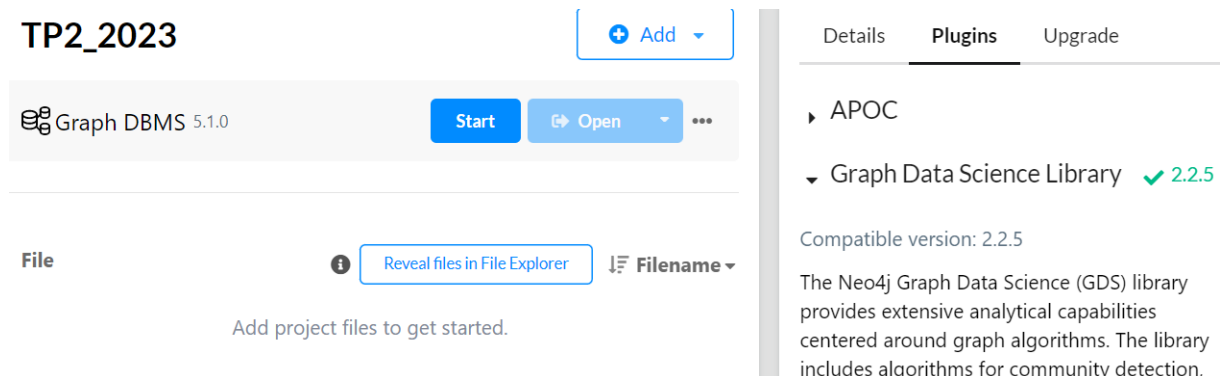


# Analyse de Graphes de données

## Neo4j : la librairie DATA SCIENCE

### I. Installation de la librairie et construction de la base de données

- Lancer Neo4jDesktop
- Créer un projet TP2, et y ajouter Local DBMS (cliquer sur ADD). Cliquer ensuite sur cette base, un menu apparaît à droite. Choisir l'onglet plugins et installer la librairie GDS.
- Lancer le projet dans le Browser : START ensuite OPEN



TP2\_2023

Graph DBMS 5.1.0

Start Open

File

Reveal files in File Explorer

Filename

Add project files to get started.

Details Plugins Upgrade

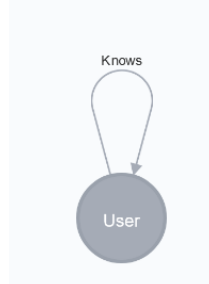
APOC

Graph Data Science Library ✓ 2.2.5

Compatible version: 2.2.5

The Neo4j Graph Data Science (GDS) library provides extensive analytical capabilities centered around graph algorithms. The library includes algorithms for community detection,

- Vérifier que la librairie est bien présente : RETURN gds.version()
- La liste des algorithmes est **CALL gds.list()**
- Créez la base de données en exécutant une à une les deux instructions fournies dans le fichier load.txt
- Vérifier le schéma de la base ainsi créée : **call db.schema.visualization()**



L'association KNOWS entre personnes possède 2 attributs : le type de la relation (relationType : studentOf, colleagueOf, advisorOf, etc.) et son intensité (weight qui mesure avec un nombre entre 0 et 10 la qualité de la relation). Vous pouvez vérifier l'adéquation de votre schéma avec celui demandé en le visualisant : **call db.schema.visualization()**

- II. **Questions : Déposer sur moodle les commandes nécessaires pour réaliser les tâches suivantes, lorsque le résultat le permet les requêtes doivent être accompagnées de leur résultat. Attention le dépôt doit se faire à la fin de la séance (clôture automatique de la zone à 18h00).**
- Donner les statistiques suivantes en utilisant les connaissances acquises durant le TP1:
    - Donner la valeur minimale, maximale, moyenne ainsi que l'écart type des interactions entre personnes (aucun algorithme de la GDS n'est nécessaire, il suffit d'avoir le degré).
    - Donner le degré min, max, moyen ainsi que l'écart type par type de relation (la relation KNOWS possède plusieurs types : voir l'attribut type)
  - Créer une projection du graphe en prenant uniquement les nœuds User, l'association KNOWS sans aucun attribut.
  - Quels est le chemin le plus court entre la personne ayant l'id 5 et celle ayant l'id 10 (en terme de nombre d'arêtes) : la requête doit retourner tous les nœuds sur ce chemin. Vous pouvez utiliser l'algorithme de Dijkstra
  - Créer une projection du graphe en prenant uniquement les nœuds User, l'association KNOWS avec l'attribut weight.
  - Reprendre la question 3 en considérant la qualité du chemin (propriété weight).
  - Donnez la distance en terme de nombre d'arêtes entre chaque paire de nœuds.
  - Reprendre la question précédente en prenant en considération les poids des arêtes
  - Rajouter une propriété « cc » aux nœuds du graphe, elle permet de spécifier la composante connexe à laquelle appartient chaque nœud ensuite afficher les nœuds avec leur composante.
    - Affichez les partitions avec pour chacune sa taille et la liste des noms des personnes.
    - Donner le nombre moyen de contact par personne pour chaque composante connexe.
    - Affichez les résultats obtenus par les autres algorithmes de détection de communautés (Louvain et Label Propagation). Est-ce que ces deux algorithmes retournent la même chose ?
  - L'importance d'un nœud dans un réseau peut être mesurée par le nombre d'interactions qu'il a avec les autres nœuds, donc son degré, rajoutez aux nœuds une telle propriété et affichez la pour toutes les personnes.
  - Calculer le diamètre du graphe (le diamètre géodésique d'un graphe est défini comme le plus grand court chemin)
  - Rajouter une propriété qui mesure combien un nœud est stratégique (un nœud est stratégique si les informations transitent souvent par lui. Une telle position intermédiaire donne à ce nœud pouvoir et influence : en général il est reconnu par le nombre de courts chemins passant par lui).
  - Quels sont les nœuds qui ont les distances les plus courtes à tous les autres nœuds ?
  - D'après le résultat de la question précédente c'est Pauline qui est la plus proche de tous les autres ? comment rendre le résultat plus pertinent ?

14. Rajoutez une propriété qui mesure l'efficacité avec laquelle chacun profite de ses relations et affichez le résultat pour tous nœuds (pagerank).
15. Les propriétés rajoutées dans les questions 9, 11 et 14 mesurent chacune de manière différente et d'un point de vue différents l'importance d'un nœud ou son influence. Ecrivez une requête qui affiche les 3 valeurs.
16. Combien de triangles on a dans ce graphe pour chaque nœud? calculer le coefficient de clustering du graphe.
17. Rajouter un nœud par composante connexe, ensuite une relation IN\_COMMUNITY allant d'un nœud vers sa composante connexe.
18. Calculer le pagerank des nœuds par composante connexe.