Moving RDBMS to NoSQL Paradigms

Hiwa Ali Faraj
Information Technology
University of Sulaimani
Sulaymaniyah , Iraq
hiwa.faraj@univsul.edu.iq

Abstract—**Current business requirements and application infrastructure paved the way toward not only SQL (NoSQL) paradigms. The business requirements impose on information technology that brings a new atmosphere for the apps that require handling massive amounts of heterogeneous data types in the limited time and provide Highly Available Transaction (HAT) requirements. Each of Relational Database Management System (RDBMS) and NoSQL models support a group of features that bring certain functionalities to utilize business requirements. The key features to determine a specific data model is technology capabilities to achieve business requirements. Due to the shortages of Atomicity, Consistency, Isolation, and Durability (ACID) transaction, relational databases are unable to manage the atmosphere's requirements, then, Brewer's theorem, data consistency, data availability and partition-tolerance (CAP) emerge an alternative to the modern business requirements. Traditional relational databases still support the applications that require to enforce relational transactions protocol, restrictions, and constraints. In parallel with both models, polyglot persistence has come out, coexisting with both models in the single web server or an enterprise server.**

This paper concentrates on the roles of various factors of moving toward NoSQL, such as handling big data, heterogeneous data types, Internet of Things (IoT), highly available transactions requirements, eschew ACID constraints, high performance, software-as-a-service business, dynamic modeling schema, and data scaling. Moreover, an overview of nature and modeling of both relational and non-relational databases, limitation of relational database, and reasoning to move to NoSQL is presented.

Keywords— **RDBMS, NoSQL, ACID, big data, CAP, HAT**.

I. INTRODUCTION

The Web 2.0 and 3.0 businesses requirements are imposing the companies to quickly move or migrate toward NoSQL. Historically file organizations of data were replaced by relational database management systems, and major vendors started to blend XML into DBMS technology. The critical moment in DBMS development advancement happened in 1970 when Edgar Frank Codd published his famous paper on the relational model of data [1]. Even since the relational database predominant model over 40 years to date, relational database come up with solutions to the main drawbacks to the file organization of data such as, redundant data, which leads to extra storage and cause inconsistent data, isolated data, meaning it is hard to query data across various separate file formats, and data dependency leads to hard working during modifying data structure [1]. When the relational database makes any successful transactions, the RDBMS has to ensure that the transaction will obey a bunch of rules. The rules are known as database characteristics such as, atomicity, consistency, isolation and durability. Then relational databases are required to enforce strict protocol in order to perform successful transactions, examples of RDBMSs are Oracle, DB2, MysqlServer, MySql, and PostgreSQL.

Relational databases have shortages such as scalability, complexity and ACID restrictions also managing relational tables over various servers is horrible. Due to the RDBMS schema restriction, any structure modification of the table's requires to redesign the schema. Then ACID hinders quick data manipulations [2]. Alternatively NoSQL provides horizontal scaling then establishes a new cluster [3]. While RDBMS have been dominant for decades to store data and transactions, currently NoSQL models have significant role and gained popularity.

As a consequence of modern world wide web (www), the server architecture frameworks require reorganizing data modeling, it requires a structure that provides availability, scalability and performance. Big data and current web applications, especially web 2.0 and web 3.0, require NoSQL model databases.

Companies are selecting a convenient model in order to fit with all the available database features [4]. Non-relational databases bring the difference schema to storing and retrieving model data [5]. It provides a schema-less model to store data and transactions, regardless of enforcing relational schema restriction [3, 5]. Hence handling of big data makes the NoSQL the most popular to use database.

The fundamental threshold between relational database and non-relational database are dimensions of big data which is known as Five Vs. The five Vs. comes from velocity, volume, value, variety and veracity. According to the paper [6], Five Vs. includes: 1) velocity refers to the quickly moving generated data, 2) volume refers to the creation of strong magnitude of data, 3) value means covenant data collection of the business, 4) variety refers to the heterogeneous data structure types: such as, structures, semi-structured and unstructured data, and 5) veracity means confirming reliable data association. The characteristics of big data dimensions force the companies to change database schema and data handling in order to fit with the current major changes of the data structure, volume and availability.

979-8-3503-2028-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00209

The discussion of many research papers includes clarifying features and categories of both relational databases as well as NoSQL, and comparing the advantages as well as shortages of both, while few studies has examined certain reasons to move RDBMS to NoSQL, such as big data, heterogeneous data types and performance, however, there are certain factors that have not been declared. This study aims to examine the different factors that would affect the database model to move from RDBMS to NoSQL Paradigms. The structure of this paper is organized as follows: Section 2 presents background literature for this study. Section 3 discusses the controversy between relational and non-relational databases. Section 4 explains the reasoning of the transformation toward NoSQL. Finally, section 5 provides a conclusion.

II. LITERATURE REVIEW

Data modeling for handling, transacting and analyzing data is curtailing research areas for decades. Chronologically data file handling was remodeled by RDBMS. Relational databases based on ACID transaction protocol still provide a solution for a lot of applications such as payroll, inventory control systems, point of sale systems and airline reservations. In addition, the nature of these apps requires enforcing ACID restrictions. Relational databases were the dominant solution for storing and data transaction of data in the era of pre-big data expulsion, hence, the traditional DBMS decline to handle big data [3, 7, 8]. Eventually, RDBMS faces various shortages for implementing big data analysis requirements [3]. Recent database management trends are moving toward the NoSQL paradigm. There are reasons behind switching data modeling. With respect to relational databases and NoSQL, many studies are introduced that mostly covered migration, differences and comparison between them, however, it was not mentioned the reasons to move RDBMS to NoSQL. While some studies focus on large scaling of data and performance.

The high performance and data storage capacity are the most common reasons behind the increased use of NoSQL database. In addition, day after day and with data growing, NoSQL has attracted in this area of research, since it performs better performance [4, 9]. RDBMS faces a lot of challenges to support huge data such as: unscalable, slow, cost, and availability[4]. Hence many enterprises have migrated to utilize NoSQL databases that respond to high speed data access[9, 10]. Leverage of NoSQL will ensure higher performance than RDBMS by applying NoSQL to database systems [11]. Currently, in the context of data storage, NoSQL databases have become alternatives to RDBMS [2].

In these studies [12, 13], different obstacles have been discussed that influence database administrators and software engineers from pre-migration to post-migration. In addition, the current procedures and different pros and cons of NoSQL migration have been discussed [13].

The use of the data is rapidly changing the nature of communication, shopping, advertising, entertainment, and relationship management. Applications that do not find ways to leverage it quickly will quickly fall behind [14]. Data generating trends to heterogeneous data types such as: structured, unstructured, and semi-structured, data comes from various resources, tweets, audio, Geolocation, text, machine logging data, and etc. Nowadays, most of the data is unstructured or semi-structured, however RDBMS schema-based approach makes it impractical to handle rapidly unstructured and semi-structured data [14]. In the context of big data performance analysis, NoSQL is better than relational databases [3, 12]. The study in [15] has discussed the most important features of NoSQL. Furthermore, it is a better solution for handling data availability and fast access.

Many studies have been conducted for the comparison of the relational and non-relational databases. On the other hand, authors in [16], considered NoSQL that the best option for the current system, which is more complex and continually updating the requirements, however if NoSQL handles features such as ACID it will be very expensive. Moreover, this study found that NoSQL is manageable and scalable, in addition it is a dynamic data model that performs unstructured big data managements very well [9].

There are many works studying the next data model. Previous studies in [2, 5] have found that NoSQL is the ultimate candidate to replace RDBMS. However, it is not to be fully alternative to the relational database. Moreover, the study in [17] showed that NoSQL provides worthy solution to the era of big data. However, it is not led to demise of relational database. On the other hand, modern enterprise solutions provide a flexible architecture of coexistence with both NoSQL and relational data models. In addition, today is the era of polyglot persistence a method that utilize various data storage to manage across different data storages requirements [17].

Data managements of big companies have been changed dramatically, therefore, many companies have start to use the NoSQL model, since it provides better solutions for handling big data problems. As consequences, many studies states migrating reasons such as big data, performance, scaling out, dynamic data modeling however this study will examine nine factors that affect database model of moving toward NoSQL.

III. THE COMPETITION BETWEEN RELATIONAL AND NON-RELATIONAL DATABASE

Relational database is based on transactional characteristics. The transactions properties are: Atomicity, Consistency, Isolation, and Durability [14, 18]. RDBMS are based on transaction characteristics pledge data integrity and consistency [14]. In order to achieve consistency, security and availability, RDBMS enforce to have schema and certain concentrates rules [4]. The synonym ACID refers to four properties which are [2, 5, 19]: 1) Atomicity: every transaction is treated as a unique task or work, it will happen or not, there is no middle opportunity, 2) Consequently: this property will not be achieved if the system faces any failure during the process, 3) Consistency: the transaction should be confirmed by predefined set or rules such as constraint, triggers and cascades, 4) Isolation: ensures that no transaction overlaps another transaction, and 5) Durability: when a transaction is committed successfully it saves the transactions into a non-volatile storage medium, even if the system fails. Relational database supports unique powerful data centers for data manipulation. Consequently, when data become massive, it not works properly to handle the segmentation for

parallel processing [2, 7, 18]. Hence, the relational database will not achieve data manipulation quickly [20]. Alternatively non-relational database support database operation over multi servers, however key relational database categories such JOINs, referential integrity and transactions cannot be attained [7, 9].

The CAP theorem supports non-relational databases and it is efficiently managing big data across distributed systems [1, 18]. The generic concept of CAP theorem was presented by Eric Brewer in 2000 [21]. He declared that, all the availability, consistency, partition tolerance in a distributed system cannot be achieved concurrently. Moreover, it explains that only two of the three of the properties consistent, available and partition-tolerant can be achieved at a time [1, 19, 22]. Similarly, the theorem states how the distributed server operated at the time when a database server failed to talk to another server [22]. The three factors of CAP theorem are:

1-Consistent, Available (CA) Systems have failure with partitions and typically deal with it with re-sync for instance, traditional RDBMS [14].

2-Consistent, Partition-Tolerant (CP) Systems have failure with availability while keeping data consistent over partitioned servers, such as BigTable and Redis [14].

3-Available, Partition-Tolerant (AP) Systems servers remain online even if they cannot communicate with each other and typically will replicate when the failure is solved, for instance: Cassandra, CouchDB. NoSQL databases can be classified according to various models, they store and retrieve data in various forms[14].

"Fig. 1" illustrates the three factors of CAP theorem.

The most common NoSQL models are:

A. Key-value stores

In this type of non-relational database data is stored in key-value pairs. Key-value save data as a collection of key-value pairs. The collection consists of groups of documents, each document uniquely identified by key-value. key-value has unique index; therefore, it has very fast access to the data, hence it has high performance and easy scaling, such as Berkeley DB, LevelDB, Dynamo, Memcached [5, 14, 15].

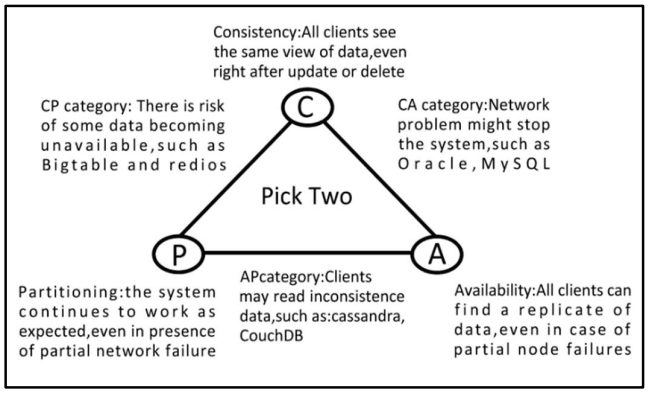

Figure 1: CAP theorem categories

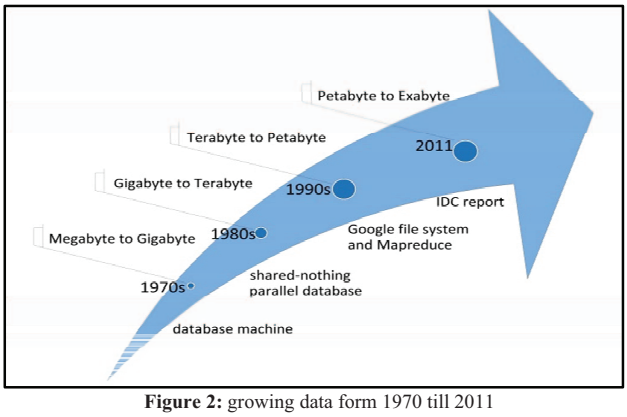

Figure 2: growing data form 1970 till 2011

B. Column oriented databases

Column oriented database stores data as a set of rows and columns. Each column of data works as an index of the database and it includes interrelated data [9, 14], examples are Google big table, HBase, Hypertable, Cassandra.

C. Graph databases

The concept of a graph database is based on the theory of graphs, It consists of three elements: node properties and lines. While node represent entities, and properties consist of the information about the entities, then edges represent the presence of a connection or relationship between two nodes[14, 15]. Graph databases include Neo4j, InfroGrid, IMS.

D. *Document stores*

Unique identification keys are used to recognize the documents in the database, the keys can be used for data manipulation. The document can store data in various formats, such as PDF, documents, XML and JSON file. In addition it supports the lists, pointers and nested documents[22].

By the above discussion, it is very important to know about The CAP theorem for designing any distributed system. For example, when transactional and ACID issues are coming in NoSQL database, there is no other option without CAP theorem [14]. RDBMS big limitations problem for supporting big data has been reviled by big companies.

IV. REASONS TO SWITCH TO A NOSQL DATABASE

Due to the big data applications, data modeling has been dramatically changed, however handling big data is the main problem applying RDBMS. in contrast NoSQL's ultimate solution to manage big data hence it supports massive data storage, dynamic schema, scale-out architecture, flexible data model and access requirements [15]. The architecture of the relational database based on operating in a single data center and their data modeling schema restricted structure based, as a consequence eventually companies such as Google, Amazon, Facebook, and LinkedIn reviled the model insufficient to support modern application requirements'[14]. In this section, the main reasons of moving to NoSQL has been discussed.

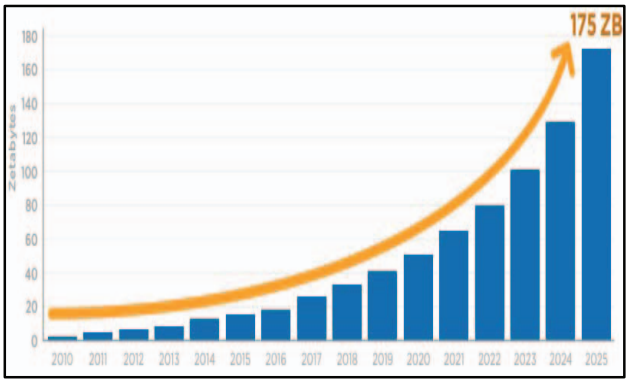

Figure 3: Annual size of the global datasphere – source: IDC Datasphere whitepaper.

A. *Big data*

Since the birth of relational databases until date data have been growing incredibly "Fig. 2", initially amount of data sized by gigabyte [23], furthermore the "Fig. 3" predicted the data to become 175 zettabytes by 2025 .

B. Heterogeneous data types; structured and unstructured data

Relational databases are highly structured and normalized as well as data stored in the tables that consist of columns and rows, each column identified by certain data types and constraints . In spite of rapidly growing data quickly, the applications trend towards unstructured data types[14]. Currently unstructured big data is the most abundant, see "Fig. 4" that illustrate how growing of unstructured data[24]. Unstructured data types such as; geolocation data, social graphs, user-generated content, email, video, audio, machine logging data, and sensor-generated data.

C. The internet of thing (IOT)

There is a rapid growth in wifi/internet-connected devices and perhaps such as sensors, household devices, factories, machines, automobile, gaming, e-commerce, geolocation, hospitals and education.

Consequently, the growth in Internet Protocol (IP) traffic is predicted to reach 4.8 zettabytes per annum [25]. Furthermore; according to Statistica the total Internet of Things in the world will be more than 75 billion connected devices [26], see "fig. 5".

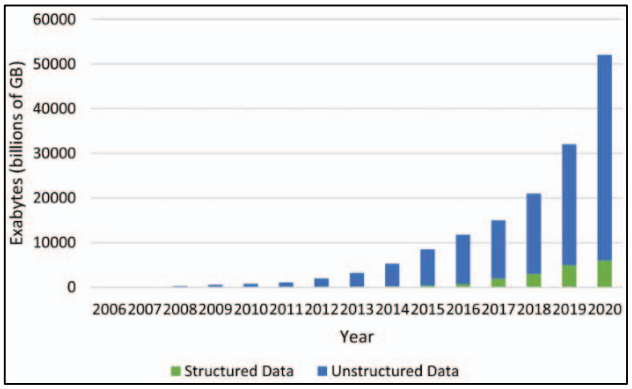

Figure 4: comparison between structured and unstructured data

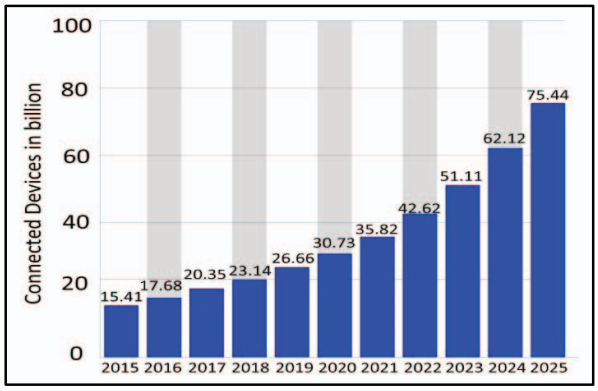

Figure 5: shows connected device in the world

D. *Highly available transactions requirements*

In order to guarantee business continuity, modern business trends require performing Highly Available Transactions (HAT). Which can be achieved by violating of ACID properties. The HAT systems reduce the delay of transaction and ensure the continuity of networks; therefore, system errors that faces recent distributed storage engine, enforce the companies to give up from ACID transaction functionalities[22].

E. Eschew acid constraints

Relational databases dominate the storage and managing technology of data. However, its lakes the ability to handle big data processes. Moreover, ACID requirements are insignificant to process huge transactions. Therefore, they are "web-scale applications, non-relational data stores, and global distribution of data centers required the creation of new alternatives" [27].

F. Performance

Due to the diffusion of the web 2.0, web. 3 applications such as social media and IOT applications, data has been growing very fast. Despite ACID working well to manage pre big data, however it is inadequate to handle big data transaction processes [4]. According to the[4, 18] the storage of data and handling performance are the big challenges for the RDBMS's models[28].

G. Cloud-based or software-as-a-service

Currently, almost all apps are hosted to the cloud venders; therefore, cloud computing requires horizontal scaling architecture which is easy for scaling services. Moreover, cloud based for hosting app reduce costs, improve mobility and security[14]. Additionally, cloud based apps provide the Database-as-a Service (DBaaS) schema to handle big data by deploying NoSQL databases, since NoSQL deployment provides high availability, fault tolerance and scalability to serve distinct client requests[21].

H. Dynamic modeling scheme

Relational databases provide static schema which consists of rows and columns. Each column has its data types and constraints. Developers found problems in modifying schemes during operations moreover any scheme modification requires the redesign of the model. In contrast NoSQL provides dynamic

schema which allows developers to modify the model during the database operations.

I. Data scaling

Relational databases support vertical scaling that requires supplementing more servers to the pool server beside adding CPU, RAM and power to the existing infrastructure. Vertical scaling is expensive, while NoSQL support horizontal scaling which includes adding more physical machines to the server. In recent years, NoSQL databases have developed as a response to the limitations of relational databases and to deliver the performance, scalability, and flexibility essential for the modern applications [2].

To sum up, moving toward NoSQL is based on the conditions that concerning with the business requirements and applications infrastructure. As much as the factors contribute with the business demand, the modeling data move toward the NoSQL, see "Fig. 6".

V. CONCLUSION

Relational databases model is based on ACID transaction properties. The model based on consistency, enforce restrict scheme and constraints, similarly it utilizes transaction protocols for data handling. The RDBMS model was dominant in the area of pre-big data. However, there are a lot of applications that are deploying relational model because of their business requirements implement restrictions and the rules applied for data handling. In contrast, relational databases have poor scalability, distributed, weak performance, high availability, low latency, expensive and face availability challenges when supporting large data.

Both data models' solutions are providing certain features and rules that support various business requirements, therefore, there is another solution that combines both relational and non-relational data models to exist in the same room, hence polyglot persistence infrastructure can be applied over an enterprise or within a single server.

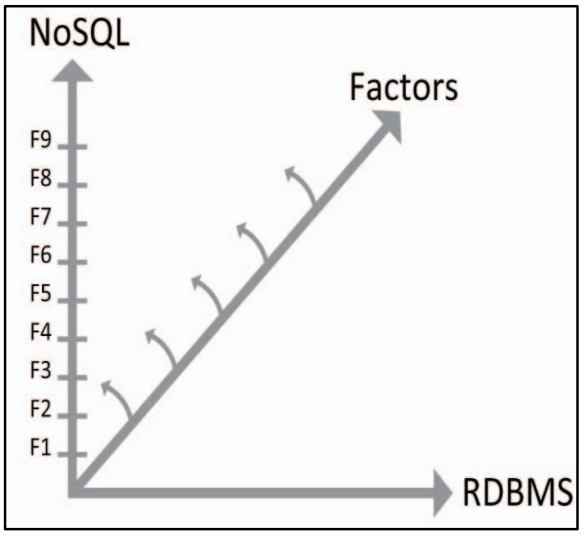

Figure 6: The factors role on moving the database model to NoSQL

This study revealed that the main reasons for migrating to NoSQL include: handling big data, heterogeneous data types, the Internet of Thing, data scaling, dynamic modeling schema, highly available transactions requirements, eschew ACID constraints, high performance, and software-as-a-service business.

REFERENCES

[1] M. Radoev, "A comparison between characteristics of NoSQL databases and traditional databases," *Computer Science and Information Technology,* vol. 5, no. 5, pp. 149-153, 2017.

[2] A. Raut, "NOSQL database and its comparison with RDBMS," *International Journal of Computational Intelligence Research,* vol. 13, no. 7, pp. 1645-1651, 2017.

[3] S. Venkatraman, K. Fahd, S. Kaspi, and R. Venkatraman, "SQL versus NoSQL movement with big data analytics," *International Journal of Information Technology and Computer Science,* vol. 8, no. 12, pp. 59-66, 2016.

[4] D. Kunda and H. Phiri, "A comparative study of nosql and relational database," *Zambia ICT Journal,* vol. 1, no. 1, pp. 1-4, 2017.

[5] M. MUS, "Comparison between SQL and NoSQL databases and their relationship with big data analytics," 2019.

[6] H. J. Hadi, A. H. Shnain, S. Hadishaheed, and A. H. Ahmad, "Big data and five V's characteristics," *International Journal of Advances in Electronics and Computer Science,* vol. 2, no. 1, pp. 16-23, 2015.

[7] S. Mukherjee, "The battle between NoSQL Databases and RDBMS," *Available at SSRN 3393986,* 2019.

[8] S. Mohanty, K. NathRout, S. Barik, S. K. Das, and A. Prof, "A study on evolution of data in traditional RDBMS to big data analytics," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 4, no. 10, pp. 230-232, 2015.

[9] D. Kumawat and A. Pavate, "Correlation of NOSQL & SQL Database," *IOSR J. Comput. Eng.(IOSR-JCE),* vol. 18, pp. 70-74, 2016.

[10] J.-K. Chen and W.-Z. Lee, "An Introduction of NoSQL Databases based on their categories and application industries," *algorithms,* vol. 12, no. 5, p. 106, 2019.

[11] Y.-L. Choi, W.-S. Jeon, and S.-H. Yoon, "Improving database system performance by applying NoSQL," *Journal of Information Processing Systems,* vol. 10, no. 3, pp. 355-364, 2014.

[12] G. Pepito, "RDBMS to NoSQL Migration: Challenges and Strategies."

[13] S. Ghotiya, J. Mandal, and S. Kandasamy, "Migration from relational to NoSQL database," in *IOP Conference Series: Materials Science and Engineering*, 2017, vol. 263, no. 4: IOP Publishing, p. 042055.
[14] S. A. Jowan, R. F. Swese, A. Y. Aldabrzi, and M. S. Shertil, "Traditional RDBMS to NoSQL database: new era of databases for big data," *J. Humanit. Appl. Sci,* vol. 29, no. 29, pp. 83-102, 2016.
[15] M. Garba and H. Abubakar, "A Comparison of NoSQL and Relational Database Management Systems (RDBMS)," *KASU Journal of Mathematical Science,* vol. 1, no. 2, pp. 61-69, 2020.
[16] W. U. Hasan, "Exploring NoSQL (Key-Value) DB vs Relational Database Management System."
[17] T. Patel and T. Eltaieb, "Relational database vs NoSQL," *Journal of Multidisciplinary Engineering Science and Technology (JMEST),* vol. 2, no. 4, pp. 691-695, 2015.
[18] M. R. Ahmed, M. A. Khatun, A. Ali, and K. Sundaraj, "A literature review on NoSQL database for big data processing," *Int. J. Eng. Technol,* vol. 7, no. 2, pp. 902-906, 2018.
[19] M. A. Mohamed, O. G. Altrafi, and M. O. Ismail, "Relational vs. nosql databases: A survey," *International Journal of Computer and Information Technology,* vol. 3, no. 03, pp. 598-601, 2014.
[20] C. Nance, T. Losser, R. Iype, and G. Harmon, "Nosql vs rdbms-why there is room for both," 2013.
[21] A. Bansel, H. González-Vélez, and A. E. Chis, "Cloud-based NoSQL data migration," in *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016: IEEE, pp. 224-231.
[22] P. Bailis, A. Davidson, A. Fekete, A. Ghodsi, J. M. Hellerstein, and I. Stoica, "Highly available transactions: Virtues and limitations," *Proceedings of the VLDB Endowment,* vol. 7, no. 3, pp. 181-192, 2013.
[23] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE access,* vol. 2, pp. 652-687, 2014.
[24] P. Azad, N. J. Navimipour, A. M. Rahmani, and A. Sharifi, "The role of structured and unstructured data managing mechanisms in the Internet of things," *Cluster computing,* vol. 23, no. 2, pp. 1185-1198, 2020.
[25] M. Cooney, "Cisco predicts nearly 5 zettabytes of IP traffic per year by 2022," (in English), Network World 2018. [Online]. Available: https://www.networkworld.com/article/3323063/cisco-predicts-nearly-5-zettabytes-of-ip-traffic-per-year-by-2022.html.
[26] Statista, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025," (in English), 2022. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. Statista Research Department.
[27] K. Machado, R. Kank, J. Sonawane, and S. Maitra, "A Comparative Study of ACID and BASE in Database Transaction Processing," *International Journal of Scientific & Engineering Research,* vol. 8, no. 5, pp. 116-119, 2017.
[28] P. Obilikwu, E. Ogbuju, and K. Kwaghtyo, "ENHANCING QUERY TIME USING A VOLUME-ADAPTIVE BIG DATA MODEL OF RELATIONAL DATABASES," *Journal of Basic Physical Research,* vol. 10, no. 1, pp. 73-93, 2021.