Relational and NoSQL Databases: The Appropriate Database Model Choice

Mohammad A. Hassan
Computer Science Department
Zarqa University
Zarqa- 13132 – Jordan
mohdzita@zu.edu.jo

2021 22nd International Arab Conference on Information Technology (ACIT) | 978-1-6654-1995-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ACIT53391.2021.9677042

***Abstract*— For over four decades, Relational database management systems RDBMS have been the primary model for data storage, retrieval and management. However, due to the continuous information growth in current organizations and the increasing needs for scalability and performance, specially while handling a very huge amount of data that generated by various new generation real time applications or social networking sites that could be unstructured or semi-structured data, poses a set of challenges to the existing RDBMS Vendors. Such challenges have created a need for adaptation alternative technologies in the field of data storage and manipulation. NoSQL technology is the alternative category of Database Management Systems that have been emerged as the solution to the ever-growing data requirements. In this paper, the advantages and the limitations of relational databases we will be presented. The NoSQL data model, types of NOSQL data stores, characteristics of each data store, advantages and disadvantages of NoSQL over RDBMS will also be discussed. The paper helps the interest users to take a review of the different database model solutions, which can serve as a base for selecting the proper database model that can satisfy their application requirements.**

Keywords*— *RDBMS, Relational Databases, SQL, NoSQL Databases, Data Stores

I. INTRODUCTION

Database model is a logical structure that describing how a database is structured and used. Several models have been proposed such as network, hierarchical, relational and Object-Oriented. Among these models, the relational database or RDBMS is the dominant one for database management since it was introduced by Codd in 1970 [1]. In common practice, it is considered the most important and central platform for most of the application to store, manipulate and retrieve data and it has been the main data management and query solution for the Information Technology industry for over four decades now.

In today's age, many applications produce a huge amount of data which might be complex, unstructured or semi-structured [2]. The massive amount of data produced by various real time applications, social media sites are impossible to get handled by the existing traditional databases such as RDBMS, which works only on structured data [3]. Another problem with relational model is that it has some scalability issues that is performance degrades rapidly as data volumes increases [4]. Such issues poses a numerous of challenges to the current relational databases vendors. The RDBMS are not so adequate to support some of the essential requirement of these modern applications such as handling large amount of unstructured data or providing flexible scalability.

To overcome these problems, a new type of non-relational databases called NoSQL has emerged. NoSQL databases try to meet the challenges and the requirements of the new class of applications. Handling complex and unstructured data efficiently while offering easy scalability are the primary advantages of NoSQL Databases [5].

The remaining of this paper is organized as follows: Section 2 presents a general overview of the traditional Databases and Database Management System (DBMS). Section 3 discussed NoSQL databases, including its characteristics, data model types, advantage and disadvantage. Section 4 highlights on the factors that should be considered while choosing the appropriate Database model. Finally, we conclude this paper in Section 5.

II. DATABASES AND DATABASE MANAGEMENT SYSTEM

A database is a collection of data and information that stored and set up in an organized style for easy access, management and updating. Data in the database could be added, deleted or modified. Database systems work by querying the information or the existing data, and then executing relevant applications against it [6]. DBMS is system software for creating and managing databases including generating reports, controlling read and write operations, as well conducting an analysis of usage. The DBMS acts as an interface between the end users and database in order to facilitate the organization and manipulation of data. The main functions of the DBMS is to manage the data, the database schema, which defines the logical structure of the database and the database engine that allows data to be obtained, updated and locked. These three essential elements support provide standardized administration procedures, concurrency, recovery, security, and data integrity [7].

A. Relational Databases

Relational Database Management System (RDBMS) is an advanced version of DBMS. It is based on the Relational model - relational algebra and tuple relational calculus- introduced by E.F.Codd. Relational databases are highly structured with a table based i.e. all the data in the table are stored as rows and columns which is often referred to as tuples and attributes. Each row contains a value of data for the corresponding column while each column has a data type which is mostly normalized. Constraints are also applied to this data. The tables are related to one another other with primary and foreign keys thus providing "Referential Integrity" among the objects and because of this these tables are also called relations. Many popular databases currently in use are based on the relational model. The major RDBMS vendors are Microsoft SQL Server, DB2, Oracle, MYSQL, etc. [8]. The main features of RDBMS systems are:

- Enforce all data to be stored in the tables that provided by an RDMBS in the form of columns and rows.
- Provide many facilities, such as primary key, that help in unique identification of the rows, and using a

common column that could be shared among more than one table.
- Index could be created to speed up data retrieval.
- Creation of virtual tables that simplify queries a long with multi-user accessibility that enabled to be controlled by individual users.

These features make RDBMS widely used across the world.

B. Structured Query Language

SQL stands for Structured Query Language which is a standard programming language for managing data held in a relational database management system. SQL was initially developed in the 70s, it is considered the most systematic tool that used by database administrators. SQL is usually used for analytical queries and writing scripts for data integration by database developers and users [9]. It includes the ability to insert, query, update, and delete data, as well as manage data schema (database structures) and control access to data in the database. Operations and queries in SQL are usually in command forms, written as statements [10]. In early 1980s, SQL became the standard programming language for relational databases [11], SQL databases and relational database systems are became refer to the same database model.

C. Relational Databases Advantages

RDBMS has been the major database solution since the ’70s and only recently new models find opportunity in the database world. In spite of the all the criticism the relational model has, its significant advantages and its great futures are still good enough for a wide range of tasks. Some of the main advantages of RDBMS are as follows [12, 13]:

- Flexibility, simplicity, ease of data retrieval and data integrity.
- Abstraction, multiuser access, automatic optimization for searching,
- A single uniform data definition language (DDL) is used for different roles (users, developer, DBA).
- A single standardized language is used for different RDBMS.
- It supports ACID constrains (atomicity, consistency, isolation, durability), each of these four qualities guaranteeing stability, security, and contribute to the ability of a transaction to ensure data integrity.

Advantages of relational database comparing to other types of databases could be summarized in the following properties. A relational database is one of the simplest existing database models, both to implement and design, and its major benefits come from its comfort, uniformity, and ease of use [14]. Even with a very limited knowledge of SQL, naive users can easily use it with the help of special tools like an online SQL Query Builder. A relational database supports powerfully in maintaining the data accuracy, data integrity, data flexibility, reducing data redundancy, and facilitates to implement security methods easily. In addition, RDBMS are optimized for writes, consistency and availability. In spite of all of these advantages, RDBMS have many problems and challenges.

D. Challenges with Relational Databases

In spite of all recognized RDBMS or SQL advantages, they have many problem and challenges that, make them highly inappropriate for certain types of projects. One of the main problems with SQL is that it is very scale as much as a database grows larger. Some of the primary problems of RDBMS are as follows [8, 12, 15]:

- The main problem with Relational databases is that they do not support high scalability, such databases are not capable of handling exponential growth of data.
- Setting up and maintaining relational databases is expensive. Also some features of relational databases are not used and hence result in higher cost as well as complexity.
- Field lengths have limits, which make it difficult to store a large amount of information in one field.
- Since data is stored in the form of tables, data cannot be easily encapsulated.
- In case of complex relational database systems it becomes difficult to share information from one database to another.
- SQL cannot handle unstructured data such as documents, e-mail, multimedia and social media efficiently.
- Multiple databases can easily become islands of information and difficult to connect the databases where they can talk to each other.
- Working with a shared relational database could result difficulties regarding performance issues and data integrity. It is hard to predict the performance characteristics of database because each application accesses the database in its own exceptional way. In the meanwhile, it is hard to ensure the data integrity because no single application has control over the data while applications may operating under different business principles.

Big companies like Amazon, Google, Facebook and LinkedIn have been revealed those limitations of the relational database in addition to the increasing demands of new applications. These shortcoming and limitations of relational database model have led to the evolution of non-relational databases, which also commonly known as NoSQL (Not Only SQL) [12]. Handling web-scale systems data, Big Data systems particularly have led to developing many NoSQL databases [4].

E. Structured vs. Unstructured Data

Structured data is data that consist of clearly described types of data and has been predefined and formatted to a set structure before being placed in data storage. Structured data are highly organized, easily analyzed and have a pattern that makes the search process easy, like in text. Inventory systems, Payroll, point of sale systems, airline reservations are all forms of structured systems since they are using structured data- the data which is stored and displayed as a set of rows, columns and tables.

Conversely, unstructured data is the type of data which usually cannot be searched easily, such data are raw and unorganized and have little or no predefined structured or form. It comes in many of unstructured data format, including, emails, social media posts, chats, presentations, spreadsheets, word processing or PDF files, digital image files as well as video and audio [16].

Relational databases are usually referred for structured data and the remaining other types usually go for the unstructured data. Examination and analysis the data of structured format is an established course with proven technology. On the other hand, examination and analysis of unstructured data is an evolving industry in its initial stages and is not considered a stabilized technology. Researchers reported that "80% of business-relevant information originates in unstructured form, primarily text." [17].

III. NoSQL Databases

Non-relational databases or NoSQL Databases are new addition to database design that do not store data in relational format and usually do not use SQL as their query language. NoSQL can handle a wide variety of data models that include objects, graph, columnar, document, and many other formats. "Not Only SQL", abbreviated to NoSQL, could handle unstructured and semi-structured data efficiently unlike the relational databases. Recently, NoSQL could be considered as a promising alternative to the traditional relational database approach. Tables in traditional relational model are used to store data with a specific predefined schema design, then data can be inserted to these tables. When working with large amount of data in distributed environments, NoSQL database is the best platform [6]. Several databases that preceded the RDMS are also to be considered as NoSQL, but this term usually refers to the modern database models that developed in the earlier part of the 21st century to handle web-scale systems data and cloud applications [11].

A. WHAT IS NoSQL?

NoSQL are the new generation of databases that mainly addressing the following properties: Being non-relational, open-source, distributed and horizontally scalable [18]. The aim of developing NoSQL is to handle modern web-scale databases and cloud applications, which began in early 2009 and is growing rapidly. Unlike relational databases, NoSQL databases do not use (rows-columns-tables) format as storage structures, and do not use the RDBMS principles, and have flexible schema approach to data management. No schema definition is required in NoSQL databases before inserting data nor changing the schema when data collection and management need evolve, and mostly they do not use SQL as their query language [19]. These databases can handle various types of data models like documents, graph, objects and many other formats. Unlike the relational databases, these databases can handle and retrieve semi-structured and unstructured data very efficiently as well as structured data.

B. The Features of NoSQL databases

Comparing to traditional relational database management systems, NoSQL databases use a different approach to store and access data. NoSQL databases do not ensure the ACID properties instead they increased availability and scalability [4]. The main features of NoSQL databases are as follows [18, 20, 21]:

- *Non-relational:* NoSQL databases do not use RDBMS principles, neither does support SQL as their query language. Furthermore, NoSQL databases do not support join operations to get advanced data unlike RDBs, instead they stored related data together to improve the speed of data access.
- *Distributed:* NoSQL databases is usually store their data in different servers, while the metadata manage the location of these stored data.
- *Open-source:* Most NoSQL databases are open source -free to download-, unlike most RDBS.
- *Horizontally scalable:* Ability to scale horizontally by increasing or decreasing multiple servers to meet the data processing capacity of NoSQL database, which leads to more performance.
- *Schema-free:* All data stored in NoSQL databases without predefined fixed Schema, unlike RDBS that need to define database schema before inserting data. Therefore, NoSQL databases can flexibly add data.
- *Replication support:* Most of the databases replication architectures are supported by NoSQL databases, which leads to high availability.
- *Simple API:* Most NoSQL databases provides an easy and simple APIs for data collection and network delivery and many other facilities that make programming easier.
- *BASE properties:* NoSQL databases do not follow the traditional relational ACID constrains, instead they follow the BASE properties, which is an abbreviation for "basically available, soft-state, and eventual consistency". The meanings are as follows:
  - **B**asically **A**vailable**:** Which means an application should work basically all the time.
  - **S**oft state**:** An application need not to be consistent all the time.
  - **E**ventual consistency**:** An application should be in some known state eventually.

NoSQL databases lose the support for ACID transactions as a trade-off for increased availability and scalability. The BASE model trade of the essential ACID properties of consistency and isolation with "availability, graceful degradation, and performance [22].

C. Types of NoSQL databases

According to the statistics, several different varieties (hundreds) of NoSQL databases have been created to support specific needs and each have different use case scenarios [23]. There are different approaches for classifying NoSQL databases into different categories and subcategories. However, most commonly used classification can be briefly described as follows [4, 8, 15, 20, 22, 24, 25]:

Key–Value Store Databases: Key-value database is combination of two main attributes: key and value that is key corresponds to a value or group of values. It is based on hash table where key is a unique identifier and points to value. It is one of the simplest databases among all of them, but are quiet efficient and powerful model. It has a simple application programming interface (API) and allows data to be stored in a schema-less manner. The data can be any primitive type or it can be an object, and the stores can hold structured as well as unstructured data. Such database support scalability - can handle large amount of data- over consistency. High concurrency, better performance, high query processing speed and options for mass storage are provided by key-value stores.

Key value data stores can be used in forums and websites for online shopping. It could be used also in states where the user's shopping cart or the user's session need to be stored for getting details like favourite products. There are many key value stores available in the market, the most popular ones are: Berkeley DB, Tokyo Tyrant, Redis, Riak, Voldemort, Memcached and Amazon's Dynamo.

Document Store Databases: Document Store Databases that also known as 'Document Oriented Databases' refers to databases that store their data in the form of documents which consist of two main attributes: key and document. These documents are somehow similar to the records in relational databases but more flexible and schemaless, the structure of documents does not need to formally specified by the user before adding documents to a collection. It is usually the application programs that verify rules about the structure of a document. The commonly format used to define documents are XML, PDF or JSON, and are addressed using a unique key which is a simple string or a URI or path. JavaScript Object Notation (JSON), is used in document store database with dynamic schema, which means different documents can have different number of fields that can be added by using unique key. These data stores are slightly more complex than the key value data stores but they offer great performance.

Document databases usually used for applications in which data need not be stored in a table with uniform structure, but instead each document could have the same or a different characteristics. Document stores serve well when the domain model can be split and partitioned across some documents. Popular fields in the document can be indexed to provide fast retrieval without knowing the key. They can be used for content management system, blog software etc. If the database have a lot of relations and normalization, document stores should be avoided. MongoDB and CouchDB Amazon DynamoDB, MongoDB Atlas and Google Cloud Firestore are the most popular document store databases.

Column Family Stores Databases: Column stores are non-relational databases and as the name suggests they store the data in column oriented tables unlike the relational databases that store the data in row oriented style. These databases store data in column families as rows that have many columns associated with a row key. A row key used to identify a specific record with a unique value and play the same role of the primary key of a relation in RDBS. Use of these row keys makes the data retrieval quicker.

Some of the main advantages of such databases are: they provides high flexibility and scalability in data storage, they are faster in read while querying, compression also get improved as all data in one file is of the same data type and can easily add new columns at any time by just creating a new file instead of rebuilding table as in relational databases. The ideal data storage methods of column stores databases make it suitable for data mining and analytic applications. Examples of the column store databases include Google's Bigtable, Cassandra and HBase.

Graph Databases: Graph databases are based on the graph theory concept which store data in form of graphs in a schema-less manner. Graph database is a collection of nodes and edges, where nodes represent entities or objects and edges represent relationship between the nodes. The graph also contain information about the properties of related to nodes. Graph databases provides an efficient data storage specially for semi-structured data. Expressing queries as traversals in graph databases making it faster than relational databases. Graph databases are follow ACID constrains and offer rollback support that guarantees the consistency of information. Graph databases can be used for a variety of applications like social networking applications, bioinformatics, security and access control, network and cloud management. There are many graph databases, but Neo4j, FlockDB, InfroGrid and IMS are the most common used databases.

D. Other NoSQL Classification

The NoSQL databases could be categorized into several other categories mentioned earlier which considered less popular. According to the official website of NoSQL database [18], such categorizes can be described as follows:

Multi-model Databases: The data format of this category of NoSQL databases contains some combination of the four types described above and therefore can support a wider range of applications. Examples of Multi-model databases include OrientDB and ArangoDB.

Object Oriented Databases: This category a combines the functions of traditional databases and object oriented programming in which data is stored in the form of objects. Such databases are commonly used in scientific research, computer aided design. Examples of object oriented database are Db4o and Versant.

Grid and Cloud Database Solutions: This category stores recent access data in random access memory and uses grid computing to speed up the time of access data from a database. According to the statistics. Examples of grid and cloud database solutions are Hazelcast and Oracle Coherence.

XML Databases: XML Databases allow data to be stored in XML format. XML databases are often linked to document-oriented databases. The data stored in an XML database can be queried, exported and serialized into any format needed. This type of database is suited for businesses with data in XML format. Examples of XML databases are Oracle Berkeley DB and BaseX.

Multi-dimensional Databases: The data in this category of NoSQL databases is stored in a multi-dimensional array in order to analyse the value of each array element. This type of database is optimized for data warehouses and online analytical processing applications. The main role of such databases is to rapidly process the data in the database in order to generate the answers quickly. Examples of Multi-dimensional databases are Intersystems cache and GT.M

Multi-value Databases: This category allows a user to assign a list of values to an attribute unlike a typical database, which is designed to hold only one value per attribute. A Multi-value database is a flexible database that features a mix of different database categories, while it is ability to store multi-dimensional data makes this model so powerful. Examples of Multi-value databases are jBASE and Model 204.

Event Sourcing: This category of NoSQL databases ensures that all changes to application state are stored as a sequence of events. Event Sourcing allows tracking all state changes that happened within an application, which is often extremely important in certain domains. Event store is the most widely discussed one of the event sourcing databases.

Time Series Databases: Databases in this category is designed to store, manipulate and retrieve data records that are part of a "time series". Time series data is often a continuous flow of data like measurements from sensors and intraday stock prices. A time-series database allow users to store large volumes of data in a format that allows fast insertion and fast retrieval to support complex analysis on that data. Examples of Time Series Databases are Informix Time Series Solution and influxdata.

According to the classification of the NoSQL database official website, more NoSQL databases categories have been described, but without giving explanation for the characteristics of these categories. Examples of these categories: Scientific and Specialized Databases, Other NoSQL Related Databases and Unresolved and Uncategorized databases.

E. Challenges with NoSQL Databases

NoSQL Database model gained popularity for its unique characteristics. In the meanwhile, NoSQL databases have many challenges that should be considered. Some of these challenges are listed below [8, 26, 27]:

- These databases do no support ACID properties. The high availability and the performance benefits at the cost of consistency. That means NoSQL databases are less reliable than the relational databases since they compromise reliability for performance.
- There is no uniformity and No specific programming interface among the different NoSQL databases types. Such databases use different query languages than standard relational, that impose an extra overhead on users to learn new query languages.
- Many of NoSQL databases are considered immature, difficult in maintenance and not good at spreading the database across multiple nodes.
- In many NoSQL databases, security is lower than relational databases, while the lack of encryption for data files and the weak authentication system in these databases are major concern.

Even though NoSQL databases are in their initial stages, they increasingly considered a practical alternative to relational databases especially for interactive web and mobile applications. The simple characteristics of these databases and their capabilities in handling unstructured data in a perfect manner, in addition to their low cost and easy scalability are making them popular choice exceedingly. It expected that in the near future, NoSQL databases will go through major improvements and will address many of its shortcomings features.

IV. CHOOSING AN APPROPRIATE DATABASE

It is important to select carefully the appropriate database model for the enterprise or the application we intend to create. Many factors should be taken into consideration before such selection. Some of these main factors are the type and the amount of data, the schema characteristics, the cost, the transactions amount and how frequent they are called [28]. Generally, for smaller and medium applications, a relational database would be a reasonable choice, while for big applications, that handle and manipulate large amount of data, a NoSQL database could be chosen. The purpose of the application that need to be developed is also a criteria for choosing a proper database model. Therefore, it is extremely important that we always identify the requirements of each application, as this is an essential issue. If the budget is an issue in developing the application, then it is a big advantage for most NoSQL databases as they are open source solution.

So, for selecting the proper database that can satisfy all of the application requirements, all issues introduced above should be taken into consideration before developing the application or the project. We can choose the most appropriate database model, a non-relational or a relational database subject to what each application needs.

IT Specialists or database administrators (DBA) should decide using traditional relational model or using a NoSQL database according to the needs of the application. They should also choose the appropriate type of NoSQL databases according to the format and features of the enterprise's operating data. After selecting the appropriate NoSQL database category, the IT Specialists can select the specific database system according to the characteristics of each database, the needs of the enterprise, as well as the popularity and the feedback about each database on websites [20].

In general, choosing to use or not to use NoSQL Database could be summarized as follows:

If the data is documents, unstructured or semi-structured with advanced query features, or if the data is schema-less or the schema is continuously changing while the consistency is preferred over availability, then use NoSQL Database.

On other hand, if the data is structured data or it is extremely relational while high availability is preferred over consistency, then do not use NoSQL Database.

In October 2021, the top databases, according to DB-Engines Ranking [29], that ranks database management systems according to their popularity is still Oracle. But among the top 10 databases, four NoSQL databases are presented, as Figure.1 shows. Among the NoSQL databases, MongoDB (Document Database), Redis (Key-Value Store), Elasticsearch (full-text Search Engine) and Cassandra (Wide-Column Database) are the leaders in their domains. The NoSQL databases, especially MongoDB, occupy more and more space on the market, but with most traditional relational databases are still constant. The growth of NoSQL databases is happening against the slowdown in the RDBMS market. Oracle, which sits atop the database popularity rankings, keeps deriving less and less of its revenue from new license sales [30].

As Fig. 1 shows, even though NoSQL databases keep winning converts, enterprises are keeping their RDBMSes around, relational databases still dominate big data and will do so for quite some time.

Rank			DBMS	Database Model	Score		
Oct 2021	Sep 2021	Oct 2020			Oct 2021	Sep 2021	Oct 2020
1.	1.	1.	Oracle	Relational, Multi-model	1270.35	-1.19	-98.42
2.	2.	2.	MySQL	Relational, Multi-model	1219.77	+7.24	-36.61
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	970.61	-0.24	-72.51
4.	4.	4.	PostgreSQL	Relational, Multi-model	586.97	+9.47	+44.57
5.	5.	5.	MongoDB	Document, Multi-model	493.55	-2.95	+45.53
6.	6.	↑ 8.	Redis	Key-value, Multi-model	171.35	-0.59	+18.07
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	165.96	-0.60	+4.06
8.	8.	↓ 7.	Elasticsearch	Search engine, Multi-model	158.25	-1.98	+4.41
9.	9.	9.	SQLite	Relational	129.37	+0.72	+3.95
10.	10.	10.	Cassandra	Wide column	119.28	+0.29	+0.18

Fig. 1. Top 10 Most Popular Databases – October 2021 [29].

V. CONCLUSIONS

This paper tries to answer the following important question: How do we choose the database model that suites our application?. More precisely, when should we select a NoSQL database model against relational database or vise-versa?

In order to answer these questions, a general comparison has been made between NoSQL databases and relational databases. The advantages and the limitations of relational databases have been reviewed. This paper highlights the basic concepts of each type of NoSQL databases along with the data features that each NoSQL database is suitable for processing. Various advantages and disadvantages of NoSQL databases as compared to relational databases have been also reviewed. Finally, factors that should be considered for selecting the suitable database model that can satisfy the application requirements have been discussed. NoSQL is increasingly considered a feasible alternative to relational databases especially for mobile applications and interactive web.

REFERENCES

[1] Codd, E. F. (1970) A relational model of data for large shared data banks, *Communications of the ACM*, 13, 6, 377-387.

[2] Lee, C. H., and Zheng, Y. L. (2015, October). SQL-to-NoSQL Schema De-normalization and Migration: A Study on Content Management Systems. In Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on (pp. 2022-2026). IEEE.

[3] Karnitis, G., and Arnicans, G. (2015, June). Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation. In Computational Intelligence, Communication Systems and Networks (CICSyN), 2015 7th International Conference on (pp. 113- 118). IEEE.

[4] Schram A, Anderson KM (2012) Mysql to nosql: data modeling challenges in supporting scalability. In: Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity. ACM, Tucson, Arizona, USA. pp 191–202

[5] Liao, Y. T. et al., (2016). Data adapter for querying and transformation between SQL and NoSQL database. Future Generation Computer Systems, **65**, 111-121.

[6] A Comparative Study of Databases with Different Methods of Internal Data Management https://pdfs.semanticscholar.org/75b1/c2b5bd593b7a47786c6cf3cb8593554aa746.pdf Tauro, C.J.M., Aravindh, S. and Shreeharsha, A.B. (2012) Comparative Study of the

[7] New Generation, Agile, Scalable, High Performance NOSQL Databases. International Journal of Computer Applications, 48, No. 20. https://doi.org/10.5120/7461-0336

[8] Behjat U Nisa, "A Comparison between Relational Databases and NoSQL Databases", International Journal of Trend in Scientific Research and Development (IJTSRD), Volume – 2, Issue – 3, Mar-Apr 2018.[S-4]

[9] Han, J., Song, M. and Song, J. (2011) A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing. Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science, Sanya, 16-18 May 2011, 351-355. https://doi.org/10.1109/ICIS.2011.61

[10] Tudorica, B.G. and Bucur, C. (2011) A Comparison between Several NoSQL Databases with Comments and Notes. 2011 10th Roedunet International Conference (RoEduNet), Iaşi, Romania, 23-25 June 2011, 1-5. https://doi.org/10.1109/RoEduNet.2011.5993686

[11] Ahsan Malik, S.M.Aqil Burney, Fawad Ahmed, "A Comparative Study of Unstructured Data with SQL and NO-SQL Database Management Systems", Journal of Computer and Communications · January 2020, DOI: 10.4236/jcc.2020.84005

[12] N. Jatana, S. Puri, M. Ahuja, I. Kathuria, D. Gosain, "A survey and comparison of relational and non-relational databases", International Journal of Engineering Research &Technology (IJERT) ISSN: 2278-0181, vol 1 Issue 6, august 2012, pp. 1-5.

[13] https://bitnine.net/blog-computing/sql-vs-nosql-comparative-advantages-and-disadvantages/

[14] https://www.educba.com/relational-database-advantages/

[15] Salah A. Jowan1*, Ramadan Faraj Swese1, Antisar Yousf Aldabrzi1, Mahmood Saad Shertil2, "TRADITIONAL RDBMS TO NOSQL DATABASE: NEW ERA OF DATABASES FOR BIG DATA", Journal of Humanities and Applied Science (JHAS) Issue No. (29) December 2016. [S-6]

[16] Sherpa Software. "Structured and Unstructured Data: What is It?", http://www.sherpasoftware.com/blog/structured-and-unstructured-data-what-is-it/.

[17] S. Grimes, "Unstructured Data and the 80 Percent Rule", http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/.

[18] NoSQL Databases. Available online: http://nosql-database.org/

[19] C. Bazar, C. Iosif, "The transition from RDBMS to NoSQL. A comparative analysis of three popular non-relational solutions: Cassandra, MongoDB and Couchbase", Database Systems Journal, vol. V no 2/2014, pp.49-59.

[20] Jeang-Kuo Chen, Wei-Zhe Lee, "An Introduction of NoSQL Databases Based on Their Categories and Application Industries", Algorithms **2019**, 12, 106; doi:10.3390/a12050106.

[21] Sullivan, D. NoSQL for Mere Mortals, 1st ed.; Pearson P T R: London, UK, 2015.

[22] Bruno Cabral, Paulo Carreiro, Marco Vieira1 and Jorge Bernardin, "Choosing the right NoSQL database for the job: a quality attribute evaluation", *Journal of Big Data* (2015) 2:18, DOI 10.1186/s40537-015-0025-0.

[23] Hecht R, Jablonski S (2011) Nosql evaluation. In: International Conference on Cloud and Service Computing. IEEE, Hong Kong, China. pp 336–41

[24] Priyanka, AmitPal, "A Review of NoSQL Databases, Types and Comparison with Relational Database", International Journal of Engineering Science and Computing, May 2016, DOI 10.4010/2016.1226.

[25] Ameya Nayak, Anil Poriya, Dikshay Poojary, "Type of NOSQL Databases and its Comparison with Relational Databases", International Journal of Applied Information Systems (IJAIS) –Foundation of Computer Science FCS, New York, USA Volume 5– No.4, March 2013 – [S-9]

[26] Harpreet kaur, Jaspreet kaur, Kamaljit kaur, "A Review Of Non Relational Databases, Their Types, Advantages And Disadvantages", International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 2, February- 2013

[27] João Ricardo Lourenço, Bruno Cabral, Jorge Bernardino and Marco Vieira, "Comparing NoSQL Databases with a Relational Database: Performance and Space", International Journal of Big Data, Vol. 2, No. 1, 2015.

[28] Cornelia Győrödi, Robert Győrödi and Roxana Sotoc, "A Comparative Study of Relational and Non-Relational Database Models in a Web- Based Application", International Journal of Advanced Computer Science and Applications, Vol. 6, No. 11, 2015. [s-0]

[29] https://db-engines.com/en/ranking

[30] Matt Asay, "NoSQL databases eat into the relational database market", Published March 4, 2015, Available: http://www.techrepublic.com/article/nosql-databases-eat-into-the-relational-database-market/, accessed July 2015.