

Database - Feature #4692

add sanity check at server startup to compare DMO indices with database indices

06/22/2020 12:27 PM - Eric Faulhaber

Status: New	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	version:
billable: No	
vendor_id: GCD	
Description	
Related issues:	
Related to Database - Feature #4011: database/persistence layer performance i... Closed	

History

#1 - 06/22/2020 12:38 PM - Eric Faulhaber

Branch 4011a eliminates the collection of index metadata from the database at startup to populate IndexHelper data structures. Instead, all index information is collected from DMO annotations.

However, if the database's indices get out of sync with the DMO annotation indices (which are only written at conversion time), we will not know it. This can cause performance problems. I'm not sure it will cause functional problems, but that may be a possibility as well.

I have left the index metadata collection logic intact (see `persist.IndexMetadataCollector`) in 4011a, but it is not used currently. At some point, we will want to re-enable this collection and add some logic to compare the metadata collected from the database with the index information collected from the DMO annotations, in order to report discrepancies. Additional indices added to the database, but not expected by converted code, should be ok (aside from the overhead of more indices to update/maintain). However, indices removed or changed at the database, which converted code expects to be in the state they were at conversion time, could cause serious performance problems.

#2 - 06/22/2020 12:48 PM - Ovidiu Maxiniuc

Additional indices are OK as they could improve performance (if we spot some series of queries that frequently use a set of columns). However, they must NOT be unique! Adding such constraints might block storing valid records on SQL.

There is another thing we might want to check: the mandatory/not null columns. Removing this attribute is not immediately an issue, as the FWD runtime will validate the records on save, but if somebody messes with the database and adds not-null constraints, again, valid records will not be accepted by SQL.

#3 - 06/22/2020 01:40 PM - Eric Faulhaber

- Related to Feature #4011: database/persistence layer performance improvements added

#4 - 06/22/2020 01:42 PM - Eric Faulhaber

Good point. We should add a comparison of legacy "mandatory" fields with database columns marked "not null" as part of the sanity check.