

Mulgara - Bug #19

Fix transaction support in Mulgara to support concurrent reads and read-only transactions. - Currently ACID failure

09/14/2006 04:51 AM - Andrae Muys -

Status:	Closed	Start date:	
Priority:	Immediate	Due date:	
Assignee:	Andrae Muys -	% Done:	0%
Category:	Mulgara	Estimated time:	0.00 hour
Target version:			
Resolution:	fixed		
Description			
<p>A correct implementation of server-side JRDF requires support for concurrent reads.</p> <p>
</p> <p>
</p> <p>Any client side round-tripping of blank-nodes requires access to read-only transactions.</p> <p>
</p> <p>
</p> <p>This requires refactoring (and some rewriting of) transaction support in the Session layer of Mulgara to support three-level transaction representation.</p> <p>
</p> <p>
</p> <p>User - We require a separate transaction object representing a users transaction. Currently represented by a javax.transaction.Transaction object on [[DatabaseSession]]. This needs encapsulating.</p> <p>
</p> <p>Query - We don't currently have any concept of a transaction at the per-query level. Without this concurrent reads can't be supported.</p> <p>
</p> <p>Method - Currently supported via methods on [[DatabaseSession]]. These need to be lifted into a separate manager object which can manage their interaction with query and user transactional contexts.</p>			

History

#1 - 09/16/2006 05:03 PM - Andrae Muys -

Violation of ACID Isolation in endPreviousQueryTransaction

Failure to close [[SubqueryAnswer]] object in

[[DatabaseSession]].endPreviousQueryTransaction() results in a

violation of ACID isolation for subqueries and counts when issuing

new queries if the Answers from previous queries have not been

closed.

The offending code is currently on line 868 of [[DatabaseSession]].java:


```
//Do not close tuples - for Jena and JRDF.  
<br/>  
//s.close();  
<br/>
```


NOTE: ACID is restored by uncommenting s.close(); however server-side JRDF is then broken. This is exposed by the following sequence:

Violation of ACID Isolation in [[DatabaseSession]].innerQuery

[[DatabaseSession]].innerQuery() fails to invalidate all Answer objects

associated with a given transaction when the transaction ends. This

is a violation of ACID isolation under the following use case:

1) Set autocommit off;

2) perform a query with a subquery;

3) write something into the store which would change the results of

the query in (2);

4) Get the subquery results from the Answer object from (2).

The subquery will not be isolated from the write.

#2 - 10/01/2006 03:09 PM - Paula Gearon

This issue subsumes the concurrent write problem described in:

http://mulgara.org/trac/ticket/2

#3 - 10/01/2006 03:14 PM - Paula Gearon

Moved to "Blocker" priority

#4 - 11/23/2006 06:49 AM - Andrae Muys -

Fixed in revision 138 in the branches/xafix-impl branch.

Please test.