Assignment 4: Web Database Applications

<Task 2: Analyse different database platform report>

Student Name: Keewoo Moon

Student Number: s3831370

In this research report, I will analyse and compare two database platforms: Oracle, which was used in the implementation of the Computerised Voting application in assignment 1, and MongoDB, which was used in the implementation of the Airbnb-lite application in assignment 2. These two applications have different system requirements and use cases, and it is essential to evaluate their advantages and disadvantages in various aspects.

1. Query Performance

The query performance of Oracle and MongoDB diverges significantly due to their distinct database models. Oracle, as a relational database, is particularly proficient at efficiently managing complex SQL queries, making it the preferred choice for structured data applications like the Computerised Voting System. Oracle's SQL engine excels at handling intricate queries and aggregating large datasets, delivering swift results through well-structured and indexed data.

Conversely, MongoDB, a NoSQL database, excels in scenarios involving semi-structured or unstructured data, such as Airbnb-Lite's document-centric model. It offers superior performance when querying and aggregating substantial amounts of unstructured data, like Airbnb listings. While MongoDB provides flexibility in querying, it may not match Oracle's efficiency when dealing with intricate joins and aggregations. In summary, Oracle, a traditional RDBMS, is ideal for handling complex and structured queries using SQL, while MongoDB, a NoSQL database, excels in scenarios demanding flexibility and performance with semi-structured or unstructured data.

2. Resource Requirements

The resource requirements of Oracle, a relational database system, are quite high. It demands substantial disc space due to its structured and index-driven architecture, which can be advantageous for compact data scenarios but leads to significant disc consumption with extensive datasets. Oracle benefits from abundant RAM for caching frequently accessed data and indexes, but this means it may require substantial memory resources. Moreover, its CPU usage can be intense, especially for complex queries and transaction management. Distributed configurations can demand substantial network bandwidth, making a robust network infrastructure necessary.

On the other hand, MongoDB, a NoSQL database, has more resource-efficient requirements. It consumes less disc space, particularly for semi-structured or unstructured data. MongoDB's memory and CPU requirements are generally more modest than Oracle's, thanks to its distributed and horizontally scalable architecture. It's also more network-efficient in scenarios requiring data distribution across multiple servers. In summary, Oracle is suitable for data-intensive applications prioritising security and data integrity, while MongoDB is a practical choice for flexible, scalable applications with an emphasis on resource optimisation.

3. Security Issues

Both Oracle and MongoDB can be secured, but the approach differs. Oracle, being a well-established relational database, has robust security features, including role-based access control and advanced

authentication methods to prevent SQL injection and other vulnerabilities. However, security implementation can be complex and may require expertise.

In contrast, MongoDB in its initial setup may not have strong security mechanisms, and it is essential to configure authentication and authorization correctly. In a real application like Airbnb-lite, security would need to be carefully managed, and the default security settings of MongoDB would not suffice, so that in the case of MongoDB, security implementation may be less straightforward and can be a potential disadvantage.

4. ACID Properties and Transaction Control

Oracle, being a relational database, adheres to the ACID properties (Atomicity, Consistency, Isolation, Durability) and provides robust transaction control, making it highly suitable for applications where data integrity and consistency are paramount, such as the Computerised Voting system.

MongoDB, on the other hand, is typically used in scenarios where ACID properties are not as strict, focusing more on scalability and flexibility. MongoDB, while improving its support for transactions in recent versions, may not offer the same level of transaction control as Oracle. Therefore, for the AirBnB-Lite application, which deals with rapidly changing data, MongoDB's approach may be more suitable, and in applications where data consistency is paramount, Oracle is the better choice.

5. Scalability

MongoDB has a strength in terms of scalability. Its flexible document-based model and horizontal scaling capabilities make it an excellent choice for applications that need to handle vast amounts of data and a growing user base. Furthermore, it is designed for horizontal scalability, which means it can handle a massive volume of data and traffic by adding more servers to the cluster. This is essential for Airbnb-Lite, given its rapid growth and the need to accommodate new hosts and listings each month.

In contrast, Oracle's scalability is more challenging and often involves vertical and complex scaling strategies, which can be costly and complex. Therefore, it may be less adaptable to rapid growth scenarios.

6. Ability to Handle Massive Volumes of Data

For massive volumes of structured data, Oracle can handle the load effectively. For the Computerised Voting application, which deals with a vast number of registered voters, Oracle's structured and efficient storage can handle such data volumes effectively.

When dealing with the large and rapidly growing dataset of Airbnb-Lite, MongoDB's ability to scale out and distribute data across multiple nodes provides a clear advantage. However, MongoDB's document-based structure allows for flexible data handling, which suits the unstructured nature of Airbnb listings and reviews. However, it may not be the ideal choice for applications that heavily rely on complex SQL-like queries. Therefore, MongoDB is exceptionally well-suited for handling massive volumes of unstructured data, making it ideal for applications with a high data influx.

7. Ability to Execute Complex Queries

For applications that require complex, multi-table joins, and intricate queries, Oracle's relational structure and SQL querying capabilities are advantageous. Oracle, with its support for SQL, is well-equipped to execute complex queries on structured data.

MongoDB, being schema-less, is less suitable for these types of queries. However, it excels at handling simpler, more straightforward queries. Because of this reason, MongoDB can execute complex

queries but is more suited for aggregations and analysis of semi-structured or unstructured data, which is the case with Airbnb-Lite's listing and review documents.

8. Data Integrity

In the context of the Computerized Voting application, data integrity is crucial, and Oracle's relational model with enforced constraints ensures high data integrity. Therefore, Oracle, with its strict data consistency and integrity mechanisms, is the better choice when data integrity is crucial, as in the Computerized Voting application.

MongoDB may be more permissive, allowing for flexibility but potentially compromising data integrity in certain scenarios, so that requires careful data modelling and validation to maintain data integrity.

9. Differences in Data Types

Oracle supports a wide range of data types, including traditional SQL data types for structured data. Oracle, being a structured RDBMS, may require more data modelling efforts to handle unconventional data types.

In contrast, MongoDB supports unstructured and complex data types when processing a variety of data, and handles JSON-like documents, making it flexible in terms of data types, which makes it advantageous for processing a variety of data, such as AirBnB-lite applications.

Security Considerations for MongoDB

In a real-world application, securing MongoDB involves several vital considerations. First, enforce authentication using mechanisms and implement role-based access control (RBAC) to manage user and application access. Encrypt data both in transit with SSL/TLS and at rest to prevent unauthorized access. Configure firewall rules to restrict network connections to trusted IP addresses, blocking external unauthorized access. Regular security audits and access control reviews are crucial, with tools like MongoDB Atlas for continuous monitoring and auditing. Opportune patch management is vital to address known vulnerabilities. Deploy MongoDB in a secure network environment to minimize public internet exposure. Finally, classify sensitive data and restrict access based on its significance, recognizing that different data types require different levels of protection. These multifaceted measures collectively ensure the security of MongoDB in real-world scenarios.

Conclusion

In conclusion, the choice between Oracle and MongoDB depends on the specific requirements of the application. For structured data and data integrity, Oracle is a strong choice, as seen in the Computerized Voting application. In contrast, MongoDB excels in scalability and performance with unstructured data, making it a suitable option for a rapidly growing application like Airbnb-lite. Both platforms have their strengths and weaknesses, and the decision should be based on the specific needs of the project.

Case studies of other implementations using these paradigms can provide additional insights. For instance, Oracle is widely used in industries that demand high data integrity, like financial institutions, while MongoDB is favoured by companies like Airbnb, Twitter, and Facebook for their ability to handle big data and scale rapidly. These case studies reinforce the notion that the choice of database platform should align with the application's unique requirements.

In the end, the selection of the database platform is a critical decision that should be made based on a thorough analysis of the project's specific needs and long-term goals.

Reference

- *Software Testing Help 2021, 'SQL Vs NoSQL Exact Differences and Know When to Use NoSQL and SQL', Software Testing Help, viewed 24 October 2023, <https://www.softwaretestinghelp.com/sql-vs-nosql/#SQL_vs_NoSQL_Security>*
- *Software Testing Help 2021, 'MONGODB Vs MySQL – Key Differences Explained', Software Testing Help, viewed 24 October 2023, <https://www.softwaretestinghelp.com/mongodb-vs-mysql/>*
- *Stack Exchange 2021, 'Concurrency in NoSQL', Stack Exchange, viewed 24 October 2023, <https://dba.stackexchange.com/questions/171011/concurrency-in-nosql>*
- *Study tonight 2021, 'Mongo DB vs SQL Databases', Study tonight, viewed 25 October 2023, <https://www.studytonight.com/mongodb/mongodb-vs-rdbms>*
- *Study tonight 2021, 'Introduction to NoSQL', Study tonight, viewed 25 October 2023, <https://www.studytonight.com/mongodb/what-is-nosql>*
- *Study tonight 2021, 'Introduction to Mongo DB', Study tonight, viewed 25 October 2023, <https://www.studytonight.com/mongodb/introduction-to-mongodb>*
- *Mongo DB 2021, 'Comparing MongoDB vs MySQL', Mongo DB, viewed 26 October 2023, <https://www.mongodb.com/en-us/compare/mongodb-mysql>*
- *W3spoint 2021, 'MONGODB Vs MySQL – Key Differences Explained', W3spoint, viewed 26 October 2023, <https://www.w3spoint.com/mysql-advantages-disadvantages>*
- *Makble 2021, 'The advantage and disadvantages of MySQL', Makble, viewed 27 October 2023, <http://makble.com/the-advantages-and-disadvantages-of-mysql>*
- *Nitya, R 2021, 'What are the advantages and disadvantages of using MySQL stored procedures?', turotialspoint, viewed 28 October 2023, <https://www.tutorialspoint.com/What-are-the-advantages-and-disadvantages-of-using-MySQL-stored-procedures>*
- *Abhishek,b 2020, 'Difference between Oracle and MongoDB', geeksforgeeks, viewed 28 October 2023, <https://www.geeksforgeeks.org/difference-between-oracle-and-mongodb/>*
- *Michel,K 2019, 'Comparison of query performance in relational a non-relation databases', ScienceDirect, viewed 28 October 2023, <https://www.sciencedirect.com/science/article/pii/S2352146519301887#!>*
- *DarkReading,S 2010, 'The Top Five Challenges In Securing Oracle Databases', DARKREADING, viewed 28 October 2023, <https://www.darkreading.com/risk/the-top-five-challenges-in-securing-oracle-databases>*
- *Vinicius,G 2020, 'MongoDB Security: Top Security Concerns and Best Practices', Percona, viewed 29 October, <https://www.percona.com/blog/securing-mongodb-top-five-security-concerns/>*