Database - Bug #3126

Incorrect compound query iteration order if a backing buffer is updated.

06/09/2016 08:51 AM - Stanislav Lomany

Status:	New	**Start date:**	06/09/2016
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 06/09/2016 08:54 AM - Stanislav Lomany

Testcase 1:

```
def temp-table tt1 field f1 as integer
                   field f2 as integer index idx1 f1.

def temp-table tt2 field f3 as integer
                   field f4 as integer index idx3 f3.

def var i as integer.
repeat i = 1 to 10:
   create tt1. tt1.f1 = i.
end.

create tt2. tt2.f3 = 2.
create tt2. tt2.f3 = 5.
create tt2. tt2.f3 = 6.
create tt2. tt2.f3 = 10.

def query q for tt1, tt2 scrolling.

open query q for each tt1, first tt2 where tt2.f3 = tt1.f1 outer-join.

DEFINE BROWSE brws QUERY q DISPLAY tt1.f1 tt2.f3 enable all
             WITH TITLE "Single Browse"
             SIZE 70 BY 7.
DEFINE FRAME MyFrame brws AT ROW 3 COLUMN 3 WITH NO-LABELS SIZE 80 BY 20.

ENABLE brws WITH FRAME MyFrame.

WAIT-FOR CLOSE OF CURRENT-WINDOW.
```

Reproduction: type "100", hold DOWN. You will see that the query loops infinitely.

#2 - 06/09/2016 08:56 AM - Greg Shah

- Subject changed from Incorrect compount query iteration order if a backing buffer is updated. to Incorrect compound query iteration order if a backing buffer is updated.

Can this be seen in the customer's application? Or is this just an issue you found in standalone testcases?

Does this have any UI implications or is it just a persistence issue?

#3 - 06/09/2016 09:03 AM - Stanislav Lomany

Testcase 2:

```
def temp-table tt1 field f1 as integer
                   field f2 as integer index idx1 f1.

def temp-table tt2 field f3 as integer
                   field f4 as integer index idx3 f3.

def var i as integer.
repeat i = 1 to 3:
   create tt1. tt1.f1 = i.
   create tt2. tt2.f3 = i.
end.

def query q for tt1, tt2 scrolling.

open query q for each tt1, each tt2 where tt2.f3 = tt1.f1.
get first q. /* 1 */
get next q.  /* 2 */
get next q.  /* 3 */

reposition q to row 1.
get next q.
tt1.f1 = 100.
message string(tt1.f1). /* 100 */

get next q.
message string(tt1.f1). /* 2 */
get next q.
message string(tt1.f1). /* 3 */

get next q.
message string(tt1.f1).  /* 100 */
get next q.
if avail(tt1) then message string(tt1.f1).
else message "N/A".  /* N/A */
```

4GL output: 100, 2, 3, 100, N/A.
P2J output: 100, 2, 3, 2, 2.

#4 - 06/09/2016 09:05 AM - Stanislav Lomany

> Can this be seen in the customer's application?

No. Found it accidentaly in standalone tescases.

> Does this have any UI implications or is it just a persistence issue?

Persistence only.

#5 - 06/09/2016 09:11 AM - Greg Shah

- Project changed from Liberty to Database