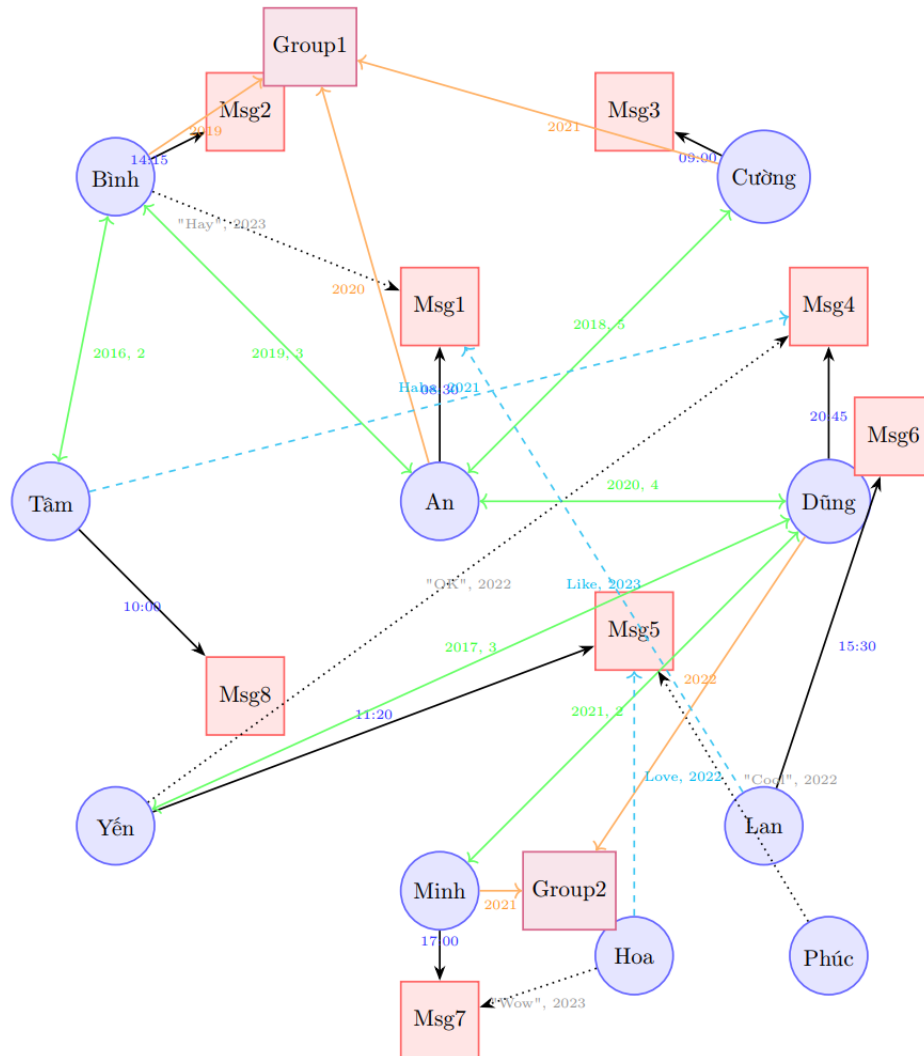


1. Tạo dữ liệu mẫu (20 điểm)

Dùng các bảng dưới đây để tạo dữ liệu trong Neo4j. Mỗi người dùng, tin nhắn, nhóm phải có ít nhất 1 quan hệ. Thêm ít nhất 3 quan hệ "FRIEND" và 2 quan hệ "REACTED_TO" dựa trên hình.



Hình 1: Hình Network Diagram Zalo

CREATE

// Tạo người dùng

```
(AN:User {name: 'An', birthYear: 1990, country: 'Vietnam', activeYears: 8}),
(BINH:User {name: 'Binh', birthYear: 1985, country: 'Vietnam', activeYears: 10}),
(CUONG:User {name: 'Cường', birthYear: 1995, country: 'Vietnam', activeYears: 6}),
(DUNG:User {name: 'Dũng', birthYear: 1988, country: 'Vietnam', activeYears: 9}),
(YEN:User {name: 'Yến', birthYear: 1992, country: 'Vietnam', activeYears: 7}),
(LAN:User {name: 'Lan', birthYear: 1987, country: 'Vietnam', activeYears: 10}),
(MINH:User {name: 'Minh', birthYear: 1993, country: 'Vietnam', activeYears: 6}),
(HOA:User {name: 'Hoa', birthYear: 1996, country: 'Vietnam', activeYears: 5}),
```

(PHUC:User {name: 'Phúc', birthYear: 1991, country: 'Vietnam', activeYears: 7}),
(TAM:User {name: 'Tâm', birthYear: 1989, country: 'Vietnam', activeYears: 9}),

// Tạo tin nhắn

(MSG1:Message {content: 'Msg1', sentYear: 2023, type: 'Text', size: 0.1, views: 50}),
(MSG2:Message {content: 'Msg2', sentYear: 2022, type: 'Image', size: 2.5, views: 200}),
(MSG3:Message {content: 'Msg3', sentYear: 2023, type: 'Video', size: 10, views: 500}),
(MSG4:Message {content: 'Msg4', sentYear: 2021, type: 'Text', size: 0.2, views: 80}),
(MSG5:Message {content: 'Msg5', sentYear: 2022, type: 'Image', size: 3.0, views: 300}),
(MSG6:Message {content: 'Msg6', sentYear: 2023, type: 'Video', size: 15, views: 700}),
(MSG7:Message {content: 'Msg7', sentYear: 2022, type: 'Text', size: 0.3, views: 120}),
(MSG8:Message {content: 'Msg8', sentYear: 2021, type: 'Video', size: 12, views: 400}),

// Tạo nhóm

(GROUP1:Group {name: 'Group1', createdYear: 2019, memberCount: 5}),
(GROUP2:Group {name: 'Group2', createdYear: 2020, memberCount: 4}),

// Quan hệ gửi tin nhắn

(AN)-[:SENT {time: '08:30', sentYear: 2023}]->(MSG1),
(BINH)-[:SENT {time: '14:15', sentYear: 2022}]->(MSG2),
(CUONG)-[:SENT {time: '09:00', sentYear: 2023}]->(MSG3),
(DUNG)-[:SENT {time: '20:45', sentYear: 2021}]->(MSG4),
(YEN)-[:SENT {time: '11:20', sentYear: 2022}]->(MSG5),
(LAN)-[:SENT {time: '15:30', sentYear: 2023}]->(MSG6),
(MINH)-[:SENT {time: '17:00', sentYear: 2022}]->(MSG7),
(TAM)-[:SENT {time: '10:00', sentYear: 2021}]->(MSG8),

// Quan hệ COMMENTED

(YEN)-[:COMMENTED {comment: 'OK', commentYear: 2022}]->(MSG4),
(BINH)-[:COMMENTED {comment: 'Hay', commentYear: 2023}]->(MSG1),
(PHUC)-[:COMMENTED {comment: 'Cool', commentYear: 2022}]->(MSG5),
(HOA)-[:COMMENTED {comment: 'Wow', commentYear: 2023}]->(MSG7),

// Quan hệ REACTED_TO

(LAN)-[:REACTED_TO {reaction: 'Like', reactYear: 2023}]->(MSG1),
(TAM)-[:REACTED_TO {reaction: 'Haha', reactYear: 2021}]->(MSG4),
(HOA)-[:REACTED_TO {reaction: 'Love', reactYear: 2022}]->(MSG5),

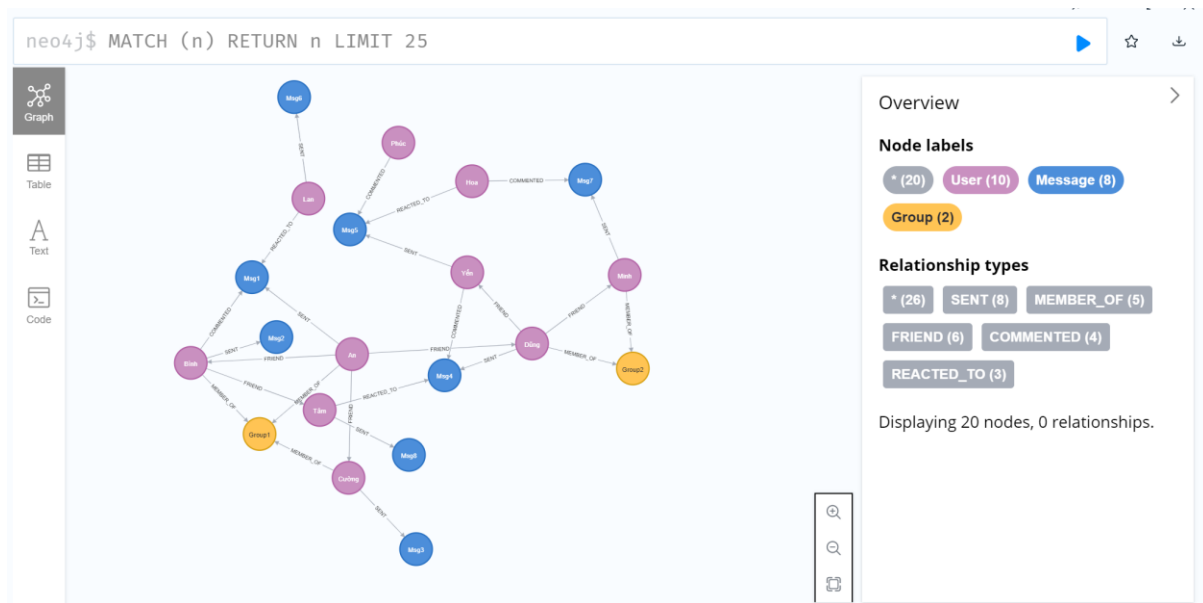
// Quan hệ MEMBER_OF

(AN)-[:MEMBER_OF {joinedYear: 2020}]->(GROUP1),
(BINH)-[:MEMBER_OF {joinedYear: 2019}]->(GROUP1),
(CUONG)-[:MEMBER_OF {joinedYear: 2021}]->(GROUP1),
(DUNG)-[:MEMBER_OF {joinedYear: 2022}]->(GROUP2),
(MINH)-[:MEMBER_OF {joinedYear: 2021}]->(GROUP2),

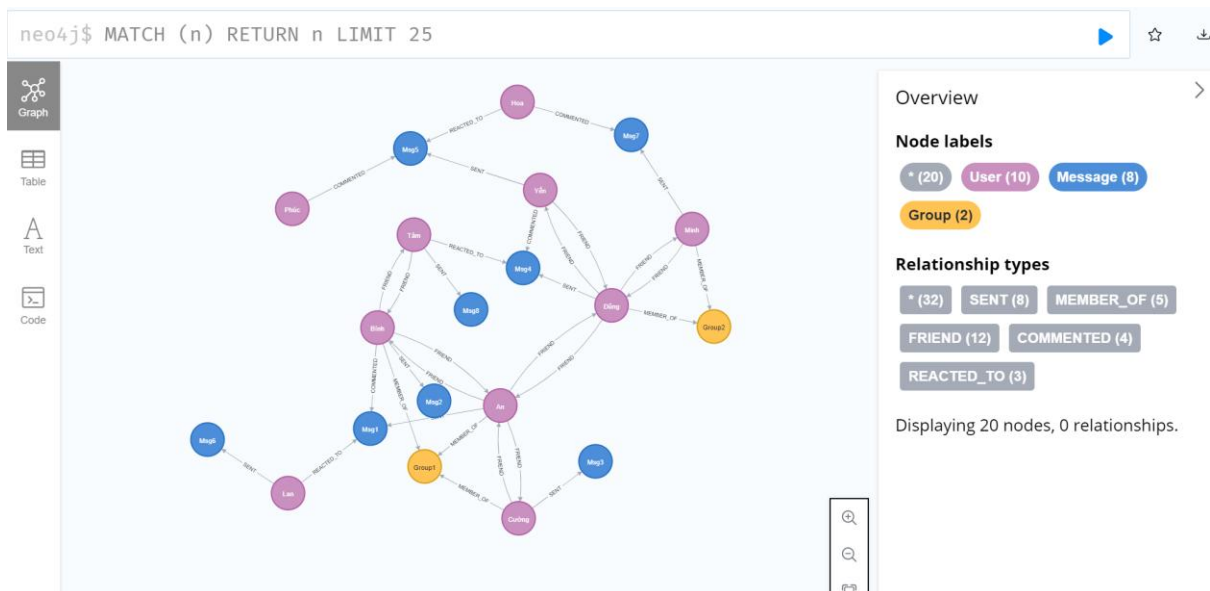
// Quan hệ FRIEND

(AN)-[:FRIEND {friendSince: 2019, messagesShared: 3}]->(BINH),
(AN)-[:FRIEND {friendSince: 2018, messagesShared: 5}]->(CUONG),
(AN)-[:FRIEND {friendSince: 2020, messagesShared: 4}]->(DUNG),
(BINH)-[:FRIEND {friendSince: 2016, messagesShared: 2}]->(TAM),

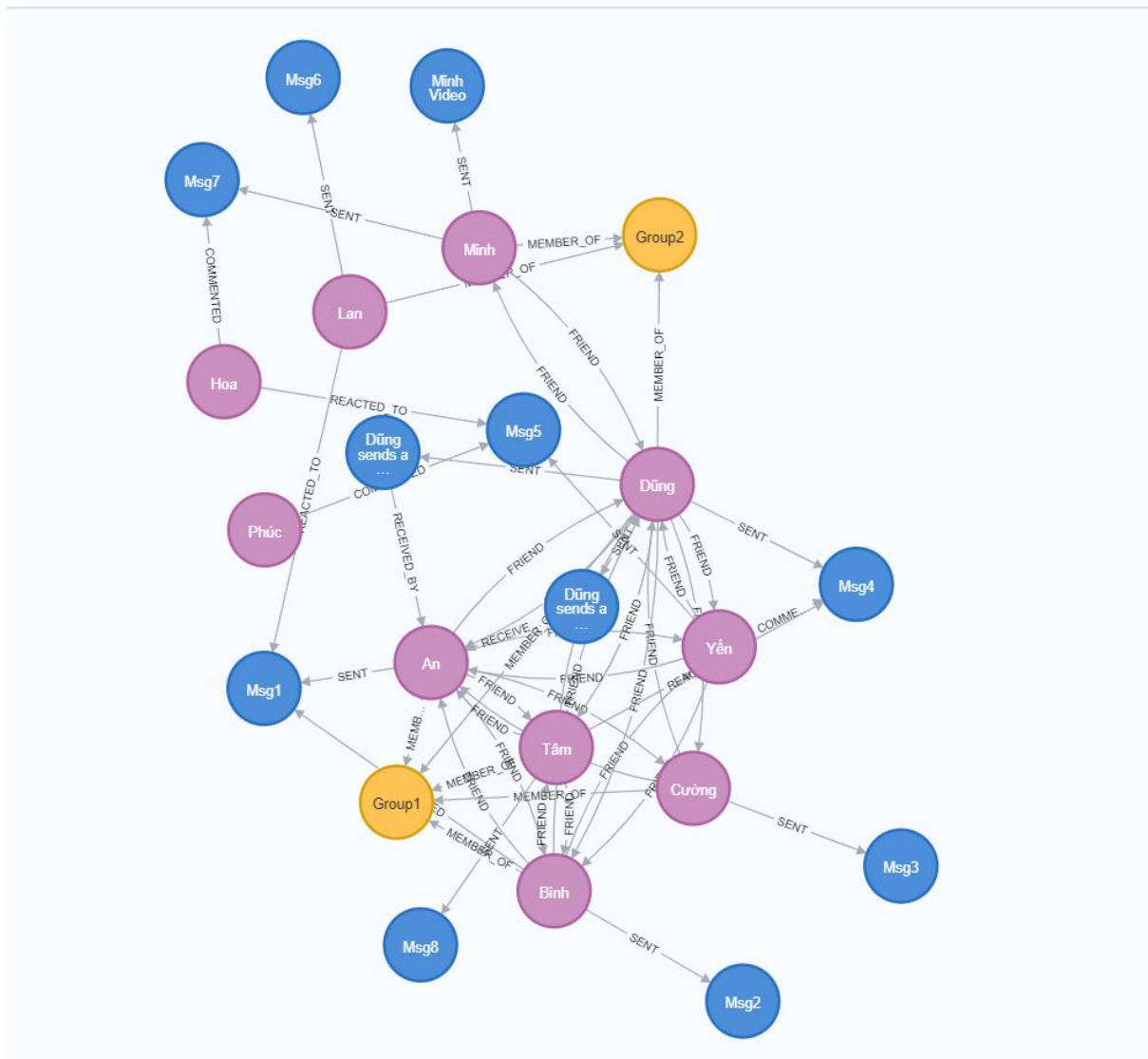
```
(DUNG)-[:FRIEND {friendSince: 2017, messagesShared: 3}]->(YEN),
(DUNG)-[:FRIEND {friendSince: 2021, messagesShared: 2}]->(MINH);
MATCH (n) RETURN n
```



```
MATCH (a:User)-[f:FRIEND]->(b:User)
MERGE (b)-[:FRIEND {friendSince: f.friendSince, messagesShared: f.messagesShared}]->(a);
```



Update 1:



2. Tìm tin nhắn và người dùng (20 điểm)

Dùng bảng và hình "Network Diagram" để trả lời:

1. Tìm tất cả tin nhắn An gửi, nhận bình luận, hoặc phản ứng từ bạn bè qua "FRIEND" hoặc nhóm "MEMBER_OF". Trả về content, sentYear, type, views, và vai trò ("Sender", "CommentedByFriend", "ReactedByGroup").

```

MATCH (an:User {name: "An"})-[:SENT]->(msg:Message)
RETURN msg.content, msg.sentYear, msg.type, msg.views, "Sender" AS role
UNION
MATCH (an:User {name: "An"})-[:FRIEND]->(friend)-[:COMMENTED]->(msg:Message)
RETURN msg.content, msg.sentYear, msg.type, msg.views, "CommentedByFriend" AS role
UNION
MATCH (an:User {name: "An"})-[:MEMBER_OF]->(group)-[:MEMBER_OF]-(member)-[:REACTED_TO]->(msg:Message)
RETURN msg.content, msg.sentYear, msg.type, msg.views, "ReactedByGroup" AS role;

```

| | | | | | |
|---|-------------|--------------|----------|-----------|---------------------|
| neo4j\$ MATCH (an:User {name: "An"})-[:SENT]->(msg:Message) RETURN msg.content, msg.sentYear, msg.type, msg.views, role | | | | | |
| Table | msg.content | msg.sentYear | msg.type | msg.views | role |
| 1 | "Msg1" | 2023 | "Text" | 50 | "Sender" |
| 2 | "Msg1" | 2023 | "Text" | 50 | "CommentedByFriend" |

2. Tìm người dùng có **activeYears** trên 8, gửi ít nhất 2 tin nhắn có **views** trên 200, và thuộc ít nhất 1 nhóm. Trả về **name**, **country**, danh sách tin nhắn (**contents**, **views**), **groupNames**.

MATCH (u:User)

WHERE u.activeYears > 8

// Kiểm tra gửi ít nhất 2 tin nhắn có views > 200

WITH u

MATCH (u)-[:SENT]->(msg:Message)

WHERE msg.views > 200

WITH u, COLLECT({content: msg.content, views: msg.views}) **AS** messages

WHERE SIZE(messages) >= 2

// Kiểm tra thuộc ít nhất 1 nhóm

MATCH (u)-[:MEMBER_OF]->(g:Group)

WITH u, messages, COLLECT(g.name) **AS** groupNames

RETURN u.name, u.country, messages, groupNames;

MATCH (LAN:User {name: 'Lan'}), (GROUP2:Group {name: 'Group2'})

CREATE (LAN)-[:MEMBER_OF {joinedYear: 2021}]->(GROUP2)

RETURN LAN, GROUP2;

Table

RAW

{}

Q

| u.name | u.country | messages | groupNames |
|---------|-----------|--|------------|
| 1 "Lan" | "Vietnam" | [{ content: "Msg6", views: 700 }, { content: "Msg8", views: 400 }] | ["Group2"] |

3. Tìm bạn bè, nhóm và tin nhắn (30 điểm)

Dùng bảng sau và hình "Network Diagram":

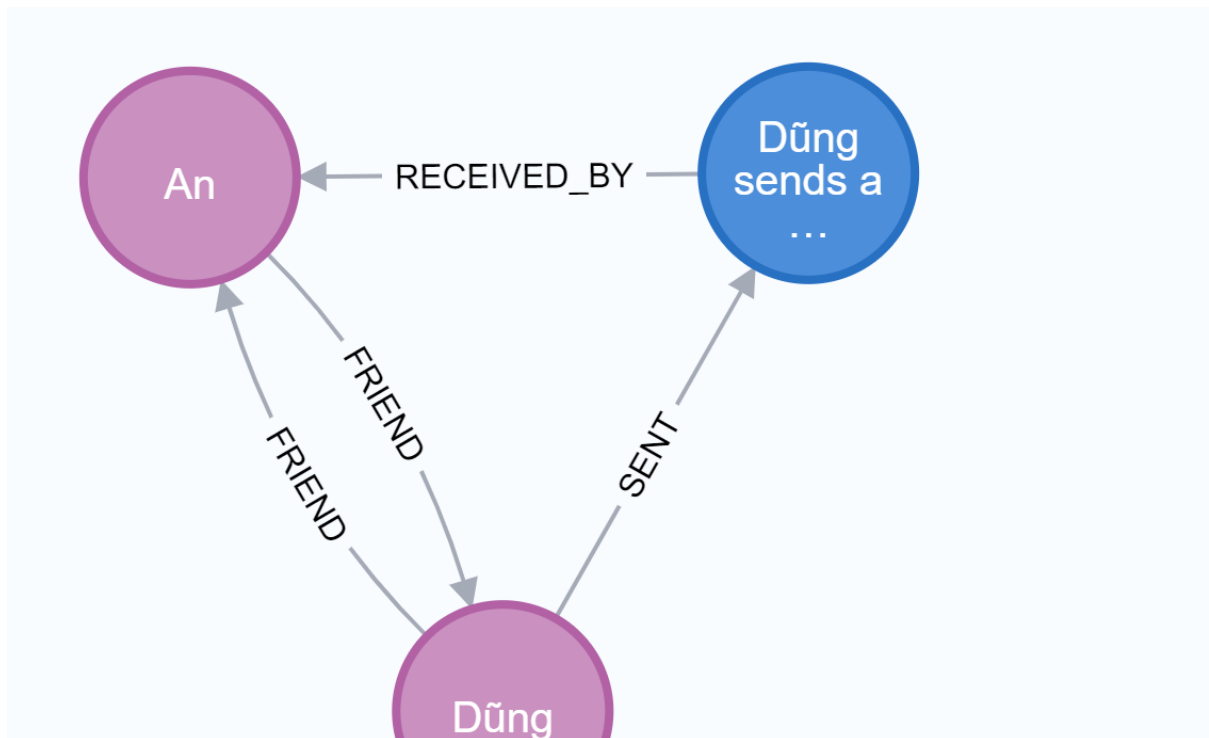
| Ng. | Loại | Kích thước | Lượt xem | Số bạn | Số nhóm |
|------|-------|------------|----------|--------|---------|
| An | Any | > 2 | 300 | 3 | 1 |
| Dũng | Video | > 10 | 400 | 2 | 1 |
| Yến | Image | > 2 | 250 | 2 | 0 |

1. Tìm người gửi tin nhắn kích thước trên 2 MB, lượt xem trên 300, là bạn của An với ít nhất 3 bạn chung qua "FRIEND" và cùng ít nhất 1 nhóm "MEMBER_OF". Trả về name, content, time, size, messagesShared, groupName, sắp xếp theo views giảm dần.

```

MATCH (dung:User {name:'Dũng'}), (an:User {name:'An'})
CREATE (msg:Message {
  content:'Dũng sends a Video Message to An',
  sentYear:2023,
  type:'Video',
  size:12,
  views:450
})
CREATE (dung)-[:SENT {time:'10:00'}]->(msg)
CREATE (msg)-[:RECEIVED_BY]->(an)
RETURN dung, msg, an;

```



```

MATCH (an:User {name: 'An'})-[:FRIEND]->(friend)-[:SENT]->(msg)
MATCH (friend)-[:FRIEND]->(commonFriend)-[:FRIEND]-(an)
MATCH (friend)-[:MEMBER_OF]->(group)-[:MEMBER_OF]-(an)
WHERE msg.size > 2 AND msg.views > 300
WITH friend, msg, group, COUNT(commonFriend) AS mutualFriends
WHERE mutualFriends >= 3
RETURN friend.name AS name, msg.content AS content, msg.time AS time,
       msg.size AS size, msg.views AS views, group.name AS groupName
ORDER BY msg.views DESC;

```

neo4j\$ MATCH (an:User {name: 'An'})-[:FRIEND]->(friend)-[:SENT]->(msg) MATCH (friend)-...

| | name | content | time | size | views | groupName |
|---|--------|------------------------------------|------|------|-------|-----------|
| 1 | "Dũng" | "Dũng sends a Video Message to An" | null | 12 | 450 | "Group1" |

2. Tìm người gửi tin nhắn Video kích thước trên 10 MB, lượt xem trên 400, là bạn của Dũng với messagesShared ít nhất 2 và cùng nhóm. Trả về name, content, size, views, groupName.

```

MATCH (minh:User {name: 'Minh'})
CREATE (msg:Message {content: 'Minh Video', sentYear: 2023, type: 'Video', size: 12, views: 450})
CREATE (minh)-[:SENT {time: '11:00'}]->(msg)
RETURN minh.name AS name, msg.content AS content, msg.size AS size, msg.views AS views;

```


| name | content | size | views |
|--------|--------------|------|-------|
| "Minh" | "Minh Video" | 12 | 450 |

```
MATCH (dung:User {name: 'Dũng'})-[:FRIEND]->(friend)-[:SENT]->(msg)
WHERE msg.type = 'Video' AND msg.size > 10 AND msg.views > 400
```

```
// Kiểm tra điều kiện messagesShared >= 2
```

```
MATCH (dung)-[f:FRIEND]->(friend)
WHERE f.messagesShared >= 2
```

```
// Kiểm tra cùng nhóm với Dũng
```

```
MATCH (friend)-[:MEMBER_OF]->(group)<-[:MEMBER_OF]-(dung)
```

```
RETURN friend.name AS name, msg.content AS content,
       msg.size AS size, msg.views AS views,
       group.name AS groupName
ORDER BY msg.views DESC;
```

| | name | content | size | views | groupName |
|---|--------|--------------|------|-------|-----------|
| 1 | "Minh" | "Minh Video" | 12 | 450 | "Group2" |

- Đếm số tin nhắn Image kích thước trên 2 MB, lượt xem trên 250 mà Yến gửi, bình luận hoặc nhận phản ứng từ bạn bè với messagesShared trên 2. Trả về số lượng.

```
MATCH (yen:User {name: 'Yến'})-[:SENT]->(msg)
WHERE msg.type = 'Image' AND msg.size > 2 AND msg.views > 250
```

```
// Tìm bạn của Yến với messagesShared > 2
```

```
MATCH (yen)-[f:FRIEND]->(friend)
WHERE f.messagesShared > 2
```

```
// Kiểm tra bình luận hoặc phản ứng từ bạn bè
```

```
OPTIONAL MATCH (friend)-[:COMMENTED]->(msg)
OPTIONAL MATCH (friend)-[:REACTED]->(msg)
```

```
RETURN COUNT(DISTINCT msg) AS totalMessages;
```


| | totalMessages |
|---|---------------|
| 1 | 1 |

4. Đường đi và phân tích (30 điểm)

Dùng bảng sau và hình "Network Diagram":

| Từ | Đến | Tin nhắn top | Yếu tố kiểm tra |
|-------|-----|---------------------------|---------------------|
| An | Lan | 4 tin xem nhiều nhất | Đường đi + Nhóm |
| Cường | Hoa | 3 tin kích thước lớn nhất | Loại tin + Phản ứng |

1. Tìm đường ngắn nhất từ An đến Lan qua "SENT", "FRIEND", "COMMENTED", "MEMBER_OF", "REACTED_TO". Tính tổng **views** của 4 tin nhắn có lượt xem cao nhất của An hoặc nhóm của An, kiểm tra tin nào trong số đó nằm trên đường đi và thuộc nhóm nào. Trả về **path**, **totalViews**, **contents**, **views**, **groupNames**.

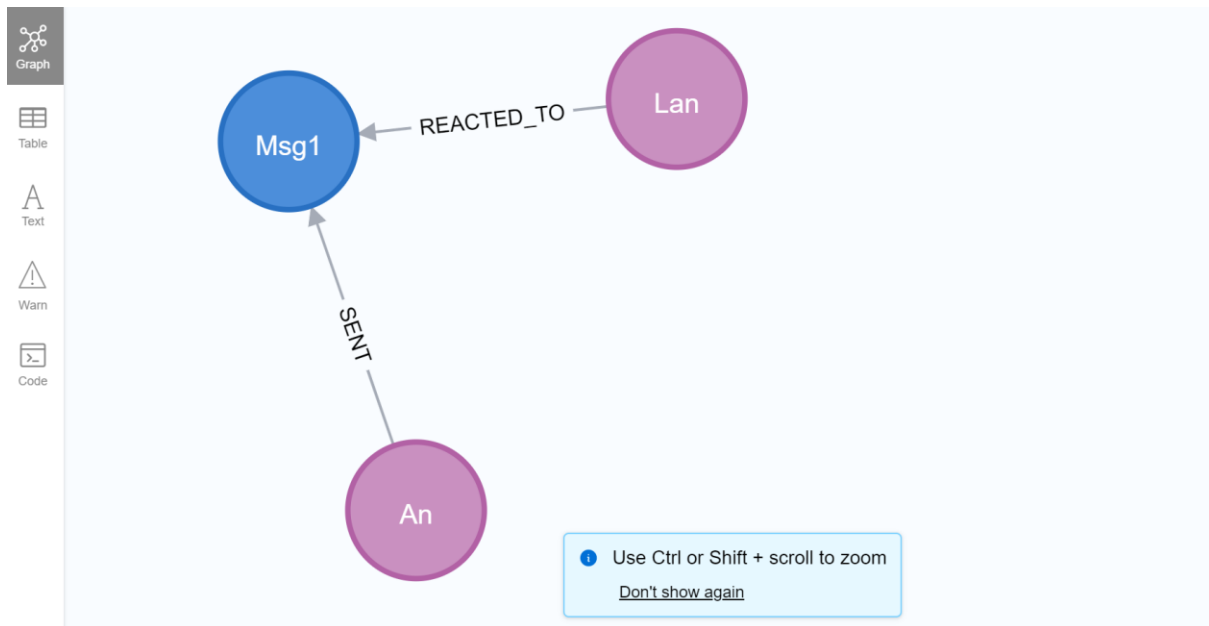
```
MATCH path = shortestPath((an:User {name: "An"})-
[:SENT|FRIEND|COMMENTED|MEMBER_OF|REACTED_TO*]-
(lan:User {name: "Lan"}))
```

```
WITH path, an
MATCH (an)-[:SENT]->(msg:Message)
WHERE msg.views IS NOT NULL
WITH path, an, msg
ORDER BY msg.views DESC
LIMIT 4
```

```
// Tính tổng lượt xem của 4 tin có lượt xem cao nhất
WITH path, an, COLLECT(msg.content) AS contents, SUM(msg.views) AS totalViews
```

```
// Kiểm tra tin nào thuộc nhóm nào
OPTIONAL MATCH (msg)-[:MEMBER_OF]->(group:Group)
```

```
RETURN path, totalViews, contents, COLLECT(DISTINCT group.name) AS groupNames;
```



2. Tìm đường ngắn nhất từ Cường đến Hoa qua các quan hệ trên. Tính tổng **size** của 3 tin nhắn kích thước lớn nhất mà Hoa hoặc bạn bè gửi/bình luận/phản ứng, kiểm tra loại tin và phản ứng trên chúng. Trả về **path**, **totalSize**, **contents**, **types**, **reactions**.

```
MATCH path = shortestPath((cuong:User)-
[:SENT|FRIEND|COMMENTED|MEMBER_OF|REACTED_TO*]-(hoa:User))
WHERE cuong.name = "Cường" AND hoa.name = "Hoa"
```

```
WITH path, hoa
```

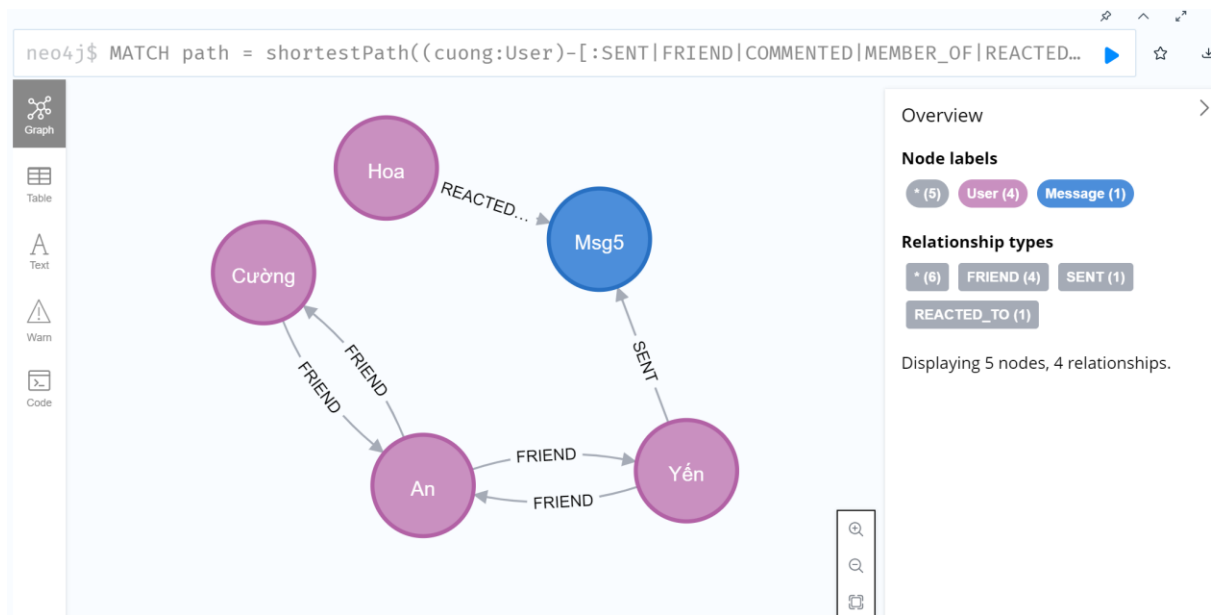
```
MATCH (hoa)-[:SENT|COMMENTED|REACTED_TO]->(msg:Message)
WHERE msg.size IS NOT NULL
```

```
WITH path, hoa, msg
ORDER BY msg.size DESC
LIMIT 3
```

```
WITH path, hoa, COLLECT(msg.size) AS sizes, SUM(msg.size) AS totalSize,
COLLECT(msg.content) AS contents,
COLLECT(msg.type) AS types
```

```
OPTIONAL MATCH (msg)-[:HAS_REACTION]->(r:Reaction)
WITH path, totalSize, contents, types, COLLECT(r.name) AS reactions
```

```
RETURN path, totalSize, contents, types, reactions
```



- Đếm số người có "FRIEND" với ít nhất 4 người khác và thuộc ít nhất 1 nhóm có `memberCount` trên 4. Trả về số lượng.

```
MATCH (u:User)-[:FRIEND]->(friend)
WITH u, COUNT(friend) AS friendCount
WHERE friendCount >= 4
```

```
MATCH (u)-[:MEMBER_OF]->(g:Group)
WHERE g.memberCount > 4
```

```
RETURN COUNT(DISTINCT u) AS totalUsers;
```

| | totalUsers |
|---|------------|
| 1 | 3 |