

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 №4  
«Транзакции»

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 \_\_\_\_\_

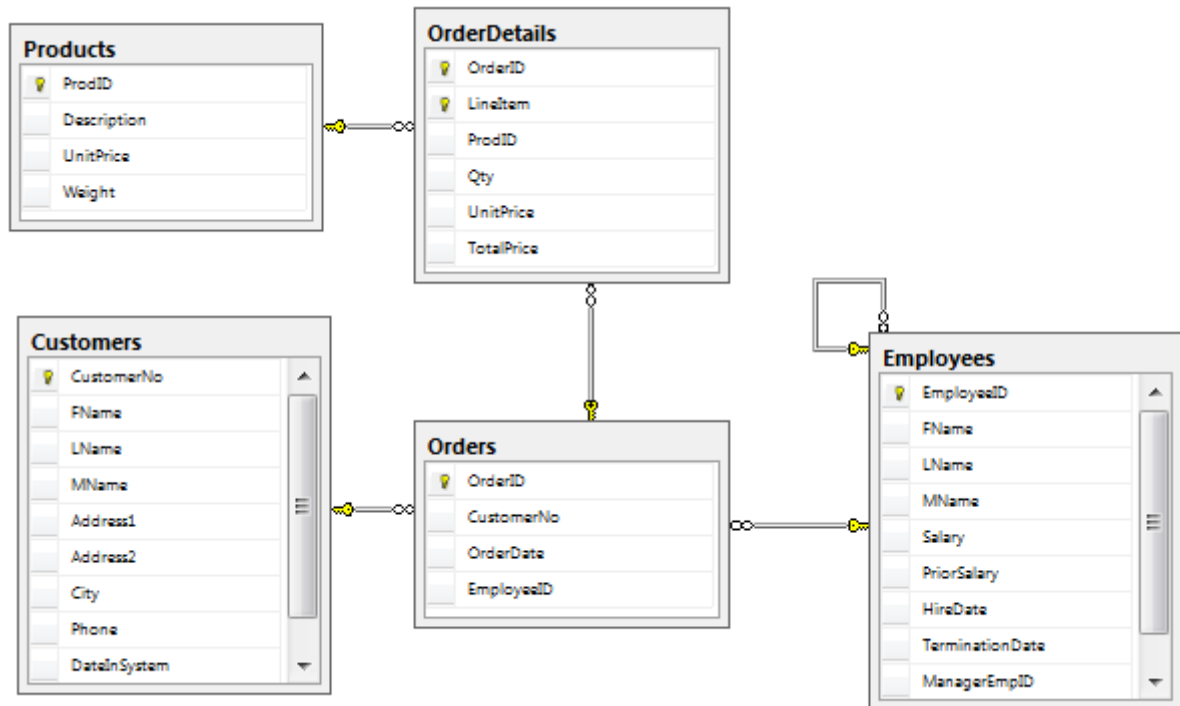
Нижний Новгород 2021

## Цель

Изучить способы работы с транзакциями.

## 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	Глуценка 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

### Создание транзакций для команд

1. BeginTransaction () – создание транзакции
2. ExecuteNonQuery () – выполнение команды
3. Commit () – применение изменений
4. Rollback () – отмена изменений

```

SqlCommand cmd = new SqlCommand("UPDATE Customers SET Phone = 'TEST' WHERE CustomerNo = 1", connection);
//cmd = new SqlCommand("UPDATE Customers SET Phone = '(052)1245789' WHERE CustomerNo = 1", connection);

try
{
    connection.Open();

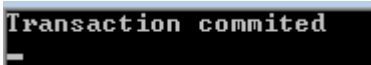
    cmd.Transaction = connection.BeginTransaction();
    cmd.ExecuteNonQuery();

    throw new Exception();

    cmd.Transaction.Commit();

    Console.WriteLine("Transaction committed");
}
catch (Exception)
{
    cmd.Transaction.Rollback();
    Console.WriteLine("Transaction rollback");
}

```



CustomerNo	FName	LName	MName	Address1	Address2	City	Phone	DateInSystem
1	Иван*+++++	Круковский	Петрович	Лужная 15	NULL	Харьков	TEST	2009-11-20

## Создание транзакций для команд

### 1 транзакция

```

SqlCommand cmd = new SqlCommand("UPDATE Customers SET Phone = 'TEST' WHERE CustomerNo = 1", connection);
cmd = new SqlCommand("UPDATE Customers SET Phone = '(052)1245789' WHERE CustomerNo = 1", connection);

Console.WriteLine("Step 1. Press any key to execute command...");
Console.ReadKey();

connection.Open();

cmd.Transaction = connection.BeginTransaction();
cmd.ExecuteNonQuery();

Console.WriteLine("Step 3. Press any key to rollback transaction...");
Console.ReadKey();

cmd.Transaction.Rollback();
connection.Close();

Console.WriteLine("Transaction rollback");

```

### 2 транзакция

```

SqlCommand cmd = new SqlCommand("SELECT LName, FName, Phone FROM Customers", connection);

Console.WriteLine("Step 2. Press any key to read Customers");
Console.ReadKey();

connection.Open();
cmd.Transaction = connection.BeginTransaction(IsolationLevel.ReadUncommitted);
SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())
{
    Console.WriteLine("{0} {1}: {2}", reader[0], reader[1], reader[2]);
}

connection.Close();

```

Данные в таблице Customers до выполнения программы:

CustomerNo	FName	LName	MName	Address1	Address2	City	Phone	DateInSystem
1	Иван*+++++	Круковский	Петрович	Лужная 15	NULL	Харьков	(052)1245789	2009-11-20

```
Step 1. Press any key to execute command...
_
```

```
Step 2. Press any key to read Customers
Круковский Иван*+++++: TEST
Дурнев Виктор: <067>4242132
Унакий Зигмунд: <092>7612343
Левченко Виталий: <053>3456788
Выжлецов Олег: <044>2134212
Андреев Николай: <099>4578596
```

```
Step 1. Press any key to execute command...
Step 3. Press any key to rollback transaction...
Transaction rollback
```

Данные в таблице Customers после выполнения программы:

CustomerNo	FName	LName	MName	Address1	Address2	City	Phone	DateInSystem
1	Иван*+++++	Круковский	Петрович	Лужная 15	NULL	Харьков	(052)1245789	2009-11-20

## Создание параметризованных запросов с помощью конкатенации строк

```
Console.WriteLine("Введите ID клиента");

var customerNo = Console.ReadLine();

// не используйте конкатенацию строк для запросов во избежание изменения структуры запроса пользователем
// для создания параметризованного запроса используется метод string.Format
string commandStr = string.Format("SELECT * FROM Customers WHERE CustomerNo = {0};", customerNo);

using (SqlConnection connection = new SqlConnection(conStr)) // создание подключения
{
    connection.Open();

    SqlCommand cmd = new SqlCommand(commandStr, connection);

    using (SqlDataReader reader = cmd.ExecuteReader()) // выполнение запроса и чтение результатов
    {
        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));
            }
        }
    }
}
```

```
Введите ID клиента
4
CustomerNo: 4
FName: Виталий
LName: Левченко
MName: Викторович
Address1: Глушенка 5
Address2: Драйзера 12
City: Киев
Phone: <053>3456788
DateInSystem: 20.08.2010 0:00:00
-----
```

Ввод недопустимых данных:

```
Введите ID клиента
1 OR CustomerNo = 2
CustomerNo: 1
FName: Иван*+++++
LName: Круковский
MName: Петрович
Address1: Лужная 15
Address2:
City: Харьков
Phone: <052>1245789
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
-----
```

## Создание и выполнение параметризованных запросов с использованием коллекции Parameters объекта SqlCommand

```
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

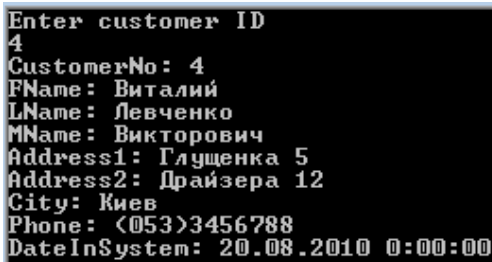
cmd.Parameters.AddWithValue("CustomerNo", customerNo); // добавление параметра в коллекцию параметров команды
connection.Open();

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));
}

reader.Close();
connection.Close();
```



```
Enter customer ID
4
CustomerNo: 4
FName: Виталий
LName: Левченко
MName: Викторович
Address1: Глушенка 5
Address2: Драйзера 12
City: Киев
Phone: <053>3456788
DateInSystem: 20.08.2010 0:00:00
-----
```

## Использование входных параметров в параметризованных запросах

```
SqlCommand cmd = new SqlCommand("SET @Parameter = 2;", connection);

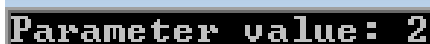
SqlParameter parameter = cmd.Parameters.Add(new SqlParameter("Parameter", System.Data.SqlDbType.Int));
parameter.Direction = System.Data.ParameterDirection.Output; // указание направления параметра

connection.Open();

cmd.ExecuteNonQuery();

Console.WriteLine("Parameter value: " + parameter.Value); // вывод на экран значения параметра после выполнения запроса

connection.Close();
```



```
Parameter value: 2
```

# Вызов хранимой процедуры с использованием команды EXECUTE T-SQL

```
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();
}

connection.Close();
```

## Процедура selectEmp

```
USE [ShopDB]
GO
-- Object: [dbo].[selectEmp]    Script Date: 03.11.2021 19:22:31 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER proc [dbo].[selectEmp] as select * from dbo.Employees
```

```
HireDate: 20.11.2009 0:00:00
TerminationDate:
ManagerEmpID: 2

EmployeeID: 4
FName: Светлана
LName: Ляleckина
MName: Олеговна
Salary: 800,0000
PriorSalary: 0,0000
HireDate: 20.11.2009 0:00:00
TerminationDate:
ManagerEmpID: 2

EmployeeID: 5
FName: Александр
LName: Жванецкий
MName: Иванович
Salary: 1500,0000
PriorSalary: 0,0000
HireDate: 20.11.2009 0:00:00
TerminationDate:
ManagerEmpID: 4
```

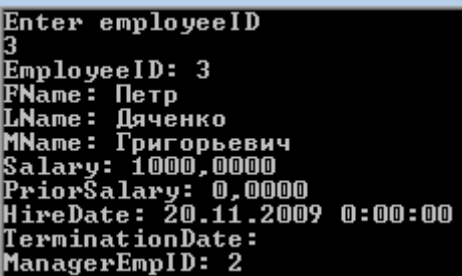
## Выполнение хранимой процедуры, принимающей параметры

```
SqlCommand cmd = new SqlCommand("proc_p1", connection) { CommandType = System.Data.CommandType.StoredProcedure };  
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();  
}  
connection.Close();
```

### Процедура proc\_p1

---

```
USE [ShopDB]  
GO  
/***** Object: StoredProcedure [dbo].[proc_p1]    Script Date: 03.11.2021 19:33:03 *****/  
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
```



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



## Выполнение процедуры, возвращающей значение

```
SqlCommand cmd = new SqlCommand("ProcedureReturnValue", connection) { CommandType = System.Data.CommandType.StoredProcedure };
SqlParameter parameter = cmd.Parameters.Add(new SqlParameter());
parameter.Direction = System.Data.ParameterDirection.ReturnValue; // после выполнения команды parameter будет содержать возвр

connection.Open();

cmd.ExecuteNonQuery();

Console.WriteLine(parameter.Value);
```

### Процедура ProcedureReturnValue

```
USE [ShopDB]
GO
/***** Object: StoredProcedure [dbo].[ProcedureReturnValue]    Script Date: 03.11.2021 19:40:24 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[ProcedureReturnValue]
AS
BEGIN
return 1;
END
```

1

## Вывод

В результате данной практической работы были изучены способы работы с транзакциями и параметризованными запросами. Были созданы транзакции для команд и выполнены различные параметризованные запросы.