

Why are methods not replayed on the PreparedStatement object in batch executor scenarios?

```
shardingsphere / jdbc / src / main / java / org / apache / shardingsphere / driver / jdbc / core / statement / ShardingSpherePreparedStatement.java
5.5.0
Code Blame 770 lines (680 loc) · 42 KB
Raw Copy Download Edit View
599 private MergedResult mergeQuery(final List<QueryResult> queryResults, final SQLStatementContext sqlStatementContext) throws SQLException {
600     MergeEngine mergeEngine = new MergeEngine(metadataContexts.getMetaData().getDatabase(databaseName),
601         metadataContexts.getMetaData().getProps(), connection.getDatabaseConnectionManager().getConnectionContext());
602     return mergeEngine.merge(queryResults, sqlStatementContext);
603 }
604
605 private void cacheStatements(final Collection<ExecutionGroup<JDBCExecutionUnit>> executionGroups) throws SQLException {
606     for (ExecutionGroup<JDBCExecutionUnit> each : executionGroups) {
607         each.getInputs().forEach(eachInput -> {
608             statements.add((PreparedStatement) eachInput.getStorageResource());
609             parameterSets.add(eachInput.getExecutionUnit().getSqlUnit().getParameters());
610         });
611     }
612     replay();
613 }
614
615 private void replay() throws SQLException {
616     replaySetParameter();
617     for (Statement each : statements) {
618         getMethodInvocationRecorder().replay(each);
619     }
620 }
621
622 private void replaySetParameter() throws SQLException {
623     for (int i = 0; i < statements.size(); i++) {
624         replaySetParameter(statements.get(i), parameterSets.get(i));
625     }
}
```



5.5.0

shardingsphere / jdbc / src / main / java / org / apache / shardingsphere / driver / jdbc / core / statement / ShardingSpherePreparedStatement.java

↑ Top

Code

Blame

770 lines (680 loc) · 42 KB

Raw



```
114 public final class ShardingSpherePreparedStatement extends AbstractPreparedStatementAdapter {
694 private int[] doExecuteBatch(final BatchPreparedStatementExecutor batchExecutor) throws SQLException {
706 }
707
708 private void initBatchPreparedStatementExecutor(final BatchPreparedStatementExecutor batchExecutor) throws SQLException {
709     DriverExecutionPrepareEngine<JDBCExecutionUnit, Connection> prepareEngine = new DriverExecutionPrepareEngine<>(JDBCDriverType.PR
710     .<Integer>getValue(ConfigurationPropertyKey.MAX_CONNECTIONS_SIZE_PER_QUERY), connection.getDatabaseConnectionManager(),
711     metaDataContexts.getMetaData().getDatabase(databaseName).getRuleMetaData().getRules(),
712     metaDataContexts.getMetaData().getDatabase(databaseName).getResourceMetaData().getStorageUnits());
713     List<ExecutionUnit> executionUnits = new ArrayList<>(batchExecutor.getBatchExecutionUnits().size());
714     for (BatchExecutionUnit each : batchExecutor.getBatchExecutionUnits()) {
715         ExecutionUnit executionUnit = each.getExecutionUnit();
716         executionUnits.add(executionUnit);
717     }
718     batchExecutor.init(prepareEngine.prepare(executionContext.getRouteContext(), executionUnits, new ExecutionGroupReportContext(con
719     setBatchParametersForStatements(batchExecutor));
720 }
721
722 private void setBatchParametersForStatements(final BatchPreparedStatementExecutor batchExecutor) throws SQLException {
723     for (Statement each : batchExecutor.getStatements()) {
724         List<List<Object>> paramSet = batchExecutor.getParametersFor(each);
725         for (List<Object> eachParams : paramSet) {
726             replaySetParameter((PreparedStatement) each, eachParams);
727             ((PreparedStatement) each).addBatch();
728         }
729     }
730 }
```

After parameter setting, methods on ShardingPreparedStatement are not replayed on actual JDBC PreparedStatement.