

## Database - Bug #5673

### Additional support for INDEXED REPOSITION open query attribute

09/16/2021 06:44 AM - Alexandru Lungu

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<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>			

#### History

##### #1 - 09/16/2021 06:54 AM - Alexandru Lungu

The INDEXED REPOSITION 4GL optimization allows query reposition to records which were not initially in the database (prior query opening). This option is silently ignored if the query is preselecting (OPEN QUERY PRESELECT) or an explicit sort clause is specified ([BREAK] BY). Also, this keyword works only for scrolling queries. Refer to testcases/uast/indexed-reposition for some testcases.

Currently, the implementation of the INDEXED REPOSITION doesn't suppress the optimization if an explicit sort clause was used. In order to fix this, the sort clause should be identified. This can be done at conversion time to either suppress the generation of setIndexedReposition or add an "explicitSorted" flag. Otherwise, the run-time may be able to detect if such clause was used.

For preselection, setIndexedReposition should be ignored. However, there are no strong testcases yet for PreselectCompoundQuery which may prove that INDEXED REPOSITION is used.

Finally, the INDEXED REPOSITION keyword seems like no optimization, especially in FWD where there is extra overhead when repositioning. Maybe further testcases and experiments can indicate some means of optimizing INDEXED REPOSITION.

##### #3 - 07/04/2022 05:25 PM - Eric Faulhaber

Alexandru Lungu wrote:

The INDEXED REPOSITION 4GL optimization allows query reposition to records which were not initially in the database (prior query opening).

It seems, based on that description, that INDEXED REPOSITION is not necessarily only an optimization, but that it actually changes the behavior of a query. Have you tested which types of repositions (e.g., to a particular RECID/ROWID, by result set offset, etc.) are made to behave differently by this option?

**#4 - 07/18/2022 12:28 PM - Alexandru Lungu**

The issue which required support for INDEXED-REPOSITION was using a REPOSITION-TO-ROWID. I can't recall if this was particular to ROWID or general (to RECID, FORWARDS, BACKWARDS, etc.). Anyways, this should be rather related to the query itself. Thus, it would make sense if the behavior is independent upon the reposition keyword.