

تذکر: این مطلب و نکته برای تا EF 6.1.3 تهیه شده‌است و ممکن است در نگارش‌های آتی آن وجود نداشته یا برطرف شده‌باشد.

کوئری ذیل را در نظر بگیرید:

```
var productsList1 = ctx.Products.Where(product => product.Id > 1)
    .Include(product => product.Category)
    .Include(product => product.User)
    .Where(
        product =>
            product.Category.Title.Contains("t") && product.Category.Id > 1 && product.Price > 100)
    .OrderBy(product => product.Price)
    .ToList();
```

به نظر شما این کوئری چند Join را ایجاد می‌کند؟

احتمالا شاید عنوان کنید که به ازای هر Include یک join خواهیم داشت. بنابراین دو جویین به جداول کاربران و گروه‌های محصول‌ها ایجاد می‌شود.

اما ... در واقعیت این کوئری را تولید می‌کند:

```
SELECT
    [Extent1].[Id] AS [Id],
    [Extent1].[Name] AS [Name],
    [Extent1].[Price] AS [Price],
    [Extent1].[CategoryId] AS [CategoryId],
    [Extent1].[UserId] AS [UserId],
    [Extent3].[Id] AS [Id1],
    [Extent3].[Name] AS [Name1],
    [Extent3].[Title] AS [Title],
    [Extent3].[UserId] AS [UserId1],
    [Extent4].[Id] AS [Id2],
    [Extent4].[Name] AS [Name2]
FROM    [dbo].[Products] AS [Extent1]
INNER JOIN [dbo].[Categories] AS [Extent2] ON [Extent1].[CategoryId] = [Extent2].[Id]
LEFT OUTER JOIN [dbo].[Categories] AS [Extent3] ON [Extent1].[CategoryId] = [Extent3].[Id]
LEFT OUTER JOIN [dbo].[Users] AS [Extent4] ON [Extent1].[UserId] = [Extent4].[Id]
WHERE ([Extent1].[Id] > 1) AND ([Extent2].[Title] LIKE N'%t%') AND ([Extent1].[CategoryId] > 1) AND
([Extent1].[Price] > 100)
ORDER BY [Extent1].[Price] ASC
```

اگر به قسمت جویین‌های آن دقت کنید دوبار جویین به جدول Categories را می‌توانید مشاهده کنید. این دو جویین حاصل یکبار Include جدول Categories و یکبار استفاده از navigation property آن در قسمت where است.

این باگ [در اینجا](#) گزارش شده، ولی به نظر هنوز برطرف نشده‌است یا مجددا ظاهر شده‌است.

برای رفع آن در حال حاضر بهترین راه حل استفاده از روش ذیل است:

```
var query2 = from product in ctx.Products
    let category = product.Category
    where product.Id > 1
    where category.Title.Contains("t") && category.Id > 1 && product.Price > 100
    select new { product, category };
var productsList2 = query2.ToList();
```

در اینجا قسمت Include کلا حذف شده و همچنین گروه‌ها توسط یک متغیر موقتی که با let ایجاد شده‌است، استفاده می‌شود. خروجی آن کوئری ذیل است:

```

SELECT
    [Extent1].[Id] AS [Id],
    [Extent1].[Name] AS [Name],
    [Extent1].[Price] AS [Price],
    [Extent1].[CategoryId] AS [CategoryId],
    [Extent1].[UserId] AS [UserId],
    [Extent2].[Id] AS [Id1],
    [Extent2].[Name] AS [Name1],
    [Extent2].[Title] AS [Title],
    [Extent2].[UserId] AS [UserId1]
FROM    [dbo].[Products] AS [Extent1]
INNER JOIN [dbo].[Categories] AS [Extent2] ON [Extent1].[CategoryId] = [Extent2].[Id]
WHERE ([Extent1].[Id] > 1) AND ([Extent2].[Title] LIKE N'%t%') AND ([Extent2].[Id] > 1) AND
([Extent1].[Price] > 100)

```

همانطور که مشاهده می‌کنید، اینبار فقط یکبار جوین به جدول گروه‌ها تشکیل شده‌است.

چند نکته:

-در کوئری let دار، اگر در قسمت select نهایی فقط product ذکر شود، هرچند جوین به جدول گروه‌ها تشکیل می‌شود اما فیلدهای این جدول انتخاب نخواهند شد.

-معادل کوئری LINQ نوشته شده را اگر بخواهیم توسط متدهای الحاقی بازنویسی کنیم، به کوئری ذیل خواهیم رسید:

```

var query2ChainedVersion = ctx.Products
    .Select(product => new { product, category = product.Category })
    .Where(@t => @t.product.Id > 1)
    .Where(@t => @t.category.Title.Contains("t") && @t.category.Id > 1 && @t.product.Price > 100)
    .Select(@t => new { @t.product , @t.category });

```

اگر علاقمند به آزمایش این باگ هستید، کدهای کامل آنرا از اینجا می‌توانید دریافت کنید:

[Sample38.zip](#)

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۵:۱۶ ۱۳۹۴/۰۱/۰۱

برنامه‌ی [DNTProfiler](#) قابلیت یافتن جویین‌های تکراری را نیز دارا است:

The screenshot displays the DNT Profiler v1.0.808.0 application interface. The top bar shows the Server Uri as http://localhost:8080 and a checkbox for Allow Remote Connections. The left sidebar contains a 'Plugins' section with 32 items, including 'Application' (2), 'Loggers' (8), and 'Alerts' (13). The 'Duplicate Joins' plugin is highlighted with a red box and a red circle containing the number 1. The main panel shows a table of duplicate joins with columns: Process Id, Process Name, AppDomain Id, and AppDomain Name. The table lists two entries for 'vstest.executionengine.x86' with AppDomain Id 2 and AppDomain Name 'UnitTestAdapter: Running test'. Below the table, the SQL command for Command Id 9 is displayed, showing a query with multiple joins and a WHERE clause. The SQL text is as follows:

```
10 [Extent3].[UserId] AS [UserId1],  
11 [Extent4].[Id] AS [Id2],  
12 [Extent4].[Name] AS [Name2]  
13 FROM [dbo].[Products] AS [Extent1]  
14 INNER JOIN [dbo].[Categories] AS [Extent2] ON  
15 LEFT OUTER JOIN [dbo].[Categories] AS [Extent3] ON  
16 LEFT OUTER JOIN [dbo].[Users] AS [Extent4] ON  
17 WHERE ([Extent1].[Id] > 1) AND ([Extent2].[Id] > 1)  
18 ORDER BY [Extent1].[Price] ASC
```