

(July 18th, 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 somewhere on the west coast of Norway where we're driving around the spectacular scenery for a couple of weeks before heading up to Svalbard to photograph polar bears and walruses. Expect to see a few blog posts with photos!

Note: to allow for everyone's summer fun, this is the last newsletter until August 29th.

Our September classes in Bellevue and Fall classes in Chicago are filling up, so be sure to book your Immersion Event in advance to avoid the class you want selling out – all the schedule details are [here](#).

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](#).

Three books for you this time, to make up for the newsletter break over the summer:

- Robert Ferguson's [The Vikings: A History](#). Although I bought this a few years ago, this book bubbled to the top of my reading list in preparation for the trip we're currently on. It does a great job of explaining why the Vikings came about, their culture, their spread through Europe and into Iceland, Greenland and North America, and their eventual demise as a people and culture as they become assimilated into Christian society in the 1000s. I've also been watching the [Vikings](#) TV series, which has tied in nicely with historical explanations from the book. Recommended.
- Graham Greene's [Journey Without Maps](#). The only other Greene book I've read is the classic [Our Man in Havana](#), for English class in high school (which I remember enjoying), so I didn't know what to expect from this nonfiction travel book. It covers Greene's travels through Liberia on the west coast of Africa in the 1930s. The country he traveled through was mostly non-colonial and tribal, so it was very interesting to read about the various customs and compare them with what I've read of travels around the

same time in other tribal areas (e.g. in Thesiger's [The Marsh Arabs](#) or Mytinger's [Headhunting in the Solomon Islands Around the Coral Sea](#)). Recommended, but only for fans of historical travel writing.

- George Steinmetz' [Desert Air](#). Steinmetz is an aerial photographer who specializes in deserts and this book is a visually stunning collection of photography of major deserts and features in them from across the world. Other books of his I've read are [African Air](#) and [Empty Quarter: A Photographic Journey to the Heart of the Arabian Desert](#), and I highly recommended all of them.

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.

Jonathan was working with a long-term remote DBA client a few weeks ago and came across an interesting deadlock scenario (he's renowned for enjoying deadlocks!) that neither of us had seen before.

The scenario involved two tables that alternated between being the table written to and the table read from, with ReadTable and WriteTable synonyms that alternated at fixed intervals, and the write table being truncated as part of the synonym swap.

As their load increased, they were seeing deadlocks involving schema-stability locks and schema-modification locks from the synonym code and the truncate code.

It turned out to be a race condition that became easier and easier to hit with higher load, and simply moving the table truncation to before the synonym swap eliminated the deadlock.

It's always worth understanding the root cause of a deadlock rather than applying some sledgehammer fix in the hope that it'll solve the deadlock. Often, simply reordering logic (where possible) can prevent the deadlock happening.

Paul's Ponderings

In this newsletter I was going to write about the pros and cons of upgrading, but went back through earlier newsletters and found an editorial on the subject from exactly five years ago, when we were looking forward to the release of SQL Server 2012, so I'm re-running it here, with additional comments.

Enjoy!

In another departure from my technical content, I blogged last week about a nasty trend I see online – berating people for simply not knowing about something rather than taking the time to explain nicely to them. You can read about it my post [Ignorance is not stupidity](#), which I followed up with a post discussing the pros and cons of regular reboots – see [here](#).

[Update 2016: I think this has lessened a bit, at least on Twitter (as I don't do forums any more). I still start out every class saying that there's no such thing as a stupid question, as everyone in the room started with zero knowledge about SQL Server.]

In this newsletter I decided to talk about upgrades, prompted by the upcoming release of SQL Server Denali.

I must admit that I'm pretty much a Luddite when it comes to most forms of technology (but I don't go as far as the group from which the word derives – see Wikipedia [here](#) – I tend to build cool techie stuff – see [here](#)). It wasn't until October 2010 that I moved off of Windows XP, to many people's amusement—including Kimberly, as there was no pressing need for me to do so. It just worked. There were no driver problems any more, all my software worked, and it was very stable.

[Update 2016: And the Luddite trend continues. I'll continue using Windows 7 until I absolutely have to move to something else. It works, it's stable, and it does everything I need to work efficiently. We also have a bunch of dive gadgetry that doesn't have drivers for newer versions of Windows. At least I got a new laptop last year...]

The major reason for holding back was I used Outlook Express and couldn't find a replacement for it to use on Windows 7—and there was no way I was going back to full-blown Outlook, which I think is just horrible. Then Kimberly decided to upgrade our laptops and I found the Thunderbird mail program and so I upgraded—and now deal with driver issues and OS bugs. Sigh.

[Update 2016: At least Windows 7 is nice and stable now. And I'm very happy to not be using Outlook, as I repeatedly see people having issues with it not syncing, or hanging, or messing up address books...]

And what about cell-phones? I swore I would never get an iPhone—until my cell-phone died finally in New York in January 2010 and I *needed* a new phone there and then. So I caved in and got one—again to the amusement of many. ☺

[Update 2016: Well, it did take me three years to move from my iPhone 3 to an iPhone 5S, and only so I could do Facetime with our kids. But now that I have reading glasses, I think I need to get a large iPhone 6. Getting old sucks...]

I find it strange that people are desperate to get their hands on the newest, latest, coolest software or gadget—and have to spend hours learning a new interface and new buttons and dealing with bugs and glitches. Maybe I'm just not hip any more...

[Update 2016: Apparently I'm still not hip...]

SQL Server is interesting though, because apart from people tinkering on their own laptops, it's usually installed in a business environment. Upgrading a server product on which your business is depending has to be done with great care, and with a lot of consideration.

For instance, when a product is first released, there are bugs in it (true of **any** software product) – do you want to be the one to hit a bug that interrupts your business? Do you really need the expense and risk of moving to a new release ASAP just because it's the new release?

[Update 2016: Now this is interesting, because the prevailing wisdom for many has been to wait for the first Service Pack to come out, but recent SP1s have introduced major bugs. I covered that in the [June 22, 2015 newsletter](#). So, there's always the potential for bugs in every release; it only further solidifies the need for adequate testing before ANY major migration!]

On the flip side, each new release has some really cool features that could really help your business and make the risk and expense of upgrading quickly worthwhile. In SQL Server 2008 it was data and backup compression that were my favorites; in Denali it's going to be the new AlwaysOn features with truly usable “read-only secondaries”.

[Update 2016: For SQL Server 2016, I think the coolest new feature is Query Store, which is covered in the demo video this time. However, I think the best thing that the SQL Product group has done for this release is to invest a bunch of time in fixing scalability problems in the existing codebase. See [here](#) for the blog posts describing these.]

And then there are corporate policies and budgets to think about. Even though you may want to upgrade, you may not be able to because of budgetary constraints. Or you may not want to upgrade but you **have** to because it's corporate policy.

Upgrading is a bit of a minefield IMHO as you also have to be very careful that if anything goes wrong, you have the option of going back. Admittedly, it has become easier over the years with the Upgrade Advisor tool and comprehensive whitepapers to help you plan and think of gotchas. Still, you need to do a lot of testing to make sure your workload isn't going to suffer from unexpected changes in performance and your code doesn't use any features that have been removed.

[Update 2016: And with this release it's even easier as you can set database-level compatibility mode, cardinality estimator, maxdop, and other settings. But there's still no substitute for proper testing before upgrading, **especially** as the new cardinality estimator isn't a 100% panacea and can lead to plan regressions. Erin's video this time is especially useful here...]

Call to action: make sure that your next upgrade is carefully planned to avoid problems. Start looking over the deprecated features lists and install the publicly available CTPs to get used to how the new version works. It doesn't mean you need to upgrade the day it's officially released but it does mean you'll be way ahead of the game once you are ready to migrate. Ultimately, it will help you to know that you're taking the upgrade step for the right reasons.

[Update 2016: Same thing applies, but you're too late now for the SQL Server 2016 CTPs. However, you can get Developer Edition completely free from Microsoft, which makes upgrade testing more palatable – see [here](#) for details – and you can upgrade from SQL Server 2005 or later to SQL Server 2016 directly – see [here](#) for details and restrictions.]

Video Demo

If you're planning an upgrade to SQL Server 2016, testing query performance with the new Cardinality Estimator (CE) is a must. But have you thought about how much easier that testing will be with Query Store? In this video Erin steps through testing a query using different compatibility modes and trace flags, and shows how Query Store can make finding the information you need extremely easy.

The video is just under 12 minutes long and you can get it:

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

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

Enjoy!

SQLskills Offerings

Our 2016 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 **New course!!**

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 **Just added!!**
- **IEPTO1:** Immersion Event on Performance Tuning and Optimization – Part 1
 - November 7-11 **Just added!!**
- **IEPDS:** Immersion Event on Practical Data Science
 - November 7-11 **Just added!!**

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