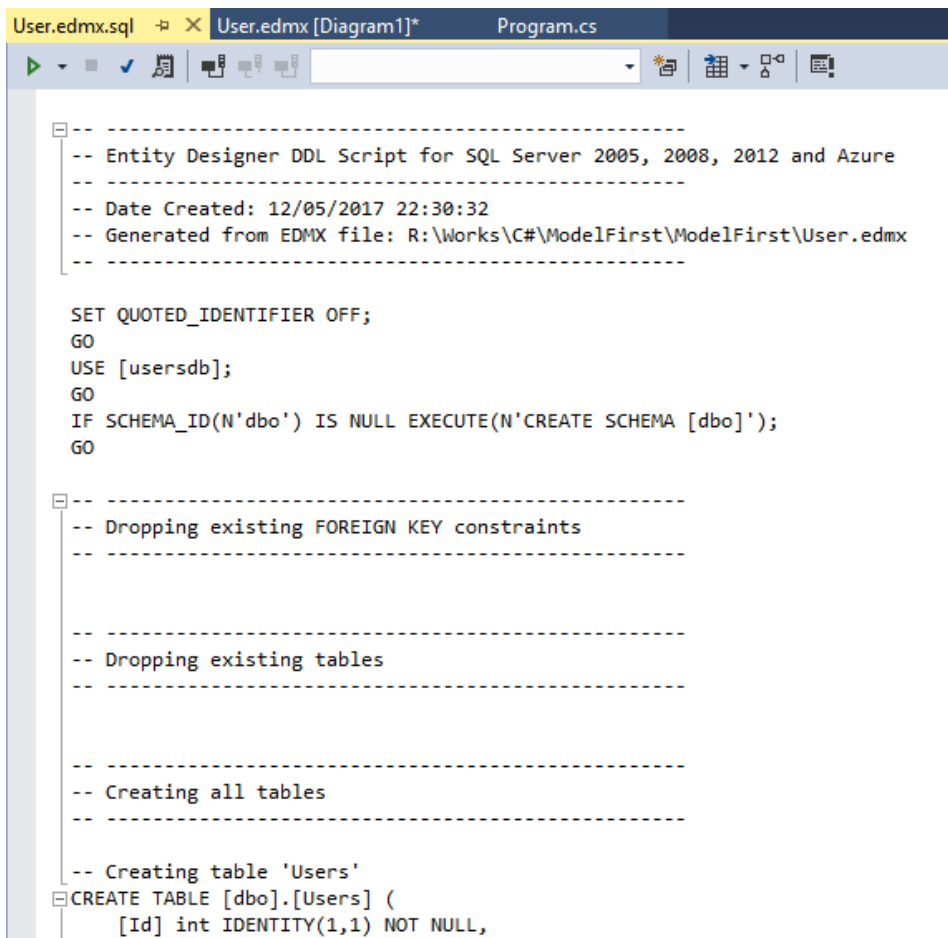
После этого будет сгенерирован скрипт базы данных:



# Подход Model First

После нажатия Finish (Готово) у нас автоматически откроется в Visual Studio файл скрипта User.edmx.sql.

В завершении нам надо будет запустить этот скрипт.

The image shows a screenshot of the Visual Studio IDE. The top toolbar includes a play button (run) and a checkmark (debug). The active window is titled 'User.edmx [Diagram1]\*' and contains a SQL script. The script is a DDL script generated by the Entity Designer. It includes comments about the generation date and file path, followed by SQL commands to set quoted identifiers, use the 'usersdb' database, and create a schema named 'dbo'. The script also contains sections for dropping existing foreign key constraints, dropping existing tables, and creating all tables. The final part of the script shows the creation of a table named 'Users' with an 'Id' column as an integer identity starting at 1 and not null.

```
User.edmx.sql  X  User.edmx [Diagram1]*  Program.cs
-- Entity Designer DDL Script for SQL Server 2005, 2008, 2012 and Azure
-- Date Created: 12/05/2017 22:30:32
-- Generated from EDMX file: R:\Works\C#\ModelFirst\ModelFirst\User.edmx

SET QUOTED_IDENTIFIER OFF;
GO
USE [usersdb];
GO
IF SCHEMA_ID(N'dbo') IS NULL EXECUTE(N'CREATE SCHEMA [dbo]');
GO

-- Dropping existing FOREIGN KEY constraints

-- Dropping existing tables

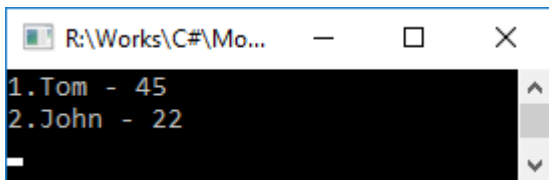
-- Creating all tables

-- Creating table 'Users'
CREATE TABLE [dbo].[Users] (
  [Id] int IDENTITY(1,1) NOT NULL,
```

# Подход Model First

В конце определим минимальный код для работы с базой данных.

```
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($"{u.Id}. {u.Name} - {u.Age}");
}
```



A screenshot of a console window with a black background and white text. The window title bar shows the path 'R:\Works\C#\Mo...'. The output consists of two lines: '1. Tom - 45' and '2. John - 22'. There is a cursor at the end of the second line.