

Database - Bug #8683

SQL operations not saved when session is closed

04/25/2024 07:04 PM - Ovidiu Maxiniuc

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

History

#2 - 04/25/2024 07:17 PM - Ovidiu Maxiniuc

In a customer application we encountered a case when SQL statements which apparently were executed were not actually committed.

The problem was identified to be operations on `_meta` tables, (not limited to `_file`, `_field`, `_index`, including the VSTs) not being performed using dedicated connections, but using a shared connection with the business/converted code. In this particular case, the 4GL code:

- deletes the physical table from mutable database;
- uses `SchemaCheck` to remove it from ORM managers, including metadata (via `MetadataManager`, which executes the `DELETE` statements);
- closes the connection.

The last step is performed outside of any business-level transactions. If there was one, we would not observe this problem. Anyway, this is the normal way of operation for all `_meta` operations, so they are not affected by business rollbacks/commits.

The result was that some records which were apparently deleted survived and provoked some unique index validation failures at the attempts to re-create those records.

So, I am thinking whether we should be doing this on a dedicated connection, maybe with `autocommit`? This increases the complexity of persistence in FWD and need re-synchronization between the connections.

As a temporary solution which was working in that particular case was to force a commit before closing of connections without `autocommit`. Because we do not have enough information to always commit at this point in the life cycle of the connection the solution was reverted.

Alternatively we can do that commit only for connections of `_meta` databases, but this is not much different than the solution above.