

TD Neo4j

Analyse de popularité sur Twitter

1 Description du cas d'étude

En tant qu'analyste de données sociales sur le réseau Twitter, afin de comprendre les dynamiques en jeu sur le réseau, on souhaiterait connaître:

- quels sont les tweets les plus retweetés,
- quels sont les utilisateurs les plus suivis,
- quels sont les utilisateurs les plus mentionnés,
- quels sont les hashtags les plus populaires.

2 Modélisation

En considérant:

- les données accessibles sur la plateforme Twitter (voir document de présentation),
- les objectifs décrits ci-dessus,
- les interrogations type ci-dessous (Q1 à Q14),

quelle modélisation mettriez-vous en oeuvre ?

3 Interrogation

On suppose le graphe modélisé ci-dessus implanté dans Neo4j. Donnez les requêtes Cypher permettant d'obtenir les besoins Q1 à Q11¹:

- Q1. le nombre d'utilisateurs
- Q2. le nombre de tweets contenant le hashtag *WhiteRabbit*
- Q3. le nombre d'utilisateurs différents qui ont tweeté un tweet contenant le hashtag *AliceInWonderland*
- Q4. le nom des followers de *@Alice*
- Q5. le nom des utilisateurs suivis par *@Alice*
- Q6. le nom des utilisateurs ayant plus de 10 followers.
- Q7. les utilisateurs étant à la fois followers et followees de *@Alice* (c'est à dire qui suivent et sont suivis par *@Alice*)
- Q8. les 10 tweets les plus populaires (ici la popularité se mesure au nombre de fois où le tweet est favori)
- Q9. les 10 hashtags les plus populaires (ici, la popularité se mesure au nombre de tweets dans lesquels ils apparaissent)

¹Les requêtes Q12, Q13 et Q14 seront directement faites en TP.

- Q10. le nom des followers des followers des followers des followers de *@Alice* (followers au 4ème degré).
- Q11. le nom des followers au dixième degré de *@Alice*
- Q12. les tweets du 30/08/21 qui mentionnent *@Alice*
- Q13. le texte des tweets des utilisateurs inscrits le 02/03/15
- Q14. le tweet le plus retweeté entre le 25/08/21 et le 02/09/21.

4 Comparaison au modèle relationnel

Comparez et discutez les réponses aux questions précédentes utilisant le langage Cypher à celles présentées ci-dessous en SQL. Pour rappel, le modèle relationnel considéré est le suivant :

```
User (idUser, screenName, name, description, createdAt, url, location, lang)
Tweet (idTweet, text, createdAt, urlEnd, source, lang, nbFavorites, #idUser)
Tweet_Url (#idTweet url, indiceStart, indiceEnd, urlExpanded)
Tweet_Media(#idTweet, idMedia, indiceStart, indiceEnd, type, url, urlExpanded, urlMedia)
Tweet_Hashtag(#idTweet, hahstag, indiceStart, indiceEnd)
Tweet_Mention(#idTweet, #idUser, indiceStart, indiceEnd)
User_Follow(#iduser, #idUserFollow)
Tweet_retweet(#idTweet, #idUser, idRetweet, createdAt, urlEnd, source)
```

[Q1] `select count(*) from User;`

[Q2] `select count(*) from Tweet_Hashtag
where hashtag='WhiteRabbit';`

[Q3] `select count(distinct idUser)
from Tweet T, Tweet_Hashtag TH
where T.idTweet = TH.idTweet
and hashtag='AliceInWonderland';`

[Q4] `select U2.screenName
from User U1, User U2, User_Follow UF
where U1.screenName='Alice'
and U1.iduser = UF.idUser
and UF.idUserFollow = U2.idUser;`

[Q5] `select U2.screenName
from User U1, User U2, User_Follow UF
where U1.screenName='Alice'
and U1.iduser = UF.idUserFollow
and UF.idUser = U2.idUser;`

[Q6] `select screenName
from User, User_Follow UF
where U.idUser = UF.idUser
group by U.idUser, screenName
having count(*)> 10;`

[Q7] `select U2.screenName
from User U1, User U2, User_Follow UF
where U1.screenName='Alice'
and U1.iduser = UF.idUser
and UF.idUserFollow = U2.idUser`

```
intersect
select U2.screenName
from User U1, User U2, User_Follow UF
where U1.screenName='Alice'
and U1.iduser = UF.idUserFollow
and UF.idUser = U2.idUser;
```

```
[Q8] select *
      from (select *
            from tweet
            order by nbFavorites desc)
      where rownum <11;
```

```
[Q9] select * from (select hashtag
                    from Tweet_Hashtag
                    group by hashtag
                    order by count(*) desc)
      where rownum <11;
```

```
[Q10] select U5.screenName
        from User U1, User U4,
        User_Follow UF1, User_Follow UF2, User_Follow UF3, User_Follow UF4,
        where U1.screenName='Alice'
        and U1.iduser = UF1.idUser
        and UF1.idUserFollow = UF2.idUser
        and UF2.idUserFollow = UF3.idUser
        and UF3.idUserFollow = UF4.idUser
        and UF4.idUserFollow = U4.idUser;
```

[Q11]

Similaire à [Q10] mais non généralisable à un degré quelconque en utilisant seulement SQL...