

Database - Feature #5967

better handling of errors received from SQL/database server

01/12/2022 11:58 AM - Ovidiu Maxiniuc

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

History

#2 - 01/12/2022 12:25 PM - Ovidiu Maxiniuc

In one of previous issue we encountered some issues with the disk space being filled up on the partition where the SQL server stores the log files. The reason was that the application entered a loop where a critical error (reported as a SQLException) was treated as a recoverable error and FWD re-issued the specific (malformed) query countless times, which was logged into database log file. Due to avalanche of the queries and the details of individual log entry, the disk partition which also holds the data quickly filled up, bringing SQL to a complete halt.

Although this was a side effect and was remedied as part of respective task, we still need to do a better job handling non-recoverable SQLException cases from other places. The main mean to do this is to do in depth investigations on the SQLState 5-character field of the exception. The problem is, although somewhat standardized and split into several return error classes, each RDBM is free to define additional values for it to handle those features which are beyond the standard. As result, the investigations must be implemented for each dialect, in order to decide the gravity of a specific error and the best option to handle it.

The specific vendorCode (obtained using SQLException.getErrorCode()) can be used in same way, but I expect the stored data to be similar.

To aid these, as a last resort, the GET DIAGNOSTICS might come in hand, but FWD does not support it right now.

Useful links:

- general SQLSTATE info on [Wikipedia](#) ;
- [PostgreSQL](#) specific codes;
- [H2](#) specific codes;
- Microsoft's [SQL Server](#) specific codes.