Journal of Global Research in Electronics and Communication
Volume 2, No. 3, March 2026
Available Online at: www.jgrec.info
REVIEWPAPER

# Database Administration Across SQL and NoSQL Systems: Trends, Techniques, and Challenges

Dr. Nilesh Jain
*Associate Professor*
*Department of Computer Science and Applications*
*Mandsaur University*
Mandsaur, India
nileshjainmca@gmail.com

***Abstract*—In the age of data-driven computing, data management has become a significant concern in managing highly scaled, heterogeneous and fast-varying loads of data. The paper also provides a comprehensive analysis of the management of databases in both SQL and NoSQL databases based on their architecture, nature of operations, trends in their performance, security issues and future trends of technology. The analysis indicates that relational (SQL) databases are very consistent and structured, and accommodate ACID-compliant transactions, which are very well suited to enterprise and transactional systems. On the other hand, NoSQL systems are highly scalable, schema-agnostic and effective with unstructured and distributed data, which is advantageous for contemporary applications, including big data analytics, cloud computing and real-time systems. Other developments covered in the paper include cloud-native databases, Data-as-a-Service (DBaaS), serverless computing, AI-based automation and distributed database models. Comparative analysis shows that there is no single best database model, and suitability of a model can be determined by the requirements of a given application and the characteristics of the workload. The conclusion of the paper is that database systems in the next-generation computing environments will increasingly embrace hybrid architecture and smart automation to enhance their scalability, security and performance.**

***Keywords*—SQL Databases, NoSQL Databases, Database Administration, Cloud-Native Databases, Distributed Systems, AI-based Database Optimization, Database Security.***

## I. INTRODUCTION

In today's fast-paced digital world, data is one of the most important resources for companies. The capability to store, manage, and analyze data is important, regardless of whether it is to gain a better insight into customer behaviour, achieve operational efficiency, or aid in decision-making [1]. With growing advancements in technology, the need for databases has grown by leaps and bounds, necessitating systems that can process large amounts of data, handle complex queries and do so quickly. Cloud computing, artificial intelligence (AI), and big data technologies have added to these demands and made organizations more innovative in their database management methods [2][3].

Traditionally, relational database has been employed by organizations to manage structured information, particularly in applications where consistency and integrity are considered the key factors. Nevertheless, as social media material, unstructured and semi-structured data, and images, videos started to rapidly grow, the existing relational systems started to demonstrate flexibility and scalability shortcomings [4]. As a result, NoSQL databases were developed to solve these problems by offering scalability in high-performance applications with a flexible data model and horizontal scaling.

As database technologies evolved, NewSQL databases are a brand-new class of systems [5]. The purpose of the systems is to unite the virtues of relational and NoSQL database systems through offering high transactions and scalability. This has led to a heterogeneous database ecosystem, necessitating more complex administrative strategies to deal with heterogeneous environments.

In the context of database administration across SQL and NoSQL systems, it is essential to emphasize key aspects such as optimization, data modelling, and scalability. Database administrators (DBAs) undertake a number of tasks which include schema design, query optimization, indexing, replica and security control of data [6]. Roles play a vital role in performance, reliability and efficient utilization of resources in both centralized and distributed databases.

The current state of database management is marked with various challenges, such as poor scaling, consistency trade-offs, and complexity of distributed systems, yet meaningful enhancements have been achieved. Moreover, AI-intensified database automation and predictive analytics are also evolving more recently, altering the way administrators have always done things, with the ability of smart optimization and real-time monitoring of systems [7]. These types of advancements are essential for enhancing performance, safety and scalability of current database management systems.

### *A. Structure of the paper*

The paper is organized as follows: Section II covers database administration for SQL and NoSQL databases. Section III reviews the new trends and technologies. Section IV addresses security issues in NoSQL databases. Section V gives a literature review. Concludes with future research directions in the last section of VI.

## II. DATABASE ADMINISTARTION ACROSS : SQL AND NOSQL

Database systems that simplify storage, management and retrieval of information are also essential components of modern computing. In essence, a database system consists of a collection of data and a management application. Data is kept in the database in an organized fashion, making retrieval and manipulation simple [8]. The structure would enable users to perform a variety of operations, including maintaining relationships among data entities, updating data, and searching for data. Administratively, these features would

necessitate effective control structures to guarantee performance, security and integrity of data in various database environments [9].

### A. Types of Database Systems

The database systems may be divided into a number of types depending on their structure, applications and management modes. Database administrators should consider this classification because each type has diverse management, optimization and scalability strategies. The most widespread ones are following:

*1) Relational databases:*

Relational databases are among most widely used database systems [10]. It displays information in tables with rows and columns. Tables are the entities themselves, and connections between tables are established using foreign keys. The standard language of interaction with relational databases is SQL. These systems may be utilized in applications that demand high levels of data integrity since they are extremely consistent, well-designed, and transactionally dependable.

Some common relational database includes:

- **MySQL:** Widely used for web applications, an open-source relational database management system.
- **PostgreSQL:** A sophisticated open-source database that is also extensible and follows the SQL standards.
- **Oracle Database:** A commercial relational database it is well-known for its scalability and performance, most frequently utilized in enterprise applications.

*2) Not only SQL (NoSQL) databases:*

NoSQL databases offer an alternative to traditional relational databases and are made to manage data that is either semi-structured or unstructured [11]. It is especially applicable to big data and real-time web applications that need to be highly scalable and flexible. NoSQLs are opposite of relational databases. NoSQL stores are usually schema-free and are horizontally scalable, creating new administrative issues of data distribution and consistency. There are several further classifications for NoSQL databases:

- **Document stores:** These databases store data in document-oriented formats. Each document may have a different structure, providing flexibility.
- **Key-value stores:** These databases use key-value pairs to store data. Because of their simplicity and effectiveness, they are appropriate for session management and caching.
- **Column family stores:** In order to facilitate effective querying and processing of massive amounts of data, instead of storing data in rows, these databases use columns.
- **Graph databases:** These databases are appropriate for applications with intricate links and networks because they are made to represent and query data in graph structures, as Table I illustrates.

TABLE I. DATABASE MANAGEMENT SYSTEMS: A NOSQL ANALYSIS

| Category | Key Value Stores | Column Databases | Document Databases | Graph Databases |
|---|---|---|---|---|
| Based on | Dynamo DB with Dynamic Hash Tables | Google's Bigtable | JSON and XML encoding are used in Lotus Notes. | Euler's Graph Theory |
| Data Model | Key/Value pairs | Columns | Key/Value Collections | Nodes, Edges, and Properties in Graph Structure |
| Applicability | Handling massive load | Distributed file systems | Web applications, information ranking, and full text searches and updates | Social networks, intelligent agencies, and the semantic web |
| Advantages | Simple and easy to implement | Quick data queries and storing enormous amounts of data | Enables for effective searching and accepts partially completed data | Complex data can be easily scaled over dispersed platforms |
| Disadvantages | Ineffective when updating or accessing a database | Very low-level API | No standard query language | Traversal of full graph is required to provide accurate findings |
| Examples | Project Voldemort and Redis | Cassandra, HBase | MongoDB, CouchDB | Neo4J, InfoGrid |

### B. Applications of database systems

The contemporary information management system cannot be devoid of database systems, which facilitate efficient storage, retrieval and processing of structured and unstructured information. They contribute to decision-making, as well as operational efficiency, with an increasing significance in the face of the rapid digital transformation of industries and applications, as follows:

- **Business applications:** Database systems are used by businesses to handle employee, sales, inventory, and customer data. ERP and CRM systems are among applications that are highly reliant on effective database management systems.
- **Healthcare:** Hospitals and other healthcare facilities employ database systems to keep track of medical histories, make appointments, and preserve patient information. Such systems enhance efficiency and medical treatment of patients.
- **E-commerce:** The online retailing stores have database systems to run product catalogs, order tracking and customer data[12]. Good data management is essential to the provision of a good shopping experience.
- **Education:** Database systems are used in the management of student record, course registration and faculty in learning institutions. This eases effective administrative processes and communication.
- **Finance and banking:** Database systems are relied upon by financial institutions to handle customer accounts, transactions and regulatory compliance. In this industry, security and data integrity are of high significance[13].

### C. Characteristics of SQL databases

SQL databases have a number of distinguishing features:

- **Structured Data:** Data consistency and integrity are ensured by the SQL databases' ability to manage structured data with defined schemas.
- **Atomicity, Consistency, Isolation and Durability (ACID) Properties:** They offer ACID in transactions, which guarantee that transactions are processed in a dependable and predictable manner [14].
- **Relational Model:** The data is organized in a table format, and a table-table relationship is established by use of foreign keys. The model aids in effective data organization and retrieval.
- **SQL Language:** A standardized language called SQL is used for database interaction, query execution, and data manipulation, which makes it easy to administer and manage the database.

## III. Features of Relational Databases and NoSQL Database Models

Relational and NoSQL database models differ in their characteristics regarding scalability, performance, consistency and data management. The key to successful database administration and system design is to understand these differences. Fig. 1 illustrates the important distinguishing properties of the two database models:

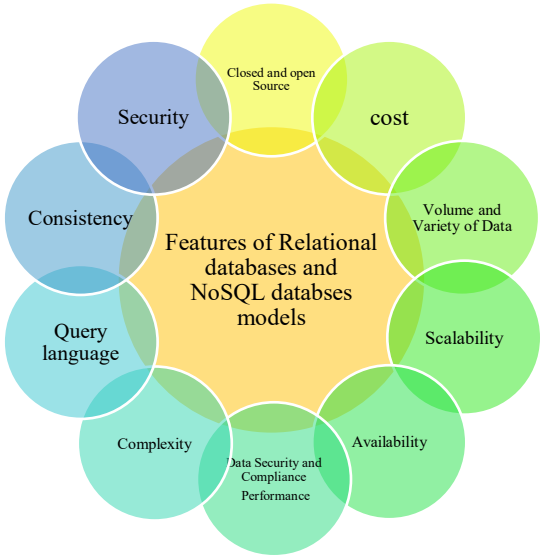


Fig. 1. Features of Relational Databases and NoSQL Databases

### 1) *Closed and Open Source*

There are both open-source and commercial relational database systems [15]. Proprietary systems like Oracle have sophisticated enterprise functionality, whereas open-source systems such as MySQL have flexible and affordable solutions. The majority of NoSQL database models, such as MongoDB, CouchDB, and Cassandra, are open source. The flexibility, community-based development and low costs are supported by the open-source character of NoSQL.

### 2) *Cost*

Relational databases are also linked to increased cost, especially proprietary systems, which can cost a lot to invest in and upgrade hardware [16]. This makes relational databases quite costly to implement at scale. NoSQL is a less expensive option because it is primarily open source. The usage of virtualized environments and inexpensive technology greatly lowers cost of maintaining NoSQL systems.

### 3) *Volume and Variety of Data*

The quantity and variety of data have greatly risen because to big data and online applications [17]. The purpose of relational databases is to handle structured data, but can be limited in handling extremely diverse or large-scale data. Large amounts of organized, semi-structured, and unstructured data may be handled by NoSQL databases, they can be applied to data-intensive applications.

### 4) *Scalability*

Upgrading a single server generally enables relational databases to scale vertically, limiting the hardware and costs. This method demands extra administrative work. In contrast, NoSQL databases support horizontal scaling using distributed commodity servers, enabling more flexible and scalable system expansion.

### 5) *Availability*

Social media, e-commerce, and cloud storage are the main drivers of increased user numbers and time spent accessing data. Single points of failure are common in relational databases by design, while being extremely powerful systems. Relational databases scale up, which further restricts availability. In today's online apps, where users rely heavily on them for daily assistance, single points of failure are ineffective. Consequently, NoSQL is a superior option for maintaining continuous user availability due to its distributed design, even in the event of hardware failures.

### 6) *Performance*

Relational databases process data significantly more slowly compared to NoSQL databases, which handle data rapidly. NoSQL employs volatile memory, which further enhances performance, in contrast to relational databases, which extract data from non-volatile memory. Volatile memory is faster than nonvolatile memory by nature. NoSQL has outperformed relational databases in Internet search applications.

### 7) *Complexity*

Relational Databases generate complex data in situations where users find it difficult to convert data into tables. Relational Databases are complex because they focus heavily on storing structured data. In other situations, complicated queries, transactions, and relational databases might not be required, such as social media, where simple reading and writing skills could be sufficient. Both unstructured and semi-structured data can be stored in NoSQL. NoSQL offers the flexibility needed to manage various types of data in their raw state without information loss since it can store both semi-structured and unstructured data.

### 8) *Query Language*

The basis of Relational Databases is solid, and there is extensive literature on SQL. Relational databases exclusively employ SQL as a data manipulation language. Among the many Relational Databases in use, there are nonetheless slight differences in their SQL implementations. Because SQL offers Relational databases have a solid basis and are well-liked by developers due to their easier implementations. Because NoSQL depends on object-oriented APIs to manipulate data, it does not yet have this foundation.

### 9) *Consistency*

The rigorous schema is more consistently applied by Relational Databases. Because availability and relational databases are incompatible, this feature forces relational

databases to compromise availability. Strong consistency helps provide an immediate, consistent view of the data once procedures are complete. Nonetheless, flexibility is more crucial than consistency in some applications, like social media. NoSQL databases have low consistency but offer higher availability. NoSQL is, therefore, a better storage choice for social media than Relational Databases.

*10) Security:*

Relational database security concerns include cross-site scripting and SQL injection. Despite these difficulties, SQL has strong security capabilities to safeguard data, including as auditing, data integrity, encryption, authorization, and authentication. The security measures are contained in the database. Instead of the database itself, middleware is used to control security in NoSQL. Databases are hence susceptible to intrusions [18][19].

### *A. Difference between RDBMS and NoSQL database*

RDBMS and NoSQL databases differ in data models, scalability, and consistency mechanisms. A comparative analysis is presented in Table II.

TABLE II. COMPARISON BETWEEN RELATIONAL AND NOSQL DATABASES

| Comparison Aspect | Relational Databases (SQL) | NoSQL Databases |
|---|---|---|
| Platform Type | Supports both open-source and proprietary systems | Primarily open-source platforms |
| Scalability Approach | Vertical scaling through hardware upgrades | Horizontal scaling using distributed systems |
| Cost Structure | Higher cost, especially for proprietary solutions | Cost-effective with lower infrastructure requirements |
| Data Handling Capability | Suitable for structured and moderate-scale data | Designed for large-scale, diverse, and unstructured data |
| Availability Model | Limited by centralized architecture and a potential single point of failure | High availability due to distributed architecture |
| Performance Behavior | Slower in large-scale or complex query operations | Optimized for high-speed read/write operations |
| Data Complexity | High due to strict schema requirements | Lower due to flexible and schema-less design |
| Query Mechanism | Uses standardized SQL with minor variations | Uses diverse query languages or APIs |
| Consistency Model | Strong consistency with strict schema enforcement | Flexible consistency (e.g., eventual consistency) |
| Security Approach | Built-in security features (authentication, authorization, etc.) | Often relies on external or middleware-based security |

## IV. DATABASE TRENDS AND TECHNOLOGIES

In fast-changing environment of data management, organizations are embracing numerous new methods of responding to the continuously increasing demands of scalability, availability, and flexibility [20]. The necessity to work with massive amounts of data, improve productivity, and take use of cloud-based solutions has led to a number of developments in recent years. These trends indicate a shift in way databases are organized, implemented, and operated, as shown in Fig. 2. The following is a list of some of the most significant database trends and technologies currently affecting the industry:


Fig. 2. Database trends and technologies

*1) Cloud-Native Database:*

Cloud-native databases have become very popular as companies continue to push their workloads to the cloud. These databases were specifically created to take advantage of cloud architecture, and they provide the advantages of scalability, flexibility, and high availability [21]. Cloud providers also managed databases that are cloud-native, unlike conventional on-premises databases, which means cross-management and maintenance are not required.

*2) NoSQL Databases:*

NoSQL databases are becoming increasingly necessary for applications that need to scale to manage enormous volumes of unstructured or semi-structured data. In contrast to conventional relational databases, which are based on SQL and pre-existing schema, NoSQL databases may be founded upon a less strict data structure, such as document stores, graph databases, column-family stores, and key-value pairs.

*3) Distributed Databases:*

Distributed databases have emerged as the foundation of the current data architecture as businesses become global and increasingly use multi-cloud or hybrid cloud approaches. Organizations may store and manage data in several places thanks to distributed databases, which guarantee high availability and low latency access.

*4) Database-as-a-Service (DBaaS):*

Businesses can obtain complete database administration services from DBaaS, a cloud-based solution, without having to handle database administration internally. Organizations can swiftly build, expand, and manage databases using DBaaS without worrying about infrastructure, software, or hardware management [22][23].

*5) Serverless Databases:*

Serverless databases are a fairly recent development in field of databases, offering a solution in which the business does not have to operate server infrastructure to host the database. Rather, the database scales automatically with demand and is billed based on actual usage. It is especially appealing when there are unpredictable or variable workloads.

*6) NewSQL Databases:*

NewSQL databases are a hybrid solution, as scalable and flexible as NoSQL systems, with conventional relational databases' robust consistency and Atomicity, Consistency, Isolation, Durability (ACID) guarantees. These databases are meant to overcome constraints of relational and NoSQL databases, as they are highly scaled, with transactional integrity.

*7) Data Security and Compliance:*

The increasing amount of sensitive data being stored in databases, organizations increasingly prioritize security and compliance above everything else. Two examples of data privacy laws that have compelled companies to reconsider how they handle and retain data are GDPR and CCPA. To assist companies in complying with these regulations, database systems with improved security features (encryption, access restrictions, and auditing) are being developed.

*8) Artificial Intelligence and Machine Learning Integration:*

The integration of AI and ML capabilities with database systems is changing way companies handle and streamline their data. Routine database administration operations such as anomaly detection, performance tuning, and query optimization may be automated with AI and ML[24]. This enables databases to be more independent and eliminates the need for human involvement, improving efficiency.

*9) Edge Databases:*

The advent of IoT devices and the increasing need to operate in real time have made edge computing a significant trend in the world of databases. By storing all data in edge databases, processing can be done closer to the point of production (e.g., IoT devices or local data centers), thereby decreasing latency and improving decision-making speed.

*10)Blockchain and Immutable Databases:*

The use of decentralized and immutable Blockchain technology in databases is being considered, especially in instances where data integrity, traceability, and auditability are important. The immutable databases based on blockchain offer an additional level of security and trust because once stored, the data cannot be changed.

### *A. Architecture and Data Models of SQL Databases*

Relational databases, sometimes known as SQL databases, are organized databases with a tabular schema that consists of rows and columns as their architecture [25]. The relational model on which SQL databases are based puts data into logically related tables based on a predefined design. Relationships between tables are established using primary and foreign keys to provide referential consistency and integrity. Each table contains records (rows), all of which have the same structure (defined by columns with fixed data types and constraints). A typical SQL database contains important sub-parts, including a transaction manager, a storage engine and a query processor. The query processor processes SQL queries and optimizes them to ensure efficient execution. The storage engine handles the physical structure, indexing and retrieval of data on disk. The transaction manager ensures ACID properties, which provide adequate concurrency control and maintain data integrity in multi-user systems.

### *B. Architecture and Data Models of NoSQL Databases*

The purpose of NoSQL databases is to provide extensive unstructured and semi-structured data models, which is a completely new architecture, with great scalability, flexibility, and capabilities to process data across distributed nodes. These databases usually enable dynamic adjustments to data models on the fly and do not impose rigid schemas. Based on the underlying data structures, NoSQL systems are classified into several types, each suited to specific data use cases and access patterns, as opposed to having a unified relational structure [26]. The architecture is distributed by design, allowing for horizontal scaling via replication and partitioning techniques (Fig. 3). Unlike SQL systems where data is normalized, NoSQL databases often use denormalized data models to improve performance for high-throughput operations.

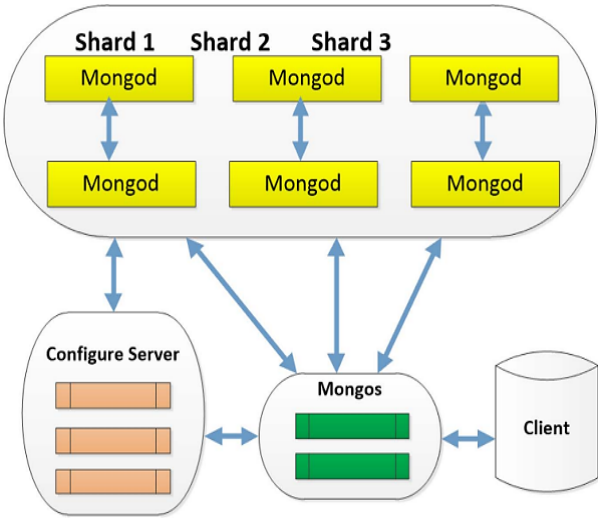


Fig. 3. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments

## V. Security Challenges in NoSQL Databases

The NoSQL databases are designed to meet the demands of large-scale data analysis, and security is not as important during the design phase. Compared to conventional databases, NoSQL databases provide a comparatively thin layer of security.

The following is a list of NoSQL databases' main security risks:

*1) Transactional Integrity:*

The NoSQL databases' soft nature makes it impossible to ensure transactional integrity. NoSQL database architecture cannot incorporate complex integrity requirements, as doing so would fail to achieve the primary goals of NoSQL, which are improved performance and scalability.

*2) Authentication Mechanisms:*

Information leakage can result from replay attacks, injection attacks, password brute force attacks, cross-site request forgery, and man-in-the-middle attacks against NoSQL databases [27]. The primary cause is the use of shoddy password storage methods and authentication procedures in NoSQL databases. While not necessary on all commodity servers, certain NoSQL databases require authentication at the local node level.

*3) Susceptibility to Injection Attacks:*

Injection attacks introduce data of their own choosing into the NoSQL database, resulting in data corruption or unavailability. Because NoSQL's design uses loosely coupled mechanisms and extremely lightweight protocols, an attacker can use a file system's backdoor to do harmful actions.

*4) Lack of Consistency:*

The three conditions of the CAP theorem (partition fault tolerance, consistency, and availability) are not all concurrently met by NoSQL databases. NoSQL databases rely on a large number of dispersed commodity servers, this does not guarantee consistent results all amount of time that certain commodity servers might not be completely synchronized with other servers that contain the most recent data.

*5) Insider Attacks:*

The insufficiency of NoSQL databases' logging and log analysis capabilities enables an insider attacker to obtain other users' sensitive data. Because NoSQL databases have a very weak security layer, users find it exceedingly challenging to keep control over their data.

### *A. Characteristics of NoSQL*

System availability, network partition tolerance, and data consistency are three essential characteristics of distributed data systems, as shown in Fig. 4. These properties are defined by the CAP theorem, which guides the design of NoSQL databases. NoSQL databases are primarily classified as follows:

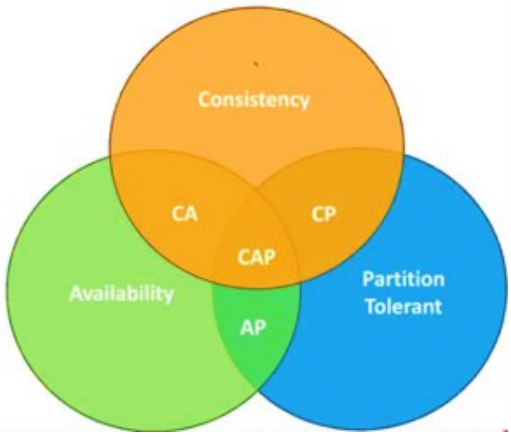


Fig. 4. Characteristics of NoSQL

- **Availability and Partition Tolerance (AP):** These systems provide eventual consistency rather than strong consistency, however, they promise superior partition tolerance and availability [28].
- **Consistency and Availability (CA):** The database ensures consistency and availability under normal conditions; however, it does not provide partition tolerance, making it less suitable for distributed environments.
- **Consistency and Partition tolerance (CP):** The database guarantees high consistency and supports partition tolerance in distributed systems, but availability may be reduced during network failures.

## VI. Literature of Review

In this section, the performance, scalability, security, and new trends in database systems, such as the introduction of AI and cloud-native architecture, are analyzed, and the main compromises between NoSQL and SQL systems are noted. The studies are summarized in Table III, and include their areas of focus, methods and findings, as well as limitations.

D. Jawale, S. Shelke, and S. Yadav, (2026) database architecture's impact on the system performance of NoSQL (non-relational) and conventional SQL (relational) database models. Transactional integrity, optimized and complex query processors are SQL databases, which are structured in a schema and are ACID-based. NoSQL databases, on the other hand, emphasize high availability, flexible data models, and horizontal scalability, making them perfect for large-scale data and real-time applications. The research applies a benchmarking methodology with the use of representative systems, such as PostgreSQL (SQL) and MongoDB/Cassandra (NoSQL) and tested in the conditions of the read-intensive, write-intensive, and mixed-workload [29].

S. Vikram (2025) discusses role of AI and ML in changing design, implementation, and SQL and NoSQL database operation in a dispersed manufacturing context. It explores the architecture development, automated query optimisation, adaptive data partitioning, anomaly detection in sensor data, and predictive indexing. The study brings out practical frameworks and implementations that have the potential of closing gap between data systems and intelligent agents, through a detailed review and technical analysis [30].

V. Panwar,(2024) examines SQL and NoSQL database security flaws, backed by recent cases of attacks and exploits that demonstrate the threat to data integrity and confidentiality. It discusses SQL injection attacks, lack of proper access controls, and NoSQL environment vulnerabilities, and offers a thorough analysis of security vulnerabilities that can be exploited to compromise sensitive data [31].

N. Kumar Miryala,(2024) provides a comparative overview of majority of recent developments in database technology, such as serverless, cloud-native, and multi-model databases. Additionally, it discusses potential effects of immutable databases and blockchain on data security and transparency, as well as how distributed and edge databases might enhance performance and accessibility [32].

S. E. A. Youssef (2023) provides a detailed discussion of database performance optimization in large-scale environments, the challenges associated with such operations, and approaches to addressing them. The traditional SQL databases, which are commonly used in the enterprise environment, are based on ACID adherence and permit an organized data management system, but often have little flexibility and scalability. Conversely, NoSQL databases are designed to manage distributed systems and unstructured data, as well as to provide scalability and performance benefits at the cost of possible trade-offs in consistency and transactional reliability [33].

V. K. Jangala,(2022) provides a thorough explanation of relational and NoSQL databases in relation to business systems, architecture, data model, consistency, performance characteristics and trade-offs. It compares the strengths of relational databases in structured, transaction-intensive, compliance-driven environments, to strengths of NoSQL databases in offering better scalability, availability and flexibility to high-throughput and distributed loads [34].

 

TABLE III. COMPARATIVE SUMMARY OF STUDIES ON DATABASE ADMINISTRATION ACROSS SQL AND NOSQL SYSTEM

| Authors (Year) | Focus Area | Objectives | Key Performance Findings | Approaches | Future limitations |
|---|---|---|---|---|---|
| D. Jawale, S. Shelke, and S. Yadav (2026) | SQL vs NoSQL performance benchmarking | Compare relational and non-relational databases under different workloads | SQL performs better in transactional consistency; NoSQL excels in scalability and high-throughput scenarios | Experimental benchmarking using PostgreSQL, MongoDB, Cassandra | Limited exploration of hybrid database models and real-time adaptive workload optimization |
| S. Vikram (2025) | AI & ML in database systems | Explore integration of AI/ML for automation and optimization in SQL/NoSQL | AI improves query optimization, indexing, and anomaly detection efficiency | Review of AI-driven frameworks and intelligent database systems | Lack of standardized AI integration frameworks and real-world deployment validation |
| V. Panwar (2024) | Database security (SQL & NoSQL) | Identify vulnerabilities and evaluate defense mechanisms | Both systems face critical risks; NoSQL introduces new attack surfaces | Security analysis of known attack vectors and case studies | Insufficient focus on unified security frameworks across hybrid and cloud databases |
| N. Kumar Miryala (2024) | Modern database trends & cloud systems | Analyze innovations in cloud-native, multi-model, and distributed databases | Multi-model and cloud-native systems improve flexibility and scalability | Comparative and trend-based analysis | Need for performance evaluation in edge and serverless environments under real-world constraints |
| S. E. A. Youssef, (2023) | Database performance optimization | Optimize performance in large-scale enterprise environments | SQL ensures reliability; NoSQL enhances scalability with trade-offs | Analytical and methodological study | Lack of adaptive optimization techniques for dynamic and heterogeneous workloads |
| V. K. Jangala (2022) | Comparative review of SQL vs NoSQL in enterprises | Evaluate architectures, consistency models, and trade-offs | Relational databases excel in structured tasks; NoSQL supports distributed workloads effectively | Comparative review of database models | Limited investigation into interoperability and integration of SQL–NoSQL systems |

## VII. CONCLUSION AND FUTURE WORK

Database systems too have changed to accommodate the emerging demands of the new data-driven applications. Although other systems can be developed that can also achieve high consistency, good data manipulation and transactional integrity, SQL databases are still utilized to perform operations on a database e.g. performance tuning, query optimization, indexing, replication and monitoring in an enterprise system. However, NoSQL databases provide greater flexibility and scalability, and more easily scale to unstructured and large-scale distributed data, this is what is required in real-time, cloud and large-data apps. The comparative analysis reveals that the two database models have certain strengths and weaknesses and functionality of the models highly depends on the needs and nature of the workload of the application. Recent innovations like cloud-native applications, Database-as-a-Service (DBaaS), AI-based automation and distributed systems are also improving the performance, scalability and efficiency of the databases. These trends are forming a more agile and smart database ecosystem.

Future research could be directed towards a hybrid database model that incorporates the benefits of NoSQL and SQL, autonomous database management system, based on AI, and increased security of the distributed environment. This will enhance more efficient, scalable and secure data management in next-generation applications.

## REFERENCES


[1] I. Qaddara, Y. Alraba'nah, and M. O. Hiari, "Evaluation of SQL and NoSQL Databases on Parallel Processing," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 4, pp. 24298–24304, Aug. 2025, doi: 10.48084/etasr.10620.

[2] J. W. Sajja, G. B. Komarina, and N. K. R. Choppa, "The Convergence of Financial Efficiency and Sustainability in Enterprise Cloud Management," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 4, pp. 964–992, May 2025, doi: 10.32996/jcsts.2025.7.4.110.

[3] S. Singh, S. A. Pahune, P. Chatterjee, and R. Sura, "Advanced Machine Learning Techniques for Prediction of Customer Churn in Telecommunication Sector," in *2025 IEEE 6th India Council International Subsections Conference (INDISCON)*, Rourkela, India: IEEE, 2025, pp. 1–6, August. doi: 10.1109/INDISCON66021.2025.11252233.

[4] A. Nayak, A. Poriya, and D. Poojary, "Type of NOSQL databases and its comparison with relational databases," *Int. J. Appl. Inf. Syst.*, vol. 5, no. 4, pp. 16–19, 2013.

[5] I. Mapanga and P. Kadebu, "Database Management Systems : A NoSQL Analysis," *Int. J. Mod. Commun. Technol. Res.*, vol. 1, no. 7, pp. 12–18, 2013.

[6] M. C. Okoronkwo, M. Mustapha, C. Markus, M. A. Adaji, and N. E. Chibunze, "A Review on Object-Oriented, Object-Relational, Relational, and NoSQL Databases: Evolution, Integration, and Performance Trade-Offs," *Open J. Phys. Sci. (ISSN 2734-2123)*, vol. 6, no. 2, pp. 61–78, Dec. 2025, doi: 10.52417/ojps.v6i2.1039.

[7] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," *J. Cloud Comput. Adv. Syst. Appl.*, vol. 2, no. 1, p. 22, Dec. 2013, doi: 10.1186/2192-113X-2-22.

[8] S. Binani, A. Gutti, and S. Upadhyay, "SQL vs. NoSQL vs. NewSQL-A comparative study," *Database*, vol. 6, no. 1, pp. 1–4, 2016.

[9] A. R. Toorpu, S. K. Vududala, A. Nerella, and B. P. Madupati, "Hybrid AI Models for Privacy-Preserving Big Data Analytics in Distributed Environments," in *2025 Global Conference in Emerging Technology (GINOTECH)*, IEEE, May 2025, pp. 1–8. doi: 10.1109/GINOTECH63460.2025.11076666.

[10] D. M. P, "Database Management Systems as a Core Technology Integrating Multiple Sectors in the Digital Era …," *Int. J. Sci. Res. Eng. Trends*, vol. 12, no. 2, pp. 1–4, 2026.

[11] A. K. Zaki, "NoSQL Databases: New Millennium Database for Big Data, Big Users, Cloud Computing and its Security Challenges," *Int. J. Res. Eng. Technol.*, vol. 03, no. 15, pp. 403–409, May 2014, doi: 10.15623/ijret.2014.0315080.

[12] S. Bjeladinovic, "A fresh approach for hybrid SQL/NoSQL database design based on data structuredness," *Enterp. Inf. Syst.*, vol. 12, no. 8–9, pp. 1202–1220, Oct. 2018, doi: 10.1080/17517575.2018.1446102.

[13] P. Parida and N. Senguttuvan, "Responsible Utilization of Cloud in Retail Banking Ecosystem," *Int. J. Comput. Appl.*, vol. 187, no. 49, pp. 34–39, 2025, doi: 10.5120/ijca2025925835.

[14] M. Salahuddin, S. Majeed, S. Hira, and G. Mumtaz, "A Systematic Literature Review on Performance Evaluation of SQL and NoSQL Database Architectures," *J. Comput. Biomed. Informatics*, vol. 7, no. 2, 2024.

[15] M. A. Mohamed, O. G. Altrafi, and M. O. Ismail, "Relational vs . NoSQL Databases : A Survey," *Int. J. Comput. Inf. Technol.*, vol. 3, no. 3, 2014.

[16] S. Sharma, R. Shandilya, S. Patnaik, and A. Mahapatra, "Leading


 

NoSQL models for handling Big Data: a brief review," *Int. J. Bus. Inf. Syst.*, vol. 22, no. 1, p. 1, 2016, doi: 10.1504/IJBIS.2016.075714.

[17] W. Kim, "Web Data Stores (aka NoSQL databases): A Data Model and Data Management Perspective," *Int. J. Web Grid Serv.*, vol. 10, no. 1, p. 100, 2014, doi: 10.1504/IJWGS.2014.058774.

[18] A. Singh, "NoSQL : A New Horizon in Big Data," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 1183–1188, 2016.

[19] A. Mittal and I. Kumar, "Trust-Aware Safe Reinforcement Learning and Graph Neural Surrogates for Real-Time Power Grid Management," in *2026 International Conference on Electronics and Renewable Systems (ICEARS)*, IEEE, Feb. 2026, pp. 1080–1085. doi: 10.1109/ICEARS67481.2026.11416721.

[20] K. Katti and Archana K, "Evolving Trends in NoSQL Databases: A Comparative Study of Scalability, Performance, and Use Cases," *Int. J. Eng. Res. Comput. Sci. Eng.*, vol. 10, no. 11, p. 50, 2023.

[21] A. Tikotikar, K. Sakibaev, U. Srilakshmi, and R. Kiran P, "Innovative Study On Database Management Systems In Modern Applications," *Int. J. Adv. SIGNAL IMAGE Sci.*, vol. 12, no. 2s, pp. 199–207, Feb. 2026, doi: 10.29284/n0s4rk20.

[22] T. Shah, "Cloud-Based Data Warehousing for Marketing Agility: Lessons from FinTech Migrations to Snowflake and AWS," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 4, pp. 1–11, 2024.

[23] M. Abourezq and A. Idrissi, "Database-as-a-Service for Big Data: An Overview," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, pp. 157–177, 2016, doi: 10.14569/ijacsa.2016.070124.

[24] A. Nerella and J. W. Sajja, "Responsible AI in Enterprise Applications: Balancing Innovation and Compliance," *Comput. Fraud Secur.*, vol. 2023, no. 7, pp. 43–52, Jul. 2023, doi: 10.52710/cfs.744.

[25] P. O'Sullivan, G. Thompson, and A. Clifford, "Applying data models to big data architectures," *IBM J. Res. Dev.*, vol. 58, no. 5/6, pp. 18:1-18:11, Sep. 2014, doi: 10.1147/JRD.2014.2352474.

[26] V. K. Tambi, "Analysis of Sql and Nosql Database Management," *Int. J. Curr. Eng. Sci. Res.*, vol. 2, no. 3, pp. 99–113, 2015.

[27] Y. Jani, "The Role of SQL and NoSQL Databases in Modern Data Architectures," *Int. J. Core Eng. Manag.*, vol. 6, no. 12, pp. 61–67, 2021.

[28] A. Makris, K. Tserpes, V. Andronikou, and D. Anagnostopoulos, "A Classification of NoSQL Data Stores Based on Key Design Characteristics," *Procedia Comput. Sci.*, vol. 97, pp. 94–103, 2016, doi: 10.1016/j.procs.2016.08.284.

[29] D. Jawale, S. Shelke, and S. Yadav, "Analyzing the Impact of Database Architecture on Performance: SQL vs NoSQL," *Int. J. Math. Comput. Res.*, vol. 14, no. 03, Mar. 2026, doi: 10.47191/ijmcr/v14iSPC3.13.

[30] S. Vikram, "Modernizing Data Infrastructure: How AI and ML are Transforming SQL and NoSQL Usage in Distributed Manufacturing," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 62s, pp. 1119–1134, 2025.

[31] V. Panwar, "Security Challenges and Solutions in Web Development-Protecting Data in SQL and NoSQL Databases," *Int. J. Eng. Appl. Sci. Technol.*, vol. 8, no. 11, pp. 15–24, 2024.

[32] N. K. Miryala, "Emerging Trends and Challenges in Modern Database Technologies: A Comprehensive Analysis," *Int. J. Sci. Res.*, vol. 13, p. 9, 2024, doi: 10.21275/MS241126103744.

[33] S. E. A. Youssef, "Optimizing Database Performance for Large-Scale Enterprise Applications: A Comprehensive Study on Techniques, Challenges, and the Integration of SQL and NoSQL Databases in Modern Data Architectures," *J. Artif. Intell. Mach. Learn. Manag.*, vol. 7, no. 1, pp. 81–92, 2023.

[34] V. K. Jangala, "Relational and NoSQL databases in enterprise systems," *Int. J. Contemp. Res. Multidiscip.*, vol. 1, no. 1, pp. 125–131, 2022.

 