

microsoft.public.sqlserver.programming: Re: Help me convince the dev manager. Please.

Re: Help me convince the dev manager. Please.

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-05/2778.html>

From: songie D (*songie_at_D.com*)

Date: 05/12/04

Date: Wed, 12 May 2004 20:17:19 +0100

Dynamic SQL's performance is a whole lot worse than stored procedure SQL – not just in the fact that it's not precompiled, but also due to the fact that it has a tendency to have DBAs complain of the generation of huge log files which dynamic SQL apparently causes SQL server to do.

It should be treated like cursors – useful at times, but used as sparingly as possible. But like others have said, there's very few database scenarios which actually NEED it, if the database is correctly structured. The 'need' for it is more likely to arise because a database project has been driven down the road of a bad design plan that can't be reversed down and needs new features adding. It sounds like this is the unfortunate scenario in your case. If it is, perhaps you should do as suggested and post what the dev manager's solution is, and what yours is, and let the group compare them?

"Rohan Hattangdi" <RohanHattangdi2003@hotmail.com> wrote in message news:OIouKJFOEHA.2480@tk2msftngp13.phx.gbl...

> All,

>

> *I cannot believe that I am posting about this again.*

>

> *I just came back from a meeting with the dev managers here and one of the dev managers insists that there must be another method out there to handle optional parameters. I have gone over the article by Erland Sommarskog and,*

> *both in conjunction with the article and via independent research, and through personal experience, firmly believe that dynamic SQL is the best way*

> *to handle optional parameters.*

>

> *There are only three options – dynamic SQL, static SQL and a hybrid solution. I keep mentioning that dynamic SQL is the industry-standard and the preferred method for handling optional parameters but the response is always that there must be a better way to handle this because dynamic SQL makes code difficult to read and to debug. I came back with the fact the dynamic SQL code is not that hard to read if structured correctly (code style and standards) and is easy to debug if all parameters are passed in*