Synergy between Relational and NOSQL Databases for Big Data Management and Shaping the Future of Database Education

Shairy
Assistant Professor in Computer Science
PCTE Institute of Engineering & Technology

Gourav
Assistant Professor in Computer Science
PCTE Institute of Engineering & Technology

Navpreet Kaur Verma
Assistant Professor in Computer Science
Guru Nanak Institute of Management & Technology

ABSTRACT
With the growing popularity of the internet, the amount of information available is also growing. Big data processing is required for Web Service 4.0 because of the necessity for extremely large-scale, highly concurrent social networking sites that are purely dynamic. The rigorous schema and robust query capabilities of traditional relational databases (RDBMS) have long dominated structured data storage. To meet the requirement for high speed, the current relational database must scale horizontally. As a result, designers must take these factors into account when creating a new class of databases known as NoSQL. This study compares the two database systems, looking at their performance, applicability in modern applications, and architectural tenets. To find out whether final-year students preferred relational or NoSQL databases for project work, a survey was administered to them. According to the findings, NoSQL databases are becoming more and more popular, especially in fields that need to handle unstructured data effectively, flexibly, and scalable. According to one analysis, 70% of students choose NoSQL systems, with MongoDB standing out as a top pick because of its document-oriented design and performance. Relational databases are still useful for applications that require transactional integrity and structured data. In order to educate students, the findings highlight the necessity for academic programs to balance the teaching of both database paradigms. Insights on database selection criteria and the possible future incorporation of industry input to improve instructional tactics are provided in the study's conclusion.

Keywords
RDBMS, ACID, NoSQL, MongoDB, CAP Theorem, Big Data

1. INTRODUCTION
Systems for managing relational databases and non-relational databases differ greatly from one another. Unlike relational database management systems, NoSQL does not store data in the conventional tabular format and retrieve it using SQL queries. However, non-relational databases have become incredibly popular in the past few years. Since simplicity fosters speed, the majority of them rely on storing basic key-value pairs. NoSQL doesn't adhere to any schema; it is known as a schema-less database management system. Carlo Strozzi used the term "NoSQL"[12] in 1998 to refer to the file-based database system he was creating. Single-node systems are unable to handle the massive amounts of data created by the latest developments in cloud computing and distributed web applications. As the Internet and cloud[14] computing continues to grow, new kinds of applications have appeared, increasing the need for database technology [1]. This type of data is often called "Big Data." Distributed databases, flexible schemas, and speed are necessary for processing such large amounts of data. Numerous open-source and commercial NoSQL database implementations exist, including BigTable [2] and HBase.

1.1 Relational Database
Every record was initially kept by hand, but as technology advanced, significant changes occurred throughout time. Databases [3] were developed to make data maintenance simpler. A database can be as simple as a text document or as complicated as a database. For these databases to function well, they must be frequently cleaned up to get rid of any redundant, inconsistent, or unclean data. The relational model is the most popular and widely used idea for storing this data. From the database pool, Structured Query Language (SQL) retrieves pertinent information due to their ease of use; relational databases are the most used kind of database. Structured Query Language, or SQL, was developed as a common high-level interface for the majority of databases and is typically used as DDL and DML for relational database management system (RDBMS) administration. Relational model-based databases, such as MySQL, MS-SQL Server, Oracle, and others, all support SQL as a query language. Data in an RDBMS is organized into multiple tables that can be accessed based on needs without requiring table modifications. Relational databases make it simple to carry out operations like join, aggregation, addition, creation, retrieval, and deletion. They also make it very simple to expand or alter already-existing tables. Relational databases rely on the ACID properties. These four characteristics work together to guarantee data integrity by ensuring that each transaction is completed consistently, preventing data corruption even in the event of system failures, and preventing concurrent transactions from interfering with one. .

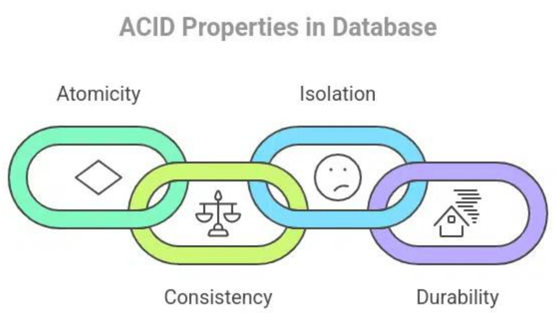


Figure 1: ACID Properties

1.2 NoSQL Database

NoSQL, an acronym for Not Only SQL, is a database management system that does not rely on relational databases. MongoDB, Cassandra, CouchDB, HBase, and other top NoSQL[10] databases are among them. NoSQL databases were developed and greatly enhanced by major internet corporations such as Amazon, Google, and LinkedIn in response to the current craze for "Big Data[13]." The primary distinction between non-relational and traditional data models is that the former are made to process large amounts of data quickly while requiring comparatively little consistency. In exchange for the performance improvement, it relaxes the ACID requirements that many relational database systems give but NoSQL[11] Database classification done according to CAP theorem.

As more people utilize the internet, more traffic is being sent to servers, such as Google, Facebook, and others. On Facebook, friends share 2.4 billion pieces of material every day. Buying a large server is the initial step in scaling, but there are two problems with this strategy: the cost and the breaking point beyond which it can be extended. Attempting to cluster information is the second method when it turns out to be extensive. Clustering is challenging because consistency is a problem if one cluster is updated while the other is not for information retrieval. Consequently, the CAP theorem is followed by NoSQL databases. Eric Brewer first introduced the CAP theorem in 2000. CAP stands for:

- **Consistency:** This indicates that when we attempt to access or get data, the same data ought to be shown. Alternatively, if a value is stored or modified to one node, another node ought to be updated automatically.
- **Availability:** This indicates that even in the event of a system failure, values can still be obtained from the system, though they might not be consistent.
- **Partition Tolerance:** This indicates that data should be able to write to and retrieve from many locations without any effects from partitioning. Relational databases don't have this feature.

The following is a rough taxonomy of NoSQL databases based on the CAP theorem and several NoSQL database concerns:

- **Partition tolerance (CP) and consistency**
  Although such a database system guarantees the consistency of the data stored in the dispersed nodes, its availability support is inadequate. MongoDB and Hbase are the primary examples of CP systems.
- **Availability and partition tolerance (AP)**
  Consistency is the main way that these systems guarantee availability and partition tolerance. Examples of AP systems are CouchDB and DynamoDB.
- **Consistency and availability (CA)**
  The replication strategy is primarily used to assure data consistency and availability, with a portion of the database not being concerned with partition tolerance [9]. The CA's systems are relational database systems.

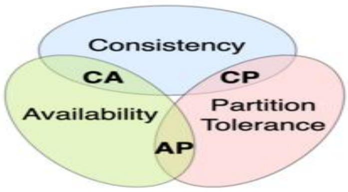


Figure 2: CAP Theorem with Example

2. REVIEW OF LITERATURE

Matallah et al.(2017)[4] shows the contrasted two NoSQL databases: MongoDB, a document database, and HBase, a NoSQL column database. With two additional workloads, G and H that concentrate on updates the two databases were compared and assessed using YCSB. Because they include insert procedures, workloads D and E were eliminated from the trials. This work's primary goal was to evaluate read, write, and load operations. 600,000 records were used to evaluate the databases in an Ubuntu environment. It was determined that HBase outperformed MongoDB in write operations and MongoDB was more effective in read operations.

Mishra et al. (2018) [5] stated that four NoSQL document databases—MongoDB, ArangoDB, OrientDB, and Elasticsearch—were assessed for performance. With the exception of workload E, which contains operations like scan, the authors employed the normal YCSB workloads. The database with the best performance, according to the tests conducted in an Ubuntu system, was MongoDB, followed by ArangoDB, OrientDB, and Elasticsearch.

Three document-type NoSQL databases—MongoDB, Couchbase, and RethinkDB—were tested for performance by the authors in [6]. These tests consist of operations like post, patch, get, and delete. In this assessment, both one and several threads were utilized. These authors found that Couchbase performed better in the majority of tests, whether using one or many threads. The one exception was the post-operation, where MongoDB produced better outcomes.

Johnson et al. (2019)[7] highlighted that the synergy between relational and NoSQL databases in the context of large data management. They talked about relational databases' inability to scale horizontally to satisfy the needs of contemporary online applications, which frequently call for processing large amounts of data in real time. Their research compared document-based systems like MongoDB and key-value stores like Redis with conventional relational models in terms of partition tolerance, availability, and data consistency as determined by the CAP theorem. By highlighting the complimentary roles that both database types can play in maximizing performance for both structured and unstructured data, the authors also investigated how to integrate both database types inside large-scale systems.

In the context of big data management, Smith et al. (2020)[8] looked at the increasing demand for scalable, effective, and adaptable data storage systems. They highlighted the shortcomings of conventional relational databases, especially when it comes to managing the unstructured and semi-structured data found in contemporary applications like social media sites and Internet of Things (IoT) systems. As alternatives to relational databases, the writers examined the rise of NoSQL databases, particularly key-value and document-based techniques. They emphasized how NoSQL databases, such as Redis and MongoDB, can handle enormous amounts of data quickly and flexibly. They suggested a hybrid approach that capitalizes on the advantages of both relational and NoSQL systems after examining their synergy.

Carvalho et al. (2022) [9] present three NoSQL document databases—Couchbase, CouchDB, and MongoDB. The databases' architecture, key benefits, and drawbacks were taken into account when characterizing them. Additionally, the authors compared the databases solely using the OSSPal technique and did not conduct an experimental review. Seven categories—functionality, operational software characteristics, software technology attributes, documentation, support and service, community and adoption, and development process—are established by the OSSpal methodology, which evaluates open-source software using both quantitative and qualitative metrics. The greatest NoSQL document database was ultimately determined by a score. MongoDB had the highest score in this evaluation, followed by Couchbase and CouchDB.

3. METHODOLOGY OF WORK

3.1 Objective

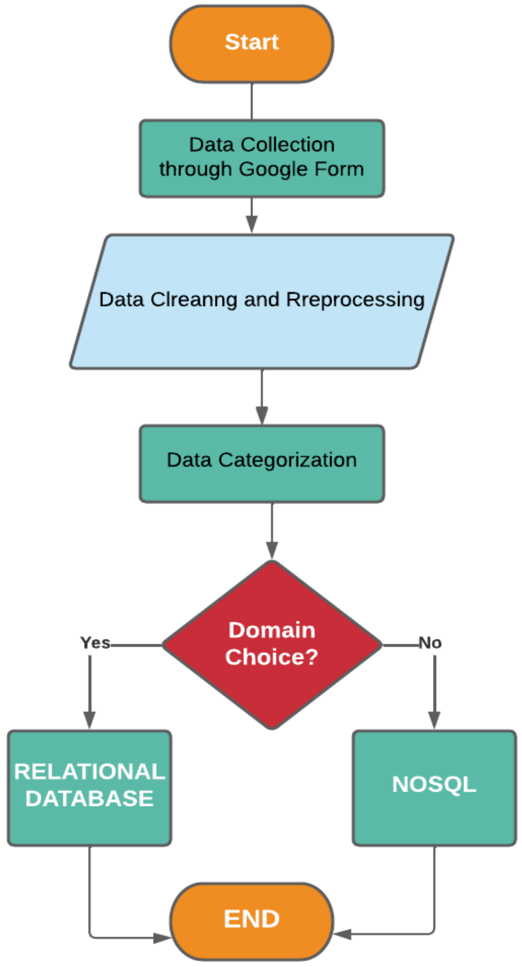
The main goal of this study is to compare Relational Database and NoSQL databases based on the preferences of final-year students pursuing Master of Computer Applications (MCA), Bachelor of Computer Applications (BCA), and Bachelor of Technology in Computer Science Engineering (B.Tech CSE) programs, The survey specifically seeks to ascertain if students prefer SQL or NoSQL databases for their senior project work, offering insights into current database usage patterns and preferences.

3.2 Data Collection Process

A Google Form, an online survey tool, was used to gather data for this investigation. The procedures listed below were used to gather data:

- **Survey Design**
  Personal data like name, class, section, roll number, and phone number were gathered via the Google Form. The survey's main focus was a single question that asked students to select between SQL and NoSQL databases for project work.
- **Distribution of the Survey**
  To ensure accessibility for all final-year students in the target programs (MCA, BCA, and B.Tech CSE), the Google Form link was distributed to students via email and social media channels like course-related groups.
- **Data Analysis Process**
  - **Statistical Tools**: Microsoft Excel was used to analyse the data. To find out if there was a significant variation in the database choices of the various student groups.
  - **Data Visualisation**: To clearly illustrate students' preferences for SQL and NoSQL databases, the results were displayed using pie diagrams and bar charts.

3.3 Flow Chart



Algorithm Steps:

1. First Step is to collect the data using a Google Form.
2. After collecting data we perform the step of Data Cleaning and Pre-processing to guarantee consistency, accuracy, and quality of data.
3. As needed, organize the data into pertinent categories or formats like BCA,MCA and B.Tech(CSE)
4. Then Database Selection according to the requirements of the domain of their interest in which they want to create final year projects.
5. After these choose a relational database should be used if the domain calls for an organized approach and select a NoSQL database if the domain demands scalability and flexibility.
6. At the end analyse the data according to student choice.

4. RESULT & DISCUSSION

Information was gathered from final year students regarding their selection rationale for the database domain for their capstone projects. The responses were grouped according to the academic program, BCA, MCA and B.Tech (CSE) so that finer pattern analysis could be performed. Relational Database Management Systems (RDBMS) and NoSQL preferences were indicated by students based on the requirements of their projects. As with most studies, this one attempted to shed some light on the current practices in the industry and the usefulness of each type of database in relation to the project's objectives and focus. To deepen the study, future work might examine the effect of project complexity on the choice of the database to better understand the reasons behind the database selection.

Both relational and NoSQL databases have distinct advantages, according to the results. Relational databases are best suited for structured data and applications that need a highly structured schema, but the analysis shows that NoSQL databases are more competitive in the current market environment. In 2025, NoSQL databases will be essential for industries due to the growing need for scalability, flexibility, and the ability to handle large amounts of unstructured data. The data's graphical representation shows the growing preference for NoSQL databases because of their capacity to meet contemporary business requirements, which makes them a more flexible option for project implementation in real-world scenarios.

4.1 Domain Analysis of B.Tech (CSE) Students

Table 1: B.TECH STUDENTS

Domain	Number of Students Selected Domain
NoSQL Database	99
Relational Database	65
Grand Total	**164**

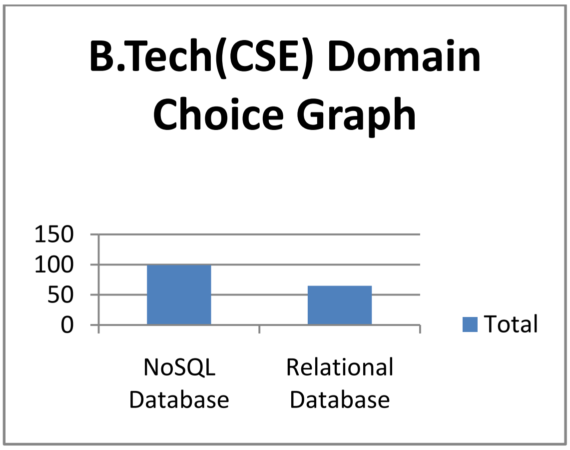


Figure 3: DOMAIN ANALYSIS OF B.TECH STUDENTS

The final year **B.Tech (CSE)** students' database domain preferences are captured in Table 1 and Figure 3. Out of 164 participants, a staggering 99 students opted for NoSQL databases while 65 students went for Relational databases. This clearly shows BTech (CSE) students prefer NoSQL databases over Relational ones. The reason for the preference could be the greater project computer science associated with managing enormous data sets and building scalable web applications, which often enable NoSQL databases to perform more effectively and are easier to use compared to their Relational counterparts. Still, the large number of students opting for Relational databases demonstrates that these systems have not lost their importance, probably for projects that demand high consistency and control over the structured data.

4.2 Domain Analysis of BCA Students

Table 2: BCA Students

Domain	Number of Students Selected Domain
NoSQL Database	112
Relational Database	26
Grand Total	**138**

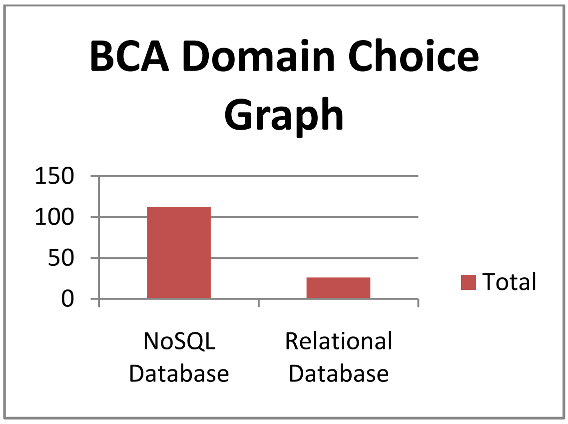


Figure 4: Domain Analysis of BCA Students

The data from BCA's last-year students' database domain interests is represented in Table 2 and Figure 4. As indicated in this class, there is a remarkable dominance of NoSQL preferences as 112 students opted for NoSQL databases while only 26 students selected Relational databases out of 138 responses. This strong bias towards NoSQL by BCA students might stem from the curriculum focus on web development and multimedia projects that make noSQL very advantageous due to its flexibility and scalability. The less students going for Relational databases suggest that BCA projects are more about data dynamic approaches than rigid structures.

4.3 Domain Analysis of MCA Students

Table 3: Mca Students

Domain	Number of Students Selected Domain
NoSQL Database	11
Relational Database	4
Grand Total	**15**

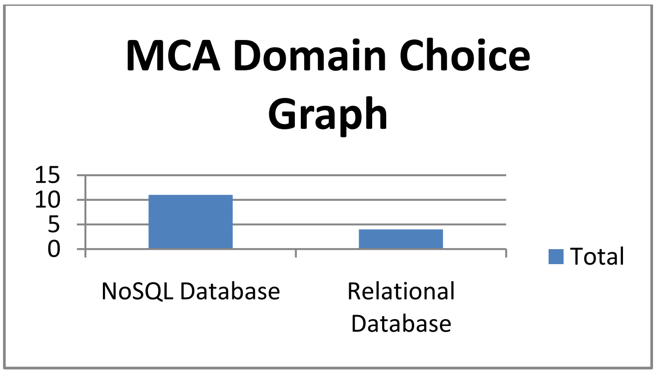


Figure 5: Domain Analysis of MCA Students

The selection of databases by MCA final year students is illustrated in Table 3 and Figure 5. Of the 15 students in this group, 11 opted for NoSQL and 4 opted for Relational databases. While this sample is the smallest of all students, it does show the trend toward NoSQL – as with the other groups. It can be assumed that the more complex application development and data analysis which is part of the MCA curriculum is what encourages students to favor NoSQL databases because of their ability to scale and handle increasingly diverse datasets. These findings must, however, be treated with caution due to the limited number of participants.

4.4 Domain Analysis of Combined Students

Table 4: Combined Final Year Students

Domain	Number of Students Selected Domain
NoSQL Database	222
Relational Database	95
Grand Total	**317**

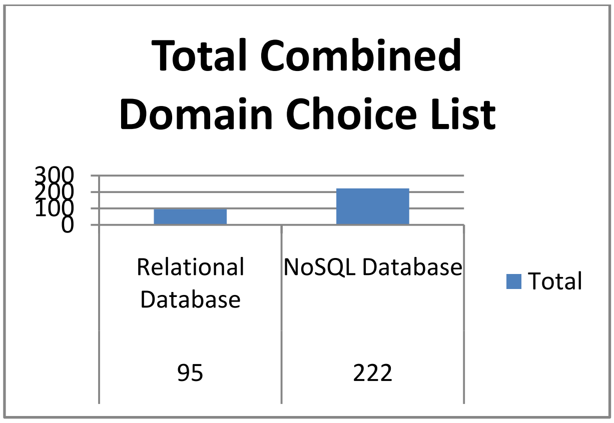


Figure 6: Domain Analysis of Combined Final Year Students

Table 4 and Figure 6 show the combined database domain preferences for all final-year students covered under the survey (B.Tech CSE, BCA, and MCA). Throughout all programs, NoSQL databases were preferred by 222 students, while 95 students preferred Relational databases, based on 317 replies. This overall finding highlights the overall trend witnessed under each individual program: an overwhelming preference towards NoSQL databases. This preference is probably a testament to the growing popularity of NoSQL in dealing with the needs of contemporary applications, especially those that involve large amounts of unstructured or semi-structured data. But it's essential to note that Relational databases continue to maintain a significant footprint, suggesting that they remain relevant for certain types of applications.

5. CONCLUSION

This study demonstrates how final-year students' preferences for database systems for their projects have changed over time. According to the analysis, there is a growing preference for NoSQL databases because of their scalability, flexibility, and capacity to manage unstructured data. These features are crucial in managing the constantly rising amounts of Big Data produced by contemporary applications like social media, e-commerce platforms, and the Internet of Things. For students who prioritise performance and flexibility, NoSQL databases have proven beneficial due to its schema-less architecture and adherence to the CAP theorem. The capacity to manage scalable, real-time data processing was mentioned by students as a key consideration when selecting NoSQL systems, especially for projects that call for fast read and write operations.On the other hand, relational databases are still important for applications that need transactional consistency and organised data storage. They are a crucial option for fields like financial systems, where consistency is crucial, because of their support for ACID characteristics, which guarantees data integrity and dependability. According to the poll results, 70% of students in all programs chose NoSQL databases, with the remaining 30% favouring traditional databases. These results highlight the necessity for academic programs to teach both new and conventional RDBMS and NoSQL technologies in a balanced way so that students have the skills necessary to handle a variety of application scenarios.

In a nutshell the emergence of NoSQL databases is a reflection of the changing needs of contemporary applications and business procedures, even while relational databases continue to be a fundamental component of structured data management. This development highlights how crucial it is to give students flexible database administration abilities so they may succeed in settings that rely heavily on data. This study could be extended in the future by examining the precise variables affecting database selection and incorporating industry input to improve instructional tactics.

6. REFERENCE

[1] Jing Han, Meina Song, and Junde Song. "A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing", In ICIS 2011, 10th IEEE/ACIS International Conference on Computer and Information Science, 2011.

[2] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, “Bigtable: a distributed storage system for structured data,” in Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7, ser. OSDI ’06. Berkeley, CA, USA: USENIX Association, 2006, pp. 15–15

[3] Donald R. Deutsch, and Eric K. Clemons, “DBMS Standards: Current Status and Future Directions,” Proceedings of the Sixth International Conference on Very Large Data Bases, VLDB Endowment, vol. 6, pp. 431–433, 1980

[4] Matallah, H.; Belalem, G.; Bouamrane, K. Experimental comparative study of NoSQL databases: HBASE versus MongoDB by YCSB. Int. J. Comput. Syst. Sci. Eng. 2017, 32, 307–317

[5] Mishra, O.; Lodhi, P.; Mehta, S. Document Oriented NoSQL Databases: An Empirical Study. In Data Science and Analytics; Springer Nature: Singapore, 2018; pp. 126–136

[6] Augusto, D.; Morais, W.; Freitas, E. NoSQL real-time database performance comparison. Int. J. Parallel Emergent Distrib. Syst. 2018, 33, 144–156.

[7] Johnson, P., Wang, Y., & Garcia, S. (2019). Exploring the integration of NoSQL and relational databases for efficient big data processing. International Journal of Data Management, 22(7)

[8] Smith, J., Brown, M., & Lee, A. (2020). The synergy between relational and NoSQL databases in big data

management: A hybrid approach. Journal of Big Data Technologies, 15(3), 234-249.

[9] Carvalho, I.; Sá, F.; Bernardino, J. NoSQL Document Databases Assessment: Couchbase, CouchDB, and MongoDB. In Proceedings of the 11th International Conference on Data Science, Technology and Applications, Lisbon, Portugal, 11–13 July 2022; pp. 557–564.

[10] Eeshan Gupta et al., "Attribute-Based Access Control for NoSQL Databases," Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, Association for Computing Machinery, New York, NY, USA, pp. 317–319, 2021.

[11] Jing Han, Haihong E, Guan Le,Jian Du, "Survey on NoSQL Database," IEEE, pp. 363- 366, 2011.

[12] Nayak, A.; Poriya, A.; Poojary, D. Type of NOSQL Databases and its Comparison with Relational Databases. Int. J. Appl. Inf. Syst. 2013, 5, 16–19

[13] SHEN D R,YU G,WANG X T,et al. Survey on No SQL for Management of Big Data[J]. Journal of Software, 2013, 24(8):1786- 1803.

[14] R Hecht, S Jablonski, "NoSQL Evaluation," International Conference on Cloud and Service Computing, IEEE, pp. 336-338, 2011.

[15] Neal Leavitt " Will NoSQL Databases Live Up to Their Promise?" IEEE Computer Society 0018-9162/10 © 2010 IEEE

[16] Bansel, H. González-Vélez and A. E. Chis, "Cloud Based NoSQL Data Migration," 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Heraklion, 2016, pp. 224-231.

[17] Matallah, H.; Belalem, G.; Bouamrane, K. Experimental comparative study of NoSQL databases: HBASE versus MongoDB by YCSB. Int. J. Comput. Syst. Sci. Eng. 2017, 32, 307–317.

[18] Mishra, O.; Lodhi, P.; Mehta, S. Document Oriented NoSQL Databases: An Empirical Study. In Data Science and Analytics; Springer Nature: Singapore, 2018; pp. 126–136