Graph Analytics

Modeling Chat Data using a Graph Data Model

Graph Data Model is perfectly suited for evaluating or analyzing various hypotheses based on Chat Data generated out of Catch the pink flamingo game.

There are following node types and edge types in the graph model of Chat Data.

- Node Types: User node, Team node, Chat Item node, Team Chat Session node
- Edge Types: Joins, Leaves, Mentioned,CreateChat, CreateSession, OwnedBy, ResponseTo, PartOf

CreateSession edge is created from User node to TeamChatSession node when a User initiates a TeamChatSession.

OwnedBy edge is created from TeamChatSession node to Team node whenever any User from a Team initiates a TeamChatSession.

Joins edge is created from User node to TeamChatSession node when a User joins a TeamChatSession.

Leaves edge is created from User node to TeamChatSession node when a User leaves a TeamChatSession.

CreateChat edge is created from User node to ChatItem node when a User creates a ChatItem.

PartOf edge is created from TeamChatSession node to Team node for all TeamChatSessions belonging to a Team

Mentioned edge is created from ChatItem node to User node when a User is mentioned in a ChatItem.

ResponseTo edge is created between two ChatItem nodes when one ChatItem is a response to another ChatItem

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:

File Name	Description	Example
1. chat_create_team_chat.csv	A line is added to this file when a player creates a new chat with their team	userid, teamid, TeamChatSessionID, timestamp 559,48,6288,14567
2. chat_item_team_chat.csv	Creates nodes labeled ChatItems. Column 0 is User id, column 1 is the TeamChatSession id, column 2 is the ChatItem id (i.e., the id property of the ChatItem node), column 3 is the timestamp for an edge labeled "CreateChat". Also create an edge labeled "PartOf" from the ChatItem node to the TeamChatSession node. This edge should also have a timeStamp property using the value from Column 3	userid, teamchatsessionid, chatitemid, timestamp 1956,6299,6305,1464235803
3. chat_join_team_chat.csv	Creates an edge labeled "Joins" from User to TeamChatSession. The columns are the User id, TeamChatSession id and the timestamp of the Joins edge	Example: userid, TeamChatSessionID, teamstamp 559,6288,12345
4. chat_leave_team_chat.csv	Creates an edge labeled "Leaves" from User to TeamChatSession. The columns are the User id, TeamChatSession id and the timestamp of the Leaves edge	userid, teamchatsessionid, timestamp 1244,6821,1464241204.0
5. chat_mention_team_chat.csv	Creates an edge labeled "Mentioned". Column 0 is the id of the ChatItem,	ChatItem, userid, timeStamp 6349,2508

	column 1 is the id of the User, and column 2 is the timeStamp of the edge going from the chatItem to the User	
6. chat_respond_team_chat.csv	A line is added to this file when a player responds to a chat post	chatid1, chatid2,timestamp 6326,6305,21564

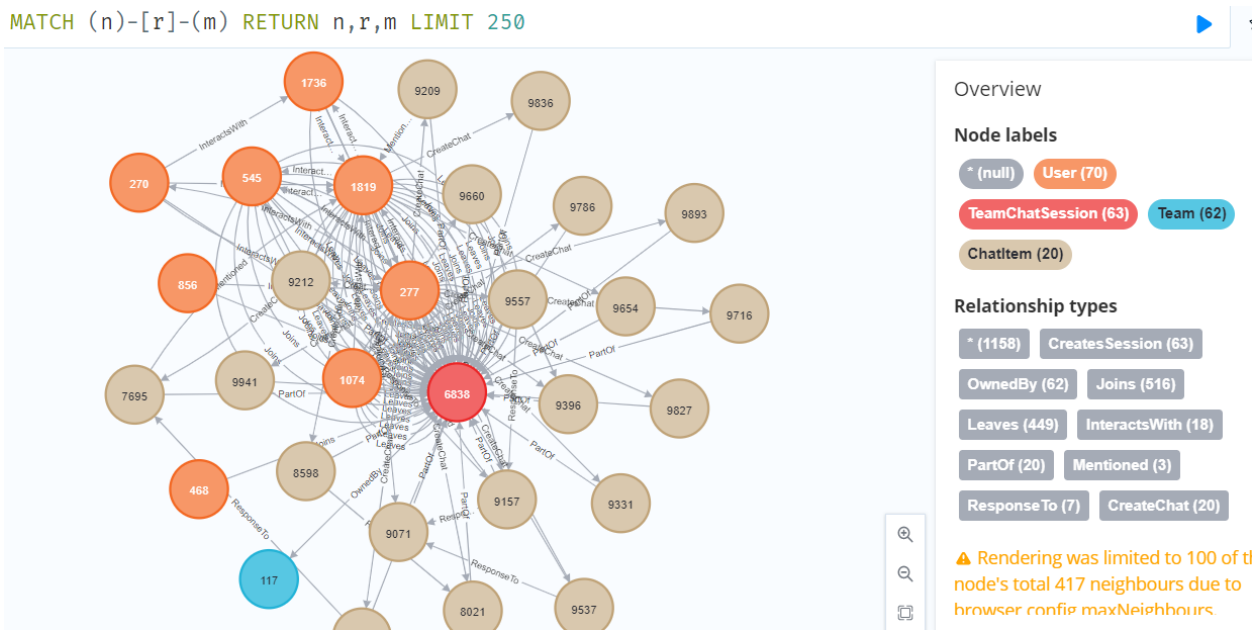
ii) Explain the loading process and include a sample LOAD command:

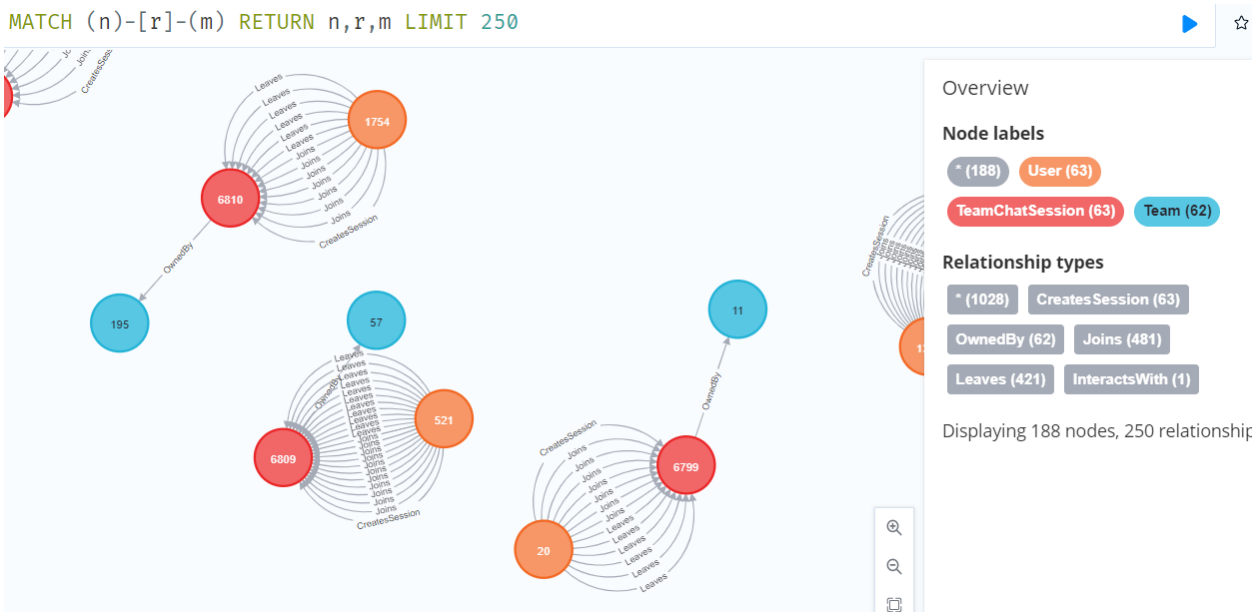
First, we defined the constraints for all the node types to be unique so that they are not duplicated when multiple relationships are created.
Following example snippet shows loading command for chat_join_team_chat.csv file.

LOAD CSV FROM "file:///chat_data/chat_join_team_chat.csv" AS row
MERGE (u:User {id: toInteger(row[0])})
MERGE (c:TeamChatSession {id: toInteger(row[1])})
MERGE (u)-[:Joins {timeStamp: row[2]}]->(c)

We are reading the file row by row. In each row of this file, first value is attributed to User node ID, second value is attributed to TeamChatSession ID while third value indicates timestamp of the *Join* relationship i.e. when User ID joined the TeamChatSession ID. Merging commands for all the three values defines the nodes and creates the specified relationship between them. Approach is similar in every file.

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 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.





Finding the longest conversation chain and its participants

The longest conversation is 9 nodes long that includes 5 unique Users as part of the conversation.

Following query produces the longest path where we can see the ChatItem ids. Then select the path and count the distinct nodes in second query.

match p=(i)-[:ResponseTo*]->(j) return i.id,j.id,length(p) as len order by len desc limit 1

match p=(i:ChatItem)-[:ResponseTo*]->(j:ChatItem)

```
where i.id=52694 and j.id=7803

with p

match (u:User)-[:CreateChat]->(k:ChatItem)

where k in nodes(p)

return count(distinct u)
```

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

Chattiest users are the ones having the most number of relationships of type 'CreateChat'.

Chattiest Users

Users	Number of Chats
394	115
2067	111
1087	109

Chattiest teams are the ones with the most number of nodes of type 'ChatItem' that are part of TeamChatSessions owned by the team.

Chattiest Teams

Teams	Number of Chats
82	1324
185	1036
112	957

User ID 999 is the 6th most chattiest User and is part of Team ID 52 which is the 7th most chattiest Team.

How Active Are Groups of Users?

To find how active the groups of users are, first we created a new edge type *'InteractsWith'* between users which are responding to each other or who mention one another. Then we compute how dense the neighborhoods are for the chattiest Users.

There is only one *InteractsWith* edge between two users even if one interacts with another multiple times via *RespondTo* or *Mentioned* relationships. This is ensured by not duplicating the edge while creating it.

Cluster Coefficient is computed as the number of edges present amongst the neighbors divided by the total number of edges possible amongst them. E.g. if chattiest user has 5 neighbors and there are 15

edges amongst these 5 users (excluding edges with the chattiest user), then cluster coefficient is 15/20 = 0.75

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
394	1.00
461	1.00
209	0.95