

## Database - Feature #7447

### Compare ValueString and ValueStringIgnoreCase faster in FWD-H2

06/19/2023 06:46 AM - Alexandru Lungu

<b>Status:</b> Closed	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b> Dănuț Filimon	<b>% Done:</b> 100%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	<b>version:</b>
<b>billable:</b> No	
<b>vendor_id:</b> GCD	
<b>Description</b>	
<b>Related issues:</b>	
Related to Database - Feature #7363: Improve H2 Value caching, hashing and eq...	<b>Test</b>

#### History

##### #1 - 06/19/2023 06:57 AM - Alexandru Lungu

- Tracker changed from Bug to Feature

There are several hits of Value.compareTo, where the operands are ValueString and ValueStringIgnoreCase. This is mainly because JDBC doesn't allow setStringIgnoreCase, so FWD-H2 is forced to "cast" the string parameters to ValueStringIgnoreCase each time they are used (usually on index look-up) against an ignore-case column.

Value.convertTo from a ValueString to ValueStringIgnoreCase implies the construction of a ValueStringIgnoreCase:

- cache look-up for String and Value;
- in case of cache miss, a ValueStringIgnoreCase is built
- the hash of the string is computed

All this work is done, so that ValueStringIgnoreCase.compareToSafe can do mode.compareString(value, ((ValueStringIgnoreCase) o).value, true). Having mode.compareString(value, ((ValueString) o).value, true) would produce the same result - they have the same underlying String.

I suggest the following solution:

- If Value.convertTo has forComparison flag set on true, the ValueString will not convert to ValueStringIgnoreCase, but stay in place. The same goes for ValueStringIgnoreCase into ValueString
- ValueString.compareToSafe and ValueStringIgnoreCase.compareToSafe will be sensitive to the other data type. Because, ignore-case has higher order, if any of the operands is ignore-case, the comparison will be ignore-case.

This change will optimize the number of Value instances created, cache look-up + override, string equals and caches.

**#2 - 06/19/2023 07:04 AM - Alexandru Lungu**

- Related to Feature #7363: Improve H2 Value caching, hashing and equals added

**#3 - 07/05/2023 03:34 AM - Alexandru Lungu**

- Assignee set to Dănuț Filimon

- Status changed from New to WIP

Danut, please check if this is still happening after your changes in [#7454](#). Of course, before the changes, there were a lot of such comparisons. I don't know if this task is relevant after the changes there. Please check if there are still comparisons between `ValueString` and `ValueStringIgnoreCase`.

**#4 - 08/28/2023 09:36 AM - Dănuț Filimon**

Alexandru Lungu wrote:

Danut, please check if this is still happening after your changes in [#7454](#). Of course, before the changes, there were a lot of such comparisons. I don't know if this task is relevant after the changes there. Please check if there are still comparisons between `ValueString` and `ValueStringIgnoreCase`.

I retested using an available customer application and there were no comparisons between `ValueString` and `ValueStringIgnoreCase`. The testing is really slow when using a conditional breakpoint in `Value.compareTo`, so I did not browse the application too much (~30 min).

**#5 - 08/29/2023 03:29 AM - Dănuț Filimon**

Great news! I retested the customer application using a different approach which allowed me to test it a lot faster and found no comparisons between `ValueString` and `ValueStringIgnoreCase`.

**#6 - 08/29/2023 03:32 AM - Alexandru Lungu**

- Status changed from WIP to Review

- % Done changed from 0 to 100

Ok, this can be closed then - let me know if there is something left here to discuss

**#7 - 08/29/2023 06:57 AM - Greg Shah**

- Status changed from Review to Closed