

Performance Comparison between SQL and NoSQL in Terms of Use with Big Data

www.doi.org/10.62341/ekaa3035

Esam. Miftah. Abdalnabi. Aboudoumat¹
Khalifa. Mohammed. Ballam²
Ashraf. Faraj. Saed Albarki³
AbdelAziz. Ibrahim. Radwan. Bader⁴

1. Computer Department, College of Science and Technology, Qumins.
2. Computer Department, College of Science and Technology, Qumins.
3. Computer Department, College of Arts and Sciences Qumins,
University of Benghazi
4. Computer Department, Higher Institute of Engineering Technologies,
Benghazi.

Esam.mouftah@gmail.com, khalifabalam@gmail.com,
Ashraf.sade@gmail.com, Abdelazizbader1@gmail.com

Abstract:

Data nowadays is growing and increasing dramatically, which has become more complex to store and process effectively using traditional database management tools. NoSQL databases have now become highly popular because of their pursuit of improving performance from traditional SQL databases, which has led to the replacement of Traditional database management systems, with NoSQL databases.

This paper aims to study NoSQL databases of various types and evaluate them in big data analysis, which helps users of this type of databases obtain an understanding of the strengths and weaknesses of their various methods to support applications that deal with huge amounts of data and compare traditional SQL databases with NoSQL databases.

We developed prototypes for the two systems in the Visual Basic language.net programming language, and we designed the

databases, non-relational databases of the MongoDB type to represent the NoSQL database, and relational databases of the SQL Server type were used to represent relational databases, and a data set containing 21,965 related records was used with bank statements. We compared the time of loading data for the two systems and performed the operations of adding, retrieving data, modification, and deletion, to test and evaluate performance and speed of implementation.

Keywords: Database, NoSQL, SQL Server, Big Data.

مقارنة الأداء بين SQL و NoSQL من حيث الاستخدام مع البيانات الضخمة

عصام مفتاح عبد النبي بودومات¹، خليفة محمد بلعم²، أشرف فرج سعد البركي³، وعبد العزيز إبراهيم رضوان بدر⁴

1. قسم الحاسوب، كلية العلوم والتقنية، قمينس.
 2. قسم الحاسوب، كلية العلوم والتقنية، قمينس.
 3. قسم الحاسوب، كلية الآداب والعلوم قمينس، جامعة بنغازي.
 4. قسم الحاسوب، المعهد العالي للتقنيات الهندسية، بنغازي.
- Esam.mouftah@gmail.com, khalifabalam@gmail.com,
Ashraf.sade@gmail.com, Abdelazizbader1@gmail.com

الملخص:

إن البيانات في وقتنا الحاضر تنمو وتتزايد بشكل كبير، مما أصبحت أكثر تعقيدا في تخزينها ومعالجتها بشكل فعال باستخدام أدوات ادارة قواعد بيانات التقليدية .حيث اصبحت الان قواعد بيانات NoSQL ذات شعبية عالية نتيجة ماحقته في تحسين الاداء من قواعد بيانات التقليدية SQL، مما ادي الي استبدال أنظمة إدارة قواعد البيانات التقليدية بقواعد بيانات NoSQL. تهدف هذه الورقة الي دراسة قواعد بيانات NoSQL بمختلف انواعها وتقييمها في تحليل البيانات الضخمة مما تساعد المستخدمين لهذا النوع من قواعد البيانات

في حصول علي فهم للنقاط القوة والضعف في اساليبها المختلفة من اجل دعم التطبيقات التي تتعامل مع كميات ضخمة من البيانات، ومقارنة قواعد بيانات SQL التقليدية مع قواعد بيانات NoSQL .

قمنا بتطوير نماذج أولية للنظامين بلغة البرمجة Visual Basic language.net، وصممنا قواعد البيانات قواعد البيانات غير العلائقية من نوع Mongo DB لتمثيل قاعدة بيانات NoSQL وتم استخدام قواعد البيانات العلائقية من نوع SQL Server لتمثيل قواعد البيانات العلائقية، وتم استخدام مجموعة بيانات تحتوي على 21,965 سجلاً يتعلق بالبيانات المصرفية. لقد قمنا بمقارنة وقت تحميل البيانات للنظامين وتنفيذ عمليات الإضافة، استرجاع البيانات، التعديل والحذف، لاختبار وتقييم الأداء وسرعة التنفيذ. **الكلمات المفتاحية:** قاعدة البيانات، NoSQL، SQL Server، البيانات الضخمة.

1. Introduction

In this era, we depend on databases and database systems so much that they have become an essential component of daily life.

And Due to the great development witnessed by the world and the great need to deal with and store data in an appropriate way for reference when needed, and given the great challenges that the world faces in choosing the appropriate methods for storing this data over time and the techniques used in the process of storing it until we got to the need to deal with a huge volume of data in Various fields such as social networking sites, banking transactions, e-marketing, air transport, and various fields of scientific research and medicine. To get the best results and many other areas that transfer millions of data in a short time and as a result of this great development in the field of data storage, which has created many challenges, the most important of which are security challenges that must be preserved. This data and the need for the emergence of new types of data warehouses commensurate with this development, which differ greatly in the way they are built, Methods of storing data and its internal structure. This paper will discuss the methods used to save and process big data and their advantages.

2. Related Work

Many studies deal greatly with the importance of big data and how to benefit from the process of storing it in a way that allows us to deal with it, store it and review it. The most important of these studies are as follows:

- [Dr. Ali bin Dheeb Al-Alkalbi – 2018]: presented a study conducted at King Saud University entitled Big Data and Decision-Making at King Saud University: An Evaluation Study of the Itqan System. Does the existing system meet the development-taking place in this field to come out with a set of results that contribute to the development of systems that deal with big data? The researcher came out with a set of results and recommendations, the most important of which is the proposal of appropriate predictive models to be included in the operations of the proficiency system. The study also showed, according to the results, that the university has a huge data volume. It is very unorganized, and therefore it is difficult for decision-makers to analyze it and obtain results from it that help them in the decision-making process due to the lack of software systems to analyze and retrieve them within the system in the university [1].

- [Tariq N. Khasawneh and others-2020]: presented a study to compare the types of database systems entitled: SQL, NewSQL, and NoSQL Databases: A Comparative Survey After conducting this study, the researchers concluded that non-relational databases are a viable alternative to traditional relational database management systems, especially with the ever-increasing volume of data. Also known as a "NoSQL" database system, it is a scalable management system that allows handling such volumes of data [2].

- [Soarov Chakraborty et al. - 2021]: published a study entitled: Performance Comparison for Data Retrieval from NoSQL and SQL Databases: A Case Study for COVID-19 Genome Sequence Dataset, where the study touched on the use of NoSQL databases for storing and processing unstructured and semi-structured data types due to the improved performance of this type

of database over SQL databases, where huge amounts of human genetic data were queried for a group of COVID-19 data, and the MY SQL database was used SQL and Cassandra and Mongo DB databases are of NOSQL type. The study showed that Cassandra and Mongo DB databases were performing well compared to MySQL in the case of unstructured data [3].

- [Wajid Ali and others – 2019]: published a study entitled: Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. In this study, we discuss a comparison between traditional SQL databases and NOSQL databases to deal with big data, data models, and data store types, features and characteristics of these sad ones, and the advantages and disadvantages of SQL and NoSQL. The four data models for NOSQL used in big data analysis were identified. These results were understandable. It is easy to implement and uncomplicated to improve SQL problems in analyzing big data, and the study determined that NoSQL is a great tool for solving the problem of data availability, in a SQL database that needs data that fits in the tables and if your information does not fit in the table, then at this stage, you must restructure the database data again [4].

3. SQL relational database and NoSQL database systems

SQL databases are called relational databases, because they use SQL queries to store, retrieve, query, and analyze data, and they are called relational databases because the stored data is very relational, that is, the data stored in this type of database is connected through a predefined schema that displays the relationships between the data [2].

NoSQL databases: An acronym for Not Only SQL, also known as a non-relational database or distributed database, it is a database management system, which does not require data to show high relationships between them, or a pre-defined schema, that allows storing a variety of data types. They can be classified into four main categories [2]:- Key-Value stores, Column-oriented, Document-oriented and Graph-oriented.

3.1 Key-Value stores:

In this method, data is stored as numeric identifiers (keys) and associated values in simple, independent tables ("hash tables") as shown in table 1. Values can be simple text strings or more complex lists and collections such as images and documents. Data lookups can usually only be done by keys, not values, which makes them faster than SQL. Therefore, this data model allows the user to request values based on specific key values. In modern data stores, information scalability is better than consistency [5].

Table 1. Key-Value Store NoSQL Database.

Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Year: 2005 Transmission: Auto

3.2 Column-oriented:

Wide column stores are also called Big Table as seen in figure 1, which is a distributed data storage system developed by Google that indexes its search and is very close to the storage method used in relational database systems because it stores data in the form of columns, which makes it accommodate attributes for each key [5].

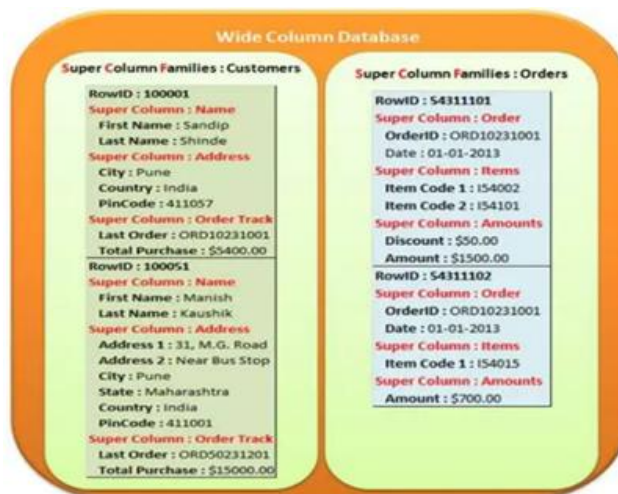


Figure 1. Wide-Column Store NoSQL Database

3.3 Document-oriented:

The document-based database is a no relational database. It is used to store data of various types, whether they are text strings, numbers, Boolean values, arrays, or objects in the form of JSON documents BSON, or XML documents. In addition, this type helps developers store data using the same document form format used in the code.

3.3.1 Key features of the documents database:

- Flexible schema: Documents in the database has a flexible schema. It

Means the documents in the database need not be in the same schema .

- Faster creation and maintenance: the creation of documents is easy and

Minimal maintenance is required once we create the document .

- No foreign keys: There is no dynamic relationship between two Documents so documents can be independent of one another. So, there is

No requirement for a foreign key in a document database.

- Open formats: To build a document we use XML, JSON, and others [6].

3.4 Graph-oriented:

Graph Databases: This type is used to store data within a graph consisting of nodes as shown in figure 2, that store data entities and links that express the relationships between these nodes.

3.4.1 Key features of graph database:

- Easy to identify the relationship between data using links.
- The Query's output is real-time results.
- The speed depends upon the number of relationships among the database Elements [6].

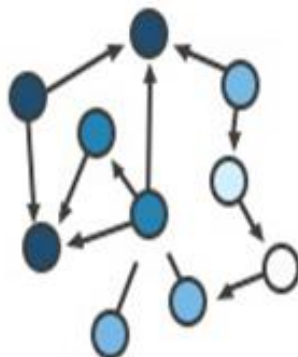


Figure 2. Graph-oriented NoSQL Database.

4. When do you use SQL and NoSQL databases?

When it comes to databases, there is no one-size-fits-all solution. This is why most companies rely on both types of databases for different use cases.

Although NoSQL database has become popular due to its speed and scalability, there are cases where SQL databases perform better. So choosing the right database is important.

4.1 Reasons to use SQL:

1. When you need ACID compliance.

2. Your data is organized and the structure hardly changes.
3. When joining and executing complex queries.

SQL-enabled companies:

PostgreSQL, MySQL, Oracle DB, Mariadb, Sqlite

Companies support No SQL:

Mongo DB, Couch DB, Cassandra, Redis, HBase... etc

4.2 Reasons to use NoSQL:

1. To store large amounts of data.
2. Need scalability.
3. Rapid development.
4. The data is not structured.

4.3 Comparison of SQL and NOSQL:

When to choosing a database, the biggest decision is picking a relational (SQL) or non-relational (NoSQL) data structure. While both databases are viable, options still there are certain key differences between the two that users must keep in mind when making a decision [6]:

- Type:

- SQL called a Relational Database
- NOSQL: called a non-relational or distributed database System.

- Storage method:

- SQL these databases are not suited for hierarchical data storage.
- NOSQL these databases are best suited for hierarchical data storage.

- Schema:

- SQL these databases have fixed or static or predefined schema.
- NOSQL: They have a dynamic schema.

- Querying:

- SQL these databases are best suited for great complex queries.
- NOSQL these databases are not so good for complex queries.

- Scalability:
 - SQL is Vertically Scalable.
 - NOSQL is horizontally scalable.
- Reliability (ACID):
 - SQL Follows ACID property.
 - NOSQL Follows CAP (consistency, availability, partition tolerance).

5. Results and discussions

Experimental results are explained in this section, we have developed prototypes of the two systems in Visual Basic language.net programming language, and database design Non-Relational Databases and Relational Databases

The Mongo DB database was used to represent the NoSQL Database, the SQL Server database was used to represent the relational databases, and a dataset containing 21,965 records related to banking data was used. It was obtained from [7], where that data was imported to a relational database (SQL) which is SQLSERVER and the same data was imported to a non-relational type database (NONSQL) in (JSON) format, which is MongoDB where the data table contains the following fields:

- Table/document: with the columns: id, name, country, date time stars, review head, review body. Figure 3 shows the data retrieval time for the two databases.

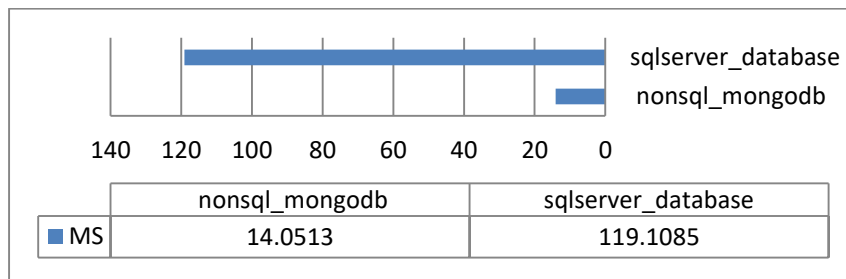


Figure 3. The data retrieval time Mongo DB and SQL SERVER

We have compared the data load time for SQL server _database, Nonsql_mongodb; the data load time was calculated as 119.1085, 14.0513, of the total data, and the loading time is taken to load 21965 Records from a database, as shown in Figure 3. Nonsql_mongodb performs better than SQL Server as SQL server has taken much time and performs worse.

Specific operations were carried out on the Mongo DB database and the SQLSERVER database represented in the following operations: addition, data retrieval, Modification and deletion, to test and evaluates the performance and speed of implementation through the two previous databases. These operations were carried out on a computer that has the following specifications: Windows 10, 64-bit, processor Intel Core I5 (1.60GHz), 8 GB RAM memory.

The focus was on using the features of the Non-Relational Database MongoDB and comparing it with the Relational Database SQL Server by applying four basic operations so that we can evaluate the performance and speed of implementation by the MongoDB database and the SQL Server database. The operations used are:

- Insert Operation.
- Select (query) Operation.
- Update Operation.
- Delete operation.

5.1 Insertion operation:

The Insert has the following SQL Server syntax and MongoDB syntax:

The following query statement was used to add a record in SQLSERVER :

```
- sqlstr = "insert into rev_for_cll (_id, name1, country, date_time,  
stars, review_head, review_body) values  
(@p1,@p2,@p3,@p4,@p5,@p6,@p7)"  
- dcom = New SqlCommand(sql, con_string)  
dcom.ExecuteNonQuery()
```

In addition, use the following query statement to add a record in MongoDB:

```
- dbcollection.Insert(sql)
```

After executing the previous query statements three times on SQLSERVER and MongoDB, the following results shown in Figure 4 were obtained:

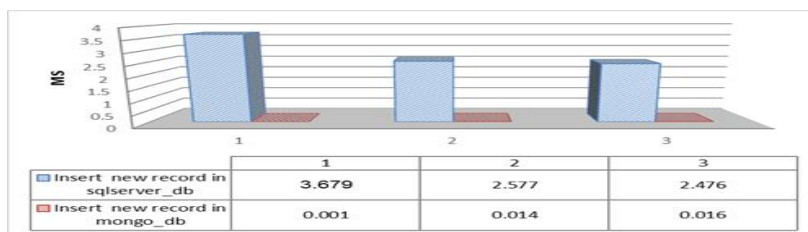


Figure 4. SQL Server vs. MongoDB insert.

Through the results shown in Figure 4, we notice that the highest performance time obtained for the addition process is 3.679 for the SQL Server data set, and the lowest performance time obtained is 0.001 for the MongoDB database.

5.2 Query operations (select):

The first selection has the following SQL Server syntax and MongoDB syntax:

Use the following query statement to display all records (21965 records) from SQLSERVER:

```
sql = "select * from rev_for_cli"
```

Use the following query statement to display all records (21965 records) from MongoDB

```
sql = dbcollection. FindAll()
```

After executing the previous query statements three times to display 21965 records from the database, SQLSERVER and MongoDB shows the following results shown in Figure 5 are obtained.

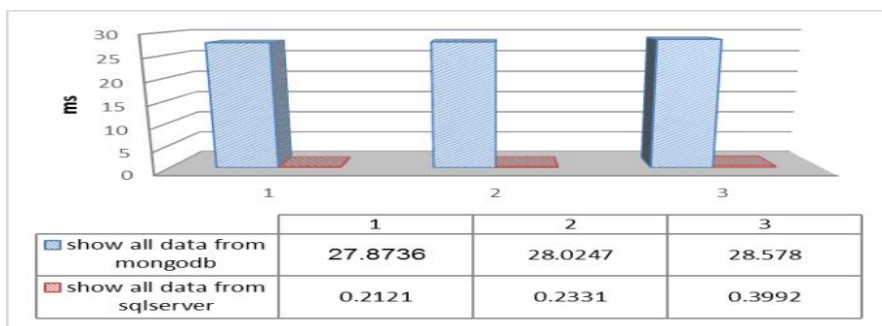


Figure 5. SQL Server vs. MongoDB select_ show all data (21965 records)

Through the results shown in Figure 5, we notice that the highest performance time obtained for displaying all data is 28.578 for the SQL Server data set, and the lowest performance time obtained is 0.2121 for the MongoDB database.

The second selection has the following SQL server syntax and MongoDB syntax:

Use the following query statement to return a record with primary key `_id` from SQLSERVER:

```
sql = "select * from rev_for_cll where id=" & textbox1.text
```

Use the following query statement to display all records from MongoDB

```
Dim sql As New QueryDocument
sql("_id") = TextBox1.Text
scol = dbcollection.Findone(sql)
```

After executing the previous query statements three times to display the data using the primary key `_id` from the database SQLSERVER and MongoDB, the following results shown in Figure 6 were obtained:

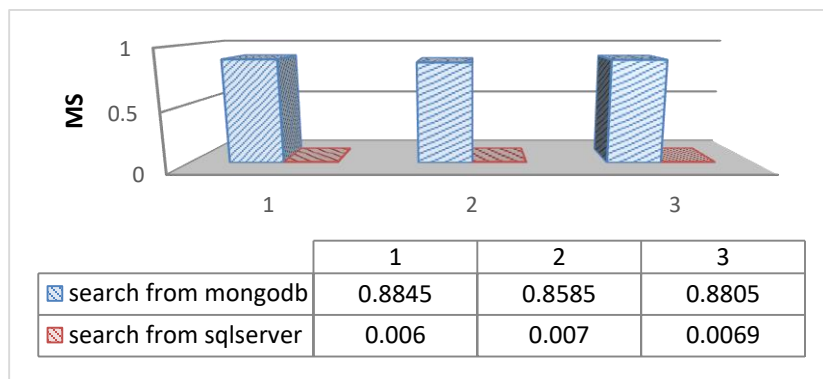


Figure 6. SQL Server vs. MongoDB select data using the primary key (_id)

Through the results shown in Figure 6, we notice that the highest performance time obtained for displaying data by the primary key is 0.8845 for the SQL Server data set, and the lowest performance time obtained is 0.006 for the MongoDB database.

The three select have the following SQL server syntax and MongoDB syntax:

Use the following query statement to display data by name from SQLSERVER:

```
sql = "select * from rev_for_cll where name =" + TextBox1.Text + " "
```

Use the following query statement to display all records from MongoDB

```
Dim sql As New Query Document
```

```
sql("name") = TextBox1.Text
```

```
scol = dbcollection.Findone(sql)
```

After executing the previous query statements three times to display the data using the primary key _id from the database SQLSERVER

and MongoDB, the following results shown in Figure 7 were obtained:

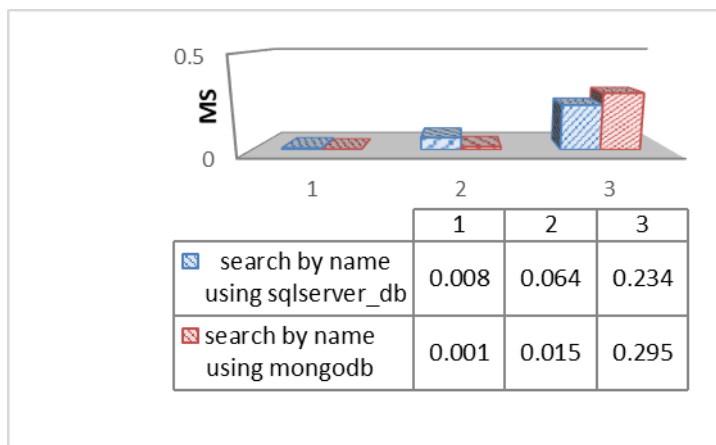


Figure 7. SQL server vs. MongoDB select data using name

Through the result ranking in Figure 7, we notice that the highest performance time obtained for displaying data by name is 0.234 for the SQL Server data set, and the lowest performance time obtained is 0.001 for the MongoDB database.

The fourth selection has the following SQL Server syntax and MongoDB syntax:

Use the following query statement to display data by date from SQLSERVER:

```
Dim da As Date = DateTimePicker1.Value.Date
```

```
sql = "select * from rev_for_cll where date_time = #" + da + " # "
```

Use the following query statement to display data by date from MongoDB

```
Dim sql As New QueryDocument
```

```
Dim date1 As Date = DateTimePicker1.Value.Date '
```

```
sql("date_time") = date1
```

Dim scol = dbcollection.Find(sql)

After executing the previous query statements three times to display the data using the date from the database SQLSERVER and MongoDB, the following results shown in Figure 8 were obtained:

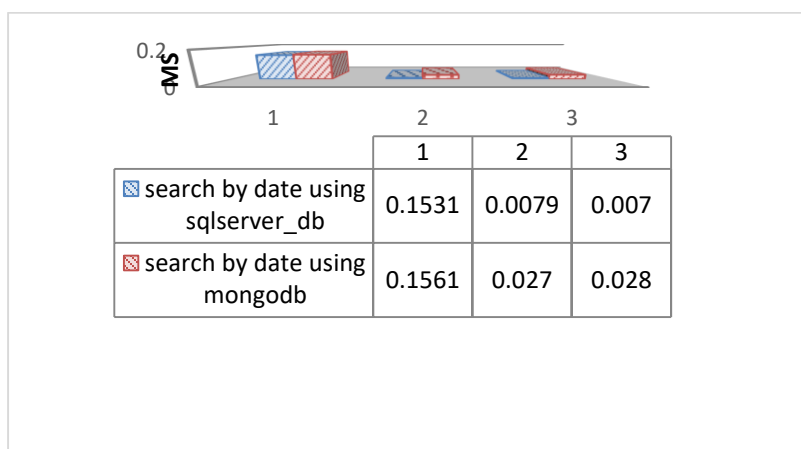


Figure 8. SQL Server vs. MongoDB select data using date

Through the results shown in Figure 8, we notice that the highest performance time obtained for displaying data by date is 0.1531 for the SQL Server data set, and the lowest performance time obtained is 0.027 for the MongoDB database.

5.3 Update operation:

To test the update operation in the databases, the following two queries were executed:

Use the following query statement to modify data by name from SQLSERVER:

```
sql = "update rev_for_cll set
name1=@p2,country=@p3,date_time=@p4,stars=@p5,review_he
ad=@p6,review_body=@p7 where @p1=" & TextBox1.Text & ""
```


Use the following query statement to modify data by name from: MongoDB

Dim sql As New QueryDocument

sql("_id") = textbox1.tex

dbcollection.Update(sql)

After executing the previous query statements three times to modify the data using the key from the SQLSERVER and MongoDB databases, the following results shown in Figure 9 were obtained:

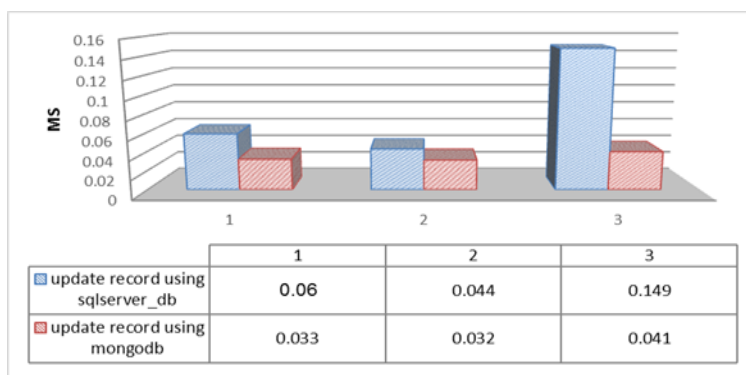


Fig. 9. SQL Server vs. MongoDB Update

Through the results shown in Figure 9, we notice that the highest performance time obtained for displaying modify data was 0.149 for the SQL Server data set, and the lowest performance time obtained is 0.032 for the MongoDB database.

5.4 Delete operation:

To test the Delete operation in the databases, the following two queries were executed:

Use the following query statement to delete data by id from SQLSERVER:

```
sql = "delete from rev_for_cll where _id= " + TextBox1.Text + "
```

Use the following query statement to modify data by id MongoDB:

```
Dim sql_del As New QueryDocument
```

```
sql_del("_id") = TextBox1.Text
```

```
dbcollection.remove(sql_del).
```

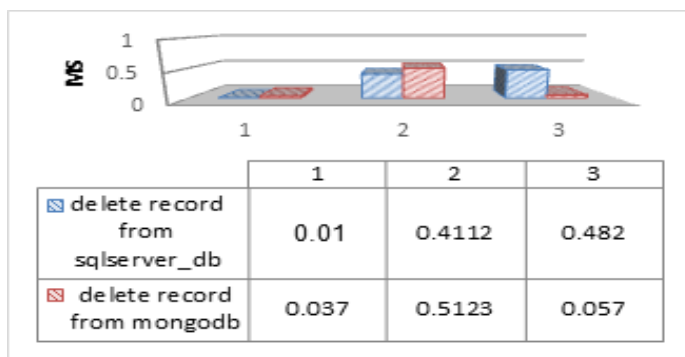


Fig.10. SQL Server vs. MongoDB delete

After executing the previous query statements three times to delete the data using the key from the SQLSERVER and MongoDB databases, the following results shown in Figure 10 were obtained: Through the results shown in Figure 10, we notice that the highest performance time obtained for displaying deleted data is 0.4112 for the SQL Server data set, and the lowest performance time obtained is 0.037 for the MongoDB database.

6. Conclusion

After studying and making a comparison between SQL and NOSQL Database in dealing with big data, it showed the shortcomings of SQL relational databases in dealing with big data and the difficulty of retrieving them faster, which led to the urgent need for the emergence of new methods and types of databases that improve dealing with this volume of data smoothly and without Problems It

is the non-relational database systems NoSQL, which is the best solution until now to deal with this huge data.

7. References

- [1] Dr.. Ali bin Theeb Al-Aklabi, Big data and decision making in King Saud University: an evaluation study for etqan system, 2018, Journal of Information Studies and Technology.
- [2] Tariq N. Khasawneh, Mahmoud Alsahlee, and Ali Safieh, SQL, NewSQL, and NoSQL Databases: A Comparative Survey, International Conference on Information and Communication Systems (ICICS) April 2020.
- [3] Soarov Chakraborty, Shourav Paul, and K. M. Azharul Hasan, Performance Comparison for Data Retrieval from NoSQL and SQL Databases: A Case Study for COVID-19 Genome Sequence Dataset, International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST'21) February 2021, Page 327-Page 331.
- [4] Wajid Ali, Muhammad Arslan Majeed, Ali Raza, and Muhammad Usman Shafique, Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics, Asian Journal of Computer Science and Information Technology , October 2019.
- [5] A B M Moniruzzaman and Syed Akhter Hossain, NoSQL Database: New Era of Databases for Big Data Analytics - Classification, Characteristics and Comparison, International Journal of Database Theory and Application Vol. 6, No. 4. 2013.
- [6] Online Banking/Financial Review Dataset. Last accessed 14 Mar. 2024.
<https://www.kaggle.com/datasets/yanmaksireviews-data-for-classification-model?resource=download>