(November 11th, 2013)

If you know someone who you think 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 Chicago where we've just begun delivering the last Immersion Event of the year (30 students taking IE1 on Internals and Performance). When we return to Redmond we'll be attending the Microsoft MVP Summit, where the various Product Groups discuss new technologies and future releases. We haven't been able to make it for the last few years so we're really looking forward to it.

Don't forget to check out our 2014 class line-up – details in the final section of the newsletter.

The most recent book I've read is Allie Brosh's Hyperbole and a Half. It's based on her popular blog where she draws cartoons and commentary about things happening in her life. Brosh, The Oatmeal, and The Onion are the internet sites that regularly make me laugh. Check out The Oatmeal's latest book too: Why Grizzly Bears Should Wear Underpants. With all the serious reading I do, it's good to have something lighthearted every so often.

Please let us know if you liked what you read/saw here and/or have any suggestions for future Quick Tips.

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

Paul's Ponderings

One of the questions we get asked every so often is why a developer or DBA should know some of how SQL Server works – the internals – and isn't that just knowledge to satisfy intellectual curiosity? As we're teaching our IE1 on Internals and Performance this week, it's a good time to address that question.

Here are a couple of examples of why it's useful to know how SQL Server does things under the covers, and how it can impact your performance.

Firstly, understanding how SQL Server stores column values in records. This is really relevant when designing an efficient table schema. Just last week I saw a case where a client was indicating "whether this value has been deleted or not" using an NVARCHAR (3) column, using the values "YES" and "NO". This will take 6 bytes for "YES", four bytes for "NO", and at least

two extra bytes in either case for the pointer in the variable-length column offset array in the record. Instead, the indicator could be stored as a BIT column, which takes only 1 bit and can be combined within one byte for up to eight separate BIT columns in a record (NOTE: When there's only one bit column then the row will still reserve one byte; rows are always on byte-boundaries, not bit-boundaries). Having said that, a one billion-record table, using the NVARCHAR (3) column (let's say with half "YES" and half "NO" values) would take 500 million * 8 + 500 million * 6 = 6.5GB, compared with 1 billion * 1 = 0.93GB. That's a huge difference. Some might say that in the grand scheme of things, 5.5 extra GB is negligible, but if many more columns have inefficient design, that adds up quickly. Think of extra disk space, extra log space, extra buffer pool space, lower data density, larger backup size, longer restore times, and so on. Kimberly just recorded a Pluralsight course that covers this in detail: SQL Server: Why Physical Database Design Matters.

Secondly, understanding how page/extent allocation works. This is most relevant when dealing with tempdb PAGELATCH_XX contention. Understanding that various allocation bitmaps need to be updated in memory whenever a page or extent is allocated explains why the contention occurs when many concurrent connections are creating and dropping small temp tables. Understanding how adding more tempdb data files allows the round-robin allocation algorithm to be used explains why adding more tempdb data files can reduce the overall PAGELATCH_XX contention on in-memory copies of allocation bitmaps (by having more bitmaps to share amongst the threads instead of just one) and lead to a workload throughput increase. See this blog post for more details.

Even something as seemingly obscure and esoteric as how SQL Server computes allocation unit IDs (which I blogged about last week) is useful to know when dealing with database corruption and you need to manually find data file pages belonging to a corrupt table because *sys.sysallocunits* is damaged so queries and *DBCC CHECKDB* fail.

Call to action: Don't dismiss knowing how SQL Server works as useless information. Knowing how SQL Server stores data, uses indexes, uses statistics, manages the transaction log, manages transactions, uses locks, uses latches, and so on can really help you design and tune your databases and workloads for optimal performance. And there's a wealth of information out the, ranging from blog posts to books to online training to in-person training. With the SQL Server community burgeoning over the last few years, there's never been a time where more information is available on how SQL Server works and how you can capitalize on that knowledge.

I'm curious to hear your thoughts on using internals knowledge, so please feel free to drop me a line, always treated confidentially, of course.

Video Demo

From Glenn:

In this Insider demo video, I explain how to interpret the results from my missing index DMV query, how to find missing indexes by querying the plan cache, and how to avoid some common mistakes that people make related to missing indexes.

The video is about nine and a half minutes long and you can get it:

- In WMV format here
- In MOV format here

The demo code is available here.

Enjoy!

SQLskills Offerings

All US classes have alumnus discounts equivalent to 25%-off the full price, and many have a special offer for the first 5 people who register for each class in 2013 – buy-1-get-1-free!

Please know that these classes are final as the hotel contracts are signed, and the classes will not be cancelled or moved for any reason, nor will the dates change.

Also, we've added a few new 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

2014 Immersion Events:

Tampa, FL

- February 3-7, 2014: **IE1**: Immersion Event on Internals and Performance
- February 3-5, 2014: **IE0**: Immersion Event for the Accidental/Junior DBA
- February 6-7, 2014: **IEHW**: Immersion Event on SQL Server Hardware
- February 10-14, 2014: **IE2**: Immersion Event on Performance Tuning
- February 10-14, 2014: **IEBI**: Immersion Event on Business Intelligence
- February 17-21, 2014: **IE3**: Immersion Event on High Availability and Disaster Recovery
- February 17-21, 2014: **IEDEV**: Immersion Event for Developers

Australia

- Sydney, NSW; March 10-14, 2014: **IE1**: Immersion Event on Internals and Performance
- Melbourne, VIC; March 17-21, 2014 **IE1**: Immersion Event on Internals and Performance

Chicago, IL

- April 28 – May 2, 2014: **IE1**: Immersion Event on Internals and Performance
- April 28 – May 2, 2014: **IEBI**: Immersion Event on Business Intelligence
- May 5-6, 2014: **IEHW**: Immersion Event on SQL Server Hardware
- May 5-9, 2014: **IE2**: Immersion Event on Performance Tuning
- May 12-16, 2014: **IE3**: Immersion Event on High Availability and Disaster Recovery
- May 13-16, 2014: **IETS**: Immersion Event on Advanced Transact-SQL
- May 19-23, 2014: **IE4**: Immersion Event on Security, PowerShell, and Developer Support
- May 19-21, 2014: **IE0**: Immersion Event for the Accidental/Junior DBA

Bellevue, WA

- June 9-13, 2014: **IE1**: Immersion Event on Internals and Performance
- June 16-20, 2014: **IE2**: Immersion Event on Performance Tuning

See 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