



Présentation de NE04J

Identifiant Noeuds et Relations

Node Pattern

Relation Pattern

Commandes :

- CALL
- FOREACH
- MERGE
- WITH
- UNION
- SET
- REMOVE
- DELETE
- LIMIT & SKIP & ORDER BY

Identifiant Nœuds & Relations

Chaque nœud et relation d'un graphe est identifié par:

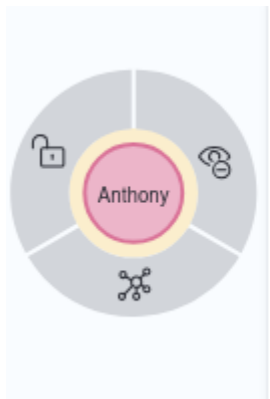
- un id (déprécié en version 5)
- un **elementid**

```
MATCH (s) WHERE ID(s) = 27 RETURN s;
```

```
MATCH (s)  
WHERE elementID(s) = "4:eb34a0e8-5c91-45c0-93b7-97d63e095dd3:27"  
RETURN s;
```

Attention :

Ne pas utiliser cet identifiant dans les requêtes.
Utilisation en interne de Neo4J



Node properties ⓘ

Acteur Personne

<elementId>	4:eb34a0e8-5c91-45c0-93b7-97d63e095dd3:27	ⓘ
<id>	27	ⓘ
Nom	Hopkins	ⓘ
Prenom	Anthony	ⓘ



Node Pattern : Jeux de test

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

```
CREATE (charlie:Personne:Acteur {Prenom: 'Charlie', Nom: 'Sheen'}),  
(oliver:Personne:Realisateur {Prenom: 'Oliver', Nom: 'Stone'});
```

```
MATCH (a:Acteur) WHERE a.Nom="Sheen"  
CREATE (a) -[:JoueDans {role: 'Bud Fox'}]->(WallStreet:Film {titre: 'WallStreet'});  
MATCH (a:Realisateur), (b:Film)  
WHERE a.Nom="Stone" AND b.titre ="WallStreet"  
CREATE (a)-[r:Realise]-> (b);
```

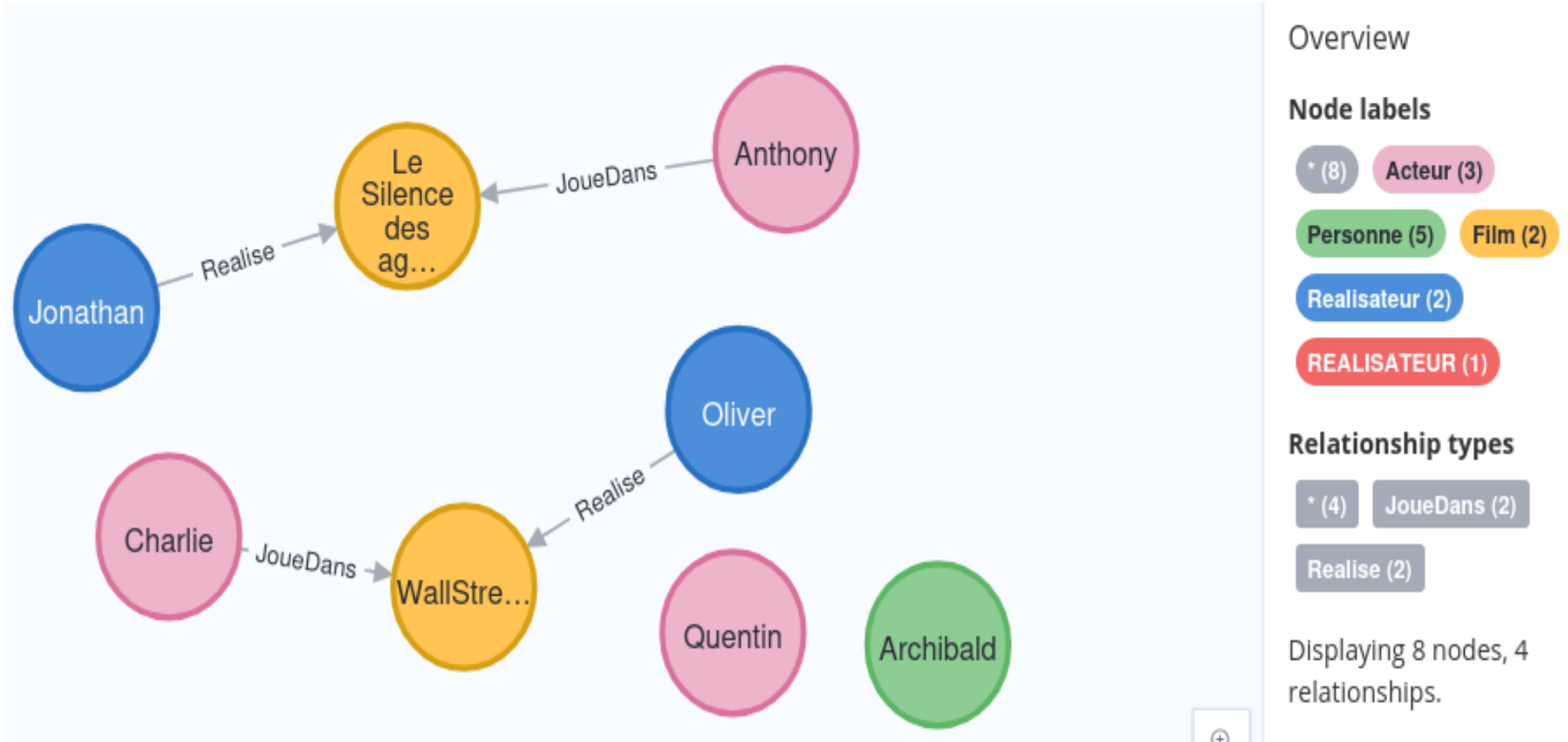
```
CREATE (a:Personne:Acteur {Prenom:'Anthony',Nom:'Hopkins'})-[r:JoueDans {role:  
'Hannibal Lecter'}]->(SilenceDesAgneaux:Film {titre: 'Le Silence des agneaux'})<-  
[:Realise]-(j:Personne:Realisateur {Prenom: 'Jonathan', Nom:'Demme'})  
RETURN a,r,j;
```

```
CREATE (a:Acteur:REALISATEUR {Prenom:"Quentin",Nom:"Tarantino"}) RETURN a;
```

```
CREATE (a:Personne {Prenom:"Archibald",Nom: "Haddock"}) RETURN a;
```

```
MATCH (n) OPTIONAL MATCH (n)-[r]-(m) RETURN n, r,m;
```

Node Pattern : Jeu de test





Node Pattern

(n:Acteur): Noeud avec le label Acteur

(n:Personne&Acteur): Noeud avec les labels Personne **ET** Acteur

(n:Film|REALISATEUR): Noeud avec le label Film **OU** REALISATEUR **OU** les deux

(n:!Acteur): Noeud sans le label Acteur.

(n:%): Noeud avec au moins un label.

(n:(!Acteur&!Realisateur)): Noeud sans les labels Personne et Realisateur

(n:Personne {Nom: "Haddock"}): Noeud ayant pour Nom "Haddock"

(n:Personne WHERE n.Prenom **starts with** 'A'): Noeud avec nom débutant par A

(n:Personne WHERE n.Prenom **ends with** 'y'):Noeud avec nom terminant par y

(n:Personne WHERE n.Prenom **contains** 'ha'):Noeud avec nom terminant par y

<https://neo4j.com/docs/cypher-manual/current/patterns/fixed-length-patterns/#node-patterns>



Relation Pattern

`(n:Acteur) - [r:JoueDans] -> (m:Film)`

`(n:Personne) - [r:!JoueDans] -> (m:Film):` Ne participe à la relation JoueDans.

`(n:Personne) - [r:!JoueDans|Realise] -> (m:Film):` Ne participe à la relation JoueDans **OU** participe à la relation Realise.

`(n:Personne) - [r:(!FRIEND_OF&!LIKES) | KNOWS] -> (m:Personne):` n en relation avec m en étant ni ami ni apprécié mais connu

`(n:Person) - [r:KNOWS WHERE r.metIn STARTS WITH 'Sto']] -> (m:Person):` n connaît m ...

<https://neo4j.com/docs/cypher-manual/current/patterns/fixed-length-patterns/#relationship-patterns>



CALL

Appel d'une procédure stockée en base

Exemple

Affiche tous les labels définis dans la base
`call db.labels();`

Affiche la structure du graphe
`call db.schema.visualization();`

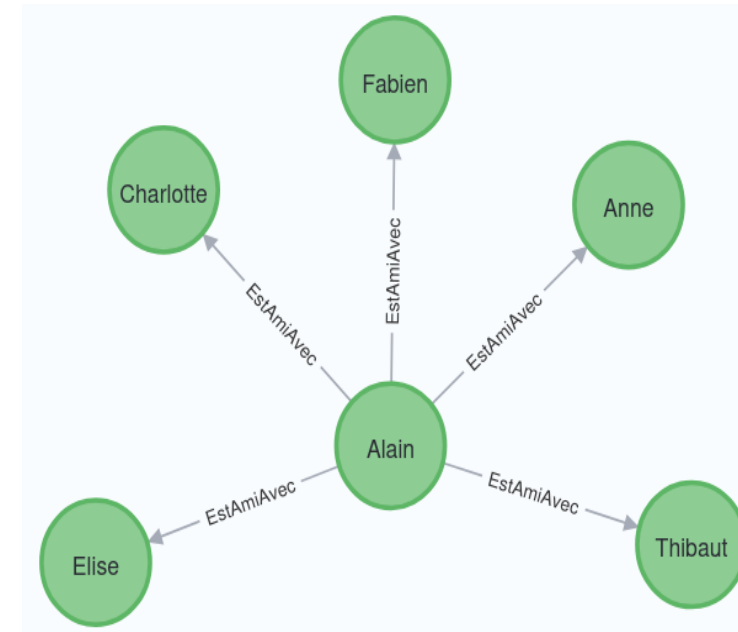
<https://neo4j.com/docs/cypher-manual/current/clauses/call/>

La boucle FOREACH permet de parcourir une collection de nœuds

Exemple

```
create (a:Personne {Prenom: "Alain"}) return a;
match (a:Personne { Prenom:"Alain" })
foreach (prenom in ["Thibaut","Anne","Fabien","Charlotte","Elise"])
|
  CREATE (a)-[:EstAmiAvec]->(:Personne { Prenom: prenom })
)
match (n)-[r:EstAmiAvec]->(m)
return n,r,m;
```

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





MERGE

Si le nœud existe déjà alors il est mis à jour
Si le nœud n'existe pas alors il est créé.

Exemple

`MERGE (n:Personne {nom: "Sylvain"});` <= Création du Noeud

`MERGE (n:Personne {nom: "Sylvain"});` <= Aucune action

`MERGE (n:Personne {nom: "Sylvain"})`

`ON MATCH SET n.dateModification = timestamp();` <= Maj du Noeud

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



WITH

La syntaxe WITH est similaire à celle de RETURN.

Il sépare explicitement les parties de la requête, permettant aux utilisateurs de déclarer les variables à transférer dans la partie suivante de la requête.

Exemple

```
MATCH (p:Person) -[r:ACTED_IN]-> (f:Movie)
WITH p, count (f) as nb_films
ORDER BY nb_films desc
RETURN p.name, nb_films
LIMIT 4;
```

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



UNION

Réalise l'union en deux ensembles

Exemple

```
MATCH (p:Person) -[r:PRODUCED] -> (m:Movie)
RETURN p.name,r,m.title
UNION
MATCH (p:Person) -[r:WROTE] -> (m:Movie)
RETURN p.name,r,m.title;
```

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



SET

MAJ /Ajout d'une propriété d'un nœud ou d'une relation
MAJ/Ajoute un label d'un nœud

Exemple

```
MATCH (f:Film) WHERE f.titre = 'WallStreet'  
SET f.pays = 'USA' return f;  <= Ajout
```

```
MATCH (f:Film) WHERE f.titre = 'WallStreet'  
SET f.pays = "Etats Unis d'Amérique" return f;  <= MAJ
```

```
MATCH (f:Film) WHERE f.titre = 'WallStreet'  
SET f.pays = null return f;  <= Supprime
```

```
MATCH (a {Nom: 'Sheen'}) SET a:Realisateur RETURN a.Nom,  
labels(a) AS labels
```

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

REMOVE

Retire une ou plusieurs propriétés d'un nœud ou d'une relation

Retire un label d'un nœud

Exemple

```
MATCH (f:Movie)
WHERE f.title = 'Cloud Atlas' SET f.pays='USA' return f;
MATCH (f:Movie)
WHERE f.title = 'Cloud Atlas' REMOVE f.pays;
MATCH (p {name: 'Peter'}) SET p = {};

MATCH (n {Nom: 'Sheen'}) REMOVE n:Realisateur;
```

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

DELETE

Supprime un nœud, une relation, tous les objets d'un graphe

Exemple

Supprime un noeud et les relations qui y sont connectées

```
MATCH (n:Personne) WHERE n.Nom = "Stone" DETACH DELETE n;
```

Supprime une relation

```
MATCH (n:Personne)-[r:JoueDans]->(f:Film)
```

```
WHERE n.Nom = "Hopkins"
```

```
AND f.titre = "Le Silence des agneaux" DELETE r;
```

Supprimer tous les nœuds et relations du graphe

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



LIMIT & SKIP & ORDER BY

LIMIT : Limite le nombre de lignes retournées

SKIP : « Passer » les n premières lignes

ORDER BY : Classement

Exemple

```
MATCH (n:Personne)RETURN n.Nom ORDER BY elementID(n);
```

```
MATCH (n:Personne)RETURN n.Nom ORDER BY n.Nom LIMIT 3;
```

```
MATCH (n:Personne)RETURN n.Nom ORDER BY n.Nom DESC LIMIT 3;
```

```
MATCH (n:Personne)RETURN n.Nom ORDER BY n.Nom SKIP 2 LIMIT 2;
```