**Lab08: Graph Model (Neo4j)**

**Objective:** To study and practice the Cypher commands in Neo4j to manage data in the graph database.

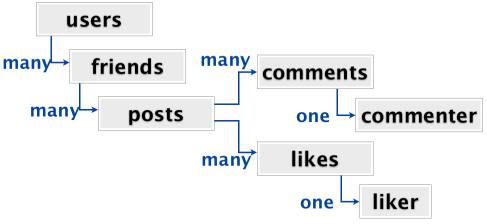
**Estimated Time:** 1.5 hours

**Number of Tasks:**

**Due:** Thursday 3rd November 2022, 11:55 P.M.

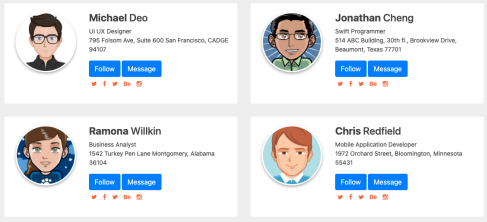
**Scenario: Social Network System**

This take-home assignment is based on the social network system as previous laboratories. The data maintained in the graph model consists of several components in the social network system, users and posts and all activities occurring in this system are making friends, commenting and reacting (likes) on user posts. The figure below shows the social network data model.



**Instructions:** For each task you are required to show the appropriate command to correspond to that task and the result to prove that your command works correctly.

**Task 1:** Consider these 4 users and create them in the graph database. Make sure you include their profile information.



CREATE

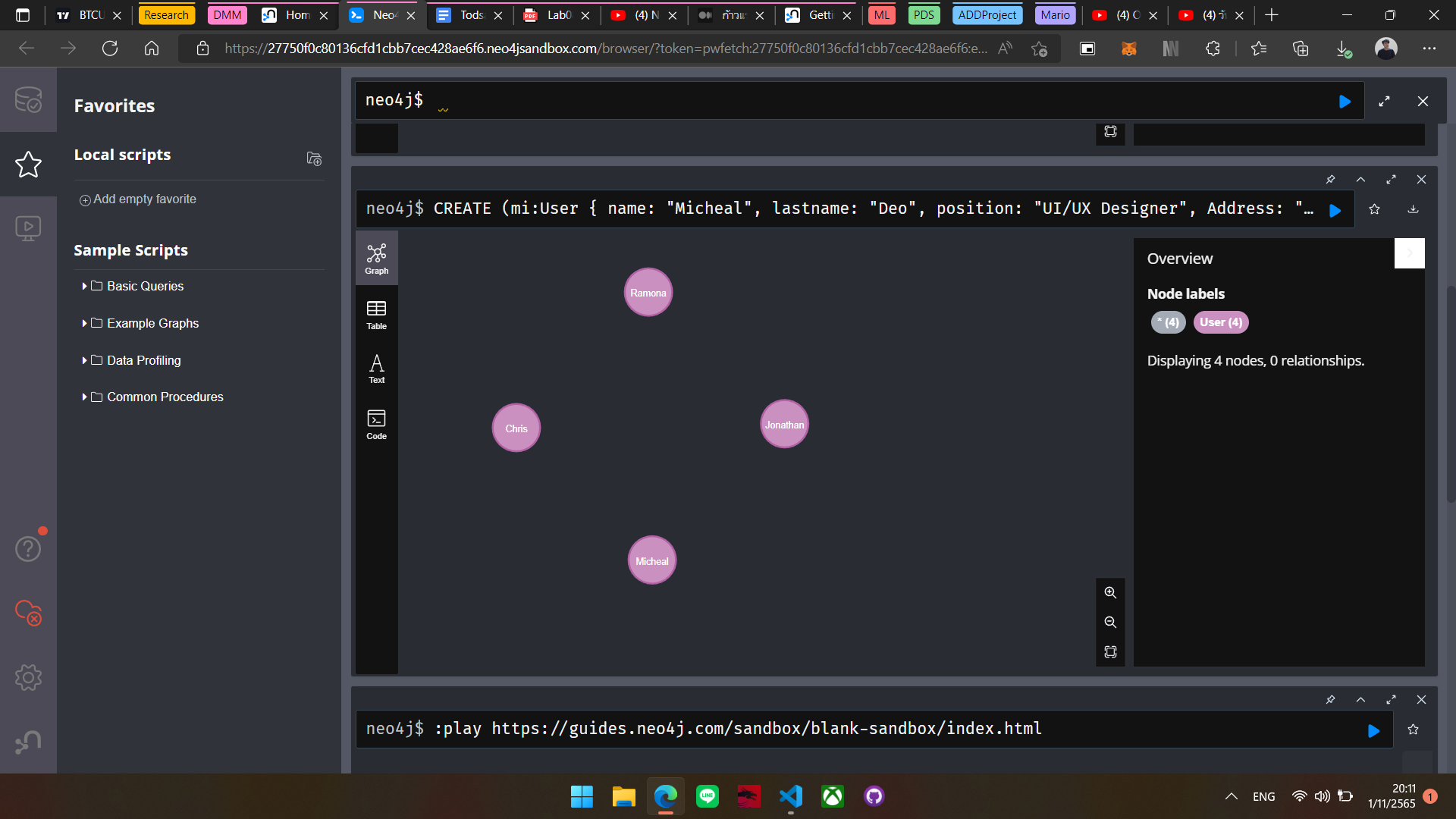
(mi:User { name: "Micheal", lastname: "Deo", position: "UI/UX Designer", Address: "795 Folsom Ave, Suite 600 San Francisco, CADGE 94107" }),

(jo:User { name: "Jonathan", lastname: "Cheng", position: "Swift Programmer", Address: "514 ABC Building, 30th fl, Brookview Drive, Beaumont, Texas 77701" }),

(ra:User { name: "Ramona", lastname: "Willkin", position: "Business Analyst", Address: "1542 Turkey Pen Lane Montgomery, Alabena 36104" }),

(ch:User { name: "Chris", lastname: "Redfield", position: "Mobile Application Developer", Address: "1972 Orchard Street, Bloomington, Minnesota 55431" })

RETURN mi,jo,ra,ch;



**Task 2:** Suppose 4 of them are each other’s friends, show a correspondent command to make them friends and show the result after adding them in the graph representation.

MATCH (mi:User), (jo:User),(ra:User),(ch:User)

WHERE mi.name="Micheal" AND jo.name="Jonathan" AND ra.name = "Ramona" AND ch.name="Chris"

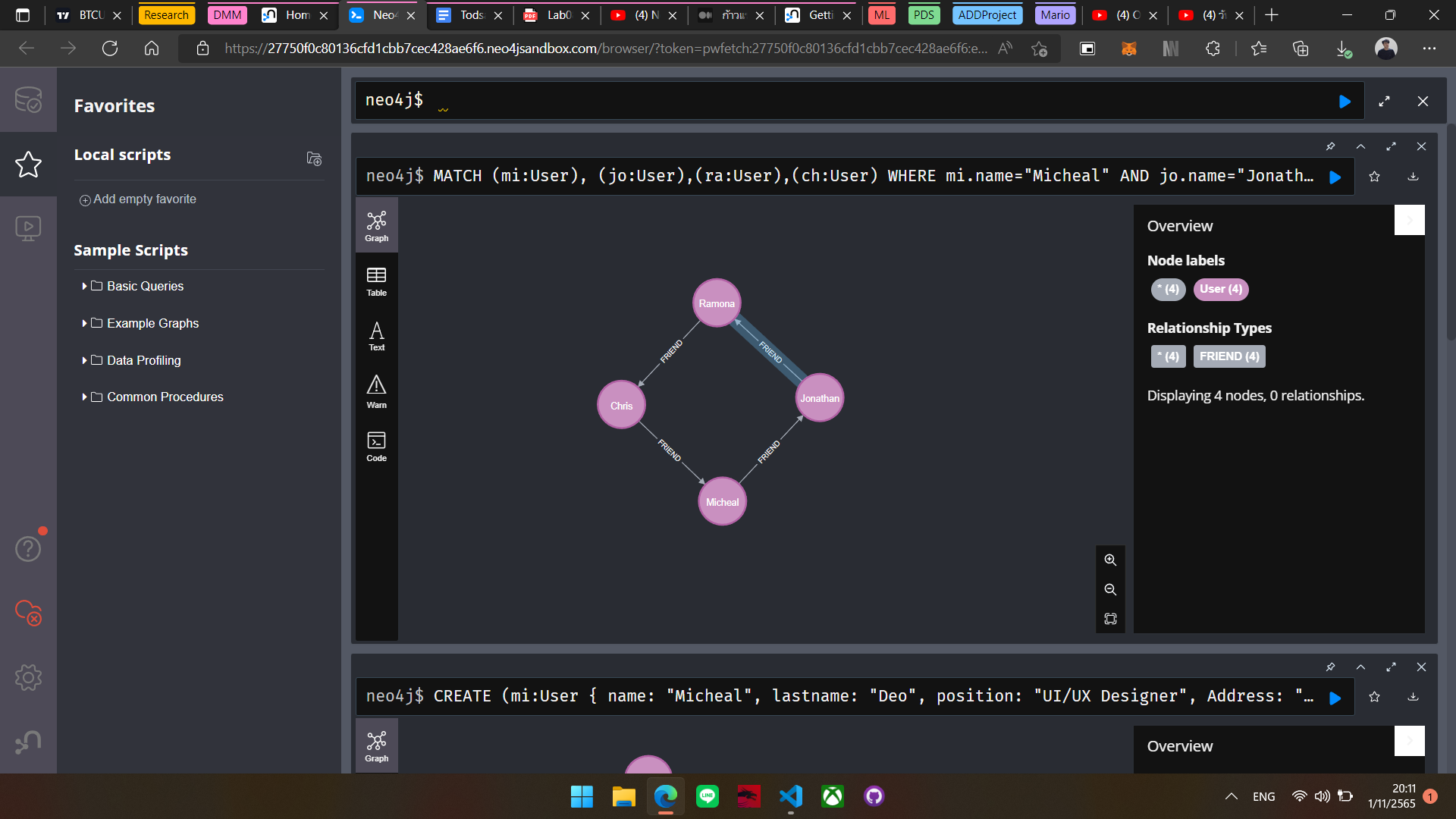
CREATE (mi)-[:FRIEND {acceptedAt: date() }]-> (jo),

(jo)-[:FRIEND {acceptedAt: date() }]-> (ra),

(ra)-[:FRIEND {acceptedAt: date() }]-> (ch),

(ch)-[:FRIEND {acceptedAt: date() }]-> (mi)

RETURN mi,jo,ra,ch;



**Task 3:** Create 4 posts from all users (1 post each) and let the owner **REACT** his/her own post **(reactions can be Like, Love, Haha, Sad, Care and Angry, choose only one for each post).** Show a successful command and the result after adding such posts in the graph representation.

MATCH (mi:User),(jo:User),(ra:User),(ch:User)

WHERE mi.name="Micheal" AND jo.name="Jonathan" AND ra.name="Ramona" AND ch.name="Chris"

CREATE (p1:Post { id:"1", message: "My name is Micheal." } ),

(mi)-[:POSTED]->(p1),

(mi)-[:REACTED { type: "Like", createdAt: date() }]->(p1),

(p2:Post { id:"2", message: "My name is Jonathan." } ),

(jo)-[:POSTED]->(p2),

(jo)-[:REACTED { type: "Love", createdAt: date() }]->(p2),

(p3:Post { id:"3", message: "My name is Ramona." } ),

(ra)-[:POSTED]->(p3),

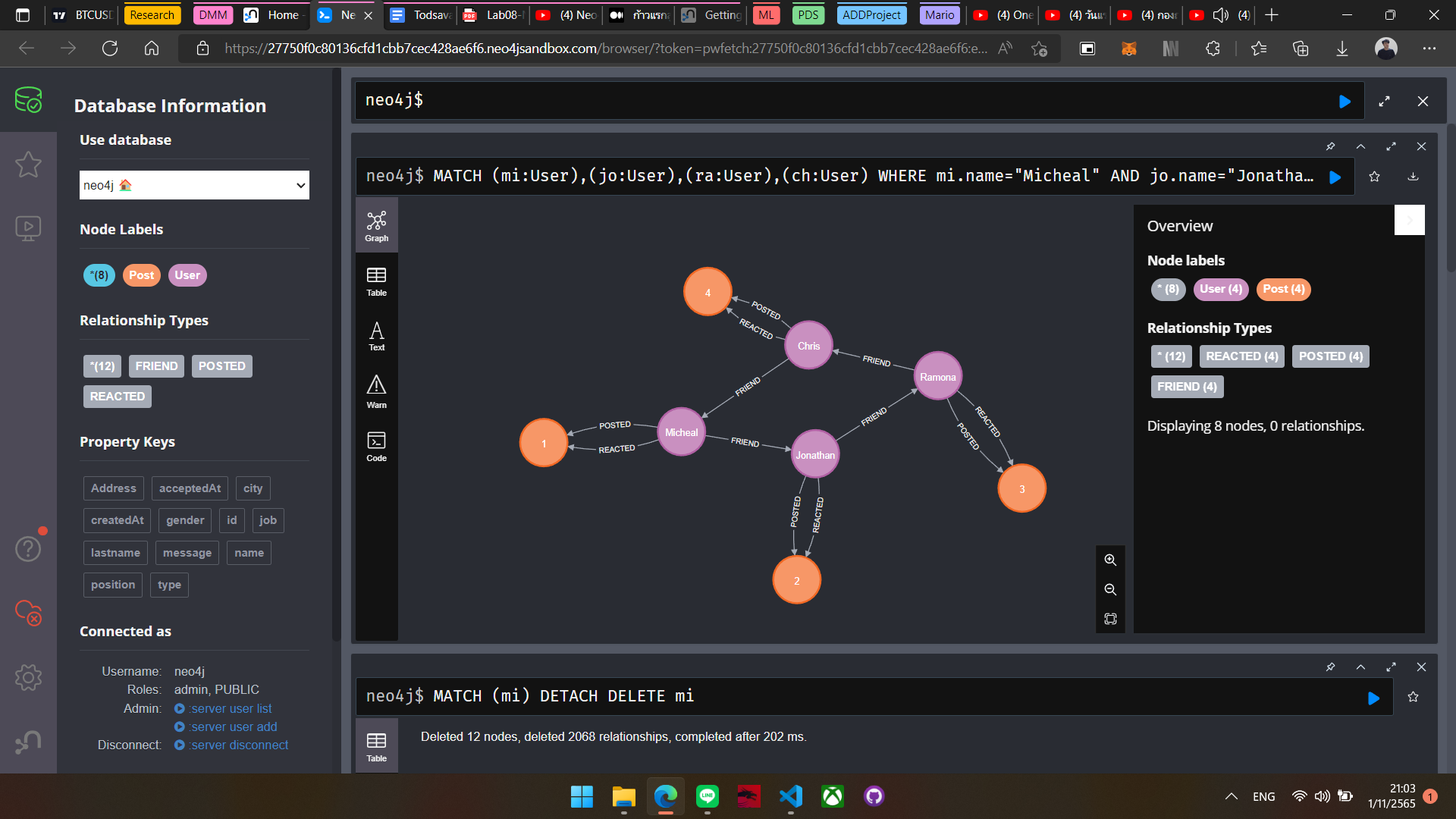
(ra)-[:REACTED { type: "Haha", createdAt: date() }]->(p3),

(p4:Post { id:"4", message: "My name is Chris." } ),

(ch)-[:POSTED]->(p4),

(ch)-[:REACTED { type: "Sad", createdAt: date() }]->(p4)

RETURN mi,jo,ra,ch,p1,p2,p3,p4;



**Task 4:** Write a command to let each user react on 2 posts from different users. The following table shows information about who reacts on whose post.

| **Post Owner** | **Michael** | **Jonathan** | **Ramona** | **Chris** |
| --- | --- | --- | --- | --- |
| **Michael** |  | Love | Like |  |
| **Jonathan** | Haha |  |  | Sad |
| **Ramona** | Care |  |  | Angry |
| **Chris** | Like |  | Love |  |

Show both the command to record the above reactions and the result in the graph format.

MATCH (mi:User),(jo:User),(ra:User),(ch:User),(p1:Post),(p2:Post),(p3:Post),(p4:Post)

WHERE mi.name="Micheal" AND jo.name="Jonathan" AND ra.name="Ramona" AND ch.name="Chris"

AND p1.id = "1" AND p2.id = "2" AND p3.id = "3" AND p4.id = "4"

CREATE

(mi)-[:REACTED { type: "Love", createdAt: date() }]->(p2),

(mi)-[:REACTED { type: "Like", createdAt: date() }]->(p3),

(jo)-[:REACTED { type: "Haha", createdAt: date() }]->(p1),

(jo)-[:REACTED { type: "Sad", createdAt: date() }]->(p4),

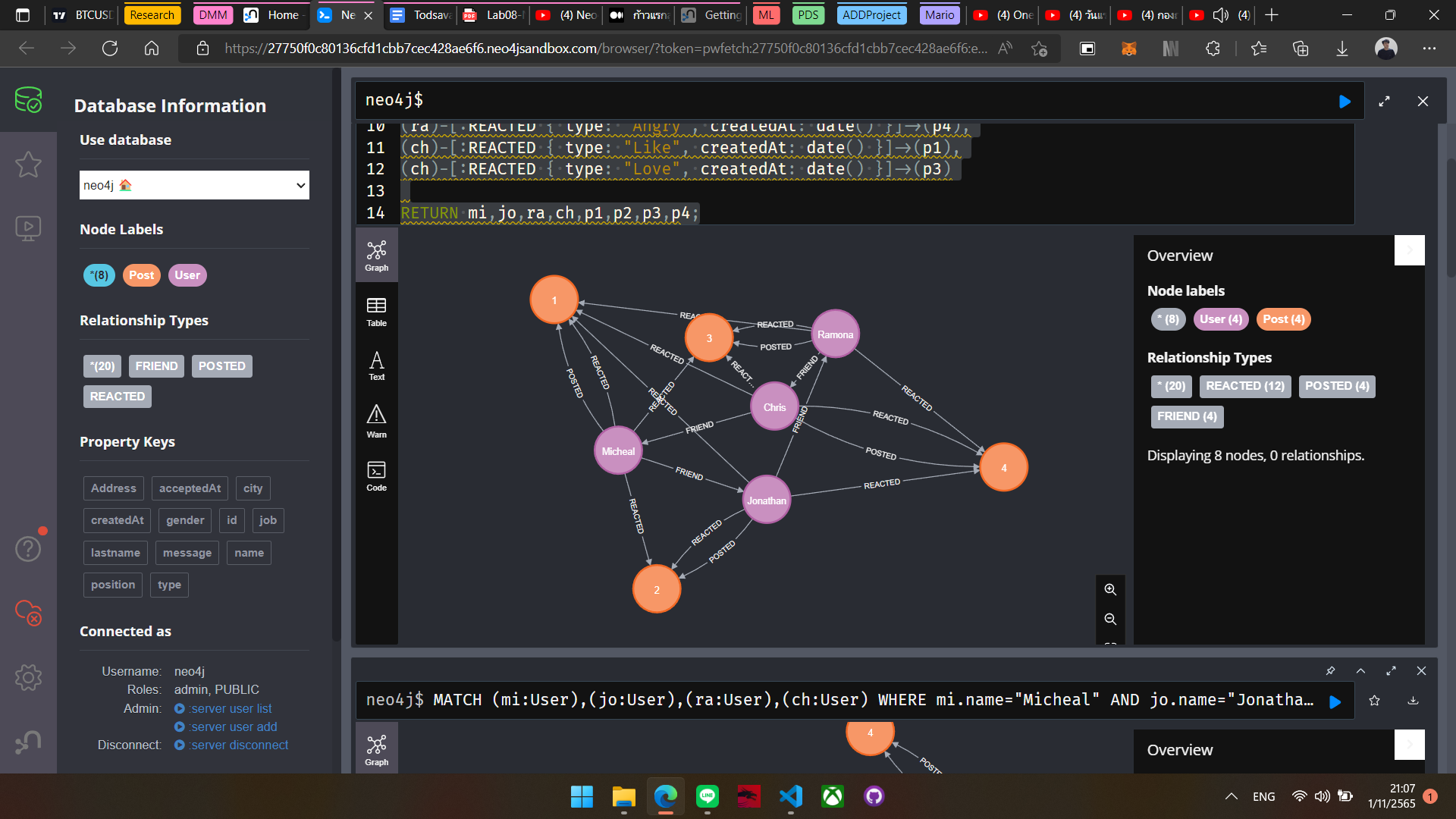
(ra)-[:REACTED { type: "Care", createdAt: date() }]->(p1),

(ra)-[:REACTED { type: "Angry", createdAt: date() }]->(p4),

(ch)-[:REACTED { type: "Like", createdAt: date() }]->(p1),

(ch)-[:REACTED { type: "Love", createdAt: date() }]->(p3)

RETURN mi,jo,ra,ch,p1,p2,p3,p4;



**Task 5:** Write a command to let each user comment on 2 posts. Each comment should store a *unique ID, message, a current date and time of creation*. The following table shows information about who comments on whose post. (Total of 8 comments for this task).

| **Post Owner** | **Michael** | **Jonathan** | **Ramona** | **Chris** |
| --- | --- | --- | --- | --- |
| **Michael** |  |  |  |  |
| **Jonathan** |  |  |  |  |
| **Ramona** |  |  |  |  |
| **Chris** |  |  |  |  |

Show both the command to record all comments and the result in the graph format.

MATCH (mi:User),(jo:User),(ra:User),(ch:User),(p1:Post),(p2:Post),(p3:Post),(p4:Post)

WHERE mi.name="Micheal" AND jo.name="Jonathan" AND ra.name="Ramona" AND ch.name="Chris"

AND p1.id = "1" AND p2.id = "2" AND p3.id = "3" AND p4.id = "4"

CREATE

(mi)-[:COMMENTED { uniqueID: "3", message: "This is awesome.", createdAt: date() }]->(p3),

(mi)-[:COMMENTED { uniqueID: "4", message: "This is awesome.", createdAt: date() }]->(p4),

(jo)-[:COMMENTED { uniqueID: "1", message: "Omg! Love it.", createdAt: date() }]->(p1),

(jo)-[:COMMENTED { uniqueID: "3", message: "Omg! Love it.", createdAt: date() }]->(p3),

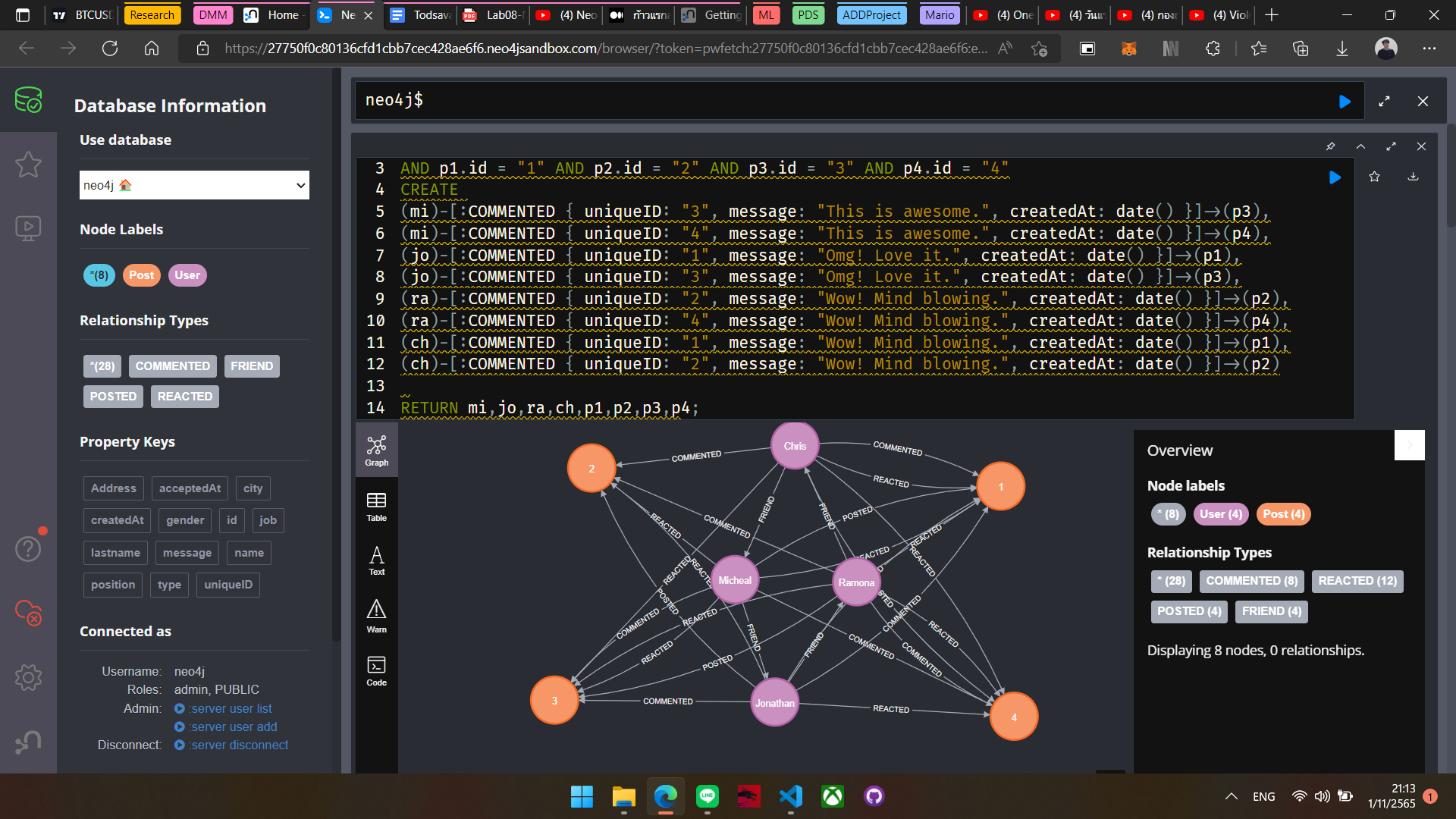
(ra)-[:COMMENTED { uniqueID: "2", message: "Wow! Mind blowing.", createdAt: date() }]->(p2),

(ra)-[:COMMENTED { uniqueID: "4", message: "Wow! Mind blowing.", createdAt: date() }]->(p4),

(ch)-[:COMMENTED { uniqueID: "1", message: "Wow! Mind blowing.", createdAt: date() }]->(p1),

(ch)-[:COMMENTED { uniqueID: "2", message: "Wow! Mind blowing.", createdAt: date() }]->(p2)

RETURN mi,jo,ra,ch,p1,p2,p3,p4;



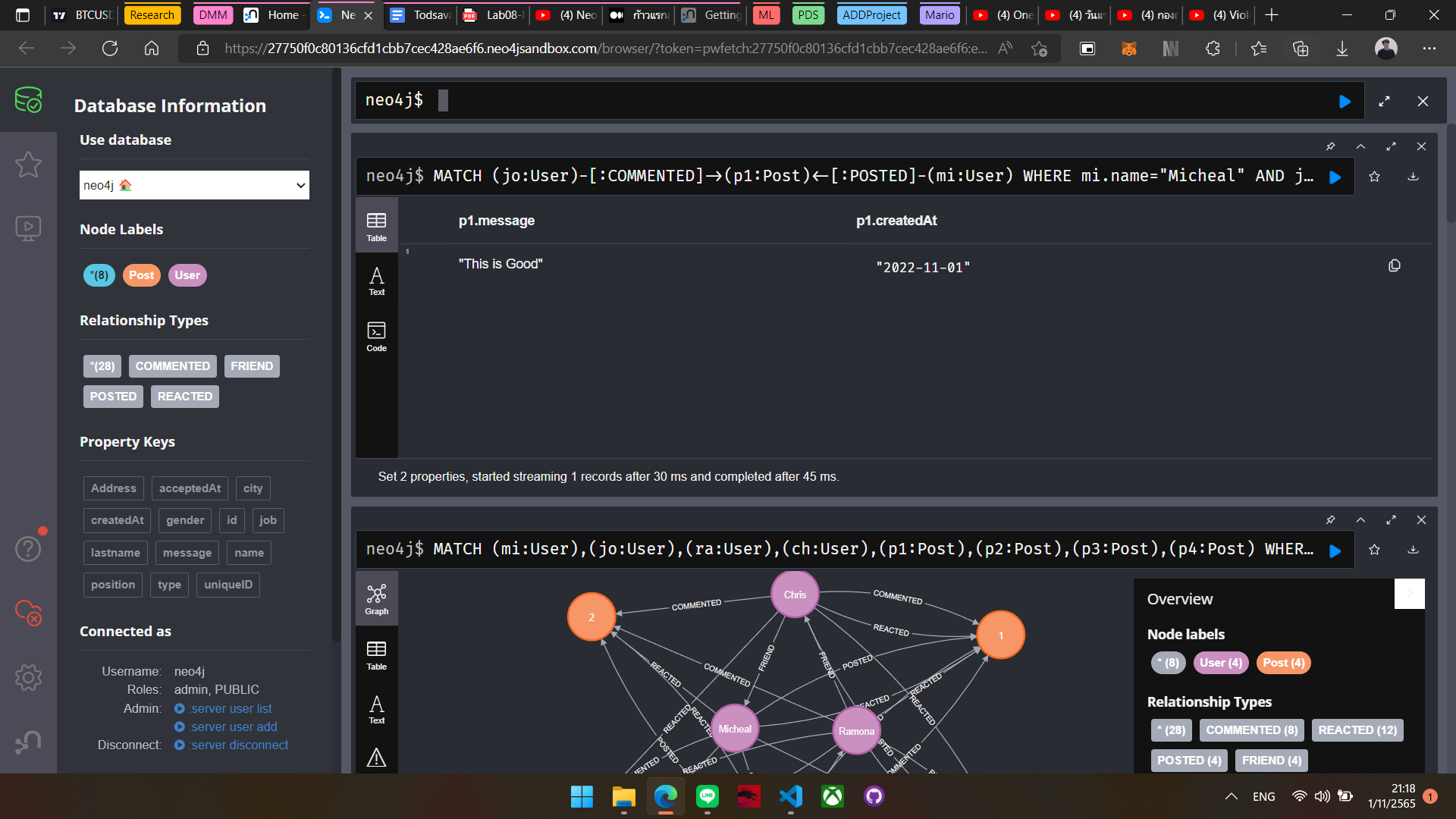
**Task 6:** Suppose Jonathan has made a typo in his comment on Michael’s post. Please show your command to update this comment and record the time and date that Jonathan has edited.

MATCH (jo:User)-[:COMMENTED]->(p1:Post)<-[:POSTED]-(mi:User)

WHERE mi.name="Micheal" AND jo.name="Jonathan"

SET p1.message = "This is Good", p1.createdAt = date()

RETURN p1.message, p1.createdAt

****

**Task 7:** Show the command to count the number of reactions that each user received. Make sure that your command will separate the type of reactions. Also show the result after executing your command.

MATCH (u:User)-[:REACTED]->(p:Post)

RETURN u.name, count(p);