

Analysis on Utilizing MySQL to Enhance Scalability, Data Security, and Reliability in Water Level Monitoring Systems

Mark Denver P. Adora

Pangasinan State University, Urdaneta City Campus

Urdaneta City, Pangasinan, Philippines

Article Info:

Received: 05 June 2025; Revised: 19 Aug 2025; Accepted: 21 Sept 2025; Available Online: 17 Dec 2025

Abstract - Water level monitoring systems execute essential functions in both flood protection and resource administration and environmental safety objectives. Real-time data collection through Internet of Things (IoT) technologies has transformed but the maintenance of databases remains a crucial obstacle mainly because it affects scalability and security alongside reliability. The structural query features and data security capabilities of MySQL make it a popular relational database which measures up well with ACID compliance despite needing additional evaluation regarding its performance capacity when handling high-frequency IoT applications. The research investigates MySQL's tools to scale up and secure and make reliable water level monitoring systems based on IoT technology through detailed analysis. The authors perform a database evaluation between MySQL, PostgreSQL and NoSQL to understand their performance in data collection and query management and real-time failover capabilities. It also shows MySQL achieves solid encryption standards and data integrity but its performance declines during high-speed data processing operations and horizontal clustering procedures cause slow failover responses. The enhancement of MySQL performance relies on essential best practices that consist of optimization for queries as well as partitioning techniques and replication management and combinations of hybrid database structures with NoSQL solutions. The final analysis shows MySQL proves effective as a structure data solution in IoT monitoring systems but integration between SQL and NoSQL systems creates optimal combinations of stability and operation speed. Future investigation should aim to maximize MySQL performance for real-time IoT applications and create AI-based security frameworks which boost reliability in vital monitoring systems.

Keywords - MySQL, IoT, Water Level Monitoring, Scalability, Data Security, Reliability, Hybrid Database Architecture.

INTRODUCTION

Monitoring of water levels is essential for resource management operations as well as flood control and safety maintenance of public infrastructure (Jan et al., 2021; Fairuz et al., 2020). Traditional monitoring methods lack real-time data capabilities so addressing rapid water level fluctuations becomes a limitation (Olisa et al., 2021). Current advancements on the Internet of Things technology enable better collection and

processing of real-time data which creates novel prospects to transform water level monitoring systems (Leal Sobral et al., 2023)., Growing data from these systems requires database frameworks which support expanding data volumes and demonstrate efficient operation when working with heavy loads (Győrödi et al., 2021). Recent study highlighted that MySQL database imposes a challenge in handling new volumes of data

generated by an IoT application and requires the emergence of NoSQL database (Mladenova & Valova, 2022). However, a study stated that as the number of data increases, MySQL responses were more stable compared to NoSQL (Rautmare & Bhalerao, 2016). In addition, several studies concluded that both MySQL and NoSQL have their own strengths depending on what query is frequently used which also affects real-time data (Rautmare & Bhalerao, 2016; Reetishwaree & Hurbungs, 2020). Replication together with clustering and partitioning capabilities enables MySQL to work efficiently with big data sets (Liu & Wu, 2024; Zmaranda et al., 2021). Furthermore, a project also showed that MySQL is suitable for IoT applications integrated in water level monitoring by which it is efficient to store and manage sensor data that facilitates seamless interaction between all system components, ensuring data accuracy and real-time updates (GB Logical, 2025).

As highlighted by (WRc, n.d.; Overmars & Venkatraman, 2020) that in real-world water monitoring systems, scalability, security, and reliability are critical where it can lead to significant inefficiencies, compromising public health, environmental protection, and sustainable water management when failures occurred (Al Faris, 2025). The proven scalability and reliability feature of MySQL established the system as an effective solution for web-based and Internet of Things applications (Yadav et al., 2024). Furthermore, limitations in data protection grow into a substantial challenge for water observation frameworks because security breaches can compromise management choices and lead to heightened risk situations during emergency situations (Adelani et al., 2024). The security platform offered by MySQL includes features that solve these challenges with encryption technology and role-based access management and secure data transmission protocols (Zhao et al., 2022).

Another essential aspect of reliable water level monitoring systems is data integrity and accountability (Vani et al., 2024). MySQL Transparent Data Encryption can effectively secure a MySQL database with minimal impact on performance (Terencio et al., 2023). MySQL auditing mechanisms established trusted data management by tracking complete user operations log as well as changes to restore compliance with standards and confirm system visibility (Győrödi et al., 2021). Current system implementations experience difficulties when attempting to achieve quality trade-offs combining system expansion with information protection together with reliability across real-time operational contexts (Anugrah et al., 2018).

Existing studies highlight the importance of real-time data capabilities in water level monitoring systems, but there is limited research on how its data storage needed to be designed to provide accurate, secured, and reliable data (Terencio et al., 2023; Jan et al., 2022). To conclude, this research will evaluate how MySQL will be implemented to strengthen the scalability along with the data security features and reliability function of water level monitoring systems. This also contributes to the exploration of effective strategies in designing MySQL databases to achieve optimal performance and scalability of water level monitoring systems.

OBJECTIVES OF THE STUDY

This study was conducted to analyze the utilization of MySQL in enhancing scalability, data security, and reliability in IoT-based water level monitoring systems. Specifically, the study sought answers to the following:

1. What is the comparative performance of MySQL, PostgreSQL, and NoSQL databases in terms of scalability, data



security, and reliability within IoT-based water level monitoring applications?

2. How can MySQL be optimized to enhance scalability, data security, and reliability in IoT-based water level monitoring systems?

MATERIALS AND METHOD

2.1 Research Design

This study has been undertaken as a systematic literature review based on the framework introduced by (Mengist et al., 2021), which proposes identifying, evaluating, and interpreting all relevant unpublished, and published literature to answer a focused and clearly stated research question (DistillerSR Inc., 2022). The research aims at examining published studies regarding MySQL’s ability to boost scalability data security reliability in IoT-based water level monitoring systems while finding optimal performance practices. Analysis of MySQL includes a comparison to PostgreSQL and MongoDB to determine its performance features along with its limitations within the framework of real-time IoT structures. The approach delivers a standardized research method to consolidate information from multiple sources which results in a complete evaluation of MySQL's performance in water level monitoring systems. The steps in the systematic literature review method are documented below.

In this systematic review, the framework was introduced by (Mengist et al., 2021) identified relevant concepts through the PICOC criteria (Population, Intervention, Comparison, Outcomes, and Context). Table 1 presents the PICOC criterion applied to the study.

Table 1. PICOC Criteria

Criteria	Description
Population	<i>IoT-based water level monitoring systems utilizing relational databases for data management.</i>
Intervention	<i>Implementation of MySQL optimization techniques, including indexing, partitioning, replication, query caching, and security enhancements.</i>
Comparison	<i>Performance evaluation of MySQL against other database systems such as PostgreSQL, MongoDB, and InfluxDB in terms of scalability, security, reliability, and data processing efficiency.</i>
Outcomes	<i>Improved scalability, data security, and reliability of MySQL in real-time IoT monitoring applications.</i>
Context	<i>Application in IoT-driven environmental monitoring systems, particularly in flood detection, resource management, and water level data analysis.</i>

2.2 Literature Review

A comprehensive literature review provides the necessary background about MySQL’s database system for IoT deployments by assessing its performance with big data workloads and data ingestion capabilities and security protocols for protecting sensitive water level data. This review was conducted to explore previous studies on:

Database Systems in IoT: The role of relational databases like MySQL in managing sensor data, particularly in water level monitoring systems.

Scalability and Reliability: How MySQL handles large datasets and ensures high availability in real-time monitoring systems.

Data Security: The security features of MySQL, including encryption, access control, and data integrity mechanisms, and their effectiveness in protecting sensitive water level data.

2.3 Data Collection Methods

Data for this research was collected from:

Academic Journals and Industry Reports:
To gain technical insights into the use of MySQL in IoT-based water level monitoring systems.

Case Studies: Real-world implementations of MySQL in water level monitoring systems and other IoT applications were analyzed to evaluate its performance in terms of scalability, reliability, and security.

Technical Documentation: MySQL documentation and performance benchmarks were reviewed to understand its capabilities and limitations.

2.4 Analytical Techniques

The research relies mainly on two evaluation methods which include comparative analysis and performance metrics evaluation. The study evaluates database systems by comparing MySQL to PostgreSQL and MongoDB and other alternatives to assess performance metrics including query response time and both data throughput, and uptime. The organized evaluation methodology produces clear knowledge about MySQL capacities and weaknesses in IoT water level monitoring which helps identify methods to maximize its operational efficiency.

RESULTS AND DISCUSSION

3.1 Literature Review

Recent studies highlight MySQL acquiring wider utilization for use in IoT applications due to its capabilities in structured data management, security aspect, and its support of ACID (Atomicity, Consistency, Isolation, Durability) compliance (Ganachari & Lakshmanasamy, 2021). A comparative study compared the MySQL database performance with NoSQL databases in IoT applications and found that the MySQL database has been performing consistently in data storage, but it has problems in high insertion rate applications within timely data environment (Rautmare & Bhalerao, 2016; (Eyada et al., 2020). In addition, a study also tested MySQL's performance in handling IoT-generated time-series data and discovered that query optimization techniques like indexing and partitioning can improve MySQL (Ganachari & Lakshmanasamy, 2021; Salo, n.d.). Furthermore, a study highlights the realization of centralized data management, calculation and analysis, charting and other functions based on relational database MySQL and B/S (Browser/Server) Architecture. In lieu of this, the application shows stability, reliability and expansibility which improves water level monitoring efficiency (Zhao et al., 2022). Moreover, a book stated MySQL supports replication and failover mechanisms which enhance data availability, but a study indicates failover recovery times in MySQL are slower compared to NoSQL databases, which prioritize distributed storage and eventual consistency models (Rautmare & Bhalerao, 2016; Bell, 2016). A study suggests using MySQL's replication strategies effectively, also combining with a layered security architecture comprising physical, network and application levels covered in security frameworks, with AI influenced threat discovering frameworks continually seeing for abnormal movements, refining security parameters, and

isolating the compromised device to improve the performance of this MySQL in real time IoT applications (Asonze et al., 2024).

In general, the wide adoption of MySQL for IoT applications continues because it offers data consistency and security features in addition to its structured query functionality. The database proves useful among IoT applications, yet its scaling ability still needs better design together with enhanced performance under heavy data flows and the need for native time-series optimization indicating the future potential of hybrid database systems which combine MySQL with NoSQL and time-series databases to optimize

execution-speed and scalability and security measures in IoT monitoring systems.

3.2 Comparative Analysis of Database Systems

The table below provides a comparative analysis of various database systems such as MySQL, PostgreSQL, and NoSQL database systems based on their scalability, security, reliability, and performance in IoT-based water level monitoring applications. This comparison highlights their strengths and limitations, offering insights into their suitability for different IoT use cases.

Table 2. Comparative Analysis of Database Systems

Database System	Scalability	Security	Reliability	Performance in IoT Applications
MySQL	Moderate (Limited Horizontal Scaling) (Ganachari & Lakshmanasamy, 2021; Salo, n.d.; Bell, 2016; Asiminidis et al., 2019; Kontogiannis et al., 2019)	Strong (Encryption, Access Control) but SQL Injection Risk (Rautmare & Bhalerao, 2016; Terencio et al., 2023; Bell, 2016; Editor, 2023; Buuck, 2022)	High (Replication & Clustering) but Slow Failover (Rautmare & Bhalerao, 2016; Bell, 2016; Asiminidis et al., 2019; Kontogiannis et al., 2019)	Moderate (Slower Write Speeds in High-Frequency Data Ingestion) (Asiminidis et al., 2019; Kontogiannis et al., 2019)
PostgreSQL	High (Supports JSON Data Type) (Asiminidis et al., 2019; Kontogiannis et al., 2019)	Strong (Advanced Authentication & Role-Based Access) (Kontogiannis et al., 2019; Editor, 2023; Buuck, 2022)	High (Better High Availability Features) (Asiminidis et al., 2019)	High (Efficient Query Execution for Complex Data) (Asiminidis et al., 2019; Kontogiannis et al., 2019)
MongoDB (NoSQL)	Very High (Optimized for IoT Time-Series Data, Supports JSON Data Type) (Asiminidis et	Moderate (Limited Role-Based Access) (Rautmare & Bhalerao, 2016)	High (Auto-Sharding, Eventual Consistency) (Rautmare &	Very High (Fast Ingestion and Retrieval) (Asiminidis et al.,



al., 2019;
Kontogiannis et al.,
2019)

Bhalerao, 2019; Kontogiannis
et al., 2019)
2016;
Asiminidis et
al., 2019)

The comparative analysis of SQL and NoSQL databases for IoT-based water level monitoring systems highlights key differences in terms of scalability, security, reliability, and performance. MySQL is very popular relational database due to its structured data management, ACID compliance for handling transactions and enhanced security. Findings however show a limitation of MySQL's scalability especially in horizontal scaling making it uncapacious to handle increasing sensor data efficiently. MySQL provides good encryption and access control mechanisms with caution but is still susceptible to SQL injections and therefore makes MySQL applications security issues in IoT applications. MySQL's replication and clustering mechanisms provide data availability but delays of fail over time affect real time responsiveness in water level monitoring systems. Moreover, high-frequency data ingestion presents challenges for MySQL, as it exhibits slower write speeds compared to NoSQL alternatives.

To address these limitations, several best practices have been identified to optimize MySQL's performance in IoT applications. Optimizing indexing strategies and using partitioning can improve query performance by reducing search time and optimizing read/writing operations in high-frequency sensor data environments (Ganachari & Lakshmanasamy, 2021; Salo, n.d.). Utilizing replication and sharding techniques enhances scalability; master-slave replication distributes read-heavy workloads, while sharding allows MySQL to

manage larger datasets efficiently, benefiting IoT-based monitoring systems (Bell, 2016; Asiminidis et al., 2019). Enhancing security measures, such as implementing parameterized queries to prevent SQL injections, using SSL/TLS encryption for secure data transmission, and carefully configuring user access privileges, can mitigate vulnerabilities in MySQL-based IoT applications (Bell, 2016; Asonze et al., 2024; Abdullah et al., 2023). In addition, adjusting MySQL's InnoDB buffer pool size, increasing query cache size, and fine-tuning memory allocation parameters can significantly enhance performance under heavy data loads (Eyada et al., 2020); Kontogiannis et al., 2019; Sumpter, 2025; Agabekov, 2024).

3.3 Performance Metrics Analysis

The evaluation of MySQL performance in IoT water level monitoring systems require a detailed analysis of three main metrics which are query response time and data throughput and system uptime. The performance metrics of MySQL directly influence how efficiently it operates as well as its scalability and reliability when dealing with real-time sensor data. Structured data management functions well with MySQL but the platform encounters difficulties with high-speed data integration and concurrent processing and automatic failover procedures. To answer this question, research studies have identified different performance outcomes for MySQL alongside its main obstacles and successful optimization techniques which are presented in the following table

Table 3. Performance Metrics Analysis of MySQL

Performance Metric	Findings	Challenges	Optimization Strategies	Supporting Studies/Literature
Query Response Time	Efficient for structured queries	Latency issues under high-concurrency and large-scale data ingestion	Implementing indexing, query caching, and partitioning	(Ganachari & Lakshmanasamy, 2021; Salo, n.d.; Kontogiannis et al., 2019)
Data Throughput	Manages moderate workloads effectively	Struggles in high-insertion rate IoT applications	Buffer pool tuning, connection pooling, and query optimization	(Eyada et al., 2020; Kontogiannis et al., 2019; Sumpter, 2025; Agabekov, 2024)
System Uptime & Reliability	Replication and clustering enhance availability	Slow failover recovery compared to NoSQL alternatives	High-availability configurations, automated failover, and optimized replication	(Bell, 2016; Asiminidis et al., 2019; Zafi et al., n.d.)

The performance of MySQL within IoT-based water monitoring systems demonstrates both beneficial characteristics as well as performance challenges. High latency occurs during MySQL queries in systems that experience high query volume and massive data ingestion while indexing and caching methods together with partitioning help improve system performance. In addition, MySQL also demonstrates efficient data throughput when handling moderate workloads, but it requires optimized buffer pools and connection pools as well as optimized queries to perform effectively in high-insertion rate IoT applications. The replication and clustering features of MySQL enhance data availability by addressing reliability issues that arise from slower failover recovery times than NoSQL systems through implementing high-availability configurations and automation of failover mechanisms and optimized replication strategies. The real-time performance efficiency of MySQL in IoT applications can be substantially improved through the

implementation of performance-based optimizations that enhance both scalability and failover management and query execution.

CONCLUSION AND RECOMMENDATION

This research examined MySQL performance together with scalability and reliability and security aspects in IoT-based water level monitoring systems to establish both pros and drawbacks. The research shows that MySQL maintains widespread usage as a relational database through its data structure management and ACID requirements and security measures yet struggles with high-speed data input and scalability limitations and immediate system recovery procedures.

The analysis showed MySQL excels at executing structured queries but exhibits weakness during high-insertion rate IoT operational demands according to studies against PostgreSQL and



MongoDB and other database systems. The main weakness of MySQL lies in its limited abilities to scale horizontally for efficient processing of escalating sensor data loads because its scalability options are less adaptable than NoSQL solutions. Besides improving IoT security with encryption and access control the database still has security issues because of its SQL injection vulnerabilities and absence of built-in anomaly detection features. The reliable data availability achieved through MySQL replication and clustering suffers from slow failover delays that reduce immediate responsiveness in real-time water level monitoring systems.

The optimization of MySQL performance in IoT-based systems requires best practices that include both query indexing and partitioning as well as replication and sharding and buffer pool tuning. The combination of MySQL and NoSQL databases for different data tasks can create hybrid architectures which balance water monitoring applications between high-speed data handling and secure data storage capabilities.

Future studies need to prioritize MySQL performance enhancement regarding high-speed IoT sensor data processing while also developing security systems through machine learning and automated failover methods for improving real-time data delivery within environmental monitoring networks.

REFERENCES

- F. Jan, N. Min-Allah, and D. Düşteğör, "IoT based smart water quality monitoring: Recent techniques, trends and challenges for domestic applications," Jul. 01, 2021, MDPI AG. doi: 10.3390/w13131729.
- M. F. M. Fairuz, M. R. A. Rejab, R. Farook, A. Hanah, N. Rashid, and M. N. M. Rifaat, "Real-Time Flood Monitoring System Using Raspberry PI," in IOP Conference Series: Materials Science and Engineering, IOP Publishing Ltd, Dec. 2020. doi: 10.1088/1757-899X/932/1/012070.
- S. C. Olisa, C. N. Asiegbu, J. E. Olisa, B. O. Ekengwu, A. A. Shittu, and M. C. Eze, "Smart two-tank water quality and level detection system via IoT," *Heliyon*, vol. 7, no. 8, Aug. 2021, doi: 10.1016/j.heliyon.2021.e07651.
- V. A. Leal Sobral et al., "A Cloud-Based Data Storage and Visualization Tool for Smart City IoT: Flood Warning as an Example Application," *Smart Cities*, vol. 6, no. 3, pp. 1416–1434, Jun. 2023, doi: 10.3390/smartcities6030068.
- C. A. Györödi, D. V. Dumșe-Burescu, R. Györödi, D. R. Zmaranda, L. Bandici, and D. E. Popescu, "Performance impact of optimization methods on MySQL document-based and relational databases," *Applied Sciences (Switzerland)*, vol. 11, no. 15, Aug. 2021, doi: 10.3390/app11156794.
- T. Mladenova and I. Valova, "Performance Study of MySQL and MongoDB for IoT Data Processing and Storage," in 2022 International Conference Automatics and Informatics (ICAI), Oct. 2022, pp. 60–63. doi: 10.1109/ICAI55857.2022.9960134.
- S. Rautmare and D. M. Bhalerao, "MySQL and NoSQL database comparison for IoT application," in 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Oct. 2016, pp. 235–238. doi: 10.1109/ICACA.2016.7887957.
- S. Reetishwaree and V. Hurbungs, "Evaluating the performance of SQL and NoSQL databases in an IoT environment," in 2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering, ELECOM 2020 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 229–234. doi: 10.1109/ELECOM49001.2020.9297028.
- Q. Liu and J. Wu, "Research on Agricultural Data Processing Based on MySQL," *Agricultural & Forestry Economics and Management*, vol. 7, no. 2, 2024, doi: 10.23977/agrfem.2024.070203.



- D. R. Zmaranda, C. I. Moisi, C. A. Győrödi, R. Győrödi, and L. Bandici, "An analysis of the performance and configuration features of mysql document store and elasticsearch as an alternative backend in a data replication solution," *Applied Sciences (Switzerland)*, vol. 11, no. 24, Dec. 2021, doi: 10.3390/app112411590.
- GB Logical, "IoT water monitor." Accessed: Feb. 25, 2025. [Online]. Available: <https://gblogical.com/our-work/iot-water-monitor>
- WRc, "The challenges of continuous water quality monitoring."
- A. Overmars and S. Venkatraman, "Towards a Secure and Scalable IoT Infrastructure: A Pilot Deployment for a Smart Water Monitoring System," *Technologies (Basel)*, vol. 8, no. 4, Dec. 2020, doi: 10.3390/technologies8040050.
- M. Al Faris, "Advanced Water Quality Monitoring 2025, How Grinviro's DIAC-X Uses AI and IoT for Real-Time Insights."
- B. Yadav, V. Shrivastava, A. Pandey, and P. Sharma, "MySQL : Database Design for Performance and Scalability," 2024. [Online]. Available: www.ijrpr.com
- F. Adeshina Adelani, E. Stefano Okafor, B. Sonimiteim Jacks, and O. Abayomi Ajala, "THEORETICAL INSIGHTS INTO SECURING REMOTE MONITORING SYSTEMS IN WATER DISTRIBUTION NETWORKS: LESSONS LEARNED FROM AFRICA-US PROJECTS," *Engineering Science & Technology Journal*, vol. 5, no. 3, pp. 995–1007, 2024, doi: 10.51594/estj/v5i3.953.
- D. Zhao, Y. Song, Z. Deng, L. Dai, and G. Pan, "Design of On-line Monitoring System for Foundation Pit Water Level Based on Raspberry Pi," in *2022 8th International Conference on Hydraulic and Civil Engineering: Deep Space Intelligent Development and Utilization Forum (ICHCE)*, Nov. 2022, pp. 406–409. doi: 10.1109/ICHCE57331.2022.10042717.
- S. Vani, P. Joshi, R. Shinde, S. Kadam, and M. Matkamwad, "Smart Water Management Grid," *International Journal of Innovative Science and Research Technology (IJISRT)*, pp. 854–859, May 2024, doi: 10.38124/ijisrt/ijisrt24may931.
- N. Terencio, Wella, and A. Sony, "Enhancing MySQL Database Security with MySQL Enterprise Transparent Data Encryption," *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 4, pp. 154–173, 2023, doi: 10.33168/JLISS.2023.0411.
- W. Anugrah, S. Suryono, and J. E. Suseno, "Real-time Geographic Information System (GIS) for Monitoring the Area of Potential Water Level Using Rule Based System," in *E3S Web of Conferences*, EDP Sciences, Feb. 2018. doi: 10.1051/e3sconf/20183111019.
- F. Jan, N. Min-Allah, S. Saeed, S. Z. Iqbal, and R. Ahmed, "IoT-Based Solutions to Monitor Water Level, Leakage, and Motor Control for Smart Water Tanks," Feb. 01, 2022, MDPI. doi: 10.3390/w14030309.
- G. Ganachari and R. Lakshmanasamy, "EVALUATING DATA MODELING FLEXIBILITY: DYNAMODB'S KEY-VALUE STORE VS MYSQL'S RELATIONAL MODEL," *International Journal of Core Engineering & Management*, no. 6, 2021.
- M. M. Eyada, W. Saber, M. M. El Genidy, and F. Amer, "Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments," *IEEE Access*, vol. 8, pp. 110656–110668, 2020, doi: 10.1109/ACCESS.2020.3002164.
- K. Salo, "Analysis and future development of the sensor data storage of an IoT application."
- D. Zhao, Y. Song, Z. Deng, L. Dai, and G. Pan, "Design of On-line Monitoring System for Foundation Pit Water Level Based on Raspberry Pi," in *2022 8th International Conference on Hydraulic and Civil Engineering: Deep Space Intelligent Development and Utilization Forum*



- (ICHCE), Nov. 2022, pp. 406–409. doi: 10.1109/ICHCE57331.2022.10042717.
- Charles. Bell, MySQL for the internet of things. Apress, 2016.
- C. U. Asonze, O. S. Ogungbemi, F. A. Ezeugwa, A. O. Olisa, O. I. Akinola, and O. O. Olaniyi, “Evaluating the Trade-offs between Wireless Security and Performance in IoT Networks: A Case Study of Web Applications in AI-Driven Home Appliances,” *Journal of Engineering Research and Reports*, vol. 26, no. 8, pp. 411–432, Aug. 2024, doi: 10.9734/jerr/2024/v26i81255.
- C. Asiminidis, G. Kokkonis, and S. Kontogiannis, “Database Systems Performance Evaluation for IoT Applications,” *SSRN Electronic Journal*, Apr. 2019, doi: 10.2139/ssrn.3360886.
- S. Kontogiannis, G. Kokkonis, S. Kontogiannis, and C. Asiminidis, “Comparing Relational and NoSQL Databases for carrying IoT data,” *Article in The Journal of Scientific and Engineering Research*, vol. 6, no. 1, pp. 125–133, 2019, [Online]. Available: <https://www.researchgate.net/publication/331062759>
- A. S. Abdullah, A. S. R, and P. Mohapatra, “Detection and Analysis of Port Scanning and SQL Injection Vulnerabilities with correlating factors in Web Applications to Enhance secure Data Transmission,” in *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, Nov. 2023, pp. 1–5. doi: 10.1109/RMKMATE59243.2023.10368777.
- A. Zafi, D. Saputra, and A. Bianto, “THE MONITORING SYSTEM FOR WATER QUALITY IS BASED ON THE INTERNET OF THINGS (IOT) AND USES A TDS SENSOR”, doi: 10.3804/ijenset.v1i2.1015.
- Editor. (2023, May 16). Database management systems (DBMS) comparison: MySQL, Postgr. AltexSoft. <https://www.altexsoft.com/blog/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>
- Buuck, B. (2022, March 7). PostgreSQL vs MySQL: Cost, performance, security, scale & ease of use | StreamSets. Software AG. https://www.softwareag.com/en_corporate/blog/streamsets/postgresql-vs-mysql.html
- Satori Cyber Ltd. (2023, July 31). Postgres access control. Satori. <https://satoricyber.com/postgres-security/postgres-access-control>
- Sumpter, B. (2025, February 24). MySQL performance tuning: Maximizing database efficiency and speed. Percona Database Performance Blog. <https://www.percona.com/blog/mysql-101-parameters-to-tune-for-mysql-performance>
- Agabekov, R. (2024, January 22). Optimizing MySQL for peak Performance — A comprehensive guide. Medium. <https://medium.com/releem/optimizing-mysql-for-peak-performance-a-comprehensive-guide-881ba1ad3ee1>
- Mengist, W., Soromessa, T., & Legese, G. (2021, September 24). Method for conducting systematic literature review and meta-analysis for environmental science research. <https://www.sciencedirect.com/science/article/pii/S221501611930353X>
- DistillerSR Inc. (2022, December 20). Eligibility criteria in a systematic review. DistillerSR. <https://www.distillersr.com/resources/systematic-literature-reviews/eligibility-criteria-in-a-systematic-review>