Subject: Tree structure
Posted by htManager on Sat, 05 Jul 2014 10:17:06 GMT
View Forum Message <> Reply to Message

Hi Tony,

I worked through the tutorials and trough the article 'A Flexible Tree Structure' to implement my own tree structure. It is very interesting to see how you can build such flexible structures.

Now my question: Is it possible to build a tree structure and a tree view with other (existing) tables with cascading primary keys? I would like to show a hirarchy where I can use the occurances of existing tables. If I insert the levels and nodes in the three tables x_tree_type, x_tree_level and x_tree_node I would have to do twice.

Subject: Re: Tree structure
Posted by AJM on Sun, 06 Jul 2014 08:40:21 GMT
View Forum Message <> Reply to Message

In theory you can use whatever key structure you like. All you have to do is amend my sample code accordingly.

Subject: Re: Tree structure
Posted by htManager on Mon, 07 Jul 2014 09:52:09 GMT
View Forum Message <> Reply to Message

ok. I will try it.
Thank you very much.

Subject: Re: Tree structure
Posted by htManager on Wed, 11 Feb 2015 16:05:10 GMT
View Forum Message <> Reply to Message

I still struggle with the generation of the tree structure. I read the tutorial several times and also looked in the Xample application.

I now have the problem that I don't see the + and - signs for expanding and collapsing my structure.

I have a primary key with three coloumns and changed the code the specified class files but with no success.

I have three levels. Under each level I inserted a node. Isn't it that level two should be expandable/collapsable in the tree structure? If not, which section of the tutorial should I read once again?

Subject: Re: Tree structure
Posted by AJM on Thu, 12 Feb 2015 09:34:00 GMT
View Forum Message <> Reply to Message

It would be impossible for me to determine what is wrong with your code just from this description. I could only do that by stepping through the code with my debugger, but that would require you to send me a copy of your, and your database schema, and your test data.

Subject: Re: Tree structure
Posted by htManager on Thu, 12 Feb 2015 13:31:50 GMT
View Forum Message <> Reply to Message

I now see what I did wrong. You can link children to a selected parent node. Sometimes it takes a little bit more time to recognize the right way.

Subject: Re: Tree structure
Posted by AJM on Fri, 13 Feb 2015 10:10:38 GMT
View Forum Message <> Reply to Message

I'm glad you got it sorted out.

Subject: Re: Tree structure
Posted by htManager on Thu, 26 Feb 2015 14:12:24 GMT
View Forum Message <> Reply to Message

Hi Tony,
it's me again. I have a new problem. I see the tree structure twice. One time with the collapsed entry node and then underneath all (the same) nodes which belong to that tree.
I have checked the queries in the _cm_getNodeData() function but I don't see my fault. The difference to your sample application is that I work here with a primary key, consisting of three colomns. Can this be the problem?
If I expand the tree I can see the correct structure but always twice.

Where do I have to search for my fault?

Subject: Re: Tree structure
Posted by AJM on Sat, 28 Feb 2015 11:20:30 GMT
View Forum Message <> Reply to Message

You have to look at the SQL queries you are generating. If you are getting the wrong data back from a query then you need to adjust that query.

Subject: Re: Tree structure
Posted by htManager on Sat, 28 Feb 2015 13:31:20 GMT
View Forum Message <> Reply to Message

I suppose that the queries are generated in the _cm_getNodeData of the class file. I copied the code from the xample file and tried to change it. But I haven't much experience in generating SQL-queries, so that I suppose that I did something wrong. Especially because I have a primary key consisting of several fields. Here is my code, maybe you can see what I am doing wrong:

```
        $this->sql_select  =  'hallen_plaetze_nodes.node_id, '
          . 'hallen_plaetze_nodes.hallen_plaetze_node_desc, '
          . 'hallen_plaetze_nodes.hallen_plaetze_nodes_logo_fname, '
          . 'hallen_plaetze_level.hallen_plaetze_level_seq, '
          . 'COUNT(child.node_id) AS child_count';
        $this->sql_from    = 'hallen_plaetze_nodes '
                          .  'LEFT JOIN hallen_plaetze_level ON '
                          .  '(hallen_plaetze_level.verbaende_art_id=hallen_plaetze_nodes
.verbaende_art_id AND '
                          .  'hallen_plaetze_level.verbaende_kuerzel=hallen_plaetze_nodes
.verbaende_kuerzel AND '
                          .  'hallen_plaetze_level.halle_no=hallen_plaetze_nodes.halle_no AND '
                          .  'hallen_plaetze_level.hallen_plaetze_level_id=hallen_plaetze
_nodes.hallen_plaetze_level_id) '
                         . 'LEFT JOIN hallen_plaetze_nodes AS child ON
(hallen_plaetze_nodes.node_id=child.node_id_snr) ';
        $this->sql_where   = '';
        $this->sql_groupby =  'hallen_plaetze_nodes.node_id, '
          . 'hallen_plaetze_nodes.hallen_plaetze_node_desc, '
          . 'hallen_plaetze_nodes.hallen_plaetze_level_id, '
          . 'hallen_plaetze_level.hallen_plaetze_level_seq';
        $this->sql_having  = '';
        $this->sql_orderby =  'hallen_plaetze_nodes.hallen_plaetze_level_id,'
          . 'hallen_plaetze_nodes.node_id';
```

Subject: Re: Tree structure
Posted by AJM on Sun, 01 Mar 2015 09:23:13 GMT
View Forum Message <> Reply to Message

I cannot possibly debug such a complicated SQL query unless I have a copy of your database schema, some test data, and a description of what you are trying to do.

You really should learn more about SQL and writing queries as without this knowledge you will find it difficult to extend the framework. You need to play with your queries outside of the framewor until you find the right query, then get the framewor to generate that query.

Subject: Re: Tree structure
Posted by htManager on Mon, 02 Mar 2015 09:44:17 GMT
View Forum Message <> Reply to Message

Ok, I will do so. I understood. Generating the correct SQL-query will lead to the correct tree view! The query result must show the entries only once, not twice!
Thank you.

Subject: Re: Tree structure
Posted by htManager on Thu, 23 Apr 2015 15:49:17 GMT
View Forum Message <> Reply to Message

I am about to improve my SQL skills as you adviced. I think I now know my fault. The difference between your xample application and my application is that in your xample application only entries are shown with tree_level_id = 1. In my application there are entries shown with tree_level_id 1 and greater. That is why I see everything twice.
Can you tell me where I have to look for if I made the correct selection? Is it in a class or in a relationship?

Subject: Re: Tree structure
Posted by AJM on Fri, 24 Apr 2015 08:07:29 GMT
View Forum Message <> Reply to Message

You have to adjust the first query to only select those entries at the top of the tree (ie: with tree_level=1). You are controlling the queries which the code issues, so it is up to you to get those queries right.

For your information I have successfully implemented tree views in my enterprise application using tables which do not include column names such as TREE_TYPE, TREE_LEVEL or TREE_NODE, and which have compound primary keys, so I know that it is definitely possible to do so. My most recent usage was to display a Bill of Materials (BOM) in my PRODUCT database which uses a single table which contains the product hierarchy, as shown below:

```
CREATE TABLE `product_component` (
 `product_id_snr` VARCHAR(40) NOT NULL,
 `product_id_jnr` VARCHAR(40) NOT NULL',
 `seq_no` SMALLINT(5) UNSIGNED NOT NULL,
 `revision_id_snr` VARCHAR(16) NULL DEFAULT NULL,
 `revision_id_jnr` VARCHAR(16) NULL DEFAULT NULL,
 `quantity` DECIMAL(15,6) UNSIGNED NOT NULL DEFAULT '1.000000',
 `start_date` DATE NOT NULL DEFAULT '2000-01-01',
 `end_date` DATE NULL DEFAULT '9999-12-31',
 PRIMARY KEY (`product_id_snr`, `product_id_jnr`, `seq_no`)
);
```

Subject: Re: Tree structure
Posted by htManager on Mon, 04 May 2015 15:48:12 GMT
View Forum Message <> Reply to Message

I solved my problem. My fault was that I used the wrong key to filter the entries with level_id '1' when the program was looking for root nodes within the tree type.
Thank you very much once again for your advice.

Subject: Re: Tree structure
Posted by AJM on Tue, 05 May 2015 08:30:30 GMT
View Forum Message <> Reply to Message

No problem. I'm glad you got it sorted.

Subject: Re: Tree structure
Posted by htManager on Mon, 01 Jun 2015 15:32:46 GMT
View Forum Message <> Reply to Message

What is the best transaction pattern to insert many nodes into another table?

I have a table where I can show with tree2 pattern the structure and above all the seats of a sportstadiom. Now I want insert all the (300-400) seats of the stadiom into a table where I want to print tickets for the games.

Both tables have composed primary keys. The primary key of the 'nodes' table is part of the primary key of the 'ticket' table. But the primary key of the 'ticket' table exists also of three more coloums. If I want to generate a popup (list2 or tree2) I get an error message that the tables are not related. I think this is because of the additional primary key components of the 'ticket' table.

Have you an idea and can you give me a hint which transaction pattern I can use?

Subject: Re: Tree structure
Posted by AJM on Tue, 02 Jun 2015 09:12:31 GMT
View Forum Message <> Reply to Message

If what you have is similar to a MANY-to-MANY relationship, which is actually implemented as a ONE-to-MANY-to-ONE with a "link" or "xref" entity in the middle, then you should look at the ADD3 pattern.

Subject: Re: Tree structure
Posted by htManager on Tue, 02 Jun 2015 14:46:07 GMT
View Forum Message <> Reply to Message

Yes, I think this is what I mean. But how can I generate the transaction pattern. Did you somewhere describe how to generate it? I only have the choice to generate add4 or add5.

Subject: Re: Tree structure
Posted by AJM on Tue, 02 Jun 2015 17:05:26 GMT
View Forum Message <> Reply to Message

This ADD3 pattern is normally listed as an option with a LIST2 pattern. If you do not want, or cannot create as LIST pattern you can always create twe script mannually. Just copy the ADD3.PHP script from the "default" directory into your subsystem directory, rename as required, amend the rerferences inside it as required, then manually create an entry on the MNU_TASK table.

Subject: Re: Tree structure
Posted by htManager on Mon, 11 Jan 2021 17:54:23 GMT
View Forum Message <> Reply to Message

Hi Tony,

I am about to create a new tree structure and I work with the Xample application as a template. I have now an error in my application when I update the level sequence number in a wrong way. I checked the behaviour in the Xample accplication and the same error comes up when I do the following:

The project type has 3 levels. If I change level 3 to 4 the error message tells me: Must not be greater than 3 in the record with level 2. If I change the sequence number in the third record back to 3 the field keeps the value 4 and the error message tells me that the value must not be greater than 3 without updating the record and I have to cancel the task.

If I change the sequence numbers in the right way, everything works fine. It's only for your information.

Subject: Re: Tree structure
Posted by AJM on Mon, 11 Jan 2021 18:50:22 GMT
View Forum Message <> Reply to Message

If you have 3 levels they are numbered 1, 2 and 3. If you want to insert a new level between 2 and 3 you must add it as level 4 first, then you can use the "Update Level Sequence Number" task to swap levels 3 and 4 around. This prevents you from specifying a sequence number that does not currently exist.

Subject: Re: Tree structure

Posted by htManager on Mon, 11 Jan 2021 20:47:28 GMT
View Forum Message <> Reply to Message

Okay, thank you for the hint. And if I inserted 5 by mistake I have to cancel and to call the task once again. That's what I meant.

Subject: Re: Tree structure
Posted by htManager on Wed, 13 Jan 2021 09:10:59 GMT
View Forum Message <> Reply to Message

I would like you to ask your advice in the following issue:

I want to show a tree structure existing of several (hierarchical) tables. You can imagine this like a sports team diagram, in football/soccer this would be FIFA -> UEFA/AFC/... -> FA/DFB/... -> ... -> Liverpool -> Saisons -> Team A -> Members/Team mates

For example: I have a table where I store the associations and the clubs like FIFA, Liverpool, FA etc. with Key A, Key B, Key C. Then I have a table where I store the teams of a club with Key A, Key B, Key C and Key D. And I have a few tables more.

I do know on which level I want to insert these nodes but how I can build up this in my table structure. Would you advice to insert a coloumn in the tree_node table where I store the tablename to look for the node_id's? If yes, in which function could I check this? Or would it be better to work with a subclass of the tree_nodes table?

On top of this problem I want to design the tree structure in a virtual private database where I can store the associations and the clubs in the shared account and only the teams and the team members are created in the respective accounts to lower the quantity of records. I implemented this in these tables yet and it works fine. I think this should be only a question of the keys and the SQL queries.

Subject: Re: Tree structure
Posted by AJM on Wed, 13 Jan 2021 10:25:18 GMT
View Forum Message <> Reply to Message

My Tree Structure is not a universal design for all possible hierarchical structures, so may not be the most optimal solution in all circumstances. For example, in my ERP system I have a PRODUCT subsystem which needs to include a Bill Of Materials (BOM) which can go down many levels. For this I have a single table which holds the entire hierarchy using just 3 columns - product_id_snr, product_id_jnr and quantity.

You should start by drawing a structure diagram which shows all the levels in your hierarchy and identify which are nodes and which are leaves. Remember that an outside table, such as PLAYER, could include a node_id column as a foreign key which points to one of the entries, at any level, in the tree structure.

Subject: Re: Tree structure
Posted by htManager on Wed, 13 Jan 2021 11:13:54 GMT
View Forum Message <> Reply to Message

I have up to 10 levels. The first 5 levels are stored in one table so that should be no problem.

According to your Bill Of Material, every table should have a coloumn node_id if an occurrence is a leaf in the tree and additionally a coloumn node_id_snr if this occurrence can be a node as well, isn't it?

I will start as you recommended with my structure diagram and try to implement the tree structure level by level from top to bottom. An advantage should be that the structure doesn't necessarily has to be as flexible as you described it in the tutorial. So I hope I will get it done.

Subject: Re: Tree structure
Posted by AJM on Thu, 14 Jan 2021 10:09:14 GMT
View Forum Message <> Reply to Message

It is always a good idea to draw the hierarchy structure on a piece of paper first so that you can draw the lines of connectivity between each element in that structure so that you can see for each node which is its parent and which are its children. You may also need to identify where connectivity is invalid so must be prevented. In my TREE structure each node can have only one parent (or no parent for the top node) but any number of children. Does the same rule apply in your structure.

Once you have drawn a structure which identifies and follows all the rules it will be easier to map that structure to tables in the database, and if the structure is right the coding will be (relatively) easy.

Subject: Re: Tree structure
Posted by htManager on Sat, 16 Jan 2021 08:43:16 GMT
View Forum Message <> Reply to Message

As you said, if you have drawn the structure it is easy to work! I changed the $this->fieldspec in the _cm_changeconfig() method to change the popUp calls according to the choosen levels. Works fine. In the _cm_popUpCall() method I can change $where to identify the correct occurences.

But with doing this I was aware that I create a second structure in which I have to insert every new team and every new player. But all I want is to show a tree structure of the existing data without modifying the (new) tree structure itself.

I don't need any either such a flexibility as you showed in your example. My structure is clear and there will be normally no (structure) changes except additional saisons, teams and payers etc.

For this reason I will try to integrate the necessary information in the existing tables like you did in

your BOM example.

As far as I see, I need three additional coloumns in the relevant tables: node_id, node_id_snr and level_id. Farther I need a table where all the levels are stored in the desired sequence. Each time a new occurence will be inserted in an existing table, the code has to determine the next node_id and the appropriate node_id_snr. This should work without any bigger problems (I hope) because a level has only one table where the relevant records are stored.

If there are faults in my thoughts, please let me know.

Subject: Re: Tree structure
Posted by AJM on Sat, 16 Jan 2021 10:06:02 GMT
View Forum Message <> Reply to Message

That seems fine to me. I have always found that being able to visualise the structure always helps. Get the structure right and the coding is easy. Get it wrong and the coding will always be hard.

Subject: Re: Tree structure
Posted by htManager on Sun, 24 Jan 2021 19:57:15 GMT
View Forum Message <> Reply to Message

I could create a proper tree structure which is working fine. Now I wanted to create a popup3 pattern with this tree structure. In http://www.tonymarston.net/php-mysql/dialog-types.html#popup you can see that this should be a list1 pattern as popup. But in the creation process I have to choose an outer table just as in the popup4 pattern. And the popup4 *.php pattern looks the same as the *.php popup3 pattern. Is this correct or do I have to do something else?

It was my fault. I chose the wrong type. Everything works fine. Sorry for the post.
