

# Neo4j - 2

Running on local

# Importing a dataset

- LOAD CSV
-

# Dans le navigateur

- neo4j status | restart | ...
  - <http://localhost:7474>
- cypher-shell
  - show databases;

\$ :server connect

### Connect to Neo4j

Database access might require an authenticated connection

Connect URL

neo4j:// localhost:7687

Database - leave empty for default

Authentication type

Username / Password

Username

Password

Connect

- **cypher-shell**
  - `show databases;`
  - `:use northwind;`
  - `CALL db.labels;`
  - `CALL db.relationshipTypes;`
  - `CALL db.schema.visualization;`
  -

# Import csv

<https://github.com/neo4j-graph-examples/star-wars/tree/main/import>

git clone [git@github.com:neo4j-graph-examples/star-wars.git](https://github.com/neo4j-graph-examples/star-wars.git)

```
> star-wars git:(main) ls import
starwars.nodes.Character.csv      starwars.nodes.Species.csv      starwars.relationships.APPEARED_IN.csv  starwars.relationships.PILOT.csv
starwars.nodes.Film.csv          starwars.nodes.Starship.csv     starwars.relationships.HOMEWORLD.csv
starwars.nodes.Planet.csv        starwars.nodes.Vehicle.csv     starwars.relationships.OF.csv
```

Mettre les csv dans le repertoire import de Neo4j  
/Users/alexis/Library/Application Support/Neo4j  
Desktop/Application/distributions/neo4j/neo4j-enterprise-  
5.24.0/import

## Ensuite LOAD CSV en spécifiant les propriétés

```
LOAD CSV WITH HEADERS FROM 'file:///starwars.nodes.Character.csv' AS row

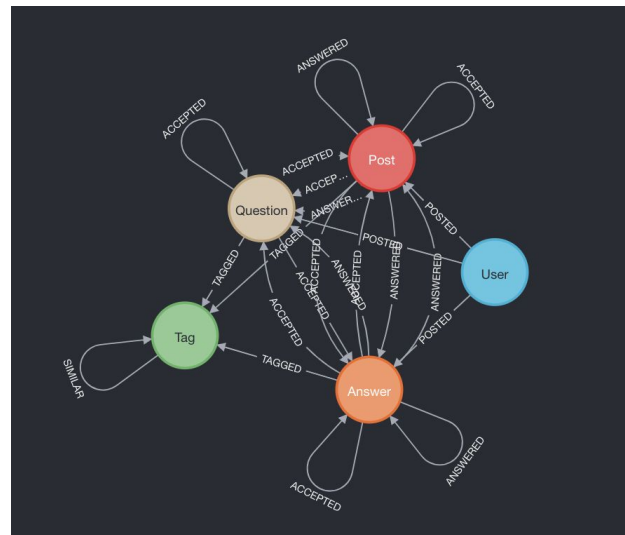
CREATE (:Character {
    id: toInteger(row.:ID),
    name: row.name,
    birth_year: row.birth_year,
    eye_color: row.eye_color,
    height: toInteger(row.height),
    gender: row.gender,
    skin_color: row.skin_color
});
```

# Demo datasets

<https://neo4j.com/docs/getting-started/appendix/example-data/>

<https://demo.neo4jlabs.com:7473/browser/?dbms=neo4j://stackoverflow@demo.neo4jlabs.com&db=stackoverflow>

```
CALL db.schema.visualization()
```



```
CALL db.labels() YIELD label
```

```
CALL apoc.cypher.run('MATCH (n:' + label + ') RETURN count(n) AS count', {}) YIELD value
```

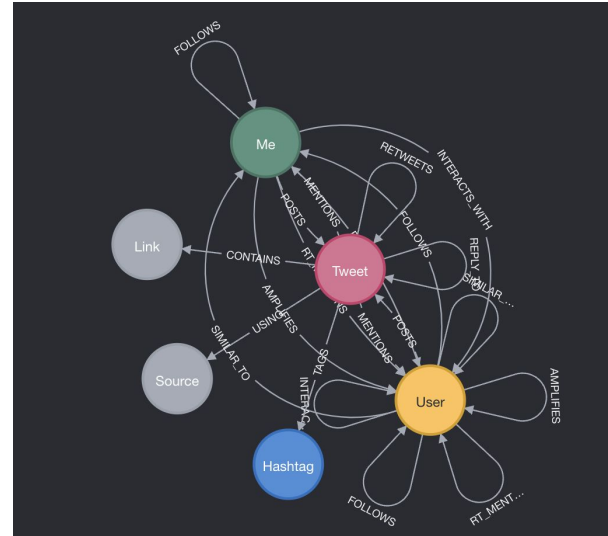
```
RETURN label, value.count AS count
```

```
ORDER BY count DESC
```



You are connected as user **twitter**

```
to neo4j+s://demo.neo4jlabs.com
```



CALL db.labels() YIELD label

CALL apoc.cypher.run('MATCH (n:' + label + ') RETURN count(n) AS count', {}) YIELD value

RETURN label, value.count AS count

ORDER BY count DESC

# Mesures de Centralité dans Neo4j Northwind

## Qu'est-ce que la Centralité?

- Mesure l'importance des nœuds dans un graphe
  - Identifie les acteurs/éléments clés du réseau
  - Plusieurs métriques différentes selon le contexte
- Degree Centrality
  - Betweenness
  - Closeness
  - Page Rank

## Définition

- Mesure le nombre de relations directes qu'un nœud possède
- Plus simple mesure de centralité
- Formule:  $C_D(v) = \deg(v)$

- In-degree: Number of incoming relationships
- Out-degree: Number of outgoing relationships
- Total degree: Sum of both

## Cas d'utilisation

- Identifier les hubs dans un réseau
- Repérer les éléments très connectés
- Première analyse rapide d'influence

# Degree Centrality - Northwind

```
// Degree centrality des employés basée sur les commandes traitées

MATCH (e:Employee)-[:SOLD]->(o:Order)
WITH e, count(o) AS DegreeScore
RETURN e.lastName AS Employee, DegreeScore
ORDER BY DegreeScore DESC
LIMIT 5
```

## Degree Centrality

- Clients basée sur le nombre de commandes
- Produits basée sur le nombre de commandes contenant ce produit
- ...

# Degree Centrality - Northwind

```
// Degree Centrality bidirectionnelle des produits (co-achat)

MATCH (p1:Product)<-[:ORDERS]->(:Order)<-[:ORDERS]->(p2:Product)
WHERE id(p1) <> id(p2)
WITH p1, count(DISTINCT p2) AS DegreeScore
RETURN p1.productName AS Product,
        DegreeScore
ORDER BY DegreeScore DESC
LIMIT 5;
```

Find the top 5 users

- who posted the most questions

- `(u:User)-[:POSTED]->(q:Question)`

- 

- who answered the most questions

- `(u:User)-[:ANSWERED]->(q:Question)`

-

## Définition

- Mesure la fréquence à laquelle un nœud se trouve sur les plus courts chemins entre d'autres nœuds
- Identifie les "ponts" dans un réseau
- Formule:  $C_B(v) = \sum(\sigma_{st}(v) / \sigma_{st})$

## Cas d'utilisation

- Détecter les intermédiaires critiques
- Identifier les goulots d'étranglement
- Repérer les nœuds contrôlant le flux d'information

Nécessite the module GDS seulement disponible en local