Asian Journal of Research in Computer Science

4(2): 1-10, 2019; Article no.AJRCOS.51946
ISSN: 2581-8260

Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics

Wajid Ali[1*], Muhammad Usman Shafique[2], Muhammad Arslan Majeed[1] and Ali Raza[1]

[1]Department of Computer Science, Government College University, Lahore, Pakistan.
[2]Department of Computer Science, University of Engineering and Technology, Lahore, Pakistan.

Authors' contributions

This work was carried out in collaboration among all authors. Author WA designed the study, performed the statistical analysis, wrote the protocol and wrote the first draft of the manuscript. Authors MUS, MAM and AR managed the analyses of the study. Author WA managed the literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2019/v4i230108
Editor(s):
(1) Prof. M. A. Jayaram, Director, Department of Master of Computer Applications, Siddaganga Institute of Technology, Tumakuru, India.
Reviewers:
(1) Michel Riguidel, France.
(2) Jorge Bernardino, Polytechnic of Coimbra, Portugal.
(3) Diab Zouhair, Larbi Tébessi University, Algeria.
Complete Peer review History: http://www.sdiarticle4.com/review-history/51946

Review Article

Received 05 August 2019
Accepted 09 October 2019
Published 21 October 2019

ABSTRACT

A key ingredient in the world of cloud computing is a database that can be used by a great number of users. Distributed storage mechanisms become the de-facto method for data storage used by companies for the new generation of web applications. In the world of data storage, NoSQL (usually interpreted as "not only SQL" by developers) database is a growing trend. It is said that NoSQL alternates with the most widely used relational databases for the data storage, but, as the name implies, it does not fully replace the SQL. In this paper we will discuss about SQL and NoSQL databases, comparison of traditional SQL with NoSQL databases for Big Data analytics, NoSQL data models, types of NoSQL data stores, characteristics and features of each data store, advantages and disadvantages of NoSQL and RDBMS.

Keywords: Database; NoSQL; relational databases; data stores; big data analytics.

**Corresponding author: E-mail: gcu.wajid.ali@gmail.com;*

1. INTRODUCTION

Expert says that information in the world doubles every two years. Every day, these large-scale websites such as Google, Facebook, Twitter and even YouTube generate huge amount of information. It is therefore a difficult job to efficiently store and retrieve this information. Larger quantities of transactions and results of testing involve structured storage solutions and database is the main solution to meet this need for structured storage and retrieval of information [1].

A database is a way to store information so that data can be stored and retrieved whenever needed. A relational database management system (RDBMS or just RDB) is a typical sort of database whose information is put away in tables. We already see that most databases utilized as a part of organizations nowadays are relational databases, instead of a flat file system or any other databases. Relational databases have the capacity to deal with huge numbers of information and complex inquiries. The information is put away in parts and bunches of tables, or 'relations'. These tables are isolated into lines (records) and segments (fields). However, with the explosion of data volume, SQL-based information querying has become a significant challenge in handling bigger databases [2].

A NoSQL database (originally called "non-SQL" or "non-relational") provides a mechanism for storing and retrieving model data in ways other than the table relationships used in relational databases. The primary objective of the NoSQL movement is to make data storage and recovery easy, irrespective of its structure and content. They are regarded alternatives to overcoming the constraints of the present persistence landscape dominated by SQL and are therefore also known as non-relational databases [3]. NoSQL frameworks are conveyed, non-relational databases intended for expansive scale information storage and for greatly parallel information handling over countless servers. They likewise utilize non-SQL dialects and instruments to communicate with information.

Although these two types (SQL, NoSQL) differ in many respects depending on the implementation they could be used for similar applications, although it is not recommended as one is not intended as an alternative to the other [4].

In this paper, we mainly focused on SQL, NoSQL databases, NoSQL data models with their types and comparison between these two databases (SQL, NoSQL).

This work has been structured as follows:

Section 2 is covered the NoSQL database context with Big Data analytics. Section 3 is relational databases, section 3.1 is on ACID properties of relational databases. Section 4 discussed NoSQL databases, section 4.1 showed its data model types, section 4.1.1, 4.1.2, 4.1.3, 4.1.4 examined its four data models, section 4.2 is characteristics of NoSQL databases. Section 5 showed comparison between RDBMS and NoSQL databases. Section 6 tells us performance of NoSQL and SQL databases for Big Data analytics, section 6.1, 6.2 is on benefits and drawbacks of NoSQL database, section 6.3, 6.4 is on benefits and drawbacks of RDBMS. Section 7 showed the concluding part of the work.

2. THE NOSQL DATABASE CONTEXT WITH BIG DATA ANALYTICS

From the recent trends reported in literature [5,6], it is evident that in today's context, there is an exponential growth of data volume that are structured as well as unstructured (Big Data) from a variety of data sources, such as social media, e-mails, text documents, GPS data, sensor data, surveillance data, etc. We can, therefore, say that Big Data is characterized by structured, semi-structured, and unstructured data from digital and non-digital resources. The main challenge is to effectively use the Big Data which is the data source for efficient decision-making by using appropriate technology for data mining [7,8].

Based on our literature study, we found that Big Data's present difficulties are due to the following general features of business [9]:

- **High data Velocity** – information streams from distinct sources and locations quickly and continuously updated.
- **Data Variety** – structured, semi-structured and unstructured data storage.
- **Volume of data** – a large number of datasets of various terabyte or petabyte sizes.
- **Complexity information** – data organized at various sites or information centers.

Big Data analytics, the process by which Big Data sets comprising a range of kinds of data are examined by companies. By using Big Data

Analytics, companies can analyze enormous volumes of data more precisely to reveal hidden patterns, unknown correlations, market trends, preferences for consumers and other useful business information [2,10]. Big Data analytics rely upon large volumes of data that require clusters for data stock to support timely and effective decision-making. Since relational databases do not apply to clusters and display efficiency problems related to large data analytics, companies are considering the need for the NoSQL movement [11].

The NoSQL schema is not fixed. It utilizes different interfaces for storing and analyzing the pure quantity of user-generated material, private information and spatial information generated by advanced apps, cloud computing, and intelligent devices [12,13].

In this context, the NoSQL DB provides a preferred solution than the SQL DB mainly because of its capacity to handle horizontal information partitioning, dynamic data processing, and performance improvement. Large internet companies (Facebook, LinkedIn, Amazon and Google) which are unable to manage services through the use of current relational databases have studied and led NoSQL to fix their issue with constantly growing information management, optimized information use, and horizontal large-scale information scalability. For IT systems with high efficiency and dynamics, the NoSQL database is the best choice compared to reliability and an extremely distributed character of three-tier Internet architecture systems and cloud computing [13,3,12]. Therefore, the required features in Big Data analyses need to be investigated further, compared to NoSQL and SQL as well as the significant variations in the efficiency of NoSQL data models. This article introduces these research results in the current context of Big Data.

3. RELATIONAL DATABASES

Data was originally stored in documents. However, as the quantity of information increased, accessing the information using files was not easy. It was a method that was slow and inefficient. As the quantity of information grew, keeping the information and collecting any records was very hard.

Hierarchical and network databases were intended as mechanisms for storage, but they did not provide a normal technique for data access. SQL came into being with the need to handle information and the desire for a normal technique of accessing information.

3.1 ACID Properties

When a transaction system makes any transaction then the system has to ensure that transaction will meet a certain characteristics. Following are some properties that must be fulfilled when a transaction made:

- **Atomicity:** Every transaction is atomic mean to say if one part of the system fails the entire system fails.
- **Consistency:** Every transaction is subject to a set of rules.
- **Isolation:** No transaction interferes to another transaction.
- **Durability:** If any person is committed the transaction then other person gets the same committed data.

ACID is essential, but only when the system is a kind of banking, finance, security systems, etc. that can be overhead for applications that need to share huge amounts of information like Google, Amazon etc.

For some of the following requirements, RDBMS does not quite fit:

- Distributed
- Scalability
- Control over performance characteristics
- High availability
- Low Latency
- Cheap

Hence to satisfy these needs the concept of NoSQL came into existence.

4. NOSQL DATABASES

As a technological environment transforms and faces new difficulties, companies progressively recognize that new methods and databases need to be evaluated to handle their information to help changing company needs and increasing complexity and development [13].

The Relational Database (RDBMS) was the dominant model for database administration [14]. But non-relational, cloud or "NoSQL" databases are now emerging in common as an alternative model for database management.

The primary motive behind this strategy is: simpler design, simpler "horizontal" scaling to

machine clusters, which is a issue for relational databases, and better accessibility control. The information structures used in NoSQL databases (e.g. key-value, graph, or document) are slightly different from those used in relational databases by default, making some activities in NoSQL quicker. The information structures used in NoSQL databases are also sometimes regarded as "more flexible" than in relational database tables. However, their total capabilities are still not disclosed [15].

In Big Data and real-time web applications, NoSQL databases are increasingly being used [16]. To emphasize that they can support SQL-like query languages, NoSQL systems are also sometimes called "Not only SQL".

4.1 NoSQL Database Types

Many NoSQL databases are accessible, but they fall within four data models outlined below [3,12,17]. Each category has its own particular characteristics, but the distinct information models are cross-checked. All NoSQL databases are generally designed for distribution and horizontal scaling, does not expose a SQL interface and may be open source [18].

NoSQL databases vary depending on their data model in their performance [19].

4.1.1 Document stores database

Document Stores Database relates to databases in in which information is stored in the form of documents. Document stores deliver excellent efficiency and choices for horizontal scalability. Documents within a document-oriented database are somewhat comparable to documents in relational databases, but are much more flexible because they are less schematic. The documents standard formats are like XML, PDF, JSON, and so on [20]. In relational databases, a record within the same database will have the same data fields and the unused data fields will be kept empty, but each document may have similar and dissimilar data in the case of document stores. A unique key that represents the document is used to address documents in the database. These keys can be a simple string or a URI or path string. In comparison with key value stores, document store is a little more complex because they allow the key value pairs to be embedded in documents which are also known as key document pairs.

For content management systems and blog applications, Document-oriented databases are suitable. Examples are the 10 G MongoDB, Apache CouchDB, Azure's DocumentDB and AWS DynamoDB, providers who use document orientated databases. The MongoDB is developed with a 10 G C++ and is an inter-plate-based, cross-platform document-oriented database. Grid File System is used to store large files in binary JSON format such as images and videos. It delivers high efficiency, consistency and persistence but is not very reliable and has a hungry resource [9].

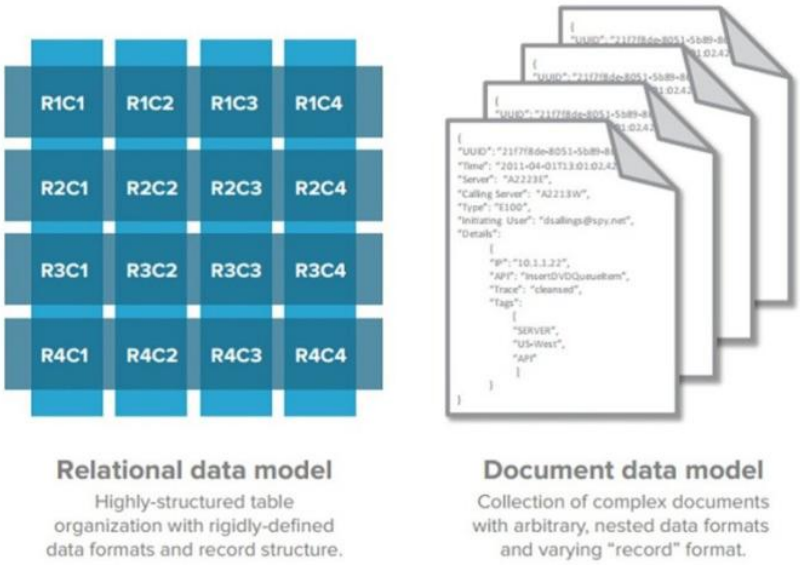

Fig. 1. Document store NoSQL database [27]

4.1.2 Key value stores database

The data stores with key-value are very simple, but they are silently effective and strong. The application program interface (API) is easy to use. The user can save the data in a schema less manner using key-value data store. The data is generally a type of programming language or object type of data. The information consists of 2 components, a string which depicts the key and the real value, producing a couple of "main value." The data saves are like hash tables in which keys are used for indexing, making them faster than RDBMS. The data model is therefore simple: a map and a dictionary that allows the user to request values based on specified key values. In modern data stores, information scalability is preferable to consistency. Therefore, ad-hoc querying and analytical characteristics such as links and aggregates were overlooked [20]. Key-value stores provide high competitiveness, quickly searching and mass storage choices. One of the weaknesses of key data store is the absence of a scheme to create a customized view of data [20].

Such key-value databases may be used as online shopping carts to create forums and websites for storing customer sessions. Amazon's DynamoDB, Cassandra, Azure Table Storage (ATS) are some remarkable examples. For internet scale apps Amazon provides DynamoDB's fully controlled NoSQL Store Service [9]. It is a distributed key value storage facility which, with its replica function, offers quick, safe, economical access to information and high availability and durability.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

Fig. 2. Key value store NoSQL database [27]

4.1.3 Graph Stores Database

Graphs database are databases that store information as graphs. The graph contains nodes and edges, which maintain the relationships between the nodes and the items. The graph also includes node-related characteristics. It utilizes an index-free adjacency method that means that each node comprises of a direct point that points towards the neighboring node. This method allows millions of documents to be accessed. The primary focus on the association between information, in a graph database [20]. Graph databases provide less effective schematic and semi-structured data storage. The queries are articulated as crossover, thus increasing the speed of graph databases over relation databases. It is simple to measure and simple to use whiteboards. Graph databases comply with ACID and promote rollback.

These data bases are designed for the development of social networking apps, bioinformatics, content management systems and cloud management services. Notable graph databases are Neo4j, Orient DB, Apache Giraph, and Titan.

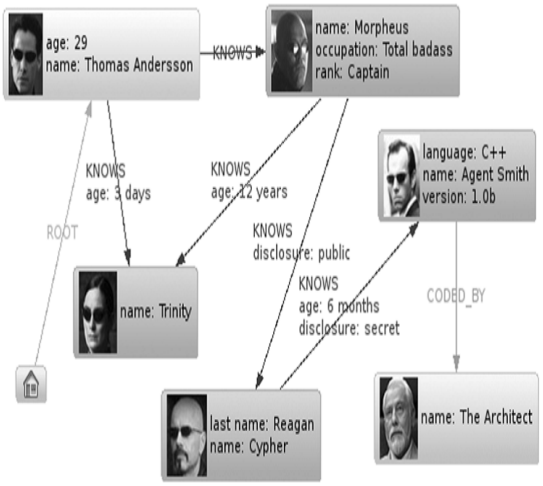

Fig. 3. Graph store NoSQL database [27]

4.1.4 Wide column stores database

The NoSQL column stores are hybrid row / column stores as opposed to pure relational bases. Although column-by-column data storage and column additions to row-based databases are shared, column stores do not store database information in lists but store the information in massively distributed architectures. Each key has one or more characteristics (rows) for each row in column stores. A column store stores its information so that less I / O activity can be

quickly added. It provides strong data storage scalability. The information saved in the database is based on the column family sort order.

Wide-column databases are perfect for data mining and Big Data analytics apps. Examples of column-oriented store suppliers include Cassandra (the high-performance of Facebook), Apache Hbase, Google's Big Table, and HyperTable. The Big Table by Google is a wide-column high-performance database, able to handle large amounts of information. It has been created using C / C++ on Google File System GFS. It is used by several Google applications such as YouTube and Gmail that have different data base latency requirements [9]. Besides the use in the Google App Engine, it is not distributed outside of Google. Big Table is conceived for simple scalability on thousands of computers, so it is hardware-tolerant.

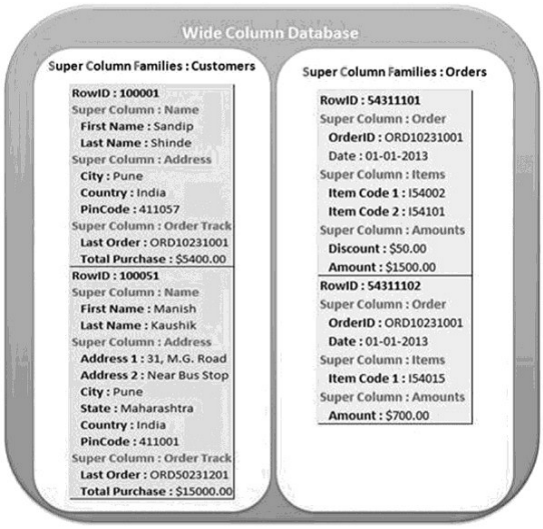

Fig. 4. Wide Column Store NoSQL Database [27]

4.2 Characteristics of NoSQL Databases

CAP Theorem – In essence, NoSQL Databases select two of the three CAP theorem principles (consistency, availability, partition tolerance). In order to achieve better availability and partitioning, many NoSQL databases have loosened the demands on consistency [21]. This resulted in BASE systems (basically available, soft-state, ultimately consistent). This implies that a compromise can be made, for example, either with low performance or offer high accessibility and low consistency with quick performance [22,21].

Scalable vertically and horizontally - Traditionally, relational databases live on a single server that can be scaled to provide scalability by adding further processors, storing and memory [21]. Replication is generally used to maintain the databases synchronized in relationship databases resident on multiple servers. NoSQL databases can be on a single server, but are designed more often to operate in a cloud of servers.

On-Disk Storage — In-Memory Dataset Relational databases live nearly always in a disk drive or storage zone network. As portion of SQL select or saved procedure activities database rows sets are placed in memory. Some NoSQL databases are built to exist in memory for speed and can continue to be stored on disk.

The NoSQL features we searched for are [23]:

- Persistence
- Replication
- High Availability
- Transactions
- Rack-locality awareness
- Economical
- Big Data Applications
- Scalable

5. COMPARISON BETWEEN RDBMS AND NOSQL DATABASES

The main reason why one would need to move to NoSQL databases is necessity in huge data storage (also called Big Data), scalability and performance reasons. Here are some tables displaying the difference in terminology and data operations between a NoSQL database and RDBMS (SQL) [1].

We have made a high-level comparison between the SQL (relational) and NoSQL (non-relational) databases on the basis of the characteristics of each database type lately reported in the literature [13,3,12,24].

Table 1. Terminologies in SQL & corresponding in MongoDB

SQL	MONGODB
Database	Database
Table	Collection
Row	document or BSON document
Column	Field
Index	Index
table joins	embedded documents and linking
primary key (specify any unique column or column combinations as primary key)	primary key (the primary key is automatically set to the _id field in MongoDB)
aggregation (e.g. by group)	aggregation pipeline

Table 2. Select query of SQL and MongoDB

SQL	MongoDB
Select * from employees	db.employees.find()

Table 3. Insert query of SQL and MongoDB

SQL	MongoDB
INSERT INTO employees VAL UES("wajid","335","BS")	db.employees.insert(name : "wa jid", Roll: "335", degree: "BS")

Table 4. Create query of SQL and MongoDB

SQL	MongoDB
Create Table employees(id int,name varchar(20),roll int)	db.createCollection("employees")

Table 5. Drop query of SQL and MongoDB

SQL	MongoDB
DROP TABLE employees	db.employees.drop()

Table 6. Summarized view of SQL and NoSQL

SQL	NoSQL
Based on ACID transactional properties such as atomicity, consistency, isolation.	Supports AID transactions and CAP theorem of distributed systems support consistency of data across all nodes of a NoSQL database.
It has vertical Scaling.	It has horizontal Scaling.
Structured Query Language are used to manipulate the data.	Query the Data efficiently. Object oriented APIs are used.
Based on pre-defined foreign keys relationships between tables in an explicit database schema. Strict definition of schemas and data type is required before inserting the data.	Dynamic database schema. Do not force schema definition in advance. Different data can be store together as required.
Softwares that use for this DB are oracle, MySQL, SQL Server.	MongoDB, Riak, Couchbase, Cassandra.

6. PERFORMANCE OF NOSQL AND SQL DATABASES FOR BIG DATA ANALYTICS

The main reason to move to NoSQL from the relational database is because of performance improvement demands. Choi et al. [13] discovered that the database NoSQL such as MongoDB offered quicker and more stable results at the cost of information coherence. Testing was performed on the basis of an open source project on an internal blog system MongoDB has found that 85% faster than a SQL databases have stored posts. NoSQL was

proposed in settings that relate to information accessibility rather than consistency [9].

The usage of MongoDB in mobile apps is described by Fotache & Cogean [25]. Some various updating operations, such as Upsert, are simpler and quicker than the SQL database with NoSQL. The use of cloud computing and NoSQL will improve the performance of mobile platforms, especially in the data layer.

In the case of a Triple Store based on Resource Description Framework (RDF) as a NoSQL database, the Ullah [26] has contrasted the results of both the relational database management scheme (RDBMS) and NoSQL. Reading a great deal of data is very comprehensive in the database and because the NoSQL database is un-structured, the storage of thousands of records is a huge amount while RDBMS uses less storage [9]. For example, the NoSQL database search for white hat took 5255 ms and only 165.43 ms to the RDBMS.

The Yahoo Cloud Serving Benchmark (YCSB) experiment was carried out by Floratou etc. [15] on RDBMS and MongoDB. They have tested the SQL client sharded database for MongoDB and client sharded databases. Tests showed that most of the benchmarks achieved higher output and reduced latency with SQL client-sharded databases. The reason for higher performance is that most read requests have been sent to pages in the buffer pool whereas the NoSQL databases tend to read shards on different nodes [9]. The research has proven that the processing power for RDBMS remains the same for NoSQL to deal with greater workloads.

6.1 NoSQL Benefits over RDBMS

- Provides a wide selection of data models
- Easily scalable
- Administrators of the database are not necessary
- Some NoSQL DB suppliers such as Riak and Cassandra can manage hardware failure
- Faster , more efficient and flexible
- Has developed very rapidly
- Used for Big data applications

6.2 NoSQL Drawbacks over RDBMS

- Immature
- No standard query language
- ACID compatibility is not possible in some NoSQL databases
- No interface specification.
- It is hard to maintain
- Less support

6.3 RDBMS Benefits over NoSQL

- It's simple to use.
- Easy to design, execute, maintain and use.
- One of the main advantages of RDBMS is that only information is saved in one place.
- It offers a variety of interfaces.
- It improves the integrity of information.
- It is secured in nature
- It has standard query language

6.4 RDBMS Drawbacks over NoSQL

- Software is costly
- Hardware overheads
- Limitations in Structure
- The lost data can hardly be recovered.
- High availability issue
- Not support Big Data applications
- Certain applications are slow in processing

7. CONCLUSION

This article contrasted SQL versus NoSQL databases and outlined the four NoSQL data models in Big Data Analytics in the context of business situations. These NoSQL data models are understandable and simple to execute and do not involve complicated methods for SQL optimization in Big Data analysis. NoSQL is a great tool to for solving data availability. In SQL database information needs to fit into the tables anyhow. If your information does not fit into the table, at that point you have to outline database structure once more. NoSQL provides schemaless data store and transactions that permit business to freely add fields to records while not the structured demand of process the schema that is a prime constraint in SQL databases. The most significant element of the movement of NoSQL databases was the many different databases accessible to developers outside legacy schemes. Now, when the information needs to determine a different storage strategy, developers do not need to rely upon the relational model. RDBMS will not go

away, definitely still necessary. However, storage requirements are very different from legacy applications for the new application generation. We conclude that NoSQL's flexible data modeling is well adapted to dynamic scalability and enhanced efficiency in Big Data analytics.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Raut AB. NOSQL database and its comparison with RDBMS. International Journal of Computational Intelligence Research. 2017;13(7):1645-1651.
2. Moniruzzaman ABM, Syed Akhter Hossain. NoSQL database: New era of databases for big data analytics-classification, characteristics and comparison. arXiv Preprint arXiv: 1307.0191; 2013.
3. Băzăr Cristina, Cosmin Sebastian Iosif. The transition from RDBMS to NoSQL. A comparative analysis of three popular non-relational solutions: Cassandra, MongoDB and couchbase. Database Systems Journal. 2014;5(2):49-59.
4. Anonymous. Introduction of MongoDB. 2018;17:8.
(Accessed 15 August 2019)
Available:http://www.mongodb.org/about/introduction
5. Mason Robert T. NoSQL databases and data modeling techniques for a document-oriented NoSQL database. In Proceedings of Informing Science and IT Education Conference (InSITE). 2015;259-268.
6. Pothuganti, Abinav. Big data analytics: Hadoop-Map reduce & NoSQL databases. International Journal of Computer Science and Information Technologies. 2015;6(1): 522-527.
7. Smolan Rick, Jennifer Erwitt. The human face of big data. Against All Odds Production; 2012.
8. Turban Efraim, Ramesh Sharda, Dursun Delen. Business intelligence and analytics: Systems for decision support. Pearson Higher Ed; 2014.
9. Venkatraman Sitalakshmi, Kiran Fahd, Samuel Kaspi, Ramanathan Venkatraman. SQL versus NoSQL movement with big data analytics. Int. J. Inform. Technol. Comput. Sci. 2016;8:59-66.
10. Ohlhorst, Frank J. Big data analytics: Turning big data into big money. John Wiley & Sons. 2012;65.
11. Kaur Pankaj Deep, Amneet Kaur, Sandeep Kaur. Performance analysis in bigdata. International Journal of Information Technology and Computer Science (IJITCS). 2015;7:55-61.
12. Prasad Abhishek, Bhavesh N. Gohil. A comparative study of NoSQL databases. International Journal of Advanced Research in Computer Science. 2014;5(5).
13. Choi Yong-Lak, Woo-Seong Jeon, Seok-Hwan Yoon. Improving database system performance by applying NoSQL. JIPS. 2014;10(3):355-364.
14. Adam Fowler. Advantages NoSQL over RDBMS. 2017;11:6.
(Accessed 22 August 2019)
Available:http://www.mongodb.org/about/introduction
15. Floratou Avrilia, Nikhil Teletia, David J. DeWitt, Jignesh M. Patel, Donghui Zhang. Can the elephants handle the NoSQL onslaught? Proceedings of the VLDB Endowment. 2012;5(12):1712-1723.
16. Anonymous. NoSQL. 2019;17:8.
(Accessed 3 September 2019)
Available:http://en.wikipedia.org/wiki/NoSQL
17. Nayak Ameya, Anil Poriya, Dikshay Poojary. Type of NOSQL databases and its comparison with relational databases. International Journal of Applied Information Systems. 2013;5(4):16-19.
18. Agrawal Rakesh, Anastasia Ailamaki, Philip A. Bernstein, Eric A. Brewer, Michael J. Carey, Surajit Chaudhuri, AnHai Doan, et al. The Claremont report on database research. ACM Sigmod Record. 2008;37(3):9-19.
19. Cassandra Planet. NoSQL databases defined and explained; 2016.
20. Nayak Ameya, Anil Poriya, Dikshay Poojary. Type of NOSQL databases and its comparison with relational databases. International Journal of Applied Information Systems. 2013;5(4):16-19.
21. Padhy Rabi Prasad, Manas Ranjan Patra, Suresh Chandra Satapathy. RDBMS to NoSQL: Reviewing some next-generation non-relational database's. International Journal of Advanced Engineering Science and Technologies. 2011;11(1):15-30.
22. MongoDB. Top 5 considerations when evaluating NoSQL databases. 2015;6:8.
(Accessed 5 September 2019)

Available:http://s3.amazonaws.com/info-mongodbcom/10gen_Top_5_NoSQL_Considerations
23. Tudorica Bogdan George, Cristian Bucur. A comparison between several NoSQL databases with comments and notes. In 2011 RoEduNet International Conference 10th Edition: Networking in Education and Research. IEEE. 2011; 1-5.
24. MongoDB. Why NoSQL? 2014;25:20. (Accessed 22 September 2019) Available:http://www.mongodb.com/nosql-explained
25. Fotache Marin, Dragos Cogean. NoSQL and SQL databases for mobile applications. case study: MongoDB versus PostgreSQL. Informatica Economica. 2013;17(2).
26. Ullah Md Arman. A digital library for plant information with performance comparison between a relational database and a NoSQL database (RDF Triple Store); 2015.
27. Moniruzzaman ABM. NoSQL database: New era of databases for big data analytics – Classification, characteristics and comparison. 2013;13:6. (Accessed 7 September 2019) Available:https://www.researchgate.net/publication/243963821_NoSQL_Database_New_Era_of_Databases_for_Big_data_Analytics_Classification_Characteristics_and_Comparison/figures?lo=1

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle4.com/review-history/51946