WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Evaluation of NoSQL Database MongoDB with Respect to JSON Format Data Representation

Ramesh Byali [a], Jyothi [b], Megha Chidambar Shekadar [c]

[a]*cse dept PDIT Hospet India ramesh.byali@gmail.com*
[b]*cse dept PDIT Hospet India*
[c]*cse dept PDIT Hospet India*

DOI: https://doi.org/10.55248/gengpi.2022.3.9.24

A B S T R A C T

The current information technology advancement is attributed to the development of more data-intensive applications. However, due to stringent limitations on data structure, data relations, and other factors, relational databases do not allow us to work with huge data sets or maintain high volume databases. Unstructured data from many industries, including a variety of forms, must be processed and stored in databases. Thus, NoSQL types offer solutions to a number of problems associated with huge data bases. Largely because of the benefits like flexibility and horizontal scalability, NOSQL (Not Only SQL) is in demand. This essay explores many No SQL database types, such as MongoDB, CouchDB, HBase, Cassandra, etc. A document-oriented, cross-platform database called MongoDB offers excellent performance, high availability, and simple scalability. The MongoDB Query Language (MQL), created for ease of use by developers, is used by MongoDB. The documentation contrasts the syntax of MQL versus SQL for typical database operations. When your data is document-centric and doesn't fit well into the schema of a relational database, when you need to support vast size, when you need to quickly prototype, and for a few additional use cases, NoSQL databases like MongoDB are a solid option.The proper encryption methods used to various crucial data fields provide data protection without adversely affecting the database's performance in terms of speed or memory usage. Due to the use of appropriate encryption techniques, this is achievable. Since document-oriented and unstructured data are generally kept in databases, it is frequently used to handle various sorts of data. Examples of such databases include MongoDB, Cassandra, CouchDB, Redis, Hyper-table, and others. Since they are open source, there is a critical necessity to offer good security and protect the user's private information when it is in transit or at rest. There is a need for a single solution that can improve the security of data transfer by providing an improved encryption method that speeds up processes and uses less memory when maintaining databases.

Keywords:Big Data, NoSQL, MongoDB, Cassandra, CouchDB, Redis, Hyper-table, HBase, Cassandra, Associative database, Relational databases.

1. Introduction

Among the prior technologies, relational databases were frequently utilized by numerous applications to gather data. The relational database had a basic structure that looked like a table with columns and records. The data in those tables can be modified, reviewed using well-defined, organized queries, and preserved in accordance with requirements [1]. Due to the advancement of technology, it is now necessary to have a huge database, support for several formats, ease of use, and flexibility. New technology is required to satisfy all of the aforementioned criteria as well as evolving needs. NoSQL [2] is a technology built on non-relational data that satisfies all of the aforementioned requirements. The unusual popularity of NoSQL databases in information systems nowadays. A database designer must choose a NoSQL database when he or she is unable to determine whether to utilize a relational or non-relational database with the vast volume of data they have acquired in a variety of formats [2]. The purpose of this paper is to outline the key elements and traits of MongoDB technology, particularly for NoSQL as a replacement for relational databases. The structure of this essay is as follows. A basic introduction to both NoSQL databases and conventional relational databases is provided in the introduction. The part where some accomplished results from various researchers are mentioned continues in the following section. Some relevant works on this subject are also mentioned in this section. a thorough explanation of the many kinds of NoSQL databases Mongo dB explanation and the results of various queries are in the third section. The fourth

* *Corresponding author.*
E-mail address: *ramesh.byali@gmail.com*

segment concludes with a conclusion [3]. The relational databases method and the constraints of existing NoSQL alternatives were intended to be solved by MongoDB, a scalable, flexible NoSQL document database platform. Application developers now have access to a level of flexibility and scalability never before possible because to MongoDB's well-known horizontal scaling and load balancing capabilities. The best worldwide cloud database service for contemporary applications is MongoDB Atlas. Developers can use Atlas to deploy fully managed cloud databases on Google Cloud, AWS, or Azure. Developers can relax knowing that they have immediate access to the availability, scalability, and compliance they need for enterprise-level application development thanks to best-in-class data security and privacy standards practices [5][6].

2. Literature Review:

The relational data model (like SQL databases) provides mechanisms to handle data consistency within the system, to manage concurrency control so that the correct results are generated for concurrent operations, and to guarantee the ACID properties of a transaction, as defined in [4]: (1) Atomic: Either all of the operations in the transaction succeed or the entire transaction is rolled back. Second, the database must be consistent, meaning that a transaction can never commit and leave the database in an inconsistent state. Third, transactions are completely separate from one another, guaranteeing that no two are impacted by the other. Transactions that have already been processed continue to be valid even if the server is restarted, demonstrating durability. A user can execute a transaction on a single or many database tables using SQL with the ACID [4] feature. There is a difference between ACID and BASE, with Brewer's CAP theorem, introduced in 2012, providing three main guarantees: (1) Concurrency, (2) Atomicity, and (3) Partition-tolerance in a scale-up approach. Non-relational data models (such as NoSQL databases) are following BASE (Basic Availability, Soft-state, Eventual consistency property). Secondly, requests are always acknowledged with feedback about their success or failure. Thirdly, the system can function normally even when arbitrarily split up because of network outages [4]. While the relational database has the advantage of collecting data in the form of tables and storing structured data, the NoSQL database may be adequate on its own [5]. NoSQL stands for "Not Only SQL," however this does not imply that it is in opposition to SQL. Both relational and non-relational databases can coexist in a NoSQL system. In situations where a lot of data is being processed, they shine. In this context, data might be either fully or partially organized. It's important to note that NoSQL databases come in a variety of flavors. Cassandra, Mongo DB, Couch DB, and HBase are just a few examples. Each option has its own unique structure and range of possible applications. Such databases have been further categorized into numerous groups based on the criteria. Since NoSQL databases fall into distinct categories, this paper explains their real-world applications considering these distinctions. However, unlike relational databases, they cannot be easily shared between applications. In cases where one type of NoSQL is required, another cannot be substituted. The designer is able to implement the most appropriate database type based on the specific requirements of the project's environment. There are four types of non-relational databases: key-value stores, column family stores, document databases, and graph databases. B. Non-Relational Databases One common NoSQL database type is the key-value store.

These databases permit queries to be performed on the database by using a key-value pair. When compared to other NoSQL databases, key-value stores' exceptional performance and scalability can be attributed to their reliance on primary-key access. CouchDB: Couchbase is a popular open-source document-based NoSQL database that was inspired by Lotus Notes. It is developed by the Apache Software Foundation and is one of many popular key-value databases. It's a database with just one node, and it performs precisely like any other database [6]. A single node installation is the norm initially, although scaling to a cluster is always an option. It lets a single database to be used across several physical or virtual machines. When compared to a standalone CouchDB instance, the capacity and availability of a clustered instance are significantly greater. Erlang is a general-purpose language, and because CouchDB doesn't require locking the database for writes, it may be used without worrying about data corruption. In addition to the HTTP protocol and JSON, files other than JSON can be attached to documents in this database. Therefore, any program or app that can read and write JSON data can use CouchDB [6]. The second type of database relies on documents as its primary data storage medium. You can save and retrieve documents in XML, JSON, and BSON formats. In the database, all information about a specific instance of an object is kept together. This is why document stores are so popular in the world of web application development. Embedded documents, which are self-describing, tree-like structures, are supported. Embedded documents often include maps, collections, and scalar values. Migration from relational databases can be made easier with the help of document databases like MongoDB, which include a very robust and complex query language and features. The most well-known document databases are MongoDB and CouchDB. Thirdly, there are column family stores, which are databases where information is represented and kept in the form of column families rather than traditional rows. Each row contains a set of columns that correspond to the row's unique key.

Databases contain information that may be accessed and manipulated as a whole. Each column set is analogous to a table in a relational database management system. Using the key, RDBMS systems can identify each row (record). Here, there are many columns in every row (fields). One of the most well-known column-family databases is Cassandra; other notable column value databases are HBase and Cassendra. Data in HBase is saved in HDFS and is logically arranged into tables, rows, and columns. HBase files are kept in HDFS internally. Cassandra: Apache Cassandra is a free and open source, decentralized/distributed storage system (database) for handling massive amounts of structured data in a variety of locations throughout the globe. It's a highly available service with no weak links [8]. Cassandra [9] has the following characteristics: • It is a column-oriented database. It is scalable, robust, and error-tolerant while maintaining a high level of consistency. Open-source software was originally developed for Facebook. • Google Bigtable provides the foundation for the data model. Amazon Dynamo forms the foundation of the decentralized architecture. Graph databases make it easy to store information about entities and the relationships between them. Nodes are another name for entities, and they also have properties. A node is simply an instance of an object within the program. Edges, which are connections between nodes, represent relationships and can have their own attributes. Nodes are sorted after taking into account their connections with others, as shown by the directionality of the edges. In this setup, the intriguing analytics framework [9] is put in place to facilitate knowledge discovery via information retrieval and filtering from document-based NoSQL with a focus on

keywords and topic mining. Neo4j is the most widely used open-source graph database, and it was built with Java technology. It has a high degree of scalability and does not require any predefined data structures (NoSQL).

3. Mongo DB Discussion:

The MongoDB database is a free and open-source document storage system that was created by MongoDB, Inc. Data in MongoDB is stored in documents that are similar to JSON and might have varying structures. When information that is related to one another is kept together, it is easier for the MongoDB query language to access it quickly. Because MongoDB makes use of dynamic schemas, it is possible to construct records without first defining the structure of the record, including the fields and the sorts of data those fields can store. It is possible to make changes to the record structure by easily adding new fields or removing fields that are already there [7]. You will easily be able to store arrays, depict hierarchical relationships, and deal with other more sophisticated structures thanks to this data model's capabilities. The documents that make up a collection are not required to have the same set of fields, and it is usual practice to denormalize data. MongoDB is known for its excellent performance. When compared to traditional databases, the majority of operations in MongoDB can be completed far more quickly. 2. The auto replication (return) option that MongoDB offers enables you to rapidly restore data in the event that the database experiences an issue. 3. Sharing makes horizontal scaling in MongoDB possible. This makes horizontal scaling possible. The process of partitioning data and storing it on many machines while ensuring that the sequence in which the data is stored is maintained is known as sharding. Vertical scaling involves adding additional resources to an already existing machine, whereas horizontal scaling involves adding additional machines to handle the data. Vertical scaling can be contrasted with horizontal scaling, which involves adding more computers.

The implementation of vertical scaling is challenging, but the implementation of horizontal scaling is straightforward. Horizontal scaling database examples: MongoDB, Cassandra etc. 4. Load balancing Horizontal scaling is what makes it possible for MongoDB to maintain and balance the load. 5. High Availability: The availability of the MongoDB database is increased thanks to the Auto Replication feature. 6. Indexing: Indexes are used to recognize and locate the data quickly in order to prevent having to search through each and every document in C. Indexes are also used to avoid having to search through each and every document in C. Data Storage in MongoDB: A portion of the programming language known as Java Script is used to underpin the storage of data in MongoDB using the JSON (Java Script Object Notation) format. It is a format for the interchange of data that is lightweight. JSON can work with all the primary data kinds, including numeric values, texts, and Boolean values, in addition to arrays and hashes [23]. Records in a traditional database can be stored as JSON, as is the case with document databases like MongoDB. The following is an example of a document that uses JSON D to hold records in the same format as tables and rows. Taking Screenshots: The Processing of Queries Inside of MongoDB, the running of the queries in MongoDB that are required to build a database, insert a document, search for a document, and delete the document. When building the schema of a database, it is impossible to know in advance all of the queries that will be executed by end users because it is impossible to predict what those queries will be. An ad hoc query is a command that is only used temporarily and has a value that is determined by a variable [8][9][10][11]. The outcome of an ad hoc query may be different each time it is run because it is determined by the variables that are being looked at. When there are hundreds or even millions of different factors that need to be taken into consideration, optimizing the way in which ad-hoc queries are processed can make a considerable impact in the overall performance of the system. The fact that MongoDB is a document-oriented database with a flexible schema is one of the reasons why it has emerged as the cloud database platform of choice for enterprise applications that call for real-time analytics.

The potential for a dramatic boost in performance is made possible by support for ad-hoc queries, which enables developers to make changes to ad-hoc queries in real time. MongoDB allows users to conduct searches using regular expressions, field queries, and range queries. Specific fields can be returned by queries, and user-defined functions can also be considered. Because MongoDB indexes BSON documents and makes use of the MongoDB Query Language, this is made possible (MQL). According to our observations, indexing is the most common problem that customer service departments for technological products fail to address with their end users. When they are used correctly, indexes are meant to increase the speed and performance of searches[11] [12]. A failure to correctly create appropriate indices can and will typically lead to a wide variety of accessibility concerns, including problems with the execution of queries and challenges with balancing the load on the system. If the database does not have the appropriate indexes, it will have to read each document individually in order to find the ones that are relevant to the query expression. However, if an adequate index is present for each query, the server will be able to carry out user requests in the most efficient manner possible. MongoDB supports a wide variety of sophisticated access patterns to datasets thanks to its extensive collection of indices and features, which include language-specific sort orders. It is important to note that MongoDB indices can be constructed on demand in order to satisfy real-time, ever-changing query patterns and the requirements of applications. In addition to this, they can be specified on any field in any of your documents, including array fields that are nested inside other arrays. When your data is only stored in a single database, it is vulnerable to several potential failure points.

These include a server crash, interruptions in service, or even good old-fashioned hardware failure [13][14][15]. Any one of these occurrences would make it extremely difficult or impossible to retrieve your data. By utilizing many servers for both backup and disaster recovery, which is made possible thanks to replication, you may avoid these risks. If you scale horizontally over numerous servers that each store the same data (or shards of the same data), you will notice a significant boost in the availability and consistency of your data. It should come as no surprise that replication also assists with load balancing. It is possible for the load to be dispersed evenly between servers when numerous users access the same data at the same time [16][17][18].For this purpose, MongoDB makes use of something called a replica set. A primary server or node is responsible for accepting all write operations and then applying those identical operations to all of the secondary servers in order to replicate the data. In the event that the major server ever suffers a catastrophic failure, it is possible that any one of the secondary servers will be chosen to take over as the primary node. If the previous primary node is

brought back online, it will function as a secondary server for the node that is now the primary one. When dealing with particularly large datasets, sharding, the process of splitting larger datasets across multiple distributed collections, or "shards," helps the database distribute and better execute what might otherwise be problematic and cumbersome queries. These protocols guarantee that the data will remain consistent. Because MongoDB ensures that each and every user gets a consistent view and quality experience with the data that they need to access, there is no requirement to add an external load balancer to the system[19][20][21][22]

4. Summary:

NoSQL databases are, by their very nature, schema-free. Because of this characteristic of NoSQL databases, it is possible to add fields without affecting the database's overall structure in any way. NoSQL databases are quite helpful when it comes to managing the tremendous increase in the amount of data that is structured, semi-structured, and unstructured. This paper focuses on a few aspects of document databases, in particular MongoDB. This article provides an explanation of the data representation in JSON format in MongoDB, along with screen pictures demonstrating the execution of a few queries in MongoDB. The query performance of MongoDB and that of other NoSQL databases could be compared in a future phase of this research's scope of inquiry. Relational and NoSQL databases can be configured together within the same database management and collaborate without obstacles if a new system that combines system of both kinds of technologies is implemented. In addition, they can combine the benefits of each technology solution to create a system that is more effective and more powerful. There is a possibility that there is a way to implement a new system that combines system of both kinds of technologies.

REFERENCES

[1] Abramova, V., & Bernardino, J. (2013, July). NoSQL databases: MongoDB vs Cassandra. In Proceedings of the international C* conference on computer science and software engineering (pp. 14-22).

[2] Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. Asian Journal of Research in Computer Science, 4(2), 1-10

[3] Becker, M. Y., & Sewell, P. (2004, June). Cassandra: Flexible trust management, applied to electronic health records. In Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004. (pp. 139-154). IEEE.

[4] Berg, K. L., Seymour, T., & Goel, R. (2013). History of databases. International Journal of Management & Information Systems (IJMIS), 17(1), 29-36.

[5] Bjeladinovic, S., Marjanovic, Z., & Babarogic, S. (2020). A proposal of architecture for integration and uniform use of hybrid SQL/NoSQL database components. Journal of Systems and Software, 168, 110633.

[6] Chandra, D. G. (2015). BASE analysis of NoSQL database. Future Generation Computer Systems, 52, 13-21.

[7] Chen, J. K., & Lee, W. Z. (2019). An introduction of NoSQL databases based on their categories and application industries. Algorithms, 12(5), 106.

[8] Cuzzocrea, A., & Shahriar, H. (2017, December). Data masking techniques for NoSQL database security: A systematic review. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 4467-4473). IEEE.

[9] de Oliveira, V. F., Pessoa, M. A. D. O., Junqueira, F., & Miyagi, P. E. (2021). SQL and NoSQL Databases in the Context of Industry 4.0. Machines, 10(1), 20.

[10] Deka, G. C. (2013). A survey of cloud database systems. It Professional, 16(2), 50-57. IEEE.

[11] Di Martino, S., Fiadone, L., Peron, A., Riccabone, A., & Vitale, V. N. (2019, June). Industrial Internet of Things: Persistence for Time Series with NoSQL Databases. In 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 340-345). IEEE.

[12] dos Santos Ferreira, G., Calil, A., & dos Santos Mello, R. (2013, December). On providing DDL support for a relational layer over a document NoSQL database. In Proceedings of International Conference on Information Integration and Web-based Applications & Services (pp. 125-132).

[13] Gessert, F., Wingerath, W., Friedrich, S., & Ritter, N. (2017). NoSQL database systems: a survey and decision guidance. Computer Science-Research and Development, 32(3), 353-365.

[14] Guimaraes, V., Hondo, F., Almeida, R., Vera, H., Holanda, M., Araujo, A., ... & Lifschitz, S. (2015, November). A study of genomic data provenance in NoSQL document-oriented database systems. In 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 1525-1531). IEEE.

[15] Rodriguez, K. M., Reddy, R. S., Barreiros, A. Q., & Zehtab, M. (2012, June). Optimizing Program Operations: Creating a Web-Based Application to Assign and Monitor Patient Outcomes, Educator Productivity and Service Reimbursement. In DIABETES (Vol. 61, pp. A631-A631). 1701 N BEAUREGARD ST, ALEXANDRIA, VA 22311-1717 USA: AMER DIABETES ASSOC.

[16] Kwon, D., Reddy, R., & Reis, I. M. (2021). ABCMETAapp: R shiny application for simulation-based estimation of mean and standard deviation for meta-analysis via approximate Bayesian computation. Research synthesis methods, 12(6), 842–848. https://doi.org/10.1002/jrsm.1505

[17] Reddy, H. B. S., Reddy, R. R. S., Jonnalagadda, R., Singh, P., & Gogineni, A. (2022). Usability Evaluation of an Unpopular Restaurant Recommender Web Application Zomato. Asian Journal of Research in Computer Science, 13(4), 12-33.

[18] Reddy, H. B. S., Reddy, R. R. S., Jonnalagadda, R., Singh, P., & Gogineni, A. (2022). Analysis of the Unexplored Security Issues Common to All Types of NoSQL Databases. Asian Journal of Research in Computer Science, 14(1), 1-12.

[19] Singh, P., Williams, K., Jonnalagadda, R., Gogineni, A., &; Reddy, R. R. (2022). International students: What's missing and what matters. Open Journal of Social Sciences, 10(02),

[20] Jonnalagadda, R., Singh, P., Gogineni, A., Reddy, R. R., & Reddy, H. B. (2022). Developing, implementing and evaluating training for online graduate teaching assistants based on Addie Model. Asian Journal of Education and Social Studies, 1-10.

[21] Sarmiento, J. M., Gogineni, A., Bernstein, J. N., Lee, C., Lineen, E. B., Pust, G. D., & Byers, P. M. (2020).Alcohol/illicit substance use in fatal motorcycle crashes. Journal of surgical research, 256, 243-250.

[22] Brown, M. E., Rizzuto, T., & Singh, P. (2019). Strategic compatibility, collaboration and collective impact for community change. Leadership & Organization Development Journal.

[23] Sprague-Jones, J., Singh, P., Rousseau, M., Counts, J., & Firman, C. (2020). The Protective Factors Survey: Establishing validity and reliability of a self-report measure of protective factors against child maltreatment. Children and Youth Services Review, 111, 104868