Comparison between relational and NOSQL databases

Conference Paper · May 2018

DOI: 10.23919/MIPRO.2018.8400041

5 authors, including:

Kosovare Sahatqija
South East European University
1 PUBLICATION **2** CITATIONS
SEE PROFILE

Jaumin Ajdari
South East European University
20 PUBLICATIONS **50** CITATIONS
SEE PROFILE

Xhemal Zenuni
South East European University
31 PUBLICATIONS **59** CITATIONS
SEE PROFILE

Bujar Raufi
South East European University
48 PUBLICATIONS **105** CITATIONS
SEE PROFILE

Comparison between relational and NOSQL databases

Kosovare Sahatqija, Jaumin Ajdari, Xhemal Zenuni, Bujar Raufi, Florije Ismaili
Contemporary Sciences and Technologies, South East European University, Tetovo, Macedonia
{ks16575, j.ajdari, xh.zenuni, b.raufi, f.ismaili}@seeu.edu.mk

***Abstract* - The relational database is provided from the traditional DBMS, which ensures the integrity of data and consistency of transactions. For many software applications, these are the principles of a proper DBMS. But in the last few years, witnessing the velocity of data growth and the lack of support from traditional databases for this issue, as a solution to it, the NoSQL (Not Only SQL) databases have appeared. These two kinds while being used for the same purposes (create, retrieve, update and manage data) they both have their own advantages and disadvantages over each other. The purpose of this study is to try and compare the research question of what are the pros and cons for each of these database' features and characteristics? This paper is a qualitative research, based on detailed and intensive analysis of the two database types, through use and comparison of some published materials during last few years.**

Keywords - NoSQL, Relational Databases, Big Data, ACID properties, BASE properties

I. INTRODUCTION

A database management system is essential in managing data efficiently and enabling users to complete various tasks with ease. Performance of this system improves the effectiveness of business processes and decreases overall costs. These systems provide a highly effective approach for handling multiple types of data.

Nowadays, organizations and companies struggle with applications which are accumulating big data on regular basis and as a result, their amount increases rapidly. The existing relational databases (usually referred as SQL databases) are widely used for the applications, but their performance degrades when the data volume is increasing, and they cannot handle the big data issue properly. Furthermore, during the development and evolution of the application, the relational database cannot modify the schema over time and the changes cannot handle different types of data. So, principally those were some of the main concerns why the NoSQL databases were introduced as an impetus for future improvement.

The NoSQL was introduced as a paradigm not to oppose the relational databases but to provide another option for the concerns that relational databases could not satisfy. So, NoSQL databases are not a replacement for the relational database but furthermore these two technologies can coexist. Storing data in the tables and in pre-defining the database schemas are some of the relational databases principles which NoSQL databases do not use or use it in very loosely coupled fashion. There cannot be a perfect solution or definitive one, but there are advantages and disadvantages of both databases.

The aim of this paper is to show the features and characteristics of each database technology especially for NoSQL as a new solution over relational databases. This paper is organized as follows. Introduction where is given an elementary introduce of the traditional relational databases and NoSQL databases. After this section continues the part where mention some achieved results from different researchers. This section also mentions some related work on this topic. A deeper introduction for the types of NoSQL databases is given on the third section, while on the fourth section NoSQL and relational databases comparison is described. On the fifth section there are presented some analysis and suggestions how to choose the right database type to software applications. Finally, the conclusions are on the sixth section.

II. RELATED WORK

There are several research papers about the NoSQL databases characteristics, features and their usage in the world of technology. There are also papers that have evaluated the performance of both NoSQL and SQL databases. In the [1], the authors elaborate the types of NoSQL databases and the advantages and disadvantages of NoSQL databases over relational databases. They defend the prospects of the NoSQL databases with a few more advancements in architecture.

Experimenting the performance and comparison of NoSQL and SQL databases have been made by the authors in [2, 4, 6] research papers. In [2], the authors have chosen to compare the performance of MongoDB as a NoSQL database and Microsoft SQL Server as a relational database. By experimenting with MongoDB and SQL Server, they have noticed that MongoDB performs faster in insert, update and simple query, whereas SQL Server performs better in update, query with non-key attributes and for aggregate queries.

The experiment in the [4] research paper was between MongoDB and MySQL. The authors have noticed that MongoDB shows faster performance compared to MySQL in insertion to the database a large amount of data. They also highlighted the features that dynamic schema of the NoSQL database offers, which are valuable in developing a big data application.

The similar results are shown in [6] in terms of

performance speed, where RavenDB and Microsoft SQL Server are compared, and the results show that on average RavenDb is 50% faster.

In [3, 5], the authors have deeply explained the feature comparison of NoSQL and relational databases including the scalability, performance, flexibility, querying, security, etc. By listing, pros and cons for both data management types, they explain why one should choose NoSQL over relational databases when developing a software application.

In research paper [10], another NoSQL and SQL databases comparison is done. The author has focused on the overhead because of the online transaction processing. The conclusion is that the overhead is not related to SQL, but to other components such are logging, locking, latching and buffer management. The author examines those four components. And avoidance of the overhead, associated with one or more of these, can provide a speedup of two. The author assumes that one can improve SQL databases to compete with NoSQL databases.

According to performance compare of both database technologies done in research paper [11], not all NoSQL databases perform better than SQL or relational databases. The authors have compared different types of NoSQL databases including MongoDB, RavenDB, CouchDB, Cassandra, Hypertable and Couchbase and as a relational database was chosen Microsoft SQL Express. The operations which were tested are read, write, delete and instantiate. A similar comparison but in different way is done in research paper [14], where the authors have done comparisons of SQL, NoSQL and NewSQL databases for the use in Internet of Things. The different way was chosen because of the different nature of data, which are sensor data. The performance results show the ranking of the technologies as NewSQL, NoSQL and then SQL databases.

The data security of NoSQL databases is a very interesting issue that is elaborated in [12, 15]. The security issue in [12] is elaborated using five factors which are authentication, access controls, secure configuration, data encryption and auditing. The result of this analysis shows that there is a significant demand to improve the security of sharded NoSQL databases. Whereas, in [15] the authors have been focused only on the security of MongoDB and Cassandra and have found that both have lack of encryption support for data files, weak authentication, and very simple authorization.

Corbellini et al. in their work [16] did a broader revision of NoSQL, referring to the four NoSQL types of databases. They have described the important features to each NoSQL database type and cover those with implemented example by explanation of the concrete implemented database management system of that type. They conclude that NoSQL, particularly, in data storage, show higher performances because of sharding and persist data storage support and in general should be considered when selecting a storage solution. The use of relational databases can easily show a weakness on the storage requirements in those situations. Furthermore, they conclude that in some situations, it may be necessary to define a hybrid data layer dividing the application data in multiple databases with different data layouts. Therefore, application data requiring a relational schema and high consistency, can be stored in a relational database, whereas the data with no need consistency, but with necessary fast access, a NoSQL database can provide an adequate solution.

Zeng et al. have made a comparison of relational and NoSQL databases through a concrete implementation solution [17]. They have made a comparison, by implementation the patient cohort identification systems in MySQL as relational platform into MongoDB and Casandra as NoSQL platforms. They have recognized the heterogeneous data sources into patient cohort identification which may involve a complication in data loading, harmonization and querying. The stated concerns are attempted to solve by use of the NoSQL. Utilizing NoSQL databases, they overcame the limitation of maximum table column count in traditional relational databases. This study indicates that NoSQL-based systems offer a promising approach for developing patient cohort query systems across heterogeneous data sources.

Ribeiro et al. in their paper [18] presents a performance comparison between the two main approaches of data storage, relational and NoSQL databases in the task of obtaining the most frequent N-grams (in Text Mining projects). The comparison was performed using a database from an international competition. It is shown that the NoSQL approach overcomes, in a statistically significant way, the relational database and they suggest Text Mining solution developers the use of the NoSQL technology to obtain better results in this field.

Kumar et al. have analyzed the problem of different data format and effectiveness in store and processing those data by use of relational database technology [19]. The use of NoSQL databases proved to be one of the best solutions for handling these kinds of schema-less data. This work comprises about the various characteristics of NOSQL databases, the performance comparison of two widely used Document-oriented NOSQL databases viz MongoDB and CouchDB are analyzed. Through two databases, for experimental use, the qualitative as well as quantitative features are analyzed.

Wisal et al., have compared Oracle relational database and NoSQL graph database using optimized queries and physical database tuning techniques [21]. They have done various experiments by execution of the various queries and show that whenever data becomes more and more connected (large number of joins) and large in size, relational databases show worse performance than NoSQL graph database. According to them, this is due that relational databases use constraints, indexes and do not store any relationship information, while NoSQL graph database stores relationship information among various nodes.

A similar comparison of relational and NoSQL database is done by Raut [20].

III. TYPE OF NOSQL DATABASES

The rapid increase of the amount of data and the problem of changing the schema database over the evolution of the different platforms were the first concerns that motivated further development of NoSQL databases. Most of the NoSQL systems are distributed databases or distributed data storage focusing on high performance, high availability, data replication and scalability as opposed to an emphasis on immediate data consistency, powerful query languages and structured data storage [7].

There are four types of the NoSQL databases that can be categorized based on their model of storing and retrieving data. Those types are key – value, column-oriented, document-based and graph databases.

A. Key-value

Key-value model is a schema-less database which is implemented using a hash table where keys are stored as indexes and a pointer that holds the actual data. This structure creates the 'key-value' pair as the model itself is named. The hash tables are suitable for lookups for simple or complex values in extremely large datasets [1, 5]. The data in the key-value database are stored in the form of rows as structured data but also can be stored as JSON or some other self-describing data format as semi structured data [7]. In overall, these databases were created for fast and efficient data management in distributed systems.

One of the more important database management system of this type, key-value NoSQL database is DynamoDB, introduced by Amazon and Amazon use it for its shopping cart.

B. Column-oriented

Column-oriented model is a wider concept of key-value architecture, organized by columns. This model is a composite approach to relational databases and key-value model schema. The data are stored in column families and rows. Every row has a row-key and a row may contain many columns. A row can have a different number of columns and in case of nested columns inside another column, those columns are called super columns [13]. This database type works very well with complex datasets as a result of its scalability. Works perfectly with the enormous amount of data in distributed systems because of its time stamping functions [13]. The first technology that introduced this schema model is BigTable, developed by Google, to handle Google's applications amount of data storage such as Gmail, Google Maps, Web site indexing, etc.

C. Document-based

The most used of NoSQL database model is the document – based database model [6]. The document-based model perfectly handles all types of data including structured, semi-structured and unstructured data. This NoSQL model stores data as collections of documents. The document can contain multiple key – value pairs, key – array pairs or even can contain nested document. The documents in a collection should be similar, but a document can contain attributes that are not necessarily need to have other documents in that collection.

Document – based database model is appropriate for use on content management systems, blog software, etc. Otherwise, it should be avoided in case that the database needs to have a lot of relations and normalizations [1].

The most used Database Management Systems, based on document – based model, are MongoDB and CouchDB.

D. Graph databases

The data in a graph database model are stored and represented as graphs, which are a collection of nodes and edges. Nodes act as objects (entities) and edges represent relationships between data. It can be considered as a model of the network of relationships between specific key-value pairs. This model can support complex data queries for a relatively short period of time, also can support ACID properties [4] and the rollback feature which ensures the consistency of data. This type of database is used when the importance is given on the relationships between data than the data itself [13].

It is important to mention that even though the graph databases have relationships they have nothing to do with relational databases [5]. The most used Database Management Systems based on graph database model is Neo4j.

IV. COMPARING SQL AND NOSQL DATABASES FEATURES

This section compares SQL to NoSQL databases features such as scalability and performance, flexibility, query language, security, data management including storage and availability of data, and their advantages and disadvantages over each other.

A. Scalability and performance

The scalability of the DBMS is very important when we choose the database management system for our software application. Relational databases (SQL databases) use vertical scalability, which means that when the volume of data is being expanded, there could be expand just the storage capacity and computing power of existing node, for example, the capacity of CPU, the RAM and the SSD of the database server [5, 7]. This kind of scalability is expensive because of grater hardware failure risk, hardware costs in means of future upgradability (hardware became older and the support is less, vendors may have some requests, hardware and software limitations, etc.), so the overall implementation cost will increase with data growth. Whereas the NoSQL databases use horizontal scalability which means that when the volume of data is rapidly growing, and the volume of data is large the system expand by adding more nodes for data storage and processing power, for example, add servers to the NoSQL database infrastructure [5]. So, the horizontal scalability of the system is a cheaper solution than the vertical scalability. Inherently the NoSQL databases support the auto – sharding feature by distributing data on different servers, which increases the performance of the database [12].

The main priority of SQL databases is to meet the

ACID (Atomicity, Consistency, Isolation, Durability) properties which are like an impossible task for NoSQL databases. By use of the horizontal scalability is difficult to meet the ACID features. The ACID properties ensure more reliability and integrity of data from SQL databases in comparison to NoSQL databases. Nevertheless, NoSQL databases rely on BASE principles (Basically Available, Soft state, Eventually consistent). Both properties are obtained from CAP theorem which is Consistency – the data is always the same in every replication on every server, Availability – the data must always be accessible (permanently available) and Partition tolerance – the database works fine despite network and machine failures. This theorem says that it is impossible to satisfy and guarantee all three aspects at the same time for distributed systems. So, there will be needed to choose just two of them [6, 8]. So, ACID properties have consistency and reliability whereas BASE properties are more flexible.

B. Flexibility

The flexibility of changing the database schema during the development or the evolution of a software application is not a feature that every DBMS can provide. The SQL databases have a static database schema that should be pre-defined before data injection and should support structured data. If there is a need to change the schema, with pre-existing data, there is a huge problem and a modification of the database schema or tables should be considered precisely, because that modification can cause service failure, decrease performance, or may need maintenance and further investments to modify application modules. While on the other hand, NoSQL databases have a dynamic schema and not necessarily need to be pre-defined. NoSQL databases can easily accommodate changes in data type / structure due to its dynamic schema design [3]. The NoSQL databases because of their data modeling are used for agile and scalable environments which will be continuously developing and evolving.

Another issue for the flexibility of the database is the data structure. The SQL databases handle just well – structured data. This can reduce the performance of the database as data volume increases. While NoSQL databases handle every kind of data including their well – structured, semi – structured and unstructured data.

C. Query language

Relational databases use a standard query language known as Structured Query Language (SQL). This query language is a powerful one and can handle complex queries through a standardized interface.

On the other side, the NoSQL databases do not have a standardized language to query and manage data. However, every NoSQL database management system vendor has created their own query language but there is a lack of creating complex queries such as aggregation on NoSQL databases. Many NoSQL systems do not provide join operation as part of their query language, so the joins need to be implemented on the application side [7].

The fact that there is not a standard query language for NoSQL databases, creates difficulties when data scientists face the challenge to understand the query language for each database. Therefore, there is a need to create a standardized query language for NoSQL databases. Hence the SQL query language is an advantage of relational databases over NoSQL databases.

D. Security

The security is an important issue for a DBMS. The relational databases have very secure mechanisms which ensure the security of the services [8]. Since the feature of sharding is considered the key to success of NoSQL databases by distributing data over servers this probably has impact in data security as the most difficult challenge for NoSQL databases. There is a concern, how the confidentiality, privacy and the security of the data are guaranteed from these systems. Most of the NoSQL databases do not have secure client-server communication and do not provide these mechanisms that can ensure security [5]. There are some key factors that should be considered when dealing with the security of databases. Those factors are authentication, access control, secure configurations, data encryption, and auditing [12]. To ensure the authentication, authorization, and auditing there should be external methods to perform the operation and should be implemented based on the NoSQL database used. It is the same way in defining the access control of the users, some of the NoSQL databases provide access control from the system, but some of them do not ensure this kind of mechanism and need to implement it from the third party.

Because of structured data in the relational databases can be easy to manage the security issues. Therefore, in the NoSQL databases, a large amount of unstructured data and the lack of encryption can affect the security of the database. Based on these data we can see clearly that there is still room for improvement for the security of NoSQL databases in the future.

E. Data management - Storage and Access

In relational databases, data stored are highly normalized and very clean. The data redundancy is avoided in a remarkable way using normalization by slicing data in small logical tables and preventing duplication. In this way, happens the improvement and usage of storage in a reasonable manner.

On the other hand, the data in NoSQL database are stored in collections without relationships and normalization between each other so this could contain data redundancy.

Replication is useful when improving the data availability. NoSQL databases practice the data replication of the database between clustered servers, in order to prevent data loss and to guarantee the security of data. The replication process is done in two ways: master-slave and master-master [7]. Master-slave replication allows the slave to take a copy of the data just for read, while the master holds the permission to write and read the data, so this way guarantees the consistency of data. While master – master replication allows reads and writes to any of the copies and this may lose the consistency.

In the relational databases, a replication is when the whole database is replicated in every site of the distributed system. It is factual that the availability of data is improved but the performance of the database operations will be decreased obviously. There is another way called no replication which replicates a fragment in just one site, here are replicated just the primary keys to show the relationships. The third way is called partial replication where there are some fragments that are replicated, and some may not. In principle, this replication over distant sites consumes a lot of time and storage, especially when having big data sets.

V. CHOOSE THE BEST DBMS

When determining the technologies and the working environment for our software application, we initially should know exactly the system requirements, the constraints that application should satisfy and the issues which can encounter during development. Different software applications have different needs and requirements and along with them easily can be determined the appropriate technologies that we should select. When it comes to DBMS there are some key points that can be valuable in this process.

If the software application priorities are integrity and consistency of data, then the answer should be relational databases. Certain platforms usually have sensitive data and they need to ensure the security of the transactions and other sensitive actions. For example, for the sensitive data as bank transactions, financial, accounting records there is a need for security and consistency of data and is essential to perform several actions as a single atomic action, so the only reliable option is RDBMS. Working with data that need to connect multiple collections into a single result should be considered the relational database. There should be considered also security as an important factor besides the consistency and availability. Moreover, when the ACID properties are necessary for the platform then there is no space for doubt that the only choice is the relational database.

When we want to go beyond the possibilities that relational databases can offer us, for example, in cases where availability, performance, and scalability are essential then NoSQL databases can be the best choice. When software application does not require or does not have priority the data integrity then NoSQL databases can be a perfect choice. NoSQL can avoid the complexity of relational databases [5]. For example, social media sites, shopping sites, email providers, etc., can be the software applications which can consider NoSQL databases as perfect solutions because of the fact that they have a large amount of data, their data grows rapidly on daily basis and, they have different types of data which can be unstructured. Another issue that we must consider is the size of the platform and if there is a need to change the data model at any time without affecting the system or application performance then NoSQL database is the answer. These databases are built to develop an agile and scalable business environment while improving and expanding more the technology systems.

However, there can be cases where software applications can benefit even more from using different solutions within the application for particular tasks. Applications that need high – speed transactions and fast response, or that perform complex queries on data in real time it is good to consider combining different or multiple database technologies for some processing needs. The combination of the SQL based database technology and currently popular NoSQL databases technologies could create a better system with greater availability, scalability, and performance.

VI. CONCLUSION

Overall NoSQL databases just as relational databases have the advantages and disadvantages over each other. The process of switching from a relational database to a NoSQL database can be very challenging in many ways. Firstly, it is needed a detailed study of both solutions, their features, and their querying options. The emerge of NoSQL databases was not intended to ruin the relational databases market, but to bring a solution for the imperfections of them. The relational databases are very used because of their traditionalism and reliability and stability. Their existence over the years has proven to the users their indisputable quality.

Principally the NoSQL users are not so much, in number, comparing to relational database users, so this could be a weak challenge to the NoSQL databases in convincing the new users on applying of this new solution. Also, the lack of having a standard query language may be a additional fact on hesitate to use the NoSQL databases. Changing or pass from one NoSQL type to another requires the need of learning from the beginning of its query language technology, which can additional complicate the use.

There is no an absolute answer, which is the best database solution. To choose the appropriate database for certain software application we should consider some database key points such as scalability, flexibility, performance, query language, security, data replication, and availability.

When ACID properties are necessary, relational database are the appropriate choice. NoSQL databases have a more flexible model comparing to the relational databases, making it easier to organize large amounts of data with varied formats and with flexible increase over time. If there are large datasets, the need of constantly schema change and there is a need for performance and flexibility then NoSQL is the perfect solution.

In general, NoSQL databases are well experienced on big data evolution and they can perfectly face a huge growth of data, but there is a lack of security issues and they should use some of external method to perform and ensure the database security. Considering the development and improvement of NoSQL database technologies the prospects are excellent.

There is possible to make a combined system of both kind of technologies, where relational and NoSQL databases can be configured together and collaborate without obstacles furthermore, they can combine the

benefits of each technology solution to create a more effective system.

REFERENCES

[1] A. Nayak, A. Poriya, and D. Poojary, "Type of NoSQL Databases and its Comparison with Relational Databases," International Journal of Applied Information Systems (IJAIS), vol. 5 - no. 4, pp. 16-19, March 2013.

[2] Z. Parker, S. Poe, S. V. Vrbsky, "Comparing NoSQL MongoDB to an SQL DB", In Proceedings of the 51st ACM Southeast Conference (ACMSE '13). ACM, New York, NY, USA, Article 5, 6 pages, April 4-6, 2013.

[3] T. Partel and T. Eltaieb, "Relational Database vs NoSQL," in Journal of Multidisciplinary Engineering Science and Technology (JMEST), vol. 2, pp. 691–695, April 2015.

[4] C. Győrödi, R. Győrödi, G. Pecherle, A. Olah, "A Comparative Study: MongoDB vs. MySQL", Conference: The 13th International Conference on Engineering of Modern Electric Systems, pp. 1-6. Oradea, 11-12 June 2015

[5] A. Oussous, F. Z. Benjelloun, A. A. Lahcen, S. Belfkih, "Comparison and Classification of NoSQL Databases for Big Data", International Journal of Database Theory and Application, vol. 6 (4), 2013

[6] Fraczek K., Plechawska-Wojcik M., "Comparative Analysis of Relational and Non-relational Databases in the Context of Performance in Web Applications", Proceedings, 13th International Conference, Beyond Databases, Architectures and Structures (BDAS 2017), pp. 153-164, Ustroń, Poland, May 30 - June 2, 2017

[7] R. Elmasri and S. Navathe, Fundamentals of Database Systems. 7th ed., Chap. 23-24, Pearson Education, 2015.

[8] M. A. Mohamed, O. G. Altrafi, and M. O. Ismail, "Relational vs. NoSQL databases: A survey" in International Journal of Computer and Information Technology, vol. 03, no. 03, pp. 598–601, May 2014.

[9] A. Abdullah, Q. Zhuge, "From Relational Databases to NoSQL Databases: Performance Evaluation", Research Journal of Applied Sciences, Engineering and Technology, pp. 434-439, May 2015.

[10] Stonebreaker, M. 2010. SQL Databases v. NoSQL Databases. Communications of the ACM, Vol. 25, No. 4, pp. 10-11.

[11] Y. Li and S. Manoharan, "A performance comparison of SQL and NoSQL databases" in IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Canada, Aug. 2013, pp. 15- 19.

[12] A. Zahid, R. Masood and M. A. Shibli, "Security of Sharded NoSQL Databases: A Comparative Analysis" in IEEE Conference on Information Assurance and Cyber Security, Pakistan, Jun. 2014, pp. 1-8.

[13] J. Bhogal and I. Choksi, "Handling Big Data using NoSQL" in IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, South Korea, Mar. 2015, pp. 393-398.

[14] F. Haleemunnisa and W. Kumud, "Comparison of SQL, NoSQL and NewSQL Databases for Internet of Things" in IEEE Bombay Section Symposium, India, Dec. 2016, pp. 1-6.

[15] L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes and J. Abramov, "Security Issues in NoSQL Databases" in IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, China, Nov. 2011, pp. 541-547.

[16] A. Corbellini, C. Mateos, A. Zunino, D. Godoy and S. Schiaffino. "Persisting Big Data: The NoSQL landscape". Information Systems. Vol. 63, pp. 1-23. Elsevier Science, 2017.

[17] Zeng N, Zhang GQ, Li X, Cui L. "Evaluation of relational and NoSQL approaches for patient cohort identification from heterogeneous data sources." Journal of Health & Medical Informatics, Vol 8(5), 2017 pp. 1- 9.

[18] Ribeiro, Jardel, et al. "NoSQL vs relational database: A comparative study about the generation of the most frequent N-grams." Conference Procedeengs, Systems and Informatics (ICSAI), 2017 4th International Conference on. IEEE, 2017, pp. 1568 - 1572

[19] Kumar, K. B. Sundhara, Srivydia and S. Mohanavalli. "A performance comparison of document oriented NoSQL databases." Computer, Communication and Signal Processing (ICCCSP), 2017 International Conference on. IEEE, 2017.

[20] Raut A. B. "NOSQL Database and Its Comparison with RDBMS". International Journal of Computational Intelligence Research, Volume 13, Number 7 (2017), pp. 1645-1651

[21] Wisal K., Ejaz A., Waseem S., "Predictive Performance Comparison Analysis of Relational & NoSQL Graph Database", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 5, 2017, pp. 523 - 530