Database - Bug #6834

RAQ navigation on records having NULL values on same index components

10/11/2022 08:48 AM - Greg Shah

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Dănuț Filimon	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	
Description			

History

#2 - 10/11/2022 08:50 AM - Greg Shah

This task is meant to properly handle the root cause of #6755.

Eric/Ovidiu: Please document the issue and solution ideas/problems here and then edit the title of this task.

#3 - 12/19/2022 01:32 PM - Ovidiu Maxiniuc

- Subject changed from deferred work from #6755 to RAQ navigation on records having NULL values on same index components

The issue is described with details and example in #6755. I will copy it here because that task contains sensitive data:

I tried to isolate the testcase with the following code:

```
DEFINE TEMP-TABLE tt6755
    FIELD f1 AS CHARACTER
    FIELD f3 AS INTEGER
    FIELD f4 AS CHARACTER
    INDEX primary-idx AS PRIMARY UNIQUE f1 f3.

CREATE tt6755. f1 = ?. f3 = 1. f4 = "Alpha".
CREATE tt6755. f1 = ?. f3 = 2. f4 = "Beta".
RELEASE tt6755.

FOR EACH tt6755:
    MESSAGE "S1:" f1 f3 f4.
END.

FIND FIRST tt6755.
    MESSAGE "S2:" f1 f3 f4.
FIND NEXT tt6755 NO-ERROR.
    MESSAGE "S2:" f1 f3 f4.
```

On OE the output is:

```
S1: ? 1 Alpha
S1: ? 2 Beta
S2: ? 1 Alpha
S2: ? 2 Beta
```

but on FWD the last FIND fails:

```
S1: ? 1 Alpha
S1: ? 2 Beta
S2: ? 1 Alpha
** No tt6755 record is available. (91)
** No tt6755 record is available. (91)
** No tt6755 record is available. (91)
S2: ? ? ?
```

Note the loop which shows all the records of the table. Initially I tried to make it this way, but the code works because the AdaptiveQuery will request the whole list of records. When navigating one at a time, the RandomAccessQuery which is also used in the converted customer code fails because of the reasons described in the above note.

When we navigate FIRST/NEXT we use a prebuilt set of queries which are meant to advance from a record to another based on the same order the query's index would do. So we fetch the first record. The next record is assumed to be the one with first n-1 index component identical, and the last component strictly the next greater. If none exist we backtrack to n-2 index component.

This algorithm works fine when using 4GL semantics, or when there are no NULL values in SQL. If NULL values are encountered on same position - as we have in our example -, SQL will be unable to guess the next record because the NULL values are not comparable. They NULL s are not equals to keep f1 fixed and iterate in f3 and it is not greater than other NULL s. As result, the SQL will not return any record and FWD assumes the OFF-END has been reached, so Beta is not printed in S2 RAQ navigation mode.

The immediate solution would be to enhance the queries with 4GL semantics of NULL-equality tests. This means each component of the navigation query (they map to index components) will have to be OR-ed with <field> is null. In this case, augmented for semantics of 4GL, like this:

current term	augmented term
tt6755.__if1 = upper(?1)	(tt6755.__if1 = upper(?1) or ?1 is null and tt6755.__if1 is null)
tt6755.__if1 > upper(?1)	(tt6755.__if1 > upper(?1) or ?1 is not null and tt6755.__if1 is null)

Since the primary-idx has two components, similar enhancing will have to be added for f3, too. This will cause performance issues because of the OR operator which will mess the SQL planner so it will not be able to correctly identify and use the expected index or probably any index at all. Alternatively, we can split the query in two but this means the number of SQL queries needed to advancement will increase exponentially. Both cases are not acceptable from the point of view of database access performance.

The original issue was worked around by slightly adjusting the customer code so that unknown / NULL values were avoided. But this is not always possible and the issue itself is difficult to be observed so this issue should be fixed in FWD.

#4 - 07/10/2023 07:18 AM - Alexandru Lungu

- Assignee set to Dănuț Filimon

- Status changed from New to WIP