ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse174922020.pdf
https://doi.org/10.30534/ijatcse/2020/174922020

A Review of Critical Security Challenges in SQL-based and NoSQL Systems from 2010 to 2019

Wilson Nwankwo
Department of Computer Science, Edo University Iyamho, Nigeria.
nwankwo.wilson@edouniversity.edu.ng

ABSTRACT

Database security has become a very critical issue for organizations and agencies that deploy databases as major data stores for their operations. The ever-increasing data volumes to be stored, maintained and manipulated, the changing user and operational requirements, and the advancement in cloud platforms and hardware have contributed to the consistent change in trends around database research and development which are in many cases directed towards the engineering of innovative data models, techniques and systems that could help overcome the security challenges already established in the existing database management systems. This paper is an articulation of the critical security threats, challenges and vulnerabilities of two widely used database management systems (DBMS): the NoSQL and SQL-based DBMS respectively. The period under review is from 2010-2019, is perceived as a decade that recorded outstanding changes in data and database engineering respectively.

Key words: Database security, NoSQL, Information Security, SQL, Data Injection, DBMS

1. INTRODUCTION

Computer-based systems have become indispensable resource in every organization and central to such systems is a DBMS. Categorization of DBMS differ from context to context. DBMSs may be categorized in view of structure, models, functionality, localization, mode of access, storage, language of access, etc. Databases (DBs) are maintained by DBMS software. Among the many categories of DBMSs, relational DBMSs also called SQL DBMSs are popular. Relational DBMSs are created following the relational concept developed by IBM in 1970[1]. Interactions with relational DBs is through SQL, a procedural language [2]. Since the 70s and for almost four decades the relational DBMSs remained the popular choice in use across fortune 500 companies. Early in the millennium it became evident to application developers that relational DBMSs built purely on the concept of relational model lacked capacity to adequately handle the size, type, structure respectively as applied to data required in different application domains owing to the increasing connected distributed systems that generate volumes of unstructured, polymorphic, semi-structured data, etc. as opposed to the traditional structured data managed by relational DBMSs. This period marked the commencement of an era of big data whereby platforms such as social media networks (e.g. facebook, whatsapp, skype, etc), stock and forex exchanges, web forums, etc. generate huge streams of data in various formats. Big data are marked by the four (4) elements or factors often termed the four 'Vs' i.e. volume, variety, variability, and velocity respectively.

Volume reflects the quantity of data generated in a given period; variety reflects the heterogeneity of data generated. Big data are often aggregated data from different data sources e.g. photographs, emails, pdf documents, video, audio, etc. In a similar vein, variability reflects elements of inconsistency in the data streams, whereas the velocity reflects the frequency or speed at which data streams are generated from the various channels in the specified ICT ecosystem.

Consequent upon the need for more robust and flexible database systems, there has been consistent growth in research and industrial software development towards advancing such data management systems that can effectively cope with the growing big data platforms. Accordingly, NoSQL DBMSs have been tagged the solution to the data storage challenges facing relational DBMSs in the last decade, following which these new database technologies have continuously gained momentum.

The NoSQL DBMS was introduced around 1998 as a variant of the existing relational DBMS through Carlo Strozzi. Stronzi's DBMS was simply called Stronzi NoSQL database [3]. The term 'NoSQL' at first sight appears to imply the absence of SQL but it simply means "Not Only SQL". It is used to describe Strozzi's lightweight open source database [3,4] which though relational was not consistent with the mandatory use of the SQL as its base language. It was in 2009 that the concept of NoSQL became widely used as a class name for a category of database technologies that were not based on the traditional relational model. The main characteristic feature of NoSQL DBMSs is their departure from the use of SQL and relational data store. NoSQL DBMSs handle the processing and storing of unstructured as well as semi-structured data in a manner considered better than SQL databases [5].

Though NoSQL DBMS deployments are on the rise, majority of data-intensive companies and corporations still rely on SQL databases. It is worthy of note that the paradigm shift in database technology towards the non-relational databases have also produced some sort of 'hybridization' in the popular traditional relational DBMSs such as Oracle, MySQL Microsoft SQL Server, IBM DB2 and PostgreSQL in that they currently adopt the multi-model concept whereby the database engines accommodate features attributable to the non-relational databases. Typical examples include capabilities such as: Document store [6], Graph DBMS [7], RDF store [8], etc.

SQL-based as well NoSQL platforms have their strengths as well as weaknesses and often times, choice and deployment of a given technology may be dependent on other factors such as security, expertise, technicality, cost, scaling, existing enterprise applications/software, etc. other than just the structural/language attachments. The aim of this paper is to present a succinct review of the security challenges that confront the current implementations of SQL and NoSQL DBMSs in enterprise applications, web applications, mobile applications, IOT devices and other mission-critical data-driven systems. In this paper, emphasis is directed to inter-node communication, encryption, authentication, authorization and data consistency.

2. DATABASE SECURITY

Irrespective of the category of DBMS deployed for use in an organization, security remains a persistent concern of not just the Database Administrators (DBA) and developers but the entire organization. In many sensitive organizations such as healthcare institutions, financial institutions, government agencies, breaches and attacks on databases could be devastating. Irrecoverable losses to the tune of billions of Dollars have been recorded. Sensitive information about individuals and highly confidential or top secret government information may be stolen and used against the individual or government thereafter. There has been cases where victims of such data theft are later intimidated and ransom demanded as a price to not releasing such information to the public, news streams, and/or social media[9]. Consequently, the owners of such data stored in corporate and social media network databases are exposed to grief and untold suffering. The consistent growth in computerization is a major factor as it is evident that the decade under review has witnessed an "explosion as regards the population entrusting to technology and allied companies with their personal data" [10, 11]. Holmes [9] submitted that ten (11) out of the recorded fifteen major historical data breaches happened between 2010 and 2019. The two largest data exposures of all time happened in 2019". According to [12] there has been an upward trend in the number of data violations.

To explore the security endowments in SQL and NoSQL platforms it is pertinent to examine underlying architecture as well as how the architecture of a DBMS is tailored to meet the fundamental pillars of information security popularly called the CIA triad [13].

2.1 Popular NoSQL and SQL DBMS distributions

Table I presents the ten (10) most popular NoSQL databases deployed in production environments within the period under review. Table II presents the four (4) most popular SQL DBMSs arranged in order of popularity according to the 2019 DB-Engines rankings [14]. Among the NoSQL implementations, MongoDB, a document-oriented NoSQL database appears to top the list according to [14] though according to a benchmarking test earlier in 2013 by EndPoint[15] on Cassandra, MongoDB and HBase respectively, it was concluded that Apache Cassandra outperformed all other contenders. It is believed that MongoDB's popularity might have been as a result of its remarkable product offerings in terms of its ability to be deployed anywhere whether on site (standalone), in the cloud, etc. Among 290 DBMSs ranked including NoSQL and SQL DBMS, Oracle DBMS, MS SQLServer, and MySQL Server have remained in the top three whereas MongoDB[16] the best performing NoSQL is ranked 5th.

Table 1: Popular NoSQL platforms in 2019

S/N	NoSQL	Model	Pricing	Year released	Developer
1	MongoDB	Document-oriented	Open source	2007	MongoDB Inc.
2	Apache Cassandra	Wide-column	Open source	2008	Apache foundation
3	Redis	Key-value (in-memory)	Open source	2009	Redis Labs
4	HBase	Wide-column	Open source	2008	Apache foundation
5	Neo4j	Graph	Open source/commercial	2010	Neo4j, Inc
6	Oracle NoSQL	Key-value	Open source/Commercial	2011	Oracle
7	DynamoDB	Key-value	commercial	2012	Amazon
8	Couchbase	Document-oriented	Open source	2010	Couchbase Inc.
9	Memcached	Key-value (in-memory)	Open source	2003	Danga interactive
10	CouchDB	Document-oriented	Open source	2005	Apache foundation

Table 2: Popular SQL DBMSs

S/N	SQL DBMS	Model	Pricing	Vendor
1	Oracle	Relational	Commercial	Oracle Corporation
2	MySQL	Relational	Open source	Oracle
3	Microsoft SQL Server	Relational	Commercial	Microsoft
4	PostgreSQL	Relational	Open source	Apache foundation
5	IBM DB2	Relational	Commercial	IBM

2.2 Architecture of NoSQL and SQL DBMSs

Principle underlying the NoSQL Technology

The NoSQL DBMSs seem to conform to the Brewer's principle which is also called the CAP principle put forward by Eric Brewer [17]. According to the principle, typical distributed data stores "cannot" offer the three fundamental guarantees simultaneously but can offer two at most. The three guarantees are: *Consistency*, *Availability*, and *PartitionTolerance*. Consistency implies that data stored would remain consistent at all times. Availability reflects the absence of any downtime, whereas 'Partition Tolerance' represents the capacity of the database to remain functional even when there is a communication failure among servers in the cluster. The implication is that servers could be segregated into groups each of which does not necessarily need to communicate with the others.

Data Models

The basic distinction between the SQL DBMS platforms and the NoSQL databases is the data model adopted in each case. Unlike the SQL DBMSs where there is a uniformity in the adoption of the tabular or relational model irrespective of the nomenclature of the DBMS, NoSQL platforms differ markedly as there are currently four(4) different data models that have been implemented and more are to evolve in the near future. The models are:

- Document
- Graph
- Key-value
- Wide-column(also called column-oriented)

NoSQL databases that adopt the document model use the document structure which is akin to the popular Javascript Object Notation –JSON[18]. Other formats used are the Binary JSON (BSON)[19], XML[20,21,22], YAML[23]. A document is composed of one or more fields.

A typical structure of a document is shown in Figure 1. Document-oriented NoSQL databases operate on two kinds of relationships: referential and embedding. As shown in Figure 1, in a referential relationship two separate documents are related through a link in a hierarchical structure. A document can also be embedded into another (in Figure 1, there are actually four documents; three are independent and connected through links whereas the *Registered_course* document is embedded in the *Student* document. The four documents constitute a *Collection* in NoSQL parlance.

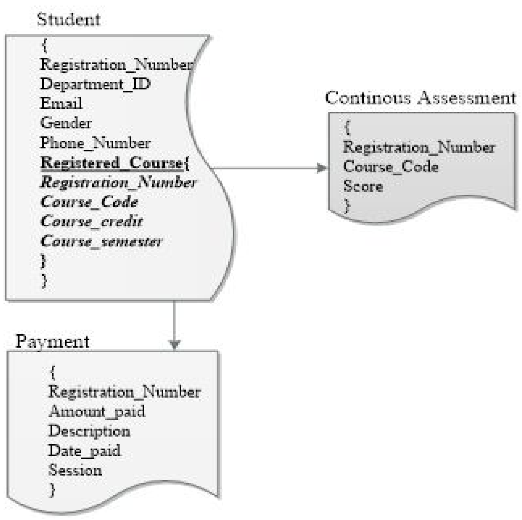

Figure 1: Document data model

Figure 2 shows an illustrative model of the Graph database. The data model comprises nodes (entities with appropriate labels) and relationships. Each node may have properties (key-value pairs akin to the columns/fields in a relational data model). Note the flexibility exhibited in a node especially the non-mandatory requirement for primary keys or joins.

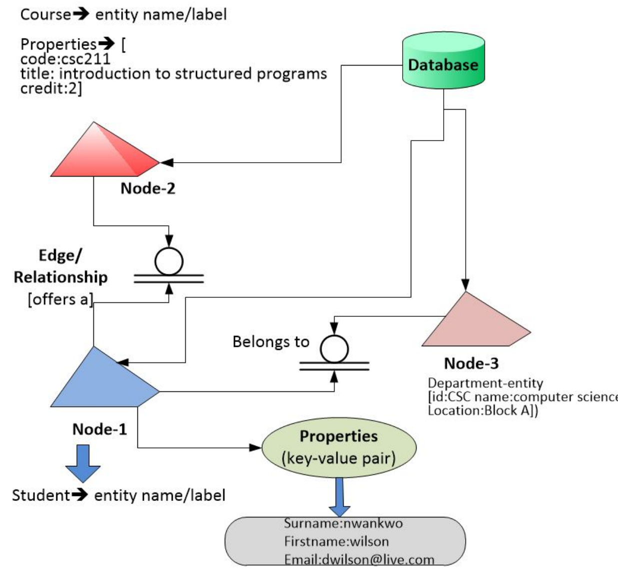

Figure 2: Graph data model

Figure 3 shows the data model for a key-value NoSQL database which is adjudged the simplest model among the NoSQL databases. It is designed to conform to map or dictionary structure. As shown in the diagram key-value model does not necessarily use keys to connect entities together hence such relationships or connections are left for the target application program to which the database is meant for to create.

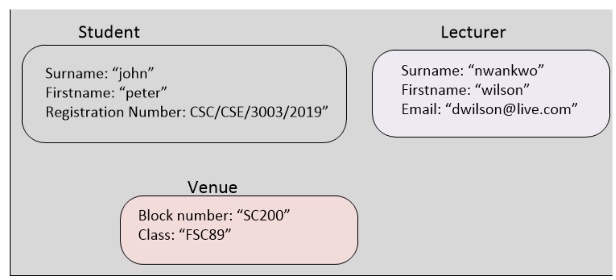

Figure 3: Key-value data model

Figure 4 shows the wide-column also called the column-oriented data model. In the wide-column model, columns are used to store data. This is similar to the fields in a relational table but differs in that related columns in a column-oriented model are grouped into a column family with each column family having many rows unlike in a relational model where data are organized in a row by row manner i.e. a row(record) is a group of related columns. Each row with its columns is associated with a row key that distinguishes it from other rows in the column family (see Figure 5). Figure 5 is a column family tagged 'lecturer' with two rows. Note the numeric row key number associated with each of the rows. Each row also has a column key which is the name of the lecturer e.g. Aliu.

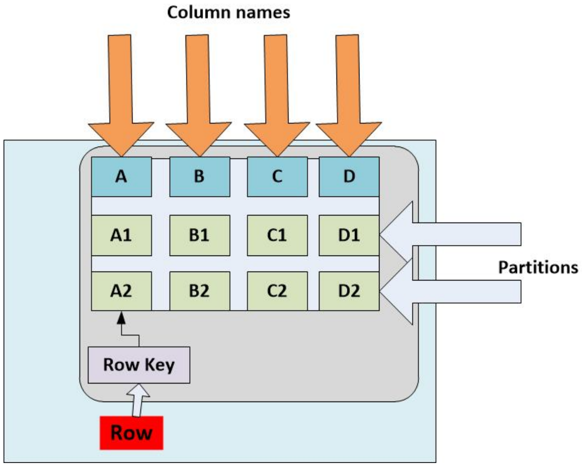

Figure 4: Wide-column data model

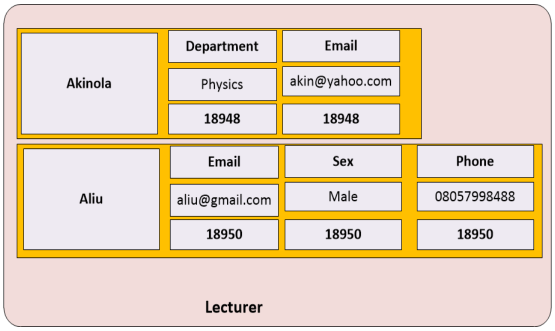

Figure 5: A Lecturer column family with two rows

Figure 6 shows the architecture of a typical RDBMS. Relational models are simply known as row-column arrangements that assume a tabular shape. Each row is a record comprising similar fields. An aggregation of the records constitute the table. Unlike the NoSQL databases where there is a flexibility of a row having different columns(akin to schema-less structure), in tables the schema would determine the column arrangement and each record must conform to the dictates of the schema. Popular RDBMSs adopt this arrangement. Thus, tables are somewhat rigid since the type of data for each column must be known before data are even stored in it. This principle promotes precision and accuracy of what is stored.

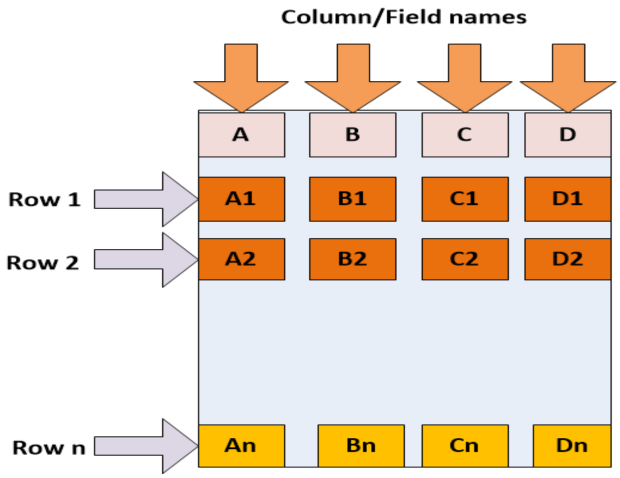

Figure 6: Relational model

2.2 The Confidentiality, Integrity and Availability (CIA) triad

The classic CIA triad of information security defines three objectives of security. CIA connotes Confidentiality, Integrity, and Availability. The understanding of how a DBMS implements each concept is important because all risks, threats and vulnerabilities against a DBMS may be a function of the DBMS's implementation of the CIA. The CIA triad is presented in Figure 7.

Confidentiality describes the prevention of unauthorized users from accessing data. It involves use of access controls, encryption, and authentication mechanisms that ensure data in a database are accessed by authorized users only. Integrity connotes the ability to prevent unauthorized modification of database data. It ensures the accuracy and completeness of data. In addition to controlling access, it further ensures that alteration of data is only done by users that are legitimately authorized to do so. On the other hand, availability reflects the DBMS's feature that provides authorized users timely and uninterrupted access to database data. Threats to availability may be malicious e.g. denial of service (DOS) or distributed DOS, or non-malicious (hardware failures, unscheduled software downtime, network failure, DBMS crash,etc.) in nature include.

Figure 6: CIA triad as applied to Databases [13]

3. SECURITY VULNERABILITIES IN NOSQL DATABASES

3.1 Weak Encryption of Data

There is a lack of strong in-built encryption in many NoSQL databases, as oftentimes, data are stored as plain text in the database. Some of existing NoSQL products do not provide encryption for data in transit. This security flaw is evident both on the client and server side. Many a time if encryption is required it has to be done at the application layer. Although there are a set of NoSQL databases that have in recent times developed means of encrypting data at rest. It should also be noted that some enterprise-level NoSQL database solutions offer encryption services. For example, MongoBD incorporates a layer of encryption using Atlas [24].

3.2 Authentication

Conventionally, NoSQL database authentication is not set by default. In MongoDB for example authentication is disabled by default, it installs with no credentials, so developers have to ensure that it runs in a trusted environment.

3.3 Authorization

Many NoSQL providers either provide basic access control systems alone or do not provide any at all. In MongoDB, for example, authorization is disabled by default. The implication is that a user has a read-only access against the database in its entirety by default [26].

3.4 Vulnerability to Query Injections

NoSQL databases also suffer from query injection attacks just like their SQL databases. Though a number of low-level techniques and protocols have been modified, the risks of injection, management of improper access control, and unsafe network exposure remain relatively high. Injection attacks are possible because a number of NoSQL databases are equipped with either an embedded RESTful API or load one using a third party extension that makes use of JSON or XML formats.
This vulnerability can allow an attacker execute malicious code via the requests payload [26].

3.5 Redundancy and Enforcement of Business logic

Unlike SQL DBMSs that provide for redundancy wherein any failure or corruption in a table may be quickly identified and possibly remedied, NoSQL databases on the other hand do not provide such features. Another gap with the major NoSQL DBMS such as HBase, Apache Cassandra, MongoDB, CouchDB, etc. is their lack of support for business logic enforcement.

3.6 Data precision and quantification

As opposed to relational DBMSs where data are thoroughly organized and structured to ensure precision and accuracy wherein the very quantity and class of data are known. Versatility and speed appear to take priority over accuracy and precision in NoSQL databases. Consequent upon the foregoing, transactional applications requiring high precision such as finance and banking operations, healthcare, etc may be susceptible to inconsistencies if driven by the existing NoSQL databases. It is possible that future developments in NoSQL technologies would address this gap.

4. SECURITY VULNERABILITIES IN SQL DATABASES

4.1 SQL Injection Attack

In a typical attack, the attacker inserts unauthorized database queries or expressions into the vulnerable data channel. The common data channels that attackers target are input parameters on the web band stored procedures. Consequent upon injection, the statements would be executed. An attacker could gain access to an entire database using SQL injections [27].

4.2 Human Enabled Vulnerabilities in Databases

In NoSQL and/or SQL-based DBs the commonest reason for data breach is the human factor. In situations where the database provider has put mechanism for securing the database, the organizations making use of the database solution needs to have person(s) with the technical know how to properly setup, configure and use those security features. Some possible human enabled vulnerabilities are discussed below:

4.3 Misconfiguration

The configuration options that are available in SQL-based and NoSQL DBs can be set in a manner that compromises the security of a database [27]. The configuration options include the database parameters and utmost caution has to be observed when setting these options. Certain attacks may be enhanced by incorrect configurations. Exploitation of misconfiguration can give an attacker access to encrypted files following which he may reset default configuration [28].

4.4 Substandard key management

Encrypting sensitive data is very important but equally important is the management of the encryption keys. If keys are inappropriately managed the system is unwittingly made vulnerable to attacks. According to [29] absence of key management is analogous to having the best lock and keeping the key of the lock under a doormat.

4.5 Lack of Segregation

This refers to the absence of differentiation as to functions or roles. Segregation connotes the specification and separation of roles; it specifies actions that may be executed by each role. By separating the powers and duties of administrators and users, it would become somewhat difficult to compromise the system where a breach occurs. This factor also contributes towards ensuring that even the internal staff are prevented from embarking on acts that may amount to database fraud or data theft. Thus, the creation of distinct roles and rights for administrators and users alike may help avert fraud, corruption, theft, etc.

5. CONCLUSION

Though NoSQL DBMSs are becoming increasingly popular and organizations are adopting them it may be submitted that it is not yet rife to depend wholly on NoSQL technologies as they are still evolving. At the moment, majority of large business organizations still depend on the SQL DBMSs such as Oracle, Microsoft SQL Server, etc.
Brown [30] notes that the NoSQL buzz has not deterred the popularity of the relational systems. According to [30], six of the topmost DBMSs in use over the period discussed are relational. The author observed that there is an ongoing 'revolutionization', re-evolving or extension of the existing industry-strength relational DBMSs through the introduction of new functionalstreams (multi-model) by the vendors, to support features that distinguishNoSQL databases. For instance, the popular SQL platforms: Oracle, MS SQLServer, PostgeSQL, MySQL currently support document stores, RDF stores, etc.
To showcase its preparedness to remain a major market share owner, Oracle also introduced its NoSQL database to compete with other NoSQL platforms. A major feature of this NoSQL database is its ease of integration with not only relational DBMSs but with other technologies. Oracle has also adopted the distributed and cloud approaches to ensure support for distributed and big data operations. It is on this note that it may be safe to conclude that the NoSQL platforms do not present a 'one cap' fits it all yet [31] and presently it cannot be adopted as a direct replacement of relational DBMSs owing to challenges including security. The foregoing argument isjustified considering thatin the over 46 billion US Dollar database market, NoSQL DBMSs, at the moment, do not account for more than 3% share of the market notwithstanding its growing campaign and acceptance in the enterprise and non-enterprise sectors. Nevertheless, NoSQL DBMSs are a potential market disruptor in the database technology sector [32-36].

In addition to handling heavy transactional workloads in companies and organizations, SQL DBMSs in general, provide more security features than their NoSQL counterparts, this may be attributed to the fact that NoSQL DBMSs are still relatively new compared to SQL DBMSs.

In the future, following the sporadic advancements in Cloud and IoT deployments [37-38] coupled with data processing requirements, it is believed that the security model of NoSQL platforms would advance to such a level that it can substantially compete with or outweigh the best security provisions in the best commercial SQL DBMSs available. In each case, it may be submitted that the greatest threat to database base security is the human factor.

REFERENCES

1. A. Bjorklund, **NoSQL databases for Software Project data**, 2011
2. R. Shabana, S.B. Imran, K.A. Rafaqut, **Challenges in NoSQL-Based Distributed Data Storage: A Systematic Literature Review**, *Electronics*, Vol. 8, No. 5, pp. 488, 2014. doi:10.3390/electronics8050488
3. A. Lith, and J. Mattson, **Investigating storage solutions for large data: A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data**, Master’s thesis, Chalmers University of Technology G¨oteborg, Sweden, 2010
4. P.J. Sadalage and M. Fowler, **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**, Addison-Wesley, New York, 2012
5. A. Holmes, **Hackers have become so sophisticated that nearly 4 billion records have been stolen from people in the last decade alone. Here are the 10 biggest data breaches of the 2010s**, 2019. [Online].Available at https://www.pulse.com.gh/bi/tech/hackers-have-become-so-sophisticated-that-nearly-4-billion- records-have-been-stolen/mvq0xfn
6. K.D. Foote, **Fundamentals of Document Databases,** 2019. Available at https://www.dataversity.net/fundamentals-of-document-databases/
7. P. Madan and A. Saxena, **Review: Graph Databases**, *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.4, No.5, 2014
8. P .Karvinen, N. Díaz-Rodríguez, S. Grönroos, and J. Lilius, **RDF Stores for Enhanced Living Environments: An Overview**, in: I Ganchev., N Garcia, C Dobre, C Mavromoustakis, R Goleva (eds) Enhanced Living Environments. Lecture Notes in Computer Science, 11369, Springer, 2019 https://doi.org/10.1007/978-3-030-10752-9_2
9. W. Nwankwo, K.C. Ukaoha, **Socio-Technical perspectives on Cybersecurity: Nigeria’s Cybercrime Legislation in Review**, *International Journal of Scientific and Technology Research*, Vol. 8, Issue 9, pp.47-58, 2019
10. P.L. Mabry, **Making sense of the data explosion: the promise of systems science**, *American Journal of preventive medicine*, Vol. 40, Issue 5, Supplement 2, pp. 159-161, May, 2011 https://doi.org/10.1016/j.amepre.2011.02.001
11. M.G. Solomon &Chapple, ***Information Security Illuminated***, Jones and Bartlett, Sudbury, Massachusetts, 2005
12. O.A .Jeremiah, O.A. Festus, **Comparative Security Vulnerability Analysis of NoSQL and SQL Database Using MongoDB and MariaDB**, *International Journal of Trends and Technology*, Vol. 67, Issue 10, 2019, doi:10.14445/22312803/IJCTT-V67I10P104
13. W. Nwankwo, ***Fundamentals of Modern Database Systems*** vol. 1, Lambert Academic Publishing, Mauritius: 2018.
14. S. Yegulalp, **Oracle remains most popular database, but MongoDB continues to rise**, InfoWorld, 2016. Available at https://www.infoworld.com/article/3019332/oracle-remains-most-popular-database-but-mongodb-continues-to-rise.html
15. Endpoint Corporation, **Benchmarking Top NoSQL Databases Apache Cassandra, Couchbase, HBase, and MongoDB**,2019. Available at https://www.datastax.com/sites/default/files/content/report/files/2019- 09/NoSQL_Benchmarks_EndPoint.pdf
16. A. Makris, K. Tserpes, G. Spiliopoulos, & D. Anagnostopoulos, **Performance Evaluation of MongoDB and PostgreSQL for Spatio- temporal Data**,*Zenodo*. 2019, DOI:10.5281/zenodo.2649876
17. K.D. Foote, **A Brief History of Non-Relational Databases**, 2019. Available at https://www.dataversity.net/a-brief-history-of-non-relational-databases/
18. MongoDB, **Top 5 Considerations When Evaluating NoSQL Databases**, 2019. [Online**].** Available at www.mongodb.com/collateral/top-5-considerations-when-evaluating-nosql-databases
19. bsonpec.org,**Specification Version 1.1**. [Online].Available at from http://bsonspec.org/spec.html
20. S.J. Pramod, ***Nosql distilled: A brief guide to the emerging world of polyglot persistence***,1st edition. Addison-Wesley Professional, Boston.
21. W. Nwankwo, B.S. Olanrewaju, and T.C. Olayinka, **A Document Interchange Framework for Tertiary Institutions**, *World Journal of Engineering Research and Technology*, vol.3, issue 5, 2017
22. W. Nwankwo, B.S. Olanrewaju, T.C. Olayinka, **Document Interchange Modeling with the Zachman Framework: Nigeria’s Institutions in Perspective**, *World Journal of Engineering Research and Technology*, vol. 4, issue 2, pp. 52-63, 2018
23. E. Goebelbecker, **YAML Tutorial: Everything You Need to Get Started in Minutes**. Available at https://rollout.io/blog/yaml- tutorial-everything-you-need-get-started/
24. D. Kirkpatrick, **Mongodb - Security Weaknesses in a typical NoSQL database**. Available from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/mongodb- security-weaknesses-in-a-typical-nosql-database/
25. T. Karavasilev and E. Somova, **Overcoming Security issues of NoSQL databases**, 2019. [Online]. Available at www.researchgate.net/publication/325229 65Overcoming_the_Security_Issues_of_NoSQL_ Databases
26. S. Singh & R.K. Rai, **A Review Report on Security Threats on Database**, *International Journal of Computer Science and Information Technologies*, Vol. 5, Issue 3, pp.3215 – 3219, 2014

27. B. Fataniya, **A Survey of Database Security Challenges, Issues and Solution**, *International Journal of Advance Research and Innovative Ideas in Education*, Vol .3,Issue 5, pp. 1022-1032, 2017
28. C.S. Ahmed, & Saeed ur Rehman Malik, **Security of an Enterprise Level RDBMS - Threats and Challenges**, *International Journal of Scientific & Technology Research*, Vol. 7, Issue 1, pp.113-116, 2018
29. Y. Guez, **6 encryption mistakes that lead to data breaches**, 2013. [Online]. Available at https://www.crypteron.com/blog/the-real-problem- with-encryption/
30. M.S. Brown, **Get The Basics On NoSQL Databases: Key Value Databases**, 2018. Available at www.forbes.com/sites/metabrown/2018/0 3/31/get-the-basics-on-nosql-databases-key-value-databases/#16b08f871aeb
31. C. Tozzi, **The Limitations of NoSQL Database Storage: Why NoSQL's Not Perfect**. 2017. [Online] Available at https://www.channelfutures.com/cloud- 2/the-limitations-of-nosql-database-storage-why- nosqls-not-perfect
32. S. Navelkar, **NoSQL: The future of data economy,** 2016. [Online].
33. J Ader, B Fitzsimmons, S Naji, **Database Software Market: The Long-Awaited Shake- up**, 2019. [Online]. Available at https://blocksandfiles.com/wp-content/uploads/2019/03/Database-Software-Market-White-Paper.pdf
34. J. Loo, **Fundamental Bull Case For MongoDB: 216%Upside,** 2019. [Online]. Available at https://seekingalpha.com/article/4287973-fundamental-bull-case-for-mongodb-216-upside
35. S. Mukherjee, **The battle between NoSQL Databases and RDBMS**, *International Journal of Innovative Research in Science, Engineering and Technology*, 2019. DOI:10.15680/IJIRSET.2019.0805107
36. S. Šuman, P. Poščić, M. G. Marković, **Big Data Management Challenges**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 1, pp.717-721, 2020 https://doi.org/10.30534/ijatcse/2020/102912020
37. S. Karimunnisa, V. Sri Kompalli, **Cloud Computing: Review on Recent Research Progress and Issues,** *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 8, No.2, pp. 216-220, 2019 https://doi.org/10.30534/ijatcse/2019/18822019
38. I.A. Atoum, N.A. Al-Jarallah,**Big Data Analytics for Value-Based Care: Challenges and Opportunities**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 8, No.6, 2019 https://doi.org/10.30534/ijatcse/2019/55862019