Database - Feature #2281

enhance schema name conversion to support literal replacements

04/10/2014 12:14 AM - Eric Faulhaber

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Vadim Nebogatov	**% Done:**	100%
Category:		**Estimated time:**	40.00 hours
Target version:	Cleanup and Stablization for Server Features		
billable:	No	**vendor_id:**	GCD
Description			

History

#1 - 04/10/2014 12:39 AM - Eric Faulhaber

- Estimated time set to 40.00

We need to enhance the schema hint syntax to allow the name conversion algorithms to be bypassed entirely, such that the schema hints provide the exact name replacements for a 4GL table name (to an RDBMS table name and DMO class name) and for a 4GL field name (to an RDBMS column name and a DMO property name). Further, we need to provide hint syntax to indicate that RDBMS table and column names are to be escaped in the generated DDL.

The proposed syntax changes are:

- Add an optional escape (true/false) attribute to the schema, table, and field elements. If escape is true at any level, all DDL names at that level and below (unless overridden by a more specific escape attribute setting in a nested element) will be escaped by the appropriate, dialect-specific escape character (e.g., double quotes) for the target database.
- Add an optional sql-table attribute and an optional dmo attribute to the table element. The value of sql-table specifies the exact table name to generate for the DDL for that table. The value of dmo specifies the exact DMO class name to generate for the Java DMO interface (and implementation class, with an appended "Impl" as we do today). If one attribute is present and not the other, use the value of the provided attribute for the missing attribute. If either attribute is present, any phrase child element is ignored (and a warning should be issued that it is being ignored).
- Add an optional sql-column attribute and an optional property attribute to the field element. The value of sql-column specifies the exact column name to generate for the DDL for that field. The value of property specifies the exact DMO property name to generate for the Java DMO interface and implementation class getter/setter methods. If one attribute is present and not the other, use the value of the provided attribute for the missing attribute. If either attribute is present, any phrase child element is ignored (and a warning should be issued that it is being ignored).

None of these changes apply to dynamically-defined temp-tables (which are not supported by our hint syntax today, anyway). In fact, the changes are not strictly necessary for static temp-tables, either, but do not go out of your way to exclude static temp-tables. If the implementation effort is the same either way, allow the enhanced syntax to apply to static temp-tables as well.

#2 - 04/10/2014 10:55 AM - Vadim Nebogatov

- Status changed from New to WIP

#3 - 04/10/2014 01:41 PM - Vadim Nebogatov

The question concerning fields with extent.
Would be possible using denormalization hints along with literal replacements hints at the same conversion? If yes: should we at first apply literal replacements hints and after that denormalization hints?

#4 - 04/10/2014 02:51 PM - Eric Faulhaber

Yes and yes.

#5 - 04/13/2014 02:40 PM - Vadim Nebogatov

The question concerning using escape attribute.

Should it be done on the level of overriding normalizer.normalizeIdentifierQuoting(...) method? How will it be correlated with using delimited-identifier/ hibernate.globally_quoted_identifiers settings?

#6 - 04/18/2014 07:21 PM - Vadim Nebogatov

I am trying to implement escape support using Hybernate's backticks ` for columns with escape hint. In this case such columns will be automatically quoted based on dialect.
My question is concerning escaping corresponding computed columns. I think they could be NOT escaped even if column itself is escaped.

For example column name with hint will be generated as `name` and will be substituted to DDL as "name", but computed column will have name __name and will be substituted as is. This approach will also require some changes in calculation of computed column name depending on presense of back tick.

#7 - 04/18/2014 09:24 PM - Eric Faulhaber

The reason for this feature is that some customers (as for the current project) want to use names where case is relevant, like FirstName. Many databases will not preserve case in table/column names. When writing SQL, they most likely will want to use the same convention whether referencing computed columns or regular columns, so I think it would make sense to escape the computed column names as well.

#8 - 04/19/2014 02:46 PM - Eric Faulhaber

Vadim Nebogatov wrote:

> The question concerning using escape attribute.
>
> Should it be done on the level of overriding normalizer.normalizeIdentifierQuoting(...) method? How will it be correlated with using delimited-identifier/ hibernate.globally_quoted_identifiers settings?

Sorry, I didn't see this question earlier. Unfortunately, I am not familiar with these settings in Hibernate. If this is still an issue, please explain what these settings are and how they are relevant to this task.

#9 - 04/20/2014 10:17 AM - Vadim Nebogatov

Eric, no problems, it is not an issue already.

It seems approach with using Hybernate's backticks is good and I already receive escaped fields and table names using hints.

Some additional issues are escaping computed columns (practically resolved) and escaping expressions in hibernate.xml used in indexes:

```
...
                       sprintf('idx_%s__%d',
                               parent.getAnnotation('table'),
                               refAst.id))
...
                  'table', sprintf('%s__%d',
                                   parent.getAnnotation('table'),
                                   extentVal),
...
```

I think to replace sprintf with new method in such cases

```
   public static String getEscapedTable(String format, String table, Long index)
   {
      StringBuilder buf = new StringBuilder();
      if (table.startsWith("`"))
      {
         table = table.substring(1, table.length() - 1);
         format = '`' + format + '`';
      }
      buf.append(String.format(format, table, index));
      return buf.toString();
   }
```

It is on testing stage. Estimation hours for this task are exceeded because of these issues, but I hope to complete them in 1-2 days.

#10 - 04/21/2014 03:35 AM - Vadim Nebogatov

Some more changes implemented:

```
'idx__%s' (generate_ddl.xml and other files)
```

```
indexName = "idx__" + indexName; in IndexHelper
```

```
Escape names in all P2J SQL statements like "drop index if exists " in all P2J dialects
```

replace

```
              table.getQuotedName(this),
              computedCol.getQuotedName(this),
```

with

```
              quote(table.getName()),
              quote(computedCol.getName()),
```

for "alter table %s add %s as %s;"

```
Quote indes name in P2J dialects in Index.buildSqlCreateIndexString(...)
```

Current issue is

```
<ast id="0" type="CONTENT" text="${table}_fkey" />
```

#11 - 04/21/2014 04:06 PM - Vadim Nebogatov

Processed formats: "%s_timestamp", "%s_offset", "alter table %s alter column %s %s as %s;"

Actually only two places '%s_fkey' on hibernate.xml are left.

#12 - 04/22/2014 03:49 PM - Vadim Nebogatov

- File vmn_upd20140422a.zip added

I have attached first results: update vmn_upd20140422a.zip contains example of new hints and corresponding DDL files. Please check using backtick symbols in corresponding tables and fields in p2j_test.p2o.

Remaining issue is escaping schema name. It is not used in DDL, so I try to understand now where it should be escaped. Code commenting and cleaning are also in progress.

#13 - 04/22/2014 03:58 PM - Eric Faulhaber

Vadim Nebogatov wrote:

> Remaining issue is escaping schema name. It is not used in DDL, so I try to understand now where it should be escaped. Code commenting and cleaning are also in progress.

Unless there is some reason Hibernate needs this, you do not have to escape the schema name. The purpose of the escape attribute at the schema element level is simply to inherit the setting for table and field names schema-wide.

#14 - 04/23/2014 02:50 PM - Vadim Nebogatov

- File vmn_upd20140423b.zip added

- File vmn_upd20140423a.zip added

I have attached update vmn_upd20140423a.zip and examples of hints and conversion results vmn_upd20140423b.zip. The code corresponds latest bzr revision 10512.

#15 - 04/29/2014 12:23 PM - Vadim Nebogatov

- File vmn_upd20140429a.zip added

I have attached update vmn_upd20140429a.zip after some fixes during testing regression and merge with latest bzr changes. The code corresponds to latest bzr revision 10516.

#16 - 04/29/2014 03:54 PM - Vadim Nebogatov

Regression 20140429_114111 completed. 3 FAILED tests:

1. tc_job_002 FAILED failure in step 40: 'Unexpected EOF in actual at page # 1.'

2. tc_item_master_031 FAILED failure in step 24: 'timeout before the specific screen buffer became available (Mismatched data at line 6, column 72. Expected 'Y' (0x0059 at relative Y 6, relative X 72) and found 'R' (0x0052 at relative Y 6, relative X 72).)'

3. tc_job_clock_002 FAILED failure in step 20: 'timeout before the specific screen buffer became available (Mismatched data at line 5, column 18. Expected '' (0x0000 at relative Y 5, relative X 18) and found ' ┌' (0x250C at relative Y 5, relative X 18).)'

I started regression before your code review only in order to find/fix some bugs. Should I continue regression?

#17 - 04/29/2014 03:56 PM - Eric Faulhaber

Vadim Nebogatov wrote:

> I started regression before your code review only in order to find/fix some bugs. Should I continue regression?

I'm sorry I've been quiet on this. I have done a very preliminary review and I will be requesting some changes (hopefully tonight or tomorrow), so please don't run through regression again yet.

#18 - 04/29/2014 04:13 PM - Greg Shah

In regard to the regression test failures, please make sure you have read the new timco.html document. It describes in detail, which tests are expected to fail and which tests will sometimes fail. You should see some useful information there.

#19 - 05/01/2014 02:39 AM - Eric Faulhaber

Code review 20140429a:

Unfortunately, there are some major problems with this update. Except for the escaping with quoted literals feature, which seems to work OK, the update does not meet the core requirements of the specification outlined in note 1. Namely, none of the dmo, sql-table, property, or sql-column attributes are honored by the schema conversion.

For example, in the samples you provided as vmn_upd20140423b.zip, the hints you specified should have caused a NewPersonImpl DMO class to be generated (instead of PersonImpl, the default), which implements the NewPerson interface (instead of Person). All references in the DMO class and interface to schedule should instead be schedule_property.

Likewise, the hinted names were not in the DDL. The SQL table name should be "new_person" rather than "person", and references to the "schedule" column should be "new_schedule_column".

You did not provide a generated Hibernate ORM document example, so I couldn't review that. However, other than the backticks, the rest of the features can't be working for that document either, based on the update.

It looks like you added code to process the new hint syntax to the TableHints class, but you either did not expose the new attributes to TRPL, or exposed them and did not use them. There should be getter methods for each of the new name-related XML attributes in note 1. You added getCustomDmoName and getCustomPropertyName methods, but they are not called from p2o.xml. You need corresponding methods in the TableHints class for the sql-table and sql-column attributes too, so they can be used from p2o.xml as well. These (or some derived value) should be stored as annotations in the *.p2o AST, so they can be accessed from other rule sets (like hibernate.xml and generate_ddl.xml).

You added helper methods for backtick processing to ProgressPatternWorker, but that is not the appropriate class to which to add these APIs, since that pattern worker is all about Progress ASTs and doesn't have anything to do with database/persistence conversion. The backtick APIs are more appropriately added to DataModelWorker, which is the main pattern worker for p2o.xml and its related rule sets.

AFAICT, the change to DynamicTablesHelper is not needed, because one cannot define schema hints for dynamically defined temp-tables.

I didn't see any dependency on the static data in the PropertyHelper class for the new methods you added to that class. If that is indeed the case, please move these methods to the DBUtils class instead, which I think is a more appropriate home.

You added a quote method to the P2JDialect interface, and you added calls to it in the concrete implementations of this interface, but the quote method doesn't appear to be implemented anywhere.

I don't think the change to HQLPreprocessor makes sense: property names should never have backticks, though the columns they are mapped to may. The column names will never appear in HQL, however.

As a general rule, please do not import individual class names into a Java class (see TableHints), unless you are resolving ambiguity, and don't add blank lines in the import statement block (sometimes an IDE like eclipse will do both without your knowledge when cutting and pasting).

Please address all these issues and post a new code update and new samples of the hints and converted output when ready.

#20 - 05/02/2014 08:00 AM - Vadim Nebogatov

- File vmn_upd20140502a.zip added

- File vmn_upd20140502b.zip added

Sorry, it is my fault. I worked last time with backticks, switched off to another bug fix, and did not include my previous changes related to other attributes.

Please check update vmn_upd20140502a.zip with results and used hints vmn_upd20140502b.zip with all your notices addressed.

> You added a quote method to the P2JDialect interface, and you added calls to it in the concrete implementations of this interface, but the quote method doesn't appear to be implemented anywhere.

Method quote is implemented in Dialect class, and I thought it would be useful to expose it in P2jDialect interface. Now quote method is used in hibernate.xml for interface p2jDialect, not for concrete dialect implementation.

#21 - 05/02/2014 08:46 AM - Vadim Nebogatov

- File deleted (vmn_upd20140502a.zip)

#22 - 05/02/2014 08:46 AM - Vadim Nebogatov

- File deleted (vmn_upd20140502b.zip)

#23 - 05/02/2014 08:48 AM - Vadim Nebogatov

- File vmn_upd20140502a.zip added

- File vmn_upd20140502b.zip added

#24 - 05/02/2014 08:53 AM - Vadim Nebogatov

- File deleted (vmn_upd20140502a.zip)

#25 - 05/02/2014 08:54 AM - Vadim Nebogatov

- File vmn_upd20140502a.zip added

#26 - 05/02/2014 10:07 AM - Eric Faulhaber

I haven't had time to look at the code yet, but there seems to be something wrong with the result. The hints file has this hint for the person table:

```
<field name="schedule" escape="true" sql-column="new_schedule_column" property="schedule_property">
```

This should result in a DMO property of "schedule_property", but it is converting to "scheduleProperty". Although the latter is preferred, it is not the verbatim result that the hint required.

This result suggests the conversion code is not bypassing the normal name conversion to take the hint's value directly, but rather is using the property hint attribute value as an input to the normal name conversion.

With this hint syntax, we are placing the burden of appropriate name choices on the author of the hints, and we should not be doing any additional processing with NameConverter[Worker].

#27 - 05/04/2014 10:49 AM - Vadim Nebogatov

- File vmn_upd20140504b.zip added

- File vmn_upd20140504a.zip added

Actually I thought that properties and class names should meet java naming conventions.

I have attached vmn_upd20140504a.zip/vmn_upd20140504b.zip with naming exactly like specified in hints, only I left adding unique suffix on names collision. Is it ok or also should not be done for hinted fields/tables?

#28 - 05/05/2014 02:42 PM - Eric Faulhaber

Vadim Nebogatov wrote:

> Actually I thought that properties and class names should meet java naming conventions.

Generally speaking, yes, we do enforce this with the default name conversion. However, for this particular feature, we are leaving the choice of name fully in the hands of the hint author. If he/she wants to break convention, so be it.

> I have attached vmn_upd20140504a.zip/vmn_upd20140504b.zip with naming exactly like specified in hints, only I left adding unique suffix on names collision. Is it ok or also should not be done for hinted fields/tables?

This is a good question.

Again, when this new type of hint syntax is used, the name choice is fully up to the hint author, so why don't we do this: under normal circumstances (i.e., when there is not a hint of this type which exactly specifies the new name), we leave the logic which disambiguates name collisions as is. So, in the absence of one of these new-style hints, everything works as it did before. However, in the case where we *do* have one of these new types of hints which specifies an exact name, we just detect the name collision and write out a warning, with enough detail for someone to easily identify and correct the error.

In the event of a name collision, the converted result will be incorrect, but that's OK -- it is up to the hint author to read the warnings and fix the hints. The idea is, we don't want to override the hinted name, rather, we want to make the hint author aware of the issue, so it can be fixed properly.

#29 - 05/07/2014 08:02 AM - Vadim Nebogatov

- File vmn_upd20140507a.zip added

- File vmn_upd20140507a.zip added

Update vmn_upd20140507a.zip with results and used hints vmn_upd20140507b.zip attached.

Removed adding unique suffix on names collision for hints and added warnings in such cases. Added tests for check such collisions. Corrected: adding to collision tables and columns maps before escaping (p2o.xml).

#30 - 05/07/2014 04:04 PM - Eric Faulhaber

Vadim Nebogatov wrote:

> Update vmn_upd20140507a.zip with results and used hints vmn_upd20140507b.zip attached.

Please note you attached the code archive (0507a) twice, but did not attach the test results archive (0507b).

#31 - 05/07/2014 04:53 PM - Vadim Nebogatov

- File deleted (vmn_upd20140507a.zip)

#32 - 05/07/2014 04:53 PM - Vadim Nebogatov

- File vmn_upd20140507b.zip added

#33 - 05/07/2014 09:45 PM - Eric Faulhaber

Code review 20140507a:

This update looks much better.

Regarding the implementation of the naming collision warning in p2o.xml, unless I'm misinterpreting the logic, it seems that one always will get the collision warning if the duplicate name (i.e., the second one encountered) was provided by a verbatim name conversion hint. However, it seems possible to bypass the warning if the first instance of the name encountered was provided by a hint, but the duplicate (i.e., the second one encountered) was generated through normal name conversion. This is inconsistent, since it is arbitrary as to whether the warning will occur or not, based on the order of processing the tables.

Now, this is a bit of an edge case, so here is what I would like you to do: if the update already has passed regression testing, commit and distribute it. Then, assess my interpretation of the logic and fix it if necessary. If the update already has passed regression testing, only a follow-up conversion regression test should be necessary.

#34 - 05/11/2014 11:49 AM - Vadim Nebogatov

Regression executed 2 times:

20140508_182010: 2 FAILED (all timeout related): tc_job_002, tc_pay_sfc_003

20140510_153509: 3 FAILED (all timeout related): tc_job_002, tc_job_clock_005, tc_job_matlcron_002

So, update vmn_upd20140507a.zip passed regression, committed and distributed

#35 - 05/11/2014 09:57 PM - Eric Faulhaber

- % Done changed from 0 to 100

- Status changed from WIP to Closed

#36 - 11/16/2016 12:07 PM - Greg Shah

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

Files

vmn_upd20140422a.zip	14.1 KB	04/22/2014	Vadim Nebogatov
vmn_upd20140423a.zip	223 KB	04/23/2014	Vadim Nebogatov
vmn_upd20140423b.zip	20.9 KB	04/23/2014	Vadim Nebogatov
vmn_upd20140429a.zip	227 KB	04/29/2014	Vadim Nebogatov
vmn_upd20140502b.zip	220 KB	05/02/2014	Vadim Nebogatov
vmn_upd20140502a.zip	192 KB	05/02/2014	Vadim Nebogatov
vmn_upd20140504a.zip	192 KB	05/04/2014	Vadim Nebogatov
vmn_upd20140504b.zip	221 KB	05/04/2014	Vadim Nebogatov
vmn_upd20140507a.zip	192 KB	05/07/2014	Vadim Nebogatov
vmn_upd20140507b.zip	221 KB	05/07/2014	Vadim Nebogatov