مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

# NoSQL Databases and Their Role in Addressing Big Data Challenges

**Soad Al-Mahdi Dera**
**Department of Computing, College of Education, University of Zintan, Libya**
Soad.dera@uoz.edu.ly

# قواعد بيانات NoSQL ودورها في مواجهة تحديات البيانات الضخمة

**سعاد المهدي ديرة**
**قسم الحاسوب ، كلية التربية الزنتان ، جامعة الزنتان، الزنتان ، ليبيا**

**Received: 10-10-2025; Accepted: 01-12-2025; Published: 25-12-2025**

**Abstract:**
The phenomenon of big data has become a defining characteristic of the technological age. Consequently, new challenges arise, including the types of data and the scale of the information available; traditional relational databases frequently prove inadequate in addressing these demands, resulting in the proliferation of NoSQL databases. This research presents a comprehensive review of NoSQL-based systems and their methodologies for managing extensive data. Furthermore, this study examines the primary categories of NoSQL databases, their optimal attributes, and their strategies for addressing the contemporary "5Vs" of Big Data: volume, velocity, variety, value, and veracity. Additionally, the research highlights the opportunities presented by NoSQL technologies in emerging fields, such as the Internet of Things (IoT), artificial intelligence, and machine learning, alongside the associated risks, including trade-offs in consistency, security challenges, and the lack of standardization. By integrating classification, comparison, and analysis, this review offers a unified perspective that can aid researchers and practitioners in the effective selection and implementation of NoSQL solutions within Big Data contexts.

**Keywords:** Big Data, NoSQL, Real-Time Processing, Scalability, IoT

**الملخص**
قد أصبح ظاهرة البيانات الضخمة (Big Data) من السمات المميزة للعصر التكنولوجي. ونتيجة لذلك، تظهر تحديات جديدة، بما في ذلك أنواع البيانات وحجم المعلومات المتاحة؛ حيث غالبًا ما تثبت قواعد البيانات العلائقية التقليدية عدم كفاءتها في تلبية هذه المتطلبات، مما أدى إلى انتشار قواعد بيانات NoSQL. تقدم هذه الدراسة مراجعة شاملة للأنظمة المبنية على NoSQLومنهجياتها في إدارة البيانات الضخمة. كما تفحص الدراسة الفئات الرئيسية لقواعد بيانات NoSQL، وخصائصها المثلى، واستراتيجياتها في التعامل مع الجوانب الخمسة المعروفة باسم "Vs5" للبيانات الضخمة: الحجم(Volume) ، السرعة(Velocity) ، التنوع(Variety) ، القيمة(Value) ، والموثوقية (Veracity). بالإضافة إلى ذلك، تسلط الدراسة الضوء على الفرص التي توفرها تقنيات NoSQLفي المجالات الناشئة، مثل إنترنت الأشياء(IoT) ، والذكاء الاصطناعي، والتعلم الآلي، إلى جانب المخاطر المرتبطة بها، بما في ذلك المفاضلات في الاتساق ، تحديات الأمان، ونقص المعايير الموحدة. ومن خلال دمج التصنيف والمقارنة والتحليل، تقدم هذه المراجعة منظورًا موحدًا يمكن أن يساعد الباحثين والممارسين في اختيار وتنفيذ حلول NoSQL بفعالية ضمن سياقات البيانات الضخمة.
**الكلمات المفتاحية:** البيانات الضخمة، قواعد البيانات NoSQL، المعالجة في الوقت الحقيقي، القابلية للتوسع، إنترنت الأشياء

## 1- Introduction

The field of Big Data is experiencing rapid growth and has become integral to almost every industry in contemporary society. It is embraced by businesses and academics to enhance the services, enable business decisions, and develop new products [1]. With volume and speed of data come numerous opportunities, but some new issues for data storage and processing [2].

Relational databases served adequately for numerous years; however, they are inadequate for the demands of contemporary Big Data. They struggle to manage excessively large datasets or data presented in diverse structures. Their static schemas also make it difficult to change easily [3],[4]. NoSQL databases were created to solve some of these challenges. They have variable data models, scale to many servers, and support many data formats. Key–value, document, column-family, and graph databases are gaining popularity in e-commerce, IoT, and social media [5]. They are faster and easier to transition for real-time applications.

مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

NoSQL databases have some issues as well. Consistency, security, and undefined standards issues have been identified by research [6]. Meanwhile, recent research suggests their increasing importance in fields such as AI and machine learning, where rapid and versatile access to data is greatly needed [7]. This study examines the landscape of NoSQL databases in the context of Big Data, highlighting their various types, features, future potential, and existing challenges.

## 2- Research Aims

The primary objectives of this research are delineated as follows:

- To classify the main types of NoSQL databases and briefly describe their key characteristics.
- To investigate how NoSQL databases respond to the fundamental "5Vs" of Big Data: volume, velocity, variety, and value.
- To offer a comparative analysis of the benefits and drawbacks of major NoSQL models.
- To explore how NoSQL facilitates opportunities in emerging fields such as IoT, AI, and machine learning.

## 3- Methodology

This paper utilized a narrative literature review approach. Peer-reviewed journals published from 2013 to 2025 were sourced from reputable electronic libraries, including IEEE Xplore, Springer, MDPI, Elsevier, arXiv, and Google Scholar. The review focused on research that examined the main types of NoSQL databases (key–value, document, column-family, and graph) and their application in Big Data management. The selected articles discussed aspects such as scalability, performance, emerging market opportunities, and ongoing challenges. The studies were meticulously analyzed, compared, and synthesized to contextualize the strengths, weaknesses, and research limitations associated with the use of NoSQL technologies in Big Data environments.

## 4- Relevance of the Study

The significance of this research lies in its comprehensive approach. Unlike other studies that examined aspects such as scalability, storage, or security in isolation, this paper integrates classification, systematic comparison, and analysis into a unified framework. In particular, it presents a systematic comparison of the main types of NoSQL databases based on data models, performance, scalability, use cases, advantages, and disadvantages. This hybrid methodology aids scholars and practitioners in comprehending the importance of NoSQL in tackling Big Data challenges and provides valuable guidance for decision-making and implementing appropriate solutions in modern environments.

## 5-Big data

The concept of Big Data (BD) is characterized by an extensive volume of data that is both intricate and ever-changing, rendering traditional management techniques ineffective, as defined by [1]. Furthermore, it is described as data that cannot be stored or utilized if it surpasses the capacity of standard systems; thus, such data is classified as BD. The demand for robust applications in contemporary industries is perpetually escalating the volume and growth of data. Technology consistently enables the extraction of data for commercial objectives [2].

The phrase "big data" is generally employed when discussing the difficulties associated with handling exceptionally large data sets. This term denotes data that is extensive, varied, intricate, and requires significant time to process. Numerous challenges arise from geographical and mathematical fields and technologies, which encompass aspects such as capture, curation, storage, search, sharing, transfer, analysis, and visualization. Nevertheless, there are not only challenges to consider. Big Data presents several benefits compared to traditional structured databases. It facilitates the management and analysis of diverse data from a single unified source. Large and intricate data sets that frequently obstruct standard data processing can be further developed, enabling the integration of information dispersed across various databases [8].

### 5.1- Types of big data

There are three primary classifications of data within the realm of Big Data. These categories include structured data, semi-structured data, and unstructured data.

#### 1- structured data

Data that can be effectively analyzed is referred to as structured data. This type of data is typically stored in a well-organized database. It pertains to any information that can be formatted into tables consisting of rows and columns within SQL databases. Such data can be readily aligned with predefined fields and includes relational keys. In contemporary times, the processing of this kind of data is regarded as one of the most sophisticated and uncomplicated approaches to information management. Relational data serves as an illustration of structured data.

**2- unstructured data**

Unstructured data does not fit well within a conventional relational database due to its lack of organization according to a predefined structure or absence of a predetermined data model. Consequently, alternative platforms have emerged for the management and storage of unstructured data, which is increasingly prevalent in IT systems and employed by organizations across various analytics and business intelligence applications. Examples include Word documents, PDFs, text files, and media logs.

Figure 1 depicts the essential distinctions between structured data, which is arranged in predetermined tables consisting of rows and columns, and unstructured data, which does not conform to a specific format or model.

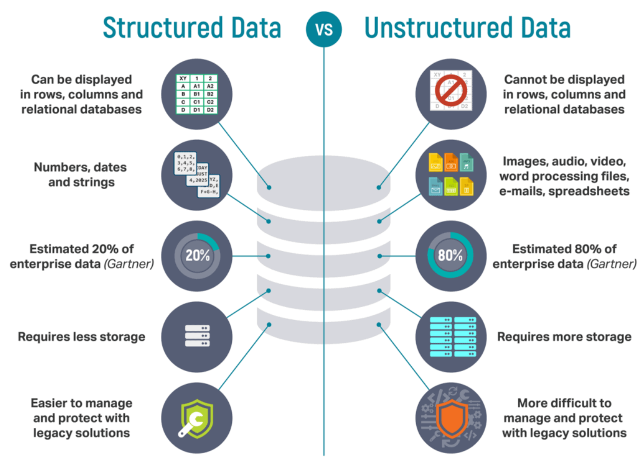


Figure 1: Comparison between structured and unstructured data formats

**3- semi-structured**

Data that is not housed within a relational database yet possesses specific organizational traits that enhance analysis is referred to as semi-structured data. This type of data is designed to optimize storage, although certain processes may be accommodated within relational databases, which can prove challenging for some forms of semi-structured data, such as XML data.

**5.2- Properties of Big Data (The 5Vs)**

Based on the literature review presented in **[9],** big data is characterized by a set of properties often referred to as the 5Vs. The fundamental characteristics include:

- **Volume:** Measures the amount of data available to an organization, which does not necessarily have to own all of it as long as it can access it.
- **Velocity:** Measures the speed of data creation, streaming, and aggregation.
- **Variety:** Measures the richness of the data representation – text, images, video, audio, etc.
- **Veracity:** Measures the understandability of the data – biases, noise, abnormality, etc.
- **Value:** Measures the usefulness of data in making decisions.

## 6- Overview of NoSQL Databases

### 6.1 Categories of NoSQL Databases

مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

NoSQL systems handle data storage and retrieval using various formats, which can be classified into four main categories of databases:

**1- Key value store:**

Key-value stores are one of the simplest NoSQL databases to work with, especially from an API perspective Using such databases allows a client to fetch a value of some key, insert a value for a key in the database, and removes a key from the database. The data blob serves as a unit for storage of the value. The database does not analyze or care about its content – this is the responsibility of the particular application. Since key-value databases depend on primary key access, their performance is usually good, and they are easy to increase when necessary [3].

Key-value stores are pairs of unique keys and the corresponding values maintained in separate hash tables. These values may be string, integers, array or object, which eliminates the need for rigid form models in data storage. Searching in these types of stores can be done solely based on key, and searching by value is not feasible. Since they work with in-memory data, they are employed for enhancing the performance of complicated SQL queries. they are suitable for applications that use a single key to access data such as online shopping carts, user profile configuration, and web session information **[10].** An example of this structure is illustrated in Figure 2

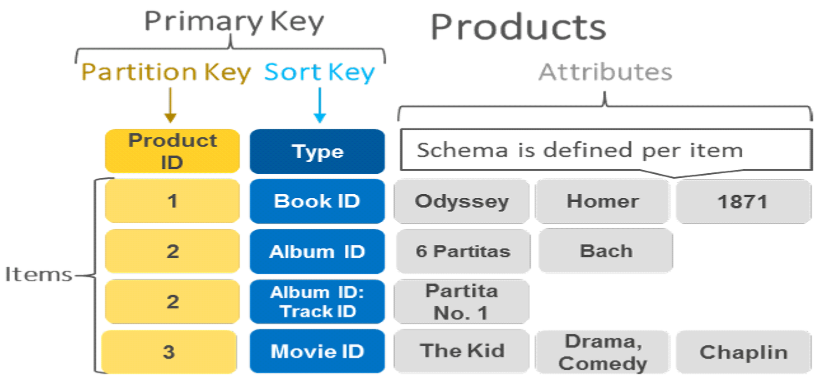


**Figure2:** An example of data stored in key-value pairs.

**2- Document stores**

Document database is a NoSQL database that is designed to store and retrieve data in the form of JSON-like documents. JSON, or JavaScript Object Notation, is an open standard data interchange format that humans and machine-readable can easily read. Documents in document-oriented databases are like documents in relational databases but more flexible because they are less structured in nature. Standard document formats include XML, PDF, JSON, etc [11]. in relational databases, there are unused and blank fields, while same-database records have common data fields. In document stores, though, one document might both share and not share similar and dissimilar data. Each document is assigned a unique key that will act as its identifier. It is straightforward to add new properties to already established documents or save new documents with multiple properties at runtime [12].

Document stores are superior to key-value stores because they store key-value pairs as documents. They are optimal in circumstances where data does not have to be stored in uniform tables but rather as documents of particular qualities. Document stores are optimal whenever the domain model can be broken down and spread across many documents. They should, however, be avoided when the database must support elaborate relationships or normalization. They are extensively used in content management systems [13]. One such representation is given in Figure3.

مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

| Key | Document |
|---|---|
| 101 | {<br>"ID": "1001",<br>"ItemsOrdered":[<br>{<br>"ItemID": "1",<br>"Quantity": "2",<br>"cost": "1000",<br>},<br>{<br>"ItemID": "1001",<br>"Quantity": "2",<br>"cost": "1000",<br>}],<br>"OrderDate": "05/11/2019"<br>} |
| 102 | "ID": "1002",<br>"ItemsOrdered":[<br>{<br>"ItemID": "2890",<br>"Quantity": "11",<br>"cost": "10000",<br>}]<br>"OrderDate": "05/11/2019"<br>} |

**Figure 3:** An example of a document data model

### 3-Graph stores

A graph database is a NoSQL database for data storage that consists of complex connections and relationships. Data in a database is represented as edges and nodes, where nodes are entities and edges are relationships between entities. Nodes and edges consist of objects which are nested key-value pairs. Keys and values can be declared in a schema to make it easy to define more complex constraints.

Twitter has a significant number of user relationships to support its tweet-following functionality. Graph databases are suitable for applications that come with location-based services, modeling the knowledge graph, and pathfinding issues, like for navigation systems. They are applied in recommendation systems and other applications with need for dealing with complicated relationships. Property graph databases are better for managing large-scale relations among numerous nodes, whereas RDF may be applied to particular information stored in a graph. FlockDB is appropriate to manage simple one-hop neighbor relations with high scaling requirements [12].

an example of a social network graph. With the edges (relations) and nodes (people), you can discover who the "friends of friends" of an individual are, i.e., the friends of Howard's friends. An example of this structure is shown in Figure 4

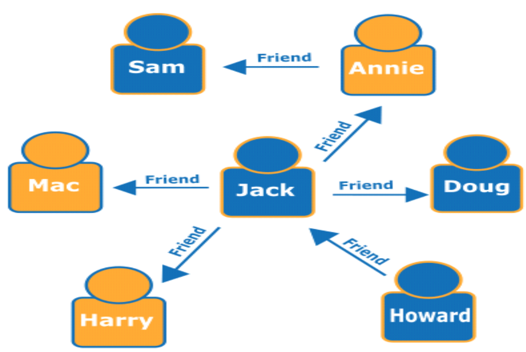


**Figure4:** An example of a Graph stores model

### 4- Column-Oriented Databases

With respect to their data model, column-family stores maintain schema-less column families, each of them including a subset of columns, and each row refers to a multitude of columns. Notwithstanding the name confusion between columnar databases and column-family ones, the former are designed to store tables partitioned by column, whereas the latter maintains schema-less column families, each of them including a subset of columns and each row referring to a multitude of columns [14].

مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

Although columnar databases share the concept of column-by-column storage with row-based databases with column-spanning, column stores do not store data in tables but instead store data in large, distributed architectures In column stores, each key is associated with one or more attributes (columns). A column store stores its data in such a way that it can be stored quickly with minimal I/O activity. It provides high scalability of data storage. The data stored in the database is based on the order of the column family. Column-oriented databases are suitable for data mining and analytics applications, where storage mode is best for frequent operations on data Some notable DBaaS providers that use column-oriented databases are described below [13]. Figure 5 provides an example of this structure

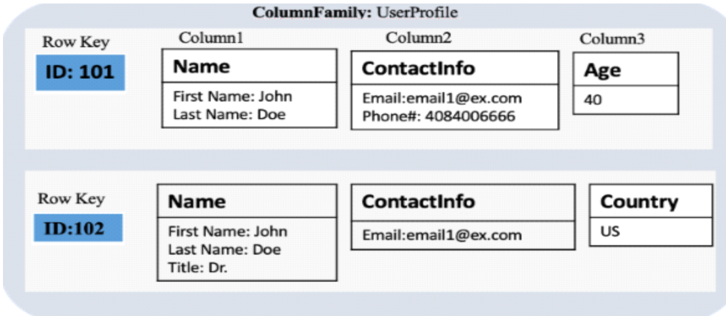


**Figure 5:** An example of the kind of data you might store in a wide-column data store

To provide clarity, Table 1 compares the main categories of NoSQL databases side by side, summarizing their data models, performance, scalability, use cases, advantages, and limitations.

**Table 1**: Comparison of NoSQL Database Types (created by the author)

| Type | Data Model | Performance | Scalability | Common Use Cases | Advantages | Limitations | |
|---|---|---|---|---|---|---|---|
| Key-Value Store | Key–value pairs | Very high for simple lookups using the key | Excellent horizontal scalability | Session storage, shopping carts, user preferences | Simple, fast, handles very large datasets efficiently | Not suitable for complex queries or relationships among data | |
| Document Store | JSON or XML documents | High for reading and writing documents | Excellent horizontal scalability | Content management systems, web applications, IoT | Flexible schema, supports diverse data types | Performance drops with complex, multi-level queries | |
| Column-Family Store | Columns grouped in families | Very high for large-scale reads and writes | Outstanding horizontal scalability | Data analytics, recommendation systems, real-time storage | Ideal for analytical queries, efficient data compression | Complex to design and manage data models | |
| Graph Database | Nodes and edges | Strong for complex relationships | Limited compared to other NoSQL types | Social networks, recommendation engines, path analysis | Excellent for modeling relationships, flexible structures | Not suitable for statistical analysis or very large sets of unconnected data | |

مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

**6.2-Key Features of NoSQL Databases**

NoSQL databases offer a range of powerful features that distinguish them from traditional relational database systems. These core characteristics, which are designed to meet the demands of modern, large-scale applications are illustrated in Figure 6

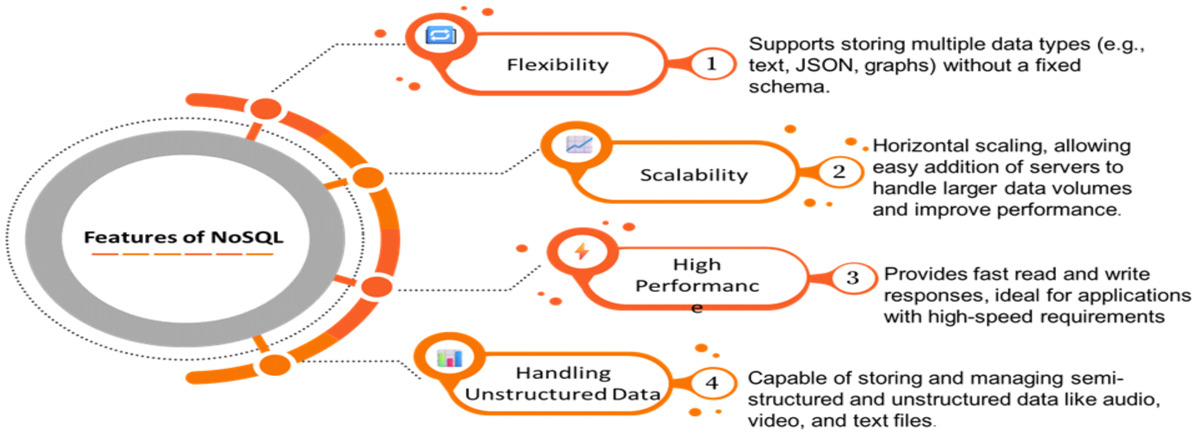


***Figure 6:*** *The primary features of NoSQL databases (*created by the author*)*

Figure 6 illustrates the primary features of NoSQL databases. These include: (1) **Flexibility**, which allows for a schema-less data model; (2) **Scalability**, referring to the ability to scale horizontally to handle large volumes of data; (3) **High Performance**, often resulting from the distributed nature of the system; and (4) the capability of **Handling Unstructured Data**, making them suitable for diverse data types

**7- Related Work (Previous Studies)**

- **Price & Baker (2021),** ***NoSQL Databases and Big Data Query Processing: Comparative Analysis***

The study provides a comparative critique of the function of NoSQL databases in storing and handling enormous amounts of data with specific focus on the exponential increase in data volume, speed, and diversity in the modern digital age. Multiple types of NoSQL databases are compared and contrasted, such as document stores, column-family, key-value, and graph databases. All the systems are compared in terms of their limitation, scalability, query performance, and schema flexibility. To manage enormous data sets efficiently, integration with distributed processing drivers such as Hadoop and Spark is also addressed in the article. Through elaborative review of existing scholarly research and actual application instances, the research offers practical comparisons that signify the strengths and limitations of every NoSQL alternative, leading database architects and decision-makers to the best big data solution.

- **Dritsas & Trigka (2025),** ***Database Systems in the Big Data Era: Architectures, Performance, and Open Challenges***

The authors examine the transition in database management systems in the time of Big Data, emphasizing architectural developments, performance issues, and ongoing problems. They conduct a thorough and exhaustive survey of the transition from legacy relational database management systems (RDBMS) to contemporary models such as NoSQL, NewSQL, and cloud-native databases. Their research documents the impetuses, trade-offs, and innovations driving these transitions. The paper categorizes databases according to data models (relational, key-value, document, column-family, graph), deployment models (on-premises, cloud-native, hybrid), scalability mechanisms (horizontal, vertical, adaptive), and consistency models (strong, eventual, hybrid). The research also considers core performance attributes like throughput, latency, fault tolerance, and cost efficiency to study whether they actually work in real-world environments. The study finds prevailing issues like data heterogeneity, security, and interoperability, listing important directions for future research. The results point to the need for new and innovative database designs to deal with the uncharted volume, velocity, and variety of Big Data, and for a general grasp of the field so as to catalyze future research on database technologies.

- **Ali et al. (2023), *A State of Art Survey for Big Data Processing and NoSQL Database Architecture***

The authors present a state-of-the-art survey exploring the capabilities of NoSQL technology in the context of Big Data processing, highlighting its increasing adoption over traditional SQL databases. The paper emphasizes the benefits of NoSQL, such as design simplicity, horizontal scalability, and enhanced availability, attributing its growing recognition to its schema-free data model, which is superior for managing large volumes of structured, semi-structured, and unstructured data. The study introduces the fundamental concepts of NoSQL, reviews relevant literature, and categorizes various types of NoSQL databases. It also provides arguments for and against the use of NoSQL. A simple SWOT analysis is performed to analyze the known advantages of NoSQL and to differentiate between SQL and NoSQL approaches. The paper concludes with comparative studies of NoSQL databases and aims to broaden the scope of previous work by offering a comprehensive overview of NoSQL's role in addressing Big Data challenges.

- **Faridoon & Imran (2021), *Big Data Storage Tools Using NoSQL Databases and Their Applications in Various Domains: A Systematic Review***

Researchers undertake a systematic literature review (SLR) to present NoSQL big data storage technologies and their deployment in different disciplines. The research deals with the challenge of data exponential growth that cannot be managed by traditional storage systems because of concerns such as schema rigidity, cost, and complexity. The main aim of this SLR is to develop a roadmap for NoSQL big data storage technologies, assess existing evidence, and identify research advancements. The review categorizes chosen papers (published between 2015 and 2020) as motivations for big data storage, employed NoSQL techniques, and key applications across fields. The paper also pinpoints existing challenges and sketches out an outline of future work plan for the big data storage area in NoSQL databases. The main findings reveal that NoSQL databases provide the solutions to scalability, flexibility, and managing heterogeneous data and thus of paramount importance in current big data environments. The authors aim to inform data scientists and researchers about learning and creating appropriate storage frameworks, discussing the role of NoSQL technologies, their limitations, and establishing future research agendas.

- **Hassan (2024), *Big Data, Big Decisions: Choosing the Right Database***.

This paper describes the most important choice of having the most suitable database solution in the era of big data, focusing on using traditional SQL databases or NoSQL databases. The writer scrutinizes a comprehensive literature review comparing the nature, pros, and cons of both databases. The research gives a comprehensive overview of how companies can make effective choices to fulfill their needs during the era of big data. Evidence suggests that although SQL databases are best suited for structured data and ACID transaction management, their inflexibility and vertical scaling shortcomings become an issue to the dynamic nature of big data. However, NoSQL databases introduce flexibility, horizontal scaling, and support for unstructured and semi-structured data, which is best suited for today's big data applications. The article enlightens the reality of the fact that NoSQL databases like Graph, Key-Value, Columnar, and Document stores are becoming increasingly significant because of their performance and agility in high-data-volume and high-data-velocity systems. It also points to factors like data interaction, speed, agility, suitable type of data, fast development, JSON support, scalability, database types, data recovery, and data security, comparing SQL with NoSQL on these grounds. The research concludes that whether to use SQL or NoSQL will be dependent on the particular data attributes and requirements of an application, with NoSQL as a strong contender in response to the dynamic nature of the big data era.

Previous research in NoSQL databases and Big Data has proven beneficial, but only in one aspect of the situation. Some authors focus just on structure or performance, while others compare NoSQL with SQL or discuss storage op tions in a particular context. Missing is a broader summary that connects these disparate pieces.
In particular, there aren't many studies that clearly classify NoSQL kinds and examine how they handle the five ma in Big Data problems: volume, velocity, Variety, Veracity and value.

Furthermore, not many studies have connected NoSQL to modern applications like machine learning, IoT, and AI under a single framework.

By offering a comprehensive literature analysis that connects the categorization, comparison, potential, and

constraints of NoSQL databases for Big Data systems, this work aims to close the gap.

## 8- Challenges Addressed by NoSQL in Big Data

مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

**Handling Large Volumes of Data:** Efficient storage and management of vast amounts of data. The main distinguishing element of Big Data settings is the massive amount of data.Big Data installations store an d manage large amounts of data.This is where conventional relational database management systems (RDBMS) fa ll short.It advances beyond the ability to store data and manage fast data streams entering databases.

The emergence and increasing popularity of NoSQL systems is intimately related to the Big Data phenomena. Growing storage demands lead to the improvement of cheaper commodity servers [14،15]. As opposed to having several high-power machines, companies start to think about distributing the data on a bigger count of less-powerful machines. Such a solution is possible with NoSQL systems that are designed to be deployed on a rather big potential number of nodes. NoSQL database systems provide a flexible design alternative to the rigid schema of RDBMS. The increase of social media, mobile users, and Internet proliferation have raised the need for new data storage systems no longer in reach of the traditional RDBMS [4]. In answer to this growing challenge, NoSQL data stores are born. They were shaped with the goal to store and handle vast amounts of data. Social networks like Facebook, LinkedIn, Twitter, etc., deal with enormous datasets and are designed using NoSQL systems. However, NoSQL systems bear with them particularities that differentiate them from the traditional RDBMS. Appling such a system in a heterogeneous context targeting specific needs remains an issue. Enterprise model is yet managed through RDBMS, while needed for combination with big data analysis tools like Hadoop. Also, transferring data from one database type to another requires data mapping operations at the model (schema) and instance levels. Import/export capabilities in both systems are rarely standardized and limited to a bunch of standard formats [16]. Moreover, NoSQL schema can be quite different from the relational schema, which makes impedance miss-matching issues unsolvable even with the use of traditional mapping tools.

1- **Supporting High-Velocity Data Streams:** Real-time processing of streaming data

In today's data-driven era, organizations are turning to stream processing platforms to handle vast and fast-moving data streams so they can make decisions in real-time. Technologies empower businesses to extract meaningful insights, identify emerging patterns, and respond to trends in real-time – essential for staying competitive in the dynamic world [17].

While technologies like Apache Kafka, Apache Flink and Structured Streaming have developed capabilities of power treatment, previous implementations would require modified growth with Java or Python which involves high maintenance costs and low flexibility. To deal with the issues that are brought about by these, new strategies have been created to reduce complexity and make development easier. For example, SQL-powered power treatment engines like KSQLDB and fine SQL offer a declarative interface that can reduce development time and simplify the implementation of real-time applications [18]. More recently, benchmarking has also shown that these systems provide physical abstractions and efficient state processing with strong guarantees at high scale. Cloud-Native Architecture like Amazon can stream for Apache Kafka (MSK) (MSK) so that organizations can scale power treatment solutions with zero infrastructure.

2- **Managing Data Variety and Complexity:** Flexible models to handle diverse data formats.

Traditional databases store data in a tabular format, normalized and structured with pre-defined data models. The strictness of these models requires data to be structured conforming to a fixed schema. However, data available in the real world comes in different formats and types, including unstructured data (free text, images, XML files, video, etc), semi-structured data (e.g., different fields for the same entity or entities), or custom serialization formats (e.g., protocol buffers, Avro or Thrift), which are generally hard to fit into fixed models [19, 20].

Multiple models have been proposed (column-family, document-store, and graph-based databases) to better handle the variety of data, following two main strategies: provide a flexible schema and/or extend current fixed models. A common point among these models relies on the ability to store nested information and expand this nesting to accommodate most of the variety in data models besides no/relational aspects of the storage itself [16]. Strictness on fixed models becomes a barrier when facing data variety and makes adaptation non-viable as costs increase. Each industry has its own data complexities, so it is very difficult a priori to accommodate such complexities with fixed models [21].

By the beginning of the decade developers were already capable of handling solid workload on a non-relational way, however, fixed models of traditional databases were incapable of coping with the growing complexity and variety of the data. Research in the then so-called NoSQL databases emerged, with the main goal of proposing flexible models for industrial needs that could handle the array of data available [19]. Since then, a large number of models has been proposed to support niche-specific data structures. Oracle, in particular, introduced nested data capabilities while maintaining relational storage [22].

## 9- Opportunities Provided by NoSQL Databases

### 1- Real-time Data Processing

In the emerging world of big data, real-time data processing has become a hot issue in both the academia and the industry. In general, when the data is collected from the web and social media feeds, real-time data processing is useful. As the map reduces expensive loop over the distributed file system, the proper indexes setup on the NoSQL databases and on some indexing mechanism has been superseded to the real-time data process [23].

NoSQL databases have become the popular storage system for data scientists for storing, preprocessing the data and for better analysis. Some of the popular NoSQL databases are Cassandra, CouchDB. These are the flexible schema-free storage systems. In the NoSQL database, the queries use the secondary indexes, map-reduce and full table scan approach to retrieve the data from the tables [24].

In the relational database system, the queries run very fast due to high normalization and relationships between the tables but as the schema is flexible in the NoSQL system some of the indexes/joins that are not possible in the NoSQL system, this could lead to run the queries slow while hitting over the huge amount of data [25]. As Hadoop is used for the offline batch processing of the data, it is not efficient to run the queries slow. However, when the small data is stored in the NoSQL database, then it is not providing the great advantage to the data scientist. There must be a proper utilization of the storage systems and its indexes when the data scientist queries over the huge quantity of data [26].

### 2-Enhancing Scalability for Distributed Systems

In recent years, a new generation of distributed data stores has emerged to perceive the technological challenges of large-scale data processing and cloud infrastructure. Design choices in inter-customer distribution design, legislature, and consistency control can severely impact the effectiveness of the hosting system. A document-oriented database is needed, where the unit of storage and retrieval is an aggregation of key-value structures [27].

This aggregation is generally a file in JSON or XML format. These document-oriented databases are engineered systems that are built from the combination of small packages and libraries. In general, it is non-trivial to determine the impact of these design choices on the performance of a system a priori. To understand the key performance trade-offs in the hosting of document-oriented databases, a thorough analytical study is undertaken. This study uses the concept of File Design to model the operation of a distributed document-oriented data store and render fundamental performance bounds on their three operational concerns: data traffic, operational load, and the safety of storage reserved for the accumulated data store. The aim is to help in the design of more effective distributed NoSQL stores and to offer cloud service providers the resources to design more principled Service Level Agreements [28].

### 3-Supporting Modern Applications (IoT, AI, Machine Learning)

In today's modern technology world, NoSQL databases are the cornerstone of smart modern applications especially in IoT (Internet of Things), AI (Artificial Intelligence) and ML (Machine Learning). Compared to traditional relational databases, NoSQL systems provide schema flexibility, scalability for growing data and efficient handling of heterogeneous data structures. This makes NoSQL ideal for environments where data characteristics evolve more often and randomly [29].

Managing IoT data that is typically unstructured and generated in enormous quantities on a real-time basis demands nimble storage systems. [29] document-oriented NoSQL designs like MongoDB and Couchbase offer efficient storage and retrieval with less rigid constraints of relational schemas. Dynamic support for emerging data types ensures IoT platforms are agile and scalable.

Security is another significant concern as data volume and sensitivity increase. According to [30], integration of AI driven anomaly detection mechanisms within NoSQL framework provides excellent improvement in identifying suspicious activities. Through machine learning models, NoSQL databases can detect anomalous patterns in real time thus enhancing the cybersecurity of intelligent systems.

مجلة الجبل للعلوم الإنسانية والتطبيقية
Aljabel Journal of Humanities and Applied Sciences
Volume 6 - Issue 2 - 2025 - Pages: 25-37

Performance tuning in NoSQL databases has attracted attention, and machine learning techniques are increasingly being explored to optimize query latency and system throughput in real-time applications.

Each of these points to the strategic importance of NoSQL databases as a robust, flexible and secure foundation for the next generation of intelligent, data-driven applications across verticals.

## 10 - Limitations and Challenges of NoSQL

### 1- Consistency vs. Availability Trade-off

Database systems are omnipresent in today's Internet-based applications. NoSQL databases have gained wide acceptance in the last decade, providing new ways for storing information. One of the implementation keys of consistency in Couchbase is to use indexes and query results [31]. The programmer should refrain from using a consistent view of the data in a distributed environment [26].

In scenarios where a particular operation encounters a failure and requires retrial, especially if it falls within a minority category of conflicting feedback, adopting intelligent conflict resolution strategies becomes a practical approach. These methods aim to reduce the likelihood of inadvertently accepting information that directly conflicts with other received data [32]. Addressing this situation presents a significant challenge, particularly in the necessity of clearly defining the essential invariants for the eventual consistency model implemented within the system architecture [33].

### 2- Security Concerns

NoSQL databases are highly scalable data storage solutions with the ability to store and manage huge amounts of structured, semi-structured, and unstructured data. NoSQL databases provide a flexible, schema-less structure alternative to the previous one-size-fits-all rigid design. NoSQL flavors have been increasing since the term NoSQL became officially recognized in 1998. Some of the flavors of NoSQL are Document Store, Key-value Pair, Version Control, Column-family, Event Store, Multimodal, Graph Database, Spatial Database, etc. However, the theory foundations of NoSQL were not comprehensive in the literature. This is responsible for the profound disagreements in the field and the difficulty in coming to agreement.

Clarifying the CAP theorem, the Jepsen test, the BASE properties of NoSQL databases, the schemes of classification, the paradigm of aggregation, and other fuzzy or often misinterpreted concepts in NoSQL systems is one of the first steps toward standardization and unification of the NoSQL world in research as well as in industry [14]. With operations further reaching to ensuring invulnerability and confidentiality of databases, this review assists in understanding the issues of big data platforms more profoundly.

### 3- Lack of Standardization

The exact definition of NoSQL databases is still undefined and debatable. The fundamental contribution of this work is twofold. Firstly, a benchmark is conducted in order to compare different NoSQL databases and hence distinguish their characteristics. A survey about 20 experts from different places and backgrounds was carried out to have a critical overview of NoSQL databases. Secondly, the major areas of ambiguity and confusion around NoSQL databases and their related concepts are presented. This is motivated by the fact that the lack of theoretical background and fragmented sources found in the literature create a lack of consensus on some NoSQL concepts. Therefore, the results of the survey are regarded at as an interesting source of new comment and interpretation about NoSQL databases, their features, structures, evolution and appropriate fields of application [14]. Traditional relational database management systems (RDBMS) are not convenient for large-scale data management, especially in distributed systems. In this sense, the design of NoSQL databases responded to the need for horizontal scalability and high availability. The demanding requirements of the new big data intensive era raised the need for flexible storage systems capable of handling huge volumes of unstructured data. NoSQL databases come to be a powerful and diverse set of databases tailored to the specific needs of industry and business, such as document databases, wide columns databases, graph databases, key-values stores, and time-series databases. However, despite the many years since the birth of NoSQL databases, the concept of NoSQL is still cloaked in ambiguity and confusion [34].

## 11- Conclusion and Future Outlook

In the future, NO SQL databases will undeniably have an even more active part with the rise of technology like Artificial Intelligence (AI), Internet of Things (IOT), Machine Learning (ML). Fast, Flexible and Adaptable require fast, flexible, and Adaptive solutions for Data which is why NoSQL is a good fit for Big and Varied Data.

But there were still obstacles to overcome. One of the major obstacles is that No- SQL systems are not consistent in the implementation of the protocols and APIs, making integration and migration hard. Security is also a big issue, more so in the age of AI and IoT where sensitive data is ubiquitous. There will be challenges for researchers and developers to come up with new methods to protect data that do not offer access and performance benefits.

Scalability, though bigger, faster applications keep testing that guarantee. Replication tweaks, blending different architectural models, and smarter, ML-guided tuning may soon carry more weight in system blueprints. Given that, and still, the long-range view remains rosy. NoSQL isn't fading to the periphery. Instead, these databases are poised to double down as backbone tech for the data-centric world. The core trio of ever-shifting data structures, infinite volume outsized loads, and lightning–quick intelligent apps needing quick pivots within milliseconds drives wide adoption. As the apparatus of extra silicon, denser fibers, and wider algorithms keep leaping and re-drafting blueprints, NoSQL will continue to stretch, adapt, and refine its recipes to match whatever threaded, steely challenge the next decade tosses skyward.

**References**

[1] B. Marr, *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results.* John Wiley & Sons, 2016.
[2] H. B. Alwan and K. R. Ku-Mahamud, "Big data: definition, characteristics, life cycle, applications, and challenges," *IOP Conference Series: Materials Science and Engineering*, vol. 769, 012007, 2020, doi: 10.1088/1757-899X/769/1/012007.
[3] T. Patel and T. Eltaieb, "Relational Database vs NoSQL," *Journal of Multidisciplinary Engineering Science and Technology*, vol. 2, pp. 691–695, 2015.
[4] M. Hassan, "Big Data, Big Decisions Choosing the Right Database," *arXiv preprint*, 2024. [Online]. Available: https://arxiv.org/abs/2405.02506
[5] E. Price and H. Baker, "NoSQL databases and big data query processing: Comparative analysis," *EA Journals*, 2021.
[6] A. Ali, S. Naeem, S. Anam, and M. M. Ahmed, "A state of art survey for big data processing and NoSQL database architecture," *International Journal of Computing and Digital Systems*, vol. 14, no. 1, pp. 297–309, 2023, doi: 10.12785/ijcds/140124.
[7] E. Dritsas and M. Trigka, "Database Systems in the Big Data Era: Architectures, Performance, and Open Challenges," *IEEE Access*, vol. 13, pp. 95068–95084, 2025, doi: 10.1109/ACCESS.2025.3572059.
[8] J. L. Pereira and M. Costa, "Decision support in big data contexts: A business intelligence solution," in *New Advances in Information Systems and Technologies*, Springer, vol. 444, pp. 983–992, 2016.
[9] G. A. Lakshen, S. Vraneš, and V. Janev, "Big data and quality: A literature review," in *Proc. 24th Telecommunications Forum (TELFOR)*, pp. 1–4, 2016, doi: 10.1109/TELFOR.2016.7818902.
[10] G. Mihai, "Comparison between Relational and NoSQL Databases," *Annals of "Dunarea de Jos" University of Galati, Fascicle I. Economics and Applied Informatics*, vol. 26, no. 3, pp. 38–42, 2020, doi: 10.35219/eai15840409134.
[11] W. Ali, M. U. Shafique, M. A. Majeed, and A. Raza, "Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics," *Asian Journal of Research in Computer Science*, vol. 4, no. 2, pp. 1–10, 2019, doi: 10.9734/ajrcos/2019/v4i230108.
[12] V. Manoj, "Comparative study of NoSQL document, column store databases and evaluation of Cassandra," *International Journal of Database Management Systems*, vol. 6, no. 4, pp. 11–26, 2014, doi: 10.5121/ijdms.2014.6402.
[13] A. Nayak, A. Poriya, and D. Poojary, "Type of NoSQL Databases and its Comparison with Relational Databases," *International Journal of Applied Information Systems*, vol. 5, no. 4, pp. 16–19, 2013, doi: 10.5120/ijais12-450888.
[14] C. Asaad, K. Baïna, and M. Ghogho, "NoSQL Databases: Yearning for Disambiguation," *arXiv preprint*, 2020.
[15] A. B. M. Moniruzzaman and S. A. Hossain, "NoSQL Database: New Era of Databases for Big Data Analytics – Classification, Characteristics and Comparison," *arXiv preprint*, 2013.
[16] A. Davoudian, L. Chen, and M. Liu, "A survey on NoSQL stores," *ACM Computing Surveys*, vol. 51, no. 2, Art. 40, 2018, doi: 10.1145/3158661.
[17] A. Owen, M. Maddog, and E. White, "Implementing real-time data analytics solutions through advanced stream processing technologies: A comprehensive approach to immediate data-driven decision-making," unpublished, 2024.
[18] Y. Gao, S. Ardalan, S. Ma, and H. Zadeh, "KSQL: Streaming SQL engine for Apache Kafka," in *Proceedings of the 22nd International Conference on Extending Database Technology (EDBT)*, pp. 664–667, 2019.

[19] M. Naeem, T. Jamal, J. Diaz Martinez, S. A. Butt, N. Montesano, M. I. Tariq, and E. De La Hoz Valdiris, "Trends and future perspective challenges in big data," in *Advances in Intelligent Data Analysis and Applications: Proceeding of the Sixth Euro China Conference on Intelligent Data Analysis and Applications*, 15–18 October 2019, Arad, Romania, pp. 309–325, Springer Singapore, 2022.

[20] A. K. Sandhu, "Big data with cloud computing: Discussions and challenges," *Big Data Mining and Analytics*, 2021.

[21] J. Sheng, J. Amankwah-Amoah, Z. Khan, and X. Wang, "COVID-19 pandemic in the new era of big data analytics: Methodological innovations and future research directions," *British Journal of Management*, vol. 32, no. 4, pp. 1164–1183, 2021, doi: 10.1111/1467-8551.12441.

[22] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, 2023, doi: 10.1016/j.ins.2022.11.139.

[23] M. Mendu, M. Krishna, G. Mahesh, and J. Pallavi, "Development of real time data analytics based web applications using NoSQL databases," *AIP Conference Proceedings*, vol. 2418, no. 1, May 2022, AIP Publishing.

[24] A. Faridoon and M. Imran, "Big data storage tools using NoSQL databases and their applications in various domains: A systematic review,"*Computing and Informatics*, vol. 40, no. 3, pp. 489–521, 2021, doi: 10.31577/cai_2021_3_489.

[25] N. Bansal, S. Sachdeva, and L. K. Awasthi, "Are NoSQL databases affected by schema?," *ETE Journal of Research*, vol. 70, no. 5, pp. 4770–4791, 2024, doi: 10.1080/03772063.2023.2237478.

[26] V. F. de Oliveira, M. A. O. Pessoa, F. Junqueira, and P. E. Miyagi, "SQL and NoSQL databases in the context of Industry 4.0," *Machines*, vol. 10, no. 1, p. 20, 2022, doi: 10.3390/machines10010020.

[27] I. Carvalho, F. Sá, and J. Bernardino, "Performance evaluation of NoSQL document databases: Couchbase, CouchDB, and MongoDB," *Algorithms*, vol. 16, no. 2, p. 78, 2023, DOI: 10.3390/a16020078.

[28] C.-O. Truică, E.-S. Apostol, J. Darmont, and T. B. Pedersen, "The Forgotten Document-Oriented Database Management Systems: An Overview and Benchmark of Native XML DODBMSes in Comparison with JSON DODBMSes," *Big Data Research*, vol. 25, p. 100205, 2021, DOI: 10.1016/j.bdr.2021.100205.

[29] M. Elkateb, M. Ashraf, and B. B. Zaidan, "Machine learning and IoT–based predictive maintenance approach for industrial applications," *Alexandria Engineering Journal*, vol. 88, pp. 298–309, 2024, DOI: 10.1016/j.aej.2023.12.065.

[30] A. S. Shaik and A. Shaik, "AI Enhanced Cyber Security Methods for Anomaly Detection," in *Machine Intelligence, Tools, and Applications*, S. Dehuri, B. B. Gupta, P. K. R. Maddikunta, and T. R. Gadekallu, Eds., pp. 401–416, Springer, 2024, DOI: 10.1007/978-3-031-65392-6_30.

[31] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo, "SQL and NoSQL database software architecture performance analysis and assessments—a systematic literature review," *Big Data and Cognitive Computing*, vol. 7, no. 2, p. 97, 2023, DOI: 10.3390/bdcc7020097.

[32] D. Brahneborg, W. Afzal, and S. Mubeen, "Resilient Conflict-free Replicated Data Types without Atomic Broadcast," in *Proceedings of the 17th International Conference on Software Technologies (ICSFT 2022)*, pp. 516–523, SCITEPRESS, 2022, DOI: 10.5220/0011314500003266.

[33] S. Ferreira, J. Mendonça, B. Nogueira, W. Tiengo, and E. Andrade, "Impacts of data consistency levels in cloud-based NoSQL for data-intensive applications," *Journal of Cloud Computing*, vol. 13, p. 158, 2024, DOI: 10.1186/s13677-024-00716-7.

[34] A. Oussous, F. Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "Big Data technologies: A survey," *Journal of King Saud University – Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018, DOI: 10.1016/j.jksuci.2017.06.001.