Міністерство освіти і науки, молоді та спорту України
Національний технічний університет України
“Київський політехнічний інститут”
Фізико-Технічний інститут

Лабораторна робота № 4
з семестрового курсу
“Проектування високонавантажених систем”

Виконала:
Студентка групи ФІ-03
Швець Катерина

Київ, 2023

Task 4 - Робота з базовими функціями граф-орієнтованої БД на прикладі Neo4j

- Без установки з Neo4j можна працювати онлайн на сайті http://neo4j.com/sandbox/ (виконувала лабораторну на сайті)

Завдання:

Змоделювати наступну предметну область:

- Є: Items, Customers, Orders
- Customer може додати Item(s) до Order (тобто купити Товар)
- У Customer може бути багато Orders
- Item може входити в багато Orders, і у Item є вартість
- Customer може переглядати (view), але при цьому не купувати Items

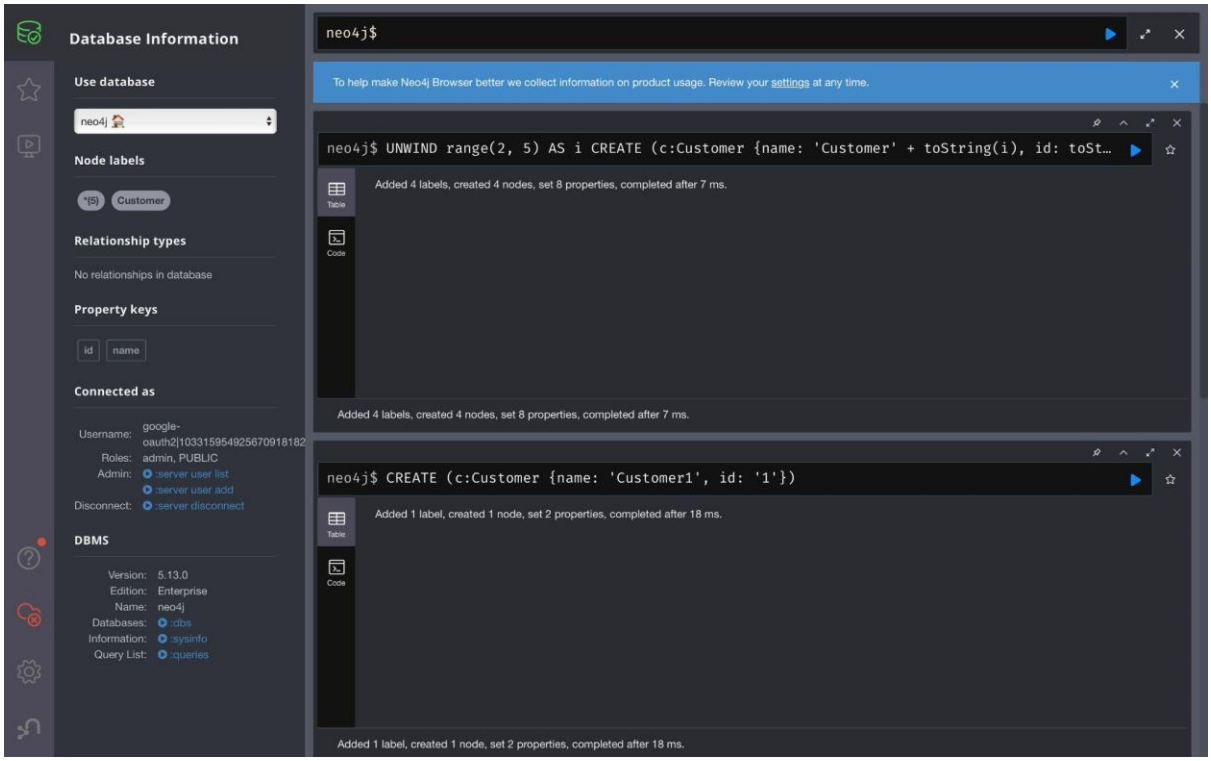
Структура Даних:

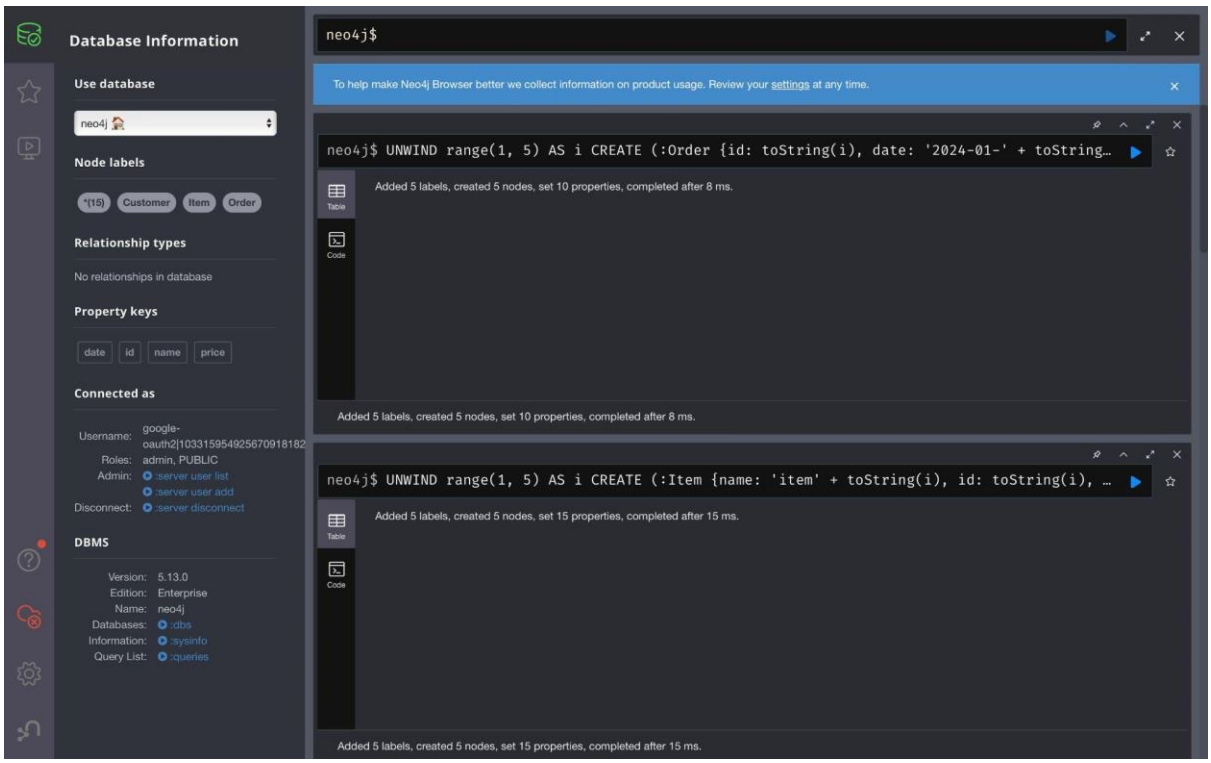
- Customer: Має атрибути, такі як ім'я, ідентифікатор тощо.
- Item: Має атрибути, такі як назва, вартість, ідентифікатор.
- Order: Зв'язує Customer та Item, може мати дату замовлення, ідентифікатор.

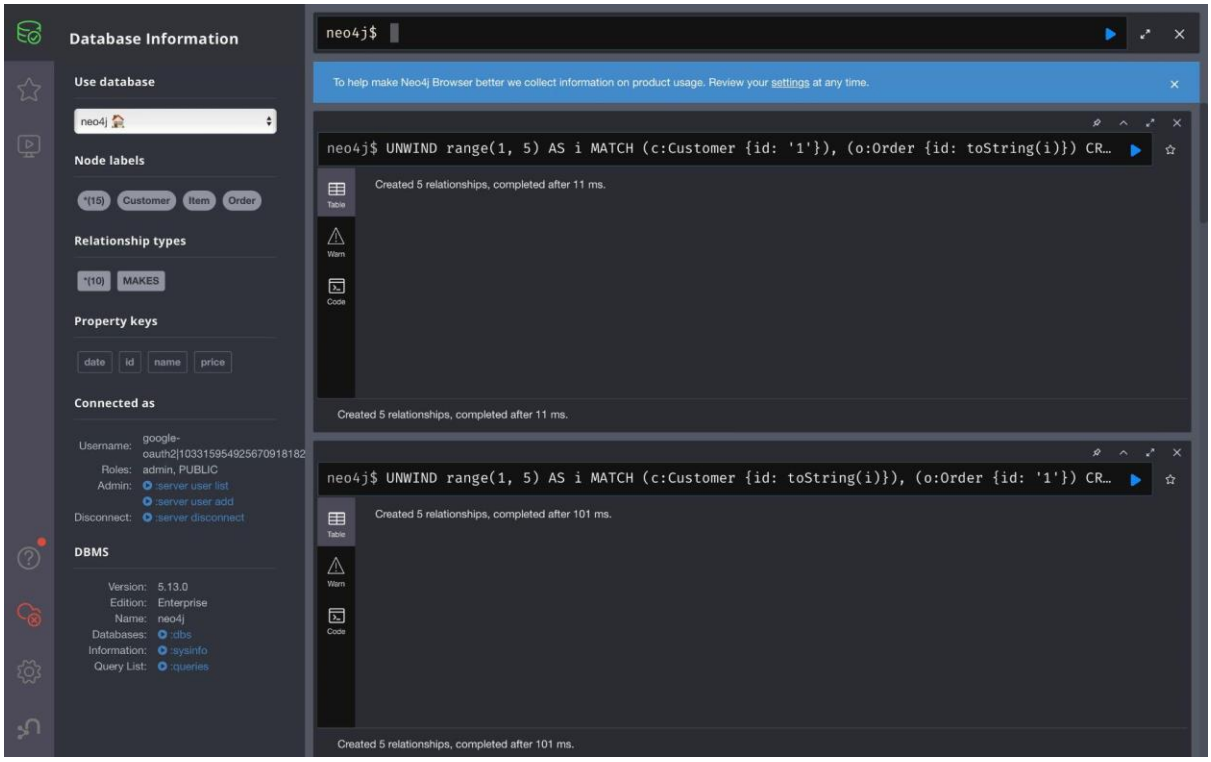
Зв'язки

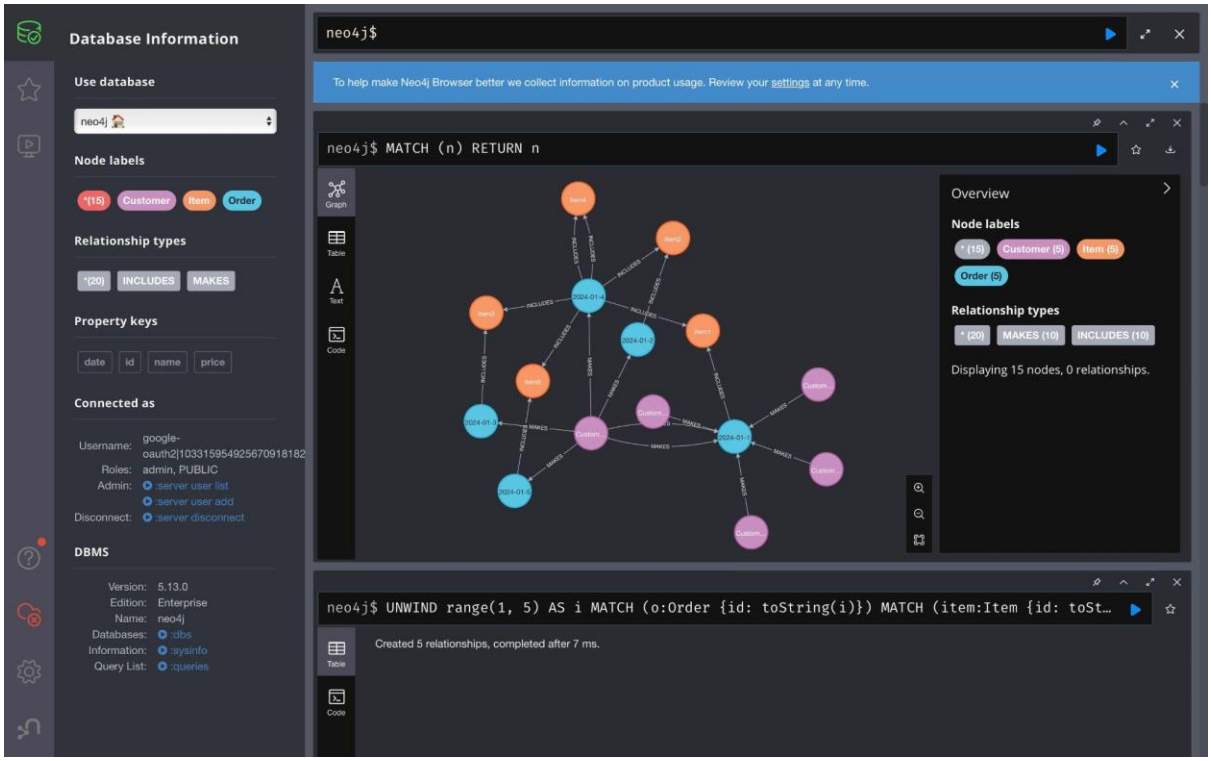
- Customer -[MAKES]-> Order: Клієнт робить замовлення.
- Order -[INCLUDES]-> Item: Замовлення включає товар.
- Customer -[VIEWS]-> Item: Клієнт переглядає товар.

Далі створю вузли та зв'язки вграфі. Для цього використовую мову запитів Cypher.



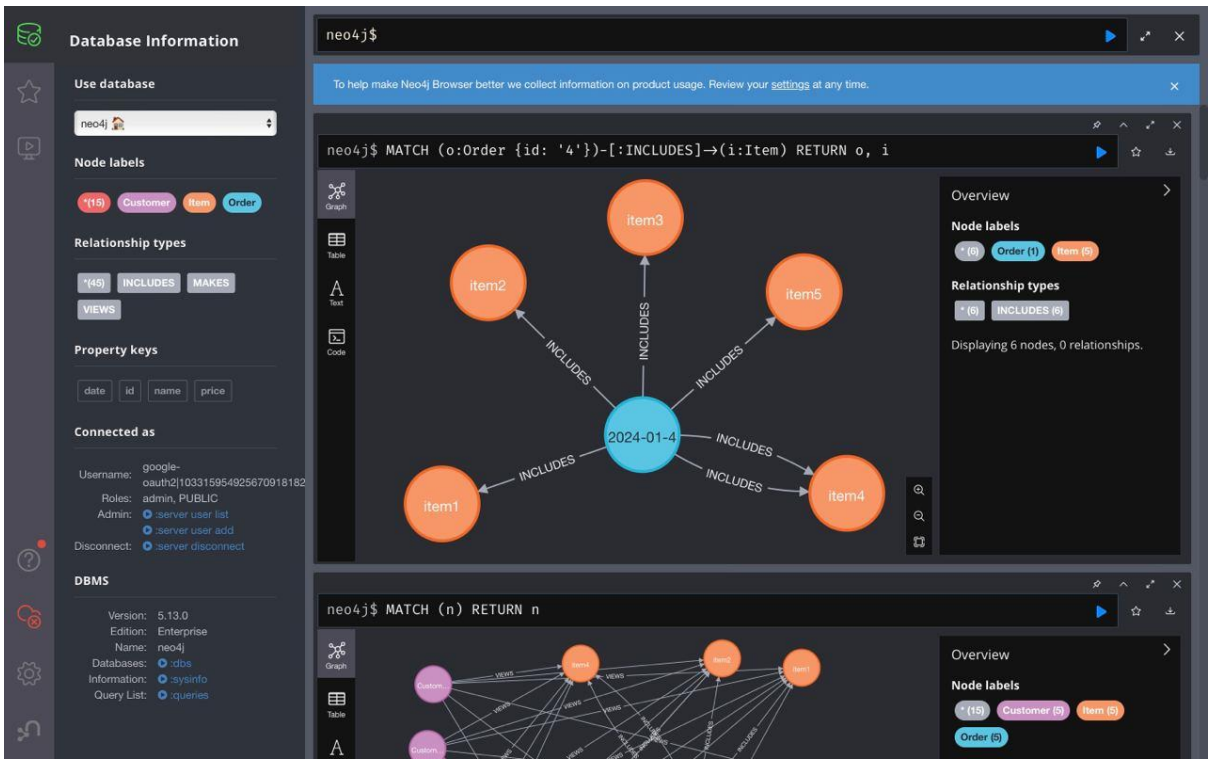

Database Information
Use database
neo4j
Node labels
*(15)
Customer
Item
Order
Relationship types
No relationships in database
Property keys
date
id
name
price
Connected as
Username: google-oauth2|103315954925670918182
Roles: admin, PUBLIC
Admin: :server user list
:server user add
Disconnect: :server disconnect
DBMS
Version: 5.13.0
Edition: Enterprise
Name: neo4j
Databases: :dbs
Information: :sysinfo
Query List: :queries
neo4j$
To help make Neo4j Browser better we collect information on product usage. Review your settings at any time.
neo4j$ UNWIND range(1, 5) AS i CREATE (:Order {id: toString(i), date: '2024-01-' + toString…
Table
Code
Added 5 labels, created 5 nodes, set 10 properties, completed after 8 ms.
Added 5 labels, created 5 nodes, set 10 properties, completed after 8 ms.
neo4j$ UNWIND range(1, 5) AS i CREATE (:Item {name: 'item' + toString(i), id: toString(i), …
Table
Code
Added 5 labels, created 5 nodes, set 15 properties, completed after 15 ms.
Added 5 labels, created 5 nodes, set 15 properties, completed after 15 ms.


Database Information
Use database
neo4j
Node labels
*(15)
Customer
Item
Order
Relationship types
*(10)
MAKES
Property keys
date
id
name
price
Connected as
Username: google-oauth2|103315954925670918182
Roles: admin, PUBLIC
Admin: :server user list
:server user add
Disconnect: :server disconnect
DBMS
Version: 5.13.0
Edition: Enterprise
Name: neo4j
Databases: :dbs
Information: :sysinfo
Query List: :queries
neo4j$
To help make Neo4j Browser better we collect information on product usage. Review your settings at any time.
neo4j$ UNWIND range(1, 5) AS i MATCH (c:Customer {id: '1'}), (o:Order {id: toString(i)}) CR…
Table
Warn
Code
Created 5 relationships, completed after 11 ms.
Created 5 relationships, completed after 11 ms.
neo4j$ UNWIND range(1, 5) AS i MATCH (c:Customer {id: toString(i)}), (o:Order {id: '1'}) CR…
Table
Warn
Code
Created 5 relationships, completed after 101 ms.
Created 5 relationships, completed after 101 ms.

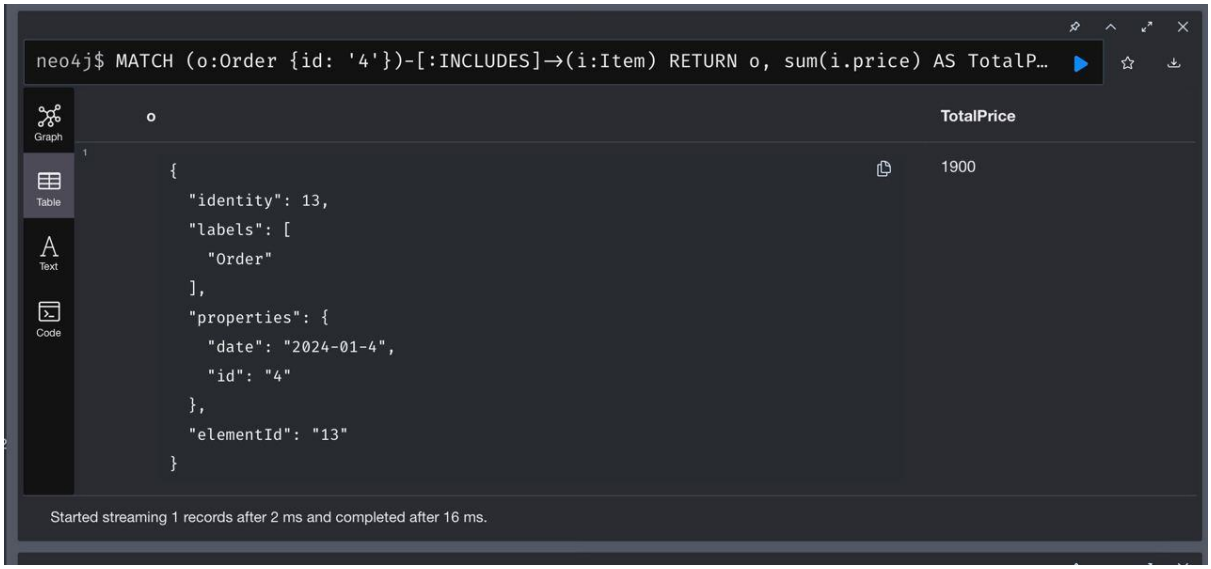


Написати наступні види запитів:

- Знайти Items які входять в конкретний Order



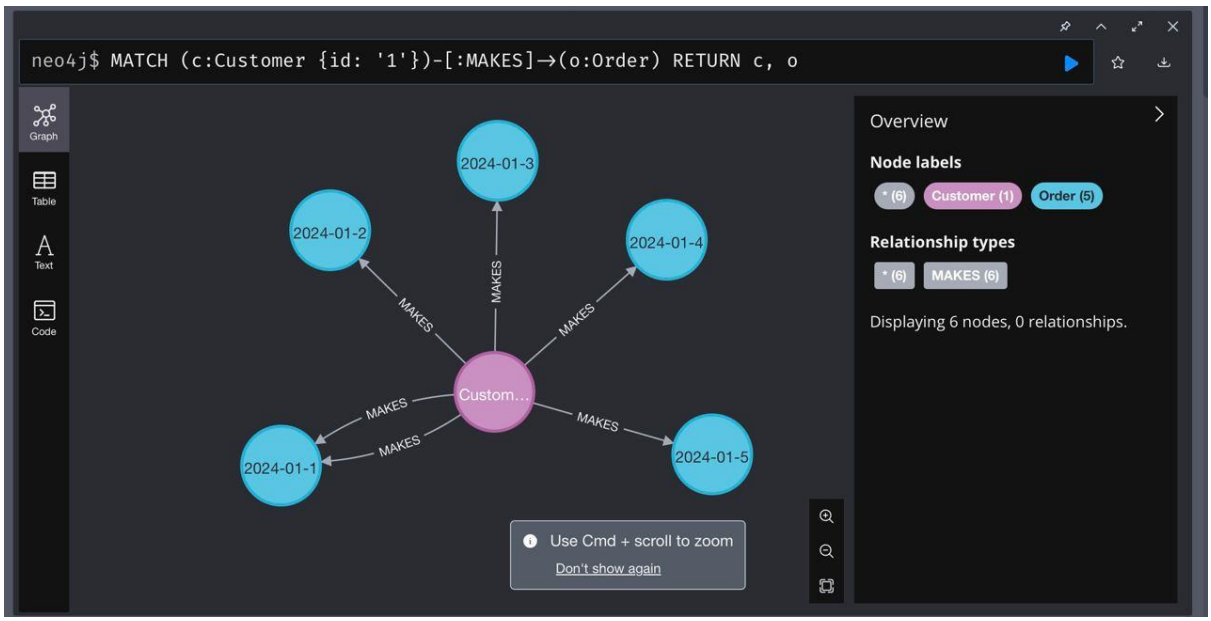
- Підрахувати вартість конкретного Order



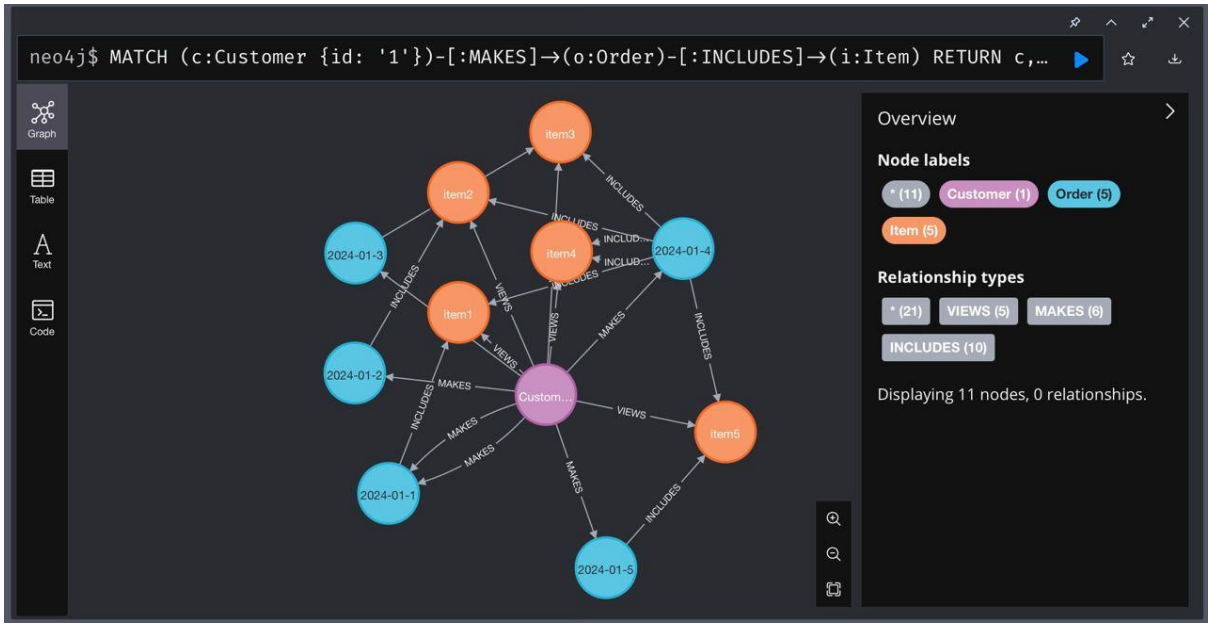

```
MATCH (o:Order {id: '4'})-[:INCLUDES]->(i:Item)
RETURN o, sum(i.price) AS TotalPrice
```

- Знайти всі Orders конкретного Customer

```
MATCH (c:Customer {id: '1'})-[:MAKES]->(o:Order)
RETURN c, o
```



- Знайти всі Items куплені конкретним Customer (через Order)




```
MATCH (c:Customer {id: '1'})-[:MAKES]->(o:Order)
RETURN c, o, i
```

- Знайти кількість Items куплені конкретним Customer (через Order)

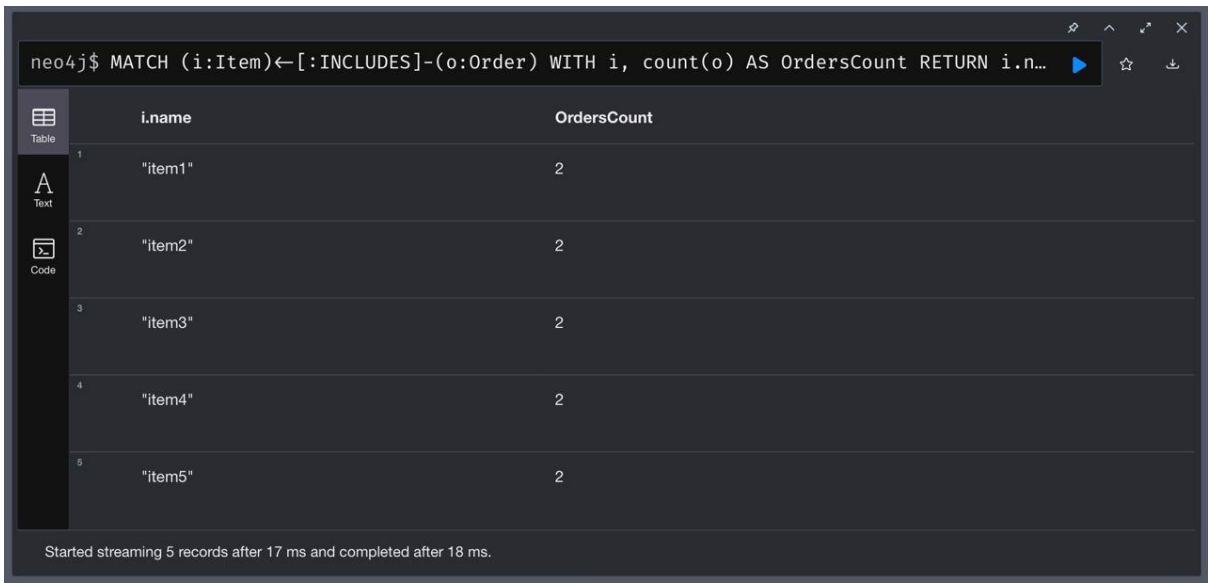
```
MATCH (c:Customer {id: '1'})-[:MAKES]->(o:Order)-[:INCLUDES]->(i:Item)
RETURN c, count(i) AS NumberOfItems
```

- Знайти для Customer на яку суму він придбав товарів (через Order)

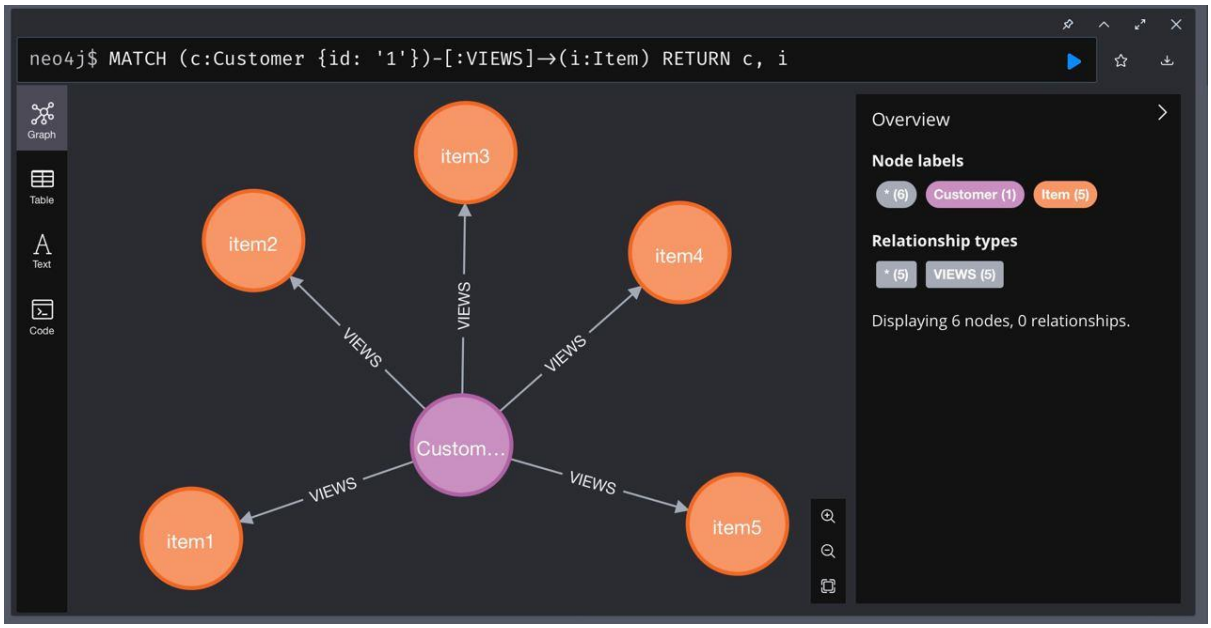

```
MATCH (c:Customer {id: '1'})-[:MAKES]->(o:Order)-[:INCLUDES]->(i:Item)
RETURN c, sum(i.price) AS TotalSpent
```

- Знайті скільки разів кожен товар був придбаний, відсортувати за цим значенням

```
MATCH (i:Item)<-[:INCLUDES]-(o:Order)
WITH i, count(o) AS OrdersCount
RETURN i.name, OrdersCount
ORDER BY OrdersCount DESC
```

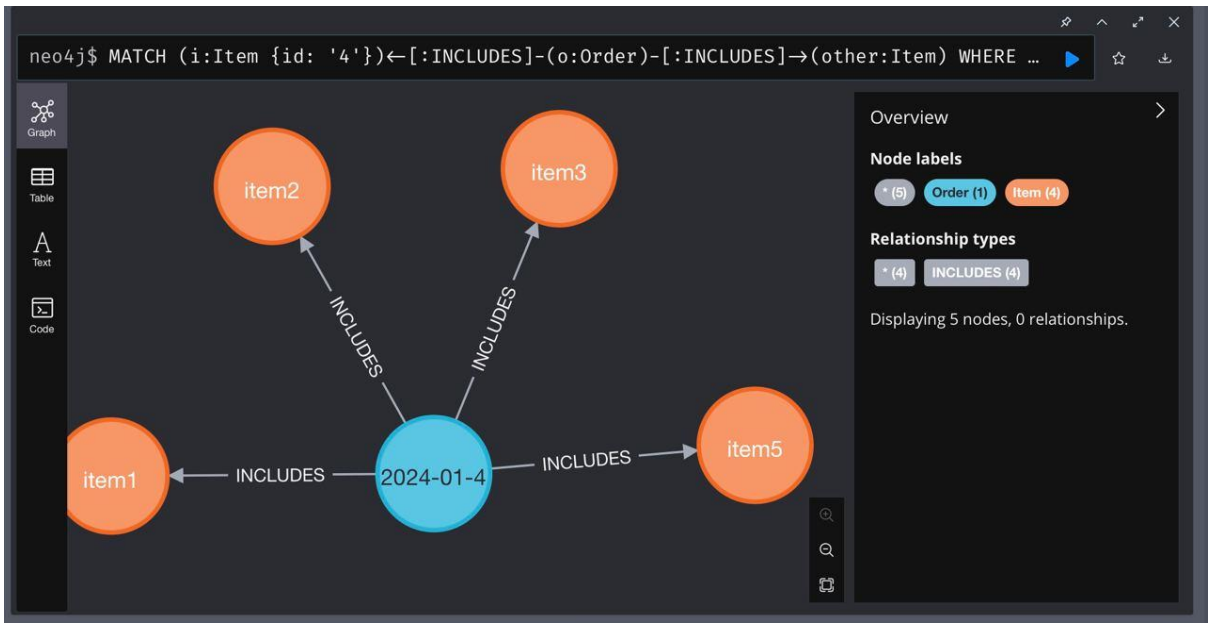


- Знайти всі Items переглянуті (view) конкретним Customer



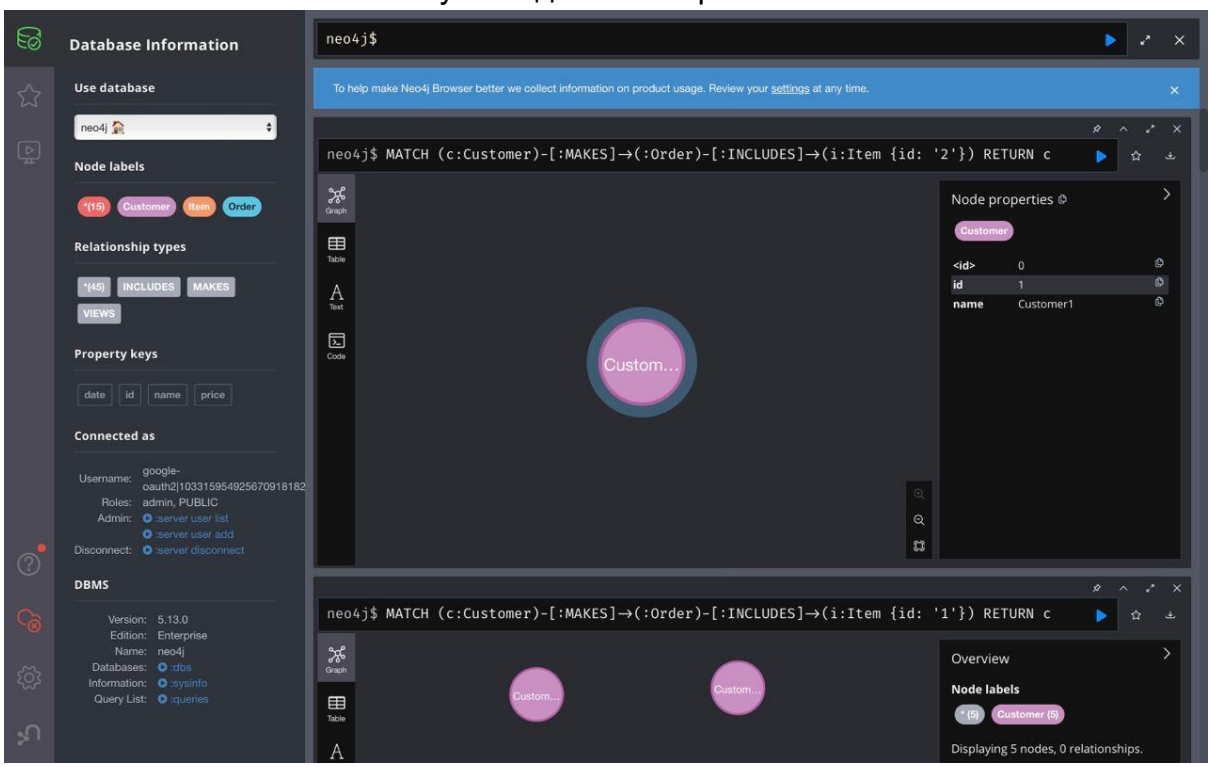

```
MATCH (c:Customer {id: '1'})-[:VIEWS]->(i:Item)
RETURN c, i
```

- Знайти інші Items що купувались разом з конкретним Item (тобто всі Items що входять до Order-s разом з даними Item)



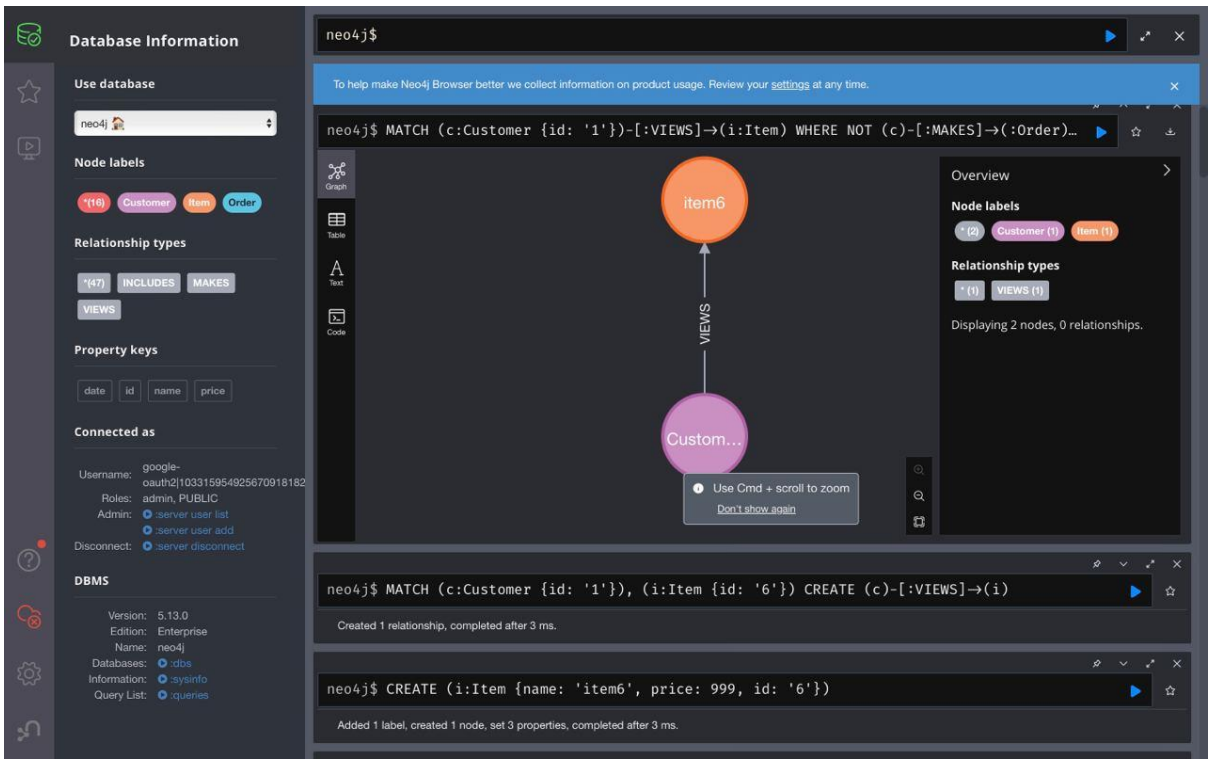

```
MATCH (i:Item {id: '4'})<-[:INCLUDES]-(o:Order)-[:INCLUDES]->(other:Item)
WHERE i <> other
RETURN o, other
```

- Знайти Customers які купили даний конкретний Item




```
MATCH (c:Customer)-[:MAKES]->(:Order)-[:INCLUDES]->(i:Item {id: '1'})
RETURN c
```

- Знайти для певного Customer(a) товари, які він переглядав, але не купив




```
MATCH (c:Customer {id: '1'})-[:VIEWS]->(i:Item)
WHERE NOT (c)-[:MAKES]->(:Order)-[:INCLUDES]->(i)
RETURN c, i
```