

Autonomous Databases Are Here. So Who Needs Database Administrators Anymore?

Quest®

The future of DBAs in the era of the autonomous database

By Clay Jackson, Database Solutions Sales Engineer, Quest®



The database world is abuzz with the prospect of autonomous databases. Like vehicles, autonomous databases are said to be self-driving, with all the benefits of lower costs and reduced error rates that come from removing the human factor of database administrators (DBAs).

Will DBAs go away as DBA responsibilities dwindle? The marketing messages around autonomous databases suggest that they will, and some DBAs (and IT directors and CIOs) are buying in. It's tempting to think that the database will run itself and labor costs will plummet. And if DBAs continue to regard their job in the same way even as companies adopt autonomous databases, then yes, many DBAs will indeed go away.

But are autonomous databases the real problem DBAs should be focused on? Which other factors are at work? How does the future look for DBA responsibilities in the world of the autonomous database? How can DBAs position themselves to thrive in such a future?

This paper examines autonomous databases from the perspectives of companies interested in adopting them and of the

DBAs they will most affect. CIOs and IT directors can use this white paper to inform their decisions about purchasing and hiring, and DBAs can use it to better understand their future in the era of the autonomous database.

THE AUTONOMOUS DATABASE: IS THAT REALLY A THING?

“The World’s First ‘Self-Driving’ Database,” says the messaging for the autonomous database. It claims “No Human Labor — Half the Cost” and “No Human Error — 100x More Reliable.”

That may be. But “human labor” refers to tuning, patching, updating and maintaining the database. Most DBAs know that database automation and scripting have taken care of a lot of that work for years.

Is it possible for a database to drive itself? If so, which parts of traditional database administration must remain in human hands? And what about everything else DBAs do?



The autonomous database itself could be in the cloud or on premises, but the way to connect to and provision it is through the cloud.

Consider the range of other tasks DBAs perform:

- Gathering business requirements
- Setting up and configuring the database initially
- Loading data — Getting it out of existing systems and into new ones
- Extracting data — Taking it from autonomous and other systems and turning it into knowledge
- Tuning SQL — Working with database developers to improve the SQL and code they write
- Enforcing security — Keeping cloud-based, autonomous databases accessible only to users who should have access
- Reporting

Even the most autonomous database cannot take care of all that, so it won't take away all the functions performed by a DBA.

But really: What is the autonomous database?

The autonomous database entails cloud data management. The database itself could be in the cloud or on premises, but the way to connect to and provision it is through the cloud.

Note that although Oracle will make 18c available in the cloud and promote it, Oracle 18c by itself is not an autonomous database. Oracle is making its 18c release

available first in its cloud machine and only to its cloud customers, but even there the database doesn't necessarily include automated cloud features such as backup, recovery, patching and machine learning. That whole feature set is in Autonomous Data Warehouse, running atop 18c.

Furthermore, the term "autonomous database" is not perfectly interchangeable with the term "Oracle 18c." (For that matter, it's possible to run Oracle 12.2 on premises or in Oracle's cloud, whichever suits the organization's IT needs.)

What is true about Oracle's autonomous database, however, is that Oracle configures, manages and maintains it, which is how they achieve the high uptime they advertise. To optimize the 18c release for its cloud environment, Oracle has added several features:

- Cloud automation that provisions and sets up hardware and software in the cloud so that entire databases can be instantiated in just a few clicks
- Advanced versions of Oracle DataGuard for disaster recovery and high availability
- Tuning to make all the components work together
- Machine learning for autonomous operation

It may seem safe for DBAs to tell themselves, "I know databases and I know cloud; therefore, database administration

as I've known it will work out just fine for me." But the truth is that autonomous databases can have a bigger impact on DBAs' careers than meets the eye. DBAs who fail to recognize the shift that comes with autonomous databases are in danger of missing the boat.

"Autonomous" doesn't mean "unbreakable"

Nature is notoriously hard on autonomous systems, especially in their early stages of development and rollout. In "Star Trek III: The Search for Spock," Scotty, the chief engineer on the starship Enterprise, aptly observed, "The more they over-think the plumbing, the easier it is to stop up the drain," which is how unpredictable conditions have occasionally led to tough outcomes with self-driving cars. Closer to the wallet, Bloomberg revealed that "hedge funds that use artificial intelligence and machine learning in their trading process posted the worst month on record in February [2018]."¹

Someone needs to understand how the autonomous trading program/vehicle/database is put together and how to repair it when something goes wrong — in mid-flight, if need be. Even if the code is sound, the input data can change dramatically or run far outside of original assumptions. When requirements change, somebody needs to know the internals well enough to modify the model.

DBAs are in a good position to play that role.

WHAT IS THE DBA OF TODAY?

DBAs are still DBAs, but most of them have far more responsibilities now than even five years ago. The trend toward multiple platforms is nudging Oracle DBAs, for instance, to learn SQL Server (or NoSQL or Hadoop or MongoDB), and vice versa, because it's what the business needs. Adoption of big data products and data analytics also leads to more demands on the time and talent of DBAs.

At the same time, database automation is gradually reducing the amount of work required for repetitive tasks like installation, configuration, storage management and workload management. Adaptive Tuning reconciles a variety of inputs,

without human intervention, to optimize database performance.

Automation is here and more is coming, so DBAs had better figure out what else to do with their time.

And their career.

The paradigm shift that deserves the DBA's attention is that the smart database administrator is actually evolving into the data administrator who sees that it's not all about the database; rather, it's all about the knowledge that helps answer business questions like "Why is product X selling so well?" "How can I sell more of product Y?" "Why is product X selling better than product Y?"

In other words, knowledge is data in business context and objectives. The business knows the questions to ask and DBAs know where to find the answers.

THE DBA OF THE FUTURE

The future belongs to DBAs who align those questions and answers to come up with knowledge about the customer based on captured data: where customers are, how they have behaved in the past and what they prefer. Companies use that data to differentiate themselves and they need professional DBAs to retrieve it.

Consider the business value to be added by mining data for knowledge about the customer:

- **Differentiation** — Most companies are trying to figure out how they can do better than an existing global brand such as Facebook, Apple, Amazon, Netflix or Google (FAANG). It takes active use of knowledge about the customer to carve out a niche, and even more to avoid being steamrolled by a giant.
- **Automated Processes** — Knowledge of the customer is the key to automating mundane tasks like package shipping and hotel check-in. It makes unremarkable processes simply disappear.
- **Customization** — With enough of the right data it is possible to offer unparalleled, individualized service through targeted marketing and promotions that know where customers are, what they like and details about them down to their sleeve length or dress size.

The smart database administrator is actually evolving into the data administrator who sees that it's not all about the database; rather, it's all about the knowledge.

¹ Burger, Dani, "Hedge Funds That Use AI Just Had Their Worst Month Ever," Bloomberg, March 12, 2018, <https://www.bloomberg.com/news/articles/2018-03-12/robot-takeover-stalls-in-worst-slump-for-ai-funds-on-record>

DBAs already know that the databases are protected. The DBAs of the future also know what's in them.

On the customer side, the devices that generate customer data are ubiquitous and the data is easy to obtain, even with directives like the General Data Protection Regulation (GDPR).

On the company side, more production systems are customer-facing and downtime is unacceptably expensive. That has caused high performance, high availability and disaster recovery to evolve from nice-to-haves into absolute requirements.

And on the DBA side, fewer of the dull tasks associated with that data require human effort. According to a survey by DBTA, "in many cases, since the routine and mundane elements in administering databases can be eliminated, DBAs will be free to focus on higher-value activities."²

DBAs already know that the databases are protected.

The DBAs of the future also know what's in them.

System design, system tuning and system understanding: A case study

In the evolution from *database* administration to *data* administration, the three elements of system design, system tuning and system understanding are paramount, as the following example illustrates.

A hypothetical company called Positive Manufacturing owns a number of plants that take orders for product. Normally, the plants run a program every day at 4:00 a.m. to generate pick lists for the day. They send the lists to computers on their forklifts to tell the drivers where to locate the loads, which the drivers start picking and loading at 6:00 a.m. Separately, order processors come in around 9:00 a.m. and start entering new orders.

Instead of picking just once per day, one of the plants decides to pick multiple times per day — at 6:00 a.m., noon and

3:00 p.m. — to keep drivers busy and to ship more product to customers faster. The first day they try that, the order processors start complaining around noon that the entry process has slowed to a crawl and that they cannot enter orders or serve their customers. After about 15 minutes that spontaneously repairs itself. The DBAs receive several complaints about the database being slow; examination reveals that it is locking because of the sharp increase in volume. They know that the picking program is causing the locking, but they don't know why.

The same locking and order processing slowdown takes place at the 3:00 p.m. cycle. It takes the DBAs most of the afternoon to discover that there is a startup parameter in the inventory program that controls the number of pieces of the inventory system that the picking process will lock before it starts its processing. It turns out that the system designer had thought, "Picking takes place at 6:00 a.m., when nobody is processing orders, so I'll lock the entire inventory table." He had added a parameter to lock the whole table while picking is going on, effectively locking everybody out of inventory so that order processors could not enter orders until the picking process had finished.

Thus, two different systems are trying to access the same data, and an important startup parameter is poorly documented. So, to coexist with the order management system, the DBAs change the parameter to lock only the current order and any other orders associated with the same products. The problem then goes away.

The point is that an autonomous database is not designed to repair itself at that level. All it will reveal is that there is a locking problem it cannot solve. And that is where skill and experience as a DBA who understands the application will pay off.

² King, Elliot, "DBAs Face New Challenges: Trends in Database Administration," Unisphere Research, December 2017, <http://www.dbta.com/DBTA-Downloads/ResearchReports/DBAs-Face-New-Challenges-Trends-in-Database-Administration-7676.aspx>



When a particular item isn't selling, the DBA's goal is for the business to approach him/her and ask for the data to explain the low volume.

THRIVING AS A FUTURE-PROOF DBA

DBAs can examine a couple of trends and do their own extrapolations into the future.

First, they can be thankful that certain tasks will take less of their time, as shown in Figure 1:

Which of these tasks are automated, performed manually, or not generally performed?

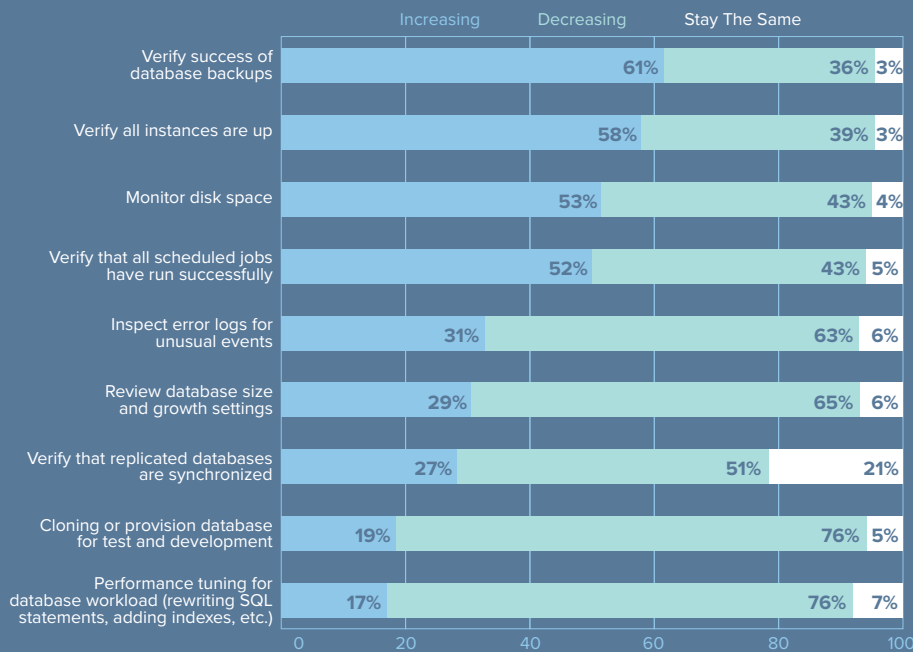


Figure 1: Tasks taking less of DBAs' time ³

³ King, p. 23.

In terms of the data management infrastructure in your organization, over the next three years, what do you think will be the top challenges facing DBAs?

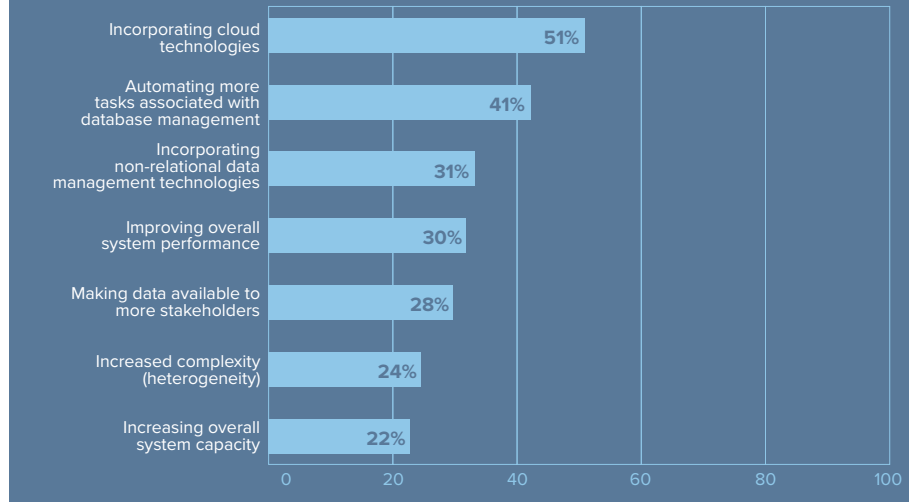


Figure 2: Upcoming challenges for DBAs ⁴

The survey results indicate that database automation is getting verification and monitoring tasks off the plate of most DBAs. Figure 2, above, shows the kinds of tasks likely to take their place.

Although those tasks may be less mundane and require more skill than the ones automation is taking over, they still revolve around *database* administration rather than *data* administration.

To truly embark on a future of building knowledge about the customer from data, DBAs will have to start getting out of their cubicle and getting around.

Getting out into the business to see what's going on

The first task is to discover everything possible about the business. Internal auditors are a valuable resource in this effort because they look for things DBAs are not accustomed to looking for. If auditors are focused on a particular part of the business or set of issues, it's probably because they suspect fraud or illegitimate activity, but more likely, they feel it's an important part of the business. Auditors know what they're looking for, but they don't have the DBA's data retrieval and reporting skills to find it.

Next, all internal customers have something that keeps them awake at night. For example, the CIO or database manager may wonder, "Can we truly rely on our backups?" — something DBAs can easily test. Or they may wonder, "What will be our recovery time if we have a disaster?" That's a more difficult question to answer, but it's more important, and DBAs are as close as anybody in the organization to being able to answer it.

When a particular item isn't selling, the DBA's goal is for the business to approach him/her and ask for the data to explain the low volume. Opening data catalogs on the item's location, its quality, the amount of time on the shelf and the number of returns, DBAs can put something together to help find out why. It's unlikely that they'll know the answer right away, but they're in a position to know about other kinds of data that can lead to the answer.

More broadly, they're looking for the answer to business-oriented questions such as "What makes the business successful?" and "In a year from now, how can we determine whether we've been successful?"

For example, getting out into the business is the only way that DBAs in companies

Knowing something about the applications that support the business is a big step on the way to greater visibility and greater value as a DBA.

⁴ King, p. 25.

without budget for monitoring tools can stumble onto pain points and database-related problems. Suppose a DBA hears a casual complaint from Accounting about having to pay overtime at the end of every month because it takes longer and longer to cut checks and issue payments to vendors. With a few lines of code, a canny DBA could tune SQL and slash the amount of time and money Accounting spends at month-end.

Finally, it's useful to know the applications that internal customers depend on to get their work done. Consider a lab manager's system that occasionally crashes in the middle of the night, with no way for anybody to know or do anything about it until the next morning when the staff has to call the help desk and request a system restart. Even if DBAs aren't trained to diagnose and troubleshoot the main problem, they can put an alarm on the system to detect failure and restart automatically, without the need to contact the help desk. Knowing something about the applications that support the business is a big step on the way to greater visibility and greater value as a DBA.

Measuring and managing

DBAs can measure success at three different levels:

- **Metrics of the business overall** — Key performance indicators, or KPIs, such as bottom-line profit/loss and growth/decline demonstrate how the organization is doing.
- **Metrics of internal customers** — Sales may want to increase its numbers by 10 percent and Operations may want to open five new sites in the next six months. What kind of data do they need to achieve those goals? What is the best way to pull it together for effective use?
- **Their own metrics** — Because DBAs are still on the hook for uptime and availability, the traditional metrics and service-level agreements (SLAs) for different groups still apply. Accounting may need system access only between 8:00 a.m. and 5:00 p.m., but global development teams may need access to their code repositories almost 24/7. Another important traditional metric is response time, where autonomous databases and the service provider hosting them introduce new variables. If users expect a response from IT within 10 minutes of an incident, then the service provider's schedule should accommodate that SLA.

After identifying *what* they're going to measure, DBAs should think next about *how* they'll measure. The downside to giving up dull tasks is that DBAs also give up control over them; they can't control service providers, for example, but they can put sensible SLAs in place, measure performance against them and hold providers responsible for any lapses. Since their internal customers have set their own metrics and goals for using a given database, the DBAs' metrics should align with them, even if it means spending money on tools for monitoring or managing the database.

Nothing motivates quite like a paycheck, so some measurements are best tied to compensation. Consider building process improvement for service providers around a cookbook of process documents. Then tie a bonus to the number and quality of process documents that DBAs create each year related to their respective databases. That will make a huge difference in how they set their priorities.

Finally, DBAs can think about tools not only to manage their databases but also to automate database performance measurement. It's reasonable to expect that autonomous databases will take care of themselves 99 percent of the time and send alerts when they cannot. That leaves DBAs free from worry about mundane things like patching servers, rebuilding fragmented indexes and adding space for growing tables.

CONCLUSION: IT'S A GREAT TIME TO BE A DATA ADMINISTRATOR

What, then, is the future of DBAs in the era of the autonomous database? What will the DBA of the future look like?

Most DBAs are not in much danger of having autonomous databases eat their lunch — at least, not all of their lunch. But smart DBAs look beyond the hype around autonomous databases, embrace the automation of many database administration tasks and position themselves for a data administration role in their organization.

They learn the business and show that they've learned it by getting out of their cube. Once they understand how their business works and how useful their knowledge can be to some business

DBAs can think about tools not only to manage their databases but also to automate database performance measurement.

Somebody is going to be next year's chief data officer. It may as well be this year's DBA.

managers, they gradually increase their value to the organization. It will be a long time before anyone figures out how to automate that.

They learn the technology that keeps their business running. That means more than learning how to write Java code; it means learning how the pieces interact and fit into the overall plumbing, how all of the database components in the organization work. They know what happens, as described above, when the inventory table is locked and how that affects other business functions, so they avoid sending themselves and their fellow DBAs on a day-long bug hunt.

The smart DBA becomes a trusted advisor by asking questions in the terms business managers understand. "Why do you need same-store sales by product by salesperson?" "Which products bring people in for the first time and which ones keep them coming back?" "Which data points don't you have that you wish you did have?"

In short, somebody is going to be next year's chief data officer. It may as well be this year's DBA. Data is only growing — in both volume and importance — so every organization needs people to make it available and turn it into knowledge of the customer.

ABOUT THE AUTHOR

Clay Jackson is a Database Systems Consultant for Quest, specializing in Database Performance Management and Replication Tools. Prior to joining Quest, Jackson was the DBA Manager at Darigold. He also spent over ten years managing Oracle, SQLServer and DB2 databases and DBAs at Washington Mutual. While at WaMu, Jackson was the Enterprise Database Compliance Officer, with responsibility for Database Security and Disaster Recovery. He also worked at Microsoft and Starbucks, is a CISM and has a Masters in Software Engineering from Seattle University.

ABOUT QUEST

At Quest, our purpose is to solve complex problems with simple solutions. We accomplish this with a philosophy focused on great products, great service and an overall goal of being simple to do business with. Our vision is to deliver technology that eliminates the need to choose between efficiency and effectiveness, which means you and your organization can spend less time on IT administration and more time on business innovation.

© 2018 Quest Software Inc. ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal

Trademarks

Quest and the Quest logo are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit www.quest.com/legal/trademark-information.aspx. All other trademarks are property of their respective owners.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (www.quest.com) for regional and international office information.