عنوان: **نحوه‌ی نگاشت فیلدهای فرمول در Fluent NHibernate**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۱/۲۱ ۱۳:۳۰:۰۰
آدرس: www.dotnettips.info
گروه‌ها: NHibernate

اگر با SQL Server کار کرده باشید حتما با مفهوم و امکان Computed columns (فیلدهای محاسبه شده) آن آشنایی دارید. چقدر خوب می‌شد اگر این امکان برای سایر بانک‌های اطلاعاتی که از تعریف فیلدهای محاسبه شده پشتیبانی نمی‌کنند، نیز مهیا می‌شد. زیرا یکی از اهداف مهم استفاده‌ی صحیح از ORMs ، مستقل شدن برنامه از نوع بانک اطلاعاتی است. برای مثال امروز می‌خواهیم با MySQL کار کنیم، ماه بعد شاید بخواهیم یک نسخه‌ی سبک‌تر مخصوص کار با SQLite را ارائه دهیم. آیا باید قسمت دسترسی به داده برنامه را از نو بازنویسی کرد؟ اینکار در NHibernate فقط با تغییر نحوه‌ی اتصال به بانک اطلاعاتی میسر است و نه بازنویسی کل برنامه (و صد البته شرط مهم و اصلی آن هم این است که از امکانات ذاتی خود NHibernate استفاده کرده باشید. برای مثال وسوسه‌ی استفاده از رویه‌های ذخیره شده را فراموش کرده و به عبارتی ORM مورد استفاده را به امکانات ویژه‌ی یک بانک اطلاعاتی گره نزده باشید).
خوشبختانه در NHibernate امکان تعریف فیلدهای محاسباتی با کمک تعریف نگاشت خواص به صورت فرمول مهیا است. برای توضیحات بیشتر لطفا به مثال ذیل دقت بفرمائید:
در ابتدا کلاس کاربر تعریف می‌شود:

```
using System;
using NHibernate.Validator.Constraints;

namespace FormulaTests.Domain
{
   public class User
   {
       public virtual int Id { get; set; }

       [NotNull]
       public virtual DateTime JoinDate { set; get; }

       [NotNullNotEmpty]
       [Length(450)]
       public virtual string FirstName { get; set; }

       [NotNullNotEmpty]
       [Length(450)]
       public virtual string LastName { get; set; }

       [Length(900)]
       public virtual string FullName { get; private set; } //از طریق تعریف فرمول مقدار دهی می‌گردد

       public virtual int DayOfWeek { get; private set; }//از طریق تعریف فرمول مقدار دهی می‌گردد
   }
}
```

در این کلاس دو خاصیت FullName و DayOfWeek به صورت فقط خواندنی به کمک private set ذکر شده، تعریف گردیده‌اند.
قصد داریم روی این دو خاصیت فرمول تعریف کنیم:

```
using FluentNHibernate.Automapping;
using FluentNHibernate.Automapping.Alterations;

namespace FormulaTests.Domain
{
   public class UserCustomMappings : IAutoMappingOverride<User>
   {
       public void Override(AutoMapping<User> mapping)
       {
           mapping.Id(u => u.Id).GeneratedBy.Identity(); //ضروری است
           mapping.Map(x => x.DayOfWeek).Formula("DATEPART(dw, JoinDate) - 1");
           mapping.Map(x => x.FullName).Formula("FirstName + ' ' + LastName");
       }
   }
}
```

نحوه‌ی انتساب فرمول‌های مبتنی بر SQL را در نگاشت فوق ملاحظه می‌نمائید. برای مثال FullName از جمع دو فیلد نام و نام خانوادگی حاصل خواهد شد و DayOfWeek از طریق فرمول SQL دیگری که ملاحظه می‌نمائید (یا هر فرمول SQL دلخواه دیگری که صلاح می‌دانید).

اکنون اگر Fluent NHibernate را وادار به تولید اسکریپت متناظر با این دو کلاس کنیم حاصل به صورت زیر خواهد بود:

```
create table Users (
       UserId INT IDENTITY NOT NULL,
      JoinDate DATETIME not null,
      FirstName NVARCHAR(450) not null,
      LastName NVARCHAR(450) not null,
      primary key (UserId)
   )
```

همانطور که ملاحظه می‌کنید در اینجا خبری از دو فیلد محاسباتی تعریف شده نیست. این فیلدها در تعاریف نگاشت‌ها به صورت خودکار ظاهر می‌شوند:

```
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
 default-access="property" auto-import="true" default-cascade="none" default-lazy="true">
 <class xmlns="urn:nhibernate-mapping-2.2" mutable="true"
 name="FormulaTests.Domain.User, FormulaTests, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null"
table="Users">
   <id name="Id" type="System.Int32" unsaved-value="0">
     <column name="UserId" />
     <generator class="identity" />
   </id>
   <property name="DayOfWeek" formula="DATEPART(dw, JoinDate) - 1" type="System.Int32" />
   <property name="FullName" formula="FirstName + ' ' + LastName" type="System.String" />
   <property name="JoinDate" type="System.DateTime">
     <column name="JoinDate" />
   </property>
   <property name="FirstName" type="System.String">
     <column name="FirstName" />
   </property>
   <property name="LastName" type="System.String">
     <column name="LastName" />
   </property>
 </class>
</hibernate-mapping>
```

اکنون اگر کوئری زیر را در برنامه اجرا نمائیم:

```
var list = session.Query<User>.ToList();
foreach (var item in list)
{
   Console.WriteLine("{0}:{1}", item.FullName, item.DayOfWeek);
}
```

به صورت خودکار به SQL ذیل ترجمه خواهد شد و اکنون نحوه‌ی بکارگیری فیلدهای فرمول، بهتر مشخص می‌گردد:

```
select
       user0_.UserId as UserId0_,
       user0_.JoinDate as JoinDate0_,
       user0_.FirstName as FirstName0_,
       user0_.LastName as LastName0_,
       DATEPART(user0_.dw, user0_.JoinDate) - 1 as formula0_, --- همان فرمول تعریف شده است
       user0_.FirstName + ' ' + user0_.LastName as formula1_ ---از طریق فرمول تعریف شده حاصل گردیده
است
   from
       Users user0_
```

نظرات خوانندگان

نویسنده: Anonymous
تاریخ: ۱۳۸۹/۱۲/۰۵ ۲۰:۲۵:۵۹

سلام آقای نصیری.
فرض کنید کلاسی مانند زیر وجود دارد:

```
public class Project
}
{ ;public virtual int Id { get; set
{ ;public virtual long ProjectCode { get; set
{ ;public virtual string ProjectName { get; set
{ ;public virtual int CreateDate { get; set
{
```

در فیلد CreateDate مقادیر زیر وجود دارد:

CreateDate
890102
891210
و ...

که تاریخ شروع پروژه ها می باشد. سوال من اینجاست که در NH کجا باید این تاریخ ها رو به 02/01/89 و 10/12/89 تبدیل کنم و در UI به کاربر نشون بدم.
با تشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۲/۰۵ ۲۰:۳۵:۰۰

سلام
دقیقا مانند مثال فوق عمل کنید. یک خاصیت private set دار را همانند مثال فوق اضافه کنید، مثلا PersianDate از نوع string .
سپس فرمولی را باید به آن در قسمت CustomMappings ذکر شده انتساب داد. برای اینکار از همان روش‌های مرسوم cast استفاده کنید به همراه substring تا بشود ابتدا مقدار عددی را به رشته تبدیل کرد و سپس با substring قسمت‌های مختلف را جدا کرد و نهایتا به هم چسباند. فقط باید دقت داشت که این فرمول باید یک فرمول معتبر SQL ایی باشد.

نویسنده: Anonymous
تاریخ: ۱۳۸۹/۱۲/۰۵ ۲۳:۳۶:۱۱

ممنون کاملا متوجه شدم.
حالا اگر بخواهیم از توابع غیر SQL استفاده کنیم باید چکار کنیم؟ برای مثال بخواهیم همین مثال بالا رو با توابع نوشته شده توسط خودمون انجام بدیم.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۲/۰۶ ۰۱:۵۱:۲۶

ببینید، توابع ویژه نمایشی سی شارپ شما، یعنی سمت کلاینت. موضوع بحث فوق سمت سرور بانک اطلاعاتی است. مقادیر در سمت سرور مطابق فرمول شما تشکیل می‌شوند. به آخرین کوئری ذکر شده در مطلب فوق دقت کنید. در حال حاضر فقط SQL Server است که امکان استفاده از توابع دات نتی را هم سمت سرور میسر کرده (از نگارش 2005 به بعد). بنابراین اگر می‌خواهید توابع ویژه‌ای را در همان سمت سرور اعمال کنید که منطق آن مثلا با سی شارپ پیاده سازی شده، باید یک CLR function مخصوص اس کیوال سرور درست کنید. بعد فرمول نگاشت فوق را بر اساس این CLR function تعیین کنید و کار می‌کند. چیزی

شبیه به همان آخرین کوئری تشکیل شده را خواهید داشت. خلاصه اینکه به نحوی باید این پیاده سازی دات نتی خودتون رو به سمت سرور ببرید.
اما سمت کلاینت شما هر کاری را می‌توانید انجام دهید. برای مثال زمان نمایش اطلاعات در WPF یا سیلورلایت از یک Converter استاندارد آن (با پیاده سازی اینترفیس IValueConverter) در حین Binding استفاده کنید. اگر با ASP.NET Webforms کار می‌کنید حین نمایش اطلاعاتی که هم اکنون در سمت کلاینت مهیا است ، مثلا جهت نمایش در یک GridView یا موارد مشابه شما خواهید داشت "myFunc(Eval("field")) و شبیه به این که myFunc باید در کدبیهایند شما پیاده سازی شود. در سایر فناوری‌ها که می‌تواند شامل موارد قبل هم باشند، نهایتا شما یک لیست دریافتی از سرور را دارید، یک حلقه با LINQ یا حالت معمولی تشکیل شده و مقادیر مدل مورد نظر ویرایش می‌شوند تا جهت نمایش مناسب شوند.
تمام این‌ها در حالتی است که قصد شما فقط و فقط تغییر نحوه‌ی نمایش است. به عبارتی الان کل دیتای فیلتر شده سمت کاربر مهیا است. شما می‌خواهید به آن شکل دهید.

حالت دیگر (حالت غیر نمایشی و استفاده در کوئری‌ها):
اگر با LINQ کمی بیشتر از اطلاعات موجود در وب کار کرده باشید احتمالا به این سوال رسیده‌اید که آیا می‌شود متد سفارشی خودمان را هم حین تهیه کوئری‌هایی از این دست استفاده کنیم؟ چون فقط یک سری extension method مشخص بیشتر وجود ندارند. اگر من extension method سفارشی خودم را تهیه کردم چطور؟
این سوال دو پاسخ دارد:
- متدهای سفارشی شما حتما روی کل اطلاعات دریافتی از سرور کار می‌کنند؛ اما بهینه نیستند. چون برای مثال myFunc سی شارپ من معادل SQL ایی ندارد که بتوانم مستقیما آن‌را سمت سرور اجرا کنم. چون نهایتا LINQ to NHibernate باید به SQL یا T-SQL ترجمه شود. به همین جهت مجبورم کل اطلاعات را دریافت کنم، مثلا 100 هزار رکورد، حالا که اشیاء دات نتی من تشکیل و کامل شده، متد سفارشی LINQ خودم را بر روی این‌ها اجرا می‌کنم. این روش کار می‌کنه ولی از لحاظ کارآیی فاجعه است.
- روش دیگر: در NH 3.0 این امکان وجود دارد ... بسط پروایدر LINQ آن با صور مختلف. که اگر وقت شد یک مطلب کامل در مورد آن خواهم نوشت.

نویسنده: Anonymous
تاریخ: ۱۳۸۹/۱۲/۰۶ ۱۵:۱۵:۲۲

از پاسخگویی شما بسیار ممنونم. من هر روز از شما مطلب جدیدی یاد میگیرم.
من قصد کشدار کردن بحث رو ندارم و اینم آخرین ارسال من در مورد این بحث است.فکر می کنم نتونستم منظورم رو واضح برسونم. فرض کنیم کلاس زیر وجود داره:

```
public class Project
}
{ ;public virtual int Id { get; set
{ ;public virtual long ProjectCode { get; set
{ ;public virtual string Name { get; set
{ ;public virtual int CreateDate { get; set

public virtual string SepratedDate
}
{ ;(get { return myFunc(CreateDate
{ ; } private set
{
{
```

من میخواهم در متد زیر لیستی از کلاس بالا رو به DataSet تبدیل کنم:

```
(public DataSet dsGetAll(bool includeArchived
}
```

```
(()using (var repository = new Repository
}
var projects = repository.Find(x => x.IsArchive == includeArchived
;(

;(()var ds = new CollectionToDataSet>(projects.ToList

;()return ds.CreateDataSet
{
{
```

ولی خطا می ده که SepratedDate در جدول وجود نداره!!!

```
{".'Invalid column name 'SepratedDate"}
could not execute query
select project0_.Id as Id15_, project0_.ProjectCode as ProjectC2_15_, project0_.Name as Name15_, ]
project0_.IsArchive as IsArchive15_, project0_.CreateDate as CreateDate15_, project0_.SepratedDate as
Seprated6_15_ from tblProject project0_ where case when project0_.IsArchive=1 then 'true' else 'false'
end=case when @p0='true' then 'true' else 'false' end
```

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۲/۰۶ ۱۷:۲۱:۱۳

- راه یک: مطالب مقاله فوق. یک قسمت آن custom mapping است که میگه لطفا این فیلد رو در کوئری با فرمول تشکیل بده نه با همین فیلدی که من اینجا اضافه کردم. این رو ندید بگیر، بجاش در SQL نهایی یک فرمول بذار، نه صاف همین فیلد رو تا من به خطا برنخورم.
- راه دو: مطالب کامنت قبل. (یعنی از زمان داشتن ToList که همه چیز سمت کلاینت است به بعد ... هر کاری دوست داشتید با این اطلاعات انجام دهید)
- راه سه: در همان قسمت custom mappings می‌شود نوشت map.IgnoreProperty الی آخر. به این صورت خاصیت تعریف شده شما در کوئری SQL ظاهر نمی‌شود تا مشکل درست کند. اطلاعات بیشتر: (+)

نویسنده: Anonymous
تاریخ: ۱۳۸۹/۱۲/۰۶ ۱۹:۰۵:۰۷

!!!Thanks. Excellent