

Database - Bug #6485

tracking-changes can be set if a temp-table has at least a buffer attached to a dataset

06/03/2022 08:11 AM - Constantin Asofiei

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

History

#2 - 06/03/2022 08:21 AM - Constantin Asofiei

- Assignee set to Ovidiu Maxiniuc

There are some problems with tracking a record delete in the before-table:

1. TRACKING-CHANGES can be set to a temp-table if there is at least a buffer attached to a dataset. Currently, the DEFAULT-BUFFER is checked, which is incorrect - this needs to be fixed.
2. a delete on a buffer not attached to a dataset will create the before-table record, if the temp-table has TRACKING-CHANGES set to true
3. this TODO in TemporaryBuffer.delete() needs to be solved:

```
                // keep the reference to the original datasource-rowid
                // TODO: is this only for DATA-SOURCE? because for a simple TEMP-TABLE (in a DATASET), t
his
                //          is not set.
```

A testcase which can be used to check the above is this:

```
def temp-table tt1 before-table btt1 field f1 as int index ix1 is unique f1.
def temp-table tt2 before-table btt2 field f1 as int index ix1 is unique f1.
def buffer buftt2 for tt2.

def dataset ds1 for tt1.
def dataset ds2 for buftt2.
// def dataset ds3 for tt2.

do transaction:
create tt1.
tt1.f1 = 10.
release tt1.
end.

do transaction:
temp-table tt1:tracking-changes = true.
find first tt1.
delete tt1.
temp-table tt1:tracking-changes = false.
end.

def var hbtt2 as handle.
def var hdr1 as handle.

create buffer hbtt2 for table "tt2".
create data-source hdr1.
hdr1:add-source-buffer(hbtt2, ?).
buffer tt1:attach-data-source(hdr1).
```

```
find first btt1.  
message "btt1" btt1.f1.  
buffer-copy btt1 to tt2.  
find first tt2.  
message "tt2" tt2.f1.  
  
temp-table tt2:tracking-changes = true.  
message buffer btt1:save-row-changes().  
temp-table tt2:tracking-changes = false.  
  
find first btt2.  
message "btt2" btt2.f1.
```

But, some other tests are required to see if a dynamic buffer is attached to a dataset and after that deleted, does TRACKING-CHANGES remain set, if it was previously set?

The current patch is in 6129a/13915, but it needs a proper fix at least for TRACKING-CHANGES.

#3 - 06/03/2022 01:34 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

- % Done changed from 0 to 70

I have fixed the first part (r13917).

I tested the 2nd issue and it works as expected because you already removed the extra condition which caused it to fail.

I am investigating the last issue to see whether we should keep the reference to the original datasource-rowid.

#4 - 06/03/2022 03:52 PM - Ovidiu Maxiniuc

- % Done changed from 70 to 100

- Status changed from WIP to Review

The the reference to the original datasource-rowid definitively must be kept. I found some other issues related to datasets/datasources and fixed them.

Committed revision 13919.

#5 - 06/03/2022 04:56 PM - Constantin Asofiei

Ovidiu, 13919 regresses the POC tests, I think this change is at fault, from TemporaryBuffer:

```
    if (sourceRowid != null)
    {
        ((TempRecord) b4Buffer.getCurrentRecord())._datasourceRowid(sourceRowid);
    }
}
```

Please provide a test which shows that this is required. For now, I've disabled the test in 6129a/13921. Please don't enable it until I understand better why this is needed and how this affects the customer's app.

Either the fix is incorrect or the root cause for what disabling this solves is not found yet.

#6 - 06/03/2022 05:00 PM - Ovidiu Maxiniuc

- Status changed from Review to Test

Here is the testcase is used:

```
// output to "6485.out".

def temp-table src-tt1 field src-f1 as int index ix1 is unique src-f1.
create src-tt1. src-f1 = 10000.
create src-tt1. src-f1 = 10001.
create src-tt1. src-f1 = 10002.
create src-tt1. src-f1 = 10003.
release src-tt1.

define data-source dt-src for src-tt1.

def temp-table tt1 before-table btt1 field f1 as int index ix1 is unique f1.
def temp-table tt2 before-table btt2 field f1 as int index ix1 is unique f1.
def buffer buftt2 for tt2.

def dataset ds1 for tt1.

for each src-tt1:
    message buffer src-tt1:name rowid(src-tt1) src-tt1.src-f1 buffer src-tt1:ORIGIN-ROWID buffer src-tt1:DATA-
SOURCE-ROWID.
end.
message "refill:"
    buffer tt1:attach-data-source(data-source dt-src:handle, "src-f1,f1")
    buffer tt1:fill().
for each tt1:
    message buffer tt1:name rowid(tt1) tt1.f1 buffer tt1:ORIGIN-ROWID buffer tt1:DATA-SOURCE-ROWID.
end.

temp-table tt1:TRACKING-CHANGES = true.
def buffer buf-tt1 for tt1.
find last buf-tt1.
    message "Deleting" buffer buf-tt1:name rowid(buf-tt1) buf-tt1.f1 buffer buf-tt1:ORIGIN-ROWID buffer buf-tt1
:DATA-SOURCE-ROWID.
    delete buf-tt1.
find last buf-tt1.
    message "Altering" buffer buf-tt1:name rowid(buf-tt1) buf-tt1.f1 buffer buf-tt1:ORIGIN-ROWID buffer buf-tt1
:DATA-SOURCE-ROWID.
```

```

buf-ttl.f1 = -999.
release buf-ttl.
create buf-ttl.
message "Created" buffer buf-ttl:name rowid(buf-ttl) buf-ttl.f1 buffer buf-ttl:ORIGIN-ROWID buffer buf-ttl:
DATA-SOURCE-ROWID.
release buf-ttl.

for each bttl.
message buffer bttl:name bttl.f1 buffer bttl:ORIGIN-ROWID buffer bttl:DATA-SOURCE-ROWID.
end.

```

The expected output is:

```

Must run ATTACH-DATA-SOURCE on buffer src-ttl in order to get DATA-SOURCE-ROWID. (14276)
src-ttl 0x00000000000001900 10000 ? ?
Must run ATTACH-DATA-SOURCE on buffer src-ttl in order to get DATA-SOURCE-ROWID. (14276)
src-ttl 0x00000000000001901 10001 ? ?
Must run ATTACH-DATA-SOURCE on buffer src-ttl in order to get DATA-SOURCE-ROWID. (14276)
src-ttl 0x00000000000001902 10002 ? ?
Must run ATTACH-DATA-SOURCE on buffer src-ttl in order to get DATA-SOURCE-ROWID. (14276)
src-ttl 0x00000000000001903 10003 ? ?
refill: yes yes
ttl 0x00000000000003100 10000 ? 0x00000000000001900
ttl 0x00000000000003101 10001 ? 0x00000000000001901
ttl 0x00000000000003102 10002 ? 0x00000000000001902
ttl 0x00000000000003103 10003 ? 0x00000000000001903
Must run ATTACH-DATA-SOURCE on buffer buf-ttl in order to get DATA-SOURCE-ROWID. (14276)
Deleting buf-ttl 0x00000000000003103 10003 ? ?
Must run ATTACH-DATA-SOURCE on buffer buf-ttl in order to get DATA-SOURCE-ROWID. (14276)
Altering buf-ttl 0x00000000000003102 10002 ? ?
Must run ATTACH-DATA-SOURCE on buffer buf-ttl in order to get DATA-SOURCE-ROWID. (14276)
Created buf-ttl 0x00000000000003104 0 ? ?
bttl 10003 ? 0x00000000000001903
bttl 10002 ? 0x00000000000001902
bttl 0 ? ?

```

#7 - 06/03/2022 06:30 PM - Constantin Asofiei

Ovidiu, I think I have the recreate. See this:

```
def temp-table tt1 before-table btt1 field f1 as int index ix1 is unique f1.
def temp-table tt2 before-table btt2 field f1 as int index ix1 is unique f1.
def temp-table tt3 before-table btt3 field f1 as int index ix1 is unique f1.
def dataset ds1 for tt1.
def dataset ds2 for tt2.
def dataset ds3 for tt3.

create tt3.
tt3.f1 = 100.
def var hdr1 as handle.
create data-source hdr1.
hdr1:add-source-buffer(buffer tt3:handle, ?).
buffer tt1:attach-data-source(hdr1).
buffer tt1:fill().
find first tt1.
message "1 tt1" tt1.f1 buffer tt1:data-source-rowid. // has

// delete the record
do transaction:
  temp-table tt1:tracking-changes = true.
  find first tt1.
  delete tt1.
  temp-table tt1:tracking-changes = false.
end.

find first btt1.
message "2 btt1" btt1.f1 buffer btt1:data-source-rowid.

def var hdr2 as handle.
create data-source hdr2.
hdr2:add-source-buffer(buffer tt2:handle, ?).
buffer tt1:attach-data-source(hdr2).
message "3 btt1" btt1.f1 buffer btt1:data-source-rowid.
```

it shows in OE:

```
1 tt1 100 0x00000000000003900
2 btt1 100 0x00000000000003900
3 btt1 100 ?
```

In FWD, line 3 shows a valid rowid (with 6129a/13921 rev backed out locally). I think ATTACH-DATA-SOURCE will clear any DATA-SOURCE-ROWID at all the after-table and before-table records. Does this make sense?

#8 - 06/03/2022 06:38 PM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

In FWD, line 3 shows a valid rowid (with 6129a/13921 rev backed out locally). I think ATTACH-DATA-SOURCE will clear any DATA-SOURCE-ROWID at all the after-table and before-table records. Does this make sense?

It looks like this. As long as between the message 2 and 3 there is no other action! And even the attach-data-source is done on a different buffer. And this makes it harder to implement. On attach-data-source event, both before and after tables must be updated (it can be done in bulk, nonetheless) but also the records loaded in buffers - as this exact case.

#9 - 06/04/2022 09:31 AM - Constantin Asofiei

STDERR for the spawned client is set like this ClientBuilder.localStart:

```
pb.redirectError(ProcessBuilder.Redirect.INHERIT);
```

I don't know what this 'inherit' mode does, but no messages appear in any log, client or server.

CA: above was meant #6487

#10 - 06/06/2022 08:33 AM - Ovidiu Maxiniuc

Sorry, I do not understand your last note. Was it meant to be posted on other thread, maybe?

#11 - 06/06/2022 08:38 AM - Constantin Asofiei

Ovidiu Maxiniuc wrote:

Sorry, I do not understand your last note. Was it meant to be posted on other thread, maybe?

Yes, sorry.

#12 - 06/07/2022 07:08 AM - Constantin Asofiei

- Status changed from Test to WIP

- % Done changed from 100 to 90

This change in TemporaryBuffer.copyAllRows (which was added a comment in [#6482-13](#), 6129a/13930):

```
if (false && sourceRowid != null)
{
    // OM: this code is correct, but CA found some cases where it is not copied
    ((TempRecord) b4Buffer.getCurrentRecord())._datasourceRowid(sourceRowid);
}
```

is dependent on fixing the problem related to resetting the `_datasourcerowid` value, documented in [#6485-7](#). If I re-enable this test, then #6277 will fail, and this is not acceptable (at this time).

We can't re-enable this code until this is fixed, and 6129a can't be merged to 3821c until this is fixed.

#13 - 06/17/2022 02:23 PM - Constantin Asofiei

Ovidiu, please work on fixing [#6485-7](#) issue. I have a case where the disabled code in [#6485-12](#) needs to be backed out, but if I do this, then I go into the [#6485-7](#) issue... so this needs to be fixed.

#14 - 06/17/2022 02:29 PM - Constantin Asofiei

Constantin Asofiei wrote:

I have a case where the disabled code in [#6485-12](#) needs to be backed out,

I meant the code is correct and needs to be:

```
if (sourceRowid != null)
{
    ((TempRecord) b4Buffer.getCurrentRecord())._datasourceRowid(sourceRowid);
}
```

#15 - 06/17/2022 08:31 PM - Ovidiu Maxiniuc

Interesting. I added some more lines to that testcase. If the data-source is detached or replaced with another, the attribute returns ?. If later the original data-source is attached back, the attribute is kept!

It all resumed to check whether the current rowid value of the attribute is still valid for the buffer in the current dataset.

Committed revision 13944.

However, during my investigations, I encountered some other flaws. I will fix and commit in subsequent changesets.

#16 - 06/18/2022 02:21 AM - Constantin Asofiei

Ovidiu, thank you, it solves the issue I had in #6277.

#17 - 06/20/2022 06:18 PM - Ovidiu Maxiniuc

- *% Done changed from 90 to 100*

I added the complete implementation of DATA-SOURCE-ROWID(num/name) in revision 13947.

#18 - 07/06/2022 02:17 PM - Constantin Asofiei

- *Status changed from WIP to Test*