



ORM для .NET Core: наш вклад в open-source

Обо мне

Фадеев Алексей Сергеевич
Старший C# разработчик
Компания Sibedge, г.Томск

Сфера деятельности

Backend-разработка
Работа с СУБД, оптимизация
Используем: PostgreSQL, Microsoft SQL Server

PgConf 2019

Большой обзорный доклад по ORM

<https://pgconf.ru/2019/242760>

Проблемные запросы

Нескалярные типы: поиск по элементу

Рекурсивные запросы

Некоторые сложные конструкции, например `COUNT(DISTINCT ...)`

Open Source проекты готовы к нашим вкладам!

Плюсы ORM

Проверка синтаксиса

Особенно при изменении структуры

Защита от SQL-инъекций

Экранирование спецсимволов

Тестирование на моках

Проблемные запросы

```
SELECT "CustomerId", COUNT(*) AS "DealCount"  
FROM "Deal"  
GROUP BY "CustomerId"
```

```
var items = context.Deals  
    .GroupBy(x => x.CustomerId)  
    .Select(g => new  
    {  
        CustomerId = g.Key,  
        DealCount = g.Count()  
    })  
    .ToList();
```

Проблемные запросы

```
SELECT "CustomerId",  
       COUNT(*) AS "DealCount",  
       COUNT(DISTINCT CompanyId) AS "CompanyDealCount"  
FROM "Deal"  
GROUP BY "CustomerId"
```

```
var items = context.Deals  
    .GroupBy(x => x.CustomerId)  
    .Select(g => new  
    {  
        CustomerId = g.Key,  
        DealCount = g.Count(),  
        // CompanyDealCount = g.DistinctCount(x => x.CompanyId)  
        CompanyDealCount = g.Select(x => x.CompanyId).Distinct().Count()  
    })  
    .ToList();
```

Библиотека EFSqlTranslator

Библиотека генерирует SQL-запросы вместо Entity Framework

Запрос выполняется через Dapper

```
Install-Package EFSqlTranslator.Translation
```

```
Install-Package EFSqlTranslator.EFModels
```

```
.Net Standard
```

```
var query = context.Employees  
    .Where(x => x.Salary > 1000000);
```

```
var moneybags = db.Query(query,  
    new EFModelInfoProvider(db),  
    new PostgreSQLObjectFactory(),  
    out var sql);
```







НЕПроблемные запросы

```
SELECT "CustomerId",  
       COUNT(*) AS "DealCount",  
       COUNT(DISTINCT CompanyId) AS "CompanyDealCount"  
FROM "Deal"  
GROUP BY "CustomerId"
```

```
var itemsQuery = context.Deals  
    .GroupBy(x => x.CustomerId)  
    .Select(g => new  
    {  
        CustomerId = g.Key,  
        DealCount = g.Count(),  
        CompanyDealCount = g.DistinctCount(x => x.CompanyId)  
    });
```

```
var items = db.Query(query, new EFModelInfoProvider(db),  
    new PostgresQlObjectFactory(), out var sql);
```


Commits on Aug 22, 2019

update version ethanli83 committed on 22 Aug ✓	 73f6cd2	
#57 - Aggregation functions could generate ArgumentNullException ethanli83 committed on 22 Aug	 8856231	
Merge pull request #56 from alexeyfadeev/master ... ethanli83 committed on 22 Aug ✓	Verified  f0e38f6	





Commits on Aug 8, 2019

DistinctCount worked incorrect on child relation. Fixed. alexeyfadeev committed on 8 Aug ✓	 facb7ac	
--	---	---

Commits on Aug 7, 2019

DistinctCountTranslationTests added. ... alexeyfadeev committed on 7 Aug ✓	 df54635	
--	---	---

Commits on Jul 31, 2019

update version ethanli83 committed on 31 Jul ✓	 7c1c7eb	
Add unit test ethanli83 committed on 31 Jul ✓	 0f9070d	

Commits on Jul 30, 2019

Merge pull request #55 from alexeyfadeev/master ... ethanli83 committed on 30 Jul ✓	Verified  8a5a3cc	
---	--	---

Commits on Jul 24, 2019

#53 - Feature: DistinctCount alexeyfadeev committed on 24 Jul ✓	 033d7fe	
---	---	---

Поддержка нескалярных типов

Массивы, hstore

```
public class Model
{
    // text[]
    public string[] Keywords { get; set; }

    // integer[]
    public int[] Ids { get; set; }

    // timestamp[]
    public DateTime[] TimeStamps { get; set; }

    // hstore
    public Dictionary<string, string> ExtraFields { get; set; }
}
```

Поддержка не скалярных типов

Работает маппинг в обе стороны

Условия поиска по элементу не поддерживается на стороне ORM

Уже реализовано в EFSqlTranslator!

Условия поиска по элементу массива на стороне ORM

```
var itemsQuery = context.Articles
    .Where(x => x.KeyWords.Contains("database"));
```

```
var items = db.Query(query, new EFModelInfoProvider(db),
    new PostgresqlObjectFactory(), out var sql);
```

```
SELECT * FROM "Articles"
WHERE 'database' = ANY ("KeyWords")
```

Подан pull-request в EFSqlTranslator

Условия поиска по ключу hstore (key-value тип)

```
var itemsQuery = context.Articles
    .Where(x => x.Tags.ContainsKey("author"));
```

```
var items = db.Query(query, new EFModelInfoProvider(db),
    new PostgresqlObjectFactory(), out var sql);
```

```
SELECT * FROM "Articles"
WHERE "Tags" ? 'author'
```

На очереди pull-request в EFSqlTranslator

Условия поиска по значению ключа hstore (key-value тип)

```
var pair = new KeyValuePair<string, string> ("year", "2020");
```

```
var itemsQuery = context.Articles  
    .Where(x => x.Tags.Contains(pair));
```

```
var items = db.Query(query, new EFModelInfoProvider(db),  
    new PostgresqlObjectFactory(), out var sql);
```

```
SELECT * FROM "Articles"  
WHERE "Tags" @> 'year=>2020'
```

Планы: массивы

Более сложные условия поиска по элементу массива на стороне ORM

```
SELECT * FROM "Products"  
WHERE 4 <= ALL ("Ratings")
```

Планы: массивы

Более сложные условия поиска по элементу массива на стороне ORM

```
SELECT * FROM "Products"  
WHERE 4 <= ALL ("Ratings")
```

```
var query = context.Products  
    .Where(x => x.Ratings.All(r => r >= 4));
```


Планы: jsonb

Более сложные условия поиска по элементу массива на стороне ORM

```
SELECT * FROM "Items"  
  WHERE data @> '{ "address": { "postcode": "634000" } }'
```

```
var query = context.Items  
    .Where(x => x.Data  
    .Contains(new { address = new { postcode = "634000" } }));
```

Dapper – проблема с массивами

```
var itemsQuery = context.Articles  
    .Where(x => x.KeyWords.Contains("database"));
```

```
SELECT * FROM "Articles"  
WHERE 'database' = ANY ("KeyWords")
```

OK

Dapper – проблема с массивами

```
var itemsQuery = context.Articles
    .Where(x => x.KeyWords.Contains("database"))
    .Select(x => x.Data);
```

```
SELECT data FROM "Articles"
WHERE 'database' = ANY ("KeyWords")
```

Error

Dapper – проблема с hstore

```
var pair = new KeyValuePair<string, string> ("name", "Alexey");
```

```
var itemsQuery = context.Articles  
    .Where(x => x.Tags.Contains(pair));
```

```
SELECT * FROM "Articles"  
WHERE "Tags" @> '"name"=>"Alexey"'
```

Npgsql.PostgresException: "42883: оператор не существует: hstore @> text"

Далёкие планы

Запросы с рекурсией

Зависит от возможностей C#/LINQ

Далёкие планы: другие ORM

EntityFramework Core

LINQ to DB

Разработчик .NET

Может приносить пользу Postgres

Может развивать open-source

Присоединяйтесь!

Контакты

FadeevAS@sibedge.com

<https://vk.com/fadeev>

<https://www.facebook.com/alexey.fadeev.3745>

Ссылки

EFSqlTranslator

<https://github.com/ethanli83/EFSqlTranslator>

Большой обзорный доклад по ORM

<https://pgconf.ru/2019/242760>



ВСЁ!