

سلام :

سال نو مبارک ! امیدوارم سال بسیار خوبی در پیش داشته باشید :

از زمانی که استفاده از ORM‌های Code First رایج شده ، اجرای اسکریپت‌های طولانی جهت ایجاد دیتابیس خیلی استفاده ندارد ، اما حالت خاص همیشه پیش می‌آید.
مثلا قصد داریم پیش از آغاز برنامه پس از ایجاد دیتابیس توسط Entity Framework به یک سری جداول فیلدی با نوع داده‌ی Geometry اضافه کنیم. یا باید به دیتابیس یک سری Stored Procedure و View اضافه کرد.
Script ها ی Generate شده توسط SQL Server حاوی دستور Go هستند. ADO.NET اجرای Script که حاوی Go باشد را پشتیبانی نمی‌کند.

اما روش‌ها و ترفندهای زیادی برای اجرای یک فایل Script طولانی حاوی دستور Go روی دیتابیس وجود دارد :

```
private static string GetScript()
{
    string path = AppDomain.CurrentDomain.BaseDirectory +
        @"Scripts\script.sql";
    var file = new FileInfo(path);

    string script = file.OpenText().ReadToEnd();
    return script;
}

private static void ExecuteScript()
{
    string script = GetScript();

    //split the script on "GO" commands
    var splitter = new[] { "\r\nGO\r\n" };
    string[] commandTexts = script.Split(splitter,
        StringSplitOptions.RemoveEmptyEntries);
    foreach (string commandText in commandTexts)
    {
        using (var ctx = new DbContext())
        {
            if (!string.IsNullOrEmpty(commandText))
            {
                ctx.Database.ExecuteSqlCommand(commandText);
            }
        }
    }
}
```

در اینجا به جای آنکه تلاش کنیم یک فایل را روی دیتابیس یک جا اجرا کنیم دستورات را جدا کرده و تک به تک اجرا می‌کنیم.

بروزرسانی: [روش بهتر](#)

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۷:۱۷ ۱۳۹۲/۰۱/۰۳

ممنون. میشه برای بهبود سرعت insert های زیاد، از [SQL Bulk Copy](#) هم استفاده کرد.

نویسنده: شاهین کیاست
تاریخ: ۱۹:۲۲ ۱۳۹۲/۰۱/۰۳

بله درسته،

در سناریویی که برای خودم پیش آمده بود کوئری ها فقط Insert نبودند و همچنین فقط قرار بود یک بار در ابتدای برنامه اجرا شوند برای همین روی سرعت حساسیت نداشتم.

نویسنده: مهدی موسوی
تاریخ: ۱۹:۵۳ ۱۳۹۲/۰۱/۰۳

سلام؛ سال نو مبارک !)

آقای کیاست، بهترین روش برای اجرای Script هایی که حاوی کلمه کلیدی GO هستند، استفاده از SMO است. به کد زیر دقت کنید:

```
Server server = new Server(new ServerConnection() { ConnectionString = cnnStr });  
server.ConnectionContext.ExecuteNonQuery(sql);
```

اینجا همه چیز توسط SMO کنترل میشه و دیگه نیازی به آنالیز Script اصلی بر اساس عبارت GO نیست.

موفق باشید.

نویسنده: شاهین کیاست
تاریخ: ۰:۱۴ ۱۳۹۲/۰۱/۰۴

سلام قربان ؛

خیلی ممنون.

در استفاده از SMO در NET 4 مشکل داشتم در نتیجه از این روش که نیازمند به Assembly های SMO نیست استفاده کردم. کامنت شما را در انتهای مطلب قرار دادم.

نویسنده: میثم خوشبخت
تاریخ: ۱۷:۱۸ ۱۳۹۲/۰۱/۰۵

شاهین جان

ممنون از مطلب مفیدت

من این مورد رو قبلا نوشتم. به همین روشی که شما نوشتید. منتها تو یه Query با مشکل مواجه شد و اجرا نشد. واسه همین از روش زیر استفاده کردم که همه جا جواب میده:

```
var commandTexts=Regex.Split(script,@\"\\s*GO\\s+");  
foreach(var commandText in commandTexts)  
{  
    // Execution Code  
}
```

نویسنده: مهدی موسوی

تاریخ: ۱۵:۳۶ ۱۳۹۲/۰۱/۰۶

سلام.

خیر، همه جا جواب نمیده! عرض کردم، نیازی به جداسازی Script بر اساس GOهای موجود در اون ندارید. Script زیر رو در نظر بگیرید (بعنوان مثالی نقض) تا متوجه بشید چرا کد شما نیز ایراد داره:

```
USE [MyDB]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[GO AND EAT](
  [Id] [int] IDENTITY(1,1) NOT NULL,
  [MyVal] [decimal](18, 0) NOT NULL,
  ) ON [PRIMARY]
GO
```

لطفا اینو در نظر بگیرید که منظورم این نیست که با تصحیح Regex Pattern می‌تونید این مشکل رو نیز برطرف کنید، منظورم اینه که با استفاده از SMO می‌تونید از نوشتن کدهای Error-Prone دوری کنید. فقط همین. (:

موفق باشید.

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۸ ۱۳۹۲/۰۱/۰۶

[پشت صحنه SMO](#) برای علاقمندان در مورد نحوه جدا سازی batch ها از یکدیگر در عمل.

نویسنده: علی
تاریخ: ۱۵:۴۷ ۱۳۹۲/۰۲/۲۴

برای اجرای چنین دستوراتی نیازی به تکه تکه کردن نداریم فقط کافیه go رو از اسکریپت حذف کنیم خودش همرو اجرا می‌کنه قبلا خودم چنین کاری رو انجام دادم

نویسنده: محسن خان
تاریخ: ۱۶:۵۸ ۱۳۹۲/۰۲/۲۴

کل این بحث در مورد نحوه یافتن صحیح این Go بدون تخریب اطلاعات جانبی بود. حالا می‌خوای حذفش کن یا تکه تکه‌اش کن!