

Database - Bug #2219

problematic dynamic query test cases

01/07/2014 12:22 PM - Eric Faulhaber

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

History

#1 - 01/07/2014 12:24 PM - Eric Faulhaber

Originally reported by Stanislav in [#2120](#):

Yet another minor bug. Testcase:

```
def temp-table tt field f1 as integer.  
  
def var h1 as handle.  
def var h2 as handle.  
def var q as handle.  
  
h1 = buffer book:handle.  
h2 = buffer tt:handle.  
  
create query q.  
q:set-buffers(h1, h2).  
q:query-prepare("for each book, each tt").  
q:query-open.  
q:get-next.
```

Warning:

```
[11/14/2013 22:37:35 TMT] (com.goldencode.p2j.persist.Persistenc$Context:WARNING) [00000001:00000007:syman-->  
local/p2j_test/primary] rolling back open transaction during context cleanup
```

#2 - 01/07/2014 12:25 PM - Eric Faulhaber

Originally reported by Stanislav in [#2120](#):

Variation of the bug above. Testcase:

```
def temp-table tt0 field f1 as integer.
def temp-table tt field f1 as integer.

def var h1 as handle.
def var h2 as handle.
def var q as handle.

h1 = buffer tt0:handle.
h2 = buffer tt:handle.

create query q.
q:set-buffers(h1, h2).
q:query-prepare("for each tt0, each tt").
q:query-open.
q:get-next.
```

Stacktrace:

```
[11/14/2013 23:04:29 TMT] (com.goldencode.p2j.persist.Persistence$Context:WARNING) [00000001:00000006:syman-->
local/_temp/primary] rolling back open transaction during context cleanup
[11/14/2013 23:04:29 TMT] (com.goldencode.p2j.persist.Persistence$Context:WARNING) [00000001:00000006:syman-->
local/_temp/primary] rolling back orphaned transaction
[11/14/2013 23:04:29 TMT] (com.goldencode.p2j.persist.Persistence$Context:SEVERE) Error closing Hibernate sess
ion during context cleanup
com.goldencode.p2j.persist.PersistenceException: [00000001:00000006:syman-->local/_temp/primary] unable to rol
lback current transaction
    at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4615)
    at com.goldencode.p2j.persist.Persistence$Context.closeSession(Persistence.java:5095)
    at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4613)
    at com.goldencode.p2j.persist.Persistence$Context.cleanup(Persistence.java:5341)
    at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:618)
    at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:616)
    at com.goldencode.p2j.security.ContextLocal$Wrapper.cleanup(ContextLocal.java:381)
    at com.goldencode.p2j.security.SecurityContext.cleanup(SecurityContext.java:498)
    at com.goldencode.p2j.security.SecurityManager.endContext(SecurityManager.java:6772)
    at com.goldencode.p2j.security.SecurityManager.popContextWorker(SecurityManager.java:6710)
    at com.goldencode.p2j.security.SecurityManager.popAndRestoreSecurityContext(SecurityManager.java:3933)
    at com.goldencode.p2j.net.RouterSessionManager.restoreContext(RouterSessionManager.java:1065)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:191)
    at java.lang.Thread.run(Thread.java:722)
Caused by: org.hibernate.TransactionException: rollback failed
    at org.hibernate.engine.transaction.spi.AbstractTransactionImpl.rollback(AbstractTransactionImpl.java:
215)
    at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4602)
    at com.goldencode.p2j.persist.Persistence$Context.closeSession(Persistence.java:5095)
    at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4613)
    at com.goldencode.p2j.persist.Persistence$Context.cleanup(Persistence.java:5341)
    at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:618)
    at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:616)
    at com.goldencode.p2j.security.ContextLocal$Wrapper.cleanup(ContextLocal.java:381)
    at com.goldencode.p2j.security.SecurityContext.cleanup(SecurityContext.java:498)
    at com.goldencode.p2j.security.SecurityManager.endContext(SecurityManager.java:6772)
    at com.goldencode.p2j.security.SecurityManager.popContextWorker(SecurityManager.java:6710)
    at com.goldencode.p2j.security.SecurityManager.popAndRestoreSecurityContext(SecurityManager.java:3933)
    at com.goldencode.p2j.net.RouterSessionManager.restoreContext(RouterSessionManager.java:1065)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:191)
    at java.lang.Thread.run(Thread.java:722)
Caused by: org.hibernate.TransactionException: unable to rollback against JDBC connection
    at org.hibernate.engine.transaction.internal.jdbc.JdbcTransaction.doRollback(JdbcTransaction.java:167)
    at org.hibernate.engine.transaction.spi.AbstractTransactionImpl.rollback(AbstractTransactionImpl.java:
209)
    at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4602)
    at com.goldencode.p2j.persist.Persistence$Context.closeSession(Persistence.java:5095)
    at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4613)
```

```
at com.goldencode.p2j.persist.Persistence$Context.cleanup(Persistence.java:5341)
at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:618)
at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:616)
at com.goldencode.p2j.security.ContextLocal$Wrapper.cleanup(ContextLocal.java:381)
at com.goldencode.p2j.security.SecurityContext.cleanup(SecurityContext.java:498)
at com.goldencode.p2j.security.SecurityManager.endContext(SecurityManager.java:6772)
at com.goldencode.p2j.security.SecurityManager.popContextWorker(SecurityManager.java:6710)
at com.goldencode.p2j.security.SecurityManager.popAndRestoreSecurityContext(SecurityManager.java:3933)
at com.goldencode.p2j.net.RouterSessionManager.restoreContext(RouterSessionManager.java:1065)
at com.goldencode.p2j.net.Conversation.run(Conversation.java:191)
at java.lang.Thread.run(Thread.java:722)
```

Caused by: org.h2.jdbc.JdbcSQLException: The object is already closed [90007-169]

```
at org.h2.message.DbException.getJdbcSQLException(DbException.java:329)
at org.h2.message.DbException.get(DbException.java:169)
at org.h2.message.DbException.get(DbException.java:146)
at org.h2.message.DbException.get(DbException.java:135)
at org.h2.jdbc.JdbcConnection.checkClosed(JdbcConnection.java:1386)
at org.h2.jdbc.JdbcConnection.checkClosedForWrite(JdbcConnection.java:1374)
at org.h2.jdbc.JdbcConnection.rollback(JdbcConnection.java:460)
at org.hibernate.engine.transaction.internal.jdbc.JdbcTransaction.doRollback(JdbcTransaction.java:163)
at org.hibernate.engine.transaction.spi.AbstractTransactionImpl.rollback(AbstractTransactionImpl.java:
```

209)

```
at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4602)
at com.goldencode.p2j.persist.Persistence$Context.closeSession(Persistence.java:5095)
at com.goldencode.p2j.persist.Persistence$Context.rollback(Persistence.java:4613)
at com.goldencode.p2j.persist.Persistence$Context.cleanup(Persistence.java:5341)
at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:618)
at com.goldencode.p2j.persist.Persistence$1.cleanup(Persistence.java:616)
at com.goldencode.p2j.security.ContextLocal$Wrapper.cleanup(ContextLocal.java:381)
at com.goldencode.p2j.security.SecurityContext.cleanup(SecurityContext.java:498)
at com.goldencode.p2j.security.SecurityManager.endContext(SecurityManager.java:6772)
at com.goldencode.p2j.security.SecurityManager.popContextWorker(SecurityManager.java:6710)
at com.goldencode.p2j.security.SecurityManager.popAndRestoreSecurityContext(SecurityManager.java:3933)
at com.goldencode.p2j.net.RouterSessionManager.restoreContext(RouterSessionManager.java:1065)
at com.goldencode.p2j.net.Conversation.run(Conversation.java:191)
at java.lang.Thread.run(Thread.java:722)
```

#3 - 03/30/2014 02:53 PM - Eric Faulhaber

- Assignee set to *Stanislav Lomany*

#4 - 04/21/2014 12:07 PM - Stanislav Lomany

The issue is caused by an unclosed implicit transaction opened by a query. It doesn't cause an error for static queries because these queries are closed when they are deleted as static resources. For dynamic queries the query is closed when the external procedure in which it was opened finishes. IMO the best place to close the query is where static resources are deleted - in `ProcedureManager.scopeFinished`.

#5 - 04/22/2014 02:50 AM - Constantin Asofiei

Stanislav Lomany wrote:

The issue is caused by an unclosed implicit transaction opened by a query. It doesn't cause an error for static queries because these queries are closed when they are deleted as static resources. For dynamic queries the query is closed when the external procedure in which it was opened finishes. IMO the best place to close the query is where static resources are deleted - in `ProcedureManager.scopeFinished`.

The problem here is that a dynamic resource can survive a procedure ran persistent - please test what happens if you expose the query handle to the caller, and the code in note 2 is ran in a persistent procedure.

#6 - 04/22/2014 07:20 AM - Stanislav Lomany

The problem here is that a dynamic resource can survive a procedure ran persistent - please test what happens if you expose the query handle to the caller, and the code in note 2 is ran in a persistent procedure.

Constantin, what did you mean by "can survive a procedure ran persistent"? Query is closed when the persistent procedure is deleted. Is that what you expected or you think that the query scope can be longer or shorter in some cases?

#7 - 04/22/2014 07:57 AM - Constantin Asofiei

Stanislav Lomany wrote:

The problem here is that a dynamic resource can survive a procedure ran persistent - please test what happens if you expose the query handle to the caller, and the code in note 2 is ran in a persistent procedure.

Constantin, what did you mean by "can survive a procedure ran persistent"? Query is closed when the persistent procedure is deleted. Is that what you expected or you think that the query scope can be longer or shorter in some cases?

In previous note, you mentioned that you wanted to close the query when the procedure finishes, not when it gets deleted. The idea is that a dynamic resource can survive the persistent procedure which created (the INSTANTIATING-PROCEDURE), and it will survive until its INSTANTIATING-PROCEDURE (the creator) gets deleted.

The QueryWrapper.resourceDelete code handles the resource deletion (which is called by ProcedureManager, when a persistent procedure gets deleted), but I think that it's missing a query close call (as you mentioned) - please check if QueryWrapper.resourceDelete can help.

#8 - 05/01/2014 11:04 PM - Stanislav Lomany

Guys, we have a problem with dynamic queries which are converted to AdaptiveQueries. Consider testcase:

```
DEF TEMP-TABLE tt FIELD f1 AS INTEGER.  
CREATE tt. tt.f1 = 1.  
CREATE tt. tt.f1 = 2.  
DEF VAR q AS HANDLE.
```

```
RUN proc.  
q:GET-NEXT.  
MESSAGE STRING(tt.f1).
```

```
PROCEDURE proc:  
  CREATE QUERY q.  
  q:SET-BUFFERS(BUFFER tt:HANDLE).  
  q:QUERY-PREPARE("FOR EACH tt").  
  q:QUERY-OPEN.  
  q:GET-NEXT.  
  MESSAGE STRING(tt.f1).  
END.
```

Output is "1,1". While equivalent testcase with the static query produces correct output "1,2":

```
DEF TEMP-TABLE tt FIELD f1 AS INTEGER.  
CREATE tt. tt.f1 = 1.  
CREATE tt. tt.f1 = 2.  
DEFINE QUERY q FOR tt.
```

```
RUN proc.  
GET NEXT q.  
MESSAGE STRING(tt.f1).
```

```
PROCEDURE proc:  
  OPEN QUERY q FOR EACH tt.  
  GET NEXT q.  
  MESSAGE STRING(tt.f1).  
END.
```

Dynamic query has the following Progress-equivalent code:

```
OPEN QUERY inMemQuery1 FOR EACH tt.
```

Which is converted to

```
public class InMemP2jQuery1  
{  
  public TempRecord1.Buf tt = null;  
  
  AdaptiveQuery query0;  
  
  public void execute()  
  {  
    query0 = new AdaptiveQuery(tt, (String) null, null, "tt.id asc");
```

```

    }

    public P2JQuery getQuery()
    {
        return query0;
    }
}

```

While equivalent static conversion uses QueryWrapper:

```

final QueryWrapper query0 = new QueryWrapper("inMemQ", false);

public void execute()
{
    externalProcedure(new Block()
    {
        public void body()
        {
            RecordBuffer.openScope(tt);
            query0.assign(new AdaptiveQuery(tt, (String) null, null, "tt.id asc"));
            query0.open();
        }
    });
}

```

The difference between a pure AdaptiveQuery and an AdaptiveQuery assigned to a QueryWrapper is that a QueryWrapper sets standalone flag to false, while standalone queries perform cleanup and resets results when the current scope (not necessary top-level) ends (see PreselectQuery.registerCleaner).

Do you think that we should convert a dynamic query using QueryWrapper?

#9 - 05/02/2014 12:14 AM - Stanislav Lomany

- File svl_upd20140502a.zip added

Update for review for the original issues (except the one noted in the previous note).

#10 - 05/02/2014 12:28 AM - Eric Faulhaber

Stanislav Lomany wrote:

Do you think that we should convert a dynamic query using QueryWrapper?

Yes, assuming CREATE QUERY is semantically the dynamic equivalent of OPEN QUERY, since the semantics of a FOR EACH loop are different than an OPEN QUERY FOR EACH query.

#11 - 05/02/2014 02:38 AM - Constantin Asofiei

Stanislav Lomany wrote:

The difference between a pure AdaptiveQuery and an AdaptiveQuery assigned to a QueryWrapper is that a QueryWrapper sets standalone flag to false, while standalone queries perform cleanup and resets results when the current scope (not necessary top-level) ends (see PreselectQuery.registerCleaner).

Do you think that we should convert a dynamic query using QueryWrapper?

I think this might be a runtime problem, not a conversion problem: looking into the QueryWrapper.prepare and QueryWrapper.assignImpl, I think prepare should have called the assignImpl, and not just save the delegate: calling assignImpl should ensure that standalone is set to false for the query.

#12 - 05/02/2014 09:52 AM - Eric Faulhaber

OK, please make the change.

#13 - 05/04/2014 05:47 AM - Stanislav Lomany

- File svl_upd20140504a.zip added

Constantin, thank you for the idea!

#14 - 05/07/2014 10:07 PM - Eric Faulhaber

Code review 20140504a:

Update looks good. Please regression test and commit/distribute if it passes.

Does this fix both variants of the problem in notes 1 & 2?

#15 - 05/08/2014 07:02 AM - Stanislav Lomany

Does this fix both variants of the problem in notes 1 & 2?

Yes.

#16 - 05/19/2014 07:34 AM - Stanislav Lomany

- Status changed from New to WIP

#17 - 05/19/2014 07:34 AM - Stanislav Lomany

- Status changed from WIP to Review

#18 - 05/19/2014 04:33 PM - Eric Faulhaber

- % Done changed from 0 to 100

- Status changed from Review to Closed

#19 - 11/16/2016 12:07 PM - Greg Shah

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

Files

svl_upd20140502a.zip	26.2 KB	05/02/2014	Stanislav Lomany
svl_upd20140504a.zip	26.2 KB	05/04/2014	Stanislav Lomany