Subject: SQL Syntax error
Posted by gpatti on Wed, 15 Jul 2009 23:15:27 GMT
View Forum Message <> Reply to Message

Tony,

I've managed to generate a SQL syntax error through use of $selections in a where clause. I think there is a bracket missing in the generated SQL.

The sequence of events is:

1. I am starting with a LIST2 transaction and selecting some rows
2. I call a 'dummy' ADD2 transaction so that I can select some additional criteria before using scriptNext() in _cm_pre_insertRecord to call an output task.

This works fine if I select 0 rows or 1 row (giving me output for all rows matching the additonal criteria or the single row I have selected. However, if I select more than one row I get a SQL syntax error.

The call is scriptNext($next_task, $where) with
$where = "crf_type.crf_type_id = '$crf_type_id' AND $selections";

However this generates the following SQL WHERE clause:
WHERE (exam.exam_id='6646' AND exam.rdcaccount_id='7') OR (exam.exam_id='6647' AND exam.rdcaccount_id='7')) AND crf_type.crf_type_id='T')

I think there should be a double opening bracket after the WHERE keyword to match the double closing bracket surrounding the $selections criteria that is grouped with the OR clause.

Let me know if you need further information to assist with this.

Thanks,

Graham

Subject: Re: SQL Syntax error
Posted by AJM on Thu, 16 Jul 2009 09:33:35 GMT
View Forum Message <> Reply to Message

What version of Radicore are you using? I've tried this with the latest version and I can't get it to fail.

Subject: Re: SQL Syntax error

Posted by gpatti on Thu, 16 Jul 2009 09:47:41 GMT
View Forum Message <> Reply to Message

I'm using 1.49

Looking at my code there is another aspect that may be contributing to this. I am adding a join to another table in _cm_pre_getData() according to the procedure described in one of the FAQs. This has been working fine until the addition of the $selections criteria as I described earlier.

```
        if (empty($this->sql_from)) {
            // construct default SELECT and FROM clauses using parent relations
            $this->sql_from    = null;
            $this->sql_groupby = null;
            $this->sql_having  = null;
            $this->sql_from    = $this->_sqlForeignJoin($this->sql_select, $this->sql_from,
$this->parent_relations);

          // add code to obtain grade amount
          $this->sql_select .= ", crf_type.crf_type_id";
          $this->sql_from   .= " LEFT JOIN crf_type ON branch.crf_type_id=crf_type.crf_type_id";
      } // if
```

I'm also going to try a couple of things in the meantime.

Graham

Subject: Re: SQL Syntax error
Posted by AJM on Thu, 16 Jul 2009 10:08:00 GMT
View Forum Message <> Reply to Message

Here's another idea - instead of using a dummy ADD2 task to provide additional selection criteria before the output task why not try a SEARCH task instead. You can use the new _cm_post_search() method to identify the output task which needs to be called next. I added this functionality as I had a need to enter additional selection criteria before calling an output task as a batch process.

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 10:24:03 GMT
View Forum Message <> Reply to Message

I'll give it a try.

Incidentally, when preparing the previous post I realized I didn't need to build the query manually using _cm_pre_getData() after all if I adjusted one of my relationships.

However, having done that, I am now just using the standard generated query but still get the

same error in the WHERE clause.

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 11:13:03 GMT
View Forum Message <> Reply to Message

Not having much success with _cm_post_search()...

The $where parameter provided does not contain the original data passed down from the parent screen, so I am losing that. Do I need to do something (in _cm_initialise?) to retain that data?

The $selection parameter is empty, regardless of what selections I make.

What might I have done wrong?

Subject: Re: SQL Syntax error
Posted by AJM on Thu, 16 Jul 2009 13:52:42 GMT
View Forum Message <> Reply to Message

I have just checked what is happening on my system, and I cannot see the problem you are having. When I press the SUBMIT button in the SEARCH screen and see what is available in the _cm_post_search() method I have the following:

the $where string contains whatever was selected in the current SEARCH screen.
the $selection string identifies whatever entries were selected in the screen prior to the SEARCH.

When the OUTPUT1 task is activated these strings are available in the _cm_pre_getData() method as:

$where contains the $selection string in the SEARCH task.
$this->sql_search contains the $where string in the SEARCH task.

You might want to try the attached files as I have made some slight alterations since the last release.

(attachment deleted, updated in later post)

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 14:48:37 GMT
View Forum Message <> Reply to Message

The $where string IS containing what I have selected in the search screen, but ONLY that. When

the search was opened it contained the primary key passed down from the parent screen, and I still need that to pass on to the next task, but it is being lost.

Redarding the new versions these are producing a login error (rdcaccount_id[0]: sys0189) so I can't try it.

Subject: Re: SQL Syntax error
Posted by AJM on Thu, 16 Jul 2009 15:41:12 GMT
View Forum Message <> Reply to Message

This attachment contains the files which were missing is the previous post, so this should not produce the error that you encountered.

The missing $where string passed down from the parent LIST2 task is by design. It is shown in the SEARCH screen as non-editable as it is fixed within the LIST2 task and cannot be changed. Because it is non-editable it is not included in the POST array when the SUBMIT button inside the SEARCH screen is pressed, which also means that it is not included in the $where string produced by the SEARCH screen. This does not matter when the SEARCH screen passes control back to the LIST2 task as the original $where string is still in the LIST2 task, and will be merged with whatever is passed back in the $search string.

In your case you want to trap the contents of $where from the LIST2 task so that it can be added to the $search string which is passed to the next task. You can trap this in the $where string in the _cm_initialise() method. Save it to a class variable so that it is available to you in the _cm_post_search() method.

File Attachments
1) radicore-2009-07-16-update.zip, downloaded 2457 times

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 16:03:29 GMT
View Forum Message <> Reply to Message

Actually Tony, that's exactly what I've done regarding $where to get the right data passed to the next task. I've also used that technique to get $selection passed through also (although I will see if the updated files make that unnecessary). So the functionality is working correctly now.

There is a side effect of using append2ScriptSequence() rather than scriptNext().

With scriptNext() once my output task is completed (it's an output3) control returns to the dummy add screen and I can change the additional selection criteria and SUBMIT to run another output, or CANCEL the screen to return to the LIST2.

With append2ScriptSequence() after the output is completed control also returns (to the search screen this time), but now on pressing SUBMIT or CLEAR I am always returned to the LIST2

screen, and the list is now filtered, which I don't want.

I used append2ScriptSequence() because of the notes in your example. Can scriptNext() be used instead, or would you suggest an alternative way of changing the behaviour?

Subject: Re: SQL Syntax error
Posted by AJM on Thu, 16 Jul 2009 16:21:14 GMT
View Forum Message <> Reply to Message

Yes, you can use scriptNext() instead of append2ScriptSequence(). You need to choose the one you want depending on what behaviour is best for you. If the next task is a batch task then append2ScriptSequence() is best.

BTW, I've decided that instead of having to create your own code to save the contents of the $where string which was passed down from the parent LIST2 task it would be much easier if I saved the value inside std.search1.inc so that it is automatically available inside _cm_post_search(). I haved changed line 57 to read as follows:

```
    if (!empty($selection)) {
     $dbobject->sql_selection = $selection;
    } else {
        $dbobject->sql_selection = $where;
    } // if
```

$dbobject->sql_selection will be made available as the $selection argument inside the _cm_post_search() method.

This means that the contents of $where will only be used if $selection is empty. You should not need any custom code inside _cm_initialise() to save the contents of either $where or $selection.

File Attachments

1) std.search1.zip, downloaded 2556 times

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 16:27:50 GMT
View Forum Message <> Reply to Message

Thanks and I'll give that a try now.

I can confirm that $selection is now available in _cm_post_search with the new files.

Regarding the return to the LIST2 and the data being filtered it's occurred to me that this is probably because I am now using a SEARCH tsk rather than an ADD task (rather than the use of append2ScriptSequence().)

I'll use scriptNext() because I prefer being able to action subsequent output tasks, but if I am correct, what would I need to do to remove any filtering on return to the LIST2?

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 16:37:29 GMT
View Forum Message <> Reply to Message

This latest version of std.search1.inc has 'broken' what was 'fixed' in the slightly earlier version you gave me!!

$selection is now coming through NULL, and $where is still as it was, containing the items selected in the search screen.

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 16:53:52 GMT
View Forum Message <> Reply to Message

Also Tony, I am back where I started now with the SQL syntax error. It appears that the SQL error only occurs with the use of scriptNext() in the way it is handling $selection.

Interestingly, changing the order of the variables in the $where statement generates slightly different versions of the same syntax error. The brackets seems to get a bit confused.

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 17:02:41 GMT
View Forum Message <> Reply to Message

scriptNext() is working!!

However, it seems the variable order is significant after all. I now have
$where = "$selection AND branch.crf_type_id = '$crf_type_id'";
The difference since the very first post is the order of the selection criteria variable and $selection. It fails if I put them the other way round. I had also changed a relationship so that the table qualifier is different, but I don't think that was significant. I'd accidentally had an additional 'AND' when I did the previous post so that solves the different versions of the same syntax error. Sorry about that one.

Subject: Re: SQL Syntax error
Posted by AJM on Thu, 16 Jul 2009 17:12:13 GMT
View Forum Message <> Reply to Message

Does that mean your problems are now solved?

Subject: Re: SQL Syntax error
Posted by gpatti on Thu, 16 Jul 2009 17:20:21 GMT
View Forum Message <> Reply to Message

Yes thanks.

I can save $selection and $where in the initialise method as that works ok.

It might just be worth you noting the problem with the variable order in case you get chance to look at it in the future (if you can reproduce it) or in case anyone else comes across it.

But everything is fine for now.

Thanks,

Graham