

BANCO DE DADOS NoSQL – 2024/2025

DICAS 1: Dicas nos comandos para o trabalho

1. Faça a separação dos nós de mensagem que contém o campo `referenced_tweets` dos que não tem. As mensagens que não possuem este campo são chamadas de mensagens originais e a hashtag estará presente em todas estas mensagens, sem exceção.
2. Ao criar os nós com esta propriedade/atributo (`referenced_tweets`), se você usar um `UNWIND` (`UNWIND tweet.referenced_tweets as referenced_tweets`) todas as mensagens que não tiverem este campo serão ignoradas e os nós não serão criados. Cuidado para não deixar estes nós de fora do seu banco, pois são os mais importantes para responder à questão 02.
3. Sugiro usar o comando `FOREACH` no lugar de `UNWIND` em `referenced_tweets` e hashtags, para evitar o problema mencionado acima. Exemplo:

No lugar de:

```
CALL apoc.load.directory('*.*json')
YIELD value
WITH value as arquivos
ORDER BY arquivos DESC
CALL apoc.load.json(arquivos)
YIELD value
UNWIND value.data AS tweet
UNWIND tweet.referenced_tweets as referenced_tweets
UNWIND tweet.entities.hashtags AS hashtags
MERGE (t:Tweet {tweet_id: tweet.id})
ON CREATE SET ...
MERGE (h:Hashtag {tag:
apoc.text.replace(apoc.text.clean(hashtags.tag), '^a-zA-Z0-9', '')})
...
```

Fazer:

```
CALL apoc.load.directory('*.*json') YIELD value
WITH value as arquivos
CALL apoc.load.json(arquivos)
YIELD value
UNWIND value.data AS tweet
MERGE (t:Tuite {id_tuite: tweet.id})
ON CREATE SET t += {
texto: tweet.text,
...
...
...
}
FOREACH (hashtags IN tweet.entities.hashtags |
MERGE (h:Hashtag {hashtag:
apoc.text.replace(apoc.text.clean(hashtags.tag),'^a-zA-Z0-9', ''}))
MERGE (h)<-[:POSSUI]-(t)
)
FOREACH (ref_tweet IN tweet.referenced_tweets |
SET t.tipo_ref = coalesce(t.tipo_ref, []) + [ref_tweet.type], t.id_ref
= coalesce(t.id_ref, []) + [ref_tweet.id]
)
...
```

4. Outra dica é usar os comandos de filtragem de string para normalizar as hashtags, como mostrado acima:

```
apoc.text.replace(apoc.text.clean(hashtags.tag),'^a-zA-Z0-9', '')
```

5. Usar o coalesce para elementos que estão em lista no arquivo JSON fará com que você recrie a lista como um atributo de forma automática:

```
coalesce(t.tipo_ref, []) + [ref_tweet.type], t.id_ref =
coalesce(t.id_ref, []) + [ref_tweet.id]
```

6. Crie seus nós e relacionamentos em um único comando e use estes mesmos nós criados nas duas questões. Não é correto termos que importar e recriar o banco de

dados inteiro sempre que quisermos encontrar alguma informação. Criamos uma vez e usamos os dados no banco e não mais nos arquivos JSON.

7. Se o comando

```
CALL apoc.load.directory('*.json') YIELD value
```

resultar em erro de que o comando não existe, mesmo após outros comandos APOC terem funcionado significa que você está usando o DBMS em uma versão acima da 5, na qual a biblioteca APOC ainda não foi completamente incorporada. Para resolver, ou não use ele e use uma lista com os nomes dos arquivos para importá-los ou crie um novo DBMS e selecione uma versão 4.* na hora da criação (Isso não é a versão do seu software neo4j e sim versão do banco criado e você pode ter diversos bancos em versões diferentes no seu computador).

8. Se você terminou seus testes de comandos, antes de rodar definitivamente o comando de criação dos nós e arestas, apague tudo o que estiver no banco, para não influenciar nas demais etapas de forma a induzir algum erro:

```
MATCH (n) DETACH DELETE n;
```

9. Para as questões 02 e 03 parte I, mostre apenas os comandos MATCH/RETURN feitos e não os comandos de criação do banco. Estes comandos de criação deverão ser colocados apenas na parte I da questão 01.

10. Para diferenciar o que é nó de mensagem original do que é nó de mensagem de citação, resposta ou retweet, faça os seguintes comandos:

```
// Com o banco de dados criado, alterar a etiqueta de identificação dos nós de Tuite para as mensagens que são de retweeted, quoted e replied_to:  
//Comando 1: Retuites:  
MATCH (t)  
WHERE "retweeted" IN t.tipo_ref //busca a informação de tipo_ref em todos os nós e se tiver e for "retweeted", altera a etiqueta do nó de Tuite para Retuite, permitindo a separação do que é mensagem original do restante.  
REMOVE t:Tweet  
SET t:Retuite;
```

```
//Comando 2: Replied_to:
MATCH (t)
WHERE "replied_to" IN t.tipo_ref //busca a informação de tipo_ref em
todos os nós e se tiver e for "replied_to", altera a etiqueta do nó
de Tuite para Resposta_para, permitindo a separação do que é
mensagem original do restante.
REMOVE t:Tweet
SET t:Resposta_para;

//Comando 3: Quoted:
MATCH (t)
WHERE "quoted" IN t.tipo_ref //busca a informação de tipo_ref em
todos os nós e se tiver e for "quoted", altera a etiqueta do nó de
Tuite para Citacao, permitindo a separação do que é mensagem
original do restante.
REMOVE t:Tweet
SET t:Citacao;
```

Estes comandos permitirão encontrar a resposta da questão 02 mais facilmente (a hashtag principal está presente em todas as mensagens originais, não analise mensagens de retweet, pois elas deveriam ser cópias das mensagens originais e, apesar de sofrerem alterações por alguns usuários, herdam todas as hashtags e texto das mensagens originais).