(April 30th, 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.

Quick Tips for our Insider friends!

Hey Insiders,

We had a great week last week in Chicago, with the whole SQLskills team teaching a total of 5 classes (IE0, IEPTO1, IEUpgrade, IECAG, and IEAzure). This week we start three more Immersion Events with our IEPTO2 and our partner classes on PowerShell and BI Strategies / Analytics. And for our final week in Chicago, we have two more classes running: Practical Data Science and SQL Server Integration Services.

And, if you can't attend an in-person class, we also have a new, live, online class debuting in May! Erin will be presenting our new ***Online IEQS: Immersion Event on Solving Common Performance Problems with Query Store***. This will be delivered live via WebEx on May 22-24 (roughly 12-13 hours of content including open Q&As; similar to two full workshop days without leaving the comfort of your home/office!). It's priced at only US$795. **Move fast to claim your seat!** See here for all the details.

Note: you can get all the prior Insider newsletters here.

SQLskills News

In-person US classes: We have the following classes remaining for 2018: Learning SSIS and Practical Machine Learning next week in Chicago, and IEPTO1 in Bellevue, WA in June. **We don't have any additional classes planned in the US in 2018;** see here for our 2018 Immersion Event class schedule.

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). See here for 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 60 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

One of the books I've read recently is *Why Nations Fail: The Origins of Power, Prosperity, and Poverty* by Acemoglu and Robinson. The book puts forth a theory that nations with inclusive economic and political institutions are much more likely to succeed than those with extractive institutions (extractive meaning that the populace is exploited by a small elite for their own gain, inclusive meaning that all citizens are treated equally and property rights are protected by law). It also explains why inclusive institutions developed in some parts of the world and not in others, with exploitative colonialism being a major historical impediment to inclusivity. Lots of interesting case studies and history – highly recommended!

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.

I was helping someone out with a corruption problem over email last week, and it was the usual case of not having viable backups. The most recent backup without the corruption was several weeks old and they didn't want to go back that far because of the hassle of having to then extract all the recent data from the corrupt database and insert it into the older copy.

Unfortunately for them, their log was also corrupt (I suspect something was badly wrong with their I/O subsystem) so they had to use *WITH CONTINUE_AFTER ERROR* (more on this in the *Ponderings* section) to make the restore sequence work, and then had no choice but to run emergency-mode repair. They attempted it and it failed with the following messages:

> *Msg 41836, Level 16, State 1, Line 44*
> *Rebuilding log is not supported for databases containing files belonging to MEMORY_OPTIMIZED_DATA filegroup.*
> *Msg 7909, Level 20, State 1, Line 44*
> *The emergency-mode repair failed. You must restore from backup.*

Because they were using In-Memory OLTP, emergency-mode repair didn't work. They then faced the onerous task of restoring the older backup and extracting data from the corrupt database to try to bring the older copy up-to-date.

Bottom line: There's no substitute for having valid backups. That means you have to make sure that DBCC CHECKDB is running regularly and you're testing that your backups allow you to restore a clean copy of your critical databases. Ever since we put emergency-mode repair into SQL Server 2005, there's been no guarantee that it works in all situations. And, as the complexity of your database increases and you use more and more advanced features, you need to both have a good backup strategy and test your restore capabilities regularly.

Paul's Ponderings

Continuing from the *Curious Case* above, I want to go into some more detail on the *CONTINUE_AFTER_ERROR* option.

SQL Server 2005 introduced this option for both for *BACKUP* and *RESTORE* commands. For backups, the option tries to force a corrupt database to be backed up (I don't see it used very often for this), and for restores, it tries to force a corrupt backup, or backup of a corrupt database, to restore. In both cases, the option to use is *WITH CONTINUE_AFTER_ERROR*, which instructs the *BACKUP* and *RESTORE* commands to try to cope with the error and continue. However, depending on how corrupt the database or backup is, it's not always possible.

Why would I want to backup a corrupt database, you may ask? Consider this: you have a corrupt database with no backups. The only way to attempt recovery in that case is to use the repair functionality in *DBCC CHECKDB*. Even though you can put the *DBCC CHECKDB* command inside a transaction and roll it back, there are infinite corruption possibilities and repair doesn't cope with them all. Therefore, prudence dictates that you take a backup of the database before attempting the repair, just in case something goes wrong. If the backup operation fails, you can try to use *CONTINUE_AFTER_ERROR*. That's safer than having to set the database offline so the files can be copied – because it may not be possible to bring the database online again, even using *EMERGENCY* mode.

The use case for *CONTINUE_AFTER_ERROR* during restore is much more obvious: you've lost the database and only have corrupt backups. If the backup fails to restore because of a corruption problem in the database or transaction log within the backup, the only option is to try to force it to restore using *CONTINUE_AFTER_ERROR* and then either run repair or try to extract data manually.

There is one catch to beware of though – think very carefully about restoring a transaction log backup that requires *CONTINUE_AFTER_ERROR* to successfully restore because doing so guarantees that you're introducing corruption into the restored database. It may be better to terminate the restore sequence before that corrupt transaction log backup, resulting in a non-corrupt database with some data loss, than to proceed and have a corrupt database that needs to be repaired.

One insidious case of a corrupt log backup can happen unexpectedly during a disaster situation, and it's related to using the bulk-logged recovery model. Let me explain…

The first log backup after a minimally-logged operation has to back up all the log generated since the last log backup, plus all the data extents changed by the minimally-logged operation, otherwise the log backup does not contain enough information to fully reconstitute the effects of the minimally-logged operation. You'd end up with a set of allocated pages for the index with no

contents. In fact it's worse – you'd end up with a set of allocated pages for the index that haven't even been formatted – massive corruption.

Now consider the case where you've switched into the bulk-logged recovery model and performed a minimally-logged operation, such as an index rebuild. And then before the next log backup can be taken, a disaster occurs and the data files are destroyed.

One of the first things you're likely to attempt is a tail-of-the-log backup. Through SQL Server 2008 this will fail, as the log backup needs to access to the inaccessible data files. From SQL Server 2008 R2 onwards, the log backup succeeds, telling you that under the covers it used *CONTINUE_AFTER_ERROR*, even though you didn't ask it to. Remember this is only in the special case of a tail-of-the-log backup after a minimally-logged operation and with the data files inaccessible.

This behavior is badly broken. (*Kimberly and I aren't fans of this "feature."*)

It's broken because the resulting log backup is useless. If you restore it as part of your restore sequence while recovering from the disaster, it will corrupt your database, as I described above. In my opinion the behavior should have been left as it was with SQL Server 2008, as I can't see the rationale for automatically creating a useless, corruption-causing log backup.

But remember, this is a special case.

In summary, and of course, the best solution is not to have to use *CONTINUE_AFTER_ERROR* at all; have a redundant copy of the database and have good backup and backup testing strategies.

Call to action: If you're involved in a disaster recovery effort, don't forget that the *CONTINUE_AFTER_ERROR* option exists. Make sure you've practiced with it before having to do it for real so you know the kinds of problems you might come across from restoring a corrupt backup, or backup of a corrupt database. And if you're ever in the situation where you need to perform a tail-of-the-log backup, pay close attention to the output of the *BACKUP LOG* command. If it mentions having to use *CONTINUE_AFTER_ERROR* then that backup you just took will result in database corruption if you restore it.

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.

Intel Threat Detection Technology

On April 17, 2018, Intel announced two new security-related technologies that are part of what they call Intel Threat Detection Technology. One of these is Accelerated Memory Scanning, which will allow products such as Windows Defender to use the integrated graphics processor unit that is present is many Intel processors to scan system memory for threats (rather than using the CPU itself).

Using the GPU rather than the CPU can substantially reduce the CPU utilization during scanning operations. This capability is available in 6th, 7th, and 8th generation Intel Core processors (which means the Intel Skylake, Kaby Lake, and Coffee Lake processors).

The second component is called Advanced Platform Telemetry, which uses system telemetry along with machine learning algorithms to improve threat detection, reduce false positives, and minimize the impact on performance. This is meant for use with the Cisco Tetration platform for data centers.

AMD 2nd-generation Ryzen Desktop Processors

On April 19, 2018 AMD released four new processors in their new 12nm 2nd generation Ryzen desktop processor family. These include the Ryzen 7 2700X, Ryzen 7 2700, Ryzen 5 2600X and Ryzen 5 2600. The Ryzen 7 parts have eight-cores and sixteen threads, while the Ryzen 5 parts have six-cores and twelve threads.

All of these processors will work with existing Socket AM4 motherboards, along with new X470 chipset Socket AM4 motherboards. This new line of processors is also significantly less expensive than the equivalent original Ryzen desktop processors were when they were launched last year. For example, the new Ryzen 7 2700X is $329.00 while the old Ryzen 7 1800X was $499.00. The new processors also come bundled with very good quality CPU coolers.

Despite what you may have heard, these are not Ryzen 2 processors or the Zen 2 architecture. The proper name is 2nd generation Ryzen processors (Pinnacle Ridge), using the Zen + architecture. They use Precision Boost 2 (which is similar to Intel Turbo Boost), and eXtended Frequency Range 2 (XFR2) which provides even higher processor clock speeds for more cores based on the thermal headroom available.

This means that you will see even better performance if you invest in the best CPU cooling solution possible. The top-end Ryzen 7 2700X processor has better multi-threaded CPU performance than the Intel Core i7-8700K in most benchmarks, while the single-threaded performance is quite competitive with the Core i7-8700K on most benchmarks. Overall, these processors are getting very positive reviews.

As you would expect, there are many detailed reviews of these processors out already:

- AnandTech
- [H]ardOCP

- Tom's Hardware
- PC Perspective
- Serve the Home

Red Hat Enterprise Linux 7.5

On April 10, 2018, Red Hat announced general availability of Red Hat Enterprise Linux 7.5. RHEL is one of the Linux distributions that will be supported by Microsoft SQL Server 2017. The other supported Linux distributions are SUSE Enterprise Linux (SLES) v12 SP2 or later, and Ubuntu 16.04 LTS or later.

Speaking of supported Linux distributions, Microsoft has a special promotion with SUSE, offering a 100% discount on the SUSE licensing cost, along with a 30% discount on the SQL Server 2017 licensing. They have a PDF that explains this in more detail.

#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.

We've had a spate of replication issues at clients since the last newsletter, so that's the theme for the #TBT section this time. Here are some resources for you:

- Joe's Pluralsight course on *SQL Server: Transactional Replication Fundamentals* that has 2 hours explaining what replication is, how to set it up, how to monitor and troubleshoot it, and how to combine it with other HA technologies
- Whitepaper on *SQL Server Replication: Providing High-Availability Using Database Mirroring* that I wrote for Microsoft back in 2008, but is still completely relevant today and can be extrapolated to working with Availability Groups too.
- Old whitepaper on *Proven SQL Server Architectures for High Availability and Disaster Recovery* that I wrote for Microsoft in 2010, and isn't listed on their site any more, but has an interesting set of architectures, including one using peer-to-peer replication.
- And a bunch of blog posts:
    - *In defense of transactional replication as an HA technology*
    - *REPLICATION preventing log reuse but no replication configured*
    - *The Transactional Replication Multiplier Effect*
    - *When is the Publication Access List required?*
    - And check out the Repltalk blog on MSDN that's been a fount of replication knowledge since 2010.

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

- Paul: *Read committed doesn't guarantee much…*
- Glenn: *SQL Server 2017 Cumulative Update 6*

I hope you find these useful and interesting!

Video Demo

The demo video this time is from Glenn's recent Pluralsight course *SQL Server: Understanding, Configuring and Troubleshooting Database Mirroring*. In the demo Glenn shows how to identify common problems that occur when using database mirroring, such as failures that cause failovers and large send or redo queues.

The video is about 7 minutes long and you can get in MOV format here.

And the demo code is here.

Enjoy!

Upcoming SQLskills Events

We have lots of events coming up in 2018 – from our ***online* IEQS** course to our own LIVE, in-person Immersion Events in both the U.S. and London; all events are open for registration. Every event has a different focus and different benefits – from deep-technical training in our online courses and in-person IEs to wide-ranging topics at SQLintersection where you can learn more effectively how to keep moving forward in both your database and your career! And, of course, one benefit all our in-person events provide is networking.

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

Online, May 2018

- **IEQS**: Immersion Event on Query Store **** NEW ****
  - May 22-24

Chicago, IL, May 2018

- **IESSIS1**: Immersion Event on Learning SQL Server Integration Services

    - May 7-11
- **IEPML**: Immersion Event on Practical Machine Learning
    - May 7-11 **(** NEW **, *only 2* seats remaining)**

Bellevue, WA, June 2018

- **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1
    - June 18-22 **(** Buy 2, get 1 free!, *only 8* seats remaining **)**

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
- **IECAG**: Immersion Event on Clustering and Availability Groups
    - September 12-13
- **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