Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MULTI-CRITERIA ANALYSIS BETWEEN NOSQL DATABASES CATEGORIES TOWARD A COMPLETE MIGRATION FROM RELATIONAL DATABASE

[1]ABDELHAK ERRAJI1, [2]ABDERRAHIM MAIZATE, [2]MOHAMMED OUZZIF

[1] RITM ESTC Laboratory Hassan II University, ENSEM Casablanca, Morocco

[2] RITM ESTC Laboratory Hassan II University, ESTC Casablanca, Morocco

E-mail: [1] erraji.abdelhak@gmail.com, [2]maizate@hotmail.com, 3 ouzzif@gmail.com

ABSTRACT

In recent times, the world has become highly dependent on computer science which has become extends to various vital and secondary areas of life, such as social communication, security, commerce, marketing, training, and other fields, as many institutions have become dealing with a huge amount of information to Storage and use for their benefit, in order to manage the present and the future well. This situation made many institutions need several means to keep pace with this great development of information in terms of quantity and quality, without losing what they could possess of previous information stored within databases, often relational system which provide easy ways of dealing with the information, also provides the possibility of applying several complex and important calculations and elicit. Developers in recent days have came out with what's called a NoSQL database, that surpasses its previous predecessor, these developers have also made many programs and tools that works with this new structure, that's characterized in 4 different categories, each one differs from the other in its fundamentals, that's why finding a way to migrate Data Base with all of data and effects from a relational system to this new structure became a must. In this paper, we will study in-depth and comprehensively the different categories of the NoSQL system, by following the comparative approach using WSM method and depending on a set of characteristics and features, in order to determine the optimal category that enables us to completely migrate all data from a relational system with all its details, and capabilities towards a NoSQL system, in order to put the first step to start studying the migration of data from a relational system to a NoSQL system

Keywords: *NoSQL; Big Data; Migration; Relational Database; WSM;*

1. INTRODUCTION

The remarkable development of humanity around the world has led to the desire to transcend borders, and bypass restrictions between the various populations of the world, which led to the development of communication between people and the flow of a huge amount of information between them, which enriched databases with important information that became exploited directly or allowing us to obtain other important information, an encouragement for the development of global trade and profits and the fields of science, health, security, politics and others. And that is through analyzing the information and exploiting it in future expectations and studying the movements of individuals and institutions and helping to make decisions through reporting digitization. This situation made the possession of more and more important information, as well as the methods of studying, analyzing and relying on it in order to achieve its successes, its development, and continuity in the market and the rivalry of rivals, a goal for every company and institution, especially those competing. For this, these institutions need new data systems that are sophisticated and capable of storing and exploiting the huge amount of accumulating information quickly and abundantly, making up the so-called big data.

This situation made traditional data systems, including relational ones, define a clear deficit, especially in the matter of analyzing and exploiting large data in small times, despite having a strict mathematical system that enables the completion of complex or simple mathematical formulas to derive new information from those stored in it.

This situation gave more space for the NoSQL database system to appear and sit on the throne of information and databases, due to the high efficiency that it showed in reality. This situation

has caused many companies to use this latter system and migrate their data from the relational system to this new system, despite its modernity and the recent research carried out around it by researchers and companies to raise its efficiency and quality until the need for the relational system is eliminated.

On the other hand, until the maturity of the NoSQL system, the relational system has an important set of advantages that make dispensing with it at the present time a risk with uncalculated consequences, which we will separate in the following paragraphs, Therefore, most companies and institutions that want to move to the NoSQL system find themselves forced to use the both systems, which raises the cost of dealing with information and the cost of investing in both systems together and the programs affiliated with each system and the human resources qualified to deal with these systems and continuous training in both systems.

In this context, researchers seek to create and create ways and means to transfer information from the relational system to the NoSQL system, but this research still has not reached the maturity level and the ability to dispense with the relational system. This research, including this research, will continue until it reaches a real starting point that changes the course of scientific research and makes the NoSQL system dominate the rest of the database systems.

This desired dominance is also linked to the means of analysis and the mechanisms for exploiting this data, which has known tangible and continuous progress and remarkable development in recent years.

In this research, we will seek an in-depth analytical study and comparison of the various types of NoSQL databases in order to determine the most appropriate type to migrate a relational database with all its data, details and capabilities to the NoSQL system, based on a set of important and separating characteristics and features. Through a comparative method approved in many scientific research.

2. LITERATURE

In recent years, several comparative studies between the categories of NoSQL databases have been carried out by researchers [1] - [2] - [3] - [4] - [5]. Others prefer to evaluate only the performance [6] - [7] or only the Cloud [8]. These studies are all created to compare these categories from the performance, integrity, reliability, interoperability, cloud support, complexity request and security, etc.

but our study aims to complete the old studies by the realization of a comparative study between all categories NoSQL databases from a structure and useful point of view, to know that it is the most suitable category to make a complete migration from a relational database without neglecting any data coming from the model or the semantics of the relational database. We carried out this comparison by the multi-criteria analysis method (WSM). The choice of the target category and result of the study is justified by a set of criteria carefully chosen and classified according to the rules of the method (WSM).

3. RELATIONAL DATABASES AND NOSQL DATABASE OVERVIEW

3.1. Relational Database Management System

In view of the relational system of databases, it is controlled by a large set of rules and principles of storage and verification of the integrity of information and data in terms of quantity and data type, and even the field of acceptance of values, as well as the rules of management, which guarantee the possibilities of recording operations in tables, their enumeration and the possibility of Developing formulas to calculate and derive other new and important information that is not stored and does not originally exist in databases.

By studying this type of database, we find that it is characterized by being a complex of several overlapping and tightly organized information in different and integrated layers, in addition to the appendices and suffixes that increase the efficiency of these systems.

These overlapping parts in the construction of this type of data can be summarized in: stored information, information structure, and the semantic that frames the structure and determines its roles, its domain of influence, and the possible operations between the various elements, the operations through the structure itself or the possible ones we can do with it, and we could also determine the elements qualified to carry out those operations. In addition, a set of suffixes are added, represented in stored programs, conditions and stored calculate formulas to as well, the index and precautions in deleting and adding information so that there is no defect in the integrity of the stored information.

3.2. NoSQL DataBases

One of the most important features of a NoSQL Database system is that we liberate the large amount of rules that control the storage system and the structure of information, contrary to the relational system where these restrictions must be

complied, to have the flexibility to store more and diverse information without restrictions.

This situation made these modern systems capable of storing very different information, without specific structure.

Although it has become possible to store any information regardless of its source, form and type, the big difference between the stored information and the absence of a structure to control the storage makes it difficult to exploit and analyze the information.

But logic requires the existence of a certain structure, even if it is flexible or comprehensive, so that we can exploit this data and information. This deficiency opened the door for creating programs to help build a specific and flexible system with a set of rules controlling the database structures. The thing that added to the superiority of this system is that it is able to store all the information with the option to place it within a specific structure or without it.

These NoSQL systems are organized into four important categories, where each category undergoes a specific philosophy and general form of information storage and management.

In this paragraph, we will present these items with their introduction and components and shortcomings of each category separately

NoSQL databases are used for many applications. There are different categories of NoSQL databases are defined based on these structure follows:

1. Key value
2. Column oriented
3. Document oriented
4. Graph based

3.2.1. key Value Category

This way of storing data is intuitive, simple, understandable, efficient and fast, it is based in their storage of data on the principle of the key / value pair, the key of which is unique in the entire database and for each key there is only one value, it's like a Map or a dictionary. The present structure is fast in indexing, that is to say that access to values is much faster. In regards to the response time of requests using this technique is very small. The data is stored with the less format scheme [9]. This storage technique is very suitable for distributed storage. In a scenario where relationships or structures are required, key-value storage is not suitable [8]. The key value store is used in web sessions or on any user specific information site. The reason is that user data (i.e. value) should not be viewed directly; it must be accessible by using the key. [12]

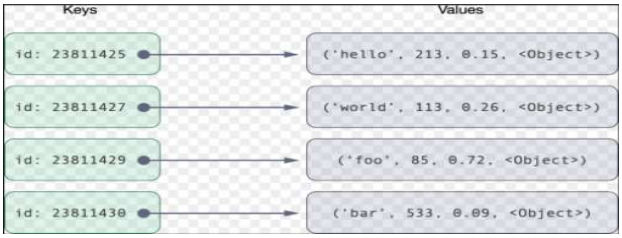

Figure1: KeyValue Store NoSQL DataBase [13]

3.2.2. Column Store Category

This way of storage, does not store the values in the rows so that each row represents a real object whose values represent an integrity depends on the primary key as in the case of the BDDR, on the other hand the philosophy of this way of storage is to store the values in the form of columns. unlike relational databases, whose values are stored in rows, which can have NULL values in their cells to represent blank information, which wastes memory, and have empty boxes in their tables, presenting a minus in the way of storage and a remarkable drawback, the column-oriented storage structures have come to overcome this drawback, knowing that it regroups all the data dispatched in a single and giant table.

This way of storage will not subsequently need the notion of joining tables as in the case of BDDRs, which decreases the complexity of creating the data source before projection and restriction during queries, but at the same time it penalizes the derivation of the data coming from the calculation formulas.

The column-oriented storage structures distribute the data on several groups, therefore a large amount of data can be easily manipulated and processed, which guarantees the scalability and its improvement, as well as the simplicity of unstacking on Mapreduce [10], therefore easily deployed for big data. Column-oriented databases offer better indexing and a better query structure than key-value databases [11].

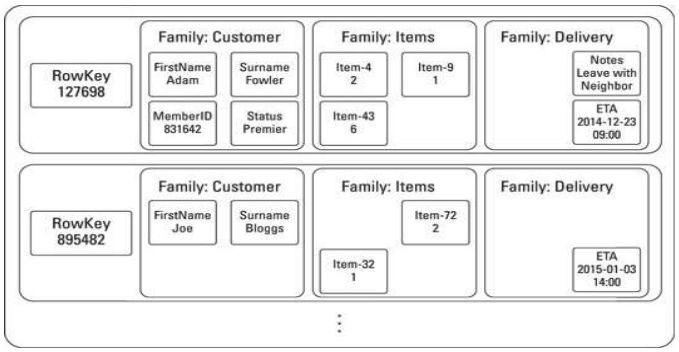

Figure 2: Column Store NoSQL DataBase[14]

3.2.3. Document Store Category

This way of data storage stores data as documents rather than simple values, or values by column or by row. This way of storage is often used in technologies XML [extensible markup language] or JSON [Javascript Object Notation] to store information as documents.

The advantage of using the JSON format is that different programming object structures can be easily mapped into this format [12].

This way of storage is based on a dynamic diagram; this makes it possible to easily add a field anywhere in its documents, which has present a big advantage with respect to fixed schema structures such as column-oriented databases or relational databases, where a new attribute must be added for all records, if the values are not known; then many null values are going to be added. This type of storage provides indexing based on the primary key [11].

This storage method is very suitable for blogs and content management systems.

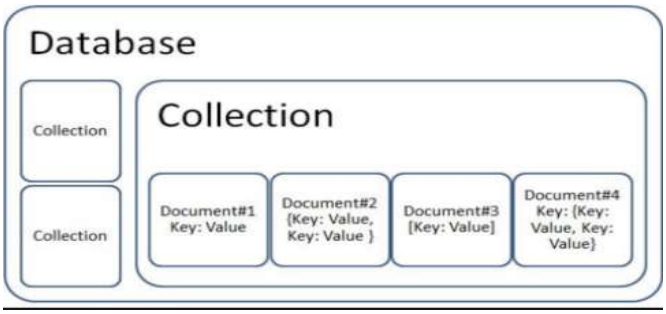

Figure3: Document Store NoSQL DataBase[13]

3.2.4. Graph Store Category

This way of data storage uses the notion of the graph, where a set of nodes are connected. This storage mode has no predefined schema, and can be used by databases based on graphs according to a very dynamic scheme. So it is the better mode for semi-structured data and unstructured data. Also this type of storage is better suited to social networking sites, which considers users connected like the nodes of a graph, and at the same time the source of information. Depending on this case, data can be stored using nodes as users and edges as that connection between users. Graphic databases present difficulties during scalability horizontal, i.e. when the nodes are going to be connected, and then they are distributed in the clusters, with considerable difficulty in crossing or handling them.

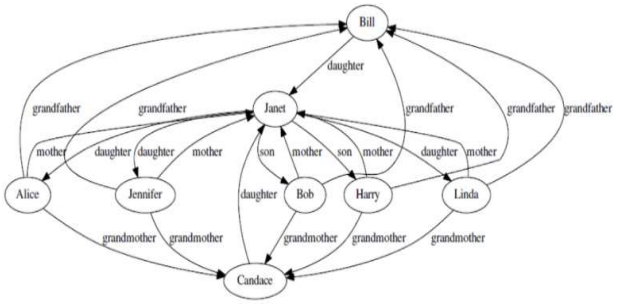

Figure4: Graph Store NoSQL DataBase[15]

4. MULTI-CRITERIA COMPARATIVE STUDY

The objective of this comparison is to determine the NoSQL category that favors a complete migration from relational databases, knowing that each category adopts a structure that does not implement certain rules of the relational model in order to serve as a processing mode. This mode of treatment is adapted to the needs of the profession of a set of organizations. For example, the need for a messaging application has nothing to do with the need for a banking application. Indeed, in a messaging application, the possibility of sending messages must be available at all times, and the loss of a few messages does not cause a major problem in the functioning of this system, nor in its activity. On the other hand, in a banking application, the loss of a few operations forms a major problem in the operation of the system, which requires it to be stopped in order to rectify and correct this problem.

Also we notice that the relational model is endowed with a set of constraints, making its strength and its weakness at the same time, of which the objective is to keep the coherence and the integrity of the stored data, but at the same time they weighed down the processing, for example the use of the primary key in the determination of the rows and the use of the information rows as a basis of storage, will increase the difficulties in the joins and the extraction of the data, for a very large mass of data. Therefore, NoSQL categories are based on other principles to overcome this difficulty, such as

using column-based storage, according to a new philosophy.

This philosophy is based on the fact that several queries aim to display the columns linked with an identifier, and therefore this approach stores data in the columns with a link with an identifier. The latter approach aims to get rid of the join. The elimination of joins destroys the possibility of spawning other new information, as in the case of relational databases. Column-oriented storage is concerned with the speed of data retrieval, but at the same time destroys the generation of new data, because it has excluded joins. This is because joins are based on two main operations: Cartesian product and restriction, with row-based storage and small or medium-sized tables to make the operation successful, while column-oriented storage stores data in based on columns, instead of rows and uses big tables in their storage philosophy, which shows the exclusion of joins by nature of storage. This approach is created to suit the need of certain applications, which do not require joins, such as messaging applications

In the case of the document-oriented category, we have formatting of information in the form of a document or an object such as JSON, but this structure is very soft and flexible, allowing the storage of any document. This complicates the definition of a standard structure, and consequently it lowers the efficiency of data extraction, knowing that their interaction language is based on a possible structure in principle. This weakness in this structure blocks the evolution of the tools that implement it, for example MongoDB is a tool of this category has resorted to integrating a Framework, which is called mongoose to adopt a considerable structure to organize and control the data on MongoDB, knowing that the establishment of this structure by mongoose is not mandatory for the operation of MongoDB.

From these two categories, which represent the same situation for the other remaining categories, we notice that their storage principles neglect some principle and foundations of the relational system in their philosophy. This will generate a waste of information when migrating from the relational system to a NoSQL system according to one of these categories, knowing that this waste will be functional, structural or semantic.

After having seen a global view on each NoSQL category of Databases, we will now develop a multi-criteria analysis between these categories. Multi-Criteria decision Analysis, or MCDA, is a valuable tool that can be applied to many complex decisions. It can solve complex problems that Include qualitative and/or quantitative aspects in a decision making process [16].

The score of a category is calculated based on a number of criteria. So far we have identified nineteen, based on the characteristics of each of the categories [17]. We have determined nineteen important criteria after a detailed analysis of relational databases to find its equivalent in NoSQL categories study. We succeeded to identify these criteria which are gathered in three kinds: structure, functional and semantics of relational Database. Knowing that the semantic aspect represents the philosophy and the raison d'être of databases.

Our comparative study aims to find out which category has little or no information waste during such a migration. To do this, we are choosing a set of criteria of a functional, structural and semantic nature, as well as a system to give more value to our comparison result. This study forms a basis for all studies that aim to develop an approach for migrating data from a relational system to a NoSQL system.

4.1 Multi-criteria Analysis Method

To make the comparison between the categories of NoSQL Database using a number of criteria, there are several possible mathematical methods. These methods can be divided into three main families [18]-[19]-[20]-[21]-[22]:

• **Complete aggregation (top-down approach):** This approach seeks to aggregate the n criteria to reduce them to a single criterion.

• **Partial aggregation (bottom-up approach):** This approach seeks to compare potential actions or rankings to each other, and to establish between them outranking relations.

• **Local and iterative aggregation:** This approach looks primarily for a starting solution. Thereafter, we proceed to an iterative search to find a better solution. Table1 shows the different existing multicriteria methods sorted by family [18]-[19]-[20]-[21]-[22].

Table 1: Example Of Multi-Criteria Analysis Methods

Family	Methods
Complete aggregation (top-down approach)	TWO WAY ANOVA WSM Method (Weighted Sum Method or Sum of s cores) WPM Method (Weight Product Method or Ratios multiplication) AHP Method (Analytic Hierarchy Process) MAUT (Multi Attribute Utility Theory)
Partial aggregation (bottom-up approach)	ELECTRE, Prométhée, Melchior, Qualifex, Oreste, Regim...
Local and iterative aggregation	IMPROVING CONES METHOD, GOAL PROGRAMMING, STEM, Branch and Bound

4.2 Weighted Sum Method (WSM)

For our analysis, we chose the Weight Sum Method (WSM). Indeed, this method allows to find the best possible category by assigning a weight to each comparison criterion, it allows to take into account all the criteria according to their value and without a criterion penalizing the other criteria [18]-[19]-[20]-[21]-[22].
This method is based on five key elements:

- Potential n actions set A={a1,a2,a3,…,an} ai, where i=1,2,…,n
- M different criteria cj where j=1,2,…,m
- Criteria weights pj for each criteria where j=1,2,…,m
- Evaluations or judgments eij for each action on each criteria where i=1,2,...,n, j=1,2,…,m
- max or min ∑eij*pj for i=1,2,⋯,n, in our case we need to maximize this function to have the better solution

4.3 Comparison Criteria and Weight

After a detailed analysis of relational systems, we notice that they contain data and constituents classified into three types: data, data contained in the structure, and data semantic.

We present in this chapter the nineteen comparison criteria cited on which the comparative study will be based, we notice that these criteria are based on the characteristics of each of the NoSQL databases category in nineteen global criteria to ensure better analysis and optimize the comparison. In the following table, we present these elements according to their nature, these criteria are:

Table 2: explanation of criteria

Natures	Criteria	Significance
Structure	Level of structure	Degree of importance of the structure
	Data type	Distinction of type
	Check constraint	To control a validate data in column or field
	Unique constraint	To control uniqueness
	Not null constraint	To check existence of data
	Primary key	Field to define authors of one entity
	Index	Index and double index
	Foreign key	Field of another structure
Functional	View	Stored query
	Function & procedure	Stored programs
	Trigger	To control data row's validate
	Aggregation	Aggregation, Analyzing and calculation functions
	Querying	The existence of a language to attack BDDs
	Programming	To ameliore evaluate and extracting data
Semantic	Relational Ship	Connection between several structures
	Dependence between fields	Dependence between attributes
	Join operation	The power to link multiple tables or objects, to build a larger data source
System	Scalability	Performance of operation Duplicate and sharding of Data
	Flexibility	Flexibility to change the structure

These criteria are classified according to the following order of importance:

Level of structure=Check constraint=Primary key=Index=Function &

procedure=Trigger=Aggregation= Querying=Programming=Relational Ship=Join operation =Scalability=Flexibility>Data type=Foreign key>Unique constraint=Not null constraint=Dependence between fields=View

And therefore the WSM weight accorded are as represented in tab3 below:

Table 3: Criteria Weight

Criteria	Weight
Level of structure Check constraint Primary key Index Function & procedure Trigger Aggregation Querying Programming Relational Ship Join operation Scalability Flexibility	**3**
Data type Foreign key	**2**
Unique constraint Not null constraint Dependence between fields View	**1**

4.4 Multi-Criteria Choice Matrix

The first step in applying the WSM method is the carrying out of the multi-criteria choice matrix. This matrix' columns contain the NoSQL categories to be compared and its lines contain the different criteria with the weight assigned to each criterion according to its importance.

In the cells, and for each category of the NoSQL Database, there is a score assigned to the criterion cited at the beginning of the line, multiplied by its weight. This score is assigned according to the characteristics of each category. This product will generate a value, which will participate in the sum of the values of the same column to generate an overall score for each category. The scores assigned to each cell can have four values:

3 meaning EXISTING and IMPORTANT in NoSQL category,

2 meaning MUST to be REDEPLOY in NoSQL category,

1 meaning EXISTING and LESS IMPORTANT in NoSQL category,

0 meaning not EXISTING in NoSQL category,

[18]-[19]-[20]-[21]-[22]:

 3: means that the intended NoSQL category has an EQUIVALENT of the criterion, which is more important.

 2: means that the intended NoSQL category has an EQUIVALENT to be REDEPLOY of the criterion.

 1: means that the intended NoSQL category has an EQUIVALENT of the criterion, which is less important.

 0: means that the intended NoSQL category has not an EQUIVALENT of the criterion.

Table 4: Multi-Criteria Choice Matrix

Criteria	KeyValue Store	Column Store	Document Store	Graph Store
Level of structure (W:3)	1*3	3*3	3*3	2*3
Data type (W:2)	2*2	3*2	3*2	3*2
Check constraint (W:3)	0*3	0*3	3*3	1*3
Unique constraint (W:1)	0*1	0*1	3*1	1*1
Not null constraint (W:1)	0*1	0*1	3*1	1*1
Primary key (W:3)	3*3	3*3	3*3	3*3
Index (W:3)	3*3	3*3	3*3	3*3
Foreign key (W:2)	0*2	0*2	3*2	3*2
View (W:1)	0*1	0*1	0*1	0*1
Function & procedure (W:3)	1*3	1*3	2*3	0*3
Trigger (W:3)	0*3	0*3	2*3	0*3
Aggregation (W:3)	1*3	3*3	3*3	1*3
Querying (W:3)	3*3	3*3	3*3	2*3
Programming (W:3)	2*3	2*3	2*3	2*3
Relational Ship (W:3)	0*3	0*3	3*3	3*3
Dependence between fields (W:1)	2*1	2*1	2*1	2*1
Join operation (W:3)	0*3	0*3	3*3	1*3
Scalability (W:3)	3*3	3*3	3*3	3*3
Flexibility (W:3)	3*3	1*3	3*3	3*3
WSM	**66**	**74**	**128**	**88**

4.5 Curve and Comparison Histogram

Figure 5 shows the distribution of the four curves for final scores of each category against the comparison criteria.

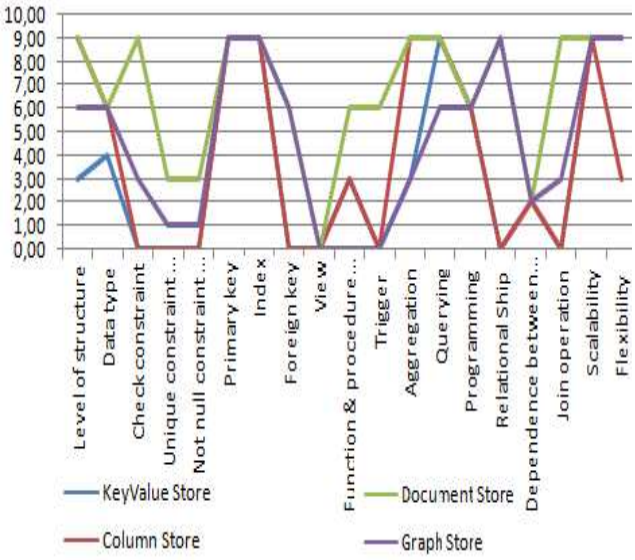

Figure 5: Distribution of Ratings Against The Criteria

The histogram in Figure 6 shows the final score for each category. The best score obtained is 128 /141, it shows a net benefit to the document Store category against the set of selected criteria and over the other categories, it is followed by the Graph Store category, then it is followed by the Column Store category and the key Value store category comes last. We can notice that none of these categories could reach the perfect score 141/141 according to this comparative approach.

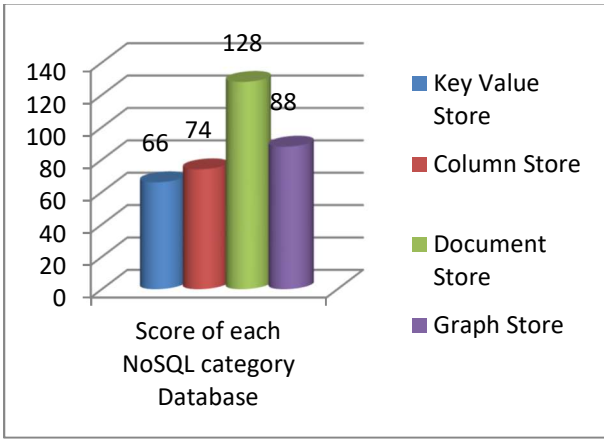

Figure 6: Categories NoSQL DataBase final notation

5. CONCLUSION

In this article, we carried out a multi-criteria comparative study between the different categories of NoSQL databases, based on the WSM multi-criteria method, and using crucial criteria deduced from the structure and semantics of the relational database. These criteria are very important because they are the key elements of a full migration to bigData; indeed, if we manage to create an equivalent for each of these criteria, then we will have a complete and satisfactory migration to NoSQL databases, and at the same time we manage to free ourselves from the obligation of using relational systems.

REFERENCES:

[1] Bathla, G., Rani, R., & Aggarwal, H. (2018). Comparative study of NoSQL databases for big data storage. *International Journal of Engineering & Technology*, *7*(2.6), 83-87.

[2] Kunda, D., & Phiri, H. (2017). A comparative study of nosql and relational database. Zambia ICT Journal, 1(1), 1-4.

[3] Qader, M. A., Cheng, S., & Hristidis, V. (2018, May). A comparative study of secondary indexing techniques in LSM-based NoSQL databases. In Proceedings of the 2018 International Conference on Management of Data (pp. 551-566).

[4] Bagga, S., & Sharma, A. (2020, March). A comparative study of NoSQL databases. In The International Conference on Recent Innovations in Computing (pp. 51-61). Springer, Singapore.

[5] Banane, M., Belangour, A., & Labriji, E. H. (2018). RDF data management systems based on NoSQL Databases: a comparative study. International Journal of Computer Trends and Technology (IJCTT), 58, 98-102.

[6] Veronika Abramova,Jorge Bernardino, Pedro Furtado :Which NoSQL Database? A Performance Overview. Open Journal of Databases (OJDB) ,Volume 1, Issue 2, 2014

[7] Veronika Abramova,Jorge Bernardino, Pedro Furtado: EXPERIMENTAL EVALUATION OF NOSQL DATABASES.

[8] Dayne Hammes, Hiram medero , Harrison Mitchell: Comparison of NoSQL and SQL Databases in the Cloud , Proceedings of the Southern Association for Information Systems Conference, Macon, GA, USA March 21st–22nd, 2014

[9] Nayak A ,Poriya A and Poojary D (2013), Type of NOSQL Databases and its Comparison with Relational Databases, IJAIS, vol. 5, no. 4,pp.16-19.

[10] Dean J and Ghemawat S (2008), Mapreduce: simplified data processing on large clusters, Communications of the ACM, vol. 51, no. 1,

pp. 107–113, http://doi.acm.org/10.1145/1327452.1327492

[11] Grolinger K, Higashino WA, Tiwari A and Capretz MAM (2013) , Data management in cloud environments : NOSQL and NEWSQL data stores, Journal of Cloud Computing : Advances, Systems and Applications, vol. 2 no.1.

[12] BATHLA, Gourav, RANI, Rinkle, et AGGARWAL, Himanshu. Comparative study of NoSQL databases for big data storage. International Journal of Engineering & Technology, 2018, vol. 7, no 2.6, p. 83-87.

[13] https://medium.com/@mark.rethana/introduction-to-nosql-databases-c5b43f3ca1cc

[14] https://www.dummies.comprogrammingbig-datacolumnar-data-in-nosql

[15] https://aws.amazon.com/fr/blogs/aws/new-store-and-process-graph-data-using-the-dynamodb-storage-backend-for-titan/

[16] Batouta, Z. I., Dehbi, R., Talea, M., & Hajoui, O. (2015). Multi-criteria analysis and advanced comparative study between automatic generation approaches in software engineering. Journal of Theoretical & Applied Information Technology, 81(3).

[17] Hajoui, O., Dehbi, R., Talea, M., & Batouta, Z. I. (2015). An advanced comparative study of the most promising nosql and newsql databases with a multi-criteria analysis method. Journal of Theoretical and Applied Information Technology, 81(3), 579.

[18] Olson, D. L. (2001). Comparison of three multicriteria methods to predict known outcomes. European Journal of operational research, 130(3), 576-587

[19] Schärling, Alain, Décider sur plusieurs critères, Presses Polytechniques Romandes, 1985, 304 pages.

[20] Talbi, El-G., Méthodes d'optimisation avancées, LIFL CNRS. [Online]. Available: (http://www.lifl.fr/~talbi/Coursoptimisation.pdf)

[21] Zeleny, M., Multiple criteria decision making, McGraw-Hill, Columbia University, 1982, 563 pages.

[22] Lehoux, N., & Vallée, P. (2004). Analyse multicritère. Ecole Polytechnique de Montréal.