Univerzitet u Istočnom Sarajevu, Ekonomski fakultet Brčko
University of East Sarajevo, Faculty of Economics Brcko

Zbornik radova Ekonomskog fakulteta Brčko
Proceedings of Faculty of Economics Brcko

http://zbornik.efb.ues.rs.ba

Časopis Ekonomskog fakulteta Brčko
Volumen 16, Sveska 1, 2022, str. 11-20
Journal of Faculty of Economics Brcko
Volume 16, Issue 1, 2022, pp. 11-20

Klasifikacija
Pregledni naučni članak
Classification
Review article

UDK: 004.655.3:004.652.4
DOI: 10.7251/ZREFB2216011B

ADVANTAGES AND CHALLENGES OF NOSQL COMPARED TO SQL DATABASES - A SYSTEMATIC LITERATURE REVIEW

Rade Božić[a,1]

[a]Faculty of Business Economics Bijeljina, University of East Sarajevo, Bosnia and Herzegovina

ARTICLE INFO
Received 10/20/2022
Sent to revision 10/21/2022
Accepted 10/28/2022
Available online 12/08/2022

Keywords:
NoSQL
SQL
advantages
challenges

ABSTRACT

Relational data models represent a long-standing standard in database technologies. A unique language for writing queries, fixed schema, vertical scalability, consistency, durability are just some of their basic features. However, the development of information technologies and intensive recording of data leads to the emergence of Big Data, where problems arise when using SQL databases. Large amounts of semi-structured and unstructured data cannot be stored in a simple way, so new systems in data storage are being developed - NoSQL databases. The aim of this paper is to present the advantages and challenges of NoSQL compared to SQL databases through a systematic literature review. Based on the set research questions, the main features where NoSQL databases have advantages were formed, but the challenges they face during implementation and operation in relation to SQL databases were also presented. The research results were presented and interpreted.

Introduction

Relational data models have been for a long time the standard choice in database implementation. They use the concept of relations in a mathematical sense to represent data in a form of tables that contain values. Columns in the table represent attributes that are defined by the name, type and format of the data. The schema defines the structure of the database and the DBMS (Database Management System) is used to implement the relational model through the use of a standard language for writing queries - SQL (Structured Query Language). For this reason, the relational model is often called SQL (de Oliveira, Pessoa, Miuyagi and Eigi, 2022). Relational databases are designed so that a row in one table can be related to one or more rows in another table. For this reason, it is possible to build complex interconnection structures. Queries return an answer in the form of an individual set of rows, which means that during execution they must use JOIN mechanisms that connect tables during execution.

The structure of these databases is fixed and previously known, and their scaling is done vertically - the server on which they are located must improve performance in order to support their expansion. The advantage is also in the use of a single language for writing queries across different databases such as e.g. MySQL, Postgres, Microsoft SQL server, Oracle Database... They also support ACID (Atomicity, Consistency, Isolation and Durability) transactions that must have atomicity (if an error occurs in the transaction, the entire transaction is terminated and the database remains unchanged), consistency (transactions must contain correct data), isolation (simultaneous transactions result as if they were executed one after the other, isolated) and durability (transactions remain recorded even when the conditions for their execution cease) (Haerder and Reuter, 1983). SQL databases for a long time have been the standard choice for data storage. The curriculum and program in a large number of educational institutions was precisely related to mastering the SQL language and working with the database that supports it.

[1] rade.bozic@fpe.ues.rs.ba

However, with the rise of Big Data (large amounts of data), it was necessary to develop new systems that will manage this data in a more efficient way. This means better scalability, performance, flexibility, working with semi-structured and unstructured data, etc. This was a reason for the developement of NoSQL databases. They become a worthy competitor to SQL databases due to their features that eliminate some previously observed shortcomings of storage systems. The aim of this paper is to identify the main advantages of NoSQL compared to SQL databases, as well as the challenges they face during their usage, through a systematic review of the literature. A literature review was conducted because previous papers focused on the differences between these two types of databases without focusing on the advantages and disadvantages (or challenges) of NoSQL databases. Although the technologies are developing rapidly, the goal of the paper is not a comparison of the previous advantages and disadvantages of NoSQL databases compared to the current state, but their presentation through a selected period of time. The procedure proposed by Barbara Kitchenham (Kitchenham, 2004) was used in the research process. The results that have been obtained are presented and interpreted in a separate chapter in the form of answers to the research questions for conducting literature review.

1. NoSQL databases

The term "NoSQL" is made up of two words: "No" and "SQL" implying that this technology excludes the use of the SQL language. It is assumed that the early users of this term tried to indicate the absence of a Relational Database Management System (RDBMS), but due to its prevalence this term has remained until today. It is an umbrella term for all databases that are not based on established RDBMS principles. This means that NoSQL does not represent a single product or technology, but a class of products and a set of different concepts related to data storage and management (Tiwari, 2011).

The first documented use of the term NoSQL was by Carlo Strozzi in 1998 during his visit to San Francisco where he presented his relational database. He used this term because his database was accessed through shell scripts instead of the standard SQL language. The use of this new term caused frustration among SQL users because this way of working was dominant at the time. However, the NoSQL term indicated a desire to find a better way to create queries that would provide programmers with an easier way to understand and read long SQL queries (Fowler, 2015).

NoSQL does not have a formal, generally accepted definition. It represents a form of data storage mechanism that is fundamentally different from RDBMS (E. S. Gupta and Sabharwal, 2015). NoSQL database stands for Not Only SQL. Its mechanism is modeled in such a way that it does not contain tabular relations such as e.g. SQL Server. The data structure is simple and designed according to specific data types so that scientists in database field have can choose the architecture that best suits them. The database is structured in the form of a tree, columns, graphs or key-value pairs. However, a NoSQL database can support SQL-like queries (Ghavami, 2016).

The biggest difference between NoSQL and SQL databases is the support or absence of the SQL language. However, over time it has become difficult to establish a strict DBMS affiliation between these two database systems. Some NoSQL databases have added SQL interfaces to facilitate integration into traditional environments. Also the difference between ACID and BASE (Basically Available, Soft state and Eventually consistent) features is slowly disappearing because NoSQL databases have added ACID features, and SQL databases have achieved consistency to enable their extension. Some NoSQL databases support a tabular structure, while some SQL databases support the organization of data in the form of documents or columns. Therefore, the term NoSQL slowly grew into the previously mentioned term Not Only SQL (Hills, 2016).

NoSQL databases can be categorized into four types of organizational structures (Hills, 2016):

- key-value: a simple organization where each record has one key (attribute) that is indexed for faster searches, and the value is an object for storing arbitrary data whose structure is unknown to the DBMS,
- graph: data storage by defining nodes and edges of graphs,
- document: data storage in a form similar to standard text documents usually has a lot of nesting,
- columns: storing data in the form of table columns as opposed to the traditional organization in the form of rows.

Some of the most famous databases belonging to these structures are shown in table 1 (data taken from the website www.geeksforgeeks.org (2022)).

Table 1: The most popular databases by structure

Structure of NoSQL database	Most popular databases
Key/Value	Redis, Riak
Graph	Neo4J
Document	MongoDB, CouchDB, RavenDB
Columns	Hypertable, HBase, Cassandra

Source: www.geeksforgeeks.org

The four characteristics associated with most NoSQL databases are as follows (Fowler, 2015):

- Absence of schema: database schema describes all possible data and structures in relational databases. With NoSQL databases, schemas are optional, so it is possible to store data without a pre-designed schema.
- Non-relational: in relational databases, connections are established between the tables in which the data is stored. With NoSQL databases, it is possible to store information combined in one record with all the details that belong to it.
- Hardware adaptability: some databases are designed to work with specialized hardware. With NoSQL databases, this is not necessary and cheap servers can be used for easy storage capacity expansion.
- High distributability: With distributed databases, it is possible to store and process data from several machines (servers). With NoSQL databases, server clusters can be used to store data in one large database.

2. Methodology of systematic literature review

The literature review was conducted according to the procedure proposed by Barbara Kitchenham, and consists of three parts: planning the review, conducting the review, and reporting the results. The defined objective of the literature review is to present the advantages and challenges of NoSQL databases in relation to SQL databases. In order to fulfill it, it is necessary to define research questions whose answers will achieve the defined goals. Two questions were defined:

- Research question 1: "What are the advantages of using NoSQL databases over SQL databases?"
- Research Question 2: "What are the challenges of deploying NoSQL databases compared to SQL databases?"

The first research question talks about the advantages of NoSQL databases compared to SQL databases, which is also the reason for their creation, development and use. This also applies to problems represented in relational data models that are solved by NoSQL databases. Another research question concerns the challenges and disadvantages faced by NoSQL databases compared to SQL databases. Do they, despite solving certain problems, create new ones that are potentially bigger than the previous ones? The aim of the paper is not only to present the advantages but also the disadvantages of the mentioned databases.

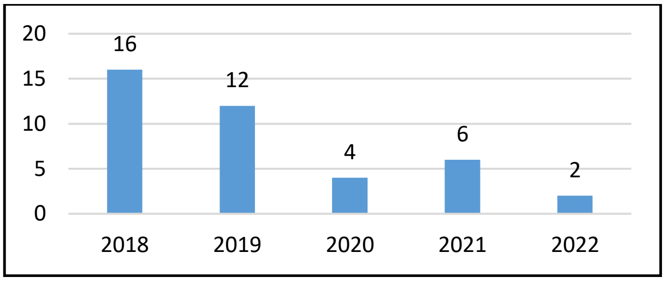
The inclusion criteria are presented through the research strategy, which contains the collection of papers from two scientific research bases: Web of Science (WoS) and Scopus. The papers included in the literature review date from 2018 until the moment of conducting the research. A longer period of time is not covered due to intensive development of NoSQL technologies and data storage systems in general. Only those papers written in English were included, and the types of documents analyzed were articles from scientific journals and papers from scientific conferences (eng. article, conference paper).

The search entry in both databases is: “nosql” and “sql” and (“advantage*” or “benefit*” or “challenge*”). The goal of the search was to cover both terms (both NoSQL and SQL) with the presence of the words "advantage", "benefit" and "challenge" both in the singular and in the plural (use of the * sign). The search was performed by title and abstract. After applying the inclusion criteria, the Scopus database provided 76 results, while WoS provided 49, which makes a total of 125 papers. Exclusion criteria were applied to the obtained results by analyzing each individual paper. Eliminated papers:

- do not contain specific answers to research questions,
- are oriented only to one type of database without reference to the other,
- present literature reviews,
- they do not clearly define what is an advantage and what is a challenge from the mentioned features.

After eliminating process, the Scopus database provided 36, and WoS 24 papers, which makes a total of 60 papers (48% of the total search result). It is important to note that 20 papers were duplicates, so after their elimination, a total of 40 papers remained and were used in the analysis procedure (32% of the total search result). Quality evaluation criteria were not used for elimination in this case.

Chart 1: Number of papers by year



Source: author's research

Chart 1 shows the number of papers by year of publication. The largest number of selected papers was published in 2018 (16), while 12 papers were published in 2019. Significantly smaller number of papers was selected in next three years. It should be taken into consideration that the year 2022 is still in progress.

3. Answers to research questions - results

After the analysis of the selected papers, answers to the research questions were formed. In the first question, 12 features were crystallized in which NoSQL databases have advantages and 14 in which they have challenges compared to SQL databases. There was no paper that was exclusively about the disadvantages of these databases without mentioning any advantages. A large number of papers contained more than one advantage or challenge. Table 2 shows the answers to both research questions.

Table 2: Advantages and challenges of NoSQL versus SQL databases

Advantages of NoSQL databases	Num. of papers	Challenges of NoSQL databases	Num. of papers
Horizontal scalability	17	Absence of SQL language	5
Flexibility and customizable scheme	17	Absence of JOIN operations	4
Convenience for managing Big Data	11	Security problems	3
Query Execution Speed (Performance)	9	Absence of ACID transactions	3
Defining aggregation from the application level	2	Without standards and consistency	2
Base recoverability and fault tolerance	2	Problems in migration from SQL to NoSQL database	2
Absence of SQL language and JOIN operations	1	No query interface	2
Absence of ACID transactions	1	Queries without mathematical background	1
Reduced energy consumption	1	It is difficult to choose an adequate type of database	1
Reducing costs by hiring cheaper equipment	1	Reduced ability to detect anomalies	1
A large number of database types	1	Increased disk and memory usage	1
Constant development and mass use	1	Without indexing large amounts of data	1
		Data redundancy	1
		Supporting is difficult	1

Source: author's research

3.1. The first research question "What are the advantages of using NoSQL databases compared to SQL databases?"

The largest number of papers was about four key advantages of NoSQL databases compared to SQL: horizontal scalability, flexibility and adaptive schema, query execution speed through performance monitoring and convenience for managing Big Data. Other advantages were mentioned in a significantly smaller number of papers, which is shown in table number 2.

Horizontal scalability is a feature that was mentioned as an advantage of NoSQL databases in 42.25% (17) of the total number of analyzed papers. It implies the possibility of expanding the system, network or process if there is a need for it. In this case, it primarily refers to the growth of recording large amounts of data (Big Data) and the need for their storage. Unlike SQL databases that use vertical, NoSQL databases have the ability to expand horizontally with the use of less demanding additional hardware, which further implies lower investment costs. For this reason, it is more difficult to fulfill the ACID principles (Sahatqija, Ajdari, Zemuni, Raufi and Ismaili, 2018). This is important especially when recording time series data where scalability is one of the key features. In a survey conducted in 2017, it was established that 68% of respondents use NoSQL databases when working with time series data (Telnarova, 2018). Herrera-Ramírez, Trevino-Villalobos and Viquez-Acuna (2021) indicated the importance of this feature when working with geospatial data. Pop, Antal, Cioara, Anghel, Sera et al. (2019) talked about the combination of NoSQL databases with blockchain technology, which allows reducing the frequency of registration of energy transactions in the chain by means of scalability. Scalability as an advantage is briefly mentioned by the authors de Oliveira et al. (2022), Somasekhar, Patra, and Raju (2021), Kalayda (2021), Wu (2021), Schreiner, Duarte, and Mello (2020), Syed, Syeda, Prior, Zozus et al. (2020), Lu, Xu and Lan (2019), Khennou, El Houda Chaoui and Khamlichi (2019), Srivastava (2019), Daniel (2019), Bjeladinovic (2018), Zhang and Xu (2018) and Hassani and Ayachi Ghannouchi (2018).

Flexibility and an adaptable scheme implies the absence of a fixed scheme, which enables the storage of semi-structured and unstructured data (de Oliveira et al., 2022). The globally accepted data format is composite and unstructured without limitations, defined type or template. Storing such data can be a challenge. NoSQL databases were introduced because of this (Ul Haque, Mahmood and Ikram, 2018). Flexibility also means adapting to the structure of a new stored record in the database (Bjeladinovic, 2018). Not every DBMS system can provide this capability. SQL databases have a static (fixed) schema that is defined before storing data to support structured data. If there is a need to change the scheme, a big problem arises related to the modification of the tables, which can further cause unsuccessful system operation and high costs. NoSQL has a dynamic schema that does not need to be predefined (Sahatqija et al., 2018) (Somasekhar et al., 2021). This is related directly to designing the schema from the application layer. A large number of studies talk about this advantage (Khennou et al., 2019). Halim and Hutagalung (2022) used a NoSQL database for remote sensing because their system required flexibility. For example. one user wanted to measure temperature and humidity, while another wanted data on brightness, noise and humidity. Each user's request for a different field has just led to the use of a NoSQL database as a solution to the mentioned problem when recording data read by telemetry. This feature also plays an important role in the development of mobile Web technologies, social networks and electronic commerce (eng. ecommerce) (W. Wu, 2021). Flexibility is also mentioned by Kotenko, Krasov, Ushakov and Izrailov (2021), Chouliaras and Sotiriadis (2020), Syed et al. (2020), Srivastava (2019), Lu et al. (2019), Daniel (2019), Hamouda and Zainol (2019), Zhang and Xu (2018) and Cisofret and Ciuciu (2018).

Query execution speed with NoSQL databases has better performance compared to SQL databases. This is also influenced by the fact that all the necessary information can be stored in one document, unlike SQL databases where it is stored in several tables that are interconnected through the use of JOIN-s (Cisofret and Ciuciu, 2018). Zhang and Xu (2018) state that NoSQL databases have better performance in CRUD operations, which affects the ACID capabilities of transactions (as well as the aforementioned scalability). This was practically tested by Balasubramanaian, Jayapal and Janakiraman (2020) by forming two systems with SQL (VirScaffSQL) and NoSQL base (VirScaffNoSQL). They conducted 10,000 CRUD (eng. create, read, update and delete) operations where the VIRScaffNoSQL system needed 120MS to execute, while the VIRScaffSQL system needed 500MS. They also conducted the experiment with 1,000,000 queries where VIRScaffNoSQL required 900MS, and VIRScaffSQL 2,500MS. With this, they confirmed the advantages in the speed of query implementation. Similar testing was performed by Lian, Miao, McGuire and Tang (2018) on MySQL and MongoDB databases, where they determined that the speed of writing and reading data from the database is higher with MongoDB. The experiment was conducted by Chakraborty, Paul and Hasan (2021) where they conducted four queries with different attributes on three databases (MySQL, MongoDB and Cassandra) and determined that on average Cassandra has the highest query execution speed. Herrera-Ramírez et al (2021), Kalayda (2021), Daniel (2019) and Bjeladinovic (2018) also briefly wrote about speed and performance.

Big Data management is directly related to other advantages such as scalability. With the increase in the number of internet users, web applications, e-commerce, social networks and storage capacities, large amounts of data is generated daily and the goal is to record as much of it as possible in order to extract usable information. Before the emergence of this term (Big Data) SQL databases were dominant (Bansal, Soni and Sachdeva, 2022). However, their extensibility was not satisfactory enough to advance NoSQL databases. This also led to the emergence of migrations from SQL databases to NoSQL, causing a large number of problems and challenges (Kim, Ko, Jeon and Le, 2018). Quicker access to a large amount of data is also one of the advantages in addition to solving the problem of intensive growth of the amount of data (Ghule and Vadali, 2018). The management of big data as one of the main reasons for the development of NoSQL databases is also discussed by Herrera-Ramírez et al. (2021), Somasekhar et al. (2021), Kalayda (2021), Wu (2021), Anusha and Usha Rani (2020), Srivastava (2019), Daniel (2019), Minukhin, Fedko and Gnusov (2018).

Defining aggregations from the application level, in contrast to SQL databases, can also be done on the upper layer where the application is located. Data is organized in a such way that it can be directly accessed without complex SQL queries and JOIN-s (J. Wu et al., 2021). If there is a change in data in SQL systems, it is necessary to redesign the entire application, while this is not necessary with NoSQL databases, because it is at the application level that a new structure can be generated (Schreiner et al., 2020). Database recovery and fault tolerance are automatically provided by NoSQL databases. Replication is a key feature of the mentioned advantages (Somasekhar et al., 2021) (Srivastava, 2019).

For some authors such as Daniel (2019), the absence of the SQL language is an advantage because for a certain number of programmers it is an additional effort to master it due to a large number of features that are sometimes unnecessary. The same author also believes that the absence of ACID transactions is an advantage because it enables greater efficiency in distributed systems. He also states that this feature has been supported in the MongoDB database since version 4.0, so listing this as a shortcoming can be considered partially. J. Wu, Ni and Zhi Xiao (2021) state that the absence of JOINs is also an advantage because they are the basis of demanding queries. Mahajan, Blakeney, and Zong (2019) conducted an interesting experiment emphasizing energy consumption expressed in joules (J). They used 100GB of Twitter data for testing by implementing measurements using NSF tools. They measured the energy consumption for identical queries where they found that MongoDB consumes 41.37J as opposed to MySQL (292.09J) and Cassandra (4628.36J). Hassani and Ayachi Ghannouchi (2018) cite as an advantage the choice of many types of NoSQL databases and that this technology is fast growing and focused on development.

3.2. The second research question: "What are the challenges in implementing NoSQL databases compared to SQL databases?"

NoSQL databases are distinguished not only by the advantages they have over SQL databases. There are also challenges that affect their application in practice. This is also the main reason why SQL databases are still used. As this literature review has included papers since 2018, certain challenges have been overcome in different ways. However, a detailed comparison of the current state with the past was not made in this paper.

Most authors consider the absence of the SQL language to be the biggest drawback or limitation of NoSQL databases. Almost every NoSQL database uses its own way of creating queries (eng. access language) which differs from one database to another depending on the model or specific product (Somasekhar et al., 2021) (Hassani and Ayachi Ghannouchi, 2018). This makes impossible to create a unique language that would serve to work with these types of databases. Due to the different approach and data structure, the learning curve of NoSQL databases is steep (Schreiner et al., 2020). This is also the reason for the creation of hybrid databases that allow creating queries in the SQL language over NoSQL databases. This approach was also taken by Kim et al. (2018) where they used Apache Phoenix to send SQL queries to HBase. They noticed problems with denormalization, secondary indexes and JOIN processes. If the data is classified in different databases, there is a problem that occurs while changing database because there is no unique language. This can cause problems in the system, and SQL databases guarantee security with such data storage structures (Jimenez-Peris, Patino, Brondino, & Vianello, 2018).

JOIN operations are missing, which is also considered by a large number of authors as a deficiency directly related to the absence of the SQL language (Sahatqija et al., 2018) (Telnarova, 2018) (Kim et al., 2018). However, some databases such as MongoDB offer certain ways of linking (referencing) based on CRUD operations (Bansal et al., 2022). JOIN operations affect the creation of complex queries but also affect their performance. For this reason, some authors, as already mentioned, consider the lack of these operations as an advantage.

The lack of ACID transactions often forces developers to implement their own code to meet its features, which can lead to the creation of a complex system (Somasekhar et al., 2021). NoSQL databases, through achieving better performance in CRUD operations, caused weaker ACID transaction capabilities (Zhang and Xu, 2018) (Kim et al., 2018).

Security problems relate to lack of encryption, weak authentication and authorization. The heterogeneity of NoSQL databases is precisely the main obstacle that prevents the application of SQL security principles to them (Alomari and Noaman, 2019). N. Gupta and Agrawal (2018) list the following security issues of NoSQL databases in 2018:

- SQL databases have integrated security systems, while NoSQL databases have them at a much weaker level,
- NoSQL database clustering options provide additional challenges to security practices,
- data in NoSQL databases are stored as text without an encryption mechanism,
- passwords in NoSQL databases are encrypted with MD5 or PBKDF2 algorithms, which are not very secure,
- weak authorization and authentication methods for defense against brute force attacks,
- resistance to injection (eng. injection) via JSON, strings, views, etc.

Distributing data across multiple servers is a challenge for system security. Most NoSQL databases lack secure client-server communication and security mechanisms. At the same time, a large amount of unstructured and unencrypted data affects security. Looking at the complete picture, there is room for development (Sahatqija et al., 2018). As the applied technologies are rapidly developing, it is possible to conduct special research related to the solution of the mentioned security problems and what is their status in the current period.

Because of the advantages achieved, migration from SQL to NoSQL databases occured. This brings corresponding challenges and problems when carrying out this process. Kim et al. (2020, 2018) mention denormalization, JOIN operations, support of complex SQL queries, automatic indexing (Telnarova, 2018) and optimization of SQL queries for NoSQL as problems.

Relational models use relational algebra rather than mapping. NoSQL database models have sacrificed the mathematical support that SQL databases have in order to gain freedom when writing queries. Although key-value databases have partially solved the problems of relational databases, they do not have a rigorous mathematical foundation, which results in difficulties in creating complex queries and understanding the connection between data. On the other hand, relational databases can use queries based on relational algebra that can analyze data logically (J. Wu et al., 2021).

Lian et al. (2018) while conducting the experiment mentioned earlier (comparing the performance of MongoDB and MySQL) found that in addition to MongoDB having a higher speed of writing and reading data, it also has a higher usage of disk (hard drive) and memory (RAM). MongoDB's disk size scales incrementally because it uses a pre-allocation algorithm designed to support predetermined data sizes. The purpose of this mechanism is to prevent disk fragmentation to ensure that documents are stored in contiguous blocks which leads to inefficient use of storage space. The experiment also showed that the use of RAM memory is very similar between the mentioned databases, but it was used at a slightly higher level with the MongoDB database.

The absence of standards (such as a table in SQL) and consistency that is not guaranteed represent just one more of the disadvantages of using NoSQL databases (Coșofreț and Ciuciu, 2018) (Somasekhar et al., 2021). Because there is no single NoSQL language, this also affects the limited support and maintenance of the system when working with these databases. In addition, there is also the absence of an interface for creating queries in an easier and simpler way (Hassani and Ayachi Ghannouchi, 2018) (Bansal et al., 2022).

It was stated earlier that certain authors consider the choice between multiple types of NoSQL databases to be an advantage, Bansal et al. (2022) believe that this is still a disadvantage and an aggravating circumstance due to the confusion that this procedure can cause for users. They also add that it is not an easy task to determine the most adequate type of database for solving a specific problem. Chouliaras and Sotiriadis (2020) talk about the problem of anomaly detection in NoSQL systems and propose the RADAR system they developed as a solution. It is based on continuous monitoring of resource utilization (CPU, memory, disk utilization, etc.) with the use of various machine learning algorithms to generate behavioral patterns. SQL databases store normalized and prepared data, avoiding redundancy. With NoSQL, data is stored in collections without interconnection and normalization so that redundancy is possible (Sahatqija et al., 2018).

Based on the analyzed papers, it is observed that they speak more about the advantages of these databases than about the limitations or challenges in their use. It is also important to note that certain authors consider the corresponding feature to be an advantage, while others claim that it is a limitation in the operation of NoSQL databases such as e.g. a large number of structure types, absence of SQL language, ACID transactions and JOIN operations. Also, during the period covered by the papers, certain characteristics of NoSQL were improved, whose presentation and comparison with the achieved results can be the basis for future research.

Conclusion

In order to present the advantages and challenges of NoSQL databases compared to SQL databases, a systematic literature review was conducted. In order to fulfill it, two research questions were designed that served as criteria for selecting papers and identifying advantages and challenges. In the analysis process itself, 40 papers were included that provided answers to the research questions. The answer to the first question crystallized 12 features according to which NoSQL databases have an advantage over SQL databases. The largest number of works pointed to the advantages of Big Data management, flexibility and adaptive scheme, then horizontal scalability, performance, defining aggregations from the application level, a large number of database types, constant development and mass use, etc. As a challenge, 14 characteristics are singled out, where most papers talk about the omission of a unique or SQL language, the absence of JOIN operations and ACID transactions, security, standards and consistency problems, etc. It should be noted that certain authors observe the same characteristic differently. Some consider it an advantage, while others consider it a challenge, such as, for example lack of JOIN operations, SQL language, a large number of database types, etc. Due to the constant development of technology, future research can be conducted in order to look at the shortcomings from earlier periods and the current state related to those shortcomings or challenge.

References

1. Alomari, E., & Noaman, A. (2019). SecloudDB: A unified API for secure SQL and NoSQL cloud databases. ACM International Conference Proceeding Series, 38–42. https://doi.org/10.1145/3358505.3358511.
2. Anusha, K., & Usha Rani, K. (2020). Performance evaluation of spark SQL for batch processing. Advances in Intelligent Systems and Computing, 1054, 145–153. https://doi.org/10.1007/978-981-15-0135-7_13.
3. Balasubramanaian, N., Jayapal, S., & Janakiraman, S. (2020). A contrivance to encapsulate virtual scaffold with comments and notes. International Arab Journal of Information Technology, 17(3), 338–346. https://doi.org/10.34028/iajit/17/3/7.
4. Bansal, N., Soni, K., & Sachdeva, S. (2022). Journey of Database Migration from RDBMS to NoSQL Data Stores. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13167 LNCS, 159–177. https://doi.org/10.1007/978-3-030-96600-3_12.
5. Barbara Kitchenham. (2004). Procedures for Performing Systematic Reviews. Keele University Technical Report.
6. Bjeladinovic, S. (2018). A fresh approach for hybrid SQL/NoSQL database design based on data structuredness. Enterprise Information Systems, 12(8–9), 1202–1220. https://doi.org/10.1080/17517575.2018.1446102.
7. Chakraborty, S., Paul, S., & Azharul Hasan, K. M. (2021). Performance Comparison for Data Retrieval from NoSQL and SQL Databases: A Case Study for COVID-19 Genome Sequence Dataset. International

Conference on Robotics, Electrical and Signal Processing Techniques, 324–328. https://doi.org/10.1109/ICREST51555.2021.9331044.

8. Chouliaras, S., & Sotiriadis, S. (2020). Real-Time Anomaly Detection of NoSQL Systems Based on Resource Usage Monitoring. IEEE Transactions on Industrial Informatics, 16(9), 6042–6049. https://doi.org/10.1109/TII.2019.2958606.
9. Coșofreț, G., & Ciuciu, I. (2018). Superstore sales reporting: A comparative analysis of relational and non-relational databases: Short paper. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10697 LNCS, 98–102. https://doi.org/10.1007/978-3-319-73805-5_10.
10. Daniel, G. (2019). UMLto [No] SQL : Mapping Conceptual Schemas to Heterogeneous Datastores.
11. de Oliveira, V. F., Pessoa, M. A. de O., Junqueira, F., & Miyagi, P. E. (2022). Sql and nosql databases in the context of industry 4.0. Machines, 10(1). https://doi.org/10.3390/machines10010020
12. Fowler, A. (2015). NoSQL For Dummies. John Wiley & Sons, Inc.
13. Ghavami, P. (2016). Big Data Governance: Modern Data Management Principles for Hadoop, NoSQL & Big Data Analytics.
14. Ghule, S., & Vadali, R. (2018). Transformation of SQL system to NoSQL system and performing data analytics using SVM. Proceedings - International Conference on Trends in Electronics and Informatics, ICEI 2017, 2018-Janua, 883–887. https://doi.org/10.1109/ICOEI.2017.8300833.
15. Gupta, E. S., & Sabharwal, N. (2015). Practical MongoDB: Architecting, Developing, and Administering MongoDB.
16. Gupta, N., & Agrawal, R. (2018). NoSQL Security. In Advances in Computers (1st ed., Vol. 109). Elsevier Inc. https://doi.org/10.1016/bs.adcom.2018.01.003.
17. Haerder, T., & Reuter, A. (1983). Principles of transaction-oriented database recovery. ACM Computing Surveys (CSUR), 15(4), 287–317. https://doi.org/10.1145/289.291.
18. Halim, D. K., & Hutagalung, S. (2022). Towards data sharing economy on Internet of Things: a semantic for telemetry data. Journal of Big Data, 9(1), 1–24. https://doi.org/10.1186/s40537-021-00549-0.
19. Hamouda, S., & Zainol, Z. (2019). Semi-structured data model for big data (SS-DMBD). DATA 2019 - Proceedings of the 8th International Conference on Data Science, Technology and Applications, Data, 348–356. https://doi.org/10.5220/0007957603480356.
20. Hassani, A., & Ayachi Ghannouchi, S. (2018). Exploring the Integration of Business Process with Nosql Databases in the Context of BPM. In Advances in Intelligent Systems and Computing (Vol. 736). Springer International Publishing. https://doi.org/10.1007/978-3-319-76348-4_74.
21. Herrera-Ramírez, J. A., Treviño-Villalobos, M., & Víquez-Acuña, L. (2021). Hybrid storage engine for geospatial data using NoSQL and SQL paradigms. Revista Tecnología En Marcha, 34, 40–54. https://doi.org/10.18845/tm.v34i1.4822.
22. Hills, T. (2016). NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software. Technics Publications.
23. Jimenez-Peris, R., Patiño-Martinez, M., Brondino, I., & Vianello, V. (2018). Transaction management across data stores. International Journal of High Performance Computing and Networking, 12(4), 418–427. https://doi.org/10.1504/IJHPCN.2018.096709.
24. Kalayda, A. V. (2021). Promising directions for the development of modern databases. Journal of Physics: Conference Series, 2131(2). https://doi.org/10.1088/1742-6596/2131/2/022087.
25. Khennou, F., El Houda Chaoui, N., & Khamlichi, Y. I. (2019). A migration methodology from legacy to new electronic health record based OpenEHR. International Journal of E-Health and Medical Communications, 10(1), 55–75. https://doi.org/10.4018/IJEHMC.2019010104.
26. Kim, H. J., Ko, E. J., Jeon, Y. H., & Lee, K. H. (2018). Migration from RDBMS to Column-Oriented NoSQL: Lessons Learned and Open Problems. Lecture Notes in Electrical Engineering, 461, 25–33. https://doi.org/10.1007/978-981-10-6520-0_3.
27. Kim, H. J., Ko, E. J., Jeon, Y. H., & Lee, K. H. (2020). Techniques and guidelines for effective migration from RDBMS to NoSQL. Journal of Supercomputing, 76(10), 7936–7950. https://doi.org/10.1007/s11227-018-2361-2.
28. Kotenko, I., Krasov, A., Ushakov, I., & Izrailov, K. (2021). An approach for stego-insider detection based on a hybrid nosql database. Journal of Sensor and Actuator Networks, 10(2). https://doi.org/10.3390/jsan10020025.

29. Lian, J., Miao, S., McGuire, M., & Tang, Z. (2018). SQL or NoSQL? Which Is the Best Choice for Storing Big Spatio-Temporal Climate Data? In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11158 LNCS. Springer International Publishing. https://doi.org/10.1007/978-3-030-01391-2_32.
30. Lu, W., Xu, Q., & Lan, C. (2019). SOOCP: A platform for data and analysis of space object optical characteristic. Information (Switzerland), 10(10). https://doi.org/10.3390/info10100296.
31. Minukhin, S., Fedko, V., & Gnusov, Y. (2018). Enhancing the performance of distributed big data processing systems using Hadoop and PolyBase. Eastern-European Journal of Enterprise Technologies, 4(2–94), 16–28. https://doi.org/10.15587/1729-4061.2018.139630.
32. Pop, C., Antal, M., Cioara, T., Anghel, I., Sera, D., Salomie, I., Raveduto, G., Ziu, D., Croce, V., & Bertoncini, M. (2019). Blockchain-based scalable and tamper-evident solution for registering energy data. Sensors (Switzerland), 19(14). https://doi.org/10.3390/s19143033.
33. Sahatqija, K., Ajdari, J., Zenuni, X., Raufi, B., & Ismaili, F. (2018). Comparison between relational and NOSQL databases. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings, 216–221. https://doi.org/10.23919/MIPRO.2018.8400041.
34. Schreiner, G. A., Duarte, D., & dos Santos Mello, R. (2020). Bringing SQL databases to key-based NoSQL databases: a canonical approach. Computing, 102(1), 221–246. https://doi.org/10.1007/s00607-019-00736-1.
35. Somasekhar, G., Patra, R. K., & Srujan Raju, K. (2021). The Research Importance and Possible Problem Domains for NoSQL Databases in Big Data Analysis. Lecture Notes in Networks and Systems, 215(February 2022), 433–439. https://doi.org/10.1007/978-981-16-1941-0_43.
36. Srivastava, N. (2019). Nosql - A solution for handling big data. International Journal of Engineering and Advanced Technology, 9(1), 7606–7610. https://doi.org/10.35940/ijeat.A9896.109119
37. Syed, S., Syed, M., Syeda, H. B., Prior, F., Zozus, M., Penning, M. L., & Orloff, M. (2020). Document oriented graphical analysis and prediction. Studies in Health Technology and Informatics, 270(August), 183–187. https://doi.org/10.3233/SHTI200147.
38. Telnarova, Z. (2018). Using relational databases for time series data. AIP Conference Proceedings, 2040(November). https://doi.org/10.1063/1.5079066.
39. Tiwari, S. (2011). Proffesional NoSQL. John Wiley & Sons, Inc.
40. Top 10 Open-source NoSQL Databases in 2020. (2022). Www.Geegksforgeeks.Org. https://www.geeksforgeeks.org/top-10-open-source-nosql-databases-in-2020/.
41. Ul Haque, A., Mahmood, T., & Ikram, N. (2018). Performance comparison of state of art nosql technologies using apache spark. In Advances in Intelligent Systems and Computing (Vol. 869). Springer International Publishing. https://doi.org/10.1007/978-3-030-01057-7_44.
42. Wu, J., Ni, D., & Xiao, Z. (2021). N-Tier Soft Set Data Model: An Approach to Combine the Logicality of SQL and the Flexibility of NoSQL. Mobile Information Systems, 2021. https://doi.org/10.1155/2021/5567234.
43. Wu, W. (2021). SQL2X: Learning SQL, NoSQL, and MapReduce via Translation. In SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.1145/3408877.3432541.
44. Zhang, C., & Xu, J. (2018). A unified SQL middleware for NoSQL databases. ACM International Conference Proceeding Series, 14–19. https://doi.org/10.1145/3220199.322.