

Catch Flamingo Game Data Files Structure

1.) Modeling Chat Data using a Graph Data Model

The graph represents a model of interactions among Users,Team,Chat Sessions and Chat Items.

The entities of the graph are defined as follows;

Nodes:User,Team,Team,ChatSession,ChatItem

Edges:CeateChat,Join,Leaves,Mentioned,OwnedBy,ResponseTo,PartOf

2.) Creation of the Graph Database for Chats

2.1) Schema of the 6 CSV files

Down below is provided the whole se of tables which describe the structure of each single data files.

Each table includes the name and a description for all of the fields.

File:chat_create_team_chat.csv

A line is added to this file when a player creates a new chat with their team.

userid	The User Identifier
teamid	The Team Identifier
TeamChatSessionID	The Team Chat Session Identifier
timestamp	Time when the Chat Had been created. Property of the relationship "CreatesSession" between the nodes.User and TeamChatSession Property of the relationship "OwnedBy" between the Team and TeamChatSession

File: chat_item_team_chat.csv

A line is added to this file for each Item Chat

Field Name	Description
userid	The User Id
teamchatsessionid	The Team Chat Session Id
chatitemid	The Chat Item Id
timestamp	Time when the TeamChatSession Had created Or the Time when a particular Chat Item had been part of a Chat Session. Property of the relationship "PartOf" between the nodes ChatItem and Teamchatsession.. Property of the relationship "Creates" between the nodes.User and TeamChatSession

File: chat_join_team_chat.csv

A line is added to this file when a player Joins a new chat with their team.

Field Name	Description
userid	The User Id
teamchatsessionid	The Team Chat Session Id
teamstamp	Time when the an User joined a Chat . Property of the relationship "Join" between the nodes. User and TeamChatSession

File: chat_leave_team_chat.csv

A line is added to this file when a player Leaves a chat with their team.

Field Name	Description
userid	The User Id
teamchatsessionid	The Team Chat Session Id

teamstamp	Time when the an User Leaved a Chat . Property of the relationship “Leaves” between the nodes.User and TeamChatSession
-----------	---

File: chat_mention_team_chat.csv

A line is added to this file when a Item Chat is mentioned by a User

Field Name	Description
ChatItemid	The Chat Item Id
userid	The User id
teamstamp	Time when the a chat Item had been mentioned in a chat Property of the relationship “Mentioned” between the nodes.ChatItem and User-

File: chat_respond_team_chat.csv

A line is added to this file when a chat is a response to another chat

Field Name	Description
chatid1	The Id of the response chat
chatid2	The Id of the original chat
teamstamp	Time when the a chat Item had been mentioned in a chat. Property of the relationship “RespondsTo” between the nodes. ChatItem

2.2) Loading process explanation and a sample LOAD command

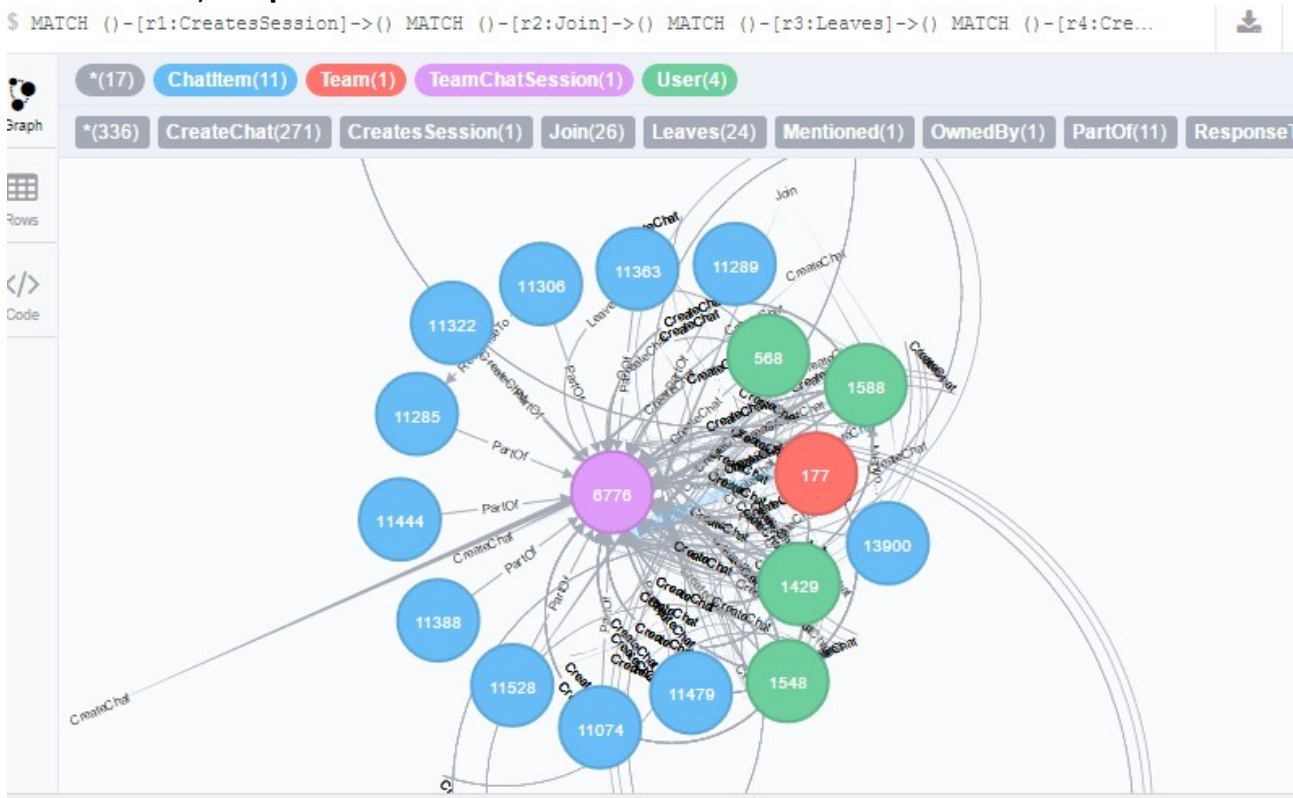
During the loading process the nodes and relationships are created,the values of the node attributes and the properties of the relationships are taken from the corresponding line items.

- 2.2.1) Load in memory the csv file specified in the command parameters
- 2.2.2) Creates nodes with specified attribute values
- 2.2.3) Creates Relationships among nodes

Here is a sample loading command for chat_leave_team_chat.csv file

```
LOAD CSV FROM "file:///C:/files/chat_leave_team_chat.csv" AS row
MERGE (u:User {id: toInt(row[0])})
MERGE (c:TeamChatSession {id: toInt(row[1])})
MERGE (u)-[:Leaves{timeStamp: row[2]}->(c);
```

2.3) Graph Screenshot



3.) Finding the longest conversation chain and its participants

3.1) Report the results including the length of the conversation (path length)
10 Items,9 Relationships

3.2) How many unique users were part of the conversation chain
10

3.3) Describe your steps

3.4 Find the longest chain

```
match p=i1-[:ResponseTo*]->i2
return p order by length(p) desc limit 1
```

3.5 Get the chat session Id

```
MATCH p=i-[:ResponseTo*]->i2 where length(p) = 9
```

```
MATCH (i:ChatItem)-[t:PartOf]->(c:TeamChatSession) where i.id in (extract(n in nodes(p)|n.id))
return p,t,c
```

3.6 Get out the Users by using the Chat Session Id

```
match (u:User)-[r:Join]->(c:TeamChatSession) where c.id = 7212
return distinct u
```

4.) Analyzing the relationship between top 10 chattiest users and top 10 chattiest teams

Chattiest Users :

```
match (u:User)-[r:CreateChat]->(c:TeamChatSession)
return u , count(r) as OutDegree
order by OutDegree desc
```

Users	Number of chats
394	115
2067	111
1087	109

Chattiest Teams:

```
match (t:Team)<-[r1:OwnedBy]-(c:TeamChatSession)
match (u:User)-[r2:CreateChat]->(c:TeamChatSession)
return t.id , c.id,count(r2) as InDegree
order by InDegree desc
```

Teams	Number of chats
82	1324
185	1036
112	957

Conclusion :

The chattiest Users are not part of the Chattiest teams.

5.) **How Active Are Groups of Users?**

Build a sub-graph containing all the chattiest users and its neighbors.

The sub-graph should include a new direct relationship between the chattiest nodes.

For each chattiest node user divide 1 by the sum of its neighbors plus the number of the neighbors nodes connected to each other .

The table below shows the results.

User Id	Coefficient
394	0,917
2067	0,7678
209	0,9523