IOT DATA MANAGEMENT USING MYSQL: A SCALABLE AND EFFICIENT STORAGE FRAMEWORK

Binu Mol T. V

**Associate Professor, Department of Computer Science, KKTM Government College, Pullut, Kodungallur, Thrissur, Kerala, India Email: binumoltv@gmail.com*

1. Abstract

The growth of Internet of Things (IoT) devices produces enormous real-time data. This leads to difficulty in data storage, scalability and retrieval. Traditional database systems find difficulty in handling large volume of sensor data keeping low-latency performance. This paper proposes an IoT data management framework based on a relational database platform MySQL which is scalable and high performing.

The proposed framework includes optimized database schemas, indexing mechanisms, partitioning approaches and replication methods which can improve system performance and reliability. The study shows that query response time is reduced, storage utilization is increased and more scalability in dealing huge data. The study ahows that an optimized MySQL database can be used as backend solution for IoT applications.

2. Keywords: *IoT, MySQL, Database Management System, Data Scalability, Real-Time Processing, Query Optimization, Data Partitioning*

Received: 13/04/2026 Revised: 25/05/2026 Acceptance: 01/06/2026 Publication: 08/06/2026

3. Introduction

The Internet of Things (IoT) links billions of intelligent devices that continuously produce both structured and semi-structured data. IoT generated data must be managed efficiently as IoT applications need timely data processing and reliable storage. These are essential for decision making and system performance. Some of the applications are smart agriculture, healthcare systems, environmental monitoring and industrial automation.
IoT environments require high data write throughput, low-latency query processing, scalable system architectures, and strong data consistency and they should be capable of efficiently managing and analyzing large volumes of continuously generated sensor data.
Relational database management systems such as MySQL have strong data consistency, structured schema design, and reliable transaction support. By optimizing MySQL it can have efficient storage, processing, and management of IoT-generated data. Though NoSQL databases are frequently adopted for IoT applications, MySQL can be used with these optimizations.

Databases used in Internet of Things Environments

Various database technologies can be used for different IoT application requirements [1]:

Time-series databases such as InfluxDB, TimescaleDB, and OpenTSDB are used for handling timestamped sensor data like telemetry, monitoring metrics, and log analytics.

NoSQL databases can be used to manage semi-structured and unstructured data efficiently. JSON-based data are stored in Document-oriented databases like MongoDB and Couchbase and in wide-column databases such as Apache Cassandra and ScyllaDB support distributed IoT environments.

Relational databases, like PostgreSQL and MySQL, are used in transactional IoT applications, such as billing systems, inventory control, and asset management.

Graph databases like Neo4j and JanusGraph excel are used in analyzing relationships among interconnected devices, making them valuable for network topology analysis, device dependency mapping, and anomaly detection.
Database Management Systems (DBMSs) are divided into two, one is relational databases and the other is non-relational (NoSQL) databases. MySQL is a relational DBMS and MongoDB is a non-relational DBMS.

Selecting database for Iot Architecture

In an IoT (Internet of Things) architecture, selecting an appropriate database depends on the specific application requirements, including data volume, the need for real-time processing, the nature of the data structure, and scalability demands [2].

Scalability

To handle huge volumes of data generated by IoT applications, the database system must support efficient scalability. There are two types of scaling like vertical and horizontal. By increasing the hardware resources like processor power, memory, or storage capacity, Vertical scaling improves the performance of a single server. The physical and cost constraints limits the vertical scaling approach. Data and workloads can be distributed across multiple servers in the case of Horizontal scaling, which allows the system to expand more flexibly and support larger data volumes. A load balancer can be used to improve horizontal scalability by distributing incoming requests evenly among multiple database servers or clusters, which in turn increase the performance and resource utilization [3].

Cost

The cost for purchasing a database is very high compared to open-source database solutions. Though open source database are cost-effective, it also require expenses related to technical support, maintenance, customization, and application development when evaluating their overall cost-effectiveness [3].
MySQL is widely regarded as a secure and dependable database platform that powers numerous prominent websites and applications. Its proven performance, stability, and reliability make it a suitable backend database solution for IoT environments [4].

Figure 1 shows the process flow involved in the collection, management, and storage of IoT data.

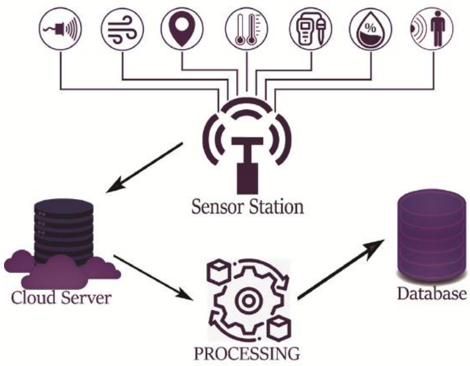

Figure 1

4. Proposed Framework Architecture

System Components

1. **Server)**
   MySQL server is used for data storage, indexing, partition management, and replication to ensure efficient performance, scalability, and reliability.
2. **Analytics and Visualization Layer IoT Devices/Sensors**
   Real-time operational or environmental data is generated from connected devices.
3. **Data Acquisition Layer**
   Edge devices and microcontrollers, such as Raspberry Pi, collect sensor readings and HTTP or MQTT communication protocols are used to transmit them .
4. **Application Layer**
   Incoming data received by API Server is validated and processed before storing it in the MySQL database.
5. **Database Layer (MySQL**
   Dashboards and reporting tools for real-time monitoring, data analysis, and decision support is done in this layer.

Architecture Flow

IoT Sensors → Data Gateway → Application/API Server → MySQL Database → Analytics and Visualization Dashboard
The above workflow shows how the system components are used in the movement of data.Data is moved from IoT sensors through a gateway and then API server, where it is processed and stored in a MySQL database and then to analytics dashboard for monitoring and analysis.

5. Database Design Strategy

5.1 Schema Optimization

Here Third Normal Form (3NF) is used for designing a database, where device information is separated from sensor measurement data to minimize redundancy and improve data integrity. Various data types like INT, FLOAT, and DATETIME are used to optimize storage utilization and enhance query performance.

Sample Table Structure:

Devices Table

- device_id (Primary Key)
- location
- device_type

Sensor_Data Table

- record_id (Primary Key)
- device_id (Foreign Key)
- temperature
- humidity
- time stamp

IoT sensor data is organized and normalized the information using multiple tables according to sensor types to satisfy the MySQL schema. MySQL can be used in an IoT environment which will give an efficient database structure for managing sensor data. This will enable smooth integration of data from various sources, and improve analysis of collected sensor information [5].

5.2 Performance Optimization Techniques

Creating indexes on fields such as device_id and timestamp, using date-based range partitioning, and the InnoDB storage engine will enhance performance. Connection pooling and batch insertion techniques can be implemented to manage large volumes of incoming data .This will improve overall database performance.

6. Scalability Techniques

Server resources can be used for improving scalability through vertical scaling. System capacity can be expanded with MySQL replication-based horizontal scaling. Master–slave replication can be used to support read-heavy operations, and load balancing mechanisms can be employed to distribute client requests efficiently across multiple replica servers.

7. Experimental Setup

The experimental setup consisted of a Raspberry Pi serving as the data generation device and an **Intel** Core i5 server with 8 GB RAM acting as the database server. The software environment included a Linux operating system, MySQL 8.x for data storage and management, and Python for simulating IoT sensor data .

Performance Metrics:

Metrics such as query response time, data insertion throughput, CPU usage, and storage growth rate, which collectively assess the system's efficiency, performance, and scalability are used for performance evaluation.

Comparing MySQL with MongoDB, approximately 83.7% lower latency is achieved by MongoDB than MySQL. Database size increases in a linear manner in the MongoDB reference architecture. On the other hand, MySQL have periods of steady growth interspersed with phases of relative stability. MySQL shows superior performance when handling smaller data insertion workloads, though its performance degrades more noticeably as the workload increases. MongoDB maintains better scalability under higher data volumes [6].

Sensor measurements are stored as separate records in MySQL for each sensor reading, enabling structured data organization and management. The results shows that MySQL have strong performance in high-throughput environments and, in certain scenarios, outperforms MongoDB [7].

One of the most extensively adopted database management systems worldwide is MySQL. It is used in a wide range of commercial, industrial, and scientific applications. The advantages of MySQL make it a suitable choice for IoT deployments. Some of the notable features of MySQL is its robust data security, which is a necessary requirement for IoT environments. Many prominent websites and organizations use MySQL due to its secure and reliable platform for protecting sensitive data [8].

Scalability: Rapid and unpredictable growth of data occur in IoT applications and hence scalable database solutions is a must. Efficient management of data is possible in MySQL and it process large volumes of data generated by various IoT devices connected.

High Performance and Availability: Huge workloads are manageable by MySQL and the continuous data streams produced by IoT applications are handled by MySQL. As it is open source and its high availability ensures uninterrupted operation.

Open-Source Adaptability: As MySQL is an open-source database customization is possible according to specific IoT application requirements. As it is accessible to large community of developers, administrators, and support resources more flexibility can be provided.

The SQL Diagnostic Manager for MySQL ia a tool effectively used for monitoring, optimizing, and managing MySQL databases that power IoT infrastructures [4].

8. Results & Analysis (Sample Structure)

Data Rate (records/sec)	Avg Insert Time (ms)	Query Time (ms)
100	5	3
500	12	6
1000	25	10

Approximately 40% query execution time is decreased in indexing, significantly improving data retrieval performance. Partitioning enhances the efficiency of querying large datasets as the amount of data scanned is reduced during operations. Replication also increased read scalability, which enable the system to handle large number of read requests and improving overall database performance.

Both structured and semi-structured sensor data can be efficiently managed by Oracle's MySQL HeatWave. In-database machine learning capabilities are integrated in it which allow users to build, train, and deploy machine learning models directly through SQL without transferring data to external analytics platforms. HeatWave supports large-scale real-time analytics by using highly optimized parallel and vectorized query execution which in turn delivers fast and efficient processing of MySQL data. Additionally, it is flexible enough to operate in edge and cloud environments, allowing MySQL to run on devices such as Raspberry Pi at the edge and within Kubernetes-managed containers in the cloud[9].

9. Discussion

The study shows that MySQL can effectively handle moderate to high IoT workloads when optimized properly. Though NoSQL databases have more flexibility in handling various data formats, MySQL also offers strong ACID compliance, which ensures robust data integrity through structured schemas. MySQL have the advantage of using more complex relational queries which is suitable for many IoT and enterprise applications.

10. Conclusion

The study shows that MySQL can serve as a scalable and high-performance storage backend for IoT applications when used with optimized schema design, indexing, partitioning, and replication mechanisms. Overall system efficiency can be improved by the proposed framework which in turn ensures data reliability, and supports the requirements of real-time data processing and analytics.

Future study can be made on incorporating AI-based query optimization techniques, integrating MySQL with big data processing frameworks, and comparative study on NoSQL database solutions.

References

1. https://stonefly.com/blog/iot-data-storage-architectures-databases/
2. https://www.tencentcloud.com/techpedia/135369
3. https://www.tigerdata.com/learn/how-to-choose-an-iot-database
4. https://webyog.com/implementing-the-internet-of-things-with-mysql/
5. Using MySQL for Storing and Analysing Sensor Data in IoT System Kiran H M HBRP Publication Page 72-78 2025
6. Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments Mahmoud Eyada , Walaa Saber , Mohammed M. El Genidy, And Fathy Amer 2020
7. Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments Mahmoud Eyada , Walaa Saber , Mohammed M. El Genidy, and Fathy Amer2020
8. https://ozgurozkok.com/implementing-the-internet-of-things-with-mysql/
9. https://www.linkedin.com/pulse/why-mysql-hidden-gem-ai-powered-iot-applications-ivg4c