(October 17th, 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 New York, where we're on the road again for three weeks after being at home during this year's PASS Summit. The Summit was excellent yet again and I believe it was the largest one ever with more than 4000 attendees. Jon, Kimberly, and I all had a blast presenting our sessions (and Jon's XEvents pre-con), and attending other people's sessions. There's a lot of cool stuff coming in SQL Server 2012!

The most recent book I've read is E.M. Forster's *A Passage to India*. I love reading books about India and this one is a masterful portrayal of the British Raj in the early 20th century – their haughty arrogance as colonialists and their misunderstanding of the Indian society and culture they're in. The characters are involved in a scandal which brings out the heated worst behavior in both sides of the community. I strongly recommend it.

Check out the final part of the newsletter for details of our **new remote DBA service**, plus our projected class schedule through August 2012. Our winter 2012 classes are now open for registration!

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 couple of years the majority of the consulting clients Kimberly has dealt with have been using or wanted to use partitioning in some way. Before that? Not so much. It seemed to take quite a bit of time before interest in partitioning really kicked in. Partially, we suspect that it just took time after SQL Server 2005 shipped for the data volumes to increase to the point where it's reasonably common to encounter partitioning.

However, that's just the Enterprise Edition Partitioned Tables feature that was introduced in SQL Server 2005. You don't need Enterprise Edition to effectively partition – you can split your data into multiple tables and then use a view (Partitioned Views) to access the entire data set (or portions of it) when required. And in fact there are very good reasons NOT to use table/index partitioning as the ***sole*** source of partitioning goodness.

You don't strictly *have* to partition just because your data volume becomes really large, but the benefits of doing so can be tremendous. So what are the possible benefits?

- Manageability

- Availability/recoverability
- Performance

For manageability, I'm talking about adding and removing large chunks of data to/from your dataset. I'm sure you've come across trying to delete several million rows from a table – its slooooow. But with partition switching, it can be a metadata-only operation. And the same thing applies to inserting data. This is the core concept of the *sliding-window* scenario.

There's also the issue of maintaining indexes. If you have one gigantic clustered index, removing fragmentation in the 'current' portion of the data means operating on *all* the data. With partitioning, you can just remove fragmentation in the data partition that is fragmented – and this is one of the best benefits of using Partitioned Views (PVs) over Partitioned Tables (PTs). With PVs you split the data into separate tables (e.g. by year) and rebuild/defrag the 'current' table using an online operation With PTs, you can rebuild at the partition-level but only using an offline rebuild. So you might want to use the partitioning strategy that Kimberly promotes, which is to combine table/index partitioning of most of the data, with a standalone table that is the current portion of the data that gets fragmented, and a view over everything. The standalone table *can* be rebuilt online.

Often a hybrid approach (using BOTH PTs and PVs) is actually best.

For availability/recoverability, spreading data across separate filegroups is beneficial because when a disaster occurs, you restore only those filegroups that are damaged as well as only those necessary to get the most critical applications running again. This is partial database availability and along with PTs and Online Operations requires Enterprise Edition. But, if you have a lot of data (e.g. TBs), having the ability to do this and vastly reduce downtime can be worth the price of the Enterprise licenses.

Coincidentally, as I'm writing this, Kimberly is on stage at PASS discussing VLDB disaster recovery strategies – go Kimberly!

But, really leveraging these features requires architecting for them. You can't restore individual filegroups when you only have a single filegroup – you must be partitioning data in some way to leverage this. Even this can be done effectively with or without table/index partitioning as you can place specific tables and indexes in their own filegroups.

And what about performance? Partitioning is not inherently a performance feature – plain and simple. However, by moving parts of your data to isolated portions of your I/O subsystem, you may be able to increase performance (for instance, a table that is both for DW and OLTP queries can be split to separate the sequential and random I/O characteristics of the data access). Also, by splitting data into separate tables you might be able to have better statistics and/or more frequent maintenance to the core, critical components which may also improve performance. There's actually a lot here and not as much time to cover it with just a quick newsletter. This is something that we hope to blog more!

What's interesting though is that people often think that by partitioning the data it will help limit how much data is processed by a query – but that's no different than just using the correct indexes so that the search/scan predicates can be efficiently processed. There are some benefits but that's definitely not the primary reason to partition!

Summary: partitioning is often very misunderstood or not even considered, and can have a dramatic effect on your environment when implemented for the right reasons. This has been a real whirlwind introduction to why you may want to consider it. For much more in-depth guidance, check out the two whitepapers on partitioning:

- Partitioned Tables and Indexes in SQL Server 2005 (which Kimberly wrote)
- Partitioned Table and Index Strategies Using SQL Server 2008

Call to action: consider the size of your data and the operations you perform on it (or struggle to perform on it) and think whether some form of partitioning would help you out – either in terms of manageability, scalability, or even performance.

I'm really interested to know your thoughts on partitioning—feel free to drop me a line, confidentially as always.

Video Demo

Last week at PASS someone came up before my session and asked about how to figure out what's using space in tempdb. I have some code that analyzes it and thought it would make an interesting demo for you. Additionally, I was shocked by how many folks aren't using SQL Sentry's awesome and free Plan Explorer tool so I thought I'd add a quick introduction to it as well. The entire video is about 7 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

We've just launched a new "remote DBA" service where we provide repeated mini-health checks 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.

Registrations are continuing for the three remaining Immersion Events left this year – a *new* Developer Immersion Event in Chicago, October 24-28, our Internals and Performance (IE1) plus a *new* BI Immersion Event in Atlanta, December 5-9. Our IE2 offering in Chicago in October is 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 **NOW**:
  - February 27 – March 2, 2012: Internals and Performance (IE1) in Tampa, FL
  - March 5-9, 2012: Performance Tuning (IE2) in Tampa, FL
  - March 12-16, 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