Graph-Based Supply Chain Network Optimization & Risk Mitigation

Data Model and Data Population

By

Razi Haider (rh06882@st.habib.edu.pk)
Moid Raza Khan (mk07216@st.habib.edu.pk)

March 28, 2024

Dhanani School of Science and Engineering

Habib University

Spring 2024

1 Data Model

1.1 All Feature Explanation

- Product Type: The type of product associated with specific data in the supply chain.
- SKU (Stock Keeping Unit): Unique code used to identify a particular product.
- Price: The price of the product or item in the supply chain.
- Availability: Information about product availability.
- Number of Products Sold: The number of products that have been sold in a certain time period.
- Revenue Generated: Total revenue generated from product sales in a certain time period.
- Customer demographics: Information about customer characteristics, such as age, gender, geographic location, etc.
- Stock Levels: The number of products still available in stock at any given time.
- Lead Times: The time required to order or receive products from suppliers.
- Order Quantities: The number of products ordered in one order or shipment.
- Shipping Times: The time required to ship products from the warehouse or distribution center to customers.
- Shipping Carriers: Companies or services used to ship products to customers.
- Shipping Costs: Costs associated with shipping products, including delivery fees and additional fees.
- Supplier Name: Name of supplier or vendor who provides products or materials to the company.
- Location: The physical location associated with the data in the supply chain, such as the location of a warehouse or distribution center.
- Lead Time: The time required to obtain products or materials from a particular supplier.

- Production Volumes: The number of products produced in a certain time period.
- Manufacturing Lead Time: The time required to produce a product, from ordering materials until the product is ready.
- Manufacturing Costs: Costs related to the production process, including raw material costs, labor, etc.
- Inspection Results: Results of product or material quality inspection.
- Defect Rates: The level of defects or defects in the products produced.
- Transportation Modes: The transportation mode used to send products, such as land, sea or air.
- Routes: Routes or paths used to send products from one point to another in the supply chain.
- Costs: Costs related to various aspects of the supply chain, including transportation costs, production costs, and other costs.

1.2 Nodes with Properties & Rationale

- **Product Node** (productType, sku, price)
  This node represents the different product SKUs in the supply chain. It captures essential product information such as type, price, and availability.
- **Supplier Node** (supplierName, location)
  This node represents the suppliers in the supply chain network. It includes information about the supplier's name and location.
- **Customer Node** (customerDemographics)
  This node represents the customers who purchase the products. It includes information about customer demographics.
- **Transporter Node** (shippingCarrier, transportMode, route)
  This node represents the transportation carriers that ship products from distributors to customers. It includes information about the transporter's name and the mode of transportation used.
- **Location Node**
  This node represents the different locations (cities) involved in the supply chain network. It allows for modeling the geographic distribution of entities.

1.3 Relationships with Properties & Rationale

- **SUPPLIES** *(Supplier to Product)*: availability, stockLevels, leadTimes, productionVolumes, manufacturingLeadTime, manufacturingCosts

  The SUPPLIES relationship connects suppliers to products and includes properties related to availability, stock levels, lead times, production volumes, manufacturing lead times, and manufacturing costs.

- **SHIPS** *(Supplier to Transporter)*: orderQuantities, shippingTimes, shippingCosts

  The SHIPS relationship represents the shipping of products from suppliers to transporters, and it includes properties like order quantities, shipping times, and shipping costs.

- **DELIVERS** *(Transporter to Customer)*: costs

  The DELIVERS relationship represents the transportation of products from transporters to customers, with the associated costs as a property.

- **PURCHASES** *(Customer to Product)*: numberSold, revenueGenerated, inspectionResults, defectRates

  The PURCHASES relationship captures the customers' purchases of products, including the number of products sold, revenue generated, inspection results, and defect rates.

- **LOCATED_IN** *(Supplier to Location, Customer to Location, Transporter to Location)*: No properties

  The LOCATED_IN relationship connects suppliers, customers, and transporters to their respective locations (cities).

1.4 Graph Property Model

You can view the Graph Property Model Here

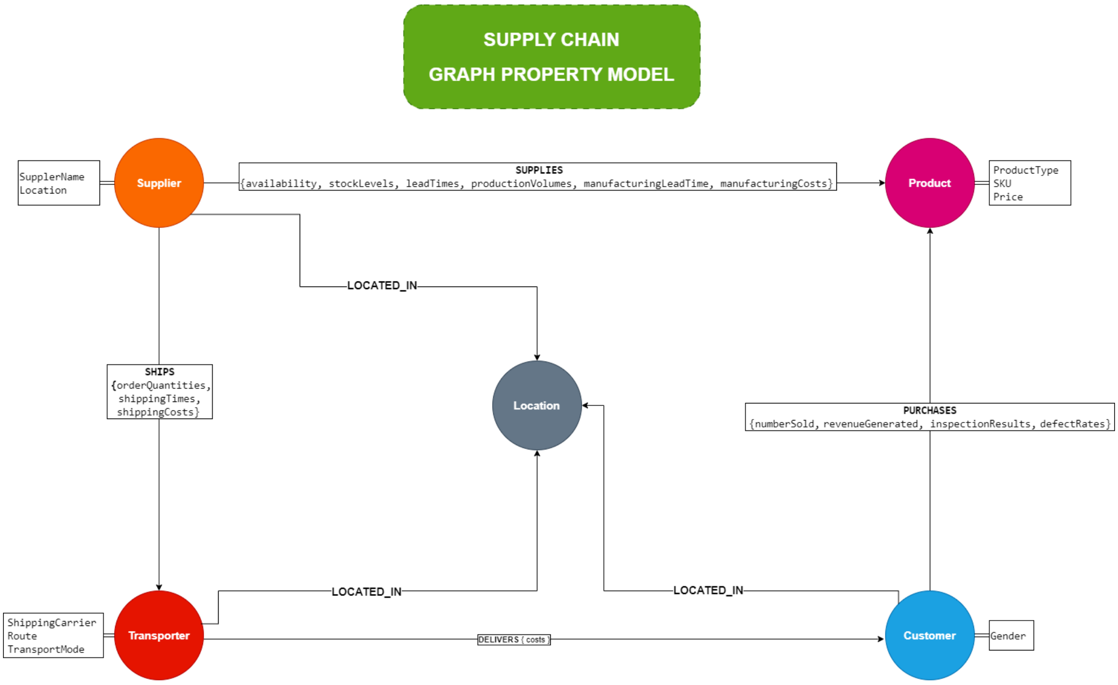

Figure 1: Graph Property Model

2 Data Population

2.1 Data Extraction & Transformation

- Data Loading:

```
import pandas as pd
supply_data = pd.read_csv("supply_chain_data.csv")
supply_data.head()
supply_data.columns
```

- Checking for Missing Values:

```
# Check for missing values
missing_values = supply_data.isnull().sum()

# Display columns with missing values and the count of
    missing values
missing_values = missing_values[missing_values > 0]

if not missing_values.empty:
    print("Columns with missing values:")
    for column, count in missing_values.items():
        print(f"{column}: {count} missing values")
```

```
else:
    print("There are no columns with missing value")
```

Output: There are no columns with missing value

- Checking for Duplicate Values:

```
if supply_data.duplicated().any():
    print(f"There are as many as {supply_data.duplicated().
  sum()} duplicate data.")
else:
    print("There are no duplicate data.")
```

Output: There are no duplicate data.

2.2 Data Loading in Neo4j

2.2.1 Importing Nodes

```
// Create Product nodes
LOAD CSV WITH HEADERS FROM 'file:///supply_chain_data.csv' AS row
CREATE (:Product {productType: row.Product type, sku: row.SKU,
   price: toFloat(row.Price)})

// Create Supplier nodes
LOAD CSV WITH HEADERS FROM 'file:///supply_chain_data.csv' AS row
MERGE (s:Supplier {supplierName: row.Supplier name})
ON CREATE SET s.location = row.Location;

// Create Customer nodes
LOAD CSV WITH HEADERS FROM 'file:///supply_chain_data.csv' AS row
MERGE (c:Customer {demographics: row.Customer demographics});

// Create Transporter nodes
LOAD CSV WITH HEADERS FROM 'file:///supply_chain_data.csv' AS row
MERGE (t:Transporter {carrier: row.Shipping carriers,
   transportMode: row.Transportation modes, route: row.Routes});

// Create Location nodes
LOAD CSV WITH HEADERS FROM 'file:///supply_chain_data.csv' AS row
MERGE (l:Location {name: row.Location});
```

2.2.2 Importing Relationships

```
// Create relationships
LOAD CSV WITH HEADERS FROM 'file:///supply_chain_data.csv' AS row

```

```
// SUPPLIES relationship
MATCH (s:Supplier {supplierName: row.Supplier name}), (p:Product {
    sku: row.SKU})
MERGE (s)-[:SUPPLIES]->(p)
ON CREATE SET
     s.availability = toInteger(row.Availability),
     s.stockLevels = toInteger(row.Stock levels),
     s.leadTimes = toInteger(row.Lead times),
     s.productionVolumes = toInteger(row.Production volumes),
     s.manufacturingLeadTime = toInteger(row.Manufacturing lead
    time),
     s.manufacturingCosts = toFloat(row.Manufacturing costs);

// SHIPS relationship
MATCH (s:Supplier {supplierName: row.Supplier name}), (t:
    Transporter {carrier: row.Shipping carriers})
MERGE (s)-[:SHIPS]->(t)
ON CREATE SET
     s.orderQuantities = toInteger(row.Order quantities),
     s.shippingTimes = toInteger(row.Shipping times),
     s.shippingCosts = toFloat(row.Shipping costs);

// DELIVERS relationship
MATCH (t:Transporter {carrier: row.Shipping carriers}), (c:
    Customer {demographics: row.Customer demographics})
MERGE (t)-[:DELIVERS]->(c)
ON CREATE SET t.costs = toFloat(row.Costs);

// PURCHASES relationship
MATCH (c:Customer {demographics: row.Customer demographics}), (p:
    Product {sku: row.SKU})
MERGE (c)-[:PURCHASES]->(p)
ON CREATE SET
     c.numberSold = toInteger(row.Number of products sold),
     c.revenueGenerated = toFloat(row.Revenue generated),
     c.inspectionResults = row.Inspection results,
     c.defectRates = toFloat(row.Defect rates);

// LOCATED_IN relationship
MATCH (s:Supplier {supplierName: row.Supplier name}), (c:Customer
    {demographics: row.Customer demographics}), (t:Transporter {
    carrier: row.Shipping carriers}), (l:Location {name: row.
    Location})
MERGE (s)-[:LOCATED_IN]->(l)
MERGE (c)-[:LOCATED_IN]->(l)
MERGE (t)-[:LOCATED_IN]->(l);
```

2.3 Final Graph Model in Neo4j

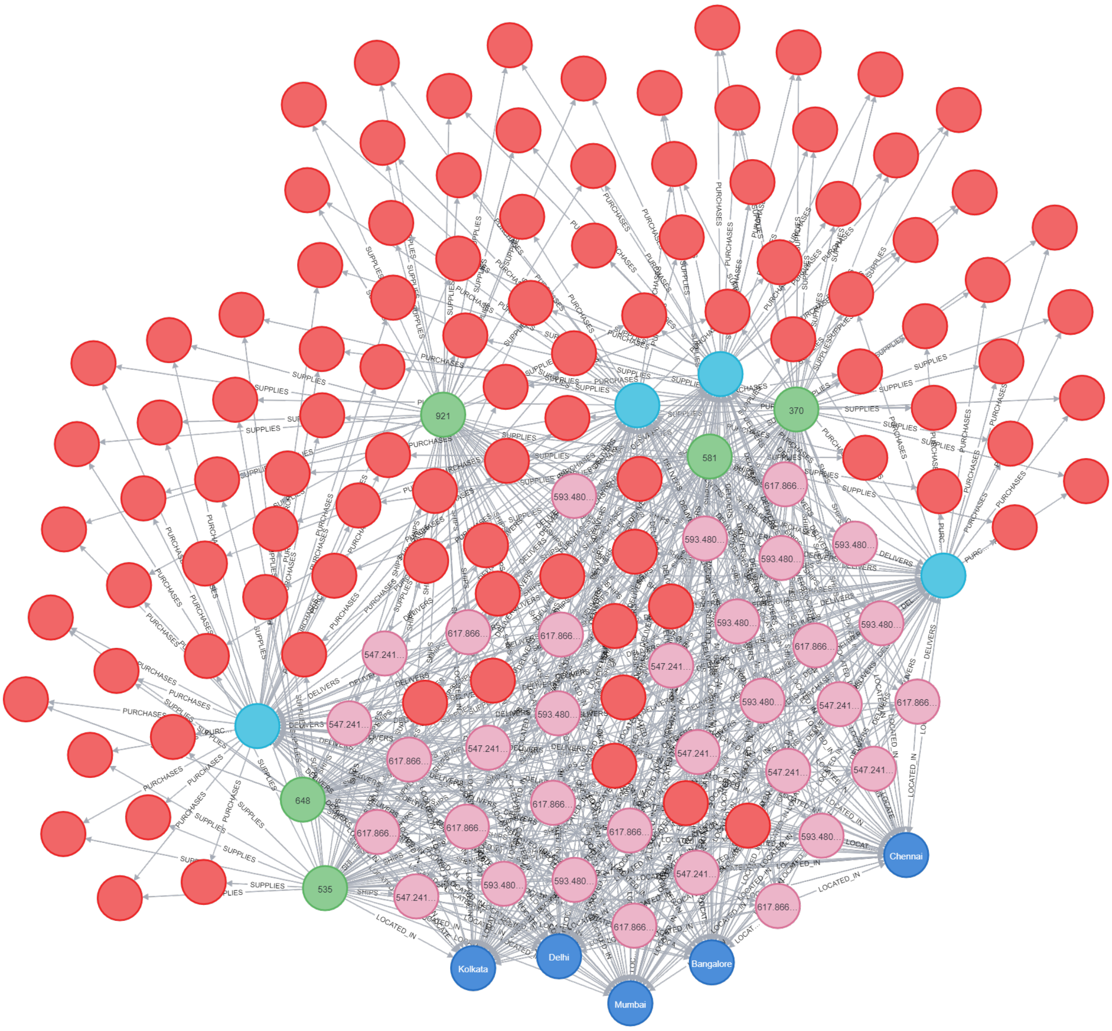

Figure 2: Neo4j Graph Property Model

References

- Supply Chain Data Set from Kaggle