Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

NoSQL Databases: Types, Features, and Applications

Yashika Yadav
241302098
Department/School of Engineering and Technology, Shree Guru Gobind Singh Tricentenary University, Budhera, Gurugram, Haryana, India

Tiya Sharma
241302130
Department/School of Engineering and Technology, Shree Guru Gobind Singh Tricentenary University, Budhera, Gurugram, Haryana, India

Suraj Nayak
241302135
Department/School of Engineering and Technology, Shree Guru Gobind Singh Tricentenary University, Budhera, Gurugram, Haryana, India

Database Management System
Under the guidance of **Ms. Kanika Malik,** Assistant Professor

Abstract - NoSQL databases emerged as an important alternative to traditional relational database systems because modern applications generate large volumes of structured, semi-structured, and unstructured data and often require horizontal scalability and high availability. This report reviews three published research works on NoSQL databases. Davoudian, Chen, and Liu provide a broad survey of NoSQL stores using data model, consistency model, data partitioning, and the CAP theorem as key design dimensions [1]. Corbellini et al. examine the selection of NoSQL databases according to quality attributes and application requirements [2]. Abadi et al. review the NoSQL landscape and group systems according to structural similarity and important design characteristics [3]. The reviewed literature is used to explain the major NoSQL types, features, applications, trade-offs, comparative findings, and requirement-based recommendations.

Index Terms - NoSQL databases, database selection, data models, scalability, consistency.

I. INTRODUCTION

A database management system stores and retrieves data for applications and users. Traditional relational databases organize data into tables and are highly effective when strong consistency, relationships, and transactional processing are central requirements. However, the growth of web applications, cloud computing, social media, and big-data systems has increased the need for databases that can scale across many machines and accommodate flexible or rapidly changing data structures. NoSQL is commonly understood as "Not Only SQL".

Rather than representing all data through the relational table model, NoSQL systems use several models, including keyvalue, document, wide-column, and graph models. Research shows that these systems make different trade-offs among scalability, consistency, availability, query flexibility, and operational complexity [1]–[3]. Therefore, choosing a NoSQL database should depend on the application's data model, access patterns, workload, and non-functional requirements.

II. LITERATURE REVIEW

A. Davoudian, Chen and Liu—A Survey on NoSQL Stores (2018)

Davoudian, Chen, and Liu published "A Survey on NoSQL Stores" in ACM Computing Surveys. The paper explains NoSQL design through four major, non-orthogonal principles of distributed database systems: data model, consistency model, data partitioning, and the CAP theorem [1]. The authors discuss the strategies associated with these principles and their strengths and drawbacks, using representative academic and industrial NoSQL technologies as examples.

The paper is particularly useful for understanding why NoSQL databases are not a single technology. Different systems make different design choices. Data partitioning and replication help distributed systems scale, while consistency models determine how quickly updates become visible across nodes. The CAP theorem is therefore important when evaluating distributed NoSQL systems because consistency, availability, and partition tolerance involve fundamental trade-offs [1].

B. Corbellini et al.—Choosing the Right NoSQL Database for the Job (2015)

Corbellini and co-authors published "Choosing the right NoSQL database for the job: a quality attribute evaluation" in the Journal of Big Data (Springer). The paper emphasizes that NoSQL is a broad category containing distinct systems with different use cases. It identifies four common categories: key-value stores, document stores, column stores, and graph databases [2].

The study highlights quality attributes and selection criteria rather than assuming that one NoSQL technology is best for every application. Important factors include scalability, availability, consistency, performance, data model, and the application's requirements. This supports a requirementdriven approach to database selection [2].

C. Abadi et al.—Persisting Big-Data: The NoSQL Landscape (2017)

The paper "Persisting big-data: the NoSQL landscape", published in Information Systems, reviews NoSQL database implementations designed for large-scale data persistence. The authors group databases according to structural similarity and compare them across multiple dimensions [3]. The work explains that massively accessed web applications and very large datasets create requirements that can be difficult for traditional relational systems to satisfy.

The paper is useful for connecting NoSQL database design with real application needs. It shows that NoSQL systems should be evaluated by their data organization, scalability, distribution

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

mechanisms, consistency behavior, and suitability for particular workloads [3].

D. Major Types of NoSQL Databases

Type	Data Model	Typical Strength	Example Applications
Key-Value	Key associated with a value/object	Very fast simple lookups; easy horizontal scaling	Caching, sessions, user preferences, realtime counters
Document	Semi-structured documents such as JSON/BSON	Flexible schema and nested data	Content management, product catalogs, user profiles, web/mobile applications
Wide-Column	Rows with flexible/dynamic columns grouped by column families	Large distributed workloads and high write throughput	Time-series data, event logs, analytics, largescale operational systems
Graph	Nodes and relationships/edges	Efficient traversal of connected data	Social networks, recommendation systems, fraud/network analysis

TABLE I MAJOR TYPES OF NOSQL DATABASES

E. Common Features of NoSQL Databases

- Horizontal scalability: capacity can be increased by distributing workloads across additional machines.
- Flexible or schema-flexible data models: useful when application data changes frequently or is semi-structured. • Distribution and replication: data can be partitioned and replicated across nodes to support availability and scale.
- High throughput and low-latency access for suitable workloads, especially simple or predictable access patterns.
- Different consistency choices: systems may provide strong consistency or weaker/eventual consistency depending on design.
- Application-oriented modeling: data can be organized around the application's access patterns rather than normalized relational tables.

F. Applications of NoSQL Databases • Web and mobile applications: Flexible user profiles, content, sessions, and rapidly changing application data.

- E-commerce: Product catalogs and inventory-related data where attributes can vary between products.
- Social networking: Large user-generated datasets and, for graph systems, relationships between users.
- Real-time analytics and event processing: High-volume event, log, and time-series workloads.
- Caching and session management: Fast key-value access for frequently requested data.
- Recommendation and fraud analysis: Relationship-heavy workloads where graph traversal is important.

III. COMPARISON OF THE REVIEWED STUDIES

A. Comparison of the Three Research Papers The three selected research papers provide different perspectives on NoSQL databases while addressing the common problem of managing modern, large-scale and distributed data. Davoudian et al. [1] provide a broad theoretical survey of NoSQL stores, Corbellini et al. [2] concentrate on quality attributes and database selection, while Abadi et al. [3] examine the overall NoSQL landscape and classify different database implementations.

Research Paper	Main Focus	Key Criteria	Contribution	Best Use in This Study
Davoudian, Chen & Liu (2018) [1]	Design of NoSQL stores	Data model, consistency, partitioning, CAP	Explains architectural trade-offs across NoSQL systems	Understanding types and distributed design
Corbellini et al. (2015) [2]	Database selection	Quality attributes and application requirements	Shows that database choice should be requirementdriven	Choosing a suitable NoSQL type
Abadi et al. (2017) [3]	NoSQL landscape for big data	Structural similarity, scalability, distribution	Reviews and compares NoSQL implementations	Connecting database types to large-scale applications

TABLE II
COMPARISON OF THE THREE REVIEWED RESEARCH PAPERS

Davoudian, Chen, and Liu [1] published A Survey on NoSQL Stores in ACM Computing Surveys in 2018. The study provides a broad review of NoSQL database systems and explains their design using four major dimensions: data model, consistency model, data partitioning, and the CAP theorem.

The paper demonstrates that NoSQL databases should not be treated as one homogeneous technology. Different systems make different architectural choices depending on their intended workloads and requirements. Data partitioning and replication allow distributed systems to handle large amounts of data and support scalability. At the same time, consistency models determine how and when updates become visible across different nodes. The CAP theorem is particularly important in understanding distributed NoSQL databases because distributed systems must deal with consistency, availability, and partition tolerance. The paper therefore provides the theoretical foundation for understanding why different NoSQL systems make different trade-offs [1]. This research is important for understanding the technical and architectural foundations of NoSQL databases.

Corbellini [2] and co-authors published Choosing the Right NoSQL Database for the Job: A Quality Attribute Evaluation in the Journal of Big Data in 2015. The study emphasizes that NoSQL is a broad category containing different database systems rather than one common technology. It identifies four major categories: key-value stores, document stores, column stores, and graph databases. The research focuses strongly on quality attributes and selection criteria. Important factors discussed include scalability, availability, consistency, performance, data model, and application requirements. This indicates that there is no single NoSQL database that is best for every situation. The major contribution of this paper is therefore a requirement-driven approach to NoSQL database selection

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

[2].

Abadi et al. [3] published Persisting Big-Data: The NoSQL Landscape in Information Systems in 2017. The paper reviews NoSQL database implementations designed to handle large-scale data persistence. The study groups databases according to structural similarity and compares them across multiple dimensions. It connects the development of NoSQL databases with the growth of massively accessed web applications and very large datasets. The research highlights important characteristics such as data organization, scalability, distribution mechanisms, and consistency behavior. It shows that NoSQL systems were developed to address requirements associated with large-scale and distributed applications. This paper therefore contributes to understanding the overall NoSQL database landscape and the relationship between database design and large-scale application requirements [3].

Overall Comparison of the Three Studies

The three research papers complement each other. Davoudian et al. [1] mainly provide a technical and theoretical foundation, explaining data models, consistency, partitioning, and CAP-related trade-offs. Corbellini et al. [2] provide a more application-oriented perspective, emphasizing quality attributes and requirements when selecting a NoSQL database. Abadi et al. [3] provide a landscape-oriented perspective, explaining the classification and characteristics of different NoSQL implementations. Together, the three studies establish that NoSQL databases provide flexibility and scalability for modern distributed applications, but their benefits depend on selecting the appropriate database model and understanding the tradeoffs involved.

B. Comparative Analysis of NoSQL Types

Criteria	Key-Value	Document	Wide-Column	Graph	Suitable Workload
Data structure	Key + value	JSON/BSON-like document	Rows/column families	Nodes + edges	Depends on data relationships
Schema flexibility	High	Very high	High	High	Rapidly changing data
Relationship handling	Limited	Moderate	Limited/indirect	Excellent	Connected/relationship-heavy data
Typical strength	Fast lookup	Flexible application data	Distributed largescale writes	Graph traversal	Accesspattern dependent
Common examples	Redis	MongoDB	Cassandra/HBase	Neo4j	Varies by application

TABLE III
COMPARATIVE ANALYSIS OF NOSQL TYPES

IV. FINDINGS

NoSQL Databases Are Designed for Modern Large-Scale Applications

The reviewed studies indicate that NoSQL databases have become important because modern applications frequently deal with large volumes of data and distributed workloads. Traditional relational systems may not always be the most suitable choice when applications require flexible data structures and horizontal scalability.

NoSQL Is a Collection of Different Database Models NoSQL is not a single database technology. The literature identifies different models, including key-value, document, wide-column/column, and graph databases. Each model organizes and accesses data differently and is therefore suitable for different application requirements.

Data Model Is an Important Selection Factor

The type of data being stored has a major influence on database selection. Document databases are suitable for semi-structured data, key-value databases are suitable for simple key-based access, wide-column systems are useful for large distributed datasets, and graph databases are suitable when relationships between entities are central.

Scalability Is a Major Advantage of NoSQL

Horizontal scalability is an important feature of many NoSQL systems. Instead of relying only on a single powerful machine, workloads and data can be distributed across multiple machines. This makes NoSQL suitable for applications that experience increasing data volume and user demand.

NoSQL Provides Flexible Data Structures Unlike traditional relational systems that generally use predefined tables and schemas, many NoSQL databases support flexible or schema-flexible data models. This is particularly useful when application data changes frequently or contains semi-structured information.

Distribution and Partitioning Support Large Workloads The reviewed literature emphasizes data partitioning and distribution as important aspects of NoSQL architecture. Data can be distributed across multiple nodes, allowing systems to handle larger workloads and support scalable database operations.

Consistency and Availability Involve Trade-Offs

NoSQL database systems may make different choices regarding consistency and availability. In distributed environments, these choices become particularly important because system design must account for network partitions and the required level of consistency.

Performance Depends on the Workload NoSQL databases can provide high throughput and lowlatency access for suitable workloads. However, performance is not identical across all NoSQL databases. It depends on the database model, data organization, access pattern, workload, and system architecture.

NoSQL Databases Are Application-Oriented

The reviewed studies show that NoSQL databases can organize data according to application access patterns rather than strictly following normalized relational structures. This approach can simplify data access for certain large-scale applications.

NoSQL Is Not a Complete Replacement for Relational Databases

The research does not suggest that NoSQL databases should completely replace relational databases. Instead, database selection should depend on application requirements. Relational databases remain appropriate when strong consistency, relationships, and transactional processing are central

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

requirements, while NoSQL can be advantageous for flexible, distributed, and large-scale workloads.

V. RECOMMENDATIONS

Select the Database According to Application Requirements

NoSQL database selection should begin with the application's requirements rather than choosing a database simply because it is classified as NoSQL. Factors such as data structure, workload, scalability, performance, consistency, and availability should be considered.

Select the Appropriate NoSQL Data Model

The database model should match the nature of the data. Document databases are suitable for semi-structured and nested data, key-value stores for simple high-speed lookups, wide-column databases for large distributed workloads, and graph databases for highly connected data.

Evaluate Scalability Requirements

Applications expected to experience significant growth should evaluate horizontal scalability carefully. The selected database should be able to distribute data and workload effectively across multiple nodes when required.

Consider Consistency Requirements

Before selecting a NoSQL database, the required level of consistency should be identified. Applications where immediate consistency is essential may require different database characteristics from applications that can tolerate eventual or weaker consistency.

Consider Availability and Distribution

For applications operating in distributed environments, availability and data distribution should be important selection criteria. Replication and partitioning strategies should be evaluated according to the application's requirements.

Match Database Selection With Access Patterns The way an application reads and writes data should be considered before selecting the database model. A system optimized for simple key-based access may not be equally suitable for relationship-heavy queries or complex document-oriented workloads.

Evaluate Performance Using the Actual Workload Performance should not be judged only by general claims about NoSQL databases. The selected technology should be evaluated according to realistic data volumes, read/write patterns, concurrency, and expected application workload.

Consider Operational Complexity

Organizations should consider the complexity involved in deploying, maintaining, scaling, monitoring, and securing a distributed NoSQL system. The technical advantages of a database should be balanced against the resources required to operate it.

Do Not Select NoSQL Without Understanding Its TradeOffs

NoSQL databases provide important advantages, but they also involve trade-offs. Before adoption, organizations should understand the implications of consistency, availability, partitioning, scalability, and data modeling.

Use a Requirement-Driven Database Strategy

The overall recommendation from the reviewed literature is to follow a requirement-driven approach. Database selection should be based on data model, access patterns, performance expectations, scalability, consistency, availability, and the operational environment rather than selecting a technology solely because it is popular.

VI. CONCLUSION

The reviewed research establishes that NoSQL databases became important because modern applications often require scalable, distributed, and flexible data management. Unlike conventional relational databases that generally depend on predefined schemas and tabular structures, NoSQL databases provide multiple data models, including key-value, document, column-family, and graph models. The literature consistently emphasizes that NoSQL systems differ in their data models, consistency mechanisms, partitioning strategies, and quality attributes [1]–[3].

The comparison of the three reviewed research papers provides a comprehensive understanding of NoSQL databases, their design principles, and their application requirements. Davoudian et al. [1] explain the technical foundations of NoSQL through data models, consistency models, data partitioning, and the CAP theorem. Corbellini et al. [2] emphasize quality attributes and application requirements for selecting an appropriate NoSQL database. Abadi et al. [3] provide a broader view of the NoSQL landscape and its relationship with large-scale data persistence.

The findings show that NoSQL databases offer important capabilities such as horizontal scalability, flexible data models, distribution, replication, and high performance for suitable workloads. However, these advantages are associated with architectural and operational trade-offs, particularly regarding consistency, availability, partitioning, and data modelling. The study also confirms that different NoSQL models are designed for different purposes. Keyvalue, document, wide-column, and graph databases provide different approaches to storing and accessing information.

The reviewed literature indicates that NoSQL databases are particularly useful in distributed and large-scale environments where scalability, flexibility, availability, and low-latency data access are important. However, the studies also show that NoSQL databases involve trade-offs related to consistency, querying, transaction support, data modeling, and system complexity. Therefore, the choice of a NoSQL database should be based on the application's data structure, access patterns, workload, consistency requirements, scalability needs, and operational environment rather than treating NoSQL as a universal replacement for relational databases. A careful, requirement-driven selection process can help organizations obtain the benefits of NoSQL while managing its associated trade-offs.

VII. REFERENCES

[1] A. Davoudian, L. Chen, and M. Liu, “A Survey on NoSQL Stores,” ACM Computing Surveys, vol. 51, no. 2, Art. no. 40, 2018, doi: 10.1145/3158661.

[2] C. Corbellini, A. Mateos, G. Zunino, and A. Godoy, “Choosing the right NoSQL database for the job: a quality attribute evaluation,” Journal of Big Data, 2015, doi: 10.1186/s40537-015-0025-0.

[3] M. Abadi et al., “Persisting big-data: The NoSQL landscape,” Information Systems, vol. 63, pp. 1–23, 2017, doi: 10.1016/j.is.2016.07.009.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)