

## عنوان: اعمال توابع تجمعی بر روی چند ستون در Entity framework

نویسنده: وحید نصیری

تاریخ: ۸:۰۱۳۹۲/۰۸/۰۷

آدرس: [www.dotnettips.info](http://www.dotnettips.info)

برچسب‌ها: Entity framework

فرض کنید که می‌خواهیم معادل کوئری زیر را که اعمال توابع تجمعی به چند ستون است،

```
SELECT sum([Rating_TotalRating]), sum([Rating_TotalRaters]), sum([Rating_AverageRating]) FROM [BlogPosts]
```

در Entity framework به کمک LINQ to Entities تهیه کنیم.

نکته‌ای که در اینجا وجود دارد، نبود گروه بندی (حداقل به ظاهر) در کوئری نوشته شده است. اما واقعیت این است که یک بانک اطلاعاتی به صورت ضمنی در مورد یک چنین کوئری‌هایی نیز گروه بندی را انجام می‌دهد. برای اینکار، کل رکوردهای مدنظر را یک گروه تصور می‌کند.

اگر سعی کنیم چنین کوئری را توسط عبارات LINQ ایجاد کنیم، در سعی اول به چنین کوئری خواهیم رسید که اصلاً کامپایل نمی‌شود:

```
context.BlogPost.Select(r =>
    new
    {
        Sum1 = r.Sum(x => x.RatingTotalRating),
        Sum2 = r.Sum(x => x.RatingTotalRaters),
        Sum3 = r.Sum(x => x.RatingAverageRating)
    }).FirstOrDefault();
```

بنابراین به نظر می‌رسد که شاید بهتر باشد از روش ذیل استفاده کنیم:

```
var sum1 = context.BlogPost.Sum(x => x.RatingTotalRating);
var sum2 = context.BlogPost.Sum(x => x.RatingTotalRaters);
var sum3 = context.BlogPost.Sum(x => x.RatingAverageRating);
```

این روش کار می‌کند و نهایتاً معادل نتایج کوئری اول را نیز حاصل خواهد کرد؛ اما با سه بار رفت و برگشت به بانک اطلاعاتی که اصلاً بهینه نیست.

راه حل: ایجاد گروه بندی ضمنی SQL به صورت صریح در عبارات LINQ

```
context.BlogPost
    .GroupBy(dummyNumber => 0)
    .Select(r =>
        new
        {
            Sum1 = r.Sum(x => x.RatingTotalRating),
            Sum2 = r.Sum(x => x.RatingTotalRaters),
            Sum3 = r.Sum(x => x.RatingAverageRating)
        }).FirstOrDefault();
```

در این کوئری جدید که بر اساس عدد ثابت صفر گروه بندی شده است، یک چنین SQL ایی تولید می‌شود:

```
SELECT TOP (1)
    [Extent1].[K1] AS [K1],
    Sum([Extent1].[A1]) AS [A1],
    Sum([Extent1].[A2]) AS [A2],
    Sum([Extent1].[A3]) AS [A3]
FROM ( SELECT
    0 AS [K1],
    [Extent1].[RatingTotalRating] AS [A1],
    [Extent1].[RatingTotalRaters] AS [A2],
    [Extent1].[RatingAverageRating] AS [A3]
```

```
FROM [dbo].[BlogPosts] AS [Extent1]  
) AS [Extent1]  
GROUP BY [K1]
```

ابتدا یک ستون فرضی با مقدار ثابت صفر به رکوردها اضافه می‌شود. سپس بر اساس این ستون فرضی، کلیه ردیف‌ها گروه بندی شده و در ادامه توابع تجمعی بر روی آن‌ها اعمال می‌گردند. به این ترتیب تعداد رفت و برگشت‌ها به بانک اطلاعاتی به همان یک مورد کاهش خواهد یافت.