Graph Analytics

Modeling Chat Data using a Graph Data Model

Use Neo4j graphics to model chat data and perform analysis of relationships between teams, users, and chats.

Creation of the Graph Database for Chats

Describe the steps you took for creating the graph database. As part of these steps

i) Write the schema of the 6 CSV files
ii) Explain the loading process and include a sample LOAD command
iii) Present a screenshot of some part of the graph you have generated. The graphs must include clearly visible examples of most node and edge types. Below are two acceptable examples. The first example is a rendered in the default Neo4j distribution, the second has had some nodes moved to expose the edges more clearly. Both include examples of most node and edge types.

Chat_create_team_chat.csv : when a user create a team chat session

Chat_join_team_chat.csv : when a user joins a team chat session

Chat_leave_team_chat.csv : when a user leaves a team chat session

Chat_item_team_chat.csv : when a new chat is created in a team chat session

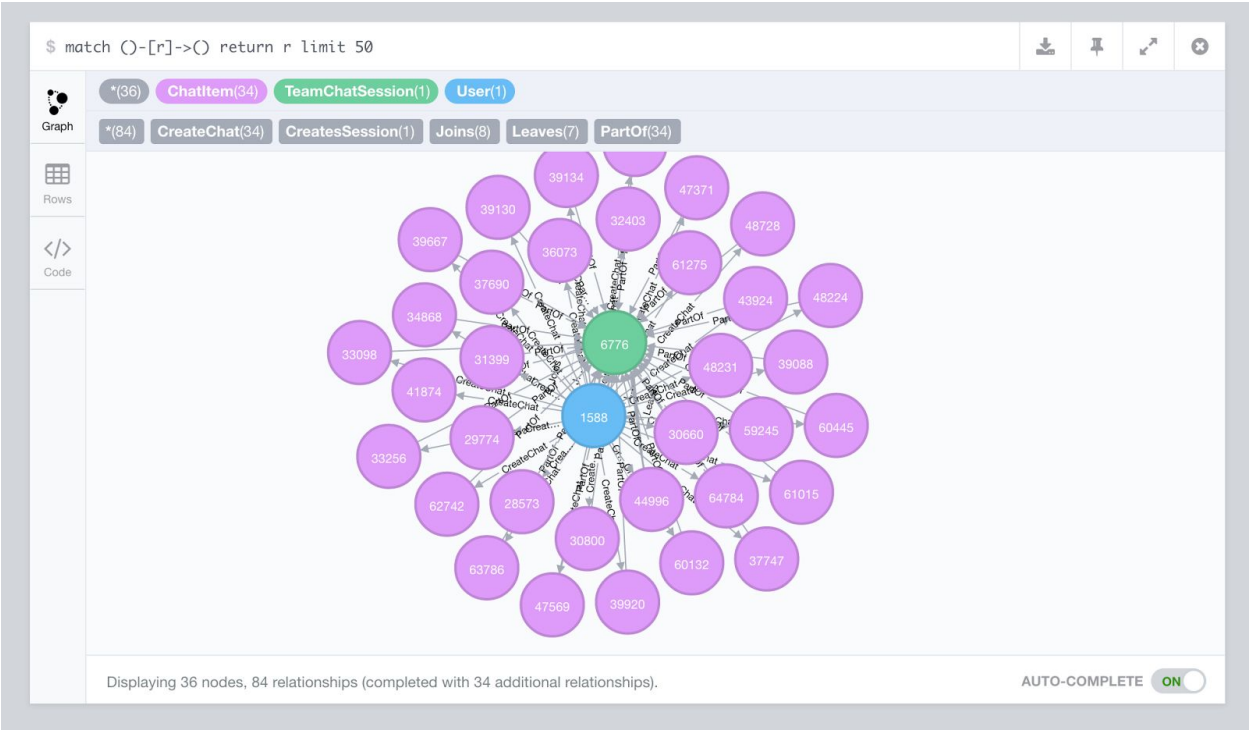
Chat_mention_team_chat.csv : when a user mentions a specific chat item

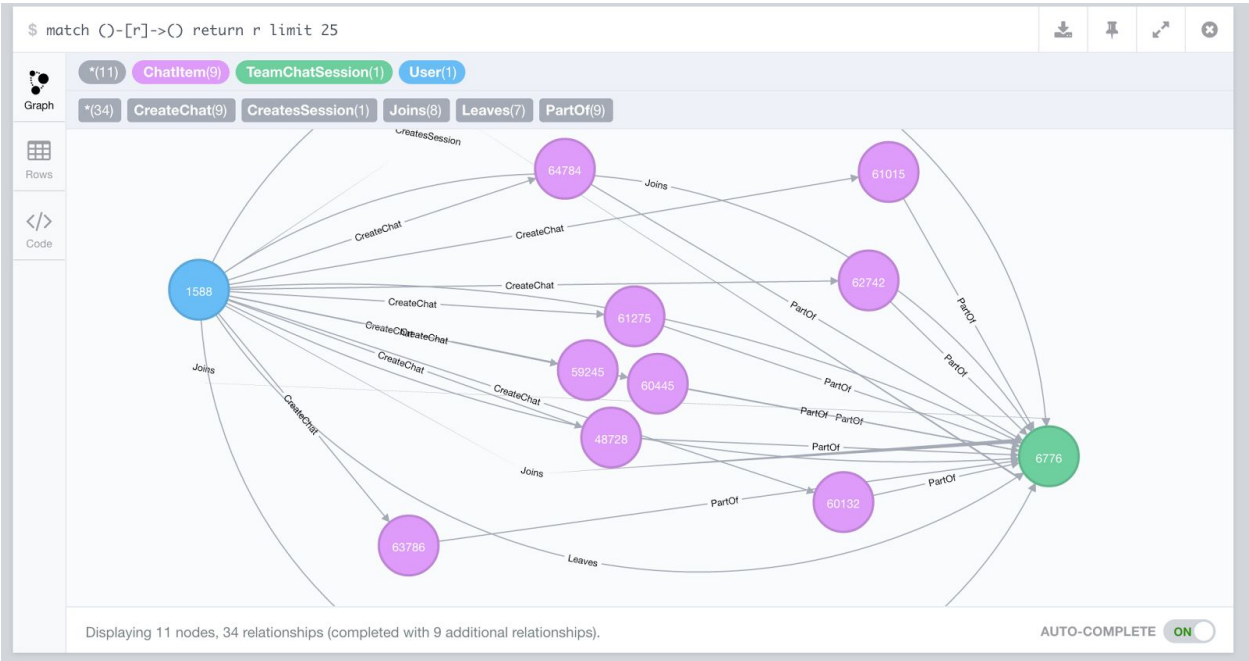
Chat_respond_team_chat.csv : when a user responds a chat to another user

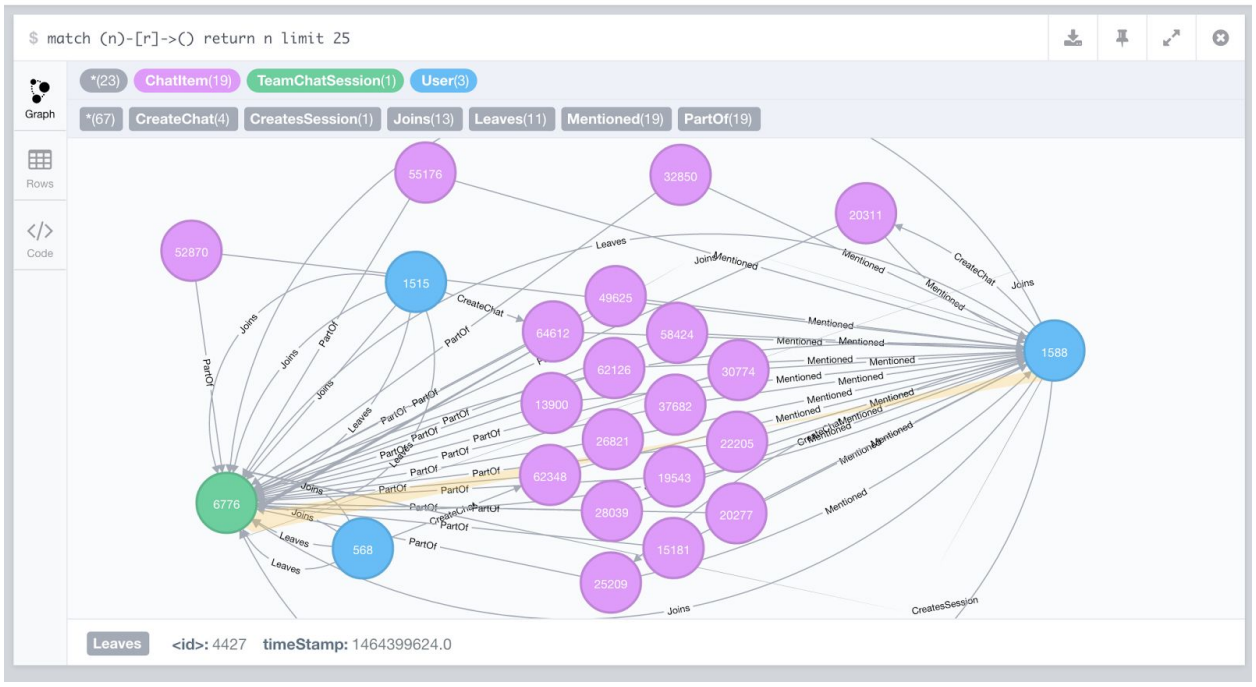
Sample LOAD script:

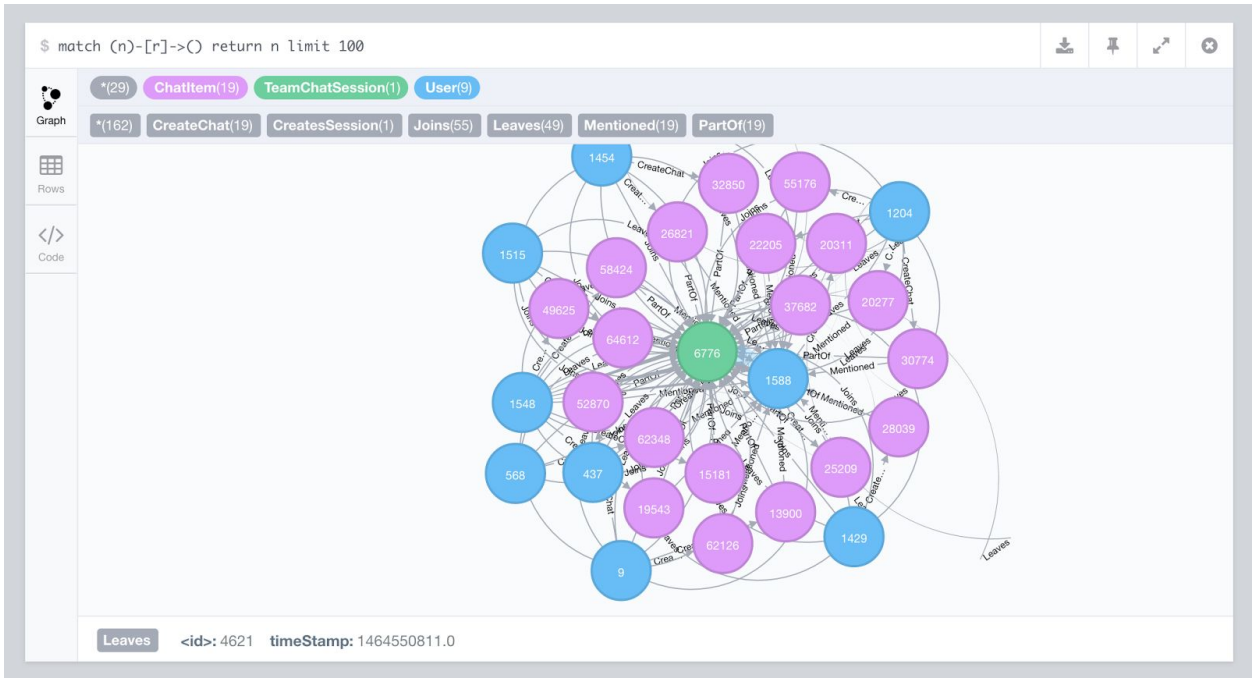
```
LOAD CSV FROM
"file:///Users/EstherChang/Documents/R/Big_Data_Capstone/big_data_capstone_datasets_and_scripts/chat-data/chat_join_team_chat.csv" AS row

MERGE (u:User {id: toInt(row[0])})

MERGE (c:TeamChatSession {id: toInt(row[1])})

MERGE (u)-[:Joins{timeStamp: row[2]}]->(c)
```

Screenshot:


$ match ()-[r]->() return r limit 50
Graph
Rows
Code
*(36)
ChatItem(34)
TeamChatSession(1)
User(1)
*(84)
CreateChat(34)
CreatesSession(1)
Joins(8)
Leaves(7)
PartOf(34)
6776
1588
Displaying 36 nodes, 84 relationships (completed with 34 additional relationships).
AUTO-COMPLETE
ON


$ match ()-[r]->() return r limit 25
Graph
Rows
Code
*(11)
ChatItem(9)
TeamChatSession(1)
User(1)
*(34)
CreateChat(9)
CreatesSession(1)
Joins(8)
Leaves(7)
PartOf(9)
1588
6776
Displaying 11 nodes, 34 relationships (completed with 9 additional relationships).
AUTO-COMPLETE
ON





Finding the longest conversation chain and its participants

Report the results including the length of the conversation (path length) and how many unique users were part of the conversation chain. Describe your steps. Write the query that produces the correct answer.

Longest chain : 10 (directional)

Unique users : 5

STEPS:

```
match p=(i1:ChatItem)-[:ResponseTo*]->(i2:ChatItem)

return length(p)

ORDER BY length(p) DESC limit 1
```

```
match p=(a)-[:ResponseTo*]->(c) where length(p)=10

with p

match (i:ChatItem)<-[r:CreateChat]-(u:User)

where i in nodes(p)

return count(distinct u)
```

This kind of search may be relevant to Eglence Inc. business plan because Eglence Inc. can know what subject attracts more conversations and target it as a new business plan to increase revenue.

Analyzing the relationship between top 10 chattiest users and top 10 chattiest teams

Describe your steps from Question 2. In the process, create the following two tables. You only need to include the top 3 for each table. Identify and report whether any of the chattiest users were part of any of the chattiest teams.

STEPS:

```
match (u)-[:CreateChat*]->(i)

RETURN DISTINCT count(u.id), u.id

order by count(u.id) desc limit 10
```

Chattiest Users

Users	Number of Chats
394	115
2067	111
209	109
1087	109(there's a tie in the third place)

STEPS:

```
match (i:ChatItem)-[:PartOf]->(c:TeamChatSession)-[:OwnedBy]->(t)

return distinct count(t.id), t.id

order by count(t.id) desc limit 10
```

Chattiest Teams

Teams	Number of Chats
82	1324
185	1036
112	957

Finally, present your answer, i.e. whether or not any of the chattiest users are part of any of the chattiest teams.

Steps:

```
match (u:User)-[r:CreateChat]->(i:ChatItem)
with u.id as uid, order by count(i) desc limit 10

match (u:User)-[:Joins]->(c:TeamChatSession)-[:OwnedBy]->(t:Team)
where u.id=uid

return distinct u.id, t.id
```

Yes. There is one chattiest user (999) belongs to one chattiest team (52).

This kind of search may be relevant to Eglence Inc. business plan because there is a direct link between the chattiest user and his/her team being also the chattiest, Eglence could target the chattiest team with paid chatting stickers/emojis or something like that to gain revenue.

How Active Are Groups of Users?

Describe your steps for performing this analysis. Be as clear, concise, and as brief as possible. Finally, report the top 3 most active users in the table below.

Steps:

```
Match (u1:User)-[:CreateChat]->(i:ChatItem)-[:Mentioned]->(u2:User)

create (u1)-[:InteractsWith]->(u2)

Match (u1:User)-[:CreateChat]->(i:ChatItem)-[:ResponseTo]->(i2:ChatItem),
(u2:User)-[:CreateChat]->(i2:ChatItem)

create (u1)-[:InteractsWith]->(u2)

Match (u1)-[r:InteractsWith]->(u1) delete r
```

Finding coefficient Example:

```
MATCH (u1:User) WHERE u1.id = 394

MATCH (u1)-[:InteractsWith]-(n:User)

return count(distinct n)


MATCH (u1:User) WHERE u1.id = 394

MATCH (u1)-[:InteractsWith]-(n:User)

MATCH (n)-[:InteractsWith]->(o:User) where o <> u1

RETURN n.id,o.id, count(case when (n)-->(o) then 1

else 0

end) as value
```

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
394	0.75
2067	0.64
209	0.83

User 394 has k=4 neighbors, therefore k*(k-1) = 12. There are total of 9 edges in neighbors of neighbors.

User 2067 has k=8 neighbors, therefore k*(k-1) = 56. There are total of 36 edges in neighbors of neighbors.

User 209 has k=7 neighbors, therefore k*(k-1) = 42. There are total of 35 edges in neighbors of neighbors.