

Database - Bug #3367

collision between sequence and table names

11/08/2017 03:28 AM - Ovidiu Maxiniuc

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	50%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#1 - 11/08/2017 03:53 AM - Ovidiu Maxiniuc

- Status changed from New to WIP

Constantin found the following exception in log while doing an import:

```
WARN: SQL Error: 0, SQLState: 42809
Nov 06, 2017 7:27:24 PM org.hibernate.engine.jdbc.spi.SqlExceptionHelper logExceptions
ERROR: ERROR: "employee" is not a sequence
org.hibernate.exception.SQLGrammarException: ERROR: "employee" is not a sequence
    at org.hibernate.exception.internal.SQLStateConversionDelegate.convert (SQLStateConversionDelegate.java:122
)
    at org.hibernate.exception.internal.StandardSQLExceptionConverter.convert (StandardSQLExceptionConverter.java:49)
    at org.hibernate.engine.jdbc.spi.SqlExceptionHelper.convert (SqlExceptionHelper.java:125)
    at org.hibernate.engine.jdbc.spi.SqlExceptionHelper.convert (SqlExceptionHelper.java:110)
    at org.hibernate.engine.jdbc.internal.proxy.AbstractStatementProxyHandler.continueInvocation (AbstractStatementProxyHandler.java:130)
    at org.hibernate.engine.jdbc.internal.proxy.AbstractProxyHandler.invoke (AbstractProxyHandler.java:81)
    at com.sun.proxy.$Proxy5.executeQuery (Unknown Source)
    at org.hibernate.loader.Loader.getResultSet (Loader.java:1926)
    at org.hibernate.loader.Loader.executeQueryStatement (Loader.java:1727)
    at org.hibernate.loader.Loader.doQuery (Loader.java:852)
    at org.hibernate.loader.Loader.doQueryAndInitializeNonLazyCollections (Loader.java:293)
    at org.hibernate.loader.Loader.doList (Loader.java:2411)
    at org.hibernate.loader.Loader.doList (Loader.java:2397)
    at org.hibernate.loader.Loader.listIgnoreQueryCache (Loader.java:2227)
    at org.hibernate.loader.Loader.list (Loader.java:2222)
    at org.hibernate.loader.custom.CustomLoader.list (CustomLoader.java:331)
    at org.hibernate.internal.SessionImpl.listCustomQuery (SessionImpl.java:1783)
    at org.hibernate.internal.AbstractSessionImpl.list (AbstractSessionImpl.java:231)
    at org.hibernate.internal.SQLQueryImpl.list (SQLQueryImpl.java:156)
    at com.goldencode.p2j.schema.ImportWorker$Library.initializeSequences (ImportWorker.java:872)
    at com.goldencode.expr.CE154.execute (Unknown Source)
    at com.goldencode.expr.Expression.execute (Expression.java:391)
    at com.goldencode.p2j.pattern.Rule.apply (Rule.java:491)
    at com.goldencode.p2j.pattern.RuleContainer.apply (RuleContainer.java:583)
    at com.goldencode.p2j.pattern.RuleSet.apply (RuleSet.java:98)
    at com.goldencode.p2j.pattern.PatternEngine.apply (PatternEngine.java:1650)
    at com.goldencode.p2j.pattern.PatternEngine.processAst (PatternEngine.java:1529)
    at com.goldencode.p2j.pattern.PatternEngine.processAst (PatternEngine.java:1477)
    at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:1032)
    at com.goldencode.p2j.pattern.PatternEngine.main (PatternEngine.java:2108)
```

After preliminary investigations the failure is caused by a name collision between sequences and tables share the same namespace in PostgreSQL (and most likely other database engines). In P4GL they reside in separate namespaces, so it is perfectly legal for a table and a sequence to share

the same name. Since the sequences were added later, the initial conversion code assumes that the table names are already unique and there will not be any name collision.

The solution for this issue is to add a dedicated suffix for sequence or better, an incrementing one. Alternatively, hints can be added in order to allow a better customization of the involved entities.

#2 - 11/08/2017 11:20 AM - Ovidiu Maxiniuc

Although at first I tried to fix the issue in TRPL I realized that NameConverterWorker is a better place. It can store the processed table names for each schema and the reports generated by it are consistent (the true final converted name is listed).

I created branch 3367a and committed my changes as revision 11197. Please review. I've started a quick conversion tests, meanwhile.

#3 - 11/08/2017 12:49 PM - Ovidiu Maxiniuc

The standard conversion test on devstv01 finished as expected, without any changes in generated code.

The convert.middle.list conversion on client sources also ended with success. Since there is only one collision, only the classes related to employee table were changed (Employee, EmployeeImpl, EmployeeImpl.hbm and <schema>.meta.xml).

There is a small inconvenient. Because in .df file the employee sequence is declared before the table with same name, the sequence gets the unindexed SQL name. The SQL table gets named employee_1 and the indexes accordingly.

#4 - 11/08/2017 01:25 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

There is a small inconvenient. Because in .df file the employee sequence is declared before the table with same name, the sequence gets the unindexed SQL name. The SQL table gets named employee_1 and the indexes accordingly.

I would prefer it be the other way around. Generally speaking, I think tables are the more "important" resource and should take precedence in the resolution of naming collisions with sequences, though I understand this is not convenient, given how the conversion of these resources is ordered in our current process.

Considering the urgency of the fix, we should leave it for now (note however that I still need to review the code). We'll have to come back around and address this shortly, though.

#5 - 11/08/2017 02:05 PM - Eric Faulhaber

Code review 3367a/11197:

Thank you for the quick turnaround on this fix.

The code looks fine to me, with the caveat that we ultimately want to change the priority of sequence over table naming as noted previously.

However, the fix I have in mind requires more testing and we'll have to come back to it later.

For future reference, I believe we can get around the priority issue by re-sorting the schema AST top-level nodes such that tables come before sequences. We do something similar to this with DMO properties during the P2O phase. Then, your fix should work naturally, but with table names given precedence. However, this may have a risk of side effects (can you think of any offhand?), so I want to hold off on it for now.

Given that your initial testing went well, I plan to run a quick conversion regression test and merge this to trunk if that goes well.

#6 - 11/08/2017 03:28 PM - Eric Faulhaber

- % Done changed from 0 to 50

Eric Faulhaber wrote:

...I plan to run a quick conversion regression test...

Never mind, I see you did that already. Task branch 3367a was merged to trunk revision 11197.

I am leaving this issue open until we can come back around and fix the priority issue.