

Петров Иван      ИУ5-35Б      РК2

Имитация связи много-ко-многим в LINQ производится с помощью таблицы, выполняющей роль связи между двумя другими.

Пусть у нас есть класс Data (класс данных) и класс DataLink (связь между списками):

```
public class Data
{
    /// <summary>
    /// Ключ
    /// </summary>
    public int id;

    /// <summary>
    /// Для группировки
    /// </summary>
    public string grp;

    /// <summary>
    /// Значение
    /// </summary>
    public string value;

    /// <summary>
    /// Конструктор
    /// </summary>
    public Data(int i, string g, string v)
    {
        this.id = i;
        this.grp = g;
        this.value = v;
    }

    /// <summary>
    /// Приведение к строке
    /// </summary>
    public override string ToString()
    {
        return "(id=" + this.id.ToString() + "; grp=" + this.grp + "; value=" + this.value + ")";
    }
}

public class DataLink
{

```

```

    public int d1;
    public int d2;

    public DataLink(int i1, int i2)
    {
        this.d1 = i1;
        this.d2 = i2;
    }
}

```

Создадим объекты данных классов:

```

static List<Data> d1 = new List<Data>()
{
    new Data(1, "group1", "11"),
    new Data(2, "group1", "12"),
    new Data(3, "group2", "13"),
    new Data(5, "group2", "15")
};

static List<Data> d2 = new List<Data>()
{
    new Data(1, "group2", "21"),
    new Data(2, "group3", "221"),
    new Data(2, "group3", "222"),
    new Data(4, "group3", "24")
};

static List<DataLink> lnk = new List<DataLink>()
{
    new DataLink(1,1),
    new DataLink(1,2),
    new DataLink(1,4),
    new DataLink(2,1),
    new DataLink(2,2),
    new DataLink(2,4),
    new DataLink(5,1),
    new DataLink(5,2)
};

```

С помощью let мы можем сформировать временную таблицу (временный запрос) temp1. Мы берем все данные l из таблицы связей lnk (с помощью from), где первичный ключ таблицы связей равняется ключу первой таблицы.

Перебирая все данные t1 из временного запроса temp1, а также все данные у из таблицы d2, мы ищем все записи из второй таблицы, ключи которых соответствуют вторичному ключу таблицы-связи:

```
Console.WriteLine("Имитация связи много-ко-многим, использование let, проверка условия");
var lnk3 = from x in d1

    let temp1 = from l in lnk where l.d1 == x.id select l

    from t1 in temp1

    let temp2 = from y in d2 where y.id == t1.d2 && y.value == "24"
                select y
    where temp2.Count() > 0

    select x;

foreach (var x in lnk3) Console.WriteLine(x);
```

Таким образом мы можем получить все записи из первой таблицы, которые соответствуют какому-либо значению записи из второй таблицы (в данном примере это значение равняется 24).

Причем вместо строк

```
let temp2 = from y in d2 where y.id == t1.d2 && y.value == "24"
            select y
            where temp2.Count() > 0
```

**можно использовать следующие:**

```
let temp2 = from y in d2 where y.id == t1.d2
            select y
            where temp2.Any(t=>t.value == "24")
```