



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ПОСТРЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

Отчёт по лабораторной работе № 4

«Построение объектных запросов на примере MS SQL Server»

Выполнил:
студент группы ИУ5 – 23М

Крутов Т.Ю.

Преподаватель:

Виноградова М. В.

2020г.

1. Работа с коллекциями – LINQ to object

Создадим консольное приложение в среде MS Visual Studio на языке C#.

```
using System;
using System.Linq;
using System.Collections.Generic;
namespace lab4
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] authors = {"Зигмунд Фрейд", "Эрнест Хемингуэй", "Олдос Хаксли",
                                "Иммануил Кант", "Алексей Толстой", "Лев Толстой", "Дмитрий Семёнов",
                                "Вильям Шекспир", "Владимир Маяковский", "Джозеф Раттц", "Виктор Крам"};

            one();
            two();
            tree();
            four();

            unique();
            five();

            void one()
            {
                IEnumerable<string> items = authors.Where(p => p.StartsWith("B"));
                foreach (string item in items)
                    Console.WriteLine(item + '\n');
                Console.WriteLine('\n');
            }
            void two()
            {
                var selectedAuthors = from a in authors
                                      where a.ToUpper().StartsWith("B")
                                      orderby a
                                      select a;
                foreach (string author in selectedAuthors)
                    Console.WriteLine(author);
            }
            void tree()
            {
                var uniqueAuthors = from a in authors
                                    select authors.Distinct().Count();

                foreach (int a in uniqueAuthors)
                {
                    Console.WriteLine(a);
                    break;
                }

                Console.WriteLine(uniqueAuthors.First());
            }
            // Группировка
            void four()
            {
                var groupAuthors = from a in authors
```

```

        group a by a[0];

foreach (var grA in groupAuthors)
{
    Console.WriteLine(grA.Key + " " + grA.Count());

    foreach (var g in grA)
        Console.WriteLine($"{t}{g}");
}

// Группировка на лету

void five()
{
    var group = authors.GroupBy(x => x[0]).Select(g => new
    {
        name = g.Key,
        count = g.Count()
    });
    foreach (var gr in group)
    {
        Console.WriteLine(gr);
    }
    Console.WriteLine(group);
}

// Уникальные строки лямда-выражение.
void unique()
{
    var unique = authors.Select(x => x.Distinct()).Count();
    Console.WriteLine(unique);
}

}

}

```

Вывод программы после запуска:

```
Консоль отладки Microsoft Visual Studio

Виктор Крам
Вильям Шекспир
Владимир Маяковский
11
11
1 1
1 1 Зигмунд Фрейд
1 1 Эрнест Хемингуэй
1 1 Олдос Хаксли
1 1 Иммануил Кант
1 1 Алексей Толстой
1 1 Лев Толстой
1 2
1 2 Дмитрий Семёнов
1 2 Джозеф Раттц
1 3
1 3 Вильям Шекспир
1 3 Владимир Маяковский
1 3 Виктор Крам
11
{ name = 3, count = 1 }
{ name = 3, count = 1 }
{ name = 0, count = 1 }
{ name = И, count = 1 }
{ name = А, count = 1 }
{ name = Л, count = 1 }
{ name = Д, count = 2 }
{ name = Б, count = 3 }
System.Linq.Enumerable+SelectEnumerableIterator`2[System.Linq.IGrouping`2[System.Char,System.String],<>f__AnonymousType0`2[System.Char,System.In
t32]]
```

2. Работа с базой данных - LINQ TO SQL

2.1. Создание БД

В среде MS SQL Server была разработана схема базы данных, а таблицы заполнены данными.

accou...	userID	login_	Pass
1	1	hsdhgfh	456
2	4	y54wt	sfgh
3	10	fgdfjgdfhello	gffxshn...
6	5	acclogin	passpass
9	3	qwerty12	kd;da
10	3	daaaD	13213
11	6	LOGINIOGIN	b@931...
13	7	rfgeg	ergwerge
14	8	dafadsg	sadgfsag
* NULL	NULL	NULL	NULL

2.2. Создание windows приложения на языке C#. Настройка соединения с БД. Создание windows форм.

Создадим приложение вида Windows Application Forms. Для настройки соединения с БД выберем сервер базы данных через обозреватель серверов и добавим необходимые таблицы. Используя класс DataContext, получим возможность обращаться к таблицам базы данных, как к объектам класса.

Результаты выполнения запросов будем выводить в поле типа DataGridView.

Добавление в БД через запрос Linq to sql:

```
ACCOUNT acc;
acc = new ACCOUNT
{
    userID = Convert.ToInt32(textBox1.Text),
    login_ = textBox2.Text,
    Pass = textBox3.Text
};
db.ACCOUNT.InsertOnSubmit(acc);
db.SubmitChanges();

textBox1.Clear(); textBox2.Clear(); textBox3.Clear();

MessageBox.Show("Добавлено!");

dataGridView1.DataSource = db.ACCOUNT;
```

Обновление:

```
private void button3_Click(object sender, EventArgs e)
{
    ACCOUNT acc = db.ACCOUNT.Where(x => x.accountID == accID).Single();
    MessageBox.Show(acc.login_ + "-" + acc.Pass + "\n" + "UPDATED");
    acc.userID = Convert.ToInt32(textBox1.Text);
    acc.login_ = textBox2.Text;
    acc.Pass = textBox3.Text;
    db.SubmitChanges();
}
```

Удаление:

```
private void button2_Click(object sender, EventArgs e)
{
    ACCOUNT acc = new ACCOUNT();
    acc = db.ACCOUNT.Where(x => x.accountID == accID).Single();
    MessageBox.Show(acc.login_ + "-" + acc.Pass);
    db.ACCOUNT.DeleteOnSubmit(acc);
    db.SubmitChanges();
    Del();

    MessageBox.Show("Удалено!");
    dataGridView1.DataSource = db.ACCOUNT;
}
```

Фильтрация, группировка, сортировка:

```
dataGridView1.DataSource = dbF.ACCOUNT.Where(x => x.login_.Length < 7).Select(y => new {
    accountID = y.accountID,
    userID = y.userID,
    login_ = y.login_,
    Pass = y.Pass
}).OrderBy(x=> x.login_);
```

```
dataGridView1.DataSource = db.ACCOUNT.GroupBy(x => x.userID).Select(y => new {userID = y.Key , Amount = y.Count() }).OrderBy(y=>y.Amount);
```

Переход по связи 1-М:

```
dataGridView1.DataSource = from a in db.ACCOUNT
    where a.ORDERS.Any(x => x.order_date >= Convert.ToDateTime("12.03.2020"))
select new { a.userID, a.login_, OrderDate = a.ORDERS.First(x => x.order_date >= Convert.ToDateTime("12.03.2020")).order_date };
```

Переход по связи М-1:

```
dataGridView1.DataSource = from b in db.ORDERS
    where b.ACCOUNT.accountID == 1
select new {
    b.accountID,
    b.ammount_items,
    b.orderID,
    b.order_date,
    b.ACCOUNT.login_,
    b.ACCOUNT.Pass
};
```

Any , all:

```
dataGridView1.DataSource = from a in db.ACCOUNT
    where a.ORDERS.All(x => x.accountID > 2)
select new { a.userID, a.login_,};
```

Загрузка данных в DataGridView выполняется с помощью создания экземпляра DataContext и вызова в нем необходимой таблицы:

```
private void button4_Click(object sender, EventArgs e)
{
    DataClasses1DataContext dbUPD = new DataClasses1DataContext();
    dataGridView1.DataSource = dbUPD.ACCOUNT;
}
```

Общий интерфейс программы представлен следующими рисунками:

Form1

Аккаунты Заказы Тексты

accounts

	accountID	userID	login_	Pass
▶	1	1	hsdhgfh	456
	2	4	y54wt	sfgh
	3	10	fgdfjgdfhello	gffxshndfg
	6	5	acclogin	passpass
	9	3	qwerty12	kd;da
	10	3	daaaD	13213
	11	6	LOGINIOGIN	b@931@#(!

ID пользователя

[Filter](#)

Логин

Пароль

Order

	orderID	accountID	ammount_items	order_date
▶	1	3	1	12.03.2020
	2	1	1	31.03.2020
	3	1	1	17.02.2020
	4	1	2	01.01.2020
*				

ID аккаунта

Количество

Дата

Тип
☐ Первые

Список используемой литературы:

1. Методические указания по выполнению лабораторной работы.
2. Джозеф Раттц-мл. LINQ: язык интегрированных запросов в С# для профессионалов. И.Д. Вильямс, 2008. – 560с.