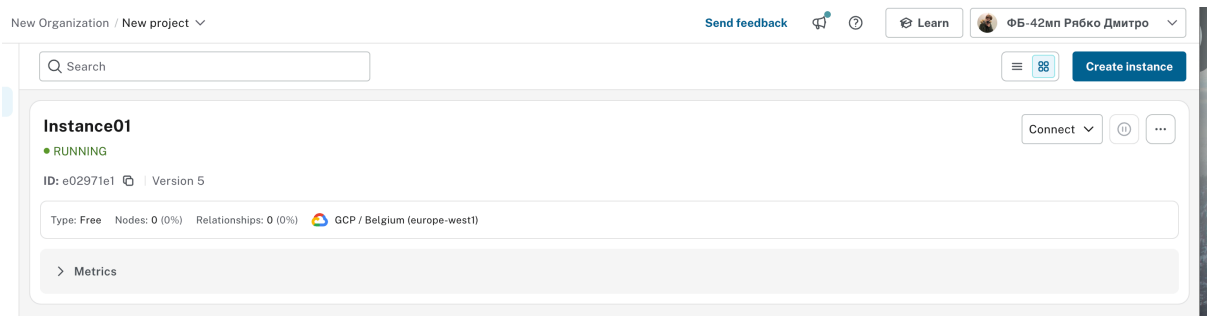
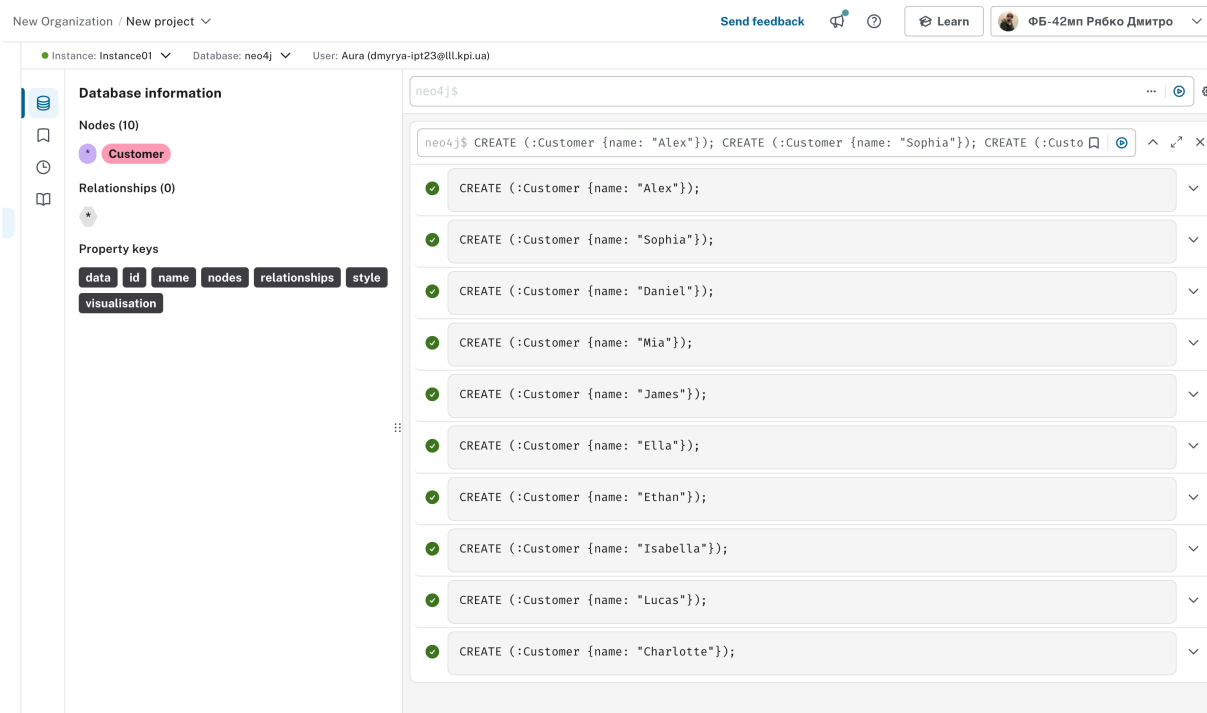


# Рябко Дмитро ФБ-42мп



## Створення клієнтів

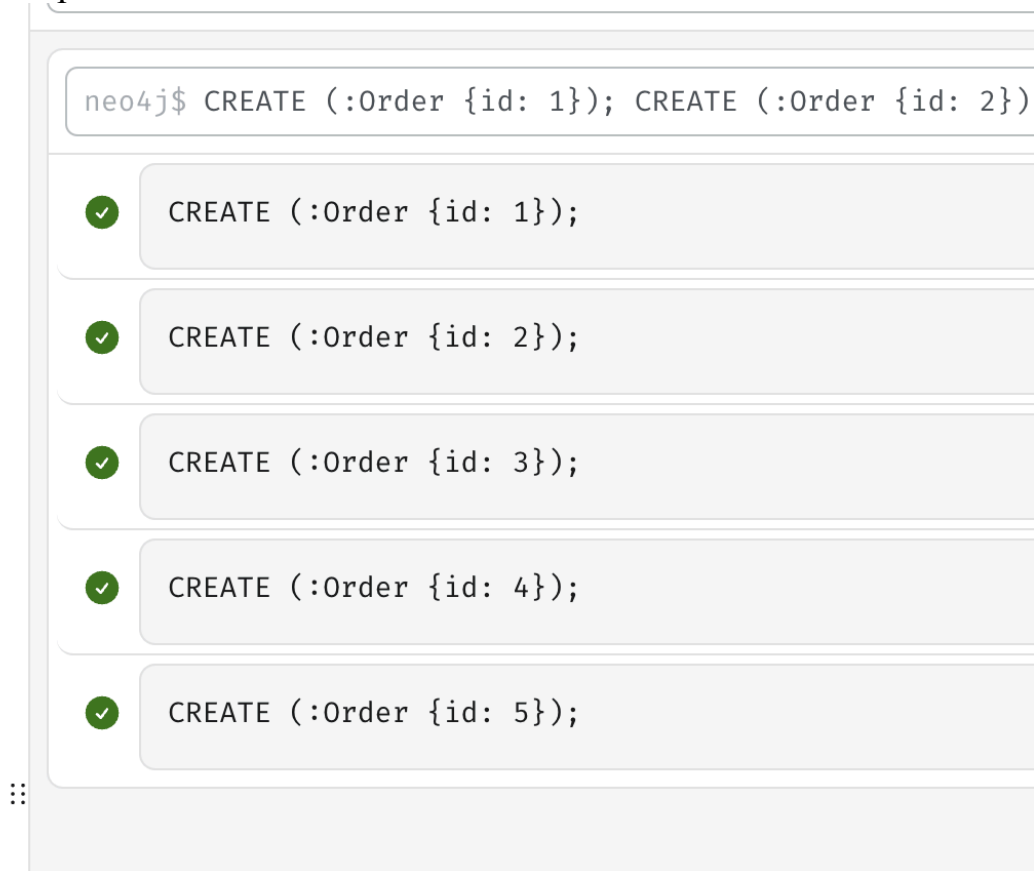


## Створення товару

```
1 CREATE (:Item {name: "Smartphone", price: 800, likes: 0});
2 CREATE (:Item {name: "Smartwatch", price: 250, likes: 0});
3 CREATE (:Item {name: "Camera", price: 1200, likes: 0});
4 CREATE (:Item {name: "Laptop", price: 1500, likes: 0});
5 CREATE (:Item {name: "Wireless Earbuds", price: 150, likes: 0});
```



Створення замовлень:



Клієнти та замовлення

```
neo4j$ MATCH (c:Customer {name: "Alex"}), (o:Order {id: 1}) CREATE (c)-[:PLACED]->(o);
```

✓

MATCH (c:Customer {name: "Alex"}), (o:Order {id: 1})

✓

MATCH (c:Customer {name: "Sophia"}), (o:Order {id: 2})

✓

MATCH (c:Customer {name: "Daniel"}), (o:Order {id: 3})

✓

MATCH (c:Customer {name: "Mia"}), (o:Order {id: 4})

✓

MATCH (c:Customer {name: "James"}), (o:Order {id: 5})

## Замовлення та товари:

```
1 MATCH (o:Order {id: 1}), (i:Item {name: "Smartphone"})
2 CREATE (o)-[:CONTAINS]->(i);
3
4 MATCH (o:Order {id: 1}), (i:Item {name: "Smartwatch"})
5 CREATE (o)-[:CONTAINS]->(i);
6
7 MATCH (o:Order {id: 2}), (i:Item {name: "Camera"})
8 CREATE (o)-[:CONTAINS]->(i);
9
10 MATCH (o:Order {id: 3}), (i:Item {name: "Laptop"})
11 CREATE (o)-[:CONTAINS]->(i);
12
13 MATCH (o:Order {id: 4}), (i:Item {name: "Wireless Earbuds"})
14 CREATE (o)-[:CONTAINS]->(i);
15
16 MATCH (o:Order {id: 5}), (i:Item {name: "Smartphone"})
17 CREATE (o)-[:CONTAINS]->(i);
18
19 MATCH (o:Order {id: 5}), (i:Item {name: "Laptop"})
20 CREATE (o)-[:CONTAINS]->(i);
21
```

✓

MATCH (o:Order {id: 1}), (i:Item {name: "Smartphone"})

✓

MATCH (o:Order {id: 1}), (i:Item {name: "Smartwatch"})

✓

MATCH (o:Order {id: 2}), (i:Item {name: "Camera"})

✓

MATCH (o:Order {id: 3}), (i:Item {name: "Laptop"})

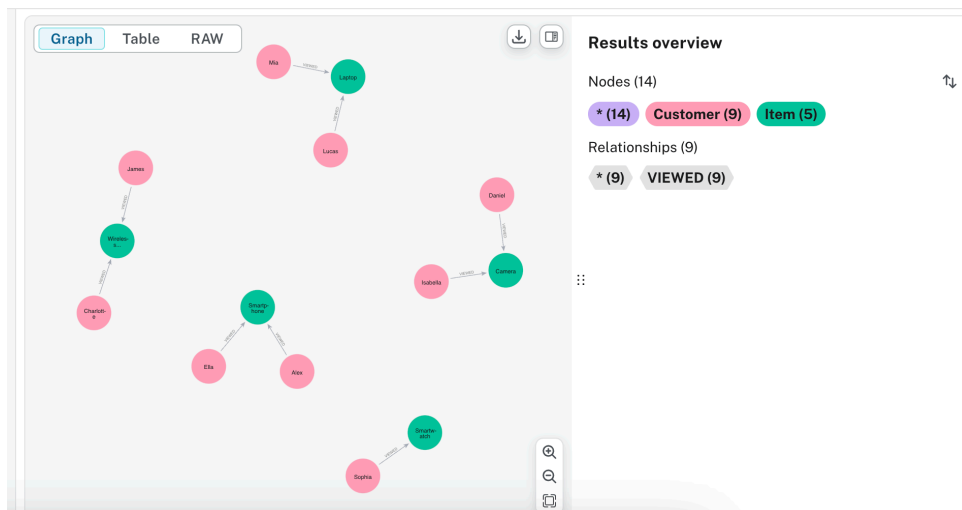
✓

MATCH (o:Order {id: 4}), (i:Item {name: "Wireless Earbuds"})

✓

MATCH (o:Order {id: 5}), (i:Item {name: "Smartphone"})

Перегляди товарів:  
(втратив випадково скріншот з виконанням)



Результат:

Nodes (20)

\* Customer Item Order

Relationships (21)

\* CONTAINS PLACED VIEWED

Property keys

data id likes name nodes price

Написати наступні види запитів:

- Знайти Items які входять в конкретний Order

```
1 MATCH (o:Order {id: 1})-[:CONTAINS]->(i:Item)
2 RETURN i.name AS ItemName;
3 |
```

Table RAW

ItemName

1 "Smartphone"

2 "Smartwatch"

- Підрахувати вартість конкретного Order

```
1 MATCH (o:Order {id: 1})-[:CONTAINS]->(i:Item)
2 RETURN SUM(i.price) AS Total_Cost;
3
```

Table RAW

Total\_Cost

1 1050

- Знайти всі Orders конкретного Customer

```
1 MATCH (c:Customer {name: "Sophia"})-[:PLACED]->(o:Order)
2 RETURN o.id AS OrderID;
3
4
```

Table RAW

OrderID

1 2

- Знайти всі Items куплені конкретним Customer (через Order)

```

1 MATCH (c:Customer {name: "Sophia"})-[:PLACED]->(:Order)-[:CONTAINS]->(i:Item)
2 RETURN i.name AS ItemName;
3

```

Table RAW

ItemName

1	"Camera"
---	----------

Started streaming 1 record after 64 ms and completed after

- Знайти кількість Items куплені конкретним Customer (через Order)

```

1 MATCH (c:Customer {name: "Alex"})-[:PLACED]->(:Order)-[:CONTAINS]->(i:Item)
2 RETURN COUNT(i) AS ItemCount;
3
4

```

Table RAW

ItemCount

1	2
---	---

Started streaming 1

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

```

1 MATCH (c:Customer {name: "James"})-[:PLACED]->(:Order)-[:CONTAINS]->(i:Item)
2 RETURN SUM(i.price) AS Total_Spent;
3
4

```

Table RAW

Total\_Spent

1	2300
---	------

Started stream

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

```

1 MATCH (:Order)-[:CONTAINS]->(i:Item)
2 RETURN i.name AS ItemName, COUNT(*) AS TimesPurchased
3 ORDER BY TimesPurchased DESC;
4
5

```

Table RAW

ItemName	TimesPurchased
1 "Smartphone"	2
2 "Laptop"	2
3 "Smartwatch"	1
4 "Camera"	1
5 "Wireless Earbuds"	1

Started streaming

- Знайти всі Items переглянуті (view) конкретним Customer

```

1 MATCH (c:Customer {name: "Alex"})-[:VIEWED]->(i:Item)
2 RETURN i.name AS ViewedItem;
3

```

Table RAW

ViewedItem

1 "Smartphone"

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

```

1 MATCH (o:Order)-[:CONTAINS]->(i:Item {name: "Smartphone"})
2 MATCH (o)-[:CONTAINS]->(other:Item)
3 WHERE other.name <> "Smartphone"
4 RETURN other.name AS CoPurchasedItems;
5

```

Table RAW

CoPurchasedItems

1 "Smartwatch"

2 "Laptop"

- Знайти Customers які купили даний конкретний Item

```

1 MATCH (c:Customer)-[:PLACED]->(o:Order)-[:CONTAINS]->(i:Item {name: "Smartphone"})
2 RETURN c.name AS CustomerName;
3
4

```

Table RAW

CustomerName

1 "Alex"

2 "James"

Started streaming 2 records

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

```

1 MATCH (c:Customer {name: "James"})-[:VIEWED]->(i:Item)
2 WHERE NOT (c)-[:PLACED]->(o:Order)-[:CONTAINS]->(i)
3 RETURN i.name AS NotPurchasedItems;
4
5

```

Table RAW

NotPurchasedItems

1 "Wireless Earbuds"



1. Як і в попередніх завданнях, для якогось одного обраного Item додайте поле з кількістю його лайків.
- З 10 окремих клієнтів одночасно запустити інкрементацію каунтеру лайків по 10\_000 на кожного клієнта
- зробіть так щоб не було втрат та перевірте щоб фінальне значення було 100\_000
- заміряйте час роботи

```
def increment_likes(item_name, increments):
    driver = GraphDatabase.driver(
        URI, auth=(USER, PASSWORD)
    )
    with driver.session() as session:
        for _ in range(increments):
            session.run(
                """
                MATCH (i:Item {name: $item_name})
                SET i.likes = i.likes + 1
                """
            ,
            item_name=item_name,
        )
    driver.close()
```

```
neo4j$ MATCH (i:Item {name: "Smartphone"}) RETURN i.likes;
```

Table RAW

i.likes

1 59070

Started stream

```
neo4j$ MATCH (i:Item {name: "Smartphone"}) RETURN i.likes;
```

Table RAW

i.likes

1 80275

Started streaming 1 record

```
neo4j$ MATCH (i:Item {name: "Smartphone"}) RETURN i.likes;
```

Table RAW

i.likes

1 100000

Started streamir

```
(kali@kali)-[~/Desktop/pvs]
$ /bin/python3 /home/kali/Desktop/pvs/lab3_riabko.py
Increment completed: Smartphone!
Total time: 421.91 sec.
```