Oracle spins up an autonomous JSON document database

Omdia view

Summary

Oracle has introduced a new database. Available immediately, Oracle Autonomous JSON Database serves as a specialized JSON document store. This new database isn't however entirely new. Instead, it derives the functionality from Oracle's flagship cloud database, Oracle Autonomous Database, creating an affordable and easy-to-use entry point for developers specializing in JavaScript Object Notation (JSON) development.

One database masquerading as many

There's something strange going on at Oracle. For several years now, the vendor from Redwood Shores, California, has sought to push its advantage in the marketplace as the leading relational database platform provider by adding more and more functionality to its flagship database, Oracle Database. Dating back to the late 90s, when the company added object-relational support and thereby bringing in a raft of new use cases to its database, Oracle has worked tirelessly to make Oracle Database a highly adaptive data platform capable of handling a wide array of disparate data types as well as operational and analytical workloads.

Into this core data platform, Oracle has also poured a tremendous amount of engineering work specific to simplifying and automating the use and management of this highly converged (multi-model) database, culminating in the release of Oracle Autonomous Database in 2018. At that time, Larry Ellison, Oracle's executive chairman and CTO went so far as to predict that this self-driving, self-healing database would eventually become the only database on offer from Oracle. Why then has Oracle introduced another database?

On the surface, the introduction of Oracle Autonomous JSON Database aligns nicely with what has become a perennial market trend toward specialization. There are databases built just to handle time series data as with IoT transactions or operational log analysis. There are databases designed to organize and analyze relationships between entities. And there are databases concerned solely with handling the stateless, real-time rigors of cloud-native software transactions. This last example describes Oracle's new Autonomous JSON Database, which specializes in storing, managing, and analyzing semi-structured documents, adhering to the JSON data-interchange format.

Typified by solutions such as MongoDB, Couchbase, Amazon DocumentDB, and Google Cloud Firebase, these document stores (also referred to as NoSQL databases because they use APIs rather than a SQL query language), have become extremely popular with software developers because of their simplicity and ease of use. Rather than having to model their objects into rows and columns, developers can stay within a format and way of thinking that is natural to them (the JSON document format).

Historically, this simplicity and flexibility came at a high cost because NoSQL databases often eschewed many of the safety measures available within SQL-centric databases such as transaction and data integrity, not to mention giving up SQL itself for data manipulation and query. For this reason, most relational databases have added some degree of support for semi-structured JSON documents, and many pure-play document databases incorporate relational database controls such

as atomicity, consistency, isolation, and durability (ACID). The result is a market rife with multi-model databases that look like document stores one day and relational databases the next, but function as both on any given day.

Two databases living as one

With this logic, it seems reasonable to suggest that enterprise buyers should simply invest in one big multi-model database. Doing so would greatly cut down on the management, maintenance, and resource spend associated with data siloed among many disconnected data repositories, as well as eliminate data fragmentation. However, it is important to note that many of these databases still have a long way to go in bringing together NoSQL and SQL benefits into one database that converges data types, workloads, and enterprise-class functionality that many companies rely on from Oracle Database. Most of these multi-model databases are by their very nature complex, demanding systems that are not always able to offer best-of-breed functionality for each and every available mode, including document store, graph analysis, and time series.

Enter Oracle Autonomous JSON Database. Built specifically for JSON developers, this new cloud database tries to deliver the best of both worlds by striking a balance between capability and complexity. Available immediately as a native JSON data store, Oracle Autonomous JSON Database seeks to provide full document storage functionality without imposing any of the complexities of operating, managing, tuning, or troubleshooting the data tier. Instead, Oracle hopes to give developers some peace of mind with a document store that "just works", leaving all of the underlying database management to Oracle's AI-informed automation. The idea is simple: free developers to focus on what they do best: build actual applications that sell.

How does Oracle Autonomous JSON Database do this? First, it provides create, read, update, and delete (CRUD) functions via both native SQL statements and a multi-language NoSQL-style document API. This API, called SODA (Simple Oracle Document Access), provides native API calls for major programming languages and can also be invoked via RESTful web services. In addition, the database's built-in command line interface (SQLcl) has been enhanced to not only parse SQL and PL/SQL but also SODA statements, creating a highly unified access method for developers, database admins, and data analysts alike. For example, database administrators can use PL/SQL to set up and execute stored procedures, and business analysts can use SQL to readily report across JSON document collections.

Of particular interest are Oracle's efforts to expand on the scale and scope of JSON itself. Because orthodox JSON uses standard text documents to convey information, it can present some performance limitations, primarily with indexing and searching very large JSON documents. For this reason, vendors including MongoDB and Oracle have created their own binary renditions of JSON. Just as MongoDB uses its own version (BSON), Oracle uses a native, binary representation of JSON data called OSON, which allows the vendor to efficiently parse, process, and analyze JSON data in-memory.

Regardless of interface, Oracle Autonomous JSON Database goes beyond SQL access to provide full ACID functionality by default. These transactions are started implicitly, can span multiple documents and collections, and have no size or time limit. This resolves developer challenges such as getting the wrong results, stale data, or having to think about consistency trade-offs. Staying on the path of simplicity, the database also provides a different type of index called JSON Search Index. It can index

an entire JSON document, making it easy for developers because they don't have to guess which attributes might be queried and set up the required indexing ahead of time. However, in the spirit of giving developers choice, the database also still provides the capability of secondary indexes if needed.

The self-managed database

If Oracle's new JSON database functioned like a traditional, standalone JSON database, these enhanced capabilities might create management complexities or impose expensive performance demands. Oracle Autonomous JSON Database is not a standalone database, however. It isn't even a new database.

Instead, Oracle Autonomous JSON Database is part of the Autonomous Database family and uses the same cloud infrastructure and database technology as Oracle Autonomous Transaction Processing and Oracle Autonomous Data Warehouse services. Oracle Autonomous JSON Database is a new workload type for Oracle Autonomous Database that allows IT buyers to stand up a JSON-specific database quickly and at a low cost while still enjoying the benefits found within Oracle Autonomous Database.

These benefits stem from ML-informed automation of security patches and software fixes, the common management routines, query optimization, even compute and memory resource allocation, and a pay-per-use model. As an example, Oracle Autonomous JSON Database uses dynamic and independent allocation of both storage and processing, which frees users from having to figure out ahead of time the most optimal configuration for future application requirements. Furthermore, these allocation units can be scaled up and down independently and immediately without interrupting the supporting application. This decoupling will enable developers to start small and grow their database as their applications and workloads grow or diminish. This can be done manually or via an auto-scale option that when enabled will allow the database to allocate compute resources by itself to best meet the performance needs for a given workload. Together, these capabilities work to make the database practically self-managing, lessening dependence on database administrators (DBAs) and freeing up more time for application development, a situation that will find favor with both JSON developers and IT administrators.

Building on this automaticity, Oracle can go well beyond targeting mature enterprise customers in need of hardened JSON development options. An aggressive subscription pricing model starting at $2.74 per hour for on-demand access, availability via Oracle's always-free tier, and the option to upgrade to the full Oracle Autonomous Transaction Processing with a single click makes this new database a tempting entry point for customers, startups, and developers just breaking into cloud-native development practices. As an example, customers will pay $240 per month per Oracle Compute Unit (OCPU) for pure JSON access, a price node widely off other cloud JSON database offerings.

Caveats and opportunities

Oracle's new database is not without drawbacks. As with most managed cloud services, Oracle Autonomous JSON Database demands a degree of vendor lock-in because Oracle Autonomous Database functions exclusively as a part of Oracle Cloud Infrastructure, Exadata Cloud@Customer, and Dedicated Region Cloud@Customer. The trade-off revolves around capability and convenience.

Wth Oracle's SODA APIs, for example, JSON support and converged database capabilities are also supported within the Oracle Database. Likewise, Oracle Autonomous JSON Database integrates natively with Oracle's flagship analytics product, Oracle Analytics Cloud, creating a highly synergistic environment for customers committed to the Oracle portfolio.

Even with these points of integration, Oracle's new database has some room left to grow. For instance, Oracle's SODA APIs are limited to Java, JavaScript/Node.js, Python, and C/C++. Fortunately, these languages drive a great deal of cloud-native development but this still leaves out native access for influential languages including C#, Haskell, Rust, Perl, PHP, Ruby, and Scala. For companies rooted in these languages, Oracle does offer SODA via RESTful web services with its Oracle REST Data Services (ORDS) product. In addition, all these languages allow for embedded SQL calls as another direct point of ingress, pointing to a growing market reliance on SQL as the true lingua franca for data processing and analysis.

Despite these constraints and concerns, Omdia believes that Oracle Autonomous JSON Database represents a unique advancement within the data and analytics marketplace. By building a single, multi-model database, automating the management of that database, and then breaking out best-of-breed functionality, Oracle has engineered a seemingly future-proof data platform capable of meeting customers where they are and making life easier for developers and accelerating the pace of application development, both now and in the future.

To do this, Oracle is plying this data platform by including the Oracle APEX low-code/no-code development interface with Oracle Autonomous JSON Database. This move will certainly find favor with developers of all levels looking to rapidly prototype new applications, and the company is already eyeing further best-of-breed databases and database tools that inherit all of the benefits found within the Oracle Autonomous Database. Clearly, Oracle's focus on the developer community has intensified in recent months and its investments in the machine learning-powered Autonomous Database look to give the company some leverage over best-of-breed rivals as it aims to reach a whole new generation of customers.

Appendix

Further reading

"MongoDB aims to break down data silos for developers," INT002-000294 (June 2020)

"Oracle taps into low-code/no-code development to help fight COVID-19,"INT002-000289 (May 2020)

"TigerGraph readies no-code graph analysis database," INT002-000277 (March 2020)

"Oracle looks to take the "Oops!" out of data science," INT002-000273 (February 2020)

2020 Trends to Watch: Analytics and Data Management, INT002-000272 (February 2020)

Author

Bradley Shimmin, Chief Analyst, AI Platforms, Data & Analytics

askananalyst@omdia.com

Citation Policy

Request external citation and usage of Omdia research and data via citations@omdia.com.

Omdia Consulting

We hope that this analysis will help you make informed and imaginative business decisions. If you have further requirements, Omdia's consulting team may be able to help you. For more information about Omdia's consulting capabilities, please contact us directly at consulting@omdia.com.

Copyright notice and disclaimer