

Subject: Today's posts from Brent Ozar Unlimited for 09/20/2016

From: Brent Ozar Unlimited® (help@brentozar.com)

To: aronet59@yahoo.com;

Date: Tuesday, September 20, 2016 9:52 PM

Here's our latest blog post.

Is this email not displaying correctly?
[View it in your browser.](#)



Tara Kizer says:

[Asynchronous Database Mirroring vs. Asynchronous Availability Groups](#)

When Database Mirroring came out in SQL Server 2005 Service Pack 1, we quickly dropped Log Shipping as our Disaster Recovery solution. Log Shipping is a good feature, but I can failover with Asynchronous Database Mirroring faster than I can with Log Shipping.

When Always On Availability Groups (AG) came out in SQL Server 2012, we were excited to get rid of Transactional Replication, Failover Clustering and Database Mirroring. It solved our reporting needs (your mileage may vary), our High Availability needs and our Disaster Recovery needs.

But what if you only have Disaster Recovery needs and want to use Availability Groups since Database Mirroring is deprecated? Make sure you have quorum setup properly!

Let's look at the big difference between Asynchronous Database Mirroring and Asynchronous Availability Groups.

ASYNCHRONOUS DATABASE MIRRORING

For asynchronous Database Mirroring, all we need is two servers: the principal at the primary site and the secondary at the DR site. Setup async mirroring between the two, and you're done. If the secondary server goes down, production still continues. The transaction log isn't clearing when the log backups occur because the principal still needs to send those log records to the secondary. As long as you have enough disk space where the transaction log resides to support this until the secondary server comes back online, production still continues. Of course if you run out of disk space, users will start receiving errors. But that can take some time and often times is enough time to get the secondary back online.

ASYNCHRONOUS AVAILABILITY GROUPS

We still need two servers for our asynchronous AG: the primary replica at the primary site and the secondary replica at the DR site. Once the two servers are in the same Windows Server Failover Cluster (WSFC), setup the Availability Group with the two replicas, specifying asynchronous-commit for the availability mode. Now take the secondary replica down and see what happens. Did the primary replica also go down? We've heard from many people that their primary replica goes down when they do maintenance on their secondary replica, and they aren't sure why. It's because quorum and/or voting is not setup properly. Availability Groups use WSFC which requires quorum. You'll need another resource at the primary site to achieve quorum. I usually use a file share witness. I did use a shared disk on one AG though. Once those are setup, modify the WSFC's quorum to whichever you picked. For instance, "Node and File Share Majority" or "Node and Disk Majority". Now when the secondary replica goes down, production continues on the primary replica because the cluster still has a quorum (2 are up, 1 is down).

Both servers and the third resource all have quorum votes by default. In this simple example where we have just two servers, it would be okay for the secondary replica to have a vote. It may not be okay if you had a more complicated setup, like the one I setup for the system that needed HA, reporting and DR. I learned about quorum votes fast when a network interruption between the two sites took production down! That system had 5 replicas (1 primary, 4 secondary): 3 at the primary site, 2 at the DR site. So far so good. But we also had a 3rd server at the DR site in the cluster. This was using SQL Server 2012, so we were limited to 4 secondary replicas. That 3rd server at the DR site was a cold standby (we decided against Log Shipping to it), and it too had its quorum vote enabled. The cluster lost quorum when the network between the two sites went down as there were 6 votes and 3 went "down".

If you are using Windows 2012 R2, you've got Dynamic Quorum! It is enabled by default. The votes can be dynamically adjusted by the cluster as needed. Dynamic Quorum would have come in handy for that system I mentioned, but this was before Windows 2012 R2 came out.

Brent says: something to think about – [the book on database mirroring](#) is 400 pages. The book on Availability Groups...still isn't out yet. It's not delayed because it's easy, it's delayed because it's hard. It's doable, it's just...harder than mirroring.

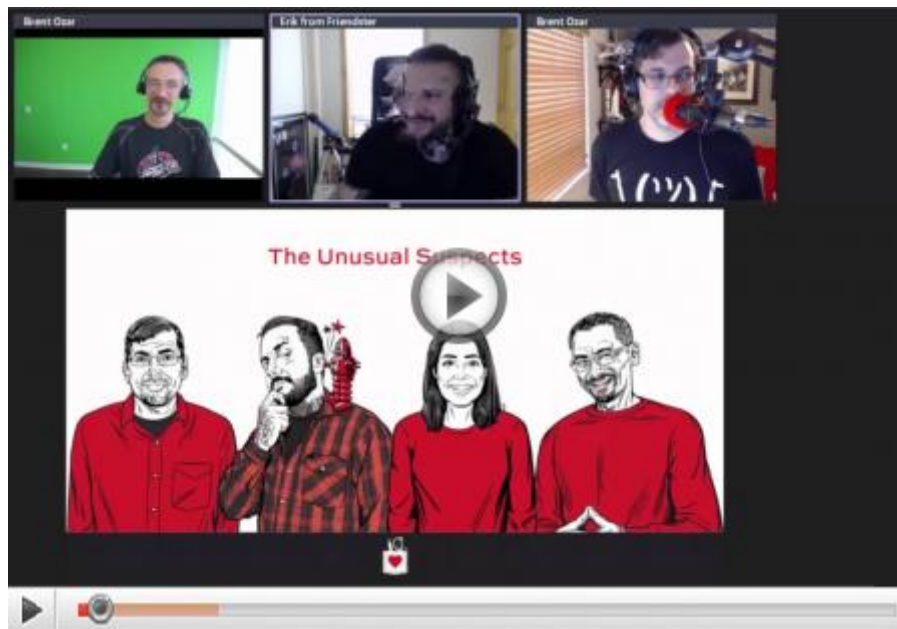
[Read the comments.](#)

Brent Ozar Unlimited Team says:

[\[Video\] Office Hours 2016/09/14 \(With Transcriptions\)](#)

This week, Brent, Richie, and Erik discuss execution plans for table value functions, how to prepare for the Senior DBA course, source control software, generating fragmentation, backups, and the new TV shows the team is looking forward to this season.

[Here's the video on YouTube:](#)



You can [register to attend next week's Office Hours](#), or [subscribe to our podcast](#) to listen on the go.

If you prefer to listen to the audio:

Enjoy the Podcast?

Don't miss an episode, subscribe via [iTunes](#), [Stitcher](#) or [RSS](#).

Leave us a review in [iTunes](#)

Office Hours Webcast – 2016-09-14

Brent Ozar: It's a quarter after, we might as well go ahead and get started. Let's see what we've got here. Folks, feel free to type in your questions over in the questions pane in GoToWebinar. We will work through them one by one. Don't hold your questions until the end because people inevitably end up putting in their questions right now, yours won't get answered.

Erik Darling: Speak up, fellow humans.

Brent Ozar: Speak up, fellow humans.

How can I see the execution plan of a table-valued function?

Brent Ozar: Mandy says, “How can I look at the execution plan of a table-valued function? I know there’s drawbacks, I just can’t replace this one at the moment and I’m trying to figure out a way to improve its performance.” How do you find the execution plans for table-valued functions?

Erik Darling: You can get the estimated plan of whatever query calls it. It will have a little box underneath that shows what the function does.

Brent Ozar: When we say estimated plan, is that a different shape from the actual plan that will execute? What’s different between the term estimated plan and actual plan?

Erik Darling: Great question. It doesn’t give you any of the actual values that get returned. It just gives you what SQL estimates. SQL goes to the optimization stuff, up to the point where it would actually execute the query—it doesn’t actually execute it. It just gives you the plan with the basics. Like, “We think this many rows are going to come back. So we’re going to do this, this, and this to get you your query.” I know that works with scalar valued functions. If you’re talking about multi statement table-valued functions, I think that also works. But with a table-valued function, if it’s an actual table-valued function, I don’t think you get the separate execution show beneath.

Brent Ozar: Yeah, I believe you only get it on estimated plans. The other way you can do it that’s really useful is sp_BlitzCache. Sp_BlitzCache, when you run that and sort by executions, there’s a sort order parameter. If you sort by executions, it will tell you the queries that ran the most often. Most likely it’s going to be your function. One of the top ten will be your function.

Do you have any good tips for preparing for the Senior DBA Class?

Brent Ozar: Next question, “Do you guys have any good tips on preparing for the [Senior DBA course](#) next week?” Yes, if you’re in America, get caffeine. It starts at 3:00 a.m. Central Time. I am certainly going to be getting myself some caffeine. The best thing I can tell you—this one is targeted at the European audience. It starts at 9:00 a.m. in London. The best advice I can give you is go run sp_Blitz to go look at all of the gotchas on your SQL Servers. Then the really big one is go get our download pack, [our First Responder Kit](#). In there, there’s a worksheet for your RPO and RTO, your recovery point objective and recovery time objective. Spend some time getting to know that form and thinking about what the values are for your most mission critical SQL Servers. A lot of what senior DBAs do is driven around RPO and RTO.

How can I tell what queries are blocking my nightly stored procedure?

Brent Ozar: Next question, “What the easiest way to check for queries that are blocking my nightly stored procedure? Please don’t tell me I have to be at my desk at midnight.”

Erik Darling: Turn off your index maintenance because I’m willing to bet that’s what it is. Aside from that, I would actually turn on the [Blocked Process Report](#) for that.

Brent Ozar: Have you actually used the Blocked Process Report?

Erik Darling: To some degree, with some success, yeah. [Michael Swart has that Blocked Process Report reader](#) which is really handy too. Which is really the best way to do that because I would never do that on my own. That, or you could log sp_whoisactive to a table. Then you can get the blocking stuff from that as well.

Brent Ozar: Oh, I love that even more. If you go to [brentozar.com/responder](#) there is an [article by Tara Kizer on how you log sp_whoisactive to a table](#). Sp_whoisactive is a great stored procedure by Adam Machanic that tells you what’s going on right now. It includes execution plans too. You can run that every minute. I would just set up a scheduled job to run it every minute during the time window that you care about, during your nightly job. That’s fantastic. Blocked Process Report Viewer is cool too.

What accounts should I use for replication security between domains?

Brent Ozar: Paul asks, “I am building a distribution server to handle replication between our local site and a data center. How and what accounts would you set up locally and on/or the domain. There are going to be a couple separate domains involved. Any other advice on setting up a distribution server? Also, will you start reading these questions ahead of time before you start saying them out load?” [Laughter] I don’t know. I have no good advice on that whatsoever. Any of you guys? No?

Erik Darling: Nada.

Richie Rump: No.

Brent Ozar: Yeah, we’ve got nothing on that. If you go to [Amazon and look at the books on replication, there’s one by \(UPDATE: Sebastian Meine\)](#). That would be the one that I would recommend there. Or just post this one on Stack Exchange, it’s actually a really good question for Stack Exchange. Break it up into a couple pieces. Have one about the security and another one about best practices around that.

Erik Darling: Yeah, that’s a really, really specific situation. I don’t think you’re going to find that covered like in a textbook example. What you’re doing is a bit out of bounds for what most people do with replication. I’ve seen some crazy merge replication strategies, and that’s bananas.

Brent Ozar: There’s nothing to say it’s wrong with it. Totally okay. It’s just unusual.

Erik Darling: Yeah.

What source control do you recommend?

Brent Ozar: “Which source control software do you recommend?” Richie, that’s a good question.

Richie Rump: No, Brent, you. You love source control. I know you’ve got a couple of your favorites.

Brent Ozar: Absolutely. Visual SourceSafe is by far and away the best product out there. (laughter)

Richie Rump: No. No. I still have corruption nightmares from dealing with Visual SourceSafe. I go back, man. I go all the way back to CVS and Visual SourceSafe and all that other junk that we had back in the days. Git is by far the best these days. I know a lot of companies that still use TFS. Just to show you what the folks think about TFS, there’s now a Git to TFS piece of software where you could use Git, then it moves it to TFS. GitHub is free. It’s really the best thing that’s out there. It’s great. We use it here. I use it personally off hours. Go GitHub. Rah GitHub.

Brent Ozar: I hate source control. I hate it with a burning passion because it’s hard, it’s complex. You are dealing with merges and all this. And I’ve even grown to love Git. Between the clients for it and the website, GitHub is fantastic. Fantastic is the wrong word. It’s doable. It hardly sucks at all.

Erik Darling: I like Git. I just wish that there was some like big, blaring sign that says, “You are working with this branch.” Or like, “You branched from master. Not from dev” or “you did this.” Yell at me, man. Make it really obvious. There’s nothing worse than doing something and being like a half hour into it and then realizing, “Oh, I branched from master and now I break everything... ugh.”

Richie Rump: Dammit, Erik, you did it again. Stop doing this to me.

Brent Ozar: There needs to be a dumbed down version of GitHub for the rest of us. Don’t let me touch master. Just let me work over in dev. Force me to start a new branch. Never allow be to commit directly to dev. Yeah, it’s pretty insane.

Richie Rump: So for those of you who just want to get—I know you guys don’t like this, but I use it all the time—it’s called SourceTree. It’s a free editor by Atlassian to help you with some Git-y stuff. So if you’re more visual oriented as opposed to command line oriented, you could use SourceTree and it helps get over that. You still have to understand how Git works and how the Git process works, but once you get over that, you don’t have to be command line typing it like you would in Git. I use both. Command line and I switch back to SourceTree when I want to do a quick check-in. Check it out, SourceTree.

Why does the number of rows affected seem off for updates?

Brent Ozar: The next one we'll go tackle is, "I've got an update query that runs. Whenever it runs I get the little number of rows returned thing back in messages. I think I'm only updating 500 rows but it keeps saying 1,000 rows affected. Is there anything that will affect what it says the number of rows affected is?"

Erik Darling: Hmm. Is there a trigger on the table?

Brent Ozar: I like that. That make sense. Someone else answers too, in the Q and A, someone else put in triggers, which means someone else has been burned by this as well.

Erik Darling: I was thinking back to when I was writing that stupid trigger for the DBA Days stuff. I was like, "Oh, yeah, that's a lot of..." What I tend to do is set no count on and just blindly put a print statement at the end of the trigger to tell me how many rows are affected by the trigger. So I have a separation. It's like, okay, the trigger affected this many rows on its own. So that gives you the count of how many things.

Brent Ozar: Genius. Plus, what I love about the print is that then people know that there was an actual trigger that was involved in an execution.

Erik Darling: Yeah, you just set something to row count and print rows affected by trigger, blah, blah, blah. You can see exactly how many rows the trigger hit. In my case, it was like 250 million.

Brent Ozar: You could also start a clock at the beginning of the trigger. Stop it again at the end and say "This number of milliseconds was burned up waiting for the trigger." I'm joking, people. Please don't do this.

Erik Darling: What you could do is have your trigger kick off an extended events session and measure disk... Get back all the information about the trigger... Don't do that.

What was the most surprising thing you learned while building [Paste the Plan](#)?

Brent Ozar: So folks, we're out of questions. This is probably a first ever for Office Hours. Feel free to type in any questions that you've got. As long as they're not asking questions, that means we get to ask questions. I'm going to start asking Richie questions about Paste the Plan. This week we went live with our new Paste the Plan in order to share execution plans online. What was the most surprising thing you learned out of building [Paste the Plan](#)?

Richie Rump: Gosh, there was a lot of things I had to learn. The biggest problem was using the new technologies that I had never used. I had never built really anything substantial, that's been in the cloud. The hard part was I couldn't use any of the tools that I have used for 15 years on that. So, C#, MSSQL. We could have gone MSSQL but we didn't. MVC. All the, ASP.NET, all the stuff that I've really honed and now it's just kind of second nature, I had to learn something new. Even the language which was Node.js which is JavaScript on the server, that is completely different than you would think that's JavaScript on the client. They're two different things. Figuring out those nuances,

figuring out how do I actually do processing in the cloud using AWS Lambda functions. And even to our own database. We did the analysis, saying, “How much would it cost if we used DynamoDB versus SQL Server? What’s the difference there?” We did all that work to try to figure out what would be the best bang for our buck for Paste the Plan. And if this is just a small little thing, we wouldn’t be out thousands of dollars because we’re spinning up all these EC2 instances, the databases, and all that stuff. So it was definitely eye opening to get into the cloud and figuring out, “What’s this server listing? How does it work? How do I connect my client into the server and write to the database?” All that stuff. It took me a little bit longer than I expected it to but it was fun getting into and trying to figure out the problems and how to work around some of those things.

Erik Darling: When you’re learning a new language like you did for that, how much do you go out and look for example code versus how much do you try and just bang it out yourself for the first time and see what you can do?

Richie Rump: The difference I think between Node.js and the more traditional languages, like Java and C#, is that the community wants to install packages. There’s all these little packages that does all these little bits of code. It’s kind of understood that you would just go in and download a package that does it for you. So there’s all these little bits of code out there that actually help you do certain things. One of the big interesting things was we need to do a lot of XML parsing. We’re parsing XML. We’ve got to read it. We’ve got to parse it. We’ve got to validate it. We’ve got to do all this stuff to it. There’s nothing in Node that lets you do XML parsing. It’s not there. So the community has taken a C project and ported it over into Node. We use that in order so that we can do all our XML processing. Whereas opposed to C# and JavaScript, it’s just there. You don’t have to think about it. So that’s just kind of the paradigm for the Node community and how they tend to work. It’s backfired on them because one guy, a few months ago, he pulled one of his packages from the library and about 15,000 projects broke because they were using that one piece of code. So it is a fragile ecosystem but it seems to work pretty well.

Erik Darling: Wasn’t that code like one line too?

Richie Rump: Yeah, it was a one-liner. It was I think like pad right or pad left or something. Yeah, it was silly. But again, that’s the culture of Node and that’s kind of how they work. That’s how that community has decided to work, not to say it’s good or bad, that’s just the way it is.

Are you okay with doing edition – not version – upgrades in place?

Brent Ozar: We got a few questions in. Mandy says, “I’m thinking about going from Standard edition to Enterprise edition of SQL Server 2014. I know I’m not supposed to do a version upgrade in-place. I know you guys don’t like doing version upgrades because you don’t have a back out plan. What about an edition upgrade? Would you be comfortable doing an in-place edition upgrade?”

Erik Darling: I'm pretty amenable to that. I think that's probably the simplest way to do it. You don't really need much of a back out on that, unless like you know, something explodes.

Brent Ozar: You're taking an outage to do it, but it's going to be so short relative to building a whole new box.

Erik Darling: Because they don't really have to add anything. It's just kind of a skew where they're just like, "You can do all the cool stuff now. Welcome to the big kids table."

Brent Ozar: What you have to do is restart set up and then go up, up, down, down, left, right, left, right, B, A, B, A, start. Then immediately it starts using the Enterprise edition features.

Erik Darling: When you're walking through the Secret Woods... Left, left, up, down. Then throw a boomerang.

Brent Ozar: This is where our transcriptionist is going to hate our guts.

Richie Rump: Place the bomb on the far wall.

Erik Darling: You have to kneel in front of the statue with the red orb. It only opens if you have the fire chain.

Brent Ozar: Make sure in Contra that you just stick with the spray gun. Don't change guns. The spray gun is the one you want to take through all of the levels... This is why we can't have nice things.

How do I create fragmentation for demos?

Brent Ozar: "Is there a way to create fragmentation to the table? I want to test my index optimization solution but when I hand my work to a QA team, I'm not seeing anything it needs to defrag." Oh man, we wrote code just like this. What did you end up doing, Erik, in order to generate fragmentation?

Erik Darling: There are a couple approaches you could take. One is GUIDs. GUIDs will always be fragmented because they insert all out of order. Another thing that you can do is if you have date columns, or not unique ascending ID columns, you can always just insert a range of values into those over and over again. They'll become, not like massively fragmented, but they'll get pretty fragmented over time if you dump in random dates from spreads of time, or random numbers from spreads of time between a range. If you just keeping dumping in the numbers 1 to 200, you're going to have to put numbers in all over the place in that index. If it's the leading key of the index then it's going to get kind of chopped up. Another thing that kind of works too is if you have longer text fields. You can update those. Sometimes you get pages to move around that way too. You get page splits and whatnot.

Brent Ozar: We're really focusing on building new tables and then dumping data into there, which is what I would recommend for you if you're going to test defrags. You don't

want to do it with existing tables because they just won't move around that much.
Erik Darling: My big question for this fellow human is why aren't you using Ola Hallengren's index defrag routine? It's free and it works really well. If you have a real job, it's so helpful.

Brent Ozar: ola.hallengren.com. O-L-A dot H-A-L-L...

Erik Darling: I'm going to paste that in before Brent falls off the spelling bee bus.

Brent Ozar: It has things like a timer on it, it will work for a certain number of minutes. It will break out gracefully. It will work on just the fields that you want, just the tables that you want, all kinds of things that are just spectacular. And it's totally open source so you can take it and then change it if you want instead of writing a new one from scratch. He keeps it updated with new features and versions, all kinds of stuff.

Erik Darling: It's also version and edition aware too, so you're not going to try and do something crazy on 2012 that you can do on 2014.

What backups should I do on the primary or the secondary?

Brent Ozar: A fellow human asks, "I have backup jobs running on my secondary for both the database and transaction logs. What's the best practice for doing backups on the primary?"

Erik Darling: Tara?

Brent Ozar: Where are you, Tara Kizer? Based on your terminology, I think you're using Always On Availability Groups. In that case, what I would actually recommend is always do your log backups on the primary because if the business tells you we're not allowed to lose more than say five minutes of data, or ten minutes of data, if synchronization breaks between the primary and your secondaries, or just if it gets behind, the secondary may be backing up old log data. It may not have current data. So I'm always going to run my log backups on the primary. My fulls, I may be okay offloading to a secondary since I can just go back to an older full backup and keep restoring all the transaction logs since. But whenever you backup from a secondary, there's two important things that you need to understand. One is that you have to license it because offloading backups equates to offloading production work. Two, means you have to run CHECKDB on it because if you're going to do a backup on something, you need to know that it's corruption free. Just because you're doing CHECKDB on the primary doesn't mean that the secondary where you're doing the backups is actually clean.

What's the name for joins on multiple fields?

Brent Ozar: Angie says, "In my company I'm seeing joins that are on multiple fields. For example, I'm joining from person to address and the joins in there are not just on ID but on a couple of IDs or a couple of different key fields. Is there a name for this kind of

join?" Oh man, that's a really good question. I don't know if I've ever seen a name for this kind, like compound keys?

Erik Darling: Multi-join predicates or something?

Brent Ozar: Yeah, or multi-predicate joins? I don't think I have ever seen a name for that.

Richie Rump: I used to just call them a join.

Erik Darling: They're totally valid and reasonable because a lot of times what happens is you'll find lazier developers who just know, "Oh, I have this key to this key." Then they get a bunch of duplicate data. They throw in a distinct rather than the appropriate join. So it's totally okay if you're joining on multiple keys. I'd just want to make sure that all of the keys that you need for the join are indexed together so that you're not having to key lookup, do a predicate key lookup somewhere else, or do a nasty hash join, or have a filter later on in the query.

Brent Ozar: Is there anything wrong with it? Not at all. Totally okay.

What's the CHECKDB parameter for indexed view checks?

Brent Ozar: Next question. "On the webinar yesterday," I did a webinar for Idera on filtered indexes, indexed views, and computed columns.

Erik Darling: Oh, sexy.

Brent Ozar: Right? Artisanal indexes. "Brent said you had to use a special parameter for CHECKDB whenever you have a filtered index." It was actually about indexed views. "I didn't get a chance to write it down. What was it and do Ola Hallengren's scripts do that?" Yes. It is extended logical checks. Extended logical checks will build a copy of your indexed view. So it does take more time. It builds a copy of your indexed view and compares to what's actually in the indexed view. By default, CHECKDB does not do that for you. So does it take longer? Yeah. Absolutely.

Erik Darling: I was going in the total opposite direction of that. I thought he wanted to just skip over everything. I was just like, physical only.

Brent Ozar: To explain, Erik's check does less work. Erik's check checks even less things, physical only, which is faster, much faster.

Erik Darling: Erik's checks used to have to go over like 50 terabytes of data at nights. Erik's checks had to go fast. Get out of the way.

Brent Ozar: Because at the end of the day, if you have indexed view corruption, it just means—just—just means data that's going to come back isn't accurate. But you can totally recreate that by dropping the index view and recreating it. So when you're in Erik's position and you've got 50 terabytes of data, at night you just can't check everything. It just doesn't work.

Erik Darling: Actually I have a question about that. If an indexed view is corrupt, how likely is it that it is just the view and not the underlying data? If you went to select from an indexed view and SQL is just like, "No, corruption." What are the odds it's the indexed view versus the underlying data or pages?

Brent Ozar: This is what's going to blow your mind. There's a known bug in indexed views when you use things like the merge command. When you merge against the underlying tables, the indexed view isn't updated, or it's updated with nulls. So it's physically not corrupt. It's fine physically. But logically, the data that's in it is corrupt. So when people do a select against it, they don't get errors. They don't get any corruption errors at all. It's just that the data that's coming back is wrong.

Erik Darling: That's so bad.

Brent Ozar: To me, that's the worst kind.

Erik Darling: It's not even indexed views though, it's merge.

Brent Ozar: Yeah. It's only merge with indexed views.

Erik Darling: But merge on its own does all sorts of other stuff, like skip foreign key stuff, and just referential integrity kind of goes out the window in a few different cases. Aaron Bertrand, I don't know the link off hand, but he has a long list of merge bugs. It's sad to look through them because they're like years old. You look through the list and so many of them are "Closed, this won't fix." "Closed, it's by design." "Still open." You're looking like, "Oh, god." Don't use merge. I beg of you. Please don't.

Richie Rump: I love the functionality of merge. I love it. As a developer, it makes things a whole lot easier, especially when you're dealing with data warehouse-y stuff. But, the bugs, ugh. It makes me scared.

Erik Darling: And the syntax.

Richie Rump: The could get used to the syntax. The syntax is weird but I could get used to that. But, ugh, bugs.

Brent Ozar: Paul says, "What would you use besides merge?" Unfortunately, it's plain old roll your own insert, update, and delete statements.

Erik Darling: The nice thing about those is that you can optimize them separately. You don't have to worry about, "Does this whole merge work good?" You get to have three separate things that work and you can fix them on their own and make sure they work on their own. You can batch them really well, which I'm a fan of.

Brent Ozar: So if you have even any slight interest in using merge, google for merge and the name Aaron Bertrand. There's a great huge two-page long list of bugs that are open with the merge statement.

Erik Darling: I'll just try to find the link while you guys talk.

Richie Rump: That's a feature. Not a bug. They're features.

Brent Ozar: Corruption is a feature.

Small talk about offices and company retreats

Brent Ozar: Someone says I need pictures on my wall. The catch with that is I use this for recording training videos, so it needs to be entirely green. The other walls in my office have lots of art. I have all kinds of crazy stuff including, I still haven't mounted it yet, but I have a whole bunch of graphics from the video game Asteroid that I'm going to stick across my wall. Plus, I have tires from an Audi. My office is like a garage now.

Erik Darling: But from whose Audi? We don't know.

Richie Rump: It is Chicago, you know, they kind of do that kind of thing.

Brent Ozar: It was parked on the street, that's just like giving permission.

Erik Darling: I wish my office were that well put together. I ordered my desk and my desk is coming, and a bookcase coming. For now, I just have some boxes and a telescope and a *Dune* poster hanging up behind me.

Brent Ozar: So the telescope, do you do it from inside the place or do you go up on a roof deck? What do you do?

Erik Darling: At our last apartment we had roof access, so we could go up to the roof and just look at stuff. You get the ambient light noise, so you can't see everything but we had some pretty good views. There was one night we caught Jupiter with several moons really well. There was one night where we caught Saturn where you could make out the rings. That was nice. We don't have roof access at this apartment unfortunately so I'm probably just going to keep the telescope around and stare at my neighbors creepily. [Laughter]

Brent Ozar: That's not a bad idea for a company excursion, like whenever we go do a retreat, see if we could find a telescope, that would be awesome.

Erik Darling: Yeah, because Tara likes to hike so it would get us out in the middle of nowhere. We could do a night out, tents and telescopes, getting too drunk to use telescopes.

Brent Ozar: Which reminds me of my other idea for the next company retreat. I have two bad ideas but this is one of them. A road trip. We pile everybody into a couple of campers and we go road tripping. We'd have to fly somewhere interesting, where it was like national parks or whatever. That could be fun.

Erik Darling: I'd be in for that but I'd feel bad because neither of us drive because we're city folks with Metra cards.

Brent Ozar: Yeah.

Erik Darling: On the other hand, we're the only ones who can always drink, so.

Brent Ozar: That's right. That's actually kind of a bonus. Suddenly I might have lost my license. Do you have another chair over there?

Richie Rump: Yeah, that's my chair. Since I redid the whole house and hardwired most of everything, I just have network equipment. I shoved most of it in the closet but it's just all boxes all over the place.

Erik Darling: Yeah, how's that going? Do you have full wired internets yet?

Richie Rump: Yeah.

Erik Darling: Nice.

Richie Rump: Mostly everything is wired now. I still need to cut a few more cables but the whole closet part is done. I got the last piece in yesterday afternoon. Plugged it all in, it's all on backup now. So I've got like four hours of backup when the network goes down.

Erik Darling: Check you out.

Richie Rump: During a hurricane, it helps out a lot to have all that stuff on a backup. You could actually watch TV and stuff.

Erik Darling: Cross your fingers, no hurricanes.

Brent Ozar: What do you do for cable? Do you have cable then? Or DIRECTV?

Richie Rump: Nothing. We cut DIRECTV off after 15 years earlier this year. We stream everything.

Brent Ozar: I was like, how are you getting TV? That makes much more sense.

Brent Ozar: We'll take a pair of questions. One person says, "Extended logical checks, does it take longer on databases if it doesn't have indexed views?" It also checks other things. Things like persisted computer columns, but there's a few other logical devices that it will go check. Also, one person says, "Can you create an indexed view on top of a view that has multiple joins calling other views?" Go hit the Books Online page on indexed views. There's a big, huge, long list of limitations. One of them is no self joins, no left outer joins. There's a bunch of things that you can't do inside an indexed view. Not uncommon there.

Erik Darling: They are disappointingly limited.

Brent Ozar: Yeah.

Erik Darling: You have to have a COUNT_BIG.

Brent Ozar: Not a count small, a COUNT_BIG.

Erik Darling: A regular count.

Brent Ozar: Even though they're the same thing under the covers, they both use bigints under the covers. Thanks, everybody, for hanging out with us this week. We'll see you next week on Office Hours. Adios, everybody.

Erik Darling: Later.

[Read the comments.](#)

Brent Ozar says:

[PasteThePlan.com Winners and Sample Plans](#)

When we announced [PasteThePlan](#) last week, we wondered what kinds of plans you folks would paste in. Six of you won free [Everything Bundles](#):

- Ajay Dwivedi
- Samuel Jones
- Stefan Gabriel
- Stephan Schon
- Steve Armistead
- Vesa Juvonen

But rather than stepping through their plans (which are awesome too), I'm going to show a few bits and pieces from the hundreds of plans you all pasted in during the first week.

[Long batch](#) – this one's got tons of issues, but even just the first statement has one of my favorites: parsing a string in order to build a subquery, and then checking to see if elements of that string are numeric. Cardinality estimation hell.

[Helpful comment block](#) – hey, whenever you have to read long queries in foreign languages, it's always good to start with a useful comment block. At least SET NOCOUNT ON is documented.

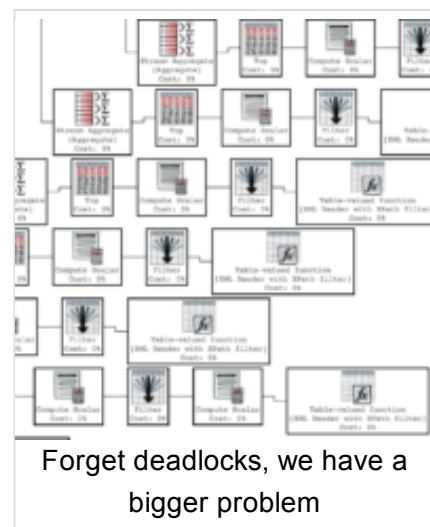
[Deadlock checking](#) – first thought: ooo, wow, someone's checking for deadlocks. Second thought: OH GOD, THAT PLAN. (Pictured at right.)

[Seemingly simple select](#) – it's just a one-line select * from a view – how bad can it be? Holy cow, how many tables are in this view?!?

[Shredding URL strings in the WHERE clause](#) – not to mention trimming phone numbers. Hoo wee.

[Dirty reads and table variables](#) – actually, Dirty Reed and the Table Variables sounds like a great name for a band.

[Nested selects causing table scans](#) – when people tell me Entity Framework writes bad code, I'm going to point them to this query and say no, YOU write bad code.



It's been so much fun watching folks paste plans in and [get help on StackExchange](#). We're already coming up with lots of ideas on how to help make this process easier for you and get you tuning advice faster. Stay tuned!

(Get it, tuned? Tune? Oh, fine, I'll go back to T-SQL.)

[Read the comments.](#)

[follow on Twitter](#) | [subscribe to our feed](#) | [forward to a friend](#)

Copyright © 2016 Brent Ozar Unlimited®, All rights reserved.

We're sending you this because you signed up for the BrentOzar.com Community. If you no longer want these community recaps, please unsubscribe. We're all about staying at Inbox Zero too.

Our mailing address is:

Brent Ozar Unlimited®
9450 SW Gemini Dr
ECM #45779
Beaverton, OR 97008

[Add us to your address book](#)

[unsubscribe from this list](#) | [update subscription preferences](#)