Rapport du TP1 , Neo4j

DONE BY :

- **IMANE MALIKI**

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

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

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

```
// Créer des relations
CREATE (p1)-[:ACTED_IN]→(m1)
CREATE (p2)-[:ACTED_IN]→(m2)
CREATE (p3)-[:ACTED_IN]→(m3)
CREATE (p4)-[:DIRECTED]→(m2)
CREATE (p4)-[:DIRECTED]→(m3)
```

Created 7 nodes, created 5 relationships, completed after 7 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

- Pour trouver tous les films dans lesquels Keanu Reeves a joué :

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

	Movie	Year
1	"The Matrix"	1999

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

```
neo4j$ MATCH (m:Movie {title: 'Inception'})←[:DIRECTED]-(p:Person) RETURN p.name AS Director
```

	Director
1	"Christopher Nolan"

Requête 3 : Trouver tous les acteurs d'un film :

- Pour trouver tous les acteurs dans un film, par exemple "The Dark Knight" :

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

	Actor
1	"Christian Bale"

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
```

	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

Requête 5 : Trouver les films aimés par un utilisateur

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 59 ms.

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

	Movie
1	"The Matrix"
2	"Inception"
3	"The Matrix"
4	"Inception"