

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по практической работе №3

«DataReader. Параметризованные SQL команды.»

по дисциплине

Базы данных

РУКОВОДИТЕЛЬ:

(подпись)

профессор каф. ВСТ
Мисевич П. В.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Сухоруков В.А.
(фамилия, и.,о.)
19-В-2
(шифр группы)

Работа защищена «__» _____

С оценкой _____

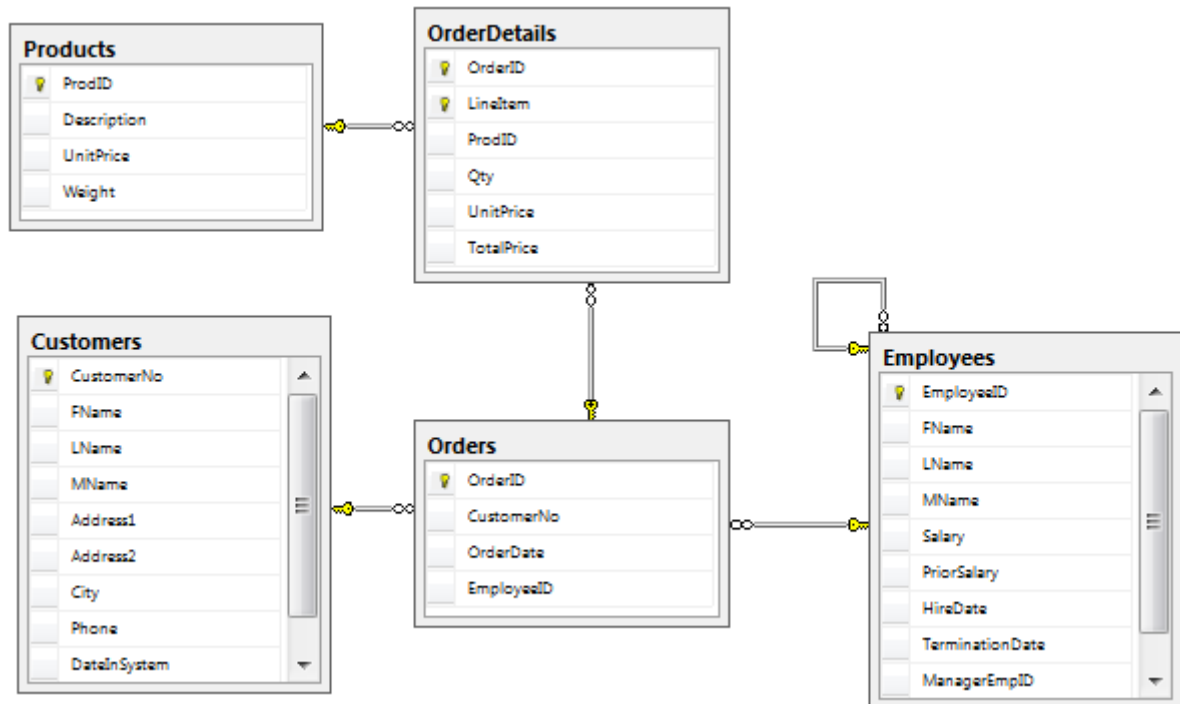
Нижний Новгород 2021

Цель

Изучить принцип работы объекта `DataReader`. Изучить способы работы с параметризованными командами.

Ход выполнения

База данных `ShopDB` состоит из таблиц `Products`, `OrderDetails`, `Orders`, `Customers`, `Employees`.



Содержание таблицы `Customers`:

CustomerNo	FName	LName	MName	Address1	Address2	City	Phone	DateInSystem
1	Иван*+++++	Круковский	Петрович	Лужная 15	NULL	Харьков	3222233322	2009-11-20
2	Виктор	Дурнев	Викторович	Зелинская 10	NULL	Киев	(067)4242132	2010-08-03
3	Зигмунд	Унакий	федорович	Дихтяревская 5	NULL	Киев	(092)7612343	2010-08-17
4	Виталий	Левченко	Викторович	Глушценка 5	Драйзера 12	Киев	(053)3456788	2010-08-20
5	Олег	Выжлецов	Евстафьевич	Киевская 3	Одесская 8	Чернигов	(044)2134212	2010-09-18
20	Николай	Андреев	Петрович	Лужная 9	NULL	Харьков	(099)4578596	2009-06-16

DataReader

`DataReader` представляет собой поток записей только на чтение, по которому можно передвигаться только вперед, последовательно от записи к записи. Работа с ним очень напоминает чтение данных из последовательного файла.

В процессе чтения в каждый момент времени в оперативной памяти клиентского компьютера находится одна-единственная текущая запись. Поэтому `DataReader` требует очень мало памяти клиента, даже если общий объем выборки данных велик.

Кроме того, каждый объект `DataReader` занимает для работы отдельное соединение (`Connection`), так что при необходимости одновременно читать данные из нескольких источников вам придется создать несколько одновременно открытых соединений с источником данных. После того, как на данном соединении был открыт `DataReader`, никакие другие операции не могут быть выполнены с данным соединением вплоть до закрытия `DataReader` методом `Close`.

Для чтения информации о всех клиентах с помощью `DataReader` и вывода её на консоль в табличном виде можно использовать следующую программу:

```
string conStr = @"Data Source=.\SQLEXPRESS; Initial Catalog=ShopDB; Integrated Security=True;"; // создание строки подключения
SqlConnection connection = new SqlConnection(conStr); // создание подключения
connection.Open();

SqlCommand cmd = new SqlCommand("SELECT * FROM Customers", connection); // построение команды, возвращающей данные в табличном представлении
SqlDataReader reader = cmd.ExecuteReader();

// с помощью объекта SqlDataReader можно просматривать результаты запроса строка за строкой
// метод Read() возвращает значение true или false в зависимости от того, достигнут ли конец пакета строк, пришедших от сервера
// так же метод Read при каждом его вызове перемещается к следующей строке пакета, пришедшего от сервера
while (reader.Read())
{
    for (int i = 0; i < reader.FieldCount; i++)
    {
        Console.WriteLine(reader.GetName(i) + ": " + reader[i]);
    }
    Console.WriteLine(new string('_', 20));
    Console.ReadKey();
}
reader.Close();
connection.Close();
```

Результат её выполнения:

```
CustomerNo: 1
FName: Иван*+++++
LName: Круковский
MName: Петрович
Address1: Лужная 15
Address2:
City: Харьков
Phone: 3222233322
DateInSystem: 20.11.2009 0:00:00

CustomerNo: 2
FName: Виктор
LName: Дурнев
MName: Викторович
Address1: Зелинская 10
Address2:
City: Киев
Phone: <067>4242132
DateInSystem: 03.08.2010 0:00:00

CustomerNo: 3
FName: Зигмунд
LName: Чнакий
MName: федорович
Address1: Дихтяревская 5
Address2:
City: Киев
Phone: <092>7612343
DateInSystem: 17.08.2010 0:00:00
```

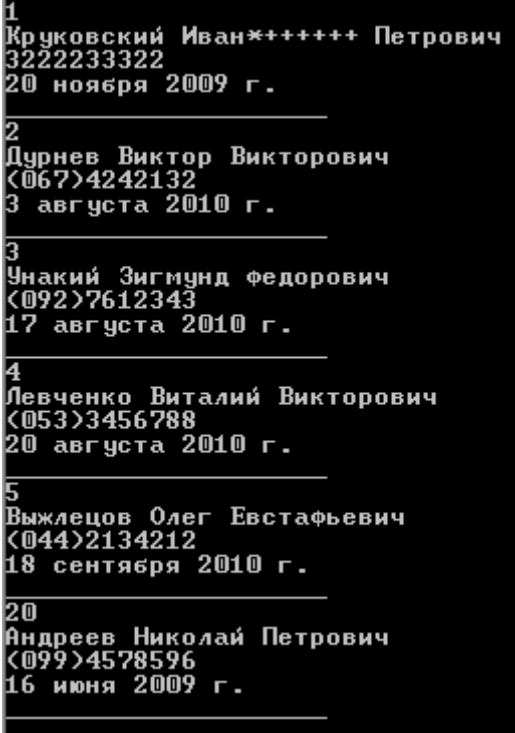
DataReader может выводить данные отдельных столбцов, по их наименованию. В примере ниже выводится информация по столбцам «LName», «FName», «MName».

```
SqlCommand cmd = new SqlCommand("SELECT * FROM Customers", connection); // построение команды, возвращающей данные в табличном представлении

// метод ExecuteReader возвращает новый объект SqlDataReader
using (SqlDataReader reader = cmd.ExecuteReader())
{
    // с помощью объекта SqlDataReader можно просматривать результаты запроса строка за строкой
    // метод Read() возвращает значение true или false в зависимости от того, имеется ли следующая строка, которую можно посмотреть
    // так же метод Read при каждом его вызове перемещается к следующей строке набора строк, пришедших от сервера
    while (reader.Read())
    {
        Console.WriteLine(reader[0]); // вывод на экран ID клиента используя перегрузку оператора с целочисленным индексом
        Console.WriteLine(           // вывод на экран ФИО клиента используя перегрузку оператора со строковым индексом
            reader["LName"] + " " +
            reader["FName"] + " " +
            reader["MName"]
        );
        Console.WriteLine(reader[7]); // вывод на экран номера телефона клиента
        Console.WriteLine("{0:D}", reader[8]); // вывод на экран поля DataInSystem клиента
        //Console.WriteLine(reader.GetFieldValue<DateTime>(8));

        Console.WriteLine(new string('_', 20));
    }
} // при выходе из блока using reader будет закрываться автоматически
connection.Close();
```

Результат выполнения:



```
1
Круковский Иван***** Петрович
3222233322
20 ноября 2009 г.

2
Дурнев Виктор Викторович
<067>4242132
3 августа 2010 г.

3
Чнакий Зигмунд Федорович
<092>7612343
17 августа 2010 г.

4
Левченко Виталий Викторович
<053>3456788
20 августа 2010 г.

5
Выжлецов Олег Евстафьевич
<044>2134212
18 сентября 2010 г.

20
Андреев Николай Петрович
<099>4578596
16 июня 2009 г.
```

Параметризированные SQL команды

Параметризированные SQL команды используются для выборки данных по какому-либо параметру.

В первой программе организуем запрос информации о клиенте по его номеру.

```
var commandStr = "SELECT * FROM Customers WHERE CustomerNo = @CustomerNo;"; // строка с запросом

Console.WriteLine("Enter customer ID");
var customerNo = Console.ReadLine(); // получение ID клиента от пользователя

SqlConnection connection = new SqlConnection(conStr); // создание подключения
SqlCommand cmd = new SqlCommand(commandStr, connection); // создание команды

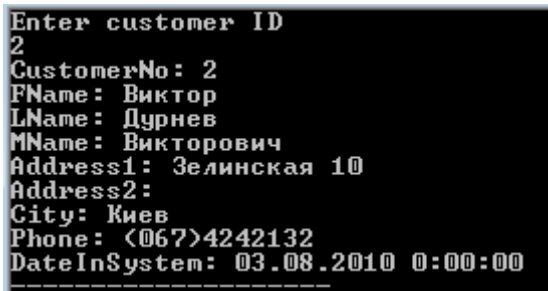
cmd.Parameters.AddWithValue("CustomerNo", customerNo); // добавление параметра в коллекцию параметров команды
connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())
{
    for (int i = 0; i < reader.FieldCount; i++)
        Console.WriteLine("{0}: {1}", reader.GetName(i), reader[i]);

    Console.WriteLine(new string('-', 20));
}
Console.ReadKey();
reader.Close();
connection.Close();
```

Результат выполнения:



```
Enter customer ID
2
CustomerNo: 2
FName: Виктор
LName: Дурнев
MName: Викторович
Address1: Зелинская 10
Address2:
City: Киев
Phone: <067>4242132
DateInSystem: 03.08.2010 0:00:00
-----
```

С помощью параметра запроса EXECUTE можно вызвать хранимую процедуру.

```
// код хранимой процедуры selectEmp: CREATE proc dbo.selectEmp
// as select * from dbo.Employees

string conStr = @"Data Source=.\SQLEXPRESS; Initial Catalog=ShopDB; Integrated Security=True;"; // создание строки подключения
SqlConnection connection = new SqlConnection(conStr);

SqlCommand cmd = new SqlCommand("EXECUTE selectEmp", connection); // создание команды, выполняющей хранимую процедуру selectEmp
connection.Open();

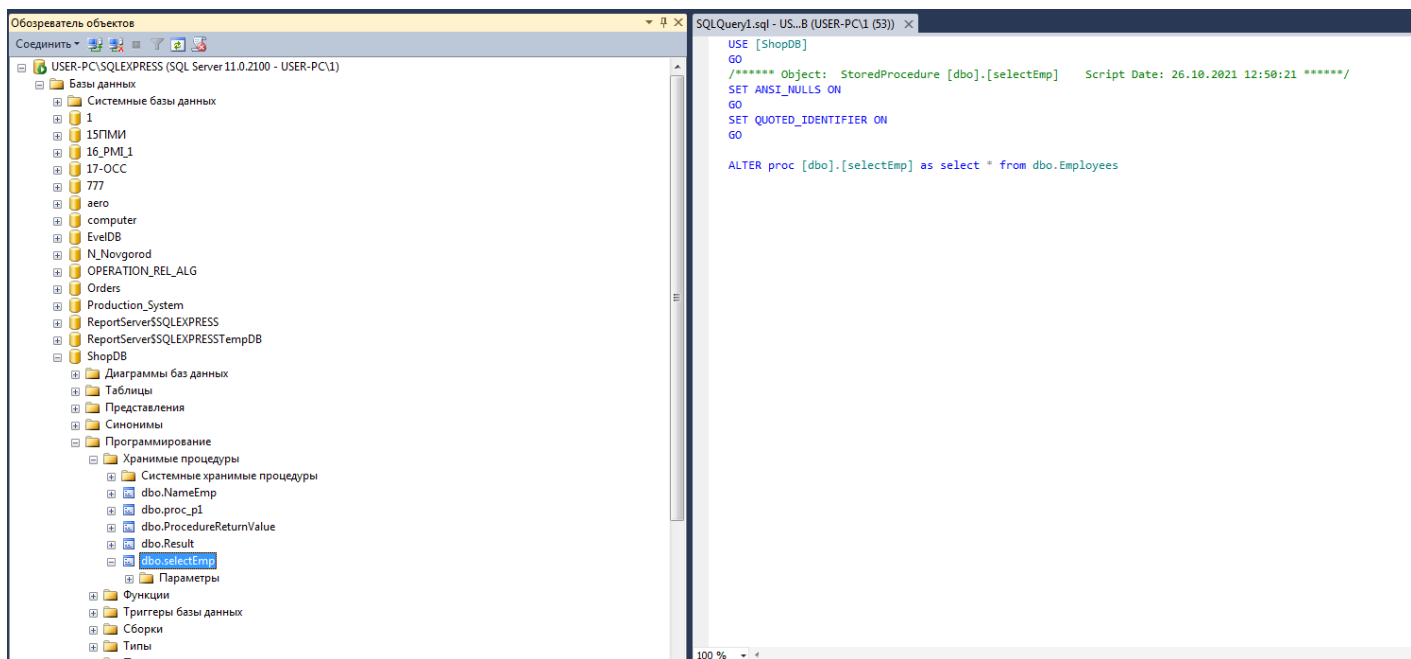
SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())
{
    for (int i = 0; i < reader.FieldCount; i++)
        Console.WriteLine("{0}: {1}", reader.GetName(i), reader[i]);

    Console.WriteLine();
    Console.ReadKey();
}

connection.Close();
```

Хранимая процедура находится в соответствующей папке базы данных ShopDB.



Результат выполнения:

```
EmployeeID: 1
FName: Василий
LName: Лященко
MName: Петрович
Salary: 5000,0000
PriorSalary: 800,0000
HireDate: 20.11.2009 0:00:00
TerminationDate:
ManagerEmpID:

EmployeeID: 2
FName: Иван
LName: Белецкий
MName: Иванович
Salary: 2000,0000
PriorSalary: 0,0000
HireDate: 20.11.2009 0:00:00
TerminationDate:
ManagerEmpID: 1

EmployeeID: 3
FName: Петр
LName: Дяченко
MName: Григорьевич
Salary: 1000,0000
PriorSalary: 0,0000
HireDate: 20.11.2009 0:00:00
TerminationDate:
ManagerEmpID: 2
```

Хранимые процедуры можно вызывать с параметрами, передаваемыми в консоль. Процедура proc_p1 выводит информацию о работнике с передаваемым номером.

Процедура proc_p1:

```
USE [ShopDB]
GO
/***** Object:  StoredProcedure [dbo].[proc_p1]    Script Date: 26.10.2021 12:55:33 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[proc_p1] @EmployeeID int as select * from dbo.Employees where EmployeeID=@EmployeeID
```

Код программы:

```
string conStr = @"Data Source=.\SQLEXPRESS; Initial Catalog=ShopDB; Integrated Security=True;"; // создание строки подключения
SqlConnection connection = new SqlConnection(conStr);

Console.WriteLine("Enter employeeID");
int employeeID = int.Parse(Console.ReadLine()); // получение данных от пользователя

SqlCommand cmd = new SqlCommand("proc_p1", connection) { CommandType = System.Data.CommandType.StoredProcedure }; // создание команды, вызывающей хранимую процедуру
cmd.Parameters.AddWithValue("@EmployeeID", employeeID); // добавление одного параметра

connection.Open();

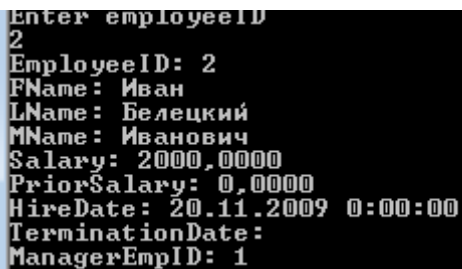
SqlDataReader reader = cmd.ExecuteReader(); // выполнение команды

while (reader.Read())
{
    for (int i = 0; i < reader.FieldCount; i++)
        Console.WriteLine("{0}: {1}", reader.GetName(i), reader[i]);

    Console.WriteLine();
    Console.ReadKey();
}

connection.Close();
```

Результат выполнения:



```
Enter employeeID
2
EmployeeID: 2
FName: Иван
LName: Белецкий
MName: Иванович
Salary: 2000,0000
PriorSalary: 0,0000
HireDate: 20.11.2009 0:00:00
TerminationDate:
ManagerEmpID: 1
```

Вывод

В ходе выполнения практической работы были получены знания по использованию объекта `DataReader`. Были освоены параметризированные команды.