

Database - Feature #4613

add missing support for database user defined functions (UDF) if they are supported in runtime

04/14/2020 06:58 PM - Eric Faulhaber

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		version:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to Database - Feature #3813: misc DB features part deux			Closed

History

#1 - 04/14/2020 09:15 PM - Eric Faulhaber

- Subject changed from add missing support for built-in functions in WHERE clauses, if they are supported in runtime to add missing support for database user defined functions (UDF) if they are supported in runtime

FWD supports many built-in functions for use in WHERE clauses, such that they are executed at the database server using technologies such as PL/Java (PostgreSQL) or Java function aliasing (H2). These are known as user-defined functions, or UDFs. Over time, support for built-in functions as UDFs has lagged their implementation in the FWD runtime environment. Many are implemented as UDFs, but there are a number missing.

To rectify this situation, **the first step is to identify which built-in functions are implemented in FWD, but which are missing UDF support.** Certain functions need to be supported as UDFs; others can just be evaluated on the application server as regular, runtime, built-in functions, and the result passed to a query as substitution parameters. Generally speaking, the following characteristics determine that a function needs to be supported as a UDF:

- the function references (i.e., has a parameter that is) a buffer/table or can reference a buffer field, whose value needs to be determined at the database;
- it does not require access to state that only exists at the application server.

If a built-in function can be evaluated without data from the database, it will be referenced from a converted WHERE clause as a query substitution parameter.

To identify which built-in functions meet the above criteria and have not been implemented yet, start with rules/gaps/expression.rules. Look at the lines that begin func.put(... Any function which we currently support at a level higher than cvt_lvl_none and rt_lvl_none is a potential candidate.

Those which have been implemented as database UDFs will be found in rules/annotation/where_clause.rules, with the comment, "functions (including operators implemented as functions) executed on the server", and the test getNodeBoolean("server_op"). The code here injects the UDF alias for a given built-in function into an HQL/FQL where clause.

Next, look at the Java runtime class persist.pl.Functions. This class defines static Java methods which form a "shim" layer API to the backing runtime implementations for the built-in functions. The UDF aliases injected by where_clause.rules are the short names of these static methods. The implementations of these methods are generally very simple, and serve only to invoke the backing runtime method which performs the work of the built-in.

However, not all built-in functions reflected in rules/gaps/expression.rules should be implemented as UDFs. Once a built-in function is found, which has higher than cvt_lvl_none/rt_lvl_none support and which is not implemented by where_clause.rules and the Functions class, it needs to be examined to determine whether:

1. it can be invoked within a WHERE clause; and
2. it meets the criteria for a UDF as defined above.

Candidate built-in functions which meet all these requirements, and which are not yet implemented as UDFs should be added to a list of missing UDFs.

Certain database-related built-in functions do not need to be vetted; we already have determined they will not be supported as UDFs:

- AVAILABLE

- AMBIGUOUS
- CURRENT-CHANGED
- LOCKED
- NEW

#2 - 06/08/2020 07:37 AM - Greg Shah

- Related to Feature #3813: misc DB features part deux added

#3 - 08/13/2020 10:52 AM - Greg Shah

Some items mentioned in [#3813](#) should be handled as part of this task:

Built-In Function Usage in WHERE Clauses

New use cases such as ROW-STATE and VALID-OBJECT need to be handled.