

The Curse of the Cursors: Why You Don't Need Cursors in Code Development

We frequently encounter developers who love to use cursors throughout their code, but these are an impediment to SQL database performance.

You really don't need cursors in your code. Here's why.

A cursor is typically used in databases to identify a record location. As Techopedia explains, when a file is open, a cursor, "points to the first record in the file, and using various commands the cursor can move to any location within the file."

This uses memory. Although the amount of memory used is very small, in aggregate all these cursors—especially unclosed cursors—can accumulate and translate into performance issues.

What's Wrong with Using Cursors?

We are not exaggerating when we say that we often have people ask, "What's wrong with 2 or even 3 nested layers of cursors?"

Later, the same people will come to us asking, "Why is this database running slow?"

When we then ask them whether they used cursors, the response is "Yes, I did."

The issue is that those developers who want to use cursors are often coming into SQL development from the Oracle world, where cursors are used. In SQL, they are not used.

There are other many other options. Sean McCown, writing in InfoWorld, suggests using temp tables to improve cursor performance, if you must use them at all. He observes:

Cursors not only suffer from speed problems, which in itself can be an issue with many operations, but they can also cause your operation to block other operations for a lot longer than is necessary. This greatly decreases concurrency in your system.

Implement Written SQL Standards

We like to have written requirements as well as written SQL standards to avoid these types of issues in database development. This includes information on how to put together a table, how to put together stored procedures, and things to avoid when you're developing. We include "no using nested cursors" as a coding standard, for example.

Having written SQL standards is key to a better development process. When we are in a scrum meeting or code review, we can refer to those coding standards and ask a team member, "Give me a good reason why you need to use a cursor here." or "Why did you code with the cursor like this? Is there any other way that you could have done this?" Typically, this diffuses any conflict.

This process makes the developer stop and consider whether there were other options. The process works particularly well when working with junior developers. These types of questions guide them such that they can articulate their own solution, whether it involves a cursor or table look up.

McCown suggests that when you must use a cursor, developers can mitigate some of the performance issues by putting the data into a temp table for comparison rather than using the live table. He adds:

You almost never need to use a cursor. There's almost always a set-based solution; you need to learn to see it.

Improving Database Performance

Eliminating cursors is just one way to improve database performance. For some other recommendations, be sure to download a copy of the whitepaper, [The 5S Approach to Improving Database Performance](#) and read our blog for SQL and DBA information.

If you would like assistance implementing and integrating any of these ideas into your operations, Datavail can work with you and your organization to effectively leverage SQL for your benefit and that of your clients and customers.

To learn more about our remote database services and how our experts can help with your ongoing SQL operations, please contact Datavail to discuss a custom solution designed for your enterprise.

About the Author: Rick Roberts

Senior SQL Server DBA/Developer, Datavail

Rick has been working with SQL Server for 12 years and still enjoys every minute. He loves implementing new SQL Server functionality and can't wait for the next SQL Server version to be released. Rick has developed and maintained Transactional & Data Warehouse database systems and has worked with SQL Server 2000 through 2014. Rick's passions are Performance Tuning and working in the Microsoft BI stack. In his off-hours, Rick enjoys hiking, biking and is planning on taking up fly fishing in the coming year.