

# 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.

### 1. Работа с коллекциями – LINQ to object

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

```
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", "Джосзеф 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 \Rightarrow x[0]).Select(g \Rightarrow 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);
        }
    }
}
}
```

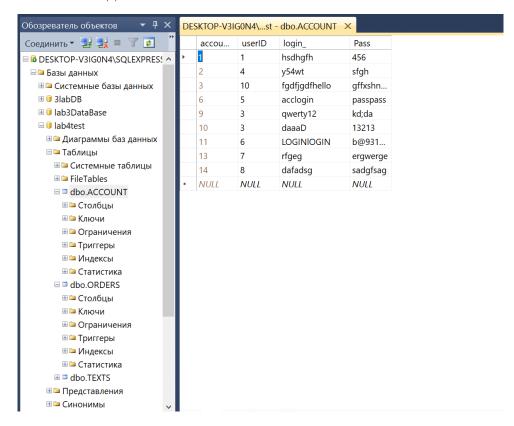
### Вывод программы после запуска:

```
| Saktop Kpam |
```

### 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.



### 2.2. Создание windows приложения на языке С#. 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 запросов будем выполнения выводить В поле типа DataGridView.

Добавление в БД через запрос Ling 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:

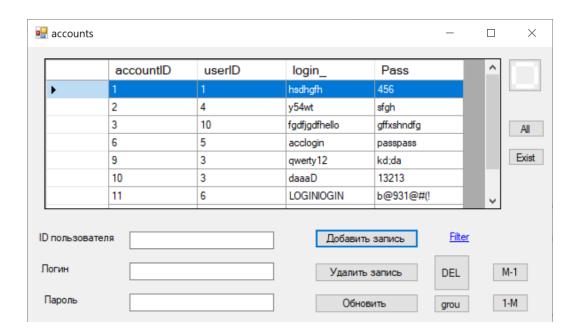
```
dataGridView1.DataSource = dbF.ACCOUNT.Where(x => x.login .Length <</pre>
              7).Select(y => new
                 accountID = y.accountID,
                 userID = y.userID,
                 login_ = y.login_,
                 Pass = y.Pass
                 }).OrderBy(x=> x.login_);
    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 \Rightarrow x.accountID > 2)
                            select new { a.userID, a.login_,};
```

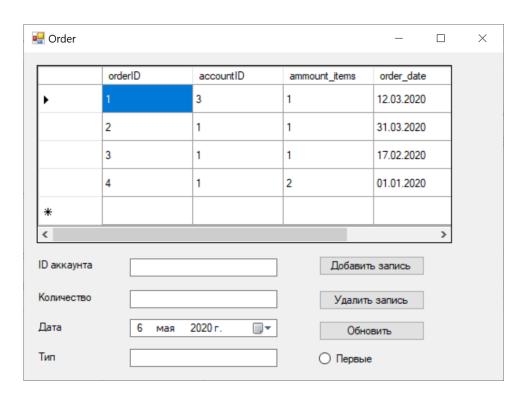
Загрузка данных в DataGridView выполняется с помощью создания экземпляра DataContext и вызова в нем необходимой таблицы:

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

Общий интерфейс программы представлен следующими рисунками:







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.