

Dialects در NHibernate کلاس‌هایی هستند جهت معرفی ویژگی‌های خاص بانک‌های اطلاعاتی مختلف؛ مثلاً SQL Server 2008 چه ویژگی‌های جدیدی دارد یا SQL Server CE 4.0 که جدیداً ارائه شده، امکان تعریف offset را در کوئری‌های خود میسر کرده (چیزی که قرار است در نگارش بعدی SQL Server اصلی(!) در دسترس باشد)، اکنون چگونه می‌توان این ویژگی را فعال کرد (باید Dialect آن به روز شود و ... همین). یک سری Dialect از پیش تعریف شده هم برای اکثر بانک‌های اطلاعاتی در NHibernate وجود دارد. ممکن است این Dialects پیش فرض الزاماً خواسته شما را برآورده نکنند یا مو به مو مستندات بانک اطلاعاتی مرتبط را پیاده سازی نکرده باشند و سؤال این است که اکنون چه باید کرد؟ آیا باید حتماً سورس‌ها را دستکاری و بعد کامپایل کرد؟ به این صورت هر بار با ارائه یک نگارش جدید NHibernate به مشکل خواهیم خورد چون باید کل عملیات تکرار شود.

خبر خوب اینکه می‌توان از همین Dialects موجود ارث بری کرد، سپس مواردی را که نیاز داریم override کرده یا به کلاس مشتق شده افزود. اکنون می‌توان از این Dialect سفارشی به جای Dialect اصلی استفاده کرد. در ادامه با یک نمونه آشنا خواهیم شد.

فرض کنید Dialect انتخابی مرتبط است با SQL Server CE استاندارد. کوئری ساده زیر را می‌نویسیم، به ظاهر باید کار کند:

```
var list = session.Query<SomeClass>().Where(x=>x.Date.Year==2011).ToList();
```

اما کار نمی‌کند! علت این است که تمام Dialects در NHibernate از یک Dialect پایه مشتق شده‌اند. در این Dialect پایه، تعریف تابع استخراج year از یک تاریخ به نحو زیر است:

```
extract(year, ?1)
```

اما در SQL CE این تابع باید به صورت زیر تغییر کند تا کار کند:

```
datepart(year, ?1)
```

و ... این Override انجام نشده (تا نگارش فعلی آن). مهم نیست! خودمان انجام خواهیم داد! به صورت زیر:

```
using NHibernate;
using NHibernate.Dialect;
using NHibernate.Dialect.Function;

namespace Test1
{
    public class CustomSqlCeDialect : MsSqlCeDialect
    {
        public CustomSqlCeDialect()
        {
            RegisterFunction("year", new SQLFunctionTemplate(NHibernateUtil.Int32, "datepart(year,
?1)"));
        }
    }
}
```

خوب تا اینجا ما یک Dialect جدید را با ارث بری از MsSqlCeDialect اصلی تولید کرده‌ایم. مرحله بعد نوبت به معرفی آن به NHibernate است. اینکار توسط Fluent NHibernate به سادگی زیر است:

```
var dbType = MsSqlCeConfiguration.Standard
...
```

```
.Dialect<CustomSqlCeDialect>());
```

پس از آن کوئری LINQ ابتدای بحث بدون مشکل اجرا خواهد شد چون اکنون می‌داند که بجای `extract year` ، باید از تابع `datepart` استفاده کند.

مرحله بعد هم می‌تواند تهیه یک `patch` و ارسال به گروه اصلی برای به روز رسانی پروژه NH باشد.