

Database - Bug #6579

QUERY-PREPARE validation

07/05/2022 08:20 AM - Marian Edu

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

History

#1 - 07/05/2022 08:26 AM - Marian Edu

In 4GL the query string is validated in query-prepare method and if anything wrong with it an error is being thrown at that point, in FWD this seems not to be the case and invalid query would lead to an 'persistence' exception being thrown when the query-open method is executed.

In <large_customer_application> 'dynamic query view' (ALT+Q) using a query string like for each wfn-op no-lock where wfn-op.zknaam will cause the following exception and lead to a crash:

```
[07/05/2022 14:32:03 EEST] (com.goldencode.p2j.persist.Persistence:WARNING) [0000000D:00000032:bogus-->local/m
tc/primary] error executing query [select wfnOp__1 from WfnOp__Impl__ as wfnOp__1 where (upper(rtrim(wfnOp__1.
zknaam))) order by upper(rtrim(wfnOp__1.kode)) asc, upper(rtrim(wfnOp__1.zknaam)) asc, wfnOp__1.srtfak asc, u
pper(rtrim(wfnOp__1.faknr)) asc, upper(rtrim(wfnOp__1.fakdagb)) asc, wfnOp__1.jaar asc]
[07/05/2022 14:32:03 EEST] (com.goldencode.p2j.persist.Persistence:SEVERE) [0000000D:00000032:bogus-->local/mt
c/primary] error executing query [select wfnOp__1 from WfnOp__Impl__ as wfnOp__1 where (upper(rtrim(wfnOp__1.z
knaam))) order by upper(rtrim(wfnOp__1.kode)) asc, upper(rtrim(wfnOp__1.zknaam)) asc, wfnOp__1.srtfak asc, up
per(rtrim(wfnOp__1.faknr)) asc, upper(rtrim(wfnOp__1.fakdagb)) asc, wfnOp__1.jaar asc]
com.goldencode.p2j.persist.PersistenceException: Error while processing the SQL list
Caused by: org.postgresql.util.PSQLException: ERROR: argument of WHERE must be type boolean, not type text
```

#3 - 07/05/2022 08:34 AM - Greg Shah

- Project changed from Base Language to Database

#4 - 07/06/2022 05:29 PM - Ovidiu Maxiniuc

The QUERY-PREPARE will process the 4GL code and create a complete in-memory class including a FQL for the 4GL code. At any error during this complex processing, it will stop with error and try to emit a similar message/code as 4GL. Apparently this particular case slips through the conversion undetected.

Marian,

Please send me an email with the necessary steps for reproduction of the above error so I can understand better what is wrong.

Thank you!

#5 - 07/07/2022 02:22 AM - Marian Edu

Ovidiu Maxiniuc wrote:

The QUERY-PREPARE will process the 4GL code and create a complete in-memory class including a FQL for the 4GL code. At any error during this complex processing, it will stop with error and try to emit a similar message/code as 4GL. Apparently this particular case slips through the conversion undetected.

Marian,

Please send me an email with the necessary steps for reproduction of the above error so I can understand better what is wrong.

Thank you!

Ovidiu, did crafted a quick 4GL test that shows that - you're right, some validation is being done there, it looks like at times is even more strict than the one in 4GL.

Problem seems to be with non-logical field used in where without any operator/value. It does look like you already know the data type of the field at that point cause there is a validation of the data type, this is a case where no value is provided. Otherwise when a string value is being used on a non character field 4GL seems to be OK with it, no error is being thrown by prepare/open/get (the later will not find anything though).

The test in testcases project - query/static/methods/query_prepare/test_query_prepare_where.p.