



Dans Neo4j, une relation est un élément à l'aide duquel nous relierons deux nœuds d'un graphe. Ces relations ont la direction, le type et la forme des modèles de données. Ce deuxième TP vous apprend à -

- Créer des relations
- Créer une relation entre les nœuds existants
- Créer une relation avec le label et les propriétés

1. Création d'une relationship

Nous pouvons créer une relation en utilisant la clause CREATE. Nous spécifierons la relation entre les crochets "[]" en fonction de la direction de la relation qu'il est placé entre le trait d'union " - " et la flèche " → " comme indiqué dans la syntaxe suivante.

Voici la syntaxe pour créer une relation en utilisant la clause CREATE.

```
CREATE (node1)-[:RelationshipType]->(node2)
```

Tout d'abord, créez deux nœuds Ind et Dhawan dans la base de données, comme indiqué ci-dessous.

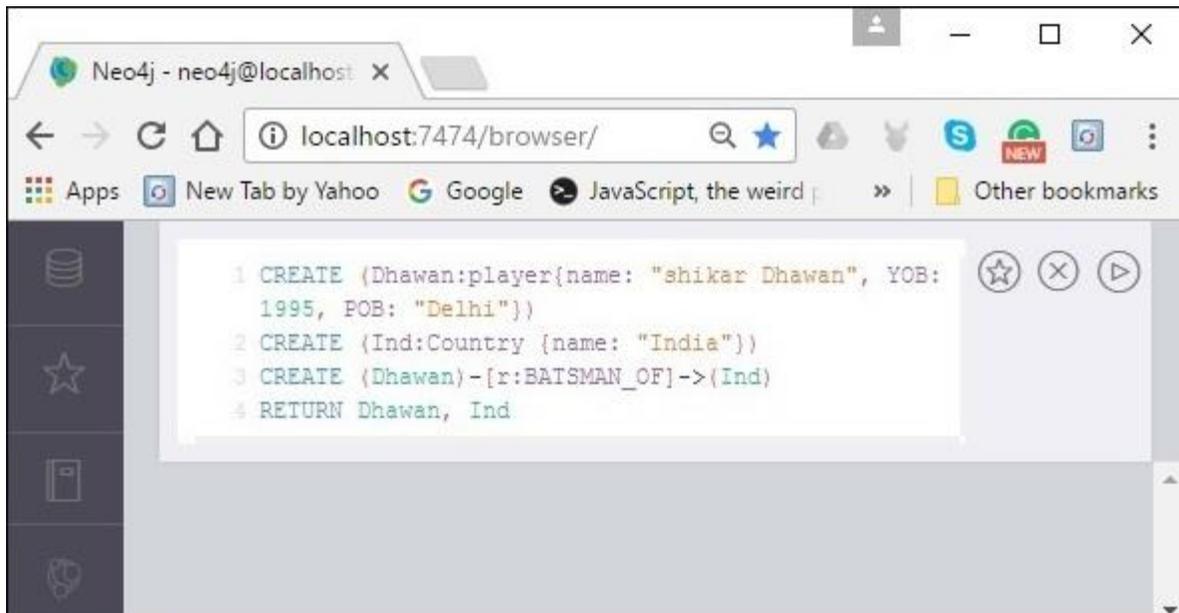
```
CREATE (Dhawan:player{name: "Shikar Dhawan", YOB: 1985, POB: "Delhi"})
CREATE (Ind:Country {name: "India"})
```

Maintenant, créez une relation nommée BATSMAN_OF entre ces deux nœuds comme :

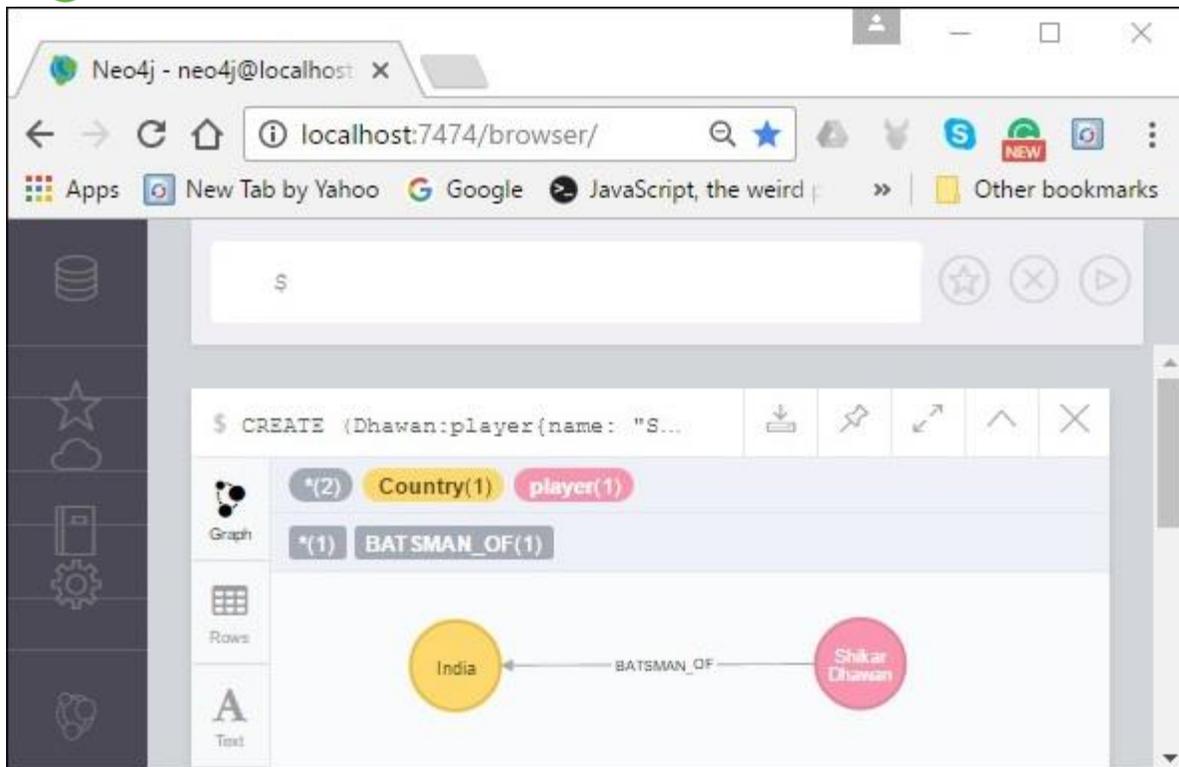
```
CREATE (Dhawan)-[r:BATSMAN_OF]->(Ind)
```

Enfin, retournez les deux nœuds pour voir la relation créée.

```
RETURN Dhawan, Ind
```



Lors de l'exécution, vous obtiendrez le résultat suivant.



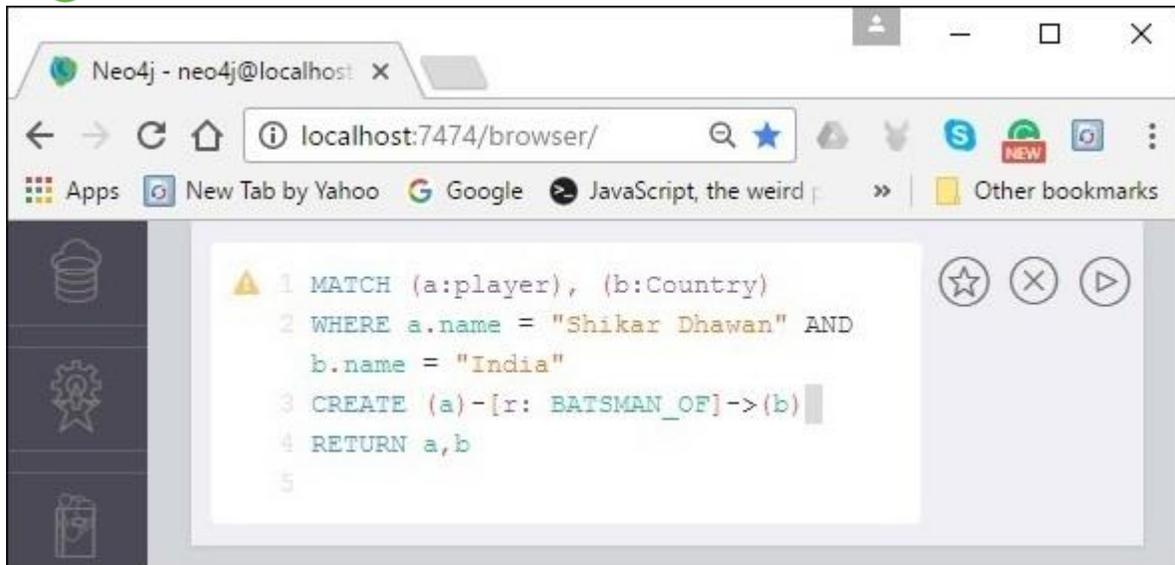
2. Creation d'une relationship entre deux nœuds existants

Vous pouvez également créer une relation entre les nœuds existants en utilisant la clause MATCH. Voici la syntaxe pour créer une relation en utilisant la clause MATCH.

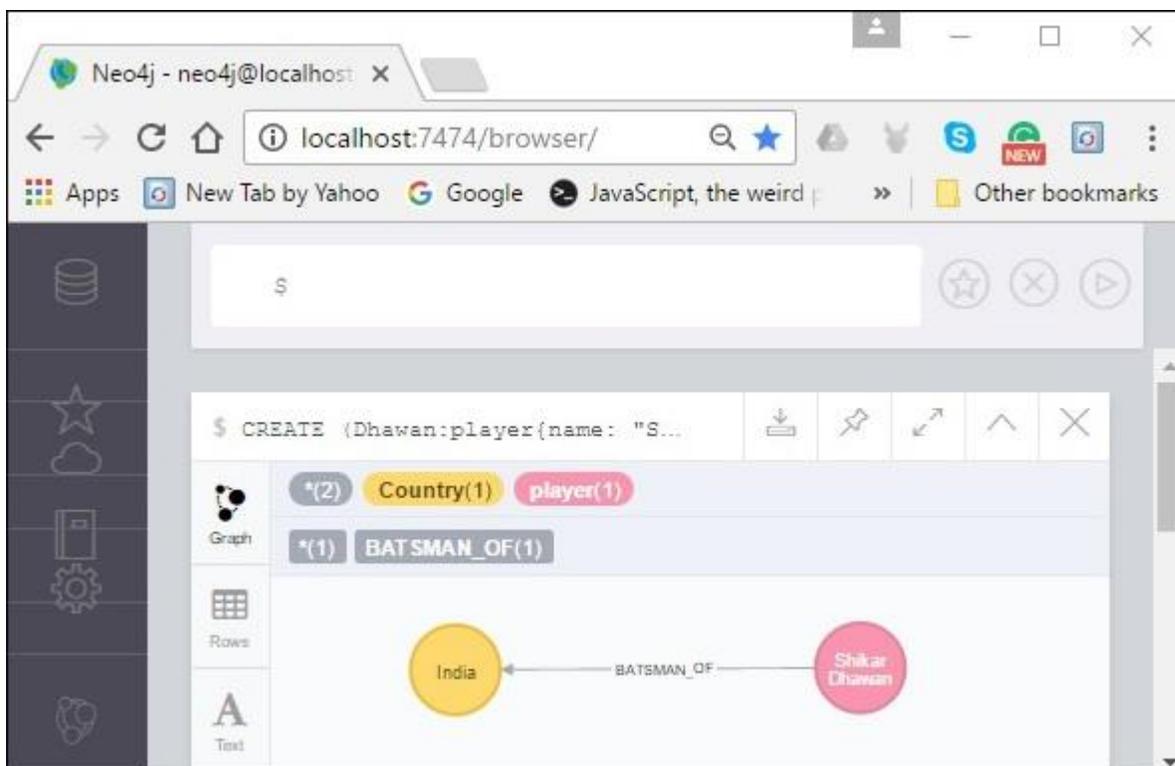
```
MATCH (a:LabeofNode1), (b:LabeofNode2)
  WHERE a.name = "nameofnode1" AND b.name = "nameofnode2"
CREATE (a)-[: Relation]->(b)
RETURN a, b
```

Vous trouverez ci-dessous un exemple de requête CQL qui crée une relation à l'aide de la clause de MATCH.

```
MATCH (a:player), (b:Country) WHERE a.name = "Shikar Dhawan" AND b.name = "India"
CREATE (a)-[r: BATSMAN_OF]->(b)
RETURN a,b
```



Lors de l'exécution, vous obtiendrez le résultat suivant.



3. Creation d'une relationship avec label et propriétés

Vous pouvez créer une relation avec le label et les propriétés en utilisant la clause CREATE.

```
CREATE (node1)-[label:Rel_Type {key1:value1, key2:value2, . . . n}]-> (node2)
```

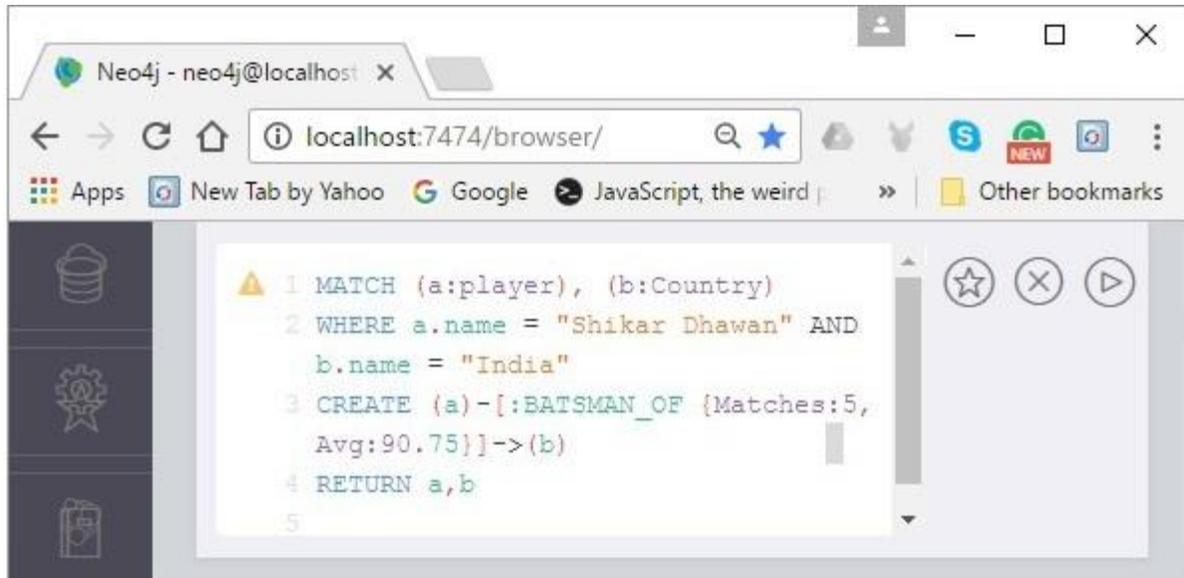


Voici un exemple de requête de chiffrement qui crée une relation avec l'étiquette et les propriétés.

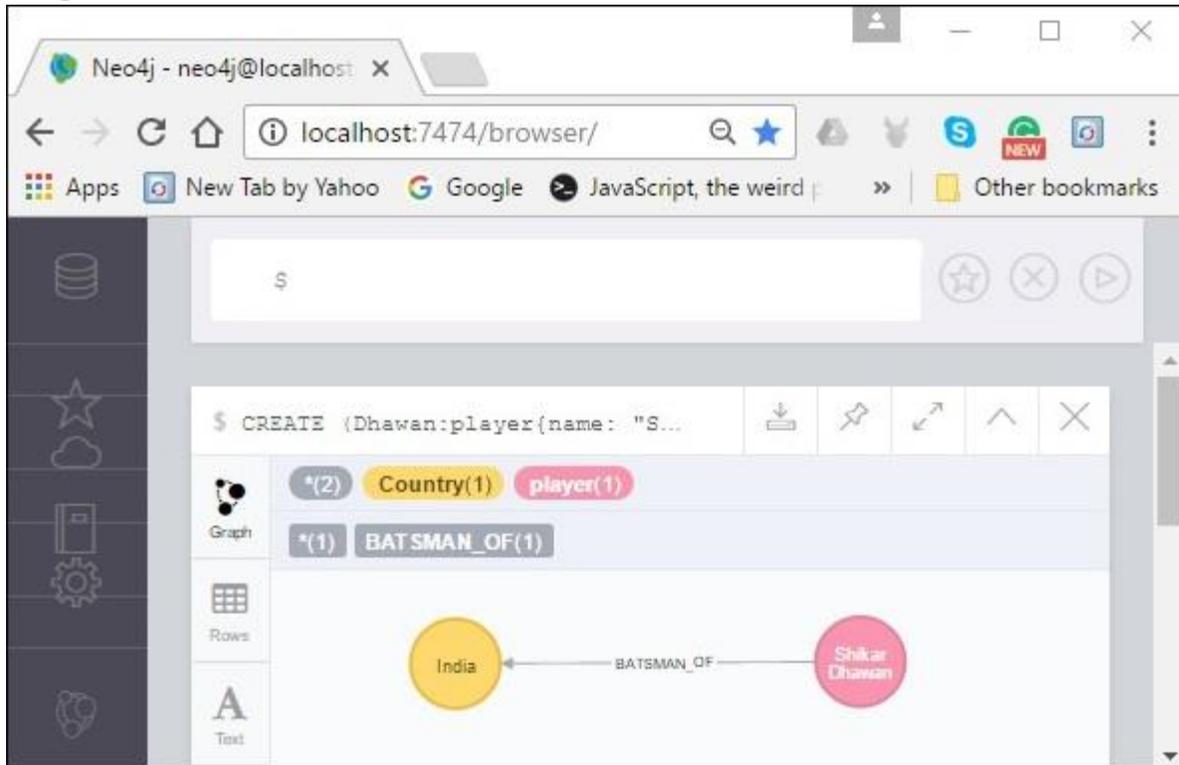
```
MATCH (a:player), (b:Country) WHERE a.name = "Shikar Dhawan" AND b.name = "India"
```

```
CREATE (a)-[r:BATSMAN_OF {Matches:5, Avg:90.75}]->(b)
```

```
RETURN a,b
```



Lors de l'exécution, vous obtiendrez le résultat suivant.



4. Création d'un chemin complet

Dans Neo4j, un chemin est formé en utilisant des relations continues. Un chemin peut être créé en utilisant la clause de création.

Voici la syntaxe pour créer un chemin dans Neo4j en utilisant la clause CREATE.

```
CREATE p = (Node1 {properties})-[:Relationship_Type]->
(Node2 {properties})-[:Relationship_Type]->(Node3 {properties})
RETURN p
```



```
1 CREATE p = (Dhawan {name:"Shikar
Dhawan"})-[:TOPSCORRER_OF]->
(Ind{name: "India"})-[:WINNER_OF]->
(CT2013{name: "Champions Trophy
2013"})
2 RETURN p
3
```

Lors de l'exécution, vous obtiendrez le résultat suivant.

