

Rapport NEOJ4

Aya Hassan

1. Créer la structure de la base de données

Nous allons commencer par créer la structure du graph pour une base de données de films. Nous allons définir des nœuds Movie (Film) et Person (Personne) et créer des relations comme ACTED_IN (pour les acteurs), DIRECTED (pour les réalisateurs), et LIKES (pour les utilisateurs qui aiment des films).

Voici un exemple simple de schéma pour créer les nœuds et les relations :

```
1 // Créer des nœuds Movie (Film)
2 CREATE (m1:Movie {title: 'The Matrix', released: 1999, tagline: 'Welcome to the Real World'})
3 CREATE (m2:Movie {title: 'Inception', released: 2010, tagline: 'Your mind is the scene of the crime'})
4 CREATE (m3:Movie {title: 'The Dark Knight', released: 2008, tagline: 'Why so serious?'})
```

Added 3 labels, created 3 nodes, set 9 properties, completed after 34 ms.

```
1 // Créer des nœuds Person (Acteurs et Réalisateur)
2 CREATE (p1:Person {name: 'Keanu Reeves', born: 1964})
3 CREATE (p2:Person {name: 'Leonardo DiCaprio', born: 1974})
4 CREATE (p3:Person {name: 'Christian Bale', born: 1974})
5 CREATE (p4:Person {name: 'Christopher Nolan', born: 1970})
```

Added 4 labels, created 4 nodes, set 8 properties, completed after 8 ms.

```
1 // Créer des relations
2 CREATE (p1)-[:ACTED_IN]->(m1)
3 CREATE (p2)-[:ACTED_IN]->(m2)
4 CREATE (p3)-[:ACTED_IN]->(m3)
5 CREATE (p4)-[:DIRECTED]->(m2)
6 CREATE (p4)-[:DIRECTED]->(m3)
```

Created 7 nodes, created 5 relationships, completed after 21 ms.

2. Exemples de requêtes

Maintenant que nous avons des données de base, nous allons exécuter quelques requêtes Cypher pour interagir avec ces données.

Requête 1 : Trouver les films par acteur

Si vous souhaitez trouver tous les films dans lesquels un acteur a joué, vous pouvez utiliser la requête suivante. Par exemple, pour trouver tous les films dans lesquels Keanu Reeves a joué :

```

1 MATCH (p:Person {name: 'Keanu Reeves'})-[:ACTED_IN]→(m:Movie)
2 RETURN m.title AS Movie, m.released AS Year

```

	Movie	Year
1	"The Matrix"	1999
2	"The Matrix"	1999
3	"The Matrix"	1999
4	"The Matrix"	1999
5	"The Matrix Reloaded"	2003
6	"The Matrix Revolutions"	2003
7		

Started streaming 10 records after 7 ms and completed after 13 ms.

Pour trouver qui a réalisé un film spécifique, comme "Inception", vous pouvez exécuter :

```

1 MATCH (m:Movie {title: 'Inception'})←[:DIRECTED]-(p:Person)
2 RETURN p.name AS Director

```

	Director
1	"Christopher Nolan"
2	"Christopher Nolan"
3	"Christopher Nolan"

Started streaming 3 records after 6 ms and completed after 8 ms.

Requête 3 : Trouver tous les acteurs d'un film
 Pour trouver tous les acteurs dans un film, par exemple "The Dark Knight" :

```
1 MATCH (m:Movie {title: 'The Dark Knight'})←[:ACTED_IN]-(p:Person)
2 RETURN p.name AS Actor
```

Table	Actor
1	"Christian Bale"
2	"Christian Bale"
3	"Christian Bale"

Started streaming 3 records after 8 ms and completed after 10 ms.

Requête 4 : Trouver les films sortis après une certaine année
Pour trouver tous les films sortis après 2000 :

```
1 MATCH (m:Movie)
2 WHERE m.released > 2000
3 RETURN m.title AS Movie, m.released AS Year
```

Table	Movie	Year
1	"Inception"	2010
2	"The Dark Knight"	2008
3	"Inception"	2010
4	"The Dark Knight"	2008
5	"Inception"	2010
6	"The Dark Knight"	2008
7		

Started streaming 26 records after 6 ms and completed after 7 ms.

Requête 5 : Trouver les films aimés par un utilisateur (si vous avez des utilisateurs)

Si vous avez des utilisateurs et voulez suivre les films qu'ils aiment, vous pouvez étendre votre modèle avec des nœuds User et des relations LIKES.

```
1 // Créer un utilisateur et aimer des films
2 CREATE (u1:User {username: 'john_doe'})
3 CREATE (u1)-[:LIKES]→(m1)
4 CREATE (u1)-[:LIKES]→(m2)
5
```

Added 1 label, created 3 nodes, set 1 property, created 2 relationships, completed after 10 ms.

```
1 // Requête : films aimés par l'utilisateur 'john_doe'
2 MATCH (u:User {username: 'john_doe'})-[:LIKES]→(m:Movie)
3 RETURN m.title AS Movie
```

```
1 MATCH (m:Movie)←[:ACTED_IN]-(p:Person)
2 RETURN m, p
```

