# Comparison between MongoDB and SQL

Divyansh Saxena
Undergraduate Student
Computer Department
V.E.S Institute of Technology

Vaibhav Malpani
Undergraduate Student
Computer Department
V.E.S Institute of Technology

Vikramank Singh
Undergraduate Student
Computer Department
V.E.S Institute of Technology

Mrs. Lifna CS
Assistant Professor
Computer Department
V.E.S Institute of Technology

***Abstract* – Data is increasing at an unbelievably high rate and thus there is a need of a language that can manage such large data efficiently. In 1980s, SQL got released with a sole purpose of managing the data. But things started changing after 2005. The amount of data generated in past 2 years is more than the data generated previously in history of entire human race. Thus, there was a need of a new programming language that was different than SQL in structure but similar in functionalities provided which led to NoSQL. This paper gives a detailed comparison of NoSQL and SQL programming languages which will help readers in deciding which language to use in different situations.**

## I - Introduction

Everyday thousands of terabytes of data is generated via various sources which includes social media, e-commerce and other sources. Some storage system is required to store this data. This system is called as a database. There are different types of databases depending upon the type of data to be stored in it. Basically, the data is of two types, Structured Data and Unstructured Data. Structured data is more of an organized data which follows strict pattern along whole data. The most common example of structured data is a table with fixed number of rows and columns. And the most important and unique feature of structured data is that each row or data element comprises of the values of columns predetermined by the table. No additional data or information can be added to a new row. On the other hand, there is unstructured data which is opposite of structured data. The key feature of this type of data is its flexibility and unorganized structure. No tables are formed in such data. The most common way to express such data is by document, column or a graph. Each document of this unstructured data can have different fields attached to it unlike structured data.

These were two basic types of data that are currently formed in huge numbers on daily basis and need different types of databases to store them

## II – Types of Databases

The two different types of data discussed in previous section need different type of database to store them. They are Relational database and NoSQL database. The former one is used to store the structured data and later one for unstructured data. Relational database is commonly formed of rows (tuples) and columns which represents a table. Whereas, NoSQL database is used to store unstructured data and it has different forms like documents, graphs or columns. Most common form is document form which is very easy to write and interpret for humans and its format is also very informal unlike tables.

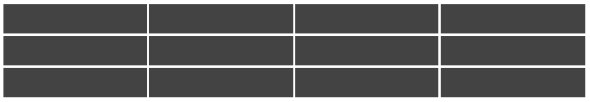

Fig 1 – Relational Database


Fig 2 – NoSQL Database

Data stored in these databases need to be managed.
To manage the data stored in these databases, there is a need of programming language that could help user to interact with the data stored in these databases. As these databases namely Relational and NoSQL have languages to work with. Data stored in Relational

databases are managed with the programming language called as SQL, Structured Query Language, while data stored in NoSQL databases are managed with different programming languages based on the format of data. In today's word, the most common unstructured data format is the document and to manage this type of data, MongoDB is used as the programming language. This is also known as the NoSQL language.

## III – Programming Language

To understand the basics of any programming language, there is a need to understand the structure of it. SQL is used for managing the structured data and has essentially three parts – Database, Table, Tuple. Database is a collection of tables that together describes a single system. For example, Hospital Management Database may have different tables such as Medicines, Employees, and Patients. Table is a combination of rows and columns containing different values that depicts some information. In Hospital Management Database, Medicines and Employees can be used as tables containing some information. Last part is a Tuple, which is also called row that contains information. There can be any number of tuples in a table inside a database. The SQL database disintegrates from top to bottom in following order – Database > Table > Tuple. MongoDB is a programming language that is used to manage the unstructured data stored in NoSQL databases. This has essentially three parts similar to SQL – Database, Collection, and Document. Database is a group of many Collections. Collection is a group of documents.

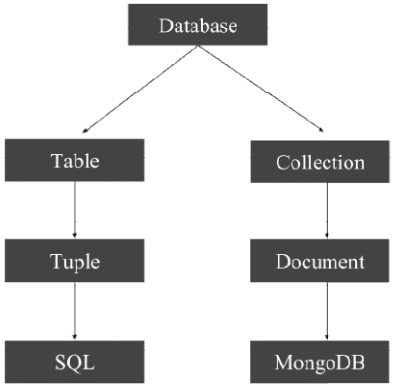


Fig 3 – Programming Language Structure

These programming commands either in SQL or MongoDB are commonly known as scripts. On execution of these scripts, data stored in the databases are managed. Some basic operations that can be performed using SQL are Select, Where, And/Or, Order By, Insert Into, Update and Delete.

The Select statement is used to select data from the database. The result is stored in a result table called the result-set.

```
SELECT column_name, column_name
FROM table_name;
```

The Where statement is used to extract only those records/tuples that fulfil a specified criterion.

```
SELECT column_name, column_name
FROM table_name
WHERE column_name operator value;
```

The And operator displays a tuple if both the first condition and the second condition are true. The Or operator displays a tuple if either the first condition or the second condition is true.

```
SELECT * FROM table_name
WHERE column_name operator value
AND column_name operator value;
```

The Order By keyword is used to sort the result-set by one or more columns. The Order By keyword sorts the records in ascending order by default. To sort the records in a descending order, you can use the Desc keyword.

```
SELECT column_name, column_name
FROM table_name
ORDER
BY column_name ASC|DESC, column_name ASC|D
ESC;
```

The Update statement is used to update existing records in a table.

```
UPDATE table_name
SET column1=value1,column2=value2
WHERE some_column=some_value;
```

The Delete statement is used to delete rows in a table.

```
DELETE FROM table_name
WHERE some_column=some_value;
```

Some basic operations that can be performed using MongoDB are create database, drop database, create collection, drop collection, insert document, query document, update document, delete document etc.

MongoDB use Database is used to create database. The command will create a new database, if it doesn't exist otherwise it will return the existing database.

use DATABASE_NAME

MongoDB db.dropDatabase() command is used to drop a existing database. This will delete the selected database. If you have not selected any database, then it will delete default test database.

db.dropDatabase()

MongoDB db.createCollection(name,options) is used to create collection. In the command, name is name of collection to be created. Options is a document and used to specify configuration of collection.

db.createCollection(name,options)

MongoDB's db.collection.drop() is used to drop a collection from the database. To insert data into MongoDB collection, you need to use MongoDB's insert() or save() method. To query data from MongoDB collection, you need to use MongoDB's find(). Update() and save() methods are used to update document into a collection. The update() method update values in the existing document while the save() method replaces the existing document with the document passed in save() method. MongoDB's remove() method is used to remove document from the collection. remove() method accepts two parameters. One is deletion criteria and second is justOne flag.

db.COLLECTION_NAME.drop()
db.COLLECTION_NAME.insert(document)
db.COLLECTION_NAME.find()
db.COLLECTION_NAME.update()
db.COLLECTION_NAME.remove()

The above mentioned MongoDB scripts manage the data stored in NoSQL. MongoDB (from *humongous*) is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster. MongoDB is developed by MongoDB Inc. and is published as free and open-source software under a combination of the GNU Affero General Public License and the Apache License. As of July 2015, MongoDB is the fourth most popular type of database management system, and the most popular for document stores.

## IV – COMPARISON

SQL stores data in form of tables and the database schema and the set up rules are predefined as per the requirement to control the relationship between fields in the table. In SQL, related information is stored in different tables, but is associated using JOIN function which leads to minimized duplication. In case of MongoDB, data is stored in JSON-like documents. The related information is stored in same document which leads to faster query access and processing speed. In MongoDB, there is no need to define the structure such as the fields or their values. Structure of the data can be changed at any instant by adding new fields or removing existing fields.

The key features that make MongoDB stand out in front of SQL are flexibility, scalability, out-of-box replication and auto-sharding. Flexibility is an important feature of MongoDB that makes it better than SQL as in today's world, there is a variety of data and it keeps changing every moment. So predetermining the structure of the database table will to create another table for new data thus resulting in slower processing even though there are features like JOIN in case of SQL. MongoDB excels in this parameter. As it stores the data in document format, any new field can be added to a particular document at any instant. Scalability is another important advantage of MongoDB over SQL. MongoDB being a NoSQL supports horizontal scalability. Horizontal scalability means to add more and more nodes to the existing system. In case of databases, a single database is partitioned and stored in different clusters with high CPU and RAM power distributed equally on all servers. This makes it very scalable as even though the amount of data is huge, horizontal scaling makes it easy and fast to handle this much of data. This is not the case in SQL, it works vertically.

In SQL, single system is burdened with more and more data. The only way the speed doesn't get affected is that the CPU and RAM power are improved as the data increases. But there is an end to it which makes MongoDB better than SQL in terms of scalability. Auto sharding is another important feature of MongoDB that makes it standout when compared to SQL. Sharding means to disintegrate a single system to different servers horizontally. This process sometimes takes place automatically which makes MongoDB exceptional and is thus used by more than 1000+ customers worldwide including FIFA, eBay etc. These features are not available in SQL.

| | SQL | MongoDB |
|---|---|---|
| Rich Data Model | No | Yes |
| Dynamic Schema | No | Yes |
| Typed Data | Yes | Yes |
| Data Locality | No | Yes |
| Field Updates | Yes | Yes |
| Easy for Programmers | No | Yes |
| Complex Transactions | Yes | No |
| Auditing | Yes | Yes |

Fig 4 – Comparison Table

But, there are various use cases when using SQL over MongoDB actually makes the system better in terms of speed. Applications that require complex, multi-row transactions (e.g., a double-entry bookkeeping system) would be good examples. MongoDB is not a drop-in replacement for legacy applications built around the relational data model and SQL. A concrete example would be the booking engine behind a travel reservation system, which also typically involves complex transactions. While the core booking engine might run on SQL, those parts of the app that engage with users – serving up content, integrating with social networks, managing sessions – would be better placed in MongoDB.

## V - SUMMARY

Briefly, systems that are very complex should prefer using SQL or relational databases and the rest of the use cases should prefer using MongoDB.

There are some instances where both MongoDB and SQL are used. These are also termed as hybrid systems as they utilize both kind of methodology to store the information. For example, most of the latest ecommerce websites use both deployment methods. The product catalog which has the list of various types of products and this list is ever increasing; MongoDB certainly is the best choice and is thus used in such cases. But when the checkout system is considered which involves transaction system and is relatively very complex; MongoDB is certainly not the better choice and thus various ecommerce use SQL here. So, there are instances where hybrid deployment works better than using either of them.

## VI – REFERENCES


[1] Rupali Arora, Rinkle Rani Aggarwal, International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013

[2] Vatika Sharma, Meenu Dave, International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 8, Aug-08

[3] MongoDB and MySQL Compared, White Paper, https://www.mongodb.com/compare/mongodb-mysql

[4] en.wikipedia.org/wiki/Scalability#Horizontal_and_vertical_scaling

[5] Sanobar Khan , Prof.Vanita Mane SQL Support over MongoDB using Metadata International Journal of Scientific and Research Publications, Volume 3, Issue 10, October 2013 1 ISSN 2250-3153

[6] Katarina Grolinger, Wilson A Higashino, Abhinav Tiwari and Miriam AM Capretz, Data management in cloud environments: NoSQL and NewSQL data stores, Grolinger et al. Journal of Cloud Computing: Advances, Systems and Applications

[7] Shubham Kumar www.jboss.org/dms/judcon/2013india/presentations/day2track2session6.pdf