

# Base de données orientée graphe chez Stack Overflow.





## Objectifs du TP :

- Mise en place de la base de données Stack Overflow.
- Pratiquer les opérations CRUD (Create, Read, Update, Delete) dans Neo4j.
- Pratiquer l'agrégation dans Neo4j.



## Partie 1 : Installation du projet

Nous allons commencer par importer la base de données utilisée pour ce TP. Pour ce faire, cliquez sur le bouton  et sélectionnez «File». Sélectionnez maintenant le fichier dump fournis sur Moodle.

Il ne vous reste maintenant plus qu'à cliquer sur les trois petits points et faire «Create new DBMS from Dump».

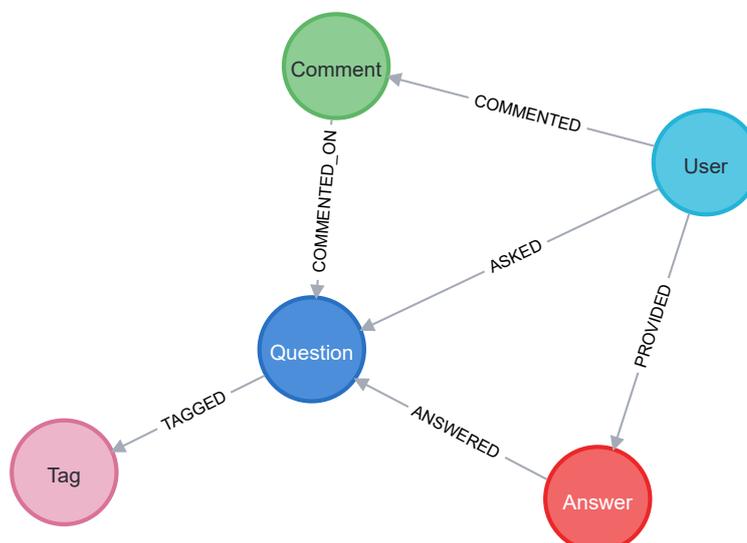


Utiliser la version V5 au minimum de Neo4j. Exemple : V5.12.0

La base de données vous affiche un statut « active », vous pouvez commencer à travailler en ouvrant votre base de données.

Durant ce TP, vous allez travailler sur une extraction de la base de données de Stack Overflow. Une représentation de la base de données vous est donnée ci-dessous.

On y retrouve différents nœuds : étiquette (*Tag*), question (*Question*), commentaire (*Comment*), utilisateur (*User*) et réponse (*Answer*). Ces nœuds sont reliés par des relations, par exemple : un *User* peut poser une *Question* ; la relation est modélisée par le lien *Asked*.





## Partie 2 : Where is Bobis ?

En utilisant le cours et son support, répondez aux différentes questions sur l'accès en lecture aux données. Vous détaillerez la requête dans son format Cypher. Le résultat retourné devra correspondre avec le retour attendu dans ce sujet. Comme dans tous les SGBD, il existe potentiellement plusieurs façons d'arriver à un résultat donné. Cependant, la performance est votre unique objectif ! Vous privilégieriez donc la méthode la plus performante.



**Question 1 :** Retournez le *User* qui s'appelle « Bobis »

n
(:User {display_name: "Bobis",uuid: 12152568})



**Question 2 :** Retournez uniquement le uuid du *User* qui s'appelle « Bobis »

n.uuid
12152568



**Question 3 :** Retournez uniquement le uuid du *User* qui s'appelle « Bobis ». Le nom de la colonne dans le résultat doit s'appeler « UUID\_de\_Bobis ».

UUID_de_Bobis
12152568



**Question 4 :** Combien de questions Bobis a t-il posé sur Stack Overflow ?

nb_question
1



**Question 5 :** Quel est le contenu de la question la plus ancienne de Bobis ?  
L'utilisation de Order by et Limit n'est pas une solution convenable.

Résultat : [I need to create a search application \(searching for recipes by ingredients\) that uses the neo4j ...](#)



**Question 6 :** Combien de personnes ont répondu à cette question de Bobis ?

Aucune personne n'a répondu à la question de Bobis.



**Question 7 :** Trouver la question (ou les questions si ex aequo) avec le plus de commentaires de Stack Overflow. L'utilisation de Order by et Limit n'est pas une solution convenable

q.title	NumComments
"neo4j - pagination return same results even when skip and order by are specified"	13



**Question 8 :** La Stack Overflow Conf arrive à grand pas ! Cette journée est l'occasion pour Stack Overflow de partager un ensemble d'accomplissements et de nouveautés pour la communauté. C'est notamment le moment très attendu pour dévoiler la « [Stack Overflow Survey 2023](#) ».

Durant cette conférence, quelques minutes seront dédiées pour élire et remercier les meilleurs contributeurs de la plateforme. Dans cet objectif, on vous demande de confectionner la requête nécessaire en respectant les règles suivantes.

Nous allons baser la sélection sur 3 scores qui devront être ensuite additionnés pour donner le score final de l'utilisateur :

- **Le score des commentaires :** 1 point par commentaire qui est posté sur une question autre que la sienne.
- **Le score des questions :** Normalisez le `view_count` d'une question sur une échelle entière de 0 à 150 (cf sur la [normalisation](#)). C'est autant de points que recevra le rédacteur de cette question.
- **Le score des réponses :** Si l'utilisateur est dans les 5% des personnes ayant le plus de réponses, il recevra 50 points supplémentaires. L'ajout des 50 points ne se fera qu'une seule fois, quel que soit le nombre de réponses.

Pour la réalisation de cette requête, vous utiliserez les éléments suivants : max, min, count, sum, percentileDist et coalesce. Vous devrez vous référer à la documentation en ligne pour comprendre ceux que vous ne connaissez pas.

Nous vous conseillons de réaliser les scores un par un et de les stocker sur l'utilisateur. Vous travaillerez ensuite à calculer la somme.

Ci-dessous les résultats que vous devez obtenir :

userName	finalScore
"cybersam"	157
"fbiville"	156
"Jefferey John Smith"	150
"Harsh Verma"	147
"Tomaž Bratanič"	144
"jose_bacoy"	134
"JoyfulPanda"	131
"lpt"	124
"InverseFalcon"	104
"lovgrandma"	104





### Pour aller plus loin : The Shortest Challenge

L'objectif de ce challenge est de proposer la requête Cypher avec le moins de caractère possible. Vous pouvez utiliser n'importe quelle méthode tant que l'objectif est atteint.

La *Question* avec le *uuid: 69162801* possède un *Tag name: cypher*. Cependant, en regardant le contenu de cette question, je me rends compte que les *Tag* suivants seraient intéressants à rajouter : R, rglobi, Globi, coding, help.

Créez et ajoutez les *Tag* à cette *Question*.

*Record à battre : 138 caractères*

*Par : Julien Breton*

*Année : 2023*





## Partie 3 : Press F to pay respects

Toujours en utilisant le cours et son support, répondez aux différentes questions sur la suppression de données dans Neo4j.

Comme toutes les entreprises, Stack Overflow doit respecter le RGPD (Règlement Général pour la Protection des Données). Dans ce cadre, lorsqu'un utilisateur fait appel à son [droit à l'effacement](#), Stack Overflow doit supprimer les données le concernant. Après discussion avec le DPD (Délégué à la Protection des Données) vous avez constaté qu'il n'existait aucun traitement automatique. Vous êtes chargé de la mise en conformité.

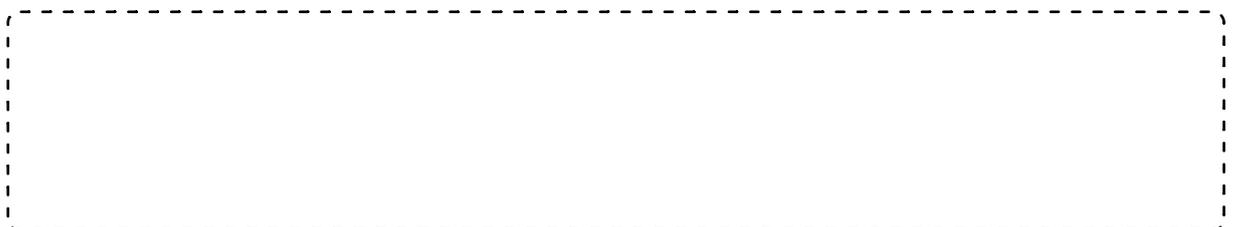


**Question 9 :** Le *User* « Graphileon » souhaite supprimer son compte de Stack Overflow, vous profitez de cette occasion pour établir les différentes requêtes qui seront exécutées dans le script. Avant de vous lancer à corps perdu dans la suppression, listez les *Question*, *Comment* et *Answer* auxquels « Graphileon » est connecté directement.

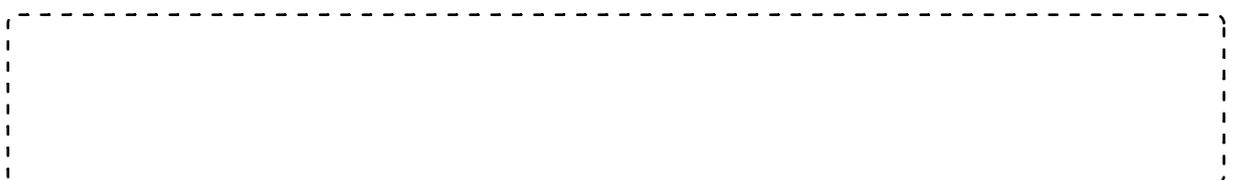


**Question 10 :** Nous commencerons par supprimer les liens *TAGGED* entre les *Question* et les *Tag*. Donnez la requête correspondante.

5 liens doivent être supprimés



**Question 11 :** Nous avons décidé que tous les *Comment* et *Answer* de l'utilisateur seraient supprimés. Donnez la requête correspondante.





**Question 12 :** Nous devons maintenant supprimer toutes ses *Question* mais aussi toutes les *Answer* et *Comment* postés dessus. L'objectif est de supprimer correctement les données et de ne pas laisser de «données mortes». En production, un oubli de ce type pourrait avoir de sérieuses conséquences.



Prendre en compte le cas dans lequel une question a été posée mais ne possède aucune *Answer* ou *Comment*. Voir la documentation du optional match en ligne.



**Question 13 :** En utilisant de nouveau la requête de la question 11, vérifier qu'il ne reste plus de nœud lié à notre *User*. Il serait aussi intéressant de vérifier qu'il n'existe pas de nœuds *Comment* et *Answer sans lien* (voir la documentation) vers un nœud *Question*. Dans le cas contraire, chercher l'erreur dans vos requêtes précédentes.



**Question 14 :** Si la question 14 est correcte, il ne nous reste plus qu'à supprimer l'utilisateur.





UNIVERSITÉ  
TOULOUSE III  
PAUL SABATIER



UNIVERSITÉ TOULOUSE III



This work © 2023 by Julien Breton is licensed under CC BY-NC-SA 4.0.

To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>