uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Database

Lourdes Moreno López
Paloma Martínez Fernández
José Luis Martínez Fernández
Rodrigo Alarcón García

Lab demo 4 (Topic neo4j (3.4))

OpenCourseWare

Data Base

Bachelor in Data Science and Engineering

SUBJECT: Lab demo 4 (Topic 3.4: neo4j)

uc3m Universidad Carlos III de Madrid

In this lab demo, we will work the neo4j application and will use the language Cypher.

Part 1: Create nodes and relationships with their properties

1.1. Create nodes with the given properties. (see Figure 1)

CREATE (n {name: {value}})

Example:

```
CREATE (Maria:Person {name:'Maria', born:1974})
CREATE (Juan:Person {name:'Juan', born:1977})
CREATE (Jose:Person {name:'José', born:1981})
CREATE (Hugo:Person {name:'Hugo', born:1980})
CREATE (Natalia:Person {name:'Natalia', born:1977})
CREATE (Roberto:Person {name:'Roberto', born:1975})
CREATE (Rosa:Person {name:'Rosa', born:1982})

CREATE (UC3M:University {name:'Universidad Carlos III de
Madrid', central_office:'Leganés'})
CREATE (UCM:University {name:'Universidad Complutense de
Madrid', central_office:'Madrid'})
```

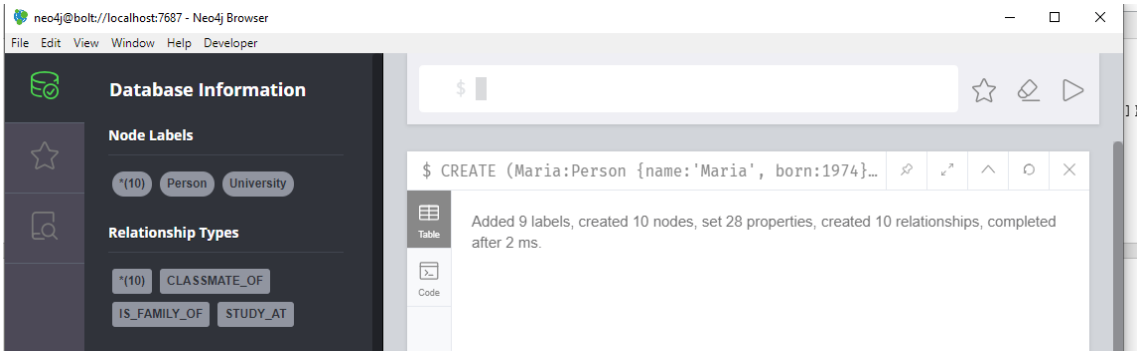


Figure 1

Recover all nodes and relationships (display graph) (see Figure 2 and Figure 3)

MATCH (n) RETURN n

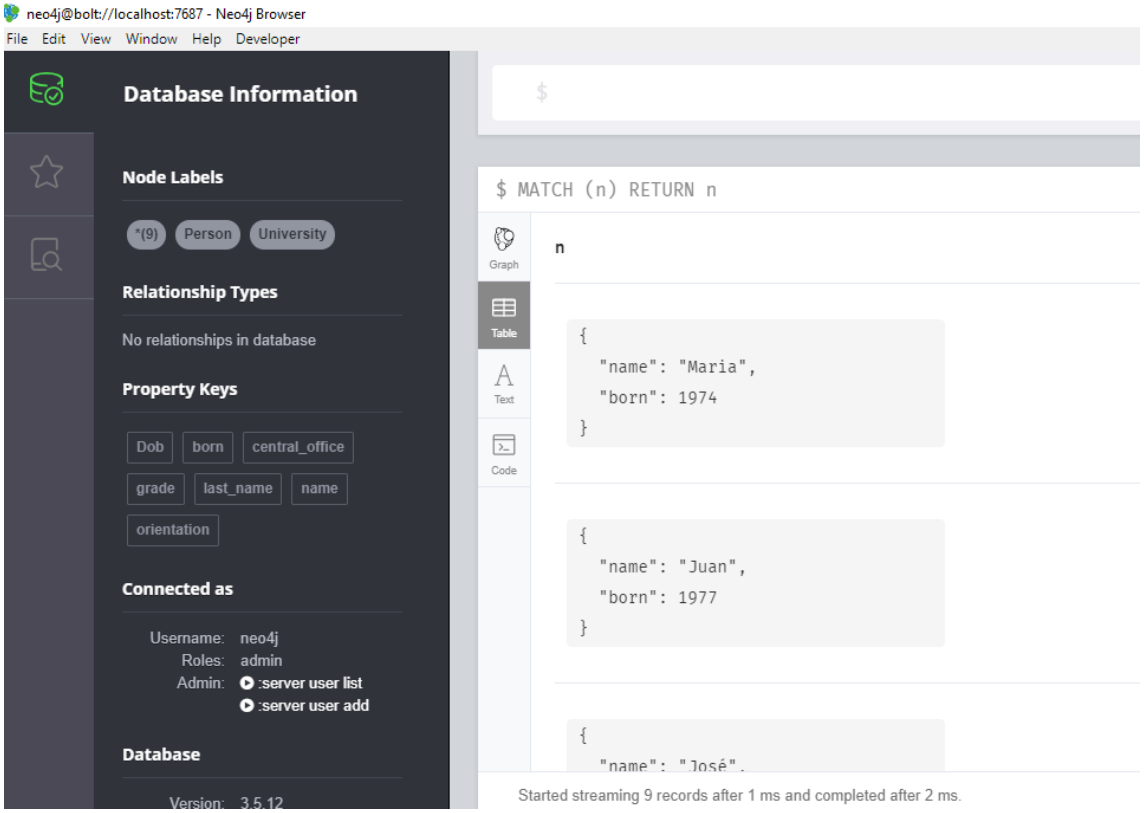


Figure 2

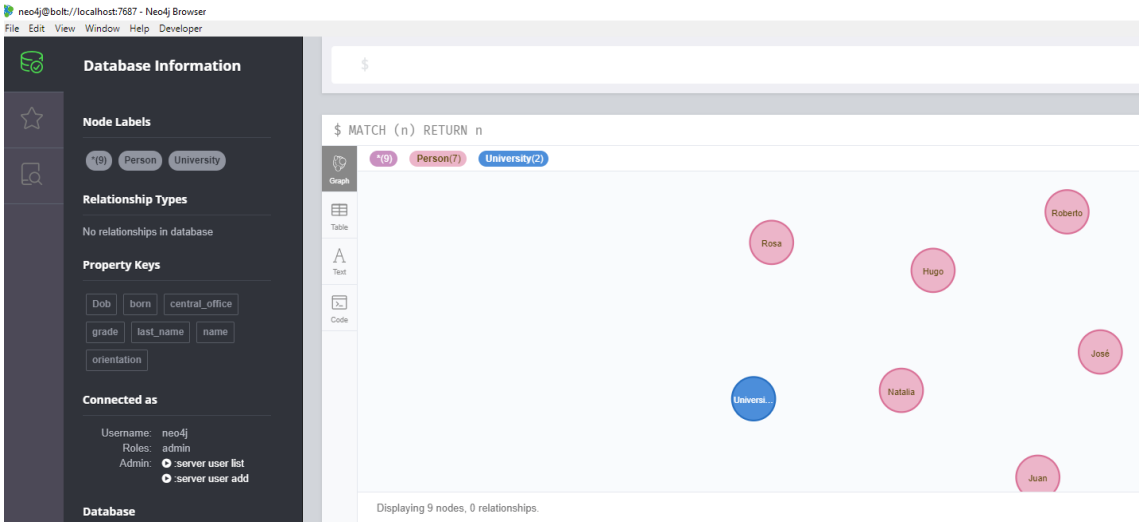


Figure 3

1.1. Create relationships with the given properties

CREATE (n)-[r:KNOWS]->(m)

```
CREATE
(Maria)-[:CLASSMATE_OF {role:['Database course']}]->(Juan),
(Maria)-[:CLASSMATE_OF {role:['Discrete mathematics course']}]-
>(Jose),
```

```
(Juan)-[:CLASSMATE_OF {role:[' Discrete mathematics course']}]-
>(Hugo),
(Roberto)-[:CLASSMATE_OF {role:['Automata theory and compilers
course']}]->(Natalia),
(Miriam)-[:CLASSMATE_OF {role:['Automata theory and compilers
course']}]->(Rosa)

CREATE
(Maria)-[:STUDY_AT {campus:['Leganes']}]->(UC3M),
(Rosa)-[:STUDY_AT {campus:['Paraninfo']}]->(UCM),
(Juan)-[:STUDY_AT {campus:['Colmenarejo']}]->(UC3M),
(Hugo)-[:STUDY_AT {campus:['Colmenarejo']}]->(UC3M)

CREATE
(Juan)-[:IS_FAMILY_OF { relationship:['cousin']}]->(Rosa)
```

Recover all nodes and relationships (display graph) (see Figure 4)

MATCH (n) RETURN n

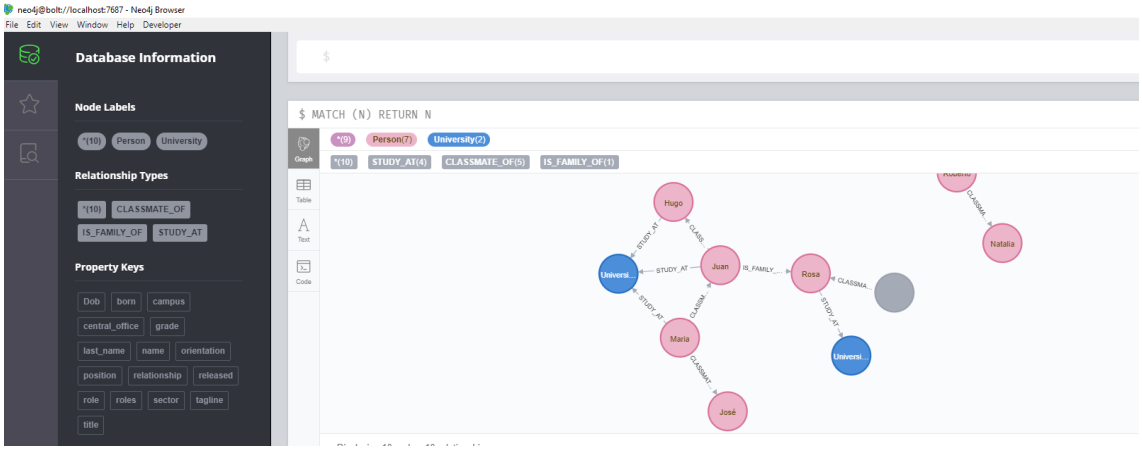
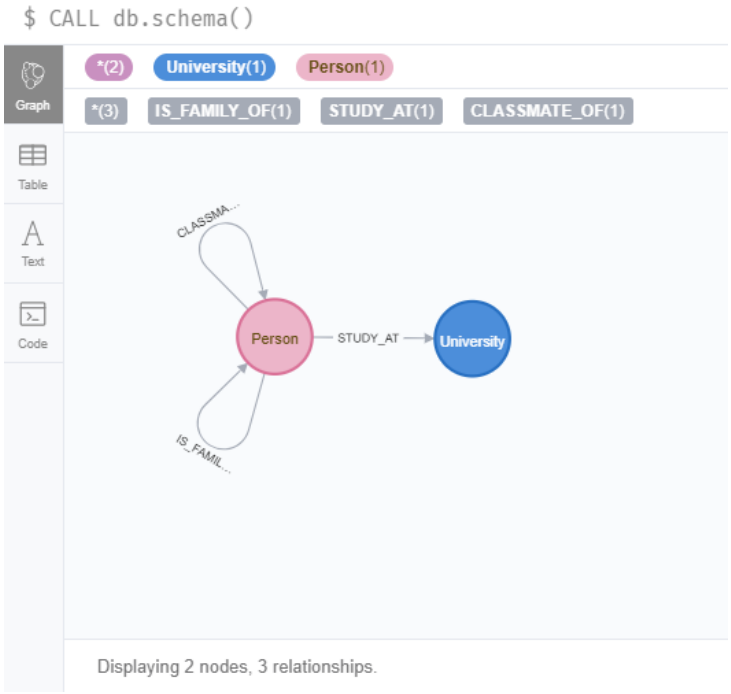


Figure 4

Write a query to display the schema of your database.

```
CALL db.schema()
```



Part 2: Queries

2.1. Search by node property

Search for Maria (see Figure 5)

```
MATCH (hername {name: "Maria"}) RETURN hername
```

Figure 5

2.2: Search by node and relationship

Search Maria's classmates (see Figure 6)

```
MATCH (Maria {name: "Maria"})-[:CLASSMATE_OF]->(classmates)
RETURN classmates
```

Figure 6

2.3 Search by node and relationship

All people who study at UC3M (see Figure 7)

```
MATCH (people)-[:STUDY_AT]-> (UC3M {name: 'Universidad Carlos
III de Madrid'}) RETURN people.name
```

Figure 7

2.4. Search by two chained relationships

List of Maria's classmates (see Figure 8)

```
MATCH (Maria {name: 'Maria'})-[:CLASSMATE_OF]->()-
[:CLASSMATE_OF]->(classmates_of_classmate) RETURN
classmates_of_classmate.name
```

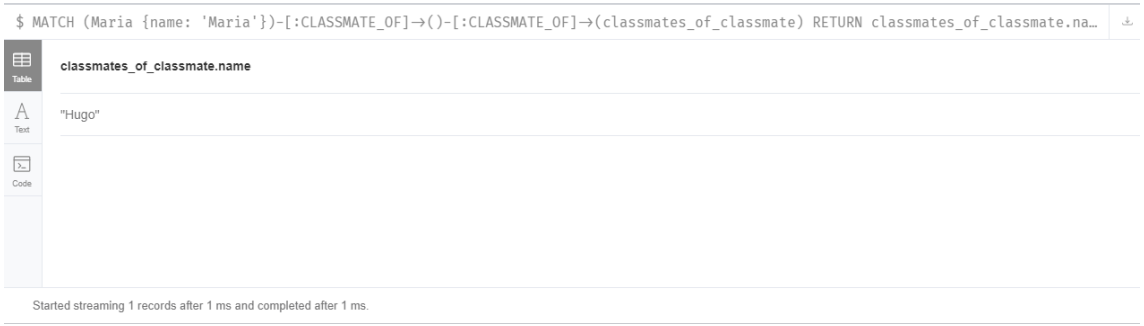


Figure 8

2.5. Search for a relationship

List of people and who study in universities (see Figure 9)

```
MATCH (person)-[:STUDY_AT]->(university) RETURN person.name,
university.name
```

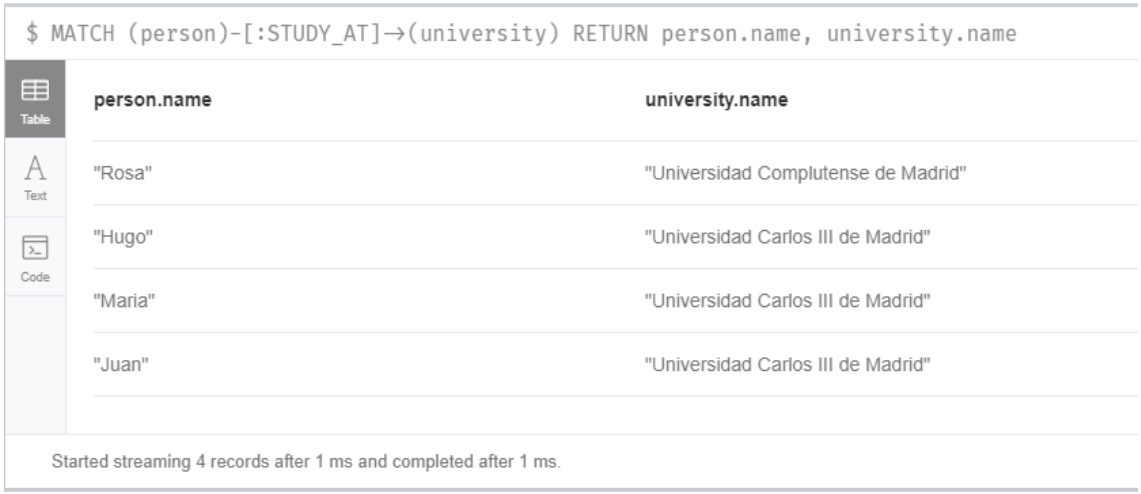


Figure 9

2.6: Search with conditions

People born after 1979 (see Figure 10)

```
MATCH (p:Person) WHERE p.born > 1979 RETURN p.name
```

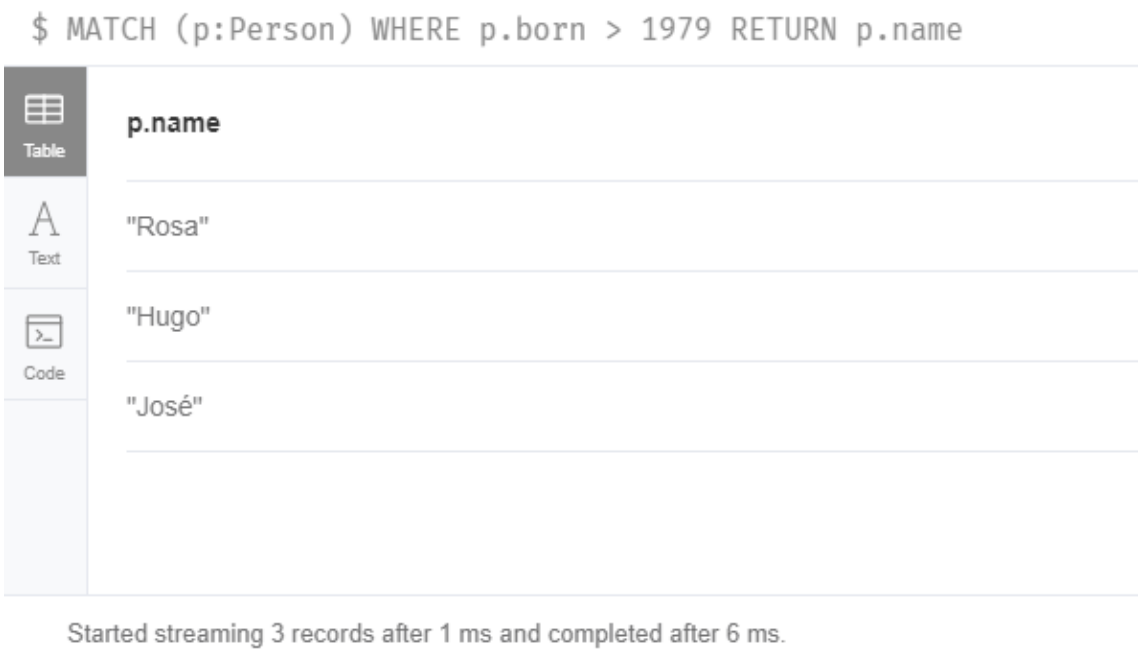


Figure 10

PART3 Update elements of the graph

3.1. Update properties to a node (see Figure 11)

```
MERGE (p:Person {name: 'Maria'}) SET p.age = 28, p.hair =
'brown' RETURN p
```

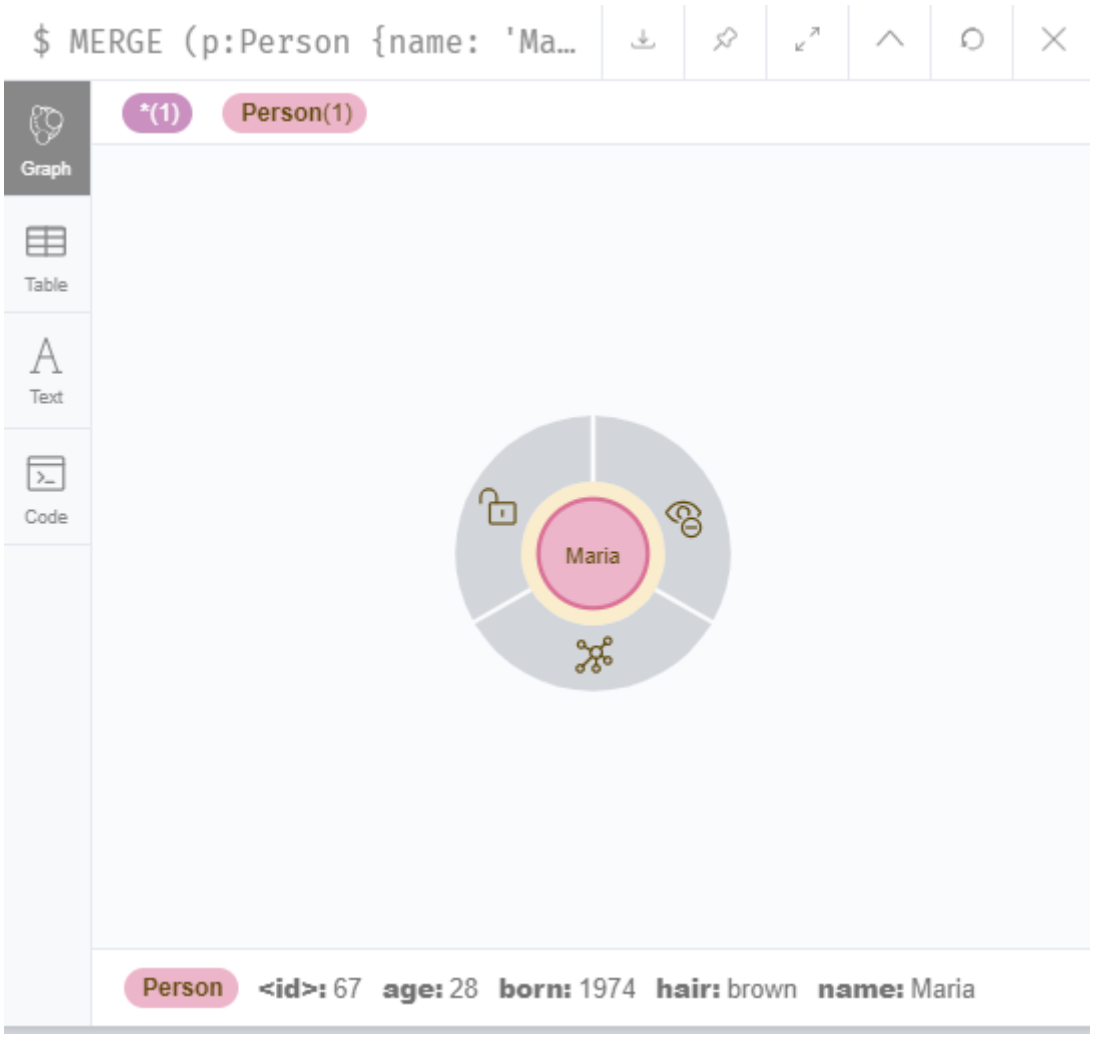


Figure 11

3.2 Update properties to a relationship (see Figure 12)

```
MERGE (Maria)-[r:CLASSMATE_OF]->(Juan) SET r.ages =1 RETURN r
```

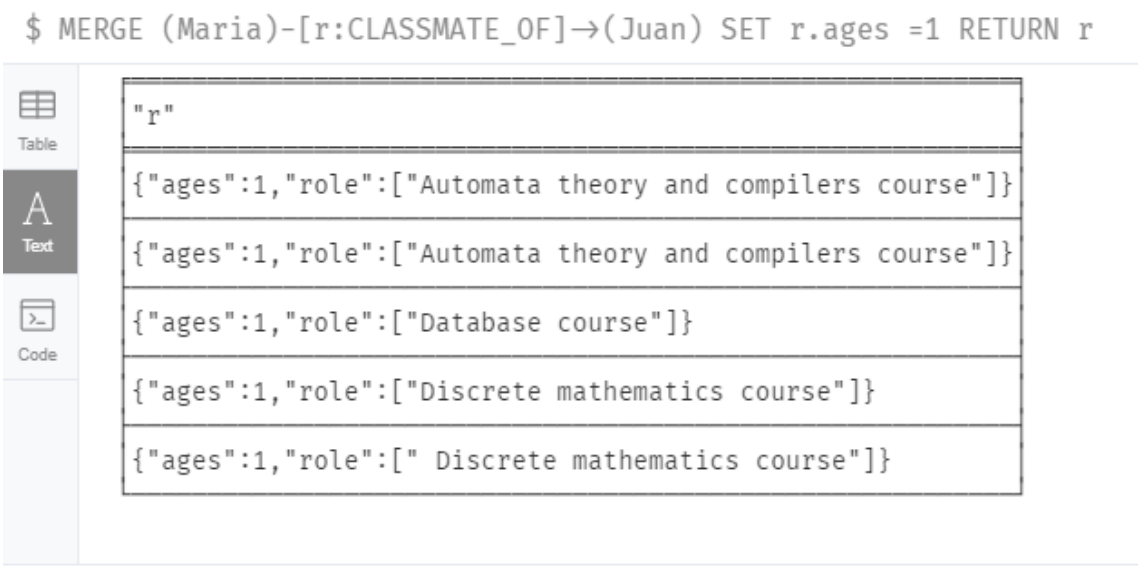


Figure 12

PART4: Aggregation functions

List of the years of birth of the people grouped and indicating the number. (See Figure 13)

```
MATCH (p:Person) return p.born, count (*)
```

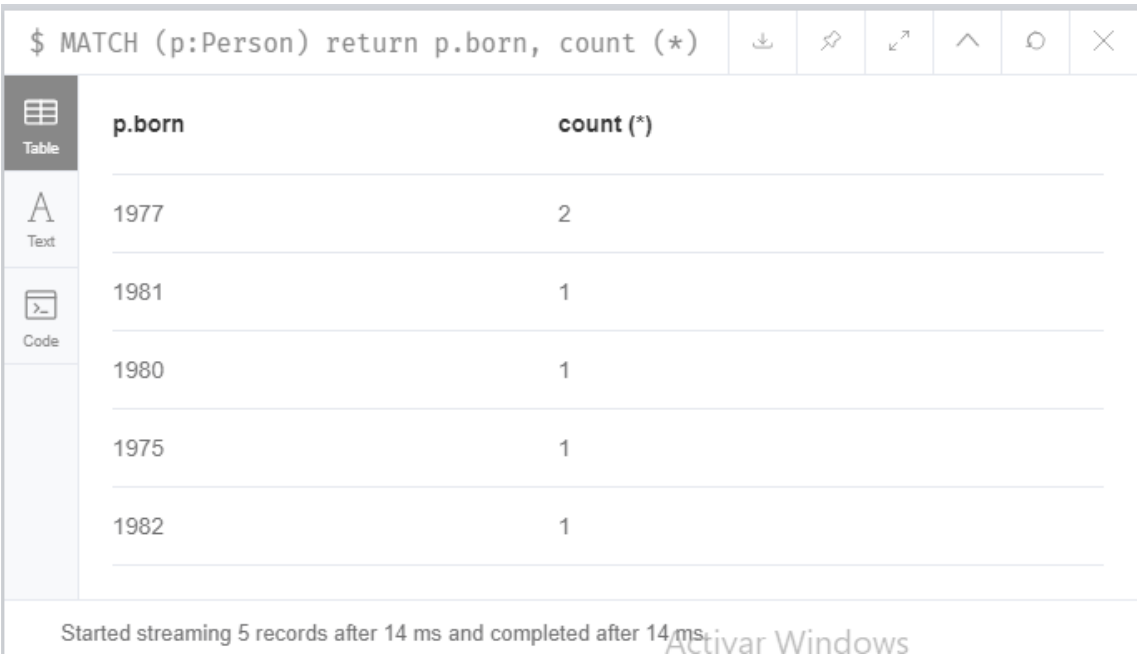


Figure 13

```
MATCH (p:Person)--(u: University)RETURN u.central_office,
count(*)
```

(see Figure 14)

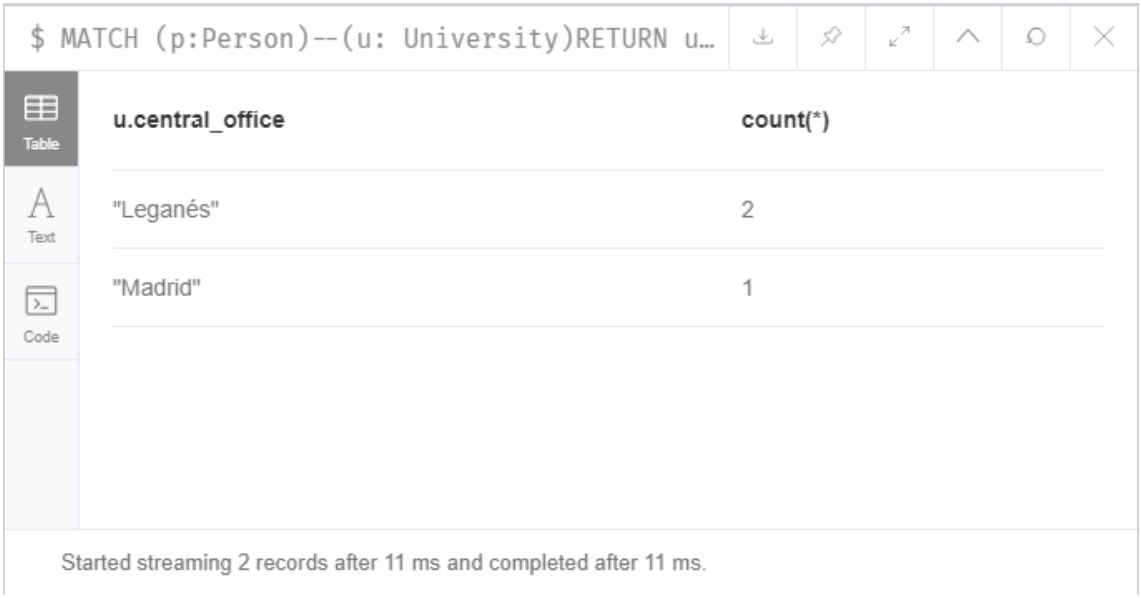


Figure 14

PART4: Delete graph elements

4.1. Delete relationships between nodes

```
MATCH (Maria)-[:CLASSMATE_OF]->(Juan) DELETE r
```

4.2. Delete nodes

In order to delete the nodes, the relationships between them must be deleted.

```
MATCH (p:Person {name: 'Maria'}) DELETE p
```

Return error because the relationships between them must be deleted

```
MATCH (p:Person {name: 'Maria'}) -[r]-() DELETE p, r
```

4.3 Delete the entire graph

```
MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r
```