International Journal of Research Publication and Reviews

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

The new Era of Database Management System using Mongo DB

Manish Verma

Arya College of Engineering & I.T, Jaipur, Rajasthan

ABSTRACT

This study delves into the core attributes, benefits, and practical implementations of non-relational databases, specifically highlighting MongoDB. It seeks to elucidate the reasons behind MongoDB's superiority over traditional relational databases, particularly in the realm of handling vast datasets. The comparative database in focus here is MySQL.

MongoDB distinguishes itself with its remarkable features, such as adaptability, scalability, automatic sharding, and replication. Large-scale data situations and real-time online applications have greatly benefited from this database technology, solidifying its position as a leader in the field.

Keywords: NoSQL, MongoDB, automatic sharding, data aggregation.

1. INTRODUCTION

The landscape of database management systems underwent a significant transformation starting in the 1980s with the inception of Relational Database Management Systems (RDBMS). These systems became the standard for storing information, particularly in applications like financial records, manufacturing, logistics, and personnel data. However, the limitations of traditional RDBMS became apparent as applications supporting millions of concurrent users emerged.

Handling substantial volumes of data posed a challenge for conventional relational databases.

Non-relational databases are referred to as "NoSQL" databases, a phrase that Carlo Strozzi first used in 1998 in response to the demand for more effective data management. NoSQL has developed into a term that now means "Not Only SQL."

When it comes to effectively managing semi-structured and unstructured data (such emails, multimedia, and social media), NoSQL databases have a huge advantage. Key-value stores, document stores, column-oriented databases, and graph databases are the different types of NoSQL databases.

Founded in October 2007, MongoDB Inc. created MongoDB, a flexible document- oriented database that runs on several platforms. In 2009, MongoDB moved from being a component of a Platform as a Service (PaaS) product that was originally planned to an open-source development paradigm. Since then, MongoDB has been the backend software of choice for well- known websites and services like The New York Times, eBay, Craigslist, Foursquare, and eBay.

With its easy scalability, high availability, and outstanding performance, MongoDB is built in C++. Collections and documents are MongoDB's basic building blocks.

Collections are similar to RDBMS tables in that they are stored in a database.

Collections are groups of MongoDB documents, and unlike RDBMS, documents within the same collection do not necessitate a uniform structure or common fields; MongoDB supports dynamic schema. The database uses a document-based query language that is

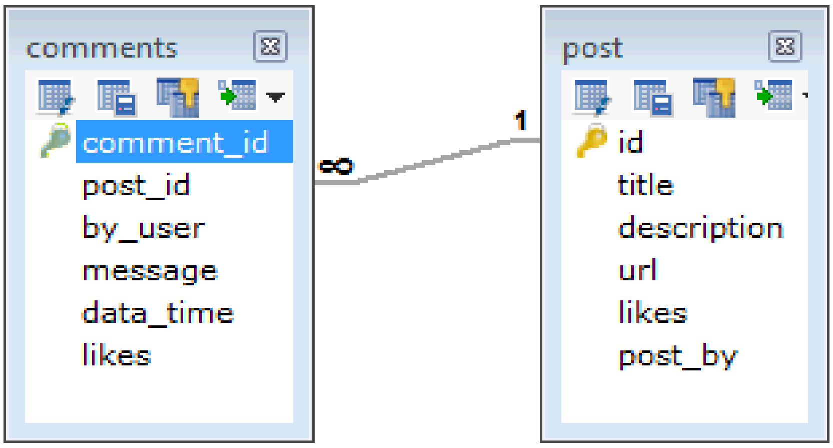
similar in strength to SQL to enable dynamic queries on data that is stored as JSON documents.

MongoDB has several noteworthy features, such as high availability, replication, and auto-sharding.

Consequently, MongoDB is widely used in user data management, data hubs, mobile and social infrastructure, big data analytics, and content management and delivery. Many different types of data are supported by MongoDB, including dates, objects, binary data, code, regular expressions, min/max keys, integers, strings, Booleans, doubles, and null values.

2. DATA MODEL MONGODB

MongoDB is a novel data storage system that uses documents formatted in BSON. BSON, a binary representation of JSON documents, serves as the foundation. Like how tables are arranged in conventional relational databases, similarly organized documents are grouped into collections. MongoDB's fields match columns in the MySQL paradigm, and its documents match rows. The way that MongoDB organizes its data contrasts sharply with MySQL, a relational database. Relational databases such as MySQL tend to disperse information about a particular record over several tables, whereas MongoDB gathers all relevant information into a single document, greatly simplifying the storage structure.



3. FEATURES OF MONGODB

MongoDB revolutionizes data storage with its dynamic and adaptable data model.

Unlike traditional databases, MongoDB's flexible structure accommodates data in various formats, facilitating effortless data modifications. Its hallmark feature, elastic scalability, ensures seamless expansion by distributing data across multiple servers, eliminating growth limitations.

One of MongoDB's standout advantages lies in its high performance compared to conventional relational databases. This performance is gauged through metrics like throughput and latency, showcasing MongoDB's efficiency at any scale. Unlike relational databases that rely on join

operations, MongoDB utilizes document

embedding and linking, reducing the need for complex table joins and enhancing data localization.

MongoDB's document-centric approach empowers each data structure to differ, enabling the addition of new fields without disrupting the central system catalogue or necessitating system

downtime. This adaptability, coupled with data locality and a dynamic schema, further amplifies MongoDB's appeal.

Key-value, range, geospatial, search, text search, and aggregation framework queries are among the query types that improve the querying capabilities of the database. Additionally, MongoDB allows map-reduce queries, offering a variety of options for data retrieval. Indexes play a pivotal role in optimizing data access, seamlessly integrated into MongoDB without relying on external application code.

In addition to these features, MongoDB incorporates failover mechanisms such as replica sets, ensuring data reliability.

Within a replica set, a primary server handles write operations, while multiple secondary servers manage reads. An arbiter server assists during failovers without storing data, ensuring smooth transition to the next primary server.

MongoDB also introduces the concept of aggregates, grouping related entities and value objects for more efficient data management. To handle large documents, MongoDB employs GridFS, allowing seamless storage and retrieval. Moreover, MongoDB's compatibility with various operating systems like Windows, Linux, Mac, and Solaris ensures its accessibility across diverse platforms.

In essence, MongoDB's groundbreaking features, ranging from its dynamic data model and query capabilities to advanced failover mechanisms and sharding innovations, position it as a cornerstone in modern database technology. Its ability to seamlessly adapt and scale makes it an indispensable choice for handling large- scale and real-time data applications.

4. COMPARISON: MONGODB VS MYSQL

MongoDB is a well-known non-relational database system that offers improved flexibility and horizontal scalability at the expense of some safety features of relational databases, such as referential integrity.

The main differences between these two database systems are substantial, making the choice between them more a matter of approach than pure technicality.

MySQL is a mature relational database system, providing a familiar database environment for seasoned IT professionals.

a. User-friendliness: MongoDB vs. MySQL Developers find MongoDB to be a compelling option. Anyone with programming experience can quickly and easily understand its data storage philosophy.

Data is stored by MongoDB in collections without a set schema. Since it stores data in a flexible manner, it is especially useful for developers who wish to use a database to support the development of their applications but may not be experts in databases.

This flexibility is a big benefit over MySQL: relational databases work best when the concepts of normalization, referential integrity, and relational database design are understood.

For teams developing applications that do not require all of the security features provided by relational systems, MongoDB offers a flexible developer interface. It can store documents of different schemas, including unstructured data sets. Web applications that can serve unstructured, semi-structured, or structured data from the same MongoDB collection without the need for structured schemas are a common example of this type of application.

When creating relational database solutions, updating or changing existing applications that are already integrated with a relational system, or designing solutions for users with a lot of experience with traditional SQL scripting, MySQL is frequently selected. Applications requiring intricate yet strict data structures and database schemas spanning numerous tables might benefit more from relational databases.

An application used in banking that needs to maintain precise point-in-time data integrity and very strong referential integrity as well as transactional guarantees is a common example of such a system.

It should be made clear, though, that MongoDB also supports the ACID (atomicity, consistency, isolation, and durability) characteristics of transactions. This gives developers more freedom to create a transactional data model that scales horizontally in a distributed setting without affecting the speed of multi- document transactions.

b. Security: MongoDB vs. MySQL

MongoDB uses a flexible permission structure in conjunction with the widely used role-based access control model. Once a user is assigned a role, that role gives them access to particular datasets and database functions. Every communication is secured by TLS, and data at rest can be encrypted by writing encrypted documents to MongoDB data collections with a master key that MongoDB never has access to.

MySQL has an identical authentication mechanism to MongoDB and supports the same encryption features. Along with roles, users can also be granted privileges, which grant them control over specific database operations and the ability to work with specific datasets.

c. Comparison of MONGO DB vs MYSQL Commands

MYSQL Commands	MongoDB Commands
SELECT * FROM table	db.collection.find()
SELECT * FROM table WHERE user=’Akshay’	db.collection.find({user=” Akshay”})
SELECT * FROM table ORDER BY Age	Db.collection.find.
DISTINCT	.distinct()
GROUP	.group()

Table 2: Fig (a) Retrieval of data in MySQL and MongoDB Modelling of data in MongoDB database differs from relational database. Various modeling techniques can be used, depending on the needs of the application. Normalization on collections and document embedding are the most popular modeling techniques.

There is a drawback to the embedding feature. In other words, it might result in a situation where documents get larger after they are created, which could harm database performance.

5. ADVANTAGES OF MONGODB

MongoDB operates within a schema-less framework, falling into the document store category. In this paradigm, a collection can house diverse documents, each with varying fields, content, and sizes. This inherent flexibility stands as a significant advantage, eliminating the need for complex joins and allowing different documents to coexist seamlessly.

One of MongoDB's core strengths lies in the clarity of its single object structure. Unlike relational databases, it does not require intricate mappings of application objects to database entities. Additionally, MongoDB offers deep query capabilities, empowering users to perform intricate searches and analyses with ease.

Furthermore, MongoDB excels in its scalability. The system allows for effortless scale-out, eliminating the complexities associated with scaling traditional relational databases. This ease of scalability is crucial for accommodating growing data sets and evolving application needs.

MongoDB leverages internal memory efficiently, facilitating swift data access. By storing the working set in memory, it ensures rapid retrieval and processing of data, enhancing overall performance.

Furthermore, MongoDB's flexibility in indexing enables programmers to build indexes on any attribute. The database's overall usability is improved and the development process is made even simpler by the support for secondary indexes, which also makes them transparent to developers.

6. APPLICATIONS OF MONGODB

Non-relational databases have gained immense popularity in the realm of

extensive data processing and real-time web applications, exemplified by tech giants like Facebook, Yahoo, Google, and Amazon. Beyond these major players, their utility extends to content management, delivery services, and mobile and social infrastructure. Among the array of NoSQL databases, MongoDB stands out as the optimal choice for managing user data. Its versatility also finds a niche in data hubs, making it indispensable for small to medium-sized non-critical sensor applications, especially when prioritizing swif t write performance.

CONCLUSION:

Given how recent the NoSQL trend is, a lot of researchers are drawn to this class of databases. Key-value stores in NoSQL databases, like MongoDB, offer an

effective framework for combining massive amounts of data. Complex data, such as an array, object, or reference, can be stored in a single field by MongoDB. Object mapping is quite simple in this kind of database. Development is faster with MongoDB than with MySQL thanks to features like data replication and auto splitting. Replication, auto splitting, horizontal scalability, and flexibility are all offered by MongoDB. For big data applications, MongoDB is a preferable option over MySQL database.

Performance-wise, it outperforms relational databases. We can select the best NoSQL database based on the application's requirements.

REFERENCES:

[1] Mr. Shanthanu Kanade, Mrs. Anuradha Kanade, and Dr. Arpita Gopal"A Study of Normalization and Embedding in MongoDB" IEEE International Advance Computing Conference (IACC), 2014

[2] Andrada OLAH, George PECHERLE, Robert GYORODI, and Cornelia GYORODI "A comparative analysis: MongoDB vs. MySQL" 13th International Conference on Engineering of Modern Electric Systems (EMES), June 11–12, 2015.

[3] K. Sanobar, M. Vanita, "SQL Support via Metadata for MongoDB"

[4] mongodb.com/compare/mongodb- mysql