IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 24 March 2025, accepted 15 May 2025, date of publication 21 May 2025, date of current version 5 June 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3572059

SURVEY

Database Systems in the Big Data Era: Architectures, Performance, and Open Challenges

ELIAS DRITSAS AND **MARIA TRIGKA**
Department of Informatics and Computer Engineering, University of West Attica, Egaleo Park Campus, 122 43 Athens, Greece
Corresponding author: Elias Dritsas (idritsas@uniwa.gr)

The publication of the article in OA mode was financially supported by HEAL-Link.

ABSTRACT The advent of Big Data has fundamentally transformed the database management systems (DBMS) field, necessitating the development of innovative paradigms, architectures, and technologies to address unprecedented challenges. Despite their historical dominance, traditional systems falter under the high velocity, massive volume, and diverse variety of Big Data. These limitations have catalyzed the emergence of alternative solutions such as not only structured query language (NoSQL), NewSQL, and cloud-native databases, each offering unique approaches to scalability, flexibility, and performance optimization. This survey provides a comprehensive and systematic overview of the evolving database ecosystem in the Big Data era. It delves into the historical progression from traditional relational DBMS (RDBMS) to modern paradigms, emphasizing these transformations' motivations, trade-offs, and innovations. The classification of databases based on data models, deployment strategies, scalability mechanisms, and consistency models is explored in depth, providing a structured framework for understanding their diverse capabilities. Furthermore, critical performance characteristics, including throughput, latency, fault tolerance, and cost efficiency, are analyzed to assess their effectiveness in real-world applications. By highlighting persistent challenges such as data heterogeneity, security, and interoperability, this survey outlines key research directions, fostering a holistic understanding of the domain and inspiring future advancements in database technologies.

INDEX TERMS Big data, databases, performance optimization, scalability, heterogeneity.

I. INTRODUCTION

The exponential growth of data over recent decades, often described as the "Big Data revolution", has fundamentally transformed the landscape of database management. The proliferation of digital devices, primarily driven by the Internet of Things (IoT) and social media platforms, is driving this transformation. The concept of Big Data, characterized by its "3Vs", namely volume, velocity, and variety, has rendered traditional DBMS increasingly inadequate. Despite their robustness and ubiquity, RDBMS face scalability, schema flexibility limitations, and the handling of unstructured or semi-structured data [1], [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Ayman El-Baz.

Addressing these limitations requires a departure from conventional practices. The rise of NoSQL databases demonstrates the necessity for horizontal scalability and flexibility in schema design. NewSQL, in contrast, seeks to retain the ACID (Atomicity, Consistency, Isolation, Durability) properties of traditional systems while achieving the scalability demanded by Big Data workloads. Meanwhile, cloud-native databases capitalize on the distributed nature of cloud computing to deliver elastic scaling and enhanced fault tolerance [3], [4].

Real-world needs have driven the adoption of these technologies. Applications in e-commerce, healthcare, finance, and social media generate diverse datasets that require real-time processing and analysis. Traditional RDBMS often struggles with these workloads due to its rigid schema requirements and reliance on vertical scaling. The innovative architectures of NoSQL, NewSQL, and cloud-native databases provide the tools necessary to meet these industries' evolving needs [5].

A. MOTIVATION

The rapid expansion of the database landscape has introduced a wide array of architectures, features, and trade-offs. While these developments have addressed critical limitations of traditional relational systems, particularly in terms of scalability, flexibility, and data heterogeneity, they have also increased the complexity of understanding and evaluating modern solutions. The diversity of paradigms, from NoSQL and NewSQL to cloud-native models, presents challenges in assessing their relative advantages, limitations, and suitability for specific application contexts.

Much of the existing discourse in the field has explored individual technologies or emphasized particular dimensions such as scalability or consistency. However, a cohesive and up-to-date synthesis that captures the architectural evolution, performance considerations, and classification of database systems across multiple dimensions remains limited. In particular, few comprehensive resources align technical characteristics with real-world deployment strategies and emerging research needs.

This survey is motivated by the need to address that gap. It offers an integrated and structured perspective on modern database systems in the Big Data era, connecting foundational concepts with practical evaluations. This work aims to foster a more precise and actionable understanding of the current database ecosystem by organizing dispersed insights and highlighting persistent challenges.

B. METHODOLOGY

This survey adopts a structured and integrative methodology to examine modern database systems' evolution, classification, and performance aspects in the Big Data era. The study is grounded in a comprehensive review of literature published after 2018, a period that marks a critical phase in the database landscape with the rise of cloud-native and serverless architectures, NewSQL platforms, and increasing emphasis on multi-model capabilities, cost-efficiency, and sustainability. This timeframe was chosen to reflect the shift in research and industry priorities toward scalable, intelligent, and flexible data systems, all of which are central to the classification, performance evaluation, and future directions discussed in this survey.

The collected material was systematically analyzed to extract recurring architectural patterns, design principles, and performance considerations observed across modern database systems. This analysis informed the development of the database classification framework, which is structured around four primary dimensions: data models, deployment strategies, scalability features, and consistency models. The framework was refined to balance conceptual clarity with practical relevance across diverse application domains.

In parallel, performance-related characteristics, such as throughput, latency, fault tolerance, and cost efficiency, were synthesized from published benchmarks, technical documentation, and empirical studies. The survey also identifies key challenges in scalability, data heterogeneity, security, and interoperability, as well as emerging research directions focused on multi-model systems, artificial intelligence (AI) integration, and sustainability.

Moreover, Figure 1 supports this methodology by illustrating the logical flow of the survey and the interrelations among key topics. It maps the transition from traditional relational databases to modern paradigms and visually anchors the classification framework within the broader context of architectural evolution, performance analysis, and open research challenges. This structure provides a coherent roadmap for understanding and comparing the capabilities of database systems in the Big Data era.

C. CONTRIBUTION

The contributions of this survey go beyond existing works by providing a structured and comprehensive analysis of database technologies in the Big Data era. This work presents an integrated perspective on evolution, classification, and performance characteristics of modern databases. Specifically, this survey:

- Traces the progression from traditional RDBMS to modern database paradigms, identifying the technical and contextual shifts that shaped each innovation. Offers a detailed classification framework based on critical attributes such as data models, deployment strategies, scalability mechanisms, and consistency approaches.
- Provides an in-depth examination of database performance characteristics, including latency, throughput, and fault tolerance, tailored to diverse real-world scenarios.
- Highlights unresolved challenges and emerging opportunities, such as data interoperability and adaptive consistency models, guiding further advancements in the field.

The remaining paper is organized as follows. Section II presents a classification framework for modern databases based on data models, deployment strategies, scalability features, and consistency models. Next, Section III delves into the evolution of databases, covering traditional, NoSQL, NewSQL, and cloud-native paradigms. Furthermore, Section IV explores critical performance characteristics. Section V identifies the key challenges facing databases in the Big Data era. Moreover, Section VI outlines future research directions to address the limitations of current systems and leverage emerging technologies. Section VII interprets the findings and compares them with existing research works. Finally, Section VIII summarizes and concludes the survey.

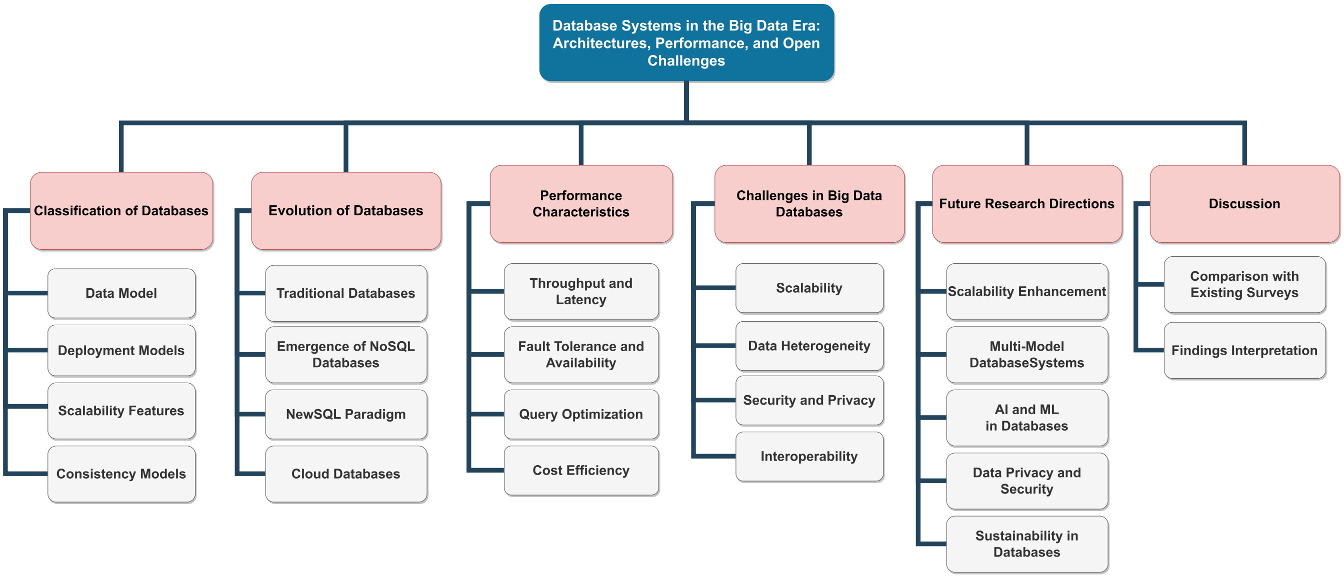


FIGURE 1. **Logical structure of the survey, mapping out the key topics and their interrelations.**

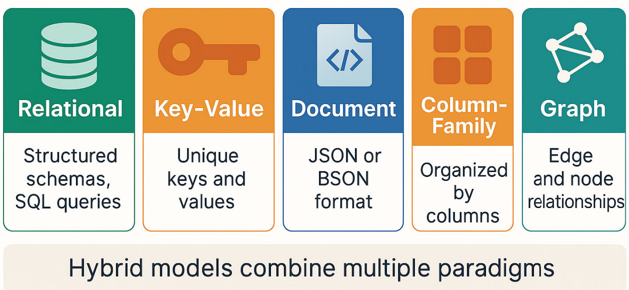


FIGURE 2. **Data models taxonomy.**

II. CLASSIFICATION OF DATABASES

Database classification in the Big Data era requires a multidimensional framework that reflects the design principles, operational contexts, and application domains unique to modern systems. This section provides an expanded exploration of the primary axes of classification, offering an enriched understanding of their scientific and practical implications.

A. DATA MODEL

Data models in Figure 2 define the fundamental structure of a database, determining how data is stored, accessed, and manipulated. Relational models, underpinned by structured schemas and SQL queries, dominate traditional transactional systems where data consistency and integrity are critical. However, their limitations in handling unstructured data and dynamic workloads have spurred the adoption of alternative models [6], [7].

Key-value databases simplify data organization, associating unique keys with corresponding values. This model is ideal for applications requiring high-speed lookups, such as caching and real-time analytics. Document-oriented databases like MongoDB expand on this concept, enabling storing semi-structured data in formats such as JSON or BSON. These systems excel in domains requiring flexible schema designs, such as content management and mobile applications [8], [9], [10].

Column-family databases, including Apache Cassandra and HBase, optimize for high write-throughput and scalability by organizing data into columns rather than rows. This makes them particularly suitable for large-scale analytical workloads like time-series data processing. Graph databases, exemplified by Neo4j and Amazon Neptune, provide an edge-centric data model, facilitating complex relationship queries in social networks, recommendation systems, and knowledge graphs. The emergence of hybrid models reflects an evolving need to integrate multiple paradigms, combining the strengths of relational, key-value, and graph-based systems within a unified framework [11], [12], [13], [14].

B. DEPLOYMENT MODELS

As shown in Figure 3, deployment models shape how databases are hosted, managed, and accessed. Traditional on-premises deployments offer maximum control over infrastructure, making them indispensable for industries with stringent security and compliance requirements. However, they often entail high operational costs and limited scalability, particularly under variable workloads [15], [16].

Cloud-native databases represent a transformative shift, leveraging the scalability and flexibility of cloud computing. These systems decouple compute and storage resources, enabling elastic scaling and seamless integration with other cloud services. Providers like Amazon, Google, and

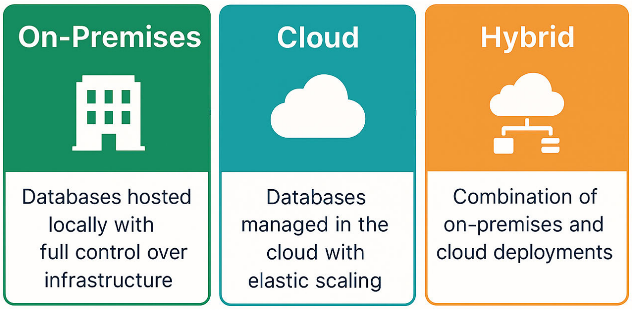


FIGURE 3. Deployment models for database systems.

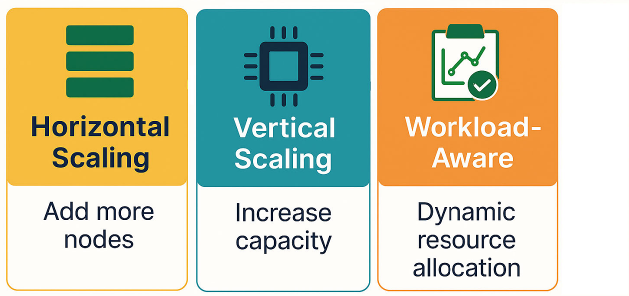


FIGURE 4. Scalability strategies in modern databases.

Microsoft offer managed database solutions with automated backups, disaster recovery, and global replication features. This accessibility has democratized database management, allowing organizations of all sizes to benefit from enterprise-grade infrastructure [17], [18], [19].

Hybrid deployments bridge the gap between on-premises and cloud systems, combining the control of local infrastructure with the scalability of cloud platforms. This model is particularly advantageous for organizations seeking to comply with data sovereignty regulations while leveraging cloud technologies' cost efficiency and flexibility. As workloads continue to diversify, adopting hybrid models is expected to grow, addressing the needs of modern, distributed applications [20], [21].

C. SCALABILITY FEATURES

Scalability is a defining characteristic of modern database systems, reflecting their ability to accommodate growing data volumes and user demands. Figure 4 illustrates the classification of databases according to their scalability strategies. Horizontal scaling, achieved by adding more nodes to a cluster, is the cornerstone of systems like NoSQL and NewSQL. Techniques such as sharding and replication enable efficient data distribution and fault tolerance, ensuring high availability and performance under heavy workloads [22], [23].

Vertical scaling, involving enhancing hardware capabilities, remains relevant for specific high-performance scenarios. However, its cost and complexity often limit its feasibility for large-scale systems. Advances in dynamic scaling mechanisms, which adapt resource allocation in real-time, are bridging the gap between horizontal and vertical approaches, enhancing system responsiveness while optimizing resource utilization [24], [25], [26].

Workload-aware resource allocation represents an emerging frontier in scalability research. These techniques promise to improve efficiency and reduce costs by predicting workload patterns and preemptively adjusting resources. As database systems become increasingly complex, scalability solutions must address both the technical and economic challenges of managing distributed, high-demand environments [27], [28].

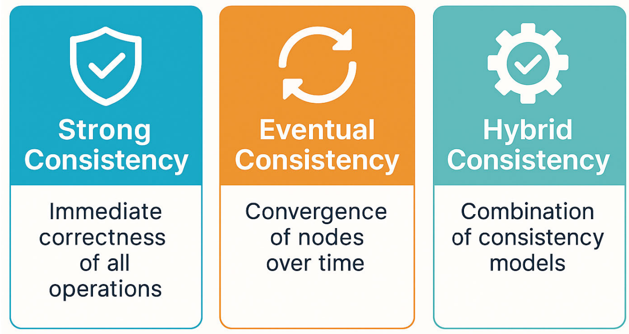


FIGURE 5. Consistency models in distributed databases.

D. CONSISTENCY MODELS

Consistency models (Figure 5) govern the behavior of distributed databases, balancing the trade-offs between performance, availability, and data accuracy. Strong consistency ensures immediate correctness of all read and write operations, making it essential for transactional systems requiring strict integrity. However, achieving this level of consistency often incurs higher latency and resource overhead, particularly in geographically distributed environments [29], [30], [31].

Eventual consistency, favored by many NoSQL databases, sacrifices immediate correctness for improved performance and availability. This model ensures that all nodes converge to a consistent state over time, making it suitable for applications where slight delays in data synchronization are acceptable, such as social media feeds or e-commerce catalog updates. Adaptive consistency approaches aim to adjust consistency levels based on application requirements dynamically, providing a balance between performance and accuracy [32], [33], [34], [35].

Hybrid consistency models, which combine strong and eventual consistency elements, represent a promising direction for future research. By offering configurable consistency guarantees, these models enable databases to cater to a broader range of use cases, addressing the diverse needs of modern applications. As distributed systems continue to evolve, consistency models will remain a critical area of innovation, shaping the future of database design [36], [37].

Table 1 provides a comprehensive classification of modern databases based on key attributes, including data models, deployment strategies, scalability features, and consistency models. It categorizes different database paradigms, outlining their characteristics, strengths, and trade-offs to facilitate a deeper understanding of their applicability across various cases.

III. EVOLUTION OF DATABASES

The evolution of databases reflects a progressive adaptation to the shifting demands of data-centric applications. From the dominance of traditional relational databases to the rise of NoSQL, NewSQL, and cloud-native systems, each phase in this evolution represents a response to specific technological and operational challenges. This section provides an in-depth examination of the stages of database evolution, emphasizing the underlying motivations and technological advancements.

A. TRADITIONAL DATABASES

Relational databases have been the cornerstone of data management since their formal introduction. Based on the relational model, these systems emphasize consistency, reliability, and structured data organization. The adoption of SQL as the standard for querying and managing relational data cemented their position as a cornerstone of enterprise computing [38].

Despite their robustness, traditional RDBMS faced significant limitations as data volumes grew exponentially. Vertical scaling, the process of improving performance by upgrading hardware, often proved insufficient due to its inherent cost and complexity. Moreover, rigid schema requirements restricted their ability to handle unstructured or semi-structured data, making them less suitable for dynamic applications like social media analytics or real-time sensor data processing [39], [40].

Efforts to address these challenges included the development of distributed relational databases and improving indexing techniques. Technologies such as in-memory processing and columnar storage formats enhanced query performance, particularly for analytical workloads. However, these incremental improvements fell short of addressing the fundamental limitations of the relational paradigm when faced with the scale and diversity of Big Data [41], [42].

The limitations of traditional databases provided the impetus for innovation. The emergence of NoSQL and NewSQL databases was driven by the need for systems that could scale horizontally, handle diverse data types, and provide real-time insights. These new paradigms have redefined the database landscape, offering solutions tailored to the unique demands of Big Data applications [43]. An illustrative summary of the key characteristics, strengths, and limitations of traditional relational databases is presented in Figure 6.

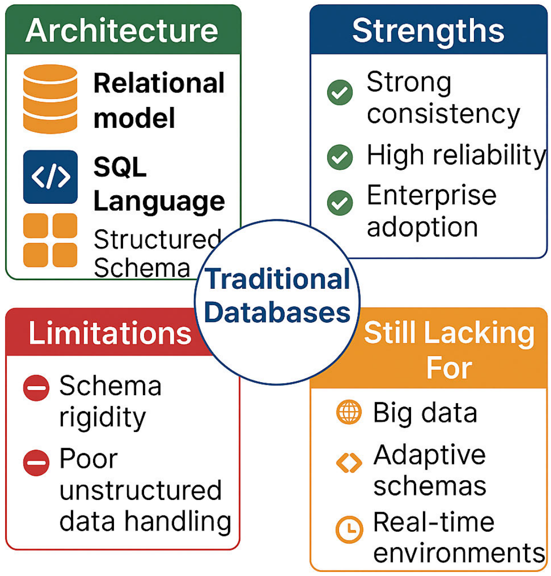


FIGURE 6. Traditional Databases: Characteristics and Limitations.

B. EMERGENCE OF NoSQL DATABASES

NoSQL databases emerged as a response to the scalability and flexibility limitations of traditional RDBMS. Designed to handle the massive scale and heterogeneity of modern data, NoSQL systems prioritize horizontal scalability and schema flexibility. Unlike relational databases, NoSQL systems typically forgo strict schema definitions, allowing for dynamic data organization and processing [44], [45].

A key strength of NoSQL databases is their support for diverse data models. Key-value stores like Redis and Riak provide a simple yet effective mechanism for associating unique keys with arbitrary data values. Document-oriented databases like MongoDB and CouchDB enable storing and retrieving semi-structured data in JSON-like documents, offering a balance between flexibility and structure. Column-family databases, such as Apache Cassandra and HBase, excel at handling wide-row datasets and high-throughput applications, making them ideal for analytical workloads. Graph databases, exemplified by Neo4j and Amazon Neptune, are uniquely suited for applications involving complex relationships, such as social network analysis and fraud detection [46], [47], [48], [49].

The scalability of NoSQL systems is achieved through techniques like sharding and replication. Sharding involves partitioning data across multiple nodes, while replication ensures data availability and fault tolerance. These strategies enable NoSQL databases to maintain high performance under heavy workloads. However, the emphasis on eventual consistency introduces challenges in scenarios requiring strict transactional guarantees [50], [51].

Research into hybrid models seeks to address these trade-offs. Combining NoSQL's scalability with the consistency guarantees of relational systems aims to deliver the

TABLE 1. Classification of modern databases by key attributes, strengths, and trade-offs.

Topic	Reference	Description	Strengths & Suitability	Limitations & Trade-offs
Relational Data Model	[6] [7]	Highlights the traditional structured schema-based approach using SQL, emphasizing data consistency and integrity.	Ensures strong ACID compliance, data integrity, and structured querying. Ideal for transaction-heavy applications like banking and enterprise systems.	Scalability challenges with vertical scaling, rigid schema restricts flexibility, and unstructured data is handled poorly.
Key-Value Databases	[8]	Explores simple key-value pair storage optimized for high-speed lookups, ideal for caching and real-time analytics.	Low-latency data retrieval, excellent for caching (e.g., Redis), simplistic storage model for fast access.	Lacks complex querying capabilities, inefficient for structured relationships, and eventual consistency may cause temporary discrepancies.
Document-Oriented Databases	[9] [10]	Discusses flexible schema designs for semi-structured data storage, suitable for content management systems.	Flexible schema supports semi-structured data (e.g., JSON in MongoDB), which is great for dynamic content.	Indexing complexity can impact performance, not ideal for complex relational queries.
Column-Family Databases	[11]	Focuses on high write-throughput and scalability for analytical workloads such as time-series data processing.	Optimized for big data analytics, excels in high-volume writes, and is widely used in IoT, logs, and event processing.	Limited transactional support, harder query optimization, requires manual tuning for efficiency.
Graph Databases	[12] [13] [14]	Explores relationship-centric data models optimized for queries in social networks and recommendation systems.	Efficient relationship traversal for connected data (e.g., social networks, fraud detection). A flexible schema allows dynamic structures.	High memory consumption, query performance degrades with deep traversal, and lacks standardization across different graph DBs.
On-Premises Deployments	[15] [16]	Discusses maximum control and security for industries with stringent compliance needs despite high costs and limited scalability.	Full data control, better security, compliance with strict regulatory standards (e.g., healthcare, government).	High infrastructure and maintenance costs, limited elasticity, and requires expert management.
Cloud-Native Databases	[17] [18] [19]	Explores elastic scaling, global replication, and seamless integration with cloud services for cost efficiency and scalability.	Highly scalable, multi-region replication, lower operational costs with managed services.	Vendor lock-in risks, latency variations, and potential data sovereignty issues.
Hybrid Deployment Models	[20] [21]	Highlights the balance between on-premises control and cloud flexibility, addressing compliance and scalability needs.	Combines on-premises security with cloud scalability, ideal for sensitive data with dynamic workloads.	Higher complexity requires careful integration to avoid latency bottlenecks and data inconsistency.
Horizontal Scaling	[22] [23]	Covers sharding and replication techniques for distributing data and workloads across multiple nodes.	Enables high scalability and fault tolerance through replication, commonly used in NoSQL and distributed databases.	Data partitioning complexities, consistency trade-offs, and higher networking overhead.
Vertical Scaling	[24] [25] [26]	Discusses enhancing hardware capabilities to improve performance, emphasizing cost and complexity constraints.	Useful for workloads requiring strong consistency, supports legacy applications without major redesign.	Expensive hardware upgrades, limited upper scaling potential, and a single point of failure.
Dynamic Scaling Mechanisms	[27] [28]	Explores real-time resource allocation using workload-aware and predictive techniques for efficiency.	Optimizes cost by dynamic scaling, reduces idle resource waste, and improves elasticity in cloud environments.	Difficult in predicting workload spikes, real-time monitoring overhead can be resource-intensive.
Strong Consistency	[29] [30] [31]	Ensures immediate correctness of all operations, suitable for transactional systems requiring strict integrity.	Essential for financial, healthcare, and compliance-driven applications where correctness is critical.	High performance overhead, increased latency in distributed environments, and scalability limitations.
Eventual Consistency	[32] [33] [34] [35]	Focuses on performance and availability, with eventual synchronization across nodes for applications like social media feeds.	Improves availability and scalability, reduces operational costs, and is commonly used in NoSQL and distributed systems.	Data inconsistencies in read operations may require conflict resolution mechanisms, which are unsuitable for critical transactional data.
Hybrid Consistency Models	[36] [37]	Combines elements of strong and eventual consistency, offering configurable guarantees for diverse applications.	Balances consistency and availability, configurable based on application needs, best for multi-region applications.	Implementation complexity, higher maintenance costs, and requires sophisticated consensus mechanisms.

best of both worlds. As the adoption of NoSQL grows, these innovations will play a critical role in defining the next generation of database systems [52]. An overview of the key aspects of NoSQL databases is illustrated in Figure 7.

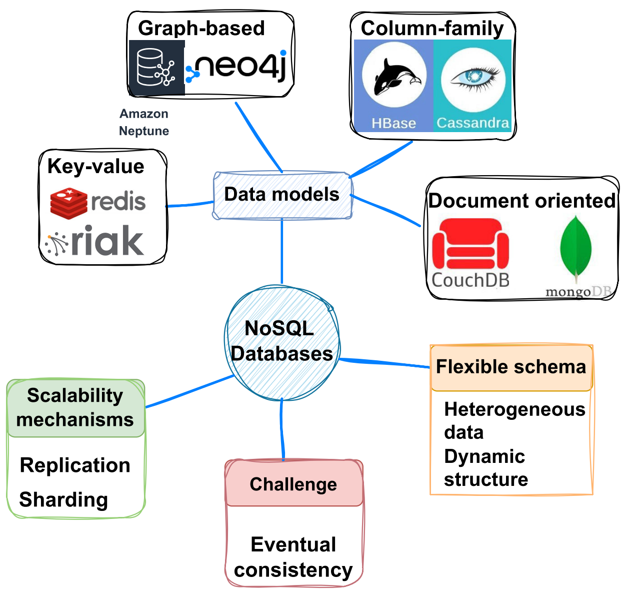


FIGURE 7. Key aspects of NoSQL databases: data model diversity, scalability mechanisms, schema flexibility, and challenges.

C. NEWSQL PARADIGM

NewSQL databases aim to bridge the divide between the scalability of NoSQL and the transactional guarantees of traditional RDBMS. By combining distributed architectures with ACID compliance, NewSQL systems offer an appealing compromise for applications requiring both consistency and high performance. Google Spanner exemplifies this approach, employing global synchronization mechanisms to achieve strong consistency across distributed nodes. Similarly, VoltDB and CockroachDB leverage in-memory processing and parallel transaction execution to deliver exceptional throughput [53], [54], [55].

The adoption of NewSQL databases is driven by their ability to handle hybrid workloads, integrating the high velocity of online transaction processing (OLTP) with the analytical depth of online analytical processing (OLAP). This makes them particularly suitable for industries like finance, e-commerce, and telecommunications, where data integrity and real-time decision-making are paramount. Despite their advantages, NewSQL systems face challenges in balancing the complexity of distributed transactions with the need for low-latency operations [56], [57], [58].

FIGURE 8. **Capabilities and architecture of NewSQL databases.**

The technical innovations in NewSQL systems, including distributed consensus algorithms and advanced query optimization, have set a new benchmark for database performance. However, their adoption remains constrained by the complexity of deployment and the need for specialized expertise. Addressing these barriers will be critical for the widespread adoption of NewSQL in diverse application domains [59], [60]. Figure 8 summarizes the core features, capabilities, and challenges of NewSQL databases

D. CLOUD DATABASES

Cloud-native databases signify a paradigm shift in database management, leveraging the scalability and resilience of cloud infrastructures. By decoupling storage and compute resources, these systems enable elastic scaling, accommodating fluctuations in workload demand without significant operational overhead. Services such as Amazon DynamoDB, Google Bigtable, and Azure Cosmos DB exemplify this trend, offering managed solutions with global availability and automated maintenance [61], [62], [63], [64].

Cloud databases democratize access to enterprise-grade capabilities, enabling organizations of all sizes to deploy, scale, and optimize their data systems with minimal upfront investment. Advanced features, such as multi-region replication and serverless architecture, enhance fault tolerance and reduce latency, meeting the demands of modern, globally distributed applications. However, these benefits are accompanied by challenges, including data residency regulations, vendor lock-in concerns, and the complexities of hybrid cloud integration [65], [66], [67].

The future of cloud databases lies in integrating AI and machine learning (ML). Predictive scaling, automated query optimization, and intelligent resource allocation are areas where cloud databases can deliver unprecedented efficiency. By addressing current challenges and harnessing emerging technologies, cloud databases will continue to play a pivotal role in shaping the future of data management [68], [69]. Figure 9 summarizes the architectural principles, intelligent features, and operational challenges of cloud-native databases.

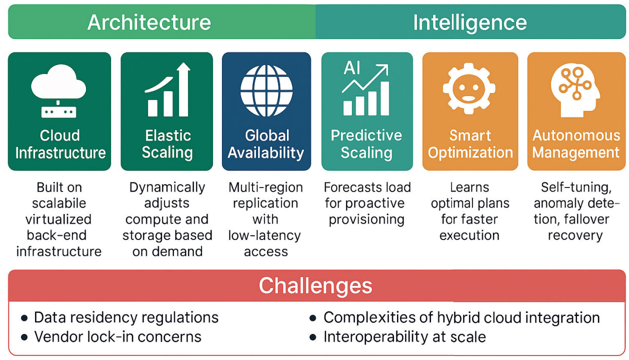


FIGURE 9. **Cloud-native databases: architecture, intelligence, and operational challenges.**

Table 2 provides a structured timeline of key advancements in database evolution, detailing how traditional RDBMS transitioned into NoSQL, NewSQL, and cloud-native paradigms. The table summarizes previous research contributions and highlights the technical innovations that each stage introduced, such as ACID compliance in relational databases, horizontal scalability in NoSQL, hybrid consistency models in NewSQL, and elasticity in cloud databases.

IV. PERFORMANCE CHARACTERISTICS

The performance characteristics of databases in the Big Data era are critical in determining their suitability for various applications. These characteristics not only define the system's efficiency and scalability but also influence its reliability and cost-effectiveness. This section provides a detailed exploration of key performance metrics, such as throughput, latency, fault tolerance, query optimization, and cost efficiency, to assess the operational capabilities of modern databases.

A. THROUGHPUT AND LATENCY

Throughput and latency are cornerstone metrics for evaluating database performance in Big Data environments. Throughput measures the volume of transactions a system can process over a given period, reflecting its capacity to handle large-scale workloads. Latency, on the other hand, indicates the time taken to complete a single operation or query, emphasizing the responsiveness of the database. High-throughput systems are essential for data-intensive applications such as online retail platforms and financial services, where millions of operations occur. Similarly, low latency is indispensable for real-time analytics and interactive applications, such as fraud detection systems and social media platforms [70], [71], [72].

Modern databases leverage various strategies to optimize throughput and latency, including parallel processing, in-memory computing, and advanced indexing techniques. For example, in-memory databases like SAP HANA reduce latency by storing data directly in the main memory, bypass-

TABLE 2. Summary of key references outlining the evolution of databases, from traditional RDBMS to modern paradigms.

Topic	Reference	Description	Technical Contributions
Foundations of Relational Databases	[38] [39]	Describes relational databases, emphasizing consistency, reliability, and SQL as the standard for querying structured data.	Introduced structured schema-based data management, ACID compliance, and SQL as a query standard.
Challenges in Relational Systems	[40] [41]	Highlights the limitations of relational systems in handling Big Data and discusses advancements like indexing and distributed relational systems.	Identified scalability limitations, introduced indexing techniques, and distributed RDBMS approaches to improve query efficiency.NoSQL introduced schema-less design, horizontal scaling, and eventual consistency. NewSQL retained ACID compliance while achieving scalability.
Transition to NoSQL and NewSQL	[42] [43]	Explores the emergence of NoSQL and NewSQL, addressing scalability, schema flexibility, and the needs of modern Big Data applications.	NoSQL introduced schema-less design, horizontal scaling, and eventual consistency. NewSQL retained ACID compliance while achieving scalability.
Introduction to NoSQL Databases	[44] [45]	Highlights the need for NoSQL databases to address scalability and flexibility challenges of traditional relational systems.	Developed document-oriented, key-value, column-family, and graph-based models for specialized use cases.
Key-Value and Document Databases	[46] [47] [48]	Discusses key-value stores for high-speed lookups and document-oriented databases for semi-structured data storage.	Enabled high-speed data retrieval, flexible schema design, and scalable data storage.
Column-Family and Graph Databases	[49] [50]	Explores column-family databases for analytical workloads and graph databases for relationship-centric queries.	Column-family databases optimized big data analytics and distributed storage; Graph databases enabled efficient relationship mapping and traversal.
Scalability and Fault Tolerance	[51]	Focuses on sharding and replication techniques to ensure performance and reliability in distributed NoSQL systems.	Implemented sharding, data partitioning, and replication to enhance fault tolerance and availability.
Hybrid NoSQL Approaches	[52]	Examines hybrid NoSQL models that integrate relational and NoSQL paradigms to combine flexibility and consistency.	Developed multi-model databases that support multiple data paradigms within a single system.
Introduction to NewSQL	[53] [54] [55]	Highlights NewSQL's objective to combine ACID compliance of relational systems with the horizontal scalability of NoSQL.	Implemented distributed transaction support, in-memory processing, and hybrid partitioning strategies.
Performance Innovations in NewSQL	[56] [57]	Discusses advancements such as in-memory processing, distributed architectures, and hybrid partitioning strategies.	Improved OLTP and OLAP workloads using real-time processing and high-performance query execution.
Challenges in NewSQL Adoption	[58] [59] [60]	Covers the complexities of distributed transactions, low-latency demands, and the need for specialized deployment expertise.	Addressed transactional scalability while maintaining consistency; introduced distributed consensus mechanisms.
Introduction to Cloud Databases	[61] [62] [63]	Highlights the scalability, elasticity, and global accessibility offered by cloud-native databases.	Enabled serverless architectures, multi-region replication, and dynamic resource allocation.
Cloud-Native Features	[64] [65]	Discusses features like multi-region replication, serverless architecture, and automated maintenance for enhanced fault tolerance.	Introduced distributed storage and compute separation, enabling elastic scalability and cost optimization.
Challenges in Cloud Integration	[66] [67]	Explores issues such as vendor lock-in, data residency regulations, and hybrid cloud complexities.	Developed cross-cloud database solutions, improving interoperability and regulatory compliance.
Future Directions in AI and Databases	[68] [69]	Emphasizes the integration of AI for predictive scaling, automated query optimization, and intelligent resource allocation.	Leveraged AI-driven query optimization, self-tuning databases, and adaptive workload management.

ing traditional disk-based storage. Additionally, techniques like query batching and asynchronous processing enhance throughput by efficiently managing transaction pipelines. However, simultaneously achieving high throughput and low latency often requires trade-offs, particularly in distributed systems where network delays and consistency requirements can impede performance [73], [74], [75].

The nature of workloads further complicates the interplay between throughput and latency. Analytical workloads, which involve complex queries on large datasets, prioritize throughput over latency, whereas transactional workloads demand immediate responsiveness. Hybrid systems attempt to balance these demands, employing adaptive resource allocation and workload-specific optimizations to maintain performance under varying conditions [76], [77].

B. FAULT TOLERANCE AND AVAILABILITY

Fault tolerance and availability are critical for ensuring the reliability of databases, particularly in distributed and cloud-based environments. Fault tolerance refers to a system's ability to continue functioning despite hardware failures, network disruptions, or software errors. High availability ensures that the system remains accessible to users, minimizing downtime and operational disruptions. Together, these characteristics are vital for mission-critical applications, such as healthcare systems, financial services, and e-commerce platforms, where data loss or unavailability can have severe consequences [78], [79], [80].

Modern databases achieve fault tolerance through replication, consensus algorithms, and failover mechanisms. Replication involves maintaining multiple copies of data across nodes, ensuring that a failure in one node does not compromise data availability. Consensus algorithms like Paxos and Raft coordinate updates across nodes to maintain consistency and prevent conflicts. Automated failover mechanisms detect failures and redirect traffic to operational nodes, maintaining seamless service continuity [81], [82], [83].

Cloud-native databases further enhance fault tolerance by leveraging cloud infrastructures' inherent redundancy and elasticity. Multi-region replication and distributed backups ensure that data remains accessible even during regional outages. However, these strategies come with trade-offs in terms of increased storage and network costs. Balancing these costs with the desired level of fault tolerance and availability remains an ongoing challenge in database design [84], [85].

C. QUERY OPTIMIZATION

Query optimization is a fundamental aspect of database performance, focusing on improving the efficiency of data retrieval and processing. It involves selecting the most efficient execution plan for a given query, minimizing resource consumption and execution time. Effective query optimization is critical for both transactional and analytical workloads, where performance bottlenecks can significantly impact user experience and decision-making processes [86], [87].

Modern query optimizers employ a combination of rule-based and cost-based strategies. Rule-based optimizers rely on predefined heuristics to choose query execution plans, while cost-based optimizers evaluate the estimated computational cost of various plans to select the most efficient one. Advances in indexing, such as B-trees and hash-based structures, further accelerate query execution by enabling faster access to relevant data [88], [89], [90].

In Big Data scenarios, the complexity of queries and the scale of datasets present unique challenges for optimization. Techniques such as predicate pushdown, query pruning, and materialized views address these challenges by reducing the amount of data processed during query execution. Additionally, distributed query engines, like Apache Spark and Presto, leverage parallel processing to handle large-scale analytical workloads efficiently [91], [92].

The integration of ML into query optimization represents a significant innovation in this area. AI-driven optimizers analyze historical query performance and system metrics to predict the best execution plans dynamically. This adaptive approach improves efficiency and enables databases to respond to changing workloads and resource conditions [93].

D. COST EFFICIENCY

Cost efficiency is crucial for database systems, particularly in cloud-based environments where resource consumption directly impacts operational expenses. It involves balancing a database's computational and storage requirements with financial constraints, ensuring that performance goals are met without excessive costs [94], [95].

Modern databases employ several strategies to optimize cost efficiency. Compression techniques reduce storage costs by minimizing the physical space required for data. Tiered storage systems, which utilize a combination of high-performance and low-cost storage mediums, balance speed and cost effectively. Workload-aware resource allocation dynamically adjusts compute resources based on demand, preventing over-provisioning and reducing idle resource costs [96], [97], [98].

The pay-as-you-go pricing model of cloud databases further enhances cost efficiency by allowing organizations to scale resources up or down based on usage. However, this flexibility comes with challenges, such as managing unpredictable costs during peak usage periods. Tools for cost monitoring and forecasting, integrated into cloud platforms, help mitigate these challenges by providing visibility into resource utilization and expenses [99], [100].

Future advancements in cost efficiency are likely to focus on energy-efficient database operations. As sustainability becomes a priority, databases are exploring low-power hardware, energy-aware query execution, and green data center practices. These innovations aim to reduce the environmental footprint of database systems while maintaining cost competitiveness [101].

Table 3 outlines an overview of key performance metrics for modern databases, focusing on strategies to optimize throughput, latency, fault tolerance, and cost efficiency. It categorizes various optimization techniques and examines their impact on application performance. It highlights their relevance in real-world scenarios such as high-transaction systems, real-time analytics, and cloud-based environments.

V. CHALLENGES IN BIG DATA DATABASES

The development and adoption of Big Data databases are accompanied by several persistent challenges that impact their effectiveness and efficiency. These challenges require innovative solutions to maintain the functionality and scalability necessary for modern applications. This section elaborates on four critical challenges: scalability, data heterogeneity, security and privacy, and interoperability.

A. SCALABILITY

Scalability remains one of the most pressing challenges in Big Data databases. The exponential growth in data volumes and the increasing complexity of applications demand systems that can scale horizontally and vertically without compromising performance. Horizontal scaling,

TABLE 3. Overview of key performance metrics for modern databases, highlighting optimization strategies, their impact on application performance, and associated trade-offs.

Topic	Reference	Description	Impact on Application Performance	Trade-offs & Considerations
Throughput Challenges	[70] [71] [72]	Focuses on optimizing high transaction volumes for real-time analytics and interactive applications.	Improves performance in large-scale transactional workloads such as online banking, e-commerce, and financial trading.	Requires significant computational resources; achieving high throughput can lead to increased hardware and energy costs.
Latency Optimization	[73] [74] [75]	Discusses strategies like in-memory computing, query batching, and asynchronous processing to reduce delays.	Essential for real-time applications such as fraud detection, recommendation systems, and IoT analytics.	Low-latency operations may sacrifice consistency in distributed environments; in-memory storage increases RAM dependency.
Workload-Specific Needs	[76] [77]	Explores balancing throughput and latency for hybrid systems handling both analytical and transactional workloads.	Enables hybrid systems to process large data volumes while maintaining fast responses for end-users.	Balancing OLAP and OLTP workloads requires sophisticated resource allocation and tuning strategies.
Fault Tolerance	[78] [79] [80]	Highlights the use of replication, consensus algorithms, and failover mechanisms to ensure system resilience.	Crucial for mission-critical healthcare, finance, and industrial automation systems.	Replication increases storage overhead and network traffic; consensus algorithms introduce latency.
High Availability	[81] [82] [83]	Focuses on maintaining accessibility through multi-region replication and automated failover strategies.	Reduces downtime for cloud-based services, improving reliability in globally distributed applications.	Multi-region replication can increase operational costs and introduce consistency challenges.
Trade-offs and Costs	[84] [85]	Discusses balancing fault tolerance and availability with the costs of storage and network resources.	Helps organizations optimize costs while maintaining a reliable system.	Over-provisioning resources for reliability can lead to unnecessary expenses; under-provisioning can cause service disruptions.
Query Execution Efficiency	[86] [87]	Covers rule-based and cost-based strategies for selecting optimal query plans and minimizing execution time.	Enhances database response time, improving user experience in analytics and reporting systems.	Optimizing complex queries requires extensive indexing and storage management, which can impact write performance.
Advanced Indexing Techniques	[88] [89] [90]	Explores B-trees, hash-based indexing, and materialized views to enhance data retrieval performance.	Improves search speed and efficiency in applications with frequent read operations.	Index maintenance can slow down write-heavy workloads; improper indexing can lead to excessive storage consumption.
Big Data Query Challenges	[91] [92] [93]	Discusses distributed query engines and ML-driven optimizations for handling large-scale analytics.	Enables faster processing of large datasets for data-driven decision-making and AI applications.	Query optimization in distributed systems can be complex; it requires tuning and workload balancing.
Cost Optimization Strategies	[94] [95]	Explores compression techniques and tiered storage systems to reduce storage and computational costs.	Reduces operational costs, making databases more cost-efficient for cloud-based and on-premises deployments.	Compression techniques can introduce overhead in decompression; tiered storage may lead to latency variations.
Dynamic Resource Allocation	[96] [97] [98]	Discusses workload-aware scaling and replica management to prevent over-provisioning and minimize idle resources.	Improves resource utilization in cloud and edge environments, optimizing system performance dynamically.	Scaling decisions must be carefully managed to avoid sudden resource shortages; predictive scaling may introduce delays.
Energy Efficiency	[99] [100] [101]	Focuses on energy-efficient operations, including low-power hardware and green data center practices.	Reduces the carbon footprint of database operations while lowering energy costs.	Energy-efficient solutions may require specialized hardware, which can increase initial investment costs.

which involves distributing data and workloads across multiple nodes, is often hindered by issues such as data partitioning, load balancing, and maintaining consistency across nodes. For example, achieving uniform workload distribution in distributed environments while minimizing cross-node communication is a complex task that directly impacts performance [102], [103].

Vertical scaling, while simpler in concept, is constrained by hardware limitations and cost considerations. Advanced techniques, such as in-memory processing and the use of hardware accelerators like graphics processing units (GPUs) and field programmable gate arrays (FPGAs), have been explored to enhance vertical scalability. However, these approaches are not universally applicable and often require significant investments in infrastructure and expertise [104], [105].

Emerging research focuses on adaptive scaling mechanisms that dynamically allocate resources based on workload patterns and predictive analytics. These mechanisms leverage ML models to forecast resource demands and optimize the allocation of computational and storage resources. By integrating such intelligent solutions, databases can achieve greater scalability while maintaining cost efficiency and performance [106].

B. DATA HETEROGENEITY

Data heterogeneity is a defining characteristic of Big Data, encompassing structured, semi-structured, and unstructured formats. This diversity poses significant challenges for database systems, particularly regarding schema design, data integration, and query processing. With their rigid schema requirements, traditional relational databases struggle to accommodate the flexibility needed to handle diverse data types and sources [107], [108], [109].

Modern databases employ techniques such as schema-on-read and multi-model support to address data heterogeneity. Schema-on-read allows data to be ingested without predefined schemas, enabling greater flexibility in handling semi-structured and unstructured data. Multi-model databases, which support multiple data representations within a single system, offer a unified platform for managing diverse datasets. Examples include ArangoDB and OrientDB, which integrate graph, document, and key-value models [110], [111].

Despite these advancements, challenges remain in ensuring efficient query processing and data transformation across heterogeneous formats. Interoperability between different data models and the development of standardized query languages are active research areas. By addressing these challenges, database systems can better support the diverse requirements of modern applications [112], [113].

C. SECURITY AND PRIVACY

Big Data databases' distributed and often cloud-based nature introduces unique security and privacy challenges. Confidentiality, integrity, and data availability are paramount, particularly in healthcare, finance, and government industries where sensitive information is involved. Threats such as unauthorized access, data breaches, and insider attacks necessitate robust security mechanisms [114], [115], [116].

Encryption is a fundamental technique for protecting data at rest and in transit. Modern databases integrate advanced encryption protocols, such as homomorphic encryption, which allows computations on encrypted data without decryption. Access control mechanisms, including role-based and attribute-based access control, further enhance security by restricting data access based on user roles and attributes [117].

Privacy-preserving techniques, such as differential privacy and secure multi-party computation, are gaining traction in addressing privacy concerns. These methods ensure that individual data points remain anonymous while enabling aggregate analyses. However, implementing these techniques in real-time database systems poses computational challenges, requiring further research and optimization [118], [119].

D. INTEROPERABILITY

Interoperability, the ability of systems to work seamlessly across different platforms and environments, is a critical challenge in the heterogeneous ecosystem of Big Data databases. Organizations often operate multiple database systems, each optimized for specific use cases, leading to data silos and integration challenges. Ensuring seamless communication and data exchange between these systems enables comprehensive analytics and decision-making [120], [121], [122].

Middleware solutions and standardized APIs are common approaches to achieving interoperability. Middleware is an intermediary layer that facilitates communication between disparate systems, while APIs provide standardized data access and manipulation interfaces. However, these approaches often introduce latency and complexity, particularly in real-time applications [123], [124].

Advancements in data integration frameworks and federated query systems aim to address these limitations. Federated systems enable queries across multiple databases without requiring data replication, providing a unified view of distributed datasets. These systems, combined with advancements in ML and natural language processing, promise to improve interoperability and enhance the usability of Big Data databases [125].

Table 4 highlights key challenges in Big Data databases, focusing on scalability, heterogeneity, security, and interoperability. It categorizes these challenges and examines their real-world impact, highlighting how they affect large-scale applications such as cloud computing, financial transactions, and multi-source data integration.

VI. FUTURE RESEARCH DIRECTIONS

The future of databases in the Big Data era hinges on addressing the limitations of existing systems while leveraging emerging technologies to unlock new capabilities. One of the most pressing areas for research is scalability enhancement. While current systems excel at horizontal scaling, they often face challenges in balancing performance and resource efficiency. Innovations in adaptive scaling algorithms, such as ML-based resource allocation, could enable databases to respond dynamically to workload fluctuations, minimizing costs without sacrificing performance. Additionally, exploring architectures that blend cloud-native features with on-premises infrastructure could address specific use cases requiring both elasticity and control [126], [127], [128].

Another critical direction is the development of multi-model database systems. These systems aim to integrate multiple data models within a single database, such as relational, graph, and document-based models. Multi-model systems promise to simplify data management for applications requiring diverse data representations, such as knowledge graphs or hybrid transactional and analytical processing (HTAP). Research into efficient query optimization and indexing strategies for such databases is crucial to unlocking their full potential [129], [130].

The incorporation of AI and ML into DBMS represents a transformative research avenue. AI-driven optimizations, including predictive query planning, automated indexing, and anomaly detection, can significantly enhance database performance and reliability. Furthermore, leveraging ML for real-time analytics and decision-making within the database layer can enable intelligent systems to adapt to dynamic workloads and user requirements. The integration of explainable AI mechanisms would ensure transparency and trust in these systems [131], [132], [133].

Addressing data privacy and security is an ongoing challenge in the context of distributed and cloud databases. Future research must focus on designing advanced encryption techniques, such as homomorphic encryption and secure multi-party computation, to enable secure data processing without compromising performance. Additionally, developing robust frameworks for regulatory compliance, data sovereignty, and secure cross-border data sharing is imperative as organizations increasingly operate in global markets [134], [135].

Lastly, sustainability in database systems is gaining importance as data centers contribute significantly to global energy consumption. Research into energy-efficient architectures, including hardware accelerators optimized for database operations and intelligent workload distribution mechanisms, can reduce the environmental footprint of

TABLE 4. Key challenges in Big Data databases, their real-world impact, and solutions addressing scalability, heterogeneity, security, and interoperability issues.

Topic	Reference	Description	Real-World Impact	Solutions
Horizontal Scaling Challenges	[102] [103]	Highlights difficulties in data partitioning, load balancing, and maintaining consistency in distributed environments.	Impacts real-time processing in large-scale applications like social media platforms and cloud storage.	Dynamic sharding, load-balancing algorithms, and auto-scaling frameworks.
Vertical Scaling Limitations	[104] [105]	Explores the constraints of vertical scaling due to hardware limitations and costs and the use of advanced techniques like in-memory processing and GPUs.	Limits the scalability of legacy enterprise systems, affecting cost-efficiency and performance.	GPU-based query acceleration, in-memory processing, and hybrid storage architectures.
Adaptive Scaling Mechanisms	[106]	Focuses on ML-based predictive resource allocation to dynamically adjust resources for greater scalability and cost-efficiency.	Improves efficiency in cloud-based and IoT environments where demand fluctuates.	AI-driven resource optimization and predictive autoscaling.
Challenges of Data Heterogeneity	[107] [108] [109]	Describes difficulties in managing diverse data types, including structured, semi-structured, and unstructured data, with traditional rigid schema systems.	Affects industries relying on multi-source data integration, such as healthcare and finance.	Schema-on-read, data lakes, and approximate query processing.
Schema-on-Read Approach	[110]	Discusses schema-on-read, enabling flexible ingestion of semi-structured and unstructured data without predefined schemas.	Benefits streaming and log-based analytics where predefined schemas are impractical.	NoSQL storage models and flexible schema data processing.
Multi-Model Databases	[111]	Explores multi-model databases like ArangoDB and OrientDB, which support multiple data representations in a single platform.	Enables diverse data storage formats in unified architectures, simplifying application development.	Hybrid databases supporting document, graph, and relational models.
Interoperability Issues	[112] [113]	Highlights the need for standardized query languages and efficient data transformation to address integration across heterogeneous systems.	Creates integration bottlenecks in multi-cloud and cross-enterprise applications.	Standardized query translation layers and cross-platform data exchange frameworks.
Security Challenges	[114] [115] [116]	Discusses the risks of unauthorized access, data breaches, and the need for encryption and access control mechanisms to ensure data security.	Critical for financial transactions, healthcare data, and government systems.	End-to-end encryption, zero-trust architectures, and access control policies.
Advanced Encryption Techniques	[117]	Covers methods like homomorphic encryption and differential privacy for secure and privacy-preserving data processing.	Enables secure analytics for sensitive data without compromising confidentiality.	Fully homomorphic encryption and privacy-preserving computation models.
Privacy-Preserving Techniques	[118] [119]	Explores differential privacy and secure multi-party computation to maintain anonymity while enabling aggregate analysis.	Protects user privacy in AI-driven analytics and large-scale data sharing.	Differential privacy models and federated learning techniques.
Challenges of Interoperability	[120] [121] [122]	Identifies issues in integrating multiple database systems, leading to data silos and hindering comprehensive analytics.	Affects cross-border data access and collaborative research platforms.	Unified metadata management and API-based integration.
Middleware and APIs	[123] [124]	Discusses middleware and standardized APIs as solutions for enabling communication between heterogeneous systems.	Facilitates seamless data exchange in multi-cloud and hybrid architectures.	RESTful APIs, middleware layers, and data virtualization platforms.
Federated Query Systems	[125]	Highlights federated query systems that allow queries across multiple databases without data replication for unified analytics.	Improves efficiency in large-scale analytics where data sources are distributed.	Distributed query execution and virtualized data access models.

database systems. Additionally, exploring edge computing paradigms for localized data processing could alleviate the strain on centralized data centers while improving latency and efficiency. By addressing these areas, future research can ensure that database systems continue to evolve to meet the demands of a data-driven world [136], [137], [138].

Table 5 presents key future research directions in database systems, focusing on advancements in scalability, multi-model integration, AI-driven optimizations, data privacy, and sustainability. It outlines how emerging technologies, such as ML-based adaptive scaling, AI-powered query optimization, and energy-efficient architectures, are shaping the next generation of database solutions.

VII. DISCUSSION

This survey has examined the evolution and classification of database systems in the Big Data era, offering a broad and integrated perspective that contrasts significantly with earlier surveys outlined in Table 6. Compared to the more specialized focus of existing works, our findings emphasize the convergence of scalability, performance, and deployment strategies across emerging database paradigms.

For instance, [139] explores data partitioning and sampling techniques in Hadoop-based systems, highlighting early efforts to manage data distribution in distributed computing environments. While their work provides a foundational understanding of partitioning strategies, it is limited in scope to specific infrastructures. Our survey extends this perspective by incorporating sharding, replication, and dynamic scaling mechanisms as broader principles of modern database design, applicable to NoSQL, NewSQL, and cloud-native databases alike.

Similarly, [140] focuses on Big Data analytics within the domain of smart grids, discussing challenges such as data variability and the integration of ML for resilience. Although valuable in its domain-specific insights, it does not engage with the broader architectural and performance-oriented concerns that are critical in evaluating database systems across multiple application domains. In contrast, our work presents a performance taxonomy, covering throughput, latency, fault tolerance, and cost efficiency, applicable to a diverse range of real-world scenarios beyond vertical applications.

A comprehensive overview of Big Data systems, components, tools, and analytics technologies offers an essential

TABLE 5. Future research directions in database systems, highlighting their potential impact on scalability, multi-model integration, AI-driven advancements, data privacy, and sustainability.

Topic	Reference	Description	Potential Impact
Scalability Enhancement	[126] [127] [128]	Adaptive scaling algorithms using ML for cost-efficient workload management. Combines cloud-native and on-premises features.	Enables highly efficient resource allocation, reducing operational costs while maintaining performance in dynamic workloads.
Multi-Model Database Systems	[129] [130]	Integrates multiple data models to simplify management and improve query optimization.	Enhances flexibility in handling diverse data types, reducing complexity in managing multiple database architectures.
AI and ML in Databases	[131] [132] [133]	Applies AI/ML for predictive planning, automated indexing, anomaly detection, and real-time analytics with explainable AI for transparency.	Improves query performance, automates data management, and enhances system adaptability while ensuring model interpretability.
Data Privacy and Security	[134] [135]	Advanced encryption and frameworks for secure, compliant, and cross-border data sharing.	Strengthens data protection in multi-cloud environments, ensuring compliance with global regulations while enabling secure data exchanges.
Sustainability in Databases	[136] [137] [138]	Focuses on energy-efficient architectures and edge computing for reduced environmental impact and better efficiency.	Reduces energy consumption in large-scale database operations, promoting green computing initiatives and lowering infrastructure costs.

backdrop to the technological landscape, as outlined in [141]. However, its emphasis is predominantly on the ecosystem level, including visualization and analytics frameworks like Hadoop and Spark, rather than the underlying database architectures themselves. Our contribution complements this systems-level overview by drilling deeper into the classification and comparative evaluation of database models, consistency mechanisms, and deployment strategies, which are crucial for selecting appropriate data management solutions in Big Data applications.

Also, [142] contributes insights into Big Data service architectures with an emphasis on cloud-based frameworks, yet their focus remains on data collection and visualization pipelines. By comparison, our survey places a stronger focus on the data storage and processing layers, systematically analyzing how cloud-native databases like Amazon DynamoDB or Google Bigtable implement elasticity, replication, and serverless capabilities.

Besides, [143] explores blockchain as a solution for secure data management in Big Data contexts, particularly for applications in healthcare and smart cities. While their work is critical in addressing privacy and integrity, it represents only one facet of the broader security landscape. Our survey incorporates a more extensive examination of encryption, privacy-preserving computation, and access control mechanisms, situating blockchain among various security techniques.

Lastly, [144] focuses on storage and placement methodologies in the cloud, particularly the non-functional aspects such as cost and data lifecycle management. Building upon their analysis, we evaluate how these methodologies manifest in modern database implementations, highlighting how elasticity, workload-aware scaling, and multi-region replication shape performance and cost efficiency.

Taken together, these comparisons underscore the broader implications of our findings. By offering a structured classification and a cross-cutting performance evaluation of traditional and modern database systems, our survey consolidates prior research and provides a foundation for future developments. The integration of architectural trends, consistency trade-offs, and AI-driven innovations enables a forward-looking perspective that informs both academic research and industry adoption in the evolving landscape of Big Data databases.

TABLE 6. Summary of surveys and descriptions in the Big Data era.

Survey	Description
[139]	Survey on data partitioning and sampling methods in big data analysis, discussing classical and novel strategies on Hadoop clusters for efficient distributed computing.
[140]	Examination of big data analytics technologies for smart grids, covering challenges like data variability, privacy, and the role of ML in improving grid resilience and efficiency.
[141]	Overview of big data system components, tools, and technologies, including visualization, distributed databases, and advancements in data analytics frameworks like Hadoop and Spark.
[142]	Discussion on big data service architectures emphasizing data collection, analysis, visualization, and cloud-based frameworks for scalable processing.
[143]	Survey of blockchain applications in big data, highlighting its integration for secure data storage, analytics, and privacy preservation in domains like smart cities and healthcare.
[144]	Analysis of cloud-based big data storage and placement methodologies, with a focus on non-functional aspects such as cost, scalability, and data lifecycle management.

VIII. CONCLUSION

This survey has provided an extensive overview of databases in the Big Data era, focusing on their evolution, classification, performance attributes, and the persistent challenges they aim to address. Big data's transformative impact has necessitated a departure from traditional relational database systems, giving rise to paradigms such as NoSQL, NewSQL, and cloud-native databases. Each paradigm offers unique strengths tailored to address the demands of scalability, flexibility, and performance optimization in increasingly complex data environments.

The historical analysis highlighted how traditional databases excelled in structured data management and transactional integrity but struggled to adapt to modern workloads' sheer scale and heterogeneity. The emergence of NoSQL databases brought unparalleled flexibility and scalability, allowing the efficient handling of unstructured and semi-structured data. NewSQL systems successfully integrated these advantages with the ACID guarantees of relational models, making them ideal for hybrid workloads. Cloud-native databases further revolutionized the field by leveraging distributed architectures to deliver elastic scalability, high availability, and global accessibility.

The classification framework presented in this survey provided a systematic understanding of database architectures, categorizing them by data models, deployment strategies, scalability mechanisms, and consistency models. This taxonomy enables researchers and practitioners to evaluate databases against specific application requirements, fostering informed decision-making. Additionally, exploring performance characteristics, including throughput, latency, fault tolerance, and cost efficiency, underscored the importance of aligning database capabilities with real-world use cases.

This survey offers a holistic understanding of the field by synthesizing diverse database systems' evolution, current landscape, and comparative strengths. It serves as a valuable resource for comprehending the dynamic shifts in database technologies and their implications for Big Data management, laying the foundation for better application-driven database design and deployment.

REFERENCES

[1] K. C. Balodi, R. Jain, T. B. K. Kumar, and D. Banerjee, "Platform revolution in the database management system industry: Evolution of SAP's business model," *J. Inf. Technol. Teaching Cases*, vol. 13, no. 1, pp. 126–133, May 2023.

[2] J. J. K. Behan, M. Ali, A. Inam, and M. T. Khan, "Comparative analysis of rdbms and nosql databases," in *Proc. DBKDA*, Jun. 2022, p. 38.

[3] N. Chaudhry and M. M. Yousaf, "Architectural assessment of NoSQL and NewSQL systems," *Distrib. Parallel Databases*, vol. 38, no. 4, pp. 881–926, Dec. 2020.

[4] S. Dähling, L. Razik, and A. Monti, "Enabling scalable and fault-tolerant multi-agent systems by utilizing cloud-native computing," *Auto. Agents Multi-Agent Syst.*, vol. 35, no. 1, p. 10, Apr. 2021.

[5] I. Chereja, S. M. L. Hahn, and O. Matei, "Benchmarking NewSQL cloud-native or cloud-hosted databases," in *Proc. 5th Comput. Methods Syst. Softw. Softw. Eng. Appl. Inform.*, vol. 1. Cham, Switzerland: Springer, Jan. 2021, pp. 296–310.

[6] C. Coronel and S. Morris, *Database Systems: Design, Implementation and Management*. Boston, MA, USA: Cengage Learning, 2019.

[7] A. Vaishnav, K. D. Pham, J. Powell, and D. Koch, "FOS: A modular FPGA operating system for dynamic workloads," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 13, no. 4, pp. 1–28, Dec. 2020.

[8] D. Basin, E. Bortnikov, A. Braginsky, G. Golan-Gueta, E. Hillel, I. Keidar, and M. Sulamy, "Kiwi: A key-value map for scalable real-time analytics," *ACM Trans. Parallel Comput.*, vol. 7, no. 3, pp. 1–28, 2020.

[9] S. Hamouda, Z. Zainol, and M. Anbar, "A flexible schema for document oriented database (SDOD)," in *Proc. 11th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, 2019, pp. 413–419.

[10] Y. Du, "Massive semi-structured data platform based on elasticsearch and MongoDB," in *Proc. 7th Int. Conf. Signal Inf. Process., Netw. Comput. Signal Inf. Process., Netw. Comput. (ICSINC)*. Cham, Switzerland: Springer, Dec. 2020, pp. 877–884.

[11] V. Sheth, "Comparative performance analysis of column family databases: Cassandra and hbase," Ph.D. dissertation, Dept. Inf. Commun. Technol., Dhirubhai Ambani Inst. Inf. Commun. Technol., Gandhinagar, Gujarat, 2023.

[12] P. Kotiranta, M. Junkkari, and J. Nummenmaa, "Performance of graph and relational databases in complex queries," *Appl. Sci.*, vol. 12, no. 13, p. 6490, Jun. 2022.

[13] F. Chinesta, E. Cueto, E. Abisset-Chavanne, J. L. Duval, and F. E. Khaldi, "Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data," *Arch. Comput. Methods Eng.*, vol. 27, no. 1, pp. 105–134, Jan. 2020.

[14] M. Massri, P. Raipin, and P. Meye, "GDBAlive: A temporal graph database built on top of a columnar data store," *J. Adv. Inf. Technol.*, vol. 12, no. 3, pp. 169–178, 2021.

[15] A. Zulkifli, "Accelerating database efficiency in complex it infrastructures: Advanced techniques for optimizing performance, scalability, and data management in distributed systems," *Int. J. Inf. Cybersecurity*, vol. 7, no. 12, pp. 81–100, 2023.

[16] F. Olariu and L. Alboaie, "Challenges in optimizing migration costs from on-premises to Microsoft azure," *Proc. Comput. Sci.*, vol. 225, pp. 3649–3659, Jul. 2023.

[17] F. Li, "Cloud-native database systems at alibaba: Opportunities and challenges," *Proc. VLDB Endowment*, vol. 12, no. 12, pp. 2263–2272, Aug. 2019.

[18] Z. Liu, H. Wu, T. Bai, Y. Wang, and C. Xu, "Towards elastic data warehousing by decoupling data management and computation," in *Proc. 4th Int. Conf. Cloud Big Data Comput.*, Aug. 2020, pp. 52–57.

[19] O. D. Segun-Falade, O. S. Osundare, W. E. Kedi, P. A. Okeleke, T. I. Ijomah, and O. Y. Abdul-Azeez, "Assessing the transformative impact of cloud computing on software deployment and management," *Comput. Sci. IT Res. J.*, vol. 5, no. 8, pp. 2062–2082, Aug. 2024.

[20] S. Shrivastava, G. Saini, and Y. Agrawal, "Multi-cloud deployments and hybrid cloud architecture," *Res Militaris*, vol. 10, no. 1, pp. 128–134, 2020.

[21] J. George, "Optimizing hybrid and multi-cloud architectures for real-time data streaming and analytics: Strategies for scalability and integration," *World J. Adv. Eng. Technol. Sci.*, vol. 7, no. 1, pp. 174–185, Oct. 2022.

[22] C. Roy, M. Pandey, and S. S. Rautaray, "A proposal for optimization of horizontal scaling in big data environment," in *Advances in Data and Information Sciences: Proceedings of ICDIS-2017*, vol. 1. Cham, Switzerland: Springer, 2018, pp. 223–230.

[23] S. Solat and F. Nait-Abdesselam, "Sharding distributed replication systems to improve scalability and throughput," in *Building Cybersecurity Applications With Blockchain and Smart Contracts*. Cham, Switzerland: Springer, 2023, pp. 101–124.

[24] X.-F. Liu, J. Zhang, and J. Wang, "Cooperative particle swarm optimization with a bilevel resource allocation mechanism for large-scale dynamic optimization," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 1000–1011, Feb. 2023.

[25] K. Razavi, M. Salmani, M. Mühlhäuser, B. Koldehofe, and L. Wang, "A tale of two scales: Reconciling horizontal and vertical scaling for inference serving systems," 2024, *arXiv:2407.14843*.

[26] V. Ramamoorthi, "AI-driven cloud resource optimization framework for real-time allocation," *J. Adv. Comput. Syst.*, vol. 1, no. 1, pp. 8–15, 2021.

[27] X. Guo, H. Gao, and Z. Zou, "WISE: Workload-aware partitioning for RDF systems," *Big Data Res.*, vol. 22, Dec. 2020, Art. no. 100161.

[28] J. L. Alvarez, "Performance analysis and optimization strategies for scalable cloud networking in high-demand environments," *Innov. Eng. Sci. J.*, vol. 4, no. 1, pp. 1–7, 2024.

[29] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," 2023, *arXiv:2303.01469*.

[30] S. Xiong, A. Cerone, A. Raad, and P. Gardner, "Data consistency in transactional storage systems: A centralised approach," 2019, *arXiv:1901.10615*.

[31] A. A. Akinyemi, "Modelling and evaluation of performance, security and database management trade-offs in cloud computing platforms. An investigation into quantitative modelling and simulation analysis of 'optimal'performance, security and database management trade-offs in cloud computing platforms (CCPS), based on stochastic activity networks (SANS) and a three-tier combined metrics," Ph.D. dissertation, Dept. Comput. Sci., Univ. Bradford, Bradford, U.K., 2020.

[32] M. Diogo, B. Cabral, and J. Bernardino, "Consistency models of NoSQL databases," *Future Internet*, vol. 11, no. 2, p. 43, Feb. 2019.

[33] K. Krishan, G. Gupta, and G. S. Bhathal, "Striking the balance: Comprehensive insights into data consistency in NoSQL realms," in *Proc. 11th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Feb. 2024, pp. 715–720.

[34] S. Kanungo and R. D. Morena, "Concurrency versus consistency in NoSQL databases," *J. Auto. Intell.*, vol. 7, no. 3, pp. 1–10, Dec. 2023.

[35] C. Gomes, E. Tavares, M. N. D. O. Junior, and B. Nogueira, "Cloud storage availability and performance assessment: A study based on NoSQL DBMS," *J. Supercomput.*, vol. 78, no. 2, pp. 2819–2839, Feb. 2022.
[36] H. N. S. Aldin, H. Deldari, M. H. Moattar, and M. R. Ghods, "Strict timed causal consistency as a hybrid consistency model in the cloud environment," *Future Gener. Comput. Syst.*, vol. 105, pp. 259–274, Apr. 2020.
[37] S. B. Ahsan, "New consistency orchestrators for emerging distributed systems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois at Urbana, Champaign, IL, USA, 2020.
[38] J. H. Lubis and E. M. Zamzami, "Relational database reconstruction from SQL to entity relational diagrams," *J. Phys., Conf. Ser.*, vol. 1566, no. 1, Jun. 2020, Art. no. 012072.
[39] M. Hassan, "Big data, big decisions choosing the right database," 2024, *arXiv:2405.02506*.
[40] Z. Hong, S. Guo, E. Zhou, W. Chen, H. Huang, and A. Zomaya, "GriDB: Scaling blockchain database via sharding and off-chain cross-shard mechanism," 2024, *arXiv:2407.03750*.
[41] I. C. Saidu, M. Yusuf, F. C. Nemariyi, and A. C. George, "Indexing techniques and structured queries for relational databases management systems," *J. Nigerian Soc. Phys. Sci.*, vol. 2024, p. 2155, Sep. 2024.
[42] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Integrating polystore RDBMS with common in-memory data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 5762–5764.
[43] P. Raj, "A detailed analysis of noSQL and newSQL databases for bigdata analytics and distributed computing," in *Advances in Computers*, vol. 109. Cambridge, MA, USA: Elsevier, 2018, pp. 1–48.
[44] P. Badgujar, "Optimizing data storage and retrieval in noSQL databases strategies for scalability," *J. Technol. Innov.*, vol. 4, no. 2, 2023.
[45] E. Popescu and A. Radu, "A comparative study of scalability and performance in nosql databases for big data storage and retrieval," *Int. J. Appl. Health Care Anal.*, vol. 5, no. 12, pp. 16–27, 2020.
[46] P. Atzeni, F. Bugiotti, L. Cabibbo, and R. Torlone, "Data modeling in the NoSQL world," *Comput. Standards Interfaces*, vol. 67, Jan. 2020, Art. no. 103149.
[47] O. Osemwegie, K. Okokpujie, N. Nkordeh, C. Ndujiuba, S. John, and U. Stanley, "Performance benchmarking of key-value store NoSQL databases," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 8, no. 6, p. 5333, Dec. 2018.
[48] P. Gómez, C. Roncancio, and R. Casallas, "Analysis and evaluation of document-oriented structures," *Data Knowl. Eng.*, vol. 134, Jul. 2021, Art. no. 101893.
[49] N. Kanakaris, D. Michail, and I. Varlamis, "A comparative survey of graph databases and software for social network analytics: The link prediction perspective," in *Graph Databases*. Boca Raton, FL, USA: CRC Press, 2023, pp. 36–55.
[50] D. Preuveneers and W. Joosen, "Automated configuration of NoSQL performance and scalability tactics for data-intensive applications," *Informatics*, vol. 7, no. 3, p. 29, Aug. 2020.
[51] B. M. Abdelhafiz, "Distributed database using sharding database architecture," in *Proc. IEEE Asia Pacific Conf. Comput. Sci. Data Eng. (CSDE)*, Dec. 2020, pp. 1–17.
[52] M. V. Sokolova, F. J. Gómez, and L. N. Borisoglebskaya, "Migration from an SQL to a hybrid SQL/NoSQL data model," *J. Manage. Anal.*, vol. 7, no. 1, pp. 1–11, Jan. 2020.
[53] S. Duggirala, "NewSQL databases and scalable in-memory analytics," in *Advances in Computers*, vol. 109. Cambridge, MA, USA: Elsevier, 2018, pp. 49–76.
[54] P. Valduriez, R. Jiménez-Peris, and M. T. Özsu, "Distributed database systems: The case for newsql," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLVIII: Special Issue In Memory of Univ. Prof. Dr. Roland Wagner*. Berlin, Germany: Springer, 2021, pp. 1–15.
[55] E. Pina, F. Sá, and J. Bernardino, "NewSQL databases assessment: CockroachDB, MariaDB xpand, and VoltDB," *Future Internet*, vol. 15, no. 1, p. 10, Dec. 2022.
[56] G. A. Schreiner, D. Duarte, G. D. Bianco, and R. D. S. Mello, "A hybrid partitioning strategy for NewSQL databases: The VoltDB case," in *Proc. 21st Int. Conf. Inf. Integr. Web-Based Appl. Services*, Dec. 2019, pp. 353–360.
[57] J. Z. A. Chung, M. M. L. Chang, and S. K. Phang, "Recent advances and benchmarking of NewSQL for OLTP and OLAP in the big data age," in *Proc. IEEE 13th Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, May 2023, pp. 168–174.
[58] T. Taipalus and H. Grahn, "NewSQL database management system compiler errors: Effectiveness and usefulness," *Int. J. Hum.-Comput. Interact.*, vol. 39, no. 20, pp. 3936–3947, Dec. 2023.
[59] A. Budholia, "NewSQL monitoring system," Master's Thesis, Dept. Comput. Sci., San José State Univ., 2021.
[60] M. Nuriev, R. Zaripova, A. Potapov, and M. Kuznetsov, "Achieving new SQL query performance levels through parallel execution in SQL server," *E3S Web Conf.*, vol. 460, May 2023, Art. no. 04005.
[61] M. Tomar, S. Ramalingam, and P. Krishnaswamy, "Cloud-native enterprise platform engineering: Building scalable, resilient, and secure cloud architectures for global enterprises," *Austral. J. Mach. Learn. Res. Appl.*, vol. 3, no. 1, pp. 601–639, 2023.
[62] R. Guo, X. Luan, L. Xiang, X. Yan, X. Yi, J. Luo, Q. Cheng, W. Xu, J. Luo, F. Liu, Z. Cao, Y. Qiao, T. Wang, B. Tang, and C. Xie, "Manu: A cloud native vector database management system," 2022, *arXiv:2206.13843*.
[63] W. Cao, Y. Liu, Z. Cheng, N. Zheng, W. Li, W. Wu, L. Ouyang, P. Wang, Y. Wang, R. Kuan, Z. Liu, and T. Zhang, "POLARDB meets computational storage: Efficiently support analytical workloads in cloud-native relational database," in *Proc. 18th USENIX Conf. File Storage Technol. (FAST)*, 2020, pp. 29–41.
[64] B. Scholl, T. Swanson, and P. Jausovec, *Cloud Native: Using Containers, Functions, and Data To Build Next-generation Applications*. Sebastopol, CA, USA: O'Reilly Media, 2019.
[65] S. Flowers, "Design and implement a replication strategy," in *Designing and Implementing Cloud-native Applications Using Microsoft Azure Cosmos DB: Study Companion for the DP-420 Exam*. Berkeley, CA, USA: Apress, 2023, pp. 83–91.
[66] M. Venugopal and C. Reddy, "Serverless through cloud native architecture," *Int. J. Eng. Res. Technol.*, vol. 10, pp. 484–496, Jul. 2021.
[67] J. Gilbert, *Cloud Native Development Patterns and Best Practices: Practical Architectural Patterns for Building Modern, Distributed Cloud-native Systems*. Birmingham, U.K.: Packt Publishing Ltd, 2018.
[68] H. Gadde, "Ai-driven data indexing techniques for accelerated retrieval in cloud databases," *Revista de Inteligencia Artif. en Medicina*, vol. 15, no. 1, pp. 583–615, 2024.
[69] A. Karunamurthy, M. Yuvaraj, J. Shahithya, and V. Thenmozhi, *Cloud Database: Empowering Scalable and Flexible Data Management*. Pondicherry, India: Quing Publications, 2023.
[70] O.-C. Marcu and P. Bouvry, "Big data stream processing," Ph.D. dissertation, Dept. Comput. Sci., Univ. Luxembourg, Esch-sur-Alzette, Luxembourg, 2024.
[71] C. Asiminidis, G. Kokkonis, and S. Kontogiannis, "Database systems performance evaluation for IoT applications," *Int. J. Database Manage. Syst.*, vol. 10, no. 6, pp. 1–14, Dec. 2018.
[72] G. Vemulapalli, "Optimizing nosql database performance: Elevating api responsiveness in high-throughput environments," *Int. Mach. Learn. J. Comput. Eng.*, vol. 6, no. 6, pp. 1–14, 2023.
[73] V. B. Ramu, "Optimizing database performance: Strategies for efficient query execution and resource utilization," *Int. J. Comput. Trends Technol.*, vol. 71, no. 7, pp. 15–21, Jul. 2023.
[74] J.-T. Yun, S.-K. Yoon, J.-G. Kim, and S.-D. Kim, "Effective data prediction method for in-memory database applications," *J. Supercomput.*, vol. 76, no. 1, pp. 580–601, Jan. 2020.
[75] Z. Wu, J. Sun, Y. Zhang, Z. Wei, and J. Chanussot, "Recent developments in parallel and distributed computing for remotely sensed big data processing," *Proc. IEEE*, vol. 109, no. 8, pp. 1282–1305, Aug. 2021.
[76] Z. Ge, D. Loghin, B. C. Ooi, P. Ruan, and T. Wang, "Hybrid blockchain database systems: Design and performance," *Proc. VLDB Endowment*, vol. 15, no. 5, pp. 1092–1104, Jan. 2022.
[77] H. Gadde, "Ai-enhanced adaptive resource allocation in cloud-native databases," *Revista de Inteligencia Artif. en Medicina*, vol. 13, no. 1, pp. 443–470, 2022.
[78] S. Maiyya, V. Zakhary, D. Agrawal, and A. E. Abbadi, "Database and distributed computing fundamentals for scalable, fault-tolerant, and consistent maintenance of blockchains," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 2098–2101, Aug. 2018.
[79] M. R. Mesbahi, A. M. Rahmani, and M. Hosseinzadeh, "Reliability and high availability in cloud computing environments: A reference roadmap," *Hum.-Centric Comput. Inf. Sci.*, vol. 8, no. 1, pp. 1–31, Dec. 2018.
[80] M. Daňhel, "Prediction and analysis of mission critical systems dependability," Ph.D. dissertation, Dept. Digit. Des., Czech Tech. Univ., Prague, Czechia, 2018.

[81] P. Nama, M. Bhoyar, S. Chinta, and P. Reddy, "Optimizing database replication strategies through machine learning for enhanced fault tolerance in cloud-based environments," *Mach. Learn.*, vol. 63, no. 3, pp. 30–44, 2023.

[82] H. Howard and R. Mortier, "Paxos vs raft: Have we reached consensus on distributed consensus?" in *Proc. 7th Workshop Princ. Pract. Consistency Distrib. Data*, Apr. 2020, pp. 1–9.

[83] Y. Aldwyan and R. O. Sinnott, "Latency-aware failover strategies for containerized web applications in distributed clouds," *Future Gener. Comput. Syst.*, vol. 101, pp. 1081–1095, Dec. 2019.

[84] S. Ramalingam, R. K. Inampudi, and P. Krishnaswamy, "Cloud-native platform engineering for high availability: Building fault-tolerant enterprise cloud architectures with microservices and kubernetes," *J. Sci. Technol.*, vol. 4, no. 2, pp. 139–177, 2023.

[85] O. Lindén, "Cross region cloud redundancy: A comparison of a single-region and a multi-region approach," M.S. thesis, Fac. Sci. Technol., Umeå Univ., Umeå, Sweden, 2023.

[86] S. J. Kamatkar, A. Kamble, A. Viloria, L. H. Fernández, and E. G. Cali, "Database performance tuning and query optimization," in *Proc. 3rd Int. Conf. Data Mining Big Data (DMBD)*, Shanghai, China. Cham, Switzerland: Springer, Jan. 2018, pp. 3–11.

[87] P. Sioulas, "Efficient concurrent analytical query processing using data and workload-conscious sharing," EPFL, Lausanne, Switzerland, Tech. Rep. EPFL-THESIS-10298, 2023.

[88] A. Katari and M. Ankam, "Data Lakes and Optimizing Query," *Int. J. Novel Res. Develop. (IJNRD).*, vol. 5, no. 9, pp. 5–21, Sep. 2020.

[89] C. Riegger, T. Vinçon, and I. Petrov, "Efficient data and indexing structure for blockchains in enterprise systems," in *Proc. 20th Int. Conf. Inf. Integr. Web-Based Appl. Services*, Nov. 2018, pp. 173–182.

[90] C. Forresi, M. Francia, E. Gallinucci, and M. Golfarelli, "Cost-based optimization of multistore query plans," *Inf. Syst. Frontiers*, vol. 25, no. 5, pp. 1925–1951, Oct. 2023.

[91] M. M. Rahman, S. Islam, M. Kamruzzaman, and Z. Joy, "Advanced query optimization in sql databases for real-time big data analytics," *Academic J. Bus. Admin., Innov. Sustainability*, vol. 4, no. 3, pp. 1–14, 2024.

[92] A. G. Chowdhury, M. Illian, Ł. Wiśniewski, and J. Jasperneite, "An approach for data pipeline with distributed query engine for industrial applications," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1005–1008.

[93] Z. Yang, in *Machine Learning for Query Optimization*. Berkeley, CA, USA: Univ. of California, 2022.

[94] F. Nilsson and D. Mihajlovic, "Optimizing databases in the cloud based on performance and cost savings," Master's thesis, Dept. Elect. Inf. Technol., Lund Univ., 2022

[95] A. Katari and D. Kalla, "Cost optimization in cloud-based financial data lakes: Techniques and case studies," *ESP J. Eng. Technol. Advancements*, vol. 1, no. 1, pp. 150–157, 2021.

[96] M. Caraman, G. Miricinschi, and A. Blîndu, "Technical scientific conference of undergraduate, master, and PhD students," Ph.D. dissertation, Dept. Softw. Eng. Automat., Tech. Univ. Moldova, Chişinău, Republic of Moldova, pp. 681–685, 2024, vol. 2.

[97] H. DaCosta, "Configurable online multi-tiered storage in a database management system," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, U.K., 2023.

[98] C. Li, J. Liu, B. Lu, and Y. Luo, "Cost-aware automatic scaling and workload-aware replica management for edge-cloud environment," *J. Netw. Comput. Appl.*, vol. 180, Apr. 2021, Art. no. 103017.

[99] Y. Li and F. Zhang, "Remote data auditing for cloud-assisted WBANs with pay-as-you-go business Model," *Chin. J. Electron.*, vol. 32, no. 2, pp. 248–261, Mar. 2023.

[100] S. Chinamanagonda, "AI-driven cloud cost management-AI tools for optimizing cloud resource allocation and costs," *Int. J. Sci. Res. (IJSR)*, vol. 12, no. 12, pp. 2135–2149, Dec. 2023.

[101] M. Niinimäki, F. Abaunza, T. Niemi, P. Thanisch, and J. Kommeri, "Energy-efficient query processing in a combined database and web service environment," *Green Comput. Strategies Competitive Advantage Bus. Sustainability*, vol. 2018, pp. 62–88, Jan. 2018.

[102] A. Uzzaman, M. M. I. Jim, N. Nishat, and J. Nahar, "Optimizing sql databases for big data workloads: Techniques and best practices," *Academic J. Bus. Admin., Innov. Sustainability*, vol. 4, no. 3, pp. 15–29, 2024.

[103] M. A. Salamkar, "Scalable data architectures: Key principles for building systems that efficiently manage growing data volumes and complexity," *Int. J. Sci. Res.*, vol. 10, no. 1, pp. 1737–1744, Jan. 2021.

[104] H. M. Waidyasooriya and M. Hariyama, "Multi-FPGA accelerator architecture for stencil computation exploiting spacial and temporal scalability," *IEEE Access*, vol. 7, pp. 53188–53201, 2019.

[105] G. Jonatan, H. Cho, H. Son, X. Wu, N. Livesay, E. Mora, K. Shivdikar, J. L. Abellán, A. Joshi, D. Kaeli, and J. Kim, "Scalability limitations of processing-in-memory using real system evaluations," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 52, no. 1, pp. 63–64, Jun. 2024.

[106] M. Abbasi, M. V. Bernardo, P. Váz, J. Silva, and P. Martins, "Adaptive and scalable database management with machine learning integration: A PostgreSQL case study," *Information*, vol. 15, no. 9, p. 574, Sep. 2024.

[107] A. M. G. El-Din and M. Senousy, "A solution for handling big data heterogeneity problem," in *Digital Transformation Technology: Proceedings of ITAF 2020*. Cham, Switzerland: Springer, 2022, pp. 185–194.

[108] L. Wang, "Heterogeneous data and big data analytics," *Autom. Control Inf. Sci.*, vol. 3, no. 1, pp. 8–15, Aug. 2017.

[109] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Approximate query processing for big data in heterogeneous databases," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 5765–5767.

[110] S. Janković, S. Mladenović, D. Mladenović, S. Vesković, and D. Glavić, "Schema on read modeling approach as a basis of big data analytics integration in EIS," *Enterprise Inf. Syst.*, vol. 12, nos. 8-9, pp. 1180–1201, Oct. 2018.

[111] E. Guliyev, "Comparative analysis of multi-model databases," Bachelor's thesis, Dept. Softw. Eng., Charles Univ., Prague, Czech Republic, 2022

[112] M. Sadeghi, A. Carenini, O. Corcho, M. Rossi, R. Santoro, and A. Vogelsang, "Interoperability of heterogeneous systems of systems: From requirements to a reference architecture," *J. Supercomput.*, vol. 80, no. 7, pp. 8954–8987, May 2024.

[113] M. Ramos, R. Sánchez-De-Madariaga, J. Barros, L. Carrajo, G. Vázquez, S. Pérez, M. Pascual, F. Martín-Sánchez, and A. Muñoz-Carrero, "An archetype query language interpreter into MongoDB: Managing NoSQL standardized electronic health record extracts systems," *J. Biomed. Informat.*, vol. 101, Jan. 2020, Art. no. 103339.

[114] Z. Sun, K. D. Strang, and F. Pambel, "Privacy and security in the big data paradigm," *J. Comput. Inf. Syst.*, vol. 60, no. 2, pp. 146–155, Feb. 2018.

[115] D. Zhang, "Big data security and privacy protection," in *Proc. 8th Int. Conf. Manage. Comput. Sci. (ICMCS)*, Jun. 2018, pp. 275–278.

[116] F. M. Abdullah, "Privacy, security and legal challenges in big data," *Int. J. Civ. Eng. Technol.*, vol. 9, no. 13, pp. 90–1682, 2018.

[117] M. Ocenas, I. Homoliak, P. Hanacek, and K. Malinka, "Security and encryption at modern databases," in *Proc. 4th Int. Conf. Cryptogr., Secur. Privacy*, Jan. 2020, pp. 19–23.

[118] S. A. Deshmukh and S. Kasar, "Applications of blockchain technology in privacy preserving and data security for real time (data) applications," *Concurrency Comput., Pract. Exper.*, vol. 36, no. 26, p. 8277, Nov. 2024.

[119] A. Cuzzocrea, C. K. Leung, A. M. Olawoyin, and E. Fadda, "Supporting privacy-preserving big data analytics on temporal open big data," *Proc. Comput. Sci.*, vol. 198, pp. 112–121, Jun. 2022.

[120] M. O. Hegazi, D. Kumar, and K. Zia, "Moving from heterogeneous data sources to big data: Interoperability and integration issues," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 207–212, 2018.

[121] M. M. Almutairi, M. Yamin, and G. Halikias, "An analysis of data integration challenges from heterogeneous databases," in *Proc. 8th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2021, pp. 352–356.

[122] J. Delgado, "Interoperability effect in big data," in *Handbook of Smart Cities*. Cham, Switzerland: Springer, 2020, pp. 875–901.

[123] V. Raghupathy, O. I. Khalaf, C. A. T. Romero, S. Sengan, and D. K. Sharma, "Interactive middleware services for heterogeneous systems," *Comput. Syst. Sci. Eng.*, vol. 41, no. 3, p. 1241, 2022.

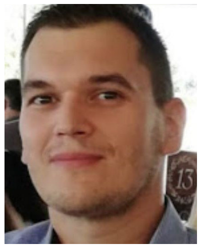
[124] P. Banerjee, "System integration, from middleware to Apis," *Int. J. Comput. Trends Technol.*, vol. 72, no. 3, pp. 37–45, Mar. 2024.

[125] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Federated query processing for big data in data science," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 6145–6147.

[126] Y. Shi, M. Dong, W. Zhang, L. Liu, Y. Zheng, L. Cui, and J. Zhang, "AdaptScale: An adaptive data scaling controller for improving the multiple performance requirements in clouds," *Future Gener. Comput. Syst.*, vol. 105, pp. 814–823, Apr. 2020.

[127] S. R. Goniwada, in *Cloud Native Architecture and Design: A Handbook for Modern Day Architecture and Design With Enterprise-Grade Examples*. New York, NY, USA: Apress, 2022.

[128] S. Misic, "Modernizing an on-premise monolithic soft-ware application to a cloud-native solution," Master's thesis, Dept. Comput. Sci., Blekinge Inst. Technol., Karlskrona, Sweden, Tech.Rep. MSE-2023-08, 2023.
[129] G. Mihai, "Multi-model database systems: The state of affairs," *Ann. Dunarea de Jos Univ. Galati. Fascicle I. Econ. Appl. Informat.*, vol. 26, no. 2, pp. 211–215, Aug. 2020.
[130] C. Zhang, "Performance benchmarking and query optimization for multi-model databases," Ph.D. dissertation, Dept. Comput. Sci., Univ. Helsinki, Helsinki, Finland, 2021.
[131] H. Gadde, "Ai-driven predictive maintenance in relational database systems," *Int. J. Mach. Learn. Res. Cybersecurity Artif. Intell.*, vol. 12, no. 1, pp. 386–409, 2021.
[132] A. P. Rodrigues, R. Fernandes, A. Bhandary, A. C. Shenoy, A. Shetty, and M. Anisha, "Real-time Twitter trend analysis using big data analytics and machine learning techniques," *Wireless Commun. Mobile Comput.*, vol. 2021, no. 1, Jan. 2021, Art. no. 3920325.
[133] A. Ouared, M. Amrani, and P.-Y. Schobbens, "Explainable AI for DBA: Bridging the DBA's experience and machine learning in tuning database systems," *Concurrency Comput., Pract. Exper.*, vol. 35, no. 21, p. 7698, Sep. 2023.
[134] A. B. Alexandru and G. J. Pappas, "Secure multi-party computation for cloud-based control," *Privacy Dyn. Syst.*, vol. 2019, pp. 179–207, Jan. 2019.
[135] F. Casalini, J. L. González, and T. Nemoto, "Mapping commonalities in regulatory approaches to cross-border data transfers," OECD Publishing, Paris, France, OECD Trade Policy Papers 248, 2021.
[136] M. H. Maturi, "Optimizing energy efficiency in edge-computing environments with dynamic resource allocation," *Environments*, vol. 13, no. 7, pp. 1–8, 2024.
[137] R. R. Pansara, "Edge computing in master data management: Enhancing data processing at the source," *Int. Trans. Artif. Intell.*, vol. 6, no. 6, pp. 1–11, 2022.
[138] Y. Zhou, S. Taneja, X. Qin, W.-S. Ku, and J. Zhang, "EDOM: Improving energy efficiency of database operations on multicore servers," *Future Gener. Comput. Syst.*, vol. 105, pp. 1002–1015, Apr. 2020.
[139] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Mining Anal.*, vol. 3, no. 2, pp. 85–101, Jun. 2020.
[140] D. Syed, A. Zainab, A. Ghrayeb, S. S. Refaat, H. Abu-Rub, and O. Bouhali, "Smart grid big data analytics: Survey of technologies, techniques, and applications," *IEEE Access*, vol. 9, pp. 59564–59585, 2021.
[141] T. R. Rao, P. Mitra, R. Bhatt, and A. Goswami, "The big data system, components, tools, and technologies: A survey," *Knowl. Inf. Syst.*, vol. 60, no. 3, pp. 1165–1245, Sep. 2019.
[142] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: A survey," *J. Internet Technol.*, vol. 21, no. 2, pp. 393–405, Mar. 2020.
[143] N. Deepa, Q.-V. Pham, D. C. Nguyen, S. Bhattacharya, B. Prabadevi, T. R. Gadekallu, P. K. R. Maddikunta, F. Fang, and P. N. Pathirana, "A survey on blockchain for big data: Approaches, opportunities, and future directions," *Future Gener. Comput. Syst.*, vol. 131, pp. 209–226, Jun. 2022.
[144] S. Mazumdar, D. Seybold, K. Kritikos, and Y. Verginadis, "A survey on data storage and placement methodologies for cloud-big data ecosystem," *J. Big Data*, vol. 6, no. 1, pp. 1–37, Dec. 2019.



ELIAS DRITSAS received the Diploma, M.Sc., and Ph.D. degrees in computer science and informatics from the Department of Computer Engineering and Informatics, University of Patras, Greece, and the M.B.A. degree from the University of Derby, U.K. He participated in many development projects funded by the EC and the Greek Secretariat of Research and Technology. He is the author and co-author of publications in machine learning and data analysis. His research interests include artificial intelligence, machine learning, big data, databases, data mining, human–computer interaction, cloud computing, security, and trust. He is a member of the Technical Chamber of Greece and the Association for Computing Machinery. He serves as a regular reviewer for several technical journals and conferences.

MARIA TRIGKA received the Diploma degree from the Department of Electrical and Computer Engineering and the M.Sc. and Ph.D. degrees in signal processing for wireless communications from the Department of Computer Engineering and Informatics, University of Patras, Greece. As a Ph.D. candidate, she was a Research Scholar under the project Strengthening Human Research Potential through Doctoral Research, co-financed by Greece-State Scholarships Foundation (IKY) and the European Union. Also, she has been involved in research projects co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH CREATE INNOVATE that emphasized artificial intelligence and machine learning for developing health and cultural services for the hard of hearing. She is the author and co-author of publications in the areas of signal processing, wireless communications, and machine learning. Her research interests include statistical signal processing and learning, with a focus on estimation theory, adaptive/distributed signal processing, and sparse representations.