Database - Bug #2420

Fully reproduce QUERY-OFF-END behavior

10/15/2014 04:08 PM - Stanislav Lomany

Status:	New	**Start date:**	10/15/2014
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 10/15/2014 04:19 PM - Stanislav Lomany

QUERY-OFF-END may return quirky values which doesn't always represent "logical" off-end state.
See the following testcases:

```
uast/adaptive_scrolling/adaptive-scrolling10-temp.p
uast/adaptive_scrolling/adaptive-scrolling10b-temp.p
uast/adaptive_scrolling/adaptive-scrolling10c-temp.p
```

As far as I've checked the rules are following:
off-end is "yes" if the position is < -0.5 of > cacheSize - 1 + 0.5 (note that strict operators are used), except the following case:
off-end is "no" if we have the full cache and the current off-end state is "no".
However I have two concerns:
1. Some code in browse widget and queries use _isOffEnd() and isOffEnd() functions to detect if we have reached the last result (i.e. position is < 0 of > cacheSize - 1). My suggestion is to make _isOffEnd() return "useful" off-end and isOffEnd() - 4GL value.
2. There is the case for which I cannot find a logical explanation (and I don't know how to implement it):

```
def temp-table tt field f1 as integer
          index idx1 f1.
def query q for tt scrolling.

create tt. tt.f1 = 1.
create tt. tt.f1 = 2.
create tt. tt.f1 = 3.

open query q for each tt. get first q. get prev q.
reposition q to row 0.
put stream s string(query-off-end("q")) " " string(current-result-row("q")) skip. /* yes */

open query q for each tt. get first q. get prev q.
reposition q to row 1.
put stream s string(query-off-end("q")) " " string(current-result-row("q")) skip. /* yes */

open query q for each tt. get first q. get prev q.
reposition q to row 2.
put stream s string(query-off-end("q")) " " string(current-result-row("q")) skip. /* yes */

open query q for each tt. get first q. get prev q.
reposition q to row 3.
put stream s string(query-off-end("q")) " " string(current-result-row("q")) skip. /* no */

open query q for each tt. get first q. get prev q.
reposition q to row 4.
put stream s string(query-off-end("q")) " " string(current-result-row("q")) skip. /* no */

open query q for each tt. get first q. get prev q.
reposition q to row 5.
```

```
put stream s string(query-off-end("q")) " " string(current-result-row("q")) skip. /* yes */

open query q for each tt. get first q. get prev q.
reposition q to row 100.
put stream s string(query-off-end("q")) " " string(current-result-row("q")) skip. /* yes */
```