Database - Bug #7821

Duplication of table records in schema namespace causes dynamic conversion to fail

09/20/2023 05:46 PM - Ovidiu Maxiniuc

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 09/20/2023 06:11 PM - Ovidiu Maxiniuc

- Subject changed from Duplication of field records in schema namespace causes dynamic conversion to fail to Duplication of table records in schema namespace causes dynamic conversion to fail

I found that the schema dictionary is created and the private nodes List is populated, in certain conditions, the NameNode items get duplicated. Here are the snapshots of the moments:

```
      at com.goldencode.p2j.schema.Namespace.add(Namespace.java:247)
      at com.goldencode.p2j.schema.Namespace.add(Namespace.java:211)
      at com.goldencode.p2j.schema.NameNode.setParent(NameNode.java:219)
      at com.goldencode.p2j.schema.NameNode.<init>(NameNode.java:174)
      at com.goldencode.p2j.schema.SchemaDictionary.addEntry(SchemaDictionary.java:4863)
      *
      at com.goldencode.p2j.schema.SchemaDictionary.loadFromAst(SchemaDictionary.java:4208)
      at com.goldencode.p2j.schema.SchemaDictionary.loadFromAst(SchemaDictionary.java:4133)
      at com.goldencode.p2j.schema.SchemaLoader.loadDefaults(SchemaLoader.java:703)
      at com.goldencode.p2j.schema.SchemaDictionary.<init>(SchemaDictionary.java:1080)
      at com.goldencode.p2j.schema.SchemaDictionary.<init>(SchemaDictionary.java:1009)
      at com.goldencode.p2j.schema.SchemaWorker.<init>(SchemaWorker.java:205)
      at sun.reflect.NativeConstructorAccessorImpl.newInstance0(NativeConstructorAccessorImpl.java:-1)
      at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
      at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
      at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
      at com.goldencode.p2j.pattern.RuleContainer.registerWorker(RuleContainer.j
```

```
at com.goldencode.p2j.schema.Namespace.add(Namespace.java:247)
      at com.goldencode.p2j.schema.SchemaDictionary$Scope.addOrReplaceNode(SchemaDictionary.java:6632)
      at com.goldencode.p2j.schema.SchemaDictionary.addEntry(SchemaDictionary.java:4916)
      *
      at com.goldencode.p2j.schema.SchemaDictionary.loadFromAst(SchemaDictionary.java:4208)
      at com.goldencode.p2j.schema.SchemaDictionary.loadFromAst(SchemaDictionary.java:4133)
      at com.goldencode.p2j.schema.SchemaLoader.loadDefaults(SchemaLoader.java:703)
      at com.goldencode.p2j.schema.SchemaDictionary.<init>(SchemaDictionary.java:1080)
      at com.goldencode.p2j.schema.SchemaDictionary.<init>(SchemaDictionary.java:1009)
      at com.goldencode.p2j.schema.SchemaWorker.<init>(SchemaWorker.java:205)
      at sun.reflect.NativeConstructorAccessorImpl.newInstance0(NativeConstructorAccessorImpl.java:-1)
      at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
      at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
      at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
      at com.goldencode.p2j.pattern.RuleContainer.registerWorker(RuleContainer.java:687)
      at com.goldencode.p2j.pattern.ConfigLoader.worker(ConfigLoader.java:814)
```

```
ava:687)
      at com.goldencode.p2j.pattern.Conf
igLoader.worker(ConfigLoader.java:814)
      at com.goldencode.p2j.pattern.Conf
igLoader.processChildElements(ConfigLoad
er.java:770)
      at com.goldencode.p2j.pattern.Conf
igLoader.loadImpl(ConfigLoader.java:617)
      at com.goldencode.p2j.pattern.Conf
igLoader.load(ConfigLoader.java:548)
      at com.goldencode.p2j.pattern.Patt
ernEngine.initialize(PatternEngine.java:
1199)
      at com.goldencode.p2j.pattern.Patt
ernEngine.<init>(PatternEngine.java:590)
      at com.goldencode.p2j.persist.Conv
ersionPool.addEngine(ConversionPool.java
:428)
      at com.goldencode.p2j.persist.Conv
ersionPool.initialize(ConversionPool.jav
a:281)
      at com.goldencode.p2j.main.Standar
dServer.bootstrap(StandardServer.java:10
98)
      at com.goldencode.p2j.main.ServerD
river.start(ServerDriver.java:536)
      at com.goldencode.p2j.main.CommonD
river.process(CommonDriver.java:555)
      at com.goldencode.p2j.main.ServerD
river.process(ServerDriver.java:1)
      at com.goldencode.p2j.main.ServerD
river.main(ServerDriver.java:1010)
```

```
      at com.goldencode.p2j.pattern.Conf
igLoader.processChildElements(ConfigLoad
er.java:770)
      at com.goldencode.p2j.pattern.Conf
igLoader.loadImpl(ConfigLoader.java:617)
      at com.goldencode.p2j.pattern.Conf
igLoader.load(ConfigLoader.java:548)
      at com.goldencode.p2j.pattern.Patt
ernEngine.initialize(PatternEngine.java:
1199)
      at com.goldencode.p2j.pattern.Patt
ernEngine.<init>(PatternEngine.java:590)
      at com.goldencode.p2j.persist.Conv
ersionPool.addEngine(ConversionPool.java
:428)
      at com.goldencode.p2j.persist.Conv
ersionPool.initialize(ConversionPool.jav
a:281)
      at com.goldencode.p2j.main.Standar
dServer.bootstrap(StandardServer.java:10
98)
      at com.goldencode.p2j.main.ServerD
river.start(ServerDriver.java:536)
      at com.goldencode.p2j.main.CommonD
river.process(CommonDriver.java:555)
      at com.goldencode.p2j.main.ServerD
river.process(ServerDriver.java:1)
      at com.goldencode.p2j.main.ServerD
river.main(ServerDriver.java:1010)
```

As result, duplicate nodes are added to the list, causing problems later, when the dynamic conversion attempts to look them up. The stack trace and error looks like this:

```
'book' is ambiguous; 2 matches found [fwd.book, fwd.book]
    at com.goldencode.p2j.schema.Namespace.findUnambiguously(Namespace.java:666)
    at com.goldencode.p2j.schema.Namespace.find(Namespace.java:529)
    at com.goldencode.p2j.schema.Namespace.find(Namespace.java:476)
    at com.goldencode.p2j.schema.SchemaDictionary$Scope.findRecordNode(SchemaDictionary.java:6446)
    at com.goldencode.p2j.schema.SchemaDictionary$Scope.findNode(SchemaDictionary.java:6732)
    at com.goldencode.p2j.schema.SchemaDictionary.findEntry(SchemaDictionary.java:4643)
    at com.goldencode.p2j.schema.SchemaDictionary.findTableNode(SchemaDictionary.java:3469)
    [...]
```

I think the addOrReplaceNode() does not work as expected. A bit deeper, in Namespace.find(), we use Collections.binarySearch() to locate (unique) nodes. This assumes the nodes structure is a correctly ordered list. But this is not the case. For my example, the fwd.book nodes could be found at position 91 and 362. Probably it all starts with a single node being incorrectly inserted. Then, because of the nature of the binary search, the subsequent inserts are performed at incorrect offsets and the same will happen with splitting the interval in half on lookup.

The truth is, there were multiple databases loaded and some of their tables carrying the same name. However, this is the normal case if we consider the common meta content. I think this might be the root of the problem: two tables with same name from different databases are added, they are somehow identical, but sorted also by the database prefix. Need more investigation for a definitive answer.