



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Проектування високонавантажених систем

Лабораторна робота №3

Робота з базовими функціями граф-орієнтованої
БД на прикладі Neo4j

Перевірив:
Родіонов А. М.

Виконав:
студент I курсу
групи ФБ-41мп
Сахній Н. Р.

Київ 2024

Мета роботи: Необхідно змодельовати предметну область у граф-орієнтованій БД на прикладі Neo4j, розробити та виконати Cypher-запити для аналізу даних, а також реалізувати механізм одночасного оновлення каунтера з використанням плагінів Neo4j та оцінити його ефективність.

Завдання до виконання:

0. INITIALIZATION of Neo4j:

1) cat ./docker-compose.yml

```
1  services:
2      neo4j:
3          image: neo4j:4.4-community
4          container_name: neo4j_graph
5          ports:
6              - "7474:7474" # HTTP-порт для браузера Neo4j
7              - "7687:7687" # Bolt-порт для Cypher-клієнтів
8          volumes:
9              - ./neo4j/data:/data
10             - ./neo4j/logs:/logs
11          environment:
12              - NEO4J_AUTH=neo4j/UIOP1234
13              - NEO4J_ACCEPT_LICENSE_AGREEMENT=yes
14              - NEO4JLABS_PLUGINS=["graph-data-science", "apoc"]
15  volumes:
16      neo4j:
17          driver: local
18
```

2) docker-compose up

```
t-1000@DESKTOP-DRI0PBB MINGW64 /d/KPI/5 курс Магістрат/Проектування ВС/Task_3-Neo4jBasics
$ docker-compose up
Pulling neo4j (neo4j:4.4-community)...
4.4-community: Pulling from library/neo4j
Digest: sha256:a646bfed7ad1c6d2dcd8026d407473834f849b62da829b61e626a388bfdcc840
Status: Downloaded newer image for neo4j:4.4-community
Recreating neo4j_graph ...
Recreating neo4j_graph ... done
Attaching to neo4j_graph
neo4j_graph | Changed password for user 'neo4j'. IMPORTANT: this change will only take effect if performed before the datab
neo4j_graph | Fetching versions.json for Plugin 'graph-data-science' from https://graphdatascience.ninja/versions.json
neo4j_graph | Installing Plugin 'graph-data-science' from https://graphdatascience.ninja/neo4j-graph-data-science-2.6.8.jar
neo4j_graph | Applying default values for plugin graph-data-science to neo4j.conf
neo4j_graph | Fetching versions.json for Plugin 'apoc' from https://neo4j-contrib.github.io/neo4j-apoc-procedures/versions.
neo4j_graph | Installing Plugin 'apoc' from https://github.com/neo4j-contrib/neo4j-apoc-procedures/releases/download/4.4.0.
neo4j_graph | Applying default values for plugin apoc to neo4j.conf
neo4j_graph | 2024-12-12 10:04:24.655+0000 INFO Starting...
neo4j_graph | 2024-12-12 10:04:25.145+0000 INFO This instance is ServerId[d74c5a11] (d74c5a11-ccde-4d40-ab1c-7a67d1dc0fe2)
neo4j_graph | SLF4J(W): No SLF4J providers were found.
neo4j_graph | SLF4J(W): Defaulting to no-operation (NOP) logger implementation
neo4j_graph | SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
neo4j_graph | SLF4J(W): Class path contains SLF4J bindings targeting slf4j-api versions 1.7.x or earlier.
neo4j_graph | SLF4J(W): Ignoring binding found at [jar:file:/var/lib/neo4j/lib/slf4j-nop-1.7.30.jar!/org/slf4j/impl/StaticL
neo4j_graph | SLF4J(W): See https://www.slf4j.org/codes.html#ignoredBindings for an explanation.
neo4j_graph | 2024-12-12 10:04:26.372+0000 INFO ===== Neo4j 4.4.39 =====
neo4j_graph | 2024-12-12 10:04:26.483+0000 INFO Progress tracking: enabled
neo4j_graph | 2024-12-12 10:04:26.484+0000 INFO Memory usage guard: minimum estimate
neo4j_graph | 2024-12-12 10:04:26.488+0000 INFO GDS license file: null
neo4j_graph | 2024-12-12 10:04:26.490+0000 INFO GDS license state: unlicensed
neo4j_graph | 2024-12-12 10:04:26.498+0000 INFO Register GraphDataScience...
neo4j_graph | 2024-12-12 10:04:26.519+0000 INFO GraphDataScience registered.
```

```

neo4j_graph | 2024-12-12 10:04:26.520+0000 INFO Register GraphDataScienceExtras...
neo4j_graph | 2024-12-12 10:04:26.521+0000 INFO GraphDataScienceExtras registered.
neo4j_graph | 2024-12-12 10:04:26.523+0000 INFO Register MetricsFacade...
neo4j_graph | 2024-12-12 10:04:26.525+0000 INFO MetricsFacade registered.
neo4j_graph | 2024-12-12 10:04:26.526+0000 INFO Register Legacy Task Store/ Registry...
neo4j_graph | 2024-12-12 10:04:26.527+0000 INFO Task Store/ Registry registered.
neo4j_graph | 2024-12-12 10:04:26.528+0000 INFO Register Legacy User Log Registry...
neo4j_graph | 2024-12-12 10:04:26.528+0000 INFO User Log Registry registered.
neo4j_graph | 2024-12-12 10:04:26.900+0000 INFO GDS compatibility: for Neo4j Settings 5.x (placeholder) -- not available,
neo4j_graph | 2024-12-12 10:04:26.901+0000 INFO GDS compatibility: for Neo4j 4.4 -- available, for Neo4j 5.11 (placeholder
neo4j_graph | 2024-12-12 10:04:26.901+0000 INFO GDS compatibility: for Neo4j 5.14 (placeholder) -- not available, for Neo4j 5.15 (placeholder) -- not available, fo
neo4j_graph | 2024-12-12 10:04:26.901+0000 INFO GDS compatibility: for Neo4j RC (placeholder) -- not available, for Neo4j 5.19 (placeholder) -- not available, for Neo4j 5.20 (placeholde
neo4j_graph | 2024-12-12 10:05:17.934+0000 INFO Initializing system graph model for component 'security-users' with versio
neo4j_graph | 2024-12-12 10:05:17.949+0000 INFO Setting up initial user from 'auth.ini' file: neo4j
neo4j_graph | 2024-12-12 10:05:17.950+0000 INFO Creating new user 'neo4j' (passwordChangeRequired=false, suspended=false)
neo4j_graph | 2024-12-12 10:05:18.295+0000 INFO Setting version for 'security-users' to 3
neo4j_graph | 2024-12-12 10:05:18.414+0000 INFO After initialization of system graph model component 'security-users' have
neo4j_graph | 2024-12-12 10:05:18.628+0000 INFO Performing postInitialization step for component 'security-users' with ver
neo4j_graph | 2024-12-12 10:05:43.457+0000 INFO Called db.clearQueryCaches(): Query cache already empty.
neo4j_graph | 2024-12-12 10:05:49.304+0000 INFO Bolt enabled on 0.0.0.0:7687.
neo4j_graph | 2024-12-12 10:05:50.280+0000 INFO Remote interface available at http://localhost:7474/
neo4j_graph | 2024-12-12 10:05:50.285+0000 INFO id: 527B4C6B21CEB7A7D3C1F79E656AD3DC141C1F7D3A25074DAF979A1B763A5F8
neo4j_graph | 2024-12-12 10:05:50.285+0000 INFO name: system
neo4j_graph | 2024-12-12 10:05:50.285+0000 INFO creationDate: 2024-12-12T10:04:34.046Z
neo4j_graph | 2024-12-12 10:05:50.286+0000 INFO Started.

```

3) Connect to Neo4j: <http://localhost:7474/browser/>

Database access not available. Please use `:server connect` to establish connection. There's a graph waiting for you.

`$:server connect`

Connect to Neo4j

Database access might require an authenticated connection

Connect URL
neo4j:// localhost:7687

Database - leave empty for default

Authentication type
Username / Password

Username
neo4j

Password

Connect

`$:server connect`

Connected to Neo4j

Nice to meet you.

You are connected as user **neo4j** to **neo4j://localhost:7687**

Connection credentials are stored in your web browser.

+ MODELLING of SUBJECT AREA

- Існують: **Items** (Товари), **Customers** (Покупці), **Orders** (Замовлення).

```
// Додавання товарів (Items)
CREATE (i1:Item {id: 1, name: 'HDR Monitor', price: 6549})
CREATE (i2:Item {id: 2, name: 'Gaming Laptop', price: 45999})
CREATE (i3:Item {id: 3, name: 'Smartphone Pro', price: 32999})
CREATE (i4:Item {id: 4, name: 'Vacuum Cleaner', price: 10999})

// Додавання покупців (Customers)
CREATE (c1:Customer {id: 1, nickname: 'Neo', email: 'neonrider@electro.com'})
CREATE (c2:Customer {id: 2, nickname: 'Cosmos', email: 'starflux@xyzmail.com'})
CREATE (c3:Customer {id: 3, nickname: 'Aero', email: 'pantomblue@webmail.com'})

// Додавання замовлень (Orders)
CREATE (o1:Order {id: 1, date: '2024-11-30'})
CREATE (o2:Order {id: 2, date: '2024-12-02'})
CREATE (o3:Order {id: 3, date: '2024-12-06'})
CREATE (o4:Order {id: 4, date: '2024-12-06'})
CREATE (o5:Order {id: 5, date: '2024-12-12'})
```

neo4j\$ MATCH (n) RETURN n LIMIT 25

Graph

Table

Text

Code

n
(:Item {price: 6549,name: "HDR Monitor",id: 1})
(:Item {price: 45999,name: "Gaming Laptop",id: 2})
(:Item {price: 32999,name: "Smartphone Pro",id: 3})
(:Item {price: 10999,name: "Vacuum Cleaner",id: 4})
(:Customer {nickname: "Neo",id: 1,email: "neonrider@electro.com"})
(:Customer {nickname: "Cosmos",id: 2,email: "starflux@xyzmail.com"})
(:Customer {nickname: "Aero",id: 3,email: "pantomblue@webmail.com"})
(:Order {date: "2024-11-30",id: 1})
(:Order {date: "2024-12-02",id: 2})
(:Order {date: "2024-12-06",id: 3})
(:Order {date: "2024-12-06",id: 4})
(:Order {date: "2024-12-12",id: 5})

- “Покупець” може додавати “Товари” до “Замовлення”, т.б. купити їх.
- Кожен “Покупець” може мати декілька “Замовлень” (доповнення)
- “Товари” можуть входити до кількох “Замовлень”, і вони мають ціну.
- “Покупець” може переглядати “Товари” без здійснення покупки.

```
// Зв'язки між покупцями та замовленнями (покупець робить замовлення)
MATCH (c1:Customer {id: 1}), (c2:Customer {id: 2}), (c3:Customer {id: 3}),
      (o1:Order {id: 1}), (o2:Order {id: 2}), (o3:Order {id: 3}),
      (o4:Order {id: 4}), (o5:Order {id: 5})

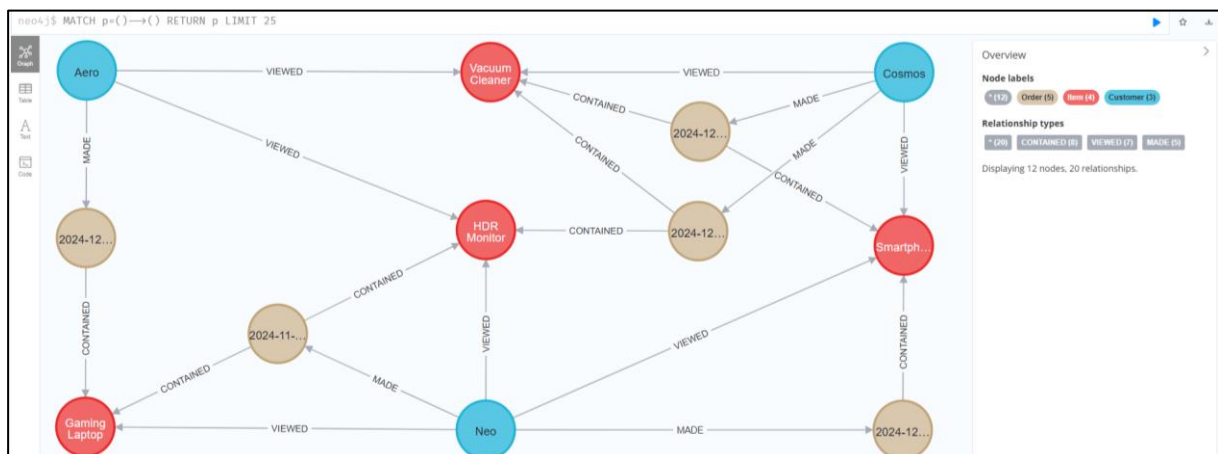
CREATE (c1)-[:MADE]->(o1), (c1)-[:MADE]->(o2), (c2)-[:MADE]->(o3),
      (c3)-[:MADE]->(o4), (c2)-[:MADE]->(o5)

// Кожне замовлення містить товари (здійснення покупки товарів)
MATCH (o1:Order {id: 1}), (o2:Order {id: 2}), (o3:Order {id: 3}),
      (o4:Order {id: 4}), (o5:Order {id: 5}),
      (i1:Item {id: 1}), (i2:Item {id: 2}), (i3:Item {id: 3}), (i4:Item {id: 4})

CREATE (o1)-[:CONTAINED]->(i1), (o1)-[:CONTAINED]->(i2),
      (o2)-[:CONTAINED]->(i3), (o3)-[:CONTAINED]->(i4),
      (o3)-[:CONTAINED]->(i1), (o4)-[:CONTAINED]->(i2),
      (o5)-[:CONTAINED]->(i3), (o5)-[:CONTAINED]->(i4)

// Покупці переглядають товари без здійснення покупки
MATCH (c1:Customer {id: 1}), (c2:Customer {id: 2}), (c3:Customer {id: 3}),
      (i1:Item {id: 1}), (i2:Item {id: 2}), (i3:Item {id: 3}), (i4:Item {id: 4})

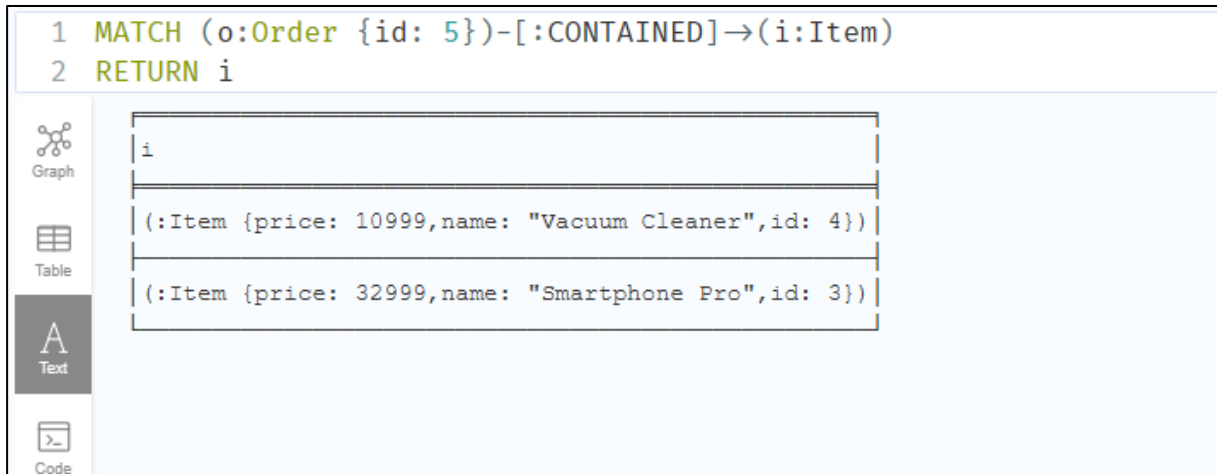
CREATE (c1)-[:VIEWED]->(i1), (c1)-[:VIEWED]->(i2), (c1)-[:VIEWED]->(i3),
      (c2)-[:VIEWED]->(i3), (c2)-[:VIEWED]->(i4),
      (c3)-[:VIEWED]->(i1), (c3)-[:VIEWED]->(i4)
```



1. TYPES of REQUESTS

- Знайти “Товари”, які входять в конкретне “Замовлення”

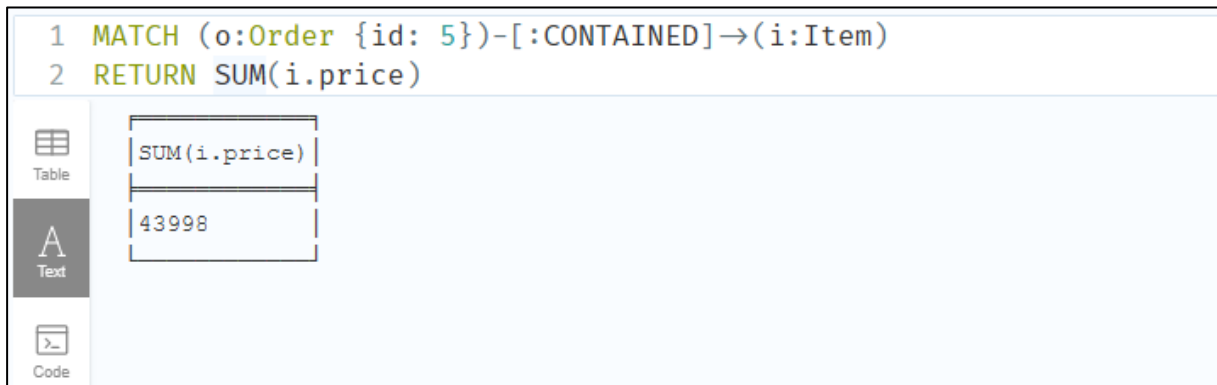
```
1 MATCH (o:Order {id: 5})-[:CONTAINED]→(i:Item)
2 RETURN i
```



i
{:Item {price: 10999,name: "Vacuum Cleaner",id: 4}}
{:Item {price: 32999,name: "Smartphone Pro",id: 3}}

- Підрахувати вартість конкретного “Замовлення”

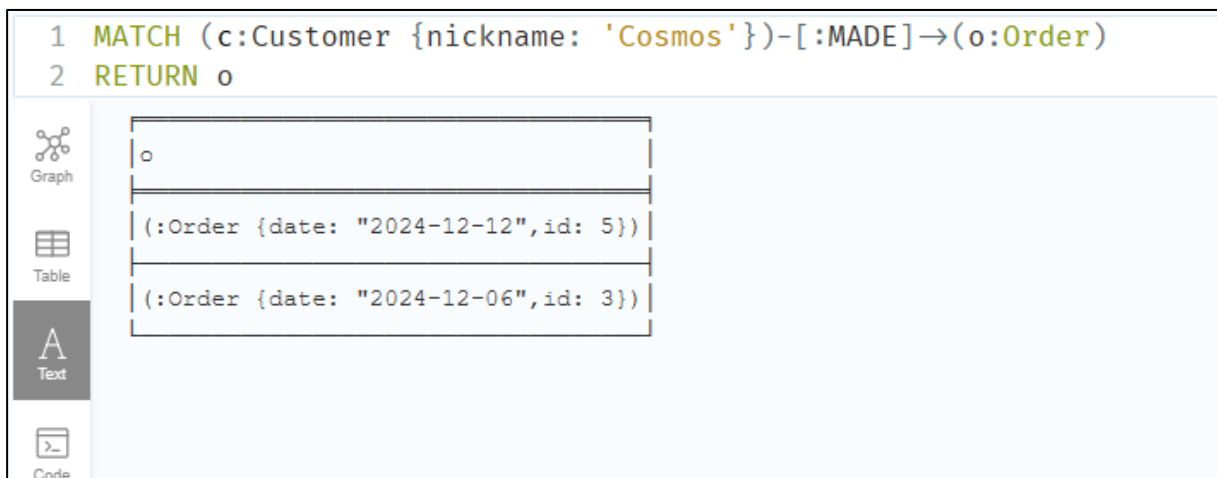
```
1 MATCH (o:Order {id: 5})-[:CONTAINED]→(i:Item)
2 RETURN SUM(i.price)
```



SUM(i.price)
43998

- Знайти всі “Замовлення” конкретного “Покупця”

```
1 MATCH (c:Customer {nickname: 'Cosmos'})-[:MADE]→(o:Order)
2 RETURN o
```



o
{:Order {date: "2024-12-12",id: 5}}
{:Order {date: "2024-12-06",id: 3}}

- Знайти всі “Товари” куплені конкретним “Покупцем” (ч/з Order)

```
1 MATCH (c:Customer {nickname: 'Cosmos'})-[:MADE]-(o:Order)-[:CONTAINED]-(i:Item)
2 RETURN i
```

Graph

Table

Text

Code

i
(:Item {price: 10999,name: "Vacuum Cleaner",id: 4})
(:Item {price: 32999,name: "Smartphone Pro",id: 3})
(:Item {price: 6549,name: "HDR Monitor",id: 1})
(:Item {price: 10999,name: "Vacuum Cleaner",id: 4})

- Знайти к-ть “Товарів”, придбаних конкретним “Покупцем” (ч/з Order)

```
1 MATCH (c:Customer {nickname: 'Cosmos'})-[:MADE]-(o:Order)-[:CONTAINED]-(i:Item)
2 RETURN COUNT(i)
```

Table

Text

Code

COUNT(i)
4

- Знайти для “Покупця”, на яку суму він придбав “Товарів” (ч/з Order)

```
1 MATCH (c:Customer {nickname: 'Cosmos'})-[:MADE]-(o:Order)-[:CONTAINED]-(i:Item)
2 RETURN SUM(i.price)
```

Table

Text

Code

SUM(i.price)
61546

- Знайти скільки разів кожен “Товар” був придбаний і відсортувати

```
1 MATCH (o:Order)-[:CONTAINED]-(i:Item)
2 RETURN i, COUNT(i) as n
3 ORDER BY n DESC
```

Graph

Table

Text

Code

i	n
(:Item {price: 6549,name: "HDR Monitor",id: 1})	2
(:Item {price: 45999,name: "Gaming Laptop",id: 2})	2
(:Item {price: 32999,name: "Smartphone Pro",id: 3})	2
(:Item {price: 10999,name: "Vacuum Cleaner",id: 4})	2

- Знайти всі “Товари”: переглянуті (view) конкретним “Покупцем”

```
1 MATCH (c:Customer {nickname: 'Neo'})-[:VIEWED]->(i:Item)
2 RETURN i
```

Graph

Table

Text

i
(:Item {price: 32999,name: "Smartphone Pro",id: 3})
(:Item {price: 45999,name: "Gaming Laptop",id: 2})
(:Item {price: 6549,name: "HDR Monitor",id: 1})

- Знайти інші “Товари”, що купувались разом з конкретним “Товаром”
(т.б. всі “Товари”, що входять до “Замовлень” разом з цим “Товаром”)

```
1 MATCH (i:Item {id: 1})<-[:CONTAINED]-(o:Order)-[:CONTAINED]->(in_one_cart:Item)
2 RETURN in_one_cart
```

Graph

Table

Text

in_one_cart
(:Item {price: 10999,name: "Vacuum Cleaner",id: 4})
(:Item {price: 45999,name: "Gaming Laptop",id: 2})

- Знайти “Покупців”, які купили даний конкретний “Товар”

```
1 MATCH (i:Item {id: 2})<-[:CONTAINED]-(o:Order)<-[:MADE]-(who_bought:Customer)
2 RETURN who_bought
```

Graph

Table

Text

who_bought
(:Customer {nickname: "Aero",id: 3,email: "pantomblue@webmail.com"})
(:Customer {nickname: "Neo",id: 1,email: "neonrider@electro.com"})

- Знайти для деякого “Покупця” товари, які він переглядав, але не купив

```
1 MATCH (c:Customer {nickname: 'Aero'})-[:VIEWED]->(only_view:Item)
2 WHERE NOT (c)-[:MADE]->(:Order)-[:CONTAINED]->(only_view)
3 RETURN only_view
```

Graph

Table

Text

only_view
(:Item {price: 10999,name: "Vacuum Cleaner",id: 4})
(:Item {price: 6549,name: "HDR Monitor",id: 1})

2. COUNTER ESTIMATION

1) `pip install neo4j`

2) `driver(uri, auth=(user, password))`

```
# Параметри з'єднання з Neo4j
uri = "bolt://localhost:7687"
user = "neo4j"
password = "UIOP1234"
```

3) `initialize_field(driver)`

```
# Ініціалізація поля "likes"
def initialize_field(driver):
    with driver.session() as session:
        session.run("MATCH (i:Item {id: 1}) \
                     SET i.likes = 0")
```

4) `increment_likes(driver)`

```
# Інкремент лічильника на '+1'
def increment_likes(driver):
    with driver.session() as session:
        for _ in range(10000):
            session.run("MATCH (i:Item {id: 1}) \
                         SET i.likes = i.likes + 1")
```

5) `if __name__ == "__main__"`

```
-----
Executing task in Neo4j...
```

```
-----
Task executed in: 2781.56 seconds
```

```
Final counter value (Neo4j): 100000
-----
```

```
neo4j$ MATCH (i:Item {id:1}) RETURN i
```



Graph



Table

```
i
```

```
(:Item {price: 6549,name: "HDR Monitor",id: 1,likes: 100000})
```

