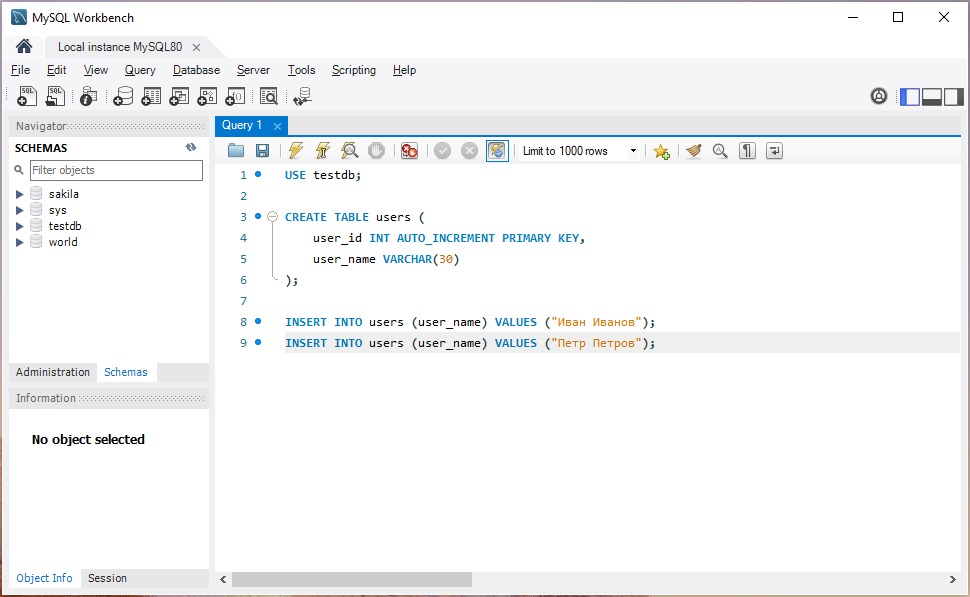
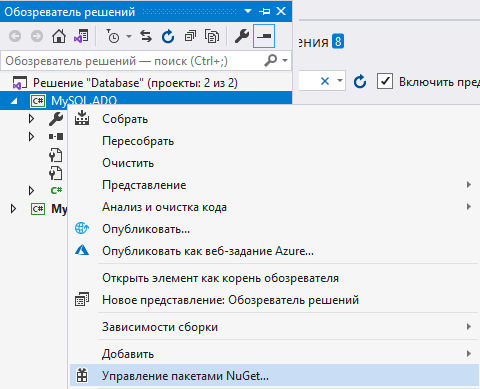
**Создание проекта с использованием ADO.NET и MySQL**

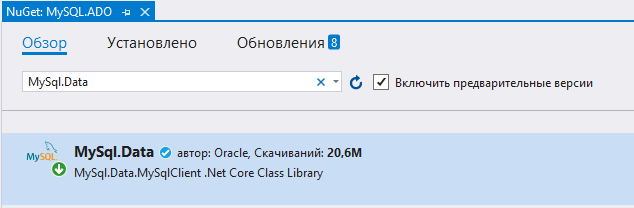
1. Для начала создадим таблицу в базе данных и наполним ее парой записей.



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



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



1. Введем следующий код в файл Program.cs:

using System;

using MySql.Data.MySqlClient;

namespace MySQL.ADO

{

internal class Program

{

private static string host = "localhost"; // Имя хоста

private static string database = "testdb"; // Имя базы данных

private static string user = "root"; // Имя пользователя

private static string password = "password"; // Пароль пользователя

private static string ConnectionString // Строка подключения

=> "Database=" + database

+ ";Datasource=" + host

+ ";User=" + user

+ ";Password=" + password;

public static void Main(string[] args)

{

MySqlConnection connection = new MySqlConnection(ConnectionString);

MySqlCommand command = connection.CreateCommand();

command.CommandText = "SELECT \* FROM users;";

connection.Open();

MySqlDataReader reader = command.ExecuteReader();

while (reader.Read())

Console.WriteLine("id {0}: {1}", reader.GetInt64(0), reader.GetString(1));

connection.Close();

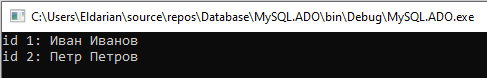
Console.ReadKey();

}

}

}

1. Запустим программу и увидим:



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

**Реализация CRUD-операций**

1. Для удобства работы создадим класс User:

internal class User

{

public int Id { get; set; }

public string Name { get; set; }

}

1. Работать с базой данных будет очень удобно через посредника – пусть таким посредником будет класс DataRequester, тогда запишем в него следующее:

internal class DataRequester

{

private static string host = "localhost"; // Имя хоста

private static string database = "testdb"; // Имя базы данных

private static string user = "root"; // Имя пользователя

private static string password = "19841986"; // Пароль пользователя

private static string ConnectionString // Строка подключения

=> "Database=" + database

+ ";Datasource=" + host

+ ";User=" + user

+ ";Password=" + password;

//…

}

1. Теперь определим первый метод – метод получения всех записей из таблицы:

public ICollection<User> SelectAll()

{

ICollection<User> result = new HashSet<User>();

MySqlConnection connection = new MySqlConnection(ConnectionString);

MySqlCommand command = connection.CreateCommand();

command.CommandText = "SELECT \* FROM users;";

connection.Open();

MySqlDataReader reader = command.ExecuteReader();

while (reader.Read())

result.Add(new User { Id = reader.GetInt32(0), Name = reader.GetString(1)});

connection.Close();

return result;

}

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

public User SelectOne(int id)

{

User result = null;

MySqlConnection connection = new MySqlConnection(ConnectionString);

MySqlCommand command = connection.CreateCommand();

command.CommandText = $"SELECT \* FROM users WHERE user\_id={id};";

connection.Open();

MySqlDataReader reader = command.ExecuteReader();

if (reader.Read())

result = new User { Id = reader.GetInt32(0), Name = reader.GetString(1) };

connection.Close();

return result;

}

1. Теперь зададим метод добавления записи:

public void Insert(User user)

{

MySqlConnection connection = new MySqlConnection(ConnectionString);

MySqlCommand command = connection.CreateCommand();

command.CommandText = $"INSERT INTO users (user\_name) VALUES (\"{user.Name}\");";

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

1. Метод обновления записи:

public void Update(User user)

{

MySqlConnection connection = new MySqlConnection(ConnectionString);

MySqlCommand command = connection.CreateCommand();

command.CommandText = $"UPDATE users SET user\_name=\"{user.Name}\" WHERE user\_id={user.Id};";

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

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

public void Delete(User user)

{

MySqlConnection connection = new MySqlConnection(ConnectionString);

MySqlCommand command = connection.CreateCommand();

command.CommandText = $"DELETE FROM users WHERE user\_id={user.Id};";

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

1. Написанный функционал необходимо протестировать, для этого перейдем в класс Program и снабдим его статическими методами:

private static void SelectAllTest(DataRequester requester)

{

IEnumerable<User> users = requester.SelectAll();

foreach (var user in users)

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

}

private static User SelectOneTest(DataRequester requester, int id)

{

User user = requester.SelectOne(id);

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

return user;

}

private static void InsertTest(DataRequester requester, User user)

{

requester.Insert(user);

}

private static void UpdateTest(DataRequester requester, User user)

{

requester.Update(user);

}

private static void DeleteTest(DataRequester requester, User user)

{

requester.Delete(user);

}

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

public static void Main(string[] args)

{

DataRequester requester = new DataRequester();

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

SelectAllTest(requester);

Console.WriteLine("\nВыборка пользователя по id 1:");

User user = SelectOneTest(requester, 1);

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

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

SelectAllTest(requester);

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

user.Name = "Чарльз Дарвин";

UpdateTest(requester, user);

SelectAllTest(requester);

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

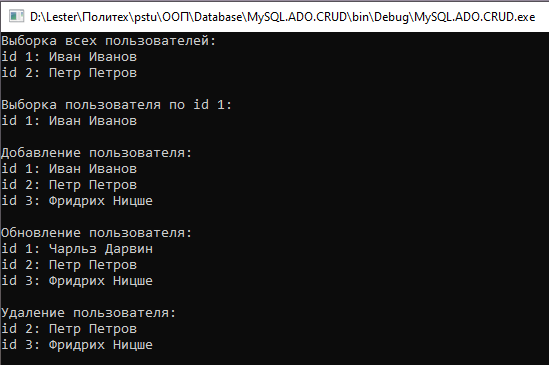
DeleteTest(requester, user);

SelectAllTest(requester);

Console.ReadKey();

}

1. Запустим полученную программу:



На этом реализация и демонстрация CRUD-операция в ADO.NET с MySQL завершены.