

IQueryable. Complex task

Легенда

Предположим, у Вас есть очень специфическая база данных, для которой еще не реализован Linq Provider (а ORM - и подавно). Перед Вами стоит задача - реализовать Linq провайдер для доступа к данным.

Задание

В качестве объекта назначения выберите любую базу данных (к примеру, MS SQL, Postgre, Oracle*). Ваш *Linq Provider* должен позволять при помощи *IQueryable* транслировать запросы с поддержкой:

1. **Операторов:**
 - a. Select ... From ... Where
 - b. >, <, =
 - c. AND
2. **Типы данных:**
 - a. Numeric
 - b. String

*Можно использовать и NoSQL, переделав соответствующим образом запрос, приведенный в задании.

Что должно получиться в итоге

- **На входе:**

Нечто наподобие примера из [задания для Week2](#) (E3SProviderTests.cs), но уже для вашей базы и в соответствии с текущим заданием:

```
[Fact(Skip = "This test is provided to show the general idea of usage.")]
0 references
public void WithProvider()
{
    var employees = new E3SEntitySet<EmployeeEntity>(searchService.Value);

    foreach (var emp in employees.Where(e => e.Workstation == "EPRUIZHW0249"))
    {
        _testOutputHelper.WriteLine("{0} {1}", emp.NativeName, emp.StartWorkDate);
    }
}
```

- **На выходе:**

Набор данных из БД в соответствии с запросом. Оформить в виде Unit-теста.

Пример конечного запроса:

```
"SELECT * FROM [dbo].[products] WHERE UnitPrice > 100 AND [ProductType]='Customised Product'";
```

(LINQ провайдер должен генерировать не менее сложные запросы/операторы или их аналоги, специфичные для выбранной БД).

т.е., создаем *CustomEntitySet<T>: IQueryable<T>* (или *CustomDbSet* - если угодно), на который нанизывается query:

```
var productSet = new CustomEntitySet<ProductEntity>(...);
```

```
List<ProductEntity> products = productSet.Where(p => p.UnitPrice > 100 && p.ProductType ==  
"Customised Product").ToList();
```

NB: если Вы не располагаете достаточным количеством времени для того, чтобы развернуть БД и реализовать код для получения данных из БД (скажем, при помощи, ADO.Net), выполните хотябы основную логику, на входе у которой будет список Entity, а на выходе - валидная строка запроса, которую можно скопировать и выполнить в СУБД ручками.

Литература и примеры

1. Изучите [основное задание для week 2](#). В нем приводится пример имплементации IQueryableProvider. Unit- тесты, приведенные в коде, помогут вам разобраться.
2. Также, вы сможете подчерпнуть полезную информацию из [поста на Хабре](#).

