



Операции с множествами: объединение, пересечение, разность

Последнее обновление: 31.10.2015



Ряд методов Linq позволяют работать с результатами выборки как со множествами, производя операции на объединение, пересечение, разность двух выборок.

Но перед использованием данных методов надо учитывать, что они проводятся над однородными выборками с одинаковым определением строк, то есть которые совпадают по составу столбцов.

Объединение

Для объединения двух выборок используется метод **Union()**:

```
using(PhoneContext db = new PhoneContext())
{
    var phones = db.Phones.Where(p => p.Price < 25000)
        .Union(db.Phones.Where(p=>p.Name.Contains("Samsung")));
    foreach (var item in phones)
        Console.WriteLine(item.Name);
}
```

Метод Union в качестве параметра принимает результаты второй выборки и объединяет ее с исходной.

При этом мы не можем объединить две разнородные выборки, например, таблицу, моделей телефонов и таблицу производителей телефонов. Однако уместна следующая запись:

```
var result = db.Phones.Select(p => new { Name = p.Name })
    .Union(db.Companies.Select(c => new { Name = c.Name }));
```

Первая выборка после метода Select будет формировать набор элементов с одним столбцом Name. Вторая выборка из таблицы компаний после метода Select также будет формировать набор элементов с одним столбцом Name. Поэтому строки в обоих выборках будут однородны, и мы их сможем объединять.

Пересечение

Чтобы найти пересечение выборок, то есть те элементы, которые присутствуют сразу в двух выборках, используется метод **Intersect()**:

```
using(PhoneContext db = new PhoneContext())
{
    var phones = db.Phones.Where(p => p.Price < 25000)
        .Intersect(db.Phones.Where(p=>p.Name.Contains("Samsung")));
    foreach (var item in phones)
        Console.WriteLine(item.Name);
}
```

Разность

Если нам надо найти элементы первой выборки, которые отсутствуют во второй выборке, то мы можем использовать метод **Except()**:

```
using(PhoneContext db = new PhoneContext())
{
    var selector1 = db.Phones.Where(p => p.Price < 25000); // Samsung Galaxy S4, Samsung Galaxy S4, iPhone S4
    var selector2 = db.Phones.Where(p => p.Name.Contains("Samsung")); // Samsung Galaxy S4, Samsung Galaxy S4
    var phones = selector1.Except(selector2); // результат - iPhone S4

    foreach (var item in phones)
        Console.WriteLine(item.Name);
}
```

[Назад](#) [Содержание](#) [Вперед](#)



[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2017. Все права защищены.