NEO4J

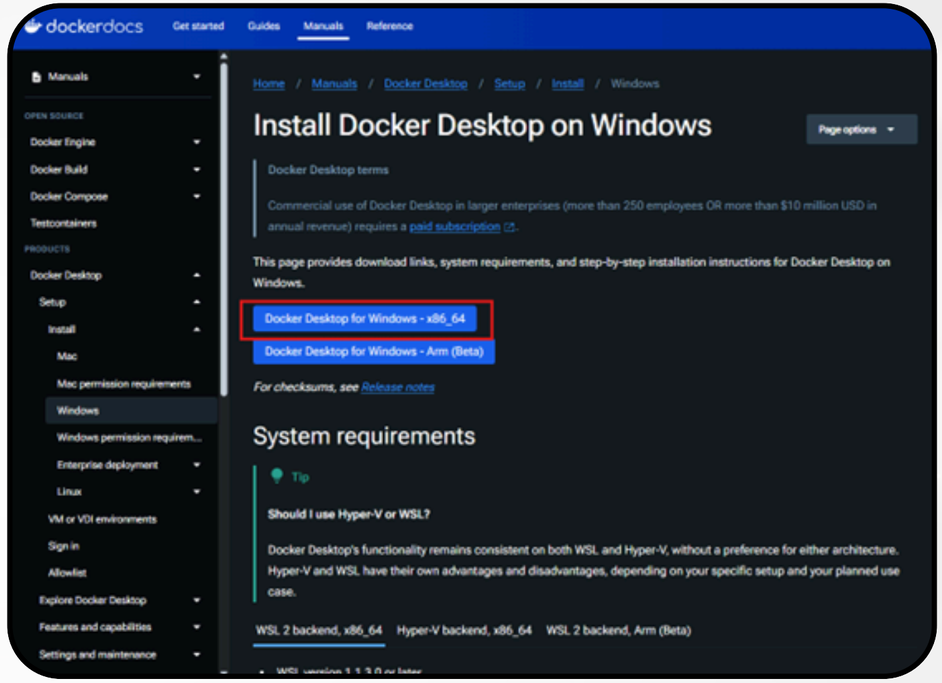
GRAPH DATABASE

นำเสนออาจารย์ : ดร.วราภรณ์ วิยานนท์

ขั้นตอนการติดตั้ง
Neo4j Database

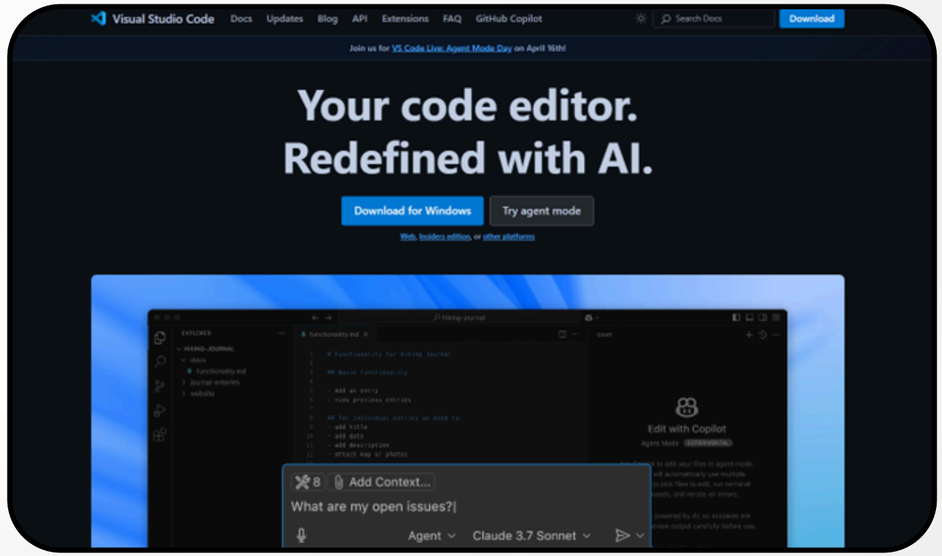
1

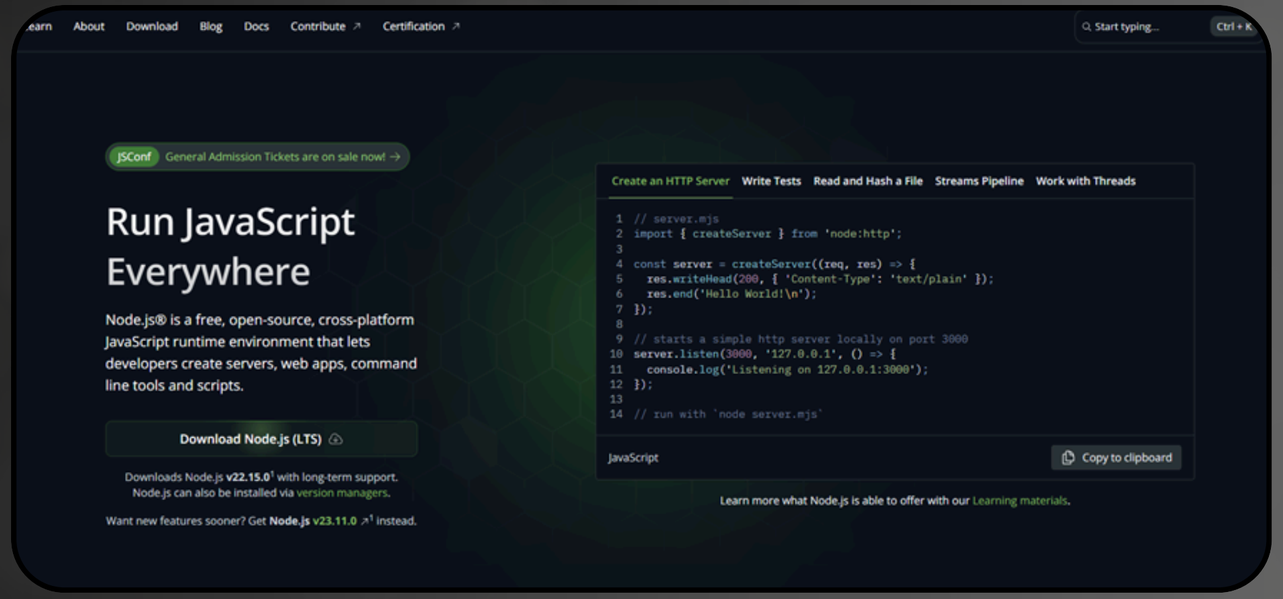
ติดตั้ง Docker desktop
https://www.docker.com/products/docker-desktop/



2

ติดตั้ง Visual Studio Code
https://code.visualstudio.com/






```
services:
  neo4j:
    image: neo4j:latest
    container_name: neo4j-container
    volumes:
      - ./neo4j_database/neo4j_data:/data
      - ./neo4j_database/neo4j_logs:/logs
      - ./neo4j_database/neo4j_plugins:/plugins
    environment:
      - NEO4J_AUTH=neo4j/Neo4j12345*
    ports:
      - "7474:7474"
      - "7687:7687"
    restart: always
```

ขั้นตอนการติดตั้ง Neo4j Database

3

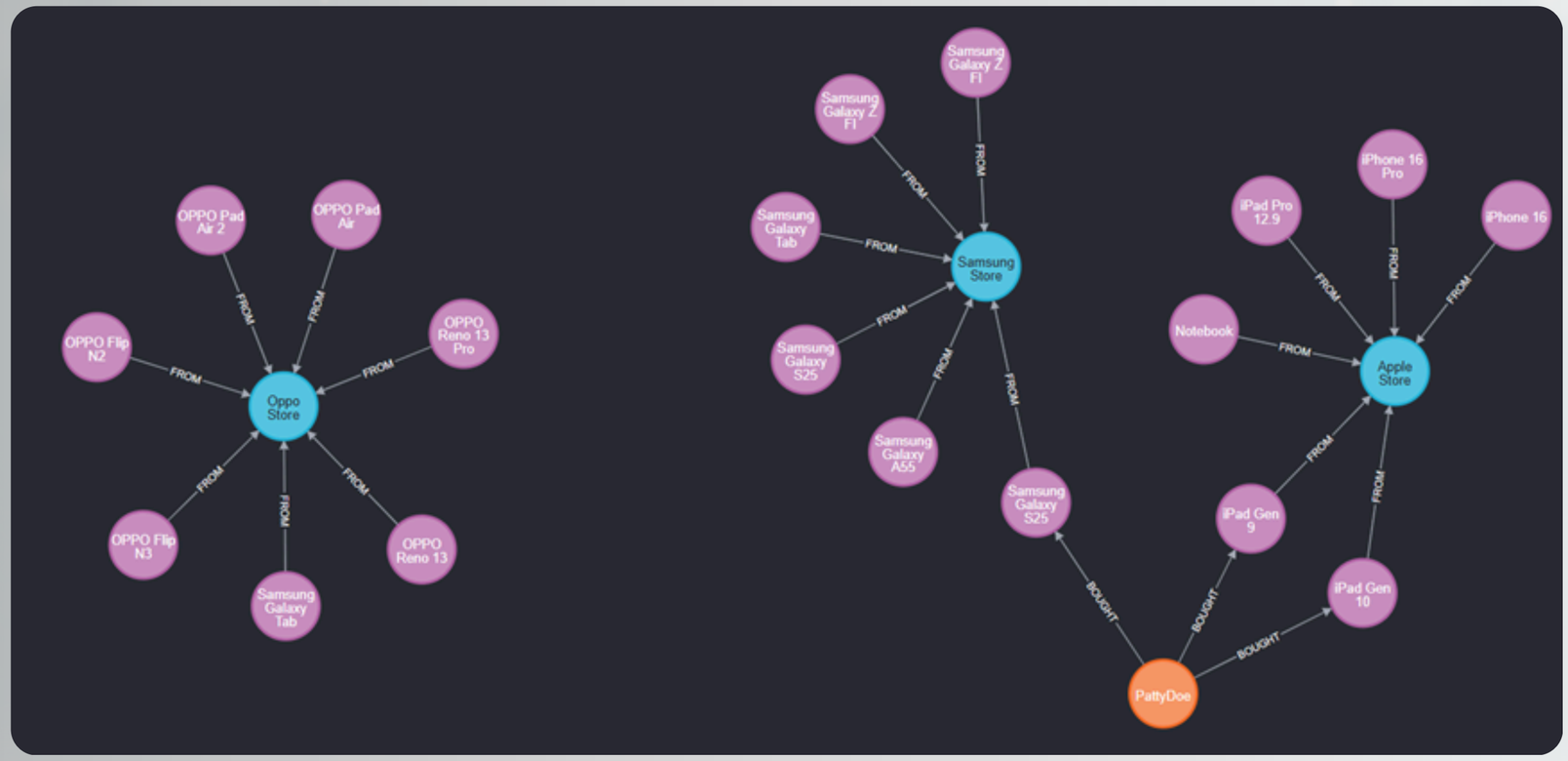
ติดตั้ง Node.js
https://nodejs.org/en

4

ติดตั้งระบบฐานข้อมูลลงบนเครื่อง โดยใช้ Docker Desktop และ Visual Studio Code และกำหนดค่าเริ่มต้นใน docker-compose.yml

CRUD OPERATION

ลักษณะของ Database



แบ่งเป็น **CRUD** สำหรับ **NODE** และ **RELATIONSHIP (EDGE)**

CRUD OPERATION

การทำเบื้องต้นด้วย **Neo4j Driver** จาก **Node.js** และ **Postman**

สร้างไฟล์ชื่อ server.js และเขียนโค้ดด้านล่างนี้

```
const express = require('express');
const bodyParser = require('body-parser');
const neo4j = require('neo4j-driver');

const app = express();
app.use(bodyParser.json());

const driver = neo4j.driver(
    'bolt://localhost:your_port',
    neo4j.auth.basic('your_username', 'your_password')
);
const session = driver.session();

app.get('/', async (req, res) => {
  try {
    await session.run('RETURN 1');
    res.send('Connected to Neo4j successfully!');
  } catch (error) {
    res.status(500).send('Failed to connect to Neo4j: ' +
  error.message);
  }
});

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server is running on
  http://localhost:${PORT}`);
});
```

Package ที่จำเป็น

```
npm init
npm install express neo4j-driver nodemon body-parser
```

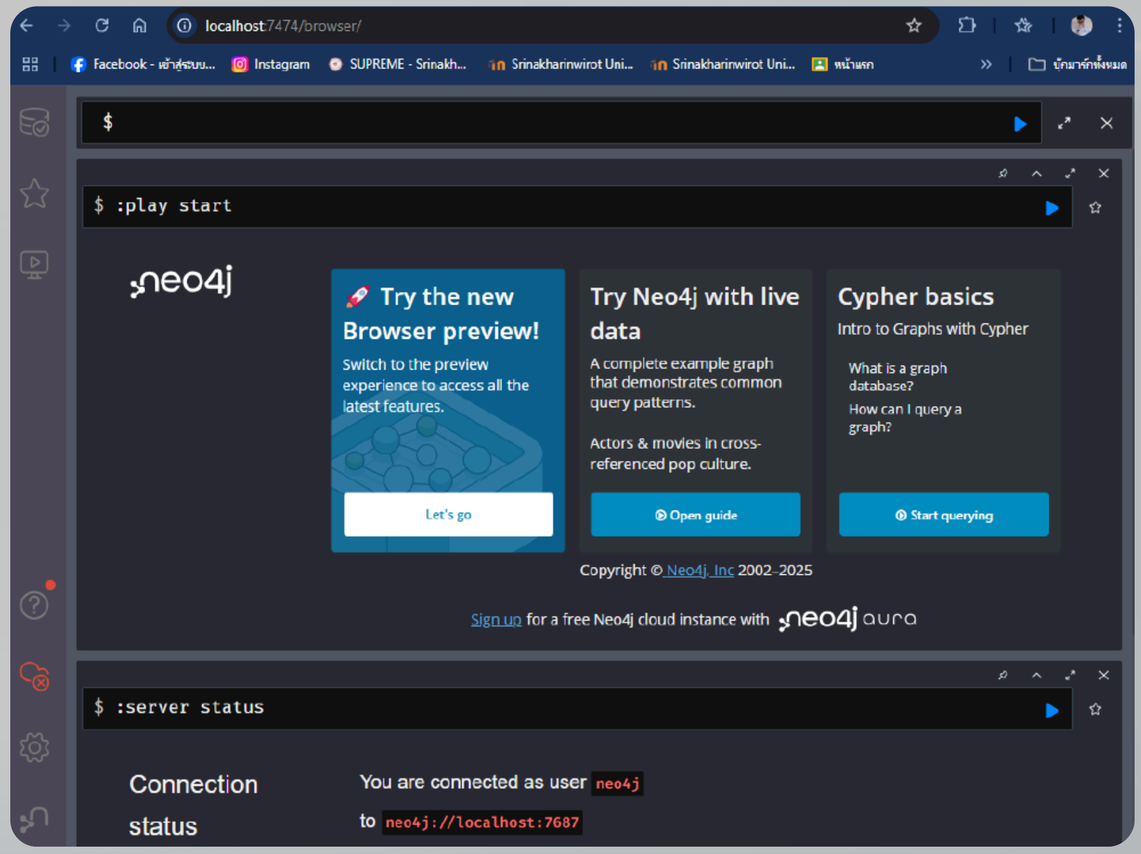
CRUD OPERATION

การทำเบื้องต้นด้วย **Neo4j Driver** จาก **Node.js** และ **Postman**

```
nodemon server.js
```

localhost:3000

```
docker-compose up -d
```



localhost:7474/browser/

CRUD OPERATION

การทำเบื้องต้นด้วย **Neo4j Driver** จาก **Node.js** และ **Postman** จัดการ **NODE** เท่านั้น

CREATE

```
app.post('/nodes', async (req, res) => {
  const { label, properties } = req.body;
  try {
    const result = await session.run(
      `CREATE (n:${label} $props) RETURN n`,
      { props: properties }
    );
    res.json(result.records[0].get('n').properties);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});
```

UPDATE

```
app.put('/nodes/:label/:id', async (req, res) => {
  const { label, id } = req.params;
  const { properties } = req.body;
  try {
    const result = await session.run(
      `MATCH (n:${label} {id: $id})
       SET n += $props
       RETURN n`,
      { id, props: properties }
    );
    res.json(result.records[0].get('n').properties);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});
```

READ

```
app.get('/nodes/:label', async (req, res) => {
  const label = req.params.label;
  try {
    const result = await session.run(`MATCH
  (n:${label}) RETURN n`);
    const nodes = result.records.map(record =>
  record.get('n').properties);
    res.json(nodes);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});
```

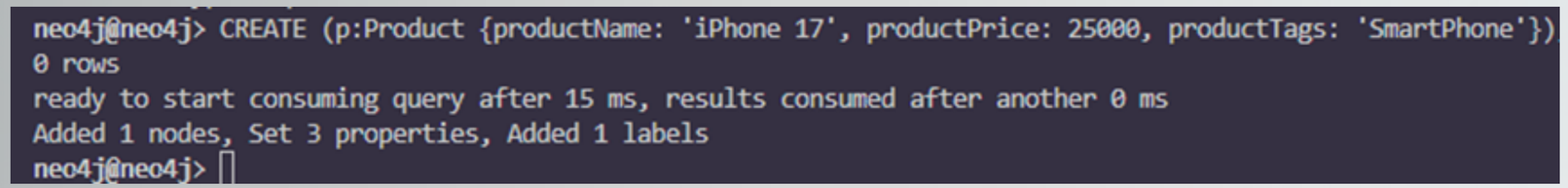
DELETE

```
app.delete('/nodes/:label/:id', async (req, res) => {
  const { label, id } = req.params;
  try {
    await session.run(
      `MATCH (n:${label} {id: $id}) DETACH DELETE n`,
      { id }
    );
    res.json({ message: 'Node deleted' });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});
```

CRUD OPERATION

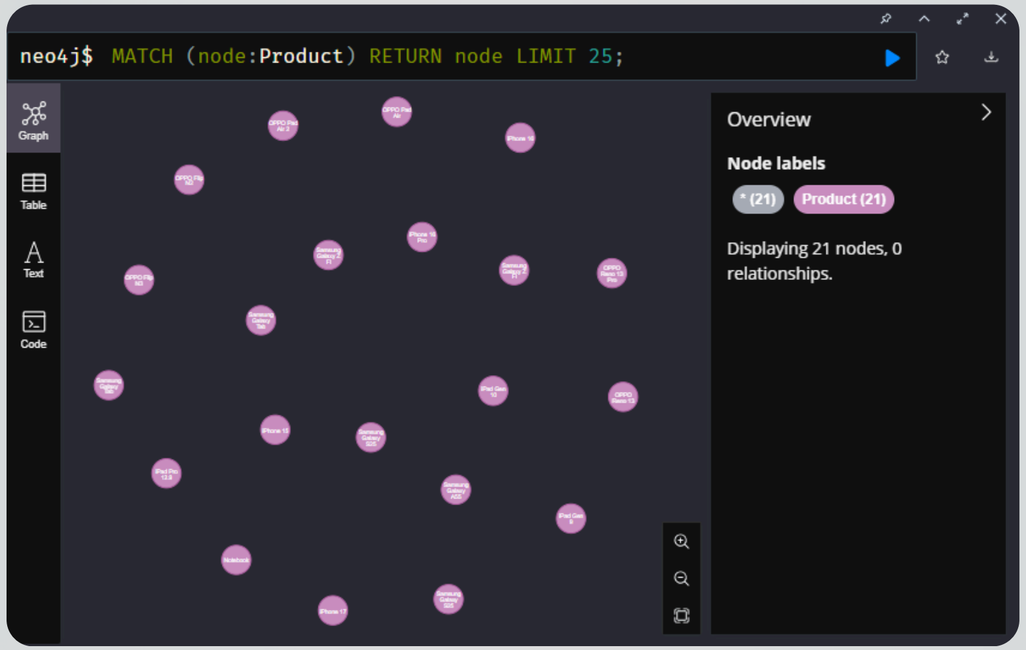
CREATE NODE ผ่าน Neo4jBrowser

CREATE NODE ผ่าน Neo4j Cypher-Shell บน Docker



CRUD OPERATION

READ NODE ผ่าน Neo4jBrowser



READ NODE ผ่าน Neo4j Cypher-Shell บน Docker

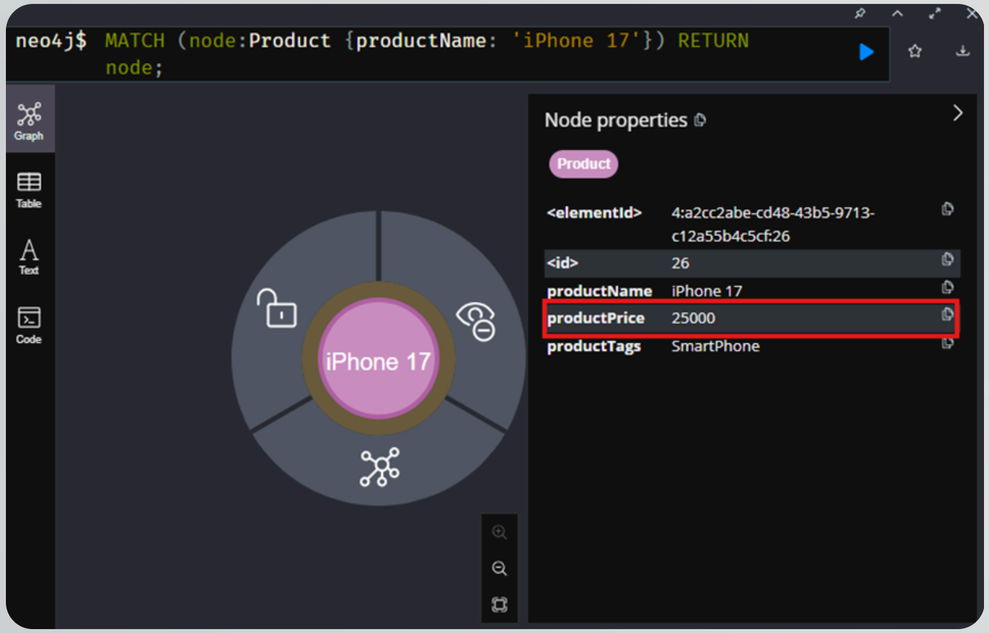
```
neo4j@neo4j> MATCH (node:Shop) RETURN node LIMIT 25;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------+
| node                                                                                                                                                      |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------+
| (:Shop {shopName: "Samsung Store", shopPhone: "0998881248", id: "1745643181263", shopAddress: "114 Sukhumvit 23, Bangkok 10110, Thailand."})              |
| (:Shop {shopName: "Apple Store", shopPhone: "0998887777", id: "1745643212838", shopAddress: "789 Green Lane, Eco Town"})                                  |
| (:Shop {shopName: "Oppo Store", shopPhone: "0998887755", id: "1745643228818", shopAddress: "123 Main Street, Cityville"})                                 |
| (:Shop {shopName: "SWUShop", shopPhone: "0999999999", id: "1745648165481", shopAddress: "114 Sukhumvit 23, Bangkok 10110, Thailand."})                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------+

4 rows
ready to start consuming query after 43 ms, results consumed after another 1 ms
neo4j@neo4j>
```

CRUD OPERATION

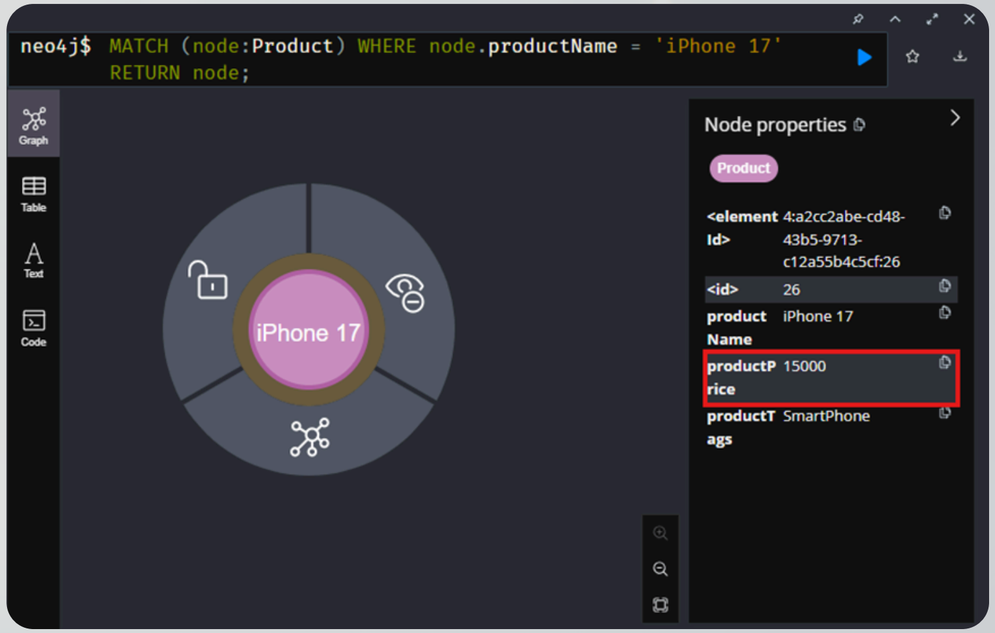
UPDATE NODE ผ่าน **Neo4jBrowser**

```
1 MATCH (node:Product {productName: 'iPhone 17'})
2 SET node.productPrice = 15000;
```

Set 1 property, completed after 62 ms.



BEFORE

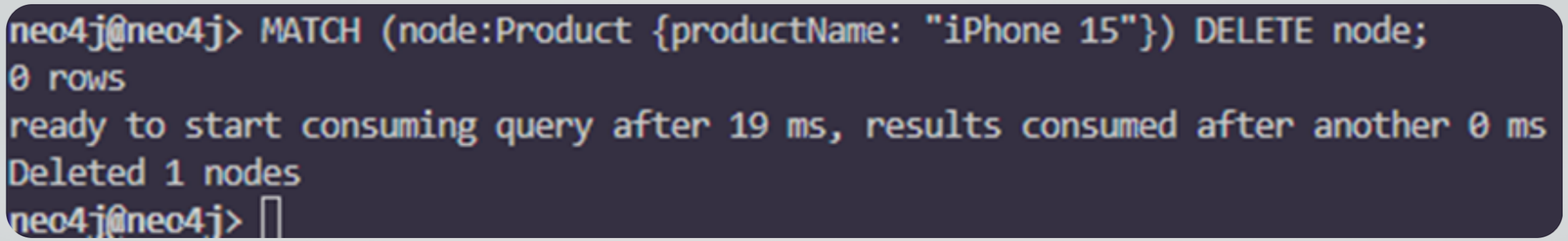


AFTER

CRUD OPERATION

DELETE NODE ผ่าน Neo4jBrowser

DELETE NODE ผ่าน Neo4j Cypher-Shell บน Docker



CRUD OPERATION

CREATE RELATIONSHIP ผ่าน **Neo4jBrowser**

สร้าง **RELATIONSHIP** จาก **NODE** แรกไปยัง **NODE** ที่สอง

```
1 MATCH (p1:Product {productName: 'iPad Gen 9'})
2 MATCH (p2:Product {productName: 'iPad Gen 10'})
3 MERGE (p1)-[:NEWEST]→(p2);
```

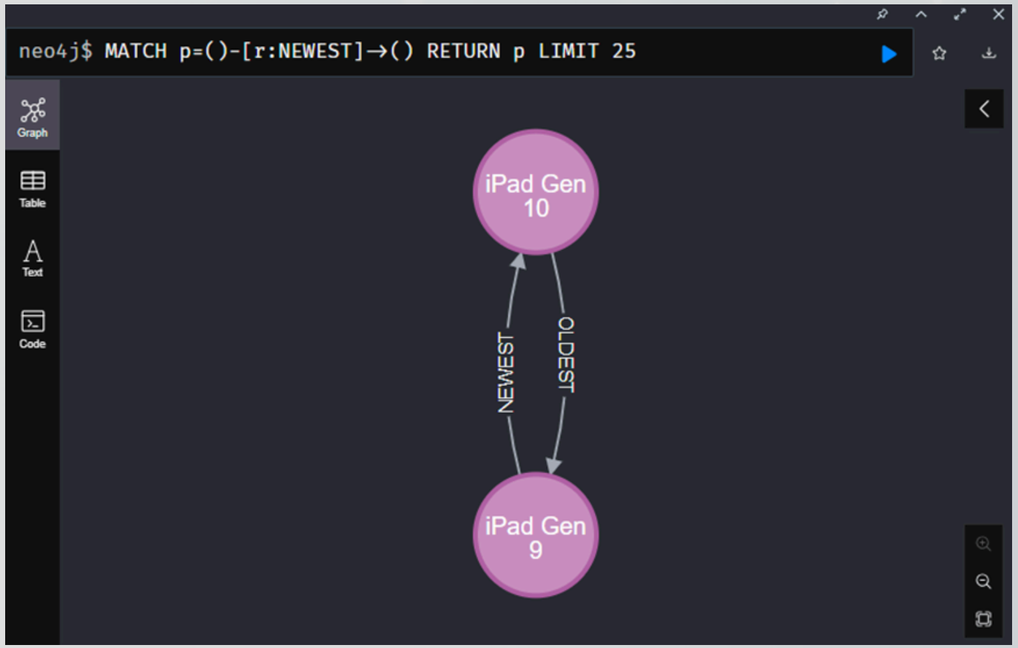
Created 1 relationship, completed after 19 ms.

สร้าง **RELATIONSHIP** จาก **NODE** ที่สองกลับมายัง **NODE** แรก

```
1 MATCH (p1:Product {productName: 'iPad Gen 9'})
2 MATCH (p2:Product {productName: 'iPad Gen 10'})
3 MERGE (p2)-[:OLDEST]→(p1);
```

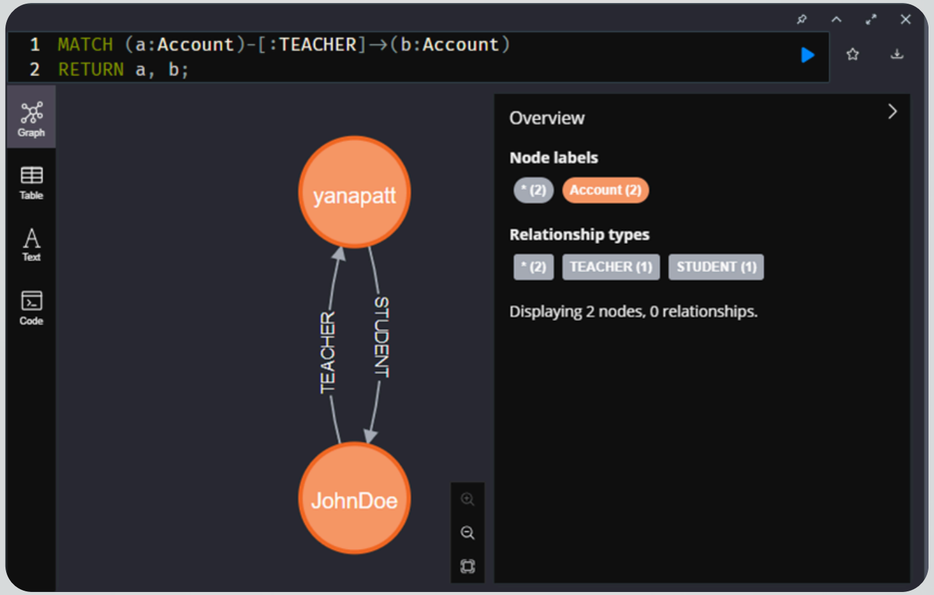
Created 1 relationship, completed after 68 ms.

ผลลัพธ์

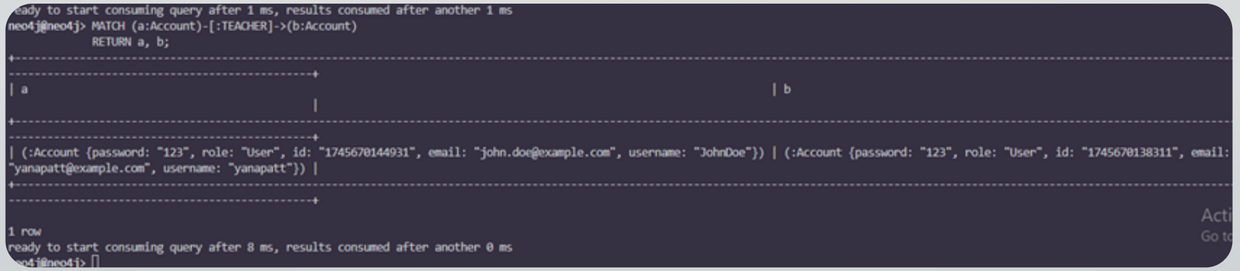


CRUD OPERATION

READ คู่ NODE ที่มี RELATIONSHIP ผ่าน Neo4jBrowser



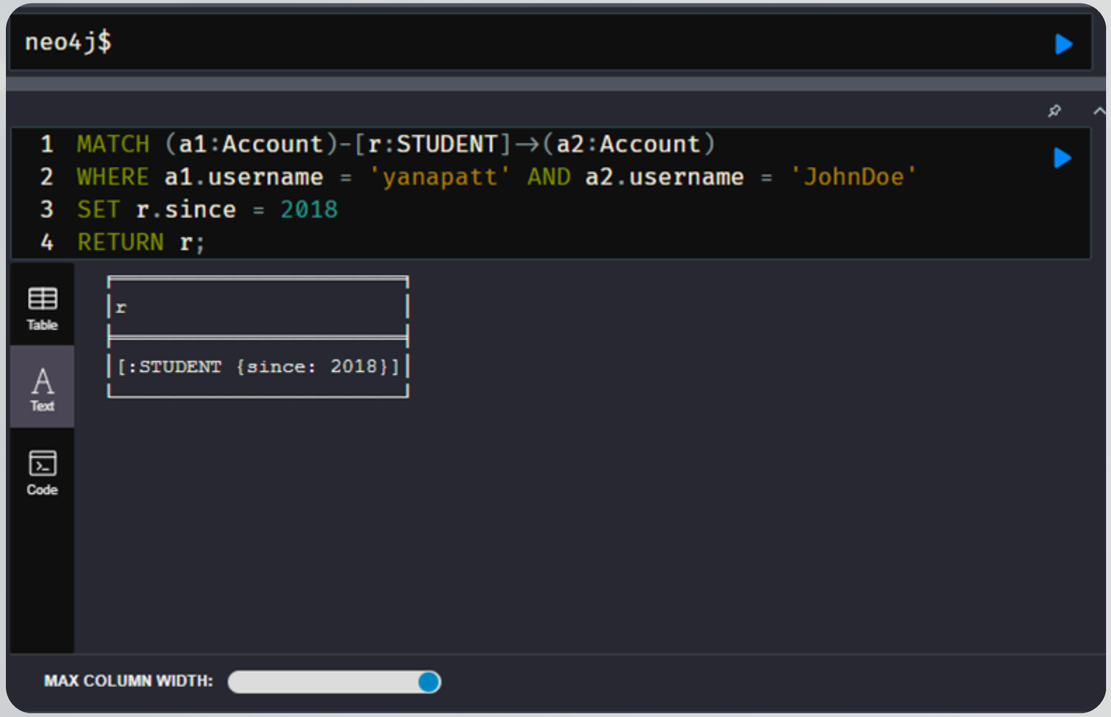
READ คู่ NODE ที่มี RELATIONSHIP ผ่าน Neo4j Cypher-Shell บน Docker



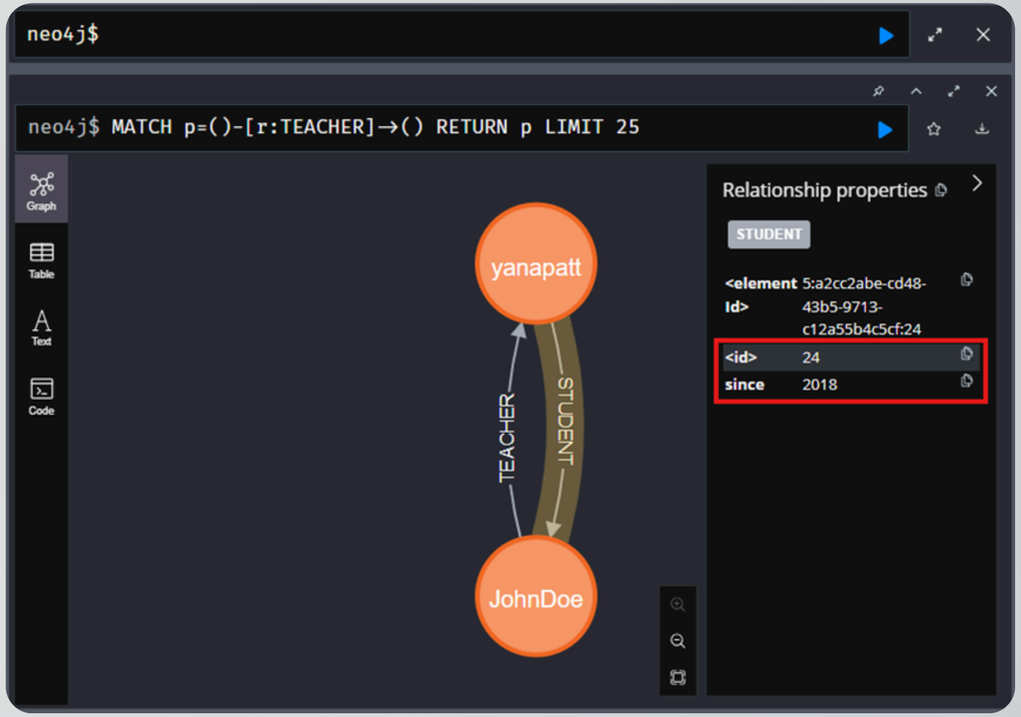
CRUD OPERATION

UPDATE PROPERTY ของ **RELATIONSHIP** ผ่าน **Neo4jBrowser**

PROPERTY ของ **RELATIONSHIP** จาก **NODE** แรกไป
ยัง **NODE** ที่สอง



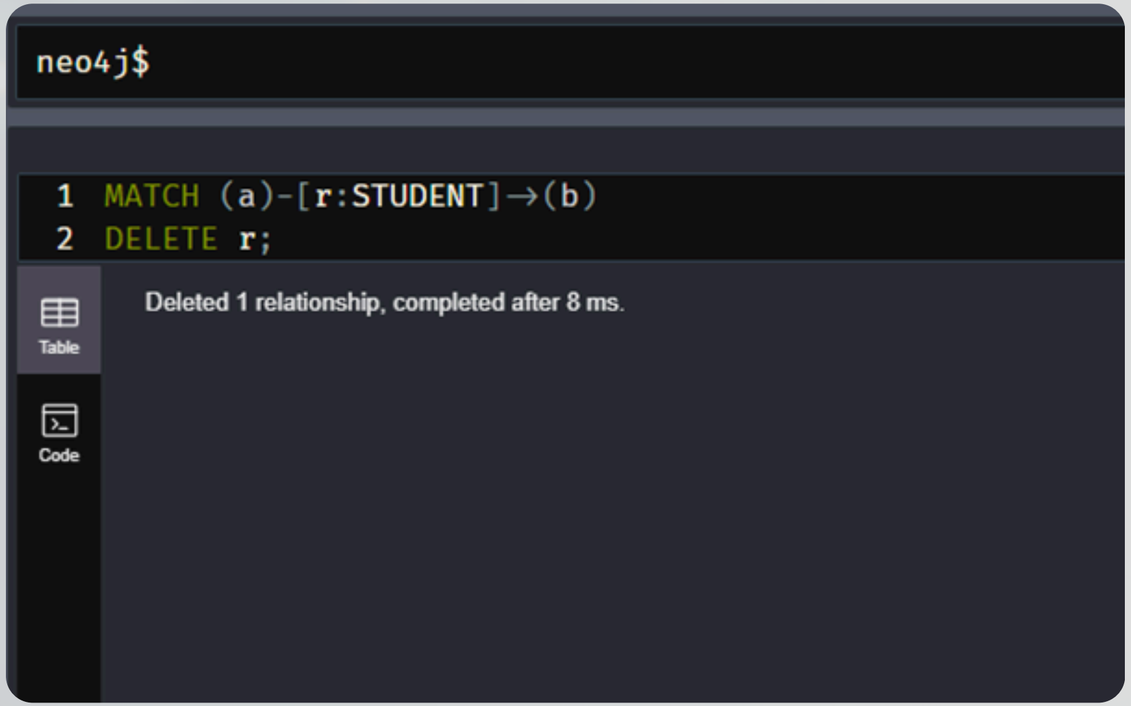
ผลลัพธ์



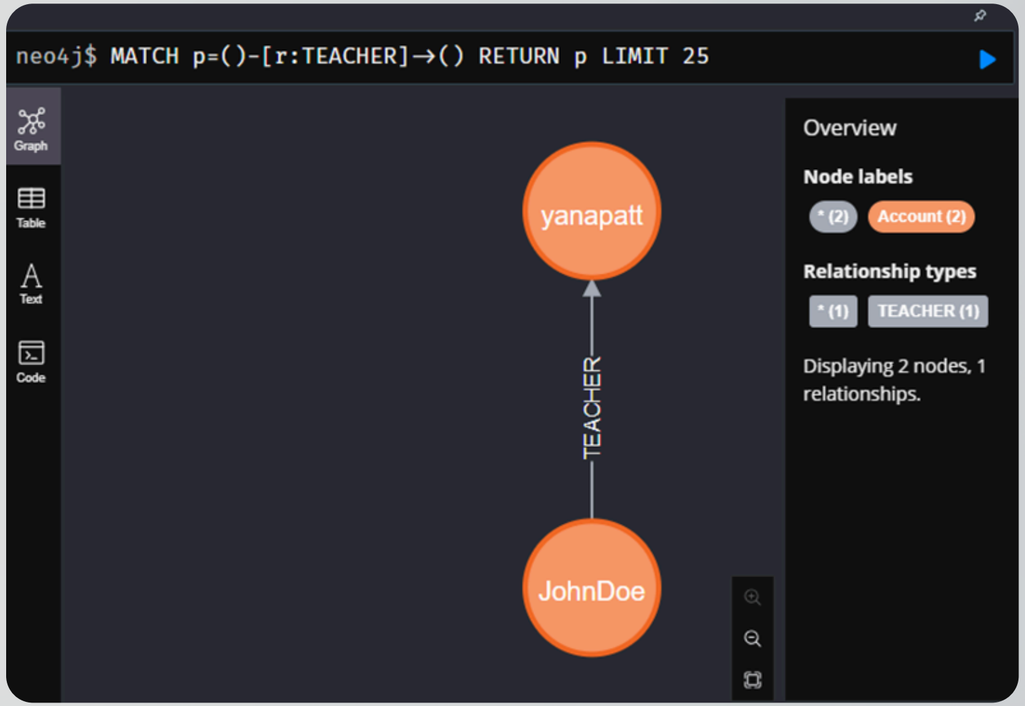
CRUD OPERATION

DELETE RELATIONSHIP ระหว่างคู่ **NODE** ที่กำหนด ผ่าน **Neo4jBrowser**

DELETE RELATION จาก **NODE** แรกไปยัง **NODE** ที่สอง

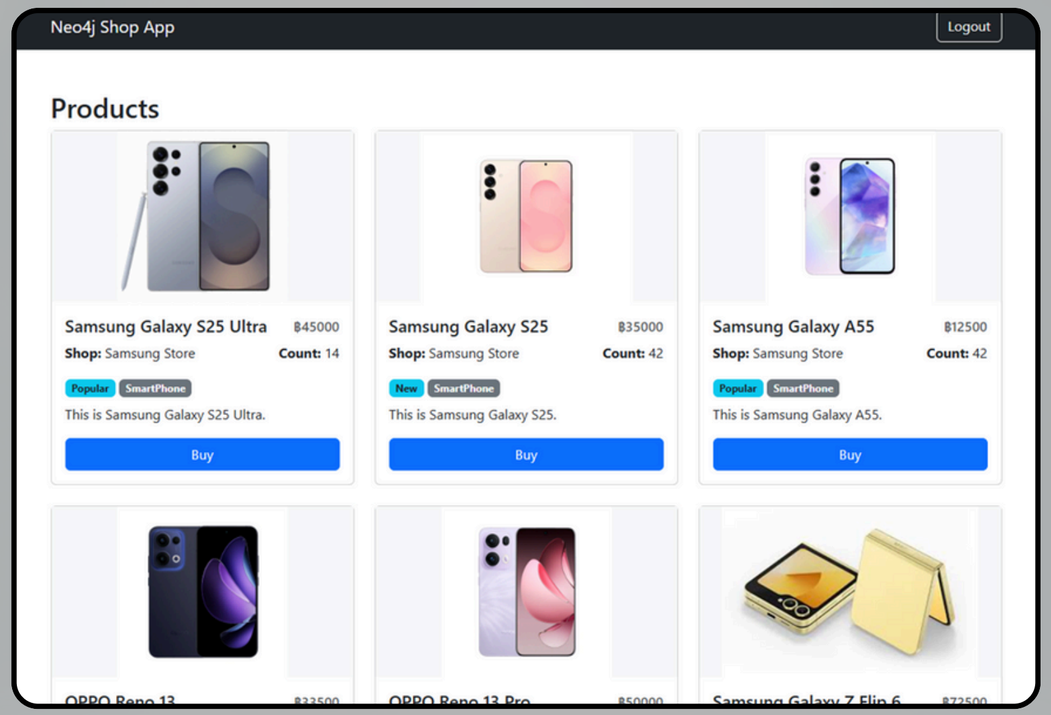


ผลลัพธ์



USE CASE

WEBSITE แนะนำสินค้า



แนะนำสินค้าที่มีหมวดหมู่เดียวกัน ที่ผู้ใช้คนนั้นซื้อเยอะที่สุด

แนะนำสินค้าที่มาจากร้านค้าเดียวกัน ที่ผู้ใช้คนนั้นซื้อเยอะที่สุด

จัดทำโดย

นายจิรัฏฐ์ อู่คงคา	66102010233
นายญาณภัทร ปานเกษม	66102010236
นายรัฐศาสตร์ จันทรา	66102010244

THANK YOU