



RUNAS RADIO



<http://www.runasradio.com>



Richard
Campbell

RunAs Radio is a weekly Internet Audio Talk Show for IT Professionals working with Microsoft products. The full range of IT topics is covered from a Microsoft-centric viewpoint.



Greg
Hughes

Text Transcript of Show #0103
(Transcription services provided by [PWOP Productions](#))



Bill Graziano Tunes Up Our SQL Server Queries!
April 1, 2009



Bill Graziano Tunes Up Our SQL Server Queries!

April 1, 2009

[Music]

Brandon Wenn: From runasradio.com, you're listening to RunAs Radio, the Internet audio talk show for IT professionals with Richard Campbell and Greg Hughes. This is Brandon Wenn, announcing show #103, with guest Bill Graziano, recorded Thursday, March 12, 2009. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com. You can follow the boys on Twitter at twitter.com/runasradio.

Richard Campbell: You're listening to RunAs Radio. I'm your host Richard Campbell, with me as always my co-host, Greg Hughes.

Greg Hughes: Hey, Richard, how are you?

Richard Campbell: I'm having a great day, man. You know, my house renovation is still going on.

Greg Hughes: Yeah, cool. So where are you at in that process? I know I've seen pictures along the way but it has been a little while.

Richard Campbell: We are at the paint, tile, and cabinetry process so I think we're about two months out of being finished.

Greg Hughes: Ah, very nice because you moved to a different house down the road and completely chopped that thing down and you're doing some really cool -- of course you're doing some pretty extravagant and cool IT work in your renovation.

Richard Campbell: Everybody should have 60 user rack space in their house, man.

Greg Hughes: Amazing, and what kind of network infrastructure are you putting enclosed?

Richard Campbell: It's all gigabyte, four internet cables to two walls of each room. These are basic requirements, man.

Greg Hughes: Yeah.

Richard Campbell: And I actually built a proper location my house for the wireless access points. They're wired in position and, you know, get proper coverage just thinking through the core issues.

Greg Hughes: Why plural wireless points?

Richard Campbell: Well, we all need fabric.

Greg Hughes: Right, exactly.

Richard Campbell: What are you thinking?

Greg Hughes: So did you do wire in the walls or did you do fiber?

Richard Campbell: Wire. I seriously considered fiber but you know with gigabyte Ethernet I don't think it's worth it.

Greg Hughes: Yeah. Maybe five years ago, we would have said fiber if you really wanted to get fancy but you can do an awful lot with the wire now.

Richard Campbell: Yeah. I seriously considered fiber a number of times, but you know, the reality is both in work and in home I only run fiber when you got to go long range. As soon as you're over 600 meters, now fiber matters but other than that it is worth the issues.

Greg Hughes: Yeah, yeah, you're right.

Richard Campbell: All right, Greg, let's introduce our. Bill Graziano is a specialist in SQL Server performance tuning and server management. Bill has nearly twenty years of database management experience. He started working with Sybase on a UNIX platform back before it became Microsoft SQL Server. He's worked with every version of SQL Server Microsoft has ever released. Bill has been a Microsoft MVP for SQL Server for four years. He is on the Board of Directors for PASS, the global user group for SQL Server professionals where he is the Vice President of Marketing. Bill is a regular presenter at the PASS conference and various user groups. He formerly worked for Accenture and Empower Trainings & Consultants. Welcome, Bill.

Greg Hughes: Hi, Bill.

Bill Graziano: Thank you.

Richard Campbell: Before we go, I really want to talk performance tuning, it's one of my favorite topics, but I think we should tell people a little bit about PASS and you sound like the guy we should talk to about that.

Greg Hughes: Yeah.

Bill Graziano: Or I should be, shouldn't I? I don't know, any organization that lets a Database Administrator, be it the VP of Marketing kind of wonder about...

Richard Campbell: I mean, PASS is all SQL Server people, right?

Bill Graziano: It is all SQL Server people, or really it's anybody that works with SQL Server. It could be database administrators, it could be



developers, it could be report developers, business intelligence professionals, access developers that are working with SQL Server, anybody that's using SQL Server, we are interested in trying to help out.

Greg Hughes: So the business side, not just the DBA type of people.

Bill Graziano: No. It's much broader than that. The big thing that we put on each year at our conference which is coming up in November...

Greg Hughes: Right.

Bill Graziano: You know, maybe 35% to 40% of our content there is related specifically to DBA work. The rest is application development, it's business intelligence, it's reporting, it's just a variety of topics around SQL Server.

Richard Campbell: Sure and it's a huge show. I mean you guys do really well in terms of size of conference. It's pretty exciting just to have a totally SQL Server-focus conference. Most show, as you go through, it's a track, not the whole show.

Bill Graziano: Correct. We have nearly 150 technical sessions related to SQL Server. We had 2500 people at our last conference which was last November. Given the economy, we were really, really happy with that. We had a European conference coming up in April in Neuss, Germany which is full of several Deuseldorf and we're really happy. The registration so far this year is well ahead of where they were last year so we're really hungry for SQL Server information.

Greg Hughes: Great.

Richard Campbell: That's going to be in just a few weeks after the show comes out so Neuss Germany, the European PASS conference.

Bill Graziano: That's correct.

Greg Hughes: Where's the next United States conference? For the people that are...

Bill Graziano: The next one is going to be in Seattle. We'll be there the week of November 3rd. We had quite a few in Seattle lately. At our last conference, we had over 300 members of the Microsoft Development Team. So there are really an awful lot of benefits for us being in Seattle and people being able to come and get their questions answered from those folks.

Greg Hughes: Sure.

Bill Graziano Tunes Up Our SQL Server Queries!

April 1, 2009

Richard Campbell: And literally from the guys who wrote the product.

Bill Graziano: Yeah.

Richard Campbell: And so that's just pretty compelling.

Bill Graziano: So you ask him a question and when they say, well, when I wrote the code that did that; this is why I wrote it that way.

Richard Campbell: I love that.

Greg Hughes: Exactly.

Richard Campbell: So you really get brass tax type stuff. So let's get into the performance tuning side of things? What sort of work do you do that you run into these kinds of problems?

Bill Graziano: From an independent consultant and I lot of times when I get called in, it's something is broken or something is slow and there's not always a lot of time to do planning and to lay out capacity plans and things like that. It's this is broken, fix it now.

Richard Campbell: Right.

Bill Graziano: So a lot of the performance tuning that I do is trying to figure out in a really short amount of time which queries are a problem, easy queries that are a problem, where are the bottlenecks and then trying to alleviate those pretty quickly while the system is running.

Richard Campbell: And so I got to think the SQL profiler is your friend.

Bill Graziano: SQL profiler is very much my friend. I spent an awful amount of time on that. I actually wrote the little utility of that use that I have released out called ClearTrace and what it does is when you run profiler it collects all those queries up, and to use stored procedures you can sometimes match up at this one array in, you know x a thousand times at a certain time frame.

Richard Campbell: Right.

Bill Graziano: What this does is a lot very similar to what read80 trace does which is the tool that Microsoft product support wrote and it aggregates all the SQL that looks the same and tells you in aggregate which queries are using the most disk resources, the most CPU, and really gives me right away these are the ones to go look at.

Greg Hughes: Cool.



Bill Graziano Tunes Up Our SQL Server Queries!

April 1, 2009

Richard Campbell: I'm surprised at how many people just don't know about profiler, that that's the tool that really bubbles the queries of concern to the top.

Bill Graziano: Yeah. You know, it's interesting to know that there's a tool out there that when you turn it on, it can look for, I think it's something like 140 different events that happened in SQL Server. You know, log ins, log outs, and a SQL statement completing the most common ones, but everything from locks to any error that the user sees you can also capture in profiler and then tell you statistics about that, what machine did it come from, how much CPU did it use, when did it occur. In fact, SQL Server is always running a trace in the background that you can go access it if you want and it's capturing information that it thinks is interesting to how the system performs.

Greg Hughes: So the tool that you released is that taking profilers output and then doing more than intelligence on that and generating new information for you to consume. Is that what it does?

Bill Graziano: That's exactly what it does. The big thing that it does is it takes the SQL statements and I try and parameterize those. So where you pass something into a where clause with a certain specific value, I try and turn that into a parameter and then I match that text against the other SQL statements.

Greg Hughes: Ah okay.

Bill Graziano: So if you select some stuff from a customer table with a certain where clause and you run out different values, I should be able to match all those up and then tell you in aggregate how often that query ran and how much CPU it used.

Greg Hughes: Sure. It sounds like you're sort of taking maybe a lot of the work that you would normally have to do by hand in automating that, at least the heavy lifting part of that so you can then take a look at it and try to solve the problem.

Bill Graziano: That's the real idea. The thing that we're really trying to get to is a lot of people when they turn on profiler, they filter for just queries that use a lot of CPU or just create that use a lot of disk and say that might get you an individual query that runs once of twice, but what you intend is that something that maybe is somewhat efficient but runs again and again and again ends up being your biggest consumer resources. So if you can make a 10% gain in something that runs, you know, 20 times a second, that adds up.

Greg Hughes: Right. So again what's the name of the tool that you did and you said you released it, and can people check it out?

Bill Graziano: The tool is called ClearTrace.

Greg Hughes: Okay.

Bill Graziano: If you go to the corporate website that I do my consulting under www.cleardata.biz, it's just a free download there.

Greg Hughes: Great.

Bill Graziano: Anybody can download it and use it.

Greg Hughes: Free is good.

Richard Campbell: Absolutely. I worry about how much of a performance tuning is focused around the infrastructure of SQL Server versus just a data you just got. I find that most people have performance problems just because the database has grown large now, probably larger than it needs to be, and they don't have a good archiving strategy just to keep their datasets under reasonable control. What's your experience in that area? Is it more about the execution of queries or just the size of the data? Where do people get their pain?

Bill Graziano: You know, I would say yes to all those as to where is the pain.

Richard Campbell: There's pain everywhere.

Bill Graziano: You know, one of the interesting things about the world now is that we have pretty amazing abilities to generate data and then we generate the data above the data and then other people collect data above the data that we generated and then sell that as more data so we end up with lots and lots of data. You know, we do see a lot of situations where people write a system, they test it, they load it up with certain amount of data and it works just fine, and then two or three or four years later the query plans or the queries they wrote that were pretty efficient on a million rows become less efficient on a hundred million rows.

Richard Campbell: Right.

Bill Graziano: So that, you know, data growth is a huge part of that. People begin to, you know, as you begin to add on the systems, I find that even companies that do really good performance testing upfront don't always do as good a performance testing as they put in what they consider a small change and that small change can have a pretty big impact.



Bill Graziano Tunes Up Our SQL Server Queries!

April 1, 2009

Greg Hughes: I know I've seen small changes have major impact where it just wasn't necessarily assumed that that would be the case.

Bill Graziano: Yeah. My favorite story about that is we were at the client and I have been doing some performance tuning for them and we made pretty good headway and then one morning the system just, it just literally shut down. The CPU would max out, the disk would max out, it would churn and become unresponsive and we couldn't figure out what it was. Nobody admitted changing anything. So we finally found the developer that admitted they had changed the one line of code and all it did was wrap a function around the column to limit the result set to just the previous seven days and the problem is as soon as they wrap a function around the column, it prevented SQL Server from using the index on that column.

Richard Campbell: Right.

Greg Hughes: Aha, aha.

Bill Graziano: Because the SQL Server will only use an index if you compare a column to a value. So the query switched from being a somewhat efficient index query to a table scan.

Greg Hughes: Right.

Bill Graziano: And since it was on the report that all the sales people use to see what calls they needed to make, it ran a lot and so we changed that one line of code and everything was better.

Richard Campbell: Wow.

Greg Hughes: Cool.

Richard Campbell: You talked a bit about procedure caching as well. That is more of the execution plans for SQL Server?

Bill Graziano: The SQL Server is kind of interesting base in that most programming that you do you're telling it how to do things, but in SQL Server you're telling it what result you want.

Richard Campbell: Right.

Bill Graziano: And so it's building its own query plan in there and so having it pick a good query plan and continue to reuse that plan is becoming one of the more interesting challenges. It's interesting. A SQL Server is progressed and they have given us better and better tools to see what goes on inside it. It really seem performance tuning kind of moved from a measured disk usage, a measured CPU and then

we got good tools to see query plans. People began to look at those. With 2005, we began to get really good tools to see what was going on in the procedure cache and now people are much more interested in Windows SQL Server create a query plan, Windows it reused it, and how do it minimize the amount of recompiles that stored procedures have.

Richard Campbell: My first question has got to be how much time in a typical query execution is dedicated to computing the query plan? Is that really a big chunk of the total time?

Bill Graziano: That's one of thing things that I wish I had a better answer for because that's one of the things that are not exposed as well. You can kind of get to it by forcing it to recompile each time.

Richard Campbell: Right.

Bill Graziano: And where you really see the impact is, you know, if you're going to run a query that's going to do a data warehouse, a big data warehouse where you can scan a million rows, the compile time for that is not going to be that big. If you're writing something that's online, it's processing transactions like maybe a credit card system and so you're looking at stored procedures or transactions that run in 10, 20, 30, 40, 50 milliseconds, then that query compile time can start to add up on that.

Richard Campbell: Yeah, that makes sense to me especially if you're hitting it a hundred times a minute or a thousand times a minute, that really will add up.

Bill Graziano: Yes, yes it does and the other thing that you get is in a lot of those OLTP systems you get a very define set of transactions which should be pretty easy to write a stored procedure and get a stable query plan, but occasionally you get people doing things inside those that makes it harder for you to reuse query plan. There's a big issue that I think a lot of people know about and began to work around but creating temporary tables and then using those temporary tables in the queries can often force stored procedures to recompile in the middle of the stored procedure.

Richard Campbell: I see. So when you want to build really performing stored procedure things, do you clear a temporary table?

Bill Graziano: Well, you know, the best thing about SQL Server is that almost any question you can ask me, the answer is it depends.

Richard Campbell: Yes.

Bill Graziano: But yeah, that's a good rule of thumb, it's that temporary tables can add a fair



Bill Graziano Tunes Up Our SQL Server Queries!

April 1, 2009

amount of overhead and can cause recompile. There are ways that you can use them in terms of creating them ahead of time and populating them and then using them, as oppose to creating them in the middle of the stored procedure that make life a lot easier on you.

Richard Campbell: Again I'm not saying never use temporary tables, but if I'm building a procedure that's highly performing, I want to run it in a fraction of a second and I want to run it a lot, then perhaps I've got to steer clear to temporary tables and find a different way to solve the problem.

Bill Graziano: I would say anything you can do to avoid temporary tables, anything you can do to avoid creating intermediate datasets...

Richard Campbell: Right.

Bill Graziano: Whether it's a temporary table, whether it's a table variable, anything you can do to avoid intermediate datasets, you'd be better off.

Richard Campbell: My next question would be as a table variable, can it behave differently from a temporary table?

Bill Graziano: It does behave differently and it's better in some cases and not as good in others.

Richard Campbell: Right.

Bill Graziano: So if you've got a really small intermediate dataset and for whatever reason you do need to use something like that, then a table variable is better. The larger it gets, the more appropriate a temporary table is.

Richard Campbell: Well, because in a temporary table, I can actually build in the access and things on it if I need to, but again, that would be the sort of thing I'm not looking at a high performance query there. I'm looking at a query that's doing something very elaborate.

Bill Graziano: Correct.

Richard Campbell: That makes sense to me. But I think the table variable, because it's got such a clear beginning and endpoint that you define it in line, you know it dies at the end of the stored procedure, it works better in those scenarios.

Bill Graziano: You know, it can work very good but typically whenever you see a table variable, it's always going to do a scan of that table variable.

Richard Campbell: Right.

Bill Graziano: Where like you said in a temporary table, you can build indexes, you can do index seeks on that, so it really depends on your used case.

Richard Campbell: Sure. Yeah, the correct answer is always it depends but I'm with you on the idea that if I build the stored procedure I want to tune, the one that profiler shows is used a lot, even if it's efficient it's just that the frequency of use is really high and so that's a substantial point of our work, I'm going to go through with the fine toothcomb, strip out anything I don't absolutely have to have in there, and I've got to imagine that in many cases you can find a different way to build your procedure without those temporary variables.

Bill Graziano: That's almost always true. There's a lot you can do with, especially now with the newer version, the SQL Server with kind of table expressions...

Richard Campbell: Right.

Bill Graziano: Which kind of let's you build an intermediate result that's on the fly. I do an awful lot with derived tables. So your Select from inside your instead of being in a table name, it's actually another select statement.

Richard Campbell: Right.

Bill Graziano: I also try and do -- I think views inside SQL Server are really underused.

Richard Campbell: I agree but where do see the view as a strength?

Bill Graziano: Also there are a couple of things they do, they do really well. I can't remember, I'm a pretty lazy DBA...

Richard Campbell: Okay.

Bill Graziano: I don't like to type much and it's a little thing, but you know, if I can build me some views to give me a lot of the information I need, have a lot of those joins done and then use those, what you'll find is that you can look like, in your mind, in how you think about it, you can build those intermediary -- those intermediate result sets. I'm going to select from this view and then I'm going to do X, Y, and Z to it. But what SQL Server does is it doesn't compile the full query with the view. So it doesn't run the view and then run the query. It collapses those together and runs that in result. I don't know if I've explained that very well.

Richard Campbell: I get your point that the fact is that view is not always materialized. You know, it



doesn't pre-computed ahead of time, it does it as it needs it and because it combines those two things together, maybe you can find a more efficient way to render the data.

Bill Graziano: Correct. Not only is the view not materialized, but the plan for how to execute the view is not materialized.

Richard Campbell: Right.

Bill Graziano: So if you have a select statement that only uses certain columns that are there, when it builds that plan it will only build the plan for those certain columns.

Richard Campbell: I thought that one of the big strength of views was the ability to build indexes against the view to build a composite index that otherwise couldn't exist.

Bill Graziano: So you could build views like that but in that case you're limited to I think it's just the enterprise edition of the SQL Server.

Richard Campbell: Okay.

Bill Graziano: So that's not a feature that's available in all the versions, but those are pretty powerful tools for doing things that you otherwise couldn't do.

Richard Campbell: I'm definitely thinking from a performance tuning point of view. The ability to say I'm not only going to create this join, I'm going to index columns between those two tables through this view to improve performance, that to me seems like a huge win.

Bill Graziano: Yeah and if you look at running that view and it stays up-to-date and you build those indexes on it, in effect you're not going back to the original table, you're building to that index view.

Richard Campbell: And what you're doing is now pushing the load off the query that you're retrieving from onto the inserts, updates, and deletes statements because now every time it does anything on those tables it has to maintain this index.

Bill Graziano: That's correct, that's the damn site.

Richard Campbell: So you're still paying the price, you're just paying it somewhere else.

Bill Graziano: I wish there more things that were free but...

Bill Graziano Tunes Up Our SQL Server Queries!

April 1, 2009

Richard Campbell: Yeah. Oddly enough there's nothing free. You're always finding you're shifting the cost around and maybe the query is performance sensitive and so that time really, really matters where the inserts, updates, and deletes are not as sensitive.

Bill Graziano: Yeah, very true.

Richard Campbell: You were talking about the tools that allow us to look into the procedure cache. What are those tools? How do we look into that?

Bill Graziano: So the biggest things that you can use are in SQL Server 2005. They added the dynamic management views or the DMVs.

Richard Campbell: Right.

Bill Graziano: And so those lets you -- basically there are queries that will tell you what's in every nook and cranny of SQL Server's memory and so the big thing that is kind of a surprise to me as I began to play with some of these is just how much of SQL Server's memory is allocated to the procedure cache, and so you might in a fairly low system that's running at 2 gigs of memory, so SQL Server gets 1.6, 1.7 gig of memory, you might find that maybe 40% of that or as much as half of that is procedure cache and not data cache.

Richard Campbell: Interesting.

Bill Graziano: So the more you can reuse query plans and have fewer query plans, the more memory you get stored in data.

Richard Campbell: Which is really what you thought you're storing in the first place.

Bill Graziano: Which is exactly what I thought I was storing in the first place.

Richard Campbell: Well, I'm surprised with the number of people who just have not ever sat down and learn DMVs as well. There's a whole lot of unique functionality there that didn't exist anywhere else.

Bill Graziano: There's a tremendous amount of functionality and, you know, as much as I work with it everyday, I'm always amazed at how often I came over something I want to know and I'm like, you know what, I'm pretty sure there's a DMV that will tell me that and I have to go read around and find it and figure out how to use it, but there are certainly a lot of them.

Richard Campbell: Yeah, just tons and tons of stuff and 2008 didn't make it any better, there's more.



Bill Graziano Tunes Up Our SQL Server Queries!

April 1, 2009

Bill Graziano: Yeah. I don't know, I don't know if that's, I don't know if I see that as worse.

Richard Campbell: Yeah, one says you say there's more visibility and then on the other side you say there's more stuff to know.

Bill Graziano: Yeah, yeah, and you know I think it will just take awhile for folks to get used to going to look there in the first place.

Richard Campbell: Right. I don't think people look there actually but I also like the fact that we can actually look and see what is cache, why are we faster now for this, and I think it's one of these big reality checks when you look at what was getting cached early in the product when we were doing our detailed performance tuning and what's being cache now, a year, two years down the road that sets and got so much larger and that the cache has a much tougher time.

Bill Graziano: Yup and you can even tell it, you know, not only what's cache but what is the query point for that particular item.

Richard Campbell: So actually going and looking to see what the query plan that has been cached is.

Bill Graziano: Yeah and it will tell you how many times it's been used, when it was last used, things like that.

Richard Campbell: That's powerful stuff. Have you run into circumstances where the query plan that gets cached, that you find in a procedure cache, is different from the one you'd see in query analyzer when you went to check it?

Bill Graziano: So what I tend to run into a lot is I'll get a query that's performing -- so the short answer is yes. The long answer involves something called parameter sniffing.

Richard Campbell: Oh yeah.

Bill Graziano: Where when you first build a query plan, it's got to make decisions about what plan to build and if you're going to make those decisions based on statistical averages, let's say you're comparing a data value to a parameter passed into the stored procedure. It either make that decision based on the explicit value that you passed in, or it makes that decision based on statistical averages of what's in the table. With 2005 and 2008, you get a great ability to tell it this is how I want you to build this query plan.

Richard Campbell: Right.

Bill Graziano: Indicates it's not building the one that you wanted it to.

Richard Campbell: So you can actually provide a hint to say I'm going to tell you what the likely values of this parameter are going to be, rather than you have to analyze the table to figure that out?

Bill Graziano: Correct and a perfect example of that might be like an active/inactive flag where you've got 99% are active but the very first time you call it, if you call it with the inactive flag it might build a very interesting query plan. The other thing that you can do that's really interesting that they put in specifically for packaged applications is that you can actually take a SQL statement so you can take the SQL statement basically out of that DMV and say whenever you see a SQL statement that looks like this, use this kind of hint to run that or this kind of plan.

Richard Campbell: Interesting, and what kind of hints can we provide it?

Bill Graziano: So the ones that I'm most familiar with are telling it when to recompile or when not to recompile, or giving it hints for specific parameter values. You know, dates we use a fair amount of. You know, just compile this plan assuming I'm giving you this kind of date...

Richard Campbell: Right.

Bill Graziano: Or an active flag or something like that. I don't find myself using those a whole lot.

Richard Campbell: Well, this is awfully close to the metal, like it's one of those things where you feel like we're getting to the point where odds are our guess is wrong.

Bill Graziano: Well, it's one of those things where once you overwrite, it default behavior. SQL Server can't correct if the scope of data changes or things like that.

Richard Campbell: Right. Yeah, you are now taking ownership.

Bill Graziano: Yeah. So I'm really trying to avoid that if I can. You know, if you can get acceptable query performance without doing that, I think you're always better off and some of those are fairly harmless like you say optimize for the active flag being through.

Richard Campbell: Right.

Bill Graziano: That's probably not going to change any time set.



Richard Campbell: But this also makes a huge case for why we shouldn't just have strings as parameters and be flipping types inside of our stored procedures.

Bill Graziano: Spoken like a true developer!

Richard Campbell: Thanks. You know, I just showed my chops there, didn't I?

Bill Graziano: Oh no. Well, actually it's interesting. It's one of the most interesting challenges to face in stored procedures and I think it's really kind of risen to the forefront now that people are better understanding query plans as how do you do with optional parameters.

Richard Campbell: Right.

Bill Graziano: Because there's just no good way in static code to do it. A lot of people are trying to get smart and they use a call less around the column in the where clause and you know, that's the only thing that doesn't guarantee you'll get a table scan in every table that you use that on.

Richard Campbell: Yeah.

Bill Graziano: Or that won't use the index, they might use a different one.

Richard Campbell: Yeah, that's the run slowly hint.

Bill Graziano: Yes, that's exactly what that is and you know there was a feature put in 2008 that you could use the option recompile hemmed with optional parameters and it would supposedly build you the correct query plan even if you use that call less.

Richard Campbell: Yeah.

Bill Graziano: Or if you use the [is no 28:19] or equals value. The problem is that there's a pretty nasty bug with that so they're saying that you should not use that and really all that leaves you is building up some type of a string, of a row select statement and issuing that.

Richard Campbell: And once again, you know, you're going to be re-computing your query plan all the time.

Bill Graziano: Yup, yup. Although if you're smart and you build a parameterize SQL statement and the text of the statement itself is identical every time and if you're always searching like customer numbers or something like that, it will reuse that portion.

Richard Campbell: Great. So I mean here's the thing, it's SQL Server really does try to help us here but there are limits to what it can do, and I find it interesting, you know, what you're saying here about there are optimizations we get to have to have long term ramifications so the fastest executing solution may not necessarily be the best in the long run.

Bill Graziano: It may not, it may not and especially look at systems that grow as fast as somebody can today...

Richard Campbell: Yeah.

Bill Graziano: You know, if I'm running a Twitter or a MySpace or Facebook or something and you get a million people a day joining, oh goodness.

Richard Campbell: Yeah, interesting problem.

Bill Graziano: Yeah.

Richard Campbell: We're coming close to the end here, Bill. Any final words, areas we haven't talked about here, stuff that people should be looking at for performance tuning?

Bill Graziano: You know, I guess my only word of advice is don't make it harder than it has to be.

Richard Campbell: Right.

Bill Graziano: And 9 times out of 10 I can solve a performance problem with just better indexing so I encourage folks to just look at how their queries are running, look for table scans and query plans and see if they can't put better indexing on, in many cases they will be able to solve their own problem.

Richard Campbell: So the fundamentals are still the main thing, isn't it? Don't worry, the DMVs always are the things that are cool but in the end a good index strategy is the best benefit.

Bill Graziano: Yeah, you're just kind of a progression you go through, you know, is this in good shape, okay, then just move on to the next thing, and where I see problems time and time again or as they get bigger, the indexes that were needed initially were probably just the primary key, and the foreign key, to starting on.

Richard Campbell: Yeah, that makes sense.

Bill Graziano: Again and again and again, just a few well placed indexes can knock down those top four, five queries and everything is just fine.



Bill Graziano Tunes Up Our SQL Server Queries!
April 1, 2009

Richard Campbell: Bill Graziano, thanks so much for coming on the show.

Greg Hughes: Thanks Bill.

Bill Graziano: Thanks for having me.

Richard Campbell: And we'll talk to you next week on RunAs Radio.