Bugs - Bug #2626

exotic WHERE predicate fails to convert

07/28/2015 04:49 PM - Ovidiu Maxiniuc

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Cleanup and Stablization for Server Features		
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

History

#1 - 07/28/2015 04:59 PM - Ovidiu Maxiniuc

The following testcase is based on an actual testcase found in the customer's test harness.

```
DEFINE TEMP-TABLE tt1
      FIELD d1 AS DATE INIT TODAY
      FIELD d2 AS DATE INIT TODAY
      FIELD l3 AS logical INIT no.

CREATE tt1.
CREATE tt1.

FOR EACH tt1
   WHERE tt1.d1 <= ?
     AND (tt1.d2 >= ? OR  tt1.d2 = ?)
     AND (tt1.l3 = TRUE OR  tt1.l3 = ?):
   message 'something!'.
END.
```

The predicate is built dynamically at runtime. The message 'something!' should not be printed at all because the 1st term of the predicate is always false (tt1.d1 <= ?).
At this moment the above query is converted in p2j as:

```
new AdaptiveQuery(tt1, "(tt1.d2 is null or tt1.d2 is null) and (tt1.l3 = true or tt1.l3 is null)", null, "tt1.
id asc");
```

Evidently, this is incorrect. The references to d1 are completely dropped. Optimally, the conversion processing should replace the whole predicate with false.

#2 - 07/29/2015 05:13 PM - Ovidiu Maxiniuc

Fixed normalization of expressions containing ? (unknown) literals.

Task branch 2626a is at revision 10902. It was forked from latest trunk revision 10901.

#3 - 07/30/2015 03:18 PM - Ovidiu Maxiniuc

The 2626a/10902 passed conversion testing on devsrv01 (identical generated sources). Runtime testing is not necessary, the affected file is used at runtime only on dynamic queries and majic does not use them.

#4 - 07/30/2015 04:17 PM - Eric Faulhaber

Code review 2626a/10902:

The changes look OK on the surface and they apparently fix this test case, but they seem to contradict the changes made explicitly to fix comparisons with unknown value back in 2009. Specifically, I'm looking at the changes around the old comment:

```
replace "literal <= unknown" or with "true"
```

Besides regression testing, did you find any broader range of test cases to check against in the testcases project?

Stas: you made these changes last time around (2009-01-23). Do you have a set of test cases you used to confirm comparison with unknown value behavior?

#5 - 07/30/2015 04:39 PM - Ovidiu Maxiniuc

Here is a test I used:

```
DEFINE VARIABLE k AS DATE INIT ?.
DEFINE VARIABLE t AS DATE INIT TODAY.

IF k > ? THEN MESSAGE "k > ?".
IF t > ? THEN MESSAGE "t > ?".
IF k >= ? THEN MESSAGE "k >= ?".
IF t >= ? THEN MESSAGE "t >= ?".
IF k < ? THEN MESSAGE "k < ?".
IF t < ? THEN MESSAGE "t < ?".
IF k <= ? THEN MESSAGE "k <= ?".
IF t <= ? THEN MESSAGE "t <= ?".
```

It will print in message area just k >= ? and k <= ?, the only cases when we have equality. From my experience there shouldn't be any difference with where predicates.

#6 - 07/30/2015 04:53 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> From my experience there shouldn't be any difference with where predicates.

Generally true, I think, but please write a simple test case with these same 8 expressions as where clauses to confirm.

> Runtime testing is not necessary, the affected file is used at runtime only on dynamic queries and majic does not use them.

I don't understand this statement. This rule set is used for static conversion, not just at runtime for dynamic queries. If it changed any HQL in the converted code, runtime regression testing is necessary.

#7 - 07/30/2015 04:59 PM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Ovidiu Maxiniuc wrote:
>
> > From my experience there shouldn't be any difference with where predicates.
>
> Generally true, I think, but please write a simple test case with these same 8 expressions as where clauses to confirm.

OK, I will do it.

> > Runtime testing is not necessary, the affected file is used at runtime only on dynamic queries and majic does not use them.
>
> I don't understand this statement. This rule set is used for static conversion, not just at runtime for dynamic queries. If it changed any HQL in the converted code, runtime regression testing is necessary.

The generated code resulted from conversion test is unchanged.

#8 - 07/30/2015 05:02 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

> The generated code resulted from conversion test is unchanged.

Sorry, I should have read the previous sentence more carefully:

> ...(identical generated sources)

#9 - 07/30/2015 05:52 PM - Stanislav Lomany

> Stas: you made these changes last time around (2009-01-23). Do you have a set of test cases you used to confirm comparison with unknown value behavior?

I don't remember what I used. unknown_handling.p and unknown_postprocess.p testcases may help.

#10 - 07/31/2015 10:15 AM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Ovidiu Maxiniuc wrote:
>
> > From my experience there shouldn't be any difference with where predicates.
>
> Generally true, I think, but please write a simple test case with these same 8 expressions as where clauses to confirm.

Here is a more complete table of relational operators in a where clause having ? (unknown) literal as operand:

```
DEFINE TEMP-TABLE tt0
   FIELD fk AS INT INIT 99 /* field with 'known' value */
   FIELD fu AS INT INIT ?. /* field with unknown value */

CREATE tt0. /* create at least one record to be found in case predicate is true */

/* compare unknown with literals (evaluable at conversion time) */
/* FIND FIRST tt0 WHERE ?  >= ?  NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "?  >= ?".  -- not compilable,
err: 223) */
```

```
   FIND FIRST tt0 WHERE 99 =   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "99 =   ?".
   FIND FIRST tt0 WHERE ?  =  10 NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  =  10".
   FIND FIRST tt0 WHERE 99 <>  ? NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "99 <>  ?".
   FIND FIRST tt0 WHERE ?  <> 10 NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "?  <> 10".
   FIND FIRST tt0 WHERE 99 >   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "99 >   ?".
   FIND FIRST tt0 WHERE ?  >  10 NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  >  10".
   FIND FIRST tt0 WHERE 99 >=  ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "99 >=  ?".
   FIND FIRST tt0 WHERE ?  >= 10 NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  >= 10".
   FIND FIRST tt0 WHERE 99 <   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "99 <   ?".
   FIND FIRST tt0 WHERE ?  <  10 NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  <  10".
   FIND FIRST tt0 WHERE 99 <=  ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "99 <=  ?".
   FIND FIRST tt0 WHERE ?  <= 10 NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  <= 10".

/* compare unknown with fields/expression (value only available at runtime) */
   FIND FIRST tt0 WHERE fk =   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fk =   ?".
   FIND FIRST tt0 WHERE fu =   ? NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "fu =   ?".
   FIND FIRST tt0 WHERE fk <>  ? NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "fk <>  ?".
   FIND FIRST tt0 WHERE fu <>  ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fu <>  ?".
   FIND FIRST tt0 WHERE fk >   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fk >   ?".
   FIND FIRST tt0 WHERE fu >   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fu >   ?".
   FIND FIRST tt0 WHERE fk >=  ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fk >=  ?".
   FIND FIRST tt0 WHERE fu >=  ? NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "fu >=  ?".
   FIND FIRST tt0 WHERE fk <   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fk <   ?".
   FIND FIRST tt0 WHERE fu <   ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fu <   ?".
   FIND FIRST tt0 WHERE fk <=  ? NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "fk <=  ?".
   FIND FIRST tt0 WHERE fu <=  ? NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "fu <=  ?".

   FIND FIRST tt0 WHERE ?  =  fk NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  =  fk".
   FIND FIRST tt0 WHERE ?  =  fu NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "?  =  fu".
   FIND FIRST tt0 WHERE ?  <> fk NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "?  <> fk".
   FIND FIRST tt0 WHERE ?  <> fu NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  <> fu".
   FIND FIRST tt0 WHERE ?  >  fk NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  >  fk".
   FIND FIRST tt0 WHERE ?  >  fu NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  >  fu".
   FIND FIRST tt0 WHERE ?  >= fk NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  >= fk".
   FIND FIRST tt0 WHERE ?  >= fu NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "?  >= fu".
   FIND FIRST tt0 WHERE ?  <  fk NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  <  fk".
   FIND FIRST tt0 WHERE ?  <  fu NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  <  fu".
   FIND FIRST tt0 WHERE ?  <= fk NO-ERROR.  IF     AVAILABLE(tt0) THEN MESSAGE "?  <= fk".
   FIND FIRST tt0 WHERE ?  <= fu NO-ERROR.  IF NOT AVAILABLE(tt0) THEN MESSAGE "?  <= fu".

MESSAGE "All done.".
```

All that P4GL outputs is the last message. Look for negation on availability of the tt0 buffer to see which predicates evaluates to true.

#11 - 07/31/2015 01:36 PM - Eric Faulhaber

Thanks, Ovidiu. If this test converts properly and behaves correctly at runtime with your fix, please merge your branch into the trunk.

#12 - 07/31/2015 04:35 PM - Ovidiu Maxiniuc

When converting the above testcase the code was not compilable. I chose to improve the normalization with for those cases. With this change, the testcase finished without any errors.
The 2626a is now at revision 10903.

#13 - 08/06/2015 08:32 AM - Eric Faulhaber

I'd like to review 2626a, so we can get it pushed to the trunk. Did you rebase it? I'm getting a mess when I try to update it.

#14 - 08/06/2015 08:46 AM - Eric Faulhaber

I guess it was rebased from trunk rev. 10907. The update looks good. Please rebase to the latest and then merge to the trunk.

#15 - 08/07/2015 07:19 AM - Ovidiu Maxiniuc

Passed conversion regression testing on devsrv01 (no changes detected in generated code) and conversion and testing on the customer's app server.
Committed revision 10916 and notification email sent to team members.

#16 - 08/07/2015 11:12 AM - Eric Faulhaber

- Status changed from New to Closed

- % Done changed from 0 to 100

#17 - 11/16/2016 12:06 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stablization for Server Features