Database - Bug #12149

Multi-table INDEXED-REPOSITION: wrong position, ERROR-STATUS and GET PREV wrap-around after a reposition by ROWID

11/10/2026 08:13 AM - Alexandru Lungu

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Alexandru Lungu	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#1 - 11/10/2026 08:14 AM - Alexandru Lungu

- Assignee set to Alexandru Lungu

Test suite

tests/query/indexed_reposition/failed_reposition_matrix/sequences_multi in testcases (36 classes, 925 ABLUnit cells). Each cell drives a prior state on a multi-table INDEXED-REPOSITION query (a step script: walks, jumps, record changes), repositions by ROWID to a target and records a trace. The expected values are OE 11.6.3 output, and all 925 cells pass on OE. FWD fails 869.

Query shapes (all INDEXED-REPOSITION, person 912101..912102 joined to pers-addr on emp-num):

Shape	Query	Cells failing
B1	EACH person, EACH pers-addr (FWD server join)	268
B2	EACH pers-addr, EACH pers-addr self-join (FWD compound query)	52
B3	EACH person, FIRST pers-addr (FWD compound query)	203
B4	EACH person, LAST pers-addr (FWD compound query)	36
B6	EACH person, EACH pers-addr OUTER-JOIN	262
B8	B1 as a dynamic query (QUERY-PREPARE)	48

Targets: D8a = the leading rowid fails the WHERE, D8b = the inner rowid fails the join's WHERE (both failed repositions), T1 = the jump pair (the reposition succeeds).

Each trace has six independent blocks, each replaying prior state + reposition: O0 the reposition itself (ERROR-STATUS, NUM-RESULTS, CURRENT-RESULT-ROW, buffer contents), O1 GET NEXT x3, O2 GET PREV x3, O3 REPOSITION-TO-ROW(1) + GET NEXT, O4 GET FIRST walk, O5 GET LAST walk (GET PREV until no row). Buffers are shown as person|pers-addr, none = buffer empty.

Issues, by number of cells affected

Most failing cells show more than one issue: 852 + 681 + 480 + 24 + 16 + 8 is more than 869.

#	Issue	Cells	Cells green if only this were fixed
1	GET LAST walk wraps around to	852	45

	the last row		
2	Wrong position / counters after the reposition	681	1
3	One ERROR-STATUS message too many on a failed reposition	480	0 (133 together with 1)
4	GET FIRST walk returns a different result set	24	1
5	7332 raised where OE raises 7331 (self-join)	16	0
6	Deleting through the first buffer after a jump does not raise "no current row"	8	8

1. GET LAST walk wraps around (852 cells)

After the GET LAST walk reaches the first row, OE's next GET PREV finds no row; FWD's returns the **last** row of the query again. In 852 cells FWD's walk is exactly OE's walk plus that one repeated row. It happens after failed (D8a, D8b) **and** successful (T1) repositions, on every shape. Not yet checked: whether it also needs a reposition at all, or happens on any multi-table GET LAST + GET PREV walk.

Example TestSeqMultiJoinableCreateQualifyingThenPrev2#testB1SN3CqP2T1 (B1, prior state N3 Cq P2, successful reposition); only O5 differs:

```
OE : O5 last n=10 rows=912102|912102/3,912102|912102/2,912102|912102/1,912101|912101/8,...,912101|912101/1
FWD: O5 last n=11 rows=912102|912102/3,912102|912102/2,912102|912102/1,912101|912101/8,...,912101|912101/1,912
102|912102/3
```

2. Wrong position and counters after the reposition (681 cells)

Where the query continues after a multi-table reposition differs from OE:

- first GET NEXT (O1), where it differs: FWD restarts at the first row of the result set (296), finds no row where OE has one (17), returns another row (15), or returns a partial row (one buffer filled, the other none) (4)
- first GET PREV (O2), where it differs: FWD returns a row where OE finds none (253), a partial row (114), or a row at another place in the result set (53)
- NUM-RESULTS / CURRENT-RESULT-ROW differ along with the rows (GET NEXT 141 / 23, GET PREV 242 / 435, TO ROW + NEXT 92 / 13)
- the buffers right after the reposition (O0) differ in 233 cells, e.g. OE leaves both buffers empty where FWD keeps the leading row

Example TestSeqMultiEachFirstDeleteLastBufferThenPrev2#testB3SJN2DiP2D8b (B3, prior state J N2 Di P2, target D8b):

```
     OE                                          FWD
O0   err=no n=0 num=1 cur=? buf=none|none        err=no n=0 num=1 cur=? buf=912101|none
O2   prev1 num=1 cur=1 buf=none|912101/1         prev1 num=1 cur=0 buf=912101|912101/1
O5   last n=2 rows=912102|912102/2,912101|912101/1
                                                 last n=3 rows=912102|912102/2,912101|912101/1,912102|912102/2
```

Example TestSeqMultiJoinableDeleteFirstBufferThenPrev2#testB1SFN3FDcP2T1 (B1, successful reposition after a delete through the first buffer): FWD's GET NEXT first returns an empty pair and its GET FIRST walk is empty, where OE walks 3 rows:

```
     OE                                          FWD
O1   next1 num=5 cur=5 buf=912102|912102/1       next1 num=4 cur=5 buf=none|none
O4   first n=3 rows=912102|912102/1,...          first n=0 rows=
```

3. One ERROR-STATUS message too many on a failed reposition (480 cells)

The first message number is right, but FWD adds one message: OE 7331 with 1 message vs FWD 2 messages (326 cells), OE 7332 with 2 messages vs FWD 3 (154 cells). Mostly D8a (270) and D8b (165) on B1, B3, B4, B6, B8.

Example TestSeqMultiJoinableWalk#testB1SN3D8b:

```
OE : O0 err=yes#7332 n=2 num=0 cur=? buf=912101|none
FWD: O0 err=yes#7332 n=3 num=0 cur=? buf=912101|none
```

4. GET FIRST walk returns a different result set (24 cells)

After the reposition the GET FIRST walk (O4) misses rows (15), returns different rows (6) or one extra row (1), plus 2 GET LAST walks that differ beyond the wrap-around. Mostly cells with a delete (Dc, Di) in the prior state; see the second example under 2.

5. 7332 where OE raises 7331 (16 cells, self-join B2)

On the B2 self-join a failed reposition whose inner rowid fails (D8b) raises 7332 with 2 messages; OE raises 7331 with 1.

Example TestSeqMultiSelfJoinWalk#testB2SED8b:

```
OE : O0 err=yes#7331 n=1 num=0 cur=? buf=912101/4|none
FWD: O0 err=yes#7332 n=2 num=0 cur=? buf=912101/4|none
```

6. Delete after a jump does not raise "no current row" (8 cells)

The test<shape>DcNoCurrentRow / DoNoCurrentRow pins (B1, B2, B3, B4, B6, B8): after a successful jump (J) the first buffer is empty on OE, and deleting its row (FIND-CURRENT EXCLUSIVE-LOCK + BUFFER-DELETE on the query buffer, or through another buffer) gets "no current row". FWD still has a record there and deletes it. After a walk off the end (E) both agree.

```
TestSeqMultiJoinableDeleteFirstBuffer#testB1DcNoCurrentRow
OE : E Dc: no current row | J Dc: no current row
FWD: E Dc: no current row | J Dc: none
```

Suggested order

1. 1 (walk wrap-around): one behaviour, present in 852 of the 869 failing cells. Fixing it alone turns 45 cells green, and it removes the noise from nearly every other failing cell.
2. 3 and 5 (ERROR-STATUS of a failed multi-table reposition): with 1, another 133 cells.
3. 6 (empty first buffer after a jump): 8 pins, likely the same root as the O0 buffer differences of 2.
4. 2 and 4 (where the query continues after a multi-table reposition): the bulk of the remaining cells, the counterpart of #11922 for jumpMultiTable.

Acceptance

- all 925 cells of failed_reposition_matrix/sequences_multi pass on FWD (as they do on OE 11.6.3)
- no test that passes today fails afterwards in the rest of tests/query/indexed_reposition (including the single-table failed_reposition_matrix suites and the tests/concurrency/indexed_reposition* multi-session batteries) or in tests/query/db