

Autonomous Database and Changing Expectations for DBAs: When Your Ship Comes In, Will You Be Waiting At the Train Station?

It's Monday, and you're a typical senior Oracle DBA with a boatload of run-of-the-mill tasks to accomplish, plus a brand-new application development project that your boss has made you the lead project manager for because you've developed a reputation within your organization for being extremely organized and focused upon the needs of your company's businesses.

As you turn to the normal weekday tasks at hand – checking if all of last night's database backups were successful, looking for any unexpected spikes in poor database performance, and thinking about how to keep your junior DBA's performance tuning expertise up-to-date – there's a knock at your cubicle. You look up and realize it's your most troublesome user – let's just call her *Bernice Pane-Diaz* - who has worked at the company for almost 40 years and who also just happens to be the cousin of the CEO.

Bernice mutters the words you most dread to hear: *"There's something wrong with the database."*

Your heart flutters a bit as you ask, "What seems to be the problem?" Bernice says, "My query is running slow again. I need it to finish by lunchtime, and it's been running since my last break about an hour ago. You better get it fixed." You know from past experience that now you're going to have to personally disengage from your schedule to take a look at what's wrong with the SQL statement, application, and maybe even the database itself. And you remember that the CEO encouraged your mentor, the prior Oracle DBA, to seek other opportunities when she accidentally got on Bernice's bad side.

In the back of your mind, a voice keeps whispering: *There's got to be a better way.*

The Safest Place For a Ship Is at Harbor ...

I talk a lot with colleagues at Oracle user events, and I detect there's a still quite a bit of confusion and concern about where our profession is heading, particularly with the onset of Oracle Cloud Infrastructure and, more recently, the Autonomous Database. Our profession demands constantly upgrading our skill sets after any minor database release, and the pressure to direct our teams of developers and DBAs to build the best systems possible demands we consider the best technology for the job at hand. But lately I've also seen some interesting sea changes in IT organizations that portend a different path for our profession.

On-premises isn't really as great (or safe) as everyone believes. A lot of CIOs and senior managers I've spoken with desperately want to get out of the datacenter business. Granted, some organizations have regulatory or security constraints that require them to remain on-premises, but for many shops it makes sense to move to a Cloud-oriented strategy to manage hardware capacity effectively and flexibly. And while there's a natural resistance to placing trust in someone else's datacenter, there's also a growing realization that our own datacenter personnel must be vetted with extreme discretion to avoid having them [suborned by outside actors](#).

The BiCloud offers intriguing possibilities. The recent announcement of the [new Oracle – Microsoft partnership](#) – what I like to think of as *The BiCloud* (with apologies to Asgard) – may finally introduce the best of both worlds for IT organizations: the unquestionable security and performance of Oracle databases when absolutely needed, paired with Microsoft SQL Server databases when less demanding performance is needed, combined with the *de facto* dominant Microsoft Office 365 for the front office.

If Data Is the New Oil, Then Bringing It to the Surface Is Job One. Finally, the era of the DBA acronym connotation of “Don’t Bother Asking” is coming to an abrupt end. The new role – what I’m calling the *Enterprise Data Architect* – is focused on getting precisely the right data to the right user community at the right time – now! – securely and without added complexity. That means we’ll need to embrace Oracle features like 18c’s Partitioned External Tables and 19c’s Hybrid Partitioned Tables so we can combine data from within our internal Oracle databases with external data in CSV, HDFS, or JSON format.

Autonomous DB: Not a Toddler Any More

And that brings us to what I see as the true promise of Autonomous Database: If used correctly and for the appropriate application workloads, it can be a true force multiplier for busy Enterprise Data Architects. As we approach Oracle OpenWorld 2019, it’s important to realize that Autonomous Data Warehouse (ADW) is almost two years old now, and Autonomous Transaction Processing (ATP) nearly 18 months old, so there’s been plenty of time to shake out any unexpected issues with either platform.

I recently leveraged ATP Serverless (ATP-S) to shake out a scalable release of Swingbench that incorporates the Transaction Processing Performance Council’s “extreme” OLTP workload (TPC-E) that simulates transaction processing for a relatively complex stock trading system. I was pleasantly surprised to find that ATP-S provided five specific database services for each set of OLTP or reporting transactions. All I had to do was select the most appropriate service to assign the transaction to, and the ATP-S instance easily handled the transaction load that I threw at it.

Even more impressive, the new *auto-scaling* feature of ATP-S detected that over time, the workloads that I was testing would perform better with the addition of OCPUs. I had originally allocated just two OCPUs to my database instance, but as the application workloads’ demands increased, ATP-S automatically added two more OCPUs. When the peak demand period had passed as my workloads neared completion, ATP-S automatically reduced the number of OCPUs from four to three, and then from three to two ... all without any manual intervention.

ATP Dedicated: A Serious Commitment

I’ve also had the opportunity recently to experiment with the latest release of Autonomous Transaction Processing known as *ATP Dedicated* (ATP-D for short). The big advantage of ATP-D is the elimination of potential “noisy neighbors” – i.e., having to share a virtualized hardware platform’s resources with other potential consumers of similar resource demands.

Once again, I leveraged my Swingbench implementation of the TPC-E application workload against an eight-OCPU ATP-D instance to see how well it could handle its demands. My experimentation also yielded a look at the implementation of another relatively new feature in Oracle 19c, [Automatic Indexing](#). Automatic Indexing is automatically activated on any new ATP-D instance, so once I’d loaded the thirty-odd tables and corresponding primary key indexes required to execute a sufficiently-robust TPC-E workload, I was able to test out how well Automatic Indexing ascertained the need for any beneficial secondary indexes.

To my satisfaction, Automatic Indexing not only identified about two dozen secondary indexes based on the workload I generated, it also implemented the handful of them that led to immediate improvement of the workload by a factor of almost **670X**. Many of the workload’s SQL statements improved their

performance by several orders of magnitude as a result of the newly-constructed indexes; in fact, two statements improved by a factor of over **110,000X**. I'll be publishing deeper details of my experiences in an upcoming white paper just after Oracle OpenWorld 2019, but for now, here's a look at the corresponding [summary report](#).

In my playbook, the best thing about Automatic Indexing is that it is a likely foil for the case of Bernice, the application user whose constant complaints couldn't be dismissed, and the disruption those complaints caused for our typical Oracle DBA's daily workload. While the need for a secondary index certainly *might* alleviate Bernice's long-running query issues, it does take time to figure out if that's really the best solution to the problem. That's time spent away from my most important task as an IT professional: *helping my application development team build better systems proactively, not reactively*.

[... But That's Not What A Ship Is Built For.](#)

There's no doubt that the role of Oracle DBA is definitely undergoing a dramatic upgrade itself as the shift towards Cloud continues at a rapidly-increasing pace, and there's no doubt that Oracle technology like Autonomous DB is will continue to contribute to the dramatic changes required for the skill sets and responsibilities of Enterprise Data Architects. Personally, I've chosen to embrace these shifts as challenges to overcome instead of a direct threat to my professional career. Perhaps it's time to contemplate the wisdom of the old saying: "A ship in harbor is safe. But that is not what ships are built for."

About the Author

Jim Czuprynski has nearly four decades of professional experience in his career in information technology, serving diverse roles at several Fortune 1000 companies before becoming an Oracle DBA in 2001. He has been an Oracle ACE Director in 2014 and is a sought-after public speaker on Oracle Database technology features, presenting often at Oracle OpenWorld, IOUG COLLABORATE, ODTUG Kaleidoscope, Oracle Development Community tours, and Oracle User Group conferences around the world.

Jim has authored over 100 articles focused on facets of Oracle Database administration to his credit since 2003 at databasejournal.com and IOUG SELECT. He has also co-authored four books on Oracle database technology. Jim's blog, *Generally ... It Depends* (<http://jimczuprynski.wordpress.com>), contains his regular observations on all things Oracle and the state of the IT industry. He is currently the Senior Enterprise Data Architect for Viscosity North America.