

## Database - Bug #5376

### Reintroduce UDFs for handling special date-related operators

05/17/2021 05:28 PM - Ovidiu Maxiniuc

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

#### History

##### #2 - 05/17/2021 06:14 PM - Ovidiu Maxiniuc

The investigation from #5325 lead to conclusion that in the current revision FWD lacks support for some + / - operators when the operands are some kind of date. As part of that task, I learned that H2 and PGSQL natively support only one operation: date +/- integer but not date +/- <other-numeric-type>. Injecting a cast to integer type on the second operand allowed the operations to be correctly performed and luckily fixed the cause of issue in that task.

However, the general case is not done because:

- the cast injection solution was not tested on MSSQL dialect. A quick investigation revealed DATEADD function for date arithmetic. Since this exists it is possible that the + operator might not work natively with MSSQL;
- only the date + numeric-type is handled. There is a second operation datetime(-tz) + numeric-type. In this case, in 4GL, the unit for second operand is the millisecond. This case was not covered by #5325. The investigations revealed that:
  - H2 supports such operation but the unit of the second operand is 1 day. I have no confirmation, but I expect that passing a fractional part (if present) represents the time of the day. For example, adding 2.5 means 2 days and 12 hours. If this is true, a conversion to a day unit should be straightforward;
  - PGSQL seems to support a more complex date and time operations. It uses a special datatype interval which can be measured in different UMs. This datatype allows other operations like \* and / to be used also;
  - MSSQL: as note above, this dialect features a very specific function to date-related operations and seems, by far, the most exotic one.

As conclusion: except for the basic operations already implemented as part of #5325, all dialects support date/time operations but in different ways. This a good part: in order to avoid the UDF written in Java which may affect the performance of the query because of their opacity to the query planner we can write dialect specific, low-level implementation for these operators.

If this is too complicated (to write and maintain) we hill have to finally rely on old-good Java UDF implementations.