

Кафедра «Систем обработки информации и управления»

**Лабораторная работа №4**  
по курсу Постреляционные базы данных

Тема: «Построение объектных запросов на примере MS SQL Server»

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-22М

Сметанкин К.И.

\_\_\_\_\_

" " \_\_\_\_\_ 2020 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В.

к.т.н., доцент

\_\_\_\_\_

" " \_\_\_\_\_ 2020 г.

## Пункты задания для выполнения:

### 1. Работа с коллекциями - LINQ to object (базовая)

В среде MS Visual Studio создать консольное приложение на языке C#. Определить и инициализировать коллекцию (список строк). Используя технологию linq to objects выполнить набор запросов (через точечный синтаксис и запросом), например:

- строки, содержащие подстроку,
  - отсортированные строки,
  - (точечный синтаксис) количество строк, являющихся уникальными,
  - группировка строк по первой букве. В результат — буква, количество строк (генерация объектов «на лету»).
- Результат отобразить на экране.

### 2. Работа с базой данных - linq to sql (базовая)

#### 2.1. Создать БД

В среде MS SQL Server Management Studio создать БД и добавить в нее таблицы по теме своего ДЗ, например

**"Hotel"(отель)**, содержит свойства:

- **hname** - **название** – строковое varchar(10), (КЛЮЧ),
- **hcountry** – **страна** – строковое varchar(15),
- **hcity** – **город** – строковое varchar(15),
- **stars** – **количество звезд** – целое.

**"Client" (клиент)**, содержит свойства:

- **fio** - ФИО – строковое, varchar(50),
- **num** - номер – целое,
- **age** - возраст – целое,
- **country** – гражданство – varchar(15),
- **ofhotel** — **отель** — varchar(10), (ВНЕШНИЙ КЛЮЧ)
- **id** - идентификатор (КЛЮЧ - РК) – целое, автоинкремент.

Открыть таблицы на редактирование и заполнить тестовыми данными.

2.2. Создать windows приложение на языке C#. Настроить соединение с созданной ранее (в п.1) БД. Подключить все таблицы. Создать форму, содержащую таблицы, набор полей и кнопок. Или создать консольное приложение.

Определить элементы и действия, например :

- Отобразить клиентов (таблицей).
- Продемонстрировать добавление, удаление и редактирование клиента.

Продемонстрировать запросы с фильтром (вывод в таблицы через дополнительные кнопки):

- Испанские отели 4 или 5 звезд,
  - Клиенты греческих отелей (полные сведения о клиентах и их отелях),
- Использовать точечный синтаксис и методы коллекций.

# Реализация

## Часть 1 (LINQ to object)

```
string[] actors = { "Операция Пластилин", "Noize NC", "Пицца", "Hospital", "Суп Харчо", "4 позиции Бруно", "П", "П"};
```

```
    // строки, содержащие подстроку
    var request1 = from a in actors // определяем каждый объект из
actors как a
                    where a.Contains("ун") //фильтрация по критерию
                    orderby a // упорядочиваем по возрастанию
                    select a; // выбираем объект
    print(request1.ToArray());

    // отсортированные строки
    var request2 = from a in actors // определяем каждый объект из
actors как a
                    orderby a // упорядочиваем по возрастанию
                    select a; // выбираем объект
    print(request2.ToArray());

    // (точечный синтаксис) количество строк, являющихся уникальными
    var stringCounter = actors.Distinct().OrderBy(t => t).Count();
    print(new string[] { stringCounter.ToString() });

    // группировка строк по первой букве. В результат – буква,
количество строк (генерация объектов «на лету»).
    var request4 = from actor in actors
                    group actor by actor[0] into g
                    select new { Name = g.Key, Count = g.Count() };
    foreach (var customPair in request4)
        Console.WriteLine($"{customPair.Name} : {customPair.Count}");

    Console.WriteLine();

    var request4_1 = actors.GroupBy(p => p[0]).Select(g => new { Name =
g.Key, Count = g.Count() });
    foreach (var customPair in request4_1)
        Console.WriteLine($"{customPair.Name} : {customPair.Count}");

    Console.ReadKey();
```

## Результат

```
4 позиции Бруно  
Hospital  
Noize NC  
Операция Пластин  
П  
П  
Пицца  
Суп Харчо
```

7

```
О : 1  
N : 1  
П : 3  
H : 1  
C : 1  
4 : 1
```

```
О : 1  
N : 1  
П : 3  
H : 1  
C : 1  
4 : 1
```

## Часть 2 (LINQ to SQL)

Созданы и заполнены таблицы в БД

```

CREATE TABLE users
(
    id            integer PRIMARY KEY,
    login         varchar(100) NOT NULL UNIQUE,
    password      TEXT,
    avatar        varchar(100) NOT NULL UNIQUE,
    karma         INT          DEFAULT 0
);

CREATE TABLE posts
(
    id            integer PRIMARY KEY,
    header        varchar(100) NOT NULL UNIQUE,
    short_topic   varchar(255) NOT NULL UNIQUE,
    main_topic    text NOT NULL,
    user_id       integer      NOT NULL REFERENCES users (id)
);

CREATE TABLE comments
(
    id            integer PRIMARY KEY,
    parent_id     integer REFERENCES comments (id),
    user_id       integer      NOT NULL REFERENCES users (id),
    post_id       integer      NOT NULL REFERENCES posts (id),
    payload       text NOT NULL
);

```

## Результат

	id	login	password	avatar	karma
1	1	user_1	password_1	/static/default.jpg	0
2	2	user_2	password_2	/static/my.jpg	0

  

	id	header	short_topic	main_topic	user_id
1	3	header 3	short topic 1	main topic 1	1
2	4	header 2	short topic 2	main topic 2	1

  

	id	parent_id	user_id	post_id	payload
1	1	NULL	1	3	отличная статья, больше не пиши
2	2	1	1	3	спасибо
3	3	1	2	3	великолепно
4	4	NULL	1	4	ну почти

Были созданы отображения таблиц с помощью linq и реализована форма win Form

```

public partial class Form1 : Form
{
    public DataClasses2DataContext db;

    public Form1()
    {
        InitializeComponent();
        db = new DataClasses2DataContext();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        dataGridView1.DataSource = db.posts;
        dataGridView2.DataSource = db.comments;
        dataGridView3.DataSource = db.users;

        toolStripAddItemButton.Enabled = true;
        toolStripDeleteItemButton.Enabled = true;
        saveToolStripButton.Enabled = true;
    }

    private void saveToolStripButton_Click_1(object sender, EventArgs e)
    {
        string idText = postId.Text;
        int id = Int32.Parse(idText);

        int userId = 0;
        string userIdText = postUserID.Text;
        if (userIdText.Length > 0)
        {
            userId = Int32.Parse(userIdText);
        }

        posts postForUpdate = db.posts.Where(it => it.id == id).Single();

        if (postHeader.Text.Length > 0)
        {
            postForUpdate.header = postHeader.Text;
        }
        if (postShortTopic.Text.Length > 0)
        {
            postForUpdate.short_topic = postShortTopic.Text;
        }
        if (postMainTopic.Text.Length > 0)
        {
            postForUpdate.main_topic = postMainTopic.Text;
        }
        if (postForUpdate.user_id != userId && userId > 0)
        {
            postForUpdate.user_id = userId;
        }

        db.SubmitChanges();

        updatePostTableView();
    }

    private void clearInput()
    {
        postId.Text = "";
    }
}

```

```

        postShortTopic.Text = "";
        postHeader.Text = "";
        postMainTopic.Text = "";
        postUserID.Text = "";
    }

private void toolStripDeleteItemButton_Click(object sender, EventArgs e)
{
    string idText = deleteIdTextBox.Text;
    int id = Int32.Parse(idText);

    posts postForDelete = db.posts.Where(it => it.id == id).Single();
    db.posts.DeleteOnSubmit(postForDelete);
    db.SubmitChanges();

    updatePostTableView();

    deleteIdTextBox.Text = "";
}

private void toolStripAddItemButton_Click(object sender, EventArgs e)
{
    string idText = postId.Text;
    int id = Int32.Parse(idText);

    string userIDT = postUserID.Text;
    int userID = Int32.Parse(userIDT);

    posts newPost = new posts();
    newPost.id = id;
    newPost.header = postHeader.Text;
    newPost.short_topic = postShortTopic.Text;
    newPost.main_topic = postMainTopic.Text;
    newPost.user_id = userID;

    db.posts.InsertOnSubmit(newPost);
    db.SubmitChanges();

    updatePostTableView();

    clearInput();
}

private void updatePostTableView()
{
    db = new DataClasses2DataContext();
    dataGridView1.DataSource = db.posts;

    dataGridView1.Refresh();
}

private void updateCommentsTableView()
{
    db = new DataClasses2DataContext();
    dataGridView2.DataSource = db.comments;

    dataGridView2.Refresh();
}

private void filterByAge_Click(object sender, EventArgs e)

```

```

    {
        dataGridView1.DataSource = db.posts.OrderBy(it =>
it.header).Select(it => it);
    }

    private void button1_Click(object sender, EventArgs e)
    {
        updatePostTableView();
    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        dataGridView2.DataSource = from o in db.comments
                                   where o.parent_id.Equals(null)
                                   select o;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        updateCommentsTableView();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        dataGridView3.DataSource = from u in db.users
                                   where u.comments.Count() > 0 &&
u.comments.Any(it => it.parent_id.Equals(null))
                                   orderby u.comments.Count, u.login
                                   select new
                                   {
                                       u.id,
                                       u.login,
                                       u.password,
                                       u.avatar,
                                       u.karma,
                                       u.comments,
                                   };
    }

    private void button4_Click(object sender, EventArgs e)
    {
        dataGridView3.DataSource = from u in db.posts.Where(it =>
it.comments.Count > 0).Select(it => it.users)
                                   group u by u.login into u
                                   select new
                                   {
                                       login = u.Key,
                                       count = u.Count(),
                                   };
    }

    private void button5_Click(object sender, EventArgs e)
    {
        db = new DataClasses2DataContext();
        dataGridView3.DataSource = db.users;
        dataGridView3.Refresh();
    }

    private void button6_Click(object sender, EventArgs e)

```



```

        {
            dataGridView3.DataSource = from u in db.users.Where(it =>
it.posts.Count == 0)

select new
{
    u.id,
    u.login,
    u.password,
    u.avatar,
    u.karma,
    u.comments,
};
        }

private void button7_Click(object sender, EventArgs e)
{
    dataGridView3.DataSource = from u in db.posts.Where(it =>
it.comments.Count > 0).Select(it => it.users).Distinct()
select new
{
    u.id,
    u.login,
    u.password,
    u.avatar,
    u.karma,
    u.comments,
};
}
}
}

```

## Результат

Lab4

Статьи

id	header	short_topic	main_topic	user_id	users
3	header 3	short topic 1	main topic 1	1	Lab4_p2.users
4	header 2	short topic 2	main topic 2	1	Lab4_p2.users
*					

ID для удаления

Показать все

Отфильтровать по заголовку

Данные для добавления/изменения

Комментарии

id	parent_id	user_id	post_id	payload	comments1
1		1	3	отличная статья...	La
2	1	1	3	спасибо	Lab4_p2.comme...
3	1	2	3	великолепно	Lab4_p2.comme...
4		1	4	ну почти	La
*					

Показать все

Корневые комментарии

Пользователи

id	login	password	avatar	karma
1	user_1	password_1	/static/default.jpg	0
2	user_2	password_2	/static/my.jpg	0
*				

Показать все

Все пользователи с корневыми комментариями

Пользователи со статьями, имеющими комментарии

Пользователи без статей

Пользователи со статьями с комментариями

## **Литература:**

1. Джозеф с. Раттц - мл. LINQ. Язык интегрированных запросов в С# 2008 для профессионалов. - Издательский дом Вильямс. - 2008 г.

## **Литература дополнительная**

2. Голощапов А. Microsoft Visua Studio 2010. - Спб.: БХВ-Петербург. - 2011 г.
3. Основы программирования на С#. - Электронный ресурс [<http://www.intuit.ru>]