forceLiterals vs forcePlaceholders

This post is inspired by the feature Enabling index hints in X++ again

<https://docs.microsoft.com/en-us/dynamics365/unified-operations/fin-and-ops/get-started/whats-new-platform-update-23#enabling-index-hints-in-x-again>

while discussing this the question were asked – do we really need “index hints” for the modern SQL Server or not.

First of all some theory about SQL statement execution

In D365 we have 2 way of executing SQL statement

**forcePlaceholders**

This is the default for SQL Statement. In this case every SQL Statement that you write in X++ processed in 2 steps – first sp\_prepare command executed with placeholders instead of actual values and the result of this sp\_prepare is the cached plan. Then sp\_executeSQL in called that executes the statement using the plan and actual values. For one statement with the different values plan is calculated only once.

How does the SQL server knows what plan to generate in sp\_prepare if the statement in new(for some values one plan can be more efficient than another)? In this case SQL uses “parameters sniffing” feature. If it gets sp\_prepare command for the unknow plan, it doesn’t execute it immediately, instead it waits for the first call of sp\_executeSQL and uses the first call values to calculate the plan.

**forceLiterals**

In this case your actual SQL statement (with the actual values) is send to SQL Server and executed as is. The drawback is that for every set of values you will need to calculate a different SQL plan.

**Parameters sniffing**

Parameters sniffing quite often becomes a reason for the performance problems. Let’s say you have warehouse where most of the items have 1 batch ID, but there are some Items and batches with general name (like “No batch”, “Empty” ). In this case when you query item by batch your actual plan is depends of first query values. If you query for the “general” batch values, SQL Server will creates a plan that starts execution with the ItemId as “general batch value” is not selective in this case. But for the most of your items it will be wrong and gives huge overhead as in most cases Batch is a unique value.

How to resolve a Parameters sniffing issue

There is no universal way to resolve Parameters sniffing issue(in some blogs ), but there are several ways to deal with it

* Create new indexes – you can create new indexes to help SQL server always choose the best plan. This often works only if you have conditions to one table only
* Use sp\_create\_plan\_guide command to force the actual plan – using this path creates a big admin overhead as you need maintain these created plans. In case you add new field in AX table you need to change all plan guides that use this table
* forceLiterals hint – in this case you send an actual values to SQL Server and it choses the best plan for these values. The overhead in this case that plan needs to be calculated each time
* Use index hint (new feature in PU23) – it is the same as sp\_create\_plan\_guide but with no admin overhead

Let’s check what is the overhead of forceLiterals usage