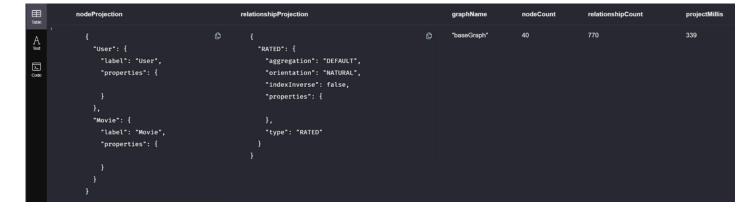
TP Neo4j

MUSTAPHA OULGHAZI

Étape 1 : Création d'une projection de base

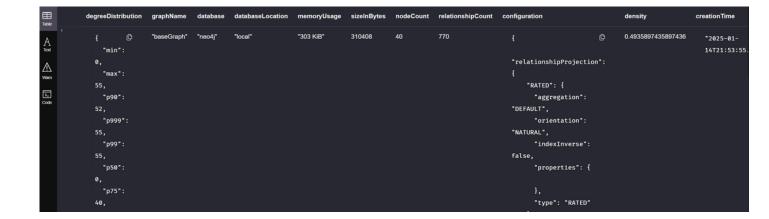
1 : Projetez un graphe contenant les nœuds User et Movie ainsi que les relations RATED

```
1 CALL gds.graph.project(
2    'baseGraph',
3    ['User', 'Movie'],
4    ['RATED']
5 );
```



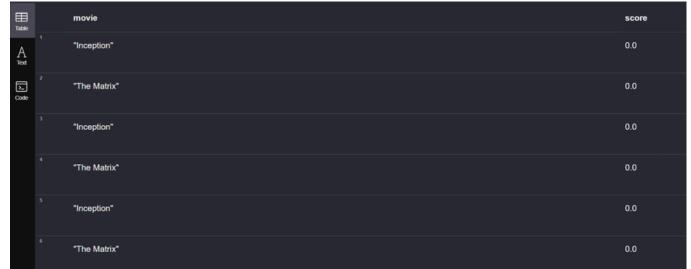
2 :Listez les graphes projetés pour vérifier la <u>projection.</u>

```
neo4j$ CALL gds.graph.list();
```



3 :Utilisez l'algorithme degree pour calculer le nombre de connexions de chaque nœud, et affichez les résultats pour les nœuds Movie.

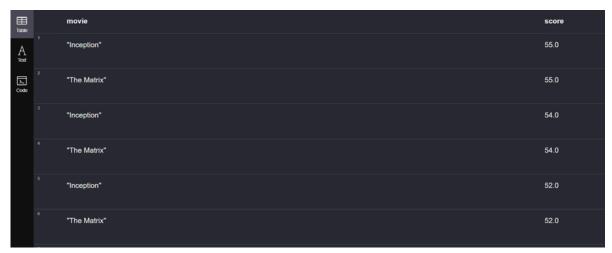
```
1 CALL gds.degree.stream('baseGraph')
2 YIELD nodeId, score
3 WHERE gds.util.asNode(nodeId):Movie
4 RETURN gds.util.asNode(nodeId).title AS movie, score
5 ORDER BY score DESC;
```



Étape 2 : Modification de l'orientation des relations

1:

```
1 CALL gds.degree.stream('reversedGraph')
2 YIELD nodeId, score
3 WITH gds.util.asNode(nodeId) AS node, score
4 WHERE labels(node) = ['Movie']
5 RETURN node.title AS movie, score
6 ORDER BY score DESC;
```



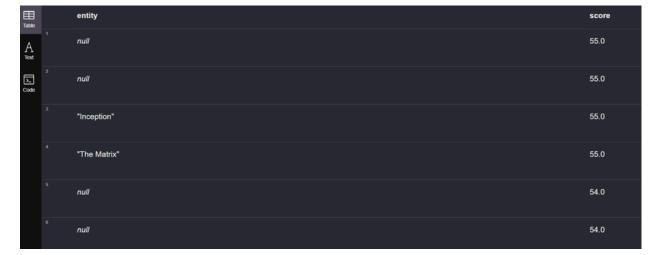
Étape 3 : Relations non orientées

 Projetez un graphe en spécifiant que les relations RATED sont non orientées.



3 : Appliquez l'algorithme degree pour analyser les connexions dans ce graphe.

```
1 CALL gds.degree.stream('undirectedGraph')
2 YIELD nodeId, score
3 RETURN gds.util.asNode(nodeId).title AS entity, score
4 ORDER BY score DESC;
```



Étape 4: Analyse avancée

1. Ajoutez une propriété weight aux relations RATED.

```
1 MATCH ()-[r:RATED]→()
2 SET r.weight = r.rating;

Set 770 properties, completed after 243 ms.

Table

Code
```

3: Projetez un graphe en incluant cette propriété et utilisez-la dans un calcul de degré pondéré.

3 :Comparez les résultats avec ceux des étapes précédentes.

```
1 CALL gds.degree.stream('weightedGraph', { relationshipWeightProperty: 'weight' })
2 YIELD nodeId, score
3 WITH gds.util.asNode(nodeId) AS node, score
4 WHERE 'Movie' IN labels(node)
5 RETURN node.title AS movie, score
6 ORDER BY score DESC;
```

