American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

© Global Society of Scientific Research and Researchers

http://asrjetsjournal.org/

New Perspectives for NoSQL Database Design: A Systematic Review

Cristofer Zdepski[a]*, Tarcizio Alexandre Bini[b], Simone Nasser Matos[c]

[a,b,c]*Federal University of Technology – Parana (UTFPR), Address Doutor Washington Subtil Chueire,330-Jardim Carvalho, Ponta Grossa, 84017-220, Parana, Brazil*

[a]*Email: cristoferz@gmail.com*

[b]*Email: tarcizio@utfpr.edu.br*

[c]*Email: snasser@utfpr.edu.br*

Abstract

The use of NoSQL databases has increasingly become a trend in software development, mainly due to the expansion of Web 2.0 systems. However, there is not yet a standard to be used for the design of this type of database even with the growing number of studies related to this subject. This paper presents a systematic review looking for new trends regarding strategies used in this context. The result of this process demonstrates that there are still few methodologies for the NoSQL database design and there are no design methodologies capable of working with polyglot persistence.

Keywords: database design; NoSQL; polyglot persistence.

1. Introduction

The constant expansion of Web 2.0 has increasingly brought to light the need to use different alternatives for data storage with greater scalability, availability, performance or even with query capabilities that are not compatible with the widespread relational model. This need has caused the emergence of several data models seeking to achieve these capabilities, as demonstrated by NoSQL data models, but which can also be called non-relational databases. Internet giants such as Google and Facebook have implemented the first solutions that have since then been made public, such as *Google BigTable* and *Apache Cassandra*, and have become references to non-relational data models.

----

* Corresponding author.

Currently, there are many database software that applies the concepts of these tools or similar in the search for solutions to the problems of scalability and availability. Scalability was not the only motivation for the emergence of the various NoSQL data models. Query formats motivated the creation of new data models capable of providing answers in a more agile and with lower computational costs. Graph-driven models are motivated by the need to navigate highly-connected data relationships, such as social media or geographic information. Despite its name, the relational model does not perform well when the need is to navigate the relationships of its entities, due to its format of storage and data extraction. The data-storage structure of graph-oriented databases differs greatly from other NoSQL models. Even so, the time of emergence of these databases coincides with the others and this is one of the reasons why the data model of graphs was also framed in the definition of NoSQL, besides the fact that it is not a relational database. Even with the large amount of software available on the market and the growing application of these models in Web 2.0-based applications, none of these models is able to handle all the requirements of a more complex application [1]. Applications such as session control, content management, logging and routing can be encapsulated in a single software and are distinct contexts to be handled by a single data model efficiently. All data models are efficient for certain contexts but do not present great performances in others. To provide this capability, applications have increasingly used the application of multiple data models, assigning each model to the context where it presents its greatest qualities. We can say that these applications uses a polyglot persistence model. The term polyglot has arisen in the context of polyglot programming, where multiple programming languages are used in an application so that the best characteristics of each can be exploited. The large number of NoSQL software currently available makes it difficult to use these NoSQL models because of their lack of standardization. While the relational models have the SQL (Structured Query Language) that makes the use of the different software uniform even with small peculiarities of each, the NoSQL models do not have any type of access standardization. Data access is often given through software-specific interfaces, or even through API integration developed by each of the manufacturers. This lack of standardization implies difficulties in adapting development teams, who need to learn the interface of each software, and difficulties in migrating between different software, even when dealing with the same data model. The lack of standardization also affects the database design in the context of NoSQL databases. Currently, there is no widely-used method definition to be used for projects involving NoSQL databases, as in relational databases. The absence of schema (or “schema-less” as it is commonly named) applied to these data models gives the impression of no need for database design. This impression is erroneous, since even if there is no schema directly defined in the database, the data need some level of standardization so that it can be used by the application. Therefore, a schema definition is required for the application's operation, determining how data is stored and recovered and to attend to this some kind of database design is required. Some authors such as [2,3] mention that these databases are developed through the application of best practices from their manufacturers and through examples. This work aims to find out what strategies are currently being developed and used for the design of applications that use NoSQL databases and also applications with polyglot persistence. A systematic review is an efficient method for this survey. This work is organized as follows: Section 2 presents the related works. Section 3 the methodology used for the review and a detailed description of its stages. Section 4 answers the research questions and a discussion of the found results. Section 5 presents the conclusions about the review.

2. Related Works

In the available literature, just a few works were done focusing on systematic literature reviews about design approaches related to NoSQL data models. However, many surveys are available in the domain of NoSQL data models and big data, focusing on many different subjects such as performance analysis, challenges on utilization of specific models [4] and benchmarks, big-data researches [5], application of NoSQL data models on different domains, and much more. This shows that the NoSQL data models have been used even more frequently on past years to solve the most diverse problems [5]. Among these surveys, [6] presents the comparison between NoSQL data models based on quality attributes, technique that is used for both design-oriented decisions and software development. A comparison between NoSQL databases and NewSQL is provided by [7]. They also concluded that there is a lack of standard query language, comparison and benchmark criteria and also limited documentation on most of these new databases. Even with many works and surveys related to NoSQL databases currently being published, there is still a literature gap regarding systematic reviews for design methodologies and techniques that can be applied to the NoSQL data models and this also applies to design methodologies capable of working with polyglot persistence. This paper aims to fill this gap and clarify which techniques is currently being developed, for which data models it can be used and their applicability on software that requires multiple data models.

3. Methodology

This work consists of a systematic review of the literature, executed according to the *Methodi Ordinatio*, proposed by [8]. This methodology describes the processes for the localization and classification of scientific papers in order to identify the ones of greater relevance through the equation *InOrdinatio*. This equation considers the impact factor, number of citations and year of publication of the papers for the definition of a relevance ranking. The use of 3 indicators to define the relevance of the works aims to increase the efficiency in the location of works that are relevant to the research topic. This ranking can then be used to eliminate less relevant papers by reducing the amount of material to be considered in the detailed reading process, reducing the risk of discard relevant ones. Even so, this method does not completely exclude the necessity for reading the selected papers. It is just a technique which aims to reduce the efforts on paper classification and allows the selection of more relevant papers to be read. The next sections provide a detailed description of the stages applied to carry out this process.

3.1. Estabilishing the research intentions

The purpose of this research is to know which methodologies are currently used for the design of NoSQL databases, as well as the design of applications that use several database models, also called polyglot persistence. The *Methodi Ordinatio* considers that the first step in this process is the definition of research questions that are capable of guiding job searches. For this work the following research questions were defined, which will guide the rest of the process:

- Q1. What are the main authors in the area of NoSQL database design?

- Q2. What are the most common application cases for NoSQL databases?
- Q3. What strategies are used for NoSQL database design?
- Q4. Which of these strategies could be applied to polyglot persistence?

These questions aim to give an overview of the methodologies currently used for the database design of applications with NoSQL data models, such as those using multiple data models, and also discover which of the methodologies found would be able to be used for multiple data models, even if the work itself does not reveal this characteristic.

3.2. Preliminary exploratory search in databases

The search for relevant papers was carried out between April and June 2019 and the following databases were used: ACM Digital Library, IEEE Xplorer, Science Direct and Springer Link. These databases were selected for their relevance in the field of research as described by [9] and their importance in the information technology field. The objective of this research was to find the best databases for the research as well as the relationship between words and with the research subject. In addition, this step also aims to identify differences in the research format of the various databases, in order to allow a more uniform definition of research in all databases. For the preliminary research, several combinations of words were used in the databases in order to identify which search strings would bring the best results for the research objectives. This evaluation was made by evaluating the titles and keywords of the papers returned, by sampling. Through the results were also found other search terms that could be relevant and these were also tested. Were also searched for terms that could be used to eliminate unwanted papers from searches, but no terms were able to eliminate them without also discarding apparently relevant papers. One example is the term "performance" which, despite discarding performance testing papers, also eliminated papers aimed at the NoSQL database design due to the large focus on performance of these databases.

3.3. Search string and database definition

Through preliminary research, it was possible to determine which combinations of words would have the best results in the search process in databases, they are:

- (NoSQL AND design)
- (NoSQL AND modeling)
- (polyglot AND persistence)

All primary search databases might be used for search. However, Springer Link needed an additional filter by *discipline*, using *Computer Science*, to eliminate papers from other areas. The publication period was between 2005 and 2019, a period of relevance for works related to NoSQL databases.

3.4. Databases final search

The final search used the same search string for all the databases with small syntax adaptations between them, aiming for greater similarity in the search criteria. Despite the syntax adaptations of search, the query results are quite different between the databases. The quantitative results of this research can be seen in the Table 1, where the results are detailed by search criteria, database and a total of papers returned.

Table 1: Database search results.

Search String	ACM	IEEE	Science Direct	Springer Link	Total
“NoSQL design”	114	238	615	199	1166
“NoSQL modeling”	128	347	651	216	1342
“polyglot persistence”	37	18	30	27	112
					2620

In Figure 1 we can observe a graph with the quantitative contributions of the databases in the final search process. It is possible to notice that the number of papers returned by the search in the database of Science Direct, with 49.5% of the results, is much higher than the others, what may represent a greater adherence to the researched topic. The same can be observed in Figure 2 where the numbers are detailed by search term. Only the search for “*polyglot persistence*” had more results in ACM when compared to Science Direct, in other terms Science Direct had the greatest results.

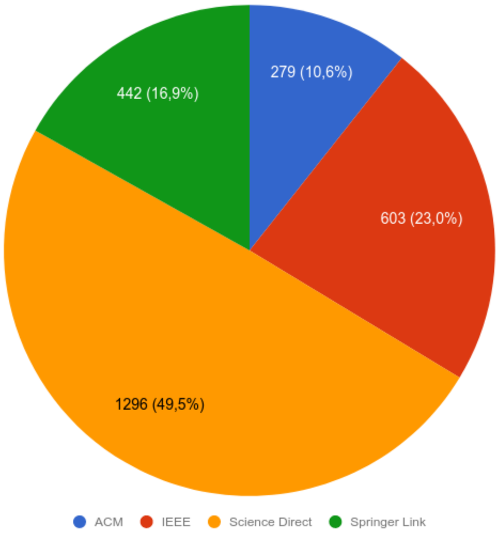

Figure 1: Database contribution.

The results were exported from databases to allow import into Mendeley software. This software was used to aid in the organization and classification of papers, as well as to allow the subsequent storage of the complete papers and to store the data in the cloud.

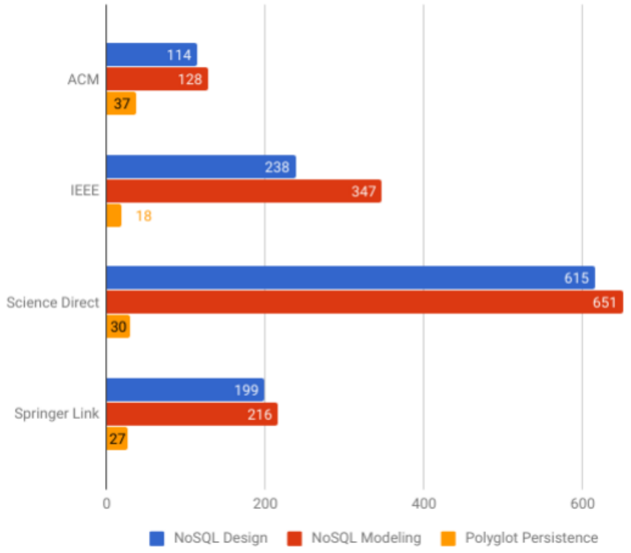

Figure 2: Detailed database contribution.

3.5. Filtering procedures

Although the definition of the search criteria largely eliminates papers not directly related to search intentions, many unrelated works may still appear among the selected papers. So a new filtering process is needed to eliminate them [8]. Many papers are repeated in the final search results because the search is performed with the same criteria in all databases. These repeated papers should be the first to be eliminated by the ease of process execution. In addition, papers from other research areas can be found and some not directly related to the research objective, due to inability to delete only by the search criteria. Mendeley software has a tool for identifying and deleting repeated papers, comparing information such as title, authors, keywords and abstract where possible. Even when the papers are not exactly the same, the software can point to a certain level of similarity. These duplicates are then displayed so the user can make the decision if they really are duplicate papers and decide whether to remove the duplicity. Applying this first filtering method, available in the tool, resulted in 1345 unique papers. The next step was the elimination of contents such as chapters of books, abstracts and editorials for not being considered relevant to the work, resulting in a total of 944 papers. The last step of the filtering process is the analysis of the title, keywords and abstract, seeking to eliminate papers that have no direct relation with the intentions of research. In this process, 750 papers were eliminated, mostly related to performance comparisons among several NoSQL database software, resulting in 194 papers to be submitted to the *InOrdinatio* equation.

3.6. Identifying impact factor, year and number of citations

With the number of papers already reduced by the previous processes we can perform the identification of the impact factor, publication year and the number of citations of each paper, needed information for the *InOrdinatio* equation. The first step in this process was to export the data to a spreadsheet where information could be stored and sorted. Mendeley software provides a tool to perform this export.

The simplest information to find was the paper's publication year because the databases already provide this information in the exported files, so the data was already available in Mendeley and consequently in the generated spreadsheet. For the impact factor, we sought a single tool that would be able to provide the impact factor for all journals, or at least for the vast majority of them. Using a single tool brings more uniformity to values, since different index values can favor some papers because they are on different scales. The tool chosen was the Scimago Journal & Country Rank (SJR), where it was possible to find a large number of journals, and the impact factor considered was the *H Index*. There was a total of 6 journals that were not found in the database and for these was considered the impact factor as 0 (zero). The reason for the application of the value 0 was that *MethodiOrdinatio* is still capable to define the relevance of the papers even if this value is zeroed. The last needed factor for *InOrdinatio* equation is the number of citations for each paper. In the same way as the impact factor, we sought a unique tool for this task, looking for a greater uniformity on results. Being capable to provide this information for all papers, we decide to use the *Google Scholar*.

3.7. Ranking papers using the InOrdinatio

With the necessary indicators for all the papers surveyed the equation *InOrdinatio* can be applied to raise the index of relevance of each one. Values were computed for all the 194 papers. The equation *InOrdinatio* can be seen in Equation 1.

$$Io = (If/1000) + \alpha * (10-(Ry-Py)) + \sum Ci \ (1) \qquad (1)$$

Table 2: Filtered papers by *InOrdinatio*.

Title	If	Py	Ci	Io
Scalable SQL and NoSQL data stores	110	2011	1388	1409,11
Uniform access to NoSQL systems	68	2014	55	97,068
Persisting big data: The NoSQL landscape	68	2016	30	86,068
Database Design for NoSQL Systems	16	2014	51	85,016
Uniform access to non-relational database systems: The SOS platform	251	2012	51	79,251
NoSE: Schema design for NoSQL applications	130	2017	12	75,13
A Polyglot Persistence approach for E-Commerce business model	29	2017	12	75,012
Sinew: A sql system for multi-structured data	119	2014	32	74,119
Schema profiling of document-oriented databases	68	2018	3	73,068
Integrating NoSQL, Relational Database, and the Hadoop Ecosystem in an Interdisciplinary Project involving Big Data and Credit Card Transactions	0	2018	2	72
Integration of Relational and NoSQL Databases	251	2018	0	70,251
BASE analysis of NoSQL database	80	2015	20	69,08
Querying a graph database - Language selection and performance considerations	72	2016	12	68,072
Data models in NoSQL databases for big data contexts	251	2011	10	66,251
Automatic schema suggestion model for NoSQL document-stores databases	17	2018	0	63,017
Mortadelo: A Model-Driven Framework for NoSQL Database Design	0	2018	0	63
Data integration over NoSQL stores using access path-based mappings	251	2011	36	57,251

In this equation, ***If*** is the impact factor previously identified, ***Ry*** the research year (2018), ***Py*** the paper's publication year and ∑***Ci*** the number of citations. The result in the equation is the index *InOrdinatio* ***Io***. The **α** factor brings some flexibility to the *InOrdinatio* index, giving the possibility of defining a relevance parameter for the publication year in the index. The higher the value of **α**, the more relevant becomes the paper's publication year. This value can vary between 0 (zero) and 10. In this research a value of 7 was used for the factor ***α*** since the most recent works tend to be more relevant in the research area. Based on the computation of the *InOrdinatio* index the papers were sorted by relevance and an acceptance criterion was defined for them. In this research only the papers with *InOrdinatio* index greater than 55 were considered, resulting in 17 papers, as can be seen in the Table 2. It is possible to observe that the number of citations had a great influence on the index of the paper "Scalable SQL and NoSQL data stores", which has the larger number of citations, being a very high index in relation to the others, even having a median impact factor.

3.8. Finding the full papers

After the previous steps of filtering, sorting and deleting we have a set of papers of greater relevance for the research and we can then search for their complete contents. This process is between the last stages because it is a laborious process and it is often not easy to find the complete content of some papers. Some of them are only available in restricted communities or need to be purchased and limiting the number of papers to be searched facilitates the research process. In order to obtain the selected papers, first we searched for their contents in the database where they were found. The Table 3 enumerates the source databases of the selected papers. Only the "BASE analysis of NoSQL database" was found in 2 databases, ACM and Science Direct.

Table 3: Selected paper source database.

Database	Number of Papers
ACM Digital Library	4
IEEE Xplorer	2
Science Direct	5
Springer Link	5

Some of the papers were not available in the databases where they were found and needed to be searched through the Internet, mainly through the *Google Scholar* that provides some missing papers. All papers were found through this method and no one had to be purchased. These papers have been inserted into Mendeley to make it easier to read and store annotations.

3.9. Reading and systematic analysis

In this step, a complete reading and systematic analysis of all 17 papers presented in Table 2 were performed. The purpose of reading is to prepare summary records and to answer the questions raised at the beginning of the systematic review process. Summary records are also useful for future research on the subject.

4. Discussion

In this section, the questions addressed at the beginning of the systematic review will be answered, according to the selected papers.

- **Q1. What are the main authors in the area of NoSQL database design?**

Among the selected papers, Rick Cattel appears to be the most relevant author, because of the paper [10], mainly by the large number of citations. We can also highlight the authors Paolo Atzeni, Francesca Bugiotti and Luca Rossi who are mentioned as authors or co-authors in 3 papers that were considered relevant in the research done by *MethodiOrdinatio*, they are: [11, 12, 13]. The papers work with logical modeling on NoSQL databases and a platform for uniform access to data.

- **Q2. What are the most common application cases for NoSQL databases?**

Among the analyzed papers, the most common cases of application of NoSQL databases are Web 2.0-oriented applications that use large amounts of unstructured or semi-structured data such as social media, content managers and particularly e-commerces. E-commerce systems are often used as an example structure in the papers, for example in [14]. The reason for using e-commerce systems is due to the fact that it has several subsystems with quite distinct non-functional requirements, as can be seen in Figure 3.

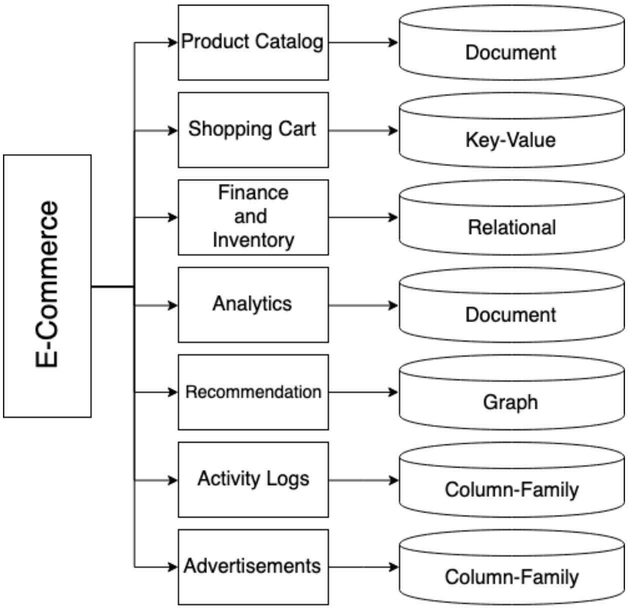

Figure 3: E-commerce structure [14]

The need for more flexible storage structures is often pointed out as one of the justifications for using NoSQL databases that have the so-called "schema-less". These structure types are often required in content managers or log recorders.

- **Q3. What strategies are used for NoSQL database design?**

Among the analyzed papers, there is no uniformity in the strategies used for the NoSQL database design. The proposals differ greatly in tools and methods used, even when dealing with the same data model. Many of them are restricted to only one data model and some even to a single NoSQL database software. NoAM (NoSQL Abstract Model) is presented on [13], as a methodology that proposes a way of representing intermediate data, regardless of the database to be used. This strategy applies only to aggregate-based models and allows the same model to be applied to key-value, document-oriented, and column-family databases. According to the authors, the intermediate model could be used with any of the aggregate data models, simply by converting the intermediate model to the desired specific model. However, the strategy does not consider particularities of each model and its primary objectives, only considers a standardized structure to be used for all. With a similar approach, [15] introduces Mortadelo: A Model-Driven Framework for NoSQL Database Design. It introduces a Generic Data Model that is composed of two blocks: The Structure Model, which contains information about domain entities and their relationships, and the Access Queries, which defines how data from the model will be requested. Based on this generic model two conversion steps are defined: A model-to-model conversion, which is a set of rules that converts the generic model into a specific NoSQL data model, and a model-to-text conversion, which converts the specific NoSQL data model created in the previous step to the implementation script on the targeted technology. Diverging from NoAM, based on access queries the framework can also work with some particularities of each NoSQL data model. [16] proposes a set of rules for conversion of a tabular structure, referring to a relational model for implementations in HBase, a column-family model database and a set of rules for the conversion to Hive, a SQL implementation based on Hadoop. The paper presents only the application in HBase and Hive being all the data stored in tabular form, which suggests that the strategy only applies to models of tabular structures like the column-family and relational. Another proposal called NoSE (NoSQL Schema Evaluator), proposes an approach that can identify the application of best practices in NoSQL databases based applications development without necessarily coding them [17]. The proposal suggests that the automated design process produces more efficient database schemas than would be produced in a rules-based approach, a process normally applied in this context. The proposal is limited exclusively to use in the column-family data model and more specifically for use with Cassandra, but suggests that it could also be adapted for use with HBase. Still, it would not be adaptable for other models, such as document-oriented, key-value or graph. The idea of following the best practices for NoSQL is also seen in [18], which presents an Automatic Schema suggestion model for NoSQL Document-Store Databases. It aims to provide a suggestion model which is applicable to NoSQL document-store databases and simplifies the further design process. As the name suggests, the model is completely focused on Document-store databases and based on some previous works from the authors regarding cardinality and some modeling guidelines. A concern frequently pointed out by the authors is the absence of an uniform access strategy to the several existing databases. A proposal for solving this problem is presented by [11], called SoS (Save Our Systems). This strategy also addresses the process of logical modeling required for data models to become compatible with the so-called “meta-layer”, which would be an intermediary step between the application and the databases to be used. Subsequently, an SOS-based programming model is presented in [12] to allow the homogeneous processing of different non-relational databases, more specifically Redis, MongoDB and HBase. The models presented are based on the concept of aggregates but the authors do not present any restriction to the use of other data models. Based on the concept of absence of schema, [19] suggests a methodology of documentation to the variants of the schema in order to list the data storage rules. This methodology is not a database design methodology itself, but only documentation of an already implemented database.

- **Q4. Which of these strategies could be applied to polyglot persistence?**

Most of the select papers propose methodologies focused only on aggregate-based models, and some even to only one of these models. The SOS methodology proposed by [11] mentions that it is possible to apply it to aggregate-based and the relational model, but it is more of a development pattern than a database design methodology. It establishes what they call the “meta-layer” responsible for the abstractions required to use different software or even distinct data models transparently to the rest of the application. However, this does not establish a method for database design, just a more flexible way of using it, which allows the transition between software and data models more easily. The NoAM methodology [13], proposes a methodology for designing databases from the creation of a layer for the definition of aggregates and future conversion for the specific models to be used. By this methodology it is possible to use the same logical model and apply it in databases of the key-value model, document-oriented or column-family. It is a methodology that allows the use of several data models, but only aggregate-oriented ones. It also does not define a way to establish the boundaries between the models, if the application of multiple models in the same application. The Mortadelo Framework [15] by the use of a generic data model, aims to became a framework that allows a single design model to be applied to multiple data models and multiple target software. The information about its access queries in the generic model provide more details about how data will be used and provide more information about how the data should be stored on different models. But once again, is not a framework designed to work with multiple data models on the same generic model, as it never defines any kind of frontier between them. [14] demonstrates an implementation model using polyglot persistence for an e-commerce system, but does not establish a methodology for database design. Its focus is mainly related to the presentation of the challenges regarding the implementation of an application that needs multiple databases.

5. Trends

The results of this systematic review shows that there is a current trend on developing design methodologies capable to provide database designers some guidelines and also bring some standardization to the process among all the different data models available. Is also possible to see that current works are bringing resources to allow the use of multiple databases on the same application, such as [11,13,15]. This multiple databases capability, also called by some authors as heterogeneous data stores, can easily be noticed as a trend by many papers related to the subject, such as [20,11, 21].

6. Conclusion

In this paper we presented a systematic review about the new perspectives regarding NoSQL database design. This work allows us to list the currently most relevant works on this subject and identify which ones are capable of working with more than one NoSQL data model and also the ones capable of being applied in the design of an application that uses polyglot persistence. The selection of relevant papers was made by the application of *Methodi Ordinatio* [8] which allows us to identify them by publication year, number of citations and the impact factor, increasing the efficiency on the process. During the paper selection process, some questions were chosen that can provide useful information for future researches in the area of NoSQL database design, such as the relevant authors, use cases, current techniques and their applicability on a polyglot scenario related to NoSQL databases. The selected papers during the process presents many different approaches aiming to solve the problem of database design for NoSQL data models. Many of these approaches focus on the design for specific data models, such as aggregate-oriented and column-family. Some of them, such as [15] and [12] propose methodologies capable of working with different data models, trying to provide some kind of standardization to the design process among the different models and softwares. Among the papers found is also possible to see that in the last years many new papers were being created trying to solve the NoSQL database design problem. This statement proves that the concern about design strategies for NoSQL databases is still under active research. Even with these approaches being created, none of them show a strategy capable of design databases for an application that uses a polyglot persistence, aiming to persist and use data across different databases for a single application. Some of these approaches provide a solution that supports different models, but does not define a way to delimit the domain between databases what seems to be part of the design strategy.

References

[1]. P. J. Sadalage, M. Fowler. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley Professional, 2012, pp. 25-38.

[2]. V. Holt, M. Ramage, K. Kear, and Heap, N. "The usage of best practices and procedures in the database community". Information Systems, 49.163-181, 2015.

[3]. R. De Virgilio, A. Maccioni, R. Torlone. "Model-driven design of graph databases" in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8824. Springer International Publishing Switzerland, 2014, pp. 172-185.

[4]. R. Shabana, B. Imran, K. Rafaqat. "Challenges in NoSQL-Based Distributed Data Storage: A Systematic Literature Review". Electronics, vol. 8, pp. 488, 2019.

[5]. J. Akoka, I. Comyn-Wattiau and N. Laoufi. "Research on Big Data - A systematic mapping study". Computer Standards & Interfaces, vol. 54, pp. 105-115, 2017.

[6]. J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira, J. Bernardino. "Choosing the right nosql database for the job: a quality attribute evaluation". Journal of Big Data, vol. 2, pp. 1-18, 2015.

[7]. K. Grolinger, W. A. Higashino, A. Tiwari, M. A. Capretz. "Data management in cloud environments: NoSQL and NewSQL data stores". Journal of Cloud Computing: Advances, Systems and Applications, vol. 2, pp. 1-22, 2013.

[8]. R. N. Pagani, J. a. L. Kovaleski, L. M. Resende. "Methodi ordinatio: A proposed methodology to select and rank relevant scientific papers encompassing the impact factor, number of citation, and year of publication". Scientometrics, vol. 105, pp. 2109-2135, 2015.

[9]. D. Buchinger, G. Cavalcanti, M. Hounsell. "Mecanismos de busca acadêmica: uma análise quantitativa". Revista Brasileira de Computação Aplicada, vol. 6, pp. 108-120, 2014.

[10]. R. Cattell. "Scalable SQL and NoSQL data stores". ACM SIGMOD Record, vol. 39, pp. 12, 2011.

[11]. P. Atzeni, F. Bugiotti, L. Rossi. "Uniform access to non-relational database systems: The SOS platform" in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7328. LNCS, 2012, pp. 160-174.

[12]. P. Atzeni, F. Bugiotti, L. Rossi. "Uniform access to NoSQL systems". Information Systems, vol. 43, pp. 117-133, Jul. 2014.

[13]. F. Bugiotti, L. Cabibbo, P. Atzeni and R. Torlone. "Database Design for NoSQL Systems" in Conceptual Modeling, vol. 8824. Springer International Publishing, 2014, pp. 223-231.

[14]. K. Srivastava and N. Shekokar. “A Polyglot Persistence approach for E-Commerce business model.” in Proceedings - 2016 International Conference on Information Science-ICIS, 2016, pp. 7-11.

[15]. A. de la Vega, D. García-Saiz, C. Blanco, M. E. Zorrilla, P. Sánchez. “Mortadelo: A Model-Driven Framework for NoSQL Database Design” in MED, 2018, pp. 41-57.

[16]. M. Y. Santos and C. Costa. "Data models in NoSQL databases for big data contexts" in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9714. LNCS, 2016, pp. 475-485.

[17]. M. J. Mior, K. Salem, A. Aboulnaga, and R. Liu. "NoSE: Schema design for NoSQL applications". IEEE Transactions on Knowledge and Data Engineering, vol. 29, pp. 2275-2289, 2017.

[18]. A. A. Imam, S. Basri, R. Ahmad, J. Watada, and M. T. González-Aparicio. "Automatic schema suggestion model for NoSQL document-stores databases". Journal of Big Data, vol. 5, pp. 46, Dec. 2018.

[19]. E. Gallinucci, M. Golfarelli, and S. Rizzi. "Schema profiling of document-oriented databases". Information Systems, vol. 75, pp. 13-25, Jun. 2018.

[20]. B. Kolev, P. Valduriez, C. Bondiombouy, R. Jiménez-Peris, R. Pau and J. Pereira. "CloudMdsQL: querying heterogeneous cloud data stores with a common language". Distributed and Parallel Databases, vol. 34, pp. 463-503, Dec. 2016.

[21]. C. Zdepski, T. A. Bini, S. N. Matos. “An Approach for Modeling Polyglot Persistence” in Proceedings International Conference on Enterprise Information Systems-ICEIS, 2018, pp. 120-126.