



BRENT OZAR
UNLIMITED

How to Develop Your DBA Career

We love to help database administrators improve their careers.

We want to help you on your path to ever greater awesomeness. In this free, handy eBook we've collected some of our best writing on how to get a DBA job, grow your skills, and ~~slay zombies~~ get promoted.

Why it's Hard to Get a Junior DBA Job	2
How to Get a Foot in the Door	5
Questions to Help You Land a DBA Job	7
How to Get DBA Training on the Job	10
How to Revive Your Stagnant DBA Job.....	13
How to Get Promoted	16
Get More Help from Us	19



Why it's So Freakin' Hard to Get a Junior DBA Job

Companies Aren't Hiring Junior DBAs. Why?

Understand how companies think about Junior DBAs and find out who you're up against. (Hint: not zombies.)



by [Brent Ozar](#)

So you're hearing all the [fun that DBAs are having on Twitter](#) and around the intertubes, and you're wondering how to get in on the fun. Maybe you're a developer or a network

admin, and you've been learning about SQL Server from books and from peers, and you're starting to look around for your first Junior DBA job.

WHERE ARE JUNIOR DBA JOB ADS?

Search for the terms "junior dba" or "senior dba" on national job sites, and the results are dramatically different:

- [CareerBuilder](#) – 296 senior, 27 junior
- [Dice](#) – 65 senior, 2 junior
- [HotJobs](#) – 170 senior, 19 junior
- [Monster](#) – 122 senior, 15 junior

To make matters worse, drill into the requirements for some of the so-called junior positions and they ask for:

- "2+ years of hands-on experience administering customer-facing databases"
- "Minimum 2 years of information technology experience is required"

- "2 years of database administrations experience including installation and configuration"

These ads make me chuckle because a DBA with two years of hands-on experience isn't looking for his next junior DBA job – he's aiming for his first senior DBA job. People don't aspire to make lateral moves – they aspire to move up the ladder.

It's not just you, and you're not missing some gold mine of hidden junior DBA job ads. Those positions just aren't advertised.

WHY AREN'T MORE COMPANIES HIRING JUNIOR DBAS?

Databases are among a company's most valuable assets. They hold sales data, payroll data, information about customers and competitors, and all kinds of proprietary data. Companies guard this data closely, and they don't want just anybody stepping up to the server to handle backups, restores, and security.

Companies may be willing to hire junior developers instead – for some reason, they don't have a problem giving a fresh-out-of-college person access to all the data. But when it comes time to actually managing the data, they want someone experienced.

As a result, even when a small company with a handful of SQL Server instances goes shopping for a DBA, they want



someone experienced. They want to hear from the candidate, “Yes, I can do what you need – I’ve already done this for a much larger shop, and I’ve got experience handling this sort of thing.” They sleep better at night knowing their DBA is not stressed out trying to figure out how to automate backups for the first time.

Every now and then, big companies will actually hire junior DBAs. This mostly happens when they’ve already got a crew of senior DBAs, but here’s the odd part: they don’t have anyone internally that wants to get promoted into the DBA team. That’s very peculiar, because in companies large enough to have DBAs already, they often have people in other teams who want to become DBAs.

KNOW YOUR COMPETITION: OTHER JUNIOR DBA CANDIDATES

The very few available junior DBA positions attract these candidates:

- **Developers** with a year or more of SQL Server programming experience who’ve decided they want to focus on SQL Server instead of development. They’ve written code that stores data in SQL Server, so they know the basics of T-SQL or LINQ or Entity Framework, understand the basics of data models, and maybe they’ve even done a little SQL Server administration on their dev boxes.
- **Windows administrators** with a year or more of Windows experience who also want to switch their focus. These people know hardware, know the basics of performance tuning, have backup/restore experience, and maybe they’ve even built a cluster or two. They might have even built SQL Servers, but they haven’t been working on SQL Server full time.
- **SQL Server DBAs** who’ve been let go, and they’re desperate. They might have worked for a company in financial trouble,

they might have been drinking on the job, who knows. These are your most dangerous competitors, because at first glance, their resumes kick sand on your resume. (Although sometimes the experience on these resumes are completely fake - but those people will get busted during the interviews.)

- **College or vocational school grads** with no production hands-on experience.

If you’re in that last category, I gotta be honest: you’re screwed. Go get a job as a junior developer or a junior Windows admin first, and then work your way into database administration. There is no classroom training that’s going to convince a company to hand over the keys to their data on your first day in the office.



Bootcamps and certification programs alone won’t get you that first job— not when your competition has experience.

You may see ads for certification programs that promise to make you DBA-ready within a week for a few thousand dollars. It won’t get you the job – at least, not when you compare yourself to the competition. After you’ve gotten started in a development or sysadmin position working around SQL Server, come back here and continue reading about how to take it to the next level.

GETTING PROMOTED TO DBA IS EASIER THAN GETTING HIRED AS ONE

If you’re a developer or Windows admin, try to get promoted internally rather than switching companies. You’ve already built



up a level of trust with your coworkers. You've already put your hands on the database server, and no data has been harmed. (Well, there was that one time, but we'll pretend people forgot about that.)

Make yourself the natural candidate by actively going after the position before the position even exists: volunteer to spend time with the DBAs, help them during scheduled outages, or just ask if you can sit around and watch during the weekend maintenance windows. Bring coffee, donuts and bacon, and be the most pleasant (but not over-the-top) person in the room.

If you've already got a job in close proximity to SQL Server, this is one of the times where getting certified in SQL Server without getting the job first does make sense. I've blogged about how [certification is the icing on the cake](#): don't get certified without at least some vague experience in SQL. Developers and Windows admins will gain knowledge from the certification training process, and it'll buy them a little bit of

credibility in the eyes of the DBA team. Not a lot – but some.

YOU'RE PROBABLY LOOKING FOR THE WRONG JOB TITLE.

Get your foot in the door as a developer or sysadmin, but if you've already got that experience, maybe you're not aiming for junior DBA. Maybe you should aim higher - and we'll discuss that next.



How to Get a Foot in the Door

Companies may not be hiring Junior DBAs in droves, but you can still get yourself that job!

by [Brent Ozar](#)



COMPANIES ARE CHEAP, DBAS ARE EXPENSIVE

You like buying stuff on sale, right? Companies do too. Companies want to

pay less and get more.

They want to hire an experienced senior database administrator for junior DBA wages. They think they've got something special – a great work environment, flexible working hours, nice plants in the lobby – and that it offsets the lower wages. It doesn't: good senior DBAs get good money, and have their pick of companies.

Whenever you see a job ad for a senior DBA (or just DBA), I bet the company has been hiring for that position for a while, and they've been unable to find the right candidate at the price they want to pay. They're getting frustrated because they wrote a big, long job description with all kinds of dreamy requirements, and now they're bummed out because Mr. Right isn't showing up on a white horse to save the day.

That's why I wouldn't hold back - throw your hat in the ring for local Senior DBA jobs. It's not career suicide: it's a case of the company asking for something unrealistic. They may not get the candidates they want for the price they want to pay, and that's your chance to get your foot in the door. Don't exaggerate your reputation, of course – be honest about your skill level and your

experience, but at the same time, don't sell yourself short.

YOU'RE DOING SQL SERVER WORK EVEN IF YOUR JOB TITLE ISN'T "DBA"

I got my start as a developer and as a network admin ([here's the story](#)). When I went to look for my first pure SQL Server job, I didn't have much on my resume and I

Training and mentoring turns Junior DBAs into Senior DBAs.



didn't really think I was all that qualified. If anything, I underestimated the bejeezus out of what I put on the resume.

When I started hiring other DBAs, though, I remembered my own experience. As a result, when I interviewed DBA candidates, I had a checklist of skills that I'd ask them:

- Have you ever had to restore a single table's contents?
- Have you set up log shipping, or done troubleshooting on it?
- Have you ever built a server connected to a SAN?

Go pick up a [SQL Server administration book](#), look at the table of contents, and check off everything that you've actually done. Even if you've only done it a few times, put it on your resume and explain that you've dabbled in it, because it'll give you a big edge over the other candidates. Don't



say that you're an expert on the topic, by any means, but the fact that you've done it is a plus.

More often than not, I'd hear candidates answer, "Well, yeah, but hasn't everybody done that?" Actually, no – some candidates haven't. Every single skill that you performed in production – not in theory – is another reason why you might get the job. Even if you've only done it once a quarter for a year, that means something.

HOW LONG HAVE YOU BEEN DOING IT?

Did your boss ask you to start backing up a SQL Server a year ago? Last year, did you start restoring the production database onto your desktop for development testing? Did you start working on making stored procedures a year ago?

You have a year of experience.

I can almost hear the angry emails coming in now from really senior DBAs who do this stuff full time, nonstop, for a living, but they've forgotten how junior-level experience works. People don't get handed

the keys to the enterprise on Day 1 and start some kind of master clock. Experience happens gradually, almost imperceptibly. There's no knightin ceremony where the CIO taps you on both shoulders with a laser pointer.

This is why so many junior-level DBA positions ask for a year or two of experience: they're expecting to hear from developers and sysadmins who've been dabbling with database tasks over time, getting their feet wet. I don't want to hire somebody who's never seen SQL Server Management Studio: I want to hire a developer who installed SSMS a year ago and has been dabbling with it ever since. He may not like going in there – it may scare the pants off him – but as long as he's been going in there grudgingly and tapping his terrified fingers on the keyboard to get his job done, then that's a plus in my book, because I'll train him the rest of the way. DBA training never ends.

Training and mentoring is the way junior DBAs become senior DBAs.

**Need to practice
interviewing for
DBA jobs?**

brentozar.com/go/hire



Questions to Help You Land a Dream DBA Job

Seven questions you MUST ask. (Bonus: they prove you're human.)



by [Kendra Little](#)

Looking back over all my years of interviewing and being interviewed, I realize that there are simple secrets to getting a great job offer.

The big secret is just this:
ask great questions.

INTERVIEWING GOES BOTH WAYS

Asking thoughtful questions in your interview process, from screening to salary negotiations, makes you more powerful. It shows you're engaged, informed, and in control of your career.

A candidate who asks good questions automatically demonstrates that they're selective, and they have choices.

But what do you ask? Here's a checklist to go through before each interview to develop questions and remind yourself what to emphasize by your questions. By all means, write down your questions and bring them.

1. BE THE CANDIDATE WHO LOVES TO LEARN

Ask at least one question about technology you're unfamiliar with. Find out what technologies are in place at your potential job before an in-person interview and do an hour of research on those you don't specialize in. When it comes up in the interview make it clear that you were

inspired by the interview process to start learning.

Don't pretend to be an expert in fields you don't know about. Do be honest about your interests and show you have initiative. Ask questions about challenges they've hit and what informed their design and implementation decisions.

Before your interview, make notes on times when you faced a difficult task, learned something new, and were able to improve something. You'll likely be asked questions



Interviewing is like swimming. At first you have no idea how to do it, but you get thrown in the water and you learn to survive. If you never practice, you'll splash around helplessly.

about your experiences when you can tell your stories.

2. ASK AT LEAST TWO SMART TECHNOLOGY QUESTIONS

Most candidates just ask basic questions about what versions of software are running. This is a great chance to set yourself apart.

What are their pain points? This is something we care a lot about at Brent Ozar



Unlimited™, but we don't mind sharing our mojo with you for your interview. This is the most interesting question you can ask: tell me where it hurts in your technology. You'll want to get lots of detail on this. Make notes and follow up on different points they cover. Ask this question of different people throughout a day of interviews and compare the responses. I promise it'll be interesting.

Are they leveraging their strengths? When you're familiar with the products they're using, think about the strengths of that product. Ask a question in a way that shows your knowledge. If you know that the JurgenPlufen can provide high availability when clustered, then ask if they're doing that. If they aren't, ask why— the reason will let you know a lot about their business. Keep your tone curious, not judgmental.

What major changes have they made? Over the history of the company, are there any revolutions in technology they can tell you about? How did they handle that change, and what would they do differently now?

Are they open to other technologies? This matters more to some people than others, but it's an important thing to know. Is this a company that looks for the right tool to solve an individual problem, or do they prefer to standardize to narrow the scope of support? There are pros and cons to both ways, but you need to know which way they go in order to see how you fit in.

Have they tried the New Hotness? If they describe a problem and you think there's a fairly obvious newer product they could buy or upgrade they could make to support that, ask about it. You want to find out why they haven't gone there— are they slow to adopt? Short on people? Short on budget for new technologies?

Good questions show you're not only a skilled technician. You're an engineer who

takes the initiative to find out how to improve an environment.

3. ASK A QUESTION ABOUT PROCESS

You want to know what processes are in place at a prospective employer. You also want to show that you're responsible.

As a candidate for DBA positions I made it a habit to ask during phone screens, "Can you tell me a bit about your change management processes?" More often than not, IT hiring managers were thrilled that I'd asked. When I asked follow up questions it was clearly hard for them to sit still.

Focus on exploratory questions— don't be critical. For a later interview, you should think about scenarios you might be in and create a hypothetical question.

"What if we decided to change the Yak so that it had air conditioning? Can you walk me through what the process of making that change would be like here?"

Be ready in case the question is turned back around to ask what YOU would recommend — that should be a home run.

4. ASK ABOUT THE BUSINESS

Before you ever talk to the company look for technical blogs, or any blogs written by employees. These are a great source for information about how things work at the company, and also a great source for questions.

Always check out recent news articles for the company as well. Care about the industry. Make sure you have a few good questions about that market and where the company is going— if you make it to higher level interviews with executives, these will be particularly useful. If you don't have much experience in that industry, it's perfectly fine to ask questions like,



“I haven’t worked for a dairy but I’ve read there are three major players. As a smaller company, how do you position yourselves in the milking industry?”

Your overall goal is to show you’re not just a technologist, you’re a potential invested employee.

5. ASK A QUESTION THAT SHOWS YOU LISTEN

On a full day of interviews, you will be able to take information you get in one interview and use it to ask questions of other people. This is one of the great reasons to take notes in your interviews.

Be careful that your questions don’t seem to pit people against each other, or slight the previous person you talked to. You want to ask questions more like,

“Harriet described the asynchronous processing she designed for the Femisphere system. Can you tell me how that works from an operational perspective?”

This shows you listened to Harriet and absorbed some of the concepts she discussed. Not everyone can do that! If you have follow up questions ready because you understand some of the operational challenges in the area, you’re in the catbird seat.

6. ASK A QUESTION THAT SHOWS YOUR STRENGTHS

Sometimes people ask what your greatest strengths are, sometimes they don’t. Figure out your greatest strengths for that position, but do it so you can ask the right questions.

Let me get one thing straight: this isn’t BS. You want to pick a few real things that set you apart and make you satisfied at work.

Here’s some example questions:

“I like to identify big changes and drive them to completion in an active

production environment. Is this a place where that’s encouraged? What are the barriers to large changes? What support is available?” Big changes work much better in some companies than others, and you want to know if you’ll always be holding your horses, or if you can make things happen. (If you ask this question, have two stories of how you’ve done this in the past ready.)

“I really enjoy specializing in certain areas of NERDERY and diving deep. This means I like to take four hours a week to do research. I can document what I learn and present it to the team. Would people be interested in that?”

“I am a generalist and I really enjoy reaching out to other teams and working out how components integrate. For example, would I have access to the configuration of the Gigabiggers and is their team open to having me sit with them once a week to learn what they do?”

A question like this shows self-awareness, and it will tell you a lot about whether you’re going to sink, swim, or run for shore.

PRACTICE, PRACTICE, PRACTICE

Interviewing is like swimming. At first you have no idea how to do it, but you get thrown in the water and you learn it to survive. If you never practice then you splash around.

If you haven’t interviewed in a while or if you are changing industries and are not sure how you’ll do, ask for help. Get connected with bloggers or people in the industry on Twitter. Go to a user group. Ask people if they would be willing to do a 45 minute practice interview with you as the candidate. Listen to their feedback. When you practice, make sure you’re asking them questions as well as responding to questions.

Don’t be afraid to go on multiple job interviews. The downside is that interviewing is hard work, and it’s unpaid. The upside is

How to Get SQL Server DBA Training on the Job

What you should – and shouldn't – expect in the way of training from a new employer, and how that affects your asking price.



by [Brent Ozar](#)

ASK ABOUT MENTORING AND TRAINING

During the interview, ask how many other database administrators are on the team and what their seniority levels are. These are the people you'll be learning from – your future mentors.

If you're joining a team of three people who've been DBAs since SQL Server 7.0, you're in good hands. If there's one other person who was a developer until about a year ago, things are going to be tougher because he'll have less time to mentor you and less skills to pass on.

If there's no other DBAs at all, you're screwed – you're going to learn things the hard way. NO, I don't mean by breaking things - that's the stupid way.

THE HARD (BUT NOT STUPID) WAY: JOINING THE SQL SERVER COMMUNITY

You should be willing to take a lower salary if you find a company with a strong, friendly group of DBAs and a management team that's committed to ongoing training. Consider it an investment in your career. Finding out how committed they are is simple: have you seen their DBAs at the local user group meeting? If so, then they're interested in furthering their career and helping out with the careers of others. If you've never seen their DBAs at a [local](#)

[PASS Chapter meeting](#), then their lack of interest in the community might predict a lack of interest in training you, too.

What's that, you say? You haven't been to a local PASS meeting yet? Then maybe YOU aren't too interested in your own training, either. This is why the PASS chapter question is one of [my Top 10 Interview Questions to Ask Senior DBAs](#). Plus, the more often you go to PASS meetings, the more the other database administrators will see you there, and if they recognize you during your interview meeting, that's bonus points. Imagine the post-interview conversation after you leave: "I remember seeing that guy at the SQL Server user group chapter. I don't remember the other candidates, come to think of it." Cha-ching!

The first year or two of a DBA job is very challenging. You're going to want help.



Asking about the company's training budget during the interview is a tougher one: it's hard to ask this question without making it sound greedy. This one is a judgement call. There's only one case where the training costs are considered a built-in part of your hiring, and that's if you go to work for a consulting company.

GOING TO WORK AS A DBA-IN-TRAINING

Some consulting companies will take you on at a lower pay rate in exchange for teaching you DBA skills while you're on the job. They have a large talent pool to draw from (compared to individual companies) and they can put you onsite with a senior person. The consulting company makes money off you, because they're paying you a far lower rate than they're billing the client. You get to learn from the senior person, and you take over as many tasks from them as possible.

The consulting company may require you to sign a contract stating you'll work for them for a certain amount of time, and if you leave any earlier, you'll be responsible for paying training costs. I have really, really bad vibes about this setup, because you're basically becoming an indentured servant. The consulting company can treat you like dirt, and here's the funniest part to me: they don't really care whether you get trained or not. They're making money off you every single day that you show up for work.

If you take this approach, here's a few things to keep in mind:

- **Don't sign an agreement** – frankly, every job involves learning on the job, and the company is making money off you every day anyway. I would love to get the chance to ask one of these interviewers, "So, did you learn anything on the job this month? Did you have to pay anybody for that privilege?"
- **Get the training costs prorated** – if you have to sign a two-year agreement and you quit after one year, then you should only be responsible for 50% of the training costs. If they tell you that the bulk of the training occurs in the first year, then get the agreement to only span one year.
- **Be very wary of non-compete agreements** – some consulting

companies have non-compete agreements saying you can't work for any of their customers for X years after you quit. Even worse, some say you can't work for any PROSPECTIVE customer – which basically means any company in their market. As part of the interview process, ask for a copy of any agreements that you'll need to sign. Otherwise, if you only get a copy of this agreement after you've already quit your current job, you're screwed.

- **Work hourly, not salary** – the consulting company is making money off you by the hour, and they will work you as long and as hard as possible. Your salary needs to be a win/win for both you and the



The last thing you want is to get six months into your new junior DBA position, find out that all your coworkers are making twice as much as you are, and feel like you got screwed.

consulting company.

DON'T REGRET YOUR ASKING PRICE

No matter who you work for, salary negotiations suck, and I don't have any good tips for how to pick your rate. However, I do have advice about what to do after you've picked your rate: write it down somewhere and tell yourself, "I am going to be happy if I get a junior DBA job for \$X/year." Put that piece of paper somewhere safe. Later, when you find out how much your coworkers or your friends are making, don't get mad: get



that piece of paper out and think back to what you were feeling like when you wrote those words down. You wanted a DBA position so bad, and you couldn't figure out how to get in. In order to get your foot in the door, you took a salary that made sense at the time.

I've done this myself in my career. I've taken positions where I knew I was wildly underpaid, but the free SQL Server training on the job more than made up for the pay cut. I knew I'd be making it up in the long term, and today, I'm typing these words from the comfort of my mansion overlooking the French Riviera while Woodhouse, my servant, feeds me grapes. (Not really. Woodhouse is cleaning the pool.)

The last thing you want is to get six months into your new junior DBA position, find out that all your coworkers are making twice as much as you are, and feel like you got screwed.

HMM – THIS IS A LOT OF SCREWING

More than once in these posts, I've said you can get screwed. This emphasizes a point I made in the first article: it's easier to get promoted than to get hired. The money's usually not as good, but if you're just doing this for money, you're in the wrong profession.


The first year or two of database administration is very challenging: you're suddenly in charge of one of the company's most expensive and risk-prone assets. There's a lot of after-hours maintenance work, and when your cell phone rings, you gotta answer it anytime, anywhere. It can be scary taking over this position. The less risks you take, the easier your transition is, and that's why getting promoted eases your transition path.

I know a lot - a LOT - of successful database administrators. The one thing they all have in

common is a passionate curiosity about databases. They're not doing the DBA job in order to cash a check - they're doing it because they love learning about how data works. If you don't have that passionate curiosity, you're just going to get frustrated by the on-call rotation.

Here's how to know if you're going to be a good DBA: you get excited at the chance to learn more about SQL Server in your spare time.

And if that's you, then hi! You're one of us now. We're all strange like that too.



Ready for face-to-face training with the Brent Ozar Unlimited Team?

brentozar.com/go/class

How to Revive Your Stagnant DBA Job

Unhappy at work? Change it.



by Kendra Little

Even when you love being a DBA, working the job can turn into a grind. There's many times when you're up all night troubleshooting just to face another day at the office handling user complaints. You hear at the water cooler that you'll need to improve performance with less hardware for the next year. You get an email that you need to be ready to virtualize everything in the next quarter, but nobody's even sure which hypervisor you'll use.

Your perfmon counters show that response times from storage are getting slower, but the SAN administrator says that everything looks better than ever. One of your servers starts blue-screening periodically. Your manager announces at the weekly team meeting that you need to stop being so reactive and start being more pro-active. Right after that, you get an alert for data corruption.

Oh, and by the way: ***we're going to need you to come in on Saturday.***

Before you know it, you're on the road to burnout. Your tour looks like this:

PHASE 1: DENIAL

It's hard to admit when you lose that loving feeling with your job. You just keep plugging away at it.

When your problems have to do either with boredom or a very high volume of reactive work, this just makes the situation worse. You're spending a lot of time, but you're mostly spinning your wheels. You're not learning or growing, you're just doing what you need to basically keep things together.

Don't envy someone with acceptance. Acceptance means that you keep disliking your job, but you've given up on improving it and rising in your career.



PHASE 2: ANGER

After a while, you get cranky. It's pretty inevitable when you're overworked in a DBA job, because you're surrounded by unhappy people. Users need data faster. Developers and vendors say the problem isn't their code, it's your servers and maintenance. Your manager isn't sure why you can't keep problems from happening in the first place. The SAN Admin gets a little tired of you stopping by with another stinky performance issue again and again.

Also, at this point people in your personal life start to get impacted as well. You're



starting to get less sleep and grumble more. You vent about politics at work and things you can't change.

You get to the point where you say things that seem perfectly reasonable in your head, but come out sounding like you're a real jerk. Sadly, you probably don't notice it.

PHASE 3: BARGAINING

Eventually you get tired of being angry. You start to think that maybe the problem isn't your job—the problem is that you're stuck in your job! So you decide to go about some self improvement.

At this point you decide to become *amazing*. You're going to learn every feature in SQL Server. You're going to know all the internals and when there's a bump in the night, you'll be the one who whips out a debugger, walks the SQL Server stack, and sends a diagnosis to the SQL Server product team. You make a deal with yourself: I'm going to read these twenty books, and then I'm going to fix my job.

Sometimes you try to make the deal with your boss. "If you send me to this training then I'll revolutionize our team."

But it's hard to revolutionize your job when you've been unhappy with your job for a



***If you're in a rut,
trust us, you're
NOT the only one who
feels it.***

good long time. You try to make big sweeping changes, but it's really hard to revolutionize processes and tasks when you've already got a full load of work.

PHASE 4: DEPRESSION

Once you get to the point where you've tried to make big changes and haven't had an impact, you get tired. You get depressed. You get so that you don't even want to finish your

PHASE 5: ACCEPTANCE

Some people reach acceptance with a job they've come to dislike, but some people never get here—they quit first.

Acceptance isn't a happy place. Don't envy someone with acceptance. Acceptance means that you keep disliking your job, but you've given up on improving it and growing in your career. The one good thing about acceptance is that you figure out some change to make it livable: you keep your hours as close to 40 a week as possible. You do just a bit over than the bare minimum at work. You focus on other things. You distance yourself a bit from it all so that it's not so much of a disappointment.

In many ways, acceptance is even more depressing than depression.

PHASE 6: ACTION

I've got some good news. When your job gets bad, you do go through something like the phases of grief. That's no fun. But the good news is that *nobody is actually dead*. And you're not dead yet, either. There are still SO MANY things you can do!

Even when you've been down a long road of trouble in a job, you can still make real changes and make a big difference at your company. You just need to get a fresh way of looking at it, and you need to make small, strategic changes. Your mistake earlier was in trying to make grand, sweeping changes. Most of the time you just can't win the war that easily.



Here are four successful strategies that I have used to make long-lasting changes at work:

- **Add a 20 minute brain workout to your day.** Block out 20 minutes for your personal learning in your calendar and stick to it. Make this your workout period for your brain, and don't let anything come between you and your learning. Especially if you're in the "acceptance" zone, this one regular dose of learning can radically change your mindset.
- **Smile at people.** Seriously. Make your face smile—it will actually change your mood, and in turn change your working relationships. If you're in a rut, trust us, you're NOT the only one who feels it. This is harder than it sounds like, and you may need to set a daily task or reminder to get good at it.
- **Break passive-aggressive patterns.** Some of the hardest burnout problems to solve are poisonous social situations. For hard social problems, find a neutral colleague and ask them for advice on how you can smooth over a bad relationship—but make sure you don't gossip or blame the other person in the process.
- **Bring in a technical influencer.** Sometimes clients bring us into short-term gigs to help confirm where the problem is in their system and provide independent validation of where long term investments should be placed. When an environment is full of finger-pointing, as a consultant we have an advantage: we can elegantly point toward a solution without getting embroiled in the political mess.

YOUR JOB DOESN'T HAVE TO STINK: AND YOU DON'T HAVE TO LEAVE IT

It's a great time to be a database administrator. Technology is growing fast and people are pushing the limits of what

they can do with SQL Server all the time. There are lots of jobs out there, but although you've been struggling, you may already have a great job.

If your team is stuck in a rut and you can figure out what you can do to make that job better, you've just become the most valuable employee on the market: the one who can turn around a bad situation. The secret to getting there is to figuring out the small ch

**You need to
figure out how to
help an ailing
SQL Server. Our
Critical Care™
helps.**

brentozar.com/go/help



How to Get Promoted

Change a few well-placed things that make a difference



by Kendra Little

There's just a few things you really need to do to set yourself apart and pave the way to a promotion. I worked

this out last week when I did some spring cleaning.

THINGS I LEARNED FROM CLEANING THE BATHROOM

I noticed something when I cleaned the upstairs bathroom: I'd put in a lot of work and scrubbed almost everything. It looked clean enough. Then I cleaned the chrome circle around the bathroom water faucet and the faucet itself.

I stepped back and the bathroom no longer looked clean enough. It looked *awesome*. There's something about the shine on chrome and a large clean faucet. The floor could have been a bit dirty, and that bathroom would still have looked awesome.

Don't promise everyone everything. What you're looking for is just a FEW good places to focus.



LIFE IS LIKE THIS

This is how the world goes— you can clean the toilet and the floor again and again and it'll look OK. Those basics are needed, but they don't make something feel fantastic to whomever uses the bathroom.

It's **a few well placed smart things** in the midst of a **good-enough environment** that make a user feel like they're using something really special.

Let's break this down:

CHANGE JUST A FEW...

You don't have to make everything awesome. You shouldn't really— you should make a *few* things awesome for your customers. If everything is shiny and full of complexity, your customers will be overwhelmed and confused. Worst of all, you'll be so irreplaceable that you can never be promoted.

If you're a DBA, this means you don't offer highly complex and detailed explanations of all of the settings and configuration to your management or the help desk. You offer summary information and aggregate performance data, and identify a few key places to give rich, complex information.

If you're a developer, this doesn't mean that you always write perfect code. But it may mean that you write a cool add-in to automate documenting code.

If you define the feature set for applications, this means you don't include every feature



customers ask for. Instead, you keep the overall interface simple, and carefully select the features that will be the most effective long term.

WELL PLACED...

Pick something noticeable and meaningful. Talk to your customers and peers about what they currently see as important. Ask them questions about how their processes work. Listen carefully.

Listen for what might simplify their life and their process, what you might be able to do to save them time.

If you talk to several customers and peers, you'll find patterns. Don't promise everyone everything— what you're looking for is a few good places to focus. These places are something where you can add a feature, develop a tool, write a report, or provide deep information. If you're in a large organization, it might just be a way that you can bring two teams together so they can help each other without your assistance. The key is that it needs to be noticeable: it needs to be something that makes a difference.



Success as a DBA today is creating an environment where you feel like you're part of something really special—and so do your customers.

SMART THINGS...

You're looking for a place where you can shine. You need this to be a "smart thing". This means it needs to be something people can easily describe. You need a quick name

to describe your contribution that sounds cool in your company culture.

Depending on your workplace, this might mean coining an acronym, making a code name, or just using industry terminology. But you want something short and memorable.

Here are some examples of names you might introduce for your features:

- **The Activity Tracker:** a daily report on the total inserts, deletes, and selects on critical tables.
- **The Build Watcher:** a utility for the nightly build that does x (there's a myriad options of utilities you can do for your builds)
- **The Hall Monitor:** a utility that tracks changes in permissions granted to databases.

IN A GOOD-ENOUGH ENVIRONMENT

While you're finding a few achievements you can create in your job that create a great experience for your customers, you don't want them to often be horrified by using basic services.

THIS CAN BE TRICKY

What if you're greatly understaffed? What if things beyond your control are a mess, and constantly make your customers unhappy?

To some extent, this is always true in all our jobs. There's always a few things that aren't perfect which cause problems for people, but the costs to fix them are so high that they don't get tackled.

What you need to do is to figure out how to get a good-enough general experience for your customers. When big problems surface, talk to them with your customers. Don't contest whether there is a problem— be open about the situation and the costs of changing things. You want your customers to understand that you listen to them. This



will help them understand when the problem is out of your scope.

If problems regularly cause critical situations for your customers, use these situations to identify just a few “smart things” you can do to show your customers you’re working for them. Create a tool that supports workarounds. Scope your plans for a few things that will improve usability, and you’ll distinguish yourself as someone worth promoting.

BEING PART OF SOMETHING REALLY SPECIAL

Success as a DBA today is creating an environment where you feel like you’re part of something really special– and so do your customers.

You don’t achieve that by bringing in unicorns and rainbows. You make yourself successful by being a great team member, and by making yourself known for a few special, noticeable, key things.



[SHOP](#) [MY ACCOUNT](#) [BAG](#) [CHECKOUT](#) [HELP](#)

Upcoming In-Person Training Classes

The Senior DBA Class of 2016

5-Day Training Class



The Senior DBA Class of 2016 In-Person Training Classes

You're a SQL Server DBA who is ready to advance to the next level in your career but aren't sure how to fully master your environment and drive the right

SQL Server Performance Tuning

5-Day Training Class



SQL Server Performance Tuning In-Person Training Classes

You're a developer or DBA who needs to speed up a database server that you don't fully understand – but that's about to change in five days of learning and fun with

Advanced High Availability and Disaster Recovery

1-Day SQLSaturday Pre-Conference

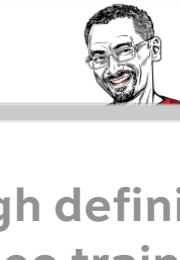


Advanced SQL Server and Disaster Recovery In-Person Training Class

You're not scared of clusters and database mirroring. You've played around with Always

Performance Tuning When You Can't Fix the Queries

1-Day SQLSaturday Pre-Conference



High definition
video training on
your desktop,
laptop, or even
your iPad.

brentozar.com/go/any

Get More Help from Us

We've got more tricks than a pony



by [Brent Ozar Unlimited®](#)

We love - no, we LOVE - helping people get relief for data pains, and we've got lots of options to help.

FIRST AID: TOTALLY FREE STUFF

We build cool troubleshooting tools and give them away for free:

- [sp_Blitz®](#) - fast SQL Server health check.
- [sp_BlitzIndex®](#) - identifies indexing madness dragging down your SQL Server.
- [Our blog](#) - thousands of articles on performance tuning, availability, and career development.
- Much more - like [our posters](#), [YouTube videos](#), and [weekly webcasts](#).

IN-PERSON TRAINING CLASSES ACROSS THE UNITED STATES

Our classes are taught by real experts with hands-on knowledge - specifically, us. We share the latest cutting-edge tips and tricks that we've learned in real-life deployments.

We're available for questions and answers - it's your chance to talk face-to-face and get personal advice on your tough challenges.

[Join us in-person at our classes!](#)

VIDEO COURSES:

\$29-\$299 FOR IN-DEPTH KNOW-HOW

- How to Think Like the SQL Server Engine, \$29 - Kick start your performance tuning with insight into SQL Server's brain.
- DBA Job Interview Question and Answer Kit, \$29 - Practice questions and more.
- Virtualization, SANs, and Hardware for SQL Server, \$299 - 5 hours of subsystem secrets.
- How to Tune Indexes and Speed Up SQL Server, \$299. 6.5 hours of quizzes, scripts, and more!

You can watch our high-definition training from your desktop, laptop, or even your iPad for 18 months.

That's just a sample - [check out the full list](#).

SQL CRITICAL CARE®:

A FASTER, SAFER SERVER IN 4 DAYS.

Tired of struggling with a slow, unreliable SQL Server?

In just 4 days, we'll work together with you to get to the root cause, explain your options, and give you a simple, prioritized action plan to make the pain stop.

We don't keep secrets: you get to keep our scripts, and you watch us work. It's like the best conference training, but in your own environment.

Schedule a free 30-minute consultation with us and learn more about our SQL Critical Care®: email Help@BrentOzar.com.

