yosr sellami



Neo4j mini project:

**E-commerce**

1. **Introduction :**

In this Neo4j mini project We'll model an e-commerce platform with various entities such as Clients, Orders, Products, Categories, and Invoices. We'll also include relationships like "PLACED\_ORDER", "CONTAINS\_PRODUCT", "BELONGS\_TO\_CATEGORY” etc…

1. **Step 1: Start Neo4j :**

the Neo4j Browser log in :

Instance is available

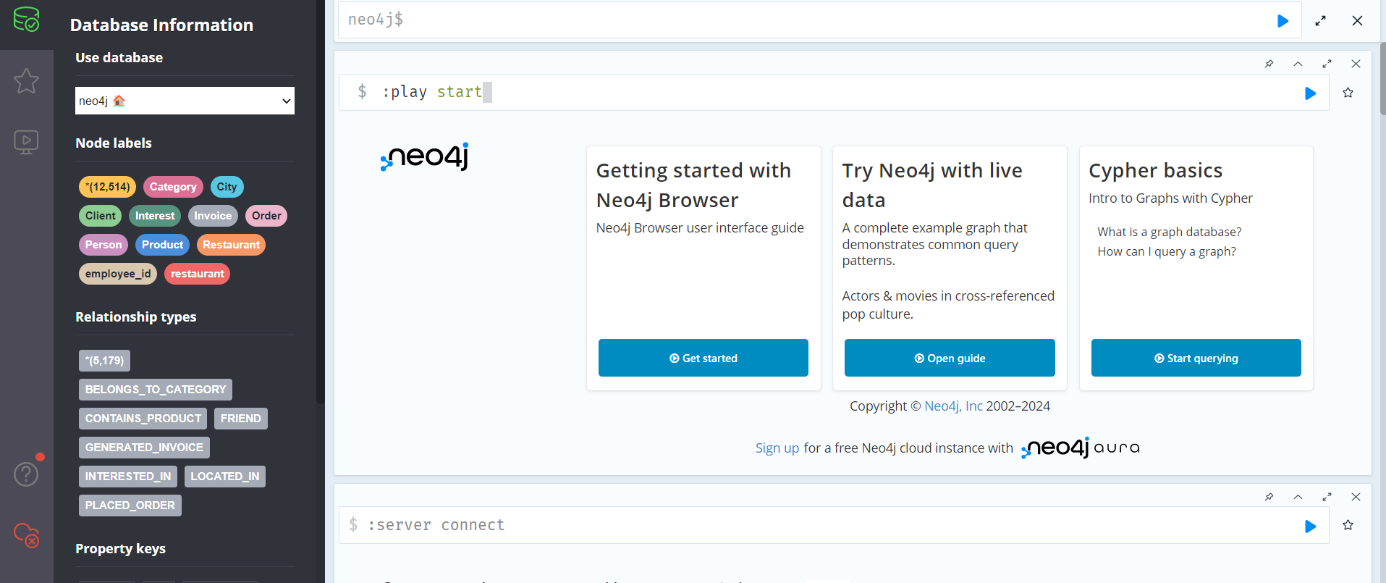
NEO4J\_URI=neo4j+s://af44b6c6.databases.neo4j.io

NEO4J\_USERNAME=neo4j

NEO4J\_PASSWORD=vod2TQ6BiMDFuiYEwh0ra7GleIvYr9XnoMevLKbuinA

AURA\_INSTANCEID=af44b6c6

AURA\_INSTANCENAME=Instance01



1. **Step 2: Define the Schema and Import Data**

Define nodes and relationships for our e-commerce platform. Using the following Cypher queries to create the data:

Create Client Nodes:

CREATE (alice:Client {client\_id: 1, name: 'Alice', email: 'alice@example.com', phone: '123-456-7890'})

CREATE (bob:Client {client\_id: 2, name: 'Bob', email: 'bob@example.com', phone: '234-567-8901'})

CREATE (carol:Client {client\_id: 3, name: 'Carol', email: 'carol@example.com', phone: '345-678-9012'})

Create Product Nodes:

CREATE (laptop:Product {product\_id: 1, name: 'Laptop', price: 1000, stock: 50})

CREATE (phone:Product {product\_id: 2, name: 'Smartphone', price: 700, stock: 100})

CREATE (tablet:Product {product\_id: 3, name: 'Tablet', price: 400, stock: 30})

Create Category Nodes:

CREATE (electronics:Category {category\_id: 1, name: 'Electronics'})

CREATE (appliances:Category {category\_id: 2, name: 'Home Appliances'})

Create Invoice Nodes:

CREATE (invoice1:Invoice {invoice\_id: 1, date: '2023-01-15', total: 1700})

CREATE (invoice2:Invoice {invoice\_id: 2, date: '2023-02-20', total: 1100})

Create Order Nodes:

CREATE (order1:Order {order\_id: 1, date: '2023-01-10', total: 1700})

CREATE (order2:Order {order\_id: 2, date: '2023-02-18', total: 1100})

Create Relationships between Clients and Orders:

CREATE (alice)-[:PLACED\_ORDER]->(order1)

CREATE (bob)-[:PLACED\_ORDER]->(order2)

Create Relationships between Orders and Products:

CREATE (order1)-[:CONTAINS\_PRODUCT]->(laptop)

CREATE (order1)-[:CONTAINS\_PRODUCT]->(phone)

CREATE (order2)-[:CONTAINS\_PRODUCT]->(tablet)

CREATE (order2)-[:CONTAINS\_PRODUCT]->(phone)

Create Relationships between Products and Categories:

CREATE (laptop)-[:BELONGS\_TO\_CATEGORY]->(electronics)

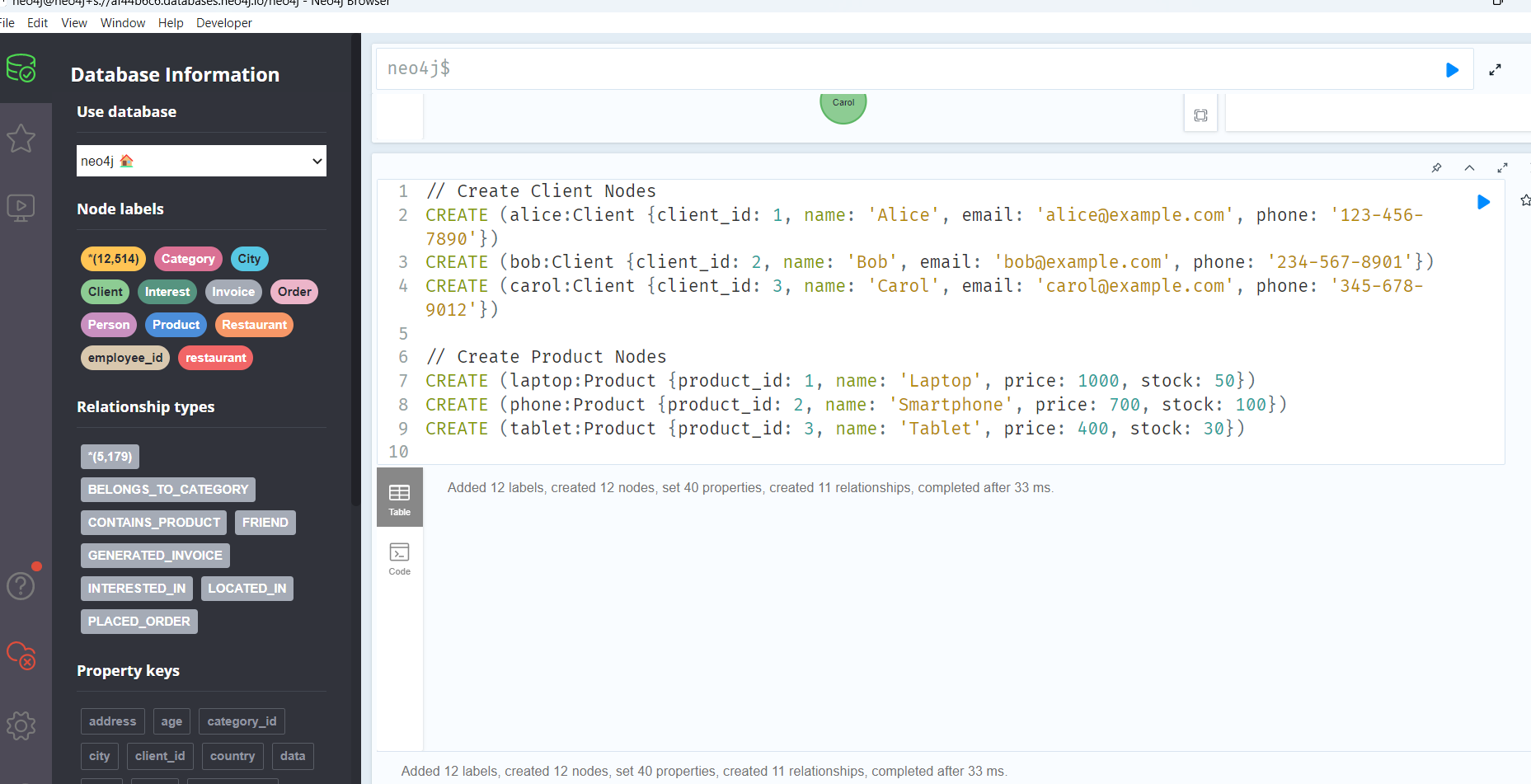
CREATE (phone)-[:BELONGS\_TO\_CATEGORY]->(electronics)

CREATE (tablet)-[:BELONGS\_TO\_CATEGORY]->(electronics)

Create Relationships between Orders and Invoices:

CREATE (order1)-[:GENERATED\_INVOICE]->(invoice1)

CREATE (order2)-[:GENERATED\_INVOICE]->(invoice2)

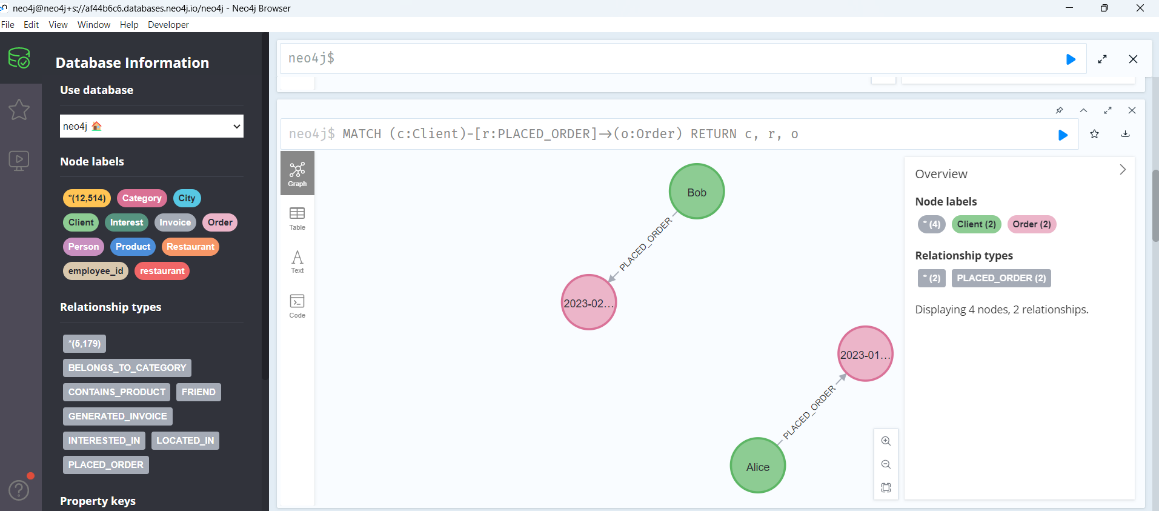


1. **Step 3: Query the Graph**

Now that we have a more complex schema, let's run some queries to explore our e-commerce platform.

1. **Get all clients and their orders**:

MATCH (c:Client)-[r:PLACED\_ORDER]->(o:Order)

RETURN c,r, o

1. **Get all orders and their products**:

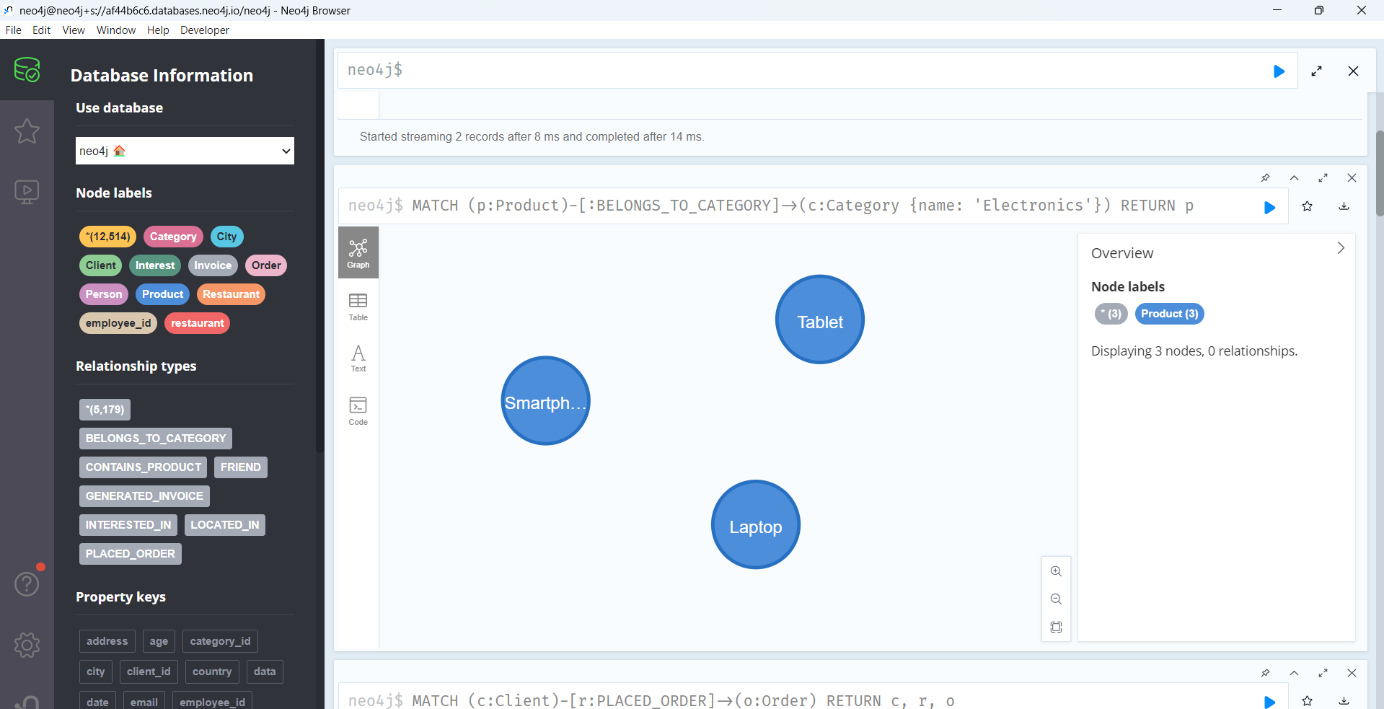
MATCH (o:Order)-[:CONTAINS\_PRODUCT]->(p:Product)

RETURN o, p

1. **Find all products in a specific category (e.g., Electronics)**:

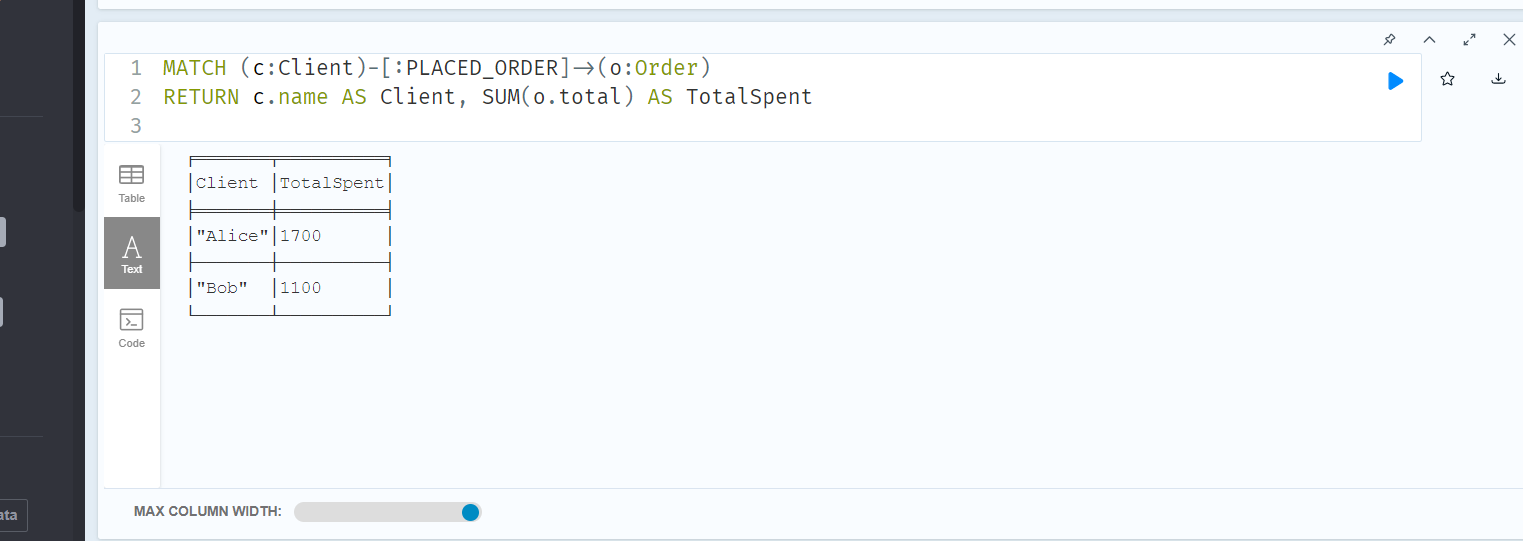
MATCH (p:Product)-[:BELONGS\_TO\_CATEGORY]->(c:Category {name: 'Electronics'})

RETURN p



1. **Find the total amount spent by each client**:

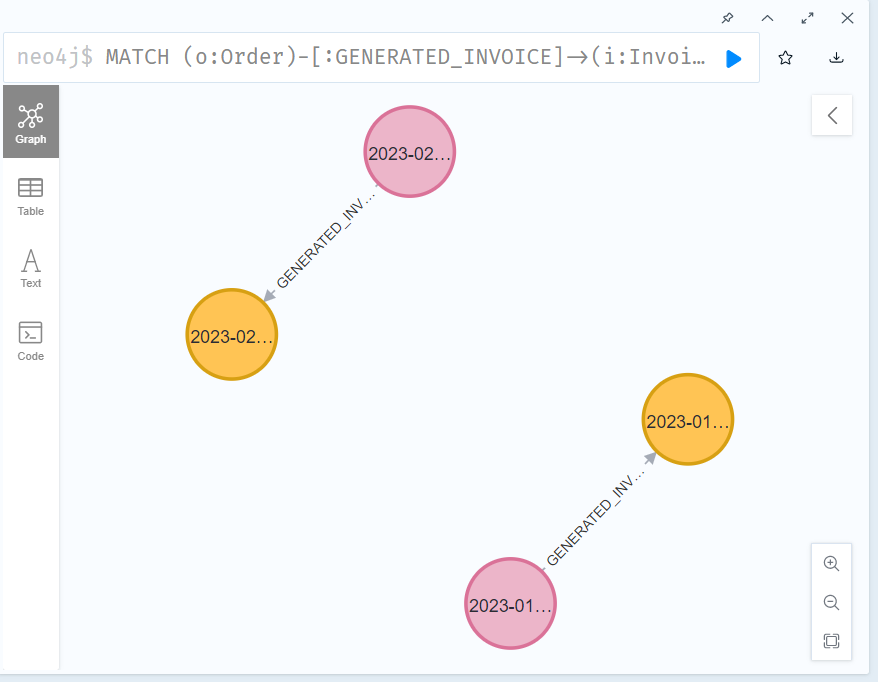
MATCH (c:Client)-[:PLACED\_ORDER]->(o:Order)

RETURN c.name AS Client, SUM(o.total) AS TotalSpent



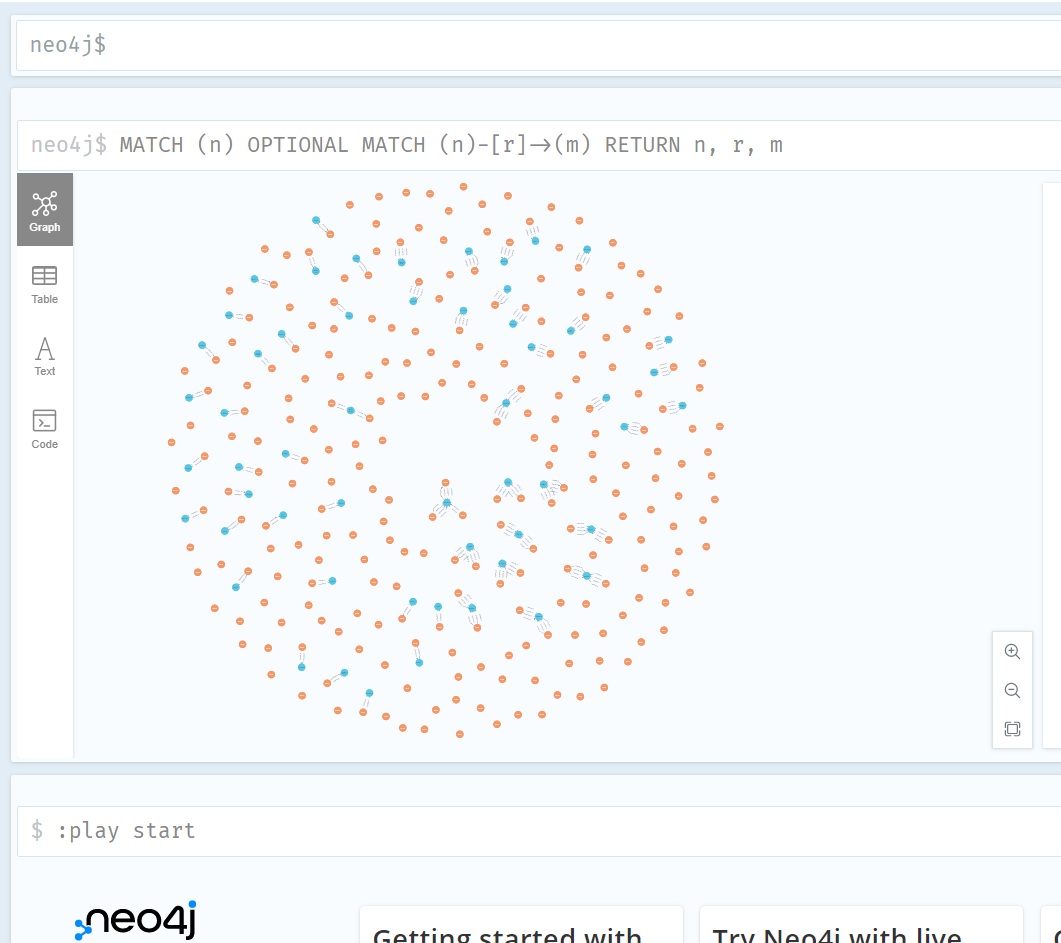
1. **Find all orders that generated invoices**:

MATCH (o:Order)-[:GENERATED\_INVOICE]->(i:Invoice)

****RETURN o, i

1. **Visualize the entire graph**:

MATCH (n) OPTIONAL MATCH (n)-[r]->(m) RETURN n, r, m

****