

Base Language - Bug #6384

dynamic query conversion issues

05/20/2022 01:50 PM - Constantin Asofiei

Status:	Test	Start date:	
Priority:	Normal	Due date:	
Assignee:	Ovidiu Maxiniuc	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#2 - 05/20/2022 02:00 PM - Constantin Asofiei

Ovidiu, this test-case which uses the before-table in a dynamic query abends with this:

```
Caused by: com.goldencode.p2j.persist.RuntimeJastInterpreter$InterpreterException: No REFERENCE named (bisomethingreallyblue)
    at com.goldencode.p2j.persist.RuntimeJastInterpreter.evalExpression(RuntimeJastInterpreter.java:1442)
    at com.goldencode.p2j.persist.RuntimeJastInterpreter.collectParameters(RuntimeJastInterpreter.java:851)
)
    at com.goldencode.p2j.persist.RuntimeJastInterpreter.callMethod(RuntimeJastInterpreter.java:944)
    at com.goldencode.p2j.persist.RuntimeJastInterpreter.execMethod(RuntimeJastInterpreter.java:659)
    at com.goldencode.p2j.persist.RuntimeJastInterpreter.interpret(RuntimeJastInterpreter.java:419)
    at com.goldencode.p2j.persist.DynamicQueryHelper.lambda$4(DynamicQueryHelper.java:559)
    at com.goldencode.p2j.persist.AbstractQuery.notifyQueryOpenListeners(AbstractQuery.java:957)
    at com.goldencode.p2j.persist.QueryWrapper.queryOpen(QueryWrapper.java:4304)
    at com.goldencode.testcases.Dynq.lambda$0(Dynq.java:49)
```

The test is this:

```
def temp-table somethingReallyBlue before-table BISomethingReallyBlue field c as char.
def dataset dsl for somethingReallyBlue .

create somethingReallyBlue.

somethingReallyBlue.c = "red".

temp-table somethingReallyBlue:tracking-changes = true.
somethingReallyBlue.c = "blue".

release somethingReallyBlue.

def var hq as handle.

create query hq.
hq:add-buffer(buffer bisomethingreallyblue:handle).
hq:query-prepare("for each BISomethingReallyBlue").
hq:query-open().
hq:get-first().
message BISomethingReallyBlue.c avail(BISomethingReallyBlue).
```

The difference is that the before-table buffer name is registered as bisomethingReallyBlue in RuntimeJastInterpreter.buffers, but bisomethingreallyblue is emitted during dynamic conversion.

The dynamic conversion and the interpreter assume that the source DMO var name will be what the runtime conversion will also emit in the JAST (case-by-case), but this is not correct in this case.

And also using this as key to identify the runtime buffer with the JAST buffer seems problematic to me. We may want to inject the legacy buffer name in the JAST and use that as key, because the legacy buffer name we know that is case-insensitive and is not affected by the conversion rules.

Please look at it and fix in 6129a.

#3 - 05/20/2022 02:23 PM - Greg Shah

- Assignee set to Ovidiu Maxiniuc
- Start date deleted (05/20/2022)

#4 - 05/20/2022 10:12 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

The issue seems familiar to me. I think I encountered something similar in the last year or so, but momentarily I do not have a clean solution.

I extracted the JAST at an incipient moment, and at least the TEMP_TABLE node for the FOR, it looks like this (see the force_dmo_alias annotation):

```
BISomethingReallyBlue [TEMP_TABLE]:8589934606 @1:33
  (schemaname=bisomethingreallyblue,
   bufname=bisomethingreallyblue,
   dbname=, recordtype=14,
   force_dmo_alias=bisomethingReallyBlue,
   bufrefkey=bisomethingreallyblue,bisomethingreallyblue,8589934609,
   bufreftype=21,
   uniquename=bisomethingreallyblue_bisomethingreallyblue,
   refid=8589934612)
```

I am a bit worried about altering TRPL code as it is shared with static conversion, too.

#5 - 05/21/2022 03:10 AM - Constantin Asofiei

- Subject changed from dynamic query for before-table uses wrong buffer name variable to dynamic query conversion issues

Ovidiu, thanks, force_dmo_alias placed me on the right track. This patch seems to have solved the issue:

```
### Eclipse Workspace Patch 1.0
#P p2j6129a
Index: src/com/goldencode/p2j/persist/DynamicConversionHelper.java
```

```

=====
--- src/com/goldencode/p2j/persist/DynamicConversionHelper.java    (revision 3599)
+++ src/com/goldencode/p2j/persist/DynamicConversionHelper.java    (working copy)
@@ -1180,11 +1180,14 @@
     while (iter.hasNext())
     {
         Aast chAst = iter.next();
-
-         if (chAst.getType() == ProgressParserTokenTypes.RECORD_PHRASE ||
-             chAst.getType() == ProgressParserTokenTypes.DEFINE_BUFFER)
+         int type = chAst.getType();
+         boolean isTempTable = type == ProgressParserTokenTypes.DEFINE_TEMP_TABLE ||
+             type == ProgressParserTokenTypes.DEFINE_WORK_TABLE;
+         if (type == ProgressParserTokenTypes.RECORD_PHRASE ||
+             type == ProgressParserTokenTypes.DEFINE_BUFFER ||
+             isTempTable)
        {
            Aast bufAst = (Aast) chAst.getFirstChild();
+            Aast bufAst = isTempTable ? chAst : (Aast) chAst.getFirstChild();
            String bufName = (String) bufAst.getAnnotation("bufname");
            if (bufName != null)
            {

```

Now I get another issue:

```

RuntimeJastInterpreter.evalExpression(Aast) line: 1442
RuntimeJastInterpreter.collectParameters(Aast, Class<?>[], int) line: 851
RuntimeJastInterpreter.callCtor(Aast) line: 776
RuntimeJastInterpreter.evalExpression(Aast) line: 1308
RuntimeJastInterpreter.doAssign(String, Aast) line: 706
RuntimeJastInterpreter.execMethod(Aast, Object...) line: 668
RuntimeJastInterpreter.interpret(String, Object...) line: 419
DynamicQueryHelper.parse(QueryProcessor, List<Buffer>, String, String, List<Buffer>) line: 542
DynamicQueryHelper.parseFindQuery(Buffer, String, LockType, String, List<Buffer>) line: 1032
$__Proxy456(BufferImpl)._find(String, BufferImpl$FindMode, LockType, Buffer...) line: 9463
TemporaryBuffer.lambda$copyAllRows$5(logical, BufferImpl, String, BufferImpl) line: 3598
1808433074.run() line: not available
ErrorManager.nestedSilent(Runnable, BiConsumer<String,Integer>) line: 695
ErrorManager.nestedSilent(Runnable) line: 656
TemporaryBuffer.copyAllRows(rowid, Temporary, Temporary, boolean, TemporaryBuffer$CopyTableMode, boolean, boolean, boolean) line: 3598
TempTableBuilder(AbstractTempTable).copyTempTable(handle, logical, logical, logical, character) line: 502
TempTableBuilder(AbstractTempTable).copyTempTable(handle, logical, logical, logical) line: 362
TempTableBuilder(AbstractTempTable).copyTempTable(handle, logical, logical) line: 342

```

I'll work on a recreate.

#6 - 05/22/2022 03:41 AM - Constantin Asofiei

The recreate for the issue in previous note is related to 'replace' mode in copy-temp-table:

```
def var htt as handle.
def var htt1 as handle.
def var hb as handle.

define temp-table tt1 no-undo
  field f1 as char
  index f1 is primary unique f1.
htt1 = temp-table tt1:handle.

create tt1.
tt1.f1 = "abc".

create temp-table htt.
htt:create-like(htt1).
htt:temp-table-prepare(htt1:name).
htt:copy-temp-table(htt1, false, true).

hb = htt:default-buffer-handle.
hb:find-first().

message hb::f1.
```

The source and destination table both have the same 4GL name, and this causes confusion during the dynamic conversion of the 4GL query, used by FWD in replace mode. I think we need to switch to FQL and not 4GL predicate for the 'find-first' emulation. In either case, using directly the FQL should be faster.

Note that this is not just for dynamic temp-tables, you may have two static temp-tables with the same name (but in different programs, so they are different 4GL temp-tables), and copy-temp-table will fail the same way.

#7 - 05/22/2022 02:24 PM - Ovidiu Maxiniuc

Constantin, please use r13909 to test the original issue. It makes both your isolated testcases work.

However, according to H2 / CA from 20190826 from DynamicConversionHelper:

"force_dmo_alias must be set only at the buffer used in the query, and not at the default buffer, when there is an explicit buffer used by the query."

This is the default buffer. I do not think this is valid any more (or I did not interpret the entry correctly). At least, I did some tests with some customer code and I did not encounter any issues.

#8 - 05/22/2022 02:25 PM - Ovidiu Maxiniuc

- *Status changed from WIP to Review*

- *% Done changed from 0 to 100*

#9 - 05/22/2022 11:43 PM - Ovidiu Maxiniuc

Constantin, please use r13911.

For some reasons, the conversion detected a collision between the buffer and ... itself and decided to rename one of them, in some cases. See #5034-2163 /2164 for additional information.

#10 - 05/23/2022 07:20 AM - Constantin Asofiei

Ovidiu, please copy the changes to 6129a.

#11 - 05/23/2022 11:37 AM - Ovidiu Maxiniuc

Done. Revision is 13866.

#12 - 06/01/2022 04:33 PM - Constantin Asofiei

- *Status changed from Review to Test*