

همونطور که میدونیم درج یکباره چندین رکورد هنگام استفاده از Entity Framework فعلا امکان پذیر نیست و باید از یک حلقه استفاده کرد و آنها رو یک به یک وارد کرد که هنگامی تعداد رکوردها زیاد باشن زمان اجرا یکم زیاد میشه. برای رفع این مشکل در EF Code First میتونین خاصیت AutoDetectChangesEnabled رو برای Context غیرفعال کنید که استفاده از این روش قبلا در [این](#) مقاله توضیح داده شده است. راه دیگه استفاده از SqlBulkCopy هست که میتوانید هنگام استفاده از ORM ها ازش استفاده کنید. اگه قبلا از ADO.NET استفاده کرده باشید و خواسته باشید تعداد زیادی رکورد رو بصورت همزمان وارد دیتابیس کنید حتما با SqlBulkCopy آشنایی دارید.

فرض کنید دارید در پروژه، از Entity Framework استفاده میکنید و یک مدل با نام Person دارید که تعریفش به صورت زیر است

```
public class Person
{
    public int PersonId { get; set; }
    public string Name { get; set; }
}
```

حالا میخوایم تعداد ۵۰۰۰ رکورد از Person رو یکجا وارد دیتابیس کنیم. برای استفاده از SqlBulkCopy، روش به این شکل هست که ابتدا یک DataTable ایجاد میکنیم. سپس ستونهای متناظر با جدول Person رو با استفاده از DataColumn ایجاد میکنیم و DataColumn های ایجاد شده رو به DataTable اضافه میکنیم و سپس اطلاعات رو وارد DataTable میکنیم و اون رو با استفاده از SqlBulkCopy وارد دیتابیس میکنیم که این روش یکم وقتگیر و خسته کننده است.

راه آسانتر استفاده از یک کتابخانه با نام EntityDataReader هست که توسط مایکروسافت نوشته شده که دیگه نیازی به ساختن DataTable نیست و این کتابخانه کارهای لازم رو خودش انجام میده. در پروژتون یک کلاس با نام EntityDataReader ایجاد کنید و سورس مربوط این کلاس رو از [اینجا](#) copy و در داخل کلاس paste کنید.

حالا یک لیست از Pesron با نام personList ایجاد مینماییم و با استفاده از یک حلقه تعداد ۵۰۰۰ تا نمونه از Person ایجاد و به لیست اضافه میکنیم.

```
var personList = new List<Person>();
for (var i = 0; i < 5000; i++)
{
    var person = new Person
    {
        Name = "John Doe",
    };
}
```

در ادامه برای استفاده از SqlBulkCopy نیاز به ConnectionString و نام جدول متناظر با کلاس Person در دیتابیس داریم.

اگر از پروژ وب استفاده میکنید میتونید با این خط کد ConnectionString رو که در فایل web.config ذخیره شده است بروگردونید که در اینجا DataConnection نام ConnectionString ذخیره شده در web.config هست.

```
var connectionString = ConfigurationManager.ConnectionStrings["DataConnection"].ConnectionString;
```

اگر از EF Code First استفاده میکنید و در تنظیمات Context خاصیت PluralizingTableNameConvention رو حذف کردیداید نام جدول dbo.Person هست و در غیر اینصورت db.People هست .

و در ادامه داریم:

```
var connectionString = ConfigurationManager.ConnectionStrings["DataConnection"].ConnectionString;
var bulkCopy = new SqlBulkCopy(connectionString) { DestinationTableName = "dbo.Person" };
bulkCopy.WriteToServer(personList.AsDataReader());
```

سرعت این روش بسیار بالاست و برای درجهای با تعداد بالا بهینه است.

برای ویرایش و حذف چندین رکورد بصورت همزمان متیونید از کتابخانه [Entity Framework Extended Library](#) استفاده کنید که امکانات دیگری هم داره و از طریق nuget هم قابل نصب است.

نظرات خوانندگان

نویسنده: مصطفی
تاریخ: ۲۰:۱۰ ۱۳۹۲/۰۳/۰۵

این روش یک مشکل خیلی بد دارد که اگر تو کلاس مدل از complextype استفاده کرده باشی دیگه همیشه از کلاس EntityDataReader استفاده کرد مشکل دوم اینه که اگر پراپرتی‌های مدلتون رو تو فایل کانفیگ به یک ستون غیر هم نام مپ کرده باشی بازم همیشه از این روش استفاده کرد

نویسنده: محسن اسماعیل پور
تاریخ: ۲۲:۱۲ ۱۳۹۲/۰۳/۰۵

برای رفع مشکل دوم میتونید از DataTable استفاده کنید و نام خاصیتی که ستون متناظرش در جدول فرق دارد رو هنگام تعریف DataColumn عوض کنید. روال کار همیشه فقط اضافه کاری داره.

نویسنده: حسنین
تاریخ: ۱۸:۴۵ ۱۳۹۲/۰۳/۰۶

جسارتا در انتهای مقاله به گمانم غلط تایپی یا اشتباه لبی باشد:
"و در غیر اینصورت db.Persons هست."

باید باشد: "و در غیر اینصورت db.People هست"

نویسنده: محسن اسماعیل پور
تاریخ: ۲۳:۱۵ ۱۳۹۲/۰۳/۰۶

با تشکر از توجه شما، اصلاح گردید.

نویسنده: وحید نصیری
تاریخ: ۸:۲۴ ۱۳۹۲/۰۳/۱۴

یک خبر خوب:

در 6 EF نسخه بتا، متدی اضافه شده به نام AddRange که افزودن تعداد زیادی رکورد رو سهولت بخشیده
[INSERTing many rows with Entity Framework 6 beta 1 and SQL Server Compact](#)

نویسنده: علی
تاریخ: ۱۱:۵۰ ۱۳۹۲/۰۳/۲۳

سلام

من خطای زیر رو دارم. برای ویرایش چند رکورد به صورت یکجا از Entity Framework Extended Library استفاده کردم. اما مشکلی دارم این است من یک جدول در پایگاه دارم که می‌خوام به صورت یکجا ویرایش کنم به این صورت که ابتدا شماره فیلد هایی که را قرار است تغییر دهم (فیلد ID) داخل یک لیست ریختم و سپس از جدول فقط اون فیلدها رو ویرایش کنم چند روش برای join جدول با این لیست امتحان کردم با join خطا می‌دهد اگر join نیز مشکلی نداشت مقدار برگشتی از نوع enumerable هست و زمانی که از update کلاس Entity Framework Extended Library استفاده میکنم خطا می‌دهد که لیستی که قرار است update شود باید از نوع dbquery باشد

Unable to create a constant value of type 'extendeexample.MyClass'. Only primitive types or enumeration types are supported in this context

سورس پروژه به همراه پایگاه داده
[project.rar](#)