

Database - Bug #7367

disable TriggerTracker for TemporaryBuffer

05/20/2023 09:40 AM - Constantin Asofiei

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

History

#1 - 05/20/2023 09:45 AM - Constantin Asofiei

A session trigger can't be registered for a temp-table (I also double-checked in OE). This means that RecordBuffer.triggerTracker should have no meaning for a record in a temp-table. I've tried disabling it in RecordBuffer.setCurrentRecord, via:

```
Index: src/com/goldencode/p2j/persist/RecordBuffer.java
=====
--- 7156a/src/com/goldencode/p2j/persist/RecordBuffer.java      (revision 4090)
+++ 7156a/src/com/goldencode/p2j/persist/RecordBuffer.java      (working copy)

@@ -11552,8 +11574,11 @@
     Record oldRecord = this.currentRecord;
     this.currentRecord = newCrtRecord;

-     TriggerTracker.releaseTracker(oldRecord);
-     this.triggerTracker = TriggerTracker.getTracker(this.currentRecord, dmoBufInterface, bufferManager);
+     if (!isTemporary())
+     {
+         TriggerTracker.releaseTracker(oldRecord);
+         this.triggerTracker = TriggerTracker.getTracker(this.currentRecord, dmoBufInterface, bufferManager);
+     }

     updateNoUndoState();

@@ -11574,8 +11599,11 @@
     this.undoable.checkUndoable(true);
 }

- TriggerTracker.releaseTracker(currentRecord);
- this.triggerTracker = TriggerTracker.getTracker(newCrtRecord, dmoBufInterface, bufferManager);
+ if (!isTemporary())
+ {
+     TriggerTracker.releaseTracker(currentRecord);
+     this.triggerTracker = TriggerTracker.getTracker(newCrtRecord, dmoBufInterface, bufferManager);
+ }

     this.currentRecord = newCrtRecord;
     this.recordChanged = false; // invalidate CURRENT-CHANGED flag (see above)
```

But this causes regressions.

Is there some state in TriggerTracker which is not related to invoking session or schema triggers?

#2 - 05/22/2023 05:01 AM - Alexandru Lungu

I recall seeing ASSIGN trigger checks for when we were executing copy-temp-table (fast or not). I don't know if they were in fact useless or not, but I know such checks are in place :)

I will take a look.

#3 - 05/22/2023 09:27 AM - Alexandru Lungu

In setCurrentRecord, there is the following condition (line 11422):

```
targetBuffer.getTriggerTracker() != null &&  
!targetBuffer.getTriggerTracker().isExecuting(DatabaseEventType.WRITE, currentRecord.primaryKey())
```

If this condition is true, doFlush = true; may execute.

- Before your change, this could happen (the trigger tracker is not null and no DatabaseEventType.WRITE is executing)
- After your change, this can't ever happen because the trigger tracker is always null

It seems to me that this should have been:

```
targetBuffer.getTriggerTracker() == null ||  
!targetBuffer.getTriggerTracker().isExecuting(DatabaseEventType.WRITE, currentRecord.primaryKey())
```

instead. Note that there are two such conditionals there.

The same goes for RecordNursery.makeVisible.

#4 - 05/22/2023 09:43 AM - Ovidiu Maxiniuc

Alex,

I do not think that is the good condition. If targetBuffer.getTriggerTracker() is null then the second operand of || is evaluated and a NPE will be thrown.

#5 - 05/22/2023 09:45 AM - Ovidiu Maxiniuc

Ovidiu Maxiniuc wrote:

Alex,

I do not think that is the good condition. If targetBuffer.getTriggerTracker() is null then the second operand of || is evaluated and a NPE will be thrown.

Sorry, you are right, I did not see you changed the != into ==, not only && into ||.

#6 - 05/22/2023 09:47 AM - Alexandru Lungu

Ovidiu Maxiniuc wrote:

Sorry, you are right, I did not see you changed the != into ==, not only && into ||.

Hmm, I edited my comment to use == null seconds after posting :) Sorry for the inconvenience

#7 - 05/22/2023 10:14 AM - Ovidiu Maxiniuc

Alexandru Lungu wrote:

Hmm, I edited my comment to use == null seconds after posting :)

That make sense. I reviewed the small piece of code from the email notification in Thunderbird. Only after clicking submit I noticed the ==.

Sorry for the inconvenience

No problem ☺☺. It happened to me as well.

Back to our code. The change looks fine, but the two conditionals must be adapted. The doFlush will be always true for temp-tables. Also armCurrentChanged(); probably should not be called since FIND CURRENT is meaningless in this case since (no external change since the _temp database is context private).

#8 - 05/23/2023 08:42 AM - Alexandru Lungu

Ovidiu Maxiniuc wrote:

Back to our code. The change looks fine, but the two conditionals must be adapted. The doFlush will be always true for temp-tables. Also armCurrentChanged(); probably should not be called since FIND CURRENT is meaningless in this case since (no external change since the _temp database is context private).

I see your point. However, I think triggerTracker is null only when currentRecord is null. In all of the previous changes, before reaching ltargetBuffer.getTriggerTracker().isExecuting, .isAvailable or .getRecord() == null is checked. This means that doFlush was true only if the currentRecord was set, which implicitly meant that the trigger tracker was set.

#9 - 05/25/2023 04:24 AM - Alexandru Lungu

Radu, can you do the changes from [#7367-1](#) and [#7367-3](#) over 7026d and do an in-depth testing of a large customer application? I want to know if we can integrate this easily and fast or there are still obvious regressions to handle.

#10 - 05/25/2023 04:25 AM - Radu Apetrii

Alexandru Lungu wrote:

Radu, can you do the changes from [#7367-1](#) and [#7367-3](#) over 7026d and do an in-depth testing of a large customer application? I want to know if we can integrate this easily and fast or there are still obvious regressions to handle.

On it. I will get back with updates as quickly as possible.

#11 - 05/25/2023 05:17 AM - Radu Apetrii

I've done some in-depth testing with a large customer application and I could not find any regression(s). Since the changes were already on my computer, I committed them to 7026d, rev. 14578.

#12 - 05/26/2023 03:20 AM - Alexandru Lungu

- Status changed from New to WIP
- % Done changed from 0 to 100

This reached 7156a. 7026d was rebased. Testing passed.

#13 - 05/26/2023 03:20 AM - Alexandru Lungu

- Status changed from WIP to Review

#14 - 05/29/2023 09:46 AM - Alexandru Lungu

- Status changed from Review to Test

Merged 7026d to trunk as rev. 14587.

#15 - 05/29/2023 10:31 AM - Ovidiu Maxiniuc

Alexandru Lungu wrote:

Merged 7026d to trunk as rev. 14587.

That makes sense. I attempted to do a review of the 7026d branch on Friday evening but it was nowhere to be found. :)

#16 - 07/10/2023 07:16 AM - Alexandru Lungu

- Assignee set to Alexandru Lungu

This can be closed.

#17 - 07/10/2023 07:16 AM - Greg Shah

- Status changed from Test to Closed