

فرض کنید در روش EF Database First می‌خواهید فیلدی به مدل اضافه شود که در دیتابیس وجود ندارد، درواقع فیلدی محاسباتی به مدل اضافه کنید. راه حل چیست؟ اولین روشی که ممکن است به ذهن برسد این است که به کلاس مدل که از جدول دیتابیس ساخته شده، فیلدی محاسباتی اضافه می‌کنیم.

```
public class Person
{
    public string FullName {
        get
        {
            return FirstName + " " + LastName;
        }
    }
}
```

به نظر راه حل درستی می‌رسد، اما مشکل این روش چیست؟ اگر مدل برنامه از روی دیتابیس بروزشود، چه اتفاقی می‌افتد؟ خب قاعدتا این فیلد محاسباتی از دست می‌رود و باید دوباره آن را به کلاس مدل جدید و بروز شده اضافه کنیم. که به نظر راه حل منطقی و خوبی نمی‌رسد. در چنین مواقعی می‌توان به جای اینکه این پراپرتی را به کلاس تولید شده از روی دیتابیس اضافه کرد، این فیلد را به یک Partial Class از کلاس تولید شده که در همان پروژه و فضای نام کلاس تولید شده از دیتابیس قرار دارد، اضافه کرد. به این ترتیب با هر بار بروز شدن مدل از روی دیتابیس، این فیلد از بین نمی‌رود.

```
public partial class Person
{
    public string FirstName {get; set;}
    public string LastName {get; set;}
}
public partial class Person
{
    public string FullName {
        get
        {
            return FirstName + " " + LastName;
        }
    }
}
```

اما این روش محدودیت‌هایی نیز دارد :

1. همه قسمت‌های Partial Class باید در یک اسمبلی باشند.
2. پراپرتی‌های درون Partial Class در دیتابیس ذخیره نمی‌شوند.
3. ونیز این پراپرتی‌ها در عبارات Linq قابل استفاده نیستند. چون عبارت Linq در نهایت به یک رشته SQL تبدیل شده و در دیتابیس اجرا می‌شود. البته با فرض این که دیتاپرووایدر، SQL باشد.

## نظرات خوانندگان

نویسنده: محسن خان  
تاریخ: ۲۱:۴۱ ۱۳۹۲/۰۸/۰۵

در مورد روش دوم؛ (با توجه به جمله بندی) آیا در روش اول این خاصیت‌های محاسباتی در بانک اطلاعاتی ذخیره می‌شوند؟ ضمناً جهت تکمیل بحث، این خاصیت‌ها (هر دو حالت) در عبارات LINQ to Objects قابل استفاده هستند.

نویسنده: جمشیدی فر  
تاریخ: ۸:۳۸ ۱۳۹۲/۰۸/۰۶

در مورد سوال اول، خیر؛ در روش اول هم این خاصیت‌ها در دیتابیس ذخیره نمی‌شوند. اینجا مساله این است که ما می‌خواهیم با هر بار اپدیت مدل از دیتابیس، فیلد محاسباتی دوباره محاسبه شود و از بین نرود.

درمورد سوال دوم، شما درست می‌فرمایید. من باید Linq رو به Linq to Entity تغییر بدم.