

A Survey of Data Handling Techniques in Next-Generation Database Systems

Dr. Parth Gautam

Associate Professor

Department of Computer Sciences and Applications

Mandsaur University, Mandsaur

parth.gautam@meu.edu.in

Abstract—The rapid expansion of digital technologies such as cloud computing, Internet of Things (IoT), artificial intelligence, and big-data applications has significantly increased the volume, variety, and velocity of data generated across modern systems. Traditional relational database management systems (RDBMS), while effective for structured and transactional data, often encounter limitations in handling large-scale, heterogeneous, and real-time data environments. Consequently, next-generation database systems, including NoSQL, NewSQL, graph, distributed, and cloud-native databases, have emerged to address these challenges through scalable, flexible, and high-performance architectures. This review paper surveys major data handling techniques adopted in next-generation database systems, emphasizing data storage models, organization strategies, query processing, optimization mechanisms, and partitioning techniques. The study further discusses advanced challenges associated with scalability, performance, security, privacy, and governance in distributed and cloud-based environments. In addition, emerging technologies such as artificial intelligence-driven database management, serverless databases, edge computing, blockchain integration, and multi-cloud optimization are examined as transformative developments shaping future database ecosystems. A comprehensive literature review summarizes recent contributions, identifies limitations, and highlights research opportunities. The findings indicate that intelligent and adaptive data handling techniques are essential for ensuring efficient, secure, and scalable database management in modern data-intensive applications.

Keywords—Next-Generation Database Systems, Data Handling Techniques, NoSQL Databases, Cloud-Native Databases, Data Partitioning.

I. INTRODUCTION

A database is an organized collection of data stored and managed electronically for efficient access, retrieval, and updating [1]. 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.

The exponential growth of digital technologies, including cloud computing, Internet of Things (IoT), artificial intelligence, and social networking platforms, has resulted in unprecedented volumes of data generation [2]. Modern applications produce highly diverse and rapidly changing datasets that require efficient storage, processing, and management techniques. Traditional relational database management systems (RDBMS), although reliable for structured data and transactional processing, often face limitations when dealing with large-scale, heterogeneous, and real-time data environments.

To address these challenges, next-generation database systems have emerged as advanced data management platforms capable of supporting distributed computing, cloud deployment, and scalable processing [3]. These systems include NoSQL, NewSQL, graph, and cloud-native databases that provide flexible schemas, horizontal scalability, fault tolerance, and high availability [4][5]. Unlike traditional databases, modern systems are designed to manage both structured and unstructured data while maintaining efficient performance across geographically distributed infrastructures.

An essential feature of next-generation databases is the adoption of advanced data handling techniques. Data handling refers to the processes involved in data storage, indexing, partitioning, replication, querying, security, and real-time stream processing. Efficient data handling techniques are necessary to ensure data integrity, minimize latency, improve query execution, and maintain reliable performance in large-scale environments. Recent developments in stream and distributed data processing have further enhanced database capabilities by enabling elastic resource allocation and continuous analytics for data-intensive applications [6].

Given the pace at which database technology is evolving, coupled with the complexities of present-day computers, it has become imperative that one understands all data-handling techniques. As such, this review article examines various data handling techniques used in advanced database systems, their advantages and disadvantages, and directions for future research.

A. Structure of The Paper

This paper is organized as follows: Section II presents an overview of next-generation database systems. Section III discusses major data handling techniques. Section IV highlights advanced issues and emerging developments. Section V reviews related literature, and Section VI concludes the paper with future research directions.

II. NEXT-GENERATION DATABASE SYSTEMS

Next-generation database systems have emerged to address the limitations of traditional database architectures in managing large, diverse, and rapidly growing datasets. The advancement of technology from file-system databases to

relational, distributed, and intelligent databases indicates changing needs in data management. This is because next-generation databases are characterized by features such as being distributed, automated, cloud-integrated, and offering higher-level security than traditional databases. There are also other types of databases, like NoSQL, NewSQL, distributed, graph, time series, and multi-model databases that offer specialized ways of data management [7].

A. Evolution of Database Systems

Database systems have evolved significantly from simple file storage mechanisms to intelligent and distributed platforms capable of handling large volumes of structured and unstructured data. Early systems relied on file-based storage, which often resulted in redundancy, inconsistency, and poor scalability. To overcome these limitations, relational database management systems (RDBMS) emerged, organizing data into structured tables with predefined schemas and becoming widely adopted due to their reliability and efficient query processing.

As digital technologies expanded, traditional relational databases faced challenges in managing large, diverse, and rapidly growing datasets. This led to the development of more scalable and flexible database architectures capable of supporting distributed computing and real-time analytics.

The rise of cloud computing further transformed database systems by enabling scalable storage, automated resource allocation, and high availability. Modern applications such as e-commerce, healthcare, and social media have also increased demand for low-latency and continuous data access [8][9]. Today, database systems extend beyond storage by incorporating automation, predictive analytics, and self-management capabilities such as self-monitoring and optimization. Overall, database evolution reflects a shift from centralized systems to intelligent, scalable, and distributed environments that support modern digital and business requirements.

B. Characteristics of Next-Generation Databases

Next-generation databases are designed to handle modern data requirements, including large-scale, diverse, and rapidly growing datasets. Unlike traditional databases, they emphasize scalability, flexibility, real-time processing, automation, and distributed computing to improve performance and efficiency.

- **Scalability:** Supports horizontal scaling across multiple servers to manage growing workloads efficiently.
- **Flexibility:** Handles structured, semi-structured, and unstructured data with adaptable schemas.
- **Real-Time Processing:** Enables low-latency data access and real-time analytics for faster decision-making.
- **Distributed Architecture:** Distributes data across multiple nodes, improving reliability and fault tolerance.
- **Automation and Intelligence:** Uses AI and machine learning for query optimization, monitoring, and resource management.
- **Security and Privacy:** Includes encryption, access control, and monitoring to protect sensitive data.
- **Cloud Integration:** Supports cloud-native storage, scalability, backup, and disaster recovery.

C. Types of Next-Generation Database Systems

Next-generation database systems include various database models developed to manage diverse and complex data requirements. These systems improve scalability, flexibility, real-time processing, and distributed operations, making them suitable for modern digital applications.

- **NoSQL Databases:** NoSQL databases are designed to handle unstructured and semi-structured data without relying on fixed schemas. They support distributed environments and are commonly used in social media platforms, recommendation systems, and real-time web applications due to their scalability and flexibility.
- **NewSQL Databases:** NewSQL databases combine the scalability of NoSQL systems with the transactional consistency of traditional relational databases. They support structured query processing while enabling high-speed transactions, making them suitable for enterprise and financial applications.
- **Distributed Databases:** Distributed databases store data across multiple physical locations while functioning as a single system. This architecture improves fault tolerance, reliability, and accessibility, especially for organizations operating across different geographical regions.
- **Graph Databases:** Graph databases organize data using nodes and relationships, enabling efficient analysis of highly connected information. They are widely used in fraud detection, social networks, recommendation systems, and biological data analysis.
- **Time-Series Databases:** Time-series databases are optimized for managing timestamp-based data generated continuously over time. They are commonly applied in IoT systems, healthcare monitoring, industrial sensors, weather forecasting, and stock market analysis.
- **Multi-Model Databases:** Multi-model databases integrate multiple data models, such as document, graph, relational, and key-value structures, within a single platform. This approach reduces operational complexity and improves interoperability between different types of data.

Figure 1 illustrates the major types of next-generation database systems, including NoSQL, NewSQL, Distributed, Graph, Time-Series, and Multi-Model databases. These database paradigms are designed to address the scalability, flexibility, performance, and data management requirements of modern applications and large-scale data environments.

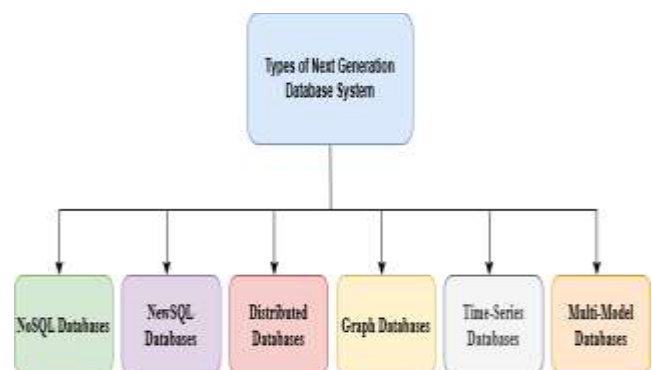


Fig. 1. Types of Next Generation Database System

III. DATA HANDLING TECHNIQUES IN NEXT-GENERATION DATABASE SYSTEMS

Data handling techniques represent the foundation of next-generation database systems, enabling organizations to efficiently manage the increasing complexity, scale, and diversity of modern data environments. Unlike conventional database architectures that primarily focus on structured transactional records, contemporary database systems must support heterogeneous datasets generated from multiple sources, including cloud applications, Internet of Things (IoT) devices, enterprise systems, social media platforms, and real-time analytics infrastructures. The growing volume and velocity of digital information have accelerated the development of advanced data handling mechanisms that improve storage efficiency, computational performance, scalability, and security.

Modern database environments employ intelligent storage models, distributed processing frameworks, replication mechanisms, and privacy-preserving strategies to ensure reliable and high-performance operations. Furthermore, the increasing reliance on real-time applications has emphasized the importance of stream processing and low-latency query optimization techniques. This section presents a detailed

TABLE I. THE MAJOR STORAGE APPROACHES

Storage Model	Characteristics	Application Areas
Relational Storage	Structured tabular format with schema	Banking, enterprise systems
Document-Based Storage	Flexible schema using JSON/XML	Web applications, content management
Key-Value Storage	High-speed data retrieval	Caching systems, online platforms
Graph Storage	Relationship-oriented architecture	Social networks, fraud detection
Column-Oriented Storage	Optimized analytical processing	Big data analytics

Relational storage remains important in transactional systems requiring strong consistency and structured query processing. However, document-oriented databases have gained popularity due to their ability to support flexible schemas and evolving data structures. Similarly, graph databases are increasingly used to handle highly interconnected datasets in which relationships among entities significantly influence analytical outcomes.

B. Data Processing and Query Optimization Techniques

Data processing is a crucial component of modern database systems, enabling rapid handling of large volumes of information while maintaining accuracy and consistency. Efficient query execution improves database performance and enhances user experience.

Query optimization techniques help reduce computational overhead and improve response time.

- **Index-Based Query Processing:** Uses indexing structures to accelerate data retrieval and reduce search complexity, improving query speed.
- **Cost-Based Optimization:** Selects the most efficient query execution plan based on estimated computational costs and system resources.
- **Parallel Query Execution:** Distributes query workloads across multiple processors or nodes to achieve faster computation and improved performance.

discussion of the major data handling approaches adopted in next-generation database systems.

A. Data Storage, Organization, and Processing Techniques

Efficient data storage and organization constitute critical requirements in next-generation database systems because data structures have become increasingly diverse and computationally demanding. Traditional relational databases rely on predefined schemas and tabular formats for storing structured information. However, modern digital ecosystems generate structured, semi-structured, and unstructured datasets simultaneously, necessitating flexible and adaptive storage architectures

1) Data Storage Models in Next-Generation Databases

Modern database systems employ different storage techniques depending on workload requirements, scalability demands, and application domains. These storage models are designed to maximize efficiency while minimizing storage redundancy and retrieval latency. Table I summarizes major database storage approaches, highlighting their characteristics and application areas across diverse computing environments.

- **Adaptive Query Processing:** Dynamically adjusts query execution strategies according to workload changes and system conditions.

Efficient query optimization is essential in high-volume environments such as e-commerce, healthcare, and financial analytics, where fast response time is critical.

- **Data Partitioning Techniques:** Data partitioning is the process of dividing large datasets into smaller, manageable segments to improve database performance, scalability, and efficient data access. It helps distribute workloads across multiple systems, reducing processing delays and improving query execution.
- **Horizontal Partitioning:** Divides data into rows distributed across different servers or clusters, improving scalability and reducing workload concentration.
- **Vertical Partitioning:** Separates data into columns based on application requirements, improving query speed and storage efficiency.
- **Range Partitioning:** Groups data according to specific value ranges, enabling faster query processing and efficient data retrieval.
- **Hash Partitioning:** Uses hashing algorithms to distribute data evenly across partitions, ensuring balanced workload distribution.

- **Dynamic Partitioning:** Automatically adjusts partitions according to workload patterns, supporting elastic scalability and load balancing.

IV. ADVANCED ISSUES AND EMERGING DEVELOPMENTS IN NEXT-GENERATION DATABASES

The next generation database management systems have evolved significantly to cope up with the growing amounts of data. In addition to providing scalable, flexible, and high-speed database management services, these advanced systems face complex problems related to performance tuning, security, privacy, and effective data governance. Solving these issues will help to develop robust and secure databases in the complex environment of today's computers. Additionally, emerging technologies like artificial intelligence, cloud-native computing, edge computing, and blockchain technology are influencing database designs and management practices to an extent that intelligent and autonomous database management systems are being developed.

A. Scalability and Performance Challenges.

Scalability and performance are important issues in next generation database systems due to massive and fast-growing datasets generated by modern applications. Databases are needed to offer high throughput, low latency and reliable query execution in distributed and cloud environments. Traditional relational databases are often limited in the handling of large-scale and heterogeneous data, and NoSQL and distributed database architectures have been used to achieve better scalability [10].

However, distributed database systems face several performance challenges with respect to data consistency, synchronization, network overhead, and complex transaction management [11]. Techniques such as replication and sharding enhance scalability but may also increase coordination costs and system complexity [12]. Additionally, inefficient query execution and resource management can negatively affect database performance, making query optimization and workload balancing essential [13].

B. Security, Privacy, and Governance Issues

The rapid growth of cloud computing, distributed databases, and big-data applications has made security, privacy, and governance critical concerns in next-generation database systems. Modern databases frequently manage sensitive information, thus increasing the risks of unauthorized access, cyberattacks, and data leakage [14]. To address these threats, advanced security mechanisms such as encryption, authentication, and fine-grained access control are widely adopted to protect data confidentiality and integrity.

Privacy preservation is another major challenge, especially in cloud-native and shared database environments where large-scale personal and organizational data are processed. Privacy-enhancing techniques, including anonymization and secure data-sharing methods, help reduce privacy risks while maintaining data usability.

In addition, effective data governance is essential to ensure regulatory compliance, data quality, ownership management, and policy enforcement. Governance frameworks support secure and transparent data management, improving accountability and trust in next-generation database systems. Therefore, strengthening security, privacy, and governance

mechanisms remains fundamental for reliable and scalable database infrastructures.

C. Emerging Technologies and Future Developments

Emerging technologies are transforming next-generation database systems by improving automation, scalability, and real-time data processing. Artificial intelligence and machine learning techniques are now being applied extensively in databases, where prediction-based maintenance can be achieved, thus reducing human interaction with the system [15].

- **AI and Machine Learning for Predictive Database Management:** AI and ML are used to automate database management tasks such as performance tuning, anomaly detection, and predictive maintenance. These technologies help identify potential failures and optimize resources automatically, reducing manual effort and improving database reliability.
- **Serverless Databases:** Serverless databases automatically scale according to workload demand without requiring manual infrastructure management. This reduces operational complexity, lowers costs, and enables faster application deployment in cloud environments.
- **Edge Computing and Edge Databases:** Edge databases process data near the source instead of relying solely on centralized cloud servers. This reduces latency and supports real-time data processing, making them suitable for IoT systems, smart cities, and autonomous applications.
- **Data Fabric and Unified Data Management:** Data fabric provides a unified architecture for integrating and managing data across cloud, edge, and on-premises systems. It supports automated governance, secure access, and consistent data management across multiple environments.
- **Multi-Cloud and Cloud-Native Database Optimization:** Future databases increasingly support multi-cloud and cloud-native architectures through unified management tools and hybrid database models. These technologies improve scalability, flexibility, and interoperability across different cloud platforms.
- **Blockchain and Decentralized Databases:** Blockchain integration enhances database security and transparency by providing immutable and tamper-resistant records. Decentralized databases use distributed architectures to improve data integrity and secure transaction management.
- **Future Developments:** Future database systems are expected to become increasingly autonomous, AI-driven, and compliance-aware. Advances in automated governance, compliance monitoring, and decentralized architectures will support secure, intelligent, and highly adaptive database environments capable of handling complex modern data workloads.

V. LITERATURE REVIEW

The literature review examines database optimization, AI-oriented databases, Big Data ecosystems, vector databases, and business intelligence systems, highlighting performance, scalability, data handling, limitations, and future research directions.

Haider and Huma (2026) systematically evaluate key optimization techniques—including indexing strategies, query optimization, denormalization, caching frameworks and data partitioning—demonstrating their effectiveness in reducing bottlenecks and improving response times. In addition, it explores emerging database paradigms such as NoSQL systems, in-memory databases, and cloud-native database solutions, assessing their suitability for modern, high-demand applications with dynamic workloads and real-time processing requirements [16].

Weng and Wu (2025) provides a comprehensive comparative analysis of PostgreSQL and MongoDB, focusing on their suitability for AI use cases. Key evaluation criteria include data modeling, query complexity, scalability, ACID compliance, indexing, and integration with AI frameworks. PostgreSQL excels in scenarios requiring strict data consistency, complex querying, and structured data, making it ideal for financial modeling, scientific research, and feature engineering. Conversely, MongoDB's schema-less design, horizontal scalability, and native support for semi-structured data align well with real-time analytics, IoT, and evolving AI datasets [17].

Dritsas and Trigka (2025) 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 [18].

Pan, Wang and Li (2024) review the existing vector database management techniques and systems. For query processing, review similarity score design and selection, vector query types, and vector query interfaces. For storage

and indexing, review various indexes and discuss compression as well as disk-resident indexes. For query optimization and execution, review hybrid query processing, hardware acceleration, and distributed search. then review existing systems, search engines and libraries, and benchmarks [19].

Hosen et al. (2024) Business intelligence (BI) is becoming increasingly important in today's data-driven environment. It connects businesses to existing databases and analyzes large volumes of data to identify valuable information. BI enhances database systems, improves operational efficiency, and supports data-driven decision-making. Artificial intelligence technologies provide relevant insights, including consumer trends, helping organizations better understand their target market and gain a competitive advantage. Analytics also assists in forecasting future market developments, while BI promotes a culture where data serves as the primary basis for decision-making [20].

F. Suter, R. F. Da Silva, A. Gainaru, and S. Klasky (2023) Data generated by scientific workflows is consumed by analysis, visualization, and machine learning components to support inference, justify scientific objectives, and monitor experiment progress. Efficient movement of intermediate data is often more critical than task scheduling. However, traditional workflow management systems mainly optimize task scheduling and assume a “move little, compute for long” model, making them unsuitable for modern data-intensive workflows. Therefore, a new workflow management approach is proposed, featuring an independently managed data plane for storing and streaming data. Workflow components in the application plane interact with this data plane, while the workflow management system acts as a control plane that coordinates workflow execution and manages connections between the application and data planes [21].

The Table II summarizes major studies on next-generation database systems, outlining research focus, contributions, advantages, limitations, and recommendations to evaluate modern data handling techniques and database performance.

TABLE II. SUMMARY OF LITERATURE REVIEW ON DATA HANDLING TECHNIQUES IN NEXT-GENERATION DATABASE SYSTEMS

Authors	Study on	Contributions	Advantages	Limitations	Recommendations
Haider and Huma (2026)	Database optimization techniques and emerging database paradigms	Evaluated indexing, query optimization, caching, partitioning, NoSQL, in-memory, and cloud-native databases	Improved performance and reduced bottlenecks	Limited focus on AI-driven optimization and adaptive database management	Explore intelligent optimization mechanisms for dynamic workloads
Weng and Wu (2025)	Comparative analysis of PostgreSQL and MongoDB for AI applications	Compared data modeling, scalability, indexing, and AI integration capabilities	Clear guidance for database selection in AI environments	Focused mainly on two database systems	Extend comparison to hybrid and multi-model databases
Dritsas and Trigka (2025)	Database evolution in the Big Data era	Comprehensive classification and performance analysis of modern databases	Broad understanding of database paradigms and trade-offs	Primarily conceptual with limited empirical validation	Conduct real-world benchmarking studies
Pan, Wang and Li (2024)	Vector database management systems	Reviewed vector indexing, storage, query processing, optimization, and benchmarks	Comprehensive overview of vector database technologies	Limited discussion of security, privacy, and governance issues	Investigate secure and scalable vector database frameworks
Hosen et al. (2024)	Business intelligence and data-driven decision making	Examined BI, AI integration, analytics, and decision support systems	Enhanced business insights and competitive advantage	Focused mainly on business applications rather than database architectures	Develop advanced AI-enabled BI systems
F. Suter, R. F. Da Silva, A. Gainaru, and S. Klasky (2023)	Data management in scientific workflows	Proposed a data-plane-based workflow management architecture	Improved data movement and workflow efficiency	Limited applicability beyond scientific workflows	Extend to broader distributed environments

VI. CONCLUSION AND FUTURE WORK

The rapid growth of data volume, variety, and velocity has created significant challenges for traditional database systems, driving the evolution of next-generation database technologies. While conventional relational databases remain effective for structured and transactional data processing, they often struggle to meet the scalability, flexibility, and real-time requirements of modern data-intensive applications. To address these limitations, advanced database paradigms such as NoSQL, NewSQL, graph, distributed, and cloud databases have emerged, offering enhanced performance, scalability, and resource efficiency. This review examined key data handling techniques in next-generation database systems, including data storage, organization, query optimization, indexing, and partitioning strategies. These techniques play a critical role in improving database performance, reliability, and scalability. Furthermore, major challenges related to performance, security, privacy, governance, and efficient resource management were analyzed to highlight their impact on modern database environments. Emerging technologies such as artificial intelligence, serverless databases, edge computing, blockchain integration, and multi-cloud database architectures were also explored due to their growing influence on database innovation.

Future research should focus on the development of autonomous and AI-driven database management systems capable of self-optimization, predictive analytics, and intelligent governance. Additionally, hybrid database architectures, advanced privacy-preserving mechanisms, and sustainable multi-cloud data management frameworks represent promising directions for enhancing the efficiency, security, and adaptability of future database systems.

REFERENCES

- [1] A. Parupalli and H. Kali, "An In-Depth Review of Cost Optimization Tactics in Multi-Cloud Frameworks," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 5, pp. 1043–1052, Jun. 2023, doi: 10.48175/IJARST-11937Q.
- [2] S. S. Gill *et al.*, "Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges," *Internet of Things*, vol. 8, p. 100118, Dec. 2019, doi: 10.1016/j.iot.2019.100118.
- [3] J. W. Sajja, "Enterprise Finance Reimagined: Harnessing ERP and Data Innovation for Next-Generation Value Creation," *Comput. Fraud Secur.*, no. 4, pp. 17–26, April, Apr. 2024, doi: 10.52710/cfs.743.
- [4] M. Kari, "AI-Assisted Query Optimization Techniques for Cloud Databases Supporting Hybrid SQL and NoSQL Workloads," *Int. J. Emerg. Res. Eng. Technol.*, vol. 6, no. 4, pp. 62–71, October, 2025, doi: 10.63282/3050-922X.IJERET-V6I4P108.
- [5] S. R. Sirikonda, "Reducing SRE Toil via Safe Autonomous Remediation in Cloud-Native Systems," *Am. J. Technol.*, vol. 5, no. 3, pp. 30–49, Mar. 2026, doi: 10.58425/ajt.v5i3.511.
- [6] D. Martinez-Mosquera, R. Navarrete, S. Luján-Mora, L. Recalde, and A. Andrade-Cabrera, "Integrating OLAP with NoSQL Databases in Big Data Environments: Systematic Mapping," *Big Data Cogn. Comput.*, vol. 8, no. 6, p. 64, Jun. 2024, doi: 10.3390/bdcc8060064.
- [7] H. B. Dama, "A Survey of MySQL Database Administration Techniques and Best Practices," *ESP J. Eng. Technol. Adv.*, vol. 6, no. 1, pp. 89–98, February, 2026.
- [8] V. K. Sharma, "Cloud Computing & IoT: 5G Focused IoT with Cloud Solutions," *Int. J. AI, BigData, Comput. Manag. Stud.*, vol. 6, no. 3, pp. 21–25, Jul. 2025, doi: 10.63282/3050-9416.IJAIBDCMS-V6I3P103.
- [9] H. Ravilla, "Building Scalable Applications with Heroku and Salesforce Integration," *Am. J. Technol.*, vol. 4, no. 3, Dec. pp. 15–36, 2025.
- [10] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo, "SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review," *Big Data Cogn. Comput.*, vol. 7, no. 2, p. 97, May 2023, doi: 10.3390/bdcc7020097.
- [11] M. R. Sundarakumar *et al.*, "A comprehensive study and review of tuning the performance on database scalability in big data analytics," *J. Intell. Fuzzy Syst.*, vol. 44, no. 3, pp. 5231–5255, Mar. 2023, doi: 10.3233/JIFS-223295.
- [12] M. Chanda, "A Low-Cost System for Acquiring Login/Logout Data for On-Ground Racks of in-Flight Entertainment Systems," California State University, 2016.
- [13] S. Solat, "Sharding Distributed Databases: A Critical Review," *J. Intell. Fuzzy Syst.*, vol. 44, no. 1, pp. 1–19, 2024, doi: 10.48550/arXiv.2404.04384.
- [14] O. S. Marpaung, D. A. Alvyn, V. William, M. S. Anggereainy, and A. Kurniawan, "Security and Privacy Issues in Cloud-Based Databases: A Literature Review," in *2023 10th International Conference on ICT for Smart Society (ICISS)*, IEEE, Sep. 2023, pp. 1–6, doi: 10.1109/ICISS59129.2023.10291433.
- [15] N. K. Miryala, "Emerging Trends and Challenges in Modern Database Technologies: A Comprehensive Analysis," no. November, 2024, doi: 10.21275/MS241126103744.
- [16] Z. Haider and Z. Huma, "Optimizing Database Architectures for High-Performance Web Applications: A Comprehensive Analysis," *Glob. Knowl. Acad.*, vol. 7, no. 1, pp. 1–13, 2026.
- [17] Yijie Weng and Jianhao Wu, "Database management systems for artificial intelligence: Comparative analysis of postgres SQL and MongoDB," *World J. Adv. Res. Rev.*, vol. 25, no. 2, pp. 2336–2342, Feb. 2025, doi: 10.30574/wjarr.2025.25.2.0586.
- [18] E. Dritsas and M. Trigka, "Database Systems in the Big Data Era: Architectures, Performance, and Open Challenges," *IEEE Access*, vol. 13, pp. 95068–95084, 2025, doi: 10.1109/ACCESS.2025.3572059.
- [19] J. J. Pan, J. Wang, and G. Li, "Vector Database Management Techniques and Systems," in *Companion of the 2024 International Conference on Management of Data*, in SIGMOD '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 597–604, doi: 10.1145/3626246.3654691.
- [20] M. S. Hosen *et al.*, "Data-Driven Decision Making: Advanced Database Systems for Business Intelligence," *Nanotechnol. Perceptions*, vol. 20, no. S3, pp. 687–704, May 2024, doi: 10.62441/nano-ntp.v20iS3.51.
- [21] F. Suter, R. F. Da Silva, A. Gainaru, and S. Klasky, "Driving Next-Generation Workflows from the Data Plane," in *2023 IEEE 19th International Conference on e-Science (e-Science)*, 2023, pp. 1–10, doi: 10.1109/e-Science58273.2023.10254849.