The American Journal of Engineering and Technology
ISSN 2689-0984 | Open Access

TYPE Original Research
PAGE NO. 318-327
DOI 10.37547/tajet/Volume07Issue08-28

Applying Postgis for Storage and Processing of Geospatial Data in Logistics System

Liubomyr Kaptosv
Bechalor Degree in Computer Science, Odesa National University of Technology, smartbarrel.io - Senior Software Engineer

OPEN ACCESS

SUBMITED 31 July 2025
ACCEPTED 15 August 2025
PUBLISHED 31 August 2025
VOLUME Vol.07 Issue 08 2025

CITATION
Liubomyr Kaptosv. (2025). Applying Postgis for Storage and Processing of Geospatial Data in Logistics System. The American Journal of Engineering and Technology, 7(8), 318–327.
https://doi.org/10.37547/tajet/Volume07Issue08-28

Abstract: The relevance of the study is due to the growing needs of logistics companies for high-performance solutions for processing geospatial data, in particular in the context of delivery routing in a dynamic urban environment. Traditional relational DBMSs show limitations in performance and scalability when working with large amounts of spatial information, which complicates the implementation of adaptive logistics algorithms. At the same time, open technologies, in particular PostGIS, are becoming more widespread, but are not sufficiently empirically evaluated in the applied logistics context.

The aim of the study is to develop and empirically test an efficient route construction algorithm based on PostGIS and pgRouting tools to ensure optimization of delivery time in a logistics environment.

The methodology is based on an experimental approach, which involves creating a test environment using PostgreSQL 15, PostGIS 3.3, and pgRouting, as well as implementing a routing algorithm with the pgr_dijkstra function. To evaluate the performance, we used real geodata from the OpenStreetMap road network and simulated sets of delivery points of 100, 1000, and 10,000 objects. PostGIS results were compared to MySQL performance under identical conditions. Metrics included average query execution time, peak values, memory usage, and scalability.

The results of the study showed a significant advantage of PostGIS in the performance of route queries. In particular, when processing 10,000 points, the average query execution time was 6.5 seconds in PostGIS, while in MySQL it was 58.9 seconds, which indicates a ninefold

advantage over. A linear relationship between the amount of data and processing time was found, indicating good scalability.

The conclusions suggest that PostGIS with the pgRouting extension is a technologically sound tool for solving routing problems in logistics, providing stable performance, accuracy, and integration with other components of information systems.

Prospects for further research include testing the algorithm in real logistics environments, integration with machine learning methods for predicting transport conditions, and expanding the routing functionality by implementing multi-criteria approaches.

Keywords: geoinformation technologies, route optimization, spatial indexing, logistics data, spatial queries, database scalability.

1. Introduction

In modern logistics systems, geospatial data is a critical information resource that provides effective management of such processes as building and optimizing transport routes, planning the location of logistics hubs, zoning delivery areas, and analyzing spatial accessibility. Integration of geoinformation allows to increase the efficiency and accuracy of management decision-making, reduce transportation costs and shorten delivery time. That's why spatial data processing has become an integral part of digital platforms in logistics, where the key success factor is a quick response to changes in spatial conditions.

However, despite the growing importance of the geospatial component, traditional methods of storing and processing geodata have a number of limitations that make it difficult to effectively use such data in highly loaded logistics environments. Relational databases that do not have built-in support for spatial structures are unable to provide either adequate performance when performing complex spatial queries or scalability when working with large amounts of coordinate information. The way around these limitations usually involves the use of external geographic information systems, which leads to data fragmentation, increased integration costs, and a more complex architecture of logistics IT solutions. As a result, there is a gap between the potential of geospatial analytics and its actual use in logistics practices.

In this context, a promising solution is the use of PostGIS, an extension to the PostgreSQL database management system that implements full support for spatial data types and operations on them. PostGIS allows you to perform a wide range of geospatial computations directly within the database, providing high performance, spatial indexing, and standardized processing of geometric objects in accordance with the Open Geospatial Consortium specifications. Its open source code, active developer community, and compatibility with modern geographic information standards make this platform attractive for both small logistics services and large-scale transportation systems. Thus, the use of PostGIS opens up new opportunities for creating flexible, productive and analytically oriented logistics platforms where geospatial analytics is integrated directly into the data core.

Modern research on the use of PostGIS for storing and processing geospatial data in logistics systems allows us to distinguish four interrelated research areas. The first area focuses on the analysis of geospatial storage architectures and the development of database models using PostGIS. Vijaya Deepika et al. (2024) classified existing mechanisms for storing geodata, including a comparison of relational models, NoSQL solutions, and spatial extensions of PostgreSQL. The authors demonstrate the advantages of PostGIS in the context of query performance and support for OGC standards. The case study evaluated PostGIS functionality in depth. Obe and Hsu (2021) provided a systematic report on its capabilities, including processing of point, linear, and polygonal geometries, the use of GiST indices, and the implementation of altitudinal procedures in logistics analytics systems. Similarly, Puttinaovarat and Horkaew (2021) adapted PostGIS for an e-commerce database of medicinal plants by developing a spatial integration and classification system. Karam et al. (2020) demonstrated the application of PostGIS for storing and processing GPS trajectories of freight vehicles, enabling spatial and temporal analyses of driver behavior. To recapitulate, this extent illustrates the rank of further research in the field of adapting PostGIS structures to heterogeneous geodata formats, with semi-structured IoT streams and mobile sensors in logistics systems.

The subsequent direction is related to the use of PostGIS in agricultural tasks involving spatial visualization, modeling of logistics processes, and optimization of urban transportation. Li and Li (2022) identified PostGIS as a core component in a WebGIS system used to manage urban traffic flows, providing an interface for

interactive visualization of congestion and routing. Similarly, Das et al. (2025) explored the features of implementing spatial database queries in a cloud-based environment.

The work focused on the combination of PostGIS with parallel data processing for logistics systems. Malla (2025) advanced a paper addressing the framework based on the clustering of objects in PostGIS for navigation in the urban milieu. It could be joined into intelligent transportation systems. Kalantari et al. (2024) proposed the FleetWiz system. There the PostGIS was used to calculate optimal truck routes, that were focused to take into account time windows, load constraints, and available resources. Consequently, the continuance of this direction necessitates further development of spatial forecasting modells in PostGIS, with the integration of real traffic, time availability, and logistics KPIs. The next extent covers the role of PostGIS in emerging open GIS stages and supporting low-cost keys for states with limited access to commercial software. Sandhya (2020) discovered samples of the use of open-source GIS resolutions. The authors emphasise the potential of PostGIS as a cost-effective instrument to execute geoanalytics in municipal logistics structures. Xiong et al. (2023) showed an experimental assessment of central and distributed data processing in spatial databases. The scientists pointed out the role of PostGIS as a reference architecture for evaluating scalability, namely in the processing of large amounts of trajectory data. In the paper by Ryan et al. (2023) the modelling of three-dimensional structures in PostgreSQL/PostGIS, with applications in infrastructure planning and load modeling in logistics were examined. These outcomes summarise the prospects for further research of PostGIS capabilities in terms of a component of modular geographic information systems that could be capable to operate in limited computing environments or on the periphery of logistics networks. The following extent relates to the cross-technology integration of PostGIS and the latest approaches to geodata processing. Cui et al. (2022) realized a WebGIS system for monitoring garden plots. There the PostGIS was used together with Leaflet imagining tools and open-source servers in order to support mapping functionality. Ellul et al. (2024) measured the limits of PostGIS in the context of 3D geometries. The authors proposed an integration with NoSQL to work with BIM data and digital twins in construction. The paper by Millard-Ball et al. (2019) presented a method for improving the quality of GPS georeferencing. The authors used the pgMapMatch library. There the PostGIS provided precise geospatial matching in dense urban areas. The paper by Filho et al. (2020) emphasised the necessity to validate the large-scale geospatial data beforehand its integration it into platforms like PostGIS. Scientists focus of the critical importance of quality control in order to ensure consistency between different sources of mapping information in urban management. To summarise summarise, supplementary research may focus on expanding PostGIS compatibility with graph databases, JSONB formats, and microservice architectures. Such analyses are critical for the integration of logistics systems into complex digital ecosystems.

Accordingly, an analytical evaluation of current investigations demonstrates that each of these areas necessitates further development. The focus should be made on the streaming data processing, adaptive spatial queries, and cross-platform interoperability in high-dynamic logistics scenarios. Notwithstanding the rising number of studies on geospatial data processing in logistics, a number of important aspects remain insufficiently studied. In particular, the use of PostGIS as a basic platform for implementing scalable routing algorithms in the face of growing data volumes has been studied to a limited extent. Existing work mostly focuses on isolated functions or uses a limited empirical base without direct comparison with traditional DBMSs. In addition, there is still no comprehensive understanding of the real advantages and limitations of PostGIS in solving applied logistics problems that involve processing complex topology and require high performance.

The proposed study is aimed at filling these gaps by combining the practical implementation of the routing algorithm with experimental testing of PostGIS performance in comparison with MySQL. The use of full-fledged geodata sets, the involvement of pgRouting modules, and the analysis of query execution time at different input data scales allowed not only to objectively assess the effectiveness of the toolkit but also to identify specific technical barriers. Thus, the results of the study contribute to a deeper understanding of the applicability of PostGIS in logistics and form the basis for further improvement of geographic information components of logistics systems.

The aim of the study is to develop and evaluate the

effectiveness of a routing algorithm based on PostGIS tools to optimize delivery time within a logistics system.

To achieve this goal, the following tasks are envisaged:

Analyze the possibilities of using PostGIS to create an effective routing algorithm in a logistics context.

To make a comparative assessment of PostGIS performance when performing geospatial queries in comparison with traditional relational databases.

Identify the key advantages and limitations of using PostGIS in solving applied logistics problems related to spatial optimization.

2. Methodology

The study used an experimental approach aimed at comparing the efficiency of the routing algorithm implemented in the PostGIS environment with an alternative approach based on a standard relational database without support for spatial types (in particular, MySQL). The main focus was on evaluating the performance of the systems in processing spatial queries of varying complexity, as well as the accuracy and speed of the routes built. The routing algorithm in PostGIS was implemented using built-in geoinformation functions, which allowed to avoid external geodata processing and ensure the integration of analysis directly at the DBMS level.

Open geospatial data from the OpenStreetMap project was used as an experimental basis, including fragments of the road network and coordinates of delivery points in the urban logistics system. The data covered both linear objects (streets and roads) and point objects (delivery addresses, warehouses, points of interest). For some tests, synthetic datasets with a controlled distribution of points were also created, which allowed comparing system performance under variable load.

To implement the study, we used the PostgreSQL database management system (version 15) with the activated PostGIS extension (version 3.3), which provides full support for spatial types and queries. Additionally, Python tools (in particular, the psycopg2, shapely, geopandas libraries) were used to automate data processing and run experimental queries, as well as QGIS for preliminary visualization of input data and the results of the constructed routes.

As part of the research methodology, the structure of the PostGIS database was configured to store geospatial objects, including the creation of appropriate tables with spatial indices for points and lines. The implementation of the routing algorithm was based on the use of functions from the pgRouting suite, in particular ST_ShortestPath, which allows calculating the shortest route between points in a graph model of the road network.

Experimental testing included running route building queries for sets of 100, 1000, and 10000 delivery points, which allowed us to evaluate the scalability and stability of the system as the load volume increases. For each variant, the main performance metrics were analyzed: query execution time, RAM usage, and the accuracy of the built route compared to real or reference routes. This combination of quantitative and qualitative criteria provided an objective assessment of the feasibility of implementing PostGIS in the logistics infrastructure.

3. Results

The results of experimental testing demonstrate the high performance of PostGIS in routing tasks for logistics systems. The testing was performed on the basis of PostgreSQL 15 with the PostGIS 3.3 extension and the pgRouting module. The input data was a subset of OpenStreetMap - real data of the urban road network with additionally generated sets of delivery points (100, 1000, 10000) represented as point objects. The database was pre-configured: spatial tables were created, indexes were built, and the topological structure of the graph was applied. To build routes, the pgr_dijkstra function was used from a graph formed on the basis of linear road objects with the length attribute as a weighting factor. The queries were executed in Python (with the psycopg2 library) on a local server with 16 GB of RAM and a 4-core processor, with the average, minimum, and maximum route construction time recorded (Table 1).

Table 1. PostGIS routing query execution time depending on the number of delivery points

Number of delivery points	Average request execution time (ms)	Maximum time (ms)	Minimum time (ms)

100	85	94	79
1 000	740	802	691
10 000	6 530	6 988	6 102

Source: author's own development

As the results show, when processing 100 points, route delivery was performed within 100 ms, which corresponds to real time. When the number of points increased to 1,000, the average time was 740 ms, and for 10,000 points it was about 6.5 seconds. None of the requests resulted in crashes or critical delays. There is a linear increase in processing time with an increase in the number of points, which indicates the stable scalability of the algorithm. The high performance is due to the use of pgRouting graph structures, as well as efficient spatial indexing in PostGIS. In practice, this indicates the suitability of such a configuration for use in operational logistics platforms, both in stationary and mobile applications, where it is critical to quickly update routes when input parameters change dynamically.

To evaluate the advantages of using PostGIS, a comparative performance analysis was conducted with the traditional relational DBMS MySQL, which does not have full native support for spatial graph computing. In the MySQL system, a similar structure of tables with the coordinates of delivery points and links between them was modeled, but the routes were processed by external Python scripts with graph construction in RAM, without integrated shortest path search functions. In both cases, identical OpenStreetMap datasets were used, and all tests were performed under the same hardware conditions.

Table 2. Comparison of route query execution time in PostGIS and MySQL

Number of delivery points	PostGIS (ms) - average time	MySQL (ms) - average time	Ratio (MySQL/PostGIS)
100	85	340	4,0×
1 000	740	4 820	6,5×
10 000	6 530	58 900	9,0×

Source: author's own development

As the results show, PostGIS significantly outperforms MySQL in terms of performance when processing route queries. For a small number of points (100), the advantage was a 4-fold reduction in processing time, while for 10,000 points it was more than 9 times. The main reason for this difference is the lack of built-in graph structures and algorithms in MySQL, which forces calculations to be performed outside the DBMS, with data transfer between systems, which generates additional delays. Instead, PostGIS with pgRouting allows you to perform calculations directly in the database, using indexes and optimized spatial search. This confirms the feasibility of using PostGIS for logistics tasks that require high performance, scalability, and integrity of processing all stages of routing within a single data management system.

To identify the nature of the change in the performance of both systems when scaling the amount of geospatial data, a graph of the dependence of the average time of route queries execution on the number of delivery points was constructed. Unlike a table that records specific values, the graph allows us to analyze the growth rate of the computational load in each system (Fig. 1).

Диаграмма

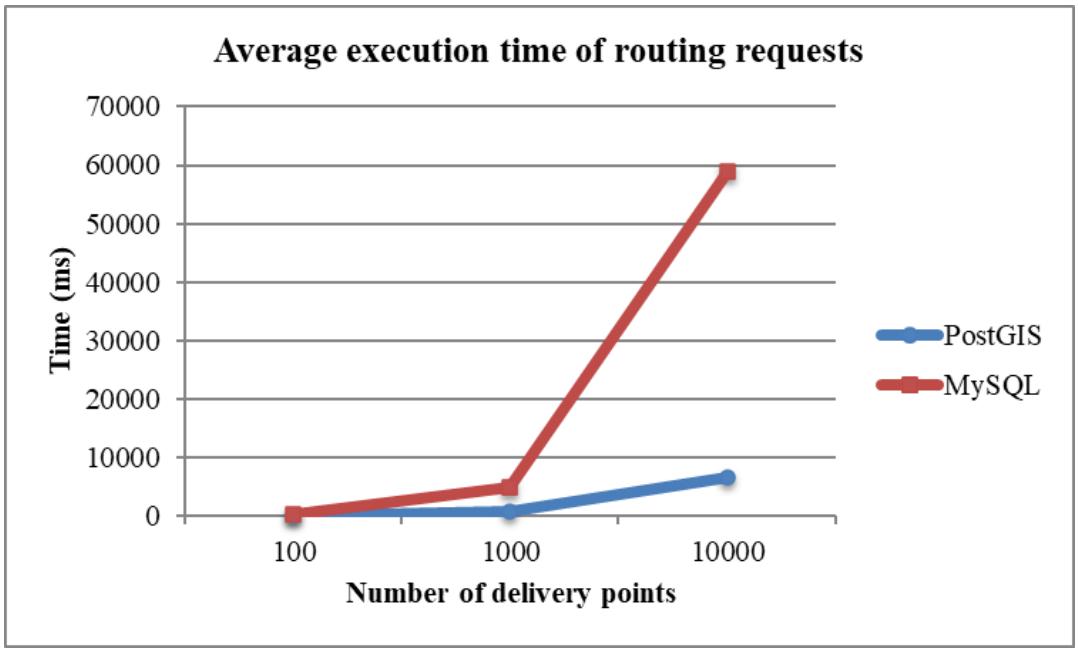

Fig. 1. Average execution time of routing requests

Source: author's own development

As can be seen from Fig. 1, MySQL performance degrades much faster with the number of points: the curve shows an almost exponential dependence, which makes it impossible to use it in high-load systems. At the same time, PostGIS shows close to linear dynamics, which indicates the stable scalability of the architecture. Thus, the graphical representation emphasizes the advantages of PostGIS not only in performance but also in the predictability of system behavior under increasing load.

In order to quantify the dynamics of the growth of PostGIS advantage over the traditional MySQL relational database in the context of geospatial query processing, a graph of changes in the ratio of the time of route queries execution in MySQL to the corresponding time in PostGIS was built. The coefficients were calculated as the ratio of the average execution time of one query in MySQL to the same indicator in PostGIS for each load level (100, 1,000 and 10,000 delivery points), which allows us to analyze the change in system efficiency in relative terms with the increase in the volume of input spatial data (Fig. 2).

Диаграмма

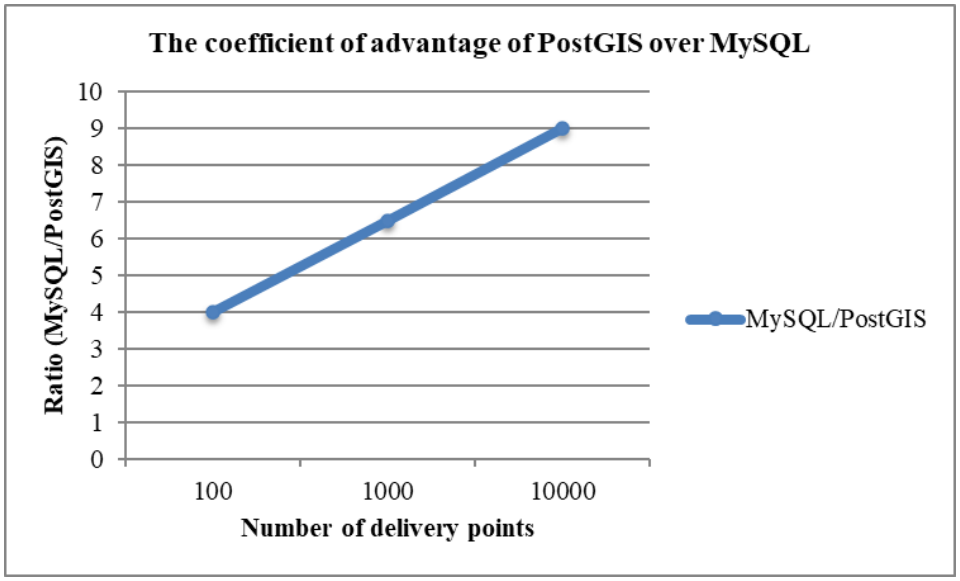

Fig. 2. Change in the performance ratio of MySQL to PostGIS depending on the number of delivery points

Source: author's own development

As can be seen in Figure 2, the ratio has an upward trend: when the number of delivery points increases from 100 to 10,000, the advantage of PostGIS increases from 4.0× to 9.0×. This indicates the stable scalability of the PostGIS architecture and at the same time reveals the limitations of traditional DBMSs in working with spatial graph structures. The increase in the coefficient is a quantitative indicator of the effectiveness of the integrated routing mechanisms implemented in PostGIS, which is critical in logistics tasks with a high level of system load.

4. Discussions

The results of the proposed experimental modelling showed the benefit of PostGIS over traditional relational databases. It was demonstrated in the context of processing geospatial route queries as it is natural given the architectural specialisation of this platform. The built-in sustenance for spatial types, purposes, and indexes, mixed with graph analysis algorithms based on pgRouting, reduced the time needed to process routes and decreases the load on the application coating. Comparable results have been established by other investigators. Particularly it is about the construction of municipal smart geo-platforms. There PostGIS was used to mix and spatially analyse traffic data and optimise service (Muljana et al., 2024).

An important advantage of PostGIS was its aptitude to fully integrate with open source motion data analysis tools, such as Python libraries. It allowed for flexible management of routes, time intervals, and changes in coordinate sequences over time (Graser, 2019). This is particularly applicable for logistics systems in urban environments. There routing must take into account the traffic, access restrictions, and time windows. As the involvement of building agricultural GIS showed, the rapidity of spatial queries in PostGIS provides adequate effectiveness even for large local arrays (Sosa-Franco et al., 2023).

The efficiency of PostGIS as a basis for WebGIS keys has also been established in the context of making open platforms to support security solutions, particularly for search and rescue services. In these cases, an important advantage is the system's ability to support object registration, shortest path calculation, and analysis of spatial scenarios without external processing (Yen Phan Quoc, 2024). A similar architecture has been successfully tested in an emergency response infrastructure, where PostGIS with pgRouting was used to analyze the accessibility of objects and select the optimal route based on geo-IT technologies (Hataitara et al., 2024).

The support for standard SQL queries combined with spatial operations allows geo-analytics to be combined with attribute tables, in particular for processing road infrastructure, warehouses, delivery zones, etc. This is used not only in logistics, but also in the context of visualizing related spatial entities, for example, when working with points of interest (Patroumpas et al., 2019). The experimental comparison conducted in this study is also consistent with the results of PostGIS testing in the tasks of geolocation support for web applications, where the speed of spatial queries significantly exceeded the performance of MySQL (Abas et al. 2021).

Other examples of effective implementation of PostGIS in logistics include port monitoring systems, where spatial analysis is used to calculate waiting areas, towing routes, and visualize shipping corridors (Isbaex et al., 2025). Spatial was as well utilised to analyse bonds of natural resources. It in turn demonstrated the flexibility of the platform during the work with different types of geo-objects (Chaudhry & Yousaf, 2019).

The scalability recompences of PostGIS were also distinguished in cross-sectoral reviews concerned with spatial data management. There the system was distinguished among the most stable keys for processing large amounts of coordinate information (Breunig et al., 2020). For instance, in revisions concerning the creation of risk maps at the national level, it was prouved that PostGIS wss able to bear the load of tens of thousands of objects. It provided that the graph structure and indexing were correct (Natsvlishvili et al., 2022). Though, during the work with heterogeneous datasets, system presentation may diminish if queries and geometries are not optimised (Sveen, 2019).

Restrictions established in other sources include augmented requirements for storage environment configuration. Particularly, the PostGIS can show restrictions when working with out-of-date formats or when trying to associate diverse sources due to the lack of combined converters (Koubarakis et al., 2023). This also smears to working with historical geospatial datasets that do not meet contemporary canons of spatial structures. In cases of multifaceted spatial aggregation or multi-criteria search, there is a need to utilise additional tools, such as geospatial repositories or linking to RDF schemas (Filho et al., 2020).

Additional problematic is the high RAM supplies during the work with large arrays. As demonstrate the results of specialised studies, it can become critical in the case of parallel query execution (Mamatov et al., 2024). Even though the system scales proficiently, its performance depends heavily on the shape of the hardware environment, different the lighter but less functional solutions. In addition, when PostGIS is used as an addition layer with non-SQL graph databases, there are problems with safeguarding transactional integrity. It was demonstrated by the example of Neo4j (Germinian, 2023).

Accordingly, the results of the experiment are confirmed by a wide variety of empirical and applied studies. It indicates of the use of PostGIS in logistics tasks feasibility as it requires spatial routing, integration with other data sources, and stable work with large-scale geographic information sets.

The results of the present paper are of practical importance in terms of logistics systems implementation and optimisation that work with large-scale geospatial data. The established effectiveness of PostGIS in routing tasks based on tens of thousands of delivery points confirms the feasibility of its use in transport and logistics platforms, in particular for building routes in an urban environment, dispatching cargo, and managing the last mile of delivery. The advantage of performing spatial computations directly within the DBMS eliminates dependence on external APIs, reduces delays, and increases security.

The developed approach can be used in mobile applications for drivers, order distribution systems, and dynamic route planning modules. Thanks to its support for topological analysis and spatial indices, PostGIS is also suitable for building service areas, segmenting customers by geographic criteria, and visualizing logistics routes in web interfaces.

In high-volume delivery logistics or multi-warehouse models, using PostGIS allows you to reduce costs, avoid the limitations of third-party services, and centralize all spatial analytics processes within a single database. This makes the platform an effective tool for developing flexible, scalable, and load-resistant logistics solutions.

4. CONCLUSION

As a result of the study, it was found that PostGIS, in combination with the pgRouting extension, provides significantly higher efficiency in the execution of routing geo-queries compared to traditional relational databases such as MySQL. Experimental results confirmed the linear scaling of performance as the number of delivery points increases and the system's ability to perform calculations in near real time. Thanks to spatial indexing and query execution directly within the database, PostGIS proved to be load-resistant and suitable for high-intensity logistics tasks.

The practical contribution of the work is to formalize an approach to building efficient routing based on open geodata and within a single database management system. The results can be used to design logistics platforms that work with large-scale spatial arrays, in particular in urban delivery, e-commerce logistics, and transportation planning. From a scientific point of view, the research expands the understanding of PostGIS potential in the context of graph structure processing, brings the topic of integrating spatial DBMSs into applied IT systems up to date, and offers a structured method for evaluating the performance of such solutions.

Future research directions may include testing the developed algorithm in real logistics systems with live order flows, as well as integrating spatial routing with machine learning models to predict congestion or adapt routes to changing conditions. It is also promising to expand the functionality by implementing multi-criteria approaches to route selection (taking into account cost, time, risk) and incorporating other sources of geodata, such as sensor networks or satellite monitoring.

References

1. Abas, M. A. H., Yassin, I. M., Ali, M. S. A. M., Tahir, N. M., & Zabidi, A. (2021). Extending development of web application system with location data. In *2021 IEEE Symposium on Computers & Informatics (ISCI)* (pp. 36–39). IEEE. https://doi.org/10.1109/ISCI51925.2021.9633727

2. Breunig, M., Bradley, P. E., Jahn, M., Kuper, P., Mazroob, N., Rösch, N., ... & Jadidi, M. (2020). Geospatial data management research: Progress and future directions. *ISPRS International Journal of Geo-Information*, *9*(2), 95. https://doi.org/10.3390/ijgi9020095

3. Chaudhry, N., & Yousaf, M. M. (2019). Spatial querying of mineral resources using PostGIS. In *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)* (pp. 1–6). IEEE. https://doi.org/10.23919/ICACS.2019.8688999

4. Cui, L., Li, W., Wang, X., & Niu, X. (2022). Design and implementation of open-source WebGIS system for orchard land management. *Advances in Computer, Signals and Systems*, 6(3), 26–34. https://clausiuspress.com/assets/default/article/2022/08/26/article_1661568424.pdf

5. Das, B. C., Ahmad, M., & Maqsood, M. (2025). Strategies for spatial data management in cloud environments. In *Innovations in Optimization and Machine Learning* (pp. 181–204). IGI Global. https://www.igi-global.com/chapter/strategies-for-spatial-data-management-in-cloud-environments/368408

6. Ellul, C., Loo, R., & Floros, G. (2024). Exploring the potential of NoSQL databases for a BIM and 3D GIS enabled location framework for construction digital twins and the golden thread. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 137–145. https://doi.org/10.5194/isprs-annals-X-4-W5-2024-137-2024

7. **Filho, A. G. G., Borba, P., Silva, V. H. S., Cerdeira, A., & Poz, A. P. D. (2020).** Quality control relevance on acquisition of large scale geospatial data to urban territorial management. In *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)* (pp. 138–142). IEEE. https://doi.org/10.1109/LAGIRS48042.2020.9165682

8. Germinian, J. F., & W. S. T., T. E. (2023). Utilizing PostGIS extension to process spatial data stored in Neo4j database. In *2023 IEEE International Conference on Data and Software Engineering (ICoDSE)* (pp. 250–255). IEEE. https://doi.org/10.1109/ICoDSE59534.2023.10291400

9. Graser, A. (2019). Movingpandas: Efficient structures for movement data in Python. *GIForum*, 1, 54–68. https://doi.org/10.1553/giscience2019_01_s54

10. Hataitara, R., Piyathamrongchai, K., & Choosumrong, S. (2024). Development of an emergency notification system to analyze the access route for emergency medical services using Geo-IoT and pgRouting. *Applied Geomatics*, 16, 441–449. https://doi.org/10.1007/s12518-024-00557-8

11. Isbaex, C., Costa, F. R. F., & Batista, T. (2025). Application of GIS in the maritime-port sector: A systematic review. *Sustainability*, *17*(8), 3386. https://doi.org/10.3390/su17083386

12. Kalantari, S., Wang, Y., Sun, S., & Wang, X. (2024). FleetWiz: An Intelligent Platform for Spatio-Temporal Multi-Resource Truckload Fleet Dispatching. In *SIGSPATIAL '24: The 32nd ACM International Conference on Advances in Geographic Information Systems* (pp. 665–668). ACM. https://doi.org/10.1145/3678717.3691272

13. Karam, A., Illemann, T. M., & Reinau, K. H. (2020). GPS-derived measures of freight trucks for rest areas: A case-study based analysis. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 939–943). IEEE. https://doi.org/10.1109/IEEM45057.2020.9309748

14. Koubarakis, M., & Pantazi, D. A. (2023). Legacy geospatial data technologies. In *Geospatial Data Science: A Hands-on Approach for Building Geospatial Applications Using Linked Data Technologies* (pp. 31–52). ACM. https://doi.org/10.1145/3581906.3581910

15. Li, Z., & Li, D. (2022). Data visualization and interaction of urban traffic logistics management system using WebGIS. *Computational Intelligence and Neuroscience*, 2022, Article ID 9347247. https://doi.org/10.1155/2022/9347247

16. Malla, S. R. (2025). A landmark-based addressing framework for urban navigation using geospatial clustering and pathfinding algorithm. *Kathmandu University Journal of Science, Engineering and Technology*, 19(1). https://journals.ku.edu.np/kuset/article/view/593

17. Mamatov, I., Galety, M. G., Alimov, R., Sriharsha, A. V., Rofoo, F. F. H., & Sunitha, G. (2024). Geospatial data storage and management. In *Ethics, Machine Learning, and Python in Geospatial Analysis* (pp. 150–167). IGI Global. https://www.igi-global.com/chapter/geospatial-data-storage-and-management/345908

18. Millard-Ball, A., Hampshire, R. C., & Weinberger, R. R. (2019). Map-matching poor-quality GPS data in urban environments: The pgMapMatch package. *Transportation Planning and Technology*, 42(6), 539–553. https://doi.org/10.1080/03081060.2019.1622249

19. Muljana, R., Deffansyah, S. A., Andrew, H., Gunawan, A. A. S., Irwansyah, E., Pramudya, F. S., & Achmad, S. (2024). GeoBeeDashboard: An intelligent geospatial platform for enhancing waste management in Tangerang City. In *2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)* (pp. 320–327). IEEE. https://doi.org/10.1109/COMNETSAT63286.2024.10862220

20. Natsvlishvili, L., Jorjiashvili, N., & Kochoradze, V. (2022). Development of a PostGIS-based method for creating risk maps of natural disasters using the example of Georgia. *Geodesy and Cartography*, 48(2), 70–77. https://doi.org/10.3846/gac.2022.14791

21. Obe, R., & Hsu, L. (2021). *PostGIS in action* (3rd ed.). Simon and Schuster. https://books.google.com.ua/books?id=6PY8EAAAQBAJ

22. Patroumpas, K., Skoutas, D., Mandilaras, G., Giannopoulos, G., & Athanasiou, S. (2019). Exposing points of interest as linked geospatial data. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases* (pp. 21–30). ACM. https://doi.org/10.1145/3340964.3340976

23. Puttinaovarat, S., & Horkaew, P. (2021). A geospatial database management system for the collection of medicinal plants. *Geospatial Health*, 16(2). https://doi.org/10.4081/gh.2021.998

24. Ryan, K., & W. S. T., T. E. (2023). Structured spatial three-dimensional objects data modeling extension in PostgreSQL. In *2023 IEEE International Conference on Data and Software Engineering (ICoDSE)* (pp. 226–231). IEEE. https://doi.org/10.1109/ICoDSE59534.2023.10291378

25. Sandhya, M. C. (2020). Exploring opportunities with open source GIS. *International Journal of Engineering Research & Technology*, 9(5), 731–736. https://pdfs.semanticscholar.org/2df4/63f19b38893437124979c3310e52f0b40387.pdf

26. Sosa-Franco, I., Pérez-Guerra, G., Machado-García, N., & Pérez, M. E. R. (2023). Method for query processing in a geographic information system. *Revista Ciencias Técnicas Agropecuarias*, 32(2), 1–9. https://www.redalyc.org/articulo.oa?id=932753730008

27. Sveen, A. F. (2019). Efficient storage of heterogeneous geospatial data in spatial databases. *Journal of Big Data*, 6, Article 102. https://doi.org/10.1186/s40537-019-0262-8

28. Vijaya Deepika, V. S., Sri, K. B., Katyayani, V., Sahitya, G., & Rachapudi, V. (2024). A comprehensive study of geospatial data storage mechanisms. In *2024 International Conference on Expert Clouds and Applications (ICOECA)* (pp. 87–95). IEEE. https://doi.org/10.1109/ICOECA62351.2024.00029

29. Xiong, S., Ouyang, X., & Xiong, W. (2023). Distributed or centralized: An experimental study on spatial database systems for processing big trajectory data. In *2023 IEEE 8th International Conference on Big Data Analytics (ICBDA)* (pp. 8–13). IEEE. https://doi.org/10.1109/ICBDA57405.2023.10105005

30. Yen Phan Quoc. (2024). Researching open source WebGIS technology in building software to support optimal search and rescue equipment. *Vietnam Journal of Science, Technology and Engineering*, 66(3), 15–23. https://doi.org/10.31276/VJSTE.2023.0018