

Database - Feature #2092

add server-side database support for several builtin functions

03/15/2013 01:14 AM - Eric Faulhaber

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Costin Savin	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Conversion Support for Server Features	version:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 03/15/2013 02:06 AM - Eric Faulhaber

The following builtin functions are used within where clauses in queries and thus need to be supported as database server-side, user-defined functions, such that they can be converted to function calls in HQL:

```
to-rowid
length
weekday
chr
date
caps
left-trim
maximum
minimum
logical
num-entries
replace
```

To implement these, follow the example of my updates in bzr rev. 10289 to add similar support for the trim builtin function, which you can see with:

```
bzr diff -c10289
```

Or, if you use meld for diffing/merging files, this may be easier to see the differences clearly:

```
bzr diff --using=meld -c10289
```

Ignore my change to where_clause_post.rules, which was unrelated to adding support for the trim function.

#2 - 03/16/2013 10:24 AM - Costin Savin

- File *cs_upd20130316a.zip* added

Added proposed update,

For ROWID I didn't know how a ROWID data type is represented in the database so I assumed it's a String (in which case the input value doesn't need to be changed)

In some cases there were no equivalent function -date, minimum, maximum, logical so I implemented them directly in the *com.goldencode.p2j.persist.pl.Functions* (if needed they can be moved to other places).

PS:

In *p2jpl.ddr* I think there was a mistake for trim : instead of *com.goldencode.p2j.persist.pl.Functions.trim* should be *com.goldencode.p2j.persist.pl.Functions.trimWS* since *Functions* doesn't contain the first one. Corrected it

#3 - 03/18/2013 04:31 AM - Costin Savin

- Status changed from *New* to *WIP*

#4 - 03/18/2013 04:31 AM - Costin Savin

- Status changed from *WIP* to *Review*

#5 - 03/18/2013 06:34 AM - Costin Savin

After some tests it seems that min and max can be applied also to character and date variables, not only numeric values (but never mixed) and ROWID is actually represented as a numeric (Long) value in the database, I will correct these and merge to latest revision

#6 - 03/18/2013 08:23 AM - Costin Savin

- File *cs_upd20130318a.zip* added

Added merged update.

Are there any other functions that need to be added? I saw DATETIME and DATETIME-TZ which fit the profile

#7 - 03/19/2013 01:29 AM - Eric Faulhaber

Costin Savin wrote:

Added merged update.

Are there any other functions that need to be added? I saw DATETIME and DATETIME-TZ which fit the profile

Not at this time.

Code review 20130318a:

- Thanks for catching and correcting the trimWS mistake in *p2jpl.ddr*. Please note I made another mistake in not adding function *trimws(text)*; to the REMOVE section at the bottom of this file. You have replicated my mistake with all of your functions. Please add drop statements for *trimws* and all of your new functions to the REMOVE section. These statements should be at the same relative locations as in the top section.
- You have added an unsupported syntax to the *p2jpl.ddr* file. This file maps user defined functions implemented in PL/Java to the backing Java implementations. PL/Java does not support variable length argument lists as far as I know. So, something like create function *getMax(numeric...)* won't work. Fortunately, the most common use case (and all we need to support for now) is a 2-argument version of maximum and minimum. Please replace the *varargs* functions with 2-argument functions.
- *where_clause.rules* looks mostly correct from a functional point of view, except that you have moved the rule for *kw_can_do* under the function type check for rowid for some reason. Please put it back under logical. Note that we are maintaining alphabetical ordering of the rules by keyword, within a function type group. Also, you have made a bit of a mess of the formatting, in terms of adding extra lines and messing up the indentation in the rules, and changing the format style (and adding an unnecessary line) in the file header. Please straighten all this out.
- In terms of choosing names for the server-side functions, please be concise, and be consistent to the degree possible with the existing naming convention in use. For instance, *kw_logical* should convert to "toLogical" instead of "getLogical", *kw_date* to "toDate" instead of "getDate"; take your cue from the other data type conversion functions which are already there. For example, *kw_int*, *kw_dec*, and *kw_string* all have "toXXXX"

names, not "getXXXX". Please see the names I'd like to use below.

- Please clean up the formatting in p2jpl.ddr. The convention I have been using may not be obvious, but it is as follows: Group all overloaded functions together, each separated by a single line. Separate these groups from one another by two lines. Same convention applies to both the top and bottom sections of the file. Maintaining this allows an easier visual grouping of related features.
- Every Progress builtin function we support is implemented somewhere in the P2J runtime. All we are doing in Functions.java is creating a thin layer which accepts and returns J2SE data types and delegates the work to those existing implementations. You should not be re-inventing the implementations here. Use rules/convert/builtin_functions.rules to see how we convert each of these functions; this will lead you to the existing runtime implementation. For recently added functions, we may only have a runtime stub, but these will be implemented soon, and we only want to do this in one place.
- Please do not disrupt the organization of methods in a class; try to organize additions in a meaningful way. In Functions.java, for instance, you have dropped the new replaceAll method in the middle of a group of existing, overloaded methods with the same name (toString). These overloaded methods are meant to be organized together for ease of maintenance. By inserting an unrelated method in the middle of this grouping, you are making future maintenance more confusing.

Costin, I don't mind if you make logical mistakes and programming errors. Given the nature of this work and the complexity of the project, we don't expect you to get everything right away. But we do expect you to be organized and detail oriented. Avoidable errors in formatting and organization make a mess of the code, which makes future maintenance more difficult. Also, it makes code reviews much more time consuming than they need to be and delays updates.

Please be sure to review the changes you have made before submitting an update. A simple way to do this is by leveraging bazaar and a diff tool, like meld. I suggest you use something like `bzr diff --using=meld` to look at all of your changes side-by-side with the original before posting them.

Please remap the names as follows:

4GL	PL/Java	Functions.java
to-rowid	toRowid	toRowid
length	lengthOf	lengthOf
weekday	toWeekday	toWeekday
chr	chr	chr
date	toDate	toDate
caps	caps	caps
left-trim	ltrimws	ltrimWS
maximum	maximum	maximum
minimum	minimum	minimum
logical	toLogical	toLogical
num-entries	numEntries	numEntries
replace	replace	replace

I don't think any of the PL/Java names above conflict with standard SQL or HQL. If we find otherwise, we can adjust them accordingly.

#8 - 03/19/2013 02:29 PM - Costin Savin

- File `cs_upd20130319a.zip` added

Re-arranged the code , solved the reviewed problems and merged with latest updates.

Testcase used to check conversion result.

```
def temp-table wf1 field f1 as int.
for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.p-rowid = to-rowid(person.first-name)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE length(person.p-rowid,"character") > 8) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE weekday(person.hire-date) < 6) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE chr(person.hours) = "A") no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hire-date = date(person.hours, person.hours, pe
rson.hours)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hire-date = date(person.hours)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hire-date = date(person.first-name)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.first-name = caps(person.first-name)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.first-name = left-trim(person.first-name)) no-lo
ck:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hours = max(person.hours, person.site-id, person
.emp-num)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hours = min(person.hours, person.site-id, person
.emp-num)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(pers-addr WHERE pers-addr.active = logical(pers-addr.link-type, "yes
/no")) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hours = num-entries(person.first-name)) no-lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hours = num-entries(person.first-name, ";")) no-
lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.hours = num-entries(person.first-name, ";")) no-
lock:
end.

for each wf1 where wf1.f1 > 0 or can-find(person WHERE person.first-name = replace(person.first-name, "Joe", "
Lando")) no-lock:
end.
```

#9 - 03/19/2013 04:29 PM - Eric Faulhaber

Code review 20130319a:

- common-progress.rules: changes look good, but see comment below about max/min.
- where-clause.rules: logic looks good. One minor format problem remains: please either remove the extra blank lines you have inserted between some function type groups (i.e., lines 449, 456, 503, 514, 521) OR add a blank line between every group, but either way, be consistent.
- rowid.java: OK.
- Functions.java: better, but you are still implementing some functions "from scratch" instead of re-using what already exists in the runtime. For example, in toDate(Integer month, Integer day, Integer year), you are constructing a java.sql.Date using a deprecated constructor. You should instead be using the same idiom you are using elsewhere: new Date(new date(new integer(month), new integer(day), new integer(year)).dateValue().getTime()). This ensures we take advantage of the logic already in the com.goldencode.p2j.util.date class for things like time zone corrections, etc. Same thing with maximum and minimum: use the implementations we have in date, integer, decimal, etc. instead of using compareTo on the J2SE wrapper types. Rename method toCaps to caps so it matches the function declaration in p2jpl.ddr. Rename toWeekDay to toWeekday.
- p2jpl.ddr: much better, but still a few problems. Typo: "create function maximum(date, data)" ("data" should be "date"). In the bottom (i.e., REMOVE) section, please group the numEntries lines together with no blank lines, separated from the other groups with one blank line, so that the format is consistent with the groupings around it (sorry, I misstated above that this section followed the same format convention as the top section). The maximum, minimum, and toWeekday functions in the REMOVE sections should all be drop function ... instead of create function ..., and the same "data" typo exists here. Rename toWeekDay to toWeekday.

Your test case points out a problem with our conversion, in that it will convert cases of max and min with more than 2 parameters to server-side HQL, but we have no runtime backing for that situation. These cases will convert and compile, but will fail at runtime. Please make the following adjustment to common-progress.rules. Where we currently have:

```
...
oldtype == prog.kw_max      or
oldtype == prog.kw_min      or
...
```

replace it with:

```
...
((oldtype == prog.kw_max or oldtype == prog.kw_min) and
 this.numImmediateChildren == 2) or
...
```

and align the "or"s accordingly. This will ensure that only those min/max function calls in a where clause which have 2 parameters are converted to server-side HQL functions. Those with more than 2 parameters will convert to client-side where clauses with the min/max function calls. Add two more tests to your test case which use min and max with only 2 parameters each to make sure this works as expected. The 2-parameter tests should convert to server-side calls in the HQL; the 3-parameter tests should convert to an AdaptiveQuery with a WhereExpression reference in the constructor.

Please save your test case as a .p file with a meaningful name and check it into the testcases project under the uast directory.

#10 - 03/20/2013 05:35 AM - Constantin Asofiei

Costin: the test you've uploaded doesn't work with the repository p2j_test.df schema definition. When modifying the p2j_test.df schema, please try not to change the type/extent of the existing fields. If you need a new field, you can add new one, but by changing it you might break existing testcases.

#11 - 03/20/2013 11:55 AM - Costin Savin

- File *cs_upd20130320a.zip* added

Added proposed update, for the testcase it seems you can't define a field as rowid in 4gl so I modified that part.
Will commit the test case under /uast/dbfunc/db-server-side-functions.p (at the moment it seems I have some problems with bazaar and didn't manage to upload it yet)

#12 - 03/20/2013 01:29 PM - Eric Faulhaber

- File *cs_upd20130320a1.zip* added

Code review 20130320a:

This one is mostly correct, except for the following:

- Functions.minimum(Date var1, Date var2) was still using compareTo on J2SE wrapper types.
- The REMOVE section of p2jpl.ddr still contained create function toWeekday... instead of drop function toWeekday....
- a number of javadoc errors in Functions.java

Also, I made a mistake in note 7 above; you were right with your first update in terms of the naming of the weekday function. It should be getWeekday instead of toWeekday (to match, getDay, getMonth, getYear). Sorry for the confusion.

Rather than going back and forth yet another time, I just made these changes in the attached update, which is currently undergoing conversion regression testing. Please compare this update with your last one to see exactly what I changed.

#13 - 03/20/2013 02:38 PM - Eric Faulhaber

Update *cs_upd20130320a1.zip* has passed conversion regression testing and is committed to bazaar rev. 10301.

#14 - 03/22/2013 07:07 AM - Costin Savin

- File *cs_upd20130322a.zip* added

Removed the extra ' character from p2jpl.ddr at line 699

#15 - 03/22/2013 08:18 AM - Greg Shah

Go ahead and check this in immediately (and distribute it). It has no effect on conversion and it is already being runtime regression tested right now. We know this doesn't hurt and it definitely makes things better, so it is safe to commit.

#16 - 03/23/2013 09:49 PM - Eric Faulhaber

- Status changed from Review to Closed

- % Done changed from 0 to 100

#17 - 04/05/2013 04:13 AM - Costin Savin

It seems "replace" already is a function name in PostgreSQL. I will replace "replace" with "replaceText" and search for any other functions that may be keywords

#18 - 04/05/2013 04:47 AM - Costin Savin

- File cs_upd20130405a.zip added

Added update

#19 - 04/05/2013 06:04 AM - Costin Savin

- File cs_upd20130405b.zip added

chr also exist in postgres , renamed it to toChr, didn't find anything for the others.

#20 - 04/08/2013 11:41 AM - Eric Faulhaber

The changes in cs_upd20130405b.zip (a superset of 0405a) look good. Please update the file header comments to reflect the chr-->toChr change and check in the update.

#21 - 04/08/2013 01:22 PM - Costin Savin

- File cs_upd20130408b.zip added

updated file header comments

#22 - 04/08/2013 01:30 PM - Costin Savin

committed to bzs as revision 10336

#23 - 11/16/2016 11:06 AM - Greg Shah

- Target version changed from Milestone 4 to Conversion Support for Server Features

Files

cs_upd20130316a.zip	61.6 KB	03/16/2013	Costin Savin
cs_upd20130318a.zip	61.8 KB	03/18/2013	Costin Savin
cs_upd20130319a.zip	67.3 KB	03/19/2013	Costin Savin
cs_upd20130320a.zip	67.3 KB	03/20/2013	Costin Savin
cs_upd20130320a1.zip	65.6 KB	03/20/2013	Eric Faulhaber
cs_upd20130322a.zip	3.42 KB	03/22/2013	Costin Savin
cs_upd20130405a.zip	12.6 KB	04/05/2013	Costin Savin
cs_upd20130405b.zip	12.6 KB	04/05/2013	Costin Savin
cs_upd20130408b.zip	12.6 KB	04/08/2013	Costin Savin