International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6S, August 2019

Research of NOSQL and SQL and Designing a Better Database Schema for Databases

Shivam Kumar Giri, Chidanandan V., Narendran Rajagopalan

ABSTRACT--- The amount of data storage and querying is increasing day by day. The requirement of data-storage, security, scalability and management is an alarming issue. Structured Query Language (SQL) databases was designed to overcome most of the trouble faced over tradition file storage system. But the alarming rate of data storage leads to Not only Structured Query Language (NoSQL) databases. in this paper, various aspects of SQL and NoSQL are compared with respect to their data management and to develop a better schema for data management is attempted.

Keywords - key, documents, collections, grouping, schema, attributes, record, data dependencies, sharding, scaling, data replication.

1. INTRODUCTION

NoSQL and SQL are two popular types of Database management systems in the contemporary world. A series of comparisons were made on the basis of various properties executed by the databases design including time of execution of queries, size/pages acquired for storing records and indexes, usability, scalability and fault-tolerance support over the two-databases. Further based on the result, an ideal schema of database is designed with the combined factors of two property database design types and its various servers' applications including Artificial Intelligence.

SQL is the cross-platform programming languages developed in 1974 by IBM (International Business Machine) with a concept of Relational Database Management System. It proves beneficial than the traditional file access system and other API in terms of fetching multiple entries from the database, the concept of indexing, key and redundancy control. Various Database Management System and open source applications were developed by various organizations. MariaDB is an open source Relational Database Management System under the GNU Public License (GPL) [7]. NoSQL is a non-relational database developed in 1960 but got popularized with many additional features in 2000's. It does not have concept of key and dependencies among the attributes and being used for large data analysis and data science. MongoDB is an open-source NoSQL based Database Management System under the Server-Side Public License (SSPL) [8].

2. THEORY

2.1 Key and indexing in Database

Keys are used to identify each of the database entry record uniquely and are used to create relationship among the entries in a table. Indexing is a data structure technique which is primarily used in the database system to efficiently retrieve records from the database file system [4]. Generally, it is created over a specific attribute of the database entries termed as index, which are used for retrieval by identifying or matching the index of the entries.

2.2 MongoDB terminologies analogue to MariaDB

The distinction of MongoDB and MariaDB are as follows [5-6]:

Table 1: Distinction of MongoDB and MariaDB terminologies

Sl. No.	MariaDB	MongoDB
1	Attribute	Field
2	Record	Document
3	Table	Collections
4	Database	Database
5	İndex	İndex

There are no join operations in MongoDB hence no Normalization is required. In MongoDB there can be redundancy of data which in turn may help in scaling of database to a larger extent.

2.3 Scalability of databases

Scalability is stated as the capability of a system to adapt to the continuous database growth in a capable manner such that the database stays stable. Scalability can be broken into two parts:

- Read Scalability: Read scalability implies that database can accommodate high value of read operations.
- Write Scalability: Write scalability implies that database can accommodate high value of write operations.

MariaDB can be read scalable up to a higher degree. Multiple Read operations can be performed and accommodated easily. MariaDB uses 2 phase locking or multiple graduality locking protocol to prevent deadlock, which in turn prevents the write scalability of the Database.

Manuscript published on 30 August 2019.
* Correspondence Author (s)
Shivam Kumar Giri, National Institute of Technology, Thiruvettakudy, 609609, Karaikal, Puducherry, India.
Chidanandan V., National Institute of Technology, Thiruvettakudy, 609609, Karaikal, Puducherry, India.
Narendran Rajagopalan, National Institute of Technology, Thiruvettakudy, 609609, Karaikal, Puducherry, India. (E-mail: narendran@nitpy.ac.in)

MongoDB is both read scalable and write scalable up to a greater extent and can handle multiple read as well as multiple write simultaneously. [9,1]

There are two type of scalability factor:

- Horizontal Scalability: Horizontal Scalability implies that the database can be extended across various other servers joined together in a pool fashion. Data can be handled and accessed from anywhere across the pooled server. It is based on partitioning of data and is easy to extend without any limitation and multiple systems can execute the instructions at the same time.
- Vertical Scalability: Vertical Scalability implies that the system is able to handle various changes in growth by adding more powerful components such as RAM, Processor, GPU etc. These are bound by a limit the system can handle during scaling and only one system processes the entire activity, since faster the execution better the results.

2.4 ACID properties of databases

A Relational Database Management System maintain Atomicity, Consistency, Isolation, and Durability [12] which is known as ACID properties ensuring accuracy, reliability, safety and integrity of the transactions performed in the database system. These properties are mandatory for accurate and reliably managing the transactions in any of the Relational Database.

2.4.1 Atomicity

Atomicity is a property which states that the operations in transaction must either be fully executed or none of them should be executed. There must be no state in a database system where the partial transaction is executed with some of its operations.

2.4.2 Consistency

Consistency is a property which state that database must remain in a consistent state after any transaction. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.

2.4.3 Durability

This property of state that database should hold all its latest updates and the committed transactions must survived even if the system fails or rolls back or the entire system get restarted.

2.4.4 Isolation

This property of a database system applies to a system where more than one transaction is executed simultaneously. The property of isolation states that any transaction executed in the system should not be affected by the existence of any other transaction present in the system.

MariaDB being a Relational Database system follows each of the above properties for the transactions. Maintaining Consistency and Isolation tends to restrict Horizontal scaling of database.NoSQL in horizontal scaling follows the BASE properties which are derived from CAP Theorem which indicate the properties that a system can cannot have consistency, availability and fault tolerance capabilities at same instance [2]. BASE property indicates the following properties:

- **Basically, Available:** This property state that the database system does guarantee availability of data in the system at all instances.
- **Soft state:** Soft state of the database system indicates that the state of the system may change over time with various transactions. The state may even change without any input to the database system.
- Eventual consistency: This property indicates that the system tends to become consistent within a time-interval, given that the system doesn't acquire any input in that particular time-interval.

2.5 Sharding and Load Balancing

MongoDB is a cross-platform, document-oriented database that provides, high performance and easy scalability ensuring effective data management with its prominent feature of auto-sharding. Sharding splits the database across multiple servers, increasing the capacity and scalability as required. This feature handles distribution of data in different nodes to maximize disk space and dynamically load balance queries [11]. Partitioning the databases appropriately is a major step that determines the efficiency of sharding. This involves choosing an index of the MongoDB, competently as a shared key for further horizontal scaling of the database.

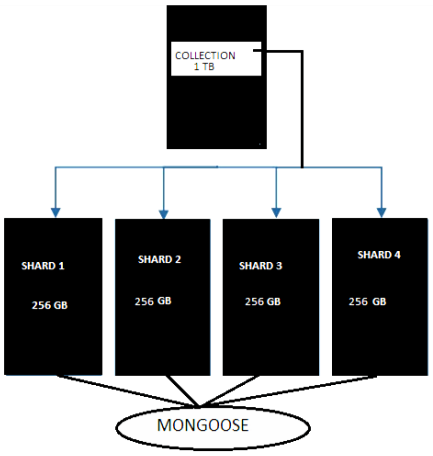

Fig1: Sharding of Collection and mongoose as router

2.7 Data Replication and Fault- Tolerance

Replication is the process of synchronizing data in the database across multiple servers. Replication provides increase in data availability by creating a multiple copy of data across different database servers which in turn increase the redundancy of data across the server. Replication protects a database from the data loss due to failure of a single server across the database server system. Replication allows you to recover from hardware failure and hence provide reliability and fault-tolerance. This feature is not present in MariaDB and most of Relational Database system. However Backup feature is possible in such systems which enable the user, a backup copy of all the table and records in the database.

However, MongoDB in clusters tends to provide the property of data replication and fault tolerance. Due to Data Replication, Data Availability increases, but on the other hand Data Redundancy also increases which leads to loss of atomicity of the database transaction.[3]. MongoDB achieves replication by the use of the replica set, which is a group of mongod instances that host the same data, setmongod is the daemon process which runs in the background and manages all the data request and accessibility. In a replica, one node is primary node that receives all write operations and the secondary operation tends to apply operations to the primary. [3] In case the primary node fails, an election takes place using voter's algorithm, elected secondary node will be assign the job of primary node till its reappearance.

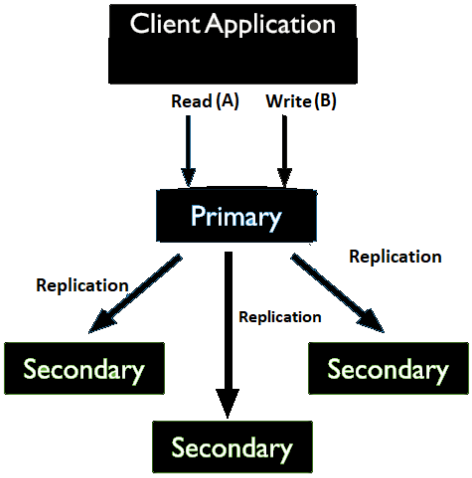

Fig 2: Data Replication in MongoDB

3. EXPERIMENTAL

3.1 Query Time Analysis using PHP in localhost

Some of the query time of certain instructions on insertion of records, file insertion in database, deletion of records, updating of data and selection of records are being analyzed for both MongoDB and MariaDB.

These reading were performed using loopback address (127.0.0.1) in an Asus STRIX Laptop with i7 Processor, 8 GB R.A.M., 64-bit Windows 10 Operating System. PHP is one of the widely used server scripting languages, basically serve as backend in most of the Website across the world.

The values are approximately same in other configurations too.

$$T_Q = T_D + T_f \quad (1)$$

T_Q =Total time to execute the query (order 10^0 s)

T_D =Total delay due to transmission and receiving frames (order 10^{-4} s)(negligible)

T_f = Total delay due to execute a query in database (order 10^0 s)

3.1.1 Pseudocode

```
<?php
session_start();
< -- Connect database --->
$time_pre_query = microtime(true);
< -- Query Here --->
$time_post_query = microtime(true);
$q_time = $time_post_query - $time_pre_query;
echo $q_time ;echo ' s';
?>
```

3.2 Analysis of size/pages required for database

MariaDB

```
>SELECT table_schema AS "Database", SUM(data_length) AS "Size (B)" FROM information_schema.TABLES GROUP BY table_schema;
>SELECT table_schema AS "Database", SUM(index_length) AS "Size (B)" FROM information_schema.TABLES GROUP BY table_schema;
```

In MariaDB the maximum size of one record is 2^{16} B (65535B).

MongoDB

```
>use db
>db.collection.stats()
```

In MongoDB stats usually result in displaying all sort of relevant data of the MongoDB schema including index, data stored, document count and metadata of the schema.

3.3 Output

The outcome of various time analyses, due to some of the widely used operations in the databases, illustrating the differences of the database management schema is as follow:

3.3.1 Time Required for Insertion of Character Data

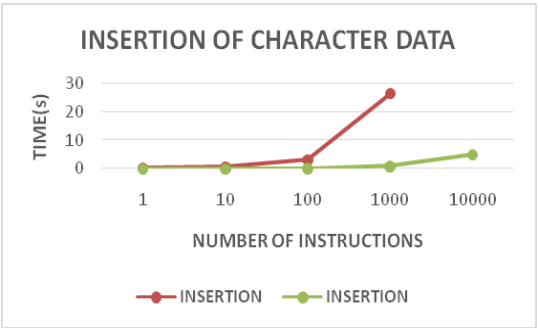

Fig 3: Graph Illustrating Insertion of Character based Data

Clearly from the Fig 3 as the number of instructions increases the time required for execution of insertion in MariaDB is much greater than that MongoDB. Therefore, insertion is costlier in MariaDB compared to MongoDB. This can be inferred from the document-oriented database system used by MongoDB.

3.3.2 Time Required for Insertion of Multimedia based Data

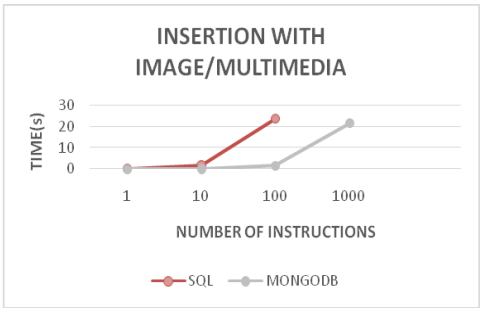

Fig 4: Graph Illustrating Insertion of Image/Multimedia based Data

MariaDB can take maximum size of 1 MB of multimedia file in a particular record whereas MongoDB use MongoBinData to store file upto 16MB and GridFS for file greater than 16 MB. Therefore, MongoDB is much more efficient in handling Big data. MariaDB can is slower than MongoDB due to the fact of 2 phase locking Protocol to avoid deadlock.

3.3.3 Time Required for Updation of Data

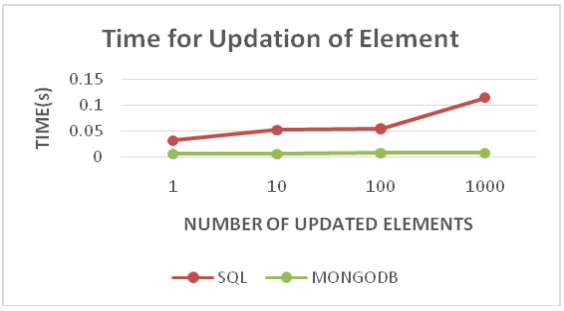

Fig 5: Graph Illustrating Updation of element in the databases

MongoDB can serve as a better database schema with respect to updation due to same reason due to 2 phase Locking protocol in SQL database.

3.3.4 Time Required for Deletion of Data

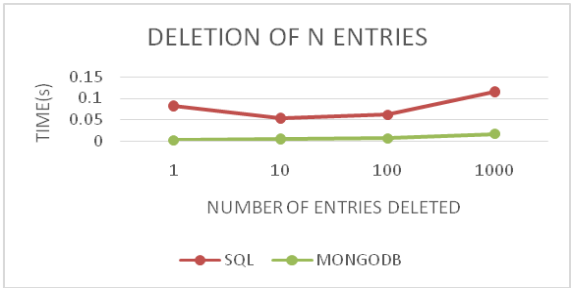

Fig 6: Graph Illustrating Deletion of entity in the databases

Similarly, the Deletion of entity is better in MongoDB than MariaDB.

3.3.5 Time Required for Selection of Data

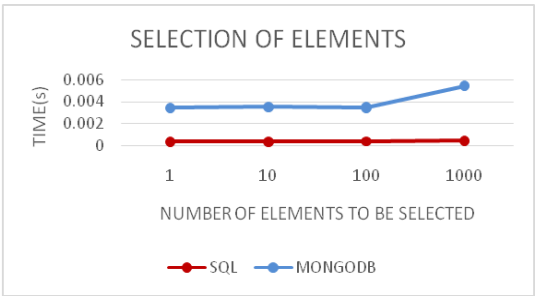

Fig 7: Graph Illustrating Updation of element in the databases

Selection of Data includes read operation of entity. Due better methodologies for indexing the MariaDB has better read scalability than MongoDB which in turn leads to better execution for the selection of elements in the database.

3.3.6 Size of Required for Data Storage

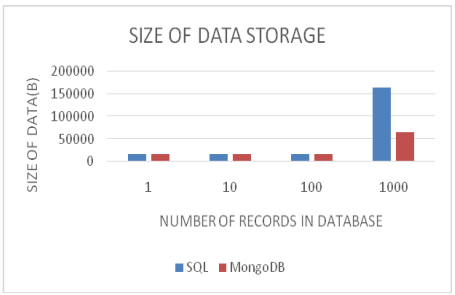

Fig 8: Graph Illustrating size acquired by entity in the databases

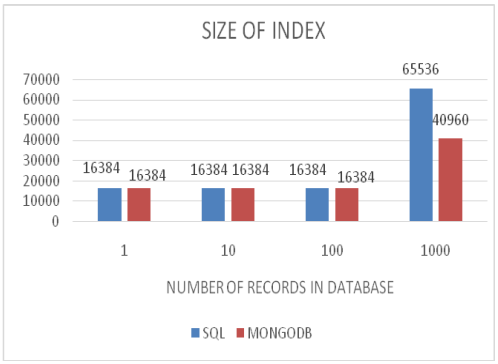

Fig 9: Graph illustrating size acquired by index in the databases

Size of Indexes and that of the data are measured in term of pages. 1 page is equivalent to 16384 Bytes of data which results. Hence for the first 100 entries, only 1 page is used for indexing and data. After which the size of data storage and indexing is quite low for MongoDB than that of MariaDB due to: Larger Key Size of MariaDB and No Fixed length records in MongoDB.

4. RESULTS AND DISCUSSION

Based on the above experimental we can conclude that the MongoDB is much more efficient in terms of vertical scaling, horizontal scaling, write scalability and write concurrency.

Due to this it is used in Real-Time databases and very Large-databases. But MariaDB is far better in terms of read scalability, selection of items as per Fig 7, follows ACID Transactions, suitable for most of day to day life applications.

In MariaDB/SQL is easy to understand and easy to code. MongoDB uses a document-oriented Database schema which is similar to JSON like structure. MongoDB generally do not use the Join operation, instead it groups all the related information in a same document. i.e. one document may contain all information of the entity, therefore in the horizontal scaling, there are many redundancies of data and high volume of data stored in each of the clustered server. It uses the load balancing technique known as sharding to manage the load across the servers, hence leading MongoDB to scale horizontally with fault tolerance.

In terms of security, MongoDB is much more secure than MariaDB as the Relational model of SQL tends to high probable to affect with various type of attack include blind SQL injections. SQL also tend to have concurrency bug [10] which tend to affect the reliability of the database

4.1 A HYBRID NOSQL APPROACH

Looking through various aspects of a relational DBMS and NO-SQL Database Management System, there tend to be a system containing a hybrid version of Both SQL and NoSQL. HYBRID NoSQL can have properties like isolation, durability, indexing and consistency from a Relation Database Manage system whereas, sharding, read write mechanism, horizontal and vertical scalability, security features, sharding and high availability from NoSQL.

4.2 Properties and Working of Hybrid system

The hybrid system to tend to have both NoSQL and MySQL properties. Therefore it follows SALT properties i.e.

- **S**trong indexing
- **A**vailability
- **L**oad balancing.
- **T**ransactional reliability

It must develop strong indexing as that of MariaDB indexing in each of individual database system, which can boost the read operations but lowers the time for write operations to some extent. As the individual System develops strong indexing with B or B+ structure and are vertical scaling, still the system will be document oriented and object based, along the horizontal scaling, there must be Auto load balancing like MongoDB. The sharding of data into small chunks must be carried out.

During Sharding and Data replication, there must be auto-index management over each node and each clustered system. Although due to availability feature, there might be a compromise with atomicity and redundancy control over the horizontally scaled system, but there will be unique index entries over individual collection or each in each of the node in case of data replication.

4.3 Features and uses

The features of above can be divided into two parts:

4.3.1Clustered systems

- Faster read/write operations
- Data replication and 24*7 Data Availability.
- Load Balancing and horizontal scaling
- Fault- Tolerance
- Vertical scalable
- Document -oriented Database Management System
- Applicable across Distributed System
- Follow the CAP Theorem

4.3.2 Individual System

- ACID Characteristics
- Vertical Scalable
- Faster Read operations than MongoDB
- Faster Write than MariaDB
- Fault Tolerance
- Document -oriented Database Management System

4.4 Application

1. HYBRID NoSQL can serve as better alternative to SQL based database schema due to faster read and faster vertical scalability.
2. HYBRID NoSQL can still serve as better option in Big Data and Real-Time system as it still executes all the features of NoSQL in the cluster Level and faster projection of Data.
3. HYBRID NoSQL can serve better alternative to storage in the field of AI (Artificial Intelligence).
4. HYBRID NoSQL can provide all the ACID transactions in an individual server system and high availability in clustered.
5. HYBRID NoSQL can used across distributed system with fault-tolerance and both horizontal and vertical Scalability.
6. HYBRID NoSQL can serve as a useful tool in data-science and data analysis.
7. HYBRID NoSQL is the combination of indexes of MariaDB with NoSQL which in turn inject some relational properties.
8. HYBRID NoSQL provide ease of joining and grouping operations. Grouping operations still need Map-Reduce operations in case of clustered index.

REFERENCES

1. MongoDB Manual 4.0: MongoDB CRUD Operations, https://docs.mongodb.com/manual/crud/,last accessed 2019/3/12.
2. MongoDB Manual 4.0: Transactions https://docs.mongodb.com/manual/core/transactions/, last accessed 2019/3/12.
3. MongoDB Manual 4.0: Replication https://docs.mongodb.com/manual/replication/, last accessed 2019/3/12.
4. MongoDB Manual 4.0: Indexes https://docs.mongodb.com/manual/indexes/, last accessed 2019/3/12.
5. MongoDB Manual 4.0: SQL to Mongo mapping https://docs.mongodb.com/manual/reference/sql-comparison/, last accessed 2019/3/12.

6. PHP: Mongo Manual: Php to Mongo Manualhttp://php.net/manual/en/mongo.sqltomongo.php, last accessed 2019/3/12.
7. MariaDB From Wikipedia, the free encyclopedia (2019)https://en.wikipedia.org/wiki/MariaDB, last accessed 2019/3/12.
8. MongoDB From Wikipedia, the free encyclopedia (2019)https://en.wikipedia.org/wiki/MongoDB, last accessed 2019/3/12.
9. Mrozek, Dariusz; Kasprowski, Paweł,Małysiak-Mrozek, Bożena; Kostrzewa, Daniel: Beyond Databases, Architectures and Structures Volume 521 || Performance Aspects of Migrating a Web Application from a Relational to a NoSQL Databases, Chapter 9, page: 107—115, 2015, ISBN: 978-3-319-18421-0,978-3-319-18422-7,
10. Fonseca, P; Cheng Li; Singhal, V; Rodrigues, R, "A study of the internal and external effects of concurrency bugs", IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), page: 221--230,2010.
11. Liu, Yimeng; Wang, Yizhi; Jin, Yi: Research on the improvement of MongoDB Auto-Sharding in cloud environment, page: 851—854, 2012, ISBN: 978-1-4673-0242-5.