Relational and Graph Database Integration for Advanced Supply Chain Analytics

Shreyasi Nath
A69034237
s1nath@ucsd.edu

Bhaavya Naharas
A69034420
bnahars@ucsd.edu

Varadraj Bartakke
A69034420
vbartakke@ucsd.edu

March 21, 2025

Abstract

This project demonstrates an integration of relational and graph database management systems, specifically PostgreSQL and Neo4j, to improve supply chain analytics. Using PostgreSQL for structured data management, followed by seamless data migration to Neo4j, the project effectively addresses the inherent complexities of supply chain data analysis. Production, food balances, trade, and pricing records are taken from FAOSTAT and carefully cleaned, standardized, and converted into time series formats for in-depth research. Through iterative graph data modeling, the project transitions from an attribute-centric to a relationship-driven approach, significantly increasing analytical precision and query performance. In addition, Cypher queries were implemented to perform detailed supply chain analyzes, uncovering vital information on trade dynamics, risk assessment, and key performance metrics. Using NeoDash for interactive data visualization, the project provides dynamic dashboards that facilitate immediate comprehension and strategic decision making. This project not only improves operational efficiency and supply chain analysis, but also lays the groundwork for future enhancements such as machine learning integration and expanded data coverage for modern supply chain optimization. Overall, the integration of database technologies, advanced analytics, and visualization tools presented in this work offers a practical framework for supply chain applications and technical insights for optimizing decision-making processes in complex chain environments.

1 Literature Review

Sl.No	Model/ Project Name	Year	Review
1.	Graph Database to Enhance Supply Chain Resilience for Industry 4.0 (Hong and Chen)	2021	This paper suggested a framework for graph databases intended to improve supply chain resilience in Industry 4.0 settings. placed a strong emphasis on using Time-to-Stockout (TTS) measurements in graph structures to dynamically manage demand and inventory relationships. shown effectiveness in both analytical insights and data searching.
2.	Supply Chain Graph Database Use Cases (Neo4j)	2022	This paper highlighted real-world applications of graph databases in supply chains, demonstrating the special capacity of graph models to offer end-to-end visibility. displayed vulnerability visualization and logistic flow optimization, highlighting supply chain networks' resilience and agility.
3.	The Game-Changing Role of Graph Technology in Supply Chain (SupplyChainBrain)	2023	This paper examined how supply chain management might be revolutionized by graph databases, with a focus on the shift from static to dynamic relationship-based analytics. illustrated how graph analytics may be used to find hidden connections, facilitate proactive risk management, and improve competitive advantage.
4.	Inventory Management Optimization with Data Analytics for a Trading Company	2023	This study focuses on employing data analytics to improve inventory management for a trading organization. It describes techniques to increase operational effectiveness and forecasting accuracy, offering practical insights that lower inventory costs and promote better supply chain performance decision-making.
5.	Building Resilience in Supply Chains: A Knowledge Graph-Based Risk Management Framework	2024	This paper presents a paradigm for managing risks and modeling intricate supply chain interdependency using knowledge graphs. The approach, which was created in reaction to COVID-19 disruptions, makes it possible to proactively identify and mitigate vulnerabilities, improving supply chains' overall robustness and resilience.

Table 1: Literature Review for Supply Chain Analsyis

2 Introduction

Effective inventory management is a crucial aspect of modern supply chain operations, ensuring the optimal balance between supply and demand. With the increasing complexity of global trade networks, businesses require data-driven approaches to track inventory, forecast trends, and optimize resource allocation.

This project focuses on using relational databases (PostgreSQL) and graph-based (Neo4j) to improve inventory management analysis. By integrating structured data such as pricing, trade volumes, and food balances with graph-based relationships, our approach provides deeper insights into supply chain dynamics.

The system is designed to address key challenges such as data integration, trade flow analysis, and inventory forecasting, offering a comprehensive view of commodity movements. Using FAOSTAT datasets, our model captures real-world trade relationships and pricing trends, enabling better decision-making for businesses and policymakers.

In addition, we implement interactive dashboards in NeoDash to visualize trade patterns, detect potential bottlenecks, and support real-time analytics. This report details our methodology, data modeling, analytics, and future enhancements, showcasing how an integrated database architecture can drive more efficient and intelligent inventory management solutions.

3 Application

The application is a Supply Chain Analysis system designed to provide a comprehensive, data-driven view of trade, relationships and market trends. It leverages a dual-database architecture that integrates a Relational Database Management System (RDBMS) using PostgreSQL and a Graph Database Management System (GDBMS) using Neo4j.

3.1 Concept

- **Future-Ready Platform:** The platform is designed with a forward-thinking approach that anticipates the evolving nature of supply chains. By incorporating machine learning techniques, the system can perform advanced demand forecasting, allowing organizations to predict and prepare for future shortages or surpluses. Furthermore, real-time data ingestion capabilities enable the platform to keep pace with rapidly changing market conditions.

- **Scalability Flexibility:** As new datasets such as population metrics or macroeconomic indicators become relevant, the platform can seamlessly integrate them. This modular approach means that the analytical capabilities can be expanded without the need for a complete architectural overhaul. The ability to incorporate additional data sources allows the system to evolve alongside business requirements, providing increasingly comprehensive insights into supply chain operations.

- **Complex Supply Chain Environment:** Managing a dispersed network demands a holistic view of inventory levels, trade relationships, and market trends. The platform addresses this complexity by consolidating diverse datasets and providing a unified view of the supply chain. This perspective is essential for effective inventory management, risk mitigation, and ensuring that each stakeholder's role is optimized within the broader network.

- **Dual-Database Architecture – Relational DBMS (PostgreSQL):** The system's dual-database architecture leverages a relational database management system, PostgreSQL, to store structured inventory data such as products, warehouses, and transaction logs. Utilizing well-defined tables, PostgreSQL ensures efficient querying, robust data integrity, and streamlined reporting, making it ideal for managing large volumes of structured data where relationships are clearly defined and require normalization.

- **Dual-Database Architecture – Graph DBMS (Neo4j):** Complementing the relational database, the system employs a graph database management system, Neo4j, to model the intricate relationships between warehouses, suppliers, and retailers. This graph-based approach offers deep insights into network connectivity, enabling the identification of potential bottlenecks and critical links within the supply chain. By visualizing and analyzing these relationships, the system can reveal alternative supply routes and pinpoint areas where operational delays might occur.

- **Decision-Making Analytics:** The platform enhances strategic decision-making through operational analytics. Graph-based queries enable the identification of alternative routes and highlight key intermediaries within the supply chain, aiding both immediate logistical problem-solving and the development of more resilient, efficient strategies over time.

- **Industry Relevance:** For food security, organizations monitoring global commodity balances can rely on the system for critical insights. In the logistics and shipping industry, providers can optimize shipping routes and manage trade flows through detailed analytics. Moreover, in competitive markets, the system supports robust decision-making with access to both real-time and historical data, ensuring that organizations maintain a competitive edge.

- **Integrated Data Sources:** A key component of the platform's strength is its integration of multiple, high-quality data sources. The system leverages FAOSTAT datasets that focus on food balances, pricing, and trade to capture real-world data on commodity flows, cost fluctuations, and global trade patterns. This comprehensive data integration creates a solid analytical foundation, enabling both detailed analysis and high-level strategic planning by providing clear insights into how goods move across borders and how market conditions impact supply chain operations.

3.2 Achievement

The implementation of this inventory management system has led to significant advancements in data analysis, decision-making, and supply chain optimization. One of the most vital achievements is the ability to provide real-time decision support, to respond to supply chain fluctuations with timely insights. The system leverages historical and real-time data to facilitate strategic inventory planning, helping businesses maintain optimal stock levels and prevent shortages or excess inventory.

Another major accomplishment is the enhanced visibility into supply chain operations. By integrating relational and graph-based databases, the system provides a distinct view of trade relationships, product movements, and price fluctuations. This capability allows organizations to analyze trade dependencies, detect inefficiencies, and optimize procurement strategies. The ability to visualize these relationships through an interactive dashboard ensures that complex datasets are transformed into actionable intelligence.

Furthermore, the system improves resource allocation by analyzing trade patterns and identifying key nodes within the supply chain. By understanding which trade routes are most efficient and where bottlenecks may occur, companies can reduce transportation costs, streamline distribution networks, and improve overall supply chain resilience. This optimization is particularly valuable for industries dealing with perishable goods, where timely deliveries are critical.

Comparative analytics has been successfully integrated into the system, allowing for the analysis of price fluctuations and demand variations. In future we aim to apply statistical models to historical data, by which the system can forecast future trends, enabling proactive decision making. Businesses can use these insights to negotiate better trade agreements, adjust pricing strategies, and plan for market shifts before they occur.

Additionally, the current system effectively integrates multiple data sources, ensuring a well-rounded approach to item movements. By combining trade, pricing, and food balance data, it enables a deeper understanding of market dynamics. This multi-faceted approach enhances decision-making across different levels of the supply chain, from producers to retailers.

Overall, the implementation of this system has resulted in improved efficiency, cost savings, and better strategic planning for supply chain stakeholders. By optimizing trade routes, and providing statistical insights, this inventory management system stands as a powerful tool for enhancing global supply chain operations.

4 Data Sources

The foundation of this inventory management analysis relies on well-structured and comprehensive datasets that provide insights into trade relationships, pricing trends, and food balance statistics. The data is primarily sourced from FAOSTAT, a global database maintained by the Food and Agriculture Organization (FAO), which provides detailed information on food production, trade, and market trends. These datasets have been integrated into a dual-database architecture using both PostgreSQL and Neo4j, enabling efficient relational and graph-based analysis.

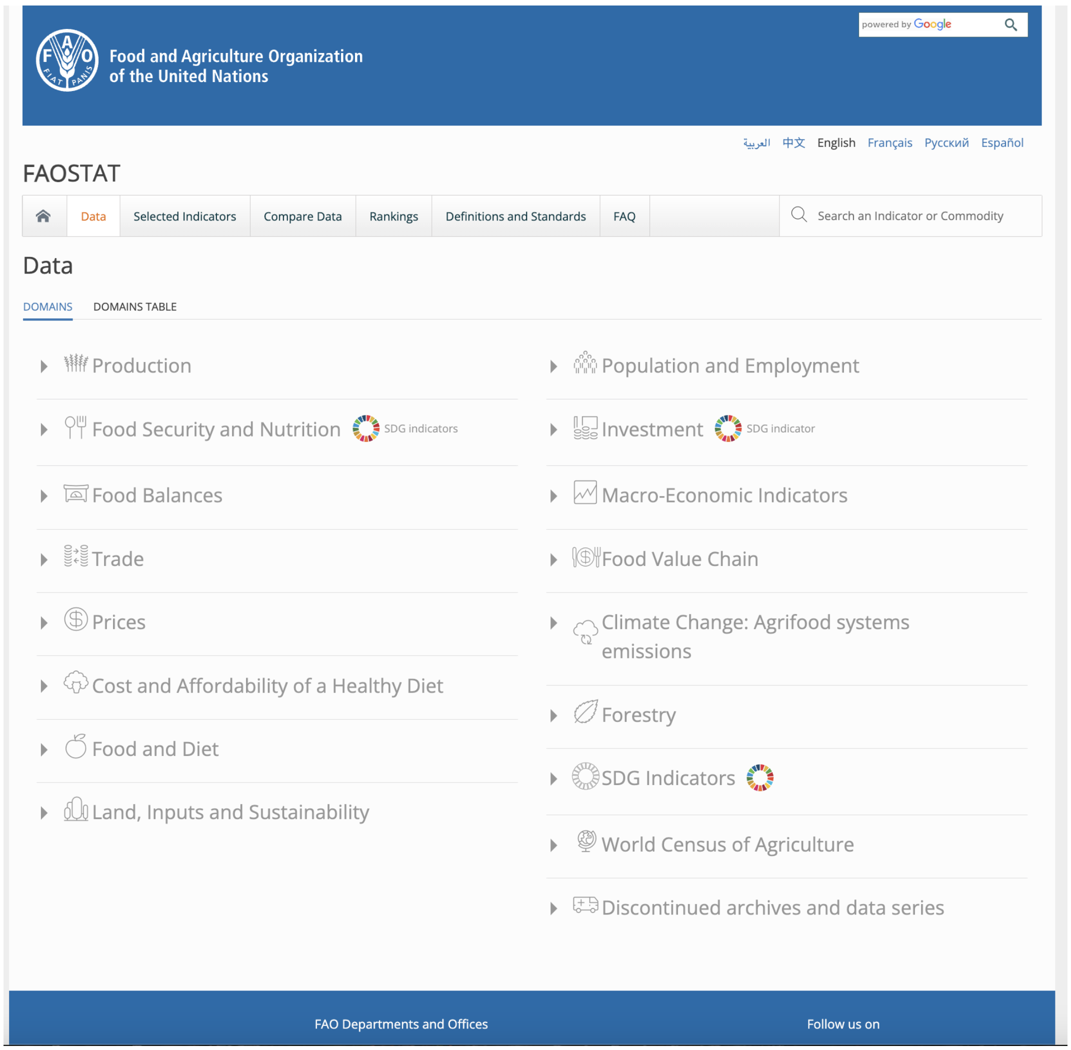

Figure 1: FAOSTAT

4.1 Role of Data Source

The role of data source in this project is essential for understanding and analyzing global trade flows, inventory levels, and price fluctuations. By leveraging trade data, it becomes possible to examine import and export transactions between countries, helping to identify supply chain dependencies and trade imbalances. This information is crucial for businesses and policymakers aiming to optimize trade routes and mitigate potential disruptions in the supply chain. Additionally, price data provides a historical perspective on market trends, enabling cost analysis, inflation tracking, and demand forecasting. Through detailed price observations across multiple years, the system can predict future price movements and assess the economic impact of various trade policies.

Another critical aspect is the analysis of food balances, which allows for monitoring the production, consumption, and stock levels of various commodities. This dataset provides valuable insights into food security, helping organizations track surplus or shortages in different regions. By integrating these datasets into a structured database system, the project enables a comprehensive evaluation of supply chain efficiency, resource allocation, and trade decision-making. The combination of these data sources ensures that inventory management is driven by accurate, data-informed strategies that align with real-world trends.

4.2 Structure of Data Sources

The data sources used in this project are structured into multiple relational tables, each serving a specific purpose within the inventory management framework. The primary datasets include trade data, price data, and food balances, all of which contain key attributes essential for inventory analysis. Each dataset is designed to ensure consistency, data integrity, and efficient querying for analytical purposes.

The **price dataset** consists of attributes such as area, item, element, and year-wise price values. It enables the analysis of commodity price fluctuations over time, allowing businesses to make informed decisions based on historical trends. The structured nature of this dataset facilitates comparative pricing studies across different regions and commodities.

The **trade dataset** records detailed information on bilateral trade transactions between countries. It includes attributes such as reporter area, partner area, item codes, trade values, and yearly trade quantities. This dataset plays a crucial role in understanding global trade dependencies, identifying major exporters and importers, and optimizing supply chain logistics. By maintaining a structured relational format, the trade dataset ensures efficient tracking of product movement across international borders.

The **food balance dataset** provides insights into the production, consumption, and stock levels of various commodities. It contains attributes such as area codes, item codes, element codes, and annual records of food supply quantities. This dataset is instrumental in assessing food security, determining self-sufficiency levels, and predicting future demand patterns. By integrating this information with trade and price data, the project creates a comprehensive model for inventory management analysis.

To support relational consistency and efficient querying, additional lookup tables such as **area codes, item codes, and element codes** are included in the database. These tables ensure standardized referencing of countries, commodities, and trade elements across all datasets, minimizing redundancy and enhancing query performance.

The following **Entity-Relationship (ER)** diagram illustrates the structure and interconnections between these datasets. The ER diagram visually represents how trade, price, and food balance data are linked through common attributes such as area codes and item codes, ensuring seamless integration of information for analytical purposes.

This structured approach to data organization ensures that the inventory management system is not only comprehensive but also scalable for future enhancements. The integration of trade, price, and food balance data allows for an in-depth analysis of global supply chain dynamics, making the system an invaluable tool for decision-makers in the logistics, agricultural, and trade industries.

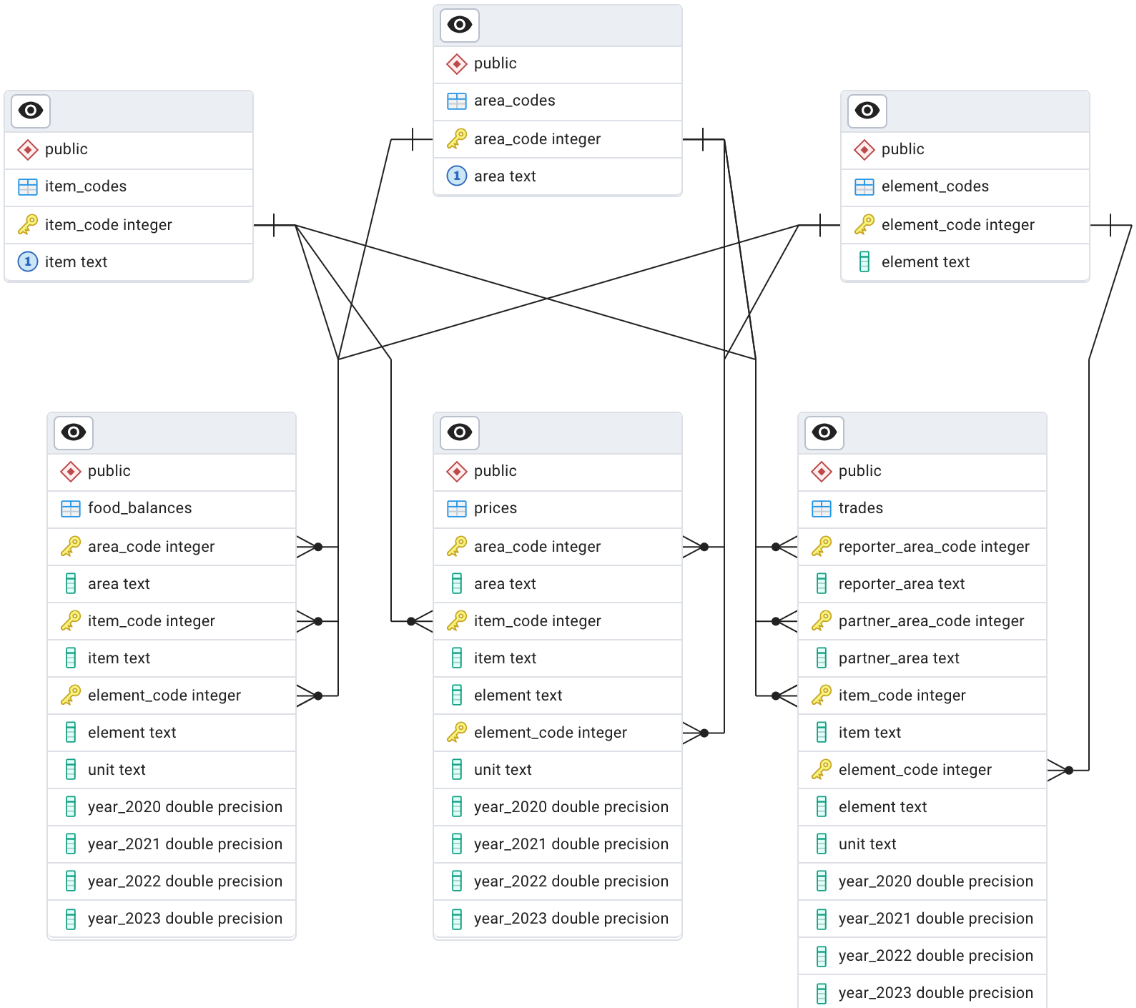

Figure 2: Entity-Relationship Diagram for Inventory Management Data Sources

5 Methodology

5.1 Data Preparation and Preprocessing

A critical phase in our approach was the comprehensive preparation of datasets sourced from FAOSTAT. Data normalization, cleansing, and preprocessing were meticulously conducted utilizing PostgreSQL, a relational database management system (RDBMS) chosen for its ability to handle structured data efficiently. This process ensured that the data was clean, structured, and reliable before integrating it into Neo4j, a graph database management system (GDBMS) optimized for analyzing relational structures and performing advanced graph analytics.

The preprocessing phase involved feature selection, handling missing values, ensuring that only relevant attributes were retained for analysis. Specific commodity categories such as Apples, Oats, Spinach, Chickens, and Wheat were selected, along with key trade metrics including Import Quantity, Export Quantity, Production, Food Supply Quantity (tonnes), and Producer Price (USD/tonne). Additionally, geographic filtering was applied to focus on trade between major countries including the United States of America, Ecuador, Uruguay, Canada, and Brazil. The structured dataset was stored in PostgreSQL tables such as ‘prices‘, ‘trades‘, and ‘foodbalances‘, ensuring consistency and referential integrity before transitioning to the graph model.

5.2 Graph Model Design and Evolution

5.2.1 Base Model and Initial Design

The initial graph model was designed to capture trade relationships by representing countries and commodities as nodes, with direct relationships representing trade interactions. Relationships such as ‘TRADES‘ were established between countries to indicate import and export flows. Additionally, a ‘HASTRADES‘ relationship connected a country to its traded items, encapsulating attributes such as production, consumption, and pricing.

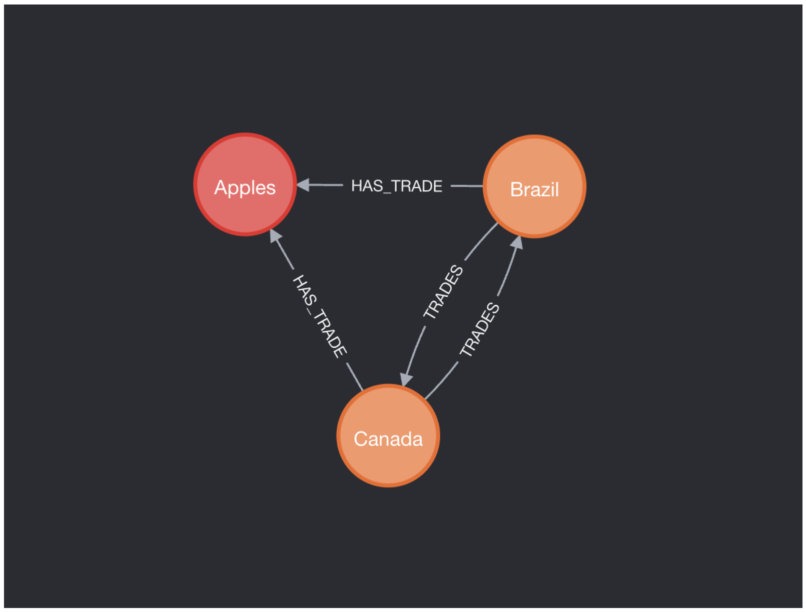

Figure 3: Old Model

While this model effectively visualized trade flows, it suffered from several limitations. It did not adequately represent time-based trends in trade or allow for detailed analysis of production and consumption dynamics. Additionally, embedding multiple attributes within a single node constrained query flexibility and increased query complexity, making it difficult to conduct detailed analyses involving multiple attributes across time.

5.2.2 Refined Model and Enhancements

Recognizing the inefficiencies in the initial design, we restructured the graph model to enhance analytical granularity and query flexibility. The refined model introduced additional nodes for ‘Year‘, ‘TradeLink‘, and ‘Element‘, enabling a more structured representation of trade dynamics. The inclusion of a ‘TradeLink‘ node facilitated a more explicit relationship between countries and traded products, capturing transactional details more effectively.

Key relationships were refined to support advanced analytics. The ‘TRADES‘ relationship was redesigned to incorporate year-wise exports, enabling a time-series analysis of trade volumes. The ‘HASTRADELINK‘ relationship between a country and a trade link provided a structured way to represent trade interactions, while ‘HASOBSERVATION‘ linked trade links to production, consumption, and loss data. Additionally, ‘FORPRODUCT‘ relationships explicitly connected trade transactions to specific commodities, facilitating detailed product-level trade analysis.

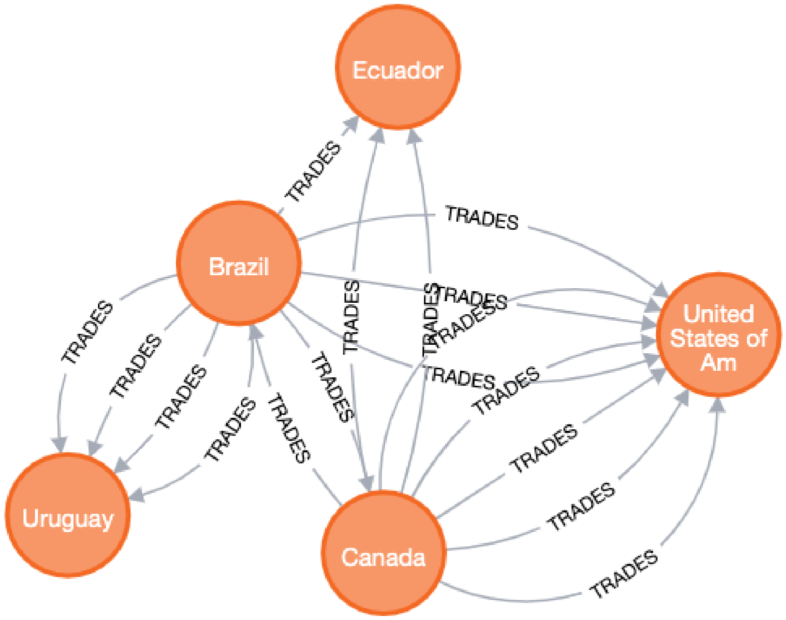

Figure 4: New Data Model

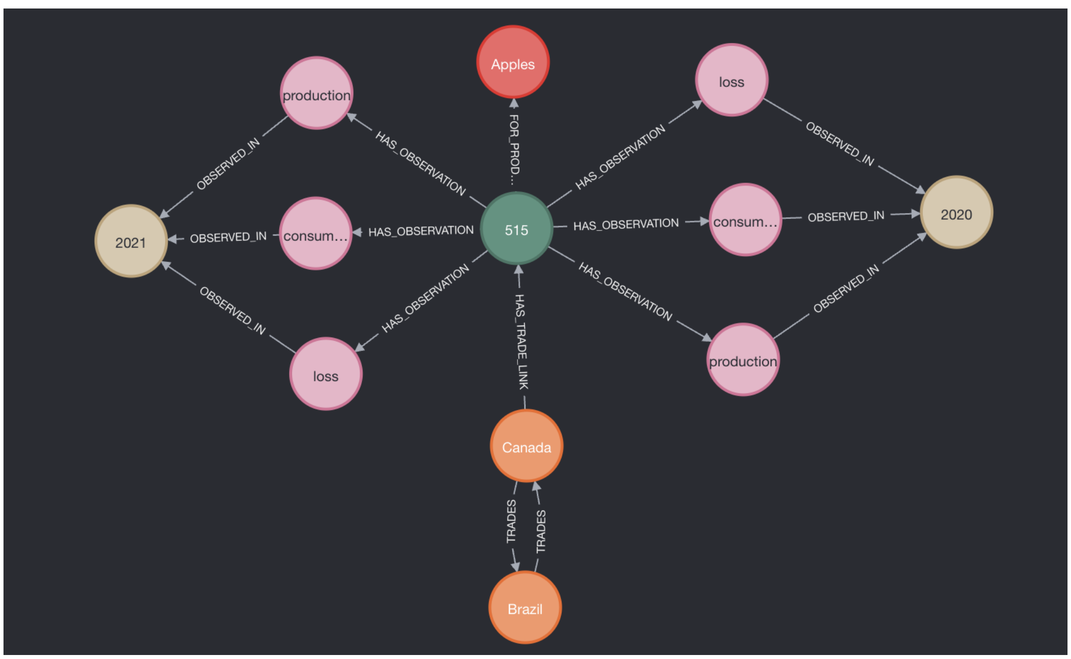

Figure 5: New Data Model with Observations

5.3 Integration of Relational and Graph Databases

The relational data model in PostgreSQL was structured to ensure efficient data management and seamless integration with Neo4j. Tables such as ‘foodbalances‘, ‘trades‘, and ‘prices‘ were created with foreign key constraints to maintain referential integrity. The data import and transformation process involved standardizing column names, handling missing values, and converting data into structured time-series formats.

Once structured, the dataset was imported into Neo4j using Cypher queries, creating nodes for 'Country', 'FoodProduct', 'Year', and 'TradeLink', and establishing relationships that captured trade interactions. Time-dependent attributes were mapped using 'OBSERVEDIN' relationships, linking observations such as production and consumption to their respective years. This hybrid approach allowed structured storage in PostgreSQL while leveraging Neo4j's powerful graph traversal capabilities.

5.4 Querying, Analysis, and Performance Optimization

With the structured graph model in place, we designed Cypher queries to extract meaningful insights from the data. Queries were optimized for performance by restructuring relationships and indexing key attributes. The system supported time-series analysis of trade patterns, allowing stakeholders to track fluctuations in production, pricing, and consumption over multiple years.

Trade dependencies between countries were analyzed using path queries, helping identify supply chain bottlenecks and alternative trade routes. Additionally, price fluctuations were examined to determine the impact of geopolitical and economic factors on commodity prices. The refined graph model significantly improved query execution time, reducing complexity and enhancing the ability to perform granular analysis across trade dimensions.

5.5 Dashboard Integration and Visualization

To ensure usability, we integrated Neo4j with NeoDash to create an interactive dashboard that allowed users to explore trade data dynamically. The dashboard provided various visualizations, including trade volume analysis, pricing trends, and supply chain connectivity maps. Users could filter data based on specific products and years, enabling a customized analytical experience.

By leveraging NeoDash, users could interact with trade networks, identify market trends, and assess supply-demand mismatches in real time. The dashboard enhanced decision-making by providing a visual representation of trade flows and allowing for deeper exploration of trade dependencies.

5.6 Final Architecture and Future Scalability

The final architecture was designed as a hybrid system, utilizing PostgreSQL for structured data storage and Neo4j for advanced relationship-based analysis. This approach enabled seamless data integration and ensured that the system could scale as additional datasets and analytics capabilities were introduced.

Future improvements include real-time data ingestion to capture live trade transactions and machine learning-driven forecasting models for predictive analytics. Enhanced synchronization mechanisms between PostgreSQL and Neo4j will further optimize performance, ensuring that the system remains scalable and adaptable to evolving trade dynamics.

6 Results

6.1 Graph Model Visualization

Using Neo4j, we successfully visualized the intricate relationships within supply chain data through a structured graph model. The visualization clearly illustrates the interconnectedness between countries, items, trade links, observations, and associated temporal data.

6.2 Analytical Outcomes with Cypher Queries

The implementation of example Cypher queries that facilitated analytical insights into the supply chain dynamics are detailed below:

Loss Percentage with respect to Oats in Canada:

```
MATCH (c:Country {name:"Canada"})-[:HAS_TRADE_LINK]->
(tl:TradeLink)-[:FOR_PRODUCT]->(p:FoodProduct {name:"Oats"})
MATCH (tl)-[:HAS_OBSERVATION]->
(op:Observation {type:"production"})-[:OBSERVED_IN]->(y:Year)
MATCH (tl)-[:HAS_OBSERVATION]->
(ol:Observation {type:"loss"})-[:OBSERVED_IN]->(y)
WHERE op.value > 0
RETURN y.year AS Year, op.value AS Production,
ol.value AS Loss, (ol.value / op.value)*100 AS LossPercentage
ORDER BY Year;
```

	Year	Production	Loss	LossPercentage
1	2020	4227300.0	253618.59	5.999540841671989
2	2021	4575800.0	274678.99	6.002862668822938
3	2022	2898619.0	174362.86	6.015376977795288
4	2023	5226465.0	313892.67	6.0058312836687895

Figure 6: Loss percentage visualization

Multi-hop between two countries:

```
MATCH path = (start:Country {name:"Canada"})-[:TRADES*1..3]->
(end:Country {name:"Uruguay"})
RETURN path
LIMIT 5;
```

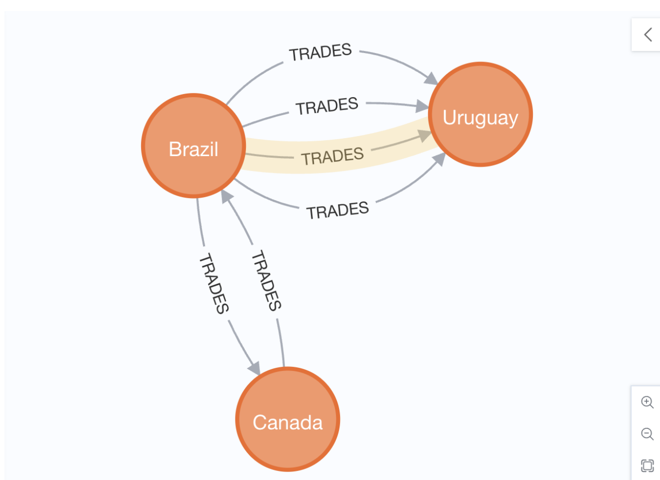

Figure 7: Multi Hop

All Elements for all countries and products:

```
MATCH (y:Year {year:2023})
```

```
MATCH (tl:TradeLink)-[:HAS_OBSERVATION]->
(o:Observation)-[:OBSERVED_IN]->(y)
MATCH (tl)-[:FOR_PRODUCT]->(p:FoodProduct)
MATCH (c:Country)-[:HAS_TRADE_LINK]->(tl)
RETURN c.name AS Country, p.name AS Product,
o.type AS Metric, o.value AS Value
ORDER BY c.name, p.name, o.type;
```

	Country	Product	Metric	Value
1	"Brazil"	"Apples"	"consumption"	930998.78
2	"Brazil"	"Apples"	"loss"	91115.32
3	"Brazil"	"Apples"	"price"	430.4
4	"Brazil"	"Apples"	"production"	1047217.0
5	"Brazil"	"Oats"	"loss"	38874.33
6	"Brazil"	"Oats"	"price"	220.9

Figure 8: All elements of countries

Subset of the graph model:

```
MATCH (n)-[r]->(m)
RETURN n, r, m
LIMIT 50;
```

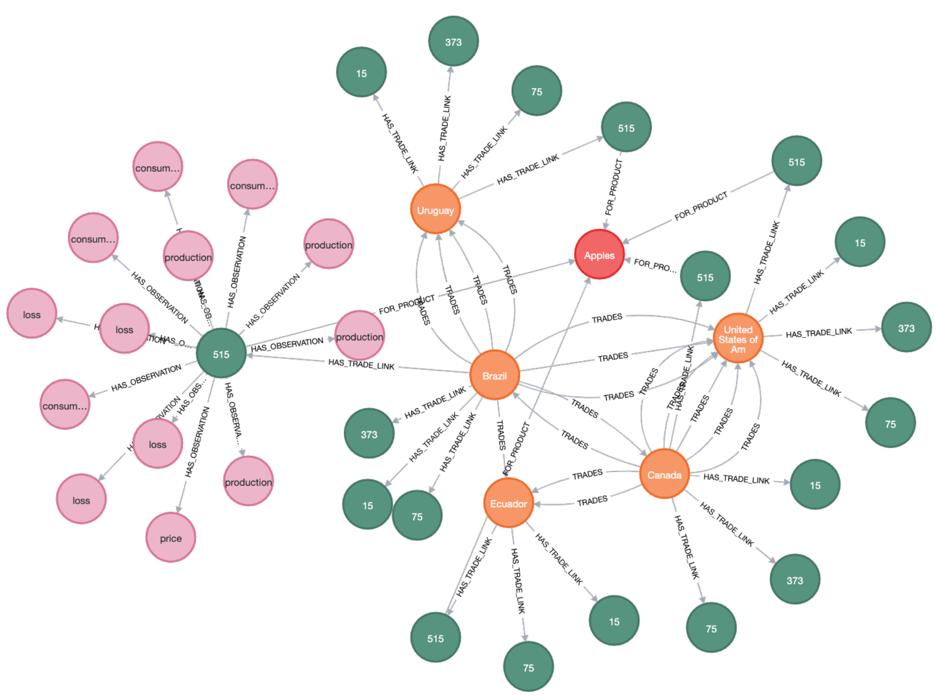

Figure 9: Subset graph

Complete Graph model:

```
MATCH (c:Country)-[:HAS_TRADE_LINK]->
(tl:TradeLink)-[:FOR_PRODUCT]->(p:FoodProduct)
MATCH (tl)-[:HAS_OBSERVATION]->
(o:Observation)-[:OBSERVED_IN]->(y:Year)
RETURN c, tl, p, o, y
LIMIT 500;
```

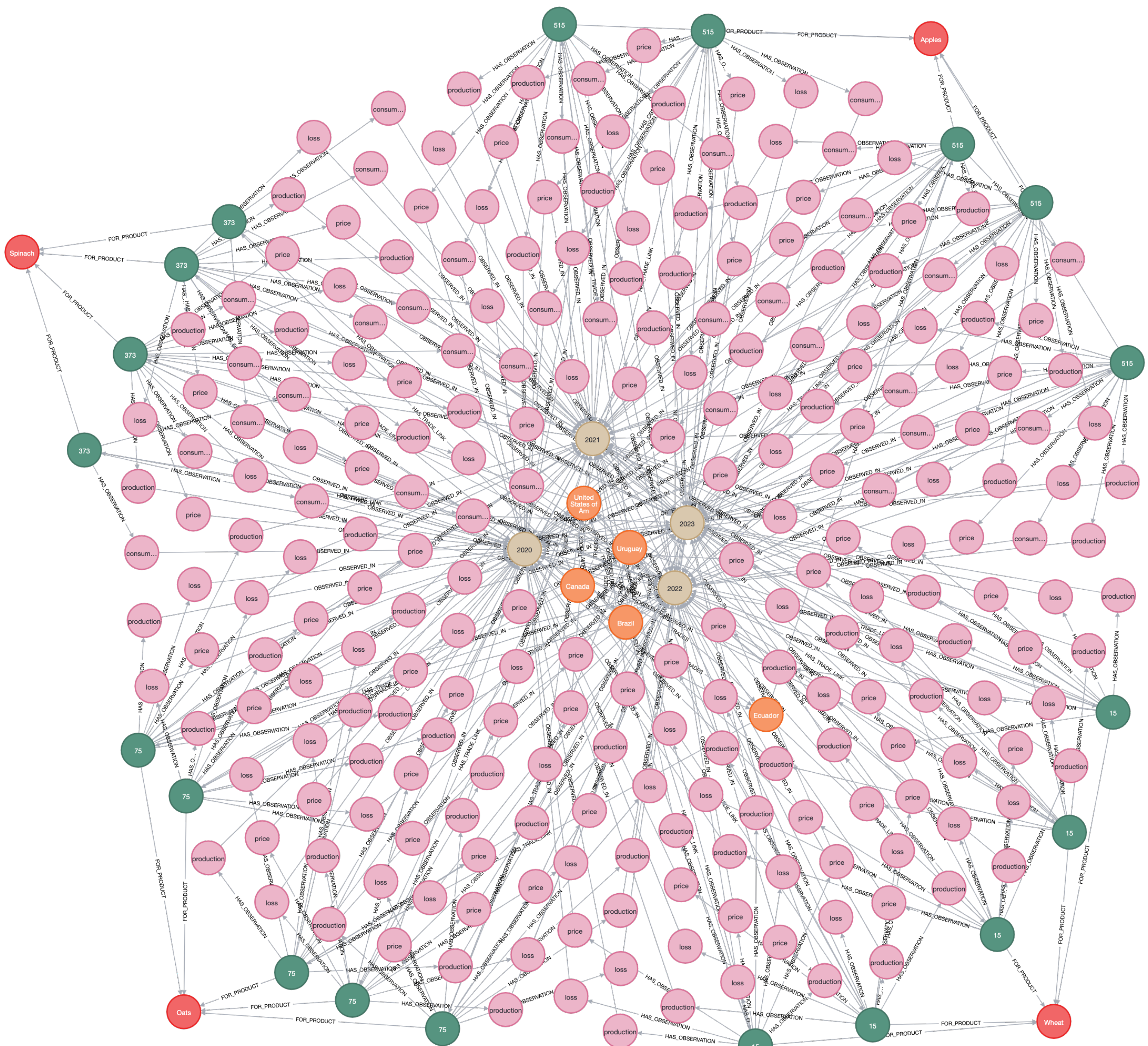

Figure 10: Full Graph

6.3 Visualization through NeoDash Dashboard

Leveraging NeoDash, we transformed complex data relationships into intuitive, actionable visualizations. Key dashboard components included:

Import - Export Pie Chart: General trade statistics.

```
CALL {
 MATCH (exporter:Country)-[t:TRADES]->(:Country)
 WHERE t.export_2023 IS NOT NULL AND
 t.product_name = $neodash_selectedproduct
 RETURN exporter.name AS country, t.product_name AS product,
 sum(t.export_2023) AS totalExports, 0 AS totalImports
 UNION ALL
 MATCH (:Country)-[t:TRADES]->(importer:Country)
 WHERE t.export_2023 IS NOT NULL AND
 t.product_name = $neodash_selectedproduct
 RETURN importer.name AS country, t.product_name AS product,
 0 AS totalExports, sum(t.export_2023) AS totalImports
}
WITH country, product, sum(totalExports) AS totalExports,
sum(totalImports) AS totalImports
RETURN country, totalExports, totalImports
ORDER BY totalExports DESC;
```

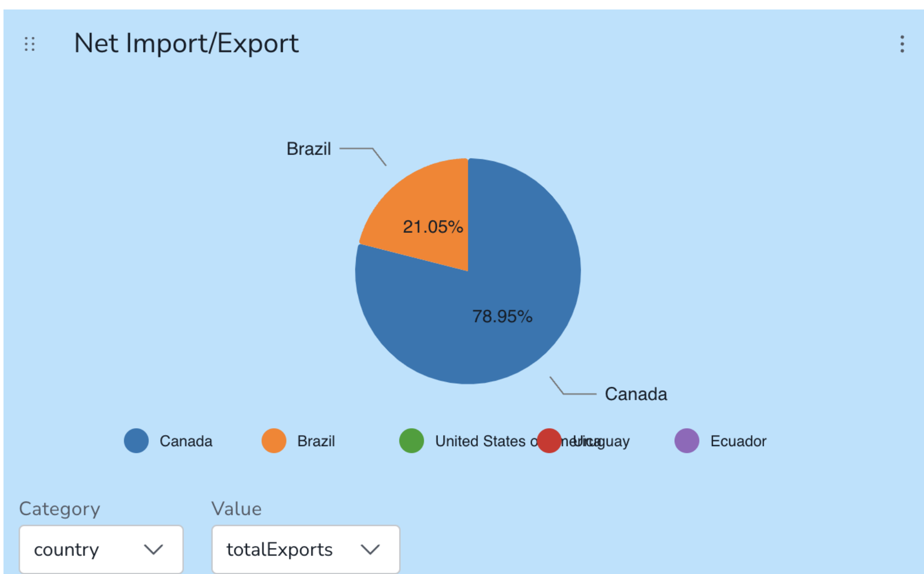

Figure 11: Net Import Export

Alternate Suppliers: Alternate trade options for items.

```
MATCH (supplier:Country)-[:HAS_TRADE_LINK]->(tl:TradeLink)-
[:FOR_PRODUCT]->(p:FoodProduct {name: $neodash_selectedproduct})
MATCH (tl)-[:HAS_OBSERVATION]->
(o:Observation {type: "production"})-[:OBSERVED_IN]->(y:Year {year: 2023})
WHERE supplier.name <> $neodash_selectedcountry AND o.value > 0
RETURN supplier.name AS AlternateSupplier,
o.value AS ProductionQuantity ORDER BY o.value DESC;
```

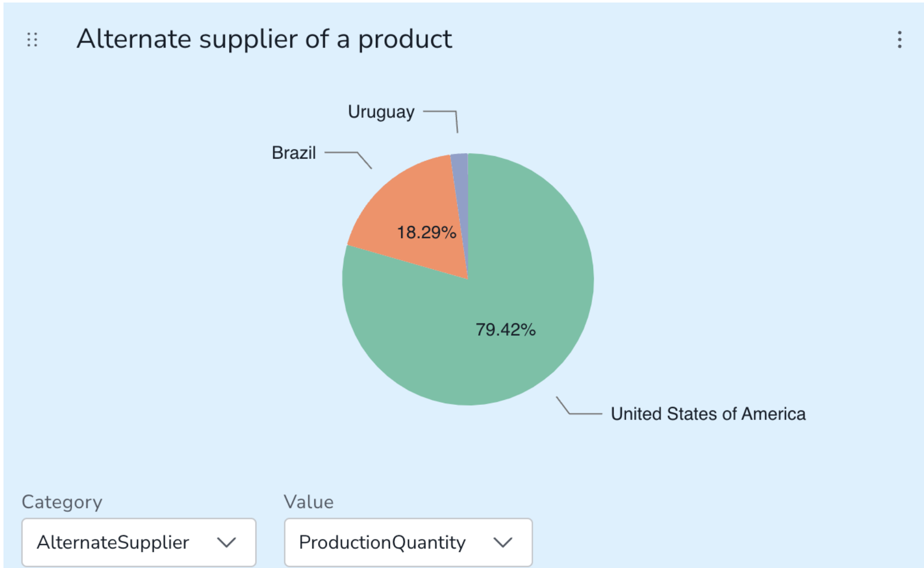

Figure 12: Shortest trade path

Shortest Path Graph: Trade relations between countries.

```
MATCH (start:Country {name: $neodash_startcountry}),
(end:Country {name: $neodash_endcountry})
MATCH path = shortestPath((start)-[:TRADES*]-(end))
RETURN start.name AS StartCountry,
end.name AS EndCountry, length(path) AS Hops, path;
```

Figure 13: Shortest trade path

Trade Ban Statistics: Impact of export bans.

```
MATCH (exporter:Country)-[r:TRADES]->(importer:Country)
WHERE toLower(r.product_name) = toLower($neodash_selectedproduct)
AND toLower(exporter.name) = toLower($neodash_selectedcountry)
```

```
AND r.export_2023 IS NOT NULL
WITH exporter, importer, r.export_2023 AS BannedExport
OPTIONAL MATCH (importer)-[totalTrade:TRADES]->(:Country)
WHERE toLower(totalTrade.product_name) =
toLower($neodash_selectedproduct)
WITH exporter, importer, BannedExport,
COALESCE(SUM(totalTrade.export_2023), 0) AS TotalImports,
CASE WHEN SUM(totalTrade.export_2023) > 0
THEN (BannedExport / SUM(totalTrade.export_2023)) * 100
ELSE 0 END AS DependencyRate
OPTIONAL MATCH (alternative:Country)-[altTrade:TRADES]->(importer)
WHERE toLower(altTrade.product_name) = toLower($neodash_selectedproduct)
AND alternative <> exporter
WITH exporter, importer, BannedExport, TotalImports,
DependencyRate, COALESCE(alternative.name,
"No Alternative Supplier") AS AlternativeSupplier,
COALESCE(altTrade.export_2023, 0) AS AltSupply
RETURN exporter.name AS BanningCountry, importer.name AS AffectedCountry,
BannedExport AS ImportFromBannedCountry, TotalImports AS TotalImports2023,
DependencyRate AS PercentageDependency, AlternativeSupplier,
AltSupply AS AlternativeSupply,
CASE WHEN AltSupply = 0 OR AltSupply < BannedExport
THEN "High Disruption Risk" ELSE "Low Disruption Risk"
END AS ImpactAssessment ORDER BY DependencyRate DESC;
```

Trade ban

BanningCountry	Canada	Canada	Canada
AffectedCountry	Brazil	Ecuador	United States of Amer...
ImportFromBannedCo...	19,472	258,322	752,628
TotalImports2023	274,707	0	0
PercentageDependency	7.088	0	0
AlternativeSupplier	No Alternative Supplier	No Alternative Supplier	Brazil

Rows per page: 5 1–5 of 7

Figure 14: Trade ban

Country-wise Revenue and Product Growth Rate: Similar structured queries provided detailed insights into revenue and growth trends by integrating consumption and pricing data for actionable business insights.

7 Conclusion

This project successfully leveraged the integration of relational and graph database technologies to address complex analytical challenges in supply chain management. The integrated solution combined PostgreSQL's structured data management with Neo4j's powerful graph analytics capabilities, offering substantial practical advantages.

Tools utilized: PostgreSQL, Neo4j, NeoDash

- **Data Integration using RDBMS and GDBMS** We successfully achieved seamless data integration between PostgreSQL (RDBMS) and Neo4j (GDBMS). The structured data was initially managed in PostgreSQL, providing transactional support, data normalization, and ensuring data consistency. Through feature selection and feature engineering, relevant data was scraped from the source (FAOSTAT) and preprocessed further. In addition, we utilized Neo4J (GDBMS) for graph modeling and this integration enabled the leveraging of Neo4j's advanced graph capabilities, allowing complex relational analyses that were challenging in traditional relational databases.

- **Hands-on Experience with Graph Data Modeling** The project aim fulfilled to provide valuable hands-on experience in graph data modeling using Neo4j. Learning RDBMS and GDBMS modeling was an iterative learning process, started from a basic model we refined our approach further, breaking down nodes and attributes into more discrete entities and clearly defined relationships. This iterative modeling process deepened our understanding of optimal graph database practices, significantly enhancing our technical ability in managing complex graph structures.

- **Data Visualization Using NeoDash** To facilitate intuitive understanding and effective presentation of supply chain insights, we employed NeoDash for data visualization. NeoDash allowed us to create dynamic, interactive dashboards with clear visualization panels and tiles, significantly improving analytical readability and decision-making support.

  Dashboard Panels and Tiles Suggestions:

  - **Node Graph Panel:** Visually represents country-to-country multi-hop relationships.
  - **Bar Chart Panel:** Displays metrics such as total exports and imports by country.
  - **Line Graph Panel:** Illustrates price fluctuations for a product over multiple years.
  - **Network Graph Panel:** Highlights relationships and trade links among countries and items.
  - **Metric Tiles:** Summarizes critical statistics including highest traded items and annual performance indicators.

  Collectively, these contributions represent significant advancements in the practical application of integrated database solutions, graph modeling, advanced analytics, and data visualization. This project offers valuable technical insights and provides a robust framework for enhancing decision-making processes in complex supply chain environments.

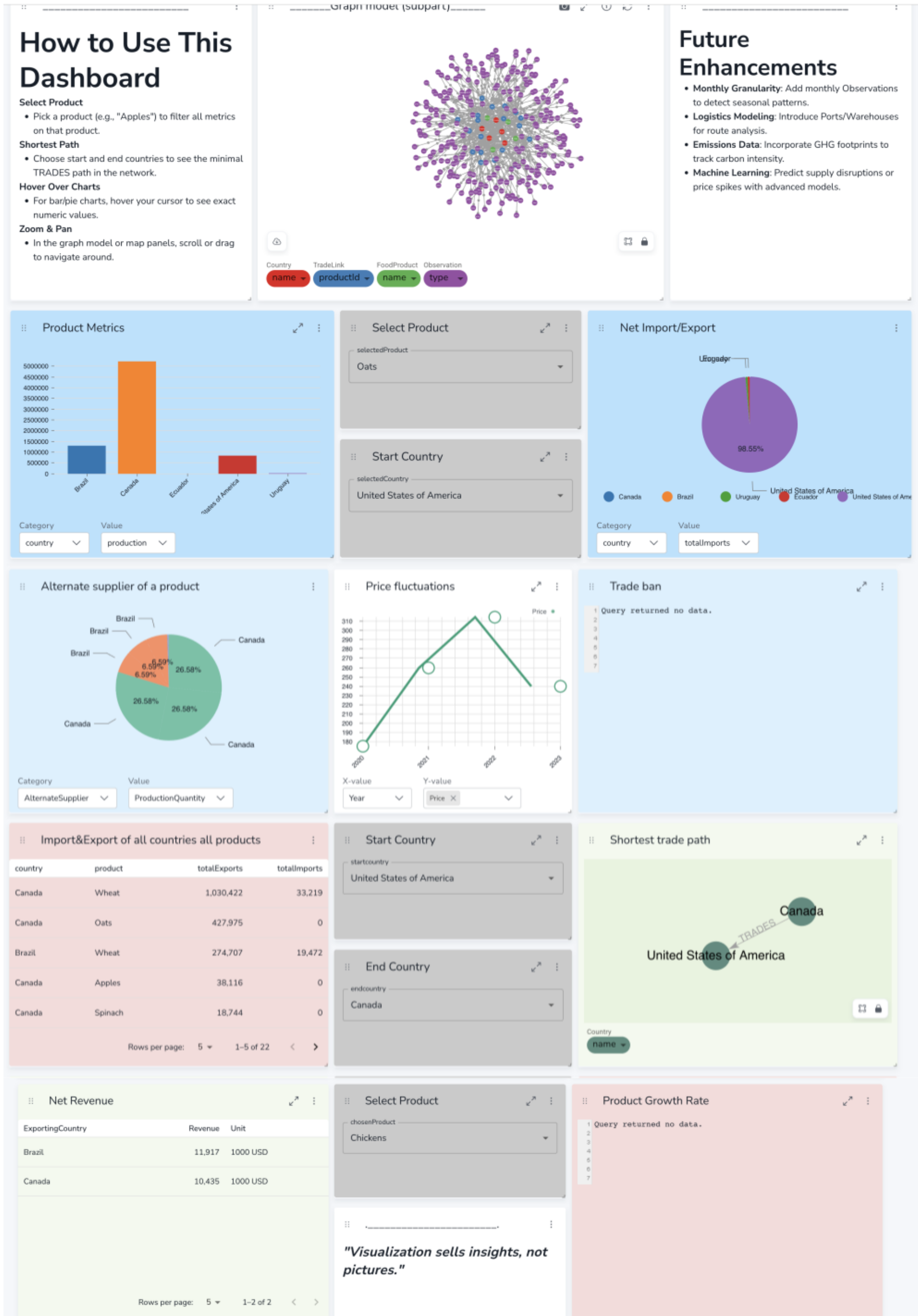

Figure 15: Dashboard of Supply Chain Analysis

8 Current Limitations

A. Partial Dataset Utilization Current Scope: The system currently leverages data on food balances, pricing, and trade. While these datasets provide crucial insights into inventory and trade flows, they represent only a portion of the factors affecting the supply chain. Impact: Missing socio-economic, demographic, and nutritional indicators may limit the system's ability to deliver a holistic analysis of supply chain dynamics, potentially overlooking external factors that can affect supply and demand.

B. Data Refresh and Updates Data is sourced from FAOSTAT and other repositories that are updated periodically. The challege is that without an automated or frequent integration process, the system risks using outdated information, leading to decisions based on stale data. This limitation can affect the accuracy of predictive analytics and real-time decision-making.

C. Complexity of Multi-DBMS Integration The system uses a dual-database approach, with PostgreSQL handling structured data (e.g., prices, trades, food balances) and Neo4j managing graph data (e.g., trade relationships). The challenge is synchronizing and maintaining consistency between these two database systems is complex. Any discrepancies in data migration or transformation processes can lead to inconsistencies. this impacts the integration complexity can increase maintenance overhead and may pose challenges when scaling or updating the system.

D. Limited Advanced Analytics Current Capabilities of the project primarily focuses on inventory management and relationship mapping using SQL and Cypher queries. So there is currently a limited use of advanced forecasting models or machine learning algorithms, which could otherwise provide deeper predictive insights. This restricts the system's ability to forecast future trends and optimize decisions under varying market conditions.

E. Scalability and Performance Constraints As the amount of data increases, the current architecture might face challenges in terms of query performance and system responsiveness. Without a scalable infrastructure, real-time analytics and interactive dashboards may suffer from lag, reducing the overall user experience.

F. Dashboard and User Interface Limitations The interactive dashboard built with tools like NeoDash provides valuable insights but may not offer comprehensive drill-down or customizable visualizations. Due to limited interactivity could hinder stakeholders from performing detailed exploratory analyses, potentially reducing the system's overall usability.

9 Future Scope

This inventory management system can be enhanced when supply chain networks change by incorporating technology, real-time data ingestion, and predictive intelligence to improve insights and resilience. Increasing the scope of data to include real-time market trends, climate impact data, and macroeconomic indicators will improve forecasting accuracy and promote an integrated approach to inventory management. Proactive decision-making will be made possible by implementing machine learning for pricing and demand forecasting in conjunction with sophisticated graph analytics in Neo4j. Larger datasets can be managed quickly by switching to cloud-based or distributed systems, such AWS or Google Cloud, and utilizing OLAP databases. While integrating logistics data—ports, warehouses, and routes—would further optimize route planning, lower costs, and promote global supply chain efficiency, guaranteeing adaptability in a constantly shifting market, interactive dashboards in Power BI or Tableau would enhance use.

10 Acknowledgement

We'd like to thank our Professor Dr.Amarnath Gupta for the insights he has provided throughout the quarter. We'd also like to thank our Teaching Assistant Zixi Chen for all the advice she kindly provided us with and all the questions she answered as we worked on this project.

11 References

References

[1] Food and Agriculture Organization of the United Nations. *FAOSTAT Database*. Available online: https://www.fao.org/faostat/en (Accessed March 2025).

[2] ResearchGate. Inventory Management Optimization with Data Analytics for a Trading Company. Available online: https://www.researchgate.net/publication/368454977_Inventory_Management_Optimization_with_Data_Analytics_for_a_Trading_Company (Accessed March 2025).

[3] Kaggle. Supply Chain Inventory Management Dataset. Available online: https://www.kaggle.com/datasets/mohammedazarudheen/supply-chain-inventory-management-data-analyst (Accessed March 2025).

[4] Neo4j. Getting Started Documentation. Available online: https://neo4j.com/docs/getting-started/ (Accessed March 2025).

[5] PostgreSQL Documentation. Available online: https://www.postgresql.org/docs/ (Accessed March 2025).

[6] arXiv. Supply Chain Analysis using Graph Models. Available online: https://arxiv.org/pdf/2401.15299 (Accessed March 2025).

[7] Kaggle. Supply Chain Dataset. Available online: https://www.kaggle.com/datasets/amirmotefaker/supply-chain-dataset/data (Accessed March 2025).

[8] Neo4j. Graph Database vs. Relational Database Concepts. Available online: https://neo4j.com/docs/getting-started/appendix/graphdb-concepts/graphdb-vs-rdbms/ (Accessed March 2025).

[9] LangChain Python Documentation. Vectorstores Module. Available online: https://python.langchain.com/v0.1/docs/modules/data_connection/vectorstores/ (Accessed March 2025).

[10] UN Comtrade Database. Available online: https://comtradeplus.un.org/ (Accessed March 2025).