

whitepaper

ZERO TO HERO — 12 ESSENTIAL TIPS FOR THE ACCIDENTAL DBA

A DBA's job can seem thankless, and DBAs can feel underappreciated because the work they do isn't always visible. But when application performance slows, and the database is being blamed, the work they do to keep the business up and running is quite evident. At the most basic level, a DBA must keep critical systems up and running 24x7. A DBA needs to make sure there is a proper strategy to bring up the environments when things go wrong inside your datacenter.

For many organizations, such as Independent Software Vendors (ISVs), it is difficult to justify a dedicated DBA. Instead, these organizations give database administration responsibilities to senior developers, making them part-time DBAs. These part-time or "Accidental" DBAs—IT professionals who have had to take on the responsibilities of database administration—need guidance and education.

This paper is for the Accidental DBA and contains 12 essential tips for working with SQL Server. These simple, yet powerful tips are filled with lessons learned on the front lines and from years of database administration experience. Following these tips can help transform the Accidental DBA into an organized DBA with clear direction for ensuring the best possible database performance.

Know Your Backup Requirements

Performing backups is a primary activity for DBAs, and all DBAs should be proficient at it. In my interactions with DBAs over the years, I always ask, "How easy is it to take a backup inside SQL Server?" Some of the answers I get are shocking. Once, a DBA answered, "I can take backups inside SQL Server without touching the keyboard, but using just the mouse." Thinking deeper, I can always say there is something about SQL Server that makes even developers working as a DBA shine—it's called simplicity in complexity.

The point here is not about how you do the backups, but about understanding how critical the backups really are to the business. Always start with basic business requirements. What are the Recovery Point Objective (RPO) and Recovery Time Objective (RTO) requirements of your teams? Based on your understanding of the business requirements, you must decide the frequency and type of the backups that need to be taken. It is important to understand the different types of backups and how they function before preparing your backup plan.

Know Your SQL Server Error Logs

Know your SQL Server instances and the number of instances that are running inside a single server box. Whenever you encounter an error or an unexpected behavior inside SQL Server, the first thing you should do is look into the SQL Server error logs.

In SQL Server 2005, the location is C:\Program Files\Microsoft SQL Server\MSSQL.x\MSSQL\LOG directory, where x is the instance ID that you are currently working on.

In SQL Server 2008, if you have a named instance, the location is C:\Program Files\Microsoft SQL Server\MSSQL10.<instance name>.

If the SQL Server instance is not starting, you can open the errorlog files using Notepad or any text editor. If the SQL Server instance is up and running, then use the `sp_readerrorlog` store procedure from SQL Server Management Studio (SSMS).

Monitor SQL Server Agent Jobs and Performance

SQL Server Agent is one of the critical services running inside a deployed SQL Server. Most of the maintenance plans, index rebuilding activities, TSQL Jobs, SSIS Jobs, and many more are configured with SQL Server Agent services. Often as a DBA, it is important to keep track of what happens inside this service.

If there are problems in this service, check the SQL Server Agent jobs. You can look up the registry for the "ErrorLogFile" key under `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL12.SQL2014\SQLServerAgent`. If you need to troubleshoot, be sure to make a note of this location. One very important piece of advice: Please review your logs for any failed jobs in the system from time to time.

DBAs should also continuously monitor database performance. Businesses don't look at the overall server infrastructure, but instead look at the database from the application's point of view. If the application is slow, it is all too common to blame the database, often without having any facts to back up the assertion. For DBAs who need to proactively monitor and troubleshoot performance issues, a 3rd-party solution such as, SolarWinds Database Performance Analyzer (DPA) for SQL Server provides a simple, visual view of current and historical performance, whether running on-premise or virtualized servers. This is especially helpful because DPA doesn't disrupt the current infrastructure and doesn't put a load on the monitored servers.

Review Application and Security Logs

In addition to the standard SQL Server error logs and SQL Server Agent logs, you should also keep an eye on the machine's error logs. Often, these logs contain important information that gets missed by other log files. Keep track specifically of the Security logs. You will likely see any access-related errors getting logged into this location.

Failed login attempts are also logged in the security log, provided that it has been configured inside SQL Server. Note that you can view the security logs even from inside of SQL Server. The security permission required for the same is a Sysadmin privilege.

Perform Security Checks

Databases and tables have to be in prime condition and without any corruption when it comes to systems running 24x7. SQL Server DBAs should understand some of the options available to maintain data integrity. For example, commands

like DBCC CHECKDB, DBCC CHECKTABLE, DBCC CHECKALLOC, and others. These are the building blocks for integrity checks, and you should know when to use each. These commands are required to check the logical structure, allocation, and structural integrity of the database tables. As a newly minted DBA, you must learn when these commands should be run. When do the jobs for these run today? Are there any errors during the run? How do you locate the errors? What is the mitigation plan if errors occur?

One of the common mistakes I have seen in a real-world production environment is that many integrity checks are well-hidden inside the maintenance plans. Because of this, DBAs sometimes neglect reviewing their current schedule. Equally important to knowing when the commands are run is keeping an eye on the errors that occur when these jobs are run.

Set Up Alerts and Notifications

Alerts are a basic building block of any monitoring program. Inside SQL Server you should have a mechanism to send an alert when something is abnormal. For example, you can set up an alert to monitor when the hard disk is running out of space, when tempdb is growing because of auto growth, when the amount of free space on the data file is below 5%, when the server's memory is beyond a certain value, for failed job steps, and more. Alerts and notifications are an important component in a DBA's toolbox.

Although SQL Server has some built-in alerting features, you should also take a look at SolarWinds DPA because it provides a powerful way to build alerts based on specific conditions.

Maintain Indexes

Indexes are like a lifeline for any database—especially for OLTP workloads. When the table is new and fresh, the allocations are clean and nice and everything seems to work seamlessly. Over time, these indexes get fragmented and take up space, which can slow performance. This is why Index maintenance is a critical part of a DBA's responsibilities. In simple terms, I have my car serviced once every six months as a regular checkup. It is not because my car is giving me any problems, but rather to make sure the car doesn't have problems when I am going for a long drive. So you should regularly perform a master checkup for database tables.

Please note that while this is an important maintenance activity, you should be careful to not to overdo this. I have seen DBAs complete a REBUILD of all tables almost every other week. My response to them has always been to ask, "Are all the indexes fragmented? Are you experiencing any problems on a particular table or an Index?"

Strategize index maintenance based on the tables or specific indexes that have problems. Keep an eye on DMV sys.dm_db_index_physical_stats for fragmentation level, and then make a judicious decision on which indexes truly need to be rebuilt.

Know Your HA and DR Requirements

Knowing your high availability (HA) and disaster recovery (DR) requirements is not as tough as it seems. If you know your requirements, then be clear about which option you will be using: Failover Clustering, Log Shipping, Replication, or Database Mirroring? On SQL Server 2012 and later, I have seen an increase in organizations opting for SQL Server AlwaysOn functionality.

It's important to not confuse or mix HA with DR requirements.

If you try to mix HA and DR, the resulting architecture is likely to be sub-optimal and prone to have problems. If the business is critical, you should push to have local redundancy (HA) and a remote stand-by site (DR) to support the business requirements. If your business has already deployed something, make sure you understand the current topology, the configurations, and the moving parts. Know how a planned DR or a real disaster recovery process will work. This helps you be prepared in the event of a real disaster.

Repair / Remove Orphaned Users

Knowing what logins are enabled and functional inside your SQL Server is one of the important tasks a DBA performs. No DBA will ever want to give away server-access control. In many environments, I have seen people do an in-place upgrade of their environments, hardly looking at any legacy code/users sitting inside their server. This can cause problems later on, as orphaned users (users inside the database that don't have an associated login on the server) can create risks. These orphaned users must be rectified or removed as part of any post-upgrade process.

Clean Old Backup Files

Know the data retention policy of your organization. It is important to make sure you do not accidentally delete backups or schedule any jobs that will delete them as part of automation. As discussed earlier, know where the backups are and try to have a proper naming nomenclature so that backups can be easily identified for:

- Data and time of backup
- Type of backup
- Server from where the backup was taken
- Database which was backed up

There is no shortcut for building a backup process, and there is no excuse for failing to evaluate that policy based on your company's requirements.

As a small tip, I know DBAs love command line utilities and automation. On Windows® Server 2003 and later, there is a neat little utility called "Forfiles," which can be used to automate the deletion of files based on time.

Know Your Storage

If you are new to the DBA role, get to know your environment. If you have become an Accidental DBA, start by finding out where your data files and log files are stored. Are they on a SAN, SAS Drives, SSD drives, SMB Share, or Azure Drive? What is the RAID levels for each of the drives? Are these physically separated drives? Also, know where the system databases are located and which drives these reside on like for tempDB, master, model, and msdb.

If you are connected via a SAN, analyze how these are connected via the various network layers. Are they connected via iSCSI or Fiber or any other mechanism? Where is the store for your backups? How much space is pending for storing these backups? What is the retention time for your backups?

Do you know if storage is a problem inside your critical database deployment? Will faster drives help? Will SSD or special hybrid drives help your workloads? These questions aren't always easy to answer, especially if you are a new DBA. With SolarWinds DPA 9.0, you can analyze the storage level of performance and its impact on a given query. You can also check the performance trends for access to data and log files. DPA also captures historical data, and can help pinpoint bottlenecks.

Know Your Service Packs

This is the simplest tip I can offer any new DBA. Know the service packs of the servers you administer. It is important to script them and keep them handy. Whenever a critical patch comes, evaluate its impact on your environment. If you are using any services mentioned in the hotfix, make sure you install on a test server to see if it impacts any functionality on your servers before pushing it to production.

Document Everything

If you are an accidental DBA, document everything that you do on a particular server. Also, make a note of the date and time when you did something and for what reasons. I have often seen people get into trouble, do a searching query, and perform some random steps suggested on a website they found. This ad hoc approach can cause more damage than you can imagine. Before trying something new and unfamiliar to you, always research and learn as much as you can before you act.

Secondly, documenting everything helps DBAs who might be joining the team in the future. Your notes and documentation can help educate new team members. They can also help you expand your career skills set.

Conclusion

A DBA is critical to an organization. Even though SQL Server makes the job function of a DBA look trivial and easy, it's the processes and checks and balances that DBAs bring to the table that are invaluable. In this paper, we discussed the high-level activities that every accidental DBA must know before embarking on this

kind of work. These are generic recommendations and I am sure many senior DBAs are already doing these processes, irrespective of their experience. These life lessons are quite important if you are starting your journey into a DBA career.

About The Author

Pinal Dave is a Developer Evangelist. He has authored 11 SQL Server database books, 14 Pluralsight courses, and has written over 2900 articles about database technology on his blog <http://blog.sqlauthority.com>. Along with 10+ years of hands-on experience, he holds a Master of Science degree and a number of certifications, including MCTS, MCDBA, and MCAD (.NET). His past work experiences include Technology Evangelist at Microsoft and Sr. Consultant at SolidQ.

About Confio and SolarWinds

Confio Software, now a part of the SolarWinds family, builds award-winning database performance analysis tools for DBAs and developers. SolarWinds Database Performance Analyzer (formerly Confio Ignite) improves the productivity and efficiency of IT organizations. By resolving problems faster, speeding development cycles, and squeezing more performance out of expensive database systems, Database Performance Analyzer makes DBA and development teams more productive and valuable to the organization. Customers worldwide use our products to improve database performance on Oracle, SQL Server, Sybase and DB2 on physical and virtual machines.

For more information, please visit: www.solarwinds.com/dpa-sql