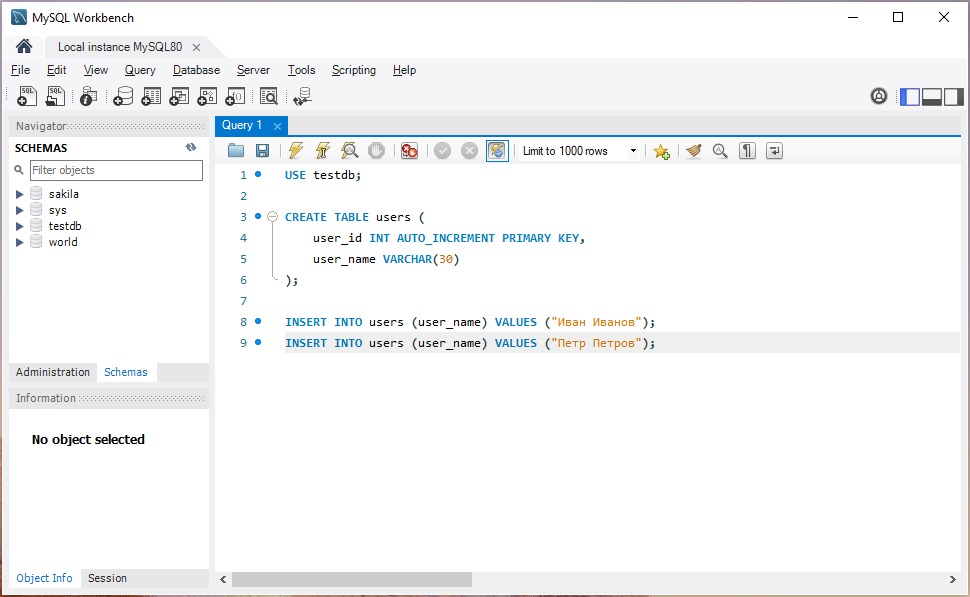
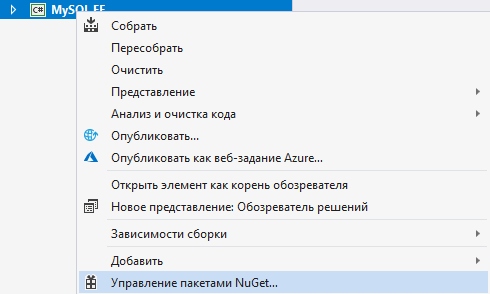
**Создание проекта с использованием Entity Framework и MySQL**

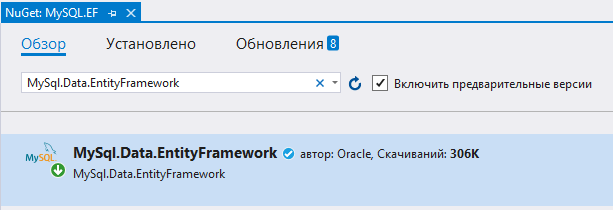
1. Для начала создадим таблицу в базе данных и наполним ее парой записей.



1. Затем создадим самый обычный проект консольного приложения .NET Framework и перейдем в «Управление пакетами NuGet».



1. Здесь установим пакет MySql.Data.EntityFramework – он потянет за собой необходимые пакеты, и они тоже установятся.



1. Откроем файл App.config и заменим элемент <entityFramework> со всем его содержимым на следующее:

<connectionStrings>

<add name="DataConnection" providerName="MySql.Data.MySqlClient"

connectionString=

"server=localhost;port=3306;database=testdb;uid=root;password=password"/>

</connectionStrings>

<entityFramework>

<defaultConnectionFactory type=

"System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework"/>

<providers>

<provider invariantName="MySql.Data.MySqlClient" type=

"MySql.Data.MySqlClient.MySqlProviderServices, MySql.Data.EntityFramework"/>

<provider invariantName="System.Data.SqlClient" type=

"System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer"/>

</providers>

</entityFramework>

1. Создадим файл User.cs со следующим содержимым:

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace MySQL.EF

{

[Table("users")]

internal class User

{

[Key]

[Column("user\_id")]

public int Id { get; set; }

[Column("user\_name")]

public string Name { get; set; }

}

}

1. Создадим файл DataContext.cs со следующим содержимым:

using System.Data.Common;

using System.Data.Entity;

namespace MySQL.EF

{

internal class DataContext : DbContext

{

public DataContext() : base("DataConnection")

{

Configure();

}

public DataContext(DbConnection connection, bool contextOwnsConnection)

: base(connection, contextOwnsConnection)

{

Configure();

}

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

private void Configure()

{

Configuration.ProxyCreationEnabled = true;

Configuration.LazyLoadingEnabled = true;

}

}

}

1. В файл Program.cs запишем следующее:

using System;

using System.Linq;

namespace MySQL.EF

{

internal class Program

{

public static void Main(string[] args)

{

using (DataContext context = new DataContext())

{

var users = context.Users.ToList();

foreach (var user in users)

Console.WriteLine("id {0}: {1}", user.Id, user.Name);

}

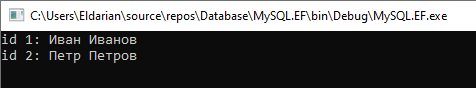
Console.ReadKey();

}

}

}

1. Запустим программу и увидим:



Убедимся, что пример работает и на этом инструкция по работе с Entity Framework с подключением MySQL завершена.

**Реализация CRUD-операций**

1. Ранее созданный класс DataContext – является посредником при работе с данными. Им очень удобно пользоваться, поэтому далее будут реализованы статические методы тестирования этого посредника в классе Program. Начнем с метода чтения таблицы:

private static void SelectAllTest(DataContext context)

{

IEnumerable<User> users = context.Users.ToList();

foreach (var user in users)

Console.WriteLine($"id {user.Id}: {user.Name}");

}

1. Подобным образом будет выглядеть метод чтения записи по идентификатору:

private static User SelectOneTest(DataContext context, int id)

{

User user = context.Users.Find(id);

Console.WriteLine($"id {user.Id}: {user.Name}");

return user;

}

1. Теперь зададим метод добавления записи:

private static void InsertTest(DataContext context, User user)

{

context.Users.Add(user);

context.SaveChanges();

}

1. Метод обновления записи:

private static void UpdateTest(DataContext context, User user)

{

context.Users.Find(user.Id).Name = user.Name;

context.SaveChanges();

}

1. Последним методом в классе DataRequester будет – метод удаления записи:

private static void DeleteTest(DataContext context, User user)

{

context.Users.Remove(user);

context.SaveChanges();

}

1. И, наконец, опишем точку входа в программу:

public static void Main(string[] args)

{

DataContext context = new DataContext();

Console.WriteLine("Выборка всех пользователей:");

SelectAllTest(context);

Console.WriteLine("\nВыборка пользователя по id 4:");

User user = SelectOneTest(context, 4);

Console.WriteLine("\nДобавление пользователя:");

InsertTest(context, new User { Name = "Фридрих Ницше" });

SelectAllTest(context);

Console.WriteLine("\nОбновление пользователя:");

user.Name = "Чарльз Дарвин";

UpdateTest(context, user);

SelectAllTest(context);

Console.WriteLine("\nУдаление пользователя:");

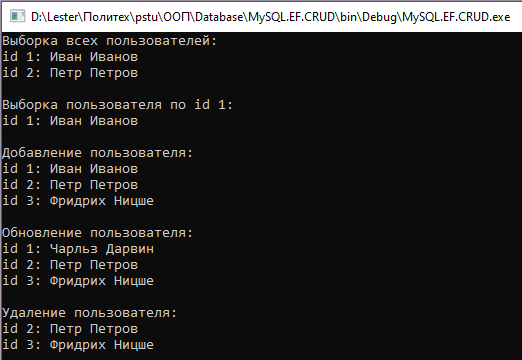
DeleteTest(context, user);

SelectAllTest(context);

Console.ReadKey();

}

1. Запустим полученную программу:



На этом реализация и демонстрация CRUD-операция в Entity Framework с MySQL завершены.