(March 9th, 2022)

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!

This newsletter is coming to you from Redmond, where the weather is slowly improving and it’s almost time to regularly head up to camp to start working with my new tractor!

SQLskills News

SQLskills Insider Sessions!

We wrote about the motivation behind our Insider Sessions and you can register for just one, or the entire series here: https://www.SQLskills.com/iSessions. We’ve started them back up in 2022 but we’re going to cut back on the frequency. **But, the best news is that we’ve posted all of the prior sessions online for streaming. For free!** Just go to the completed sessions and you’ll find all of the resources, and the videos! Enjoy.

As for the rest of this year, we’ll do roughly one a month. The next one is next Wednesday:

- **SQLskills Insider Session: Parallelize Maintenance Tasks Using Service Broker with Jonathan Kehayias**
  - Wednesday, 3/16 at 10am Pacific Time (note that Daylight Savings starts in the U.S. this coming Sunday) – details here.

Our thought: block some time – EVERY WEEK! **Dedicate 2 hours a week to Improving Your SQL skills with SQLskills**! Join us when it makes sense (hopefully always! ☺) or, read blog posts, review topics of interest, go spelunking in the documentation on a topic that’s always interested you. Stay fresh, stay current – stay ahead of the competition!

Block some time NOW, register online, and when a session is planned to run, we’ll send you a meeting link with joining instructions.

Book Review

The latest book I’ve read is Max Hastings’ *Catastrophe 1914: Europe Goes to War*. I bought this in 2014, the obvious centennial of the start of WWI, but my colossal reading list stopped me getting to it until now. What a fantastic book about such a horrendously incompetent descent into and start to the conflict! This is the first in-depth book I've read about WWI and I have more in the queue.

From Amazon: "In Catastrophe 1914, Max Hastings gives us a conflict different from the familiar one of barbed wire, mud and futility. He traces the path to war, making clear why Germany and Austria-Hungary were primarily to blame, and describes the gripping first clashes in the West, where the French army marched into action in uniforms of red and blue with flags flying and bands playing. In August, four days after the French suffered 27,000 men dead in a single day, the British fought an extraordinary holding action against oncoming Germans, one of the last of its kind in history. In October, at terrible cost the British held the allied line against massive German assaults in the first battle of Ypres. Hastings also re-creates the lesser-known battles on the Eastern Front, brutal struggles in Serbia, East Prussia and Galicia, where the Germans, Austrians, Russians and Serbs inflicted three million casualties upon one another by Christmas. As he has done in his celebrated, award-winning works on World War II, Hastings gives us frank assessments of generals and political leaders and masterly analyses of the political currents that led the continent to war. He argues passionately against the contention that the war was not worth the cost, maintaining that Germany’s defeat was vital to the freedom of Europe. Throughout we encounter statesmen, generals, peasants, housewives and private soldiers of seven nations in Hastings’s accustomed blend of top-down and bottom-up accounts: generals dismounting to lead troops in bayonet charges over 1,500 feet of open ground; farmers who at first decried the requisition of their horses; infantry men engaged in a haggard retreat, sleeping four hours a night in their haste. This is a vivid new portrait of how a continent became embroiled in war and what befell millions of men and women in a conflict that would change everything."

Enormously recommended for history buffs!

The Curious Case of…

This section of the newsletter explains recent problems we’ve helped with on client systems or been asked about over email or #sqlhelp; they might be something you’re experiencing too.

Jonathan came across an interesting case recently of log shipping restores randomly failing. You can read about it here…

Ponderings…

(From me this time – I first ran this in 2015, but it cropped up yesterday and is still completely relevant today. Enjoy!)

When people design indexes, they primarily consider how useful the index will be for the query workload. Secondarily, they’re likely to consider maintenance requirements and potential problems around index fragmentation that may occur. However, one facet of index design that's usually neglected is considering the *fanout* or *fan-out* of the index structures themselves.

The fanout of an index measures, for a page at level x in an index, how many pages it references in the level below (nearer the leaf level). The higher the fanout is, the fewer the number of levels in the index.

The index key size impacts the size of the structure needed to reference it. Specifically, the index key is pushed up to all entries (and all levels) in the index as it's used to allow navigation through the index from the root page down to the leaf level.

The larger the index key size, the fewer index records can be stored in an index page and so the lower the fanout. The lower the fanout is the more levels are required in the index, depending on the number of pages at the leaf level.

For instance, if the fanout is 10 in an index, that means each index page can hold 10 index records, referencing 10 pages at the level below in the index. If the index has 10,000 pages at the leaf level, there needs to be 1,000 pages in the level above, then 100 pages, then 10 pages, and finally the root page. That's a total of 5 levels.

For the same data, if the index fanout is changed to 100, and the index has 10,000 pages at the leaf level, the next level needs 100 pages, and then there's the root page. That's a total of only three levels.

So why do you care about the number of levels in the index, determined by the fanout?

Well, when traversing an index, at each level in the Storage Engine needs to decide how to navigate further down the index. It does this using the records in the page, to figure out where the key value is that it's searching for.

The more levels there are in the index, the more CPU each search through the index takes, and so the slower each search is. Even though the index itself may be wholly contained in memory, this operation per-level in the index requires performing a search (it's an efficient binary search, but it's still a search that uses CPU resources).

So the reason you need to care about fanout is that low fanout increases CPU usage during searches.

And fanout is controlled by index key size, so careful index key choice entails not just considering fragmentation effects, but also minimizing the key size as much as possible to make the highest possible fanout.

I wrote a detailed blog post after I originally published this editorial back in 2015 with query timing information for various key sizes and index depths – check it out: *On index key size, index depth, and performance*.

Call to Action

When you're evaluating indexes for efficiency, make sure you consider the index key size and the fanout, with a view to minimizing the number of levels in the index. You can see the number of levels in your indexes using the *sys.dm_db_index_physical_stats* DMV, and also looking at the *IndexDepth* value in the *INDEXPROPERTY* function.

Additionally, if you want to get more details on ***why*** physical database design matters – check out Kimberly's Pluralsight course on exactly this topic

#TBT

(Turn Back Time…) Blog posts we've published since the previous newsletter plus some older resources we've referred to recently that you may find useful.

Wait stats are the theme for #TBT this time. Here are some wait statistics resources for you:

- My Pluralsight: *SQL Server: Performance Troubleshooting Using Wait Statistics*, that has 4.5 hours covering waits, latches, and spinlocks
- My main post about wait statistics
- My wait stats library with info on more than 360 wait types
- The introductory whitepaper Jonathan and Erin wrote
- Microsoft advanced whitepapers I helped to produce:
    - *Diagnosing and Resolving Latch Contention on SQL Server*
    - *Diagnosing and Resolving Spinlock Contention on SQL Server*
- My three blog post series on waits, latches, and spinlocks
    - Our articles on SentryOne's sqlperformance.com about wait statistics

Posts since the last newsletter:

- Paul: *The Curious Case of… random failing log shipping restores*

I hope you find these useful and interesting!

Video Demo

In this week's demo video, Jonathan takes a look at one of the highly touted new features of SQL Server 2019, Inlining of Scalar User Defined Functions, and shows an example of where performance degrades rather than improves.

The video is 4.5 minutes long and you can get it in WMV format here.

The demo code is here.

Enjoy!

Upcoming SQLskills Events

We’ve moved all our classes to 2022 – schedule details will be published shortly.

With our new streaming system, you can now choose to attend a live, online event or purchase a recording to watch at your leisure, either individually or as part of a bundle. And all attendees of live events get lifetime access to the class recordings too!

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

LIVE, Online Immersion Events:

Spring 2022

- Details coming soon.

You can get all the details on our training options page or just go directly to our new shop.

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