

Database - Feature #2140

use positional query substitution parameters in HQLPreprocessor and related classes

05/02/2013 01:33 PM - Eric Faulhaber

Status:	Closed	Start date:	09/02/2013
Priority:	Normal	Due date:	09/05/2013
Assignee:	Stanislav Lomany	% Done:	100%
Category:		Estimated time:	24.00 hours
Target version:	Cleanup and Stabilization for Server Features		
billable:	No	vendor_id:	GCD
Description			
Related issues:			
Related to Database - Feature #1580: upgrade Hibernate to latest level		Closed	10/23/2012 05/03/2013

History

#1 - 05/02/2013 01:39 PM - Eric Faulhaber

- Estimated time set to 24.00

The query substitution parameter placeholders we emit in our converted HQL where clauses (?) are deprecated in HQL. When we moved to Hibernate 4, we were supposed to use JPA-style parameters instead (e.g., ?1, ?2, etc.).

We decided to implement this in the HQLPreprocessor (and in some related classes which may add AND conditions into the where clause, which themselves have additional placeholders) and leave the converted code alone.

See also [#1580](#), notes 17, 141, 142.

#2 - 05/02/2013 10:01 PM - Eric Faulhaber

- Due date set to 09/05/2013

- Start date changed from 05/02/2013 to 09/02/2013

#3 - 05/03/2013 04:53 AM - Stanislav Lomany

Basic idea of the solution is to:

1. Use setParameter(String, ...) in places like Persistence.getQuery (search for "setParameter" and "setParameters").
2. Change HQLPreprocessor.emit to use ?1, ?2 etc.
3. Change in the same way all other places that generate placeholders (search for "=?" and "= ?")
4. Make sure that through numeration is used in all parts that construct a single query.

#4 - 02/11/2014 04:54 PM - Stanislav Lomany

- File svl_upd20140212a.zip added

Update for review. Passed main part of regression testing. The main idea is to:

1. exclude explicit placeholders ("?", "?") from HQL string;
2. use the special class HQLExpression for HQL concatenation;
3. when HQL is fully combined, emit the HQL string with ?0, ?1 etc placeholders.
4. use HQLExpression for any HQL expression containing at least one parameter (for the case if we want to use named parameters in the future).

Server log still contains two "positional parameters are deprecated" warnings. I'll modify hibernate logging in order to pinpoint the cause of these warnings.

#5 - 02/11/2014 05:35 PM - Eric Faulhaber

How does this update take into account the parameter indices we are managing within the HQLPreprocessor (see paramIndices, createParameterIndices, getParameterIndices in that class)? With HQLExpression.toFinalExpression, we now have a parallel means of generating these index values. Will the inlining we do in HQLPreprocessor still work properly?

#6 - 02/12/2014 09:04 AM - Stanislav Lomany

How does this update take into account the parameter indices we are managing within the HQLPreprocessor (see paramIndices, createParameterIndices, getParameterIndices in that class)? With HQLExpression.toFinalExpression, we now have a parallel means of generating these index values. Will the inlining we do in HQLPreprocessor still work properly?

Yes, it works properly. All we do is change

```
"... = ? ... = ?" + setParameter(0, ...); setParameter(1, ...)
```

to

```
"... = ?0 ... = ?1" + setParameter("0", ...); setParameter("1", ...)
```

at the point where parameters are represented by a continuous array.

#7 - 02/12/2014 09:44 AM - Stanislav Lomany

- Status changed from New to WIP
- File svl_upd20140212b.zip added

Minor update for MAJIC (fixed query in TmpSumCleanupContextHook).

#8 - 02/12/2014 01:45 PM - Eric Faulhaber

Code review 20140212a:

Looks good, nice work. Once it passes the rest of regression testing, please check it in and distribute.

Minor question on the TmpSumCleanupContextHook change: any reason you didn't just hard-code the JPA-style placeholder in the constant string,

instead of using HQLExpression?

#9 - 02/12/2014 02:02 PM - Stanislav Lomany

I saw this code which is currently not active

```
if (types[k] instanceof org.hibernate.type.CompositeCustomType)
{
    query.setParameter("px" + (++posIndex), values[k], types[k]);
}
```

and thought that eventually we may go for named parameters. And in this case it will be better if we have centralized generation of placeholders so no one will have to look for hard-coded places like I did.

#10 - 02/12/2014 02:22 PM - Eric Faulhaber

This is something Greg or Ovidiu added specifically for composite data type support (specifically, datetime-tz), but there has been no decision to go to named parameters globally.

#11 - 02/13/2014 11:29 AM - Stanislav Lomany

- Status changed from WIP to Review

Committed to bzt revision 10462.

#12 - 02/24/2014 11:20 AM - Eric Faulhaber

- % Done changed from 0 to 100

- Status changed from Review to Closed

#13 - 11/16/2016 12:07 PM - Greg Shah

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

Files

svl_upd20140212a.zip	226 KB	02/11/2014	Stanislav Lomany
svl_upd20140212b.zip	2.99 KB	02/12/2014	Stanislav Lomany