NoSQL: Will it be an alternative to a relational database? MySQL vs MongoDB comparison

Basit alyasiri[1], Bushra Sahi[2], Nassir AL-Khafaji[3]
{alyasirib@gmail.com[1], bushrasahi69@gmail.com[2], nassir.farhan@stu.edu.iq[3]}

Nasiriya Technical Institute, Southern Technical University, Iraq[1,2,3]

Abstract. NoSQL databases and related technologies are rapidly evolving and are commonly used in a variety of scenarios (consistency, availability, partition tolerance), and this has sparked controversy in the database world in several, so this paper aims to shed light on the characteristics that determine the use of databases. By discussing the challenges and studying the differences in depth and trying to answer the question related to the future use of the database, we started by discussing both relational and non-relational databases, shortcomings, and challenges, then we chose a model of both types, and we studied the differences in practice in the query process by installing database servers, it is an important part of the data handling process. We also discussed the popularity of the two models adopted for several considerations and concluded that non-relational databases, despite the challenges, tend to be the first choice soon.

Keywords: Relational Database, Nosql, Database, CAP, Mongodb, Mysql Queries.

1 Introduction

Information management systems have increased in importance nowadays because of the importance of modelling and organizing the data that we need to use in the right way and at the right time. These systems provide a very efficient way of dealing with different types of data. Currently, organizations are dealing with applications that generate large amounts of data daily, so their size is growing exponentially and that is why the thinking of using a database goes beyond the limits of the relational database in use which has evolved over decades. But because there are certain limits to the use of these rules, including their limitations in dealing with big data, as well as their unsuitability for the unstructured and diversified data found in many companies, and because of their inability to update.

And add other schemes that we need to add as the work develops, and from Here comes the need to use non-relational databases that have adopted many characteristics to deal with large, structured, or unstructured data. It may also can accept the addition of new updates depending on business variables, but many organizations have not dispensed with the relational database because they work one way or the other. To manage their data because it has advantages that compete with non-relational database, which are among the challenges for NoSQL that can discuss solutions and deal with their use, Where previous works were investigated and we found that most of them focused on one topic, which is how to choose database by studying differences between different models of them, so our study was heading towards whether it is possible to

IMDC-IST 2021, September 07-09, Sakarya, Turkey

DOI 10.4108/eai.7-9-2021.2314925

solve NoSQL instead of a relational database in all cases.

2 Related Study

There have been a lot of discussions lately about the performance of SQL and NoSQL databases. However, there have been few studies comparing the performance of SQL and NoSQL databases.

[4]This paper discusses database storage structures, concepts, and differences, as well as insertion, update, deletion, and selection processes. A set of the non-relational database was selected for evaluation, and their performance was successively compared with no relational database, and it was concluded that the performance of the relational database is stronger in small data, versus big data. Non-relational databases are preferred.

[7]This paper compares the efficiency of different queries and commands in both relational database and NoSQL systems using two different sets of data of different sizes to highlight the advantages of NoSQL database and over the relational database and especially when investigating big data. The study concluded that differences in execution time were observed and demonstrated using graphs. In all cases, it can be inferred that NoSQL databases outperform relational databases.

[9] This article introduces the main types and characteristics of NoSQL data models, as well as detailed and synthetic comparison between the relational model and the NoSQL data models and concluded that The Relational Model and NoSQL Models are not mutually exclusive, but rather complement each other, and Each has advantages and disadvantages in a particular context. They represent solutions of different scenarios and coexistence within companies.

[12] This paper compares the concepts of NoSQL and Relational databases then their limitations. And address the advantages and types of NoSQL databases. . From these comparisons, the conclusion was that a relational database might be a great choice if you have structured data and need a traditional relational database. If you have unstructured and/or structured data with the potential for rapid growth, NoSQL would be the best choice.

[15]This paper did a comparative analysis to discover that non-relational databases work better than a relational database. Looking at the dataset used for the experiment, the hypothesis is that MongoDB works better only for insertion, deletion, and selection, while Oracle still does better with the update process in the available environment.

3 Database

The database has played an important role in society for centuries, and anything that carries information constantly is a type of database. Cave drawings, stone slabs, and papyrus are all true examples of the database. The history of the database is largely due to the desire to create structures that can meet the requirements of storing information accurately and permanently over long periods while enabling users to easily search for information within the stored data.

The introduction of computers opened new possibilities for database technology, which allowed to quickly generate, and access information stored on the disk; However, the mere existence of a computerized system does not automatically make the data strongly interactive. But it needs systems to manage it and languages for the processes of interacting with data in terms of querying, adding, speaking, and other operations that you perform on the database to get the required [1].

4 Relational Database

The relational database has become an integral part of most software applications, and, in the opinion of most programmers, they cannot be dispensed with due to their ease of use, organization of data, and constant storage, which enables constant data retrieval while allowing multiple users to access the same data concurrently with the ability to modify that data. Most current relational database programs are written by separate teams of programmers and are then reused by other applications [13]. SQL databases are Database Management Systems (RDBMS). Microsoft SQL Server, Oracle Database, MySQL, and IBM DB2 have historically been the most popular of these. Except for MySQL, this is also used to store data for Web applications.

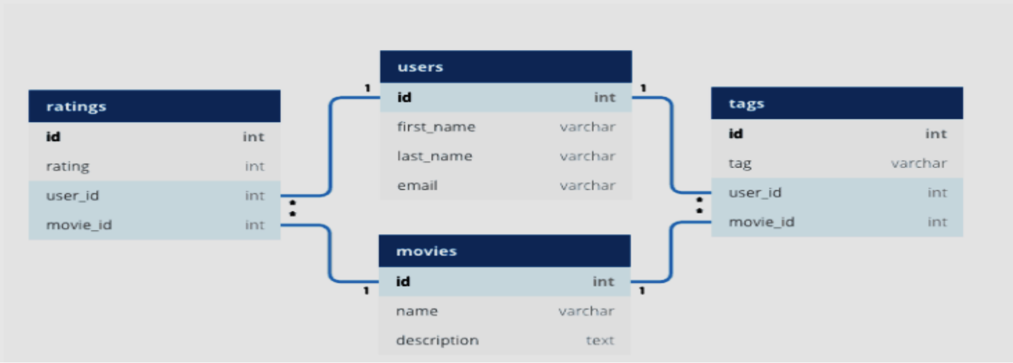


Fig 1. Example of a relational database.

4.1 The ACID Properties

The ACID properties ensure that transactions are reliable in a relational database. Atomicity, Consistency, Isolation, and Durability (ACID) are acronyms for Atomicity, Consistency, Isolation, and Durability. The ACID properties are usually followed by the RDBMS database.

1. Atomicity: A transaction must be done in its entirety or none.
2. Consistency: All transactions ensure that the database’s credibility is maintained. After a transaction, the database is not left in an invalid state.
3. Isolation: Each transaction must be run in isolation to ensure that any data inconsistencies do not a fact other transaction.
4. Durability: A completed transaction’s modifications must be retained, An ACID-compliant database ensures that only successful transactions are

completed, which provides data integrity for the database as well as the ability to recover from system failures. [12]

4.2 Challenges Faced by Relational Database

Since the 1980s, the relational model has dominated the computing industry, mostly for data storage and retrieval. After a while, relational databases began to lose their importance due to facing some challenges, and the most important of them [4]:

1. Big Data: The development of big data applications, particularly web applications, has become increasingly important in recent years. Indeed, a growing number of organizations across a variety of sectors depend on information gleaned from vast quantities of data. However, when dealing with large amounts of data, conventional relational database systems and platforms are inefficient, with slow response times and a lack of scalability, performance, and accuracy. To address the complexity of big data problems. [11]
2. Unstructured data: Difficulty dealing with relational database with unstructured data While relational databases take advantage of SQL, which has been optimized to work with structured data, SQL can become very complex when dealing with unstructured data.[12]
3. Data schema modification and development: With application creation and development, the relational database schema cannot be modified over time, and modifications cannot accommodate different types of data [14]. It became clear that conventional relational databases were incompatible with the requirements of a variety of applications and their data. This resulted in the introduction of a new technology known as NoSQL among commercial organizations and businesses.[12]

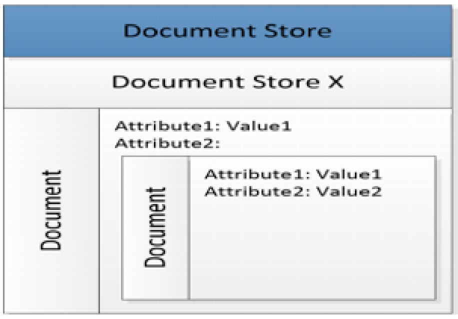
5 NoSQL Database

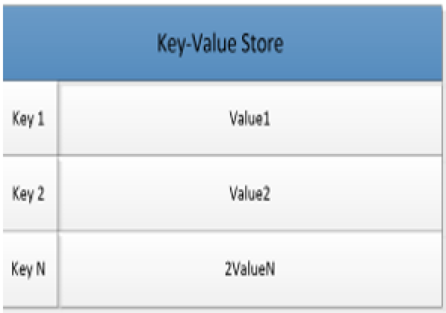
The current meaning of the term NoSQL was given by Eric Evans, in June 2009, at a meeting in San Francisco, organized by Johan Oskarsson for discussions on a new open-source distributed database. In October of the same year, Emil Efren, the founder, and CEO of Neo4J described, in a message on Twitter, the term NoSQL as “Not Only SQL Today, but NoSQL is also used as an alias for ”Next-Generation Database Management Systems mostly addressing some of the points: (being non-relational, distributed, open-source and horizontally scalable”) [9].

5.1 NoSQL Database Types

NoSQL data is stored in one of four data architecture patterns.

1. Key-Value Store Database
2. Column Store Database
3. Document Database
4. Graph Database

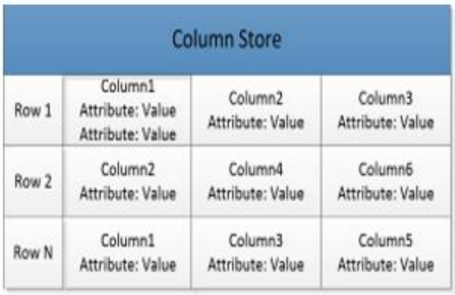




Key-Value Store	
Key 1	Value1
Key 2	Value2
Key N	2ValueN

Fig. 2. Example of a key-value store**.**

Fig. 3. Example of a Document Store.



Column Store			
Row 1	Column1 Attribute: Value Attribute: Value	Column2 Attribute: Value	Column3 Attribute: Value
Row 2	Column2 Attribute: Value	Column4 Attribute: Value	Column6 Attribute: Value
Row N	Column1 Attribute: Value	Column3 Attribute: Value	Column5 Attribute: Value

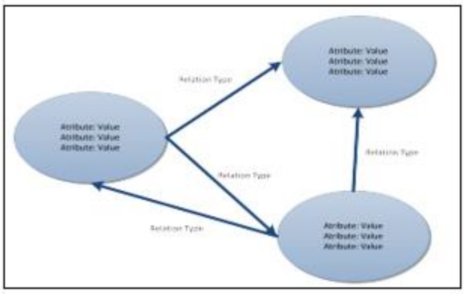


Fig. 4. Example of a Column Store.

Fig. 5. Example of a Graph Store.

5.2 Characteristics of NoSQL Database

1. Big data: is a term that refers to a group of vast amounts of structured and semi-structured data. It is used in many organizations and in various web applications and research to make critical decisions. Efficient storage in processing this large volume of data and extracting critical information with a short response time is one of the requirements in today’s competitive environment. Relational databases, which have dominated storage technology for a long period, seem unsuitable for mixed data formats. Data in tables cannot be expressed only in terms of rows and columns. NoSQL enables the development of a variety of storage formats that are easily compatible with high-speed, large, and varied data [3].
2. Scalable Horizontally scalable refers to the fact that, unlike relational database that can be scaled mainly vertically, by increasing the capacities of highly specialized and very expensive servers, NoSQL database can be distributed on multiple commodity servers, thus offering the possibility to add new servers in the system architecture, at any time, as needed.
3. The CAP Theorem Instead of ACID restrictions for a relational database, the NoSQL database, which emerged to achieve horizontal scalability, and access to data with the help of multiple nodes are based on CAP theory. CAP theory has been named based on the principles of consistency, availability, and partition tolerance. Consistency refers to having the same data at the same time on all nodes as the data is distributed. In the event of any changes,

it ensures that updates are made to all nodes and ensures that users have access to the same data. Ensure that all requests directed to the database with the help of the principle of access do not go unanswered even if some nodes (nodes) are not available [5].

4. Partitioning Tolerance allows data to be distributed to multiple nodes and create a block, even if some nodes are corrupted. Is that the system can continue to work? Also, the CAP theory, A system contains no more than two of the three principles mentioned at the same time. He expresses his ability to present it. Hence, a database System can be CA, AP, or CP at the same time [6].

5. Non-relational means that the data are represented by principles other than those used by the relational model [8].

6. Distributed refers to the fact that, unlike a relational database, NoSQL databases are used efficiently and flexibly in distributed systems with multiple servers and different environments [5].

7. Open-source means that, unlike most RDBMSs, NoSQL is open-source to provide solutions for complex applications at lower costs, although many of these systems today also offer commercial variants [2].

6 Comparison Between a Relational Database and NoSQL Stores

To study database, the general and future trend, and the best choice considering the rapid developments in the use of data in terms of type and quantity, and the best mechanism to obtain the best results. We performed a comparison between the MySQL relational database type and the NoSQL type (MongoDB). The comparison was according to the following aspects:

1. Comparison of general characteristics. Between MySQL vs MongoDB.
2. Comparison of the quality of queries Between MySQL vs MongoDB.
3. Comparison According to the CAP theorem. Between MySQL vs MongoDB.
4. Comparison by popularity Between MySQL vs MongoDB.

6.1 Comparison of characteristics Between MySQL vs MongoDB.

We provide a description and comparison of the main characteristics of relational databases and NoSQL stores according to the paragraphs mentioned in Table 1.[17]

Table 1. Comparison of the main characteristics of relational databases and NoSQL stores [17].

Name of properties	MongoDB	MySQL
Description	One of the most popular documents stores available both as a fully managed cloud service	Widely used open-source RDBMS
License	Open Source	Open Source
Data scheme	schema-free	Yes
SQL	Read-only SQL queries via the MongoDB Connector for BI	yes
Partitioning methods	Shading	horizontal partitioning, shading with MySQL Cluster
Replication methods	Multi-Source deployments with MongoDB Atlas Global Clusters	Multi-source replication
MapReduce	Yes	No
Consistency concepts	Eventual Consistency Immediate Consistency	Immediate Consistency
Foreign keys	no	yes
Transaction concepts	Multi-document ACID Transactions	ACID
Concurrency	Yes	yes
Durability	yes	Yes

6.2 Compare the Quality of Queries Between MySQL vs MongoDB.

To test the advantages of operations between relational and non-relational databases, a comparison was made between the two databases, MySQL and MongoDB in the four

1. Insert
2. Select
3. Update
4. Delete

And because the results that we obtain for testing operations in the different databases need to determine the performance and specifications of the computer used, so we used in these tests a computer that has the following characteristics, an Intel Core i5 processor and an Intel Core i5 processor (2.8 GHz) And 4 GB RAM. And Windows 10 64-bit, for that, we needed MySQL database by installing AppServ9.3.0 into MongoDB database through installing 4.4 MongoDB Compass Create and develop the same database in MySQL server and MongoDB server, and we created the same tables in both databases,

6.2.1 Insert

In the data entry test, 10,000 records were first entered into tables that do not contain data from the MySQL and MongoDB databases, organized by some tables for the hiding process, whose databases were created. The time it takes to enter.

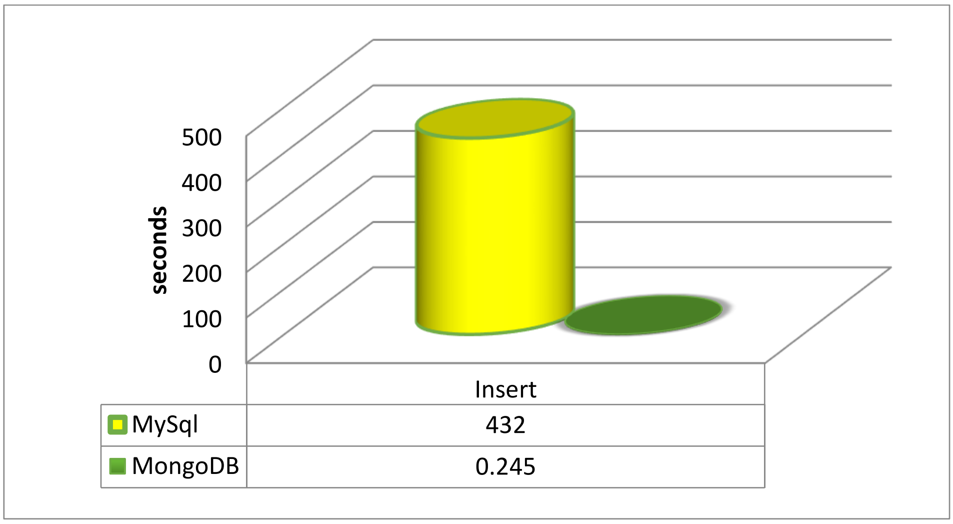


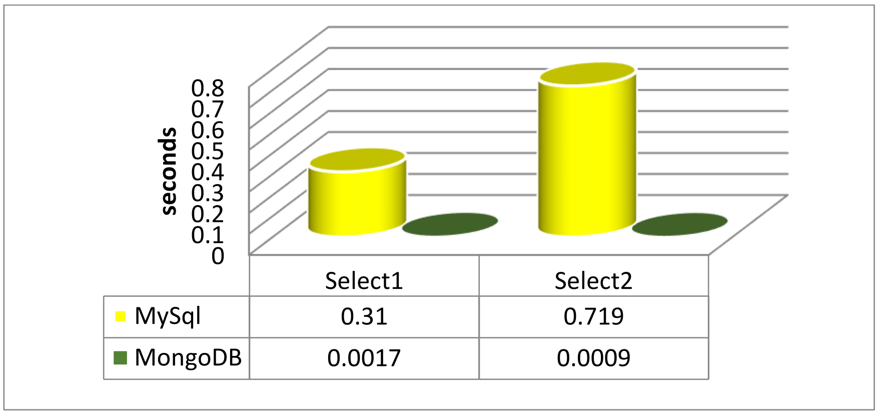
Fig 6. MySQL vs. MongoDB Insert.

Through the results, the MongoDB database took 0.245 seconds to enter 10,000 records, while the result of entering the same number of records in MySQL was 432 seconds. Hence, we conclude that MongoDB is the best performer in the process of entering a large group of records at once as shown in Fig 6.

6.2.2 Select

In the case of the record display test, two queries were executed,

1. The first is to choose all records and respond to the time required to execute this query.
2. The second is to query specific records on the condition that the time required to be executed for both rules is also calculated.



	Select1	Select2
🟨 MySql	0.31	0.719
🟩 MongoDB	0.0017	0.0009

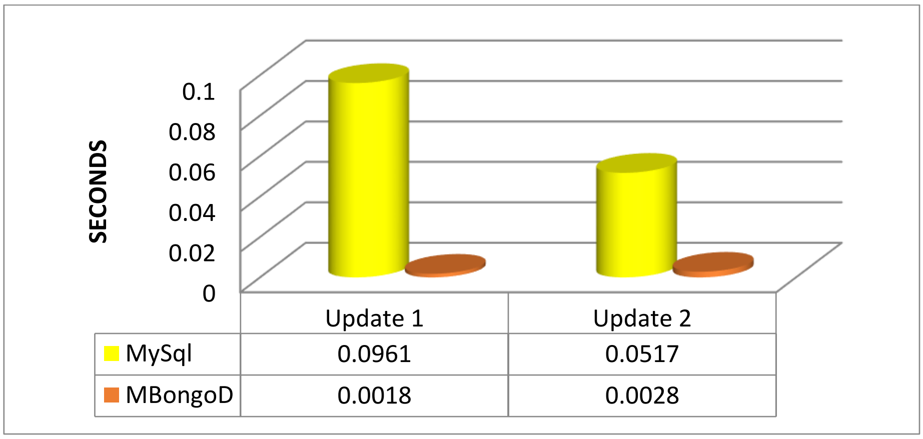
Fig 7. MySQL vs. MongoDB select.

As shown in Fig 7 the results of the first test, the query for all records took 0.31 seconds for the MySQL database, while the MongoDB database took 0.0017 seconds in the second test, the query result for specific records for the MySQL database took 0.719 seconds, and for the MongoDB database, the query result For specific records 0.0009 seconds, so the MongoDB database is the fastest in the case of displaying records in different modes and terms of the number of records tested.

6.2.3 Update

In the database update test, two operations were performed:

1. Username update test
2. Update test in the comment field for the user



	Update 1	Update 2
🟨 MySql	0.0961	0.0517
🟧 MBongoD	0.0018	0.0028

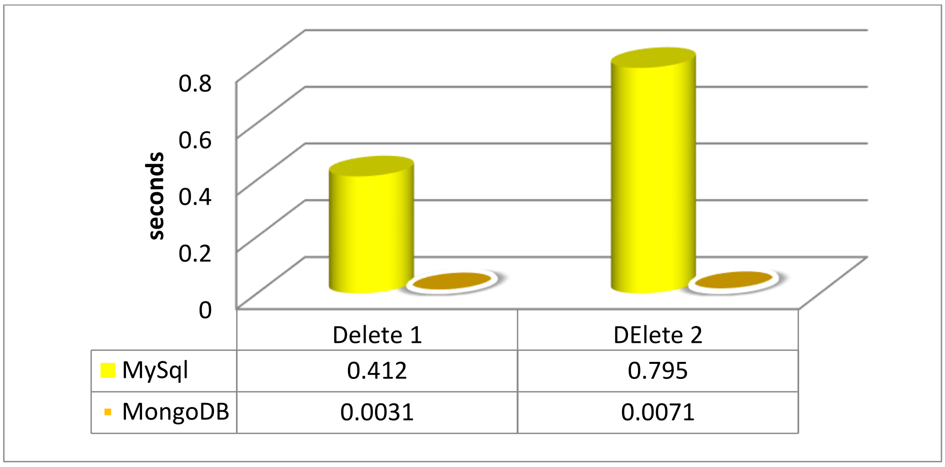
Fig 8. MySQL vs. MongoDB update.

The results shown in Fig 8 showed that the first update process took 0.0961 seconds in the MySQL database, and in the MongoDB database, the update process took 0.0018 seconds, while the second query needed 0.0517 seconds in the MySQL database to complete the update process, and MongoDB database needed to 0.0028 seconds, and from here we can say that the update process in MongoDB is better than the two mentioned queries.

6.2.4 Delete

In the process of deleting records, we have two queries:

1. Query deletes all user comments.
2. Query Delete some users' comments.



	Delete 1	DElete 2
■ MySql	0.412	0.795
▪ MongoDB	0.0031	0.0071

Fig 9. MySQL vs. MongoDB Delete.

In the results of the deletion process for both queries, it was shown that the first query completed the deletion process to 0.0412 seconds for the MySQL database, while for the MongoDB database, the time to delete was 0.0031, for the second query, the time to remove comments of specific users in the MySQL database was 0.795 seconds while the MySQL database was the time The deletion process took 0.0071 seconds, and from here it was found that the MongoDB database is the fastest in the deletion process, especially in deleting large data as shown in Fig 9.

6.3 Comparison According to the CAP Theorem Between MySQL vs MongoDB

According to CAP theory, a NoSQL database is in a CA state where it is ensured that all changes made in the database are visible in all nodes at the same time and that the system responds to all requests. However, if some nodes are damaged, the system can crash. But in MongoDB, the system in a CP state ensures that it continues to work and responds to requests even if nodes are corrupted so if the system is in this state, the system continues to work even if a section of the nodes are damaged, and

also ensures that the data is updated or consistent across all nodes But it may not respond to all requests because availability is not guaranteed in this type of database.
Therefore, MongoDB has an advantage when used in most systems that need these features [5][6].

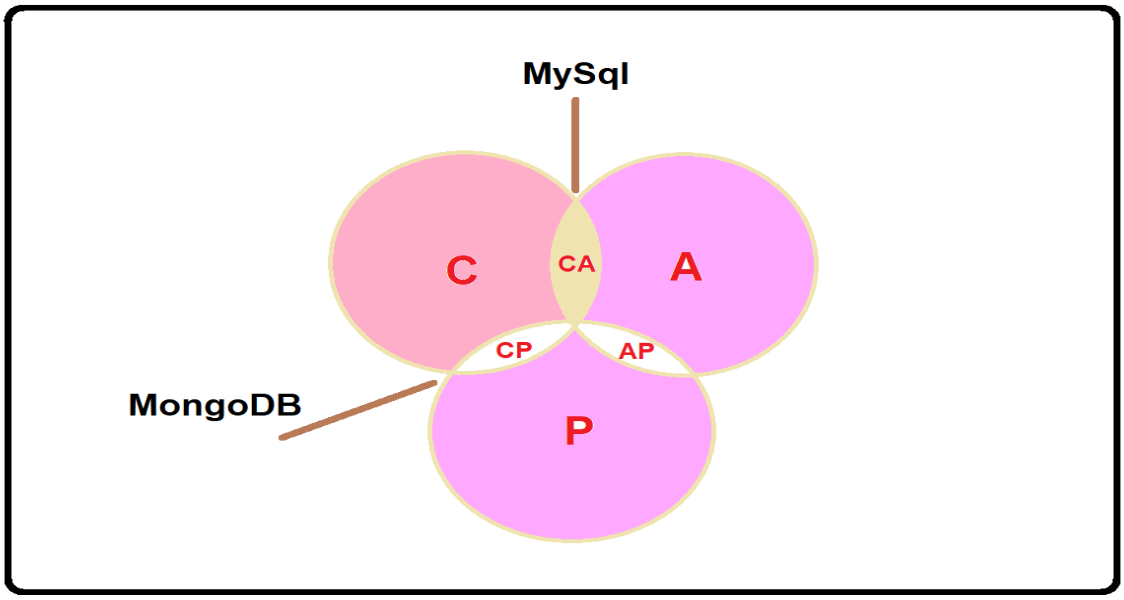


Fig 10. Classification of CAP and Databases.

6.4 Comparison by popularity Between MySQL vs MongoDB

Here we review the difference between MongoDB and MySQL concerning CAP theory MongoDB: The CAP theorem chooses Consistency and Partition Tolerance for the MongoDB database such as CP. What exactly does this imply? This ensures that all clients will be able to see the same clear view of the database. Users of one node, on the other hand, would have to wait for all other nodes to consent before being able to read or write to the database. In this case, continuity takes precedence over availability.

By various dates, a distinction was made between the MySQL database and MongoDB database, which were categorized by popularity [16]. The following parameters were used to determine the popularity of each system:

1. The number of times the system has been mentioned on the internet, as determined by the number of times the system has been mentioned in search engine queries.

2. For the time being, we’re measuring with Google and Bing. To ensure that only valid results are counted. The system has a lot of people interested in it. Google Trends uses to count search frequency.

3. The frequency of technical discussions about the system. On the IT FAQ pages, I used the number of related questions and the number of users interested. Some job offers for the system. They are the offers on the leading professional search engines

4. The number of personal files in professional networks.
5. The importance of social networks. Tweets that refer to the system count.[16]

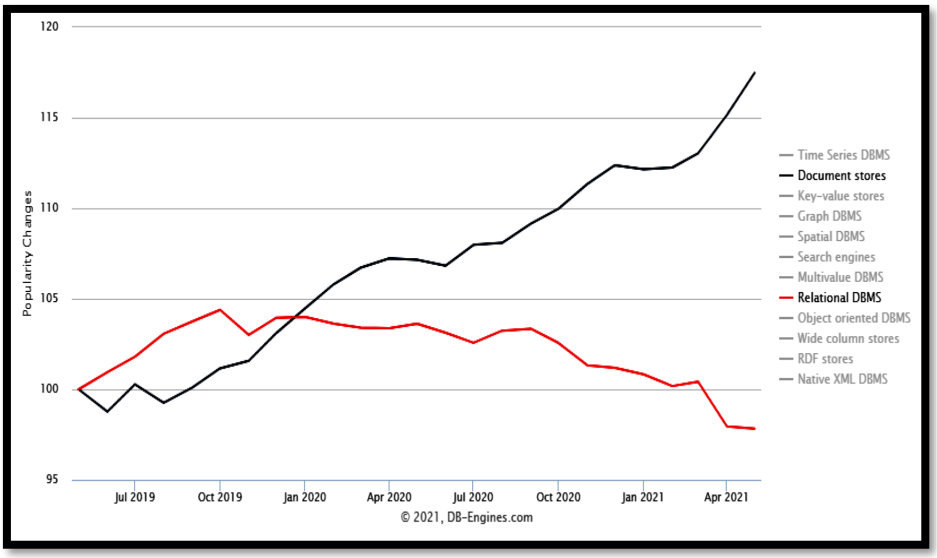


Fig 11. the historical trend of the two categories' popularity (Relational DBMS vs Document stores) from (2013-2021) [17].

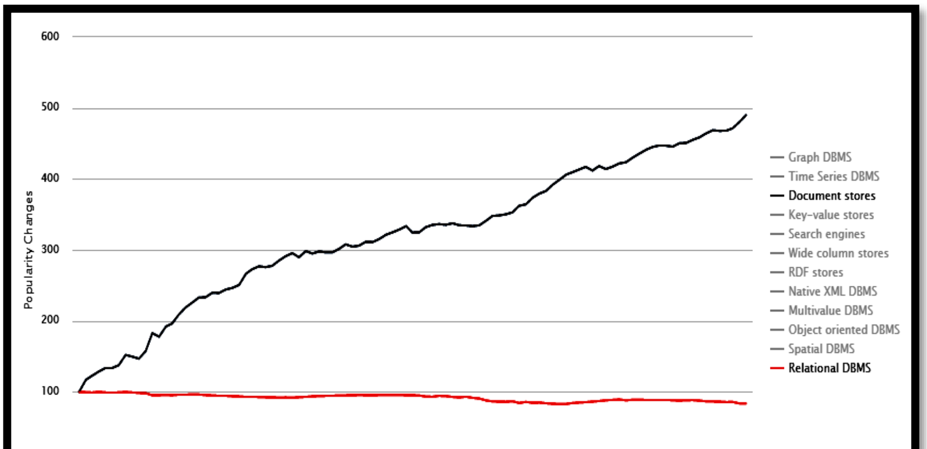


Fig 12. the historical trend of the two categories' popularity (Relational DBMS vs Document stores) from (2019 - 2021) [17].

The graph in Fig 11 shows the historical trend of popularity for the category (DBMS Relational and Document Stores) from (2013-2021). According to the criteria we mentioned for determining popularity, it showed a rise in the popularity of document stores versus relational DBMS.

The graph in Fig 12 shows the historical trend of popularity for the category (DBMS Relational and Document Stores) from (2019-2021). According to the criteria we mentioned for determining popularity, it showed a rise in the popularity of document stores versus relational DBMS.

7 Conclusion

After reviewing the challenges and differences, especially by analysing the query test comparison between MySQL and MongoDB databases, where we used servers with modern versions and a work environment that simulates the impact, we found that one of the reasons that lead us to believe in moving to NoSQL is the high performance in the four processes (input, selection, deletion, and update) which are important elements in the effectiveness of databases. By studying the analysis of the growth of the popularity of MySQL and MongoDB databases (such as discussions or job offers and other considerations), we found that MongoDB is more growing over years (from 2013 to 2021). This is an early indicator that precedes widespread use and that follows a factor Given time, it can be a logical decision to put the non-relational database at the forefront of the choice of enterprise data management, although it is currently facing challenges, according to the data of growth in the technology of this type, the data, and systems used, the increasing popularity and scope of use, the trend towards their use will not be far. After we have reviewed several studies and the ideas provided by a lot of research on selecting the right database between relational and non-relational databases, we found most of them conclude that selecting the right database follows the organization's need for how to manage its data robustly. And its success despite the existence of future problems, solutions, and frameworks that are moving towards modern databases. We found that one of the important reasons not to use non-relational databases is the lack of experience in managing this type compared to relational databases. We suggest taking the inquiries of this study and researching more in different aspects to clarify the future of databases and thus build plans according to an outlook.

References

[1] Chapter 2 - relational database. In Peter Mccaffrey, editor, An Introduction to Healthcare Informatics, pages 17–29. Academic Press, 2020.

[2] Wajid Ali, Muhammad Usman Shafique, Muhammad Arslan Majeed, and Ali Raza. Comparison between sql and nosql database and their relationship with big data analytics. Asian Journal of Research in Computer Science, pages 1–10, 2019.

[3] Gourav Bathla, Rinkle Rani, and Himanshu Aggarwal. Comparative study of nosql database for big data storage. International Journal of Engineering & Technology, 7(2.6):83–87, 2018.

[4] Roman Cereˇsˇn´ak and Michal Kvet. Comparison of query performance in relational a ˇ non-relation database. Transportation Research Procedia, 40:170–177, 2019.

[5] Raif Deari, Xhemal Zenuni, Jaumin Ajdari, Florije Ismaili, and Bujar Raufi. Analysis and comparison of document-based database with sql relational database: Mongodb vs mysql. In Proceedings of the International Conference on Information Technologies (InfoTech 2018), pages 1–10, 2018.

[6] Tuˇgba Beril Doˇgu¸c and Ahmet Arif Aydın. Cap-based examination of popular nosql database technologies in streaming data processing. In 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), pages 1–6. IEEE, 2019.

[7] Benymol Jose and Sajimon Abraham. Performance analysis of nosql and relational database with mongodb and mysql. Materials today: PROCEEDINGS, 24:2036– 2043, 2020.

[8] Sarah H Kamal, Hanan H Elazhary, and Ehab E Hassanein. A qualitative comparison of nosql data stores. Int. J. Adv. Comput. Sci. Appl, 10(2):330–338, 2019.

[9] Gianina Mihai. Comparison between relational and nosql database. Annals of the University Dunarea de Jos of Galati: Fascicle: I, Economics & Applied Informatics, 26(3), 2020. 13

[10] Anasuodei Moko and Prince Oghenekaro Asagba. Big data and nosql database architecture: A review. 2020.

[11] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. Big data technologies: A survey. Journal of King Saud University-Computer and Information Sciences, 30(4):431–448, 2018.

[12] Sowndarya Palanisamy and P SuvithaVani. A survey on rdbms and nosql database mysql vs mongodb. In 2020 International Conference on Computer Communication and Informatics (ICCCI), pages 1–7. IEEE, 2020.

[13] Shaymaa Ahmed Razoqi. Reasons of the transformed toward nosql database and its data models. JOURNAL OF EDUCATION AND SCIENCE, 29(2):82–100, 2020.

[14] Kosovare Sahatqija, Jaumin Ajdari, Xhemal Zenuni, Bujar Raufi, and Florije Ismaili. Comparison between relational and nosql database. In 2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO), pages 0216–0221. IEEE, 2018.

[15] B Sirish Shetty and KC Akshay. Performance analysis of queries in rdbms vs nosql. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), volume 1, pages 1283–1286. IEEE, 2019

[16] Link of related {https:// medium .com /edureka/ sql-vs-nosq l-db-5d9b69ace6ac}

[17] Link of related {https://db-engines.com/en/ranking_definition}