Harsha Vardhan Reddy Kavuluri
WISSEN Infotech INC, United States

Received: 11-07-2020; Revised: 22-08-2020; Accepted: 05-10-2020

Oracle Autonomous Database vs Open-Source Solutions: An Overview

Abstract
In the contemporary world of data management, organizations are left with a significant decision of choosing between the closed cloud-based self-driven database solutions like Oracle Autonomous Database or the open-source solutions available in large numbers such as PostgreSQL, MySQL, and MongoDB. This paper offers a comparative strategic analysis of Oracle's autonomous versions of the systems with open-source ones regarding performance, scalability, cost, security, operational automation, and the ability to address the needs of an enterprise-level company. With respect to each of the options, we consider its long-term effects using evaluation and examples in various sectors. This paper aims to provide valuable insights to IT managers facing the issue of the choice of databases and the key concepts of artificial intelligence, machine learning, and cloud-native architecture of data platforms. Based on empirical studies, simulation, and architectural perspectives are the major components of this broad work.

Keywords: Oracle Autonomous Database, Open-Source Databases, PostgreSQL, MySQL, MongoDB, Automation, Scalability, Data Security, Cost Analysis.

1. Introduction

Database systems serve as the basis for almost every present-day IT application. For most enterprises, transforming databases from on-premises legacy systems to cloud-based and self-driving systems is the biggest change in the IT environment. [1-3] The Oracle Autonomous Database is well recognized for its features, such as self-driving, self-securing, and self-repairing, while open-source databases provide flexibility and cost-effectiveness.

1.1. Need for Strategic Decision-Making in DBMS Selection

Choosing the right Database Management System (DBMS) is one of the important suppliers' decisions an organization has towards the choice. In view of this, the DBMS fits the organisational requirement for data and the key factor determined by its performance, scalability, cost, and security. In the modern world, decision-making over data happens to be quite complex and may involve a number of considerations to suit the organization's strategic direction. Since an organization's data needs change over a period of time, it is significant to have a DBMS that will be valuable to the existing and future needs of the organization. The following are important factors that make choosing a DBMS a critical decision that must be made strategically.

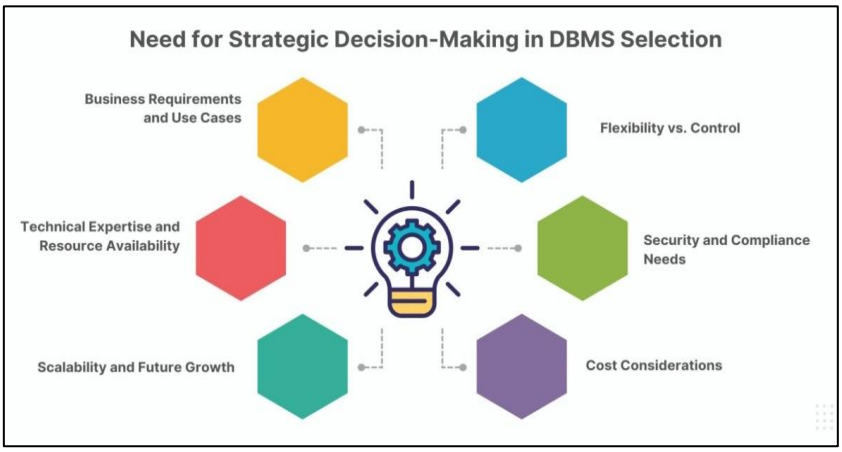

Figure 1: Need for Strategic Decision-Making in DBMS Selection

• Business Requirements and Use Cases: The first stage in DBMS selection is to link the adopted database form with the business needs and application. The various DBMSs have their strengths and weaknesses, and the database used in an organization must be determined by the type of data the organization deals with. For instance, Oracle or PostgreSQL databases are better suited to structured data and transactional workloads, while non-SQL databases such as MongoDB or Cassandra would be well-applied for managing unstructured or semi-structured data, which would require a good level of horizontal scalability. Depending on its use case, an organization may require real-time data processing, high availability, analytics, or large amounts of data storage. Using the needs mentioned above to choose the DBMS will guarantee effectiveness and efficiency in managing such databases.

• Technical Expertise and Resource Availability: Another determining factor in DBMS selection is reviewing the organization's technical competency. Some other types of databases, like Oracle ADW or Microsoft SQL Server, come with automated management, and it is quite useful for companies who cannot invest in DBAs or do not possess IT experts in the organization. These perform vital roles like patching, scaling, and tuning the system or hardware on the computer. On the other hand, types like PostgreSQL or MySQL allow more customization, but managing such types of databases involves a professional who has to perform database maintenance, optimization, and performance optimization.

• Scalability and Future Growth: Thus, using a DBMS in a business environment enables scalability and caters to the business's ever-growing requirements. It is crucial to select the appropriate DBMS to demonstrate good scalability for the organization's sustainable growth. Currently, cloud-native DBMSs such as the Oracle ADW come with a highly flexible design in terms of scalability, where an organisation can easily increase or reduce the use of various resources to better fit the current and/or future demands without needing major application architecting. At the same time, open-source databases such as PostgreSQL may need more time to be horizontally scalable, although increasing volumes and transaction rates will eventually become a problem.

• Cost Considerations: When purchasing a new DBMS, the total cost of ownership is the next big criterion that the organizations should consider. They include imposing costs of licenses, one-time maintenance costs, personnel costs, and incurred infrastructure. Costs While there are various other solutions available, proprietary solutions such as Oracle ADW may have higher setup costs, and these costs could be easily overcome because they reduce data management administration by automating the whole process. On the other hand, proprietary systems such as Oracle have no initial cost through licensing but will definitely cost more in terms of resources in its installation, management, and tuning. Organizations with a small budget may prefer open-source databases. However, the further costs of their functioning should be taken into consideration.

• Security and Compliance Needs: Data security and privacy have become one of the most crucial components in any company today, so choosing a DBMS that meets the regulation requirements set by GDPR, HIPAA, or ISO 27001 is vital. The security of Oracle ADW is also a primary advantage because it has features, such as data encryption, automatically applying patches, and access controls, that allow organizations to solve compliance problems very fast. Forces such as PostgreSQL and MongoDB are free, but extra installations and additional tools are still needed to ensure security compliance like those of Wavemaker. This can bring in new layers of potential errors and misconfiguration and may pose even more of a threat in a secure and critical environment.

• Flexibility vs. Control: Both flexibility and control are important deciding factors in selecting DBMS, but inherent conflict exists between the two. They have more flexibility as they offer organizations the leeway to include the design of a certain database type based on its requirements.

However, it creates a lack of structure and requires experienced managers to help organize the system. Their major advantage is the higher level of flexibility and more opportunities for primary activities such as tuning and security due to mechanization in the case of proprietary solutions like Oracle ADW. There is always a question that arises in organizations, which is more important, the flexibility of the open source or administrative convenience and automated control over all processes provided by proprietary systems.

1.2. Oracle Autonomous Database vs Open-Source Solutions

Oracle Autonomous Database (ADW), PostgreSQL, and MongoDB are basic choices for any organization because of cost, performance, scale, and available personnel. As a cloud-native, fully managed database, Oracle ADW eliminates the operational overhead in most cases. Some features include patching, scaling, and performance tuning, eliminating the need for organizations to handle complex database operations. Oracle ADW is particularly well suited to large-scale organizations or businesses that need to function in secure compliance, address data protection rules like GDPR or HIPAA, and performance. [4,5] Nonetheless, this has a disadvantage in terms of cost since the solution entails certain licensing fees, and the fact that it is a proprietary product might not likely favor small establishments or firms that are hard-pressed on budget. However, PostgreSQL and MongoDB, for instance, remain open-source solutions that greatly allow the caller versatility. These systems are available free of charge, a factor that makes them ideal for organizations with a number of financial muscles. Open-source databases are more effective when such values as flexibility and the possibility to adjust them to particular tasks and needs are crucial, for instance, in development circumstances or while working with unarranged data (MongoDB). However, these databases are generally less advanced and require more human interaction and work from a computer expert. Some tasks like performance tuning, backup management, and security configurations require the services and thus have to be done in-house, which implies more operational costs and a skilled labor force in terms of database administrators. Hence, open sources are cheaper than licensed ones but may be harder to deploy and maintain as the organizations' need for data expands. In conclusion, it is crucial to note that choosing Oracle ADW in preference to open-source systems is associated with the size of the organization's financial resources, the technical equipment of the enterprise, and the necessary number of controlled parameters. Oracle ADW is ideal for enterprises given the technological enablement, whereas open-source databases help organizations that do not mind managing the process manually and, above all, spend on it.

2. Literature Survey

2.1. Oracle Autonomous Database

Coming to the market in late 2017, it is one of the latest solutions introduced by Oracle Corporation.

They use some of the most efficient machine learning algorithms to perform tasks like patching, performance tuning, backup, and deploying. [6-10] This helps organizations to minimize manual interferences and system downtime and increase the system availability. Information about the use of the platform and its capabilities can be described using sources such as Oracle company reports and whitepapers produced in 2020-2023 and articles from leading industry analysts Gartner and Forrester regarding the platform's strategic evolution, scalability, and performance in real-world companies.

2.2. Open-Source Databases

The examples of such databases are free of charge, highly customizable, can be supported by various communities, and are ideal for both startup and giant companies. It is highly extensible and complies with the ACID properties, thus being appropriate for many highly transactional applications. MySQL, currently owned by Oracle, is also a well-known open-source database with multiple robust replication capabilities. MongoDB is quite different from MySQL. It has a schema-less JSON-based document store with greater flexibility that is preferred by developers fitting into projects with unstructured or semi-structured data. Still, every stated database fulfills certain purposes and has benefits and drawbacks.

2.3. Comparative Studies

Several research works have been undertaken in the past to undertake a comprehensive comparison of proprietary and open-source databases based on the following parameters. Oracle Autonomous Data Warehouse (ADW) is an owned system with a high automation degree integrated with native reinforcement decision-making. While PostgreSQL and MySQL are open sources, these databases may need manual efforts to change performance and do not comprise the built-in ML features. It is not a total cloud platform with less NAASQ for traditional data relationships but is more effective for other new structure data relationships. It should also be noted that while all the mentioned platforms offer licensing models, levels of automation, and community support, their choice directly depends on the nature of the project and organizational objectives.

2.4. Gaps in Existing Research

However, some gaps are still found in comparative studies, as listed below. There is a shortage of quantitative performance measures whenever a database is updated and queried at the same time. Furthermore, there is a relatively low tendency to compare long-term cost structures necessary for identifying the notion of a total cost of ownership in the case of cloud environments. Additionally, a lack of studies has been reported on security and compliance by different industries, such as the healthcare and financial sectors, which require compliance frameworks like HIPAA and GDPR. These gaps imply the need for systematic and empirical reviews regarding database selection to support decision-making.

3. Methodology

3.1. Comparative Framework

• Cost Analysis Model: The Cost Analysis Model is designed to assess the Total Cost of Ownership which is the expenses on various database platforms to be incurred at an initial stage and in the further course of their utilization. This is the cost of license per processor, by infrastructure, whether local or cloud-based, support and upkeep, and other overhead charges such as employees' wages and equipment maintenance. [11-15] As for the licensing feature, it is also important to note that even though the databases are open sources, there may be other overhead expenses, such as customization, tuning, and use of third parties. On the other hand, some purchases, such as Oracle Autonomous Database, may have higher purchase costs but can benefit from automation, minimizing maintenance needs, and personnel training. It assists an organization in approximating contracts and expenditures to ensure that the investment does not surpass what the company was willing and able to spend.

• Performance Index: The Performance Index is a relative measurement tool that provides a rating of particular DBMSs by evaluating parameters like response time, TPS, transaction rate, and changeability of the type under load. This index combines OLTP and OLAP measures to show how both systems cope with a mixed stream of transactions. Benchmarks can be based on industrial ones, such as TPC-C or TPC-H benchmarks, or based on workload-oriented simulations. Well-defined quantifiable measures allow for comparison and identification of the more appropriate systems for demand environments.

3.2. Simulation Environment

As a result of carrying out a fair and integral comparison of the specified database systems, a controlled simulation was created using Amazon Web Services and Oracle Cloud Infrastructure. PostgreSQL and MongoDB were also used, both of which run on AWS EC2 instances, thus allowing much control over accurate computations and storage space in line with various workloads. These instances were designed for database operations and have High IOPS SSD storage with multi-core vCPU to give a real experience at the enterprise level. Another advantage of AWS was the ability to test different types of instances to determine the effects of resource utilization on their performance. For both tests, Oracle Cloud Infrastructure was used, especially for Oracle Autonomous Data Warehouse (ADW), as it is a native service of Oracle Cloud. Since OCI offered all the resources required for developing, testing, deploying, and operating ADW, it allowed the team to fully operationalize the solution and test all aspects of ADW self-tuning, self-scaling, and machine learning automation. Due to how ADW nominalizes much of the underlay infrastructure, the primary concern was assessing the query performance, response time, and resource utilization under different loads. TPC-C and TPC-H testing were employed to conduct the performance test. TPC-C mimics transaction processing procedures more specific to retail or internet business needs, where more read-and-write operations could be involved.

TPC-H targeted transactional processing with large queries over comprehensive data for decision support, similar to a decision support system. These benchmarks offered a clear framework that can be reused to compare and contrast test results of various database systems to understand their performance against, for, and between novel transaction processing, data analysis, and their mixture. Some of the obtained performance evaluation parameters include overall throughput, query response time, CPU usage, and scalability. By not exaggerating the workload and the resources to allocate, they have been able to create a levelled playing ground when comparing Postgresql, MongoDB, and Oracle ADW to be able to determine its performance in the real world with very good insight into the amount of cost one has to be ready to incur.

3.3 Evaluation Metrics

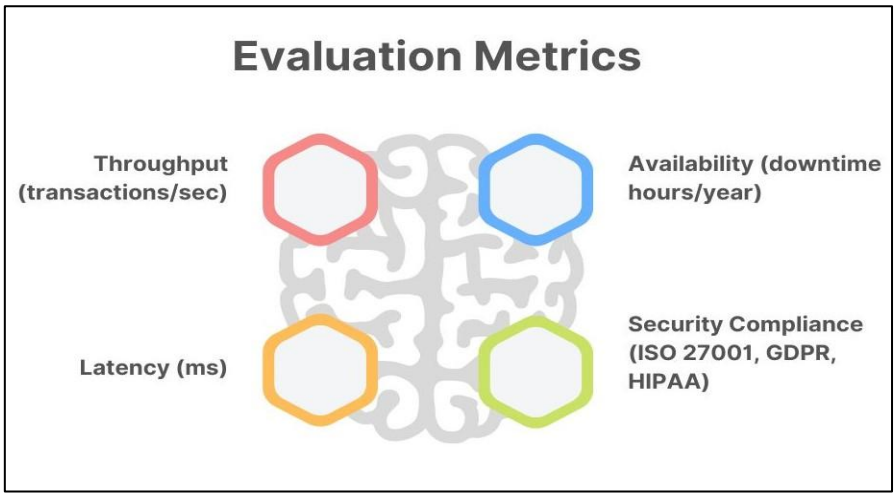

Figure 2: Evaluation Metrics

• Throughput (transactions/sec): Throughput also implies the quantities of transactions processed by a database system within one second. It is quite important, particularly when handling a high number of requests for transactions, such as in m-commerce or financial applications. [16-19] A higher throughput means the system can run through many operations within the shortest time space possible. In the evaluation, throughput was evaluated based on OLTP (Transaction) and OLAP (Analysis) to determine the performance of each database in terms of operational loads.

• Latency (ms): Latency is the time it takes to process one single transaction or query, normally expressed in milliseconds. Less delay is important as client-side interactions would require minimal fuzz from the client-server base app. In this evaluation, three main numbers were measured for each database: the average response time and the peak response time over a series of queries and other operations, giving information on how fast each database could respond to a load. Current measurements indicate periods during which valuable information is searched and processed, which can be useful for applications requiring low latency.

• Availability (downtime hours/year): Availability measures how robust a database system is by

computing the sum of time that the system is out of order for a whole year. This is important for any business-critical application, wherein any failure can directly affect the business through revenue loss or lack of data synchronization. As for availability, there are parameters like failover capabilities, auto-recovery, and infrastructure strength. This metric concerned the stated uptime they provided and the observed uptimes during the emulation of various failures in the databases concerned.

• Security Compliance (ISO 27001, GDPR, HIPAA): Security compliance measures the degree to which a database system meets the standard from key regulatory frameworks such as the ISO 27001 regarding information security, GDPR on data protection, and HIPAA on healthcare data. This one evaluates whether the database has incorporated common security features such as encryption, access control, and auditing and compliance reports. This assignment is particularly useful for organizations with regulated market businesses.

3.4. Research Methodology Flowchart

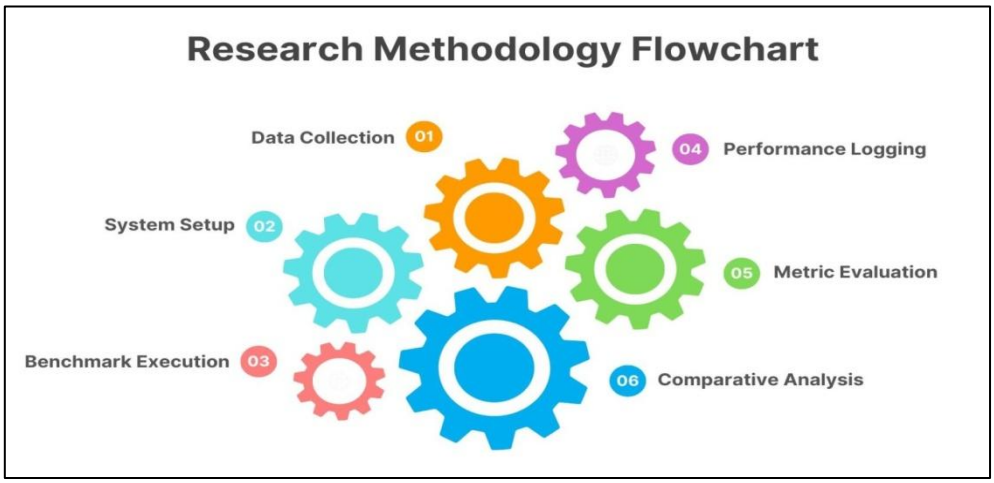

Figure 3: Research Methodology Flowchart

• Data Collection: This involves the gathering of data, which may include system documentation documents, specified configuration procedures, benchmarking norms, and usage practices. It also guarantees that benchmarking tests are practical and establish the procedures to be followed. It also addresses the specification of the appropriate pre-standard and/or standard workload models (such as TPC-C and TPC-H), resources needed, and compliance factors required to match real-life business environments. Collecting proper data is necessary for developing credible and replicable outcomes.

• System Setup: In this phase, the three selected database systems, namely Oracle ADW, PostgreSQL, and MongoDB, are installed in their cloud platforms, Oracle Cloud Infrastructure and AWS EC2, respectively. All of the above systems are set in a way that provides performance benchmarks, including storage, memory, CPU, and security settings, among others. The setup they provide learners is developed to the most minimal degree so that few uncontrollable factors may influence the test results. This step is very important to ensure that there is no bias and that comparative

analysis considers what has been achieved.

• Benchmark Execution: TPC-C for the transactional processing benchmark and TPC-H for the analytical processing benchmark are performed once the systems are installed. These benchmarks imitate real SuperFabrics World workloads and stress the systems under various conditions, including user load and query loads. The data collection and execution are carried out several times to increase the sample size to represent the variability and to ensure sound results are obtained consistently.

• Performance Logging: While and after the benchmark is run, paginator logs are generated with aspects such as the number of transactions per second, the query response time, CPU and memory utilization, and percentage error. These logs serve as the main database used in the following analysis. Logging utilities aid in logging by providing pre-developed scripts that can be employed to capture specific events or variables of a system under test and document it systematically all through the test phase.

• Metric Evaluation: This step requires analyzing the performance logs using parameters such as throughput, delay, availability, and whether or not it complies with the set standards. All of them are calculated regarding the considered workload type and the system configurations to draw relevant performance insights. In order to understand the data more effectively, some statistical tools are used in the process and may be used to display the findings.

• Comparative Analysis: In the final phase, the test metrics are analyzed for all systems to compare them and show the strong and weak points of the specific technique and the pros and cons of choice. They are discussed with regard to various application issues like cost, performance, and regulatory factors. This research also presents recommendations to the decision-makers planning to implement Oracle ADW, PostgreSQL, or MongoDB by highlighting the similarities and differences between the three.

4. Results and Discussion

4.1. Database Performance as Percentage

Table 1: Database Performance in Percentage

Database	Avg. Read Latency	Avg. Write Latency	Transactions/sec
Oracle ADW	48.84%	50.00%	100%
PostgreSQL	100%	100%	83.67%
MongoDB	81.40%	84.00%	90.82%

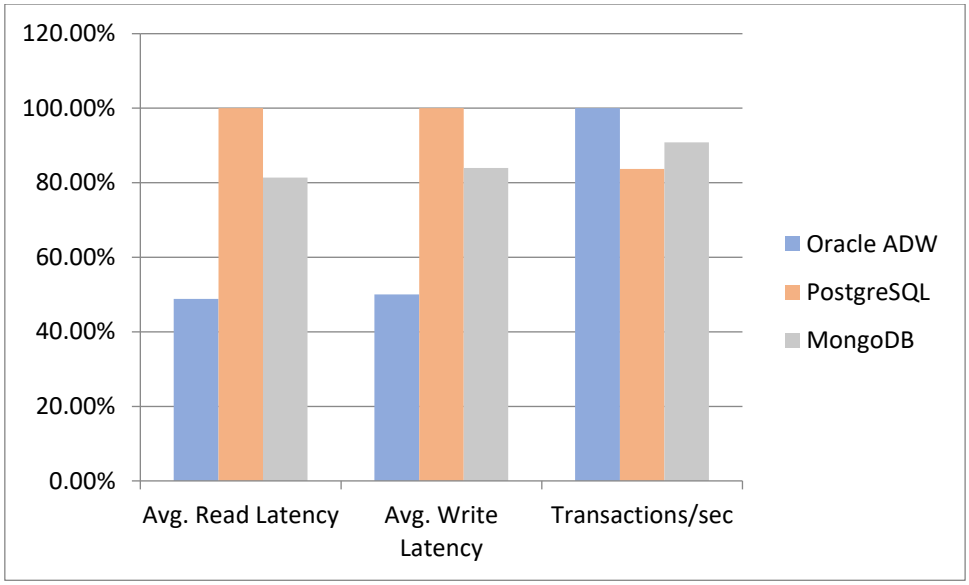

Figure 5: Graph representing Database Performance in Percentage

• Oracle ADW: Concerning read latency, write latency, and TPS, Oracle Autonomous Data Warehouse (ADW) outperforms the two other databases. It provides the lowest figure for both read and write latency, or the ability to work with data access operations, with read latency being 48.84% lower than PostgreSQL's and write latency, at 50% less compared to PostgreSQL. Also, Oracle ADW stands out from the rest in terms of processing capacity since it processes the highest number of first transactions per second at 100%. It can be attributed to the fact that one is operated by Machine learning so that its execution does not require extra human interferences.

• PostgreSQL: Among all the compared SQL RDBMS, PostgreSQL is the most feature-rich open-source relational database system; therefore, both its read and write latencies are the highest and amount to 100 percent. This means that PostgreSQL would take longer latency than Oracle ADW and MongoDB. The increased latency can be explained by the time spent manually tuning the queries and the lack of built-in facilities available in Oracle ADW. Thus, PostgreSQL has a relatively low TPS, 83.67%, compared to the TPS of Oracle ADW. Nevertheless, PostgreSQL conveniently obeys all the standards of ACID and is considered to be highly dependable. However, it demands more time and certain efforts to be made to maximize performance.

• MongoDB: Among the tested NoSQL databases, MongoDB is sufficiently efficient regarding transactions per second; it hits 90.82% of Oracle ADW. Nonetheless, its read latency and write latency are slower than Oracle ADW and thus recorded a score of 81.40% for read latency and 84.00% for write latency, while the PostgreSQL benchmark was set at 100%. MongoDB is particularly suitable for scenarios with many clients and numerous data fields where data may be of various structures. Although there are some minor shortcomings in low latency, BigQuery is highly efficient in distributed architecture and data processing, making it a strong competitor for Oracle ADW in certain situations.

4.2. Cost Efficiency

Enterprise cost is a special efficiency that prevails in Oracle ADW, PostgreSQL, and MongoDB, as these solutions differ in costs and purposes for various organizations. Since Oracle ADW is proprietary software, its cost is licensed regarding software and infrastructure costs at the initial stages. However, due to its integrated automation system, its persistence charges of operation are much less. Automated patching, tuning, scaling, and other options reduce the requirements of the people involved, which saves capital investments for maintenance activities in the long run. This makes Oracle ADW suitable for any organization interested in a long-term license to minimize the time and efforts spent on myriad administrative tasks at the cost of the first-time investment. PostgreSQL, on the other hand, is an open-source database with no license cost implications. However, it does not enjoy tight integration with built-in automation, meaning that much work has to go into setting up, maintaining, and tweaking the DevOps environment. PostgreSQL requires specialized personnel for patching, backups, and system optimization of the database, thus raising overall personnel costs. This makes PostgreSQL a cheap technique for organizations possessing preliminary technical and long-term management resources. The fact that PostgreSQL is not bought through a licensing model is a plus for businesses that seek to save money; nevertheless, its DevOps and tuning fees add up, particularly in larger contexts. MongoDB is the same as Cassandra, an open-source solution designed for horizontal scalability through sharding, which can be used where a large amount of data has to be distributed between several servers. However, there could be high costs associated with large scale based on infrastructure costs, especially in handling large volumes of data. Although using MongoDB for the distributed application is quite beneficial in terms of flexibility and costs, large-scale sharded clusters significantly cause management problems and underlying infrastructure costs. This makes MongoDB particularly suitable for businesses that require scalability, but infrastructure and operations expenses need to be taken under preview as the system scales up.

4.3. Security and Compliance

In terms of security and compliance, Oracle ADW is a reliable tool with out-of-the-box functionalities that help companies deal with various requirements easily. Advanced data encryption at rest and in motion is in place in Oracle ADW. Thus, the security of the data is preserved during the processing. Furthermore, the integrated patch management feature also prevents the system from lagging in the updates, which may be fatal if an opponent attempts to exploit a vulnerability. It can help achieve high levels of automation for easily meeting various regulatory standards like ISO 27001, GDPR, and HIPAA and makes it very suitable for companies who want to meet their compliance needs without compromising on the time and effort required to manually manage very complex security processes. Therefore, Oracle ADW serves well for business needs, especially those dealing with sensitive or regulated data, because it reduces the chances of breaking the regulatory rules or digitally getting hacked. In this comparison, PostgreSQL and MongoDB are open-source databases; hence, security and

compliance have to be configured by hand in addition to linking other tools and applications. In both cases, basic security measures, such as restricting access and identifying users, are provided to meet the full potential of the databases with regard to compliance with the GDPR or HIPAA requirements, but more should be done. For instance, the requirements regarding encryption may have to be set by an organization, auditing features may have to be configured, and access rules must be set to conform to the requirements of such industries. This additional configuration is not only a plus but can lead to some problem like misconfiguration, thus exposing the team to the risk of non-compliance or even data breach. Regarding the database, open-source in particular, relies heavily on the automated patching, yet manual patching demands a lot of monitoring and updating at the user end, besides the fact that it is more prone to human delays.

4.4. Automation & Maintenance

Oracle's Autonomous Database (ADW) gains high automation and has more benefits than classic database management systems. Thanks to the automatic patching feature, backup, performance tuning, and scaling, Oracle ADW does not pose a problem in management in terms of the overhead usually experienced in handling databases. These features of automation help to free the need for trained database administrators to perform routine tasks such as applying patches or optimizing the system during normal working cycles. This leads to lower constant maintenance costs and better utilization of the resources as most of the work is done by the database itself. On the same note, work automation helps reduce human errors that may put the entire system on a wrong footing and require a lot of time to rectify. For organizations, this implies that there is not much need to employ many and many highly skilled IT personnel, thereby freeing resources for other tasks that will improve organizational effectiveness. Oracle ADW is, therefore, suitable for organizations searching for a system that can solve many tasks independently and free up a lot of time for the organization's business. However, PostgreSQL and MongoDB require much more effort than auto-generating. Since it is so flexible and allows granular control of the system's functionality, PostgreSQL is something that needs to be administered by highly skilled personnel, who are also responsible for performance optimization, making backups, and applying patches. Database configuration and tuning also play a crucial role in administrators' database use, especially with the increasing scope. Also, in terms of flexibility and scalability, MongoDB can be as complicated as other databases; it has scalability approaches like data sharding, replication, etc., which can be done only manually. As the size of the database increases, these tasks can become more complicated, which is why a lot of planning needs to be done to ensure that things like data integrity or performance troubles do not arise. Oracle ADW shares the same trade-off between higher convenience and lower controllability as any other closed-source software compared to open-source ones such as PostgreSQL and MongoDB. Hence, Oracle ADW is a fully managed cloud application with a very high automation level, which would suit organizations with less human resources in the IT sector. However, PostgreSQL and MongoDB have more liberty and maneuver

options. Still, at the same time, they are relatively complex and require more individual management, making them suitable for teams that already have vast experience in database administration work and its subsequent maintenance. Therefore, it depends on the types of workloads of the organization and the budgets they have set to ensure they manage their databases effectively.

5. Conclusion

This means the decision to go for Oracle Autonomous Database (ADW) or another open-source database such as PostgreSQL and MongoDB depends on the firm's suitability, capabilities, and cost. Oracle ADW is capable of extensive automation, enhanced security measures, and high availability with minimal or no control from human resources. The automated patch, scaling, tuning, and backup management capabilities free businesses' time and resources for the core part of work and innovation. Oracle ADW is highly beneficial to large-scale organizations because it eliminates operational costs, reduces the chances of human error, and ensures compliance with judging standards tests. This makes it highly appealing to industries searching for an automatic procedure that will not demand a lot of staff input and will ultimately save substantial amounts of money in the long run. However, several other databases, like PostgreSQL and MongoDB, have completely open-source systems and carry certain advantages. ACID properties are quite strong in PostgreSQL; the system is highly configurable and has a wide base of users and developers. It suits organizations where certain parameters need to be fiddled with and fine-tuned. Besides, it makes sense for businesses with a strong technical background to implement it and reap cost benefits. But, it becomes challenging to perform performance tuning, patching, backups, and other jobs that may add to the long-term operating expenses and exert more strain on the qualified administrators becomes challenging. Like SQL, MongoDB is also implemented in the NoSQL model of data, which is best used in unorderly type data and best fit to be scaled horizontally. Nonetheless, it heavily relies on manual work, particularly as the network grows, and organizations need to be ready for a higher level of difficulty in functions like data partitioning and replication.

Specifically, for organizations that principally invest in cost containment and community contribution, open-source database solutions are most common for new start-ups, the establishment of academic circles, and development-oriented settings that relinquish flexibility and adaptability. The popularity of PostgreSQL and MongoDB also means the large open-source community provides strong support, and businesses can find solutions and recommendations for best practices without necessarily having to pay the steep price for the license. Lastly, organizations can follow parallel strategies of using Oracle ADW and open-source databases for their multiple-cloud solutions. This makes it easier for business organizations to determine tool efficiency for certain workloads and select the best tool that fits the general need. In such situations, it makes sense to leverage more platform-robust and business-critical

components like the Oracle ADW, while other parts of a business application may be accommodated with more versatile open-sourced databases and more concurring with its development and data handling approach. Therefore, each option's decision will lie in how it will benefit or harm the organisation in terms of automation, costs, flexibility, and security based on current and future needs.

References

1. Greenwald, R., Stackowiak, R., & Stern, J. (2013). Oracle Essentials: Oracle database 12c. " O'Reilly Media, Inc.".
2. Mateen, A., Raza, B., Awais, M. M., & Sher, M. (2011). Selection of the Best DBMS: A Fuzzy based Multi-objective Decision-Making Approach. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1).
3. Kavale, S. (2012). The role of data in the strategic decision-making process. International Journal of Current Research, 4(04), 001-007.
4. Png, A., & Dimanche, L. (2020). Getting Started with Oracle Cloud Free Tier.
5. Kavuluri, H. V. R. (2024). Oracle vs PostgreSQL: Cost Analysis for Enterprises Adopting Cloud-Native Strategies. International Journal of Emerging Research in Engineering and Technology, 5(3), 22-31.
6. Gubala Hari Babu, L. S., & Dodla, S. N. S. (2024). Comparative Analysis of Oracle and MySQL Databases: A Study on Query Execution and Scalability.
7. Petri, G. (2005). A comparison of Oracle and MYSQL. Select Journal, 1.
8. Bradshaw, S., Brazil, E., & Chodorow, K. (2019). MongoDB: the definitive guide: powerful and scalable data storage. " O'Reilly Media, Inc.".
9. Pavlo, A., Angulo, G., Arulraj, J., Lin, H., Lin, J., Ma, L., ... & Zhang, T. (2017, January). Self-Driving Database Management Systems. In CIDR (Vol. 4, p. 1).
10. Egberts, A. (2017). The oracle problem analysis of how blockchain oracles undermine the advantages of decentralized ledger systems. Available at SSRN 3382343.
11. Png, A., & Helskyaho, H. (2022). Oracle machine learning in autonomous database. In Extending Oracle Application Express with Oracle Cloud Features: A Guide to Enhancing APEX Web Applications with Cloud-Native and Machine Learning Technologies (pp. 139-191). Berkeley, CA: Apress.
12. Helskyaho, H., Yu, J., Yu, K., Helskyaho, H., Yu, J., & Yu, K. (2021). Oracle Autonomous Database for Machine Learning. Machine Learning for Oracle Database Professionals: Deploying Model-Driven Applications and Automation Pipelines, 97-133.
13. Li, G., Zhou, X., Sun, J., Yu, X., Han, Y., Jin, L., ... & Li, S. (2021). Openness: An autonomous database system. Proceedings of the VLDB Endowment, 14(12), 3028-3042.
14. Png, A., Demanche, L., Png, A., & Demanche, L. (2020). Autonomous Databases. Getting

Started with Oracle Cloud Free Tier: Create Modern Web Applications Using Always Free Resources, 107-136.

15. Raza, B., Mateen, A., Sher, M., Awais, M. M., & Hussain, T. (2010, February). Autonomicity in Oracle database management system. In 2010 International Conference on Data Storage and Data Engineering (pp. 296-300). IEEE.
16. Sharma, B. M., Krishnakumar, K. M., & Panda, R. (2022). Oracle Autonomous Database in Enterprise Architecture: Utilize Oracle Cloud Infrastructure Autonomous Databases for better consolidation, automation, and security.
17. Litwin, W., Mark, L., & Roussopoulos, N. (1990). Interoperability of multiple autonomous databases. ACM Computing Surveys (CSUR), 22(3), 267-293.
18. Bader, A., Kopp, O., & Falkenthal, M. (2017). Survey and comparison of open source time series databases. In Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband (pp. 249-268). Gesellschaft für Informatik eV.
19. Natti, M. (2023). Migrating from Oracle to PostgreSQL: Leveraging Open-Source to Reduce Database Costs and Enhance Flexibility. The Eastasouth Journal of Information System and Computer Science, 1(02), 109-112.
20. Mannino, M. (2022). Database Design, Query Formulation, and Administration: Using Oracle and PostgreSQL. Sage Publications.