

Forum:[Hack Forums](#)

Thread: [\[Entity Framework\] Multi level parent relation search.](#)

severos - March 30, 2018, 12:17 pm

I have this entity:

PHP Code

```
<?php
public class Category
{
    public Category()
    {
        Children = new HashSet<Category>();
        Products = new HashSet<Product>();
    }

    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int Id { get; set; }
    public string Name { get; set; }
    public int? ParentId { get; set; }

    [ForeignKey("ParentId")]
    public Category Parent { get; set; }
    [InverseProperty("Parent")]
    public ICollection<Category> Children { get; set; }
    [InverseProperty("Category")]
    public ICollection<Product> Products { get; set; }
}
```

and this:

PHP Code

```
<?php
public class Product
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int Id { get; set; }
    public string Name { get; set; }
    [Required]
    public int CategoryId { get; set; }

    [ForeignKey("CategoryId")]
    public Category Category { get; set; }
}
```

A category can be considered as Master/Primary if its parent is null.

Lets say that I want to search Products using category ID. It's a trivial matter if I'm looking for a lowest child category. All I have to do is use the DbContext with the DbSet<Product> with a simple LINQ query to find Products that meet the category Id.

But how would I go around writing a query that will search in sub categories? like if I search using a Master category id. I know I can use loops for it, but it doesn't seem like a nice solution to be honest, specially if you consider having multiple nesting levels.

Anyone have an elegant looking code for this? I won't consider performance much to be honest since this is a small university project, but getting info on good performance solution would be awesome for future reading.

Ninja - March 30, 2018, 12:29 pm

So you want to search for the last descendant of a Category, or am I misunderstanding it here?

severos - March 30, 2018, 12:58 pm

(Mar 30, 2018 12:29 PM)Ninja Wrote: [\[CITE\]](#) So you want to search for the last descendant of a Category, or am I misunderstanding it here?

Given a category Id, I need to get all products under that category, or its sub categories.

?DiViN3 - March 30, 2018, 5:06 pm

PHP Code

```
<?php
/// <summary>Retrieves the property info from the type with the following attribute type into a list collection.</summary>
/// <typeparam name="T">The object type to search.</typeparam>
/// <param name="attributeType">The attribute type to search for.</param>
/// <returns>The <see cref="PropertyInfo"/>list.</returns>
public static List<PropertyInfo> GetPropertyList<T>(Type attributeType)
{
    var _properties = typeof(T).GetProperties().Where(_property => Attribute.IsDefined(_property, attributeType));
    return _properties.ToList();
}
```

Usage:

PHP Code

```
<?php
var _properties = GetPropertyList<CustomType>(typeof(CustomAttribute));
```

severos - March 30, 2018, 9:32 pm

(Mar 30, 2018 05:06 PM)?DiViN3 Wrote: [\[CITE\]](#) PHP Code

```
<?php
/// <summary>Retrieves the property info from the type with the following attribute type into a list collection.</summary>
/// <typeparam name="T">The object type to search.</typeparam>
/// <param name="attributeType">The attribute type to search for.</param>
/// <returns>The <see cref="PropertyInfo"/>list.</returns>
public static List<PropertyInfo> GetPropertyList<T>(Type attributeType)
{
    var _properties = typeof(T).GetProperties().Where(_property => Attribute.IsDefined(_property, attributeType));
    return _properties.ToList();
}
```

Usage:

PHP Code

```
<?php  
var _properties = GetPropertyList<CustomType>(typeof(CustomAttribute));
```

How is that related to my question??

?DiViN3 - March 30, 2018, 10:05 pm

(Mar 30, 2018 09:32 PM)severos Wrote: [\[CITE\]](#) How is that related to my question??
Retrieves all the property info from the selected category.

severos - March 30, 2018, 10:10 pm

(Mar 30, 2018 10:05 PM)?DiViN3 Wrote: [\[CITE\]](#) Retrieves all the property info from the selected category.

Who ever said anything about retrieving property info?? Have you even used Entity Framework before? because you seem to not understand what I'm looking for here.

maype - April 1, 2018, 12:01 pm

Maybe, the best way to understand what you are doing is to learn SQL. Beside using class relation & some special method to achieve to extract you data. Entity Framework just manage SQL for you.

Good luck !

severos - April 1, 2018, 12:09 pm

(Apr 1, 2018 12:01 PM)maype Wrote: [\[CITE\]](#) Maybe, the best way to understand what you are doing is to learn SQL. Beside using class relation & some special method to achieve to extract you data. Entity Framework just manage SQL for you.

Good luck !

I already know basic SQL, but writing a stored procedure to do that search (although it might have slightly better performance) will look ugly as well, it's just moving loop ugliness from my code to the server.

maype - April 1, 2018, 12:21 pm

From what I understand from your question, you want to join the Sub Categories from the found categories right ??

I found this on stackoverflow. Maybe it can help. Sorry if I dont get ur needs.

See:

<https://stackoverflow.com/questions/5010110/entityframework-join-using-join-method-and-lambdas>

```
Code var query = db.Categories          // source  
    .Join(db.CategoryMaps,             // target  
        c => c.CategoryId,             // FK  
        cm => cm.ChildCategoryId,      // PK  
        (c, cm) => new { Category = c, CategoryMaps = cm }) // project result  
    .Select(x => x.Category); // select result
```

Sakuto - April 1, 2018, 12:46 pm

My point would be to:

- First retrieve every categories related to a parent category (letting EF doing with lazy loading, or just use Include())
- Second, retrieve every article which category is in the retrieved list (.Where(a => ids.Contains(a.Id)))

Two request, but I don't see something simpler atm.

severos - April 1, 2018, 8:57 pm

(Apr 1, 2018 12:21 PM)maybe Wrote: [\[CITE\]](#) From what I understand from your question, you want to join the Sub Categories from the found categories right ??

I found this on stackoverflow. Maybe it can help. Sorry if I dont get ur needs.

See:

<https://stackoverflow.com/questions/5010110/entityframework-join-using-join-method-and-lambdas>

```
Code var query = db.Categories           // source
    .Join(db.CategoryMaps,              // target
        c => c.CategoryId,              // FK
        cm => cm.ChildCategoryId,       // PK
        (c, cm) => new { Category = c, CategoryMaps = cm }) // project result
    .Select(x => x.Category); // select resultNo, I'm not looking to join tables at all...
```

(Apr 1, 2018 12:46 PM)Sakuto Wrote: [\[CITE\]](#) My point would be to:

- First retrieve every categories related to a parent category (letting EF doing with lazy loading, or just use Include())
- Second, retrieve every article which category is in the retrieved list (.Where(a => ids.Contains(a.Id)))

Two request, but I don't see something simpler atm.

That's actually 2 requests if I have single nesting level, consider something like this:

-Master Category

-Parent1

-Parent2

-Child2-1

-Parent3

-Child3-1

If I'm searching using Parent3 id then your method works in 2 requests. However, if I'm searching using Parent1, first query will retrieve Parent2 and Parent3 ids, then I'll have to query each of them to get lowest level children ids, then query products with the list of IDs.

Ninja - April 2, 2018, 7:17 am

(Apr 1, 2018 12:09 PM)severos Wrote: [\[CITE\]](#) it's just moving loop ugliness from my code to the server.

Isn't that the whole point of abstraction? To move "ugliness" somewhere else? If I could use a stored procedure, no matter how ugly, I would. It is cleaner on the client side.

severos - April 2, 2018, 7:28 am

(Apr 2, 2018 07:17 AM)Ninja Wrote: [\[CITE\]](#) Isn't that the whole point of abstraction? To move "ugliness" somewhere else? If I could use a stored procedure, no matter how ugly, I would. It is cleaner on the client side.

I thought abstraction splits ugliness around to tiny bits that you no longer see them? I'll probably end up using stored procedures, just wanted to see if there are other alternatives around.

Sakuto - April 2, 2018, 10:44 am

<https://stackoverflow.com/questions/11565423/most-efficient-method-of-self-referencing-tree-using-entity-framework> Taking profit of EF.

Ninja - April 2, 2018, 12:20 pm

(Apr 2, 2018 07:28 AM)severos Wrote: [\[CITE\]](#) I thought abstraction splits ugliness around to tiny bits that you no longer see them?

Isn't putting ugliness server sided exactly that?

severos - April 3, 2018, 11:10 am

(Apr 2, 2018 10:44 AM)Sakuto Wrote: [\[CITE\]](#)

<https://stackoverflow.com/questions/11565423/most-efficient-method-of-self-referencing-tree-using-entity-framework> Taking profit of EF.

I have the feeling it's going to be a really helpful link... but I can't seem to understand what's going on in there ? guess I need to rest a bit first.

But thanks for the help ?

Assault - April 3, 2018, 3:41 pm

There is no easy way to do this with the ORM if you want the database server to do the heavy lifting. You are much better off writing a stored procedure and use a recursive CTE in this scenario because of the complex requirements.

see this url for an example: [https://technet.microsoft.com/en-us/library/ms186243\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms186243(v=sql.105).aspx)

severos - April 4, 2018, 2:33 pm

(Apr 3, 2018 03:41 PM)Assault Wrote: [\[CITE\]](#) There is no easy way to do this with the ORM if you want the database server to do the heavy lifting. You are much better off writing a stored procedure and use a recursive CTE in this scenario because of the complex requirements.

see this url for an example: [https://technet.microsoft.com/en-us/library/ms186243\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms186243(v=sql.105).aspx)

Now this is really something interesting, I'll probably end up using that. Thanks so much for the link ?