

یکی از مشکلات برنامه نویسان، نوشتن هزاران رکورد در دیتابیس در مدت زمان بسیار کوتاهی است که عموماً این کار در هنگام خواندن اطلاعات از فایل‌های اکسل و گاهی از فایل‌های text ای اتفاق می‌افتد. برای مثال در زمان نوشتن این اطلاعات، با Timeout مواجه شده و اگر هم Timeout ندهد بسیار کند عمل می‌کند. در این پست قصد داریم روش نوشتن هزاران رکورد را در کسری از ثانیه توسط EF Code first مورد بررسی قرار دهیم و در نهایت مقایسه ای با AddRange در EntityFramework داشته باشیم. خوب؛ در ابتدا مدلی را با نام Personel را به شکل زیر طراحی مینماییم.

```
public class Personel
{
    [Key]
    public int PersonelID { get; set; }

    [MaxLength(15)]
    public string Name { get; set; }

    [MaxLength(25)]
    public string Family { get; set; }

    [MaxLength(10)]
    public string CodeMelli { get; set; }
}
```

سپس این مدل را در Context خود معرفی نمایید همانند کلاس زیر:

```
public class PersonalContext : DbContext
{
    public DbSet<Personel> Personel { get; set; }

    public override int SaveChanges()
    {
        return base.SaveChanges();
    }
}
```

برای ساختن دیتابیس در EF CodeFirst Entityframework میتوانید به [سری آموزشی CodeFirst در سایت جاری](#) مراجعه نمایید. اکنون همه چیز مهیا است برای انجام عملیات Bulk Insert. در ابتدا پاورشل نیوگت را باز کرده و [پکیج مورد نظر را](#) با توجه به نسخه Ef استفاده شده، به پروژه اضافه نمایید. همانند دستور زیر:

```
Install-Package EntityFramework.BulkInsert-ef6
```

بعد از نصب پکیج مورد نظر، باید لیستی از موجودیت‌ها را از یک فایل اکسل خوانده و به BulkInsert EF ارسال نماییم. برای این کار مانند زیر عمل مینماییم.

```
public ActionResult Insert()
{
    int Counter = 1000;
    List<Personel> Lst = new List<Personel>();
    // شبیه سازی خواندن رکوردها از فایل اکسل
    for (int i = 0; i < Counter; i++)
    {
        Lst.Add(new Personel
        {
            CodeMelli = "0000000000",
            Family = "Karimi",
            Name = "Mohammad"
        });
    }
}
```

```

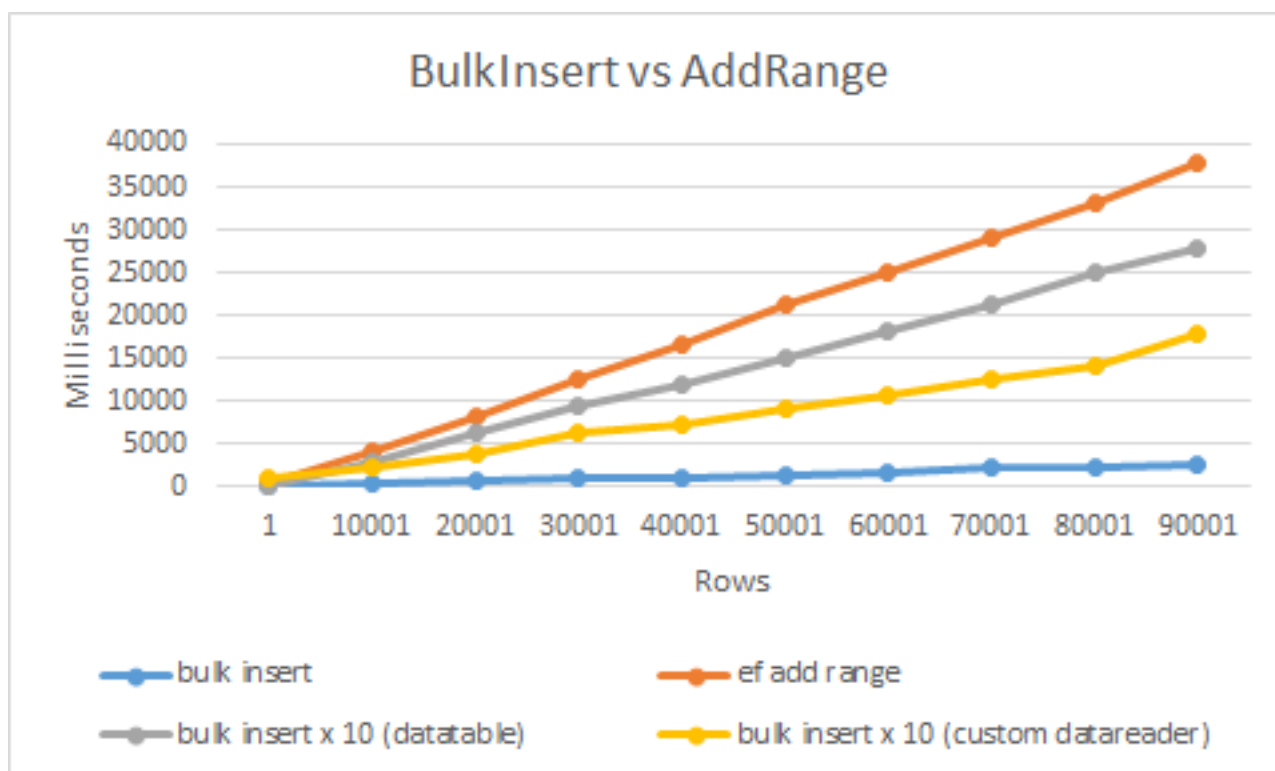
    });
}

PersonalContext db = new PersonalContext();
db.BulkInsert(Lst);
db.SaveChanges();
return View();
}

```

تنها نکته‌ی استفاده از متد BulkInsert، اضافه نمودن ارجاعی از `using EntityFramework.BulkInsert.Extensions;` به بالای کلاس جاری است.

در شکل زیر می‌توانید مقایسه‌ای بین `bulkInsert` و `AddRange` را در تعداد رکوردهای نوشته شده و مدت زمان صرف شده برای نوشتن در دیتابیس، مشاهده نمایید.



نظرات خوانندگان

نویسنده: سعید عبوسی
تاریخ: ۲۳:۲۰ ۱۳۹۲/۱۲/۲۸

سلام؛ چطور میتونم مقادیر یک دیتابیل رو به این صورت در دیتابیس درج کنم؟

نویسنده: ژوپتر
تاریخ: ۸:۲۱ ۱۳۹۲/۱۲/۲۹

```
var personnels = dataSet.Tables["Personnel"].AsEnumerable();
db.BulkInsert(personnels.ToList());
```

نویسنده: حسین
تاریخ: ۱۱:۴۲ ۱۳۹۳/۰۱/۰۳

با سلام وتشکر
کد زیر نباید یه این صورت نوشته بشه؟

```
db.Personel.BulkInsert(Lst);
```

اگر نه چجوری بدون دادن نام جدول عمل اضافه کردن رو انجام میده؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۵ ۱۳۹۳/۰۱/۰۳

خیر. این پروژه سورس باز هست؛ [اینجا](#).
با استفاده از Reflection و همچنین اطلاعات موجود در Context، جداول را تشخیص می‌دهد.

نویسنده: sharp
تاریخ: ۲۱:۳۴ ۱۳۹۳/۰۵/۲۱

سلام.
موقع استفاده از BulkInsert خطای زیر برای من رخ میده. من یک کلاس Category دارم که شامل دو فیلد CategoryID (کلید و identity) و CategoryName هستش. کدم رو به شکل زیر نوشتم :

```
using(NewsEntities context = new NewsEntities())
{
    try
    {
        List<Category> catlist = new List<Category>();
        for (int i = 0; i < 1000; i++)
        {
            Category cat = new Category { CategoryName = "cat" + i.ToString() };
            catlist.Add(cat);
        }
        context.BulkInsert(catlist);
        context.SaveChanges();
        MessageBox.Show("Added");
    }
    catch(Exception ex)
    {
        MessageBox.Show(Error.ShowError(ex));
    }
}
```

خطایی که رخ میده :

The given key was not present in the dictionary
همین کد رو با AddRange که می نویسم بدون مشکل کار میکنه.

نویسنده: وحید نصیری
تاریخ: ۲۳:۲۱ ۱۳۹۳/۰۵/۲۱

مشکلی مشاهده نشد: [BulkInsertTests.zip](#)

نویسنده: احمد
تاریخ: ۹:۲۱ ۱۳۹۳/۰۶/۲۹

سلام، من از bulk رو nuget نصب کردم و به using هم اضافه کردم ولی وقتی که به این خط می رسه

```
db.BulkInsert(entities);
```

میگه BulkInsertExtention.cs NotFound, یه پنجره باز میشه که این فایلو انتخاب کنم!
کسی به این مشکل برخورد کرده؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۶ ۱۳۹۳/۰۶/۲۹

اگر در حین کار با کتابخانه های مختلف، صفحه دیالوگ گشودن فایل به همراه پیام cs file not found مشاهده شد، این صفحه را لغو کنید تا استثنای اصلی نمایش داده شود. همچنین در EF باید Inner exception را هم بررسی کنید.
علت اصلی هم به اینجا بر می گردد که فایل pdb، به همراه کتابخانه ی مورد نظر توزیع شده و این فایل حاوی محل قرارگیری سورس کتابخانه و همچنین شماره سطر مرتبط با استثناء است. چون این سورس بر روی سیستم شما موجود نیست و فایل pdb نیز پیوست شده، صفحه ی باز کردن فایل یافت نشده، نمایش داده می شود.

نویسنده: احمد
تاریخ: ۹:۳۴ ۱۳۹۳/۰۶/۲۹

ممنون از پاسختون البته اینو فراموش کردم بگم که کد اجرا میشه! ولی خب این پنجره هم میاد!
خب من از nuget نصبش کردم، چطوری این فایلو بزارم کنار سورس؟

نویسنده: وحید نصیری
تاریخ: ۹:۳۷ ۱۳۹۳/۰۶/۲۹

فایل EntityFramework.BulkInsert.pdb را از پوشه ی bin برنامه حذف کنید. این فایل و اطلاعات موجود در آن، فقط بر روی سیستمی که اصل کتابخانه بر روی آن کامپایل شده معتبر است. برای مثال در این فایل ثبت شده فایل BulkInsertExtention.cs در مسیر c:\prog\lib\user12\bluk قرار دارد که فقط بر روی سیستم نویسنده ی اصلی این کتابخانه معتبر است. اگر شما هم نیاز به فایل pdb معتبری دارید، سورس این کتابخانه را دریافت کنید. دستی آن را کامپایل کرده و سپس ارجاعی را به اسمبلی نهایی بدهید. اینبار به صورت خودکار فایل pdb معتبری در پوشه ی bin برنامه ی شما قرار خواهد گرفت.

نویسنده: رضایی
تاریخ: ۱۱:۵ ۱۳۹۳/۰۹/۰۹

با سلام

برنامه در سیستم اصلی بدرستی کار می کند اما موقع نصب بر روی سیستم دیگر خطای زیر را می دهد

Field_ConnectionString was not found in Type System.Data.SqlConnection Parametername:PropName

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۴ ۱۳۹۳/۰۹/۰۹

- کار ما رفع باگ‌های پروژه‌های ثالث نیست. باید این مساله را در [issue tracker](#) آن‌ها پیگیری کنید. این مشکل باید در پروژه‌ی اصلی برطرف شود و نه اینجا.
- در کل اگر به فایل [providerBase](#) آن مراجعه کنید، چنین تعریفی را دارد:

```
protected virtual string ConnectionString
{
    get
    {
        return (string)DbConnection.GetPrivateFieldValue("_connectionString");
    }
}
```

سورس را دریافت کنید و مورد فوق را به نحو ذیل تغییر دهید:

```
protected virtual string ConnectionString {
    get {
        return (string)DbConnection.ConnectionString;
    }
}
```

نویسنده: سید
تاریخ: ۱۴:۲۰ ۱۳۹۳/۰۹/۰۹

راه حل اصلی اینه که دانت فریم‌ورک 4.5.1 و بالاتر نصب کنه

نویسنده: وحید نصیری
تاریخ: ۱۴:۵۹ ۱۳۹۳/۰۹/۰۹

استفاده از `Get Private FieldValue("_connectionString")` یک «هک» هست و راه حل نیست. این فیلد خصوصی می‌تواند در یک نگارش خاص وجود داشته باشد و در دیگری خیر. به همین جهت private تعریف شده‌است.

نویسنده: علی یگانه مقدم
تاریخ: ۱۷:۵۵ ۱۳۹۴/۰۷/۰۵

برای پشتیبانی از ورژن‌های اخیر EF6 [این پکیج](#) را نصب کنید