**STEPS FOLLOWED IN NEO4J**

**This file includes the commands used in Neo4j for Graph analytics.**

1. **To visualise Joins relationship**

LOAD CSV FROM " <https://raw.githubusercontent.com/meeramullamkuzhy/Catch-the-Pink_flamingo/main/chat_join_team_chat.csv>" AS data

MERGE (u:User {id: data[0]})

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

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

MATCH (u)-[r]->(c)

RETURN u,r,c

1. **To visualise Leaves relationship**

LOAD CSV FROM" https://raw.githubusercontent.com/meeramullamkuzhy/Catch-the-Pink\_flamingo/main/chat\_leave\_team\_chat.csv"AS data

MERGE (u:User {id: data[0]})

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

MERGE (u)-[:Leaves{timeStamp: data[2]}]->(c)

MATCH (u)-[Leaves]->(c)

RETURN u,Leaves,c

1. **To visualise Mentions relationship**

LOAD CSV FROM" https://raw.githubusercontent.com/meeramullamkuzhy/Catch-the-Pink\_flamingo/main/chat\_mention\_team\_chat.csv" AS line

MERGE (c:ChatItem {id: line[0]})

MERGE (u:User {id: line[1]})

MERGE (c)-[:Mentions{timeStamp: line[2]}]->(u)

MATCH (c)-[ Mentions]->(u)

RETURN c, Mentions,u

1. **To visualise RespondsTo relationship**

LOAD CSV FROM "https://raw.githubusercontent.com/meeramullamkuzhy/Catch-the-Pink\_flamingo/main/chat\_respond\_team\_chat.csv" AS line

MERGE (c:User {id: line[0]})

MERGE (u:User {id: line[1]})

MERGE (c)-[:RespondsTo{timeStamp: line[2]}]->(u)

MATCH (c)-[ RespondsTo]->(u)

RETURN c, RespondsTo,u

1. **To visualise the longest conversation**

match p=(i1:User)-[:RespondsTo\*]->(i2:User)

return p, length(p) order by length(p) desc limit 1

1. **To find active team**

MATCH (u:User)-[:Joins]->(c:TeamChatSession)

RETURN count(c) as freq

c.id as team ORDER BY freq DESC

1. **To find active users**

MATCH (c:ChatItem)-[:Mentioned]->( u:User)

RETURN u.id,  count(c) as freq

ORDER BY freq DESC