

Database - Bug #3315

infinite loop in mark_client_branch during annotations because of nest CAN-FINDs with temp and perm tables

08/10/2017 11:01 AM - Greg Shah

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Eric Faulhaber	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#1 - 08/10/2017 11:04 AM - Greg Shah

Bug report (from a customer):

The problem is with nested CAN-FINDs which mix ordinary table and temp-table. In this case FWD conversion process hangs.

I have located the reason: it is while-loop which gets executed infinitely: in rules/annotations/where_clause_prep2.rules (function mark_client_branch):

```
<action>topref = ref</action>
<action>level = topref.depth - expref.depth</action>
<while>topref.ancestor(level, prog.kw_where)
  <action>topref = topref.getAncestor(level, prog.kw_where)</action>
  <action>level = topref.depth - expref.depth</action>
</while>
```

The problem can be recreated using this code (checked in to testcases/uast/nested_can_find_mixing_temp_and_perm_tables.p):

```
def var res as logical.
def var id as int init 14.

def temp-table tt no-undo
  field txt as char.

res = CAN-FIND (FIRST book NO-LOCK
               WHERE book.book-id = id
               AND CAN-FIND (FIRST tt NO-LOCK
                             WHERE tt.txt = book.book-title)).
```

When run, the following message is printed to the console before the hang:

```
INFO: CAN-FIND-only reference to buffer p2j_test.book,p2j_test.book,-1 from TABLE (219043332157 [7@22])
```

#2 - 08/10/2017 12:48 PM - Eric Faulhaber

- File 3315_20170810a.patch added

I took a quick look at this and produced the attached fix to prevent the infinite loop. However, there is still a problem with the converted code for this test case, in that the HQL generated for the outer CAN-FIND is incorrect. The generated HQL is:

```
"book.bookId = ? and"
```

...but it should be:

```
"book.bookId = ?"
```

So, this will fail at runtime.

The KW_AND node associated with this where clause is being annotated incorrectly. This will require further investigation, but the attached patch should at least get the customer past the conversion hang.

#3 - 08/10/2017 12:50 PM - Eric Faulhaber

- Assignee set to Eric Faulhaber

- Status changed from New to WIP

TODO: add the fix from [#3315-2](#) to an existing development branch or create a branch for it.

#4 - 08/24/2017 11:49 AM - Eric Faulhaber

The patch from [#3315-2](#) was added to task branch 3293a, which was merged to FWD trunk revision 11160.

Jaroslav, at your convenience, please let us know if this patch fixes the infinite loop issue originally reported.

We still have to address the problem of incorrect HQL being generated.

Files

3315_20170810a.patch	1.24 KB	08/10/2017	Eric Faulhaber
----------------------	---------	------------	----------------