

IDERA®

SQL Server Whitepaper

# THE ADAPTIVE SQL SERVER DBA

BY SCOTT STONE

# TINKER, TAILOR, **DEVELOPER, DBA**

Tinker, Tailor, Developer, DBA is not intended as a pitch on a slightly less harrowing book or movie than the John Le Carre classic. Rather, it is a consideration of the career progressions of those working in and around SQL Server and how that progression and interaction has evolved in the past ten to fifteen years.

The phrase comes from the English children's counting rhyme that originated who knows when:

---

TINKER, TAILOR, SOLDIER, SAILOR,  
RICH MAN, POOR MAN, BEGGAR MAN, THIEF...

---

The idea was to recite the rhyme while counting some objects and your fate was determined by where you landed with no objects remaining. The variations and evolution of the simple rhyme are interesting but the point is that many careers are similarly determined in the world of SQL Server management. As chance takes us from one position to another, filling one void or another for a particular posting, the industry itself is also changing as the interaction between DBA and Developer has evolved.

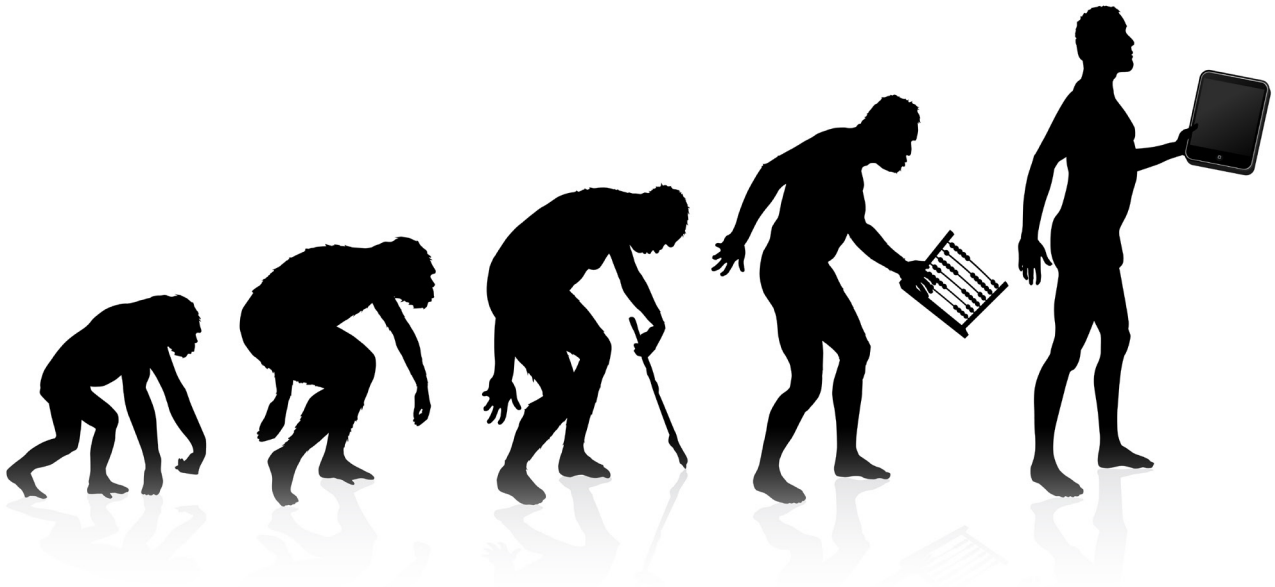
# ARE DBAS BECOMING OBSOLETE?

The DBA as defined in the 1990s is certainly headed for extinction, like dinosaurs, if not already fossilized among the raised floor datacenters of the past. But those DBAs were forced to adapt in recent years thanks to massive increases in numbers of instances, databases, and the size of databases.

We've heard too many tales of impending doom for the DBA. Thankfully it never transpired making it hard to take such predictions seriously anymore.



- 1 The first wave of predictions came from the database vendors themselves, touting the self-managing and self-tuning capabilities of new database releases. But they failed to take into account the growing complexity they were introducing in database features that outstripped the advancements in automating mundane tasks and self-tuning engines that worked “most of the time” but sometimes failed miserably.
- 2 After that came the distribution of roles and responsibilities once reserved for the DBA. DBAs lamented performance issues directly caused by the ignorance of storage administrators believing SANs were independent of the arcane DBA practices of meticulous manual storage management to reduce I/O contention. Although storage never went back to the DBA’s control, the influence of DBAs helped to restore some level of sanity to the faulty ideas that mission critical database transaction storage could be treated no differently than file storage without consequence.
- 3 Still later the virtualization wave finally hit the database, with Microsoft SQL Server embracing the concept before any other major vendor as a differentiator. Many virtualization advocates are among extremely experienced DBAs, but all preach the caveat that the DBA should control the virtual environments where databases were running. Several cautionary tales emerged of database performance hits that were traced back to ignorant system administrators moving heavy application images onto database servers without notification of any DBA. The DBAs in question proved their worth by being the ones to anticipate the problem, isolate the problem, and use the situation to justify regaining at least some input about what virtual images could share resources with critical databases and their applications.
- 4 The latest prediction of gloom for the DBA profession concerns the movement of more data to the nebulous “cloud,” with the presumption that this data will now be managed perfectly in the cloud without risk or performance impact.



# EVOLUTION OF THE DBA ROLE

## DBA to Production DBA

There was a time when all DBAs were effectively “Production DBAs” to the point that no one felt the need for a different term for “production DBA.” This period largely predated the emergence of SQL Server as a viable enterprise database, so many of you may not relate. Oracle, Informix, and Sybase ruled the open systems RDBMS world and DBAs largely managed things in the same ways the IBM DB2 DBAs had managed mainframe databases. Everyone respected the arcane rituals of backups, artful storage allocation for performance, index allocation, and statistic maintenance practiced by these specialists and no one would think of implementing a database without an experienced DBA to maintain it.

There was little direct interaction between the developers and DBAs during this era and DBAs were charged with tuning databases for maximum performance for the applications that life dealt them. The separation of duties and IT organizational structure somewhat dictated this reality. But the DBA controlled most of the database related environment directly including storage and servers.

Some either suggested tuning the application for better use of the database or bridged the gap and established themselves as “Development DBAs” to help developers write more efficient code for database access.

## DBA Specialization

As time progressed and SQL Server gradually pushed its way into the enterprise application world with each new release, so did the specialization of DBAs in large organizations. There were Application DBAs who either helped developers tune code or tuned application performance in consideration of how the database engine actually worked. But the specialization meant someone else, the Operations DBA, needed to make sure the housekeeping was maintained, backups were ready, instances were up, patches were maintained, etc. The hybrid Developer-DBA or DBA-Developers were cultivated to avoid the need for the Production DBAs to suggest changes to SQL statements or other load tuning advice.

But while the DBA expertise spread itself into different parts of the IT organization, these parts were still somewhat isolated in many cases and sometimes at odds with each other internally.

The unifying factor for some of these DBA specializations was the gradual loss of control of their environment. Storage and system virtualization emerged as threats that took control of things DBAs were used to controlling and using to manage database performance. The DBA still had responsibility for database performance but less control over things that impacted that performance.

Vendors sold tools that purported to “eliminate finger pointing” when in reality they were being used to “optimize finger pointing” by monitoring what they controlled and preferably those things around them as well that were out of their control.

## Back to Modern Times (Rise of DevOps)

About 7 to 8 years ago, we began to hear hype around a new buzzword - DevOps - which purported to move from a world of isolation and separation between production and development to one of continuous interaction and improvement. My personal experience at the time was the idea was as mythical as a unicorn because the hype did not match the reality among my customers. I am sure someone somewhere was implementing these ideas, but none of them seemed to be in large enterprise level organizations.

The unique thing about the SQL Server platform compared to its RDBMS competition was that Microsoft seemed to embrace the changes other platforms resisted. Virtualized databases are one example of such a move. While databases as a whole seemed to be the last thing to go virtual, the first databases to go virtual were always SQL Server. And now it seems to be the rule more than the exception.

In like fashion, probably because of the number of SQL Server DBAs being either explicitly or “accidentally” recruited from the ranks of developers, SQL Server seems to be at the forefront of DevOps even by companies that are not trying to implement DevOps.

# WHY SO MANY DBA CONSCRIPTS?

Accidental, reluctant, or any of the other terms that have been coined to designate these brave people filling the role of a DBA without prior training is not a new phenomenon, but it is increasing.

Some developer or IT administrator is designated as default DBA. And if they are lucky then they know enough to seek out the training needed for the job they have inherited. If not, then it takes a crisis of poor performance, lost data without backup, compromised security, or any of the other catastrophes that trained DBAs are charged with preventing to force the issue.

The culprits to this long trend toward accidental or involuntary DBAs are hard to pin down. **Here are the most likely suspects:**

- 1 Database Vendors** – For a long time the database vendors have oversold the self-managing aspects of database engines, the whole time increasing the complexity of options and tuning adjustments available in those same database engines. For every old adjustment that is automated two new variables are added to track, adjust, and optimize.
- 2 Technically Naïve Management Teams** – This is probably the biggest factor as cost control efforts allow for wishful thinking that the DBA tasks can simply be absorbed by development teams without preparation or training. The lucky ones inherit savvy developers who had prior DBA experience or know enough to seek the training they need proactively.
- 3 Evolving Application Models** – Many application vendors like to obfuscate the fact an underlying database even exists to support their application. It is not in their best interest to fully disclose the total cost of ownership likely includes dedicated administrators.

# DBA ORGANIZATION HIERARCHY

Many SQL Server DBAs in particular are being thrown into the position as an afterthought, often after a crisis of some sort convinced the corporate leaders that SQL Server was not as self-managing as they imagined. And in those cases there are lots of lone rangers out there.

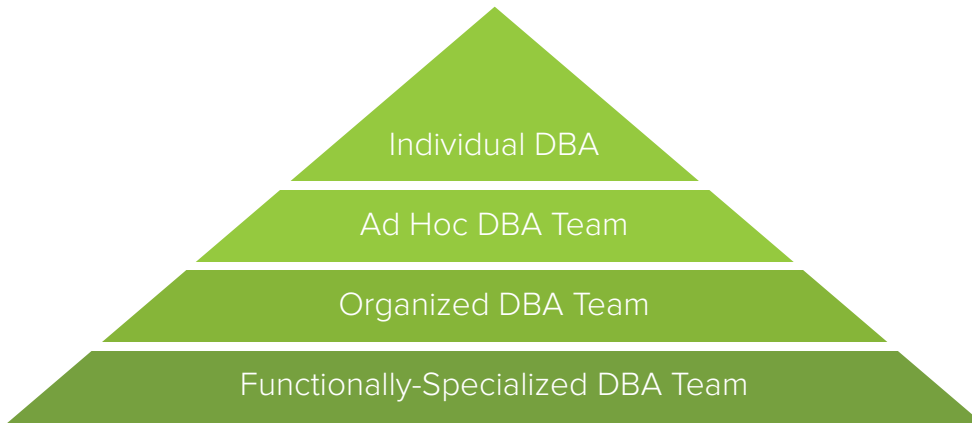
If you are in an organization large enough to have a DBA team, the next factor to consider is organization, or lack of one. Some DBA teams are loose organizations that simply tackle problems and tasks in an ad hoc manner or according to proficiency. They are called “the tuning guy” or “the capacity planner” or “the troubleshooter”.

If the organization is large enough and has had enough bad experience with the ad hoc hero approach, then some sort of structure is applied to the DBA responsibility. Sometimes it goes by technology specialty, sometimes by business unit alignment, or even by simple geography. Everyone knows who is primary on any given instance, but that person often still resorts to using other DBA specialists on the team depending on the task or situation.

Finally, large organizations do all this and more by adding the dimension of DBA specialty. There are Application DBAs who focus on application performance; Operation DBAs who make sure the lights stay on and the data is always backed up; Development DBAs who serve as the buffer to poorly designed SQL applications from developers themselves. Engineering DBAs who do nothing but work on finding tools and best practices to standardize across all those other DBAs in the organization and make them more efficient.

You can recognize a similarity to the Capability Maturity Model used to describe organizations and software development processes. The paradox is that each level of “maturity” also adds complexity and those ad hoc DBA heroes in small shops can be more efficient than highly organized teams. But you simply can’t function in a large organization with a large team of freelancing DBA heroes, no matter how good they might be as individuals.





Regardless of where you find yourself on this pyramid, you probably recognize you are part of a team. You either have to work with developers, storage administrators, virtualization administrators, or even business customers to get your job done efficiently. The same tasks that those specialized teams do in larger organizations have to be done by someone somewhere, maybe not done well, but someone is filling those roles whether they are DBAs or not.

## AND NOW TO THE POINT!

Just as SQL Server DBAs are being minted on purpose or accidentally thrust from developers into the role of DBA, many are being called upon to increase interaction with developers to improve database and, by extension, application performance.

That means we encounter more people every day who have been working in various capacities around databases, in particular SQL Server databases, throughout their career. And the line between these roles is blurring.

Developers morph into DBAs while expected to continue to help development, but become overloaded with production responsibility. Production DBA specialists are brought in to help bring a sense of consistency and order, but are expected to provide continuous performance feedback to application developers to suggest performance improvements. Developers or other IT administrators are forced into temporary “DBA duty” to cover for DBAs on leave or overworked in general.

All these hypothetical career situations are not really hypothetical at all. As DBAs are forced by career or responsibility to expand beyond the old specialist silos, communications and understanding is increasing for all.

# THE ADAPTIVE DBA LIVES ON

The SQL Server DBA of today is more like that chameleon in the picture, surviving and adapting; than the giant over-specialized and antiquated dinosaurs as they are sometimes inaccurately portrayed.

Data is more important than ever to business success. Data is the **key asset** to many modern businesses and when the management and mining of that data for business intelligence is your business' core competency, you cannot afford to totally outsource oversight and control of that key asset to a faceless vendor. The data itself may be most efficiently housed in a cloud environment, but someone with the sensibilities of a DBA for data integrity, redundancy, security, and performance will always need to be managing it... *at least for the wise business managers.*



# SQL MANAGEMENT SUITE

## COMPLETE SQL SERVER MANAGEMENT

- **SQL Diagnostic Manager** Performance monitoring, alerting and diagnostics (*screenshot shown*)
- **SQL Compliance Manager** Monitor, audit and alert on activity and changes
- **SQL Secure** View and manage SQL Server permissions
- **SQL Safe Backup** Hands-free backup across all SQL Servers
- **SQL Defrag Manager** Find and fix SQL fragmentation hot spots fast

Start for FREE



IDERA understands that IT doesn't run on the network – it runs on the data and databases that power your business. That's why we design our products with the database as the nucleus of your IT universe.

Our database lifecycle management solutions allow database and IT professionals to design, monitor and manage data systems with complete confidence, whether in the cloud or on-premises.

We offer a diverse portfolio of free tools and educational resources to help you do more with less while giving you the knowledge to deliver even more than you did yesterday.

**Whatever your need, IDERA has a solution.**

I D E R A