Database - Bug #2625

query-off-end does not work properly

07/27/2015 04:24 PM - Ovidiu Maxiniuc

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Ovidiu Maxiniuc		**% Done:**	100%
Category:			**Estimated time:**	0.00 hour
Target version:	Cleanup and Stablization for Server Features			
billable:	No		**case_num:**	
vendor_id:	GCD			

Description

Related issues:	
Related to Database - Bug #2514: incorrect return values for get{First|Next|L...	**Closed**

History

#1 - 07/27/2015 04:31 PM - Ovidiu Maxiniuc

Please check the testcase uast/query/off-end.p from testcases repository, extracted from the customer's server and simplified and anonymized. The problem can be describes as:

> in a dynamic query, the query-off-end attribute will return true/yes, even if there are other rows preselected

The commented-out static code proves that there are 3 of them.
The cause of this behavior seems to be the client-side sorting of the found rows (which will use the pointer to the current cursor and leaving it in off-end position instead pointing to 1st row as requested by the first get-next method).

The consequence of this bug is that some queries cannot be iterated (the attribute is used as a flag for leaving row looping) so the result is missing (for BPM/ETF, ex: entitySEARCHTestCases/RELATIONSHIP/APPLIC_PER.xml).

#2 - 07/27/2015 05:23 PM - Ovidiu Maxiniuc

I created the 2625a branch based on revno 10901 of trunk. The update contains the following fixes:

- fixed implementation of get(Next/Last/Prev/First/Current) methods of AbstractQuery.
- synchronized the cursor with first/next/prev/last methods in PresortCompoundQuery.

After patching the server, the testcase that lead to this issue is passing in ETF.

#3 - 07/28/2015 04:26 PM - Eric Faulhaber

- Assignee set to Ovidiu Maxiniuc

Code review 2625a/10901:

The changes look good to me, but I would like Stanislav to do a quick review as well, as he recently has made a lot of changes in this area for his browse work and may be sensitive to additional changes.

Stas: please take a quick look.

Ovidiu: please regression test and merge if it passes and Stanislav also is OK with the changes.

#4 - 07/29/2015 05:32 AM - Stanislav Lomany

I wrote a simple loop with PresortCompoundQuery. In this case the first record is missing.

```
RecordBuffer.openScope(tt2, tt);
tt.create();
tt.setF1(new integer(1));
tt.create();
tt.setF1(new integer(2));
tt.create();
tt.setF1(new integer(3));
tt2.create();
tt2.setF2(new integer(1));
tt2.create();
tt2.setF2(new integer(2));
tt2.create();
tt2.setF2(new integer(3));

forEach("loopLabel0", new Block()
{
   PresortCompoundQuery query0 = null;

   FieldReference byExpr0 = new FieldReference(tt, "f1");

   public void init()
   {
      query0 = new PresortCompoundQuery();
      query0.enableBreakGroups();
      query0.addComponent(new PreselectQuery(tt, (String) null, null, "tt.id asc"));
      query0.addComponent(new PreselectQuery(tt2, (String) null, null, "tt2.id asc"));
      query0.addSortCriterion(byExpr0);
   }

   public void body()
   {
      query0.next(true);
      message(concat(valueOf((integer) new FieldReference(tt, "f1").getValue()), " ", valueOf((integer) new Fi
eldReference(tt2, "f2").getValue()))));
   }
});
```

My overall concerns is that functions like PresortCompoundQuery.peekNext() do update cursor position. So calls like

```
Object[] row = peekNext();
cursor.next();
```

have to be carefully tested.

#5 - 07/29/2015 09:53 AM - Ovidiu Maxiniuc

Stanislav Lomany wrote:

> I wrote a simple loop with PresortCompoundQuery. In this case the first record is missing.
>
> [...]
>
> My overall concerns is that functions like PresortCompoundQuery.peekNext() do update cursor position. So calls like
> [...]
> have to be carefully tested.

Could you post/send me the original P4GL code?
Thanks.

#6 - 07/29/2015 09:56 AM - Stanislav Lomany

The version I've posted was modified manually to use two temporary tables instead of permanent and temporary (for convenience). The 4GL code was

```
def temp-table tt field f1 as integer.
def temp-table tt2 field f2 as integer.

create tt. tt.f1 = 1.
create tt. tt.f1 = 2.
create tt. tt.f1 = 3.

create tt2. tt2.f2 = 1.
create tt2. tt2.f2 = 2.
create tt2. tt2.f2 = 3.

for each book, each tt2 break by book.book-id:
   message string(book.book-id + " " + string(tt2.f2).
end.
```

#7 - 07/29/2015 12:32 PM - Ovidiu Maxiniuc

Stanislav,
thanks for spotting this regression. My testcases were not covering this case.

The missing row was caused because the cursor was not reset before sorting the result. The PresortCompoundQuery uses the Presorter to sort them. After this operation the cursor position is again altered and must be synchronized with the internal position of the sorted result.
The implementation is a little awkward because of these two indexes: an int (SimpleResults.position) and a double (Cursor.position). They must be synchronized on each pointer move operation.

Committed revision 2625a/10903.

#8 - 07/30/2015 10:52 AM - Ovidiu Maxiniuc

The revision 2625a/10903 passed the regression testing and results were saved to
/opt/secure/clients/timco/majic_test_results/2625a_10903_16cd2a2_20150730_om.zip.

#9 - 07/30/2015 12:51 PM - Eric Faulhaber

Code review 2625a/10903:

The change looks good.

Please note that the trunk has changed to rev. 10902 and there are some conflicts with this branch (AbstractQuery and Cursor). However, while the file header will need to be merged, the functionality changed does not appear to conflict. Please rebase and merge these files carefully, then test locally and merge into the trunk if all is OK.

Stas: do you have any other immediately pending updates which conflict with this branch?

#10 - 07/30/2015 12:54 PM - Stanislav Lomany

No.

#11 - 07/31/2015 01:45 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

The update was rebased to latest trunk and merged back as revision 10904 and notification email was sent to team members.

#12 - 07/31/2015 01:47 PM - Eric Faulhaber

- Status changed from WIP to Closed

- % Done changed from 0 to 100

#13 - 11/16/2016 12:06 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stablization for Server Features