

فرض کنید مدل متناظر با جدول بانک اطلاعاتی دانشجویان، به صورت ذیل تعریف شده‌است و دارای یک فیلد محاسباتی است:

```
public class Student
{
    public int Id { get; set; }

    [Required]
    [StringLength(450)]
    public string LastName { get; set; }

    [Required]
    [StringLength(450)]
    public string FirstName { get; set; }

    [NotMapped]
    public string FullName
    {
        get
        {
            return LastName + ", " + FirstName;
        }
    }
}
```

فیلد محاسباتی FullName را نمی‌توان در کوئری‌های EF بکار برد؛ زیرا کوئری‌های LINQ نوشته شده در اینجا باید قابلیت ترجمه‌ی به SQL را داشته باشند و چون در بانک اطلاعاتی برنامه، فیلدی به نام FullName وجود ندارد، نمی‌توان FullName را مورد استفاده قرار داد.

تبدیل خواص محاسباتی به کوئری‌های SQL به کمک DelegateDecompiler

هر «نمی‌توانی» را می‌توان تبدیل به یک پروژه‌ی ابتکاری کرد! و اینکار توسط پروژه‌ای به نام [DelegateDecompiler](#) انجام شده‌است.

کوئری متداول ذیل، قابل اجرا نیست و با یک استثناء متوقف می‌شود؛ زیرا همانطور که عنوان شد، FullName قابل تبدیل به SQL نیست.

```
var fullNames = context.Students.Select(x => x.FullName).ToList();
```

اما اگر همین کوئری را توسط DelegateDecompiler بازنویسی کنیم:

```
var fullNames = context.Students.Select(x => x.FullName).Decompile().ToList();
```

بدون مشکل اجرا می‌شود. اینبار FullName به صورت ذیل تبدیل به عبارت SQL معادلی خواهد شد:

```
SELECT
    [Extent1].[LastName] + N', ' + [Extent1].[FirstName] AS [C1]
FROM [dbo].[Students] AS [Extent1]
```

برای استفاده‌ی از DelegateDecompiler دو کار باید انجام شود:

الف) خاصیت محاسباتی مدنظر را با ویژگی Computed مزین کنید:

```
[NotMapped]
[Computed]
public string FullName
```

ب) از متد الحاقی Decpile در کوئری تهیه شده استفاده نمائید.

استفاده از DelegateDecompiler به همراه AutoMapper

فرض کنید ViewModel ایی که قرار است به کاربر نمایش داده شود، ساختار ذیل را دارد:

```
public class StudentViewModel
{
    public int Id { get; set; }
    public string FullName { get; set; }
}
```

کوئری ذیل که از [Project To](#) مخصوص AutoMapper جهت نگاشت اطلاعات دریافتی از بانک اطلاعاتی به StudentViewModel استفاده می‌کند، با توجه به اینکه کار نوشتن Select را به صورت خودکار بر اساس خاصیت FullName انجام می‌دهد، قابلیت اجرای بر روی بانک اطلاعاتی را نخواهد داشت:

```
var students = context.Students.Project().To<StudentViewModel>().ToList();
```

برای رفع این مشکل تنها کافی است از متد Decpile کتابخانه‌ی DelegateDecompiler به نحو ذیل استفاده کنیم:

```
var students = context.Students
    .Project()
    .To<StudentViewModel>()
    .Decpile()
    .ToList();
```

اینبار کوئری ارسال شده‌ی به بانک اطلاعاتی، یک چنین شکلی را پیدا می‌کند:

```
SELECT
    [Extent1].[Id] AS [Id],
    [Extent1].[LastName] + N', ' + [Extent1].[FirstName] AS [C1]
FROM [dbo].[Students] AS [Extent1]
```

کدهای کامل این مطلب را [از اینجا](#) می‌توانید دریافت کنید.