



Выборка и проекция из базы данных

Последнее обновление: 31.10.2015



Для выборки применяется метод **Where**. Выберем из бд все модели, производитель которых - "Samsung":

```
using(PhoneContext db = new PhoneContext())
{
    var phones = db.Phones.Where(p=> p.Company.Name=="Samsung");
    foreach (Phone p in phones)
        Console.WriteLine("{0}.{1} - {2}", p.Id, p.Name, p.Price);
}
```

Для выборки одного объекта мы можем использовать метод **Find()**. Данный метод не является методом Linq, он определен у класса DbSet:

```
Phone myphone = db.Phones.Find(3); // выберем элемент с id=3
```

Но в качестве альтернативы мы можем использовать методы Linq **First()/FirstOrDefault()**. Они получают первый элемент выборки, который соответствует определенному условию. Использование метода FirstOrDefault() является более гибким, так как если выборка пуста, то он вернет значение null. А метод First() в той же ситуации выбросит ошибку.

```
Phone myphone = db.Phones.FirstOrDefault(p=>p.Id==3);
if(myphone!=null)
    Console.WriteLine(myphone.Name);
```

Теперь сделаем проекцию. Допустим, нам надо добавить в результат выборки название компании. Мы можем использовать метод **Include** для подсоединения к объекту связанных данных из другой таблицы: `var phones = db.Phones.Include(p=>p.Company)`. Но не всегда нужны все свойства выбираемых объектов. В этом случае мы можем применить метод **Select** для проекции извлеченных данных на новый тип:

```
using(PhoneContext db = new PhoneContext())
{
    var phones = db.Phones.Select(p => new
    {
        Name = p.Name,
        Price = p.Price,
        Company = p.Company.Name
    });
    foreach (var p in phones)
        Console.WriteLine("{0} ({1}) - {2}", p.Name, p.Company, p.Price);
}
```

В итоге метод Select из полученных данных спроецирует новый тип. В данном случае мы получим данные анонимного типа, но это также может быть определенный пользователем тип. Например:

```
public class Model
{
    public string Name { get; set; }
    public string Company { get; set; }
```

```
public int Price { get; set; }  
}
```

И спроецируем выборку на этот тип:

```
var phones = db.Phones.Select(p => new Model  
{  
    Name = p.Name,  
    Price = p.Price,  
    Company = p.Company.Name  
});  
foreach (Model p in phones)  
    Console.WriteLine("{0} ({1}) - {2}", p.Name, p.Company, p.Price);
```

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



**Стилизатор изображений
Ашманова.** ▾

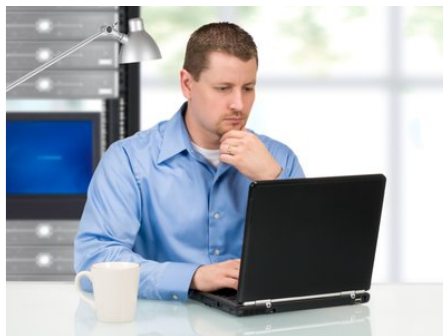
ashmanov.net



Яндекс.Директ

**Курсы Learn Python
в Минске** ▾

itstep.by



Яндекс.Директ

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

Контакты для связи: metanit22@mail.ru

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