

## Database - Bug #7737

### QueryWrapper should either show or throw errors depending on the query type

08/22/2023 07:51 AM - Dănuț Filimon

<b>Status:</b>	WIP	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Dănuț Filimon	<b>% Done:</b>	50%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No	<b>version:</b>	
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to Database - Bug #7496: finish support for query:forward-only attribute			<b>WIP</b>

### History

#### #1 - 08/22/2023 08:14 AM - Dănuț Filimon

This test fails for **7496a/rev.14703** and **trunk/rev.14704**.

```
DEFINE TEMP-TABLE tmp1 NO-UNDO FIELD f1 AS INTEGER f2 AS CHARACTER INDEX idx1 IS UNIQUE f1.
CREATE tmp1. ASSIGN tmp1.f1 = 1 tmp1.f2 = 'one'.
CREATE tmp1. ASSIGN tmp1.f1 = 2 tmp1.f2 = 'two'.
CREATE tmp1. ASSIGN tmp1.f1 = 3 tmp1.f2 = 'three'.
RELEASE tmp1.
```

```
DEFINE QUERY q FOR tmp1.
QUERY q:FORWARD-ONLY = TRUE.
OPEN QUERY q FOR EACH tmp1.
```

```
GET FIRST q.
DO WHILE AVAILABLE(tmp1):
    MESSAGE tmp1.f1 tmp1.f2.
    GET NEXT q.
END.
GET LAST q.
MESSAGE tmp1.f1 tmp1.f2.
```

```
CLOSE QUERY q.
```

#### 4GL Output:

```
1 one
2 two
3 three
Cannot LAST or PREV Query q when FORWARD-ONLY or -noautoreslist or BREAK BY enabled. (12393)
```

#### 7496a/rev.14703 and trunk/rev.14704 output:

```
1 one
2 two
3 three
Cannot LAST or PREV Query q when FORWARD-ONLY or -noautoreslist or BREAK BY enabled. (12393)
** No tmp1 record is available. (91)
** No tmp1 record is available. (91)
```

? ?

Error 12393 is thrown when GET LAST is executed, the No record available error shows up when trying to display the values from an empty buffer. On second thought, there should be no problem with the cursor invalidation (the issue name can be changed into something more descriptive).

## #2 - 08/22/2023 08:19 AM - Dănuț Filimon

- Related to Bug #7496: finish support for query:forward-only attribute added

## #3 - 08/22/2023 08:23 AM - Dănuț Filimon

Another simpler test:

```
DEFINE TEMP-TABLE tmp1 NO-UNDO FIELD f1 AS INTEGER f2 AS CHARACTER INDEX idx1 IS UNIQUE f1.  
CREATE tmp1. ASSIGN tmp1.f1 = 1 tmp1.f2 = 'one'.  
CREATE tmp1. ASSIGN tmp1.f1 = 2 tmp1.f2 = 'two'.  
CREATE tmp1. ASSIGN tmp1.f1 = 3 tmp1.f2 = 'three'.  
// RELEASE tmp1. // <- buffer is not released anymore like in the first example  
  
DEFINE QUERY q FOR tmp1.  
QUERY q:FORWARD-ONLY = TRUE.  
OPEN QUERY q FOR EACH tmp1.  
  
GET LAST q.  
MESSAGE tmp1.f1 tmp1.f2.  
  
CLOSE QUERY q.
```

4GL output:

```
Cannot LAST or PREV Query q when FORWARD-ONLY or -noautoreslist or BREAK BY enabled. (12393)
```

7496a/rev.14703 and trunk/rev.14704 output:

```
Cannot LAST or PREV Query q when FORWARD-ONLY or -noautoreslist or BREAK BY enabled. (12393)  
3 three
```

If the buffer is released after the create, we get a No record available error instead of 3 three.

#### #4 - 08/22/2023 08:37 AM - Alexandru Lungu

Is there a problem in changing recordOrShowError into recordOrThrowError?

Danut, do you have an example in which doing this change results in other regressions? If yes, can you post an example where recordOrThrowError is not right?

#### #5 - 08/22/2023 08:54 AM - Dănuț Filimon

Alexandru Lungu wrote:

Is there a problem in changing recordOrShowError into recordOrThrowError?

Danut, do you have an example in which doing this change results in other regressions? If yes, can you post an example where recordOrThrowError is not right?

Here's an example where the execution is not interrupted by the exception:

```
DEFINE TEMP-TABLE tmp1 NO-UNDO FIELD f1 AS INTEGER.  
DEFINE VARIABLE hq AS HANDLE NO-UNDO.
```

```
CREATE tmp1. ASSIGN tmp1.f1 = 1.  
RELEASE tmp1.
```

```
DEFINE QUERY q FOR tmp1 SCROLLING.  
hq = QUERY q:HANDLE.
```

```
hq:FORWARD-ONLY = TRUE.  
hq:QUERY-PREPARE("FOR EACH tmp1").  
hq:QUERY-OPEN().  
hq:GET-LAST().  
MESSAGE tmp1.f1.  
hq:QUERY-CLOSE().
```

```
MESSAGE "test".
```

```
DELETE OBJECT hq.
```

**#6 - 08/22/2023 09:08 AM - Alexandru Lungu**

- Assignee set to Dănuț Filimon

- Status changed from New to WIP

Is the fact that in the second example the query is scrolling? Or maybe because the second query is dynamic? Or maybe because the table has another structure (does the index or uniqueness matter)?

**#7 - 08/22/2023 09:11 AM - Alexandru Lungu**

Danut, can you attempt making the query forward-only **after** you prepare it, but **before** opening? Is there any difference?

**#8 - 08/22/2023 09:19 AM - Dănuț Filimon**

Alexandru Lungu wrote:

Is the fact that in the second example the query is scrolling?

It doesn't matter if the query is scrolling.

Or maybe because the table has another structure (does the index or uniqueness matter)?

Structure/index/uniqueness does not matter, I tried to bring the example in [#7722-3](#) to use the same structure but I get the same expected results from both.

Or maybe because the second query is dynamic?

This should be the case.

Danut, can you attempt making the query forward-only after you prepare it, but before opening? Is there any difference?

Yes, the dynamic query FORWARD-ONLY flag is set before the query is prepared, but this also doesn't change the results when it is done after it.

This might be strictly a difference between the types of queries used in both examples.

**#9 - 08/22/2023 09:34 AM - Alexandru Lungu**

- Subject changed from *GET-LAST* or *GET-PREV* should invalidate Cursor in a *FORWARD-ONLY* query to *QueryWrapper* should either show or throw errors depending on the query type

I see. It is weird that we don't have any similar use-case implemented in FWD: throw error if static, just show error if dynamic. We need to be 100% sure that we aren't misled by something else here.

Can you extend your testing for other kind of errors thrown by *QueryWrapper*? For instance:

- can you check with other statements like *get-first*, *get-next*, *get-prev* and *forward-only*?
- can you check *get-next*, *get-prev*, *get-last* and *get-first* with a closed query that is either static or dynamic?
- can you attempt reposition on a closed query that is either static or dynamic?
- use *GET-BUFFER-HANDLE*, *SET-BUFFERS*, *INDEX-INFO*, *SKIP-DELETED-RECORD* on either static or dynamic?
- can you use the *ON ERROR*, *LEAVE* clause for a wrapping block? Are the either static or dynamic queries honoring *ON ERROR*?

If this is a general issue we have in FWD, then we can go with a refactoring of errors in *QueryWrapper* to honor this finding.

**#10 - 08/23/2023 07:07 AM - Dănuț Filimon**

Alexandru Lungu wrote:

Can you extend your testing for other kind of errors thrown by *QueryWrapper*? For instance:

- can you check with other statements like *get-first*, *get-next*, *get-prev* and *forward-only*?
- can you check *get-next*, *get-prev*, *get-last* and *get-first* with a closed query that is either static or dynamic?
- can you attempt reposition on a closed query that is either static or dynamic?
- use *GET-BUFFER-HANDLE*, *SET-BUFFERS*, *INDEX-INFO*, *SKIP-DELETED-RECORD* on either static or dynamic?
- can you use the *ON ERROR*, *LEAVE* clause for a wrapping block? Are the either static or dynamic queries honoring *ON ERROR*?

I've tested these scenarios and found the following:

- *GET-BUFFER-HANDLE* will not be interrupted if a wrong parameter value is given, the same applies for *INDEX-INFO*.
- I don't know how *SET-BUFFERS* should be tested, I did it together with *GET-BUFFER-HANDLE* where I tried to run the function with and without any buffers set. *GET-BUFFER-HANDLE* didn't throw any errors for a *HANDLE* with no buffers, it just printed ? for any given value (int/buffer name).
- When using *ON ERROR UNDO*, *LEAVE* with both static and dynamic queries the messages after the block are displayed. In the case of static queries, messages that are in the same *DO ON ERROR UNDO*, *LEAVE* block are not displayed after the error is thrown.
- Trying to access records after closing the a dynamic query doesn't interrupt the execution, while for static queries it is. Repositioning in a query acts in the same manner.
- Finally, I found a test which doesn't interrupt for both queries. It's a simple example where you run *GET-NEXT* 3 times on a temporary table with 2 records (scrolling/forward-only/preselect are not required to reproduce). Both tests display No record available is displayed and any message before closing the query.

**#11 - 08/23/2023 07:13 AM - Alexandru Lungu**

- % Done changed from 0 to 50

Danut, please fix the cases where errors behave different for static / dynamic queries. That is, change QueryWrapper places where recordOrShow is used to honor dynamic.

**#12 - 08/23/2023 08:42 AM - Dănuț Filimon**

- % Done changed from 50 to 0

**Committed 7496a/rev.14704.** Fixed 4 errors in QueryWrapper.

I've created a test suite to test how errors are handled in 4GL and after fixing the errors I am still left with some problems (unrelated, but we should keep a record on them):

- Cannot Get on query q which is not opened. (3159) should be showed, but Cannot run GET methods on query q until it is opened. (7313) is shown instead. I suspect that this is because QueryWrapper.canGet() is more widely used than QueryWrapper.canGetStmt(). Even if both implementations look the same, the methods are used in different contexts.
- Preselect queries allow records to be retrieved even if the query is FORWARD-ONLY. (e.g you can run GET-NEXT 2 times and then GET-FIRST, GET-NEXT, no error should be thrown and records should be displayed - preselect query, forward-only = true).
- Instances where errors are shown and logged (error can be written to a file) and in some instances are not.
- Expected an error when repositioning on a closed query (4GL), but got a No record available error (FWD).

**#13 - 08/23/2023 08:54 AM - Dănuț Filimon**

- % Done changed from 0 to 50