

# *DBA JumpStart*



Successful DBAs and Data Professionals reflect on:

*“If you could give a DBA just one piece of advice  
What Would it Be?”*

*A SQL Community Project*

**By John Sansom**

# ***DBA JumpStart***

**Successful DBAs and Data Professionals Reflect on**

**“If you could give a DBA just one piece of advice  
What Would it Be?”**

**John Sansom**

**[www.johnsansom.com](http://www.johnsansom.com)**

Dear Aspiring DBA,

Starting a new career is hard, and now so more than ever.

We live in a global marketplace of ever increasing competition for fewer opportunities. Choosing the right career path is arguably one the most important decisions that you can make in life and you want to be sure you're headed in the right direction.

A career as a SQL Server DBA is both challenging and rewarding. How much so, is determined by you.

If you're on the lookout for an easy paycheck, walk away now. You're not going to find that here. If however, you relish the thought of being in control of your destiny, where a continued commitment of time and energy really can make a difference, then congratulations. You are on the path to becoming a member of one the most rewarding technology communities I know of.

It is the very same community that has brought this ebook to life. A remarkable collaboration of talent and generosity from across the global SQL Server community, all made possible entirely by the good will of the people involved.

When I first contacted Brent Ozar to see if he would be so kind as to share a few words for a blog post I was working on, asking:

*"If you could give an aspiring DBA just one piece of advice what would it be?"*

.....I had no idea that we would end up here. Creating a collection of inspirational writing from over 20 SQL Server experts and community champions.

That's the great thing about our community, we bring out the best in each other. Brent could have just gone along with my original request but instead he saw the potential for something much more when he said:

*"This is a great idea that deserves something bigger. I want to challenge you to build a free PDF around this."*

What started life as just another blog post idea, has grown into something much more than I could have imagined.

Enjoy the ebook and if you have any questions or thoughts you'd like to share you can email me at [john@johnsansom.com](mailto:john@johnsansom.com) or reach me on Twitter [@SQLBrit](https://twitter.com/SQLBrit).

I wish the very best on your DBA journey.

Sincerely,

John Sansom

[www.johnsansom.com](http://www.johnsansom.com)

**Share This Ebook!**



# Contents

<a href="#"><u>Jonathan Allen</u></a> .....	1
<a href="#"><u>Jes Borland</u></a> .....	3
<a href="#"><u>Jason Brimhall</u></a> .....	5
<a href="#"><u>Denny Cherry</u></a> .....	8
<a href="#"><u>Robert L Davis</u></a> .....	9
<a href="#"><u>Richard Douglas</u></a> .....	11
<a href="#"><u>Rob Farley</u></a> .....	19
<a href="#"><u>Argenis Fernandez</u></a> .....	21
<a href="#"><u>Grant Fritchey</u></a> .....	23
<a href="#"><u>Steve Jones</u></a> .....	25
<a href="#"><u>Kevin Kline</u></a> .....	27
<a href="#"><u>Ted Krueger</u></a> .....	30
<a href="#"><u>Jennifer McCown</u></a> .....	32
<a href="#"><u>Bob Pusateri</u></a> .....	35
<a href="#"><u>Brent Ozar</u></a> .....	36
<a href="#"><u>Chris Shaw</u></a> .....	38
<a href="#"><u>Jason Strate</u></a> .....	41
<a href="#"><u>Michael J. Swart</u></a> .....	44
<a href="#"><u>Mike Walsh</u></a> .....	46
<a href="#"><u>Chris Yates</u></a> .....	51
ABOUT THE AUTHOR.....	53
<a href="#"><u>John Sansom</u></a> .....	54

## Jonathan Allen

### ***Build a Rewarding Career: Make Relationships a Priority***

*"If you could give an aspiring DBA just one piece of advice, what would it be?"*

*- Prioritise, and Socialise.*

I have been working in IT for almost 15 years and before that I worked with back office systems involving data analysis and reporting.

Gradually over time I began developing relationships and communicating regularly with Data Professionals in similar roles at other companies.

Now I have a network of friends and acquaintances across the UK and indeed all over the world that I speak to on an irregular basis, sometimes enquiring about family or social events and other times to bounce ideas off.

This makes my work so much more fun than being insulated in an isolated office, as the only DBA in a small team. Even if you are in a big team I would argue that meeting and talking to more people in the industry will broaden your appreciation of the bigger IT world.

Sure you will do okay, I did. But it was *only* okay though. Perhaps there was a chance that years ago I could have spoken to someone who knew something, met someone that might have helped me create something better or put me in touch with an opportunity that would have given my life a different trajectory.

I was happy in my bubble and who's to know if things could have gone worse rather than better? All I can say is that since I have joined in with the SQL community I have had more fun, I feel as though I have grown both as a developer and as a person, and I've been a more valuable professional for my employers.

"So how do I get involved?", you ask.

Well the best place to start would be to go along to your local user group meeting and find out who is close to you geographically and get into the community that way. Some groups run monthly, others every couple of months and some quarterly. Get on the mailing lists and you might even be close enough to choose to visit more than one every once in a while. Perhaps you can help out running the meetings – just something as simple as being on the door greeting people while someone else sets up the AV is an amazing help.

Next you might want to attend a conference or two. In the UK we have a healthy number of opportunities for you – SQL Bits is the longest standing SQL Server conference in the UK, there are also now 3 SQL Saturdays each year and SQL Relay conferences happen all over the country too.

Maybe you are not mobile and can't get to user groups or conferences. Well, even without leaving your desk you can access some great technical content and discussion. Join [SQLServerCentral.com](http://SQLServerCentral.com) or [Ask.SQLServerCentral.com](http://Ask.SQLServerCentral.com) or [stackoverflow.com](http://stackoverflow.com) or [sqlskills.com](http://sqlskills.com) or [sqlpass.org](http://sqlpass.org) or [sqlserverfaq.com](http://sqlserverfaq.com).

There are free videos on [sqlbits.com](http://sqlbits.com) from previous conferences so you can catch up on what has happened in previous years. [sqlpass.org](http://sqlpass.org) hosts the online marathon 24 hours of PASS each year which is like tag-team learning – 24 one-hour sessions on SQL Server from people booked to speak at the annual PASS Summit. It's all free. If you do nothing else, visit those sites and get registered.

OK, you said Prioritise and Socialise but I'm only seeing Socialise...

That's right, prioritise your socialising. Be sure to do it. I delayed quite a while because I was uncertain about getting involved. Would I know enough, would I ask stupid questions, would I actually be able to help anyone and would I be welcome?

The answer is yes, in every case, for every person. The SQL Community doesn't judge you by how much you know, how stupid your questions are (believe me I have asked plenty of those in my time) or anything else. We are just glad that new people are getting involved, it means that there are more people to share with.

Don't put it off until tomorrow. Join at least one site in the references below. If you are not already involved in the SQL community in some way, then make it a goal for today. Join now, I look forward to meeting you.

## Links and References

UK user groups contact details: <http://sqlsouthwest.co.uk/other-user-groups/>

PASS: <http://sqlpass.org> / <http://www.24hoursofpass.com/>

### Q + A sites

- <http://ask.sqlservercentral.com>
- <http://sqlservercentral.com>
- <http://stackoverflow.com/>
- <http://www.sqlskills.com/>

### Other UK community sites

- <http://www.sqlserverfaq.com>
- <http://sqlrelay.co.uk>
- <http://www.sqlbits.com>
- <http://www.johnsansom.com/>

## **Bio:**

Jonathan has been working with SQL Server since 1999. He enjoys performance tuning, development and using SQL Server to provide appropriate business solutions.

He is the founder and leader of the [PASS SQL South West user group](#), is a moderator at SQL Q + A forum [ask.sqlservercentral.com](http://ask.sqlservercentral.com) and is on twitter at @fatherjack.

He's spoken at [SQLBits](#) and [SQL in the City](#), as well as many local user groups across the UK.

## Jes Borland

### ***“Be Passionately Inquisitive”***

John Sansom asks, “If you could give an aspiring DBA just one piece of advice what would it be?”

**Question everything** about the data - access to it, security of it, and how it's being used.

I don't mean this negatively. You can question why a developer wants sa access, or why the business wants to implement a feature, or why the same process has been done the same way for ten years when there is a new method that would be faster and more secure - without being negative.

The DBA is the gatekeeper of the company's data, and, as I always say, **the business's data IS the business**. Understand that this is a very important role, and should not be taken lightly. It is not your job - and it should not be - to say yes to every request, or to approve every piece of code the developers want moved into production.

It is your job to question the users as to why they want something and to ask if that fits in with the company's objectives. Does it help the business reach its goals, without spending too much money or exposing the company to too much risk?

It is your job to learn more about new features and technologies, and question if those would fit in your company. Don't always assume that the way something has always been done is the best way to do it. In one of my previous jobs, I helped replace some VB script with PowerShell - there was resistance to change, but it was more efficient in the end. That change started with me asking why we weren't using PowerShell.

When a developer wants to check code in, there should be a code review. If you are a DBA, you don't have to be a T-SQL expert, but at least know enough to identify common problems. If it's complex, question the developer, asking if she's tried multiple ways to get the information, and if it's the most efficient. If you see something that looks out of place, ask why it's there. You may learn something!

Learn how to question without being inquisitorial. You don't have to be confrontational. Never approach someone as if he has done something wrong - you are simply asking to clarify, to learn more.

The greatest thing about asking questions is that you will learn. You will learn more about SQL Server. You will learn more about the business. You will learn interpersonal and communication skills. This will help you grow - your knowledge of SQL Server will grow, and your career will grow.

Remember: question everything. You need the company's data. You need it be secure. You need to it be used appropriately. Questioning will help you meet these objectives.



**Bio:**

Jes Borland is a Consultant with Brent Ozar Unlimited and a Microsoft SQL Server MVP.

She holds an AAS - Programmer/Analyst degree, is a Microsoft Certified Professional in SQL Server 2012, and has worked with SQL Server as a developer, report writer, DBA, and consultant.

Her favourite topics are administration, automation, and SSRS. She is President of FoxPASS, founder of Tech on Tap, and a LessThanDot.com blogger.

She frequently presents at user groups, SQL Saturdays, and other community events.

She is also an avid runner and chef.

## Jason Brimhall

### ***Don't Settle for Just a Job, Have a Career***

In my day to day operations I have the opportunity to work with people in various capacities in regards to data. Sometimes it is in the capacity of a mentor, sometimes in the capacity of a consultant, and sometimes just in the capacity of the dude that fixes the problem.

I enjoy working as a database professional. There may be times when I want to scream or yell or pull out my teeth and hair. Then there are times when I just bounce off the walls with joy and pleasure. Some may call that a manic-depressive disorder. They just don't understand the true life of a data professional.



### ***Reminiscing***

In becoming a data professional, I took the long route to get to where I am. I made the decision to work with SQL and learn about it 17 years ago. I made the decision to learn about SQL because I viewed it as a really difficult thing to learn. I wanted that challenge. Then again, back then I also enjoyed the challenge of learning to configure Cisco routers.

Early on I passed the Microsoft exams for SQL 6.5. A couple of years later, I finally landed a job where I got to touch a database. That was part of my duties with being in one man shops. I worked in a few of those one man shops for a while where I had to be the exchange admin, domain admin, DBA, and even janitor at one shop. I don't miss the days of having to fix the plumbing, in between troubleshooting performance issues and checking the router for DoS attacks.

Eventually I got an opportunity with a larger enterprise to be a production DBA. All I had to do was work with SQL Server all day long. It was fun designing metrics and monitors to alert on various thresholds while saving the company oodles of money. I really thought I was learning something cool. I thought I was doing pretty good too.

Fast forward a little more and a couple of job changes, I found myself living in Las Vegas and getting more involved in the community. Boy did I learn quickly how little I actually knew about SQL Server. Sure, there was reading of posts, books and forums before that. But that just didn't quite open my eyes like becoming involved.

I soon started applying myself even more so I could learn more about SQL Server and then be able to try and teach those things to the developers where I worked. I also started working on trying be good enough to be able to teach people at User Group meetings. Throw in the efforts to answer questions on forums and writing articles - and it was an explosion of learning.

Now I present pretty regularly at User Group meetings. I travel around the world to present at SQL Saturdays. I have contributed articles and co-authored a book. I also had (still have) the sweet opportunity to participate in the Mentoring project hosted by Andy Warren. I even went so far as to challenge myself and attained the MCM. Yet, I

know that I have really only scratched the tip of the iceberg with SQL Server. There is so much to learn about SQL Server still. If I were to compare myself past to present, I would rate my skills in various areas lower now than I probably did back in the day.

Through the years, and more particularly the more recent years, I have observed many team-mates and DBAs for clients. These observations have revealed some good and some bad. When I notice certain behaviours that need to be changed, I try to use it as a teaching opportunity.

### ***Price of Rice***

One thing I find myself doing on a frequent basis is trying to gauge if I might be treating my work as a 9 to 5 job or if I am treating it like a career? Am I just punching the clock or am I investing in myself and improving my skills? Am I helping others improve their skills or am I hoarding the knowledge like an Oracle DBA?

As I observe others I can't help to ponder some of those same questions. For instance, if I encounter a veteran DBA of 10 or so years that can't perform a transaction log backup, I will wonder if being a DBA is just a job for that person. The way you treat your work duties often speaks volumes about how much you care for the quality of work you do and is also revealing in how much one values their skills.

Taking that same DBA that can't perform a log backup, I might start to wonder if there is a time investment outside of work to better their skills. I might wonder why I have to show that person five or six times how to perform that log backup. This may sound a little judgemental but it is not meant in that way. Let's call it an informal assessment to try and figure out how to help that person become more efficient at performing their job duties.

As a data professional I think it is an important thing to do. Spend some time on introspection and try to determine just how much of a career the job is. Find out if it is a career or if it is on the short end of the spectrum that points to it being just a job.

As a team lead, I like to give everybody on the team the task of taking 15-30 minutes each day (on the clock) to improve their skill-set in some way. This is a tactic that does not work in all environments and with all employers - I get that. But if that 15 minutes a day means that the team-mate will be more efficient down the road, it is a good investment. If that 15 minutes means there will be less time redoing some work, then it is time well spent.

As I mentioned earlier, there is plenty about SQL Server that I still need to learn. An important component of learning is to invest some time. It's a matter of finding a topic and then taking the time to research. I do my research by reading and then experimenting. Once I feel comfortable with that research, I will typically write about the topic. Why? It helps to solidify or to disprove some of the principles just learned. It also helps to cement that research into memory. I also like to do it because it serves as a personal archive that I can refer back to at some future point (I have done that plenty of times).

Another thing I like to do after learning about something different in SQL Server is to present it to a group of people. That group can be co-workers, a user group, or at a SQL Saturday (as a few examples). The beauty of presenting on the topic is that it

johnsansom.com

helps me to embed that knowledge a little further. It also helps me to try and gain an even deeper understanding of the topic to be able to answer questions that may arise. Best of all is that it helps to disseminate knowledge to others.

### **Recap**

For me, being a data professional equates to a career. I get that for some it is just a J O B - and that is fine. For some, it may just be a J O B because they have not figured out how to advance it into a meaningful career. Those people don't want to just be a clock puncher and want to make something more of their chosen profession.

As a data professional, I suggest the following practices to help turn your profession into a career:

1. Regular introspection - check in with yourself on occasion to keep yourself headed in the right direction.
2. Learn something new - Treat this like a cursor. Keep finding something new to learn and act on it.
3. Give Back and Get Involved - When you learn something new, teach it to somebody or post it on a blog. This helps give back to the community and more people can learn and grow.

These three simple steps can help turn just another J O B into a quality career. Invest in yourself.

### **Bio:**

Jason Brimhall has 10+ years experience and has worked with SQL Server from 6.5 through SQL 2012.

He has experience in performance tuning, high transaction environments, as well as large environments. He is currently a DB Consultant and also a Microsoft Certified Master(MCM).

He is the VP of the Las Vegas User Group (SSSOLV).

Jason is a regular blogger and you can read his articles at <http://jasonbrimhall.info>

Jason is a co-author of the book [SQL Server 2012 T-SQL Recipes: A problem solution approach](#).

## Denny Cherry

### *Keep Learning*

The best advice that I can offer someone entering the field of Database Administration is to keep learning.

If you think you know everything that there is to know about this product we deal with day-in and day-out called Microsoft SQL Server, you are wrong.



There are so many little pieces to learn about how the engine works with data, how statistics work, how memory is managed, how data is read and written, and most importantly how all of these pieces fit together just so, to make a SQL Server that runs fast.

Just to make our lives supporting this software called Microsoft SQL Server that much harder Microsoft has decided that they are going to release a new version every 2 years or so.

Instead of just having to manage one or two versions like we did back in the SQL 7 and SQL 2000 time frame, we now have to support 4 or 5 versions (I've got clients with SQL 2000 up through SQL 2012, and some will move to SQL 2014 right when it comes out).

Just because SQL Server does the same thing in the new versions (stores data) doesn't mean that things aren't different in the new versions.

This is especially true in SQL Server 2014. There are a bunch of new features and changes to existing features that will change how some very low level pieces of the database engine work, so we are back to reading and learning more so that we can keep up with the changes to the platform.

### **Bio:**

Denny Cherry is the owner and principal consultant for Denny Cherry & Associates Consulting and has over a decade of experience working with platforms such as Microsoft SQL Server, Hyper-V, vSphere and Enterprise Storage solutions.

Denny's areas of technical expertise include system architecture, performance tuning, security, replication and troubleshooting. Denny currently holds several of the Microsoft Certifications related to SQL Server for versions 2000 through 2008 including the Microsoft Certified Master as well as being a Microsoft MVP for several years.

He has written several [books](#) and dozens of technical articles on SQL Server management and how SQL Server integrates with various other technologies.

## Robert L Davis



### ***Hard Lessons: Learning to Say No***

There are many lessons that we will learn the hard way in our careers. In the long run, we are often better professionals for having learned the lesson. One of the hardest lessons I had to learn in my career was learning to say, "No". "No," to project managers who wanted data analysis right away. "No," to managers who wanted you to take on another project with a short deadline. Most importantly, "No," to myself that I wasn't going to spend every waking hour on my job.

### ***When I Was a New Data Professional***

When I got my first IT job back in 2000, I felt like I had to work twice as hard as everyone else to make up for my noobiness. I thought for sure that people would realize that I was working above my skill level if I didn't compensate for it by working harder and longer. So I compensated.

My first IT job, I was hired to help out the two existing database developers. Two months into the new job, one of the database developers I was helping transferred to a different department. I had not been told this was coming. The same morning the other database developer quit without notice. Two months into my first IT job, I was the database development team.

I had to figure out all of the things the other developers had not bothered to show me because they took care of it and I didn't need to know. Well, now I needed to know because things were not working and people were just starting to notice that they were broken. I worked 12 hour days in the office and then I worked more when I got home. I found lots of existing processes the other database developers managed that had not worked properly in months. Within two weeks of taking over the whole team, I had everything that was broken running smoothly and the new project I had been assigned was well ahead of schedule.

Sounds like I was doing great, but in reality, I was billing 60 hours a week and actually working at least 80. Even then, I wouldn't have made it through this time if not for the help of one of our application developers who helped me by creating some automated processes to replace manual processes and the help of our sysadmin who helped me fish out where everything was and how it worked.

Over time, we filled the empty slots on the team again, with me as team leader. Still, I didn't slack up on the hours I worked or the schedule with which I delivered things. The application developer who had helped me became a permanent member of the database development team as well and was really instrumental in helping taking our processes to a whole new level of automation.

Still I worked unbelievable hours. Then the dotcom balloon burst and venture capital dried up. The company I worked for had massive layoffs and eventually closed. It

reopened as a new company with a skeleton crew. I was once again the whole of the database development team. I continued working massive hours and delivering much more than one person possibly could. I was no longer a noob in IT, but I was still in overcompensate mode.

It was at this point that I decided I wasn't going to kill myself and keep doing the work of a whole team no matter how many hours I had to work. I started telling people, "No."

You want data analysis for a new client by your meeting tomorrow morning? "No, I can't get that done today. I have higher priorities right now. I can have it for you by next week."

You want me to help with the website development (yes, I was doing web development for the company too) and still deliver the data portion for it as well in the same time frame? "No. I can do one or the other, not both."

### What Happened When I Started Saying No

Much to my surprise, when I starting saying no to people, they started realizing just how much I was doing. When I started explaining to people when I could reasonably deliver something and why, they learned that I had more on my plate than a single person could possibly do.

It turns out that when I was working 80 hours a week and always said yes to every request, I made myself look like I had plenty of spare time on my hand. They didn't see the hours I worked nor the total amount of work I delivered. All they saw was that I always had time to do their requests without delay. To their perspective, the more work I did, the less it looked like I was working. I was doing myself a much bigger disservice than I had imagined.

When you sacrifice your personal time to deliver work items, nobody notices and nobody realizes how hard you are working. Learn to say, "No," and set realistic expectations and people will realize how much work you are doing and will appreciate your work a lot more. And on top of it all, your personal life won't suffer because you spend every waking hour working, like I used to do. I learned this lesson the hard way. I hope my story will help you learn it the easy way.

### **Bio:**

Robert L Davis is a SQL Server 2008 Certified Master, an experienced DBA, evangelist, speaker, writer, trainer, and consultant. He is the Principal DBA with Outerwall, Inc. and consults part-time via SQL DBA Master. He has worked with SQL Server for more than 13 years, including recent stints as a Sr. Product Consultant and Chief SQL Server Evangelist for Idera Software, Program Manager for the SQL Server Certified Master Program at Microsoft Learning, and as a production DBA at Microsoft.

## Richard Douglas

### ***Why You Must Always Remain Vigilant***

Before we get into the swing of things I'd like to thank John for the opportunity to participate in this endeavour. It's a great idea and I'm sure it will add a lot of value to our community.

In my post [Which personal traits should a #SQLPASS Director have?](#) I shared excerpts from my personal report created by StrengthsFinder 2.0. There is however another trait that I feel that every DBA should have. That trait is Vigilance, vigilance is key to a successful career in most professions. As a Database Administrator, guardian of the security and integrity of data, it is arguably the most important trait you can have.

The key thing to take away is that a good DBA is proactive not reactive. You can never become truly proactive if you do not remain vigilant. The job of a SQL Server DBA can be quite different than that of another DBA from another RDBMS platform. Part of this is due to the width of features which sees some DBAs as a jack of all trades covering tasks likes monitoring as well as designing cubes and reports. Whilst others are the victim of server sprawl and the number of instances in their estate registers in the hundreds.

Saying that you should be proactive is all well and good, but how can you as a data professional achieve this.

- Automating regular tasks
- Baselining
- Documenting
- Define your DR plan

I'll touch briefly on each of these points to cover why I believe they are important and how being vigilant helps your organisation and your career. It is however the last point that I want to talk about today and one aspect of your DR planning in-particular.

#### ***Automating regular tasks***

I'm constantly amazed by the number of IT professionals who have a manual daily check list. It's great that they have the checks and are doing them, but manually?!? Come on, there's a better way. The paid for editions of SQL Server come with something called "SQL Server Agent" which is in essence a scheduler. Even if you aren't lucky enough to be using one of the premium editions of SQL Server you can still schedule scripts through the Windows Task Scheduler.

Make sure you have the SQL Agent installed and it is set to start automatically (there's caveats with clusters). Once you have this set up you'll need to do the following:

- Set up [Database Mail](#)
- Create some [operators](#), make sure that you use distribution lists here rather than hardcoded individuals. It makes things much easier as you scale out or have changes in your departments.
- Look to see which [alerts](#) are relevant for your environment.

As a proactive or vigilant DBA there were a few jobs that I set up to make sure I was aware of everything I needed to be. Why not challenge yourself to do the same:

### **Create a job to test Database Mail**

This may sound strange, but have you ever been on call only to find out the next day that everything was down without you knowing? Relying on your mail system can be a single point of failure! Be vigilant and build in checks. If you don't receive alerts at certain times throughout the day and night you know there is a point of failure in your communication chain, at that point you can be proactive and dial in to find out what is going on.

### **Create a job to check for job failures in MSDB**

Everyday I used to have a job that read in the job history tables and checked for failures. Why did I do this? Simple, there were other teams and anyone in those teams could have created a new job which may not have had an operator associated with it. If there's no operator, or you are not in that list then you won't know there has been a failure.

### **Create a job checking your backups**

This in part goes back to the testing database mail job. I have been in situations where I didn't receive failure alerts because jobs hadn't run they had been disabled or the SQL Server Agent service had been stopped. Make sure you are never in this situation by creating a job that will report to you all backups that did and didn't happen in the last 24 hours. Bonus points if you check to make sure there are transaction log backups when running in the full recovery model.

### ***Baselining***

Baselining needs to be on every DBAs to do list if it's not being done already. A lot of people don't really understand why though. To answer this let me first ask you a question: How many calls a day do you receive from users saying their application is slow and how do you measure that? Perceptions of time become very distorted when people are stressed, you need a way to categorically say what the difference between now and a "good level of performance" is to know if there truly is a problem. How do we do this? You nailed it - baselines.

Simply put baselines are the average values over a period of time that allow you to compare performance levels to another point. I'm not going to tell you everything that you need to cover here as baselining is a very involved topic. You can check out a

white paper that I co-authored [here](#) that includes a portion on baselining as well as a webcast that I recorded for Dell Software called [The Day After Tomorrow;Why you need to baseline](#).

Once you have created your baseline to create your benchmark or comparison point you simply take another and take a look at the differences. This way you can quantify to the user if there are performance problems or not based upon the workloads you are seeing. Once again, this all comes down to being proactive and vigilant. Really you need to be taking your baselines regularly and for different types of workloads, all this is covered in the [webcast](#).

## **Documenting**

Yes documenting is boring and it is time consuming, but hey you now have a bunch more time from automating a load of tasks that we covered in a previous section of this post! There's lots of reasons why you should participate in both creating and reading documentation. Let me provide a few examples:

### **Creating documentation.**

I'm going to be covering DR (Disaster Recovery) plans in more detail later on. DR Plans however, are probably the most important document you are going to write, possibly only second to writing your CV/resume or your will. There's also a good chance that if you don't have good DR documentation then you are going to need up to date copies of your CV/resume and/or will sooner than you would have liked!

### **Create coding standards.**

You may think I am either mad or very anal for saying this, there is good reason though so stick with me on this. Ever heard of plan cache pollution? Plan cache pollution is where you have a number of very similar plans in the plan cache that really could have been consolidated into one plan. What does this mean to you? Well it means that this area in memory contains more information than it needs to, this is memory that could and should be used to store data in memory so you don't have to make expensive trips to your IO subsystem.

Let me show you how this works. The examples below were created in SQL Server 2012 Developer edition, they really should work in anything older than 2008 though, as long as you choose an appropriate database and query.

The first thing I am going to do is to select the database I wish to use and clear the plan cache. PLEASE DO NOT DO THIS ON A PRODUCTION INSTANCE

```
USE AdventureWorks2012;  
GO
```

```
DBCC FREEPROCCACHE;  
GO
```

We're then going to run the following queries:

```
SELECT type FROM sys.objects WHERE name='sysfiles1'
GO
SELECT type FROM sys.objects WHERE name='sysprivs'
GO
```

We're not really bothered about the results here.

Next we're going to query the plan cache to look at the plans that were created. We're going to take special note of the query\_hash attribute. Microsoft has this to say about the query\_hash attribute:

*"When searching for resource-intensive queries, you should consider how to find and tune similar queries that collectively consume significant system resources. The [sys.dm\\_exec\\_query\\_stats](#) and [sys.dm\\_exec\\_requests](#) dynamic management views provide query hash and query plan hash values that you can use to help determine the aggregate resource usage for similar queries and similar query execution plans."*

This query will allow you to look for the above queries in the plan cache:

```
-- Find the query execution statistics row(s) for the query
SELECT
    sql_handle,
    plan_handle,
    execution_count,
    query_hash
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.plan_handle) AS sql
WHERE
    sql.text LIKE 'SELECT type FROM sys.objects WHERE name %'
GO
```

The results will look something like this:

	sql_handle	plan_handle	execution_count	query_hash
1	0x020000006F5DA6239FFD830C24D9BCF582EA3F3B7C586A3...	0x060005006F5DA62300FFD9F80400000001000000000000...	1	0x046BA3EC3D3257E8
2	0x020000000C8F842B359A732192ADED1B19D900CCC1D86BD...	0x060005000C8F842B30F7D9F80400000001000000000000...	1	0x046BA3EC3D3257E8

As a vigilant DBA I would be thinking; Hang on, if these plans are similar what can I do to make them use the same plan and make better use of my resources. That's going to benefit all of my users.

How do we do this? Prepare your queries!

The queries I performed before were "ad hoc" queries. If I were to write them so that they were prepared then they would share the same plan and I can save space in my cache. This means more memory is free elsewhere, it will probably be used to store

johnsansom.com

data in my buffer pool meaning less expensive trips to my IO subsystem and that has to be worth doing.

So now if you run the following code you will see that both queries will use the same plan:

```
USE AdventureWorks2012;
GO

DBCC FREEPROCCACHE;
GO

/*
SELECT type FROM sys.objects WHERE name = 'sysfiles1'
GO
SELECT type FROM sys.objects WHERE name = 'sysprivs'
GO
*/

DECLARE @MyVarcharParm VARCHAR(20)
SET @MyVarcharParm = 'sysfiles1'
EXEC sp_executesql
    N'SELECT type FROM sys.objects WHERE name = @Parm',
    N'@Parm VARCHAR(20)',
    @MyVarcharParm;
GO

DECLARE @MyVarcharParm VARCHAR(20)
SET @MyVarcharParm='sysprivs'
EXEC sp_executesql
    N'SELECT type FROM sys.objects WHERE name = @Parm',
    N'@Parm VARCHAR(20)',
    @MyVarcharParm

-- Find the query execution statistics row(s) for the query
SELECT sql_handle, plan_handle, execution_count, query_hash
FROM sys.dm_exec_query_stats AS qs
CROSSAPPLY sys.dm_exec_sql_text(qs.plan_handle) AS sql
WHERE sql.text LIKE '%SELECT type FROM sys.objects WHERE name %'
GO
```

	sql_handle	plan_handle	execution_count	query_hash
1	0x0200000070446C085AF5B0A8E167482453EBE19792A6D6...	0x0600050070446C08B0E8D9F804000000100000000000...	2	0x0468A3EC3D3257E8

Obviously on my laptop this really won't amount to much of a saving, but what if you had recently consolidated lots of environments in a physical to virtual project, some of these databases are likely to have chatty applications and could be hitting the same SAN. Any reduction you can provide here will make an improvement to the overall performance of not just than one database, but the guest, the host and every other instance that shares that SAN.

### ***Define your DR plan.***

Your company's data is its lifeblood. Don't let HR fool you people aren't what makes a company successful, protecting its data is. Microsoft and Apple are both still here despite the departures of Bill Gates and Steve Jobs, but if one of these companies had lost the database with all their financial information and had no disaster recovery process then it would be a much different story.

Nearly every day I hear customers talking about their backup strategy and I have to correct their thinking. Having a backup strategy is not the same as having a restore strategy. If you put the process of backing up your data in front of how you are going to retrieve your data in a disaster scenario then you will come up short. At this point you'll wish you had read my section on creating documentation and having an up to date CV/resume. As a vigilant DBA it's really important that you think about how you are going to restore your databases and how long that will take and the point in time you will be able to recover to rather than how much data you can afford to store on your servers.

The week before the PASS summit I recorded a session called "[I got 99 problems but my backups aint one](#)" for the [PASS Virtual DBA Chapter](#). This section covers a few of the core messages I mention in that webcast.

When planning your restore strategy there are two main things to consider, these are:

- Recovery Point Objective (RPO)
- Recovery Time Objective (RTO)

Recovery Point Objective (RPO) is the amount of data you can afford to use. As a vigilant DBA I will ask the business owners of the databases I run for them. This is an important distinction, I do not own these databases I maintain and protect these for other people. I as a vigilant DBA am providing a service to other people within and potentially outside of my organisation. When I receive a new database I need to know who owns this from a business perspective. Once I know that I will book a meeting with that person to ask them about their data and how important it is. Only at that point can I decide which types of backups and how frequently those backups can be performed.

It is not uncommon for databases to be backed up using a full backup in the simple recovery model, but what if your business owner cannot afford to lose up to 24 hours worth of data? Are they even aware of that risk?

Recovery Time Objective (RTO) is the amount of time you will have to restore your data to a usable state. Again, I have seen some horrible "backup plans" these include monthly full backups with transaction log backups every 30 minutes. The reply I hear is, but we have a 30 minute RPO, we'll smash that. That may well be (as long as your log chain hasn't been broken) but how long could it take to redo and undo transactions in the recovery process for up to one month's worth of data? I don't mind saying that is not a job I would fancy doing!

We as a collective unit of database professionals need to be smarter and take a wider view. Yes I am covered by backing things up in this fashion. Is that really the right solution though?

Options you can use to make your backup, and subsequently restores faster. These include, but aren't limited to:

- Writing to multiple backup devices
- Increasing the BufferCount parameter
- Increasing the MaxTransferSize parameter
- Use compression

By writing to multiple backup devices at once you can actually write data away in parallel, this means we can write to multiple blocks via multiple disk spindles at the same time. This makes the write process much faster, to aid this further by using compression (2008 Enterprise, 2008R2 and on Standard edition. N.B. there are third party applications that will support previous versions and provide more than one level of compression) you are writing less information away further reducing the traditional write bottleneck. Interestingly compression isn't enabled by default, so make sure that you test the extra CPU usage won't cause you any problems and then enable it.

BOL defines BufferCount as "Specifies the total number of I/O buffers to be used for the backup operation." and MaxTransferSize as "Specifies the largest unit of transfer in bytes to be used between SQL Server and the backup media." In layman's terms. You can tune these parameters to make your backups and restores much much faster. There is of course a trade off.

Increasing the values for these values will consume extra memory during the backup / restore process. The calculation is pretty straightforward though it's BufferCount \* MaxTransferSize. One thing you may not be aware of is that this are of memory is outside of the memory you can configure in SQL Server, we used to call this area MemToLeave. If you haven't already set your maximum memory configuration setting then you really should look at it. I typically recommend that people read the memory chapter from SQL Server 2008 Internals and Troubleshooting by Christian Bolton. Christian is the lead author of the book and the author of that particular chapter, it will show you how to work out how you should set the maximum memory option for your environments.

Baselining, you thought we had already covered this didn't you? Well yes I had, but we are performance tuning and we can't tune anything without a baseline! We need to know how things are working now now so we can tune them. Microsoft doesn't display the parameter values it uses for the BufferCount and MaxTransferSize. You can however see this information if you use a couple of trace flags. Trace flag 3213 will provide the values used, these need to be used in tandem with either trace flag 3604 (to screen) or 3605 (to log).

## **Summary**

All of the points that I wanted to bring up:

- Automating regular tasks
- Baselining
- Documenting
- Define your DR plan.

Were all ways of bringing to light some of the skills you need to be successful in your job as guardian of your company's data. Being vigilant means that you know you need to use these skills and will use them to cater for all eventualities.

I hope that you have found this post useful, thank you for putting aside the time to read it.

## **Bio:**

Richard Douglas is a Systems Consultant for a vendor of SQL Server solutions where he specialises in SQL Server providing solutions and system health checks to organisations across the EMEA region.

Richard has recently taken on the responsibility of being Editor in Chief of the community site [Toad World](#) which provides articles on SQL Server and a blog syndication service, other duties see him regularly presenting webinars to both US and EMEA audiences.

He is also a keen member of the SQL Server community, he founded and runs a PASS affiliated chapter in the UK ([www.SQLStudy.org](http://www.SQLStudy.org)) and is on the organising committee for a national event called SQLRelay.

You can find him on Twitter under @SQLRich and blogs at [SQL.RichardDouglas.co.uk](http://SQL.RichardDouglas.co.uk)

[Read more](#) about Richard.

## Rob Farley

### ***Making Music Through a Closer Relationship With the Data***

Aspiring DBAs have a lot to learn. It's really easy to be shown what to do in various situations, and create a cookbook of sorts, which will be tremendously useful in their career. However, if they want to go beyond that, they need to understand things better.

My advice would be to learn how things work, so that you can take the right approach. For example, it's really helpful to understand that the best kind of data access is through non-clustered indexes, which prepare the data for your query before you run it, or that replication involves a bunch of processes spun up by SQL Server Agent to run all of the various components.

SQL Server is a brilliant product, written by some amazing people. The system is designed to be able to not only look after your data, maintaining its integrity and controlling how changes get applied effectively and quickly through the use of a transaction log, but also providing mechanisms to track the state of the data, and how to execute queries based on this.

The thing is that from time to time, there is troubleshooting to do. This might mean solving a corruption problem, or finding out why a particular query takes much longer than it should. At this point, if you only have the cookbook, you might have trouble.

To go off-piste, it's best to have a good understanding of what's going on behind the scenes, so that you can look for things that don't seem quite right. This doesn't necessarily mean being able to attach a debugger to `sqlservr.exe`, but you should be able to recognise a lot of what's going on with indexes (and heaps), the processes that run, wait statistics, and so on.

I liken it a bit to music. When I was a kid, I was taught to play the piano, and learned to move around notes quite well. Getting around the notes is a bit like typing – you just have to put your fingers in the right places at the right time.

Of course there's more to it, as you have to hit the notes with the right speed and according to the right rhythm, but there are similarities. The problem with this approach is that there's only so far you can go by following sheet music.

I could play music that was in front of me, but couldn't improvise very well until I learned about how music worked – about the relationships between chords, and progressions that worked (and ones that didn't!). Now when I play the piano, I move between chords, and have a closer relationship to the music than when I was just following instruction.

Data is the same. Business Analysts need to build a closer relationship with their data through self-service reporting, and DBAs need to build a closer relationship with the system that looks after that data. Not just understanding the logical structures of



tables, views, functions, etc., but also what's going on behind the scenes to make sure it's all okay. Then you get to start tuning your data so that it follows better chord progressions, sounding better to everyone's ears.

**Bio:**

Rob Farley of LobsterPot Solutions is an MCM, an MCT, a SQL Server MVP (since 2006), and a PASS Board member.

He blogs at [http://sqlblog.com/blogs/rob\\_farley](http://sqlblog.com/blogs/rob_farley) and tweets as @rob\_farley.

He enjoys it when he can help people, and when Arsenal wins.

## Argenis Fernandez

### *Learn By Helping Others*

I remember back in 2001 when my friend and former SQL Server MVP Carlos Eduardo Rojas was busy earning his MVP street cred in the NNTP forums (a.k.a. Newsgroups).



I always thought he was playing the Sheriff, trying to put some order in a Wild Wild West town, by trying to understand what these people were asking. He spent a lot of time doing this stuff and I thought it was just plain crazy. After all, he was doing it for free. What was he gaining from all of that work?

It was not until the advent of Twitter and [#SQLHelp](#) that I realized the real gain behind helping others. Forget about the glory and the laurels of others thanking you (and thinking you're the best thing ever - ha!), or whatever award with whatever three letter acronym might be given to you.

It's about what you learn in the process of helping others.

You see when you teach something, it's usually at a fixed date and time, and on a specific topic. But helping others with their issues or general questions is something that goes on 24x7, on whatever topic under the sun.

Just go look at sites like [DBA.StackExchange.com](#) or the [SQLServerCentral forums](#). The questions are coming in literally non-stop and from all corners of the world. And yet a lot of people are willing to help you, regardless of who you are, where you come from, or what time of day it is.

In my case, this process of helping others usually leads to me learning something new. Especially when the question is not really something I'm good at. The delicate part comes when you're ready to give an answer but you're not sure. Often times I'll try to validate with Internet searches and what have you. Often times I'll throw in a question mark at the end of the answer, so as not to look authoritative, but rather suggestive. But as time passes by, you get more and more comfortable with that topic. And that's the real gain.

I have done this for many years now on [#SQLHelp](#), which is my preferred vehicle for providing assistance. I cannot tell you how much I've learned from it. By helping others, by watching others help. It's all knowledge and experience you gain...and you might not be getting all that in your day job today. Such thing, my dear reader, is invaluable. It's what will differentiate yours amongst a pack of resumes. It's what will get you places. Take it from me - a guy who, like you, knew nothing about SQL Server.

**Bio:**

Argenis Fernandez is member of the Database Engineering team at SurveyMonkey based in Redmond, WA.

He has worked with SQL Server for over 15 years and enjoys large SQL Server farms, high-end OLTP databases, managing Windows environments, performance troubleshooting, high availability, disaster recovery, following best practices, and PowerShell scripting - along with security topics.

Argenis is also a Microsoft Certified Master on SQL Server 2008, an avid Twitter user (you can follow him at @DBArgenis), and occasional blogger on SQL Server topics at [SQLBlog.com](http://SQLBlog.com). More recently, Argenis founded the Security Virtual Chapter for PASS, which he currently leads along with fellow MCM Robert Davis (a.k.a. @SQLSoldier).

## Grant Fritchey

### ***Worried? You Should Be. OK Relax***

Getting started as a data professional is an incredibly daunting task. If you're not concerned that you're going to mess stuff up and cause a system to crash and burn, maybe you're in the wrong job.

The amount of information you have to learn is insanely huge, coupled with the fact that you are straddling application development, system administration and business needs, multiplied by the factor that all the apps, all the code and the very server structure on which you're building everything is constantly changing.

Concerned now? Good. Stay that way.

The one piece of advice I want to offer you, is that very *state of concern*. You are in a wonderful and horrifying position. If you're working in the database administration space, you're tasked with protecting the data that the business needs to run. This means you need to worry about backups and database consistency. You need to sweat whether you can run a point-in-time restore operation at 3AM. You have to worry about security on the system and whether or not you have disks laid out correctly. In short, you have stuff to worry about.

If you're working as a database developer, your worries are no less than the DBA. You have to ensure that you've got a system that will collect the data needed by the business and do it in a timely manner. You have to have appropriate data constraints in place, or you will get bad data into your business. You too need to worry.

If you're working in the business intelligence space, you don't have any room to breathe either. You have to ensure that your queries are well written, pulling back the data accurately and with adequate performance. You have to ensure that you're feeding information into the business that allows them to make correct decisions so you can keep collecting your pay check. Worry central.

Now, relax. My real piece of advice to you is *to relax*.

You are launching yourself into a vital position within most companies. You know this. So, take your responsibilities seriously. Take your time. Relax.

There's a saying I read about from the Special Forces (that my cross fit trainer also uses), slow is smooth and smooth is fast. Make sure you do the things you have to do. Slow down. Get them right. Slow is smooth. Then, you'll deploy your backup process once and it will work, the first time. Smooth is fast. You have to be the person that takes a breath before you commit that change to the data structure so that you know you've tested it so it won't break production. You're the one who will run the report twice to validate the outcome before it goes to the decisions makers. You're concerned, but you're relaxed.



**Bio:**

Grant Fritchey has more than twenty years' experience in IT. That time was spent in technical support, development and database administration. Grant currently works as a Product Evangelist at Red Gate Software.

Grant writes articles for publication at SQL Server Central and Simple-Talk. He has published books including, [SQL Server Execution Plans](#) and [SQL Server 2012 Query Performance Tuning](#). Grant has written chapters for [Beginning SQL Server 2012 Administration](#), [SQL Server Team-based Development](#), [SQL Server MVP Deep Dives Volume 2](#) and [Pro SQL Server 2012 Practices](#).

Grant Fritchey was one of the founding officers of the Southern New England SQL Server Users Group.

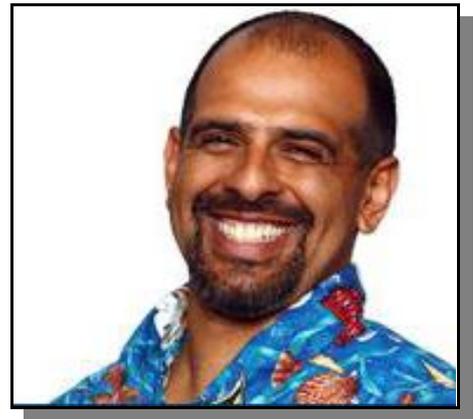
Grant has been awarded as a Microsoft SQL Server MVP since 2009.

## Steve Jones

### *Help Yourself Through Automation*

As I've worked in a variety of companies, both as a database developer and DBA, it seems as though there is one constant at all my positions: there's never enough time.

There's not enough time to get things tested before they must be delivered. There's not enough time to tune all the queries in an application. There's not enough time to audit security. There's often not enough time to get all the work done that your manager has assigned to you. It's true at work, and it's true in life. There's never enough time.



At some point in my career, I realized I wouldn't have enough time and started to invest in automation for my job. I learned how to use VBScript, DOS commands, various command line utilities, and schedulers to accomplish tasks that I didn't want to handle manually. As soon as I realized that I was performing a task on a regular basis, I looked for ways to automate the process.

If I wanted to copy and rename backup files to the network, I whipped up a VBScript with the FileSystemObject and scheduled a SQL Agent job. It ran happily for months, giving me peace of mind that I had a second copy of my backups. When I found it didn't handle database names with space, I corrected the bug in the script and let it run for years.

All sorts of tedious tasks were banished from my day with a little coding effort and the various scheduling options we have with Windows and SQL Server. I still didn't have enough time to get everything done, but I was much, much more productive than I had been in the past. My manager soon realized that I could get more work done than the other DBAs. That didn't help my time and work commitments, but it did help my job security and paycheck.

Before long I realized that there were more and more opportunities for coding. When I found corruption in a database and called Microsoft support, I learned how I could have pro-actively detected the problem much earlier. A day or two and I had another script set up to check for this issue. Over a few years, across a couple of jobs, I built a library of scripts and automated checks that I could easily install on a new SQL Server instance and automate common tasks and monitor for problems.

A few years later I started SQLServerCentral and Andy Warren was one of my partners. We built pieces of the site from scratch, and naturally I added a number of my own monitoring items to the database server. However I learned that I had barely scratched the surface in terms of what automated checks and processes could do. Andy taught me that every potential check we ran for data quality and every potential call to a DBA to query data should be automated. We learned that we could write a short script or batch to look for potential issues in our databases or data and execute them on regular schedules to alert us before problems grew out of control.

Those of us working in technology have the ability to constantly reduce the amount of time that is required for many tasks. Computers automate many tasks, and it's only natural that those of us working with computers take advantage of the capabilities built into our platforms with a little code.

If there's one piece of advice I could give to DBAs that are trying to become better at their jobs it would be that they should learn to automate tasks. Anything that you perform regularly, anything that you might get asked or told to do more than once or twice should be automated. The time you spend learning about a tool like PowerShell or the skill you gain in developing logical programming skills might prove to be much more valuable to your career than anything else you learn about SQL Server.

***Bio:***

Steve has been working with SQL Server since 1991 and been a DBA in a variety of large and small companies and industries.

In 2001 Steve founded SQLServerCentral with two partners and has been publishing technical articles and facilitating discussions among SQL Server professionals ever since.

He currently is the full time editor of SQLServerCentral, owned by Red Gate Software.

## Kevin Kline

I'm wondering if my advice might be a bit different than what the other bloggers will be giving you. For some reason, I've got it in my mind that you'll be getting more technically-oriented counsel from many of them. In my case, I've not only spent many years as a hard-core enterprise DBA, but I've also made the leap into management and leadership in some very large and successful IT organizations. And so I thought I would go "off road" a bit compared to others by providing advice about a more decidedly non-technical approach to career growth for the IT technologist.



You could probably make an argument that I'm providing two distinct pieces of advice. But I believe that these two seemingly incongruent thoughts are in fact like threads which are deeply intertwined:

1. Find and grow a strong relationship with the very best mentor you can, and, to...
2. Spend as much time as possible on deliberate practice of your technical skills.

Now, I'll explain what I mean.

### ***Mighty Mentors Cultivate Powerful Protégés***

In the first thread, I tell you to find and grow a relationship with the best mentor you can find. But what does 'the best mentor' look like? Of course, in an IT career, a good mentor is certainly one with above average technical competency. But with the grow of outstanding, high-quality bloggers, you can read lots of great technical advice on almost a daily basis from the likes of Paul Randal and the team at SQLSkills.com, Brent Ozar and his team, Adam Machanic, Aaron Bertrand, the Microsoft SQLCAT, and many others. That somewhat lessens the need to have a technical mentor just over the cubical wall.

And in our case, the very best mentors are the ones who provide not just technology wisdom, but wise counsel about office politics, the social fabric of the work, and work/life balance. So that means you'll want to identify a friendly person at least a few years more advanced in their career than you who have both a career and personal values that you'd like to emulate. This might potentially be your current boss. But in many work environments, a boss in another department might be a better choice since you might want advice about dealing with your boss, deadlines, or other things that are a little tougher for your line manager to advise on impartially. It might also be a person who's not a boss at all, but someone with real world experiences that are valuable and applicable to your situation in life.

Some companies have formal mentoring programs. That's great. Take advantage of it, if it's available. But here's the tricky part – it's really up to you to plant the seed of a relationship and to cultivate it to fruition. I've encountered many people over the

years who said "I had a great mentor, but we eventually lost touch". Maintaining that relationship falls to you, not the mentor. So make it happen and invest the time and energy into it that it needs to thrive.

One of the things that good mentors do is to help you see your blind spots. In this case, blind spots mean a couple distinct things. Your first blind spot might be in terms of promote-ability and social conduct. For example, we all need someone who likes us enough to say "Better stop bringing the garlic & basil salad in for lunch before the big monthly meeting. No one wants you to open your mouth for the rest of the day". IT pros are notorious for being a bit unrefined in the social sphere and so it's always a significant way to set yourself apart when you get good advice about how to best dress and behave in your current corporate culture. Another significant blind spot many of us have is our broader "brand". Are you known for thoroughness? Do people say good things about you and your work? A strong mentor will help you understand your own brand and how the consequences of your actions, which might not be obvious to you, will play out over the coming days and weeks. Finally, and this is where I bring in the second thread of this advice, your mentor may advise you to burnish your technology skills.

That thought leads me to....

### ***Deliberate Practice Leads to Definitive Improvement***

When you encounter a deficiency in your skills or an area where you're strong and want to get even stronger, research has shown an absolute causal link between mindful, deliberate practice and improved performance. In study after study, researchers find that performance-minded professionals, from musicians to managers to top-dollar athletes, who spend time in focused concentration to improve their skills will achieve prodigious capabilities. It requires many hours and lots of repetition. But it's literally a proven fact that practice makes perfect.

And to further tie this back in with your mentor of things both technical and social, you need to practice the soft skills in a mindful way just as you should be practising your Transact-SQL programming, query turning, and database modelling. Your mentor will help you identify the areas where you need to improve – then go do it, just like Nike says in their commercials. "Just do it".

Too busy fighting fires? Then make time for practice, just as you should make time to interact with your mentor, preferably at the start of your day before you jump in to your regular workday. When you probe many top-calibre IT professionals, you'll find that a surprising large number of them start their day with "quiet time" for study and reflection. So should you. And once you make it such a habit that it becomes part of your daily or weekly routine, you'll find in retrospect that your skills and your career hit a tipping point on the day you got "deliberate" about it.

### ***Sounds simple, doesn't it?***

Many times you get highly technical advice from highly technical people. And that is 100% useful and valuable to advance your career. But when you examine the most successful IT people, you'll usually find that it's neither their skills nor their knowledge set them apart from everyone else in the IT world. Other people in their organization

johnsansom.com

are just as skilled and knowledgeable. It's almost always a set of well-honed non-technical skills combined with a strong technical talent built through an awareness of their own strengths and weaknesses. If you enlist a mentor to help you suss out your strengths and weaknesses and who will help you to successfully navigate office politics and combine it with deliberate, mindful practice in both technology and soft skills, you'll be unstoppable. Now – go forth and conquer.

**Bio:**

Kevin Kline is a well-known database expert and software industry veteran.

A long-time Microsoft SQL Server MVP and noted leader in the IT industry, Kevin is a founder and former president of [PASS](#) and the author of popular IT books like [SQL in a Nutshell](#).

Kevin is a top-rated speaker at industry trade shows worldwide, has a monthly column at [DBTA](#) magazine, and regular webcasts at [SSWUG.Org](#).

He tweets at @kekline and blogs at <http://KevinEKline.com>, as well as at <http://ForITPros.com> for issues of IT leadership.

Kevin is the Director of Engineering Services at SQL Sentry, a leading vendor of database and business intelligence tools.

## Ted Krueger



### ***Igniting Your Unrelenting Desire to Succeed***

Wow, let's think about this question. I mean, really think about it. In a career that some could consider kind of successful, I'm looking back and thinking, "What could be the best piece of advice I could possibly give that would make or break an aspiring DBA?" As the flood of ideas pour in, I tried to set a predicate on the concept of the word advice.

Advice could mean: use this tool because it will help you save a few hundred hours a year. Let's look at [SQL Sentry's Plan Explorer](#). Damn fine product and a company I'm really liking being connected to in some way. This one tool can save you, estimated, a few hundred hours a year. But is that sound advice? Again, advice for someone aspiring or setting aspirations to set a career path cannot be restricted to a tool. Would that tool truly follow them everywhere, be allowed in every installations or really make their career fly above and beyond even mine? I'm not sure.

I digress – to become successful, we cannot rely on a singular form of how we do something. We cannot rely on a singular method or even a path that is set static and not constantly evolving (read - [What makes a successful SQL Server Pro?](#)) What we can provide someone is a sense of confidence, ambition to know they can be successful and most of all, a winning mindset that nothing is over them in accomplishing.

### ***So what is my one piece of advice?***

If you are an aspiring DBA, what I think made my career, the one thing that I found to be invaluable above all: I never gave up.

Becoming a DBA is easy. Yes, I said that and no one can argue with me. Being an everyday, no real value DBA, button pushing, cube hibernating, DBA, is pretty easily achievable. There are TONS of jobs out there. If you put a little effort into learning SQL Server, DB2, Oracle or Sybase, you'll land a DBA job. The problem is, will you be a true success? Hardly. What will make you an absolute success as a DBA, or later a data architect, consultant, data engineer, whatever your flavour, is never giving up. Always seek the best in yourself.

### ***What does that mean?***

Do you know how to create an index? Do you know how to use an INCLUDE in that index? Do you know how to partition tables? Do you know how to use compression? Do you even know how to setup a really snazzy availability groups landscape that spreads the world? OK, now...do you know how all that works? I mean, deep down and into the guts of how it all works?

Retain a mindset in which you will never give in to the fact that it, just works. Determine why it works and how it works. Putting this exhausting energy into finding

johnsansom.com

the inner answers to how SQL Server (and the others) work internally, externally and on the surface, will spell success as you become a DBA.

### ***The end of your career***

I can speak from experience in my own career, although it isn't nearly at the end, but having dear friends that have reached that point, retaining this mindset and ability to always seek the answer will equate to confidence and an abundant mind of knowledge. Knowledge that will make and create the best DBA and career you will ever know you could have had. Trust me, always look for the answer. It's right there! You just need to look for it and always retain the ambition and motivation to know, you will gain nothing but success from finding it. It will spell career success, the best jobs in a rapid career path and the best achievement you can manage to obtain – truly feeling you've been successful.

### ***Bio:***

Ted Krueger is a SQL Server MVP, Data Architect with BlueMetal Architects, and has been working in development and database administration for over 15 years. Specialties range from High Availability and Disaster Recovery setup and testing methods down to custom assembly development for SQL Server Reporting Services.

Ted blogs and is one of the founders of LessThanDot.com technology community. Some of the articles focused on are Backup / Recovery, Security, SSIS and working on SQL Server and using all of the SQL Server features available to create stable and scalable database services.

Recent publications include co-authoring of the Red Gate book, [Troubleshooting SQL Server: A Guide for the Accidental DBA](#) and [SQL Server Deep Dives Volume 2](#)

Speaking Engagements range from SQL Saturday Events to User Groups.

Notable honors and accomplishments:

- SQL Server MVP
- Microsoft SQL Server Certification Exams Author
- Finalist for the DBA of the Year 2010 Awards
- PASS Regional Mentor - North Central Region
- Co-founder and blog on LessThanDot.com
- [www.lessthandot.com](http://www.lessthandot.com)
- Simple-Talk DBA of the Day

## Jennifer McCown

### ***Deliberate Practice is Your Path to Knowledge***

There are a great many thing I tell aspiring DBAs, whether or not I've actually been asked.

"Study."

"Know what you know."

"Remember to keep rebalancing work and life."

"Talk to other DBAs. Talk to your devs. Talk to your users."

"Stop using cursors, darn it."

"A little less cream in the coffee next time."



From my many years in the field, many interviews conducted, many tragedies experienced and averted, many hours spent in extra-vocational activities, I can absolutely distil everything I want you to know down to one simple thing: Test your knowledge.

### ***Certifications***

When I say test, the audience instantly thinks certification, and runs off to Amazon to buy the self-based training kits. Of course you can test yourself by taking a test, no question about it. In fact, we often advise new DBAs to follow a certification training program – either the kit, or a class, or just the topics outlined in the "Skills Measured" section on the exam's web page.

But, understand that studying for a test, and even taking the test, is not truly testing your knowledge. You're learning, and you're testing your ability to pass the exam. Good pursuits, yes, but this is not what I'm talking about.

### ***Interviews***

A good interviewer will absolutely test what you know by asking you a bunch of questions, and staring at you severely as you answer. But again, this isn't really what I'm talking about. You prepare for interviews, good. You get interviewed, also good. But this also falls under learning and passing (or failing) a test.

So, testing your knowledge with interviews and certification exams don't count. What does?

### ***Hands On, Real World, Live-not-Memorex***

Once you learn something, do it. Find out if you really know it. Test your knowledge.

If you can't do it at work, do it at home. I'm talking about everything you learn, from the ground up.

- Heard that the new data type has a range of here to there? Create a variable of that type, and put some data in it.
- Found that you have a need to add nonclustered primary keys to tables at work? Practice doing that, in code. Check out the options available for that statement and play with them.
- Recursive CTEs have a maximum default recursion level of 100? Make a recursive CTE. See what happens when you go more than 100 levels deep. See if there's a way to change that default behaviour.
- Never tried a piecemeal restore before? Do one! Do ten! Read up a little on BOL and try out the different restrictions and rules. Break the thing on your test box. Have a little fun.

Get into types, and new code, and high availability scenarios. Get into anything you're likely to need in your career – not just at your current job, in your career. You can test most of your fundamentals on a single instance of SQL Server Developer on your personal laptop. Many of the more advanced lessons can be accomplished the same way, or with the help of virtual machines.

If you get a complicated enough thing you want to explore, see if your company will spring for a little room in a test environment. Explain that you're expanding your skill set on the advice of a Microsoft Certified Master<sup>1</sup>, to be sure that you're well prepared for the expanding data needs of the company. Companies LOVE self-study, because it's cheap, and because you're demonstrating a dedication to the job. Most will be pretty happy to give you a little space in dev if you show interest. A very few will become paranoid, and fear that you're trying to grow beyond them. Test yourself on your own time, and at least consider making their fears come true.

And remember that at any step of the way, dozens and hundreds of people who have done this before are out there on the web, waiting to answer your tweet, forum question, or blog.

### ***Blogging and...***

Blogs are another excellent way to test your knowledge, in the true "trial by fire" sense. Why on earth is blogging a true knowledge test, and Microsoft exams aren't? Because you don't truly know a thing until you've explained it. Decades upon decades of students bemoaning essay questions, and rejoicing at multiple choice, bears me out on this.

You don't know a thing until you've explained it. Or, to paraphrase Steven King, until you've written something down, how do you know what you think about it?

---

<sup>1</sup> - Go ahead, name drop...they won't recognize the name, but they'll get that they SHOULD be impressed anyway.

Explain how something works, or why it's faster or better or different from another thing in SQL Server. Blogger newbies find very quickly that it is very difficult to explain a concept clearly. You find out even more quickly where the holes in your knowledge are, and that's the key.

What's the difference between char and varchar? Well, varchar is variable length, and char is fixed. Great, now tell me how that's instantiated. If you can't explain the simple facts on the ground in a handful of words, you need to clarify your own thinking. Run an example or two for yourself, clarify it, and then write it down. This is how you build solid knowledge. What's more, it's how you save solid knowledge – any blogger worth his or her salt will have, at some point, looked up a bit of information in their own blog.

Many people take the next step and become community and conference speakers. This is certainly an option, but not a requirement.

### ***Test Your Knowledge***

I run up against gaps in my own knowledge all the time. I do my best to make sure that happens when I'm playing around on my own computer, or at worst on the development environment. I don't want to run into one of those black holes of ignorance when I'm working on a corrupt database with the VP of All the Things breathing down my neck, or when I'm talking with a client about their DR strategy, and most certainly not when I'm interviewing.

As much as you can, test your knowledge. Get your hands on the product. Explain what you've done, if only to yourself. That makes the exams, the interviews, and the job a piece of cake (*well, comparatively*).

### ***Bio:***

Jennifer McCown is a SQL Server Microsoft Certified Master, consultant, and DBA. She owns MidnightSQL Consulting, and is the Senior Editor at MidnightDBA.com, where she creates technology tutorials, articles, and the DBAs@Midnight webshow.

In 2013, Jen formulated and produced the community-written book [Tribal SQL](#).

Jen is a member, volunteer, and speaker for the Professional Association for SQL Server, the North Texas SQL Server User Group, and the PASS Women in Technology virtual chapter. She has presented at numerous SQL Saturdays, conferences, and online training events worldwide.

## Bob Pusateri

### *Break Stuff Like an Expert*

My favourite thought on this topic is **don't be afraid to make mistakes**. Mistakes are one of the ways that we learn. Make a lot of them, and you'll have many opportunities to learn. Really.

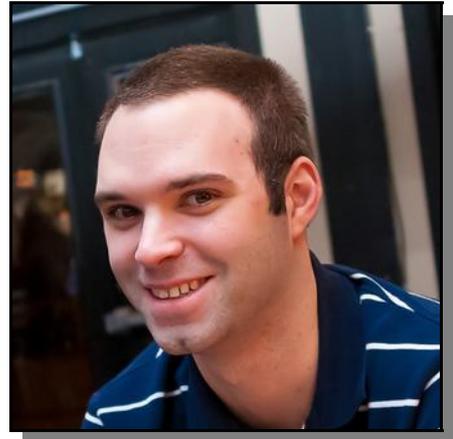
I hear the term "expert" thrown around a lot, occasionally even in contexts I agree with. To me, an expert is someone that's found an incredible number of ways to break things, and has used those experiences to their advantage. They have figured out how to fix everything that they have broken, and even more importantly, they know how to avoid breaking things in the future. Whenever something goes awry, the worst possible outcome is to not learn anything from it. So long as this isn't the case, you can always make at least some good come from a sticky situation.

This is not to say that you should go out making mistakes or breaking things. Creating problems in a production environment is still a very bad idea that could have a negative impact on your career. These mistakes are best made in development or (even better) a local sandbox instance. Practice everything there before making changes in production. Even more importantly, take some time and think about all the different things that can go wrong. If possible, make those situations happen in your DBA environment and then figure out the best way to recover from them.

Along those lines, not being afraid to make mistakes also doesn't mean you shouldn't be prepared for them. Even if they are inconvenient, simple actions such as making sure backups are up-to-date and on hand before launching a change can be the difference between looking like a rockstar for recovering gracefully from an unforeseen issue, and having egg on your face.

#### **Bio:**

Bob Pusateri is a Microsoft Certified Master and DBA in the healthcare data warehouse industry with 7 years experience on SQL Server. His interests involve internals, performance optimization and VLDBs. He is an active member of both Chicago-area chapters of PASS, a community speaker, and maintains a web presence through both twitter (@SQLBob) and his blog ([bobpusateri.com](http://bobpusateri.com)).



## Brent Ozar

### ***Reach Out and Teach***

As you transition into database administration full time, you're going to be working with more people in different departments. The developers, systems administrators, SAN admins, BI team, project managers, and end users will all interact with you because you hold the keys to the data.

It's tempting to think of yourself as an all-knowing magical guardian, and to think of others as morons.

As you gain knowledge, you might become condescending to others. You'll get angry with developers who don't write flawless queries, or project managers who think they can host a multi-terabyte data warehouse on a server with 16GB of memory.

It's also tempting to think that your current co-workers are somehow "different" and that life will be magically better when you leave this brown hole of an office and move somewhere better. It won't.

### ***Your real job is training others, and it never ends.***

Nobody wants to write crappy code. Nobody wakes up in the morning thinking, "How can I suck more at my job?" The reason they don't have all the answers is because you haven't shared the answers with them yet.

The best teachers don't make their students feel stupid. They delight students by revealing the secrets of how the world works, by making complex things seem simple.

### ***Get started without speaking at all.***

Even if you hate public speaking, you have an easy way to teach other people: video lunch and learns. Bring your co-workers into a conference room, get the company to buy food, and play a database training video up on the big screen. Here's some of my favourite sources:

- [SQLbits](#) - great conference in the UK that videotapes all of their sessions and makes them available for free online. Click Speakers at the top left and pick your favourite folks, then check out their topics.
- Vendor webcast archives - including [SQL Sentry TV](#), [Red Gate's SQL in the City](#), and [MSSQLTips](#)
- [Brent Ozar Unlimited's videos](#) - our 30-minute YouTube videos from past weekly webcasts.

Only show videos that you personally watched and liked - after all, you want to be seen as a good content curator and educator.

When the video ends, handle questions. If you know the answer, great - explain it. If



johnsansom.com

you don't, don't fake it - say, "That's a great question. I'm going to write that down and have the answer for you for next week." Then do it. (Trust me, you look dumber if you fake it.)

Next week, explain the answer. If you need slides or a demo to illustrate the concept, do it. Feel free to contact the original video speaker and ask them if they've got any slides illustrating the answer - and of course, explain that you showed their video to your staff. They'll be delighted, and they'll likely offer to help check your explanation, too.

It's easier to start training this way, by just answering questions. Eventually, as you get more comfortable, you can do a lunch-and-learn consisting solely of material you've written yourself. By then, you'll already be comfortable handling questions (which is honestly the toughest thing about presenting). You'll know what delivery methods work well with your own staff - demos, slides, or a mix. You'll know the skill levels of your audience well.

And after a couple of those internal lunch-and-learns, you'll be presenting in public - teaching other developers and database administrators.

### ***Bio:***

Brent Ozar specializes in making SQL Server faster and more reliable.

He's a Microsoft Certified Master of SQL Server and MVP, and he has over a decade of experience.

He co-authored Professional [SQL Server 2008 Internals and Troubleshooting](#), and he loves sharing knowledge at [BrentOzar.com](#) and on [Twitter](#) as @BrentO.

## Chris Shaw

### ***Do You Want a Job or a Career?***

Recently I was asked about the piece of advice I would pass along to a Jr. DBA or someone considering becoming a DBA.

I tried to determine what my one piece of advice would be, the one piece above all others that I would pass along.

I went back and forth thinking on this for weeks. The idea alone that someone would want my advice is humbling. I considered a number of things that I would pass along both technical and non-technical. I considered my personal pet peeves and if I could change the industry how would I change it?

Each consideration I reflected on was based on what I wanted to tell someone and, well, that is what I am doing here. I even thought about what I would say if I was in an elevator at the PASS Summit and I was asked what the one piece of advice would be and I imagined I only had 30 seconds to answer before all of us in the elevator went our separate directions.

Eventually I reached a point in my personal reflection where I started to think about the questions I have been asked in the past and I let that direct my answer. So when I am asked, "What is the one piece of advice I would pass along to new or Jr. DBAs?" I won't list 50 items making it sound like I just can't grasp the whole concept of what that ONE thing is....

### ***How can I become a DBA?***

This must be the number one question I am asked. I have had this discussion so many times I debate if people are serious when they ask me this question. I have tried to encourage and motivate some of them by suggesting books, websites, even blogs and twitter accounts that they should follow, yet only a few have gone on to become Database Administrators. There are times I start to think I have failed my self-appointment as DBA Ambassador. I looked at my approach a few times and adjusted it thinking I must be answering in the wrong way. Now I am not so sure. I don't think I have been fair when communicating how much of a commitment becoming a good DBA is (I have never considered myself to be a quick learner so this may just be my impression).

Granted, this collection of my thoughts is not meant to classify everyone who considers becoming a Database Administrator, but is simply a reflection on the most recent discussion I had during lunch at an event I was speaking at. This discussion happened at a table with a handful of people who, like me, were late to lunch and it was obvious a speaker ran late, yet again (that was me).

So the discussions started with the question, "How can I become a DBA?" with follow up questions along the lines of "What pays more, a Developer or an Admin?", and the question if there is more money in networking.



*I want to pause here and mention these are all valid questions. The difficulty I have with writing this post is the tone of this discussion. The obvious tone that I and others at the table received was that of someone who wasn't all that interested in becoming a DBA because of what we do, but a tone of wanting to be a DBA because of the pay or because some IT magazine said there was room for a lot of growth (again, not a bad reason to choose this field). It would be like determining you wanted to become a church pastor without having the religious belief.*

What advice did I pass along or what advice would I pass along to anyone entering the field? **Decide if this is a job or if it is a career** – If you're looking for a job, then you may just want to keep looking. Being a DBA isn't a job, it is a career...maybe even more than a career. It becomes us. Some of us who are DBAs fell into this field by accident (no one else wanted to do it) while others of us chose to be a DBA and planned our education to become a DBA. No matter how we became a DBA, the DBAs that I know of who have succeeded in what they do, treat being a DBA as way more than just a job.

In my opinion, I would describe the difference between a job and a career as:

- A job is something you do 8 hours a day, five days a week. Sure, sometimes it comes with a little overtime, however, for the most part, the time when you arrive at work and the time you leave work is the same each day. The motivation to go to the job is focused around the money, mostly because that is how one pays the bills. Some may think of a job as a bad thing but it isn't. Jobs can sustain you while you study for another career. Some may not like the idea of becoming so ingrained with their work. Many people like the idea of leaving work at 5:00 PM each day to know they are making their way home or to school, or on their way to a hobby of some sort. There isn't anything wrong with wanting a lifestyle like this. I like to think of a job as a light switch, something that can be turned off and on when the time is right.
- A career is work that you can't put down, somewhat like a good book. There is something there that draws your attention to it, something that makes you say, "You know what? I am going to skip book club today because I want to figure out why this does what it does." (Whatever it may be!) A career will leave little traces of itself scattered throughout your life: for example, I sat down the other night to watch a TV show that I find entertaining but as a commercial came on, I found myself starting to concentrate on a design issue I had been working on. Before I knew it, I had an idea. I started up the laptop and thought about making a quick note so I could get back to the show. I did a couple tests and came to realize that not only had I missed my TV show but the next show as well, not to mention it was WAY past my bedtime!

*Just a few days ago I saw a car accident just outside a local military base. I saw a long-time friend out directing traffic. My friend is over 70 years old and has spent his life working for the safety of the United States in the US Army. The police were already on scene along with the fire department and EMTs. Why was he standing out there even though he is retired? Because this was his career. He knew that by directing traffic he relieved a resource and he knew that resource could then focus on*

*assisting with the rescue efforts or the clean-up. My friend may not have thought about this decision as I have. Why? Because being a first responder isn't his job. It never was his "job". At one time he may have been paid to do it actively, but this day he did it because it is just who he is.*

The tasks of a DBA are difficult to define precisely. Sure, there are things like backups and restores that fit into the DBA category nicely, but what about performance tuning? Should that be done by whoever developed the code? What about building a data flow or the design? Being a DBA is like being a 'Jack of all Trades'. There is so much that can impact the stability of the database, anything that touches the database can impact the primary role of the DBA. Aspects that relate to hardware, operating systems and even networking can impact the core responsibilities of the DBA. Throughout my 15 plus years as a DBA, I have worked on all those aspects and more.

If the idea of long nights, early mornings and solving problems while you sit at the dinner table does not appeal to you, don't worry! It doesn't to too many of us either but we do it. We do it because we are called to do it. There is a lot of satisfaction of fixing what other people can't and doing what other people won't. If all you want is to walk out the door after your shift is over or to turn off the light switch because your work for the day is done, I recommend that you keep looking for work that fits your lifestyle a bit better. Keep in mind also that if you are motivated and driven by recognition, awards or the admiration of everyone at a company, often you will not find those things as a DBA.

If you like the idea of leaving the office at 7:00 AM when the work day is about to begin, knowing that all the faces you see coming in as you walk out of the building are going to have a good work day because you were able to fight through the sleep deprivation, face the errors that caused others to run and hide and that you've muscled through the last 12 hours of your 24 hour shift without eating anything, then being a DBA may be right up your alley. When we as DBAs are doing a good job, no maybe better said as... when we as DBAs are doing a great job, the database performance is fast just like everyone expects it to be. The database is stable just as everyone expects it to be. After whole SAN storage failure the database is back online ready for business... that is what the organization expects it to be. When a DBA is doing a great job we are meeting expectations and all too often many who are not DBAs don't know all the work it takes to meet those expectations.

**Bio:**

Chris started his database career in 1993 as a US Marine. Following the Marines Chris continued working with databases for companies such as Wells Fargo, Pulte Mortgage and Yellow Pages Inc. He later consulted with insurance companies including Anthem Blue Shield and Admini Quest. Chris has been writing and speaking about SQL Servers for over 15 years at events such as SQL Connections, Pass and SSWUG Ultimate conferences. Chris has been a contributing author on 3 books, and has received the Microsoft MVP award.

Chris' blog site at [chrisshaw.wordpress.com](http://chrisshaw.wordpress.com). He can also be reached on twitter at @SQLShaw or by email at [CShaw@Xtivia.com](mailto:CShaw@Xtivia.com)

## Jason Strate

### ***One Piece of Advice for the Aspiring DBA***

When you are an aspiring DBA, there are many things that come with that job role that you'll have to consider as part of the job. You will need to look at disaster recovery, high availability, platform performance, security and a variety of other areas that are critical to the keeping the platform up and running. As you look into all of these different areas, they are tied closely with the tasks you are assigned with whatever environment you work in. In some cases you'll need to know more or less in area or another depending completely on the environment.



Given the variety of tasks and the skills needed for DBAs is there a single piece of advice that can be given to all DBAs that has the same value to all DBAs? Yes, there is. That piece of advice is to track what you work on with daily status reports.

Daily status reports are a list of the activities that you accomplish for each day. These accomplishments are just the work that you did in a day's work and some details about that work. For instances, your accomplishments could be the following:

- Resolved corruption on production database. Issue was tied to storage drivers. Infrastructure team will deploy fix this evening with emergency patch.
- Provided performance tuning assistance to development team. Explained and demonstrated impact of implicit conversions on database performance.
- Reviewed white paper on in-memory OLTP. Technology may provide assistance with current tier 1 application INSERT bottlenecks.
- Meeting with business team to re-design reporting platform. Conversations were focused on style of report headers, no need to include me in discussion. Required meeting / 4 hours.

This list easily provides a list of what was accomplished in the day, but also provides a lot more. Consider the corruption issue, if this shows up on another server or continues to happen after the planned patch, then you have a history of that item to review and consider. With implicit conversions, if this has been an ongoing performance issue, you've shown value in not just fixing performance in production but also in preventing a re-occurrence. Also, by showing the impact of certain technologies, you can show justification to read white papers and use time during the business day to expand what you know about SQL Server. Lastly, including information on meetings attended can help provide a track record of useless meetings that are eating your time and help your manager get you out of those meetings.

The other thing that daily status report should include are tasks that you plan to work on in the following day or week and roadblocks that are preventing work from being accomplished. Forward looking tasks are critical to being productive. If you don't

know what you will need to work on tomorrow, then tomorrow, you will start the day trying to figure out what that list of tasks should be. It is much better to write that list at the end of the day while you are still thinking about what did not get accomplished on that day. Starting the day without a task list too easily allows the morning to slip by without getting anything accomplished.

Likewise, if you don't lay out what your roadblocks are, when they come up, and communicate them, then your lack of progress on a project looks to be an issue with your ability to do work. In a recent project, I had a due date that was months in the future. Provided that a couple of tasks for others was done on time, there would be plenty of time to complete the delivery and finish the project ahead of schedule. Unfortunately, roadblocks from others became an issue early on and continued until well past the delivery date. Continuously reporting on those roadblocks is the only thing that saved me from getting the brunt of the complaints when the project didn't deliver as expected.

When considering reporting on roadblocks, it may sometimes appear that they are just being done to cover your own self in the event of failures. In this case, that is exactly what is going on. It is critical to the success of any DBA to be able to clearly identify why something can't be done. Because your most important project may not be important to the person you are depending on. Reporting on roadblocks helps your manager manage the roadblocks and reach out to reprioritize either your project or the task that you are relying on. Your success depends on how you complete the tasks that you have the ability to complete.

While it seems there's a lot that goes into a daily status report, there effort really doesn't need to take more than 5 minutes a day. Just jot down what you've done, what you plan to do, and what you can't do in an e-mail and shoot it off to your manager. Your manager may not read the status reports every day, but when it matters and when the information is needed, both you and your manager will appreciate that you do this.

There is, of course, one other time when all of this information is useful that hasn't been discussed yet. That is during performance reviews. We all have to go through these and one of the hardest parts is looking back a year and figuring out what we worked on and accomplished. If you have a year's worth of status reports this becomes a fairly easy task. In fact, it should be easy to lay out exactly the value that was provided over the course of a year and provide the basis for your negotiations on salary and benefits for the upcoming year.

Now having discussed daily status reports up to this point, you may be wondering that these really have nothing to do with SQL Server or with things typically associated with being a Database Administrator. That is indeed correct, because this skill is much bigger than that. The skills required to be a DBA are generally fairly easy to learn. Often in any position, you'll get a list of the tasks that are a priority – when to perform backups, indexing, or performance tuning. What you won't always get is the mandate to deliver a status report. Whether you are a database administrator today or a solutions architect in the future, this one skill, and advice, will transcend your career.

To conclude, the most important skill you can learn and the advice that I'll give any DBA, or any other type of professional, is to learn to provide daily status reports. They'll be your insurance when things go wrong and your praise when things are

johnsansom.com

going right. They'll help prevent listless Monday mornings while you figure out what to do and help when you need to renegotiate the priority of your road-blocked tasks. Overall, they'll provide the basis for your career success.

**Bio:**

Jason is a SQL Server database architect with Pragmatic Works. He is also a Microsoft SQL Server MVP and MCM. He has over 15 years of experience developing, architecting, and managing data platforms. He writes, blogs, and presents on SQL Server and has been known to let loose with karaoke and unicorns.

## Michael J. Swart

### ***"Make Sure That You Really Love Doing It"***

John Sansom asked me to give one piece of advice to aspiring DBAs. I spent a lot of time thinking about what would be the best single piece of advice I could offer. Before I could settle on an answer, I came across something written by Robin Williams. I thought it was perfect. So I'm going to hijack his advice and use it to answer John.



Robin Williams was giving this advice to an aspiring actor during a [recent AMA \(ask-me-anything\) on Reddit](#). I like this piece of advice for everyone in general and for actors specifically. I think it's appropriate for actors because I understand show business can be such a fickle industry. It's so important to love acting because the career can be – and will be – tough.

It reminds me of another more local saying. Not every kid can make the NHL. Just as it is in show-biz, it's very difficult to "make it to the top". If you can make it to the NHL, you're one of the fortunate ones. It seems to be such an exclusive vocation.

But the I.T. field is different in an interesting way. I was recently talking to a friend at work. "What are the chances of my daughter becoming Prime Minister. I figure they're about one in thirty million." My friend pointed out that not every Canadian tries to become Prime Minister, or even a politician. So the odds of someone trying to become Prime Minister and succeeding are significantly better. And here's where the I.T. field is different. Anyone who wants to become a DBA becomes a DBA. Anyone who [wants to become a rockstar DBA](#) becomes a rockstar DBA. There's no real secret. Talent helps. Hard work helps more. But mostly it's putting in time. Putting in time is easier said than done. It's putting in time and the commitment that goes with that.

So I believe Robin Williams' advice still applies to you aspiring DBAs. If you love this field it makes the work fascinating. You start finding that problems become puzzles.

johnsansom.com

All of a sudden, you're not studying, you're satisfying curiosity. This field provides a great scope for creativity (for the creative) and great scope for community (for you social creatures).

### **Remember**

Like I mentioned, if you dedicate your career to the pursuit of becoming Prime Minister, your odds of succeeding become much much better than one in thirty million. On the other hand, for those of us who never even try, the odds are zero. So as an aspiring DBA, there will be plenty of opportunities for you to demonstrate that you want this, that you're one of the few that want it badly enough. Learn about Randy Pausch's [Brick Walls](#)

Now maybe you don't love the field. Maybe your DBA job is simply a means to an end. Maybe your DBA job enables you spend time at what you do love. That's fine. It just means that you need to have a strong work-ethic. The time commitment takes a bit more discipline.

You've chosen a great field. It will pay back what you put into it.

### **Bio:**

Michael J. Swart is a passionate database professional and blogger.

He's earned Microsoft's MVP award for SQL Server in 2012 and 2013.

Michael works at Desire2Learn an educational software company designing and maintaining many high performing solutions.

He lives in Waterloo, Ontario, Canada with his wife and two daughters.



## Mike Walsh

### ***The Attitude of the DBA***

*Disclaimer – This is my view. My view may be different than others' views. Where it's different, I advise you to consider all perspectives on this and learn from them all. I base this on 15 years of experience in technology. Well Experienced compared to some, still just a babe compared to others. I am still learning these lessons and growing so I bet we'll both learn from alternate perspectives on this.*



*I'm also basing this on a lot of tough lessons I have learned in my career. Some I'm still learning. So if you once worked with me at, say, the five-year mark of my DBA career, you might find yourself saying "Ha!! You need to practice what you preach!!" To you, I'd say, "Yes. I wish I had some of this under control even back then." I suppose that's why I was excited to share this when John asked me to share an article about thoughts for aspiring DBAs.*

### **Aspiring to be a DBA? Soft Skills First**

I'm a consultant now. But when I was a senior DBA or team lead I was often responsible for being part of the hiring decision. When I hired a strong mid-level or senior DBA, I cared a lot about experience and skills and the more senior the more obscure knowledge I would test on.

When I hired a Junior DBA, however, I cared most about the attitude. I cared more about humility and a desire to grow and learn than I did about knowing if you could walk through the best practices of post installation configuration options and settings.

You see, as a DBA, you are **stuck in the middle**. I may be biased here, but I would say that the old adage of **DBA** standing for **Default Blame Acceptor** is quite true. You see the DBA is wedged in the middle of two worlds:

### **Users**

Your users use applications that rely on databases. They don't know a lot about SANs, networks, or even hardware. They just know that their applications have databases and that the **database is probably the part that isn't working right** when they have performance problems. This may or may not be true, but they outnumber you.

### **The Non-DBA Part of the Infrastructure Team**

So sometimes you report to an infrastructure team. Sometimes you don't. Either way – you are still a bit of an outsider. Because while your users are pushing you for something, the rest of IT is trying to save money, fight other fires and deal with the biggest consumers of resources in their SANs, their virtualization layers, their servers, etc. And, **guess what?** That's normally you and your databases.

So the result here is often you're fighting for the resources you need to deliver the performance and reliability your users want, while trying to help your users understand the implications and costs of their requests.

When it comes to Storage administrators, Server administrators, Network administrators, etc. you will often find folks who are fine talking to users and approachable. But you'll find plenty of pockets of people who prefer to deal with business users through layers of abstraction. Through management, project management, etc. With DBAs, though, I find that every DBA not only needs to stay in close touch with their users, the users actually have grown to expect this. **You have to be a people person.**

And you need to be able to manage the demands of users sometimes asking for demanding things in demanding ways expecting high accountability and many of these things not being your direct responsibility. I find a good DBA can help translate from user to IT speak. The DBA can be an advocate of not only the databases they protect, but of the users who use the apps that drive their businesses.

So when I look to bring someone in the world of DBAdom, I want to make sure they have the soft skills that this standing in the gap between business and technology requires:

- **How do you present yourself and get along?** This matters. DBAs are professional, polite, team oriented people who get along with others. They listen. They can deal with people. I'd rather train an excellent service industry veteran to be a DBA than a well experienced, but obnoxious and arrogant skilled computer science graduate with tons of IT experience.
- **How do you take "disaster mode"?** If you do your job right as a DBA and you automate, test recovery, plan for failure, etc. this happens less. But even when you do all you can do – disaster still strikes. How do you handle chaos? I care about how you act when impatient CIOs demanding updates while trying to coordinate other IT teams to get you the pieces and parts you need (SAN deployments, new machines being build, new VMs provisioned, backup files moved around, etc.) I can teach SQL skills, it's so much harder to teach patience.
- **How do you troubleshoot?** With a shotgun or with a methodology? Poor troubleshooting processes make so many IT disasters worse. Again, I can teach SQL skills faster than I can teach troubleshooting.
- **Are you driven?** What do you do to learn? Do you love what you do? Do you really want to be a DBA? This is a career field that a lot of people enjoy, do you? Do you know what a User Group is? Have you read some blogs? Do you go to SQL Saturdays or SQL Bits? I want to see that you want to learn and that you are a sponge for knowledge.

So when you are preparing to get that DBA job work on some of the soft skills. You can see more about soft skills in this [blog post](#) I wrote and the comments on it.

### ***Confession Time***

So now that some of the basics are taken care of. Hear some of the ways I messed up on soft skills and learn from my mistakes. Maybe this will help you as you journey from Junior DBA to senior DBA.

### ***Hero Syndrome***

I thought that my day job was more important than it was. This doesn't mean that you shouldn't care and strive to grow and learn and use some of your spare time to do that. It does, however, mean that you shouldn't make your life about work. I did this. I felt like I had to prove something to everyone in IT and the users. Like I had to be the "hero". I spent much of my "at home" time working on work projects. I over applied myself and lived off of the occasional praise I'd get from co-workers. I was also quick in getting the "I saved the day" e-mail out..

There is a balance – and you need to start balanced. You need boundaries and guidelines on when to turn work on or off. Now emergencies happen to DBAs – it is a part of the job – especially in your first 5 years or so. You need to, however, understand what a real emergency is and what it isn't. I fell into the trap of marking everything as critical because I wanted to be needed, to be respected and to be seen as the solver of problems. Sad thing here is I was seen as that – at work – but it would backfire (read on) anyway. And the tragic thing here is it made me an absent father, an absent husband. The first years of my oldest two children's lives were a blur. I don't get any of that time back. **Don't do that – find the balance right away when you start.**

I'm still suffering from some of the after effects of that, and still don't balance like I should. I wish someone told me that 15 years ago. Most managers will let you push yourself if you push yourself. In the age of smart phones, 24\*7 coverage and automated alerts, it is easy to see the rest of the team replying to emails 24\*7 and feel this is a requirement and start down the road. It's an easier road to avoid than it is to get off of.

### ***A Little Knowledge & A Lot of Arrogance***

That hero syndrome? It led to growing quickly in my career, but it also grew to this arrogance and general unpleasantness. I fell into this trap of pushing myself incredibly hard, and when it went unnoticed I'd start to get discontent. I'd become a bit arrogant and demanding. It was a pattern I saw at a couple jobs. Here's what I learned **everyone loves and wants to hire/work with a rock star.. Except No one loves or wants to hire/work with a rock star who is also a jerk.** You see, at first the person who knows a lot and brings a ton of knowledge is great. But if you aren't a team player? If you make your manager work harder dealing with the damage in your wake? Forget it. All of the good is outweighed and you eventually become damaged goods there. **You need to be a team player.**

### ***Don't Be Bitter***

Sometimes when you are in hero mode or you have a lot of users or developers hounding you for questions, you can begin to get perhaps a bit bitter. "There is one of me and 80 of you people!" is a bad place to be. But sometimes, for a DBA, it is an easy to find yourself there. Don't let the stress of the job, the stress of management making silly decisions and not taking your advice kill your spirit. Now I don't mean stick with a bad situation forever, I don't mean don't try to help encourage open dialogue and a spirit of positive change. That's all good. What I mean, though, is don't live or die by what happens at work. Don't let the politics make you externally bitter. I personally think that every DBA has to be a leader of sorts. And sometimes leaders are placed in bad situations by the chain-of-command. A good leader doesn't whine up and down the chain here, they adapt to the situation and work at overcoming it. It's okay to have a heart to heart with management with data and metrics and facts.

It isn't okay to send out a reply all with a bitter tone. Or be a jackass of a spokesperson against management in a quarterly meeting. It's tempting. It feels great at the time, you get some high fives afterwards, but you've just secured the need for your next job eventually.

### ***Own Mistakes***

One of my favourite blog posts over the past 5 years of blogging is [this one](#) - Bill Clinton wasn't impeached for... When you make mistakes - and you will - own them. Quickly admit to them, quickly work on a plan to resolve and be open and honest. DBAs have to have integrity; we are entrusted with sensitive and critical data. That means we cannot sweep things under the carpet in cover up attempts. We need to deal with every situation we come against with integrity. **I can work with a DBA who learns from mistakes they've made and admitted. I don't think I can with one who lies or hides their mistakes - it's too risky.**

### ***Get a Hobby***

While there are a lot more tips to write, I'm going to take some of my own advice here and step away from the computer. You need a life outside of work. Family, fun, friends and hobbies - stuff like that.

Our jobs as DBAs are demanding. There is real stress at times. We are entrusted with a lot and often have pressure coming from all directions. Do something totally unrelated to your day job every once in awhile. Go for walks, play with friends, and enjoy wrestling with your kids. This goes back to balance, but through your hobbies and downtime, you recharge yourself for life. You can't do 60-80 hour work weeks a lot of weeks in a row without a lot of things breaking - both in your life and, eventually, in your environment. **Stop and look away, and remember this is only a job** - one phase in your years here on this planet. **Don't take it all so seriously.**

**Bio:**

Mike is a partner and WellDBA™ Practice Lead with Linchpin People. He is an experienced SQL Server professional and has worked with every version of SQL Server since version 6.5.

He has been in the role of DBA, Developer, Architect and Performance Lead but always leans on his DBA and performance tuning experience. He loves providing training and coaching to clients and colleagues and DBAs.

Mike has been recognized as a SQL Server MVP by Microsoft for three years in a row since 2011. He speaks at national and regional conferences and SQL Server events on DBA, Performance Tuning and Professional Development topics. He blogs regularly at [www.straightpathsql.com](http://www.straightpathsql.com) and sometimes at [www.linchpinpeople.com](http://www.linchpinpeople.com).

He lives in rural New Hampshire with his wife, three kids, some pets and a flock of backyard chickens.

## Chris Yates

### ***Two Essential Pieces to Your Success Puzzle***

*"If you could give an aspiring DBA just one piece of advice what would it be?"*

This is a question that has stood the test of time; one that has been asked by many aspiring DBAs on their journey.

Some might say that you need to be centralized and good at one technical aspect such as replication, ETL processing, database mirroring, disaster recovery, or log shipping. All the technical aspects that a DBA must learn and endure along their journey are all prevalent and are a must for success, but over my thirteen year career to date I can visualize taking a new DBA and, if they are adaptable, teach them along the way.

For me I see two very important pieces to the puzzle that many miss, few obtain, and those who do are humble in their practice.

#### ***Hunger***

Most people will look at my title of this section and be thinking, "What in the world is this guy talking about, hunger?". No, I am not speaking of food in a literal sense or being hungry for literal food, but do you have the hunger and the drive to learn?

I've been around long enough to know and understand the persona's of a lot of the all-knowing DBAs; do they exist – sure they do but the more prevalent ones I've noticed something different about and they are the ones who have a hunger or desire to better their skill set day in and day out.

There will be many days on one's journey that the DBA will experience frustration as the DBA will wear many hats along the way; however each challenge that presents itself can be viewed as an opportunity to learn and find a resolution. Write your own learning destiny, success doesn't come handed to you. It is something that you have to want and strive for.

#### ***Be Exceptional***

You may or may not have heard the term "How do I become an exceptional DBA?".

Being exceptional is going above and beyond the desired task at hand. This can have a broad meaning; how are your communication skills with others such as the business teams, development teams, release management teams, QA teams? Or when you are faced with an issue do you follow it through to the end or do you wait for someone else to help? Or is there something else noteworthy that in your field you can look on and gauge where you are at professionally?

Being exceptional is, at times, doing things that no one else wants to do. For example, the places I have worked at in my career I like to come in and first thing I ask my



boss is, "I'd like to have the things that no one else wants to do". Guess what, so far a lot of times it has been documentation on the system and processes which uncovers a plethora of information about the environment.

I have been asked by many people how do I get there? How do I achieve being exceptional? Each individual is different and circumstances surrounding that individual are different.

For me, I grew up playing sports and was heavily involved with team consensus at an early age and into college that has carried over into my career – I've always tried to surround myself with people who have been through the trials before; having a good team. Today, like so many others, I look up to people I consider top in our industry - the Brent Ozar's, Paul Randal's, Glenn Berry's, John Sansom's, Chris Shaw's, John Sterret's, and Adam Machanic's. If you have never gotten involved with the SQL community itself start now. The men and women in our industry share knowledge like no other community and a lot of times it is free. Let me ask you this....have you ever been to a technical blog by any of them, checked out SQL server forums such as SQLServerCentral, or logged onto twitter and followed these people. The wealth of knowledge you can pick up on is phenomenal and some of the free tools they offer so you don't have to re-invent the wheel such as sp\_whoisactive, sp\_blitz, or Glenn Berry's diagnostic queries are outstanding!

Whatever the situation, issue or challenge may be, take a step back and be exceptional. Provide leadership through service, and at times put away our ego and pride and just listen – you will be surprised at what you can pick up on by just being a sponge.

### ***Make a Commitment to Yourself***

I ask this question of you today....what is your hunger level and after taking inventory are you striving to be an exceptional DBA, or are you satisfied with just showing up?

Be hungry, be motivated, and be exceptional. I will not sit here and expect you to believe that every day will be a rose garden when dealing with DBA tasks, but what I can guarantee is that each person has the ability to make a difference and impact in their respective environments. It only takes one – will you be that one?

### ***Bio:***

Chris Yates is a Database Administration Manager with over thirteen years of experience in the SQL industry. His experience includes design and implementation of both OLTP and OLAP solutions as well as assessment and implementation of SQL Server environments for best practices, performance, and high availability solutions accompanied by a strong development background.

He is actively involved with the local PASS chapter (SQL Server User Group) along with Regional PASS organizations. Chris enjoys helping others in the SQL Server community and does this by contributing on several SQL forums creating [The SQL Corner](#), and speaking at several SQL functions. His passion and focus is not only with technology but also helping others along their way and career path.

## **ABOUT THE AUTHOR**

## John Sansom

John Sansom is a Microsoft Certified Master (MCM) of SQL Server and author of the popular SQL Brit blog community.

As the Technology Operations Manager of the data services team at Expedia, Inc, John Sansom is wholly responsible for one of the largest SQL Server estates in operation. His primary focus is to ensure the service availability of Expedia's data technology platforms, whilst accommodating hot deployments without compromise to performance.



Specialising in solution architecture and performance-tuning, John provides SQL Server consulting expertise to Expedia's global business teams, as well as independently through his own [company](#).

Awarded the Microsoft Community Contributor(MCC) award, he is a prolific blogger and can be found regularly writing about SQL Server and Professional Development over at [www.johnsansom.com](http://www.johnsansom.com). As a tool to further inspire excellence and encourage the development of others, John created the [SQL Community Forum](#), an entirely free community for ambitious Data Professionals to collaborate and share knowledge. Having only just recently launched, the community has already amassed over 1000 posts of insight and experience.

He also has a lot of cool friends that gave their expertise for this little ebook, and for that, he is forever grateful to them.

Always delighted to talk technology, you can reach John via email ([john@johnsansom.com](mailto:john@johnsansom.com)) and connect on [LinkedIn](#).

**Microsoft**  
**CERTIFIED**  
Master

---

SQL Server® 2008

**Share This Ebook!**

