

(September 20<sup>th</sup>, 2011)

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 bi-weekly Quick Tips is coming to you from the financial district at the tip of Manhattan Island as we're on-site with one of our clients for two weeks. Kimberly and I have been out and about at some fantastic places including the nearby [Fraunces Tavern](#) – famous for being the place where General George Washington gave a farewell dinner to his Continental Army officers in December 1783. Their dining room is in that period style (*mostly* lit with candles) – very cool!

The most recent book I've read is Jonathan Phillips' *The Fourth Crusade and the Sack of Constantinople*. Medieval history is one of my passions and the Crusades are some of the most interesting events from political, martial, and social points of view. They show the grip that the church held over many people's lives and the thirst for warfare and pillage inherent in the medieval nobility. The Fourth Crusade was a disaster and Phillips does an excellent job of explaining how it happened. Strongly recommended for history buffs.

Check out the final part of the newsletter for advance notice of a **new remote DBA service we'll be offering from October**, plus our projected class schedule through August 2012.

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

### **Paul's Ponderings**

Over the last year we've helped a few clients track down tempdb space hogs with automated monitoring built around the sys.dm\_db\_task\_space\_usage DMV to identify procedures and ad-hoc code that are misusing temp tables and generating query plans that cause memory spills to tempdb from, for example, large sorts and joins where table indexing is incorrect.

In this newsletter I'd like to describe some of the misuses of temp tables that I've seen while investigating some of the tempdb space hogs. Don't get me wrong though – temp tables are great – when they're used efficiently.

There are three main problems we see:

- Over-population of temp tables
- Incorrect indexing on temp tables
- Using a temp table where none are required

The first problem involves creating a temp table using a `SELECT ... INTO #temptable` construct and pulling in far more data into the temp table than is necessary.

The most common thing we see is pulling lots of user table columns into the temp table, where some of the columns are not used ever again in subsequent code. This is a HUGE waste of IO and CPU resources (extracting the columns from the user table in the first place) and a horrible waste of tempdb space (storing the columns in the temp table). I've seen code pulling in large varchar columns (without reason) and with million-plus row datasets...

The other facet of over-population of temp tables is pulling in too many rows. For instance, if your code is interested in what happened over the last 12 months, you don't need to pull in all the data from the last ten years. Not only will it be bloating the temp table, it will also drastically slow down the query operations.

The key to better performance is making sure your selection/projection is as focused as possible. To limit your selection, use an effective `WHERE` clause. To limit your projection, list only the necessary columns in your select list.

The second problem involves either creating indexes before populating the table (so that no statistics are generated) or creating a bunch of inappropriate indexes that are not used. The most common example we see is creating a single-column nonclustered index for each temp table column that is subsequently used in a multi-table join. Those are just taking up space for no use whatsoever. Temp tables *\*DO\** need indexes (preferably after load) but as with any form of query tuning – *only* the `RIGHT` indexes. Consider creating permanent tables that mimic what's going on in your temporary objects and then using DTA to see if it has recommendations. While DTA's not perfect, it's often WAY better than guessing.

The final problem is when a temp table is used when it is not needed. The SQL Server query optimizer is a fabulous beast and is very good at figuring out the most efficient way to execute most queries. If you choose to take some of the query operation and pre-calculate it into a temp table, sometimes you're causing more harm than good. Any time you populate a temp table you're forcing SQL Server to materialize the complete set of results of whatever query you ran. This can really limit SQL Server's ability to produce a pipeline of data flowing efficiently through a query plan and making use of parallelism and collapsing data flows when possible.

While it's true that you might be able to do better than the optimizer sometimes, don't expect that it's the case all the time. Don't just go straight to using temp tables, give the optimizer a chance – and, make sure to retest your code/expectations around service packs and hot fixes as these may have eliminated the need for temp tables as well.

In one recent example I struggled to figure out why a temp table was even being used. It contributed up to 50GB of tempdb space for a query that was run many times per day. Taking the tempdb-creation code and embedding it as a derived table in the main query completely removed the tempdb usage and took the query from 17 minutes down to a few seconds. Derived tables are a great form of query hint – and always something to try before moving to temp tables.

**Summary:** if possible, limit the amount of data that temp tables hold, create appropriate indexes after the temp table is populated, and make sure that using a temp table is actually more efficient than just letting SQL Server process the whole query in one go.

**Call to action:** take a look at your temp table usage. You may be surprised to find a lot of tempdb space and CPU resources being consumed by inappropriate temp table usage and incorrect temp table indexing. Or if you don't have time and you're suffering from horrible tempdb problems, give us a call to help you.

I'm really interested to know your thoughts on temp table usage—feel free to [drop me a line](#), confidentially as always.

### **Video Demo**

Last week I was asked whether using a clustered index was a cure for the *inconsistent analysis* problem that is inherent in the read-committed isolation level. The answer is no and I thought I'd record a short demo video showing you the problem. The video is about 6 minutes long.

I produced the video in WMV and MOV formats so everyone can watch. You can get the videos:

- For WMV: [here](#)
- For MOV: [here](#)

I recommend downloading before watching. And you can get the demo code [here](#).

### **SQLskills Offerings**

In October we're launching a new "remote DBA" service where we provide repeated mini-health audits on regular intervals (once a server has been initially health-checked) plus automated monitoring of SQL Server for problems – and what to do when a problem inevitably \*does\* arise. This is an excellent way to gain access to our team's incomparable expertise and experience – especially if you don't have a full-time DBA on staff. However, even if you do, this gives you an additional set of expert eyes to watch over your critical data.

[Let me know](#) if you're interested in getting more details. Space is going to be limited in our initial roll-out so you're hearing it here first before we announce it formally in a couple of weeks.

Registrations are continuing for our remaining Immersion Events this year – a \*new\* Developer Immersion in Chicago, October 24-28, and Internals and Performance (IE1) plus a \*new\* BI Immersion in Atlanta, December 5-9. **The IE2 offering in Chicago in October is now sold out!**

For everyone that's been asking – YES – we will be offering all four of our Immersion Events in 2012 with these classes being added to our schedule starting in October. Here's a list of some of the classes and cities we're targeting:

- Available for registration in October:

- February 2012: Internals and Performance (IE1) in Tampa, FL
  - March 2012: Performance Tuning (IE2) in Tampa, FL
  - March 2012: High Availability/Disaster Recovery (IE3) in Tampa, FL
- Available for registration December/January
  - April 2012: IE1 & IE2 in Chicago, IL
  - May 2012: IE1 & IE2 in London, UK
  - August 2012: IE1-IE2-IE3-IE4 in Redmond/Bellevue (WA) again!

See [here](#) for all the details.

### **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