SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

Exploring the Landscape of Distributed Database Management Systems: A Literature Review of Common Architectural Models

Joshua C. Reyes
Tarlac State University, Philippines

Article Info:
Received: ***25 Aug 2023;*** *Revised:* ***05 Dec 2023;*** *Accepted:* ***20 Dec 2023;*** *Available Online:* ***13 Dec 2023***

Abstract – *This literature review examines the common Distributed Database Management System (DDBMS) architectures: Client-Server, Peer-to-Peer, and Multi-DBMS. It compares their performance, availability, accessibility, and data analysis capabilities. The review also identifies key factors that differentiate their suitability for specific application scenarios based on their strengths and weaknesses. This analysis aims to provide a framework for organizations to select the most appropriate DDBMS architecture for their diverse needs.*

Keywords – *Distributed Database Management System (DDBMS), Client-Server architecture, Peer-to-Peer architecture, Multi-DBMS architecture, Performance, Availability, Accessibility, Data Analysis.*

INTRODUCTION

The expanding volume and complexity of data necessitate robust management solutions, particularly for geographically dispersed organizations. Distributed Database Management Systems (DDBMS) have emerged as a powerful tool in this domain, offering significant advantages over traditional centralized systems. As highlighted by Orlunwo & Prince (2021), DDBMS offers improved performance by leveraging multiple resources, increased availability through data redundancy, distributed access enabling remote collaboration, and enhanced data analysis capabilities across geographically separated sites.

Despite these advantages, selecting the most suitable DDBMS architecture for a specific application remains a challenge. Several common architectures exist, each with its strengths and weaknesses. Client-server, peer-to-peer, and multi-DBMS architectures offer distinct functionalities and cater to different application needs.

While existing research offers valuable insights into the individual characteristics of these architectures, a comprehensive comparison across performance, availability, accessibility, and data analysis capabilities is lacking. Azizah et al. (2022) acknowledge the importance of these factors for DDBMS selection, but a systematic comparison is still rare.

This knowledge gap presents a significant barrier to optimal DDBMS selection. Without a clear understanding of how these architectures perform in each key area, organizations may struggle to identify the most suitable solution for their specific needs.

This literature review aims to address this gap by systematically analyzing and comparing the three common DDBMS architectures (Client-Server, Peer-to-Peer, and Multi-DBMS) across performance, availability, accessibility, and data analysis characteristics. Through this analysis, the review aims to identify the key factors that differentiate the suitability of each architecture for specific application scenarios based on their strengths and weaknesses. This paper seeks to provide a comprehensive framework to guide organizations in selecting the most appropriate DDBMS architecture for their diverse application needs.

OBJECTIVES OF THE STUDY

The primary objective of this study is to explore and compare the common Distributed Database Management System (DDBMS) architectures—Client-Server, Peer-to-Peer, and Multi-DBMS—in terms of

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

their performance, availability, accessibility, and data analysis capabilities.

Specifically, this study aims to:

- Identify and examine the characteristics of common Distributed Database Management System (DDBMS) architectures, namely Client-Server, Peer-to-Peer, and Multi-DBMS architectures.
- Compare the performance of the selected DDBMS architectures in terms of speed, scalability, efficiency, and query processing capabilities.

MATERIALS AND METHODS

This study employs a systematic literature review based on the original guidelines introduced by Kitchenham (2007), which propose identifying, evaluating, and interpreting all primary studies related to the topic of interest. The steps in this review method are documented below.

Research Questions

The research questions addressed by this study are:

Table 1. The research questions addressed by this study.

RQ1	How do the common DDBMS architectures (Client-Server Architecture, Peer-to-peer Architecture, and Multi DBMS Architecture) compare in terms of their performance, availability, accessibility, and data analysis characteristics?
RQ2	What are the key factors that differentiate the suitability of each DDBMS architecture for specific application scenarios based on their strengths and weaknesses?

To ensure the selection of relevant studies, this review employed the PICOC criteria (Population, Intervention, Comparison, Outcomes, and Context). Table 2 outlines the application of the criteria in this study.

Table 2. The definition of PICOC criteria.

Criterion	Description
Population	Common Distributed Database Management Systems (DDBMS) architectures (Client-Server, Peer-to-Peer, Multi-DBMS)
Intervention	*Not applicable*
Comparison	Performance, Availability, Accessibility, Data Analysis characteristics
Outcomes	How the architectures compare based on the chosen characteristics
Context	Specific application scenarios

The RQ1 focuses on comparing the characteristics (Outcomes) of different DDBMS architectures (Population) based on performance, availability, accessibility, and data analysis capabilities (Comparison). The RQ2 shifts the (Context) to specific application scenarios. It asks what factors (Outcomes) make each architecture (Population) more suitable for different applications based on their strengths and weaknesses.

Since this study focuses on a comparative analysis of existing Distributed Database Management Systems (DDBMS) architectures, the "Intervention" element within the PICOC criteria is not applicable.

Search Process

For the formulation of the search chain, concepts were established to allow greater precision in the relevant information results. These concepts were formed from the combination of the terms previously defined in the Population, Comparison, Outcomes, and Context criteria established in Table 2 and their synonyms corresponding to each term.

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

Search terms. The population, comparison, outcomes, and context criteria were considered for the extraction of studies.

The search for literature related to the study was conducted on Google Scholar (scholar.google.com). This choice was made due to Google Scholar's ability to aggregate papers from various research databases, including but not limited to Web of Science, Scopus, PubMed, IEEE Xplore, ACM Digital, Library, and SpringerLink, making it a valuable resource for accessing relevant research dispersed across multiple databases.

Inclusion and Exclusion Criteria

To ensure the relevance and quality of the studies selected for this review, certain inclusion and exclusion criteria were established:

Inclusion Criteria

The inclusion criteria focus on the factors and conditions that need to exist so that we can select the paper and include it in the primary studies list. The criteria are mentioned below:

1. The paper answers the RQs, either with a full or a partial.
2. The paper's publication date is 2019 and above.
3. The paper should exist on one of the primary sources (Scopus, IEEE Xplorer, ACM Digital Library, Web of Science, and Google Scholar) if a snowballing search happened.

Exclusion Criteria

The exclusion criteria focus on the factors and conditions that cause the paper to be excluded from the primary studies list. The criteria are mentioned below:

1. The paper's publication date is before 2019.
2. The paper's author is not known.
3. The paper does not answer the RQs.
4. The paper is considered a duplicate of another paper with a different title, and the researcher opted for the most recent version.

Data Extraction

The data extracted from each study included the following information:

Table 3. The data extraction form format.

Field	Description	Type of Information
ID	Unique code to identify the scientific article	General
Year	Year of publication	General
Title	Title of the scientific article	General
Author	Author(s), their institution, and the country where it is located	General
Source	Source (journal or conference) and full reference	General
Abstract	Summary of the study, including the primary research questions and their corresponding answers	General
Architectural Model	Refers to a specific type of Distributed Database Management System (DDBMS) architecture discussed in the literature	General
Performance	Performance of the architecture in terms of speed, efficiency, etc., as discussed in the literature	Related to question 1
Availability	Availability of the architecture in terms	Related to question 1

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

	of uptime, fault tolerance, etc., as discussed in the literature	
Accessibility	Accessibility of the architecture is in terms of ease of use, scalability, etc., as discussed in the literature	Related to question 1
Data Analysis	Characteristics of the architecture related to data analysis as discussed in the literature	Related to question 1
Strengths	Advantages highlighted in the literature for the architecture	Related to question 2
Weaknesses	Limitations discussed in the literature for the architecture	Related to question 2
Suitability for Specific Scenarios	Specific application scenarios where the literature discusses the suitability of the architecture based on its strengths and weaknesses	Related to question 2

Data Analysis

The data was organized in a tabulated format to showcase the following information:

- The details of obtained relevant articles.
- The comparative analysis of performance across common DDBMS architectures.
- The comparative analysis of availability across common DDBMS architectures.
- The comparative analysis of accessibility across common DDBMS architectures.
- The comparative analysis of data analysis across common DDBMS architectures.
- The analysis of strengths across common DDBMS architectures.
- The analysis of weaknesses across common DDBMS architectures.
- The analysis of suitability for specific scenarios across common DDBMS architectures.

The results of the data analysis were presented in the "Results" section. A comprehensive summary of the extracted data and the synthesis of findings was provided. The data were presented using tables to facilitate understanding and comparison across studies.

RESULTS AND DISCUSSION

RESULTS

Search Results

By utilizing the search chain that was established beforehand, several articles were collected from all the databases. After applying the inclusion and exclusion criteria determined later, a total of fifteen (15) relevant articles were selected, with five (5) articles per Distributed Database Management System (DDBMS) architecture. The details of these papers are summarized in Table 4.

Table 4. The details of obtained relevant articles.

ID	Literature	Architectural Model
RR01	(Jinadu et al., 2021)	Client-Server
RR02	(Abdelhafiz, 2020)	Client-Server
RR03	(Naik, 2019)	Client-Server
RR04	(Ojekudo & Eze, 2021)	Client-Server
RR05	(Kyaw & Aung, 2020)	Client-Server
RR06	(Ezechiel et al., 2019)	Peer-to-Peer
RR07	(Ramesh et al., 2022)	Peer-to-Peer
RR08	(Fazlali et al., 2019)	Peer-to-Peer
RR09	(Sarode, 2021)	Peer-to-Peer
RR10	(Ahmed et al., 2021)	Peer-to-Peer

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

RR11	(Chiara Forresi et al., 2021)	Multi DBMS
RR12	(Holubova et al., 2021)	Multi DBMS
RR13	(Esposito et al., 2021)	Multi DBMS
RR14	(Groppe & Groppe, 2020)	Multi DBMS
RR15	(Lu & Holubová, 2019)	Multi DBMS

RQ1: How do the common DDBMS architectures (Client-Server Architecture, Peer-to-peer Architecture, and Multi DBMS Architecture) compare in terms of their performance, availability, accessibility, and data analysis characteristics?

To address this research question, a comprehensive review of relevant literature was conducted, analyzing how common DDBMS architectures (Client-Server, Peer-to-Peer, and Multi-DBMS) compare in terms of performance, availability, consistency, and security characteristics.

Table 5. The comparative analysis of performance across common DDBMS architectures.

Performance	
Client-Server	Offers improved performance through techniques like distributed storage pools, parallel processing, and optimized query processing (Jinadu et al., 2021; Abdelhafiz, 2020; Ojekudo & Eze, 2021).
Peer-to-Peer	The potential for scalability and high-throughput computing suggests potential performance benefits (Fazlali et al., 2019). Blockchain technology can enable faster transactions compared to traditional protocols (Sarode, 2021).
Multi-DBMS	Performance might vary depending on the specific implementation. Some systems might achieve good performance by optimizing for specific queries involving data from multiple sources (Forresi et al., 2021).

Table 5 shows the comparative analysis of performance across common DDBMS architectures.

Table 6. The comparative analysis of availability across common DDBMS architectures.

Availability	
Client-Server	High availability (HA) can be achieved through storage virtualization, Redundant Array of Independent Disks (RAID) technology, and backups (Jinadu et al., 2021; Abdelhafiz, 2020). Distributed architecture inherently reduces the risk of a single point of failure (Abdelhafiz, 2020).
Peer-to-Peer	Fault tolerance is a potential benefit due to data replication across peers (Ezechiel et al., 2019; Ramesh et al., 2022). Blockchain technology can further enhance fault tolerance (Ramesh et al., 2022).
Multi-DBMS	Data availability might be improved through potential distribution across different DBMS (Forresi et al., 2021). Security mechanisms using blockchain can potentially improve data availability by restricting unauthorized access (Esposito et al., 2021).

Table 6 shows the comparative analysis of availability across common DDBMS architectures.

Table 7. The comparative analysis of accessibility across common DDBMS architectures.

Accessibility

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

Client-Server	Ubiquitous data access is possible through distributed resource pools and location transparency (Jinadu et al., 2021). However, setting up access for users might be complex compared to local data access (Naik, 2019). Server-based architecture can potentially scale to accommodate more clients without impacting individual performance (Ojekudo & Eze, 2021).
Peer-to-Peer	Accessibility discussions are limited in the reviewed papers.
Multi-DBMS	A multistore concept suggests potential benefits for data accessibility across different data models or storage locations (Forresi et al., 2021). Distributed access control mechanisms can potentially improve accessibility by managing user access (Esposito et al., 2021).

Table 7 shows the comparative analysis of accessibility across common DDBMS architectures.

Table 8. The comparative analysis of data analysis across common DDBMS architectures.

Data Analysis	
Client-Server	Scalability, parallel processing, and the ability to access data from multiple heterogeneous databases make it suitable for big data analysis (Jinadu et al., 2021; Naik, 2019). However, data analysis tasks might require additional steps to locate and integrate relevant data from different servers (Naik, 2019).
Peer-to-Peer	The distributed nature of data might require additional steps for locating and integrating data for analysis tasks (Ezechiel et al., 2019; Ramesh et al., 2022). Blockchain technology can potentially facilitate secure data sharing for collaborative data analysis (Ramesh et al., 2022). Specific functionalities like using a BST for efficient data organization can improve performance for specific data analysis tasks involving multi-attribute range queries (Ahmed et al., 2021).
Multi-DBMS	A significant advantage for data analysis tasks that involve integrating data from various sources with potentially diverse formats (Forresi et al., 2021; Holubova et al., 2021; Lu & Holubová, 2019). MMDBMS allows querying across different data models, enabling comprehensive data analysis tasks (Chiara Forresi et al., 2021).

Table 8 shows the comparative analysis of data analysis across common DDBMS architectures.

RQ2: What are the key factors that differentiate the suitability of each DDBMS architecture for specific application scenarios based on their strengths and weaknesses?

To address this research question, a comprehensive review of relevant literature was conducted, aiming to identify the factors that differentiate the suitability of each DDBMS architecture (Client-Server, Peer-to-Peer, and Multi-DBMS) based on their strengths and weaknesses.

Table 9. The analysis of strengths across common DDBMS architectures.

Strengths	
Client-Server	*Scalability:* Well-suited for large-scale deployments with growing

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

	data volumes (Abdelhafiz, 2020; Kyaw & Aung, 2020). *Improved Data Availability and Fault Tolerance:* Offers high availability and redundancy mechanisms for critical data (Jinadu et al., 2021; Abdelhafiz, 2020) *Support for Heterogeneous Databases:* Can integrate with various database management systems (Naik, 2019) *Partial Data Distribution:* Enables distribution of specific data subsets for optimized management (Naik, 2019) *Optimized Query Processing:* Can leverage specific functionalities for efficient query processing within the architecture (Ojekudo & Eze, 2021)
Peer-to-Peer	*Decentralization:* Eliminates central points of failure and promotes data ownership distribution (Ezechiel et al., 2019]; Ramesh et al., 2022) *Security:* Blockchain integration can enhance data security and privacy in specific scenarios (Ramesh et al., 2022l; Sarode, 2021) *Scalability:* Generally considered highly scalable due to distributed data storage (Fazlali et al., 2019) *Improved Transaction Efficiency (with specific protocols):* Architectures using Raft consensus mechanisms can achieve high transaction efficiency (Fazlali et al., 2019) *Efficient Range Queries (with specific protocols):* Distributed Binary Search Tree (DBST) integration can enable efficient searches for specific data attributes (Ahmed et al., 2021)
Multi-DBMS	*Integration of Heterogeneous Data Sources:* Allows seamless access and querying of data from diverse Database Management Systems (DBMS) (Forresi et al., 2021; Holubova et al., 2021) *Data Modeling Flexibility:* Accommodates various data models and structures used by different Database Management Systems (DBMS) (Groppe & Groppe, 2020; [Lu & Holubová, 2019) *Unified Data Access and Querying:* Provides a single interface for accessing and querying data across different sources (Holubova et al., 2021; Esposito et al., 2021) *Potential for Improved Performance:* Carefully designed Multi-DBMS architectures might leverage platform-specific functionalities for optimized performance (Groppe & Groppe, 2020)

Table 9 shows the analysis of strengths across common DDBMS architectures.

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

Table 10. The analysis of weaknesses across common DDBMS architectures.

Weaknesses	
Client-Server	*Complexity:* Requires careful design and management compared to simpler architectures (Abdelhafiz, 2020)
	Increased Response Times: Complex communication paths between clients and servers can introduce latency (Abdelhafiz, 2020).
Peer-to-Peer	*Data Consistency and Conflicts:* Maintaining data consistency across distributed nodes can be challenging (Ezechiel et al., 2019)
	Complexity of Consensus Mechanisms: Reaching agreement on data updates in decentralized systems can introduce complexity (Fazlali et al., 2019)
	Potential Challenges with Large-Scale Adoption: Complexity might hinder broader application in large-scale deployments (Sarode, 2021)
Multi-DBMS	*Complexity of Data Integration and Management:* Integrating and managing data across various systems can be a significant challenge (Forresi et al., 2021; Esposito et al., 2021)
	Overhead of Query Optimization: Optimizing queries across diverse data sources can introduce additional processing overhead (Forresi et al., 2021)
	Challenges in Schema Design and Representation: Ensuring consistent data representation across heterogeneous systems requires careful design (Holubova et al., 2021)

Table 10 shows the analysis of weaknesses across common DDBMS architectures.

Table 11. The analysis of suitability for specific scenarios across common DDBMS architectures.

Suitability for Specific Scenarios	
Client-Server	Cloud-based big data applications (Jinadu et al., 2021)
	Applications requiring high data availability and fault tolerance (Abdelhafiz, 2020; Kyaw & Aung, 2020)
	Distributed applications with partial data distribution or needing integration with heterogeneous databases (Naik, 2019)
	Optimized query processing in distributed database systems (Ojekudo & Eze, 2021)
Peer-to-Peer	Scenarios with limited data updates and a focus on data availability (Ezechiel et al., 2019)
	Secure data sharing and collaboration in specific domains, leveraging blockchain technology (Ramesh et al., 2022)
	High-throughput distributed computing applications requiring strong consistency, potentially using Raft consensus mechanisms (Fazlali et al., 2019)

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

	Specific scenarios requiring fast and secure P2P transactions with manageable complexity, potentially using blockchain (Sarode, 2021)
	Distributed systems requiring efficient multi-attribute range queries, potentially using DBST (Ahmed et al., 2021)
Multi-DBMS	Organizations with diverse data sources requiring integration (Forresi et al., 2021; Holubova et al., 2021)
	Applications requiring seamless data integration from various sources (Esposito et al., 2021; Lu & Holubová, 2019)
	Organizations managing data across diverse platforms and models, potentially leveraging platform-specific functionalities (Groppe & Groppe, 2020)

Table 11 shows the analysis of suitability for specific scenarios across common DDBMS architectures.

DISCUSSION

This section delves into the key findings presented in the Results section, aiming to provide a comprehensive understanding of the comparative analysis of common Distributed Database Management System (DDBMS) architectures, namely Client-Server, Peer-to-Peer, and Multi-DBMS architectures. It will also address the key factors that differentiate the suitability of each DDBMS architecture for specific application scenarios based on their strengths and weaknesses.

Comparative Analysis of DDBMS Architectures

Performance. In terms of performance, the Client-Server architecture demonstrates improved performance through distributed storage pools, parallel processing, and optimized query processing techniques. However, the potential impact of network overhead on performance, compared to local data access, is acknowledged. Peer-to-peer architectures suggest scalability and high-throughput computing benefits, while Multi-DBMS performance varies depending on the specific implementation. Understanding these nuances is crucial for selecting the most suitable architecture based on performance requirements.

Availability. The availability analysis highlights that Client-Server architectures achieve high availability through storage virtualization, RAID technology, and backups. Peer-to-Peer architectures benefit from fault tolerance due to data replication and Multi-DBMS architectures potentially improve data availability through distribution across different DBMS. Blockchain technology is noted for enhancing fault tolerance and security mechanisms in both Peer-to-Peer and Multi-DBMS architectures. However, specific discussions on availability mechanisms in Multi-DBMS architectures are limited.

Accessibility. In terms of accessibility, Client-Server architectures enable ubiquitous data access through distributed resource pools and location transparency. However, the complexity of setting up access for users is noted. Peer-to-peer accessibility discussions are limited, while Multi-DBMS architectures suggest benefits for data accessibility across different data models or storage locations. Distributed access control mechanisms are identified as potential enhancers for accessibility in Multi-DBMS architectures.

Data Analysis. The data analysis comparison reveals that Client-Server architectures are suitable for big data analysis due to scalability, parallel processing, and the ability to access data from multiple heterogeneous databases. Peer-to-peer architectures may require additional steps for locating and integrating data for analysis tasks. Multi-DBMS architectures offer significant advantages for data analysis tasks that involve integrating data from various sources with potentially diverse formats.

Volume 8, Issue 1, 2023
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

Strengths and Weaknesses Analysis

Client-Server. Strengths of the Client-Server architecture include scalability, improved data availability, fault tolerance, support for heterogeneous databases, partial data distribution, and optimized query processing. However, weaknesses include complexity in design and management, leading to potential challenges and increased response times.

Peer-to-peer. Peer-to-peer architectures exhibit strengths such as decentralization, security through blockchain integration, scalability, and efficiency in specific scenarios. Weaknesses include challenges in maintaining data consistency across distributed nodes, complexity in consensus mechanisms, and potential hindrances in large-scale adoption.

Multi-DBMS. The strengths of Multi-DBMS architectures lie in the integration of heterogeneous data sources, data modeling flexibility, unified data access and querying, and the potential for improved performance. Weaknesses include the complexity of data integration and management, the overhead of query optimization, and challenges in schema design and representation.

Suitability for Specific Scenarios

The analysis of suitability for specific scenarios further emphasizes the practical applications of each architecture:

Client-Server. Suited for cloud-based big data applications, scenarios requiring high data availability and fault tolerance, distributed applications with partial data distribution, and optimized query processing in distributed database systems.

Peer-to-Peer. Ideal for scenarios with limited data updates and a focus on data availability, secure data sharing, and collaboration using blockchain, high-throughput distributed computing applications requiring strong consistency, and specific scenarios requiring fast and secure P2P transactions.

Multi-DBMS. Well-suited for organizations with diverse data sources requiring integration, applications requiring seamless data integration from various sources, organizations managing data across diverse platforms and models, and scenarios potentially leveraging platform-specific functionalities.

CONCLUSION AND RECOMMENDATION

This literature review serves as a foundational exploration of DDBMS architectures, providing insights into their comparative performance, availability, accessibility, data analysis characteristics, as well as strengths and weaknesses. The identified factors influencing the suitability of each architecture for specific scenarios contribute to the understanding of practical applications in real-world contexts.

REFERENCES

Abdelhafiz, B. M. (2020, December 1). *Distributed Database Using Sharding Database Architecture*. IEEE Xplore. https://doi.org/10.1109/CSDE50874.2020.9411547

Ahmed, S., Atanu Shome, & Biswas, M. (2021). *DBST: A Scalable Peer-to-Peer Distributed Information System Supporting Multi-Attribute Range Query*. https://doi.org/10.1109/icsct53883.2021.9642540

Azizah, N., Hartajaya, V., & Riady, S. (2022). Comparison Of Replication Strategies On Distributed Database Systems. *International Journal of Cyber and IT Service Management*, *2*(1), 20–29. https://doi.org/10.34306/ijcitsm.v2i1.70

Esposito, C., Ficco, M., & Gupta, B. B. (2021). Blockchain-based authentication and authorization for smart city applications. *Information Processing & Management*, *58*(2), 102468. https://doi.org/10.1016/j.ipm.2020.102468

Ezechiel, K. K., Kant, S., & Agarwal, R. (2019). A synchronizer-mediator for lazy replicated

databases over a decentralized P2P architecture. *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, *Greater Noida, India, 2019*(pp. 199-213). https://doi.org/10.1109/icccis48478.2019.8974513

Fazlali, M., Eftekhar, S. M., Dehshibi, M. M., Malazi, Hadi Tabatabaee, & Nosrati, M. (2019). Raft Consensus Algorithm: an Effective Substitute for Paxos in High Throughput P2P-based Systems. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1911.01231

Forresi, C., Francia, M., Galinucci, E., & Golfarelli, M. (2021). Optimizing Execution Plans in a Multistore. *Lecture Notes in Computer Science*, 136–151. https://doi.org/10.1007/978-3-030-82472-3_11

Groppe, S., & Groppe, J. (2020). *Hybrid Multi-model Multi-platform (HM3P) Databases*. https://doi.org/10.5220/0009802401770184

Hamid, S. A., Abdulrahman, R. A., & Khamees, D. R. A. (2020). What is Client-Server System: Architecture, Issues and Challenge of Client - Server System (Review). *Recent Trends in Cloud Computing and Web Engineering*, *2*(1), 1–6. https://doi.org/10.5281/zenodo.3673071

Holubova, I., Contos, P., & Svoboda, M. (2021). *Multi-Model Data Modeling and Representation: State of the Art and Research Challenges*. https://doi.org/10.1145/3472163.3472267

Jinadu, O. T., Johnson, O. V., & Ganiyu, M. (2021). Distributed Database System Optimization for Improved Service Delivery in Mobile and Cloud BigData Applications. *International Journal of Computer Science and Mobile Computing*, *10*(9), 38–45. https://doi.org/10.47760/ijcsmc.2021.v10i09.004

Kyaw, P. M., & Aung, K. (2020). *Airline Reservation System of Replication Method Approach*. Meral.edu.mm. https://meral.edu.mm/records/4153

Lu, J., & Holubová, I. (2019). Multi-model Databases. *ACM Computing Surveys*, *52*(3), 1–38. https://doi.org/10.1145/3323214

Naik, S. (2019). Accessing Data From Multiple Heterogeneous Distributed Database Systems. *Advances in Computer and Electrical Engineering Book Series*, 192–219. https://doi.org/10.4018/978-1-5225-8295-3.ch008

Ojekudo, N., & Eze, C. (2021). Optimized Query Processing In Distributed Database System Using Hybrid Model. *International Journal of Advances in Engineering and Management (IJAEM)*, *3*(7), 4111. https://doi.org/10.35629/5252-030741114121

Orlunwo , P. O., & Prince, O. A. (2021). DISTRIBUTED DATABASE MANAGEMENT SYSTEM (DBMS) ARCHITECTURES AND DISTRIBUTED DATA INDEPENDENCE. *International Journal of Computer Science and Mobile Computing*, *10*(1), 23–48. https://doi.org/10.47760/ijcsmc.2021.v10i01.004

Ramesh, P., Devadas, A., Ray, P., Ramesh, S., Joshua, T., Priyan, V., Ramesh, M., & Rajasekaran, R. (2022). Under lock and key: Incorporation of blockchain technology in the field of ophthalmic artificial intelligence for big data management - A perfect match? *Indian Journal of Ophthalmology*, *70*(6), 2188. https://doi.org/10.4103/ijo.ijo_143_22

Sarode, R. P. (2021). *Secure Transaction Commitment in Peer- to-Peer (P2P) Processes*. https://u-aizu.repo.nii.ac.jp/record/226/files/d8202102.pdf

Zhang, C., & Lu, J. (2019). Holistic evaluation in multi-model databases benchmarking. *Distributed and Parallel Databases*. https://doi.org/10.1007/s10619-019-07279-6

PLEASE INCLUDE CONTACT INFORMATION:
NAME: JOSHUA CASTRO REYES
CONTACT NO: 09095959442
EMAIL ADDRESS: MEETJCREYES@GMAIL.COM