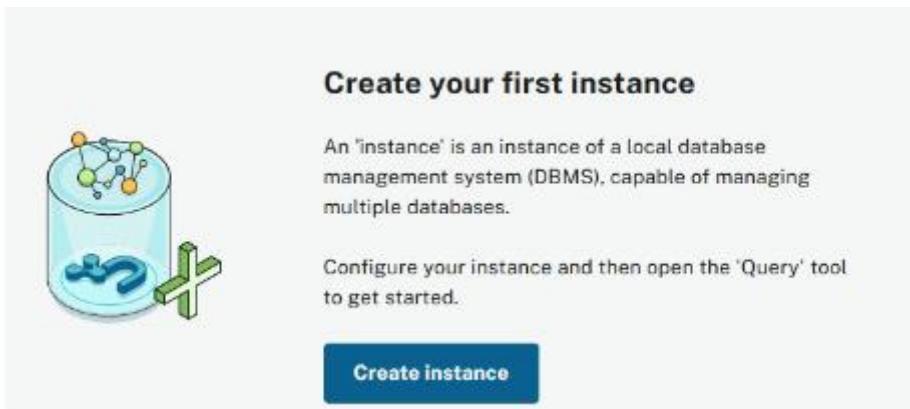


## TP6 : Neo4j

### 1. Accès à une base neo4J

#### 1.1. Préparation



- monInstance2025
- la database user n'est pas modifiable
- mdp 12345678

**Create Instance** ×

**Instance details**

Instance name  
monInstance2025

Instance names must be unique

Neo4j version  
2025.09.0

**Create database user**

Database user  
neo4j

Password  
..... (copy)

Password must be at least 8 characters long

- il faut choisir ou créer base de données

**Create database** ×

Database name  
monProjet

Cancel Create

Query Connect ^ ...

## 2. Exercice 1

Se connecter à une base

:use mabase



```
monprojet$  
neo4j$ :use monprojet  
✓ New queries will use monprojet
```

- Pour tout supprimer :

MATCH (n) DETACH DELETE n;

- Pour afficher tous les noeuds :

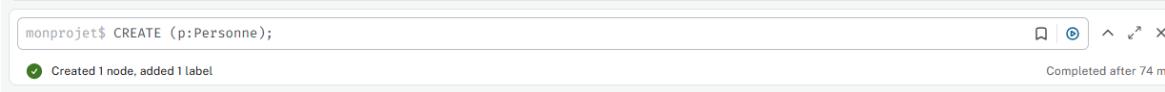
MATCH (n) RETURN n;

- Pour afficher toutes les liaisons :

MATCH p=()-[]->() RETURN p;

- Créer les personnes suivantes

- Une personne sans propriété. Personne sera une catégorie.



```
monprojet$ CREATE (p:Personne);  
✓ Created 1 node, added 1 label  
Completed after 74 ms
```

- Deux personnes avec comme propriété nom : DELHOUMI



```
monprojet$ CREATE (p1:Personne {nom: 'DELHOUMI'}), (p2:Personne {nom: 'DELHOUMI'})  
✓ Created 2 nodes, set 2 properties, added 2 labels  
Completed after 33 ms
```

- Une personne avec comme second label Prof. nom : DELHOUMI, age 52 ans



```
monprojet$ CREATE (p:Personne:Prof {nom: 'DELHOUMI', age: 52})  
✓ Created 1 node, set 2 properties, added 2 labels  
Completed after 46 ms
```

- Une personne avec comme second label Etudiant. nom : VOLEUR



```
monprojet$ CREATE (p:Personne:Etudiant {nom: 'VOEUR'})  
✓ Created 1 node, set 1 property, added 2 labels  
Completed after 24 ms
```

- Une personne avec comme second label Etudiant. nom : TRICHEUR, nom : Enzo, age : 39



```
monprojet$ CREATE (p:Personne:Etudiant {nom: 'TRICHEUR', prenom: 'Enzo', age: 39})  
✓ Created 1 node, set 3 properties, added 2 labels  
Completed after 157 ms
```

- Une personne avec comme second label Etudiant. nom : ESTJAUNE, prénom : Gilles , age : 36

```
monprojet$ CREATE (p:Personne:Etudiant {nom: 'ESTJAUNE', prenom: 'Gilles', age: 36})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 10 ms

- Une personne avec comme second label Secrétaire. nom : LUGNIER, prénom : Gwendoline , age : 30

```
monprojet$ CREATE (p:Personne:Secrétaire {nom: 'LUGNIER', prenom: 'Gwendoline', age: 30})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 32 ms

- Une personne avec comme second label Prof. nom : SECOUARD, prénom : Stéphane , age : 59

```
monprojet$ CREATE (p:Personne:Prof {nom: 'SECOUARD', prenom: 'Stéphane', age: 59})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 27 ms

- Une personne avec comme second label Prof. nom : ZIMMERMANN, prénom : Albrecht , age : 52

```
monprojet$ CREATE (p:Personne:Prof {nom: 'ZIMMERMANN', prenom: 'Albrecht', age: 52})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 15 ms

- Une personne avec comme second label Prof. nom : ZIMMERMANN, prénom : Urs , age : 49

```
monprojet$ CREATE (p:Personne:Prof {nom: 'ZIMMERMANN', prenom: 'Urs', age: 49})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 22 ms

- Une personne avec comme second label Prof. nom : ZINNERMANN, prénom : Robert , age : 45

```
monprojet$ CREATE (p:Personne:Prof {nom: 'ZINNERMANN', prenom: 'Robert', age: 45})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 14 ms

- Une personne avec comme second label Prof. nom : PORCQ, prénom : Eric , age : 28

```
monprojet$ CREATE (p:Personne:Prof {nom: 'PORCQ', prenom: 'Eric', age: 28})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 12 ms

- Une personne avec comme second label Directeur. nom : TRAVERT, prénom : Carine , age : 50

```
monprojet$ CREATE (p:Personne:Directeur {nom: 'TRAVERT', prenom: 'Carine', age: 50})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 31 ms

- Une personne avec comme second label Directeur. nom : ADOUI, prénom : Lamri , age : 55

```
monprojet$ CREATE (p:Personne:Directeur {nom: 'ADOUI', prenom: 'Lamri', age: 55})
```

✓ Created 1 node, set 3 properties, added 2 labels

Completed after 14 ms

- Une catégorie Matière avec comme propriété. nom : Mathématiques

```
monprojet$ MATCH (m:Matière {nom: 'Anglais'}) DETACH DELETE m
```

✓ Deleted 2 nodes

Completed after 39 ms

```
monprojet$ MERGE (m:Matière {nom: 'Mathématiques'}) SET m.nom = 'Anglais' return m
```

```
monprojet$ CREATE (m:Matière {nom: 'Mathématiques'}) RETURN m
```

- Une catégorie Matière avec comme propriété. nom : Anglais

- Ajouter aux nœuds existant des informations suivantes :

- Pour Voleur, ajouter les propriétés prenom = "Alexis" age = 31 ans, code\_postal = 14000 , ville = CAEN

```
1 MATCH (p:Personne {nom: 'VOLEUR'})
2 SET p.prenom = 'Alexis', p.age = 31, p.code_postal = '14000', p.ville = 'CAEN'

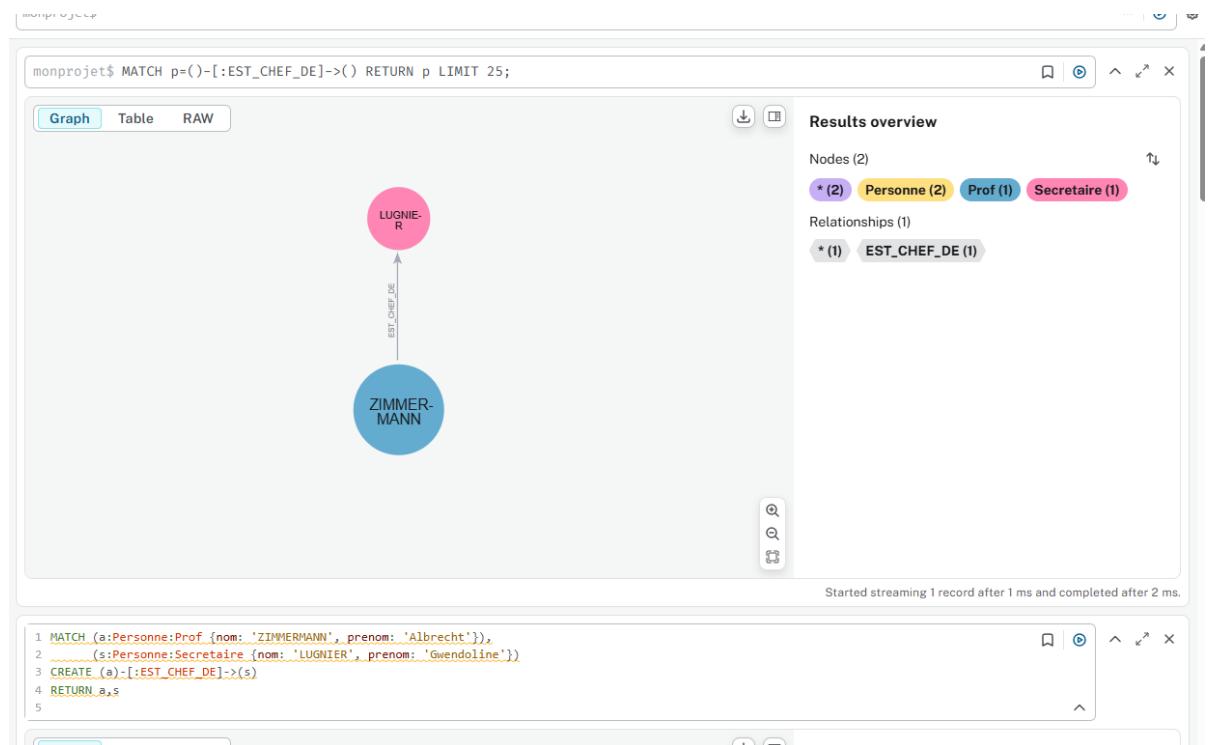
✓ Set 4 properties
```

Completed after 62 ms

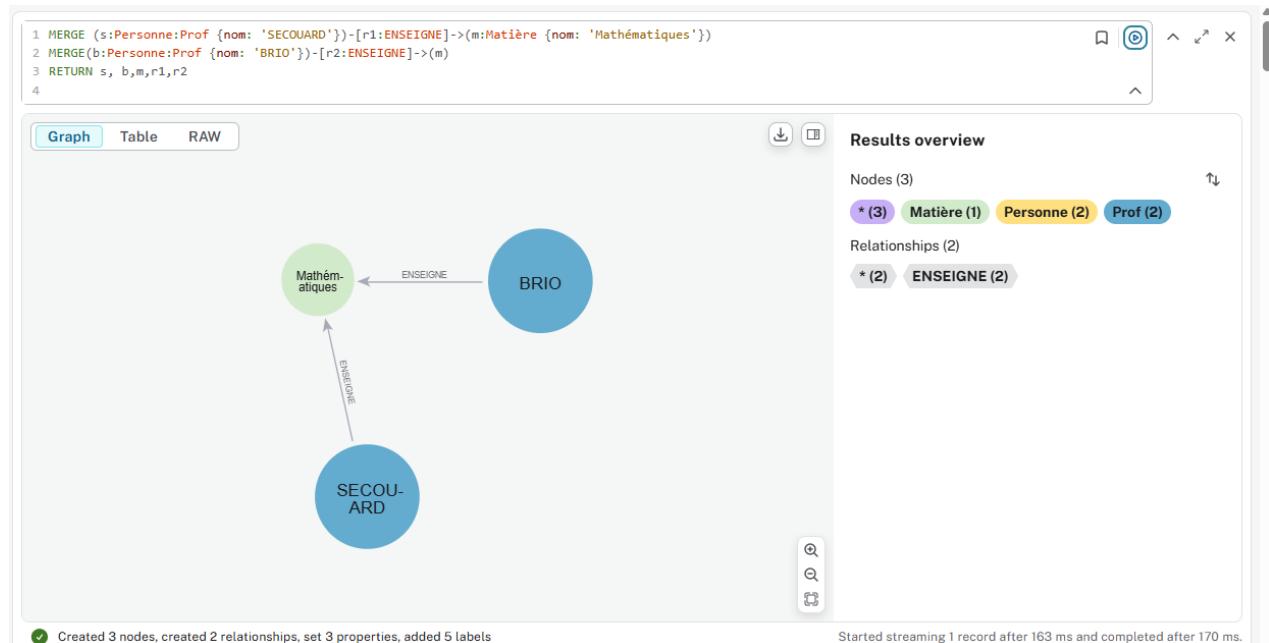
- Pour Voleur, ajouter la propriété un tableau de notes = [12,13.5,2]



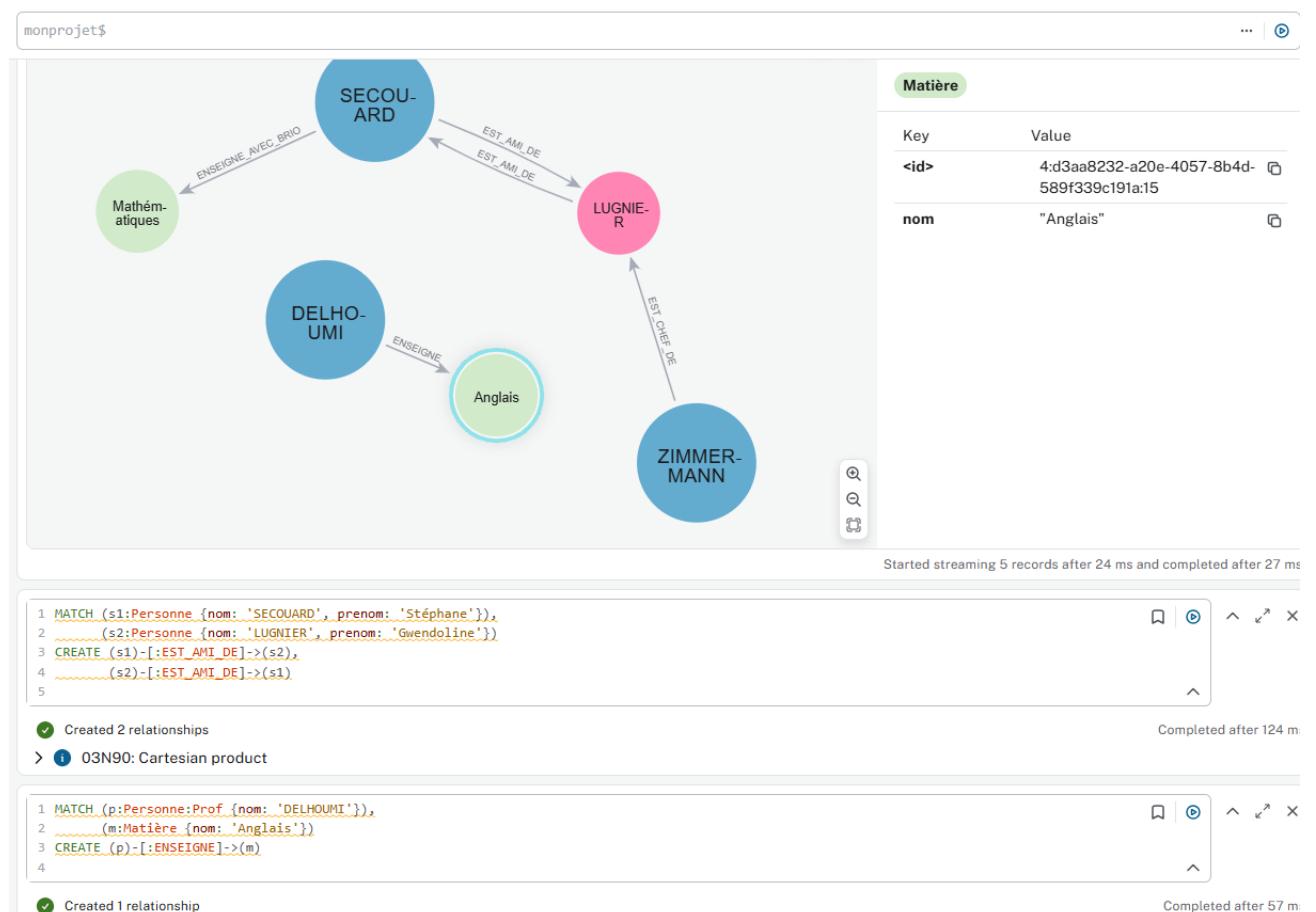
- Une relation : Zimmermann Albrecht est chef de la secrétaire



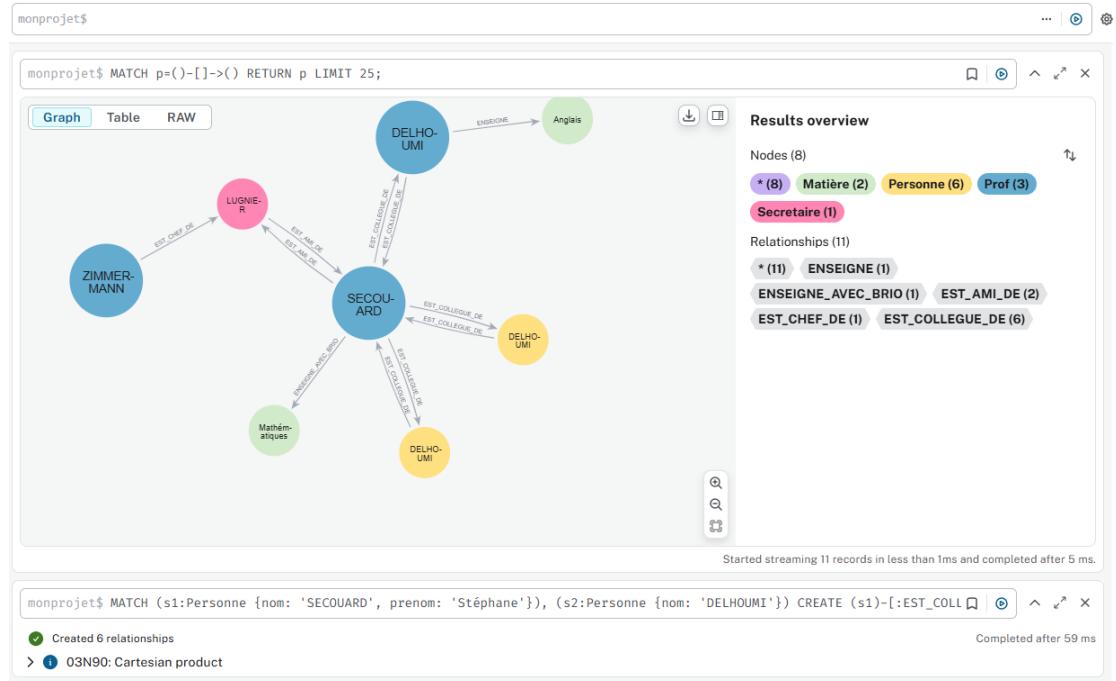
- Une relation : Secouard enseigne la matière Mathématique avec BRIO.



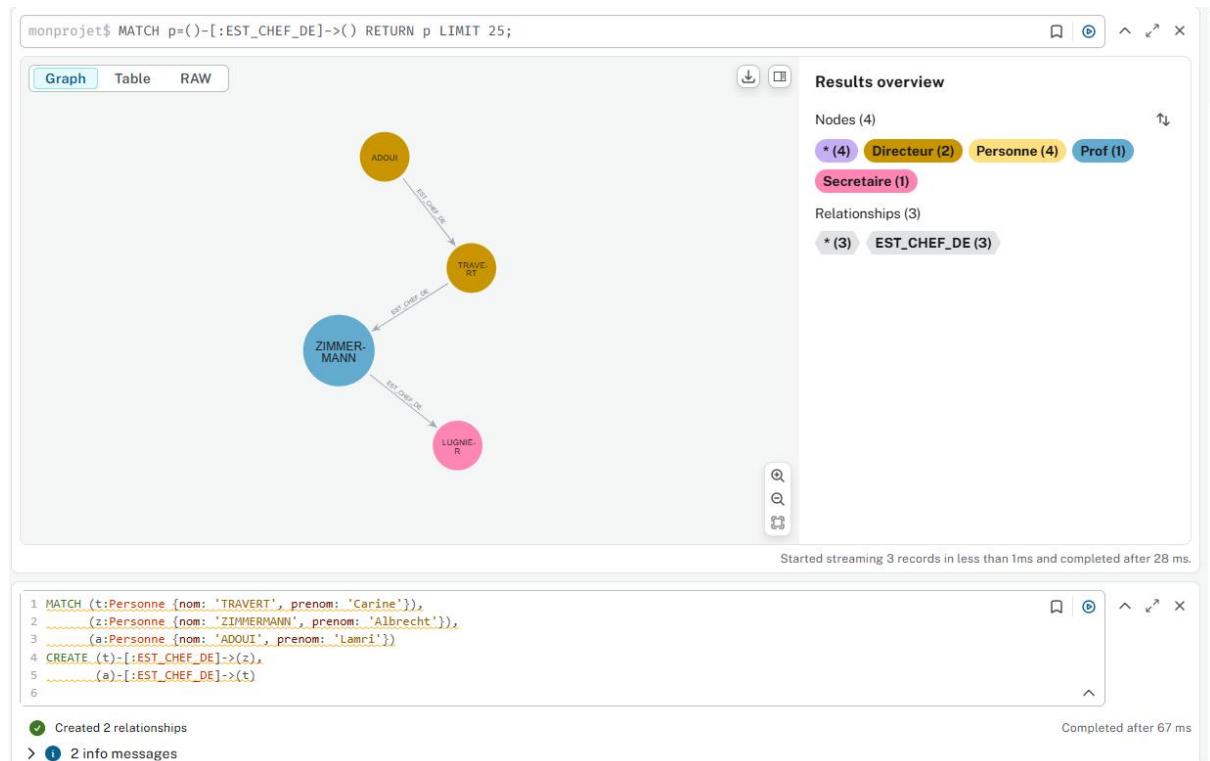
- Une relation : Delhoumi enseigne la matière Vous Savez Quoi



- Une relation : Secouard est ami de LUGNIER
- Une relation : Lugnier est ami de Secouard
- Une relation : Secouard est collègue de Delhoumi
- Une relation : Delhoumi est collègue de Secouard



- Une relation : Travert est chef de Zimmermann Albrecht
- Une relation : Adoui est chef de Travert



- Visualiser des informations

- les noeuds sans nom

```
1 //Visualiser les noeuds sans nom
2 MATCH (n)
3 WHERE n.nom IS NULL // Filtre les nœuds sans la propriété 'nom'
4 RETURN n
```

- les noeuds avec nom Delhoumi

The screenshot shows the Neo4j browser interface. The query entered is: `monprojet$ MATCH (n:Personne {nom: 'DELHOUMI'}) RETURN n`. The results overview panel indicates 3 nodes found. The graph view displays three nodes, each labeled "DELHOUMI". One node is blue, two are yellow. The status bar at the bottom right shows: Started streaming 3 records after 32 ms and completed after 34 ms.

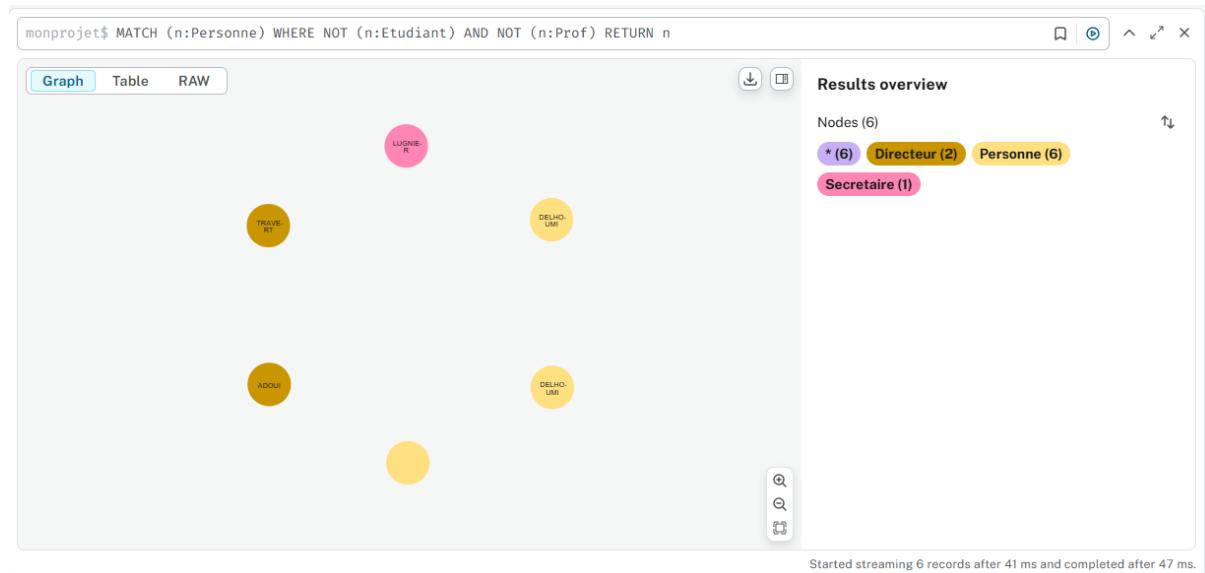
- les profs avec nom Delhoumi

The screenshot shows the Neo4j browser interface. The query entered is: `monprojet$ MATCH (p:Personne:Prof {nom: 'DELHOUMI'}) RETURN p`. The results overview panel indicates 1 node found. The graph view displays one node, labeled "DELHOUMI", which is blue. The status bar at the bottom right shows: Started streaming 1 record after 51 ms and completed after 52 ms.

◦ les profs

```
monprojet$ MATCH (p:Personne:Prof) RETURN p
```

◦ les pas étudiants ou pas profs



◦ les profs dont le nom commence par ZIM

```
monprojet$ MATCH (p:Personne:Prof) WHERE p.nom STARTS WITH 'ZIM' RETURN p
```

◦ les nœuds dont le nom contient un I



◦ le prénom du voleur

```
monprojet$ MATCH (p:Personne {nom: 'VOLEUR'}) RETURN p.prenom
```

Table RAW

p.prenom

: "Alexis"

- le prénom des profs de 59 ans

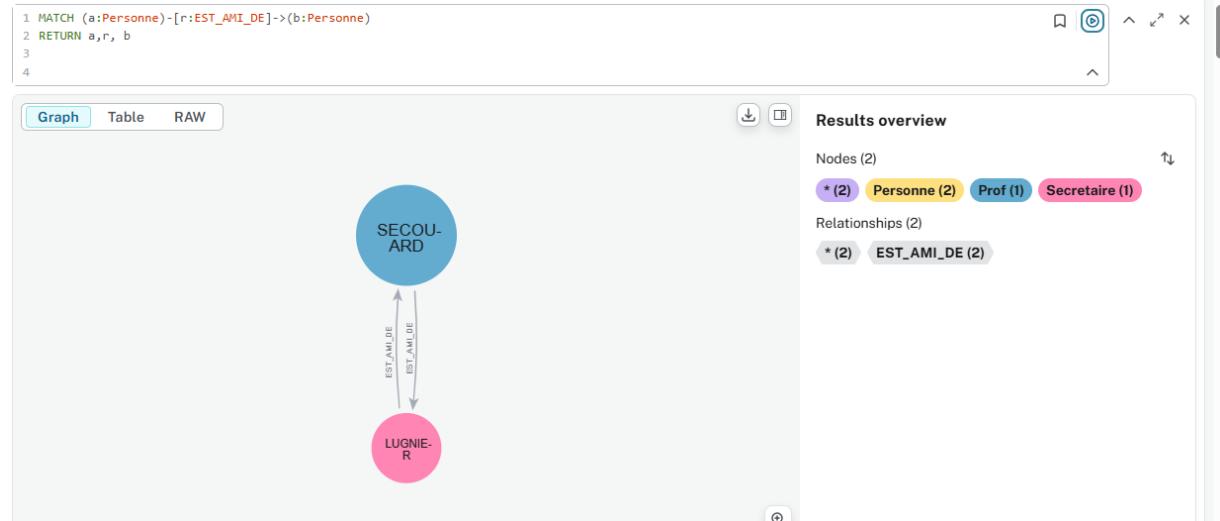
```
monprojet$ MATCH (p:Personne:Prof) WHERE p.age = 59 RETURN p.prenom
```

**Table** RAW

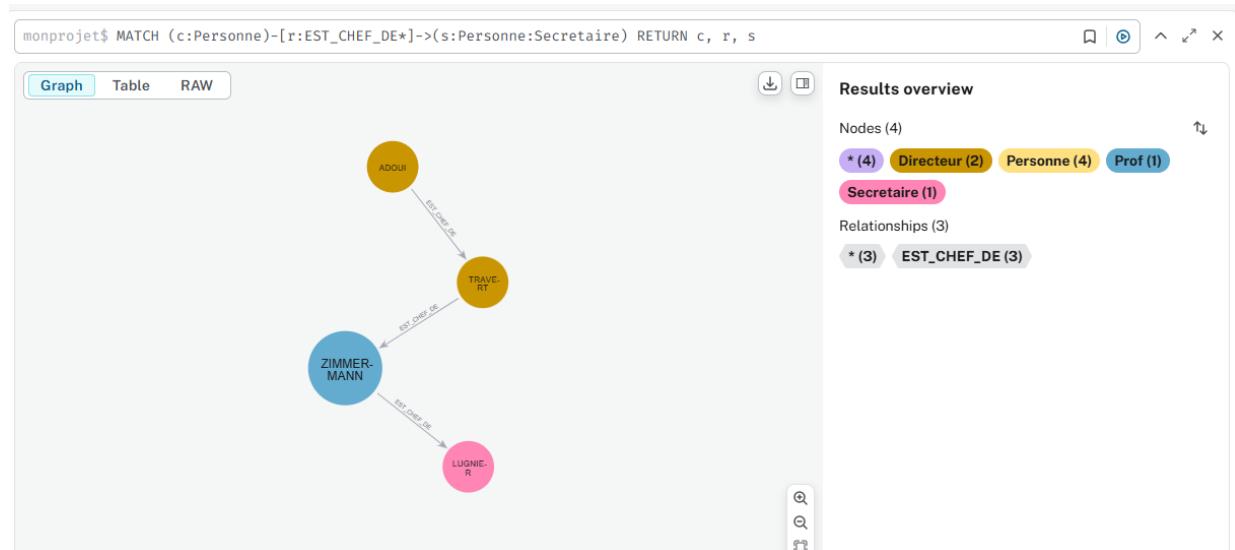
p.prenom

1 "Stéphane"

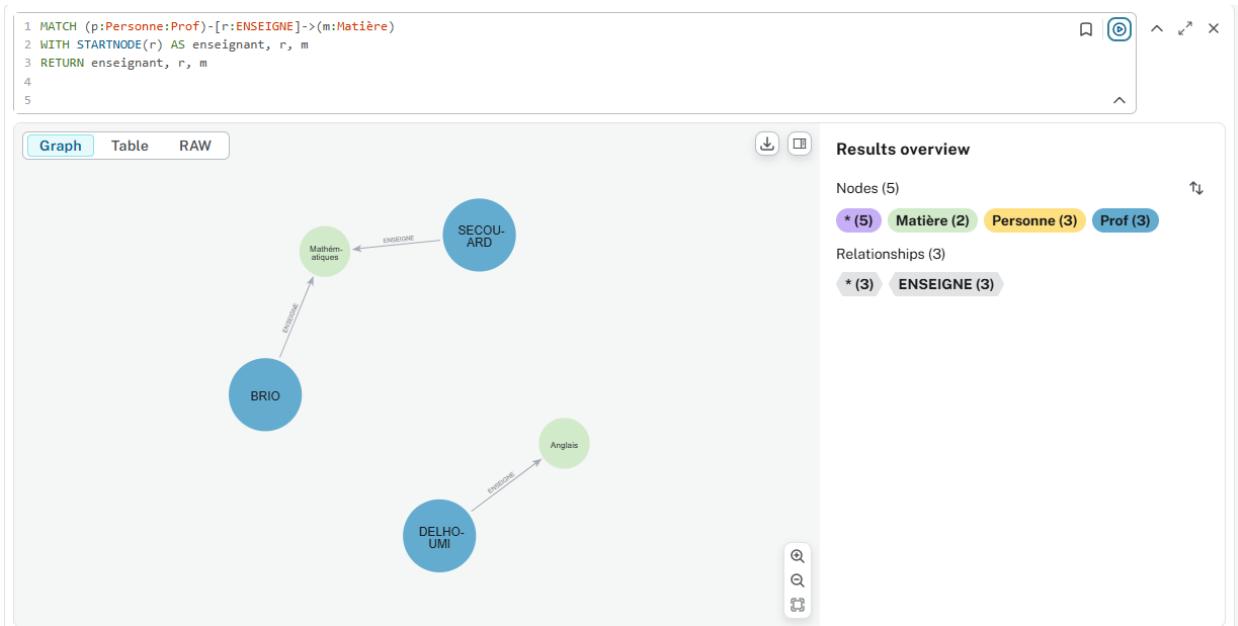
- les amis



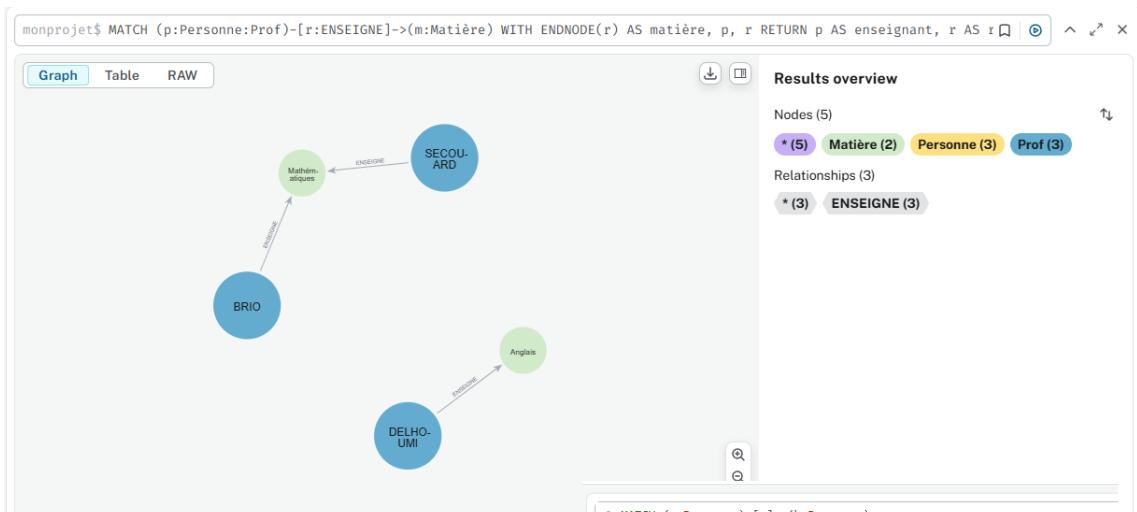
- la branche des chefs jusqu'à la secrétaire



- ceux qui enseignent (utiliser startnode)



◦ ce qui est enseigné (utiliser endnode)



◦ la requête qui donne cela

Qui	qui	De_Qui
1 "ADOUI"	"EST_CHEF"	"TRAVERT"
2 "DELHOUMI"	"EST_COLLEGUE"	"SECOURD"
3 "LUGNIER"	"EST_AMI"	"SECOURD"
4 "SECOURD"	"EST_AMI"	"LUGNIER"
5 "SECOURD"	"EST_COLLEGUE"	"DELHOUMI"
6 "TRAVERT"	"EST_CHEF"	"ZIMMERMANN"
7 "ZIMMERMANN"	"EST_CHEF"	"LUGNIER"

QUI	QUOI	DE QUI
1 "DELHOUMI"	"EST_COLLEGUE_DE"	"SECOURD"
2 "DELHOUMI"	"EST_COLLEGUE_DE"	"SECOURD"
3 "DELHOUMI"	"EST_COLLEGUE_DE"	"SECOURD"
4 "LUGNIER"	"EST_AMI_DE"	"SECOURD"
5 "SECOURD"	"EST_AMI_DE"	"LUGNIER"
6 "SECOURD"	"EST_COLLEGUE_DE"	"DELHOUMI"
7 "SECOURD"	"EST_COLLEGUE_DE"	"DELHOUMI"
8 "SECOURD"	"EST_COLLEGUE_DE"	"DELHOUMI"
9 "ZIMMERMANN"	"EST_CHEF_DE"	"LUGNIER"
10 "TRAVERT"	"EST_CHEF_DE"	"ZIMMERMANN"

- Tout le monde mais préciser quand ils sont amis ou chefs sinon NON (utiliser optional match )

```

1 MATCH (a:Personne)-[r]->(b:Personne)
2 OPTIONAL MATCH (a)-[r_ami:EST_AMI_DE]->(b)
3 OPTIONAL MATCH (a)-[r_chef:EST_CHEF_DE]->(b)
4 RETURN a.nom AS qui,
5     CASE WHEN r_ami IS NOT NULL THEN 'EST_AMI_DE'
6         WHEN r_chef IS NOT NULL THEN 'EST_CHEF_DE'
7         ELSE 'NON'
8     END AS quoi,
9     CASE WHEN r_ami IS NOT NULL OR r_chef IS NOT NULL THEN b.nom ELSE NULL END AS de_qui
10
11

```

Qui	quoi	De_Qui	
	qui	quoi	de_qui
1 "DELHOUMI"	"NON"	null	
2 "VOLEUR"	"NON"	null	
3 "TRICHEUR"	"NON"	null	
4 "ESTJAUNE"	"NON"	null	
5 "LUGNIER"	"EST_AMI"	"SECOURD"	
6 null	"NON"	null	
7 "SECOURD"	"EST_AMI"	"LUGNIER"	
8 "ESTJAUNE"	"EST_CHEF"	"LUGNIER"	
9 "ZIMMERMANN"	"EST_CHEF_DE"	"LUGNIER"	
10 "TDAVIDDT"	"EST_CHEF_DE"	"ZIMMERMANN"	

- Corriger des informations

- Supprimer les Delhoumi qui ne sont pas profs

```

1 MATCH (p:Personne {nom: 'DELHOUMI'})
2 RETURN p;
3

```

```

1 MATCH (p:Personne {nom: 'DELHOUMI'})
2 WHERE NOT 'Prof' IN labels(p)
3 DETACH DELETE p;
4

```

✓ Deleted 2 nodes, deleted 4 relationships

- Secouard n'enseigne plus avec Brio. Supprimer la liaison.

```

1 MATCH (p:Prof {nom: 'BARIO'})-[r:ENSEIGNE]->(m:Matière {nom: 'Mathématiques'})
2 DELETE r;
3

```

✓ Deleted 1 relationship

Completed after 107 ms

```
monprojet$ MATCH (p:Prof {nom: 'BARIO'})-[r]->(m) RETURN p, r, m;
```

Graph Table RAW



Results overview

Nodes (2)

\* (2) Matière (1) Personne (1) Prof (1)

Relationships (1)

\* (1) ENSEIGNE (1)



- Nommer le nœud sans nom Inconnu, prenom = Durin, age = 40

```

1 MATCH (p:Personne)
2 WHERE p.nom IS NULL
3 SET p.nom = 'Inconnu',
4     p.prenom = 'Durin',
5     p.age = 40;
6

```

```
monprojet$ MATCH (p:Personne) WHERE p.nom IS NULL RETURN p;
```

- Supprimer le code postal du Voleur

```
monprojet$ MATCH (p:Personne {nom: 'VOLEUR'}) REMOVE p.code_postal;
```

✓ Set 1 property

Completed after 77 ms

```
monprojet$ MATCH (p:Personne {nom: 'VOLEUR'}) RETURN p;
```

Graph Table RAW



Node details

Etudiant  Personne



Key Value

<id>	4:d3aa8232-a20e-4057-8b4d-589f339c191a:4
age	31
nom	"VOLEUR"
notes	[12.0, 13.5, 2.0]
prenom	"Alexis"
ville	"CAEN"

- Retirer le prénom de Inconnu et corriger son nom pour le nommer ALAPHILIPPE

```
monprojet$
```

```
monprojet$ MATCH (p:Personne {nom: 'ALAPHILIPPE'}) RETURN p;
```

Graph Table RAW



Results overview

Nodes (1)

\* (1) Personne (1)



Started streaming 1 record after 31 ms and completed after 33 ms.

```
monprojet$ MATCH (p:Personne {nom: 'Inconnu'}) REMOVE p.prenom SET p.nom = 'ALAPHILIPPE';
```

✓ Set 2 properties

Completed after 45 ms

- En utilisant Merge, créer une personne, prof nommée nom:JORT, prenom:Fabienne, age 60

```
monprojet$ MERGE (p:Prof {nom: 'JORT', prenom: 'Fabienne'}) ON CREATE SET p.age = 60;
```

✓ Created 1 node, set 3 properties, added 1 label

Completed after 130 ms

- En utilisant Merge, corriger son age. La boulette, elle n'a que 59 ans.

```

1 MERGE (p:Prof {nom: 'JORT', prenom: 'Fabienne'})
2 SET p.age = 59;
3

```

Set 1 property

Completed after 44 ms

monprojet\$ MATCH (p:Prof {nom: 'JORT'}) RETURN p;

Graph Table RAW

Node details

Prof

Key	Value
<id>	4:d3aa8232-a20e-4057-8b4d-589f339c191a:16
age	59
nom	"JORT"
prenom	"Fabienne"

- Fonctions d'agrégations

- Compter le nombre de nœuds

```

monprojet$ MATCH (n) RETURN count(n) AS nombre_de_noeuds;

```

Table RAW

nombre\_de\_noeuds

17
----

Started streaming 1 record after 29 ms and completed after 30 ms.

- Compter le nombre de nœuds par catégorie ou groupe de catégories

```

monprojet$ MATCH (n) RETURN labels(n) AS categories, count(*) AS nombre ORDER BY nombre DESC;

```

Table RAW

categories

categories	nombre
<sup>1</sup> ["Personne", "Prof"]	8
<sup>2</sup> ["Personne", "Etudiant"]	3
<sup>3</sup> ["Personne", "Directeur"]	2
<sup>4</sup> ["Matière"]	2
<sup>5</sup> ["Personne"]	1
<sup>6</sup> ["Personne", "Secrétaire"]	1

Started streaming 6 records after 1 ms and completed after 3 ms.

- Compter le nombre de nœuds par age trié du plus jeune au plus vieux

```
monprojet$ MATCH (p:Personne) RETURN p.age AS age, count(*) AS nombre ORDER BY age ASC;
```

**Table** **RAW**

age	nombre
28	1
30	1
31	1
36	1
39	1
40	1
45	1
49	1
50	1
52	1

Started streaming 13 records after 31 ms and completed after 32 ms.

- Afficher l'âge le plus petit, moyen et le plus grand

```
1 MATCH (p:Personne)
2 RETURN
3   min(p.age) AS age_min,
4   round(avg(p.age),2) AS age_moyen,
5   max(p.age) AS age_max;
6
```

**Table** **RAW**

age_min	age_moyen	age_max
28	44.64	59

Started streaming 1 record after 49 ms and completed after 50 ms.

- Afficher par personne, leur diverses relations

```
1 MATCH (p:Personne)-[r]->(c)
2 RETURN p.nom AS personne, collect(type(r)) AS relations, collect(c.nom) AS vers;
3
```

**Table** **RAW**

personne	relations	vers
"DELHOUUMI"	["ENSEIGNE", "EST_COLLEGE_DE"]	["Anglais", "SECOURD"]
"LUGNIER"	["EST_AMI_DE"]	["SECOURD"]
"SECOURD"	["ENSEIGNE", "EST_AMI_DE", "EST_COLLEGE_DE"]	["Mathématiques", "LUGNIER", "DELHOUUMI"]
"ZIMMERMANN"	["EST_CHEF_DE"]	["LUGNIER"]
"TRAVERT"	["EST_CHEF_DE"]	["ZIMMERMANN"]
"ADOUI"	["EST_CHEF_DE"]	["TRAVERT"]

Started streaming 6 records after 52 ms and completed after 53 ms.

- Afficher par relation, les diverses personnes

```
monprojet$ MATCH (p:Personne)-[r]->() RETURN type(r) AS relation, collect(p.nom) AS personnes;
```

[Table](#) [RAW](#)

relation	personnes
<sup>1</sup> "ENSEIGNE"	["DELHOUMI", "SECOURD"]
<sup>2</sup> "EST_COLLEGUE_D E"	
<sup>3</sup> "EST_AMI_DE"	["LUGNIER", "SECOURD"]
<sup>4</sup> "EST_CHEF_DE"	["ZIMMERMANN", "TRAVERT", "ADOUUI"]

Started stream

- Afficher les personnes la plus jeune et la plus vielle

```
1 // Récupérer les personnes les plus jeunes et les plus âgées
2 MATCH (p:Personne)
3 WITH
4     max(p.age) AS age_vieux,
5     min(p.age) AS age_jeune
6
7 MATCH (vieux:Personne {age: age_vieux})
8 MATCH (jeune:Personne {age: age_jeune})
9 RETURN
10    vieux.nom AS nom_vieux,
11    age_vieux AS age_vieux,
12    jeune.nom AS nom_jeune,
13    age_jeune AS age_jeune;
14
```

[Table](#) [RAW](#)

nom_vieux	age_vieux	nom_jeune	age_jeune
<sup>1</sup> "SECOURD"	59	"PORCQ"	28
<sup>2</sup> "JORT"	59	"PORCQ"	28

- oui mais comme cela

```

1 MATCH (p:Personne)
2 WITH max(p.age) AS age_vieux, min(p.age) AS age_jeune
3
4 // Personnes les plus vieilles
5 MATCH (vieux:Personne {age: age_vieux})
6 WITH collect(DISTINCT {type: 'plus_vieux', nom: vieux.nom, age: age_vieux}) AS lignes_vieux, age_jeune
7
8 // Personnes les plus jeunes
9 MATCH (jeune:Personne {age: age_jeune})
10 WITH lignes_vieux, collect(DISTINCT {type: 'plus_jeune', nom: jeune.nom, age: age_jeune}) AS lignes_jeune
11
12 // Combiner les deux listes
13 WITH lignes_vieux + lignes_jeune AS personnes
14 UNWIND personnes AS ligne
15 RETURN ligne.type AS type, ligne.nom AS nom, ligne.age AS age;
16

```

Table RAW

type	nom	age
<sup>1</sup> "plus_vieux"	"SECOUARD"	59
<sup>2</sup> "plus_vieux"	"JORT"	59
<sup>3</sup> "plus_jeune"	"PORCQ"	28

Started streaming 3 records after 125 ms and completed after 128 ms.

- la moyenne des notes du tricheur

```

1 MATCH (e:Etudiant {nom: 'VOLEUR'})
2 UNWIND e.notes AS note
3 RETURN round(avg(note),2) AS moyenne_notes;
4

```

Table RAW

moyenne_notes
<sup>1</sup> 9.17

Started streaming 1 record after 40 ms and completed after 41 ms.

- Fonctions ensemblistes

- En utilisant une fonction ensembliste, donner la requête qui les deux plus vieux et plus jeunes

```

1 // Deux plus vieux
2 MATCH (p:Personne)
3 WHERE p.age IS NOT NULL
4 WITH p
5 ORDER BY p.age DESC
6 LIMIT 2
7 RETURN 'plus_vieux' AS type, p.nom AS nom, p.age AS age
8
9 UNION
10
11 // Deux plus jeunes
12 MATCH (p:Personne)
13 WHERE p.age IS NOT NULL
14 WITH p
15 ORDER BY p.age ASC
16 LIMIT 2
17 RETURN 'plus_jeune' AS type, p.nom AS nom, p.age AS age;
18

```

Table RAW

type	nom	age
<sup>1</sup> "plus_vieux"	"SECOUARD"	59
<sup>2</sup> "plus_vieux"	"JORT"	59
<sup>3</sup> "plus_jeune"	"PORCQ"	28
<sup>4</sup> "plus_jeune"	"LUGNIER"	30

Started streaming 4 records after 173 ms and completed after 178 ms.

## Exercice 2

Pour cet exercice, aucun avertissement produit cartésien ne devra se produire

- Se connecter à une base
- Tout effacer

```
monprojet$ MATCH (n) DETACH DELETE n;
```

Deleted 4 nodes

Completed in less than 1ms

- Créer des noeuds Personnes (:Prof:Personne {nom: "SECOUARD", prenom: "Stéphane"}) (:Prof:Personne {nom: "VAGINAY", prenom: "Athénaïs"}) (:Prof:Personne {nom: "JORT", prenom: "Fabienne"}) (:Prof:Personne {nom: "DELHOUMI", prenom: "Sylvian"})
- Affiche tout

Database information

Nodes (4)

- Personne
- Prof

Relationships (0)

Property keys

- age
- code\_postal
- data
- id
- name
- nodes
- nom
- notes
- prenom
- relationships
- style
- ville
- visualisation

monprojet\$

```
monprojet$ MATCH (p:Personne) RETURN p;
```

Graph Table RAW

Node details

Personne Prof

Key	Value
<id>	4d3aa8232-e20e-4057-8b4d-5891339c191a7
nom	"SECOUARD"
prenom	"Stéphane"

Started streaming 4 records after 7 ms and completed after 8 ms.

```
1 CREATE
2   (:Prof:Personne {nom: "SECOUARD", prenom: "Stéphane"}),
3   (:Prof:Personne {nom: "VAGINAY", prenom: "Athénaïs"}),
4   (:Prof:Personne {nom: "JORT", prenom: "Fabienne"}),
5   (:Prof:Personne {nom: "DELHOUMI", prenom: "Sylvian"});
```

Created 4 nodes, set 8 properties, added 8 labels

Completed after 3 ms

- Visualiser le nœud Secouard et manipuler un peu l'interface

```
monprojet$ MATCH (vous:Personne {nom : "SECOUARD"}) return vous;
```

Graph Table RAW

Node details

Personne Prof

Customize Style

Color:

Size:

Caption:

nom prenom <id> <type>

Started streaming 1 record after 67 ms and completed after 69 ms.

- Créer des noeuds Endroit :
 

```
(:Endroit {nom: "RU"})
(:Endroit {nom: "DOMICILE"})
(:Endroit {nom: "BUREAU"})
(:Endroit {nom: "SALLE DES PERSONNELS"})
```

The screenshot shows the Neo4j Browser interface. At the top, a query is entered: `monprojet$ MATCH (e:Endroit) return e;`. Below the query, there are three tabs: Graph, Table, and RAW. The Graph tab displays four blue circular nodes arranged in a square pattern. The top-left node is labeled 'BUREAU', the top-right 'DOMICILE', the bottom-left 'SALLE DES PERSONNELS', and the bottom-right 'RU'. To the right of the graph, a 'Results overview' panel shows 'Nodes (4)' and a link to 'Endroit (4)'. A message at the bottom of the browser window states: 'Started streaming 4 records after 22 ms and completed after 23 ms.'

```
monprojet$ MATCH (e:Endroit) return e;
```

Graph Table RAW

BUREAU  
DOMICILE  
SALLE DES PERSONNELS  
RU

Results overview  
Nodes (4)  
\* (4) Endroit (4)

Started streaming 4 records after 22 ms and completed after 23 ms.

```
1 CREATE
2   (:Endroit {nom: "RU"}),
3   (:Endroit {nom: "DOMICILE"}),
4   (:Endroit {nom: "BUREAU"}),
5   (:Endroit {nom: "SALLE DES PERSONNELS"});
6
```

Created 4 nodes, set 4 properties, added 4 labels

Completed after 41 ms

- Création d'un nœud Lieu\_Travail avec comme nom Campus 3

The screenshot shows the Neo4j Browser interface. At the top, a query is entered: `monprojet$ // Vérifier la création MATCH (l:Lieu_Travail) RETURN l;`. Below the query, there are three tabs: Graph, Table, and RAW. The Graph tab displays a single orange circular node labeled 'Campus 3'. To the right of the graph, a 'Results overview' panel shows 'Nodes (1)' and a link to 'Lieu\_Travail (1)'. A message at the bottom of the browser window states: 'Started streaming 1 record after 89 ms and completed after 89 ms.'

```
monprojet$ // Vérifier la création MATCH (l:Lieu_Travail) RETURN l;
```

Graph Table RAW

Campus 3

Results overview  
Nodes (1)  
\* (1) Lieu\_Travail (1)

Started streaming 1 record after 89 ms and completed after 89 ms.

```
monprojet$ CREATE (:Lieu_Travail {nom: "Campus 3"});
```

CREATE (:Lieu\_Travail {nom: "Campus 3"});

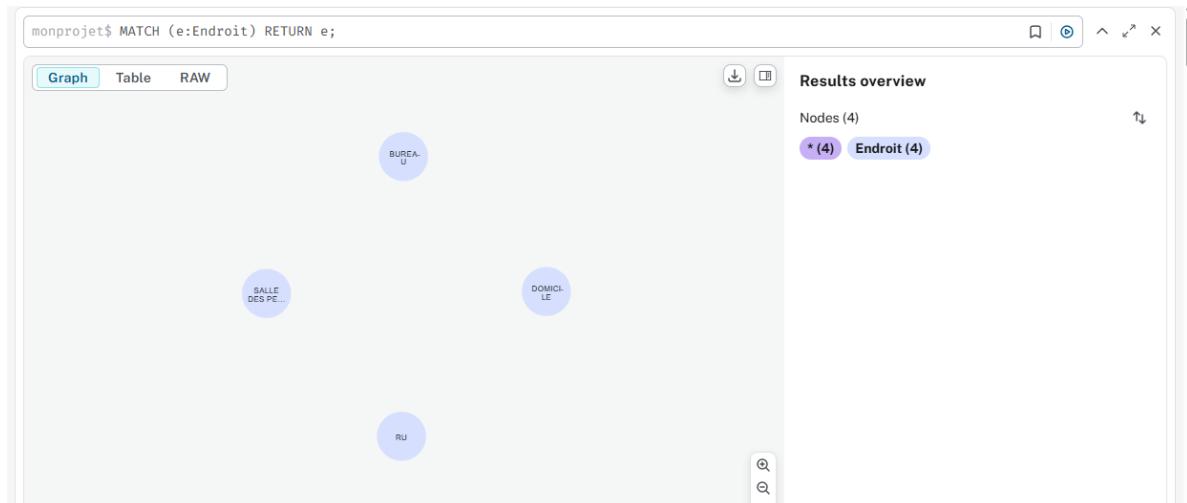
## Compléter les noeuds

```
(: Endroit {ville: "IFS", nom: "RU"})
(: Endroit {ville: "CAEN", nom: "DOMICILE"})
(: Endroit {ville: "TROARN", nom: "BUREAU"})
(: Endroit {ville: "IFS", nom: "SALLE DES PERSONNELS"})
```

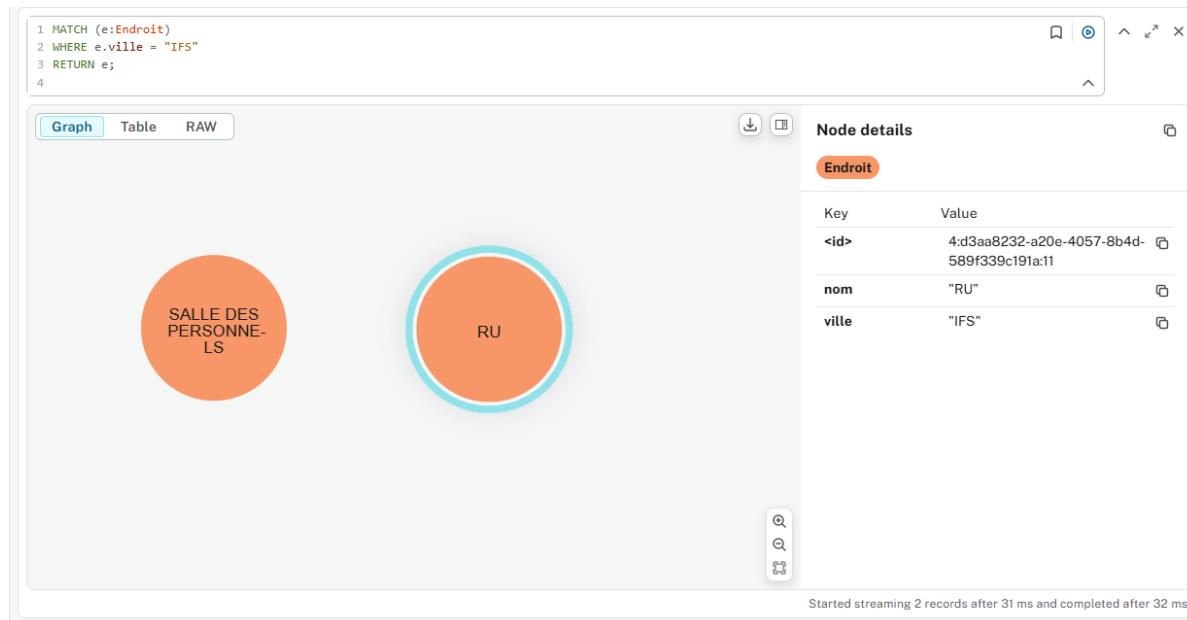


```
1 MATCH (e:Endroit {nom: "RU"})
2 SET e.ville = "IFS";
3
4 MATCH (e:Endroit {nom: "DOMICILE"})
5 SET e.ville = "CAEN";
6
7 MATCH (e:Endroit {nom: "BUREAU"})
8 SET e.ville = "TROARN";
9
10 MATCH (e:Endroit {nom: "SALLE DES PERSONNELS"})
11 SET e.ville = "IFS";
12 |
```

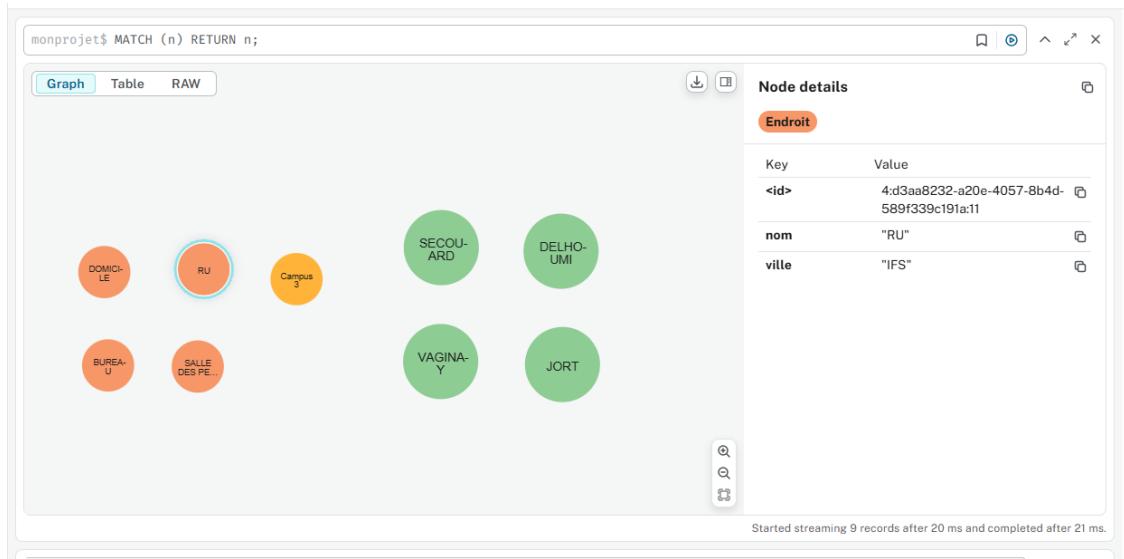
- Visualiser les endroits



- Visualiser les endroits à IFS



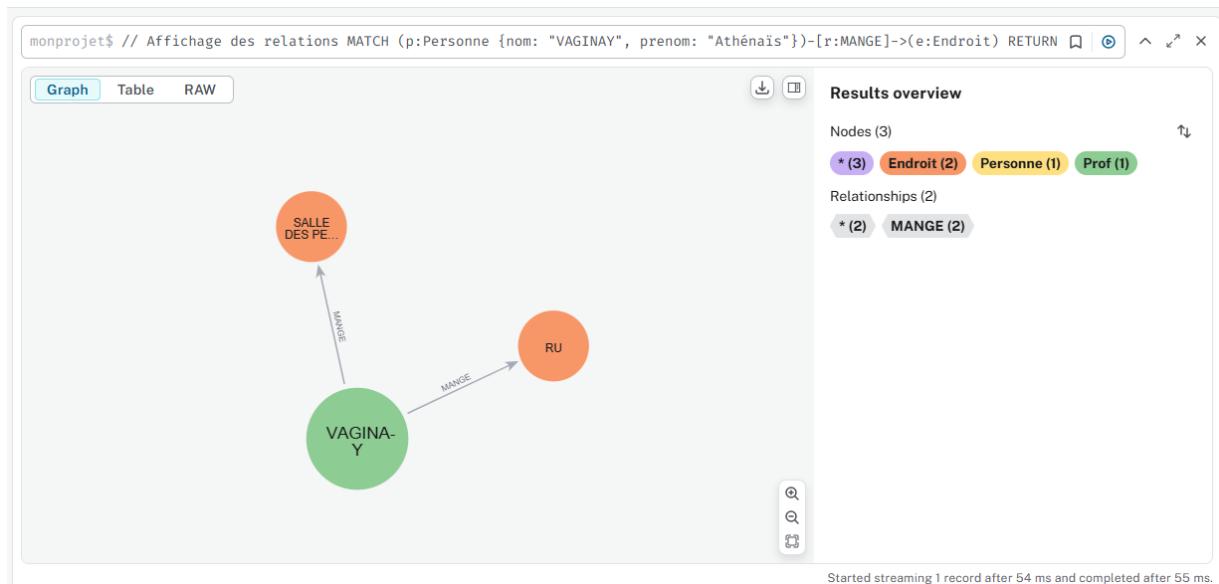
- Afficher tout



- Stéphane Secouard mange au RU

```
monprojet$ // Affichage des relations
1 MATCH (p:Personne {nom: "SECOURARD", prenom: "Stéphane"})
2 MATCH (e:Endroit {nom: "RU"})
3 CREATE (p)-[:MANGE]->(e)
4 RETURN p, e;
q
```

- Athénaïs Vaginay- mange au RU ou en Salle des Personnels (une seule requête)

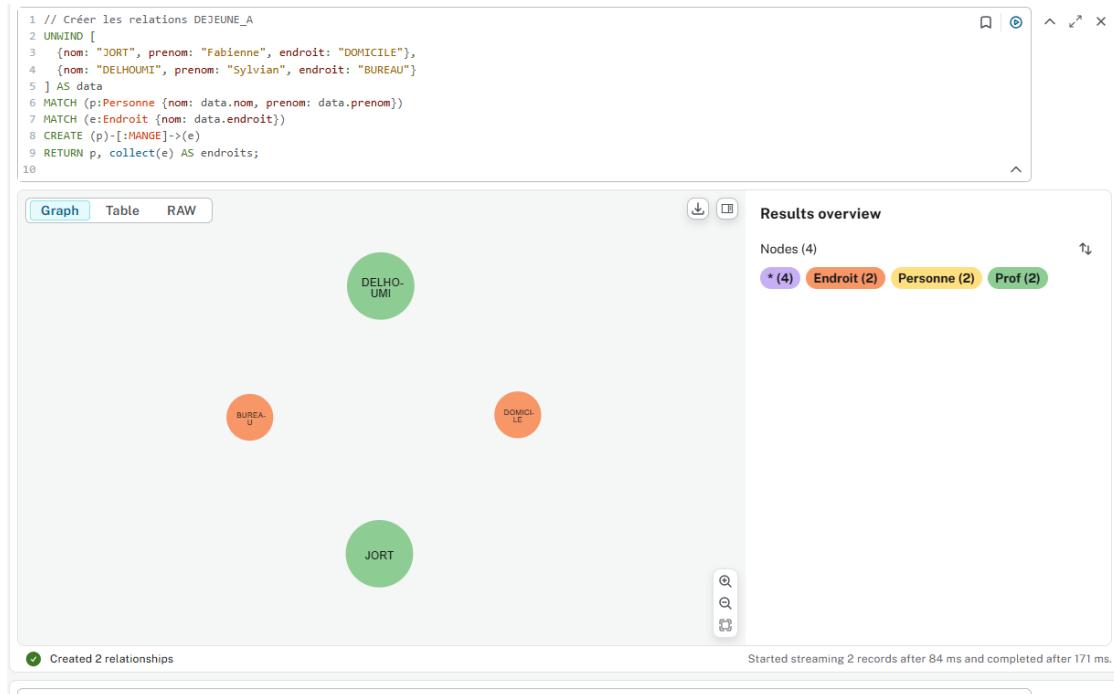


```
monprojet$ // Création des relations
1 MATCH (p:Personne {nom: "VAGINAY", prenom: "Athénaïs"})
2 MATCH (e:Endroit)
3 WHERE e.nom IN ["RU", "SALLE DES PERSONNELS"]
4 CREATE (p)-[:MANGE]->(e);
5
6
```

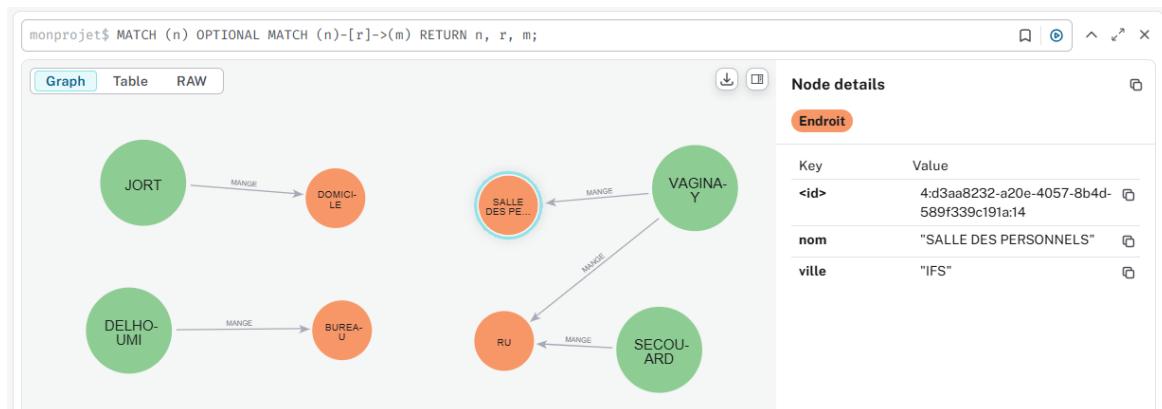
Created 2 relationships

Completed after 134 ms

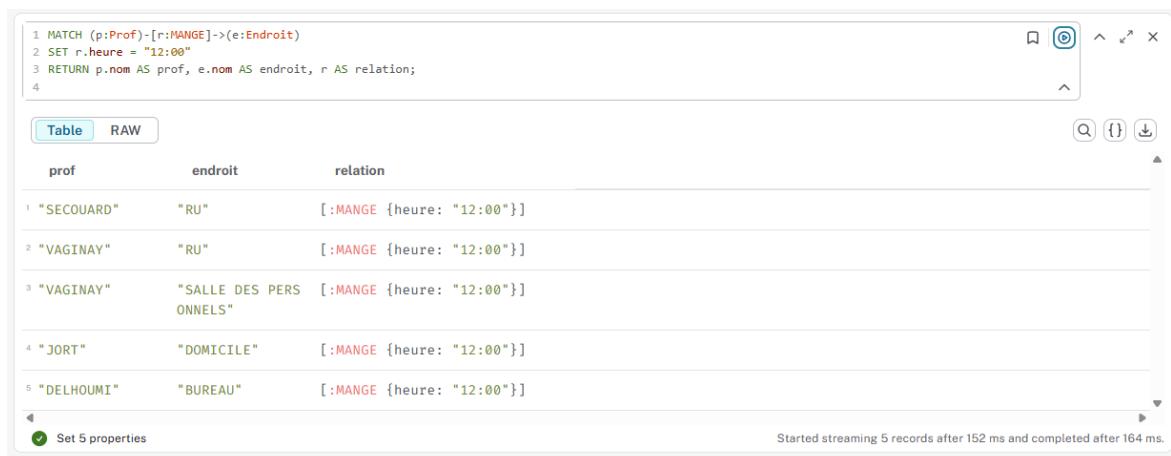
- F.Jort déjeune à son domicile alors que S.Delhoumi déjeune dans son bureau (une seule requête)



- Afficher tout



- Les profs mangent à 12:00



- Mais pas Stéphane Secouard qui mange à 11:50

```

1 // Mettre à jour l'heure de Stéphane Secouard à 11:50
2 MATCH (p:Prof {nom: "SECOUARD"})-[r:MANGE]->(e:Endroit)
3 SET r.heure = "11:50"
4 RETURN p.nom AS prof, e.nom AS endroit, r.heure AS heure;
5

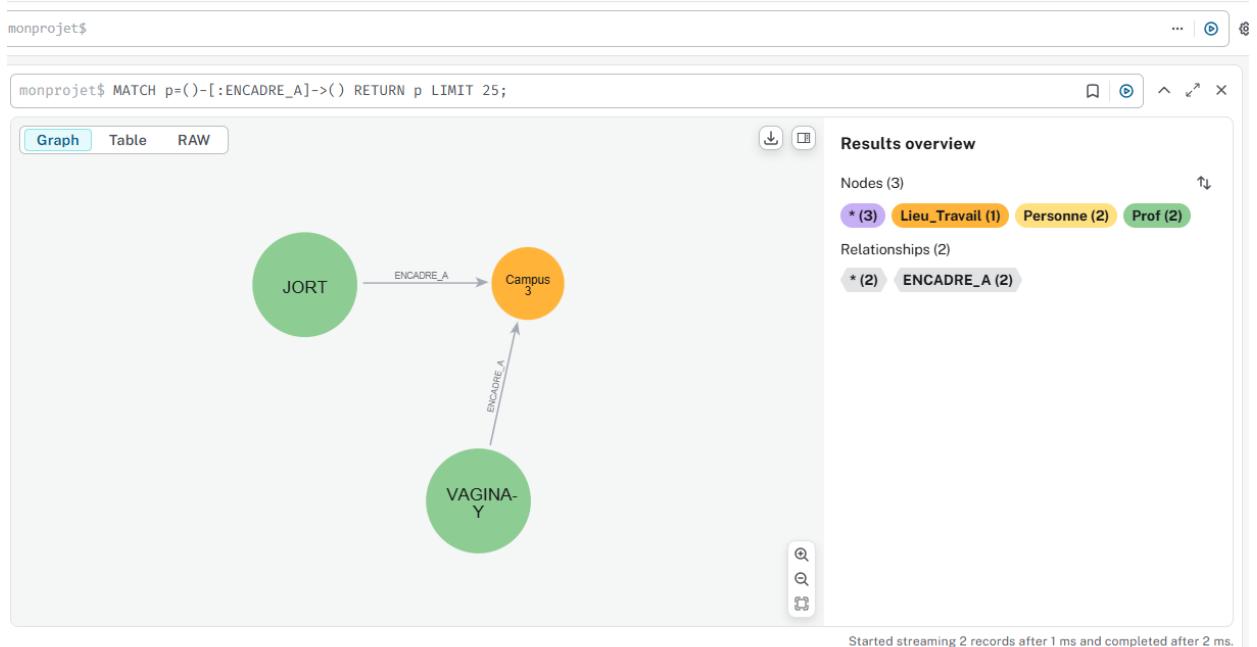
```

Table RAW

prof	endroit	heure
"SECOUARD"	"RU"	"11:50"

Set 1 property Started streaming 1 record after 70 ms and completed after 72 ms.

- F.Jort et A.Vaginay encadrent au campus 3
  - F.Jort occupe la fonction : Responsable des stages
  - A.Vaginay occupe la fonction : Directrice des études (faire un profil)



```

1 // Créer le nœud Lieu_Travail : Campus 3
2 CREATE (:Lieu_Travail {nom: "Campus 3"});
3
4 // Créer les relations d'encadrement avec la fonction des profs
5 MATCH (f:Personne {nom: "JORT", prenom: "Fabienne"}), (a:Personne {nom: "VAGINAY", prenom: "Athénais"}), (l:Lieu_Travail {nom: "Campus 3"})
6 CREATE
7 (f)-[r:ENCADRE_A {fonction: "Responsable_des_stages"}]->(l),
8 (a)-[r:ENCADRE_A {fonction: "Directrice_des études"}]->(l)
9 RETURN f, a, l;
10

```

- S.Secouard et S.Delhoumi encadrent au campus 3

```

1 // Créer les relations d'encadrement avec Campus 3
2 MATCH (s:Personne {nom: "SECOUARD", prenom: "Stéphane"}), (d:Personne {nom: "DELHOUMI", prenom: "Sylvian"}), (l:Lieu_Travail {nom: "Campus 3"})
3 CREATE
4 (s)-[r:ENCADRE_A]->(l),
5 (d)-[r:ENCADRE_A]->(l)
6 RETURN s.nom AS secouard, d.nom AS delhoumi, l.nom AS lieu;
7

```

Table RAW

secouard	delhoumi	lieu
"SECOUARD"	"DELHOUMI"	"Campus 3"

Created 2 relationships 2 info messages Started streaming 1 record after 191 ms and completed after 194 ms.

- Ajout d'une propriété à une relation
  - S.Secouard occupe la fonction : Responsable du recrutement
  - S.Delhoumi occupe la fonction : Responsable des vacataires

```

1 // Ajouter la fonction "Responsable du recrutement" à la relation de Secouard
2 MATCH (s:Personne {nom: "SECOURD", prenom: "Stéphane"})-[r:ENCADRE_A]->(l:Lieu_Travail {nom: "Campus 3"})
3 SET r.fonction = "Responsable du recrutement"
4 RETURN s.nom AS prof, r.fonction AS fonction, l.nom AS lieu;
5
6 // Ajouter la fonction "Responsable des vacataires" à la relation de Delhoumi
7 MATCH (d:Personne {nom: "DELHOUUMI", prenom: "Sylvian"})-[r:ENCADRE_A]->(l:Lieu_Travail {nom: "Campus 3"})
8 SET r.fonction = "Responsable des vacataires"
9 RETURN d.nom AS prof, r.fonction AS fonction, l.nom AS lieu;
3 |

```

- Afficher tout

