

# Neo4j

## Diseño de la BD

Neo4j es una base de datos que sirve como backend para una aplicación de gestión de la cadena de suministro. Modela entidades como regiones, naciones, proveedores, partes, líneas de ítems, pedidos y clientes, así como sus relaciones en una estructura de grafo. La aplicación tiene como objetivo admitir de manera eficiente consultas complejas relacionadas con el benchmark TPC-H, centrándose en análisis de la cadena de suministro.

## Estructura de Nodos y Relaciones

### Nodos

Region: Representa regiones geográficas.

Nation: Representa naciones, asociadas a una región específica.

Supplier: Representa proveedores con atributos como nombre, dirección y número de teléfono.

Part: Representa partes fabricadas por proveedores.

LineItem: Representa ítems en pedidos con detalles como cantidad, precio extendido y fecha de envío.

Order: Represents customer orders with details like order date and ship priority.

Customer: Representa clientes con atributos como segmento de mercado.

### Propiedades de los nodos

Region: regionkey name.

Nation: nationkey, name, y regionkey(Foreign key lincando nation a su region).

Supplier: suppkey, name, address, phone, comment, y nationkey(Foreign key lincando supplier a su nation).

Part: partkey, mfg, size, type.

LineItem: quantity, extendedprice, discount, tax, orderdate, shipdate.

Order: orderkey, orderdate, shippriority, custkey(Foreign key lincando order a su customer).

Customer: custkey, mktsegment, nationkey(Foreign key lincando customer a su nation).

### Relaciones

PART\_OF: Conecta naciones con sus respectivas regiones.

LOCATED\_IN: Asocia proveedores(suppliers) con las naciones en las que se encuentran.

SUPPLIED\_BY: Vincula partes(parts) con sus respectivos proveedores(suppliers).

CONTAINS: Asocia ítems de línea(line items) con partes(parts ) y pedidos(orders) con ítems de línea.

ORDERED\_BY: Conecta pedidos(orders) con clientes(customers).

BELONGS\_TO: Establece relaciones entre clientes(customers) y naciones, partes(parts) y proveedores(suppliers), y pedidos(orders) y regiones.

SUPPLY: Conecta partes(parts), proveedores(suppliers) e ítems de línea(line items) con información sobre el costo de suministro(ps\_supplycost).

## Querys en Cypher

Las consultas Cypher desempeñan un papel crucial en la extracción de información valiosa y análisis de la base de datos de grafo Neo4j, respaldando la aplicación de gestión de la cadena de suministro. Las consultas están diseñadas para navegar eficientemente la estructura del grafo y recuperar información relevante para la toma de decisiones y el análisis.

### Query 1 - Aggregated Lineltem

**MATCH:** Identifica todos los nodos de Lineltem que cumplen con la condición de tener una shipdate menor o igual a la fecha especificada.

**WITH:** Agrega y calcula diversas métricas (sum, avg, count) basadas en diferentes propiedades de Lineltem.

**RETURN:** OMuestra la información agregada.

**ORDER BY:** Ordena el resultado según returnflag y linestatus.

Esta consulta extrae información agregada de Lineltem, resumiendo cantidades, precios y descuentos, contribuyendo al análisis de la cadena de suministro.

### Query 2 - Supplier, Part, y Nation

**MATCH:** Recorre el grafo para conectar Part con Lineltem, Customer, Nationy Region a través de diversas relaciones.

**WHERE:** Filtra según condiciones especificadas, incluyendo size, type y region.

**ps.supplycost = (MATCH ...):** La subconsulta recupera el supplycost mínimo asociado con la misma part, supplier, y region.

**ORDER BY:** Ordena el resultado según acctbal, name y partkey.

Esta consulta obtiene información detallada sobre suppliers, partsy nations, considerando condiciones específicas y el supply cost mínimo.

Orders the result based on acctbal, name, and partkey.

### Query 3 - Customer Order y Lineltem

**MATCH:** Conecta Customer con nodos de Order y Lineltem.

**WHERE:** Aplica filtros según market segment,customer key, order key, order date range, y ship date range.

**WITH:** Agrupa datos y calcula los revenue totales para cada order.

**RETURN:** Muestra order key, revenue, order date, y ship priority.

**ORDER BY:** Ordena el resultado según revenue y order date en orden descendente.

Esta consulta proporciona información sobre customer orders and line items asociadas, filtrando por market segment y date ranges.

## Query 4 - Customer, Order, LineItem, Supplier, Nation, y Region

**MATCH:** Establece conexiones entre nodos de Customer, Order, LineItem, Supplier, Nation, y Region.

**WHERE:** Aplica condiciones para filtrar datos según propiedades de customer, order, line item, supplier, nation, region properties y date ranges.

**WITH:** Agrupa datos y calcula los revenue totales para cada nation.

**RETURN:** Muestra el nombre de la nation y los revenue totales.

**ORDER BY:** Ordena el resultado según revenue en orden descendente.

Esta consulta recupera información agregada de revenue para naciones dentro de una region y date range específicos, involucrando múltiples nodos y relaciones.

## Conclusión

El diseño de la base de datos Neo4j, utilizando un modelo de grafo con nodos y relaciones clave, demuestra ser eficiente para el análisis de la cadena de suministro. La selección estratégica de propiedades y la indexación contribuyen al rendimiento de las consultas. Las consultas Cypher, adaptadas a necesidades analíticas específicas, extraen eficientemente información valiosa, mostrando la efectividad de la estructura y las consultas elegidas para la gestión de la cadena de suministro.