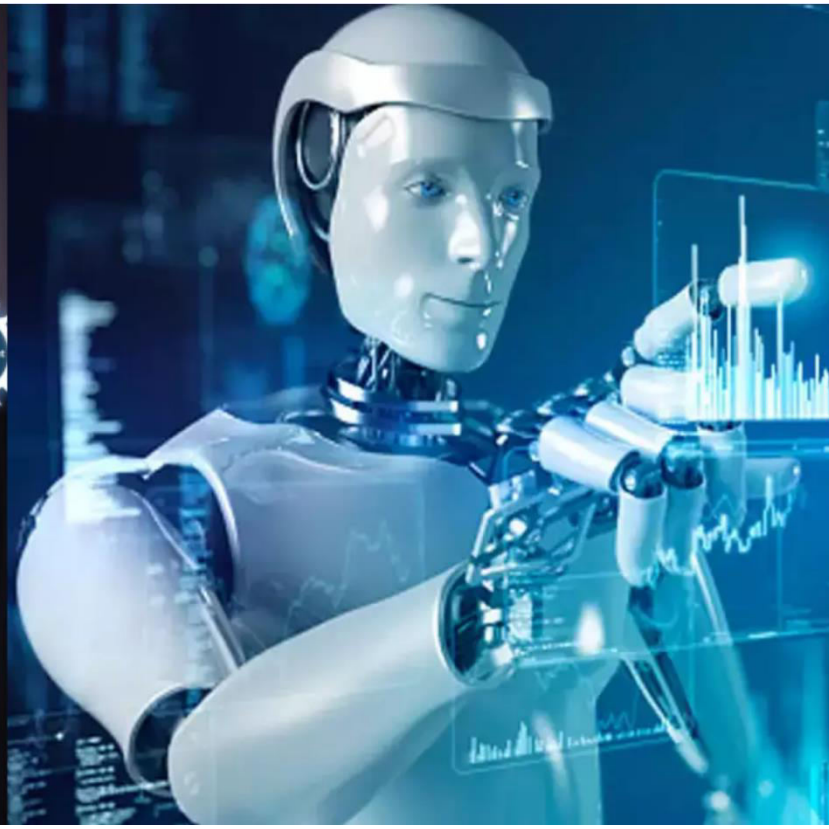
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 2, February 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 2, February 2025

|DOI: 10.15680/IJIRSET.2025.1402091|

Comparison of SQL and NoSQL Databases

Dr.G.A. Pethunachiyar, Ms.Harini Durai Samy

Assistant Professor, Department of Computer Science, School of Excellence in Law, The Tamil Nadu Dr.Ambedkar Law University, Chennai, Tamil Nadu, India

B.C.A.LLB (Hons), School of Excellence in Law, The Tamil Nadu Dr.Ambedkar Law University, Chennai, Tamil Nadu, India

ABSTRACT: The emergence of diverse database technologies has marked the evolution of data. Structured Query Language(SQL) databases have long been the cornerstone of transactional systems, offering a robust relational structure and ensuring data integrity through ACID (Atomicity, Consistency, Isolation, Durability) properties. However, the rise of Big Data and the need for scalable, flexible solutions have led to the growing adoption of NoSQL (Not Only SQL) databases. These databases, characterized by their schema-less design and eventual consistency, offer advantages in handling unstructured and semi-structured data across distributed systems. This paper explores the key distinctions between SQL and NoSQL databases, analyzing their strengths and limitations. The research adopts a comparative analysis approach, examining the architectural differences, data models, and operational characteristics of SQL and NoSQL databases. The study draws on existing literature, case studies, and practical examples to illustrate how each database type performs under various conditions. The analysis includes a review of use cases where SQL databases are traditionally preferred, such as in environments requiring high transactional consistency, and scenarios where NoSQL databases excel, particularly in handling large-scale, distributed data with flexible schemas. The findings suggest that the choice between SQL and NoSQL databases is highly context-dependent. SQL databases remain superior for applications requiring strong data consistency and complex queries, while NoSQL databases offer unparalleled scalability and flexibility for handling unstructured data in distributed systems. The paper concludes by providing guidelines for selecting the appropriate database technology based on specific project needs, including data complexity, performance requirements, and future scalability considerations.

KEYWORDS: SQL, NoSQL, Data management, Scalability, Data consistency

I. INTRODUCTION

Data has emerged as one of the most valuable assets for organizations across various sectors in the contemporary digital landscape. With the exponential growth of data generated from diverse sources such as social media interactions, e-commerce transactions, Internet of Things (IoT) devices, and more effective data management has become crucial for businesses striving to make informed decisions, enhance customer experiences, and maintain a competitive edge. According to estimates by IDC, the global data sphere is expected to reach 175 zettabytes by 2025, highlighting the urgent need for robust data management strategies that can handle this deluge of information. At the heart of effective data management lies the database—a structured system designed to store, retrieve, and manipulate data efficiently. Databases serve as the backbone of modern applications, enabling organizations to manage vast amounts of information while ensuring accessibility and reliability. Traditionally, relational database management systems (RDBMS), characterized by their structured approach to data storage and retrieval using SQL (Structured Query Language), have dominated.

However, as the volume and variety of data have increased, traditional SQL databases have faced significant challenges. The rise of data characterized by high velocity, volume, and variety has necessitated more flexible and scalable solutions capable of handling unstructured or semi-structured data. This shift has led to the emergence of NoSQL databases, which offer a different approach to data management. Unlike their SQL counterparts, NoSQL databases are designed to accommodate diverse data types without rigid schemas, allowing for greater flexibility in how data is stored and accessed.

The growing adoption of NoSQL databases reflects a fundamental change in how organizations approach data management. These databases are particularly well-suited for applications that require rapid scaling and adaptability in

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 2, February 2025

|DOI: 10.15680/IJIRSET.2025.1402091|

response to changing business needs. By leveraging schema-less designs and eventual consistency models, NoSQL databases enable organizations to manage large-scale distributed systems effectively.

This research paper aims to conduct a comprehensive analysis of SQL and NoSQL databases by exploring their key distinctions in terms of architecture, data models, operational characteristics, strengths, limitations, and suitable use cases. By adopting a comparative analysis approach, this study seeks to illuminate the contexts in which each database type excels or falters.The paper will begin with an exploration of the historical context that led to the development of both SQL and NoSQL databases. It will then delve into their fundamental differences concerning data structure, scalability options, consistency models, and query languages. Following this comparative analysis, real-world case studies will be presented to illustrate practical applications of both database types in various industries.

The findings from this research will underscore that the choice between SQL and NoSQL databases is highly context-dependent. While SQL databases remain superior for applications requiring strong transactional consistency and complex querying capabilities, NoSQL databases offer unparalleled scalability and flexibility for handling unstructured data across distributed environments.

II. BACKGROUND

Historical Context of Data Management

Data management has evolved significantly over the past several decades, driven by the increasing complexity and volume of data generated by organizations. In the early days of computing, data was often stored in flat files, which were simple text files containing records that lacked a structured format. This approach posed significant challenges in terms of data retrieval, manipulation, and integrity. As businesses recognized the need for more efficient data handling, hierarchical and network database models emerged in the 1960s and 1970s. These models allowed for more complex relationships between data entities but still lacked the flexibility and ease of use required for modern applications .The introduction of the relational model by Edgar F. Codd in 1970 marked a pivotal moment in data management history. Codd's model utilized tables to represent data and defined relationships between these tables through keys, allowing for more sophisticated querying capabilities using Structured Query Language (SQL) . This innovation laid the groundwork for relational database management systems (RDBMS), which became the dominant technology for managing structured data throughout the late 20th century. RDBMS solutions such as Oracle, Microsoft SQL Server, and MySQL gained widespread adoption due to their ability to ensure data integrity through ACID properties.

Overview of SQL Databases

SQL databases are characterized by their structured approach to data storage and retrieval. They operate on a fixed schema that defines the organization of data within tables, including the type of data each column can hold. This rigid structure enforces consistency and integrity, making SQL databases particularly well-suited for applications requiring high levels of transactional accuracy, such as banking systems and enterprise resource planning (ERP) software .One of the primary advantages of SQL databases is their powerful querying capabilities. SQL allows users to perform complex queries involving multiple tables through operations such as joins, subqueries, and aggregations. This functionality enables organizations to extract meaningful insights from their data efficiently. Furthermore, SQL databases provide robust security features, including user authentication and access control mechanisms, ensuring that sensitive information is protected from unauthorized access. Despite their strengths, SQL databases face limitations when it comes to scalability and flexibility. As organizations increasingly deal with large volumes of unstructured or semi-structured data such as social media content or sensor data traditional RDBMS solutions struggle to adapt to these changing requirements.

Emergence of NoSQL Databases

The limitations of SQL databases became particularly apparent with the advent of Big Data in the late 2000s. As organizations began generating vast amounts of unstructured data at unprecedented rates, the need for more flexible and scalable database solutions became critical This shift led to the emergence of NoSQL which offer a fundamentally different approach to data management compared to their relational counterparts. NoSQL databases are designed to accommodate a variety of data types without rigid schemas, allowing for greater flexibility in how data is stored and accessed. They can be categorized into several types based on their underlying architecture. MongoDB stores data in JSON like documents that can have varying structures, Redis uses a simple key value pair mechanism for storing data, Cassandra organizes data into columns rather than rows, optimizing read and write performance and Neo4 focuses on

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 2, February 2025

|DOI: 10.15680/IJIRSET.2025.1402091|

relationships between entities, making them ideal for applications involving complex networks such as social media analysis.

NoSQL databases prioritize scalability by enabling horizontal scaling by adding more servers or nodes to distribute the load effectively rather than relying solely on vertical scaling . This architecture allows organizations to handle massive datasets efficiently while maintaining high availability. These systems are particularly well-suited for applications that require rapid scaling and schema-less designs and eventual consistency models,

III. SQL VS NOSQL DATABASES

The choice between SQL and NoSQL databases is crucial for developers and organizations, based on performance, scalability, and the ability to manage various data types. Understanding the key differences between these two database technologies is essential for making suitable databases for particular applications. The selection of databases for the application is pivotal for organizations as it directly influences performance, scalability, and the ability to handle various data types. In this comparison , Feature based is depicted in Table 1 and Use cases are depicted in Table 2 .

Table 1 : Features based comparison of SQL and NoSQL Databases

Features	SQL Databases	No SQL Databases
Data Structures	Data is organized into rows and columns with a predefined schema, ensuring consistency across the database. Applications with stable and predictable data requirements, such as financial systems or customer relationship management platforms, can use SQL for better performance.	It uses document-oriented, key-value pairs, column family stores, or graph structures. It allows for dynamic schemas, and structured, semi-structured, and unstructured data, makes ideal for applications where data formats may change frequently or are not known in advance.
Scalability	It employs vertical scaling that adds required resources (CPU, RAM, storage) to a single server to handle increased loads.	It uses horizontal scaling that allows organizations to add more servers or nodes to distribute the load effectively across multiple machines. This architecture enables NoSQL systems to handle large volumes of data and high traffic loads more efficiently, making them suitable for applications requiring rapid scalability
Consistency Model	It adheres strictly to ACID properties to ensure reliable transaction failures or concurrent access. This strong consistency model is crucial for applications where maintaining accurate records is paramount.	It adheres CAP theorem (Consistency, Availability, Partition tolerance), which emphasizes availability and partition tolerance over immediate consistency. NoSQL systems may allow temporary inconsistencies in data during high-load scenarios or network partitions, they prioritize system availability and responsiveness.
Querying Language	SQL is the primary query language to interact with the database. Querying complex datasets through joins, subqueries, and aggregations. This capability allows users to extract meaningful insights from structured data efficiently	varying in query languages depending on their architecture. document-oriented databases like MongoDB use JSON-like syntax for queries, while graph databases like Neo4j utilize graph traversal languages.
Schema Flexibility	SQL databases require a predefined schema that must be established before any data can be inserted into the tables	Databases embrace dynamic schemas that allow developers to adapt their data models on the fly without extensive reconfiguration.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 2, February 2025

|DOI: 10.15680/IJIRSET.2025.1402091|

Speed and Efficiency	SQL databases can experience performance bottlenecks as they grow larger due to the need for joins between tables in complex queries. As the number of records increases, query execution times may rise significantly	NoSQL databases often provide faster read/write operations because they do not require joins; instead, they can retrieve data from a single document or key-value pair directly .
Data Integrity and Security	SQL databases excel in maintaining data integrity through strict adherence to ACID properties, ensuring reliable transactions even during failures or concurrent access scenarios. This makes them ideal for applications where accuracy is critical	Most of the NoSQL systems prioritize availability over immediate consistency; while they can handle high volumes of requests efficiently, they may allow temporary inconsistencies in the data during peak loads .
Handling Unstructured Data	It follows structured data structures	With the increasing prevalence of unstructured data such as images, videos, and social media post .SQL databases offer significant advantages due to their ability to store diverse formats without rigid schemas.

Table 2: Use Cases of SQL and No SQL Databases

Use Cases of SQL Data Bases	Use Cases No SQL Databases
Financial Applications:SQL databases are often employed in banking and financial services where strict consistency is paramount. Transactions involving monetary values must be processed reliably to prevent errors that could lead to significant financial losses. Customer Relationship Management (CRM) Systems: Many organizations use SQL databases to manage customer data effectively. These systems typically involve complex relationships between entities such as customers, orders, and products. The ability to perform complex queries is essential for generating reports and insights that drive marketing strategies and customer engagement. Enterprise Resource Planning (ERP): ERP systems integrate various business processes into a unified system, often using SQL databases to manage inventory, procurement, human resources, and financials. The structured nature of SQL databases allows for efficient data retrieval and reporting across different departments .	Real-Time Big Data Analytics: Organizations that require quick access to large volumes of data often turn to NoSQL databases for their ability to scale horizontally. Organization involved in real-time analytics such as monitoring user behaviour on websites that benefit from the speed and flexibility of NoSQL solutions like Apache Cassandra or MongoDB . Social Media Applications: Social media platforms generate vast amounts of unstructured data from user interactions, posts, comments, and multimedia content. NoSQL databases can efficiently store this diverse data without requiring a fixed schema, allowing for rapid development and iteration of features . Content Management Systems (CMS): Many content-heavy applications leverage NoSQL databases due to their ability to handle various content types (text, images, videos) without a predefined structure. Document-oriented NoSQL databases like CouchDB allow developers to store content in formats such as JSON or XML, making it easier to manage dynamic content .

IV. PERFORMANCE METRICS ANALYSIS OF SQL AND NOSQL DATABASES

Performance metrics are essential for evaluating the efficiency and effectiveness of database systems, whether SQL or NoSQL. Monitoring these metrics allows organizations to identify bottlenecks, optimize resource usage, and ensure that applications perform as expected.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 2, February 2025

|DOI: 10.15680/IJIRSET.2025.1402091|

Performance Metrics for SQL Databases

- Batch Requests per Second: This metric measures the total number of T-SQL batches received by the SQL Server per second. A sudden increase in batch requests can indicate a spike in user activity or potential performance issues. Monitoring this metric helps database administrators understand overall usage trends and detect anomalies in application behaviour .
- Transactions per Second: This metric reflects the total number of transactions executed across all databases on the server. It is crucial for assessing the workload on the database system. A high number of transactions per second indicates heavy usage, but it should be analysed alongside batch requests to determine if transactions are being efficiently processed .
- Read and Write Latency: These metrics measure the average time taken for read and write operations on the database files. High latency can signal performance issues with the underlying storage system or indicate that the database is experiencing I/O bottlenecks. Monitoring read and write latency is vital for maintaining optimal performance, especially in high-transaction environments .
- Page Life Expectancy: This metric indicates how long a data page remains in memory before being flushed to disk. A low page life expectancy (typically below 600 seconds) suggests memory pressure, which can lead to increased I/O operations and degraded performance. Keeping this value high is essential for ensuring efficient data retrieval from memory rather than disk .
- Cache Hit Ratio: The cache hit ratio measures how often requests for data are satisfied from the buffer cache versus requiring a disk read. A high cache hit ratio indicates that the database is effectively utilizing memory resources, resulting in faster query response times. Conversely, a low cache hit ratio may necessitate an increase in memory allocation .
- Blocked Processes: This metric tracks the number of processes that are currently blocked while waiting for resources to become available. High levels of blocked processes can lead to significant performance degradation and indicate issues with query design or resource contention within the database .
- Memory Grants Pending: This metric reflects the number of processes waiting for memory grants to execute queries. Ideally, this value should be zero; however, if it exceeds zero, it indicates memory pressure that could impact query performance and responsiveness .
- Compilations and Recompilations: These metrics track how often SQL Server compiles or recompiles execution plans for T-SQL statements per second. A high number of recompilations can indicate inefficient query design or parameter sniffing issues, leading to unnecessary overhead on the server .

Performance Metrics for NoSQL Databases

While SQL databases have specific performance metrics that are critical for monitoring, NoSQL databases also require attention to different aspects of performance:

- Throughput: This refers to the number of operations (reads/writes) completed in a given time frame and is crucial for assessing how well a NoSQL database can handle varying loads during peak usage periods .
- Latency: Measuring read and write latency is essential for understanding how quickly data can be retrieved or stored in NoSQL databases. High latency can affect user experience significantly, particularly in applications requiring real-time data access .
- Data Size and Growth Rate: Monitoring the total size of data stored and its growth rate helps organizations plan for future capacity needs and ensure that their infrastructure can support increasing volumes of unstructured data .
- Error Rates: Tracking errors during read/write operations provides insights into potential issues within the database system or application logic that may need addressing to maintain optimal performance.
- Replication Lag: In distributed NoSQL systems, replication lag measures the delay between when data is written to one node and when it becomes available on replicas. Minimizing replication lag is crucial for ensuring consistency across distributed systems.

V. FUTURE ENHANCEMENTS

The future of database technologies is being shaped by several emerging trends that redefine how data is stored, processed, and utilized. Organizations are increasingly seeking solutions that not only meet their current needs but

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 2, February 2025

|DOI: 10.15680/IJIRSET.2025.1402091|

anticipate future demands for scalability, flexibility, and performance. This section explores key trends influencing Both SQL and NoSQL databases, providing insights into how these databases are expected to develop in the future.

- Multi-Model Databases: One significant trend is the rise of multi-model databases, which combine the capabilities of both SQL and NoSQL systems into a single platform. These databases allow organizations to use different data models such as relational, document, graph, and key-value within the same application. This flexibility enables developers to choose the most appropriate model for each use case without being locked into a single database type. As businesses increasingly deal with diverse data types and structures, multi-model databases are expected to gain popularity for their ability to streamline data management processes.
- Cloud-Based Database Solutions: The shift toward cloud computing is another critical trend impacting database technologies. Cloud-based databases offer scalability, cost-efficiency, and ease of management that on-premises solutions often cannot match. As organizations migrate their operations to the cloud, they are adopting Database-as-a-Service (DBaaS) models that provide managed database solutions without the need for extensive infrastructure investment. Major cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform are continually enhancing their database offerings, making it easier for businesses to deploy and scale applications rapidly.
- Serverless Architectures:Serverless computing is gaining traction as a way to simplify application development and deployment. In a serverless architecture, developers can focus on writing code without managing the underlying infrastructure. This trend extends to databases as well, with serverless database solutions allowing automatic scaling based on demand and eliminating the need for manual provisioning of resources . This approach not only reduces operational overhead but also enhances agility, enabling organizations to respond quickly to changing business requirements.
- Artificial Intelligence and Machine Learning Integration:The integration of artificial intelligence (AI) and machine learning (ML) into database technologies is transforming how data is managed and utilized. AI-driven analytics tools can automatically optimize query performance, predict resource needs, and identify anomalies in data patterns . As organizations seek to leverage their data for competitive advantage, AI-enhanced databases will become increasingly important for automating routine tasks and providing deeper insights into data trends.
- Enhanced Security Features: With rising concerns about data breaches and cyber threats, enhanced security features are becoming a priority in database technologies. Both SQL and NoSQL databases are evolving to include advanced encryption and audits. As regulatory compliance becomes more stringent across industries, organizations will increasingly rely on databases that offer robust security capabilities to protect sensitive information.
- Focus on Performance Optimization:As data volumes continue to grow exponentially, performance optimization remains a critical focus for both SQL and NoSQL databases. Techniques such as indexing improvements, query optimization algorithms, and caching strategies will be essential for ensuring fast response times and efficient resource utilization . Additionally, emerging technologies like in-memory databases are being adopted to enhance performance by storing data in RAM rather than on traditional disk storage.

VI.CONCLUSION

This research paper presented an in-depth comparison of SQL and NoSQL databases by highlighting their distinct characteristics, strengths, and limitations. The evolution of data management technologies has led to the emergence of these two paradigms, each catering to different needs .SQL databases have long been the cornerstone of transactional systems, offering a robust relational structure that ensures data integrity through ACID properties . They excel in environments requiring high levels of data consistency and complex queries, making them ideal for applications in finance, healthcare, and enterprise resource planning . However, their rigid schema and reliance on vertical scaling can pose challenges in handling large volumes of unstructured data and adapting to rapidly changing business requirements .NoSQL databases have emerged as a powerful alternative designed to address the limitations of traditional relational databases. Their schema-less design and eventual consistency models provide unparalleled flexibility and scalability, making them well-suited for applications dealing with Big Data and unstructured or semi-structured information . Use cases in social media, e-commerce, and real-time analytics demonstrate how NoSQL databases can effectively manage diverse data types while supporting rapid growth and dynamic workloads .Both SQL and NoSQL databases offer unique advantages tailored to different applications within the realm of data management.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 2, February 2025

|DOI: 10.15680/IJIRSET.2025.1402091|

REFERENCES

[1] IDC. “The Digitization of Edge to Core.” IDC White Paper, 2018.
[2] Liu, Z.H., Hammerschmidt, B., McMahon, D., Liu, Y., & Chang, H.J. SQL and NoSQL Database Software Architecture Performance Analysis. MDPI. 2022.
[3] Stonebraker, M. SQL databases v. NoSQL databases. Communications of the ACM, 53(10), 10-11.,2010.
[4] Meier, A., & Kaufmann, M. SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management. Springer Vieweg: Wiesbaden, Germany ,2019.
[5] Dhasade, S.D. A Study of NoSQL Database. International Research Journal of Modernization in Engineering Technology and Science, 4(10), 686-690,2022.
[6] Chandra, D.G. BASE analysis of NoSQL database. Future Generation Computer Systems, 52, 13–21. 2015.
[7] Gonzalez-Aparicio, M.T., Younas, M., Tuya, J., & Casado, R. Evaluation of ACE properties of traditional SQL and NoSQL big data systems. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 1988–1995). Limassol, Cyprus. 2019.
[8] Date, C.J. An Introduction to Database Systems. 8th ed. Boston: Pearson Education Limited, 2003.
[9] Sadalage, Pramod J., and Martin Fowler. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Boston: Addison-Wesley Professional Computing Series, 2012.
[10] Stonebraker, Michael et al. “The End of an Architectural Era: (It’s Time for a Complete Rewrite).” Proceedings of the 33rd International Conference on Very Large Data Bases, 2007: 1150-1160.
[11] Codd, E.F. “A Relational Model of Data for Large Shared Data Banks.” Communications of the ACM 13, no. 6 (1970): 377-387.
[12] Elmasri, Ramez, and Shamkant B. Navathe. Fundamentals of Database Systems. 7th ed. Boston: Pearson Education Limited, 2016.
[13] Silberschatz, Abraham et al. Database System Concepts. 6th ed. New York: McGraw-Hill Education, 2011.
[14] Garcia-Molina, Hector et al. Database Systems: The Complete Book. Upper Saddle River: Prentice Hall, 2008.
[15] Chen, M., Mao, S., and Liu, Y. “Big Data: A New Frontier for Innovation and Research.” IEEE Access 2 (2014): 1148-1160.
[16] Robinson, Ivan et al. Graph Databases. Sebastopol: O'Reilly Media Inc., 2015.
[17] Hwang, Kai et al. “Distributed and Cloud Computing: From Parallel Processing to the Cloud.” Journal of Parallel and Distributed Computing 74 (2014): 1-16.
[18] Gupta, Anil et al. “Artificial Intelligence in Databases: A Review.” Journal of Computer Science 16 (2020): 123-134.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details