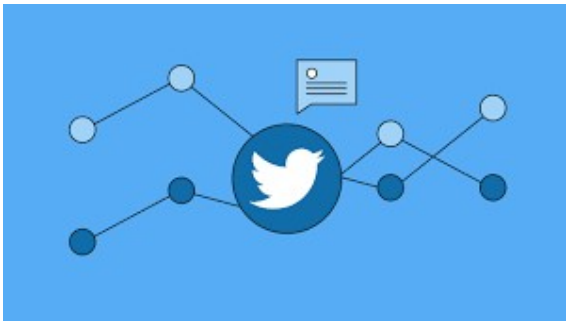
REPORT

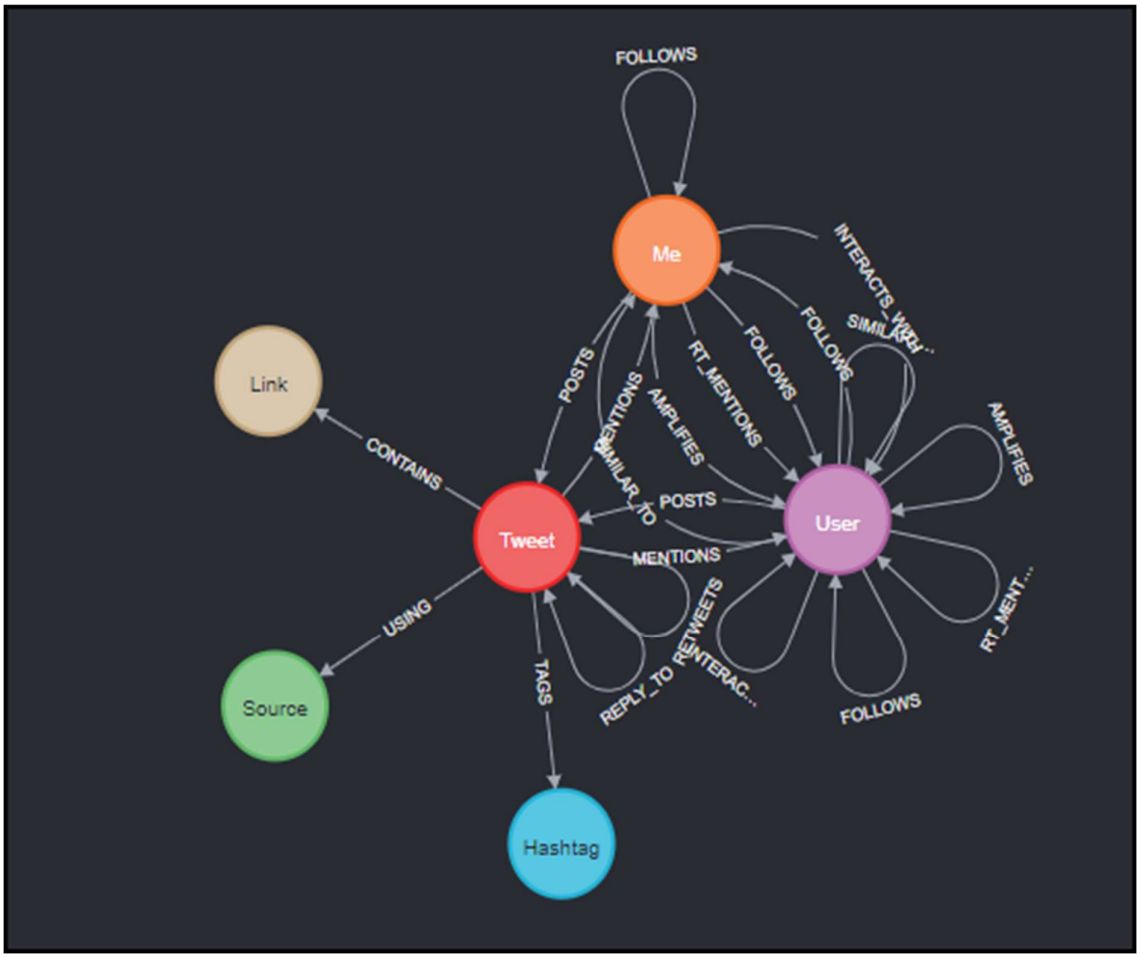
NAME : S M SUTHARSAN RAJ

PROJECT TITLE : TWITTER DATABASE – SOCIAL NETWORK ANALYSIS



0. INTRODUCTION TO THE DATASET

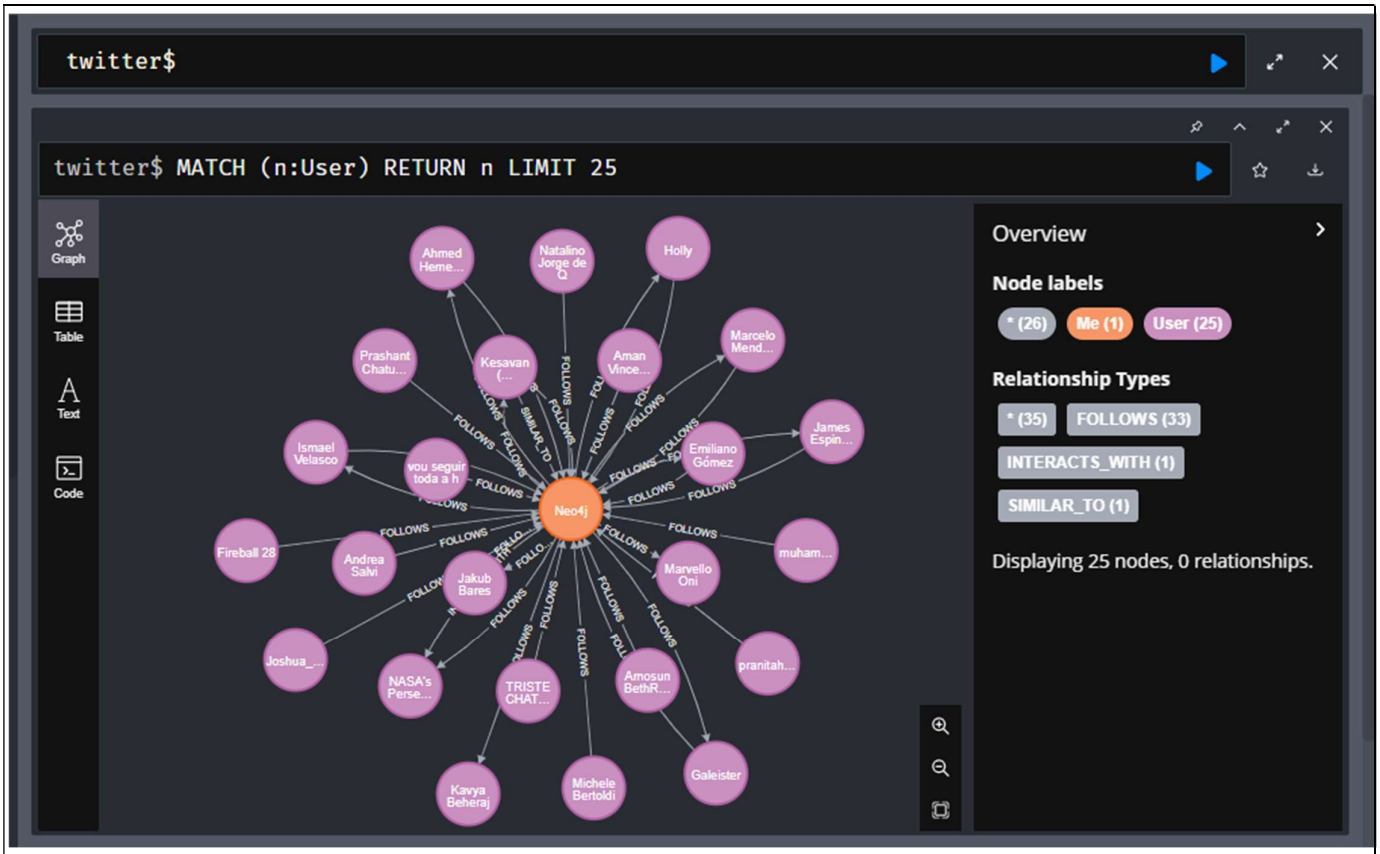
The twitter dataset used here is the one that is hosted on the neo4j server. It has many nodes and relations which pertains to the following diagram. We will display a limited size for the output to look clear and concise.



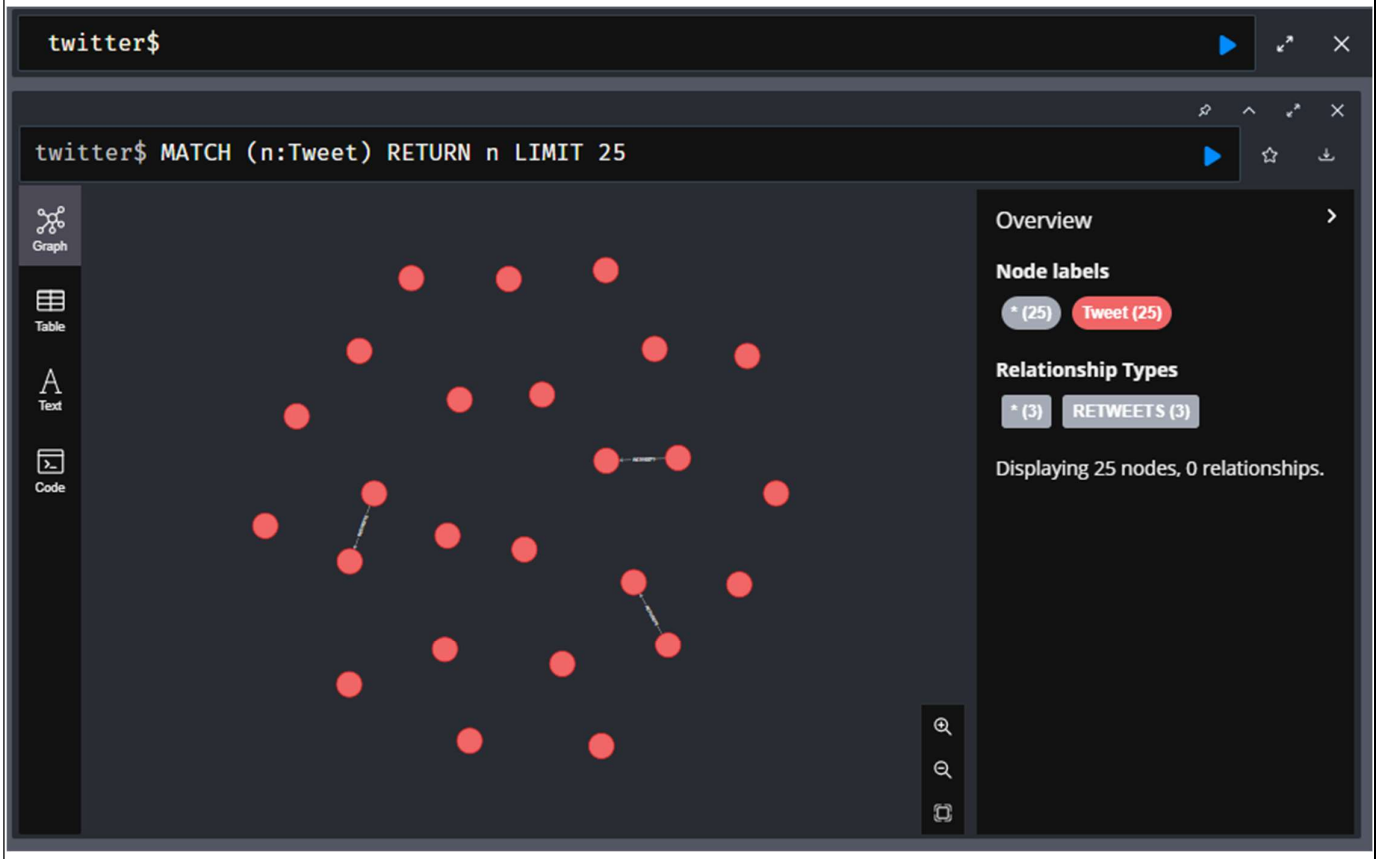
1. CREATE A GRAPH DATABASE FROM THE DATABASE :

LISTING THE NODES :-

- USER NODE

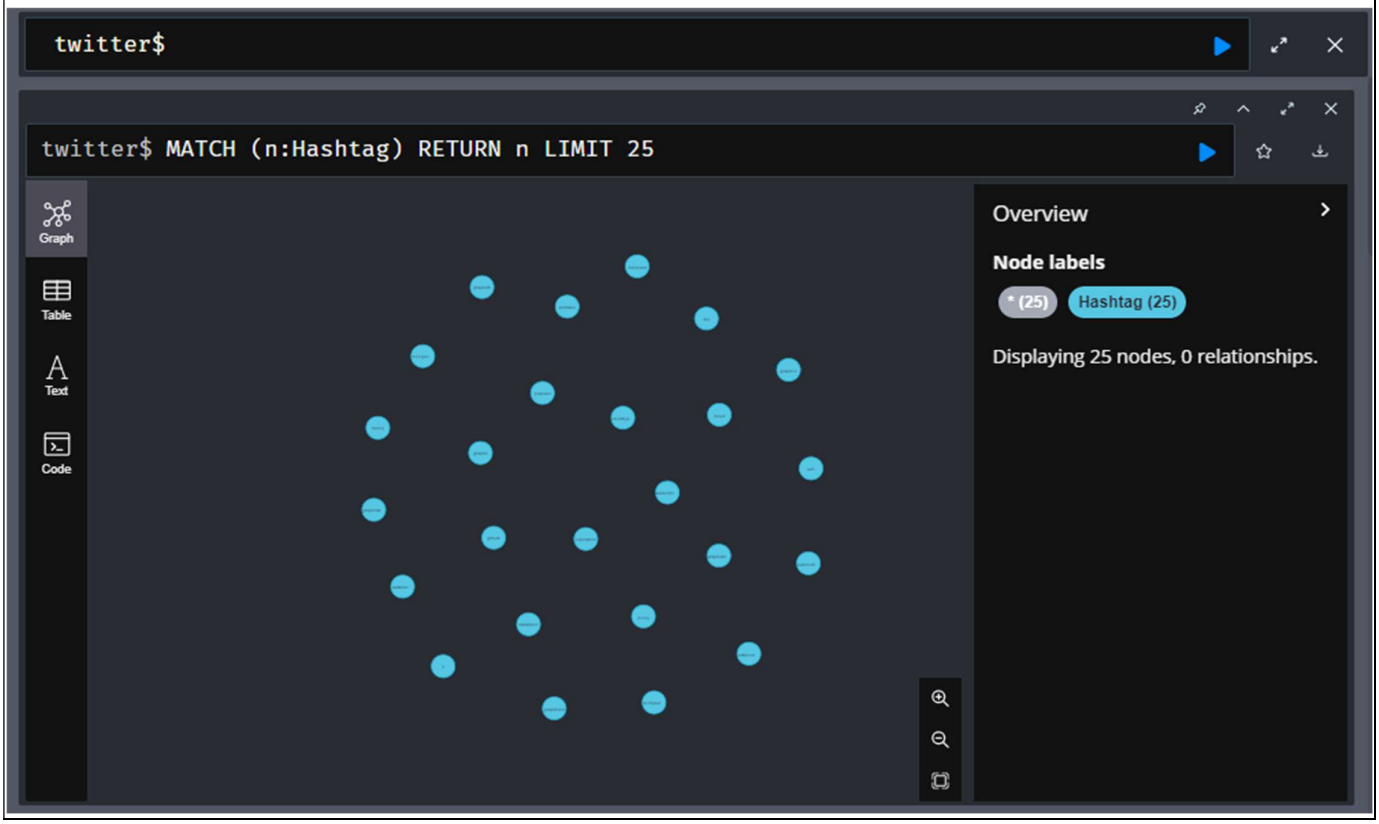

twitter$
twitter$ MATCH (n:User) RETURN n LIMIT 25
Graph
Table
Text
Code
Overview
Node labels
* (26)
Me (1)
User (25)
Relationship Types
* (35)
FOLLOWS (33)
INTERACTS_WITH (1)
SIMILAR_TO (1)
Displaying 25 nodes, 0 relationships.
Neo4j
Ahmed Heme...
Natalino Jorge de O...
Holly
Marcelo Mend...
Prashant Chatu...
Kesavan (...
Aman Vince...
James Espin...
Ismael Velasco
vou seguir toda a h
Emiliano Gómez
Fireball 28
Andrea Salvi
Jakub Bares
Marvello Oni
muham...
Joshua_...
NASA's Perse...
TRISTE CHAT...
Amosun BethR...
pranitah...
Kavya Beheraj
Michele Bertoldi
Galeister
FOLLOWS
SIMILAR_TO

- TWEET NODE

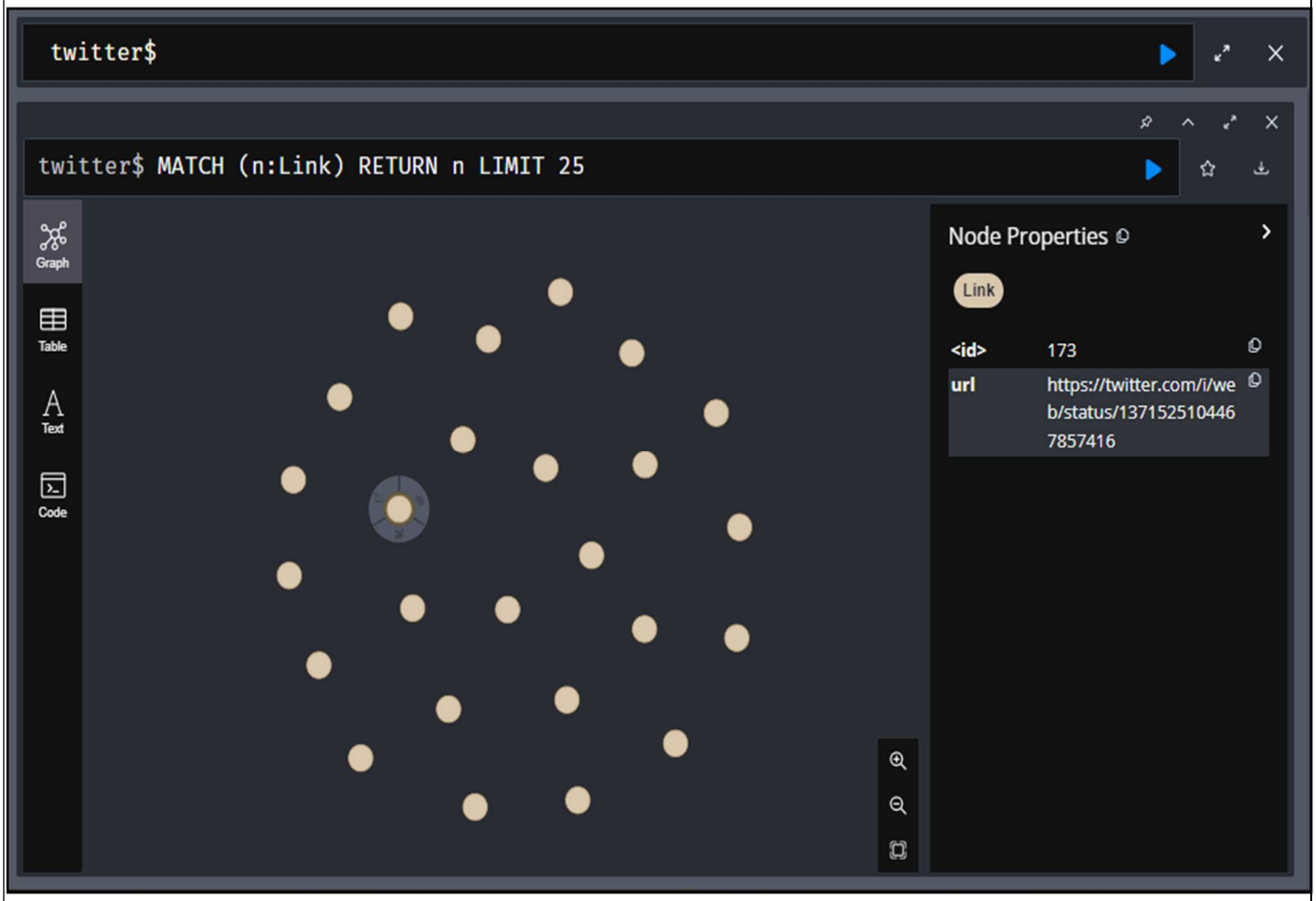

twitter$
twitter$ MATCH (n:Tweet) RETURN n LIMIT 25
Graph
Table
Text
Code
Overview
Node labels
* (25)
Tweet (25)
Relationship Types
* (3)
RETWEETS (3)
Displaying 25 nodes, 0 relationships.

- SOURCE NODE

- HASHTAG NODE

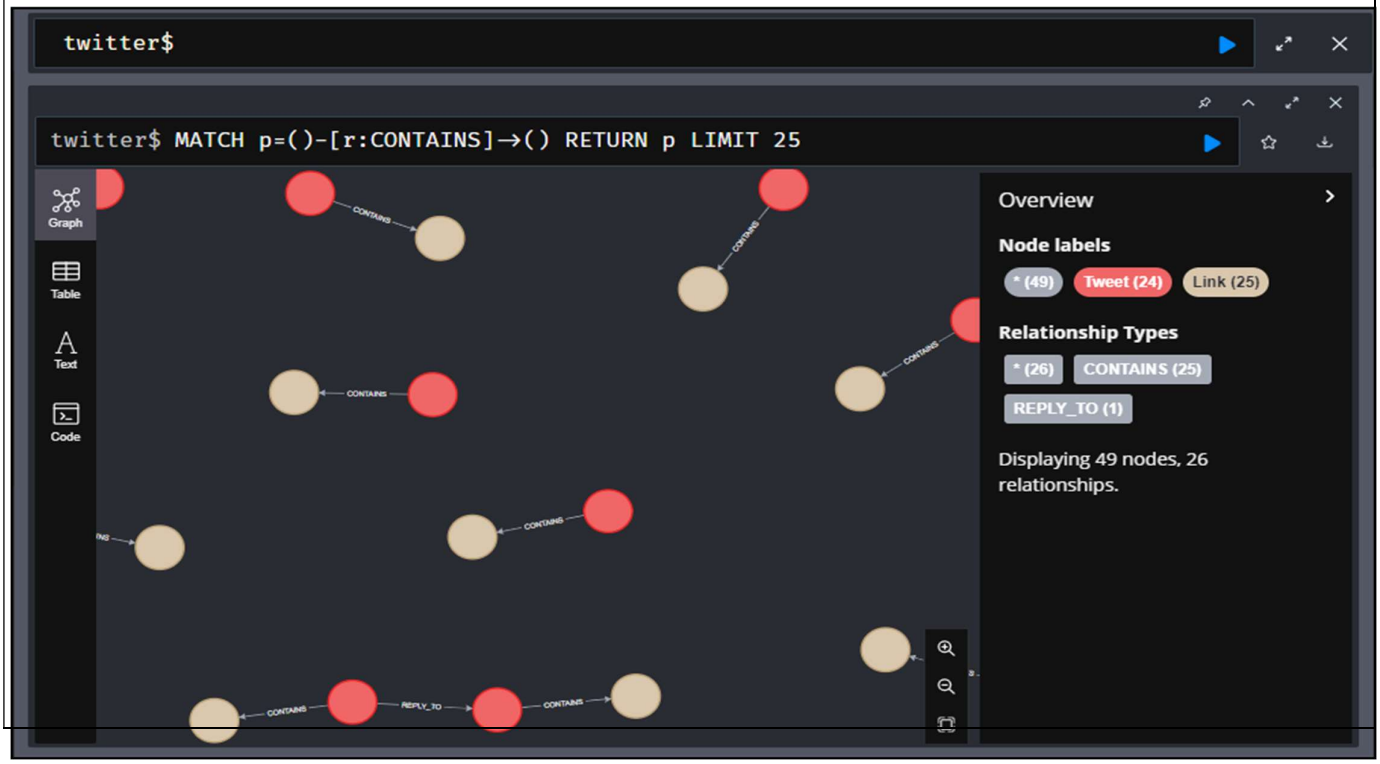


- LINK NODE

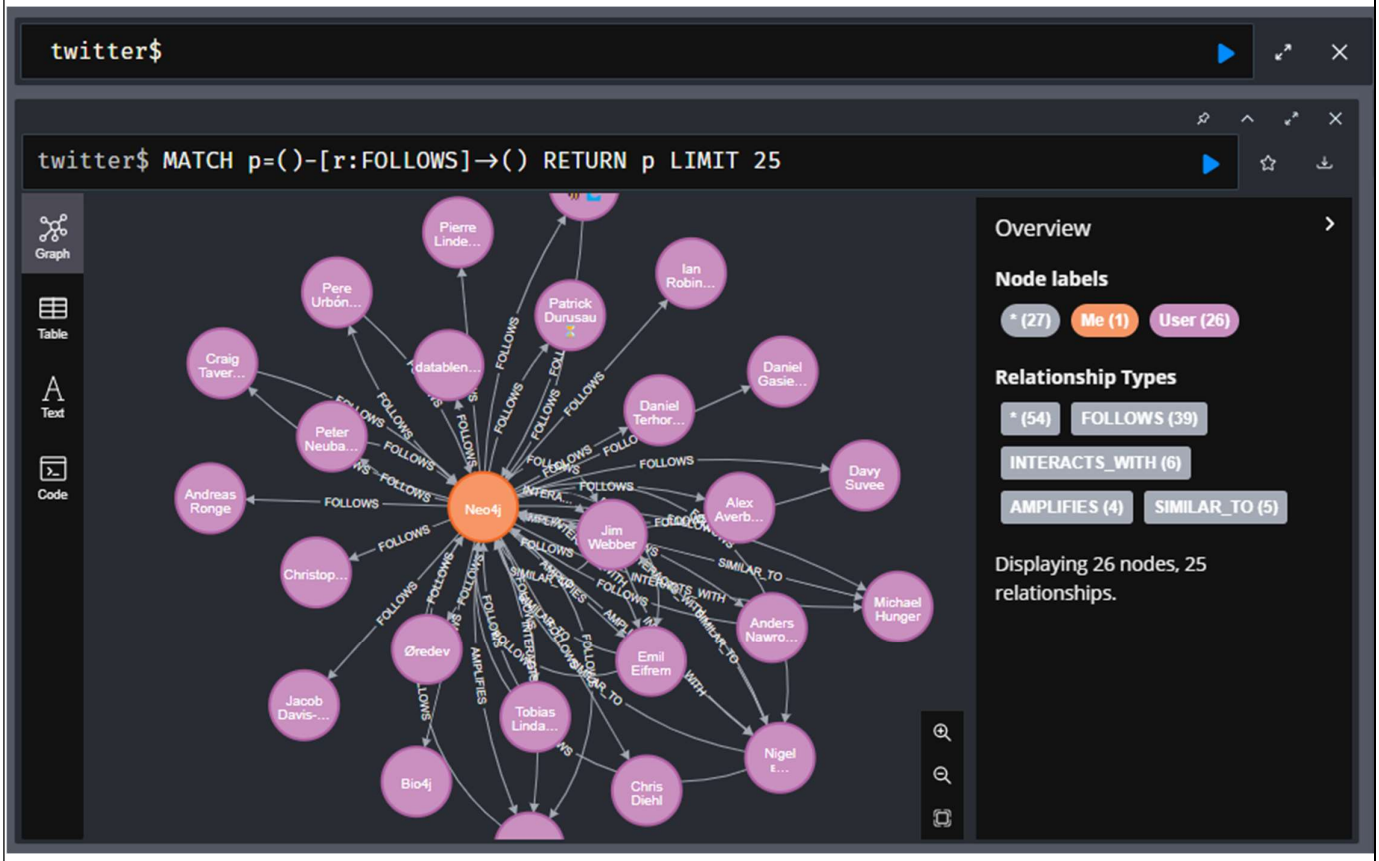


LISTING THE RELATIONS :-

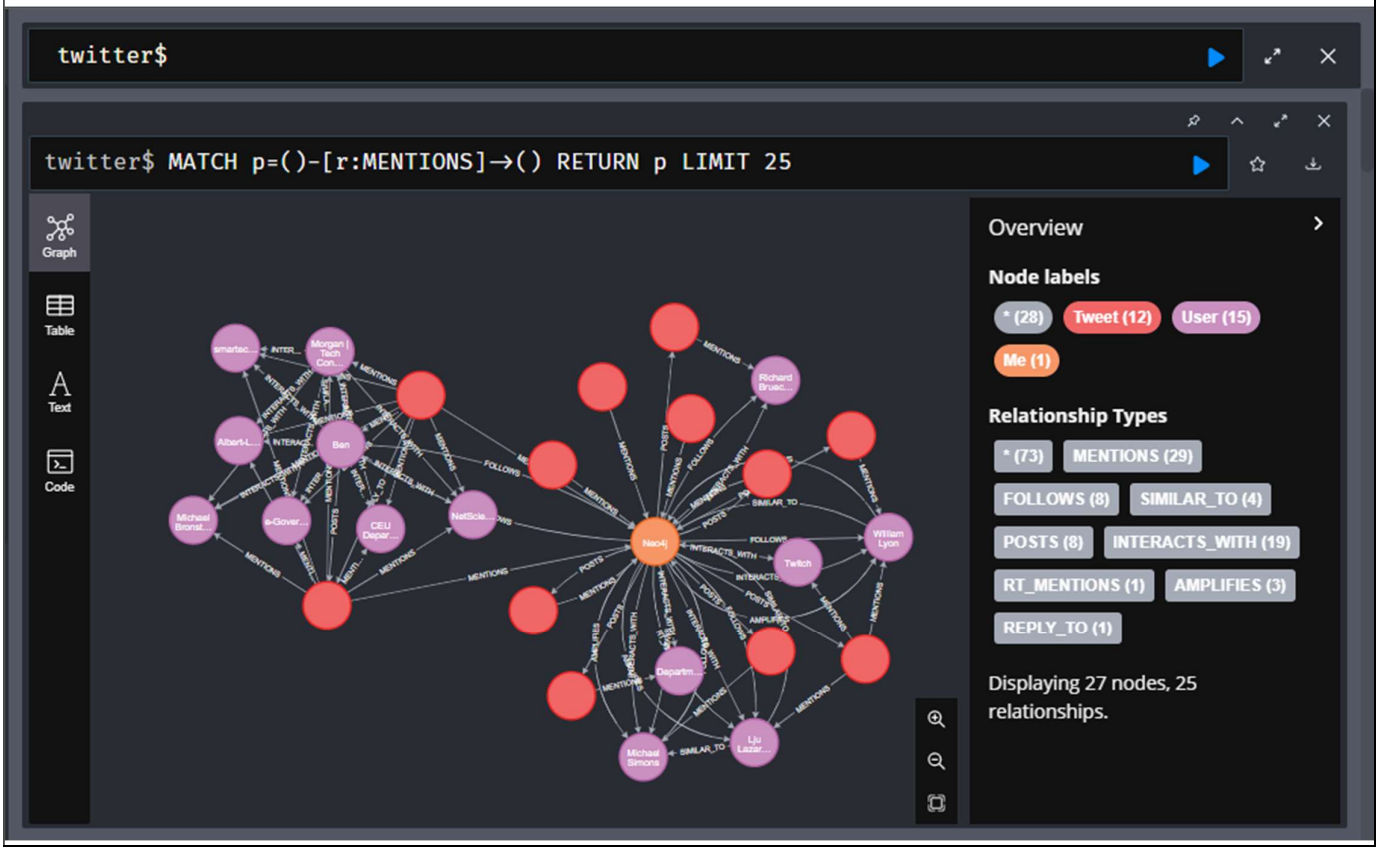
- CONTAINS



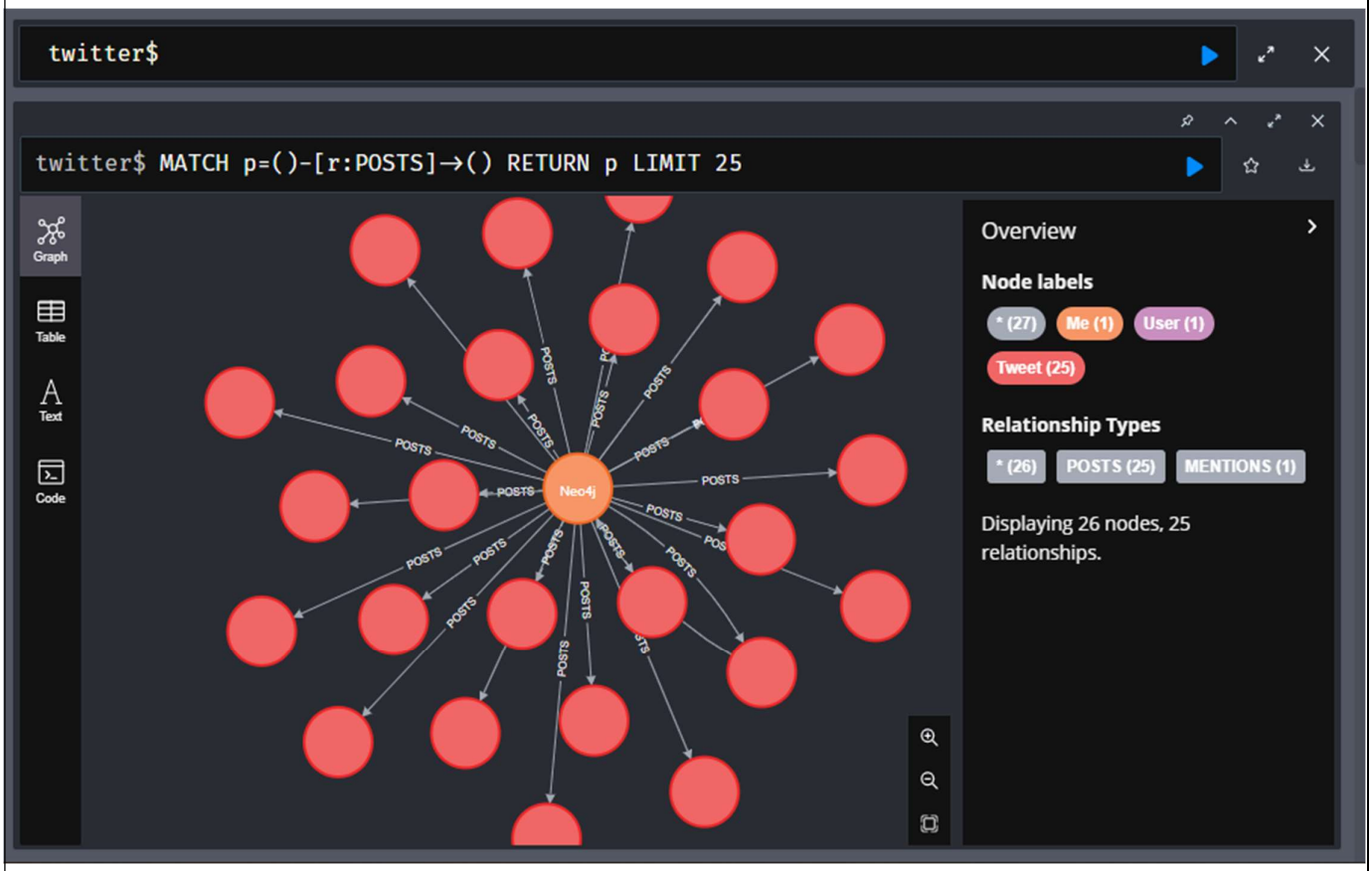
- FOLLOWS



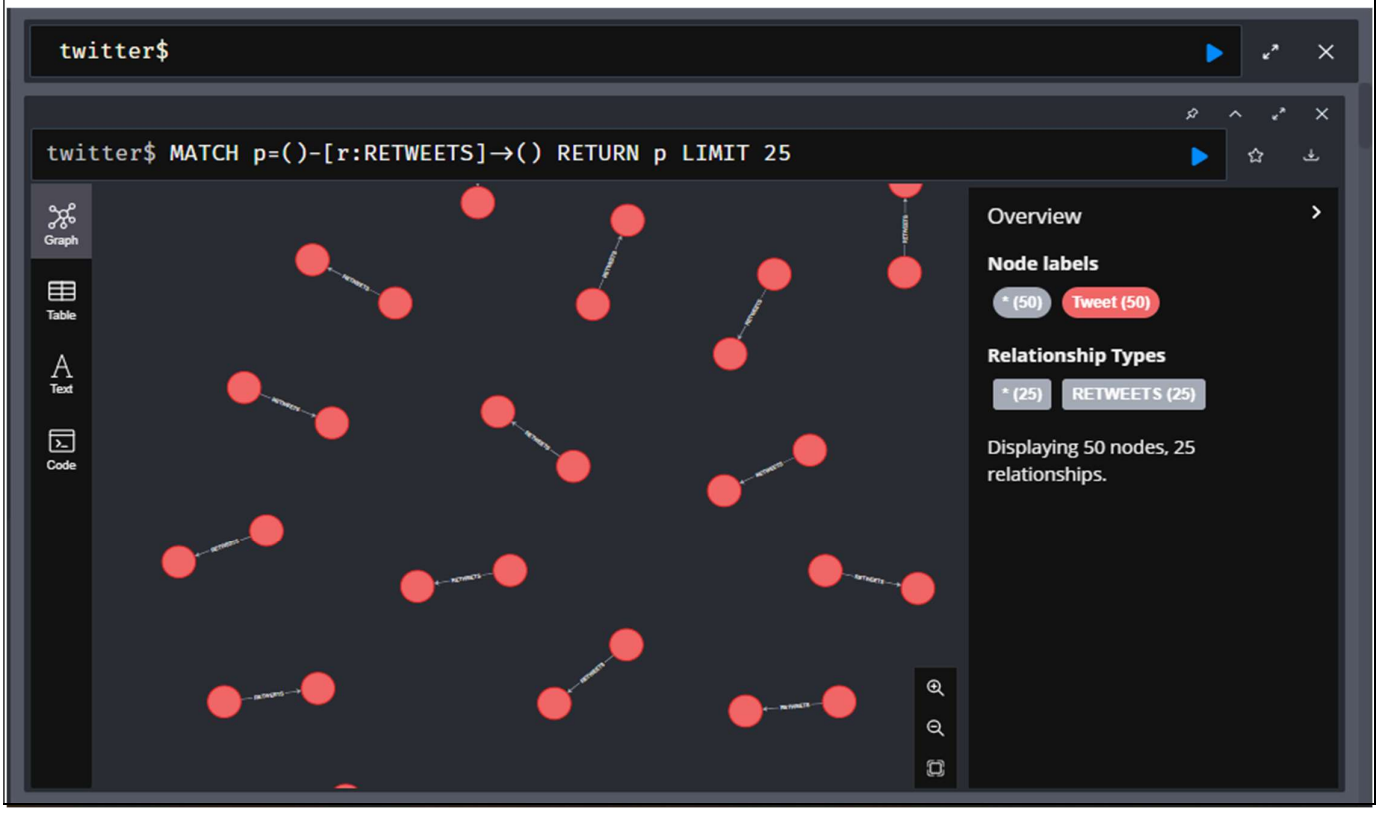
- MENTIONS



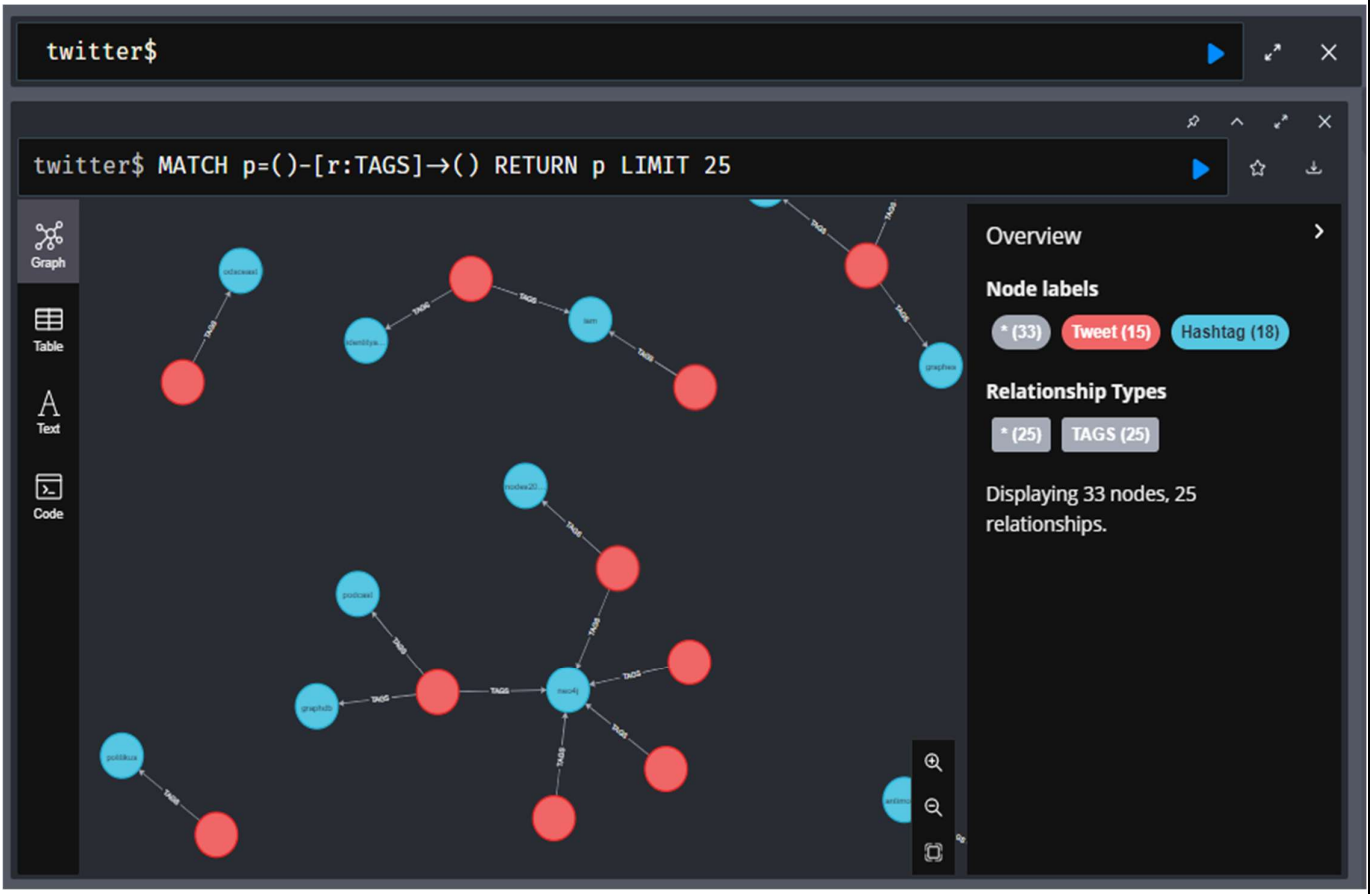
- POSTS



- RETWEETS



- TAGS



2. COMMANDS : <u>CONTENT/ COLLABORATIVE</u> BASED FILTERING :

a. Which platform are users tweeting from most often? : **CONTENT**

b. Which hashtags co-occur with #python most frequently? **CONTENT**

twitter$

```
MATCH (:Hashtag {name:'python'})←[:TAGS]-(:Tweet)-[:TAGS]→(h:Hashtag)
RETURN h.name AS Hashtag, COUNT(*) AS Count
ORDER BY Count DESC
LIMIT 5
```

Table | Text | Code

	Hashtag	Count
1	"neo4j"	6
2	"graph"	2
3	"py2neo"	2
4	"wordnet"	2
5	"graphdatabase"	1

Started streaming 5 records after 6 ms and completed after 7 ms.

c. Which tweet has been retweeted the most, and who posted it? **COLLABORATIVE**

twitter$

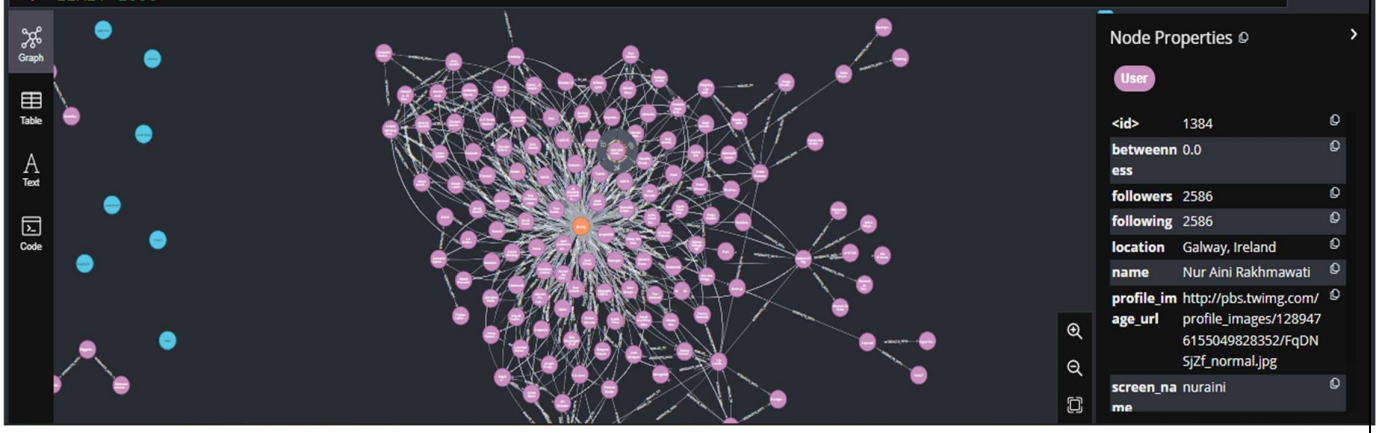
```
MATCH (:Tweet)-[:RETWEETS]→(t:Tweet)
WITH t, COUNT(*) AS Retweets
ORDER BY Retweets DESC
LIMIT 10
MATCH (u:User)-[:POSTS]→(t)
RETURN u.screen_name AS User, t.text AS Tweet, Retweets
```

Table | Text | Code

	User	Tweet
1	"ElLazal"	*null*
2	"lyonwj"	*null*
3	"rvanbruggen"	*null*
4	"kojiannoura"	*null*
5	"AlessandroNegro"	"Today is my day! Deal of the Day March 14: Half off my book Graph-Powered Machine Learning and selected title

d. Entire graph **: MISCELLANEOUS**

```
MATCH (t:Tweet)
WITH t ORDER BY t.id DESC LIMIT 2000
MATCH (user:User)-[:POSTS]→(t)-[:TAGS]→(tag:Hashtag)
MATCH (t)-[:MENTIONS]→(user2:User)
UNWIND [tag,user2] as other WITH distinct user,other
RETURN { from: {id:id(user),label: head(labels(user)), data: user},
    rel: 'CONNECTS',
    to: {id: id(other), label: head(labels(other)), data: other}} as tuple
LIMIT 1000
```



e. Links from interesting retweets : **CONTENT**

```
twitter$
```

Started streaming 10 records after 6 ms and completed after 11 ms.

```
MATCH (:User {screen_name: 'neo4j'})-[:POSTS]→
  (t:Tweet)-[:RETWEETS]→(rt)-[:CONTAINS]→(link:Link)
RETURN t.id_str AS tweet, link.url AS url, rt.favorites AS favorites
ORDER BY favorites DESC LIMIT 10
```

Table | Text | Code

	tweet	url	favorites
1	"1354525606583660549"	"https://twitter.com/i/web/status/1354483413399396353"	58
2	"1356792215306125316"	"https://twitter.com/i/web/status/1356665707220590594"	47
3	"1356792215306125316"	"https://arrows.app/"	47
4	"1350102888039772160"	"https://twitter.com/i/web/status/1350049048863121409"	36
5	"1362720758909329408"	"https://twitter.com/i/web/status/1362086924077436931"	36

f. Most tagged by an user : **COLLABORATIVE**

```
twitter$
  (h:Hashtag)←[:TAGS]-(t:Tweet)←[:POSTS]-(u:User {screen_name:'neo4j'})
WITH
  h, COUNT(h) AS Hashtags
ORDER BY
  Hashtags DESC
LIMIT 10
RETURN
  h.name, Hashtags
```

Table | Text | Code

	h.name	Hashtags
2	"graphdatabases"	75
3	"twin4j"	68
4	"graphcast"	34
5	"graphtechnology"	29
6	"graphdatabase"	20

Started streaming 10 records after 6 ms and completed after 11 ms.

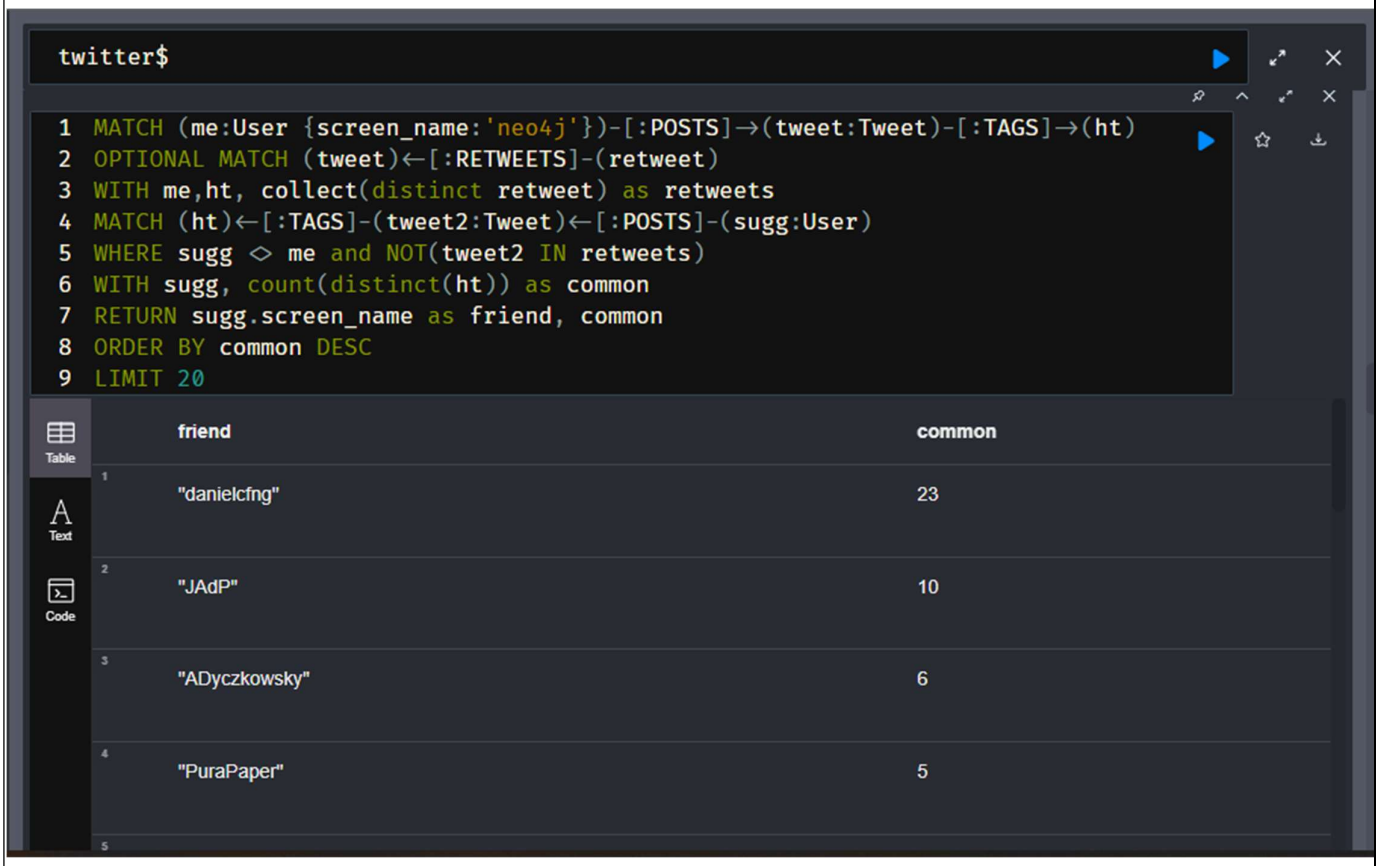
g. Users tweeting an user, but not following : **COLLABORATIVE**

```
twitter$
MATCH (ou:User)-[:POSTS]→(t:Tweet)-[mt:MENTIONS]→(me:User {screen_name:
'neo4j'})
WITH DISTINCT ou, me
WHERE (ou)-[:FOLLOWS]→(me)
AND NOT (me)-[:FOLLOWS]→(ou)
RETURN ou.screen_name
LIMIT 20
```

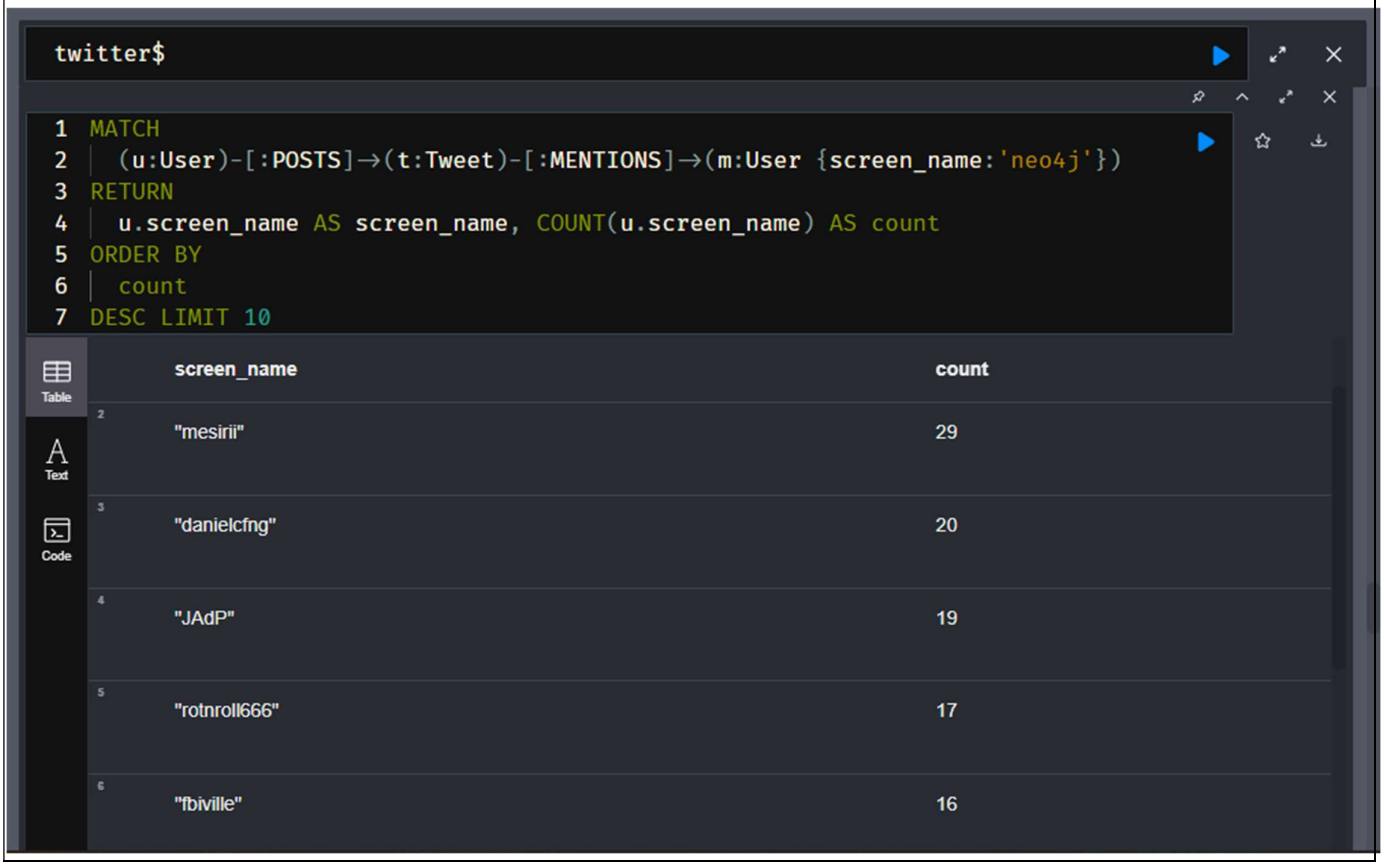
Table | Text | Code

	ou.screen_name
1	"drlynnchiu"
2	"jscarp"
3	"jughh"
4	"danfeindanfein"
5	"madjid_net"

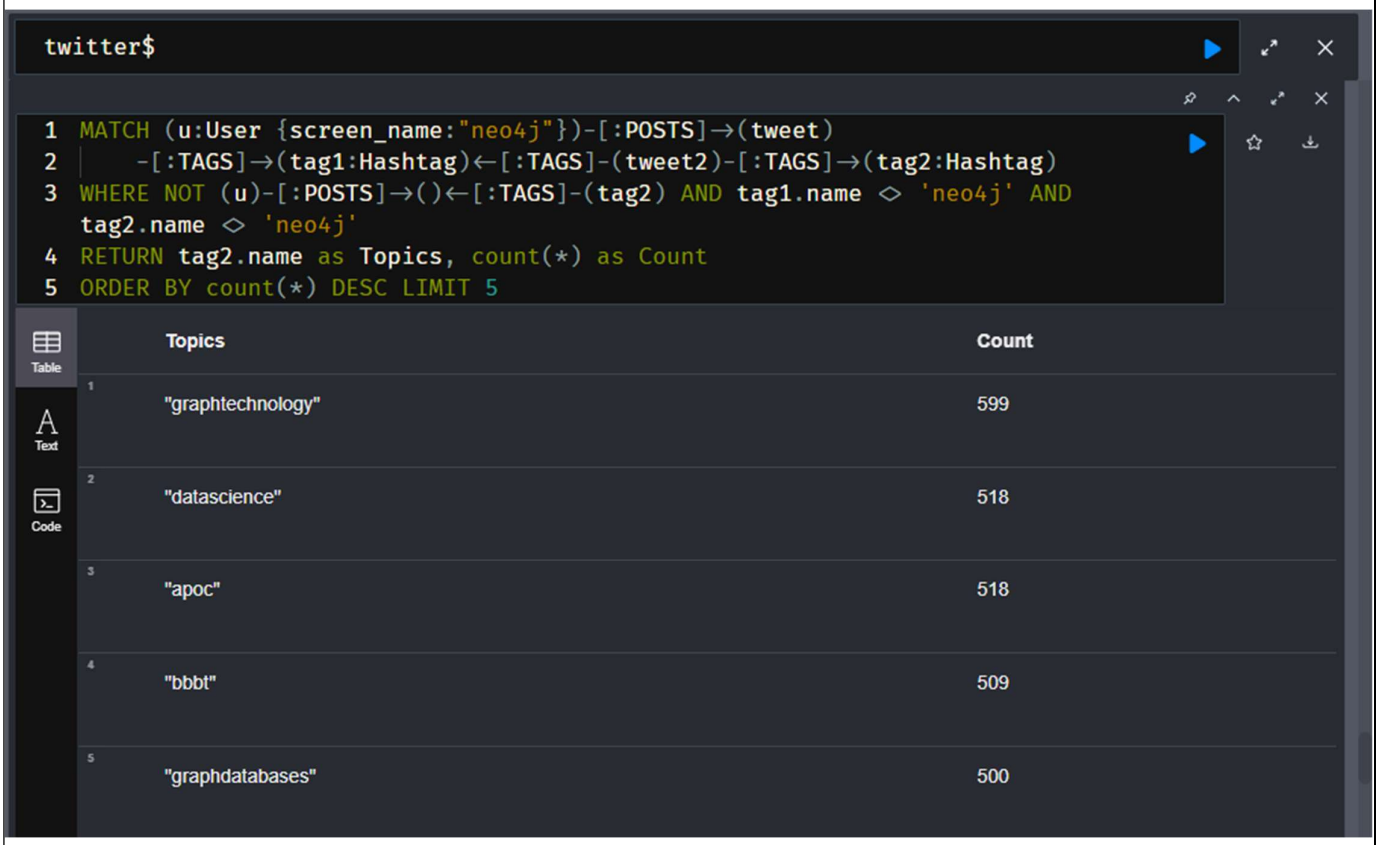
h. Users tweeting with common tags as an user : **COLLABORATIVE**



i. Top mentions of an User : **CONTENT**



j. Which other topics could we recommend for a specific user? Finding the most frequently co-occurring topics to the ones they used and that they haven't used themselves. : **CONTENT**



3. CENTRALITY MEASURE

```
twitter$
```

```
1 CALL gds.pageRank.stream('graph')
2 YIELD nodeId, score
3 RETURN gds.util.asNode(nodeId).screen_name as screenName,
4 score, gds.util.asNode(nodeId).followers as followersCount
5 ORDER BY score DESC LIMIT 10;
```

Table | Text | Code

	screenName	score	followersCount
1	"neo4j"	7832.045558041402	34507
2	"mendonca2709"	0.8073490677802425	61
3	"galeister"	0.8073490677802425	35
4	"jamesejr7"	0.8073490677802425	1025
5	"AngeliusAngel"	0.8073490677802425	170
6	"emilianogomez33"	0.8073490677802425	113

4. COMMUNITY DETECTION

```
twitter$
```

```
1 CALL gds.louvain.stream('graph')
2 YIELD nodeId, communityId
3 WITH communityId AS communityId, size(collect(nodeId)) as size
4 RETURN communityId, size
5 ORDER BY size DESC;
```

Table | Text | Code

	communityId	size
1	1	34507
2	5	2
3	24	1
4	65	1
5	75	1
6		

5. ANALYSIS/ INFERENCE :-

So, we had taken a graph of more than 40,000 users and had analysed the twitter database. We see the recommendations and various types of filtering on different criteria. Accordingly, we see the results. Most of the test was done on the user : 'neo4j' as this user had various quantity to deal with. Also we see centrality measures and community detection where only a single community (id : 1) had the highest users. So, it was majority community tweeting here. Other communities are just a single membered.
Also all the queries worked with the above relations and nodes which also makes this graph database versatile. So this was the complete Social Network Analysis on the database.

6. CONCLUSION :-

So this was the Social Network Analysis on the Twitter database all about. Future works could include, user suggestions based on locality, betweenness measures, twitter posts which are spam, likes and dislikes by the user and various others factors.

THANK YOU