The SQL vs NoSQL Differences and Similarities

Twana Asaad TAHA

Abstract - The conventional SQL database is portrayed in the conventional DBMS, which guarantees the respectability of information and consistent connections. For some product development, these are the standards of an appropriate DBMS. However, over the most recent couple of years, seeing the speed of information development and the absence of assistance from conventional databases for this problem, the Not Only SQL (NOSQL) databases were created. The two categories of databases, although being utilized for similar objectives, have their own preferences and hindrances over one another. Accordingly, the reason for this investigation is to attempt to look at the exploration question of the differences together with the similarities of both databases.

Keywords – DBMS, SQL, No Only SQL (NOSQL), Relational databases, ACID property, CAP hypothesis, BASE property.

1. INTRODUCTION

Technology has provided many significant improvements to processes and activities that organizations engage in, including enabling data to be stored in databases. Basically, a database comprises of data that is stored and accessed using a computer.

Accordingly, technology continued to improve whereby the "database management system" (DBMS) was designed, which is a system software for the purposes of not only creating but also managing one or more databases. This way, programmers as well as users are provided with a manner that is systematic when creating, retrieving, updating, and managing data in a database. Simply put, a database management system functions as the interface amongst the applications or end users and a database, which ensures that data is organized consistently and can be accessed easily. When it comes to organizations, performance of database management systems increases the viability of professional procedures and diminishes general expenses. Similarly, these frameworks offer an exceedingly effective way to handling different data types.

These days, associations and organizations battle with applications which are aggregating huge information on normal premise and subsequently, their sum increments quickly. In accordance with Coronel and Morris [1], the current relational databases – normally alluded as SQL databases – are broadly utilized for such kinds of applications; however, their performance is noted to decrease when the volume of data is expanding, in turn, the databases are not able to handle the enormous data volume issue appropriately.

Moreover, when developing or upgrading applications that aggregate huge data, the conventional relational database does not have the capacity for modifying database schemas over time, which means that any changes made will make the database not able to deal with different data types. Thus, essentially Hoffer, Ramesh, and Topi [2] state that there are a portion of the fundamental concerns relating to why the Not Only SQL (NOSQL) databases were presented as a driving force for future improvement.

Pokomy [3] articulates that rather than being introduced as an approach or model for opposing the conventional relational database, the NOSQL database was designed to provide another choice to mitigate the worries that relational databases failed to fulfill. Based on this explanation, Leavit [4] is in agreement that the NOSQL databases are not for replacing the conventional SQL database based on the fact that the two advances can exist together. Still, there are notable differences between the two kinds of databases. For example, Han, Haihong, Le, and Du [5] maintain that data storage in a table and pre-characterizing a database schemas are a portion of the conventional SQL database standards, which are not utilized in NOSQL databases. Consequently, Strauch, Sites, and Kriha [6] contend that it is difficult for there to be an ideal arrangement or complete one, yet there are both disadvantages and advantages of the two databases.

2. RELATED WORK

There are several research papers that have investigated not only the features and characteristics but also the adoption and practice of using NOSQL databases in the realm of innovation. Similarly, there are additionally studies that have assessed the exhibition of the SQL and NOSQL databases. For instance, in research study [5], the authors expand the categories of NOSQL databases together with their pros and cons over SQL databases. In their analysis, the authors [5] elaborate on the possibilities of NOSQL databases with a couple of more progressions in terms of design.

Tentatively, Nayak, Poriya, and Poojary [10] have profoundly clarified the element correlation of SQL and NOSQL databases including the security, performance, scalability, querying language, flexibility, and so forth. By determining the advantages and disadvantages for the databases, they clarify clients ought to pick NoSQL over SQL databases in the development of software applications.

While considering the performance assessment of both databases in research paper [11], it is clear that not all types of NOSQL databases execute in a superior manner over SQL databases. This way, the researchers have used the "MongoDB", which a NOSQL database to compare with the relational database Microsoft SQL Express. The activities in their investigation included testing, reading, writing, deleting as well as instantiating data in the databases.

A comparable investigation but in a manner that is different is completed in the research study [14], where the researchers have done examinations on SQL and NOSQL databases for the utilization in Internet of Things. The diverse study mechanism was picked in view of the distinctive data types that exist. The presentation results demonstrate the positioning of the innovations as NoSQL and afterward SQL databases.

Tudorica and Bucur [7] have dissected the issue of varying data types as well as the adequacy of storing and handling the data types by utilizing the conventional relation database. Subsequently, the utilization of NoSQL databases demonstrated to be perhaps the best answer when handling different data types. In this manner, the two databases were utilized for exploratory analysis to assist in determining the quantitative and qualitative concepts.

Kaur and Rani have investigated the comparison of an Oracle SQL database against a NOSQL graph database utilizing improved questions and database modification procedures [8]. The researchers have performed different examinations by assessing the different questions and demonstrate that at whatever point information turns out to be increasingly associated and huge in size, the Oracle SQL databases show more regrettable execution as compared to the NOSQL graph database. As indicated by them, this is expected that relational databases utilize indexes and constraints while not storing any relationship data, whereas the NoSQL database stores relationship data among different hubs and nodes.

3. TYPES OF NOSQL DATABASES

Tudorica and Bucur [7] express that the quick increment of volume of data together with the issue of making changes to the database schemas over the advancement of the various existing database management systems are the main worries that roused additional improvement of the NOSQL database. Currently, the majority of the NOSQL database systems comprise of databases that are distributed as well as scattered data storage that is focused on increased efficiency, enhanced accessibility, data duplication together with versatility of data instead of an accentuation on immediate consistency of data, powerful inquiry language features, and organized data storage.

Since their development, there are several categories of NOSQL databases that are classified dependent on their model of retrieving and storing data. The categories are discussed as follows;

a) Document-based

Kaur and Rani [8] state that the document-based is the most utilized kind of NOSQL database. For the most part, the document-based NOSQL database is able to impeccably handle a wide range of data types, including organized, semi-organized as well as those that are unstructured. What's more, the document-based NOSQL database is able to store data in the form of collections or accumulations of reports.

Similarly, this model has a feature whereby records can contain various types of keys. In the conventional relation database, Han, Haihong, Le, and Du [5] indicate that documents stored are required to be comparable; yet, a document can contain characteristics that are not really needed by other documents in that accumulation. In this manner, the document-based NOSQL database model is fitting for use on complex systems, like systems for managing content or blogging software. Nevertheless, Kaur and Rani [8] advise that this type of model ought to be kept away from on the off chance that that the database requires the use of normalization together with relationships.

b) Column-oriented

The column-oriented NOSQ database model illustrates a more extensive design, which is organized in terms of columns or by segments. This model demonstrates a composite way to deal with the conventional relational databases and in terms of database schemas. Basically, Hecht and Jablonski [9] indicate that data store in the database is organized in section families and columns. In addition to that, every single row in the database is provided with a key and a row may likewise contain numerous segments.

Nayak, Poriya, and Poojary [10] suggest that this database type of NOSQL database functions efficiently with complex datasets because of its adaptability. The database is additionally able to works flawlessly with the gigantic volumes of data in distributed frameworks in view of its time-stamping capacities. The principal innovation that presented this outline model is called the "Big Table," which was created by Google to provide a way to deal with Google's applications increasing data volume, for example, Gmail, Google Maps, and so forth.

c) Graph databases

In this NOSQL database model, data and information are not only stored but are also represented in the form of diagrams – specifically graphs, which are an accumulation of edges and nodes. McCreary and Kelly [11] mention that nodes in a graph function entities or objects whereas edges are used to express connections that exist between information.

As such, this type of database can reinforce complex information questions for a moderately brief timeframe, and can likewise can bolster ACID properties [12] together with the feature called "rollback" that guarantees the consistency of information. Consequently, this kind of database is utilized when the significance is given on the connections between information as opposed to the information itself [11]. McCreary and Kelly maintain that it is essential to specify that despite the fact that the chart database models portray having relationships, they do not necessarily have anything to do with the conventional relational databases [10].

d) Key-value

Key-esteem NOSQL database model is a database that contains no schema, which is executed utilizing a "hash table" where keys are put away as lists while a "pointer" is used to contain the real information. As a result, this structure is what gives the database its name "key-value." At the same time, the "hash tables" are appropriate for queries for basic or complicated qualities in very large sets of data [13].

Data and information in this type of database are put away as columns as organized information [14]. In general, this type of databases was designed for quick and productive management of data in distributed frameworks. Abramova and Bernardino indicate that an example of the "key-value" NOSQL database is called the "Dynamo DB," which was designed and is being utilized by Amazon for its shopping basket.

4. COMPARISON OF SQL AND NOSQL DATABASE FEATURES

In this section, a comparison of the features of the SQL and NOSQL databases will be discussed. This will include features of scalability and performance, flexibility, query language, and security.

a) Scalability and performance

The adaptability of the database management system is significant when selecting the database management system for particular product application. At the onset, the conventional SQL databases utilize vertical versatility, which implies that when the data volume is being increased, there could an increase as well in the storage limit together with power of computing that is existing in the node [13].

This sort of versatility is costly a result of the potential risk of hardware equipment failure, equipment costs in methods for upgrading in the future whereby equipment is bound to become older whereas support for the model becomes less, merchants may have a few solicitations, equipment and programming impediments, and so forth. Essentially, the general usage cost will increment with information development.

On the other hand, the NOSQL databases utilize horizontal adaptability, which implies that when the data volume is quickly increasing, and the volume of information is huge, then the framework expands as well by including more hubs for increasing processing power together with storing data, for instance, through adding servers to the infrastructure of the NOSQL database [15]. In this way, the flat versatility of the framework is a less expensive arrangement than the vertical adaptability. Intrinsically, the NoSQL databases bolster the feature of "sharding" by dispersing information on various servers, which expands the exhibition of the database [13, 15].

Tentatively, the main concern of SQL databases is to meet the "Atomicity, Consistency, Isolation, and Durability" (ACID) properties which resemble a difficult undertaking for NOSQL databases. The reason is because the NOSQL database implements horizontal scalability, which makes it problematic to meet the ACID properties [10]. The ACID properties guarantee greater unwavering quality and respectability of information from SQL databases in contrast with NOSQL databases.

All things considered, NOSQL databases depend on BASE standards, which stands for "Basically available," "Soft state," and "Eventually predictable." The two properties are acquired from the CAP hypothesis: Consistency – ensuring that the information is consistently the equivalent in each replication on each server; Availability – ensuring that the information should consistently be open and accessible); and Partition resistance, which means ensuring that the database works fine in spite of system and machine disappointments. This hypothesis says that it is difficult to fulfill and ensure every one of the three perspectives simultaneously for appropriated frameworks. Along these lines, there will be expected to pick only two of them [10, 12]. Along these lines, ACID properties portray consistency and unwavering quality even though BASE properties are increasingly adaptable.

b) Flexibility

The adaptability of modifying or making changes to the database during the improvement or the development of a product software is not a component that each database management system can offer. In this manner, the SQL databases utilize a static database pattern that ought to be pre-characterized before information infusion and should bolster organized information. On the off chance that it is necessary to modify, there is a tremendous issue and an alteration of the database schema or tables ought to be measured decisively, in light of the fact that that adjustment can cause server failure, decline execution, or may require upkeep and supplementary ventures to alter the database.

In contrast, NOSQL databases utilize a dynamic pattern and are also not really required to be pre-characterized. NoSQL databases can without much of a stretch oblige changes in information type/structure because of its dynamic outline plan [16]. The NoSQL databases on account of their information displaying are utilized for nimble and adaptable situations which will be constantly developing and advancing.

Tentatively, another concern for the adaptability of the database concerns the information structure. The SQL databases is able to handle this occurrence simply well because of how data is organized. However, Hammes, Medero, and Mitchell [17] state that this can lessen the presentation of the database as information volume increments. In retrospect, the NoSQL databases handle each data type as well as information including their well – organized, semi – organized and unstructured information.

c) Query language

The conventional SQL database utilizes a standard question language known as "Structured Query Language" (SQL). This question language is a ground-breaking one and can deal with complex inquiries through an institutionalized interface.

On the opposite side, the NOSQL databases do not have an institutionalized language to inquiry and oversee information. Be that as it may, each NOSQL database management framework designer has made their own question language, yet there is an absence of making complex inquiries, for example, performing aggregation functions in the databases. Apart from that, numerous NoSQL frameworks do not offer the function of joining tasks as a major aspect of their question language, so the joins should be actualized on the application side [18].

Rautmare and Bhalerao [19] allude that the way that there is anything but a standard inquiry language for NOSQL databases, makes troubles when information researchers face the test to comprehend the question language for every database. Along these lines, there is a need to make an institutionalized question language for NOSQL databases. Thus the SQL question language is a favorable position of social databases over NOSQL databases.

d) Security

Security is a significant issue for a database management system. The relational databases have exceptionally secure systems which guarantee the protection of services as well as the users [20]. Since the element of "sharding" is viewed as the way to achievement of NOSQL databases by disseminating information, this likely has sway in information security as the most troublesome test for NOSQL databases. There is an issue, relating to how the secrecy, protection and the security of the information are ensured from these frameworks.

Essentially, the majority of the NOSQL databases do not have safe customer server correspondence and also do not give these components that can guarantee security [20]. As such, there are various factors that ought to be viewed as when managing the security of databases. Those variables are confirmation, get to control, secure designs, information encryption, and reviewing [12].

To guarantee the verification, approval, and inspecting there ought to be outside techniques to play out the activity and ought to be executed dependent on the NOSQL database utilized. It is a similar route in characterizing the entrance control of the clients, a portion of the NoSQL databases give access control from the framework, however some of them don't guarantee this sort of instrument and need to actualize it from the outsider.

On account of organized information in the conventional relational databases can be anything but difficult to deal with the security issues. In this manner, in the NOSQL databases, a lot of unstructured information and the absence of encryption can influence the database protection [20]. In view of these information we can see plainly that there is still opportunity to get better for the protection of NOSQL databases created in the coming future.

5. CONCLUSION

Generally speaking NOSQL databases similarly as the conventional relational databases demonstrate pros and cons over one another. Regardless, the process of changing to a NOSQL database from an SQL database can be trying from numerous points of view. For instance, it is crucial to provide a point by point investigation of the two arrangements, their highlights, and their questioning choices. Still, the development of NOSQL databases was not planned to demolish the market of the relational database; instead, it was designed to bring an answer for the flaws of both databases. Nevertheless, the relational databases are extremely utilized on account of their conventionalism, dependability as well as stability. Their existence throughout the years has demonstrated to the clients their unquestionable quality.

Essentially the NoSQL clients are less, in number, contrasting with the conventional relational database clients, which means that this could be a frail test to the NOSQL databases in persuading the new clients on applying of this new arrangement. Likewise, the absence of having a standard query processing language might be an extra certainty on user hesitation to utilize the NOSQL databases. Altogether, when selecting the fitting database for certain product application, it is imperative to think of some as database key focuses, for example, the query processing language, data availability, flexibility, data replication, performance, and scalability.

At the point when ACID properties are important, the investigation has determined that the relational database (SQL) is the more suitable decision. On the other hand, the NOSQL databases have a progressively adaptable model contrasting with the relational databases, making it simpler to sort out a lot of information with changed arrangements and with adaptable increment after some time. In the event that there are huge datasets, there is always the need for making changes to the database schema and there is a requirement for both flexibility and performance, which means that NOSQL is the ideal solution.

All things considered, the NOSQL databases are all around experienced on enormous information advancement and they can perfectly handle huge volumes of data, yet there is an absence of security issues and they should utilize some of outside technique to perform and guarantee the database security. Consequently, bearing in mind the advancement and development of NOSQL database, the future prospects will undoubtedly be exceptional.

REFERENCES

[1] Coronel, C. and Morris, S., 2016. *Database Systems: Design, Implementation, & Management*. Cengage Learning.

[2] Hoffer, J.A., Ramesh, V. and Topi, H., 2011. *Modern Database Management*. Upper Saddle River, NJ: Prentice Hall.

[3] Pokorny, J., 2013. NoSQL Databases: *A Step to Database Scalability in Web Environment*. International Journal of Web Information Systems, 9(1), pp.69-82.

[4] Leavitt, N., 2010. *Will NOSQL Databases Live Up To Their Promise?* Computer, 43(2), pp.12-14.

[5] Han, J., Haihong, E., Le, G. and Du, J., 2011, October. *Survey on NOSQL Database.* In 2011 6th International Conference on Pervasive Computing and Applications (pp. 363-366). IEEE.

[6] Strauch, C., Sites, U.L.S. and Kriha, W., 2011. *NOSQL Databases.* Lecture Notes, Stuttgart Media University, 20.

[7] Tudorica, B.G. and Bucur, C., 2011, June. *A Comparison between Several NOSQL Databases with Comments and Notes.* In 2011 RoEduNet international conference 10th edition: Networking in education and research (pp. 1-5). IEEE.

[8] Kaur, K. and Rani, R., 2013, October. Modeling and querying data in NoSQL databases. In *2013 IEEE International Conference on Big Data* (pp. 1-7). IEEE.

[9] Hecht, R. and Jablonski, S., 2011, December. NoSQL evaluation: A use case oriented survey. In *2011 International Conference on Cloud and Service Computing* (pp. 336-341). IEEE.

[10] Nayak, A., Poriya, A. and Poojary, D., 2013. Type of NOSQL databases and its comparison with relational databases. *International Journal of Applied Information Systems*, *5*(4), pp.16-19.

[11] McCreary, D. and Kelly, A., 2014. Making sense of NoSQL. *Shelter Island: Manning*, pp.19-20.

[12] Have, C.T. and Jensen, L.J., 2013. Are graph databases ready for bioinformatics? *Bioinformatics*, *29*(24), p.3107.

[13] Li, Y. and Manoharan, S., 2013, August. A performance comparison of SQL and NoSQL databases. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (pp. 15-19). IEEE.

[14] Abramova, V. and Bernardino, J., 2013, July. NoSQL databases: MongoDB vs cassandra. In *Proceedings of the international C* conference on computer science and software engineering* (pp. 14-22). ACM.

[15] Parker, Z., Poe, S. and Vrbsky, S.V., 2013, April. Comparing NoSQL mongoDB to an SQL database. In *Proceedings of the 51st ACM Southeast Conference* (p. 5). ACM.

[16] Lee, C.H. and Zheng, Y.L., 2015, October. SQL-to-NoSQL schema denormalization and migration: a study on content management systems. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 2022-2026). IEEE.

[17] Hammes, D., Medero, H. and Mitchell, H., 2014. Comparison of NoSQL and SQL Databases in the Cloud. *Proceedings of the Southern Association for Information Systems (SAIS), Macon, GA*, pp.21-22.

[18] Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y. and Wilkins, D., 2010, April. A comparison of a graph database and a relational database: a data provenance perspective. In *Proceedings of the 48th annual South East regional conference* (p. 42). ACM.

[19] Rautmare, S. and Bhalerao, D.M., 2016, October. MySQL and NoSQL database comparison for IoT application. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)* (pp. 235-238). IEEE.

[20] Wu, C.M., Huang, Y.F. and Lee, J., 2015. Comparisons between mongodb and ms-sql databases on the twc website. *American Journal of Software Engineering and Applications*, *4*(2), pp.35-41.