

(January 29th, 2019)

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](#).



Note: As an Insider, you can read all prior Insider newsletters [here](#).

Quick Tips for our Insider friends!

Hey Insiders,

This newsletter is coming to you from Redmond, where we're home until April, working hard on new content and enjoying the crisp, clear weather.

You can save \$100 off any online class when you use the discount code 'newsletter' for new registrations over the next two weeks!

And if you're in Europe, also check out [Erin's SQLBits precon](#) on February 27th!

SQLskills News

You can find all the details of our upcoming classes (pricing, logistics, registration, curriculum) by drilling down from the class schedule page [here](#), and there's a list of classes at the bottom of the newsletter.

Live, ONLINE classes: we're delivering an online class every two weeks through the end of **March**. It's the perfect way to keep learning, keep your systems moving forward, and stay motivated! And our format has received rave reviews: each class is held over three, *slightly-longer-than-half* days with a combination of lecture/demo and open Q&A. You can deep-dive into important content as well as keep up with your office workload! This is the place to learn new tech and get your questions answered!

Live, IN-PERSON classes: we're back in Chicago in April/May 2019 and our classes are open for registration, including a **new class on Power BI** taught by Tim.

And, if you're looking for a great place to learn best practices in migration, troubleshooting, and performance, join us in Florida in June for our [Spring SQLIntersection](#) conference where we're going to focus on this incredibly important process. **Glenn's Ponderings** this month will illustrate tools that can help and how to reduce regressions. SQLIntersection will not only help you get to a newer version of SQL Server but it will teach you the best new features to leverage once you're there!

Finally, even if you can't join us in person, I've put out a call for **2019 remote user group sessions** and we've got 32 scheduled this year already! If you'd like one of us to present for your user group, check out my blog post [here](#).

Book Review

The most recent book I've read is Jorge Carrión's [Bookshops: A Reader's History](#). The book is essentially a love letter to wonderful bookshops around the world (most famous, some not so much) that Carrion has visited. He gives a lot of history of them, how they came about, how they were cultural and social centers, which famous authors frequented them, and sometimes how they ceased to be. I was glad to see one of my favorite bookshops, Hatchards in Piccadilly, London, discussed. The book has lots of ideas for places to visit, and it was interesting reading about how some of these famous places came about. Recommended for bibliophiles!

Paul's Glenn's Ponderings

(This is a guest editorial from Glenn that's a combined editorial and Curious Case – enjoy!)

As SQL Server 2008 and SQL Server 2008 R2 rapidly approach the end of [Extended support from Microsoft on July 9, 2019](#), and with SQL Server 2014 also falling out of Mainstream support on July 9, 2019 (joining SQL Server 2012, which [fell out of Mainstream support on July 11, 2017](#)), I am seeing an increasing number of organizations that have been migrating to a modern version of SQL Server. I define a modern version of SQL Server as SQL Server 2016 or higher.

I see this as a good development overall, since SQL Server 2016 and higher actually have many useful, new features that make them much better products than their predecessors. Migrating to a modern version of SQL Server also usually means using newer, faster hardware and storage, running on a current version of Windows Server, which is also very beneficial.

Despite all of this, I have seen a decent number of cases where organizations have migrated from an older version of SQL Server to a modern version of SQL Server on new hardware and a new operating system, and then be unpleasantly surprised by performance regressions once they are in Production. How can these performance regressions be occurring, and what steps can you take to help prevent them?

The main culprit in most of these performance regressions is a combination of lack of knowledge, planning, and adequate performance testing. Unlike older versions of SQL Server, modern versions of SQL Server have several important performance-related configuration options that you need to be aware of, understand, and **actually test** against your workload. Most people who run into performance regressions have done what I call a “blind migration” where they simply restore their databases from the older version to the new version of SQL Server, with no meaningful testing of the performance effects of these different configuration options.

So, what are these key configuration options that you need to be concerned with from a performance perspective? The most important ones include [your database compatibility level](#), the cardinality estimator version that you are using, your [database-scoped configuration options](#), and [what trace flags you are using](#). Since SQL Server 2014, the database compatibility level affects the default cardinality estimator that the query optimizer will use. Since SQL Server 2016, the database compatibility level also controls other performance related behavior by default. You have the ability override many of these database compatibility level changes with database scoped configuration options and query hints. There are actually a rather large number of different combinations of settings that you have to think about and test. So, what are you supposed to do?

Microsoft Database Experimentation Assistant

In my ideal scenario, you would use the free [Microsoft Database Experimentation Assistant](#) (DEA) to capture a relevant production workload. This involves taking a full database backup, then capturing a production trace that covers representative high priority workloads. While this is going on, I would run some of my [SQL Server Diagnostic Queries](#) to get some baseline metrics from your legacy instance.

Once you have done that, you can then restore that backup to your new environment, and replay the production trace multiple times in your new environment. Each time you do this (which also includes a fresh restore from that original full production database backup), you will use a different combination of these key configuration settings. You have to make the database configuration/property changes *after* each restore, *before* you replay the DEA trace.

The idea here is to see which combination of these configuration settings yields the best performance with your workload. Here are some relevant, likely combinations:

- Use the default native database compatibility level of the new version
- Use the default native database compatibility level of the new version and use the query optimizer hotfixes database-scoped configuration option
- Use the default native database compatibility level of the new version and use the legacy cardinality estimator database-scoped configuration option
- Use the default native database compatibility level of the new version and use the legacy cardinality estimator database-scoped configuration option and use the query optimizer hotfixes database-scoped configuration option
- Use the existing database compatibility level of the old version
- Use the existing database compatibility level of the old version and use the query optimizer hotfixes database-scoped configuration option

This level of DEA testing may not be practical if you have a large number of databases, but you should really try to do it on your most mission critical databases. Barring that, I would try to do

as much testing of your most important stored procedures and queries as possible, using these different configuration settings.

Finally, if no adequate testing is possible you can follow Microsoft's [recommended upgrade sequence](#) (in your new production environment, after you go live), which is:

- Upgrade to the latest SQL Server version and keep the source (legacy) database compatibility level
- [Enable Query Store](#), and let it collect a baseline of your workload
- Change the database compatibility level to the native level for the new version of SQL Server
- Use [Query Store](#) (and [Automatic Plan Correction](#) on SQL Server 2017 Enterprise Edition) to fix performance regressions by forcing the last known good plan

You also have all of the other new “knobs” of [database-scoped configuration options](#), query-level hints, and trace flags available to you. You may have to do some additional work on some queries [with USE HINT query hints](#). Ideally, you would have done enough testing so that you already have a pretty good idea of the “best” combination of these settings for your workload, but many organizations don't actually do that.

Keep in mind that for each of the new QP features over the last two versions ([Adaptive Query Processing](#) in SQL Server 2017 and [Intelligent Query Processing](#) in SQL Server 2019), Microsoft exposes the ability to disable specific behavior at the database scoped configuration or query USE HINT scope. Microsoft generally recommends that if you do find regressions related to a specific feature, try disabling it at lower granularities first, so you can still benefit from all of the rest of the improvements you get from the latest database compatibility level.

Query Tuning Assistant

Microsoft is shipping a new tool called [Query Tuning Assistant](#) (QTA) in SSMS 18.0. QTA can guide you through the recommended database compatibility level upgrade process in a wizard-fashion, collecting the baseline workload in Query Store, bumping up the database compatibility level, and then comparing performance with the post-upgrade workload collection. At the end of this process, if performance regressions are detected, rather than moving back to the previously known good plan, the QTA will actually suggest hint-based improvements that can be deployed for individual queries (using plan guides), without having to necessarily move back to the legacy CE.

Call to action: With a bit of work beforehand, you should be able to avoid nasty surprises once you go into production on your upgraded version of SQL Server.

Glenn's Tech Insights

This section of the newsletter highlights recent news and views from the hardware and Windows worlds that we think will be interesting to SQL Server community members.

One Year of Free Support Remaining for Windows 7

On January 14, 2020, Microsoft's free extended support for Windows 7 will end. That means no more updates or fixes, including security fixes, unless you are an organization that is willing to pay for updates in one of two ways. First, [Microsoft will sell](#) Windows 7 Extended Security Updates (ESUs) through January 2023 on a per device basis, with the price increasing each year. Second, organizations can buy the upcoming [Microsoft Virtual Desktop](#) service, which lets you run Windows 7 remotely in Azure virtual machines, and also get Windows 7 ESUs free of charge. See:

- [Helping customers shift to a modern desktop](#)
- [Enhance your security – shift to a modern desktop now](#)

Mainstream support for Windows 7 with SP1 ended on January 13, 2015. Windows 7 was introduced in July 2009, and it has had a good run. Many people know and love Windows 7 and they are certainly very familiar and comfortable with it after nearly ten years.

From a SQL Server perspective, Windows 7 has become more difficult to use on a personal workstation since SQL Server 2016 and newer will not install or run natively on Windows 7. From a tooling perspective, the story is better so far, since SSMS 17.x, SSMS 18 Preview 6, and Azure Data Studio **will** install and run on Windows 7. I expect that this story will change in the future though.

From a hardware and device driver perspective, the Windows 7 story is less positive. Most new and newer desktop and laptop hardware [explicitly does not support](#) Windows 7. For example, Dell has this cautionary language on the support pages for many of their recent and current models:

“Before attempting to install Windows 7 or 8, [check your processor type](#). Systems with 7th Generation Intel® processors are not eligible for Windows 7/8/8.1 downgrades.”

What this means is that if you have a system with an Intel Kaby Lake or newer processor, only Windows 10 is officially supported and neither Intel or Microsoft provide Windows 7 drivers. This is in alignment with both Intel and Microsoft support policies. You also lose support for [new hardware performance features](#) such as [Intel Speed Shift](#) unless you are running Windows 10.

Seagate Introduces IronWolf 110 SSD Product Line for NAS

Seagate [has introduced](#) a family of 2.5” 7mm SATA TLC SSDs that are designed for network attached storage (NAS) use. These range from a 240GB model to a 3.85TB model. [These drives](#)

[are meant for prosumer and entry-level enterprise](#) NAS usage, and they offer lower power operation and better performance than traditional hard drives at the expense of less capacity per drive.

This gives you the option of using the larger sizes for capacity drives or using the smaller sizes for tiering or caching in your NAS. Especially if you have 10 GbE connectivity in your NAS, you need to think about whether the drive performance in your NAS becomes your main bottleneck or not. Being able to use flash storage in your NAS will let you address this issue.

Converting a Vintage Sun Ultra 24 Workstation to an AMD EPYC Workstation

Patrick Kennedy at [ServeTheHome](#) found a pristine (still in the original box with full documentation) [twelve-year-old Sun Ultra 24](#) workstation with an Intel Core 2 Duo E8600 processor and 4GB of DDR2 RAM in an ATX form-factor case.

He kept the case, but replaced the motherboard, power supply, processor, memory, video card, and storage. He used a 16C/32T frequency-optimized [AMD EPYC 7371 processor](#) with 256GB of RAM, along with fast, modern storage and 100 GbE network support. BTW, that is a good AMD EPYC processor for production SQL Server usage.

This will be a very capable workstation, and the article describing the parts list and build process is quite interesting. Personally, I would prefer a system built around an AMD Ryzen Threadripper 2950X processor, even though that would limit me to 128GB of RAM.

#TBT

(Turn Back Time...) This section highlights some older resources we've referred to recently that you may find useful, plus blog posts we've published since the previous newsletter.

The theme for the TBT this time is on-disk structures:

- Kimberly's Pluralsight course: [SQL Server: Why Physical Database Design Matters](#)
- Kimberly's Pluralsight course: [SQL Server: Indexing for Performance](#) (which goes into lots of detail on index structures)
- Paul's blog posts on the basics:
 - [Inside the Storage Engine: Anatomy of a record](#)
 - [Inside the Storage Engine: Anatomy of a page](#)
 - [Inside the Storage Engine: Anatomy of an extent](#)
 - [Inside the Storage Engine: IAM pages, IAM chains, and allocation units](#)
 - [Inside The Storage Engine: GAM, SGAM, PFS and other allocation maps](#)
- Paul's blog post categories:
 - [Inside the Storage Engine](#)
 - [On-disk Structures](#)

Here are a few of the blog posts we've published since the last newsletter:

- Glenn: [New Pluralsight Course Published](#)
- Erin: [Important Query Store Fixes – January 2019](#)

I hope you find these useful and interesting!

Video Demo

From Erin: Columnstore indexes have been available since SQL Server 2012, and Microsoft continues to make improvements to the feature. But one area where I've run into issues is estimates, which then affect query performance. In this video we take a look at what statistics exist for columnstore indexes in SQL Server.

The video is just over 11 minutes long and you can get it [here](#).

The demo code is [here](#).

Enjoy!

Upcoming SQLskills Events

Our first set of 2019 live, in-person events has been announced for Chicago in April/May and we've also published our full 2019 Q1 lineup of live, online classes.

Each and every event has a different focus as well as different benefits – from deep-technical training in our Immersion Events to wide-ranging topics at SQLintersection where you can learn more effectively how to keep moving forward in both your environment and your career! And, of course, one benefit you'll always get from in-person events is networking; we hope to meet/see you at an event soon!

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](#)
- [So why do you want to come to our training? And the winners are...](#)
- [Community blog posts about our classes](#)
- [Immersion Event FAQ](#)

LIVE, ONLINE Immersion Events:

- **IEPUM2017:** Immersion Event on Planning and Implementing an Upgrade/Migration to SQL Server 2017

- January 29-31 (started today)
- **IEQUERY:** Immersion Event on Fixing Slow Queries, Inefficient Code, and Caching/Statistics Problems
 - February 12-14
- **IETLB:** Immersion Event on Transactions, Locking, Blocking, Isolation, and Versioning
 - February 26-28
- **IEVLT:** Immersion Event on Very Large Tables: Optimizing Performance and Availability through Partitioning
 - March 12-14
- **IECS:** Immersion Event on Columnstore Indexes
 - March 26-28 **** NEW ****

LIVE, IN-PERSON Immersion Events:

Chicago, IL, April/May 2019

- **IEPTO1:** Immersion Event on Performance Tuning and Optimization – Part 1
 - April 29-May 3
- **IECAG:** Immersion Event on Clustering and Availability Groups
 - April 29-30
- **IEPowerBI:** Immersion Event on Power BI, Power BI Report Server, and SSRS
 - April 29-30 **** NEW ****
- **IE0:** Immersion Event for the Junior/Accidental DBA
 - May 1-3
- **IEUpgrade:** Immersion Event on Upgrading SQL Server
 - May 1-3
- **IEPTO2:** Immersion Event on Performance Tuning and Optimization – Part 2
 - May 6-10
- **IEPML:** Immersion Event on Practical Machine Learning
 - May 6-10
- **IEAzure:** Immersion Event on Azure SQL Database, Azure VMs, and Azure Managed Instance
 - May 6-9

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