

Database - Bug #3172

QUERY-PREPARE can be used on static queries (and the sorting can be dynamically changed)

08/19/2016 12:55 PM - Greg Shah

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

History

#1 - 08/19/2016 12:57 PM - Greg Shah

Please review the following articles:

<http://knowledgebase.progress.com/articles/Article/000032702>
https://community.progress.com/community_groups/openedge_general/f/26/t/12020
<http://knowledgebase.progress.com/articles/Article/P39990>
<http://stackoverflow.com/questions/18267870/static-vs-dynamic-queries-in-openedge>

Of interest is that static queries can have their WHERE clause and BY clause modified dynamically.

On a positive note, the buffer(s) can't be changed.

#3 - 06/23/2020 01:17 PM - Roger Borrello

This code generates a NPE on trunk-11347:

```
def temp-table tt1 field f1 as int index ix1 is unique f1.
def var h as handle.

def query q1 for tt1.
h = query q1:handle.
message "S1" h:num-buffers h:is-open h:dynamic.
open query q1 for each tt1 no-lock.
message "S2" h:num-buffers h:is-open h:dynamic.
h:query-prepare("for each tt1 no-lock indexed-reposition").
message "S3" h:num-buffers h:is-open h:dynamic. // 1 no no FWD abends

h:query-prepare("for each tt1 where f1 = '' no-lock indexed-reposition").
message "S4" h:num-buffers h:is-open h:dynamic. // 1 no no FWD abends

/*
We need some more tests to see how the static query q1 gets affected
if its query predicate is changed to something else, like adding a
WHERE clause, another buffer, etc.
*/

create query h.
h:add-buffer(buffer tt1:handle).
h:query-prepare("for each tt1 no-lock indexed-reposition").
message "D1" h:num-buffers h:is-open h:dynamic.
```

#4 - 06/23/2020 04:01 PM - Roger Borrello

Will there be different code-paths for a temp-table versus a standard table? I am walking through the testcase, trying to get familiar with the life-cycle of the predicate/buffers.

#5 - 06/23/2020 04:49 PM - Eric Faulhaber

Stack trace?

#6 - 06/23/2020 04:55 PM - Eric Faulhaber

Nevermind, got it.

```
Caused by: java.lang.NullPointerException
    at com.goldencode.p2j.persist.PreselectQuery.getTableCount(PreselectQuery.java:3515)
    at com.goldencode.p2j.persist.AbstractQuery.numBuffers(AbstractQuery.java:3036)
    at com.goldencode.p2j.persist.QueryWrapper.numBuffers(QueryWrapper.java:5456)
    at com.goldencode.testcases.DynStatQry.lambda$execute$0(DynStatQry.java:58)
    ...
```

#7 - 06/23/2020 06:04 PM - Roger Borrello

Eric Faulhaber wrote:

Nevermind, got it.

Are you looking at this? Did you see the notes in #4208-736 ?

#8 - 06/23/2020 06:09 PM - Eric Faulhaber

- *File 3172_ecf_20200623a.patch added*

Yes and no :-)

I am attaching a patch because I didn't have a branch to which to commit. The attached patch fixes the NPE, but it needs review by someone more familiar with the dynamic query implementation.

The core problem is that QueryWrapper is determined to be dynamic or static at construction, and any assignment of a delegate query of the opposite nature later does not update this state. This leads to incorrect handling of API calls that assume a static query when a dynamic one has been assigned, and presumably vice-versa. My fix updates the state of the dynamic flag to true when a dynamic query is assigned to the wrapper.

The fix is not orthogonal and maybe it should be, in the event a wrapper assigned dynamic at construction can be assigned a static delegate query later.

Constantin, please take it from here.

#9 - 06/23/2020 07:54 PM - Roger Borrello

Eric Faulhaber wrote:

Yes and no :-)

I am attaching a patch because I didn't have a branch to which to commit. The attached patch fixes the NPE, but it needs review by someone more familiar with the dynamic query implementation.

The core problem is that QueryWrapper is determined to be dynamic or static at construction, and any assignment of a delegate query of the opposite nature later does not update this state. This leads to incorrect handling of API calls that assume a static query when a dynamic one has been assigned, and presumably vice-versa. My fix updates the state of the dynamic flag to true when a dynamic query is assigned to the wrapper.

The fix is not orthogonal and maybe it should be, in the event a wrapper assigned dynamic at construction can be assigned a static delegate query later.

Constantin, please take it from here.

That patch alleviated the NPE, and we now result in:

- "Unable to PREPARE TEMP-TABLE buffer object. (9046)"
- "TEMP-TABLE must be PREPARED before referencing NAME/DEFAULT-BUFFER-HANDLE etc. (9040)"

#10 - 06/24/2020 01:33 AM - Eric Faulhaber

Roger Borrello wrote:

That patch alleviated the NPE, and we now result in:

- "Unable to PREPARE TEMP-TABLE buffer object. (9046)"
- "TEMP-TABLE must be PREPARED before referencing NAME/DEFAULT-BUFFER-HANDLE etc. (9040)"

Are you using a different test case? I didn't get those errors.

#11 - 06/24/2020 05:28 AM - Constantin Asofiei

Eric Faulhaber wrote:

Constantin, please take it from here.

OK, I'm working on it.

#12 - 06/24/2020 05:58 AM - Constantin Asofiei

Eric Faulhaber wrote:

I am attaching a patch because I didn't have a branch to which to commit. The attached patch fixes the NPE, but it needs review by someone more familiar with the dynamic query implementation.

We can't make a static query resource dynamic. The query resource is still reported as 'non-dynamic', even if its delegate was changed via query-prepare (and the delegate is dynamic).

... in the event a wrapper assigned dynamic at construction can be assigned a static delegate query later.

I don't think so. 4GL doesn't allow access to the underlying actual query (the delegate in FWD). Only the query resource can be moved from a handle to another, and all access is done through this resource.

I think my idea with the dynamicDelegate will work, but I need to check all the 'dynamic' behavior with a static query resource. And how the static query behaves once it was prepared with a dynamic delegate.

#13 - 06/24/2020 07:10 AM - Constantin Asofiei

Roger, the fix is in 3821c rev 11363.

Please test with the customer scenario.

#14 - 06/24/2020 08:40 AM - Roger Borrello

Constantin Asofiei wrote:

Roger, the fix is in 3821c rev 11363.

Please test with the customer scenario.

I'll let you know.

#15 - 06/24/2020 08:55 AM - Roger Borrello

Roger Borrello wrote:

Constantin Asofiei wrote:

Roger, the fix is in 3821c rev 11363.

Please test with the customer scenario.

I'll let you know.

The NPE is gone. I'll post to #4208 for customer details. I'll proceed with debugging that application.

Files

3172_ecf_20200623a.patch	410 Bytes	06/23/2020	Eric Faulhaber
--------------------------	-----------	------------	----------------