

Database - Bug #8589

Deprecate the shared temporary database completely

04/05/2024 03:05 AM - Alexandru Lungu

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

#1 - 04/05/2024 03:45 AM - Alexandru Lungu

In the original support for `_temp` database, one single in-memory database was created. All sessions used that single instance with LOCAL TEMPORARY tables. Back then, it was safe to have a long living database in which session would do their clear-up. However, due to the architecture of H2, there were still synchronizations done even for different sessions (mostly locking the meta tables, shared cached, etc.). At that point, GCD team wasn't mostly into changing the H2 internals. Thinking about it today makes me think that "shared" mode is still viable within FWD-H2.

Anyway, with recent changes to H2, the private `_temp` databases option was introduced: one session - one database. This allowed the usage of NO_LOCK and disable any synchronization features for better performance at memory expense. Also, with very latest changes, the direct-access layer was introduced, which allowed accessing the `_temp` database resources directly (faster queries without SQL).

AFAIK, everything should still work for shared `_temp` database, but from my POV this is a burden to support. Starting with #8582, there were differences in how we should differently support the features. Also, performance-wise, shared `_temp` database is slower (although more memory efficient). Testing nevertheless is more complex.

My POV is that private `_temp` databases should be the default and keep shared database as a possible `directory.xml` possibility, but acknowledge its possible drawbacks.

#2 - 04/05/2024 06:11 PM - Eric Faulhaber

Unless there is a *significant* memory savings or some other substantial advantage of shared mode, I think we should remove shared mode entirely.

If we leave it there and make it a configuration option, even if it is no longer the default, it will remain a burden to support. We will still have to maintain and test it, forever.

Greg and I discussed this and he disagrees, but I think we need to understand how much memory is at stake here.