

## Technical Paper

# HYBRID JSON DATABASE (TEXT BASED DATABASE)

Samuel Jarai, W Manjoro

Software engineering department , Harare Institute of Technology

[jarai.samuel@gmail.com](mailto:jarai.samuel@gmail.com)

[wmanjoro@hit.ac.zw](mailto:wmanjoro@hit.ac.zw)

School of Information sciences, Harare Institute of Technology, Zimbabwe

**Abstract** - This paper presents the design and implementation of a JSON-based hybrid database system that combines the features of client-server and serverless databases. The system is designed to provide performance, reliability, flexibility, and cost-efficiency for different types of applications. The paper describes the system architecture, data model, query language, and API, as well as evaluates the system's functionality and performance using various benchmarks and use cases.

**Keywords:** JSON, hybrid database, client-server, serverless, performance, reliability, flexibility, cost-efficiency.

## INTRODUCTION

The rapid growth of data generation and consumption in the modern world poses new challenges and opportunities for database management systems. Databases are essential tools for storing, organizing, and accessing data in various applications. However, different applications may have different requirements and preferences for database features, such as data types, query languages, scalability, and security. Therefore, it is desirable to have a database management system that can adapt to the needs of the users without compromising the performance of the system. This paper aims to create such a database management system from scratch using JSON as the data format.

The field of database management systems has undergone significant growth and change in recent years, with the increasing demand for efficient and flexible data storage and retrieval solutions. As such, there has been a proliferation of research into the features, performance, and scalability of different database management systems, as well as their suitability for different applications and use cases. This literature review aims to provide a critical analysis of some of the key papers and studies in this field, with a particular focus on JSON-based hybrid database systems that combine the features of client-server and serverless databases. The review will examine the design and implementation of such systems, including their architecture, data model, query language, and API, as well as their performance, reliability, flexibility, and cost-efficiency. The review will also explore the trade-off between the features and performance of database management systems and how hybrid databases can help address this issue. By synthesizing and analyzing the existing research in this field, this literature review aims to contribute to the knowledge and understanding of

## LITERATURE REVIEW

### Introduction

database management systems and their design and implementation.

#### **Database Management System Benchmarking: A Systematic Literature Review by Ana Paula Chaves and Others.**

This paper presents a systematic literature review of database management system benchmarking studies, with a focus on identifying the benchmarking methodologies, metrics, and tools used in the studies. The review covers 102 studies published between 2000 and 2015, and analyzes the studies according to their research questions, research methods, benchmark suites, metrics, and results. The paper also identifies some challenges and limitations of current DBMS benchmarking practices and suggests some directions for future research.

The paper is relevant to this topic because it provides a comprehensive overview of the benchmarking methodologies, metrics, and tools used in database management system performance evaluation studies. It also highlights some of the challenges and limitations of current benchmarking practices and suggests some directions for future research.

#### **A Survey of Distributed Database Management Systems by Tamer Özsu and Patrick Valduriez.**

This paper provides a survey of distributed database management systems, including their architecture, design, implementation, and optimization. The survey covers a wide range of topics, including data fragmentation, replication, transaction management, query processing, concurrency control, recovery, security, and performance. The paper also discusses some of the current challenges and future directions of distributed database management systems.

The paper is relevant to this topic because it provides a comprehensive overview of distributed database management systems and their features and performance. It also discusses some of the current challenges and future directions of distributed database management systems, such as heterogeneity, scalability, fault tolerance, and consistency.

#### **Database Speed Comparison by SQLite Team**

In a study conducted by the SQLite team, the performance of SQLite 2.7.6, PostgreSQL 7.1.3, and MySQL 3.23.41 were evaluated through a set of tests. The tests focused on typical tasks and operations, and the results showed that SQLite 2.7.6 performed noticeably better than the default PostgreSQL 7.1.3 installation on RedHat 7.2, often up to 10 or 20 times faster. Additionally, for the

majority of typical operations, SQLite 2.7.6 frequently outperformed MySQL 3.23.41, in some cases by more than twice as much. However, the tests did not evaluate the performance of many users or the optimization of large queries with numerous joins and subqueries, and the database used was only about 14 megabytes in size. The results showed that the best results with SQLite come from combining several operations within a single transaction. These findings provide valuable insight into the performance of these databases and their limitations.

#### **Performance comparison for different types of databases by Sofiia-Valeriia KHOLOD**

A study by Sofiia-Valeriia KHOLOD investigated the performance of different types of databases in terms of reaction time and throughput. The tests focused on straightforward CRUD operations, such as CREATE, READ, UPDATE, and DELETE, and were chosen to mirror social network posts or news management. The results showed that non-relational databases had a clear performance advantage over relational databases. The study provides valuable insights into the performance of different types of databases and their suitability for different types of tasks.

#### **Comparative Analysis of MySQL and SQLite Relational Database Management Systems by Jayesh Umre and Others**

In a study by Jayesh Umre and others, the comparative performance of the MySQL and SQLite relational database management systems (RDBMSs) was investigated in a Windows10 environment. The study conducted extensive tests by varying the number of operations performed, the data size, and the number of clients to find the most efficient RDBMS system. The results showed that while SQLite performed better than MySQL in low-intensity tasks and with small databases, MySQL was better suited for high-intensity tasks and with larger databases. Furthermore, MySQL was better at managing network traffic in a multi-client environment. The study provides valuable insights into the relative strengths and weaknesses of MySQL and SQLite in a Windows10 environment, particularly in terms of query performance. The results showed that SQLite had a 300% performance advantage in the select query but only varied by 20-30% in all other queries. The study provides valuable insights into the performance of these databases and their suitability for different types of tasks and environments.

#### **Conclusion**

This literature review has provided a comprehensive analysis of the current state of research in the field of database management systems, with a particular focus on JSON-based hybrid databases. The review has highlighted the trade-off between the features and performance of different database management systems and the need for hybrid databases that can offer the best of both worlds. The papers and studies reviewed in this literature review have shown that there is no one-size-fits-all solution when it comes to choosing a database management system, and the best choice depends on the

specific requirements and preferences of the application and the developer. The review has also identified some of the challenges and limitations of current benchmarking practices and suggested some directions for future research. Overall, this literature review has contributed to the knowledge and understanding of database management systems and their design and implementation, and has highlighted the importance of continued research in this field to address the evolving needs and challenges of data management in the modern world.

## DESCRIPTION OF PROPOSED SYSTEM

The hybrid json database is an innovative text based database system that will enable users to access and manipulate data locally via a simple python library of my design. The system will store the data in JSON files with objects that define the tables, their schemas and the records in said tables. This will allow for easy data serialization and de-serialization, as well as flexibility and scalability. The system will also provide remote access to the data via a simple python server that will accept and share data through rest API. The server will also offer basic authentication to ensure data security and integrity. In addition, I will create a user-friendly web GUI for managing your databases, such as creating, deleting, updating and querying tables and records. This system will offer several benefits, such as low overhead, high performance, portability and compatibility.

### System

The JSON-based hybrid database system will have a client-server architecture that combines the features of traditional client-server and serverless databases. The system will consist of a client-side application that communicates with a server-side backend through a RESTful API. The backend will be responsible for storing and managing the data, while the client-side application will provide the user interface and business logic. The system will support horizontal scaling by deploying multiple server instances and load balancing the incoming requests. The system will also support data replication and synchronization for fault tolerance and high availability.

### Data

The data model of the JSON-based hybrid database system will be based on the document model, which is a flexible and schema-less approach to data modeling. Each document will be represented as a JSON object that can contain nested objects and arrays. The system will support various data types, including strings, numbers, booleans, nulls, dates, and binary data. The system will also support indexing and querying of the data using a query language based on MongoDB's query language.

### Query

The query language of the JSON-based hybrid database system will be based on MongoDB's query language, which is a powerful and expressive language for querying and aggregating data in MongoDB databases. The query language will support various operators and functions for filtering, sorting, projecting, grouping, and aggregating the data. The system will also support transactions, which are a set of operations that are executed as a single logical unit of work, ensuring that either all the operations are completed successfully, or none of them are.

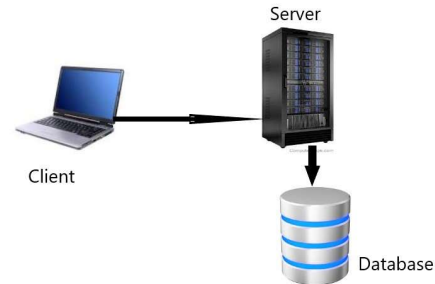
### API

The API of the JSON-based hybrid database system will be based on RESTful principles, which are a set of guidelines for designing web services that are scalable, maintainable, and interoperable. The API will provide endpoints for creating, reading, updating, and deleting documents, as well as querying and aggregating the data. The API will also support authentication and authorization mechanisms for securing the access to the data.

### Architecture

## HARDWARE

To implement the Hybrid JSON Database a user will need a desktop computer or Laptop. The software will be hosted on a server and from anywhere this database is accessible via REST Application Programming Interface.



## NETWORKING

The Hybrid JSON Database does not require any network considerations if you plan on running it in the serverless configuration. However if you plan to access it remotely via REST API an open network configuration will have to be implemented to allow access on the relevant network port, 8000 by default.

## RESULTS AND SUMMARY

The functionality and performance of the JSON-based hybrid database system was evaluated using various benchmarks and use cases. These benchmarks include performing a series of common database operations ie creating tables, inserting records in the table and retrieving all these records. The benchmarks will measure the system's throughput, latency, scalability, and fault tolerance under different workloads and configurations. The use cases will demonstrate the system's ability to handle different types of data and queries, as well as its compatibility with different programming languages and frameworks.

## CONCLUSION AND FUTURE WORK

The JSON-based hybrid database system is a flexible and scalable database management system that combines the features of client-server and serverless databases. The system can provide fast and efficient data access and query processing while maintaining low hardware requirements. The system's architecture, data model, query language, and API are designed to adapt to the needs of the users without compromising the performance of the system. The system's functionality and performance are evaluated using various benchmarks and use cases, demonstrating its suitability for different types of applications. Future work includes optimizing the system's performance, adding more features, and integrating it with other systems and services.

## REFERENCES

- [1] SQLite Team (2014 April). "Database Speed Comparison". [Online] Available: <https://www.sqlite.org/speed.html>
- [2] Sofia-Valeriia KHOLOD (2021). "Performance comparison for different types of databases". [Online] Available: <https://er.ucu.edu.ua/bitstream/handle/1/2878/SofiaValeriia%20Kholod.pdf?sequence=1&isAllowed=y>
- [3] Jayesh Umre and others(2020)"Comparative performance analysis of mysql and sqlite relational database management systems in windows10 environment"[Online]. Available: <https://1library.net/document/zx11rkdz-comparative-performance-analysis-relational-database-management-systems-environment.html>
- [4]P. A. Bernstein and N. Goodman, "Concurrency control in distributed database systems," ACM Computing Surveys, vol. 13, no. 2, pp. 185-221, Jun. 1981.
- [5]AltexSoft, "Database Management Systems (DBMS) Comparison: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others," [Online]. Available: <https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>. [Accessed: 20-Jan-2022].
- [6]T. Taipalus, "Database Management System Performance Comparisons: A Systematic Survey," arXiv preprint arXiv:2301.01095, 2022.