Database - Bug #5800

Conversion issue with CAN-FIND

11/02/2021 11:13 AM - Igor Skornyakov

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

Description

History

#1 - 11/02/2021 11:14 AM - Igor Skornyakov

The conversion of the following program

```
def var ts1 as datetime no-undo.
def var ts2 as datetime no-undo.

FIND FIRST udftests
WHERE udftests.test-name = '_minusts2'
  AND CAN-FIND(LAST udftests where udftests.fdatetime = ts1) = CAN-FIND(LAST udftests where udftests.fdatetime
 = ts2)
```

fails:

```
    [javac] /media/ias/ssd1T/_gcdc/fwd/testcases/src/com/goldencode/testcases/udfs/T1.java:42: error: no suita
ble constructor found for FindQuery(Buf,String,<null>,String,logical,LockType)
    [javac]             (P2JQuery.Parameter) () -> isEqual(new FindQuery(udftests, "udftests.fdatetime = ?", n
ull, "udftests.id asc", isEqual(udftests.getFdatetime(), ts1), LockType.NONE).hasAny(), new FindQuery(udftests
, "udftests.fdatetime = ?", null, "udftests.id asc", isEqual(udftests.getFdatetime(), ts2), LockType.NONE).has
Any()),
    [javac]                                                     ^
    [javac]     constructor FindQuery.FindQuery(DataModelObject,String,Supplier<logical>,String,DataModelObjec
t,Object[]) is not applicable
    [javac]       (argument mismatch; logical cannot be converted to DataModelObject)
    [javac]     constructor FindQuery.FindQuery(DataModelObject,String,Supplier<logical>,String,DataModelObjec
t,LockType) is not applicable
    [javac]       (argument mismatch; logical cannot be converted to DataModelObject)
    [javac]     constructor FindQuery.FindQuery(DataModelObject,String,Supplier<logical>,String,Object[],LockT
ype) is not applicable
    [javac]       (argument mismatch; logical cannot be converted to Object[])
    [javac] /media/ias/ssd1T/_gcdc/fwd/testcases/src/com/goldencode/testcases/udfs/T1.java:42: error: no suita
ble constructor found for FindQuery(Buf,String,<null>,String,logical,LockType)
    [javac]             (P2JQuery.Parameter) () -> isEqual(new FindQuery(udftests, "udftests.fdatetime = ?", n
ull, "udftests.id asc", isEqual(udftests.getFdatetime(), ts1), LockType.NONE).hasAny(), new FindQuery(udftests
, "udftests.fdatetime = ?", null, "udftests.id asc", isEqual(udftests.getFdatetime(), ts2), LockType.NONE).has
Any()),
    [javac]                                                                                                 ^
    [javac]     constructor FindQuery.FindQuery(DataModelObject,String,Supplier<logical>,String,DataModelObjec
t,Object[]) is not applicable
    [javac]       (argument mismatch; logical cannot be converted to DataModelObject)
    [javac]     constructor FindQuery.FindQuery(DataModelObject,String,Supplier<logical>,String,DataModelObjec
t,LockType) is not applicable
    [javac]       (argument mismatch; logical cannot be converted to DataModelObject)
    [javac]     constructor FindQuery.FindQuery(DataModelObject,String,Supplier<logical>,String,Object[],LockT
ype) is not applicable
    [javac]       (argument mismatch; logical cannot be converted to Object[])
    [javac] Note: Some messages have been simplified; recompile with -Xdiags:verbose to get full output
```

```
    [javac] 2 errors
```

4GL accepts the program as correct.

The converted code is:

```
public class T1
{
   Udftests.Buf udftests = RecordBuffer.define(Udftests.Buf.class, "fwd", "udftests", "udftests");

   @LegacySignature(type = Type.VARIABLE, name = "ts1")
   datetime ts1 = TypeFactory.datetime();

   @LegacySignature(type = Type.VARIABLE, name = "ts2")
   datetime ts2 = TypeFactory.datetime();
   /**
    * External procedure (converted to Java from the 4GL source code
    * in udfs/t1.p).
    */
   @LegacySignature(type = Type.MAIN, name = "udfs/t1.p")
   public void execute()
   {
      externalProcedure(T1.this, new Block((Body) () ->
      {
         RecordBuffer.openScope(udftests);
         new FindQuery(udftests, "upper(udftests.testName) = '_MINUSTS2' and ?exists(from Udftests as udftests
 where udftests.fdatetime = ?)exists(from Udftests as udftests where udftests.fdatetime = ?)", null, "udftests
.id asc", new Object[]
         {
            (P2JQuery.Parameter) () -> isEqual(new FindQuery(udftests, "udftests.fdatetime = ?", null, "udftes
ts.id asc", isEqual(udftests.getFdatetime(), ts1), LockType.NONE).hasAny(), new FindQuery(udftests, "udftests.
fdatetime = ?", null, "udftests.id asc", isEqual(udftests.getFdatetime(), ts2), LockType.NONE).hasAny()),
            ts1,
            ts2
         }).addExternalBuffers(udftests, udftests).first();
      }));
   }
}
```

#2 - 11/02/2021 11:57 AM - Eric Faulhaber

- Start date deleted (11/02/2021)

The conversion has gone quite wrong. I don't think we've come across a case before which compares the results of two CAN-FINDs in a WHERE clause. The FindQuery should look something like this:

```
new FindQuery(udftests, "upper(udftests.testName) = '_MINUSTS2' and exists(from Udftests as udftests where udf
tests.fdatetime = ?) = exists(from Udftests as udftests where udftests.fdatetime = ?)", null, "udftests.id asc
", new Object[]
{
   ts1,
   ts2
}).first();
```

#3 - 11/02/2021 12:11 PM - Igor Skornyakov

Eric Faulhaber wrote:

> The conversion has gone quite wrong. I don't think we've come across a case before which compares the results of two CAN-FINDs in a WHERE clause. The FindQuery should look something like this:
>
> [...]

Actually, if we compare the result of CAN-FIND with the value of a logical variable, we get similar error. Comparison with the value of a logical **field** is converted correctly.