

(September 12th, 2016)

If you know someone who would benefit from being an Insider, feel free to forward this PDF to them so they can sign up [here](#).



Quick Tips for our Insider friends!

Hey Insiders,

This newsletter is coming to you from Bellevue, WA where we've just started the first day of our IEPTO1 class. We're doing IEPTO1 this week, IEPTO2 and our new Advanced SSIS class next week, and then we're off to Dublin. Incidentally, if you're in Europe, the Dublin IEPTO1 class in October is the **last time we'll be teaching in Europe until 2018** – see [here](#) for details.

SQLintersection session and workshop details have been posted; we have an exciting line-up scheduled for our show! If your team needs architectural advice and sessions from speakers who not only know their technology but know how to convey it – this is the place to be! Be sure to use the 'SQLskills' discount code to save \$50 on registration. Check it out [here](#).

We've decided to make our special offer **single-instance health check for only US\$2,500 the permanent price for the first health check for new clients**, which is more than 1/3 off the original price! Details about why our health checks are so cost-effective are [here](#).

The latest book I've read is Steven Erickson's [Deadhouse Gates: A Tale of the Malazan Book of the Fallen](#). This is the second book in Erickson's epic 10-volume fantasy series. The overall story is very involved, with magic, immortals, empires, battles, and lots of intrigue and the book throws you right in from page one. The book easily stands on its own, but is better read after the first one. All the books are 700-1000+ pages, so a real treat to read. Highly recommended!

Note: you can get all the prior Insider newsletters [here](#).

The Curious Case of...

This section of the newsletter explains problems we've found on client systems; they might be something you're experiencing too.

A quick one this time: Erin was working with a large client recently to track down the cause of a configuration issue they were seeing.

The DBA had increased the max server memory for the instance, to allow the buffer pool to be larger to hold more of the workload data in memory, but the amount of memory being used by the instance would not increase.

It turned out that the DBA had also turned trace flag 834 to enable large pages (see [this blog post](#) from Bob Ward for more details). When this trace flag is enabled, SQL Server's memory is allocated during instance start-up, and if the max server memory setting is raised, additional memory will not be allocated until the next time the instance is restarted.

If someone is going to enable a trace flag, make sure that all the (potentially detrimental or non-intuitive) side effects of the trace flag are understood.

Paul's Ponderings

While my first cup of coffee was brewing on Saturday morning, I quickly checked my email and saw an interesting question, which I'll paraphrase here:

To be able to tune the performance of a stored procedure, is it necessary to understand all the code in the procedure?

And then I went outside to join Kimberly in the hot tub and we got into a debate about the answer. Can you guess what conclusion we came to?

My first thought was that you at least have to know the aim of the procedure, such as roughly how many rows it should return, so you can judge whether the query plan looks pretty optimal for the goal of the procedure, and to do that you'd need to understand some of the code.

Kimberly countered that some simple troubleshooting could help you to understand *what the problem might be* by saying you could add `OPTION (RECOMPILE)` to offending statements, which she discusses in [this post](#). If the performance and the plan shape changes drastically (because the statement gets a new plan each time it's executed), then you'd know it's a parameter sensitivity issue. Technically, that wouldn't require any understanding of the procedure's code and would prevent parameter sensitivity problems from using a plan that worked well for some parameters but not others.

She cautions that this should not be a long term solution but it might be a good and very simple test to get a feel for where the problem is. Permanently fixing it might mean leaving `OPTION (RECOMPILE)` in the procedure or it might mean some code changes so that you don't recompile *every* time. Kimberly has another post that addresses some of the ideas and options [here](#).

But what if you're trying to troubleshoot something like excessive `PAGEIOLATCH_SH` waits coming from the procedure's execution? You need to know whether it's expected for the procedure's plan to be driving reads or not – so you do need to have *some* information...

Kimberly made a further point that sometimes if a procedure has grown very complex over time (sometimes years of change after change after change), it may be better to spend time understanding all the goals of the procedure and try writing the functionality from scratch. This

is especially true if the code is old and manually implements some functionality that is now natively present in T-SQL, like windowing functions, for example.

We debated some more and agreed that although it's usually not necessary to understand every line of code, usually, it *is* necessary to understand the aim of the procedure and metrics like how often it will be executed.

I reckon that most of you probably guessed that we'd come to the conclusion "it depends", but our conclusion is "generally, no, most of the time you don't need to understand all the code in a procedure to tune it".

Call to action: as with any performance tuning, the more information you have about the problem and what it is you're tuning, the more easily you'll be able to tune. Understanding what problem you're trying to solve is the key to efficient performance tuning – I don't think it's worth the time poring over every line of a procedure before tuning it as there are many things you can do with very limited knowledge of what the procedure's code is going.

Video Demo

In this demo Jonathan takes a look at the really cool new index analysis feature in SQL Sentry's Plan Explorer and how to use it to gauge the effectiveness of index changes during query tuning. (And by the way, just this week **SQL Sentry announced that Plan Explorer Pro is now completely free** too! See [here](#) for details.)

The video is just over 10 minutes long and you can get it:

- In WMV format [here](#).
- In MOV format [here](#).

The demo code is available [here](#).

Enjoy!

SQLskills Offerings

We will be announcing our 2017 line-up in our next newsletter so stay tuned! As for 2016, our classes are all open for registration (listed below), including three new classes in Chicago in November added because of popularity. We hope to see you at a class or a conference this year!

To help your boss understand the importance of focused, technical training, we've also added a few items to help you justify spending your training dollars with us:

- [Letter to your boss explaining why SQLskills training is worthwhile](#)
- [Community blog posts about our classes](#)

- [Immersion Event FAQ](#)

Upcoming Immersion Events

Bellevue, WA

- **IEPTO1:** Immersion Event on Performance Tuning and Optimization – Part 1
 - September 12-16
- **IEPTO2:** Immersion Event on Performance Tuning and Optimization – Part 2
 - September 19-23
- **IESSIS2:** Immersion Event on Advanced SQL Server Integration Services
 - September 19-22

Dublin, Ireland (*returning to Europe in 2018 NOT 2017*)

- **IEPTO1:** Immersion Event on Performance Tuning and Optimization – Part 1
 - October 3-7

Chicago, IL

- **IE0:** Immersion Event for Junior/Accidental DBAs
 - November 7-9
- **IEPTO1:** Immersion Event on Performance Tuning and Optimization – Part 1
 - November 7-11
- **IEPDS:** Immersion Event on Practical Data Science
 - November 7-11

Click [here](#) for the main Immersion Event Calendar page that allows you to drill through to each class for more details and registration links.

Summary

We hope you've enjoyed this issue - we really enjoy putting these together.

If there is anything else you're interested in, we'd love to hear from you - [drop us a line](#).

Thanks,
Paul and Kimberly

Paul@SQLskills.com and Kimberly@SQLskills.com