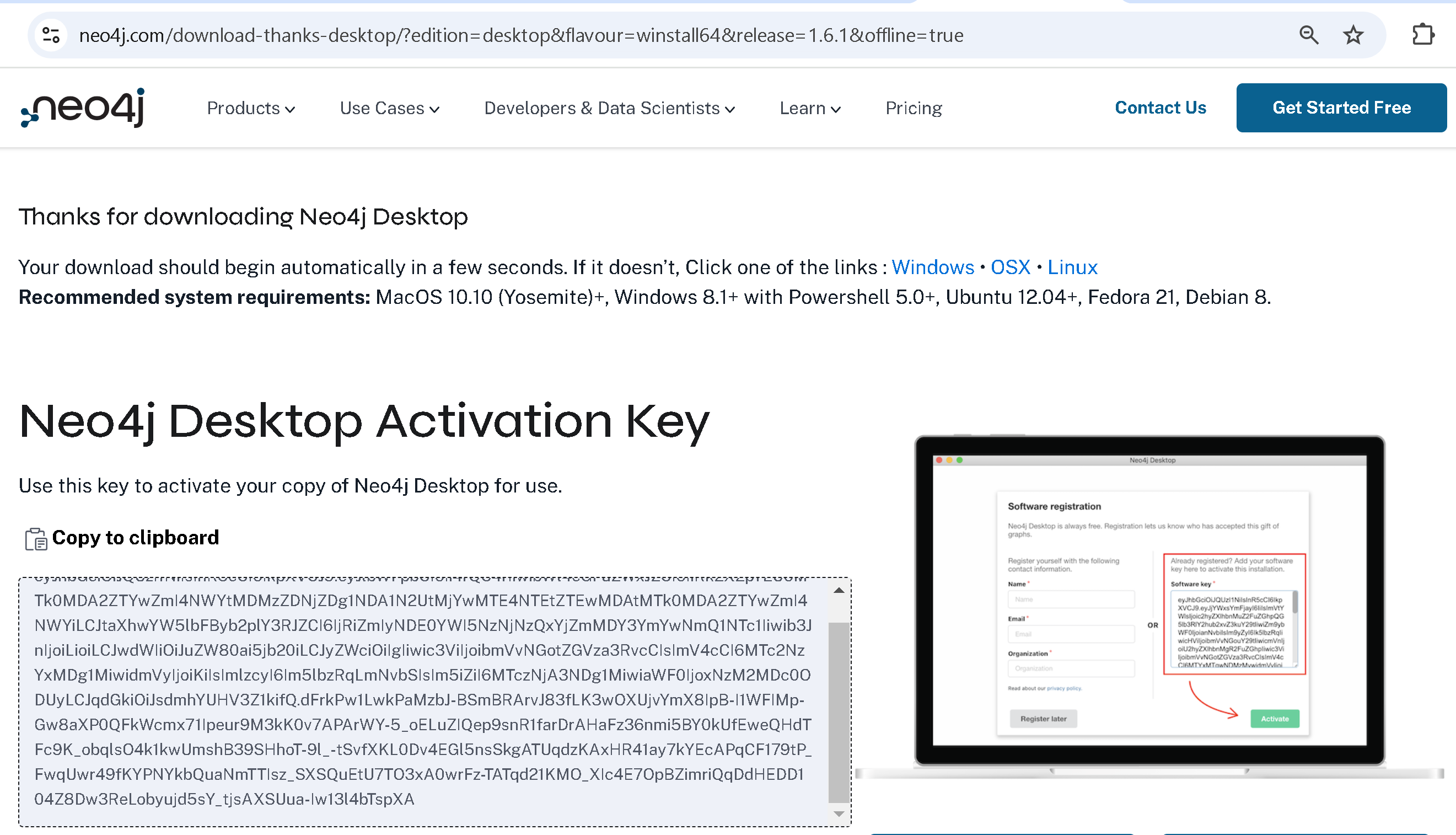
**Họ tên: Nguyễn Văn Minh MSSV: 22003405 Lớp: DHKTPM18A**

## **NEO4J**

## **Cài đặt Neo4j Desktop**

<https://neo4j.com/download/>



## **Create a database**

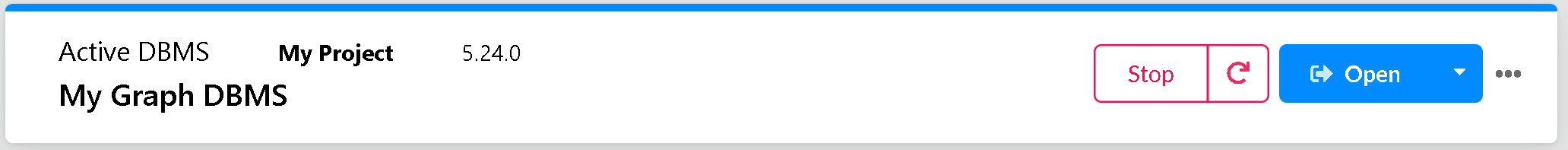
Add 🡪 Local DBMS.

Give the project name, graph DBMS name, and password. Then click the “Create” button.

Click the “Start” button on the graph DBMS.

A screenshot of a computer

Description automatically generated



Click the “Open” button, select the “Neo4j Browser” or enter "http://localhost:7474" on the browser.

A screenshot of a computer

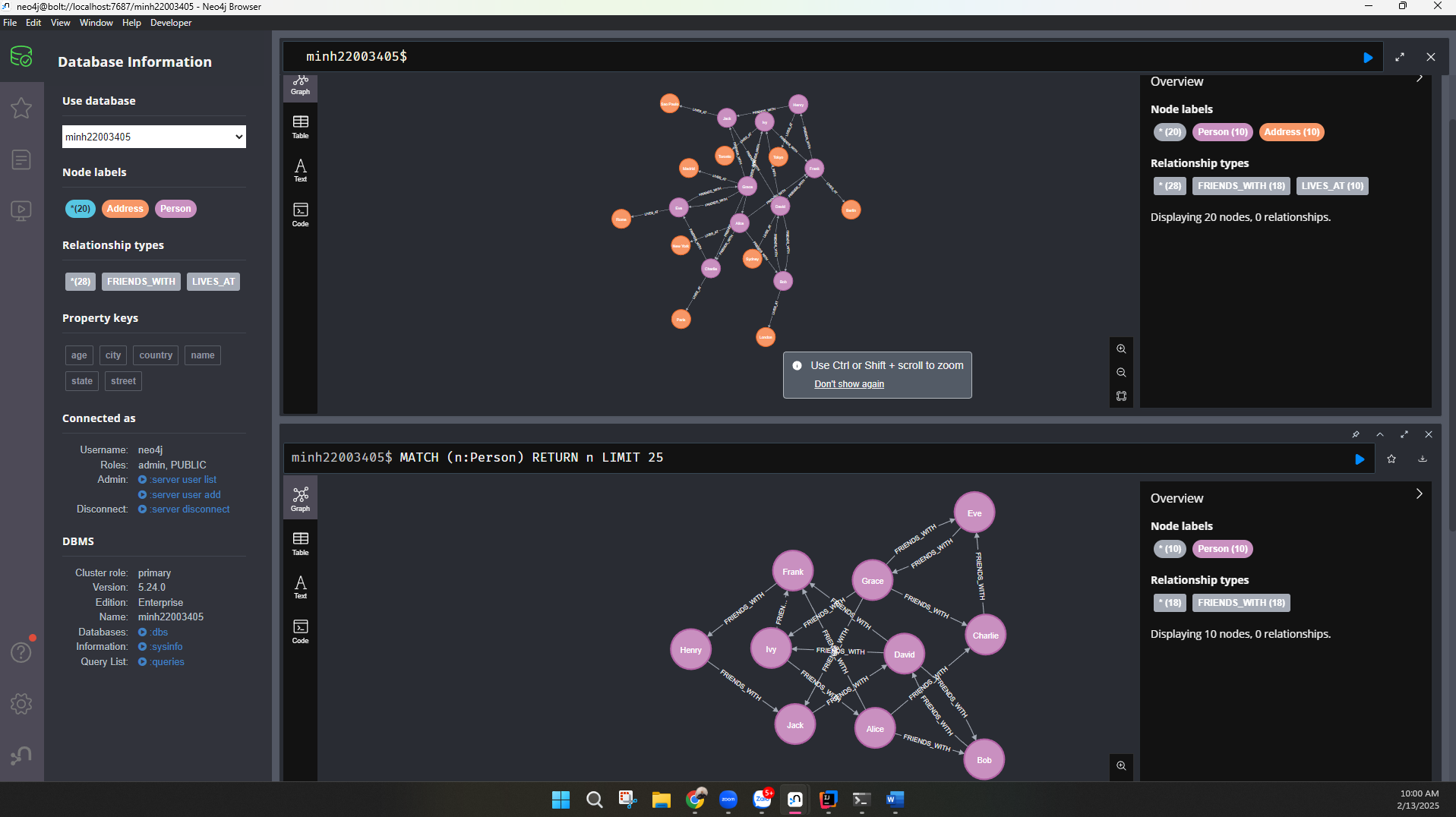
Description automatically generated

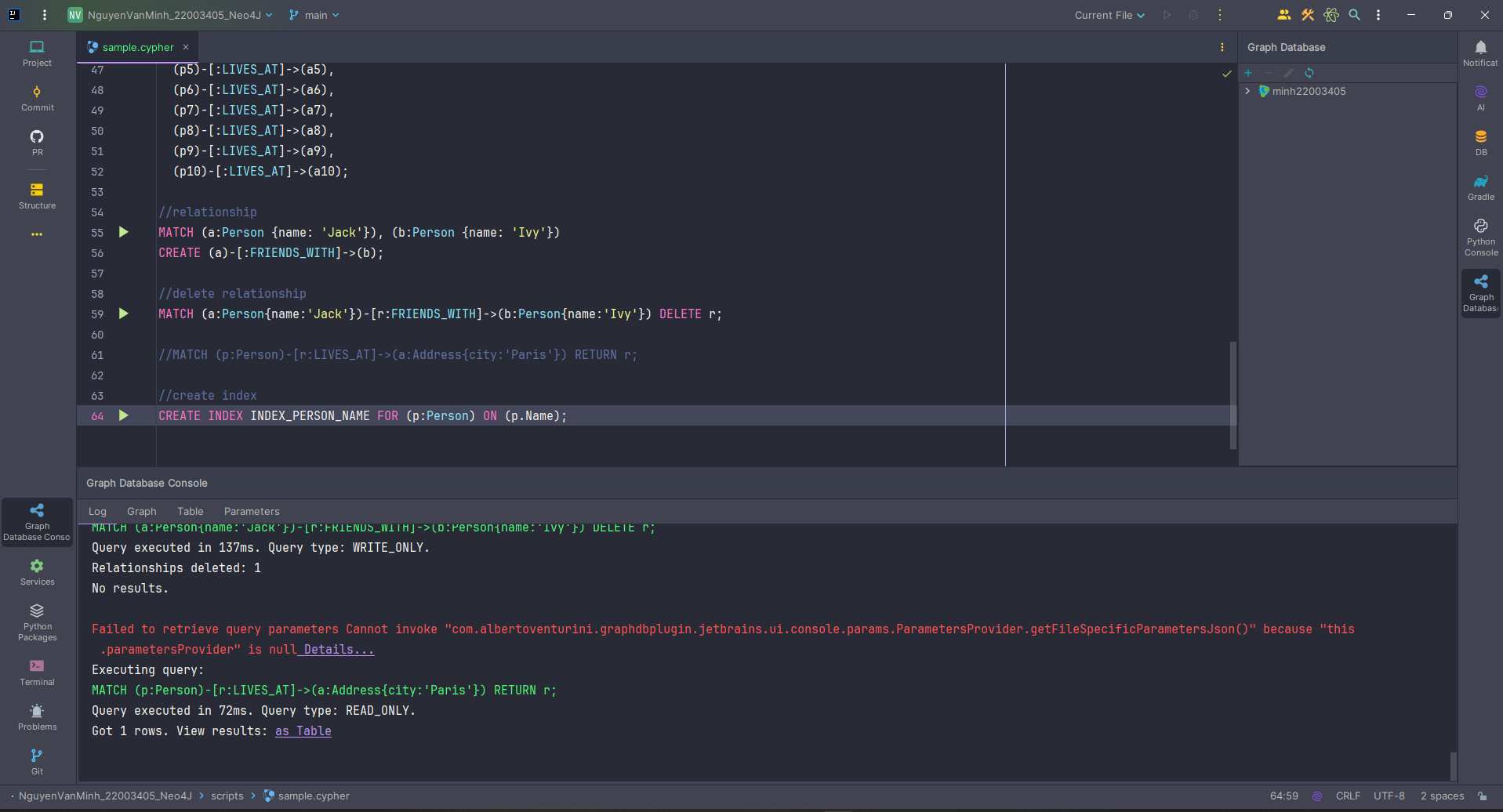
<https://neo4j.com/docs/getting-started/>

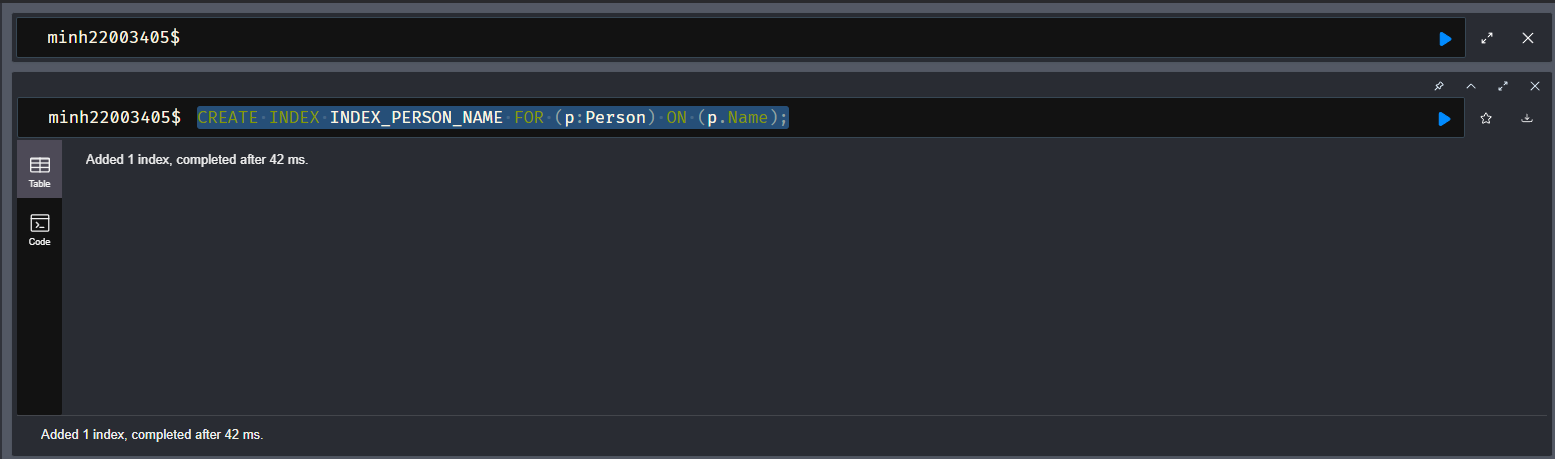
## **Neo4j Overview**

A diagram of a cloud computing system

Description automatically generated





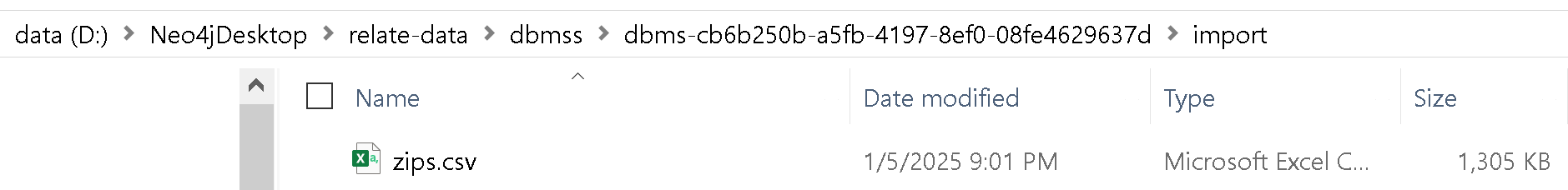


## **Load data from CSV file**

<https://neo4j.com/docs/cypher-manual/current/clauses/load-csv/>

1. **Copy zips.csv vào folder import**

D:\Neo4jDesktop\relate-data\dbmss\dbms-cb6b250b-a5fb-4197-8ef0-08fe4629637d\import

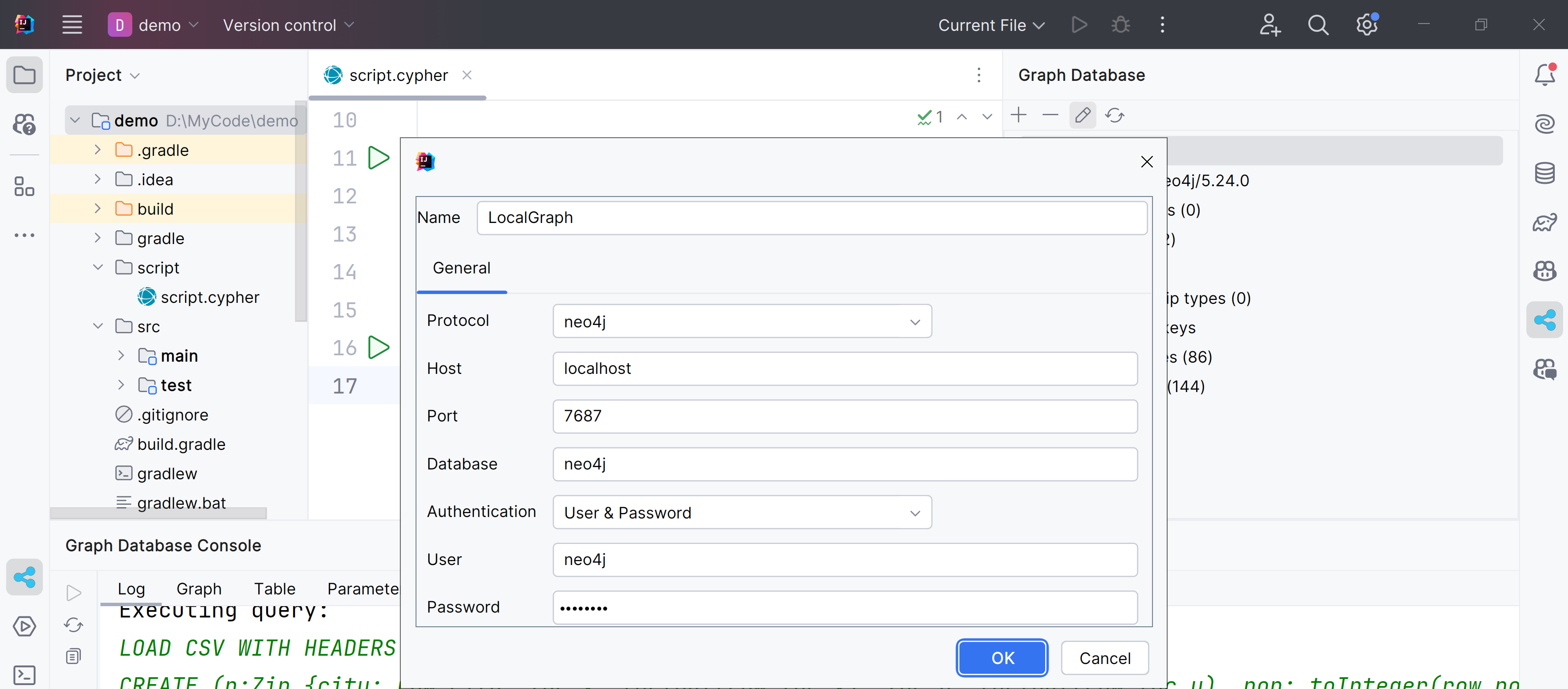


1. **In IntelliJ IDEA, install the “Graph Database” plugin**

A screenshot of a computer

Description automatically generated

1. **Create a new data source**



1. **Create a .cypher file, write and run the following scripts**

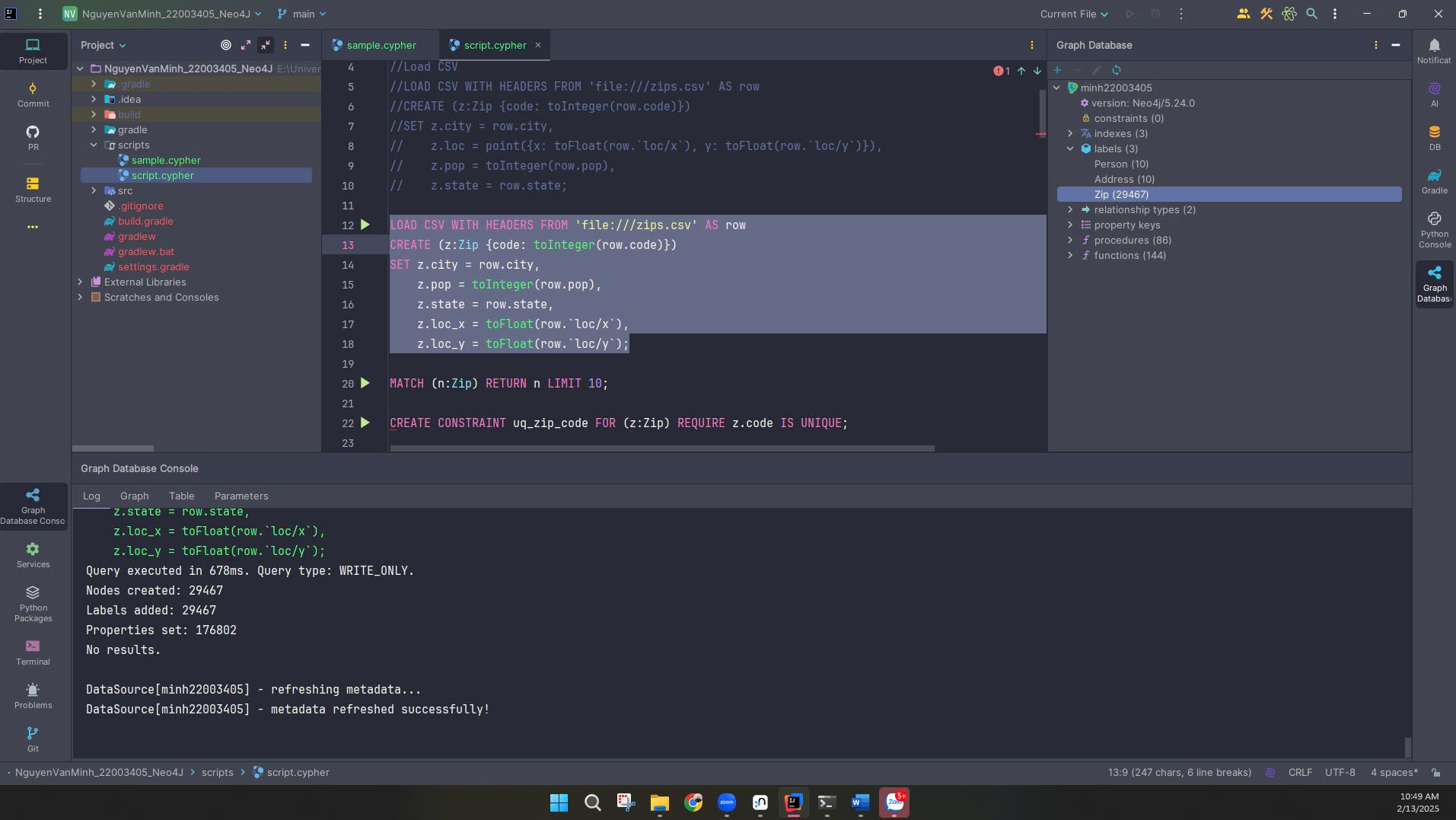
LOAD CSV WITH HEADERS FROM 'file:///zips.csv' AS row  
CREATE (z:Zip {code: toInteger(row.code)})  
SET

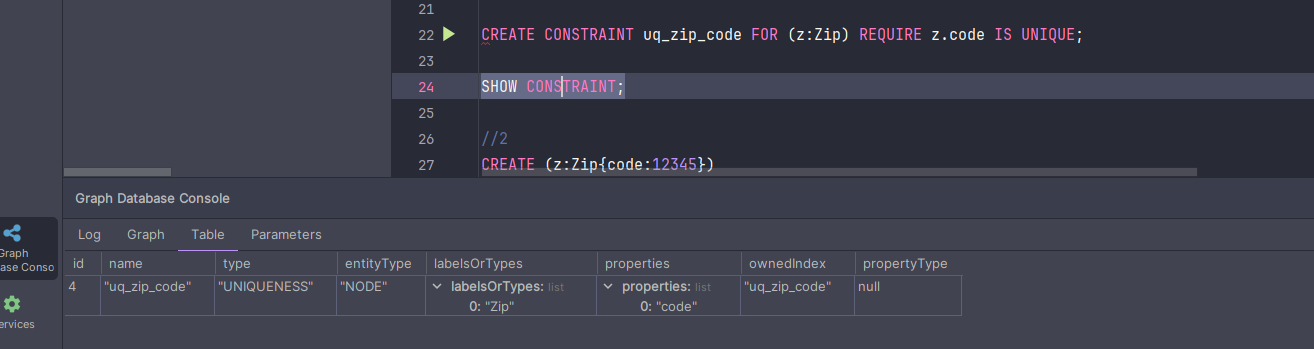
z.city = row.city,  
 z.pop = toInteger(row.pop),  
 z.state = row.state,  
 z.loc\_x = toFloat(row.`loc/x`),  
 z.loc\_y = toFloat(row.`loc/y`);

**Create unique index on :Zip(zip\_id)**

CREATE CONSTRAINT uq\_zip\_code FOR (z:Zip) REQUIRE z.code IS UNIQUE;

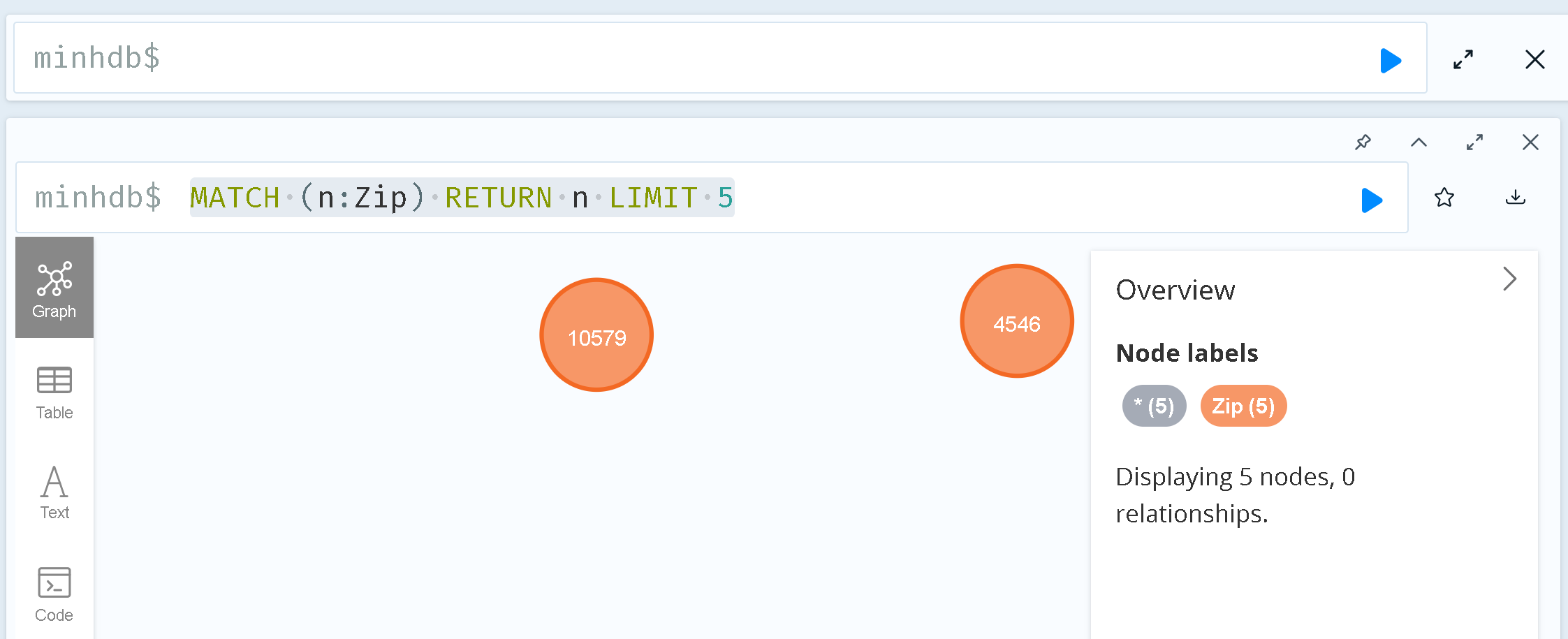
1. **The result**

****

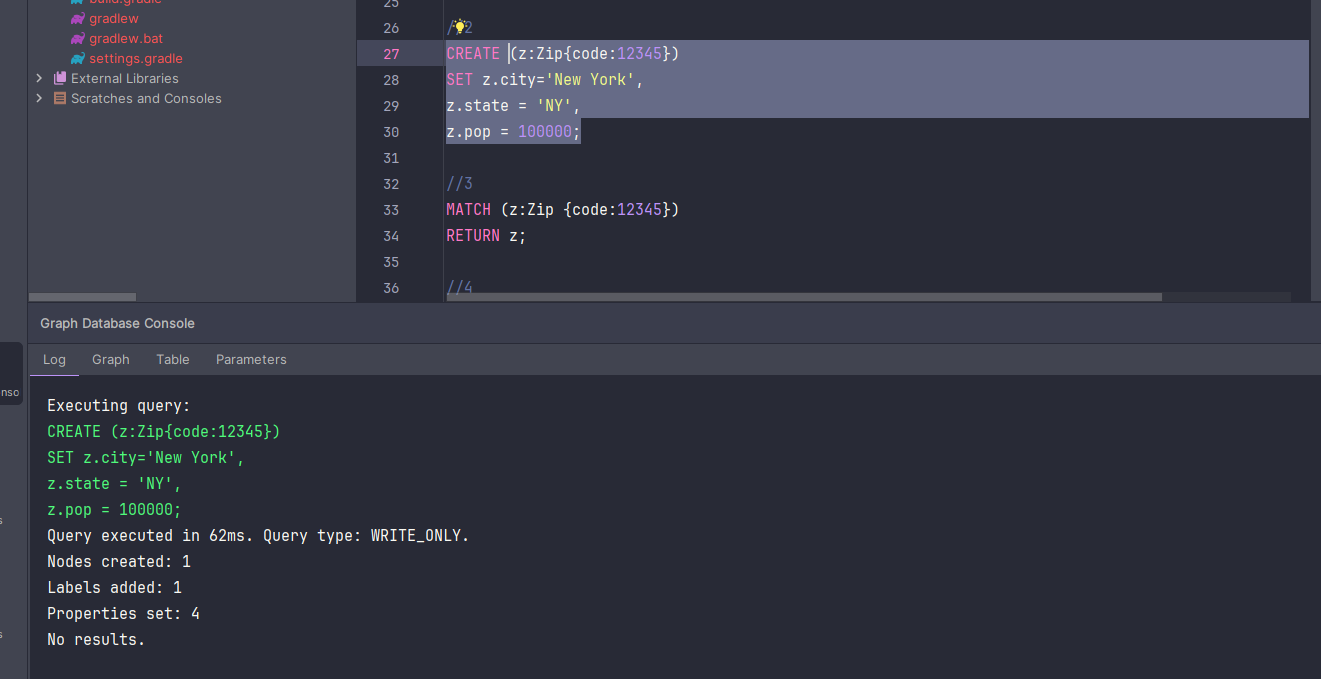
****

1. **Query**
2. List n nodes

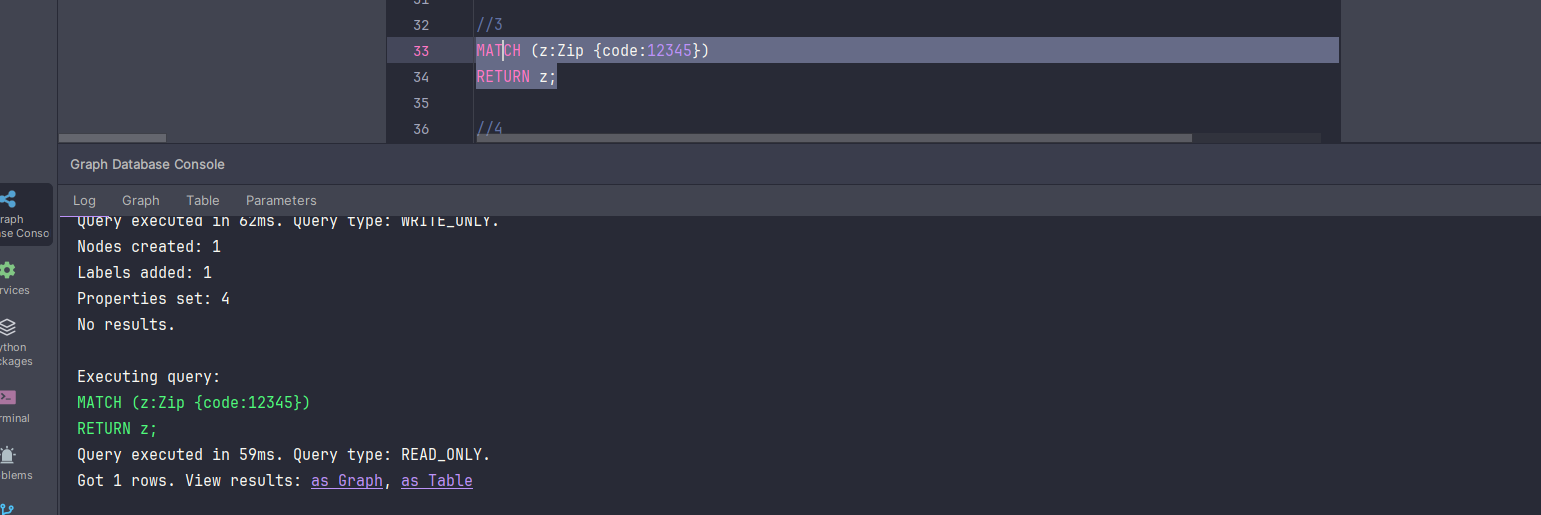
MATCH (n:Zip) RETURN n LIMIT 5



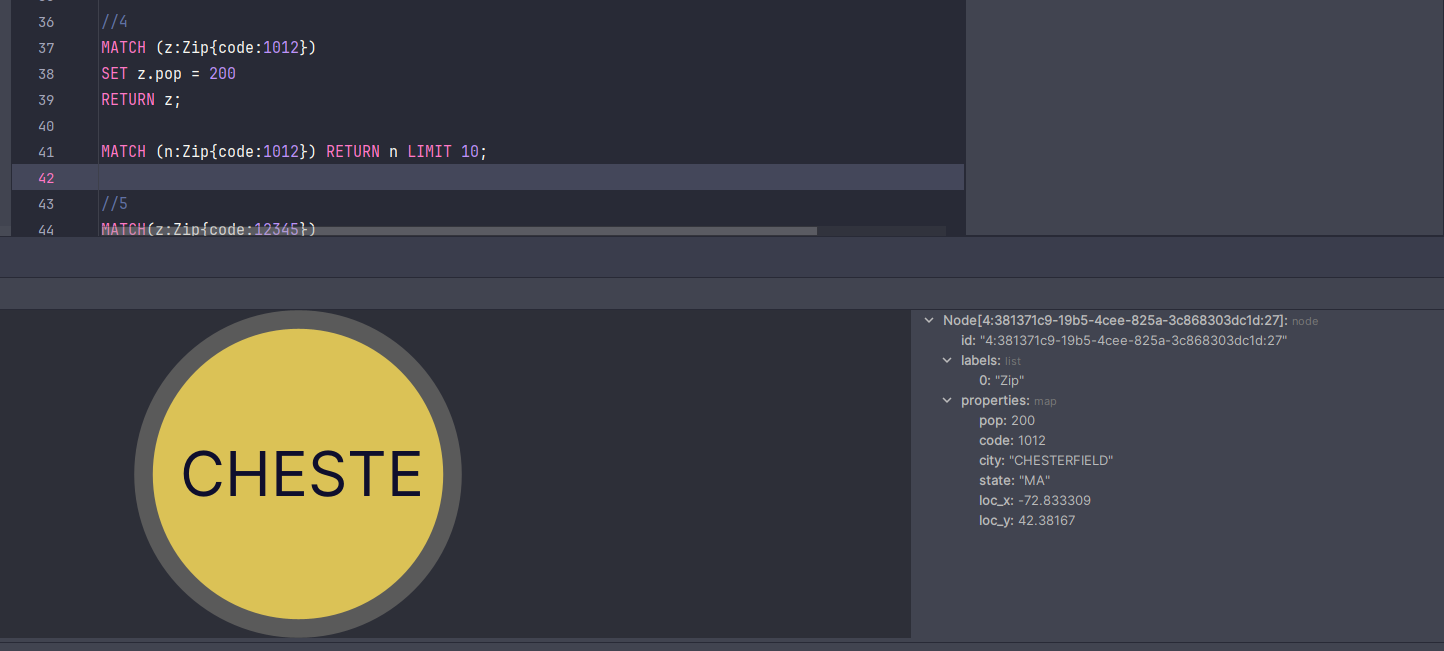
1. Create a new node



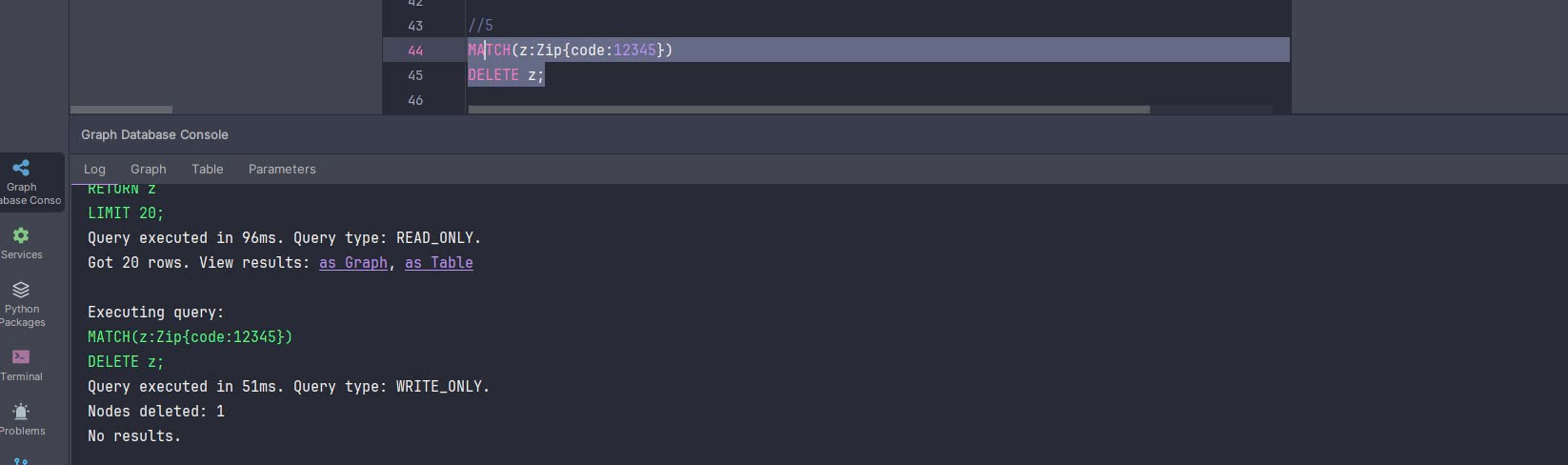
1. Tìm một node khi biết zip code



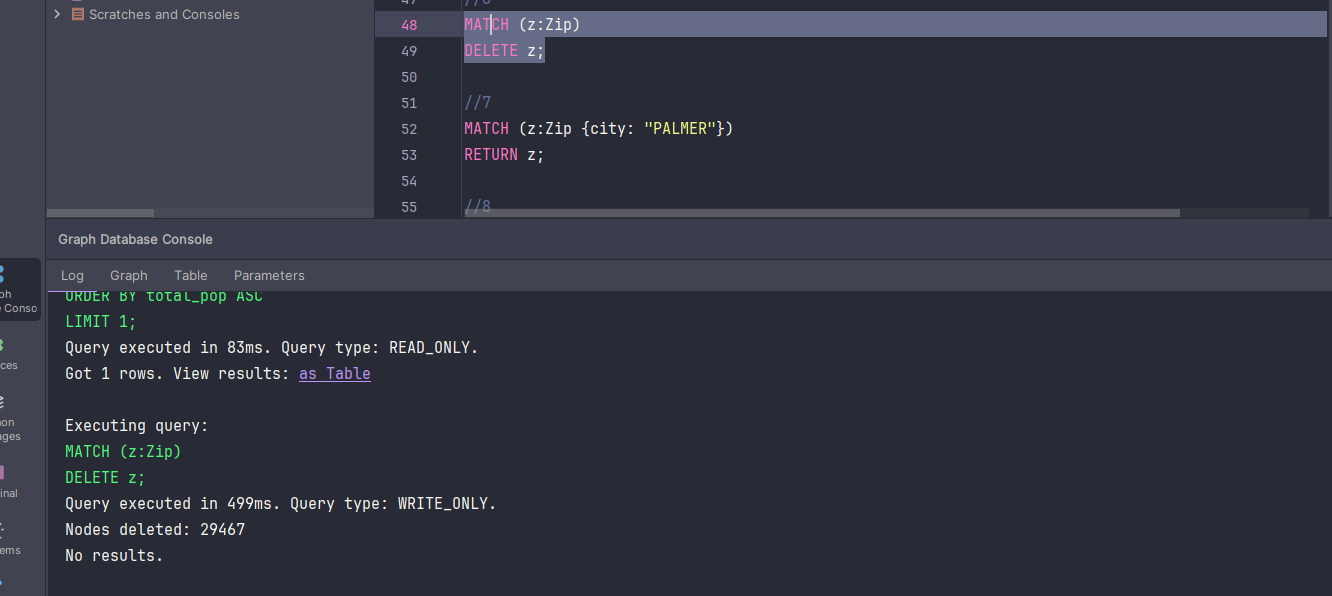
1. Cập nhật thông tin của một node khi biết zip code



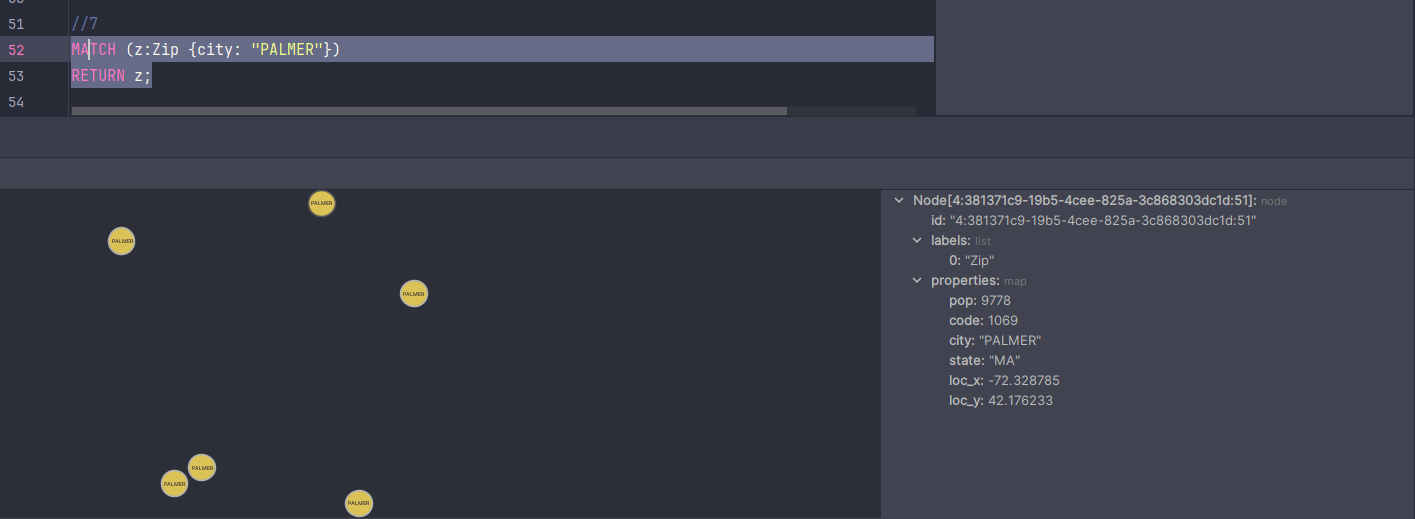
1. Xóa một node khi biết zip code



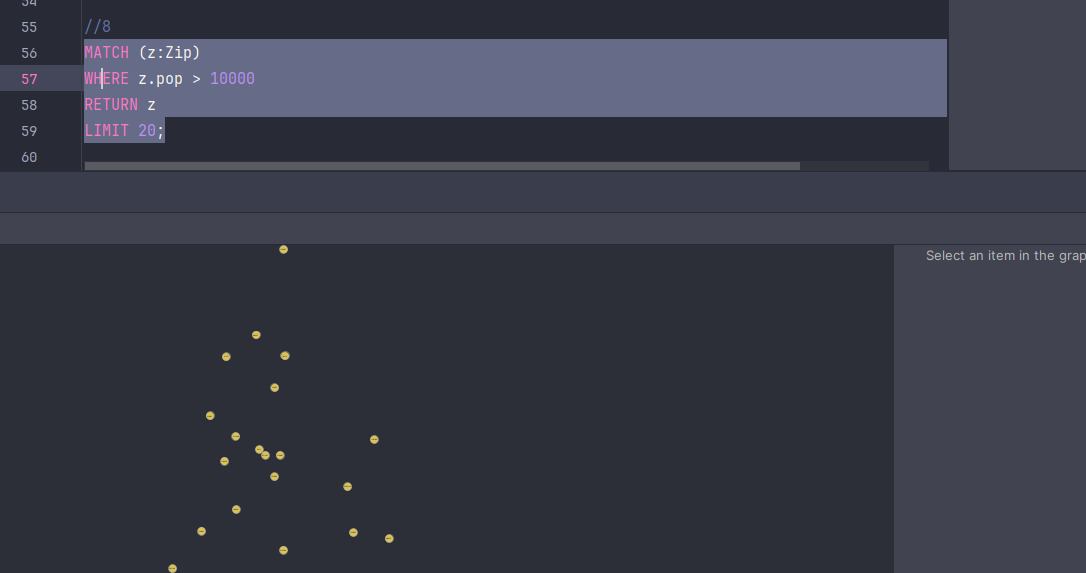
1. Xóa tất cả các node



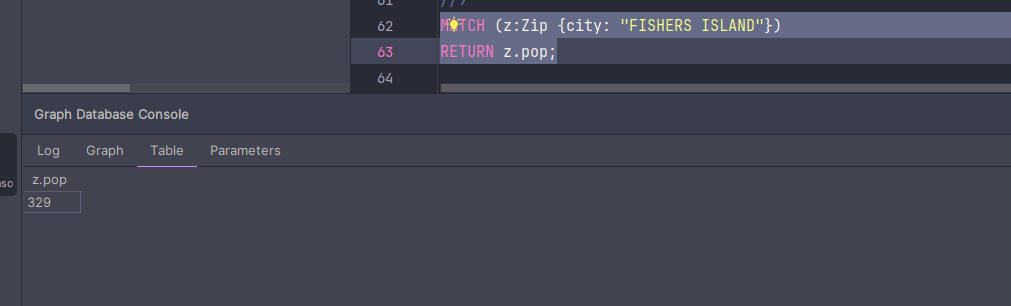
1. Tìm các node có city là PALMER



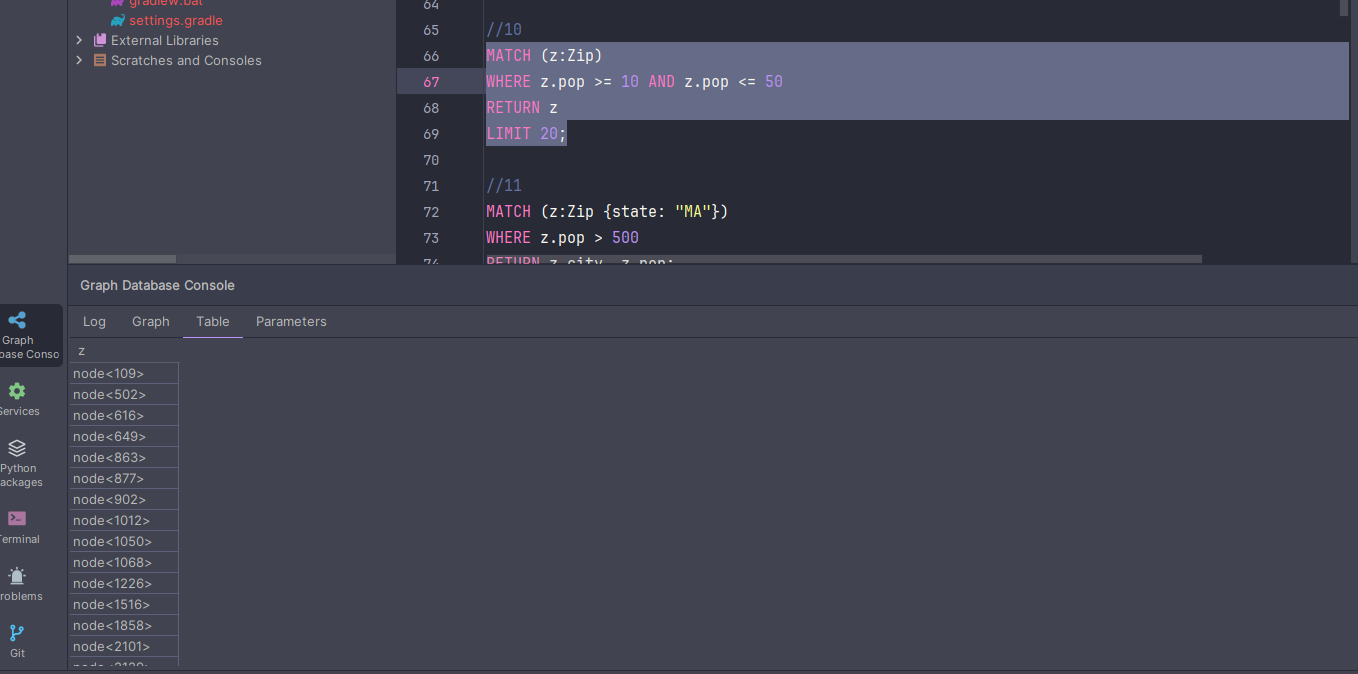
1. Tìm các node có dân số >100000



1. Tìm dân số của thành phố FISHERS ISLAND



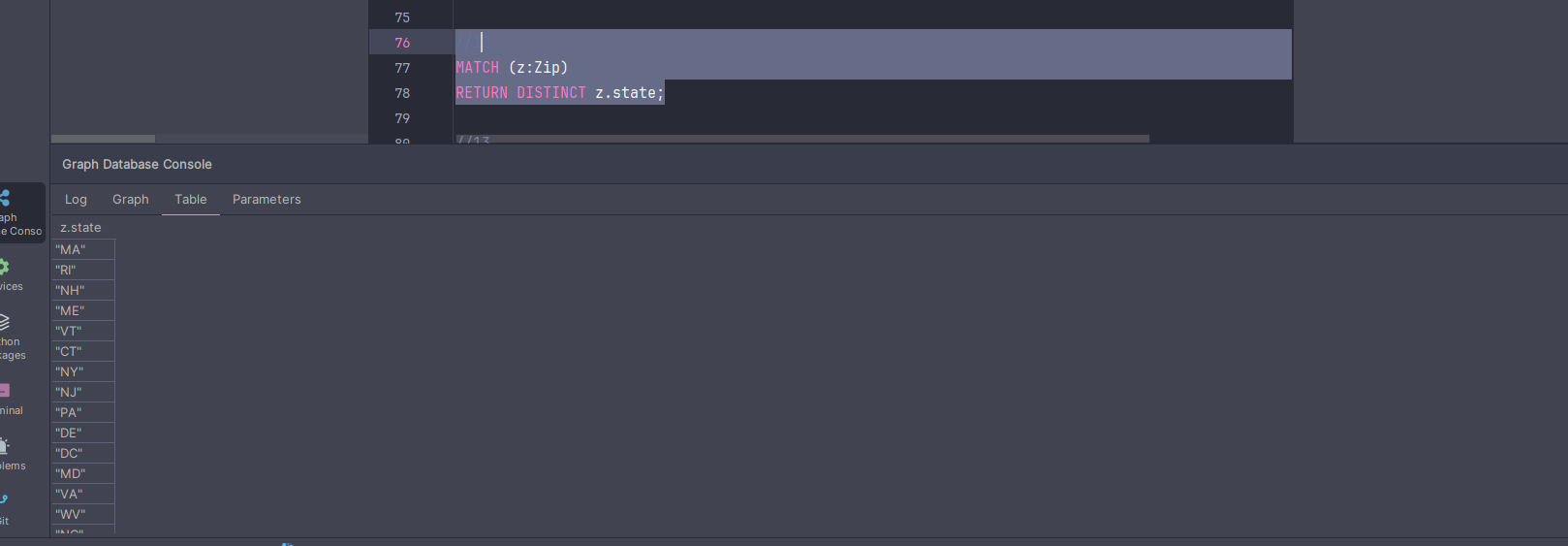
1. Tìm các thành phố có dân số từ 10 – 50



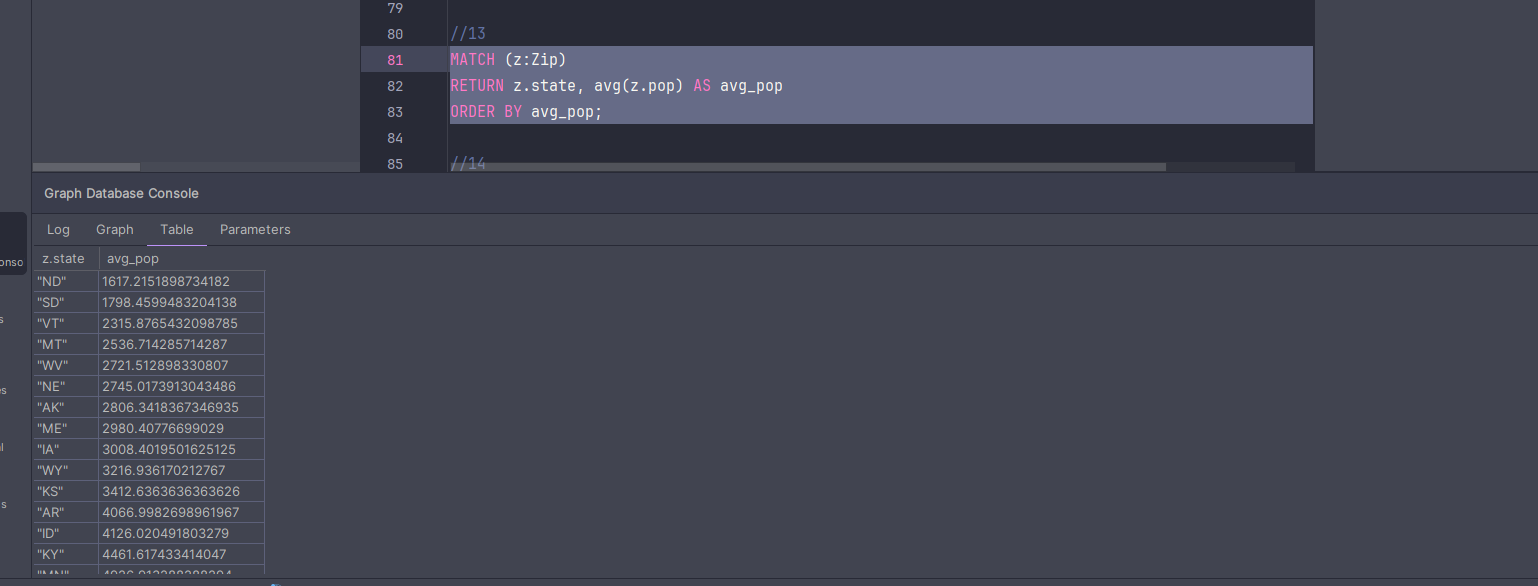
1. Tìm tất cả các city của bang MA có dân số trên 500



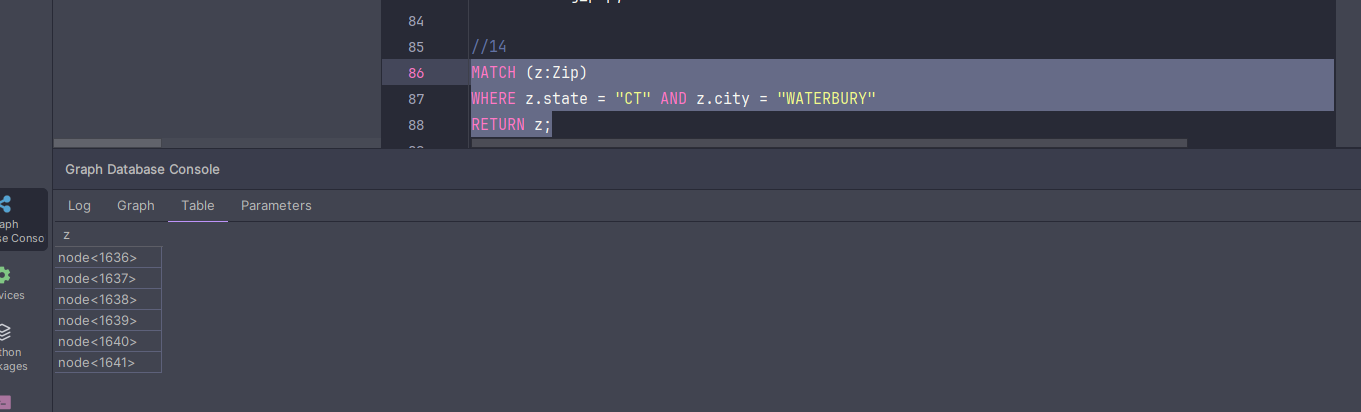
1. Tìm tất cả các bang (không trùng)



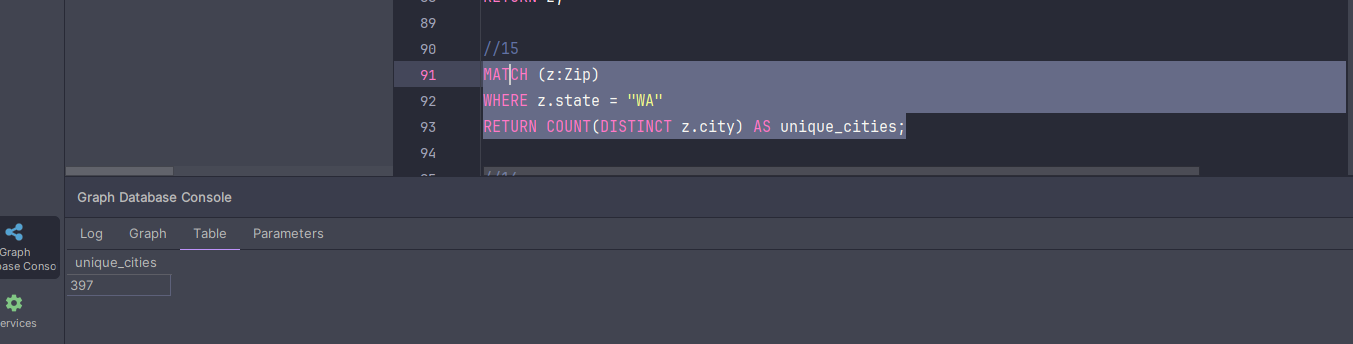
1. Tính dân số trung bình của mỗi bang



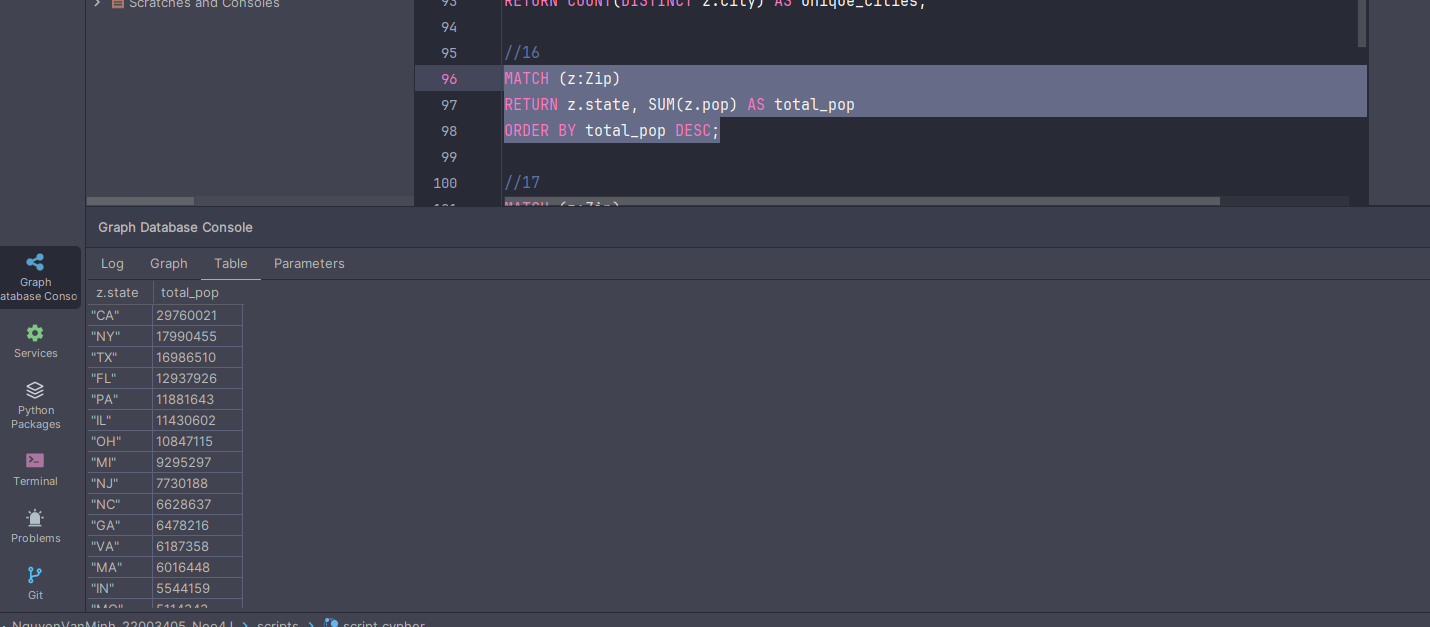
1. Tìm những document của bang 'CT' và thành phố 'WATERBURY'



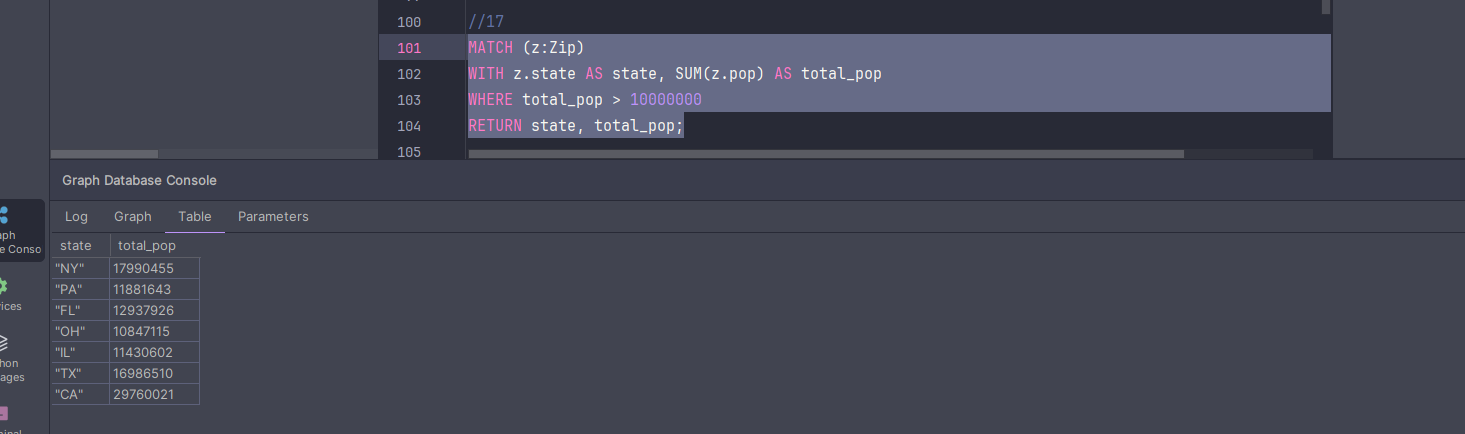
1. Bang WA có bao nhiêu city (nếu trùng chỉ tính 1 lần)



1. Tổng dân số của từng bang, sắp xếp theo tổng dân số giảm dần

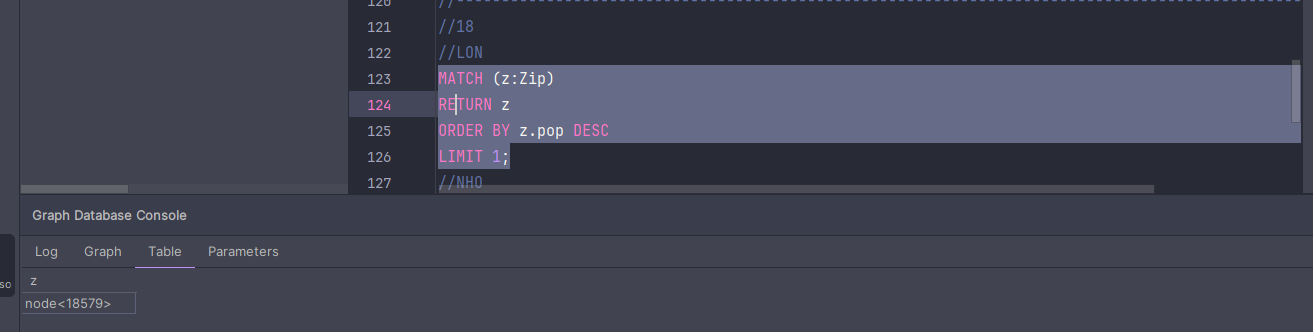


1. Tìm tất cả các bang có tổng dân số trên 10000000

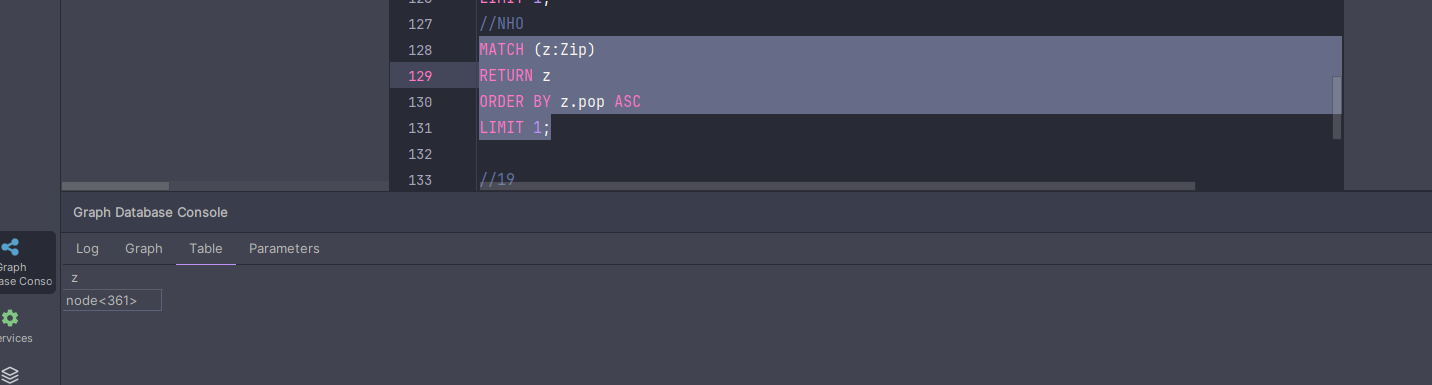


1. Tìm các node có dân số lớn (nhỏ) nhất

LỚN NHẤT

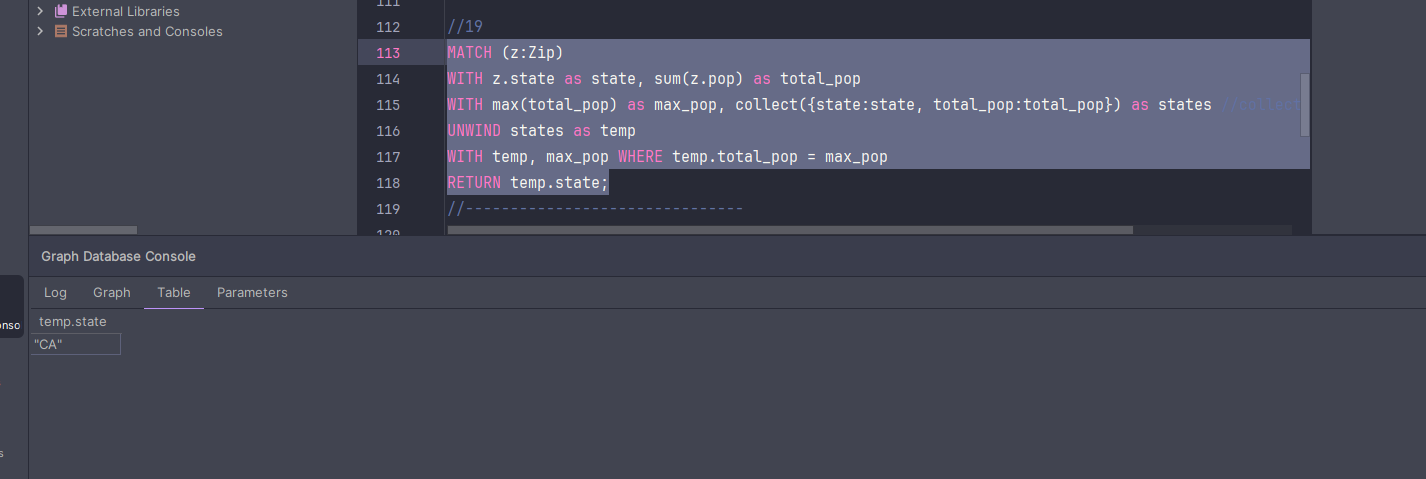


NHỎ



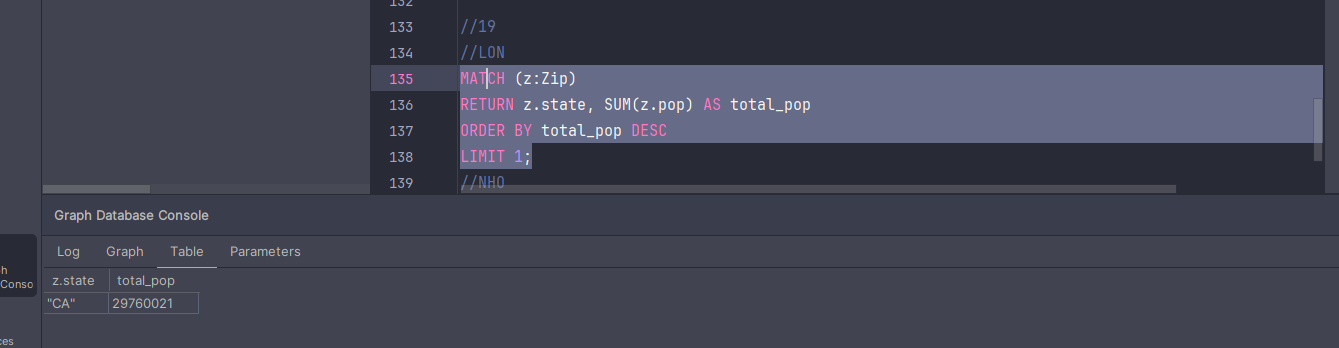
1. Tìm bang có tổng dân số lớn (nhỏ) nhất

Cách 1

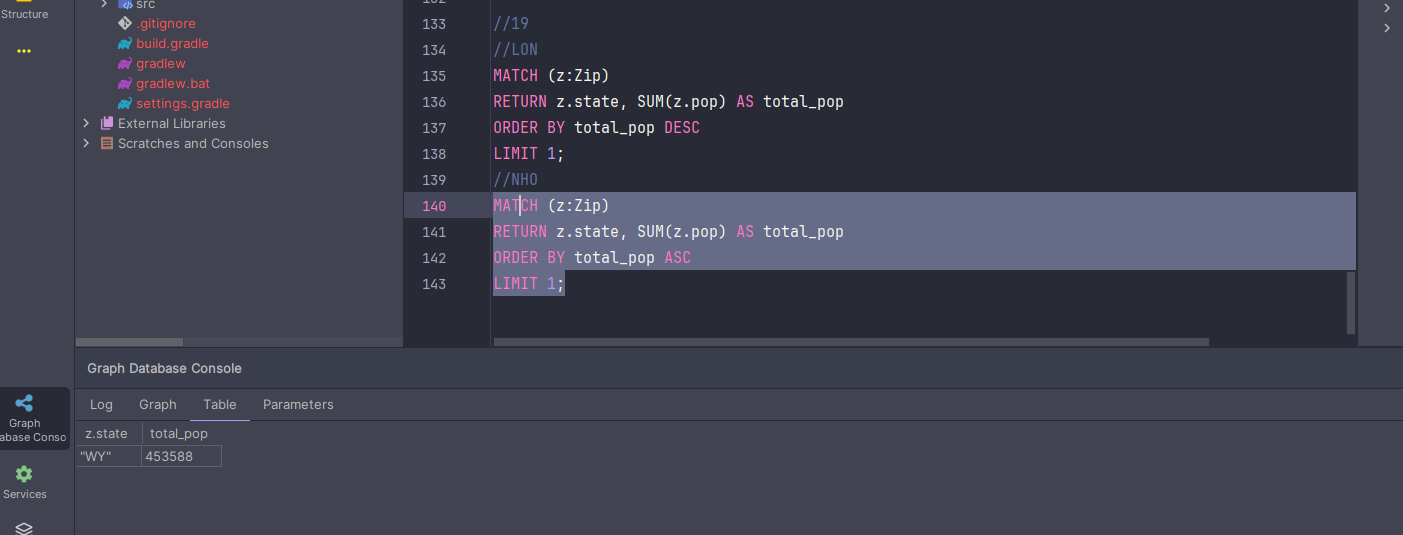


Cách 2

LỚN



NHỎ



# Bài tập

Bài toán quản lý các khóa học của trường đại học.

|  |  |  |  |
| --- | --- | --- | --- |
| **Course** | | | |
| **course\_id** | **name** | **hours** | **dept\_id** |
| CS101 | Programming | 4 | CS |
| CS201 | Algorithms | 3 | CS |
| CS202 | Systems | 3 | CS |
| MA101 | Algebra | 3 | Math |
| MA201 | Calculus | 4 | Math |
| MA301 | Analysis | 4 | Math |
| MU104 | Jazz | 3 | Music |
| EE102 | Circuits | 3 | EE |
| IE101 | Proabability | 3 | IE |
| IE102 | Statistics | 3 | IE |

|  |  |  |
| --- | --- | --- |
| **Student** | | |
| **student\_id** | **name** | **gpa** |
| 11 | Bush | 3.0 |
| 12 | Cruz | 3.2 |
| 13 | Clinton | 3.9 |
| 22 | Sanders | 3.0 |
| 33 | Trump | 3.8 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Department** | | | | |
| **dept\_id** | **name** | **dean** | **building** | **room** |
| CS | Computer Science | Rubio | Ajax | 100 |
| Math | Mathemagics | Carson | Acme | 300 |
| EE | Electrical Engineering | Kasich | Ajax | 200 |
| IE | Industrial Engineering | Cruz |  | 200 |
| Music | Musicology | Costello | North | 100 |

| **Enrollment** | |
| --- | --- |
| **course\_id** | **student\_id** |
| CS101 | 11 |
| MA101 | 11 |
| CS101 | 12 |
| CS201 | 22 |
| MA201 | 33 |
| EE102 | 33 |
| MA301 | 22 |

# Mô hình CSDL quan hệ

student(**student\_id**, name, gpa)

department(**dept\_id**, name, dean, building, room)

course(**course\_id**, name, hours, dept\_id) *//dept\_id là khóa ngoại tham chiếu dept(dept\_id)*

enrollment(course\_id, student\_id) *//course\_id là khóa ngoại tham chiếu course(course\_id) và student\_id là khóa ngoại tham chiếu student(student\_id)*

# Triển khai mô hình Đồ thị (Graph Model)

<https://arrows.app>

A red circle with arrows

AI-generated content may be incorrect.

A red circle with black text

AI-generated content may be incorrect.

* Chuyển đổi mô hình dữ liệu quan hệ sang mô hình dữ liệu đồ thị
  + Một row tương ứng là một node.
  + Tên table là tương ứng với tên label của node.
  + Join hoặc khóa ngoại tương ứng là một relationship giữa các node.
* Rows 🡪 Nodes, Table names 🡪 labels
  + Mỗi dòng trong bảng student trở thành một node có label là Student trong mô hình đồ thị.
  + Mỗi dòng trong bảng course trở thành một node có label là Course.
  + Mỗi dòng trong bảng department trở thành một node có label là Department.
* Joins to relationships
  + Giữa Student và Course có một relationship tên là ENROLLED *(sinh viên đã ghi danh vào khóa học).* Mỗi dòng trong bảng enrollment tương ứng thể hiện một relationship.
  + Giữa Course và Department có một relationship tên là BELONGS\_TO *(khóa học thuộc về khoa).*

# Importing the Data using Cypher

* [Link](https://www.mediafire.com/file/st9qczs39iyicpg/courses.rar/file) tải csv data
* Giải nén và copy các CSV files vào thư mục import của Neo4j DBMS

\.Neo4jDesktop\relate-data\dbmss\dbms-70f621bb-94c3-4eca-8740-cd10f4d5882b\import\courses

* Viết code Cypher để thực hiện:
  + Tạo các indexes hoặc các constraints cho dữ liệu trong đồ thị
  + Load dữ liệu từ các CSV files
  + Tạo các relationships giữa các nodes

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Add connection
  + Scheme: Neo4j
  + Host: localhost
  + Port: 7687
  + Username: neo4j
  + Password: 12345678
  + Database: coursedb
* Load dữ liệu từ các CSV files
* Tạo các indexes hoặc các constraints cho dữ liệu trong đồ thị
* Tạo các relationships giữa các nodes

# Querying the Graph

1. Liệt kê danh sách n sinh viên
2. Tìm kiếm sinh viên khi biết mã số
3. Tìm danh sách khóa học thuộc của một khoa nào đó khi biết mã khoa
4. Cập nhật name = “Mathematics” cho department\_id = “Math”
5. Cập nhật name = “Rock n Roll” cho department\_id = “Music”
6. Thêm khóa học vào khoa IE: IE202, Simulation, 3 hours.
7. Xóa toàn bộ các khóa học
8. Liệt kê tất cả các khoa
9. Liệt kê tên của tất cả các trưởng khoa
10. Tìm tên của trưởng khoa CS
11. Liệt kê tất cả các khóa học của CS và IE
12. Liệt kê danh sách các tên của các sinh viên đăng ký học khóa học CS101
13. Tổng số sinh viên đăng ký học của mỗi khoa
14. Tổng số sinh viên đăng ký học của mỗi khoa, kết quả sắp xếp theo mã khoa
15. Tổng số sinh viên đăng ký học của mỗi khoa, kết quả sắp xếp theo số sinh viên
16. Liệt kê danh sách tên của các trưởng khoa mà các khoa này không có sinh viên đăng ký học
17. Danh sách khoa có số sinh viên đăng ký học nhiều nhất
18. Danh sách sinh viên có gpa >= 3.2, kết quả sắp xếp giảm dần theo gpa