International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

June 2023 Issue 55 Volume 15 Number 2 Pages 155-161

ANALYSIS AND COMPARISON OF NOSQL DATABASES WITH RELATIONAL DATABASE: MONGODB AND HBASE VERSUS MYSQL

M. Sais A. Sghir N. Rafalia J. Abouchabaka

Department of Computer Science, Computer Research Laboratory (LaRI), Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, manar.sais@uit.ac.ma ayoub.sghir@uit.ac.ma, arafalia@yahoo.fr abouchabaka3@yahoo.fr

Abstract- The development of the Internet industry, the advent of new web technologies, and the trend toward what is now known as web 3.0 are accompanied by an increase in data generation, and new challenges regarding the storage and processing of Big Data. Since this Big Data entered the IT sector, the use of relational systems is limited, and the ability to handle the high volume and variability of data types has become an uphill task for traditional systems. As a result, an effort is required to store and process this massive data in order to prevent information loss and ensure better management of the Big Data. Different approaches have emerged, including the NoSQL Databases. Various NoSQL databases with different types are able to control the exponential growth of data. NoSQL systems are promising solutions for Big Data storage and management challenges due to their many qualities, such as the high availability and scalability of distributed systems that require rapid access time and cannot tolerate any delay in the event of failure The purpose of this paper is to compare and test the performance of two generations of databases: HBase and MongoDB, a more sophisticated generation known as NoSQL, and MySQL, a more traditional relational generation.

Keywords: Big Data, Storage, Relational Databases, NoSQL, MySQL, HBase, MongoDB.

1. INTRODUCTION

Relational databases have long been the preeminent, if not the only, method of storing and managing data [1]. However, in recent years, we have witnessed an explosion of data due to technological advances and the expansion of web applications. This evolution of data is called big data. The term refers to data sets characterized by very large size (volume), complexity (variability), and growth rate (velocity) which is difficult for it to be analyzed and managed with traditional tools [31]. The demands for rapid data storage and processing have increased dramatically, and traditional technologies and tools, including relational databases and data processing software, are struggling to manage or analyze this massive amount of data. The traditional systems have some limitations and can be problematic in some use cases [2], [3]. These constraints prompted the development of NoSQL, an efficient and easier method of dealing with enormous volumes of data. NoSQL [4] is a non-relational database system that is fundamentally different from a traditional relational database that meets all the above requirements. The system contains multiple database types to accommodate different data structures and provide greater scalability, better data models, and extremely superior performance.

MongoDB is a relatively new database model that belongs to the NoSQL database paradigm and is an alternative to relational databases. It is a general-purpose, distributed, document-oriented storage system. Collections are the structural units of the MongoDB database, each collection consists of a set of documents, and each document is an ordered set of keys and associated values [5]. It is also based on the principles of sharding and replication to ensure fault tolerance. One of the goals of this paper is to present the MongoDB database and compare its performance with a classical relational database (MySQL).

The HBase database is another more widely used solution. It is a non-relational, open-source, multi-dimensional database capable of storing and processing massive amounts of data (nearly a billion rows of data). HBase is a columnar storage-based system Hadoop uses HDFS as a Big Data file storage system, and for MapReduce as a core component of Big Data processing and Zookeeper as a collaborative service [6]. One of the purposes of this paper is to compare the storage and operating principles of HBase with the relational database MySQL and to present some storage features of HBase.

The structure of the paper is as follows. Related work is presented in section 2. Section 3 is devoted to a literature review, where we analyze the main aspects of the two database generations, relational and NoSQL, and compare the storage structure of the MySQL database with the two NoSQL paradigms; HBase and MongoDB the other is to testing the performance of MongoDB and MySQL during database query activities. Finally, we conclude with Section 5.

In Section 4, we present two experiments, one is devoted to comparing the storage pattern of HBase with MySQL and to testing the flexibility of HBase, and

2. RELATED WORK

MySQL is one of the most frequently used relational databases. The major problem with these classic databases is the lack of scalability, as the data must fit into predefined tables or structures. In the context of Big Data, complex data cannot be handled by conventional models and databases, and structured database efficiency changes in an unstructured environment, especially when consistency and scalability become an issue in the operation of the transactional database management system [7]. HBase and MongoDB are two of the NoSQL databases, with different systems and storage structures for processing masses of unstructured data.

Many white papers and blogs have attempted to compare and analyze a standard SQL database with a column store-based database (HBase). The authors of [8] attempt to introduce the HBase database, some of its features, its relationship with Apache Hadoop, and its limitations in managing distributed data compared to other NoSQL database management systems. The second part of this work is devoted to the comparison of the temporal performance of HBase databases and the traditional MySQL database when data volume increases.

In [9], the authors highlight some of MongoDB's query support features, as well as a careful comparison of the query language, transactions, and security. In the same context, the paper [10] focuses on evaluating the efficiency. The results of performance tests between MongoDB and MS-SQL revealed that MongoDB surpasses the latter database management system.

Similar work may be found in [11], where the authors present a performance comparison of two databases from two different generations, MySQL and MongoDB. After analyzing the results of insert and search operations of 25000 records. They claim that non-relational databases outperform MySQL, while relational databases have horizontal scaling problems.

3. REVIEW OF LITERATURE

Relational data models such as SQL databases provide specific mechanisms to manage data consistency and concurrency control in the system, ensuring accurate output for concurrent operations, and ultimately ensuring ACID properties of transactions. Unlike NoSQL databases that rely on the three components as a condition in their extension of the CAP theory. The term NoSQL means "Not Only SQL," but that doesn't mean it's antithetical to SQL; rather, it's quite good compared to relational databases that only rely on structured data to store them as database rows. With this, we can create compatibility between SQL and NoSQL systems [12].

3.1. Relational DBMS

As computing grows in large organizations, there is a growing need to organize data into models, allowing a clearer separation between the logical representation of data and the physical organization. The relational database or RDBMS was born, and since it was developed by Edgar Codd in 1970 [12], it has become the dominant database management model, widely used in most applications to store and restore data, and has been the primary data storage solution for the IT industry.

A relational database system (RDBMS) [13] is a database with a compact and uniform structure that stores data in the form of rows and columns in a structured format. This storage structure is used to locate the required data or specified values in the database. This database supports Controlled Structured Query Language (SQL), which has a fixed schema that covers the storage of related data. Most SQL databases provide four essential data management features known as ACID properties (atomicity, consistency, isolation, and durability):

- **Atomicity:** A transaction is an atomic unit of processing in a database., which means that all changes to the data are made completely, or not at all.
- **Consistency preservation:** Consistency means that a transaction must obey the data integrity constraints of the database.
- **Isolation:** Isolation means that a transaction must appear to execute independently of other transactions, even if many transactions are executed simultaneously.
- **Durability or permanency:** Means that the changes made to the database by committed transactions must be preserved. Failure must not result in the loss of these modifications.

3.2. NoSQL

In a Big Data environment, traditional data models cannot handle complex and unstructured data. To save time and efficiency, many researchers have successfully found several alternative solutions to this problem in different domains and directions [14-16].

NoSQL database is one of the solutions and tools to manage Big Data. Large data sets can be managed more flexibly using NoSQL systems that provide so-called schema-less modeling, a flexible approach to data modeling in which data semantics are embedded in flexible connection topologies, and an associated storage model. NoSQL is based on a flexible model that ensures autonomous data distribution and elastic use of storage, computing, and network capacity without the need to permanently store specific data in a physical location. In addition, data access latency and performance are improved by the integrated data cache [17]. The concept of NoSQL is not well defined and can refer to many different databases. In general, there are four types of NoSQL databases, namely key-value databases, document databases, columnar databases, and graph databases. Each database is defined by its properties.

Since relational databases meet the ACID properties. Distributed databases are designed to support three fundamental data processing properties known as the CAP theorem. The CAP theorem states that it is impossible to provide more than two of three guarantees simultaneously [18] as shown in the figure 1. The first is consistency, which states that regardless of the number of copies, the data in the database only exists in one visible state. The second principle is availability, which states that the data must be accessible as long as the system is operational (whether distributed or not). The third option, partition tolerance, refers to the system's ability to function even when server connectivity is unstable.

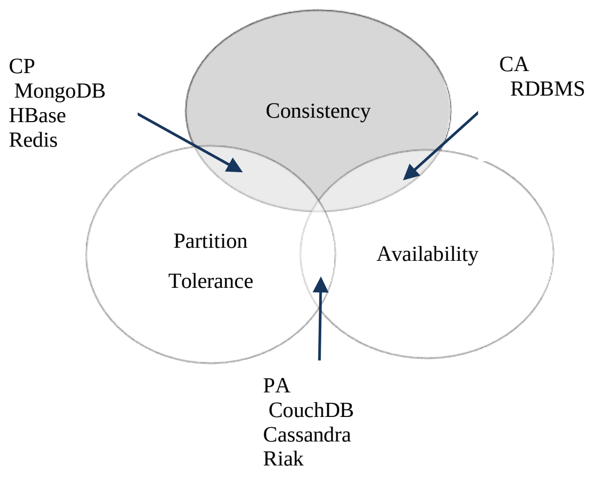


Figure 1. CAP theorem with supported NoSQL Databases

In this work, we will focus on the study of robustness and performance testing of databases. HBase, which is column-centric, and MongoDB, which is document-centric.

3.2.1. Apache HBase

HBase is an open-source, distributed, key-value, and multidimensional NoSQL database [19] It was created for the HDFS distributed storage system [20]. It is written in Java and supported by the Google Bigtable article [21]: A Distributed Storage System for Structured Data. HBase provides Hadoop with all the functions of Bigtable, relies on HDFS to store Big Data, MapReduce for processing and Zookeeper as a collaborative service. HBase is very useful for storing huge amounts of data [22].

HBase is based on the table storage structure, which is composed of rows and columns divided into several column groups, and this data cannot be specified only by row key. Due to its dynamic capabilities, HBase can store and analyze one billion rows of data simultaneously, and this capacity can be expanded at any time [23]. Furthermore, HBase is intended for high throughput and minimal latency and to Allow rapid and random read/write operations on big data collections

An HBase system consists of four main components: HMaster, the ZooKeeper cluster, RegionServers (RS) and HBaseClient (HTable). HMaster is in charge of a variety of administration functions, including monitoring all RegionServer instances in the cluster, resizing regions, and duplicating data across HRegionServers. The task of locating RegionServers servicing a particular range of rows falls to HBaseClient (keys). HRegionServers respond to client requests by retrieving or updating data. Concurrent access to data stored in the HBase cluster is maintained by the ZooKeeper cluster.

3.2.2. MongoDB

MongoDB is a document-oriented, distributed, open-source NoSQL database. MongoDB stores data in JSON documents with a variable structure. In addition, the MongoDB engine can be considered a database engine that efficiently handles JSON objects. Because MongoDB uses dynamic schemas, with which a collection of records and documents can be created without specifying the structure, for example the type of data it will contain. A record's structure can be changed by simply adding new fields or removing existing fields [24], [25]. Data is divided into shards to ensure scalability, and each shard is replicated across a group of nodes for fault tolerance [5].

Documents in the collection do not need to have the same set of fields, as storage arrays, hierarchical structures and other complex structures can define the data model. In addition, the MongoDB paradigm offers a high degree of flexibility that traditional SQL approaches do not, and deploying a MongoDB cluster is simple and fast [26].

4. EXPERIMENTATION AND DISCUSSION

Comparisons between document-based database, column-based database and SQL database can be made from different angles. The first part of this section specifically compares several aspects of the MySQL database and the HBase database. The main objective of this comparison is to extract the advantages of the HBase structure for data storage and manipulation. The second part specifically evaluates and tests the performance of a MongoDB database against MySQL by examining and comparing the performance of four CRUD operations defined in the database.

4.1. Comparative Study: MySQL VS Hbase

Despite the similarity between HBase and the relational database in terms of row and column storage, HBase is column-oriented, whereas relational databases are row-oriented. Both databases store their data separately on hard disks [27]. The purpose of this part is to present two storage architectures, relational storage, and NoSQL storage. The experiment attempts to prove the differences in storage characteristics between MySQL and HBase databases, as well as to highlight the advantages of HBase over relational databases.

4.1.1. Database Migration from MySQL to HBase

In order to extract the advantages of the HBase database over MySQL at the storage level, we try in the first part of this test to migrate a database from a MySQL relational architecture to another with a different architecture (HBase). Retail_db is the database used in this test which contains the following fields.

- Products
- Categories
- Departments
- Customers
- Order_items
- Orders

➢ Import Database from MySQL into HDFS with Sqoop: The migration of the retail_db database from MySQL to HBase is not done in a direct way, it must be transferred to HDFS first. Apache Sqoop is the right solution to easily transfer structured data while preserving the structure. The first command below allows us to create an empty folder in HDFS in which the database will be stored, and the second command allows us to import the database tables to the empty folder already created using Apache Sqoop.

```
$ hadoop fs -mkdie -p /staging/mysql/retail_db
$sqoop import-all-tables --connect jdbc: mysql://localhost:3306/retail_db --username root –password cloudera --warehouse-dir '/staging/mysql/retail_db' -m 1
```

➢ Import Database from HDFS into HBase with Sqoop: The second step is to import the database from HDFS to HBase using Sqoop. Sqoop does not use a default table name when importing into HBase. First of all, it is necessary to create empty tables with column families in HBase, and then run the table import command as shown in the following commands.

```
hbase(main) :001 :0> create 'categoriesdemo', 'categoryfm'
& sqoop import          --conncet jdbc: mysql://localhost/retail_db –table categories -- hbase-table categoriesdemo
column-family categoryfm --hbase-row-key Category_id -m1
```

4.1.2. Flexibility and the Concept of Timestamp in the HBase Database

Hbase storage is based on columns to store data. HBase database systems allow a flexible schema, we can add new data without conforming to a schema model. Whereas MySQL requires a strictly defined schema. The advantage of HBase is that it can dynamically add or remove data columns without affecting performance. In addition, HBase is essentially a write-optimized store, adding new data is cheaper, as it is an append-only operation, and the number of column families that may be created and the number of columns in each family have no theoretical upper limit. Again, it all depends on the needs of the customer, the creator of the HBase table, and the capacity of the cluster. A new family column is added using the "alter" command.

In contrast to other databases, HBase may store the same data in a variety of ways and does not distinguish between writing new rows to a table and updating existing ones, it can store the same data very differently. Each update operation is an addition to a new version of the same row of data. Versions of the same row is distinguished using a time stamp value called a timestamp. The timestamp mechanism is the date to the nearest millisecond that the update was made, it can be applied to record multiple values. Each column family's maximum number of versions saved by HBase can be specified. HBase typically keeps the last three iterations of column values while automatically deleting old versions [28]. Figure 2 shows the three values stored in the 'categoryfm_4:player2' column, each value is specified by a unique timestamp.

```
Hbase(main):035: >   get   'categoriesdemo',  '1 '  , { COLUMN=>
'categoryfm_4:player2', VERSIONS=>4}

COLUMN                              CELL
  categoryfm_4:palyer2              timestamps=1571176505761,
value=aya
  categoryfm_4:palyer2              timestamps=1571176498301,
value=sara
  categoryfm_4:palyer2              timestamps=1571176027187,
value=said
3  row(s)  in  0.0730  seconds
```

Figure 2. The timestamp concept in HBase

4.2. Data Storage Structure and Performance Evaluation test of Document Store (MongoDB) and SQL Database (MySQL)

The benefits of utilizing the non-relational MongoDB database over the relational MySQL database may be compared and shown in a variety of ways. First, each related database's basic data organization and storage concepts are introduced. The capabilities of each database type's query language are compared and examined.

4.2.1. Data Organization: MongoDB vs MySQL

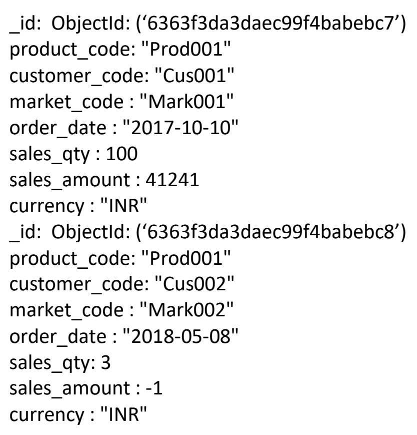
A relational database's data structure is based on a data model that offers a declarative way to express data and queries. Rows and columns are commonly used to organize and store data, creating two-dimensional tables known as relationships. Each relationship has several columns (sets of named attributes), each associated with a specific domain. Tables can be linked together by primary or foreign keys. These distinct identifiers signify the many connections that exist across tables, and these connections are frequently explained by various types of data models. A relation can contain a series of rows called tuples (records). The tuple contains N components, corresponding to the N attributes of the belonging relationship. The schema is represented by the name of the relationship and its collection of attributes[29].

Unlike relational databases, document databases store data in documents, which are collections of key-value pairs. Documents are typically stored in the JSON or BSON format (a combination of "binary" and "JSON," or JavaScript Object Notation), which can be thought of as a binary or numeric representation of JSON documents. Strings are used to represent the keys, and values might be primitive types (such as integers or strings) or structures (such as arrays or objects) [30]. Lists, pointers, embedded arrays, and nested documents are also supported. This facilitates data retrieval and frequently eliminates the need for expensive joins.

In order to compare the storage architecture and test the robustness of the performance, we are running the same database. We tried importing the database into MySQL and then migrating a copy of the database into MongoDB. Table 1 and Figure 3 show how the table 'transactions' is stored in MySQL and MongoDB.

Table 1. The 'transactions' in MySQL

Product _code	Customer _code	Market_ code	Order_ date	Sales _qty	Sales_ amount	currency
Prod001	Cus001	Mark001	2017-10-10	100	41241	INR
Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
Prod002	Cus003	Mark003	2018-04-06	1	875	INR
Prod002	Cus003	Mark003	2018-04-11	1	583	INR
Prod002	Cus004	Mark003	2018-06-18	6	7176	INR
Prod003	Cus005	Mark004	2017-11-20	59	500	USD
Prod003	Cus005	Mark004	2017-11-22	36	250	USD
Prod003	Cus005	Mark004	2017-11-23	39	21412	INR



```
_id: ObjectId: ('6363f3da3daec99f4babebc7')
product_code: "Prod001"
customer_code: "Cus001"
market_code : "Mark001"
order_date : "2017-10-10"
sales_qty : 100
sales_amount : 41241
currency : "INR"
_id: ObjectId: ('6363f3da3daec99f4babebc8')
product_code: "Prod001"
customer_code: "Cus002"
market_code : "Mark002"
order_date : "2018-05-08"
sales_qty: 3
sales_amount : -1
currency : "INR"
```

Figure 3. The table 'transactions' in MongoDB

4.2.2. Performance Evaluation

One of the most crucial factors in selecting a database technology is how well the system performs during CRUD activities (create, read, update, and delete). MongoDB and MySQL were subjected to a series of performance tests (read, update and delete operations) as representative systems based on relational data and document models.

• Read Operation

We have executed the following selection query on MySQL and MongoDB to test the performance of the read operation.

– Selection of all customers in the database with the number of products ordered by each customer.

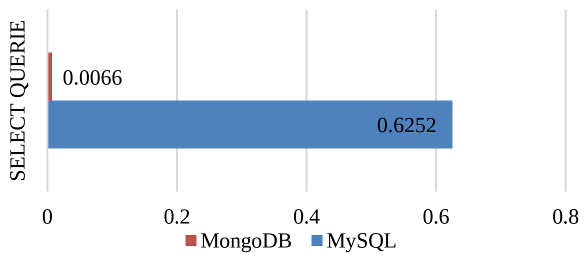


Figure 4. MySQL vs. MongoDB select operation

The select query executes in 0.0066 seconds in MongoDB and 0.6252 seconds in MySQL. From the graph, we notice that MongoDB took less time to execute the select operations than MySQL, as shown in Figure 4.

• Update Operation

To test the update operation in the databases, the following query is executed:

– Update the date of a customer's transaction.

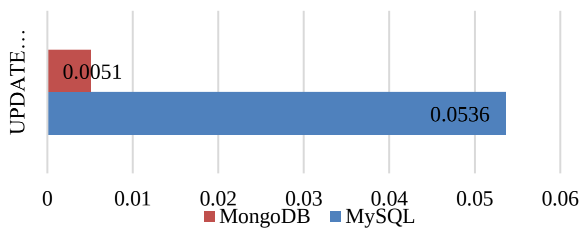


Figure 5. MySQL vs. MongoDB update operation

In MongoDB, the update query takes 0.0051 seconds while in MySQL, it takes 0.0536 seconds. According to the graph, MongoDB requires less time to complete update operations than MySQL, as illustrated in Figure 5.

• Delete Operation

For each database, we have run a delete query in the same manner as the previous procedures mentioned above.

– Delete all orders (transactions) of a customer.

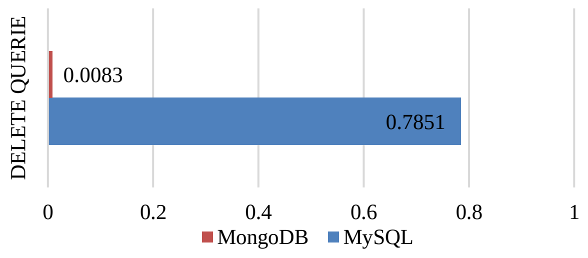


Figure 6. MySQL vs. MongoDB delete operation

The same observation as with the earlier operations. In MongoDB, the delete query takes 0.0083 seconds to complete, but it takes 0.7851 seconds in MySQL. Figure 6 illustrates that MongoDB executes delete operations more quickly than MySQL.

MongoDB is faster than MySQL. MongoDB's speed is due to its document-based storage structure. This means that MongoDB stores data in a single document for an entity and can read and write data much faster. On the other hand, MySQL can exhibit slow performance when handling large amounts of data. MySQL stores tables in a normalized way, and if you want to retrieve or modify data, you have to write and read data in many tables, which increases the server load and affects server performance.

5. CONCLUSION

The current generation of data at an exponential rate poses several problems and challenges to traditional technologies and creates a need for better technologies and media to store and process massive amounts of data. There is a demand for new solutions to store our data quickly and sustainably, process it and restore it easily and efficiently. The traditional relational systems that have dominated for several years are powerless in terms of storage and processing of massive and heterogeneous data. On the other hand, the database movement that has emerged in the last few years manages to offer better performance than the dominant RDBMS.

If we try to review the concepts of relational databases and NoSQL databases, we can understand the motivation behind NoSQL databases and why many large enterprises use them. NoSQL databases differ from traditional databases in many aspects such as structured schema, complexity, transactional approach, big data storage management, and performance, which has led to the use of NoSQL in cloud computing and eventually in data warehouses. This transition from relational to non-relational databases is challenging in many ways. To identify the optimum option for a certain application and construct a non-relational database that has precisely the same query capabilities and operations as the database it is replacing, all viable non-relational database types must be carefully studied.

Two comparative studies have been done in this paper The goal of the first one is to compare the storage structure of the HBase database with MySQL, and also to extract the added value of this type of columnar storage. The second one tests the performance of data query operations between MongoDB and MySQL. HBase shows more flexibility, and MongoDB shows its high performance for data storage and processing compared to MySQL.

ACKNOWLEDGEMENTS

This study is being funded by Morocco's National Center for Scientific and Technical Research (CNRST).

REFERENCES

[1] V. Abramova, J. Bernardino, "NoSQL Databases: MongoDB vs Cassandra", International C* Conference on Computer Science and Software Engineering, pp. 14-22, New York, NY, USA, July 2013.
[2] D. Kumawat, A. Pavate, "Correlation of NOSQL and SQL Database", OSR Journal of Computer Engineering (IOSR-JCE), pp. 70-74, October 2016.
[3] C.A. Gyorodi, D.V. Dumse Burescu, D.R. Zmaranda, R.S. Gyorodi, G.A. Gabor, G.D. Pecherle, "Performance Analysis of NoSQL and Relational Databases with CouchDB and MySQL for Application's Data Storage", Applied Sciences, Vol. 10, No. 23, January 2020.
[4] W. Ali, M.U. Shafique, M.A. Majeed, A. Raza, "Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics", Asian Journal of Research in Computer Science, pp. 1-10, October 2019.
[5] W. Schultz, T. Avitabile, A. Cabral, "Tunable Consistency in MongoDB", Proc. VLDB Endow., Vol. 12, No. 12, pp. 2071-2081, August 2019.
[6] H. Xu, "Research on Mass Monitoring Data Retrieval Technology Based on HBase", J. Phys.: Conf. Ser., Vol. 1871, No. 1, p. 012133, April 2021.
[7] S. Shetty, B.D. Rao, S. Prabhu, "Growth of Relational Model: Interdependence and Complementary to Big Data", International Journal of Electrical and Computer Engineering (IJECE), Vol. 11, No. 2, April 2021.
[8] V.D. Jogi, A. Sinha, "Performance Evaluation of MySQL, Cassandra and HBase for heavy write Operation", The 3rd International Conference on Recent Advances in Information Technology (RAIT), pp. 586-590, March 2016.
[9] S. Soni, M. Ambavane, S. Ambre, S. Maitra, "A Comparative Study: MongoDB vs MySQL", International Journal of Scientific and Engineering Research, Vol. 8, No. 5, pp. 120-123, January 2017.
[10] C.M. Wu, Y.F. Huang, J. Lee, "Comparisons Between MongoDB and MS-SQL Databases on the TWC Website", American Journal of Software Engineering and Applications, Vol. 4, No. 2, May 2015.
[11] B.D. Damodaran, S. Salim, V.M. Surekha, "Performance Evaluation of MySQL and MongoDB Databases", International Journal on Cybernetics and Informatics (IJCI), Vol. 5, pp. 387-394, April 2016.
[12] E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", Commun. ACM, Vol. 13, No. 6, pp. 377-387, June 1970.
[13] H. Bodepudi, "Faster the Slow Running RDBMS Queries with Spark Framework", International Journal of Scientific and Research Publications, Vol. 10, No. 11, pp. 287-291, 2020.
[14] A. O'Driscoll, R. Sleator, "Synthetic DNA: The Next Generation of Big Data Storage", Bioengineered, Vol. 4, March 2013.
[15] M. Sais, N. Rafalia, J. Abouchabaka, "Intelligent Approaches to Optimizing Big Data Storage and Management: REHDFS System and DNA Storage", Procedia Computer Science, Vol. 201, pp. 746-751, January 2022.
[16] M. Sais, N. Rafalia, J. Abouchabaka, "Enhancements and an Intelligent Approach to Optimize Big Data Storage and Management: Random Enhanced HDFS (REHDFS) and DNA Storage", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 50, Vol. 14, No. 1, pp. 196-203, March 2022.
[17] S. Begum, "A Deeper Examination of NoSQL Database Models and Characteristics", International Journal of Research Publication and Reviews, Vol. 3, No. 10, October 2022.
[18] M.A.G. Mhmoud, O.S. Fatoh, "Study on SQL vs. NoSQL vs. NewSQL", Journal of Multidisciplinary Engineering Science Studies (JMESS), Vol. 3, pp. 1823 1821, June 2017
[19] L. George, "HBase: The Definitive Guide: Random Access to Your Planet-Size Data", Beijing Koln: O'Reilly Media, 1st Edition, p. 556, October 2011.
[20] K. Shvachko, H. Kuang, S. Radia, R. Chansler, "The Hadoop Distributed File System", The 26th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1-10, May 2010.
[21] F. Chang, et al., "Bigtable: A Distributed Storage System for Structured Data", ACM Trans. Comput. Syst., Vol. 26, No. 2, pp. 4:1-4:26, June 2008.
[22] M.M. Al Momani, "Cluster Mechanism for Hot Data Storage in HBase System", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 14, No. 3, June 2019.
[23] X. Huang, L. Wang, J. Yan, Z. Deng, S. Wang, Y. Ma, "Towards Building a Distributed Data Management Architecture to Integrate Multi-Sources Remote Sensing Big Data", The 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, United Kingdom, pp. 83-90, June 2018.
[24] J.K. Chen, W.Z. Lee, "An Introduction of NoSQL Databases Based on Their Categories and Application Industries", Algorithms, Vol. 12, No. 5, May 2019.
[25] V. Guimaraes, et al., "A Study of Genomic Data Provenance in NoSQL Document-Oriented Database

Systems", IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1525-1531, November 2015.
[26] R. Byali, "MogoDB: A NoSQL Database with Amazing Advantages and Features", International Journal of Research Publication and Reviews, Vol. 3, No. 10, pp. 1468-1471, October 2022.
[27] M.U. Hassan, I. Yaqoob, S. Zulfiqar, I. Hameed, "A Comprehensive Study of HBase Storage Architecture-A Systematic Literature Review", Symmetry, Vol. 13, p. 109, January 2021.
[28] L. Yan, Z. Zhang, D. Yang, "Temporal RDF(S) Data Storage and Query with HBase", CIT. Journal of Computing and Information Technology, Vol. 27, No. 4, June 2020.
[29] R. Rai, P. Chettri, "Chapter Six - NoSQL Hands On", Advances in Computers, P. Raj, G.C. Deka, (Eds.) Elsevier, Vol. 109, pp. 157-277, 2018.
[30] S. Wang, G. Li, X. Yao, Y. Zeng, L. Pang, L. Zhang, "A Distributed Storage and Access Approach for Massive Remote Sensing Data in MongoDB", ISPRS International Journal of Geo-Information, Vol. 8, p. 533, November 2019.
[31] B. Jabir, N. Falih, "Big Data Analytics Opportunities and Challenges for the Smart Enterprise", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 47, Vol. 13, No. 2, pp. 20-26, June 2021.

BIOGRAPHIES

Name: **Manar**
Surname: **Sais**
Birthday: 13.06.1996
Birth Place: Fes, Morocco
Bachelor: Mathematical and Computer Sciences, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, 2017
Master: Big Data and Cloud Computing, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, 2019
Doctorate: Student, Department of Computer Science, Computer Research Laboratory (LaRI), Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, Since 2020
Research Interests: Big Data, Cloud Computing, Artificial Intelligence
Scientific Publications: 3 Papers

Name: **Ayoub**
Surname: **Sghir**
Birthday: 03.08.1996
Birth Place: Sale, Morocco
Bachelor: Mathematical and Computer Sciences, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, 2018
Master: Big data and Cloud Computing, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, 2020
Doctorate: Student, Department of Computer Science, Computer Research Laboratory (LaRI), Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, Since 2020
Research Interests: Big Data, Cloud Computing, Data Science
Scientific Publications: 1 Paper

Name: **Najat**
Surname: **Rafalia**
Birthday: 04.11.1968
Birth Place: Kenitra, Morocco
Bachelor: Applied mathematics, Department of Mathematics, Faculty of Sciences, Mohammed V University, Rabat, Morocco,1992
Master: Postgraduate Diploma (DEA) in Computer science, Department of Computer Science, Faculty of Sciences, Mohammed V University, Rabat, Morocco, 1994
Doctorate: Doctorate in Computer Sciences, Departement of Computer Science, Ibn Tofail University, Kenitra, Morocco, 2017
The Last Scientific Position: Prof., Department of Computer Sciences, Ibn Tofail University, Kenitra, Morocco, Since 2019
Research Interests; Distributed Systems, Multi-Agent Systems, Concurrent and Parallel Programming, Big Data
Scientific Publications: 14 Papers, 2 Theses

Name: **Jaafar**
Surname: **Abouchabaka**
Birthday: 27.02.1968
Birth Place: Guersif, Morocco
Bachelor: Fundamental Mathematics, Department of Mathematics, Faculty of Sciences, Kadi Ayad University, Marrakech, Morocco, 1992
Master: Postgraduate Diploma (DEA) in Mathematics, Department of Mathematics, Faculty of Sciences, Mohammed V University, Rabat, Morocco, 1994
Doctorate: Computer Sciences Applied to Mathematics, Department of Computer Science, Mohammed V University, Rabat, Morocco, 2001
The Last Scientific Position: Prof., Department of computer Sciences, Ibn Tofail University, Kenitra, Morocco, Since 2005
Research Interests: Concurrent and Parallel Programming, Distributed Systems, Artificial Intelligence, Big Data
Scientific Publications: 55 papers, 2 Theses