RunAs Radio

http://www.runasradio.com

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.

Text Transcript of Show #058
(Transcription services provided by PWOP Productions)

Erin Welker Partitions Our Data!
May 21, 2008

[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 #58, with guest Erin Welker, recorded Thursday, May 8, 2008. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com.

Richard Campbell: Hi, this is Richard Campbell. You're listening to RunAs Radio and with me as always, my co-host, Greg Hughes.

Greg Hughes: Hey everybody. How are you doing, Richard?

Richard Campbell: I am well, sir. How is Portland today?

Greg Hughes: It's good. It's kind of gray, but we've had a couple nice days. I got the boat out last weekend and had a lot of fund.

Richard Campbell: Nice.

Greg Hughes: Yeah.

Richard Campbell: Out on the salt?

Greg Hughes: No, on the freshwater, on the river. I went down to Willamette River. It's a little boat and low jet drive so it has this intake port, I set up a stick and developed an analog problem as they say.

Richard Campbell: Ah, very nice. You know what they say a boat is. It's a hole in the water you throw money into.

Greg Hughes: Pretty much, yeah. The money that you put into the boat initially is the small part. It's the ongoing keeping it going and fixing it that really drills into your bank account.

Richard Campbell: Absolutely.

Greg Hughes: But it was a lot of fun. It's nice that spring is coming up and everything is turning green.

Richard Campbell: Yeah, it's all good and with springtime comes TechEd US so we're going to be there for the IT week in Orlando. That's the second week of June.

Greg Hughes: Yup. It sounds like we have a lot of fun things coming up. Of course, we're going to be doing Speaker Idol.

Richard Campbell: Again, yeah.

Greg Hughes: I have some panel discussions on the stage and we'll make some RunAs Shows from that and a whole variety of other things.

Richard Campbell: And TechEd is such an amazing opportunity to talk to some amazing folks from Microsoft. There's really no better place to go to get access to really remarkable talent around the products that we have to use every day.

Greg Hughes: Yeah, it's a huge number of people who, you know, if you're a blog reader and you read the things that are published on MSDN and the TechEd sites and whatnot, then you realize that there's a large number of people at Microsoft that have these huge brains and this capacity to be able to explain things and to meet them in person and to have a conversation is really a valuable thing.

Richard Campbell: And we're going to talk to as many as we can and capture them on recordings and publish them here.

Greg Hughes: Yup, it's going to be a lot of fun.

Richard Campbell: And if there's some shows you'd like to hear about, particular people or particular discussions, send us an email, info@runasradio.com.

Greg Hughes: Your suggestions are what drive this show.

Richard Campbell: All right, Greg. Let's introduce Erin. Erin Welker spent 25 years in Information Technology development, management, database administration and business intelligence. She began working with SQL Server in version 1.11…

Greg Hughes: Wow.

Richard Campbell: Analysis Services, SSIS, DTS, and Reporting Services since their inception. She was privileged to be a member of the Project REAL team, and has written whitepapers on data warehousing in SQL Server 2005. Her current focus is PerformancePoint Planning. Welcome Erin.

Erin Welker: Thank you.

Richard Campbell: SQL 1.11.

Erin Welker: Yes.

Greg Hughes: When was that?

Erin Welker: That was early '90s.

Richard Campbell: Yeah, no kidding, and that was the Sybase product entirely I imagine.

Erin Welker: Exactly.

Greg Hughes: Cool.

Erin Welker: It's come a long, long way.

Richard Campbell: Well, just the breakaway from Unix and that whole sort of space is amazing, really.

Erin Welker: Yeah, we were under OS 2.

Richard Campbell: I remember I jumped in at about 4.2 and that's when NT first came along and nobody thought NT was going to go anywhere. I really liked it and NT 3.1 didn't have a GUI. It was all command line and I loved that.

Erin Welker: Well, it actually saved our lives. We went live with NT maybe a couple of months after it came out. Before that, we were planning on going with OS 2, which our tests just said will not work. So, we took a gamble on NT and it paid off big time.

Richard Campbell: And even then, that early version of NT had NTFS and that was really the key to the whole thing.

Erin Welker: Yeah, but one thing, it's funny, I was just thinking about this the other day how we were playing with SQL Server 1.11 and 12 meg of RAM was as far as it would go.

Richard Campbell: Wow, 12 megabytes.

Erin Welker: 12 meg.

Greg Hughes: That's a whole different world.

Erin Welker: We thought that was a lot.

Richard Campbell: It was tons. Project REAL, what was that?

Erin Welker: Have you not heard of Project REAL?

Richard Campbell: I have not heard of Project REAL.

Erin Welker: Really? Well, that was a Microsoft-initiated project and they brought in some partners and I got to be one of them and it was all about taking a real live customer and taking them from SQL Server 2000 and migrating them to 2005. This customer was Barnes & Noble…

Richard Campbell: Cool.

Erin Welker: Who was running a 1.2 terabyte database, data warehouse. So, it was actually not converting Barnes & Noble themselves, but to take their data and mask it and then go through a best practices scenario and then produce source code, whitepapers, basically "here's how you do it" or "how we recommend you do this."

Richard Campbell: So, the right way to migrate from 2000 to 2005.

Erin Welker: Yeah, and more so since Analysis Services and SSIS were such big jumps from prior versions of product. How do you utilize the utilities properly?

Richard Campbell: So, not just moving the database itself, but going into the new data warehousing or OLAP tools.

Erin Welker: Right, right. So, we wrote, I didn't personally, but members of the team wrote a full-pledged ETL on SSIS and then published best practices on that. My focus was on partitioning, which of course was new in 2005 and how to best implement that.

Richard Campbell: Right, so actually taking tables and spreading them across multiple machines?

Erin Welker: No. Partitioning actually can only be done within a database, so it was spreading database across potentially multiple disks.

Richard Campbell: Right.

Erin Welker: And the key with partitioning that when I present, certainly, one of the first things I say is this is all about management. So, if you have a multi-terabyte database, how do you create indexes, how do you backup, how do you do all of these things, especially in a data warehouse where a huge portion of the database is potentially inactive. No data is changing, so there is no need for that data to participate some of those maintenance activities. Does that make sense?

Richard Campbell: Yeah, sure. Last month's data is not going to change again, so why keep it in the active partition?

Erin Welker: Exactly. So, as you're maintaining your indexes, as you're doing your backup, all those things that as a DBA we know we're supposed to do to keep a healthy database, partitioning allows you to do virtually all of those activities at a partition level. So, instead of maintaining 2 terabytes of data, you're actually maintaining maybe 100 or so gig. We all know how to

do that. Well, I don't know if you know how to do that, but…

Richard Campbell: Oh sure, but, you know, these are all things we're concerned about. I mean partitioning is an interesting -- the trick is how do you automate that so it's relatively painless? How do I know when it's next month's data and just move on? Do I actually have to copy stuff around or that just happening innately?

Erin Welker: Well, actually, that's one thing we did do as part of Project REAL is built some routines into the ETL process so that it says, "Okay. Hey, it's brand new." Actually, Barnes & Noble partition on a weekly basis. So, during the ETL process, they would say, "Hey, we've got a new week. Let's see if we've already created this partition. If we haven't, let's create it." It automates basically dealing with that data and making sure that it goes into the right partition.

Richard Campbell: Does the application know it's switching from partition to partition? Is it just speaking to stored procedures? How does that connection work?

Erin Welker: Well, the beauty of partition tables is in essence, in 2005, every table is partitioned, but most tables just have a single partition. So, every table you refer to as this comprehensive table name, that if it's partitioned into 100 partitions, for example, you only query the table and then underneath, SQL Server collects the data from the various partitions.

Richard Campbell: And then you're defining a partitioning rule that would actually allow it to know where to write.

Erin Welker: Exactly, a partition function.

Richard Campbell: Ah, I see.

Erin Welker: So, in a data warehouse, for example, in probably 99.9% of the cases, we're going to partition on a date.

Richard Campbell: Right because that's the one thing we can count on, time goes by.

Erin Welker: Right.

Richard Campbell: So, for most journaling type of data like sales data and so forth, you're just going over time and you're just setting parameters whether it be a week or a month?

Erin Welker: Correct. We actually did some performance testing as well, I think it was during Project REAL, I've done it at some point, regarding as you're bringing in data in a data warehouse, probably 80% or more of it is going to the most recent partition. So, what we found is when you're updating one partition heavily, it's probably best to switch out that partition to an external table, drop the indexes, strip constraints and everything that will slow down a load so our updates re-add all that stuff and then switch it back into the partition table.

Richard Campbell: Don't you run a risk when you do that strip down and reload, which obviously is faster that then you don't have to manage constraints that can't be reapplying violations in the data?

Erin Welker: Well, hopefully your ETL process is, number one, taking care of that and also in very large data warehouses, oftentimes we don't keep constraints and the reason why that is, is for performance purposes. Given the fact that our updating of the data warehouse is so heavily managed through the ETL process, we can be pretty sure that the ETL programming has made sure those relationships or constraints have been…

Richard Campbell: Right. There's only one way to load a data warehouse, that's thru the ETL process, and it essentially is handling all the constraints for you, so why do you keep all the data.

Erin Welker: Right, as opposed to an OLTP application where those updates are coming from all over the place.

Richard Campbell: Right. It makes a lot of sense. Now, I've always thought of partitioning in the context of OLTP, really for performance reasons that because I can keep my active set relatively small, I'm able to keep things quite nimble in that respect, the indices aren't too big and so on, but how does this apply in data warehousing?

Erin Welker: Well, I have found that the performance gains in a data warehouse on partitioning are relatively small and I guess it depends on the scenario, but data warehouse by definition is oftentimes spanning a very large period of time. So, the gains we might get as far as being able to query subsets of the data are usually negated by the overhead of putting it all back together again so to speak. I'm not saying that you will gain nothing in terms of performance with partitioning. I'm saying it's primarily a manageability feature, not a performance feature…

Richard Campbell: Yeah, why change it?

Erin Welker: That may sometimes get performance benefits from.

Richard Campbell: Okay, that makes a lot sense to me.

Greg Hughes: What about the backup side of things? I know that historically, people have had a hard time backing up large amounts of data. It seems to me that if you can partition, split things up, you might be able to improve your backup performance.

Erin Welker: Right, and backup if I recall, I've been in PerformancePoint world for a few months now so I'm kind of having to switch gears again, if I recall, you cannot back up at a partition level, but what you can do is back up a file group. So, if you align your partitions with file group, then essentially you have implemented a partition level backup system. So, for instance, if my most current partition is on a file group that's separate from all the other partitions, then I can just backup that file group and I have essentially backed up the partition. Now, if you get into a recoverability scenario, then the main thing you got to worry about is recovery is kind of like to make sure everything is in sync and the way you do that is to make those file groups, the set partitions, the older partitions reside upon read-only and at that point, the restore says, "Oh, you've been read-only as of this point," so I know I don't have to worry about log time checking.

Richard Campbell: That's an interesting element I think of partitioning here is being able to mark off -- it's one table, yet pieces are marked as read-only.

Erin Welker: Right.

Richard Campbell: I just find that fascinating. It's a partially read-only table.

Erin Welker: And I kind of like that as a DBA because then I can be sure that nobody is going in there ad hoc and doing something outside of the routine system.

Richard Campbell: You are limiting the range of damage to a very large table effectively. It's another kind of constraint.

Erin Welker: Right. Yeah, that's a good way of looking at it.

Richard Campbell: Yeah, it's a fantastic idea.

Greg Hughes: I'm the step back guy here, so why don't we step back one second. It's my understanding -- I've managed database administrators, I've never been one myself, so I'm kind of the IT management guy that doesn't do anything, he just manages the people that do the real work, right?

Erin Welker: I've met people like you. Actually, I've been you in previous lives.

Greg Hughes: I'm thinking from that perspective, it's my understanding that in SQL 2005 and I've managed software teams that have taken advantage of this partitioning before and I understand the benefits of it, but some of the nuts and bolts I don't really have a clue about, but there's a difference between a partition table, which we've talked quite a bit about, and a partition index, which you've touched on, but can you talk about the relationship between those and kind of how they impact each other. What do you think about like from an architecture standpoint, those sorts of things?

Erin Welker: Honestly, I don't know that I'm going to provide too much enlightenment there because what I generally do as they would say is align my index partitioning with my table partitioning and in fact if you think about it, if you create a clustered index, then the partitioning function you use for your clustered index is your table function. I think from the standpoint of manageability, I would have to stretch to find a scenario where you would then place non-clustered indexes using a different partitioning scheme because you would want to keep them all together, so as you archive or do various pieces of management, you'll want them all to kind of be aligned with each other.

Richard Campbell: I was trying to imagine a scenario where it never makes sense to not have the index partitioned exactly the same as the table.

Erin Welker: I'm sure the scenario exists.

Richard Campbell: Yeah, but it seems unlikely to me, really.

Erin Welker: I'm hard-pressed to think of it.

Greg Hughes: You mentioned sort of the physical disk partitioning. What have you seen -- I know that we explored and actually leveraged that for some performance gains just in terms of, this may not be data warehousing-specific, but since we've touched on it, but gains in high IO scenarios, so we have multiple tables, multiple databases and you're doing a lot of IO.

Erin Welker: Okay. So, you're talking about segmenting pieces of the data on disk subsystems?

Greg Hughes: Yeah, physically separate I guess spindles as you might call them.

Erin Welker: Okay. I think I've seen a fair amount of that in 2005. I think this is a complaint to point out a new behavior coming up in 2008, and Erik

Hanson would be proud of me for pointing this out, there was an issue in 2005 or is an issue in 2005 if you have a query that ends up examining a small number of partitions, more than one but not many more than one. So, let's say you're spanning two partitions.

Richard Campbell: Right.

Erin Welker: The way 2000 works is it will spawn a thread for each partition it queries.

Richard Campbell: Now, is that 2000 or 2005?

Erin Welker: Oh, I'm sorry, 2005.

Richard Campbell: Right.

Erin Welker: Thank you for clarifying that.

Richard Campbell: Okay.

Erin Welker: So, if you have a box that has 32 processors on it or more, you're greatly under utilizing the processing capabilities of that server.

Richard Campbell: Well, because normally doesn't SQL Server 2005 allow multiple threads against a partition?

Erin Welker: No. Now, there is one caveat to that. If you are querying a single partition, then there's intra-partition querying and it will spawn as many threads as it deems.

Richard Campbell: Okay. So, it's not that the table has to be only one partition, but that the query processor has figured out the data I'm looking for is in one partition and then it would use, could use multiple threads.

Erin Welker: Exactly. So, this scenario actually came out of Barnes & Noble.

Richard Campbell: Oh really?

Erin Welker: Two partitions on a weekly basis and so imagine if you have a query against an, I'm trying to remember, well, you can kind of do the math, 52 times, say, five years, so over 250 partitions.

Richard Campbell: Right.

Erin Welker: Then they do a query that says, "Okay, compare the sales from this week versus those last week." So, I'm querying two partitions and so then I'm going to spawn two threads.

Richard Campbell: And it will only spawn two threads.

Erin Welker: Only spawn two threads and it gets worse than that. Let's say that I'm comparing sales today versus yesterday.

Richard Campbell: And it's across the partition barrier.

Erin Welker: So, you would have this query we'd say, pretty quickly most times because if I'm within the same partitioning, I'm spawning many threads.

Richard Campbell: Right.

Erin Welker: And it comes back like that, but then once a week or one day a week, I'm spanning two partitions and suddenly the query gets slow for no reason.

Richard Campbell: Right. So, preparing Monday and Tuesday, lightning fast; comparing Sunday and Monday where it jumps the partition, it would suddenly decrease the performance dramatically.

Erin Welker: Correct.

Richard Campbell: And from the outside, you go, "What's going on?" It would just be totally baffling.

Erin Welker: Exactly. So, in 2008, they've changed the behavior to essentially throw all the threads at each partition and start with partition 1 from the query, throw all the threads at that, return results, then go to the next one and go to the next one. So, this is all being said to address the earlier point of how do you deal with the disk. Well, with this new behavior, you want each and every partition to take full advantage of many disks, many spindles so that any given partition will take full advantage of the underlying IO subsystem.

Richard Campbell: Interesting. So, now, to talk about manifestation here, normally to split up partitions onto separate spindle sets, I would put them in separate file groups and then position the file groups on different spindle sets, but because of this threading behavior, only any one given spindle set is going to be used in a given query anyway because only one partition is going to be worked on at a timeframe given query.

Erin Welker: So, if your strategy pre-2008 is to say I'm going to thread partitions across their own kind of individual pieces of the IO subsystem in order to take advantage of that parallelism, that's not going to work in 2008.

Richard Campbell: No. That was the whole vision of that was here, I'm going to put these three

partitions and I all want to query on three separate sets of spindles thinking parallel execution will run all three spindles at the same time, but the threading constraint totally undermines that. I'm better off to spread those partitions across all of those spindles because more spindles is always better.

Erin Welker: Right. In essence, I think they've implemented the best strategy for most cases, which I'm sure that was their goal.

Richard Campbell: Well, I've got to think this has got something to do with query consistency more than anything that ended up limiting the threads.

Erin Welker: That's kind of huge, especially if you have some C-level executive that ultimately are submitting queries through whatever type of interface and one day runs one way and the next day it runs different ways, then that's not good.

Richard Campbell: One thing I learned as a database guy was don't make the executives uncertain and distrust the data.

Erin Welker: Oh, and multiply that to about 10 to 100 times in a data warehouse.

Richard Campbell: No kidding, yeah. Oh, I get chills. So, what other things have happened in 2008? I mean obviously this threading issue I think is a very important thing to understand, the difference between 2005 and 2008, but partitioning really only came to exist in 2005, so this is really version 2 we're looking at in 2008. Are there other major improvements or changes?

Erin Welker: The main one that comes to my mind and I'll take this moment to plug the blog as I blogged about it probably about two months ago. I actually found this out by reading Paul Randal's blog who if you read his blog, you know he blogs much often and about just the most juicy stuff.

Richard Campbell: You're talking about the guy who wrote DBCC or at least led the team to write DBCC. He knows SQL Server very deeply.

Erin Welker: Yes, yes he does. He pointed out that there is a locking behavior change in 2008 and the default behavior does not change. However, you have the ability to overwrite the default locking behavior and I'll probably not remember the minute details of this, but essentially you have, I think it was if you query various partitions, you were subject to the same lock escalation as if you had a single partition table.

Richard Campbell: Right. In 2005, there was no concept of a partition lock. If you got an escalation of a few pages where normally if it was only one table, it would lock the whole table. That still happens even if you partition it.

Erin Welker: Right, and you know it doesn't take much before it escalates to that lock.

Richard Campbell: Right.

Erin Welker: So, if you're querying just a few partitions, suddenly you've locked everybody else out.

Richard Campbell: Thinking that the whole concept of a partition table is you're going to let it get massive.

Erin Welker: Right.

Richard Campbell: Huge numbers of partitions is a vast table. To have that whole thing locked is kind of frightening.

Erin Welker: So, in 2008, there is an option to override that behavior and have it consider the lock escalation at a partition level, not at the table level.

Richard Campbell: Interesting. So, now we do have a partition level lock, but it doesn't happen by default?

Erin Welker: No, by default, which in a way I kind of question, but then again I haven't really played around with this and I'm sure they did when they were making these design decisions, but the default is still that -- and this is not necessarily how it takes locks, but when it's at the lock escalation level.

Richard Campbell: Right.

Erin Welker: So, the default behavior, at least when Paul blogged about it, was still that it would consider the escalation to the table level.

Richard Campbell: That's got to be consistency going back to previous versions. You don't want to change locking behavior when somebody upgrades.

Erin Welker: You're probably exactly right.

Richard Campbell: That would be my gut reaction is that's just something you don't do.

Erin Welker: Well, we might also look at it as an advanced option feature too. You probably don't want people playing around with that unless they really know what they're doing and usually if they know to go do that, then they've done their research.

Richard Campbell: So, then you can also switch it over to a mode where it will just escalate to a partition

level if it can and if it's not partition table, it still goes to a table and so on. I mean it's a very interesting problem how to introduce this capability and not control things.

Erin Welker: Yeah and you kind of got to consider as you mentioned 2005 is the first release with partitioning. Now, there were partition views in 2000 and I think partition views get a bad rap. Actually, they can sometimes outperform partition table. So, there is actually still kind of two partitioning options in SQL Server from a management standpoint, probably partition tables are the better way to go.

Richard Campbell: But it's an interesting discussion because the idea of a partition view then is to not partition the table just to do it at a view, really a logical construct rather than an actual table construct.

Erin Welker: Right. The problem with partition views that some customers experienced was you consider the partition view is sitting on top of a bunch of tables that each one represents a kind of a partition, but the query optimizer views them each as their own individual table with potentially a different indexing strategy.

Richard Campbell: Right.

Erin Welker: Therefore, it has to optimize or look for the best index method, access method, for each and every one of those so your compile time can go through the roof.

Richard Campbell: Yeah, you're getting a much more complex query plan to deal with all of that.

Erin Welker: Right and in actuality, you may have the same indexes on each and every one, but the optimizer doesn't know that.

Richard Campbell: No.

Erin Welker: It can't guarantee that.

Richard Campbell: No, it can't guarantee because it can be different, it has to at least check so that's just going to take time.

Erin Welker: Right.

Richard Campbell: But it strikes me that the partition view is really an intermediary step. If I've got an existing database where I haven't partitioned data, partitioning existing data isn't easy. That's a lot of work, isn't it?

Erin Welker: It certainly can be. Actually, it's not -- if you did use partition views in 2000, it's actually a pretty easy move to partition tables.

Richard Campbell: Okay, and that seems to be a sort of best practice approach that you're going to get more performance benefit out of actually partitioning a table, especially in 2008 with the new locking rules and so forth and a better threading model.

Erin Welker: Right, and we've got -- you were talking about new features in 2008. There is actually another one I'm very excited about that's index views. In 2005, if you had index views on top of your partition table, basically you had to drop and recreate them every time you switch partitions in or out of the partition table.

Richard Campbell: Right.

Erin Welker: You might say, "Well, why is that such a big deal?" Well, if you're doing a lot of querying on a relational data warehouse and when I say that you're just using it to source an OLAP cube, so if you're querying that database, index views can be great as summarizations of data to really speed up those queries so that you're not constantly aggregating.

Richard Campbell: It's like an intermediary aggregate essentially between…

Erin Welker: Exactly.

Richard Campbell: So, you're talking about running an OLAP cube in more of a real time view rather than pre-aggregating the data.

Erin Welker: Right. So, essentially, what you're using index views to do is exactly the same thing you do in analysis services. To speed it up, you're pre-aggregating. However, in 2005, index views on a partition table are virtually unusable because the table is huge almost by definition and if you happen to recreate that index view, we could potentially be talking hours.

Richard Campbell: You might as well re-compute the cube, which is what you're trying to avoid in the first place.

Erin Welker: Right.

Richard Campbell: I see this as really a significant improvement in folks who want to do real time analysis where we're steadily loading data and rather than going through that sort of load process and re-aggregation, we have this partition. It's the partition align index view is what you're talking about where the index view is bound to the partition, that's storing

the aggregate, so as data is being loaded into the partition, it's re-computing those aggregates and then your real time cube is able to just take advantage of those aggregates.

Erin Welker: Yeah. In actuality, I'm speaking more for -- there are a lot of customers I've been to in the last couple of years who don't want to use Analysis Services.

Richard Campbell: Really?

Erin Welker: I don't agree with that, but, you know, they have actually some good business reasons for that. Maybe it's the new skill set that they aren't prepared to support or maybe one customer I had had a bad experience where somebody who designed the cube I would say not in the most desirable way because it was quickly deprecated as business rules changed.

Richard Campbell: Right.

Erin Welker: So, they had the bad experience and they said, "We don't want to go that way again because we tried it and it didn't work." Being a consultant, sometimes we have to recognize the battle is not going to be won.

Richard Campbell: Yeah, you're not going to win that one. How am I going to help these guys? This is an alternative approach to that.

Erin Welker: Right.

Richard Campbell: It's a fascinating angle on some of the new capabilities we're getting with partitioning in 2008. It makes me wonder what we're going to get next. You know, they say Microsoft's technology, the third version is always the great one. This is only the second one. I wonder what they're going to do next.

Erin Welker: Yeah. Well, one of my number one wish list along the lines of index views is that they improve the optimization to where more queries can take advantage of them. There are some matching rules that are in place that are easy to bypass.

Richard Campbell: So, there's just not enough sophistication in the query processor to take advantage of all of the limiting that partitioning provides.

Erin Welker: Right and I know that the powers that be are aware of it. As you know, there's only so much you can push.

Richard Campbell: Sure, but for the past two years, I found especially in 2005, I felt like I couldn't do much to a query plan anymore. Most of the time when I looked at a query plan, it was right.

Erin Welker: You know what? I second that 100%. There are so many times where it's like, "Oh, it's not doing that. I know better."

Richard Campbell: Yeah.

Erin Welker: You know, I force it to do my way and sure enough…

Richard Campbell: Yeah, I know, it was right, but with some of these partitioning features and arguably it's because they're new, this is where you would find that the query plan wasn't running efficiently because there were some elements the query planner doesn't understand around the partition potentials.

Erin Welker: Such as?

Richard Campbell: Well, I'm just thinking exactly the way you were describing that the query planner doesn't always take advantage of what partitions do for boundaries of data.

Erin Welker: Well, actually, and I may have said something other than what I meant, what I meant was index view.

Richard Campbell: Right.

Erin Welker: So, the beauty of index views is that you can have a query that goes directly against one of the tables that the index view covers and the optimizer says, "Oh, well, I can get all of that out of the index view," and it does that for you automatically.

Richard Campbell: That's cool.

Erin Welker: But there are matching rules that have to be observed in order for that to happen, so if all the moons are aligned and, you know…

Richard Campbell: But sometimes it doesn't happen and that's going to show up in a query plan that it doesn't come up with the best matching rule set.

Erin Welker: Right. You can take advantage of that. In fact, you can change the query to just send it to the index view and the thing with index views, we're not talking about 10% performance improvement or even it got twice as fast. We can sometimes get 100 times faster with an index view.

Greg Hughes: Wow.

Richard Campbell: Dramatic, dramatic improvements.

Erin Welker: It can be a huge performance benefit.

Richard Campbell: It's nice to know as DBAs, we still have a job. You know what I mean?

Greg Hughes: That's for sure.

Erin Welker: Yeah. I think we'll always have a job. I've been around as I've said, for more than 25 years now and I think since day 1, they said we're going to be out of a job in five years.

Richard Campbell: Somehow it never seems to happen.

Erin Welker: Yeah. In fact, we make it more and more complex, so they're more dependent upon us.

Richard Campbell: Yup, no question about that.

Greg Hughes: I know I've seen indexes make and break commercial software that depends on SQL Server and if it wasn't for the DBAs, there wouldn't be a product.

Erin Welker: Right, right. As a consultant, I bank on the fact that Microsoft has created products, several products in fact, that on the surface are very easy to implement, but knowing what's going on underneath the covers can greatly enhance that implementation.

Greg Hughes: Absolutely.

Richard Campbell: Erin, I think we're about out of time. Any final words?

Erin Welker: Not that I can think of. I'll mention my blog again, not so much to promote myself, but to promote various other SQL Server experts that are on that site blogging about issues that are near and dear to our hearts.

Richard Campbell: Right, and that's www.sqlblog.com/blogs/erin_welker. Erin, thanks so much for coming on the show.

Greg Hughes: That was great, Erin. Thank you.

Erin Welker: Oh, thank you. I enjoyed it.

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