Database - Bug #7113

Problem with nested CAN-FIND conversion

02/12/2023 04:18 AM - Igor Skornyakov

Status:	Review	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 02/12/2023 04:25 AM - Igor Skornyakov

Consider the following statement:

```
FOR EACH Customer WHERE CAN-FIND(
        FIRST Order WHERE Order.CustNum = Customer.CustNum 
                    AND CAN-FIND(
            FIRST OrderLine WHERE OrderLine.OrderNum = Order.OrderNum
                              AND CAN-FIND(
                FIRST Item WHERE Item.ItemNum = OrderLine.ItemNum
            )
        )
    ):
    MESSAGE Customer.CustNum.
END. 
```

The query is converted to (for MariaDB, formatted):

```
select customer.recid from Customer__Impl__ as customer 
where (
   exists(
      from Order___Impl__ as order__1 
      where checkError_BB(initError_B(false),  (order__1.custNum = customer.custNum or order__1.custNum is nul
l and customer.custNum is null) 
                 and exists(
                        from OrderLine__Impl__ as orderLine_1 
                        where checkError_BB(initError_B(false), orderLine_1.orderNum is null 
                                   and exists(
                                       from Item__Impl__ as item_1 where item_1.itemNum is null
                                    )
                              )
            )
        )
   )
)
order by customer.custNum asc
```

which doesn't look correct.

In addition at runtime we get an exception:

```
Caused by: java.util.NoSuchElementException
        at java.util.ArrayDeque.removeFirst(ArrayDeque.java:285)
        at java.util.ArrayDeque.pop(ArrayDeque.java:522)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter$2.ascent(FqlToSqlConverter.java:2195)
        at com.goldencode.ast.AnnotatedAst$1.notifyListenerLevelChanged(AnnotatedAst.java:2734)
        at com.goldencode.ast.AnnotatedAst$1.hasNext(AnnotatedAst.java:2643)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateExpression(FqlToSqlConverter.java:2535)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateWhere(FqlToSqlConverter.java:2088)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processSelect(FqlToSqlConverter.java:1418)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processStatement(FqlToSqlConverter.java:912)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateExpression(FqlToSqlConverter.java:2525)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateWhere(FqlToSqlConverter.java:2088)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processSelect(FqlToSqlConverter.java:1418)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processStatement(FqlToSqlConverter.java:912)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.toSQL(FqlToSqlConverter.java:599)
        at com.goldencode.p2j.persist.orm.Query.lambda$createSqlQuery$0(Query.java:355)
        at com.goldencode.p2j.jmx.NanoTimer.timer(NanoTimer.java:129)
        at com.goldencode.p2j.persist.orm.Query.createSqlQuery(Query.java:370)
        at com.goldencode.p2j.persist.orm.Query.list(Query.java:298)
        at com.goldencode.p2j.persist.Persistence.list(Persistence.java:1512)
        at com.goldencode.p2j.persist.ProgressiveResults.getResults(ProgressiveResults.java:1112)
        at com.goldencode.p2j.persist.ProgressiveResults.moveTo(ProgressiveResults.java:943)
        at com.goldencode.p2j.persist.ProgressiveResults.moveTo(ProgressiveResults.java:801)
        at com.goldencode.p2j.persist.ProgressiveResults.next(ProgressiveResults.java:415)
        at com.goldencode.p2j.persist.ResultsAdapter.next(ResultsAdapter.java:161)
        at com.goldencode.p2j.persist.AdaptiveQuery.next(AdaptiveQuery.java:1676)
        at com.goldencode.p2j.persist.PreselectQuery.next(PreselectQuery.java:2621)
```

Tested with 6129c/14808.

#2 - 02/13/2023 03:02 AM - Igor Skornyakov

The problem is at the program conversion (not only runtime).
The query from the previous noye is converted to

```
            query0.initialize(customer, "exists(from Order_ as order_ where order_.custNum = customer.custNum
and exists(from OrderLine as orderLine where orderLine.orderNum = ? and exists(from Item as item where item.it
emNum = ?)))", null, "customer.custNum asc", new Object[]
            {
               (P2JQuery.Parameter) () -> order_.getOrderNum(),
               (P2JQuery.Parameter) () -> orderLine.getItemNum()
            });
            query0.addExternalBuffers(item, orderLine, order_);
```

#3 - 02/13/2023 04:16 AM - Constantin Asofiei

- Start date deleted (02/12/2023)

What's the problem with the query? What would be the correct variant? If you change the FQL manually to the correct version, does this solve the runtime issue?

#4 - 02/13/2023 04:23 AM - Igor Skornyakov

Constantin Asofiei wrote:

> What's the problem with the query? What would be the correct variant? If you change the FQL manually to the correct version, does this solve the runtime issue?

I understand that tested exists would not contain parameters - like the first one.I have not tryied to change it manually since I was working on a different task.
Should I start analyzing this issue?
Thank you.

#5 - 02/13/2023 04:25 AM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
>> What's the problem with the query? What would be the correct variant? If you change the FQL manually to the correct version, does this solve the runtime issue?
>
> I understand that tested exists would not contain parameters - like the first one.I have not tryied to change it manually since I was working on a different task.
> Should I start analyzing this issue?
> Thank you.

So instead of ? it should be Order.OrderNum and OrderLine.ItemNum, respectively? If yes, just do a quick test just to check if the issue is with conversion only.

#6 - 02/13/2023 04:52 AM - Igor Skornyakov

Constantin Asofiei wrote:

> So instead of ? it should be Order.OrderNum and OrderLine.ItemNum, respectively? If yes, just do a quick test just to check if the issue is with conversion only.

After manual change the query to

```
query0.initialize(customer, "exists(from Order_ as order_ where order_.custNum = customer.custNum and exists(f
rom OrderLine as orderLine where orderLine.orderNum = order_.orderNum and exists(from Item as item where item.
itemNum = orderLine.itemNum)))", null, "customer.custNum asc");
```

The FQL looks correct:

```
select customer.recid from Customer__Impl__ as customer
where (exists(
    from Order___Impl__ as order__1
    where checkError_BB(initError_B(false), (order__1.custNum = customer.custNum or order__1.custNum is null a
nd customer.custNum is null)
     and exists(
         from OrderLine__Impl__ as orderLine_1
         where checkError_BB(initError_B(false), (orderLine_1.orderNum = order__1.orderNum or orderLine_1.orde
rNum is null and order__1.orderNum is null)
           and exists(
               from Item__Impl__ as item_1
               where (item_1.itemNum = orderLine_1.itemNum or item_1.itemNum is null and orderLine_1.itemNum i
s null)))))))
order by customer.custNum asc
```

However, I still get an exception:

```
Caused by: java.util.NoSuchElementException
        at java.util.ArrayDeque.removeFirst(ArrayDeque.java:285)
        at java.util.ArrayDeque.pop(ArrayDeque.java:522)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter$2.ascent(FqlToSqlConverter.java:2196)
        at com.goldencode.ast.AnnotatedAst$1.notifyListenerLevelChanged(AnnotatedAst.java:2734)
        at com.goldencode.ast.AnnotatedAst$1.hasNext(AnnotatedAst.java:2643)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateExpression(FqlToSqlConverter.java:2536)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateWhere(FqlToSqlConverter.java:2089)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processSelect(FqlToSqlConverter.java:1419)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processStatement(FqlToSqlConverter.java:913)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateExpression(FqlToSqlConverter.java:2526)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.generateWhere(FqlToSqlConverter.java:2089)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processSelect(FqlToSqlConverter.java:1419)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.processStatement(FqlToSqlConverter.java:913)
        at com.goldencode.p2j.persist.orm.FqlToSqlConverter.toSQL(FqlToSqlConverter.java:600)
```

Should I start working on this?
Thank you.

#7 - 02/13/2023 04:54 AM - Constantin Asofiei

One more thing: is this a regression in 6129c compared to trunk?

#8 - 02/13/2023 05:02 AM - Igor Skornyakov

Constantin Asofiei wrote:

> One more thing: is this a regression in 6129c compared to trunk?

No, I get the same exception with the trunk

#9 - 02/13/2023 05:03 AM - Igor Skornyakov

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
> > One more thing: is this a regression in 6129c compared to trunk?
>
> No, I get the same exception with the trunk

The conversion is also broken with the trunk.

#10 - 02/13/2023 05:19 AM - Constantin Asofiei

Thank you for checking, you can continue with your normal tasks.

#11 - 02/13/2023 09:13 AM - Igor Skornyakov

- Status changed from New to WIP

#12 - 02/13/2023 05:20 PM - Ovidiu Maxiniuc

The nested CAN-FIND is an old issue. There were some fixes in time (see #3040, #3465, #3771, and others) and the issue seemed fixed lately.

The (manual) fix in #7113-6 is correct. This is what the conversion should have generated.

The problem which causes the NoSuchElementException is an unbalanced access to aliases stack. In processSelect there is a pair: aliases.push(topLevelAliasMap); and then down if (aliases.pop() != topLevelAliasMap) which should do that and also validate if the process was correct. However, there is another aliases.pop(); in generateExpression(), as the stack shows. I think that is not correct (any more).

After removing it the query seems to be almost correct. There are some extra RPARENS ()) added at the end of the WHERE predicate. It seems to be that they come from the "emulate the ASCENT" block: some nodes are re-walked multiple times causing these tokens to be added to sb output.

#13 - 02/14/2023 01:22 AM - Igor Skornyakov

Ovidiu Maxiniuc wrote:

> The nested CAN-FIND is an old issue. There were some fixes in time (see #3040, #3465, #3771, and others) and the issue seemed fixed lately.
>
> The (manual) fix in #7113-6 is correct. This is what the conversion should have generated.
>
> The problem which causes the NoSuchElementException is an unbalanced access to aliases stack. In processSelect there is a pair: aliases.push(topLevelAliasMap); and then down if (aliases.pop() != topLevelAliasMap) which should do that and also validate if the process was correct. However, there is another aliases.pop(); in generateExpression(), as the stack shows. I think that is not correct (any more).
>
> After removing it the query seems to be almost correct. There are some extra RPARENS ()) added at the end of the WHERE predicate. It seems to be that they come from the "emulate the ASCENT" block: some nodes are re-walked multiple times causing these tokens to be added to sb output.

Thank you, Ovidiu!

#14 - 02/20/2023 08:40 AM - Igor Skornyakov

- File can-find.diff added

Fixed nested CAN-FIND conversion (see attached .diff).
Please review.
Thank you.

Working on runtime issue (#7113-6).

#15 - 02/20/2023 08:52 AM - Igor Skornyakov

Igor Skornyakov wrote:

> Fixed nested CAN-FIND conversion (see attached .diff).
> Please review.
> Thank you.
>
> Working on runtime issue (#7113-6).

Sorry, the converted code is still not correct.

#16 - 02/20/2023 09:48 AM - Igor Skornyakov

- File can-find.diff added

Igor Skornyakov wrote:

> Igor Skornyakov wrote:
>
> > Fixed nested CAN-FIND conversion (see attached .diff).
> > Please review.
> > Thank you.
> >
> > Working on runtime issue (#7113-6).
>
> Sorry, the converted code is still not correct.

The conversion looks OK now.
Please review.
Thank you.

#17 - 02/20/2023 10:55 AM - Igor Skornyakov

Created task branch 7113a.

#18 - 02/20/2023 11:20 AM - Igor Skornyakov

Committed fix for the nested CAN-FIND to 7113a/14485.

#19 - 02/20/2023 02:28 PM - Igor Skornyakov

The only remaining issue is that generated SQL statement contains extra closing brackets) at the end of the WHERE clause.
Working on it.

#20 - 02/21/2023 12:26 PM - Igor Skornyakov

- Status changed from WIP to Review

Fixed SQL generation.
Committed to 7113a/14896.

Please review.
Thank you.

#21 - 04/28/2023 08:34 PM - Ovidiu Maxiniuc

- Assignee set to Igor Skornyakov

Igor,
I have fixed the extra closing parenthesis as part of another task. Please do not put additional effort in that.

I had a look at 7113a/14896, but I did not do a full review. What I did instead was I put the testcase from note-1 to test, without your patch. It converted as you described and, with my runtime patch, it executed without errors. I do not see anything wrong with the conversion.

I recommend to put the status of this task to "duplicate".

#22 - 04/29/2023 04:02 AM - Igor Skornyakov

Ovidiu Maxiniuc wrote:

> Igor,
> I have fixed the extra closing parenthesis as part of another task. Please do not put additional effort in that.
>
> I had a look at 7113a/14896, but I did not do a full review. What I did instead was I put the testcase from note-1 to test, without your patch. It converted as you described and, with my runtime patch, it executed without errors. I do not see anything wrong with the conversion.
>
> I recommend to put the status of this task to "duplicate".

Ovidiu,
Sorry, but I do not understand. The conversion w/o my patch was incorrect and, AFAIR, you agreed with this. The fact that runtime changes makes it run w/o errors doesn't make it **logically** correct. Please double check your fix.

I stopped working on this two months ago waiting for a code review.

BTW: in what branch have you made your changes?
Thank you.

#23 - 05/02/2023 01:26 PM - Ovidiu Maxiniuc

> Sorry, but I do not understand. The conversion w/o my patch was incorrect and, AFAIR, you agreed with this. The fact that runtime changes makes it run w/o errors doesn't make it **logically** correct. Please double check your fix.

I am sorry. I read the task in FFD mode, being focused on the runtime issue and skipped the lines about conversion. Indeed, the current conversion is incorrect (I can confirm with my local test) and must be fixed. In my test I analysed the work done by fql2sql converter. The FQL expression it received was syntactically correct although - as you rightfully said - not correct from the POV of business logic.

> I stopped working on this two months ago waiting for a code review.

OK, I will do the review for it as I have finish my other urgent tasks.

> BTW: in what branch have you made your changes?

The branch is 7259a.

#25 - 11/29/2023 11:10 AM - Eric Faulhaber

- Assignee changed from Igor Skornyakov to Ovidiu Maxiniuc

Ovidiu, 7259a was merged to trunk some time ago. Did it fix this issue? If not, are the changes in 7113a needed?

#26 - 11/29/2023 11:56 AM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Ovidiu, 7259a was merged to trunk some time ago. Did it fix this issue?

7259a contained changes in FQL processor, which happen at runtime, in a later stage of query processing. I remember it fixed that issue.

> If not, are the changes in 7113a needed?

I looked at 7113a now (for the first time, apparently). Beside the runtime issue I notified Igor to avoid work duplication in FqlToSqlConverter because I fixed in 7259a, there are some changes at conversion (where_clause_prep.rules) and FqlPreprocessor level (about UDFs ?). I will analyse the testcase from first note with current FWD to understand what Igor was looking at. There is a short discussion with Constantin about it in notes 2-6.

Files

can-find.diff	4.9 KB	02/20/2023	Igor Skornyakov
can-find.diff	5.41 KB	02/20/2023	Igor Skornyakov