

در مورد طراحی Self Referencing Entities پیشتر مطلبی را در این سایت [مطالعه کرده‌اید](#).  
 یک مثال دیگر آن می‌تواند نظرات چند سطحی در یک سایت باشند. نحوه تعریف آن با مطالبی که در قسمت هشتم عنوان شود تفاوتی نمی‌کند؛ اما ... زمانیکه نوبت به نمایش آن فرا می‌رسد، چند نکته اضافی را باید در نظر گرفت. ابتدا مثال کامل زیر را در نظر بگیرید:

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Data.Entity;
using System.Data.Entity.Migrations;
using System.Linq;

namespace EFGeneral
{
    public class BlogComment
    {
        public int Id { set; get; }

        [MaxLength]
        public string Body { set; get; }

        public virtual BlogComment Reply { set; get; }
        public int? ReplyId { get; set; }
        public ICollection<BlogComment> Children { get; set; }
    }

    public class MyContext : DbContext
    {
        public DbSet<BlogComment> BlogComments { get; set; }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            // Self Referencing Entity
            modelBuilder.Entity<BlogComment>()
                .HasOptional(x => x.Reply)
                .WithMany(x => x.Children)
                .HasForeignKey(x => x.ReplyId)
                .WillCascadeOnDelete(false);

            base.OnModelCreating(modelBuilder);
        }
    }

    public class Configuration : DbMigrationsConfiguration<MyContext>
    {
        public Configuration()
        {
            AutomaticMigrationsEnabled = true;
            AutomaticMigrationDataLossAllowed = true;
        }

        protected override void Seed(MyContext context)
        {
            var comment1 = new BlogComment { Body = "نظر من این است که" };
            var comment12 = new BlogComment { Body = "پاسخی به نظر اول", Reply = comment1 };
            var comment121 = new BlogComment { Body = "پاسخی به پاسخ به نظر اول", Reply = comment12 };

            context.BlogComments.Add(comment121);
            base.Seed(context);
        }
    }

    public static class Test
    {
        public static void RunTests()
        {
            Database.SetInitializer(new MigrateDatabaseToLatestVersion<MyContext, Configuration>());

            using (var ctx = new MyContext())
            {
                var list = ctx.BlogComments
            }
        }
    }
}
```

```

        //where ...
        .ToList() // fills the childs list too
        .Where(x => x.Reply == null) // for TreeViewHelper
        .ToList();

        if (list.Any())
        {
            // ...
        }
    }
}

```

در مثال فوق کلاس نظرات به صورت خود ارجاع دهنده (خاصیت Reply به همین کلاس اشاره می‌کند) تعریف شده است تا کاربران بتوانند تا هر چند سطح لازم، به یک نظر خاص، پاسخ دهند. در اینجا یک چنین جدولی با اطلاعاتی که ملاحظه می‌کنید تشکیل خواهند شد:

The screenshot shows the Object Explorer on the left with the database 'testdb2012' expanded. The 'dbo.BlogComments' table is selected, showing its columns: Id (PK, int, not null), Body (nvarchar(max), null), and ReplyId (FK, int, null). The 'Keys' section shows the primary key 'PK\_BlogComments' and the foreign key 'FK\_BlogComments\_BlogComments\_ReplyId'. The 'SQLQuery2.sql - ...istrator (56)\*' window on the right contains the following query:

```

1 SELECT TOP 1000 [Id]
2   , [Body]
3   , [ReplyId]
4 FROM [testdb2012] . [dbo] . [BlogComments]

```

The 'Results' tab at the bottom shows the following data:

	Id	Body	ReplyId
1	1	نظر من این است که	NULL
2	2	پاسخی به نظر اول	1
3	3	پاسخی به پاسخ به نظر اول	2

یک نظر ارائه شده و سپس دو نظر تو در توی دیگر برای این نظر ثبت شده است.

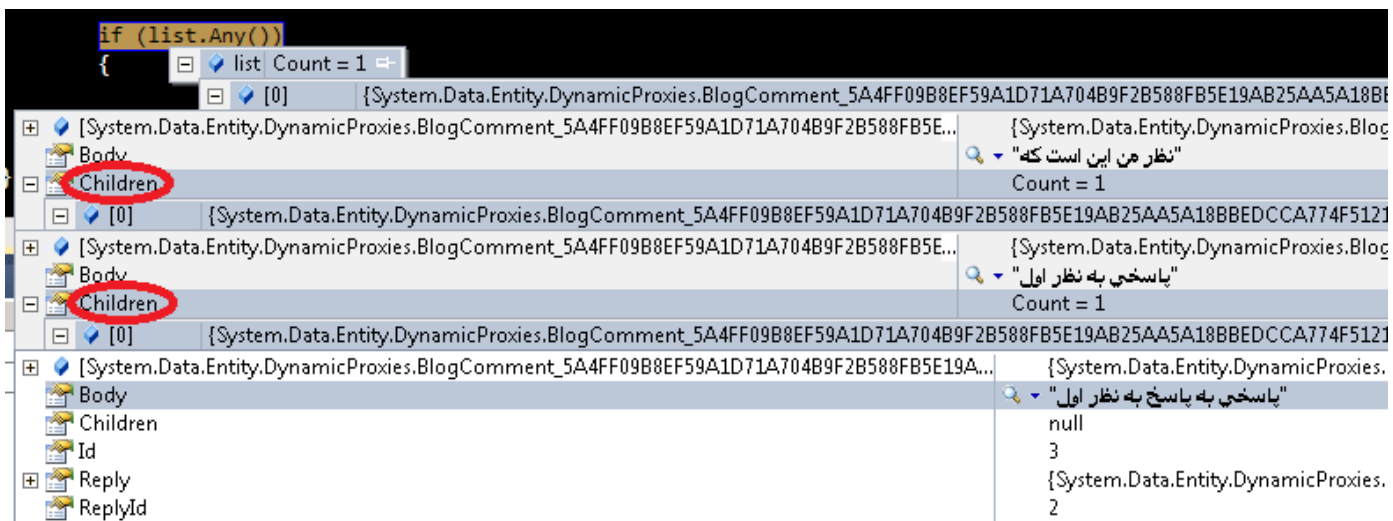
اولین نکته اضافه‌تری که نسبت به قسمت هشتم قابل ملاحظه است، تعریف خاصیت جدید Children به نحو زیر می‌باشد:

```

public class BlogComment
{
    // ...
    public ICollection<BlogComment> Children { get; set; }
}

```

این خاصیت تأثیری در نحوه تشکیل جدول ندارد. علت تعریف آن به توانمندی EF در پرکردن خودکار آن بر می‌گردد. اگر به کوئری نوشته شده در متد RunTests دقت کنید، ابتدا یک ToList نوشته شده است. این مورد سبب می‌شود که تمام رکوردهای مرتبط دریافت شوند. مثلاً در اینجا سه رکورد دریافت می‌شود. سپس برای اینکه حالت درختی آن حفظ شود، در مرحله بعد ریشه‌ها فیلتر می‌شوند (مواردی که reply آن‌ها null است). سپس این مورد تبدیل به list خواهد شد. اینبار اگر خروجی را بررسی کنیم، به ظاهر فقط یک رکورد است اما ... به نحو زیبایی توسط EF به شکل یک ساختار درختی، بدون نیاز به کدنویسی خاصی، منظم شده است:



### سؤال:

برای نمایش این اطلاعات درختی و تو در تو در یک برنامه وب چکار باید کرد؟  
 تا اینجا که توانستیم اطلاعات را به نحو صحیحی توسط EF مرتب کنیم، برای نمایش آن‌ها در یک برنامه ASP.NET MVC می‌توان از یک [TreeViewHelper](#) سورس باز استفاده کرد.  
 البته کد آن در اصل برای استفاده از EF Code first طراحی نشده و نیاز به اندکی تغییر به نحو زیر دارد تا با EF هماهنگ شود  
 (متد `Count` و `ToList` موجود در سورس اصلی آن باید به نحو زیر حذف و اصلاح شوند):

```
private void AppendChildren(TagBuilder parentTag, T parentItem, Func<T, IEnumerable<T>>
childrenProperty)
{
    var children = childrenProperty(parentItem);
    if (children == null || !children.Any())
    {
        return;
    }
    //...
```

## نظرات خوانندگان

نویسنده: مهدی پایروند  
تاریخ: ۹:۲۷ ۱۳۹۱/۰۵/۲۶

آیا میشه این خاصیت رو اینطور پیاده سازی کرد که:

```
public class Person
{
    // other properties

    [Required]
    public virtual Person RelatedPerson {get; set;}
}
```

حالا برای شروع در متد Seed یا معرفی اولین شخص با توجه به این که این RelatedPerson نمیتواند خالی باشد چطور میتوان خود شخص را به این پراپرتی معرفی کرد و بعنوان روت اشخاص از آن استفاده کرد و مپ آن به چه شکل است.

نویسنده: وحید نصیری  
تاریخ: ۹:۳۱ ۱۳۹۱/۰۵/۲۶

نمی‌شود. ویژگی Required را حذف کنید تا فیلد nullable شود. در مورد مباحث اولیه نگاشت آن [به این مطلب](#) مراجعه کنید.

نویسنده: مهدی پایروند  
تاریخ: ۹:۳۵ ۱۳۹۱/۰۵/۲۶

در مورد NHibernate هم این محدودیت وجود دارد؟

نویسنده: وحید نصیری  
تاریخ: ۹:۳۸ ۱۳۹۱/۰۵/۲۶

این محدودیت نیست. کار یک ORM نگاشت اطلاعات کلاس‌های شما به جداول بانک اطلاعاتی است. زمانیکه سمت بانک اطلاعاتی این فیلد باید null پذیر باشد چون ریشه یک درخت تشکیل شده والدی ندارد، سمت کدهای شما هم به همین ترتیب باید تعریف شود تا نگاشت به نحو صحیحی صورت گیرد.

نویسنده: مهدی پایروند  
تاریخ: ۹:۵۴ ۱۳۹۱/۰۵/۲۶

در مورد تعریف ORM حق با شماست ولیا اینکه میشه این رکورد رو تو بانک ایجاد کرد، پس شاید راهی برای اینکار در ORM هم باشه، من کلاس رو این شکلی تعریف کردم و مطمئن هستم حین این ذخیره سازی که این رکورد، رکورد اول جدول هم هست مثلا شناسه اون حتما با یک ذخیره میشه.

نویسنده: وحید نصیری  
تاریخ: ۹:۵۷ ۱۳۹۱/۰۵/۲۶

اولین رکورد یا به عبارتی ریشه یک درخت، ریشه‌ای ندارد. این ریشه نال رو چطور در بانک اطلاعاتی تعریف و ذخیره می‌کنید؟ بحث ما Id اولین رکورد نیست. بحث کلید خارجی است که باید نال پذیر باشد و به همین جدول هم اشاره می‌کند و نمایانگر رکورد والد یک رکورد خاص است (جدول خود ارجاع دهنده). در همین مطلب جاری به تصویر اول دقت کنید. بحث ما فیلد ReplyId است که نال پذیر است. این ReplyId کلید خارجی اشاره‌کننده به همین جدول هم هست.

نویسنده: مهدی پایروند

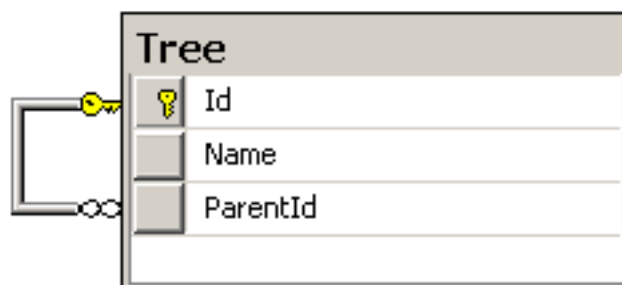
موضوعی که من می‌خواهم براس تو ORM راهی پیدا کنم مربوط به اینه که بتونم کلاسم رو تغییر ندیم و از بتونم فیلد پرنتم رو غیر نال تعریف کنم، این اسکریپت رو ببینید:

```
CREATE TABLE [dbo].[Tree](
  [Id] [int] IDENTITY(1,1) NOT NULL,
  [Name] [nvarchar](10) NOT NULL,
  [ParentId] [int] NOT NULL,
  CONSTRAINT [PK_Tree] PRIMARY KEY CLUSTERED
(
  [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Tree] WITH CHECK ADD CONSTRAINT [FK_Tree_Tree] FOREIGN KEY([ParentId])
REFERENCES [dbo].[Tree] ([Id])
GO

ALTER TABLE [dbo].[Tree] CHECK CONSTRAINT [FK_Tree_Tree]
GO
```



حالا برای ایجاد رکورد اولی می‌دونید که شناسه اون یک هست و برای همین میشه رکورد فیلد [ParentId] رو پر کرد من این امتحان رو انجام دادم و مشکلی پیش نیومد

	Id	Name	ParentId
1	1	Root	1

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۰:۹

فیلد parent-id رو دستی مقدار دهی کردی؟ یا با EF مقدار دهی شد؟

نویسنده: مهدی پایروند  
تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۰:۱۵

تو این مورد با بانک ولی می‌گم حتما باید برای سمت ORM هم راهی باشه

نویسنده: وحید نصیری  
تاریخ: ۱۰:۱۷ ۱۳۹۱/۰۵/۲۶

parent-id کلید خارجی است. برای مقدار دهی آن (ثبت اولیه رکورد مرتبط) اگر null پذیر نباشد نیاز است حتما رکورد اشاره کننده به آن وجود خارجی داشته باشد. به همین دلیل باید در این حالت خاص آن را نال پذیر تعریف کرد؛ چون رکورد ریشه، والدی ندارد و کلید خارجی آن نال خواهد بود. همچنین در این حالت خاص مورد بحث ما، کلید خارجی به خود جدول جاری اشاره می‌کند (و اگر نال پذیر نباشد کل رکورد ریشه، در بار اول ثبت آن، قابل ذخیره سازی نیست).

نویسنده: وحید نصیری  
تاریخ: ۱۰:۳۷ ۱۳۹۱/۰۵/۲۶

برای درک بهتر این موضوع، سعی کنید دستور زیر را اجرا کنید (از management studio استفاده نکنید):

```
INSERT INTO [Tree]
  ([Name]
  ,[ParentId])
VALUES
  ('12'
  ,2)
```

قابل ثبت نیست. ضمناً امکان مقدار دهی دستی ParentId هم در اینجا تا زمانیکه رکورد ثبت نشده باشد، میسر نیست (کاری که management studio به صورت دستی انجام داده، چند مرحله کار بوده نه صرفاً یک insert معمولی).

نویسنده: مهدی پایروند  
تاریخ: ۱۰:۴۵ ۱۳۹۱/۰۵/۲۶

بله ممنون، همین موردی که می‌گید چند مرحله رو باید بشه با ORM بدست آورد، با SQL Profiler میشه از کارکرد SQL Managment نمونه برداری کرد، البته شاید این کار خروجی مناسبی برای سمت کد نداشته باشه.

نویسنده: وحید نصیری  
تاریخ: ۱۰:۵۰ ۱۳۹۱/۰۵/۲۶

طراحی مدنظر شما اصلاً متداول نیست. برای مطالعه بیشتر مراجعه کنید به این مقاله:

[SQL Server CTE Basics](#)

در اینجا هم NULL int ManagerID تعریف شده.

نویسنده: Mina  
تاریخ: ۲۳:۴۳ ۱۳۹۱/۰۹/۱۷

با سلام؛

اگر ما در مدل یک فیلد به عنوان مثال status داشته باشیم و بخواهیم آنهایی که status شان ok است را برگردانیم

```
var model = _efComment.List(p => p.PostId == postId);
var list = model.ToList()
.Where(p => p.ComentStatus == ComentStatus.Ok)
.Where(x => x.Reply == null)
.ToList();
```

کد بالا جواب نمیده چون اگه پدرش status با وضعیت ok داشته باشه فرزندانش با هر وضعیتی باش میاره راه حلی به نظرتون میاد؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۹/۱۸ ۰:۱۴

همان ابتدای کار از && برای فیلتر کردن استفاده کنید

```
p => p.PostId == postId && p.ComentStatus == ComentStatus.Ok
```

نویسنده: میلاد محسنی  
تاریخ: ۱۳۹۱/۱۱/۱۹ ۲۲:۲۹

با سلام.

برای تهیه مدل‌های خود ارجاع دهنده، نظراتان در خصوص استفاده از hierarchyID چیست؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۱۱/۱۹ ۲۳:۰۶

[قابل انتقال نیست](#) به سایر بانک‌های اطلاعاتی. یکی از اهداف کار با ORM‌ها مستقل کردن برنامه از بانک اطلاعاتی است. مثلاً بتوانید با SQL CE هم کار کنید که این نوع داده رو پشتیبانی نمی‌کنه و خیلی از بانک‌های اطلاعاتی دیگر نیز به همین ترتیب.

نویسنده: مهدی سعیدی فر  
تاریخ: ۱۳۹۱/۱۲/۰۱ ۱۹:۴۴

با سلام

بنده مدل زیر را دارم که مربوط به صفحاتی هستند که والد هم دارند.

```
public class Page
{
    public virtual int Id { get; set; }
    public virtual string Title { get; set; }
    public virtual DateTime? CreatedDate { get; set; }
    public virtual DateTime? ModifiedDate { get; set; }
    public virtual string Body { get; set; }
    public virtual string Keyword { get; set; }
    public virtual string Description { get; set; }
    public virtual string Status { get; set; }
    public virtual bool? CommentStatus { get; set; }
    public virtual int? Order { get; set; }
    public virtual User User { get; set; }
    public virtual User EditedByUser { get; set; }
    public virtual ICollection<Comment> Comments { get; set; }
    public virtual int? ParentId { get; set; }
    public virtual Page Parent { get; set; }
    public virtual ICollection<Page> Children { get; set; }
}
```

و با دستور زیر می‌خواهم از آن کوئری بگیرم:

```
this._pages.ToList().Where(page => page.Parent == null).ToList();
```

دستور فوق به خوبی کار می‌کنه. ولی وقتی با که دستوراتی که توسط mini-profiler لاگ شده را می‌بینیم که اخطار duplicate reader را می‌دهد.

برای هر page موجود دستور زیر را صادر می‌کند

```
SELECT
[Extent1].[Id] AS [Id],
[Extent1].[Title] AS [Title],
[Extent1].[CreatedDate] AS [CreatedDate],
[Extent1].[ModifiedDate] AS [ModifiedDate],
[Extent1].[Body] AS [Body],
[Extent1].[Keyword] AS [Keyword],
[Extent1].[Description] AS [Description],
[Extent1].[Status] AS [Status],
[Extent1].[CommentStatus] AS [CommentStatus],
[Extent1].[Order] AS [Order],
[Extent1].[ParentId] AS [ParentId],
[Extent2].[Id] AS [Id1],
[Extent2].[Title] AS [Title1],
[Extent2].[CreatedDate] AS [CreatedDate1],
[Extent2].[ModifiedDate] AS [ModifiedDate1],
[Extent2].[Body] AS [Body1],
[Extent2].[Keyword] AS [Keyword1],
[Extent2].[Description] AS [Description1],
[Extent2].[Status] AS [Status1],
[Extent2].[CommentStatus] AS [CommentStatus1],
[Extent2].[Order] AS [Order1],
[Extent2].[ParentId] AS [ParentId1],
[Extent2].[User_Id] AS [User_Id],
[Extent2].[EditedByUser_Id] AS [EditedByUser_Id],
[Extent1].[User_Id] AS [User_Id1],
[Extent1].[EditedByUser_Id] AS [EditedByUser_Id1]
FROM [dbo].[Pages] AS [Extent1]
LEFT OUTER JOIN [dbo].[Pages] AS [Extent2] ON [Extent1].[ParentId] = [Extent2].[Id]
```

می‌خواستم ببینم کاری میشه کرد تا سر بار این کوئری را کمتر کرد؟

در ضمن اگر بخواهم viewmodel را طوری تعریف کنم تا فیلدهای اضافی مانند createddate و user و ... که در هنگام نمایش منوی آبخاری به آنها نیازی ندارم چه کار باید کرد؟ چون من هر کاری کردم نتونستم parent را برای viewmodel به خوبی تعریف کنم.

ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۲/۰۱ ۲۱:۲۹

- این خروجی SQL لاگ شده مطلب جاری (با تمام توضیحات و نگاشت‌های آن) توسط برنامه مطمئن SQL Server Profiler است:

```
SELECT
[Extent1].[Id] AS [Id],
[Extent1].[Body] AS [Body],
[Extent1].[ReplyId] AS [ReplyId]
FROM [dbo].[BlogComments] AS [Extent1]
```

منطقی هم هست. چون در ToList اول، کار با دیتابیس تمام و قطع می‌شود. ToList دوم سمت کلاینت اجرا می‌شود. یعنی تشکیل درخت نهایی توسط امکانات LINQ to Objects انجام می‌شود و نه هیچ کار اضافه‌ای در سمت سرور.

- اگر اینجا join اضافی پیدا کردید ... حتما مشکلی در تنظیمات نگاشت‌ها دارید.

- اگر duplicate reader دارید شاید بخاطر [lazy loading](#) سایر خواص راهبری است که تعریف کردید مانند User و EditByUser و غیره. این‌ها اگر قرار است نمایش داده شوند، پیش از ToList اول باید توسط متد الحاقی Include به صورت eager loading تعریف شوند تا lazy loading و duplicate reader نداشته باشید.

- برای فیلتر فیلدهای اضافی، پیش از ToList اول، با استفاده از Projection و نوشتن یک Select، موارد مورد نیاز را انتخاب کنید.

نویسنده: مهدی سعیدی فر

تاریخ: ۱۳۹۱/۱۲/۰۱ ۲۲:۳۰



با راهنمایی شما مشکل Duplicate Reader با نوشتن دستور به شکل زیر حل شد.

```
this._pages.Include(page => page.Children).ToList().Where(page => page.Parent == null).ToList();
```

الان مشکلی دارم اینه که نمیتونم به select خوب بنویسم تا فقط مواردی را که میخوام برگردونم. مثلا من viewmodel را این شکلی تعریف کردم.

```
public class NavBarModel
{
    public virtual int Id { get; set; }
    public virtual string Title { get; set; }
    public virtual string Status { get; set; }
    public virtual int? Order { get; set; }
    public virtual NavBarModel Parent { get; set; }
    public virtual ICollection<Page> Children { get; set; }
}
```

توی select زدن نمیدونم چه شکلی باید کد بزنم تا parent و children هم به صورت خودکار پر شوند.

در حقیقت من می‌خواهم این موارد را در یک navigation bar به صورت منوی آبدشاری نشون بدم.

در ضمن شما می‌تونید در نشان دادن اطلاعات فوق به صورت منوی آبدشاری با امکان تعریف فرزند تو در تو بدون محدودیت راهنماییم کنید. من کدی واسش نوشتم ولی متاسفانه برای پیاده سازی نامحدود بودن فرزندان منو مشکل دارم.

ممنون

نویسنده: مهدی سعیدی فر  
تاریخ: ۱۱:۴۰ ۱۳۹۱/۱۲/۰۴

سلام

راستش من منوهای چند سطحی پویا را برای bootstrap navbar نوشتم. الگوریتمش را مجبور شدم به صورت بازگشتی بنویسم. اگر کسی می‌تونه به صورت غیر بازگشتی بگه ممنونش میشم. این کد به partialpage برای navbar هست که هرکسی برای bootstrap به راحتی میتونه استفاده کنه.

```
@model IEnumerable<DomainClasses.Page>
@helper ShowNavBar(IEnumerable<DomainClasses.Page> pages)
{
    foreach (var page in pages)
    {
        if (page != null)
        {
            if (page.Children.Count == 0)
            {
                <text><li><a tabindex="-1" href="#">@page.Title</a></li></text>
            }
            if (page.Children.Count > 0 && page.Parent == null)
            {
```



بنابراین در حالت فعلی تمام فیلدهای وابسته از بانک اطلاعاتی استخراج نمی‌شوند مگر اینکه lazy loading را به نحوی تبدیل به eager loading کرده باشید.

- ضمناً در حالت فعلی اگر دقت کرده باشید پیش از ToList اول یک سه نقطه گذاشته شده است. یعنی اینجا می‌تونید where بنویسید. می‌تونید Select بنویسید و به صورت اختصاصی یک سری خاصیت مشخص رو انتخاب کنید و خیلی از کارهای دیگر.

نویسنده: ali.rezayee  
تاریخ: ۱۵:۱۹ ۱۳۹۲/۰۱/۱۴

با تشکر فراوان از پاسخ کامل جنابعالی.

مشکل من از اینجا شروع شد که با Json.Net خواستم نتیجه این کوئری را به Json تبدیل کنم، من دستور زیر را نوشتم، پس از اولین اجرا رکورد اول دو فیلد Body و Id را دارد که کاملاً درست است، اما رکورد بعدی که فرزند رکورد اول است شامل تمام فیلدهای جدول BlogComment و تمام جداول مرتبط با آن است.

ممنون از شما به خاطر صرف وقت گرانبها.

```
var list = context.BlogComment.Where(p => p.UserId == 100)
    .Select(p => p.Body, p.Id, p.Children)
    .ToList();
```

نویسنده: وحید نصیری  
تاریخ: ۱۶:۱۰ ۱۳۹۲/۰۱/۱۴

از ویژگی Newtonsoft.Json.JsonIgnore برای عدم serialization یک خاصیت خاص به JSON استفاده کنید.

نویسنده: ali.rezayee  
تاریخ: ۲۱:۴۰ ۱۳۹۲/۰۱/۱۴

با تشکر فراوان، مشکل من با Newtonsoft.Json.JsonIgnore حل شد.

اما من برای انتخاب تعدادی خصوصیت از کد زیر استفاده کردم، در نتیجه نهایی که در عکس مشخص است، فقط رکورد اول شامل فیلدهای مشخص شده توسط من است، فرزندان این رکورد حاوی تمام فیلدها هستند. البته در این کدها از using استفاده نشده چون باعث خطای The ObjectContext instance میشود. باز هم ممنون برای اختصاص وقت.

```
public partial class BlogComment
{
    public BlogComment()
    {
        this.Children = new HashSet<BlogComment>();
    }

    public int Id { get; set; }
    public string Body { get; set; }
    public Nullable<System.DateTime> DateSend { get; set; }
    public Nullable<System.DateTime> DateRead { get; set; }
    public bool IsDeleted { get; set; }
    public int UserId { get; set; }

    [Newtonsoft.Json.JsonIgnore]
    public virtual BlogComment Reply { get; set; }

    public int? ReplyId { get; set; }
    public virtual ICollection<BlogComment> Children { get; set; }
}
```

```
public JsonResult Index()
{
    var ctx = new testEntities();
    var list = ctx.BlogComments
        .Where(p => p.Id == 1)
        .Select(p => new
        {
            p.Id,
            p.Body,
            p.Children
        })
        .ToList();

    JsonResult jsonNetResult = new JsonResult();
    jsonNetResult.Formatting = Formatting.Indented;

    jsonNetResult.SerializerSettings = new JsonSerializerSettings()
    {
        ReferenceLoopHandling =
ReferenceLoopHandling.Ignore
    };

    jsonNetResult.Data = list;
    return jsonNetResult;
}
```

```
[
  - {
    Id: 1,
    Body: "نظر 1",
    - Children: [
      - {
        - Children: [
          - {
            Children: [ ],
            Id: 4,
            Body: "نظر 1-2",
            DateSend: "2013-03-03T00:00:00",
            DateRead: null,
            IsDeleted: false,
            UserId: 1,
            ReplyId: 2
          }
        ],
        Id: 2,
        Body: "نظر 1-1",
        DateSend: "2013-03-03T00:00:00",
        DateRead: null,
        IsDeleted: false,
        UserId: 1,
        ReplyId: 1
      }
    ]
  }
]
```

نویسنده: وحید نصیری  
تاریخ: ۲۲:۵۹ ۱۳۹۲/۰۱/۱۴

علت این است که p.Children به تمام خواص عمومی شیء BlogComment اشاره می‌کند؛ این رو هم همیشه یک سطح دیگر با Projection سبک‌تر کرد و یا بجای Projection در حالت شما ساده‌تر است که JsonIgnore را روی تمام خواصی قرار دهید که نباید توسط JSON.NET بررسی شود. با توجه به lazy loading، این خواص virtual توسط EF در بدو امر بارگذاری نمی‌شوند و

همچنین چون توسط JSON.NET به دلیل JsonIgnore معرفی شدن واکاوی مجدد نخواهند شد، بنابراین مشکلی از لحاظ کارایی یا حجم بالای خروجی نخواهید داشت.

نویسنده: ali.rezayee  
تاریخ: ۲۳:۳۲ ۱۳۹۲/۰۱/۱۴

ممنون از لطف شما.  
عالی و جامع بود.

نویسنده: مولانا  
تاریخ: ۲۰:۳۹ ۱۳۹۲/۰۲/۱۲

دوست عزیز.  
از مطلب مفید شما متشکرم. من هم مشکل شما رو دارم. آیا به جواب رسیدید؟

نویسنده: مولانا  
تاریخ: ۲۲:۴۶ ۱۳۹۲/۰۲/۲۴

باسلام.  
برای مدل‌های خود ارجاع دهنده در هنگام حذف یک رکورد، خطای زیر بوجود می‌آید.

The DELETE statement conflicted with the SAME TABLE REFERENCE constraint

چاره کار چیست؟ (می‌خواهم با حذف یک فولدر تمام محتویات آن نیز حذف شوند).

نویسنده: ایمان محمدی  
تاریخ: ۱۱:۳۱ ۱۳۹۲/۰۴/۱۰

سلام  
در مثال بالا چون نظرات با یک پست مرتبط هستند قاعدتا براساس یک پست کامنت‌ها رو محدود می‌کنیم. ولی اگر بخواهیم یک کامنت خاص رو با زیر مجموعه هاش دریافت کنیم شرط جستجو رو چطور باید بنویسیم؟  
راهکار زیر در sql هست که دیتابیس رو به sql server محدود می‌کنه و معادل لینک هم فکر نکنم داشته باشه.

```
DECLARE @Table TABLE(
    ID INT,
    ParentID INT,
    NAME VARCHAR(20)
)

INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 1, NULL, 'A'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 2, 1, 'B-1'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 3, 1, 'B-2'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 4, 2, 'C-1'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 5, 2, 'C-2'

DECLARE @ID INT
SELECT @ID = 2

;WITH ret AS(
    SELECT*
    FROM@Table
    WHEREID = @ID
    UNION ALL
    SELECTt.*
    FROM@Table t INNER JOIN
    ret r ON t.ParentID = r.ID
)

SELECT *
```

FROM ret

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۳:۵۱

نیازی نیست زیاد پیچیده فکر کنید. یک لیست ساده رو واکنشی کنید (مثلا یک کامنت که سه زیر مجموعه مرتبط دارد با یک select ساده)، اتصال نهایی آن‌ها در سمت کلاینت خودکار است. به عبارتی شکل دهی تو در تو در اینجا در سمت کلاینت انجام می‌شود و نه سمت سرور. سمت سرور آن، یک select معمولی از رکوردهای مورد نظر بیشتر نیست.

```
var specifiedCommentId = 4; // دارای یک سری زیر کامنت است
var list = ctx.BlogComments
    .Include(x => x.Reply)
    .Where(x => x.Id >= specifiedCommentId && (x.ReplyId == x.Reply.Id || x.Reply ==
null))
    .ToList() // fills the childs list too
    .Where(x => x.Reply == null) // for TreeViewHelper
    .ToList();
```

نویسنده: ایمان محمدی  
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۶:۱۰

ممون ولی مشکلی با سمت کلاینت ندارم مشکل من شرطی هست که بدون واکنشی کل سطرها ، خود ارجاعی رو تا هر سطحی که هست پیمایش کنه .

فکر کنم سوالم رو اینجوری مطرح کنم بهتر باشه، ما تعدادی کارمند داریم که عضو گروه‌های زیر هستند. گروه بندی خود ارجاع دهنده هست.

```
GroupRoot {id=1, Name="گروه اصلی",ParentId=null}
Group2 {id=2, Name="بازرگانی",ParentId=1}
Group3 {id=3, Name="فروش",ParentId=2}
Group4 {id=4, Name="فروش قطعات",ParentId=3}
Group5 {id=5, Name="خدمات",ParentId=1}
person1 {name="Ali", GroupId=2} // بازرگانی
person2 {name="reza", GroupId=3} // فروش
person3 {name="iman", GroupId=3} // فروش
person4 {name="hamid", GroupId=4} // فروش قطعات
person5 {name="hasan", GroupId=4} // فروش قطعات
person6 {name="Ahmad", GroupId=5} // خدمات
person7 {name="vahid", GroupId=1} // گروه اصلی
```

گزارش 1: نمایش گروه بازرگانی شناسه 2 به همراه زیر مجموعه هاش {بازرگانی،فروش،فروش قطعات}

گزارش 2 : نمایش پرسنل گروه فروش و زیر مجموعه‌های گروه فروش {reza,iman,hamid,hasan}

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۶:۳۳

کوئری رو که در پاسخ سؤال قبلی شما نوشته شد، یکبار اجرا کنید. تمام رکوردها رو واکنشی نمی‌کنه. فقط از یک آی دی خاص شروع می‌کنه و مواردی رو که ReplyId اون‌ها با idها یکی است انتخاب می‌کنه. یعنی فقط یک سری زیر خانواده خاص.

مسایل و پروژه‌های شخصی خودتون رو هم در انجمن‌ها پیگیری کنید.

نویسنده: وحید نصیری

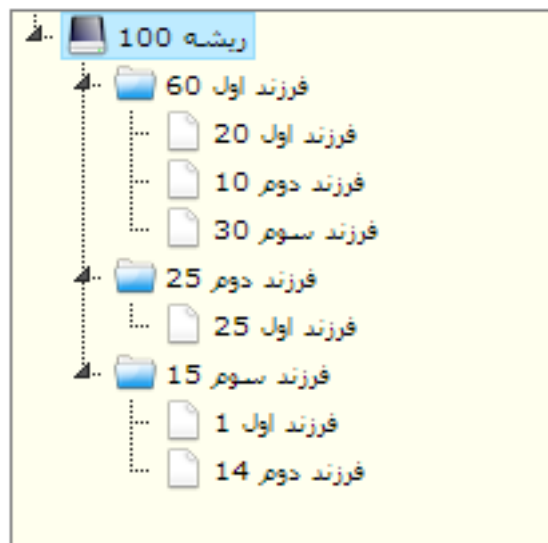
تاریخ: ۲۱:۲۱ ۱۳۹۲/۰۴/۱۹

برای تکمیل بحث، روش مایکروسافت برای حل مساله مدل‌های خود ارجاع دهنده:  
[Recursive or hierarchical queries using EF Code First and Migrations](#)  
 از نوشتن مستقیم SQL و ایجاد View استفاده کرده.

نویسنده: ali.rezayee

تاریخ: ۱۲:۵۹ ۱۳۹۲/۰۴/۲۴

با سلام و عرض تشکر بابت این مطلب کامل



در خصوص مدل‌های خود ارجاع دهنده، اگر بخواهیم مثل عکس بالا هر نود جدیدی که ثبت میشود مقادیر امتیاز تمام والد‌های آن تا ریشه تغییر کند بهترین راه حل چیست؟  
 به بیانی دیگر اگر به شکل دقت کنید هر نود جدیدی که ثبت میشود امتیاز تمامی والد‌های آن تا ریشه به روز میشود، برای این کار راه حلی که به نظرم رسید این بود که با ثبت هر نود جدید، تمامی والد‌های آن واکشی و مقدار امتیاز آن به روز شوند تا به ریشه برسیم؛ راه حل دیگر نوشتن تریگر در سمت دیتابیس است. فکر میکنم در هر دو حالت سر بار زیادی داشته باشیم. آیا راه بهتری وجود دارد؟  
 ممنون

نویسنده: محسن خان

تاریخ: ۱۴:۷ ۱۳۹۲/۰۴/۲۴

کم هزینه‌ترین روش: جمع‌ها رو سمت کلاینت مدیریت کنید و اصلاً اطلاعات جمع آن‌ها رو سمت سرور ثبت نکنید. زمانیکه این TreeView در حال رندر هست، جمع گره‌های والد رو بر اساس فرزندانش محاسبه و نمایش بدید.

نویسنده: ali.rezayee

تاریخ: ۱۴:۲۳ ۱۳۹۲/۰۴/۲۴

بسیار ممنونم از پاسخ شما.  
 روش شما برای زمانی که تمام درخت لود شود و به کلاینت فرستاده شود کاملاً عالی است. اما اگر با Ajax مرحله به مرحله نودها لود شود دیگر این روش پاسخ نمیدهد چون پدر از امتیازات تمامی فرزندانش بی خبر است.  
 باز هم ممنون



نویسنده: محسن خان  
تاریخ: ۱۴:۳۲ ۱۳۹۲/۰۴/۲۴

برای نمایش اول کار: جمع کل ریشه اصلی مساوی است با جمع فرزندان که والد غیر null دارند. یک کوئری سبک بیشتر نیست. مابقی جمع‌های درخت رو میشه سمت کلاینت مدیریت کرد.

نویسنده: امیرحسین جلوداری  
تاریخ: ۱۳:۲۲ ۱۳۹۲/۰۵/۰۷

روش بهینه‌ی حذف پدر و فرزندان اون چیه؟! ... ممنون میشم اگه رو همین مثال لطف کنین

نویسنده: وحید نصیری  
تاریخ: ۱۳:۳۰ ۱۳۹۲/۰۵/۰۷

- نمی‌تونید والد رو یک ضرب حذف کنید. از آخرین فرزند به اول ریشه (والد ریشه جاری) باید (یکی یکی) حذف کنید تا قیود تعریف شده مانع حذف اطلاعات نشوند.  
- یا اینکه ... soft delete انجام بدید. یک فیلد bool تعریف کنید که این رکورد خاص deleted هست یا خیر.

نویسنده: مهدی سعیدی فر  
تاریخ: ۱۳:۳۲ ۱۳۹۲/۰۵/۰۷

من از این استفاده می‌کنم. (مدل کامنت و پاسخ هاش)

```
public void Remove(int id)
{
    var selectedComment = _comments.Find(id);
    _comments.Where(x => x.ParentId == id).Load();
    _comments.Remove(selectedComment);
}
```

نویسنده: امیرحسین جلوداری  
تاریخ: ۱۴:۲۳ ۱۳۹۲/۰۵/۰۷

روش شما رو تست کردم ولی جواب نداد!... فکر میکنم این ساختاری که شما تعریف کردید برای حالت cascadeDelete جواب میده ...

طبق فرمایش آقای نصیری کاری که من کردم اینه (روی سیستم گروه و زیرگروه):  
1 - واکنشی گروه انتخاب شده و فرزندان آن با استفاده از الگوریتم اول سطح:

```
private Stack<Group> GetChildsAndRoot(Group group)
{
    var stack = new Stack<Group>();
    var queue = new Queue<Group>();
    stack.Push(group);
    queue.Enqueue(group);
    while (queue.Any())
    {
        var currGroup = queue.Dequeue();
        foreach (var child in currGroup.Childs)
        {
            queue.Enqueue(child);
            stack.Push(child);
        }
    }
    return stack;
}
```

2- پاک کردن پشته‌ی بازگشتی :

```
var group = _groupRepository.GetByID(id);
var nodes = GetChildsAndRoot(group);
while (nodes.Any())
{
    _groupRepository.Delete(nodes.Pop());
}
_unitOfWork.SaveChanges();
```

نویسنده: مهدی سعیدی فر  
تاریخ: ۱۵:۱۶ ۱۳۹۲/۰۵/۰۷

نه کار می‌ده. اون هم در حالتی که cascade نباشه.  
این کدی که نوشتم دقیقاً مال پروژه‌ی IRIS هست. یک بار کدش را در اون پروژه ببینید؛ ضرری نداره.

نویسنده: ایلینا اکبری فرد  
تاریخ: ۱۱:۱۸ ۱۳۹۳/۰۳/۱۷

با سلام.  
چگونه می‌توان childها را نیز بر اساس فیلد دلخواه مرتب سازی کرد. برای مثال کوئری زیر تنها سطح اول را مرتب می‌کند و فرزندان را مرتب نمی‌کند.

```
var query = _tEntities.AsNoTracking()
    .Where(p => p.Parent_Id == null && p.IsActive == true)
    .OrderBy(sortExpression);
var result = query.ToList();
return result;
```