TP: Neo4j

Asnaoui Lahcen

Étape 1 : Création d'une projection de base

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

```
CALL gds.graph.project(
  'baseGraph',
  ['User', 'Movie'],
  ['RATED']
);
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	projectMillis
{ "User": { "label": "User", "properties": { } }, "Movie": { "label": "Movie", "properties": { } } }	{ "RATED": { "aggregation": "DEFAULT", "orientation": "NATURAL", "indexInverse": false, "properties": { }, "type": "RATED" } }	"baseGraph"	40	770	339

2 :Listez les graphes projetés pour vérifier la projection.

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

degreeDistribution	graphName	database	databaseLocation	memoryUsage	sizeInBytes	nodeCount	relationshipCount	configuration	density	creationTime
{ "min": 0, "max": 55, "p90": 52, "p999": 55, "p99": 55, "p50": 0, "p75": 40,	"baseGraph"	"neo4j"	"local"	"303 KiB"	310408	40	770	{ "relationshipProjection": { "RATED": { "aggregation": "DEFAULT", "orientation": "NATURAL", "indexInverse": false, "properties": { }, "type": "RATED"	0.4935897435897436	"2025-01-14T21:53:55.

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

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

	movie	score
1	"Inception"	0.0
2	"The Matrix"	0.0
3	"Inception"	0.0
4	"The Matrix"	0.0
5	"Inception"	0.0
6	"The Matrix"	0.0

Étape 2 : Modification de l'orientation des relations

1 :

```
CALL gds.graph.project(
  'reversedGraph',
  ['User', 'Movie'],
  { RATED: { orientation: 'REVERSE' } }
);
```

	nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	projectMillis
1	{ "User": { "label": "User", "properties": { } }, "Movie": { "label": "Movie", "properties": { } } }	{ "RATED": { "aggregation": "DEFAULT", "orientation": "REVERSE", "indexInverse": false, "properties": { }, "type": "RATED" } }	"reversedGraph"	40	770	14

2 -3 :

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

	movie	score
1	"Inception"	55.0
2	"The Matrix"	55.0
3	"Inception"	54.0
4	"The Matrix"	54.0
5	"Inception"	52.0
6	"The Matrix"	52.0

Étape 3 : Relations non orientées

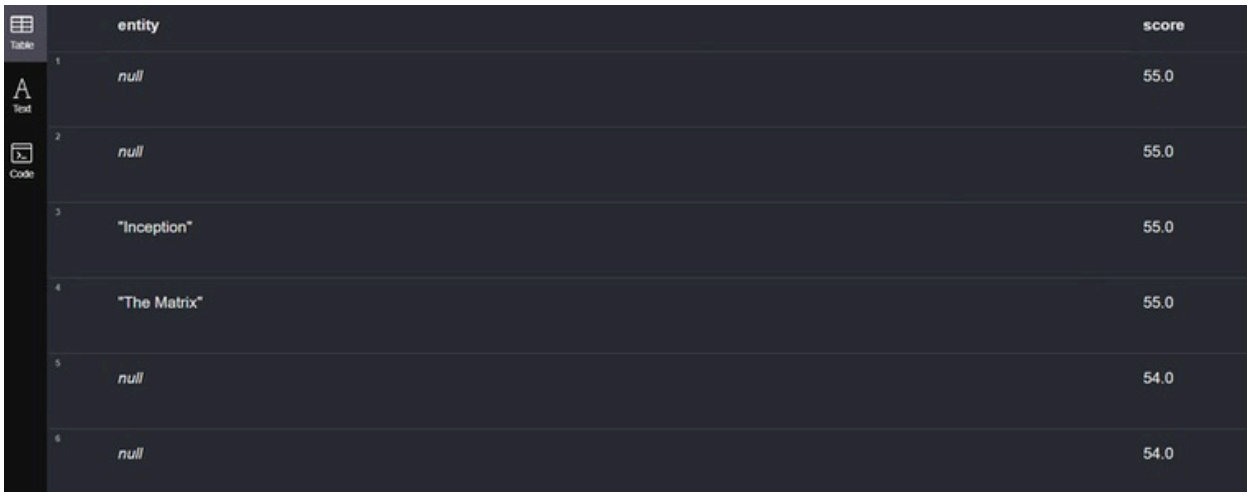
1. Projetez un graphe en spécifiant que les relations RATED sont non orientées.

```
CALL gds.graph.project(
  'undirectedGraph',
  ['User', 'Movie'],
  { RATED: { orientation: 'UNDIRECTED' } }
);
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	projectMillis
{ "User": { "label": "User", "properties": { } }, "Movie": { "label": "Movie", "properties": { } } }	{ "RATED": { "aggregation": "DEFAULT", "orientation": "UNDIRECTED", "indexInverse": false, "properties": { }, "type": "RATED" } }	"undirectedGraph"	40	1540	13

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

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

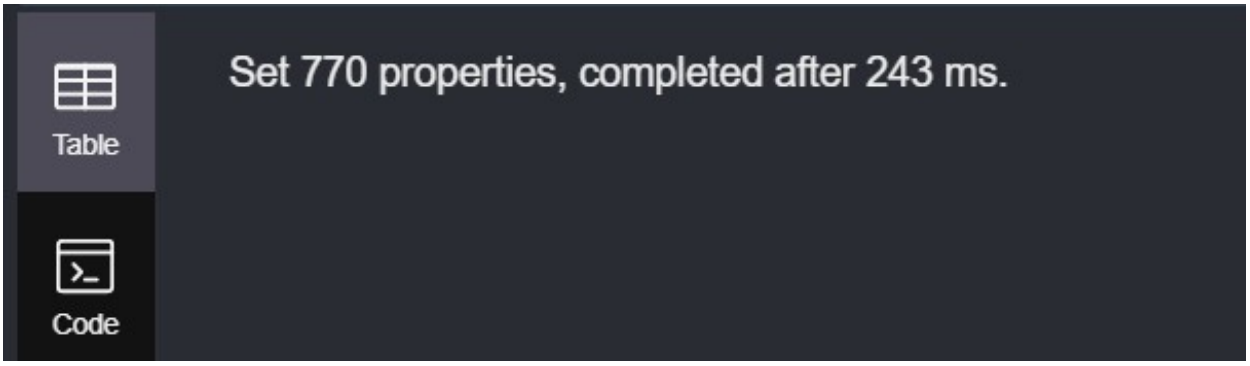


	entity	score
1	*null*	55.0
2	*null*	55.0
3	"Inception"	55.0
4	"The Matrix"	55.0
5	*null*	54.0
6	*null*	54.0

Étape 4 : Analyse avancée

1.Ajoutez une propriété weight aux relations RATED.

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



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

```
1  CALL gds.graph.project(
2    'weightedGraph',
3    ['User', 'Movie'],
4    {
5      RATED: {
6        orientation: 'NATURAL',
7        properties: 'weight'
8      }
9    }
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	projectMillis
1	1	"weightedGraph"	40	770	61


```
{
  "User": {
    "label": "User",
    "properties": {

    }
  },
  "Movie": {
    "label": "Movie",
    "properties": {

    }
  }
}
```

```
{
  "RATED": {
    "aggregation": "DEFAULT",
    "orientation": "NATURAL",
    "indexInverse": false,
    "properties": {
      "weight": {
        "aggregation": "DEFAULT",
        "property": "weight",
        "defaultValue": null
      }
    },
    "type": "RATED"
  }
}
```


3 :Comparez les résultats avec ceux des étapes précédentes.

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

	movie	score
1	"Inception"	0.0
2	"The Matrix"	0.0
3	"Inception"	0.0
4	"The Matrix"	0.0
5	"Inception"	0.0
6	"The Matrix"	0.0