
Subject: LIST3 linked from LIST1
Posted by [rafs](#) on Wed, 08 Jul 2015 23:08:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

LIST3: As I scroll through the parents, or children, the grandchild does not change parent --< children --< grandchildren
The parent and child data changes when I scroll at those levels, but grandchild remains unchanged.

I have linked this LIST3 from the parent's LIST1.

Subject: Re: LIST3 linked from LIST1
Posted by [AJM](#) on Thu, 09 Jul 2015 09:44:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

As you scroll through the parent entries the primary key of the current parent is passed down to child entity where it will enable you to scroll through all available entries one at a time. The primary key of the current child entity is then passed down to the grandchild entity where it will enable you to scroll through all available entries one page at a time.

You need to ensure that the SELECT list in each SQL query includes the primary key for that table. If you miss this out there is no value which can be passed down to the child entity.

If there are any differences between the name of the primary key in the parent table and the name of the corresponding foreign key in the child table, then you must convert the field name manually in your code as the framework cannot do this automatically.

Subject: Re: LIST3 linked from LIST1
Posted by [rafs](#) on Thu, 09 Jul 2015 20:17:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

AJM wrote on Thu, 09 July 2015 05:44 If there are any differences between the name of the primary key in the parent table and the name of the corresponding foreign key in the child table, then you must convert the field name manually in your code as the framework cannot do this automatically.

I am following the convention that every table's primary key is called 'id', and every table's foreign key is called <referenced-table>_id. This is the convention that I picked up from CakePHP, and probably RoR. This convention was a requirement for the project. So, for example, packages has many files (WHERE packages.id = files.package_id), and files have many lines (WHERE files.id = lines.file_id).

The parent to child part works when using this convention, but that the child to grandchild linking does not.

Subject: Re: LIST3 linked from LIST1
Posted by [rafs](#) on Thu, 09 Jul 2015 20:55:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

AJM wrote on Thu, 09 July 2015 05:44 If there are any differences between the name of the primary key in the parent table and the name of the corresponding foreign key in the child table, then you must convert the field name manually in your code as the framework cannot do this automatically.

I thought that the point of defining the relationships in the data dictionary was to handle these linking fields for me? When I defined the child relationships for the parent tables, it explicitly asks me to specify which field in the junior table is the foreign key to the primary key of the senior table, so this mapping should be handled automatically by the framework, no?

For example, Figure 13 in the first tutorial. Your example db x_option table is linked to x_pers_opt_xref WHERE "x_option.option_id = x_pers_opt_xref.option_id". I assumed that the fields need not have the same name.

Subject: Re: LIST3 linked from LIST1
Posted by [rafs](#) on Thu, 09 Jul 2015 22:10:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

AJM wrote on Thu, 09 July 2015 05:44... then you must convert the field name manually in your code ...

Where?

Subject: Re: LIST3 linked from LIST1
Posted by [rafs](#) on Thu, 09 Jul 2015 23:12:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok, I see your documentation, <http://www.tonymarston.net/php-mysql/database-design.html#field.names.and.content>
I am renaming my db fields, and will try again.
Thanks for the documentation; I am learning a lot.

Subject: Re: LIST3 linked from LIST1
Posted by [AJM](#) on Fri, 10 Jul 2015 08:43:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

rafs wrote on Thu, 09 July 2015 21:17
I am following the convention that every table's primary key is called 'id', and every table's foreign key is called <referenced-table>_id. This is the convention that I picked up from CakePHP, and probably RoR. This convention was a requirement for the project. So, for example, packages has

many files (WHERE packages.id = files.package_id), and files have many lines (WHERE files.id = lines.file_id).

That may be a convention that some newbies with lesser experience follow, but the convention that I and my fellow developers have been following since the 1970s is that wherever possible the same column should have the same name regardless of which table it's on. All the SQL samples you see on any vendor's website still follow this convention, which is why you see "customer.customer_id" and not "customer.id". Note also the use of snake_case and not CamelCaps.

Subject: Re: LIST3 linked from LIST1
Posted by [AJM](#) on Fri, 10 Jul 2015 08:52:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

rafs wrote on Thu, 09 July 2015 21:55
I thought that the point of defining the relationships in the data dictionary was to handle these linking fields for me? When I defined the child relationships for the parent tables, it explicitly asks me to specify which field in the junior table is the foreign key to the primary key of the senior table, so this mapping should be handled automatically by the framework, no?

Not quite correct. This dictionary information can be used when including JOINS into an SQL query as the code iterates through ALL the parent_relation entries. This is documented in Using Parent Relations to construct sql JOINS. When receiving a \$where string from another object I do not know which table this came from or which of the possible relationships to use.

Subject: Re: LIST3 linked from LIST1
Posted by [AJM](#) on Fri, 10 Jul 2015 08:55:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

rafs wrote on Thu, 09 July 2015 23:10AJM wrote on Thu, 09 July 2015 05:44... then you must convert the field name manually in your code ...

Where?

If you look in /menu/classes/mnu_nav_button_jnr.class.inc and /menu/classes/mnu_nav_button_snr.class.inc you will see code in the _cm_pre_getData() method which does this. You could also use the _cm_initialise() method if you wished.

Subject: Re: LIST3 linked from LIST1
Posted by [AJM](#) on Fri, 10 Jul 2015 09:01:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

rafs wrote on Fri, 10 July 2015 00:12Ok, I see your documentation,
<http://www.tonymarston.net/php-mysql/database-design.html#field.names.and.content>

I am renaming my db fields, and will try again.

It is not necessary to convert the column names in your database, you can convert the contents of the \$where string instead. I have to do this in the framework in those places where I have two foreign keys on a table which link to the same primary key, such as in the MNU_MENU table where menu_id and task_id_jnr both link to MNU_TASK.task_id. In cases like this it is simply not possible to rename the two foreign keys to match the same primary key, so I must deal with this in the code.

Subject: Re: LIST3 linked from LIST1

Posted by [rafs](#) on Sat, 11 Jul 2015 22:19:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

I originally skipped over your articles on database stuff. I have read them now. I guess I am rated as a novice with borderline nincompoop tendencies.

Anyways, I didn't see anything explicitly about the following (hypothetical) situation. A parent has many children, but always has a favorite (favourite) child. Are there any special considerations to be aware of? I would assume that I can just define each of these FK relationship in the data dictionaries, but I wonder if I will have to add some code since parent.child_id_favorite and child.child_id have different names.

parent

```
parent_id int primary key,  
parent_name char(10),  
child_id_favorite int,  
FOREIGN KEY (child_id_favorite) REFERENCES child(child_id)
```

child

```
child_id int primary key,  
parent_id int,  
child_name char(10),  
FOREIGN KEY (parent_id) REFERENCES parent(parent_id)
```

Thanks.

Subject: Re: LIST3 linked from LIST1

Posted by [AJM](#) on Sun, 12 Jul 2015 09:28:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

What you have there is a circular relationship, but having a foreign key linking to a primary key with a different name is something that occurs more than once in the MENU database. For example:

parent (MNU_TASK)

task_id VARCHAR(80)

child (MNU_MENU)

menu_id VARCHAR(80) REFERENCES parent(task_id)

task_id_jnr VARCHAR(80) REFERENCES parent(task_id)

A relationship is a relationship, even if it is circular, so you handle it as any other relationship.
