

3

Évaluer Neo4j : CYPHER

Nous savons désormais ce que fait Neo4j, il nous faut maintenant examiner le comment. Nous allons étudier CYPHER, un langage d'interrogation spécifique à Neo4j. Il y a sans aucun doute un lien de parenté entre CYPHER et SQL : nous retrouverons des notions connues telles que le regroupement, l'agrégation, l'union, etc. CYPHER diffère de SQL par la structure de la requête, car ce langage d'interrogation propose également des moyens dédiés à l'interrogation d'un graphe, moyens permettant de retrouver efficacement les données fortement connectées.

CYPHER permet d'exprimer des modèles de chemins à parcourir au travers des nœuds et des relations pour extraire de l'information ou en modifier. Ce langage utilise le principe des identificateurs nommés pour faire référence à des éléments de la requête. Enfin, CYPHER permet de transformer un graphe en ajoutant/supprimant/modifiant des nœuds, des relations ou des propriétés.

Ce que nous allons voir...

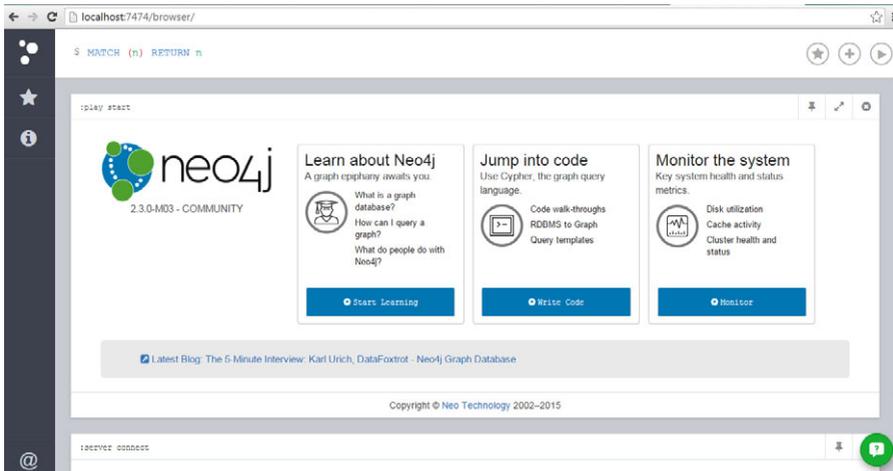
- exprimer un modèle de chemin (*pattern matching*) ;
- choisir un chemin ;
- utiliser les identificateurs ;
- étudier les types de données ;
- utiliser les labels ;
- lire les données et la structure du graphe ;
- modifier les données et la structure du graphe ;
- poser des contraintes d'unicité ;
- manipuler l'indexation depuis CYPHER.

3.1. Préparation aux exemples

Rien de tel que la pratique pour appréhender et mémoriser les différentes composantes de CYPHER. Pour chaque exemple, nous vous donnerons le script CYPHER de création du graphe ainsi que les exemples de requêtes associées. Nous vous recommandons fortement de créer une nouvelle base à chaque fois qu'un nouveau graphe est utilisé.

Si vous utilisez votre propre installation de Neo4j (en local), vous pouvez passer par la console de Neo4j accessible à l'adresse <http://localhost:7474> depuis un navigateur web.

Figure 3.1 : Console Neo4j



Si vous souhaitez évaluer CYPHER, mais qu'aucune installation de Neo4j n'est à votre disposition, vous pouvez utiliser la [console du site Neo4j](#). Le bouton CLEAR DB permet de supprimer le graphe en cours d'utilisation afin d'en créer un nouveau.

Figure 3.2 : Console du site internet de Neo4j

The screenshot shows the Neo4j web console interface. At the top, there's a navigation bar with 'console.neo4j.org' and buttons for 'Clear DB', 'Help', 'Share', 'Toggle Viz', and 'Options'. Below the navigation bar, there's a message: 'If you want to share your graph, just do it with Share'. The main area displays two queries and their results.

Query 1:

```
START n=node(*)
MATCH n-[?]-m
WITH n, r
DELETE n, r
```

Query took 11 ms and returned no rows.
Updated the graph - deleted 7 nodes and 7 relationships [Result Details](#)

Query 2:

```
CREATE (sylvain:Personne { nom:"Sylvain" }),(christophe:Personne { nom:"Christophe" }),(philippe:Personne { nom:"Philippe" }), sylvain-[:PRETE { Livre:"Les fleurs du mal" }]->christophe, christophe-[:VENDS { Livre:"Les fleurs du mal" }]->philippe, philippe-[:OFFRE { Livre:"Les fleurs du mal" }]->sylvain
RETURN sylvain, christophe, philippe
```

Query took 50 ms and returned 1 rows.
Updated the graph - created 3 nodes and 3 relationships set 6 properties [Result Details](#)

The results of the second query are shown in a table:

sylvain	christophe	philippe
(0:Personne (nom:"Sylvain"))	(8:Personne (nom:"Christophe"))	(7:Personne (nom:"Philippe"))

On the right side of the console, there is a graph visualization showing three nodes: 'Personne Sylvain', 'Personne Christophe', and 'Personne Philippe'. Edges connect them: 'Personne Sylvain' is connected to 'Personne Christophe' (labeled 'PRETE'), 'Personne Christophe' is connected to 'Personne Philippe' (labeled 'VENDS'), and 'Personne Philippe' is connected to 'Personne Sylvain' (labeled 'OFFRE').

Enfin, si vous souhaitez conserver une trace de vos activités où que vous soyez, vous pouvez ouvrir gratuitement un compte sur le site de [GrapheneDB](#) pour créer une base Neo4j hébergée (supportant 1000 nœuds et 10000 relations pour le compte gratuit). La console d'administration sera exactement la même que celle de la version standard de Neo4j. Cliquez sur le bouton LAUNCH en bas de la page d'accueil de votre compte pour accéder à la console d'administration de Neo4j (voir [Figure 3.3](#)).

Note > À la différence de la console Neo4j, GrapheneDB est également accessible en REST.

Il existe d'autres services de ce type (voir [Annexe 3 : Ressources](#)) et une image Docker officielle pour Neo4j est également téléchargeable sur le [site de Docker](#).

Figure 3.3 : Page d'accueil d'un compte sur la plateforme GrapheneDB

The screenshot shows the GrapheneDB dashboard for a trial account. At the top, there is a navigation bar with the GrapheneDB logo, links for Docs, Blog, and Support, and a user profile for sylvain.roussy@free.fr. Below the navigation bar, the breadcrumb trail shows 'DATABASES > TRIAL_GRAPHENE'. The main content area is titled 'trial_graphene' and includes a 'Sandbox' section with a storage usage indicator (2 MB of 512 MB Storage, 0% Usage) and an 'Upgrade / Clone' button. Below this, there are two summary cards: one for 'Version' (Neo4j Community Edition 2.2.5, updated 9 days ago) and another for 'Nodes' (8 of 1,000) and 'Relationships' (15 of 10,000). At the bottom, there is a 'Tools' section with a 'Neo4j Web Admin interface' and a 'Launch' button. A small note at the bottom right states 'SSL not available. Read More'.

3.2. Requêtes de base

Avant d'aller plus avant et vous présenter en détail les requêtes les plus courantes, voici à quoi ressemble une requête CYPHER minimale pour commencer à interroger notre graphe :

```
MATCH (n) RETURN n
```

Cela signifie : *Retrouve tous les nœuds du graphe et place-les dans le résultat.*

La lettre *n* est un identificateur, ce n'est pas un élément de syntaxe. Cet identificateur va représenter tour à tour chaque nœud. Pour 100 nœuds dans notre graphe, nous aurons un résultat de 100 lignes, chaque ligne étant, dans ce cas, un nœud.

Voyons à présent ceci :

```
MATCH modele_de_chemin RETURN elements_de_resultats
```

Dans ce modèle de requête, nous allons exprimer un modèle de chemin (au sens graphe) plutôt qu'utiliser un nœud seul. De même, nous verrons qu'il est possible de choisir et de modifier les éléments d'un résultat.

Cela ne vous paraît pas très clair pour l'instant ? Pas de panique, d'ici quelques pages vous aurez compris où nous voulions en venir. En attendant, cela devrait vous suffire pour aborder la section suivante qui, justement, traite les *modèles de chemins*.

Réinitialisation du graphe

Selon les exemples qui vont suivre, il peut vous être demandé de réinitialiser votre graphe, c'est-à-dire de supprimer la totalité des nœuds et des relations qui y sont présents. La requête CYPHER suivante vous permettra d'effectuer cette opération :

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

Si vous utilisez une version 2.3 ou supérieure de Neo4j, une syntaxe plus courte est utilisable :

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

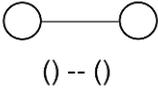
Note > Si vous utilisez la console du site Neo4j, vous pouvez supprimer le graphe en cours en cliquant simplement sur le bouton **CLEAR DB**.

3.3. Exprimer un modèle de chemin (*pattern matching*)

CYPHER est conçu pour être parfaitement adapté à l'interrogation des graphes, aussi son expression est assez proche du langage naturel. Afin de décrire des structures relationnelles, la syntaxe de CYPHER utilise un schéma semblable à de l'ASCII Art.

Considérons, ci-dessous, deux nœuds anonymes reliés par une relation non dirigée. Sur la partie supérieure de l'image se trouve le diagramme représentant le graphe, sur la partie inférieure son expression.

Exemple 3.1 : Deux nœuds anonymes, une relation non orientée

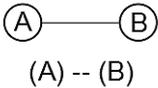


Chaque couple de parenthèses représente un nœud, et le double tiret la relation. En CYPHER, cela s'écrit :

```
MATCH ()--()
```

Ci-après un autre exemple avec des nœuds identifiés :

Exemple 3.2 : Deux nœuds identifiés, une relation non orientée

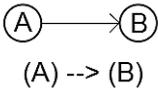


En CYPHER, cela s'écrit :

```
MATCH (A)--(B)
```

Nous allons à présent diriger la relation du nœud A vers le nœud B et vérifier son expression :

Exemple 3.3 : Deux nœuds identifiés, une relation orientée de A vers B



En CYPHER, cela s'écrit :

```
MATCH (A)-->(B)
```

Ainsi que du nœud B vers le nœud A, ce qui donne le résultat suivant :