(June 25th, 2018)

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,

We've just finished our final US in-person class for 2018, and for the rest of the year we'll be focusing on live, online classes. We have **three brand-new live, online classes with special newsletter discounts that expire next Monday** – see below!

Additionally, we still have a few seats remaining in our London classes in September; they're our last, in-person Immersion Events to be delivered this calendar year!

SQLskills News

Live, online classes: there are three new classes available for registration:

- August 28-30: ***IEPUM2017****: Immersion Event on Planning and Implementing an Upgrade/Migration to SQL Server 2017* (taught by Glenn)
- October 9-11: ***IETLB****: Immersion Event on Transactions, Locking, Blocking, Isolation, and Versioning* (taught by Kimberly)
- October 23-25: ***IEQUERY****: Immersion Event on Fixing Slow Queries, Inefficient Code, and Caching/Statistics Problems* (taught by Erin, Jonathan, and Kimberly)

These classes will be delivered live via WebEx 10am-3pm PST, Tuesday-Thursday (roughly 12-13 hours of content including open Q&As; similar to two full workshop days without leaving the comfort of your home/office!) and you also get access to the class recordings. By dedicating only 3 half-days of your time you still have time to get some work done during the day and with lifetime access to the recordings, you get amazing ROI!

Each of these classes are priced at US$699 and we're offering a combo package of all three for US$1,749, saving US$350. And as a special discount ONLY for newsletter subscribers, **use the discount code 'newsletter' before July 1st (this offer will NOT be extended) to save US$100 off each class (US$599 each) or an additional US$150 off the combo price (US$1,599 for the three-course combination for a total saving of US$500!)**. Click on the class links above for all the details.

In-person London classes: We're bringing four of our Immersion Events to London in September: IEPTO1 and IEPTO2, plus our new classes: **IEAzure** on Azure, Azure VMs, and azure Managed Instance and **IECAG** on clustering and availability groups **(register for one of the new classes and get the other one half-price!)** See here for all the details.

Finally, even if you can't join us in person, I've put out a call for 2018 remote user group sessions. In 2017, we did more than 100 of these around the world and we have set up more than 70 for 2018 already! If you'd like one of us to present for your user group, check out my blog post here. **Note: Tim has a new user group session on Azure Managed Instance** that he's happy to present to your group – see here for details.

Book Review

The latest book I've been reading is J. R. R. Tolkien's *The Lord of the Rings: 50th Anniversary Edition* (the link is the paperback edition, but I have a magnificent leather-bound edition that cost quite lot as this is one of my favorite books). Unless you were living in a cave in 2001-2003, you can't have missed at least hearing about the three Lord of the Rings movies that dominated the cinema in each of those years, with the final one (The Return of the King) being one of only three movies ever to win 11 Academy Awards (alongside Titanic, and the 1959 Ben-Hur).

I first read TLOTR when I was 11, read it again in college, and then again in 2000 while on parental leave from Microsoft after my first daughter was born, making this my fourth reading. After 18 years, I'd forgotten how rich the storytelling is, and how many things were left out of the movies. The story is very complicated, but can be boiled down to: an evil ring must be taken into the heart of the most dangerous place (Mordor) in the land (Middle Earth) so it can be destroyed, and the task falls to a hobbit, about the most unlikely of all the good races in Middle Earth (men, dwarves, elves, hobbits, and so on). All kinds of side stories happen, leading up to the final battles. You don't *have* to read *The Hobbit* first, but it certainly helps.

This is an absolutely wonderful book, and I can't recommend it enough! And the movies are just stunning, especially in their longest Director's Cut editions – I've lost count of how many times I've watched them.

The Curious Case of…

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

A quick one this time: Erin was working with one of our remote DBA clients last week and they had a problem where a reporting application had stopped working and they couldn't connect to the instance.

Erin checked it out, and sure enough, reports were throwing errors when attempting to connect to the instance, and connections failed through SSMS as well.

However, Erin was able to connect to the instance remotely using the Dedicated Admin Connection (DAC), as we'd configured the instance to allow remote admin connections, and fix the problem.

Bottom line: Make sure that your have the 'remote admin connections' option enabled so you can make use of the DAC if you're unable to connect to an instance normally (otherwise you're probably looking at restarting the instance). The DAC is a separate scheduler, memory, and thread to allow you to connect to an otherwise unavailable instance and investigate and hopefully fix the problem. It's been there since SQL Server 2005 and you can read more about it in Books Online here.

Paul's Ponderings

I've been teaching all this week so was looking for an old Ponderings to rerun, and I was inspired to pick this one from four years ago after discussing some of the client issues the team has been dealing with this week. And I've added a few comments. Enjoy!

Every day in our lives, most of us seek to minimize risk. Examples include assessing traffic to choose the safest point to pull out into the road, choosing flights in a multi-flight trip to avoid flight delays causing a missed connection, and delaying rocket launches based on the probability of bad weather occurring.

Ok – we don't all do the last one, but you get the idea. It's a natural human tendency to want to limit our risk of possible disaster and negative outcomes.

Here's an example of forced risk aversion. A few years ago I was performing a health check for a client of a client. Our client writes casino-management software and their client was a casino – we'll call them X. I identified a performance issue that could be mitigated through the creation of several nonclustered indexes and some code changes. Our client said that wouldn't be possible for 18 months because the Gaming Commission in the state where X resides doesn't allow software changes until they've been extensively investigated and tested, to avoid any risk of instability or fraud.

The only other solution I could think of was to max out the server memory and drop in some SSDs, to make the server cache as much of the workload as possible and store the rest on the fastest possible I/O subsystem. This would be expensive to the client, but X was a casino, so I figured they could afford it. Problem solved.

That was a case where there was no choice, but often when dealing with SQL Server performance problems that require code changes – and sometimes extensive code changes – I find that people don't stop to consider less risky alternatives.

Changing code is inherently risky. Most code is not documented, even to a minimal standard of 'what does each block do and why'. It's quite horrifying to see really, especially being a former developer and development manager at DEC/Digital and Microsoft where code complexity demanded rigorous documentation (and even then, it was often inadequate).

[Edit 2018: And sometimes developers don't even test the code and slap it straight into production, causing all kinds of problems – including wasted money from then requiring consultants like us to untangle the mess.]

My point is that allowing developers to make changes to a mostly undocumented and sometimes poorly-understood code base should give you sleepless nights. The risks involved include instability, loss of functionality, introducing security flaws, introducing other performance problems, and a host of other potential side-effects. So what are you to do?

Consider the alternatives:

1. Can you avoid/lessen the performance problem by buying more memory for the server and storing more of the workload in the buffer pool?
2. Can you avoid/lessen the performance problem by boosting the speed of the I/O subsystem – most likely by going to solid-state storage?
3. Can you upgrade your server to one that has newer processors with faster per-core performance?
4. Can you create a small number of nonclustered indexes that will provide a net gain in performance, even considering the added runtime maintenance, regular index maintenance, and increased storage required for them?

The first three of these cost money, but so does making a code change. You're paying the salary of developers, testers, project managers, and others involved in making the change. New hardware is a one-time capital expense.

Numbers 1, 2, and 4 are also things that can be done within days, providing almost immediate relief of a performance problem, whereas changing code *properly* takes a lot of time.

[Edit 2018: #4 might not be possible if the code is in an application where changing the schema would in validate a support contract/warranty.]

Don't get me wrong – I'm not saying that you can use hardware to completely remove the need to change code, or that by default you should throw hardware at a performance problem. As your workload increases, you'll likely hit the point where you *have* to change the code to fix the problem, but you can often delay that point, giving you time to properly design, prototype, test, and implement the right fix in a robust, less risky way.

Call to action: The next time you're considering changing existing code to solve a performance problem, ask yourself if there's a less risky, and possibly cheaper, short-term alternative. And try

to avoid the view that capital expense is a "worse" cost than making a code change. If the code change involves people who you pay money, and carries a risk of causing more problems that will take more people who you pay money to fix, plus annoy clients and possible lose business, which solution is cheaper in the long run? I don't want you to throw hardware at performance problems by default – I just want you to consider the alternatives to making extensive code changes.

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.

AMD Announces Ryzen Threadripper 2 Processor

On June 5, 2018, AMD announced the upcoming AMD Ryzen Threadripper 2, which is on track for a Q3 2018 release (supposedly in August). This new HEDT processor family will use the "12nm Zen +" architecture and will have up to 32 cores (with 64 threads) with a 250W thermal design power (TDP) rating.

These new processors will work with existing AMD X399 motherboards (after a BIOS update), which also means that they will have the same four memory channels and 64 PCIe 3.0 lanes as the first generation AMD Ryzen Threadripper processors.

A high-end desktop (HEDT) system built with this processor family gives you a very high level of compute, memory, and storage performance that is very useful for running multiple instances of SQL Server, running multiple concurrent VMs, etc.

AMD Threadripper Exchange Program

In an interesting marketing ploy aimed directly at Intel, AMD is offering to let the first 40 U.S.-based winners of the Intel 8th Gen Core i7-8086K Limited Edition Sweepstakes to exchange their Intel Core i7-8086K processor for an AMD Ryzen Threadripper 1950X processor. As AMD puts it:

> *"We appreciate the advancements they've helped drive with the x86 architecture over the last four decades. But, we're ready to take it from here. That's why we're giving 40 performance-hungry enthusiasts in the U.S. an opportunity to celebrate the next 40 years of high-performance computing by trading in their commemorative processor prize for our CPU that enables you to work, play and create with heavy metal."*

If you want a processor for the best single-threaded performance, the Intel Core i7-8086K processor is a better choice. If you are doing heavy multi-threaded work, the AMD Ryzen Threadripper 1950X is a better choice.

AMD EPYC 2 Server Processors

AMD has revealed that the 2nd generation AMD EYPC server processors will be sampling with hardware partners in 2H 2018 and will be released in 2019. These are code-named "Rome", and will use a 7nm manufacturing process. They are expected to have up to 64 physical cores, plus SMT (for a total of 128 logical cores).

The other specifications, such as clock speeds are still TBD. This type of processor will continue to make AMD a viable competitor to Intel in the data center for more types of server workloads. Having actual competitive server processors from AMD and Intel is a welcome development for database professionals.

High Memory Capacity Laptops

Several hardware vendors, such as Dell and Lenovo have announced new laptop models that support 128GB of RAM, along with new six-core Intel Xeon E-2176M, multiple M.2 NVMe storage slots and two external Thunderbolt 3 ports.

Specifications like these give you a lot of compute, memory, and storage capacity for running pretty heavy workloads on a laptop. Of course, I still think that you can get a lot more capacity for less money in a new desktop system, but if you do need to be mobile, this is your best alternative.

#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 #TBT this time is deadlocks so here are some resources for you:

- Jonathan's Pluralsight courses: *SQL Server: Deadlock Analysis and Prevention*
- The deadlock entry from our Accidental DBA blog-post series
- Jonathan's very long SimpleTalk article on *Handling Deadlocks in SQL Server*, which is an excerpt from his popular (and free) ebook *Troubleshooting SQL Server: A Guide for the Accidental DBA*

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

- Erin: *Updating Statistics with Ola Hallengren's Script*
- Erin: *Can you force a plan for a different query with Query Store?*
- Erin: *Query Store and the Plan Cache Flushing*
- Erin: *Why aren't you using Query Store?*

- Paul: *New live online training class in October: Fixing Slow Queries, Inefficient Code, and Caching/Statistics Problems*
- Paul: *New live online training class in October: Transactions, Locking, Blocking, Isolation, and Versioning*
- Paul: *New live online training class in August: Planning and Implementing an Upgrade/Migration to SQL Server 2017*
- Glenn: *AMD Ryzen Threadripper 2990X CPU-Z Leaks?*
- Glenn: *Performance and Stability Related Fixes in Post-SQL Server 2016 SP2 Builds*

I hope you find these useful and interesting!

Video Demo

In this video, Tim demonstrates the capabilities of Dynamic Data Masking, which allows administrators to limit sensitive data exposure by masking it to non-privileged users.

Dynamic Data Masking has been available in Azure SQL Database for years and was introduced on-premises with SQL Server 2016. Dynamic Data Masking allows you to mask how sensitive data appears in the result set. Utilizing Dynamic Data Masking does not physically change how the data is stored in the database; instead it applies a masking rule at the query processing layer. By default, masking rules are applied to all users except for those excluded. Built in masking functions are 'default' which uses 'x's to mask the field, 'credit card' which only exposes the last four digits, 'email' which exposes the first letter and replaces the domain with xxx.com, 'random number' which generates a random number according to the selected boundaries and actual data types, and 'custom text' which exposes the first and last characters and adds a custom padding string in the middle.

The video is just over 8 minutes long and you can get it in WMV format here.

The demo code is here.

Enjoy!

Upcoming SQLskills Events

We have events coming up in 2018 – from our ***new, live, online*** courses to our own live, in-person Immersion Events to our own conference: SQLintersection; all of our Fall events are open for registration.

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
  - August 28-30 **(** NEW **)**
- **IETLB**: Immersion Event on Transactions, Locking, Blocking, Isolation, and Versioning
  - October 9-11 **(** NEW **)**
- **IEQUERY**: Immersion Event on Fixing Slow Queries, Inefficient Code, and Caching/Statistics Problems
  - October 23-25 **(** NEW **)**
- **Special savings on the three-course combination when registering before July 1; see the SQLskills News section above for full details!**

LIVE, IN-PERSON Immersion Events:

London, UK, September 2018

- **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1
  - September 10-14
- **IEAzure**: Immersion Event on Azure SQL Database and Azure VMs
  - September 10-11 **(** Buy IEAzure, get IECAG half-price **)**
- **IECAG**: Immersion Event on Clustering and Availability Groups
  - September 12-13 **(** Buy IECAG, get IEAzure half-price **)**
- **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2
  - September 17-21

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