**Graph Analytics**

**Modeling Chat Data using a Graph Data Model**

This is a Graph Model for chats created by users of the game “To Catch The Pink Flamingo”. The Graph Model has 4 types of nodes (**Users**, **Teams**, **Team Chat Sessions** and **Chat Items**)

Each node has its own unique ID. There are various types of edges, each represents a different kind of interaction between nodes

Edges such as ‘**CreatesSession**’, ‘Joins’ and ‘Leaves’ describes the interaction Users have on the Chat Sessions

Edges such as ‘**CreatesChat**’ describes the interaction Users have on the Chat Items

Edges such as ‘**PartOf**’ describes which Chat Sessions the Chat Items belong to

Edges such as ‘**OwnedBy**’ describes which Teams the Chat Sessions belong to

Edges such as ‘**Mentions**’ and ‘**ResponsesTo**’ describes the interaction ChatItems have towards Users or other Chat Items respectively

**Creation of the Graph Database for Chats**

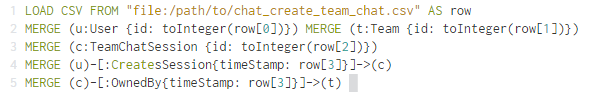
Describe the steps you took for creating the graph database. As part of these steps

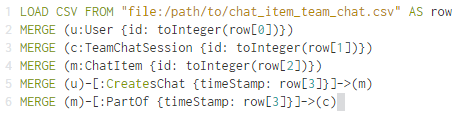
1. Write the schema of the 6 CSV files

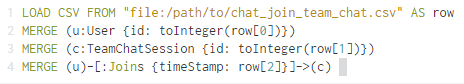
|  |  |
| --- | --- |
| Chat\_create\_team\_chat | * userid: ID of user: Int * teamid: ID of team: Int * TeamChatSessionID: ID of chat session: Int * timestamp: Timestamp   -- for userid **CREATING** TeamChatSessionID  -- for TeamChatSessionID **OWNEDBY** teamid |
| Chat\_item\_team\_chat | * userid: ID of user: Int * teamchatsessionid: ID of chat session: Int * chatitemid: ID of chat item: Int * timestamp: Timestamp   -- for userid **CREATING** chatitemid  -- for chatitemid **PARTOF** teamchatsessionid |
| Chat\_join\_team\_chat | * userid: ID of user: Int * TeamChatSessionID: ID of chat session: Int * timestamp: Timestamp   -- for userid **JOINING** TeamChatSessionID |
| Chat\_leave\_team\_chat | * userd: ID of user: Int * teamchatsessionid: ID of chat session: Int * timestamp: Timestamp   -- for userid **LEAVING** teamchatsessionid |
| Chat\_mention\_team\_chat | * ChatItem: ID of chat item: Int * userid: ID of user: Int * timestamp: Timestamp   -- for ChatItem **MENTIONING** userid |
| Chat\_respond\_team\_chat | * chatid1: ID of chat item: Int * chatid2: ID of chat item: Int * timestamp: Timestamp   -- for chatid1 **RESPONDINGTO** chatid2 |

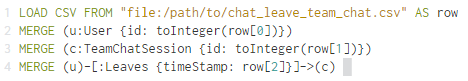
1. Explain the loading process and include a sample LOAD command

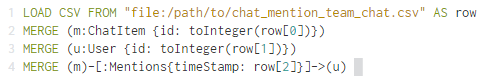
* The loading process determines **which columns** in the CSV file should be **used as integer ids** for the **nodes** and **which single column** should be used as **timestamp of the interaction**.
* Nodes are read as (**nodechar**: **Nodetype** {id: toInteger(row[**row order**])}). Bolded fields are filled in by the Neo4j user
* Edges are read as –[: **Interactiontype** {timestamp: row[**order of timestamp column**]}] –>. Bolded fields are filled in by the Neo4j user
* The CSV **headers** reveal what **type of nodes** that is whereas the **type of interaction** has to be **guessed** via the **file name**. There can be **more than one** type of interaction
* If there are many types of interaction, the timestamp column is shared by them





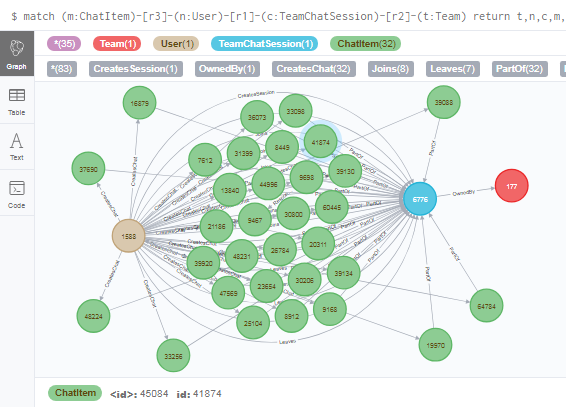








1. Present a screenshot of some part of the graph you have generated. The graphs must include clearly visible examples of most node and edge types. Below are two acceptable examples. The first example is a rendered in the default Neo4j distribution, the second has had some nodes moved to expose the edges more clearly. Both include examples of most node and edge types.

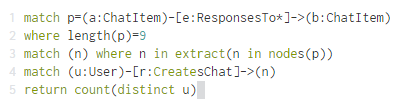


**Finding the longest conversation chain and its participants**

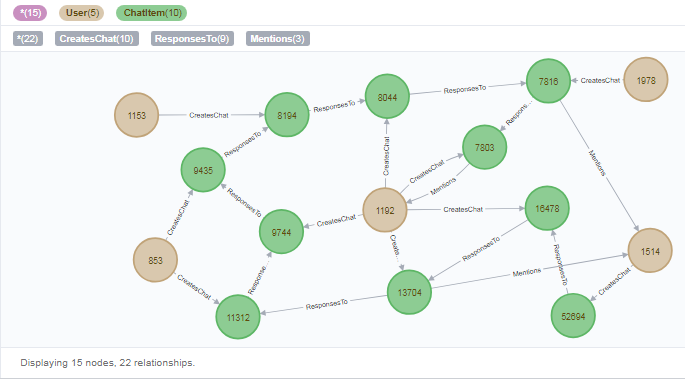
Report the results including the length of the conversation (path length) and how many unique users were part of the conversation chain. Describe your steps. Write the query that produces the correct answer.



Longest conversation chain’s length = 9



Number of unique users in this chain = 5 (1153, 1192, 1514, 1978, 853)



The entire longest conversation chain path, consisting of 5 users and 10 chat items

**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.

**Chattiest Users**

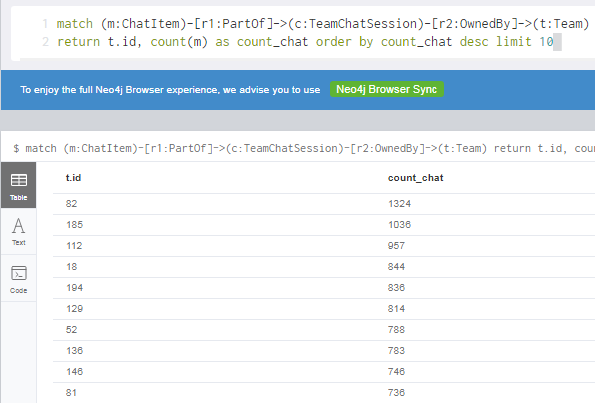
* Find all Users –CreatesChat->ChatItem relationships.
* Then count number of these ChatItems for each User
* Then rank in descending order by this count (limit to top 10)



|  |  |
| --- | --- |
| **Users** | **Number of Chats** |
| 394 | 115 |
| 2067 | 111 |
| 1087 ties with 209 | 109 |

**Chattiest Teams**

* Find all ChatItem –PartOf-> TeamChatSession –OwnedBy-> Team relationships.
* Then count number of these ChatItems for each Team
* Then rank in descending order by this count (limit to top 10)



|  |  |
| --- | --- |
| **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.

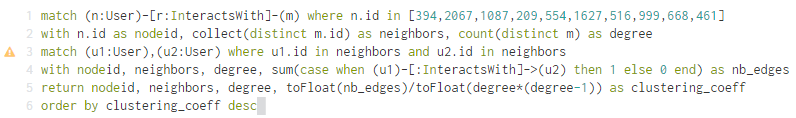
* Out of 10 chattiest users, only **user 999** belongs to the chattiest **team 52**
* **No other relationship observed**

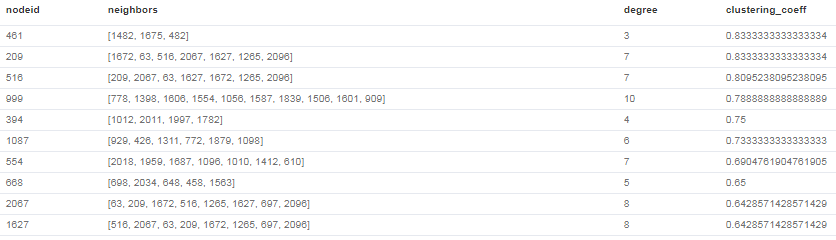
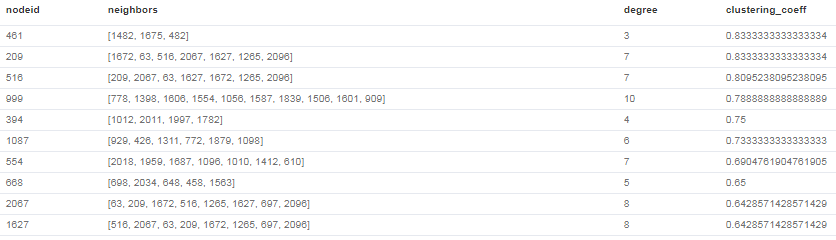
**How Active Are Groups of Users?**

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.

There are two ways to consider connection between neighbors’ nodes:

1. Consider the edges as **directed edges**. That means if there are 2 nodes interacting with each other that connection should be **counted twice** & maximum number of connections **= k\*(k-1)**



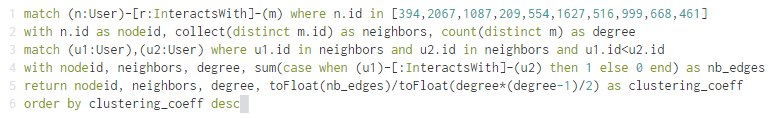


THIS METHOD SEEMS TO BE THE ONE **IMPLIED BY THE ASSIGNMENT INSTRUCTION**. HOWEVER THE RESULTING COEFFICIENT VALUES **DON’T MATCH THE GRADING RUBRIC**

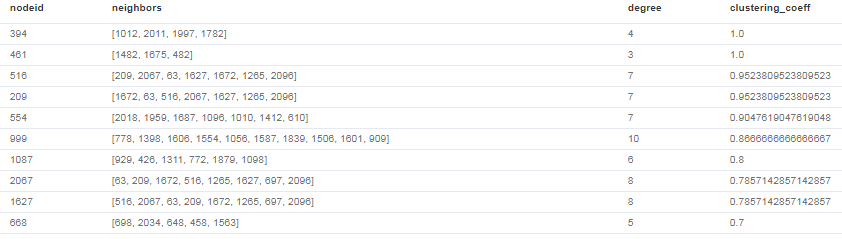
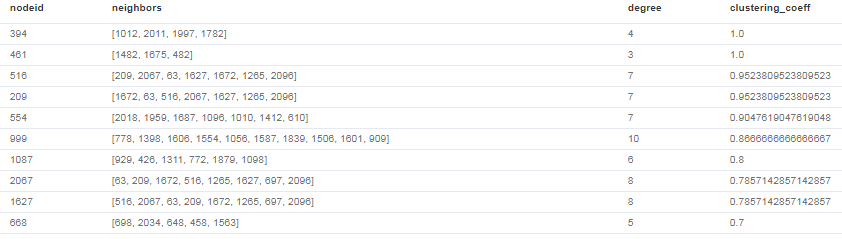
**Most Active Users (based on Cluster Coefficients) – METHOD 1**

|  |  |
| --- | --- |
| **User ID** | **Coefficient** |
| 461 | 0.8333 |
| 209 | 0.8333 |
| 516 | 0.8095 |

1. Consider the edges as **un-directed edges**. That means if there are 2 nodes interacting with each other that connection should be **counted exactly once** & maximum number of connections = **k\*(k-1)/2**



**The u1.id < u2.id is to prevent double counting**



THIS METHOD **DOESN’T** SEEM TO BE THE ONE **IMPLIED BY THE ASSIGNMENT INSTRUCTION**. HOWEVER THE RESULTING COEFFICIENT VALUES **MATCH THE GRADING RUBRIC.** BUT **GOD KNOWS WHY** THEY ARE REGARDED AS **THE TOP 3**

**Most Active Users (based on Cluster Coefficients)**

|  |  |
| --- | --- |
| **User ID** | **Coefficient** |
| 394 | 1.0 |
| 461 | 1.0 |
| 516 | 0.9524 |