# Part 1:

**userID** and **sessionID** could be used as a composite primary key, making each record unique. **sessionId** being a long int would ensure that there are enough unique numbers to ensure each record is uniquely identifiable. **missionID** could be an int to uniquely identify mission numbers. **isHit** would be a Boolean value, either yes for True or no for False. A **flamingoId** attribute could be added to track which type of flamingo has been hit or missed for each click, using an int to represent different types of flamingos.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| userID: long | sessionID: long | timestamp: dateTime | clickedPoint: coordinate | missionID: int | isHit: boolean | flamingoId int |
| 100 | 4356 | 10/12/2015::14:15:09 | (4,8) | 13 | yes | 3 |
| 101 | 3241 | 10/23/2015::14:15:19 | (20,5) | 18 | no | 7 |
| 102 | 4537 | 11/4/2015::14:15:20 | (17,43) | 21 | no | 19 |

# Part 2:

**Nodes / Entities:** User, Chat Session, Chat Text

**Edges / Actions:** Leave, Join, Start, Write, Contain, Mention

1. When a user joins a chat session, check which team they are from and increment a counter which represents the amount of active chat sessions open for that team.
2. Record the count and size of messages a user writes historically during a chat session; aggregate if the average count and size of messages decreases as the time gets closer to when the user leaves the chat session.
3. Record all words written into chat text through out a chat session. Count up the instances of each word to see which are most frequently used.
4. Record all messages sent by each user in a chat session and count up the number of messages from each user to see who sends the most. This could also be averaged over the time a user is in a chat session to record how many messages are sent per minute.
5. For a given user, sum the number of sessions a user has joined, minus the number of a sessions a user has left, resulting in the active count of sessions a user is currently in.

# Part 3:

**Flamingo Properties:**

* Feather-colour: “red”, “orange”, “pink”, “green”
* Beak-colour: “blue”, “yellow”, “purple”
* Foot-shape: “triangular”, “circular”, “square”
* Hat-type: “Fedora”, “Bowler”, “Flat”, “Fez”, “Cowboy”, “Viking”
* Hat-colour: “Brown”, “Black”, “Orange”, “Red”, “green”
* Feather-pattern: “Diamond”, “Polka-dot”, “Triangle”, “Circle”, “Oval”

3 new groups can be added to the root node on the tree to help store user information, chat information and click data.

The user data object can store information like the user’s id, username, team, and their score and rank

**Users:**

* Id
* username
* teamId
* total\_score
* rank

The chat data object stores information regarding chat sessions, including the team and session id for a chat, the users in that chat, a history of chat actions such as users leaving and joining, and the messages and timestamps.

**Chats:**

* teamId
* sessionId
* users
  + userId
* history
  + chat\_action
  + userId
  + timestamp
* messages
  + sender
  + text
  + timestamp

The clicks data object can be used to store information about each click within a play session. This would be linked with the game information but separate as it would contain only click related information, not information about the level and mission etc. It contains identity information identifying a user, the game session and mission they are in, and information about the flamingo they are trying to click and whether they successfully clicked it or not.

**Clicks:**

* userId
* sessionId
* timestamp
* clickedPoint
* missionId
* isHit
* flamingoId