Bài tập lớn BigData

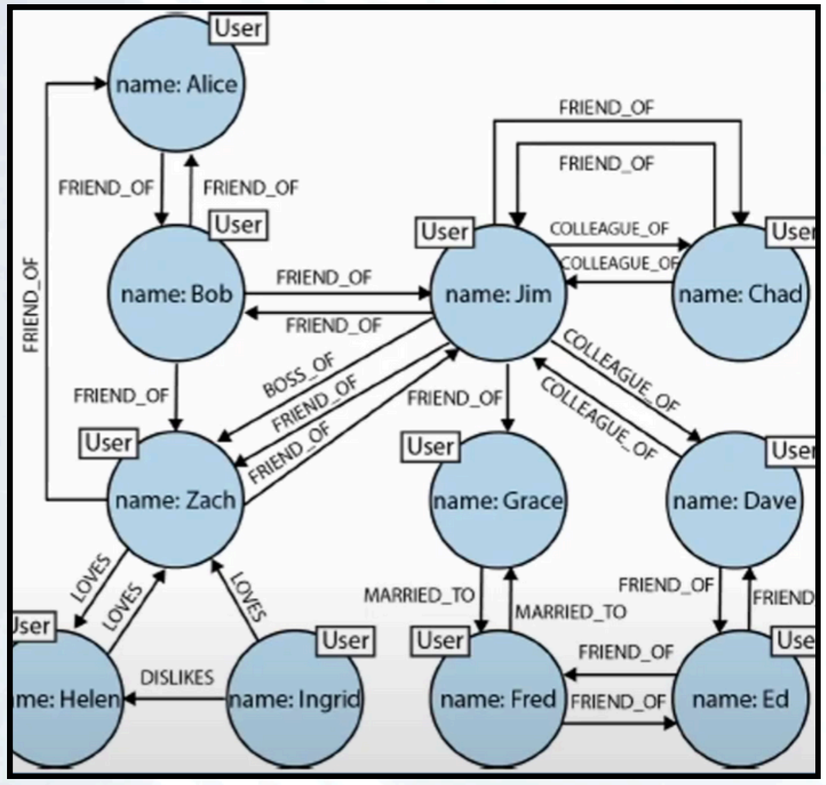
Nhóm 4

Graph Analytics With Chat Data Using Neo4j

Thành viên nhóm

- 22022522 Đàm Thái Ninh
- 22022653 Long Trí Thái Sơn
- 22022548 Hoàng Đăng Khoa

Graph DataBase

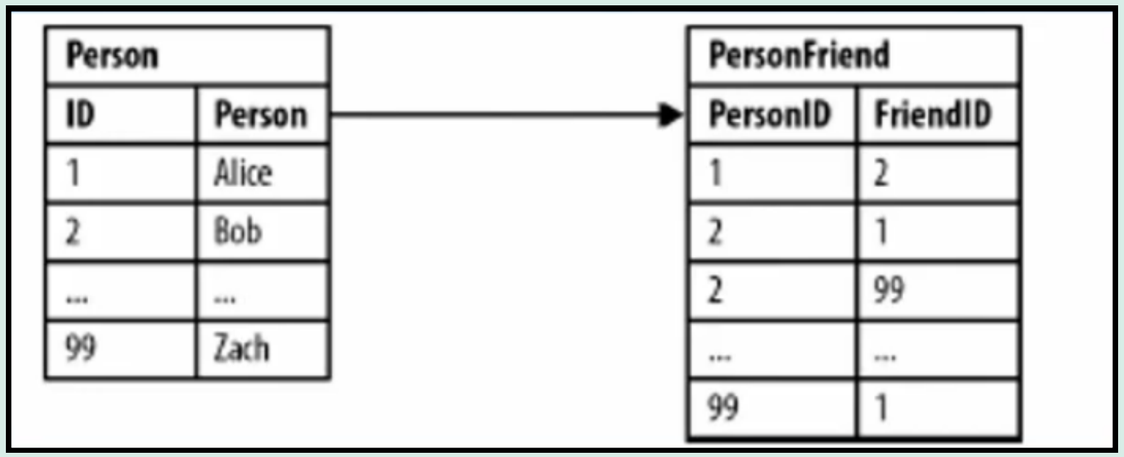


Graph Data sử dụng các nút, cạnh và thuộc tính để biểu diễn và lưu trữ dữ liệu

Lợi ích

- Hiệu quả
- Linh hoạt
- Nhanh chóng

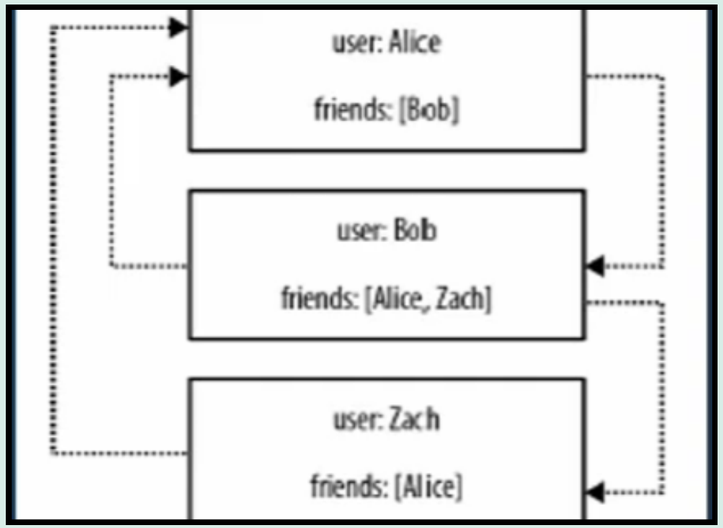
SQL, NOSQL và Graph Database



Person	
ID	Person
1	Alice
2	Bob
...	...
99	Zach

PersonFriend	
PersonID	FriendID
1	2
2	1
2	99
...	...
99	1

SQL

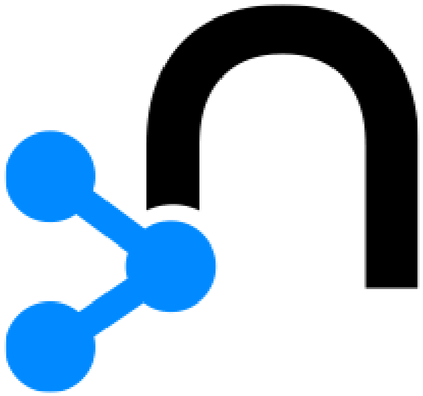


NOSQL

Depth	RDBMS execution time(s)	Neo4j execution time(s)	Records returned
2	0.016	0.01	~2500
3	30.267	0.168	~110,000
4	1543.505	1.359	~600,000
5	Unfinished	2.132	~800,000

Source: Neo4j in Action

Neo4j



Là gì?

Neo4j là một cơ sở dữ liệu đồ thị, cho phép lưu trữ và truy vấn mối quan hệ giữa các dữ liệu một cách hiệu quả

Neo4j Browser

Thực hiện truy vấn

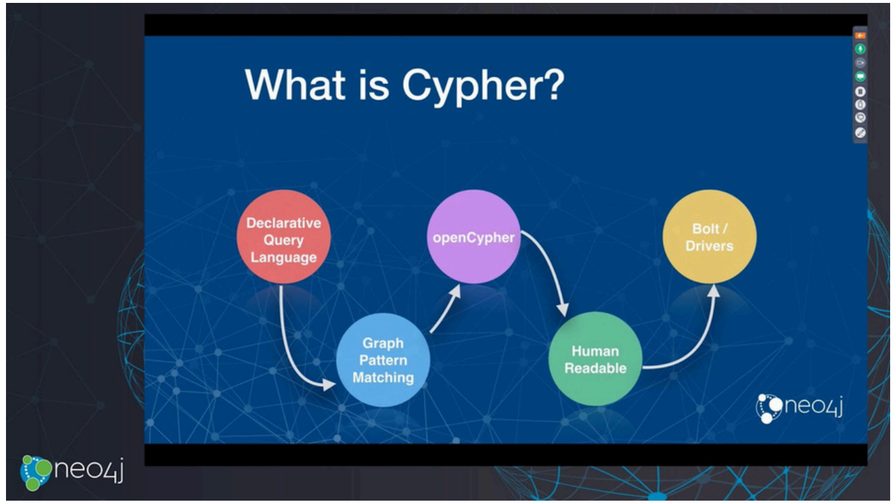
Cách dùng

Có thể tạo ràng buộc duy nhất cho các nút và cạnh, sau đó tải dữ liệu từ các tệp CSV và xây dựng đồ thị

Neo4j Bloom

Visualize dữ liệu

Cypher – Ngôn ngữ truy vấn của Neo4j



Tạo Node

```
CREATE (n:Person {name: 'Alice', age: 30})
```

Tạo Relationship

```
MATCH (a:Person {name: 'Alice'}), (b:Person {name: 'Bob'})
CREATE (a)-[:FRIEND_OF]->(b)
```

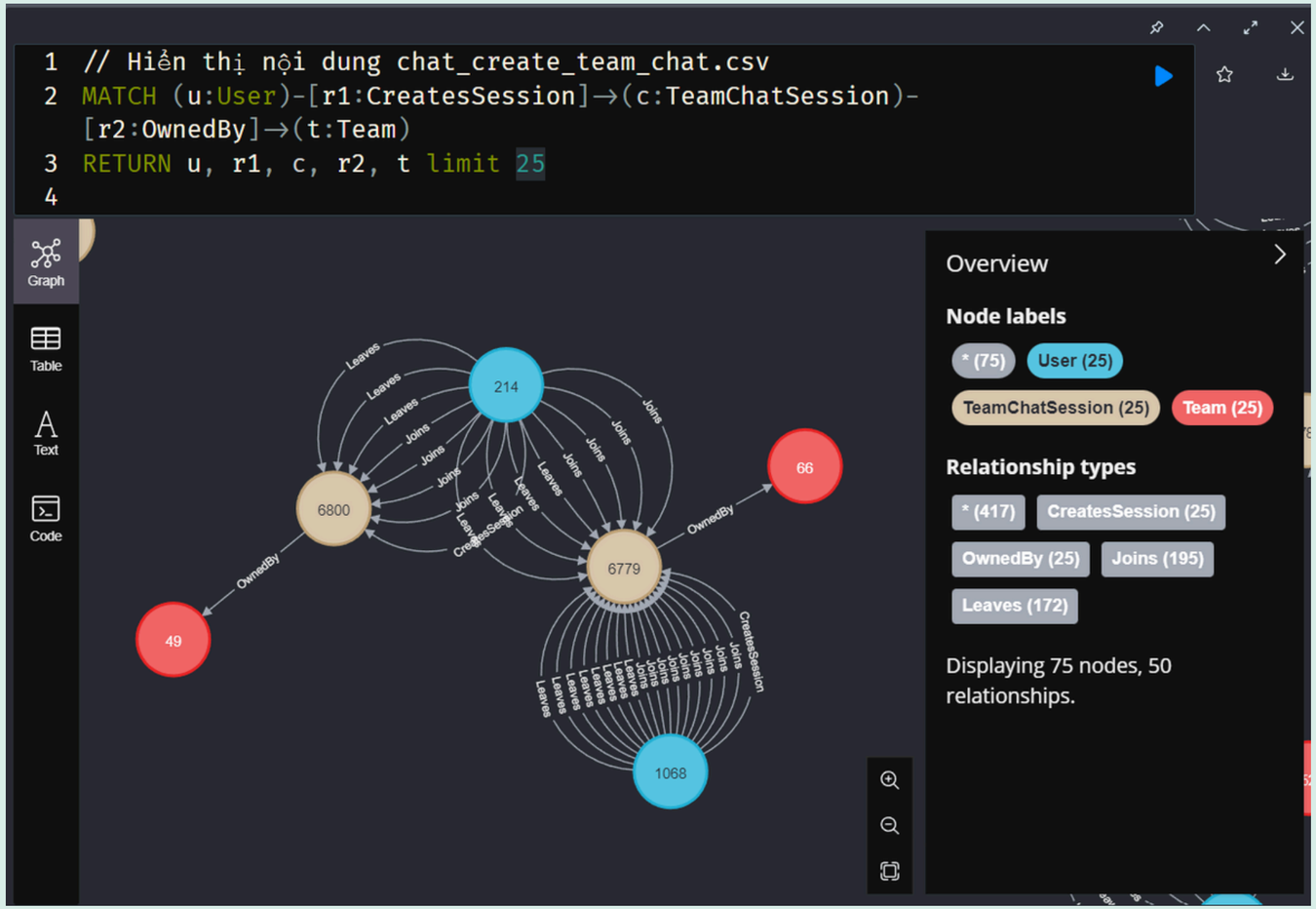
Truy vấn Node

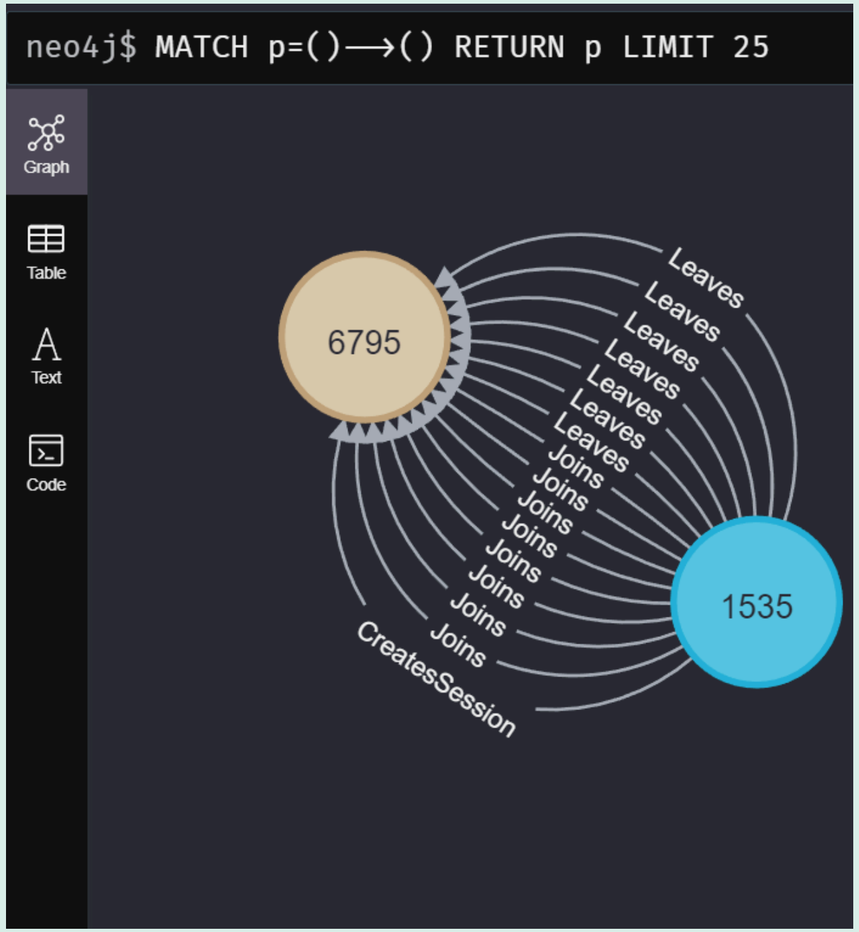
```
MATCH (n:Person)
RETURN n
```

Graph Analytics of Catch the Pink Flamingo Chat Data Using Neo4j

File Name	Description	Fields
chat_create_team_chat.csv	userid	the user id assigned to the user
	teamid	the id of the team
	teamChatSessionID	a unique id for the chat session
	timestamp	a timestamp denoting when the chat session created
chat_item_team_chat.csv	userid	the user id assigned to the user
	teamchatsessionid	a unique id for the chat session
	chatitemid	a unique id for the chat item
	timestamp	a timestamp denoting when the chat item created
chat_join_team_chat.csv	userid	the user id assigned to the user
	teamChatSessionID	a unique id for the chat session
	timestamp	a timestamp denoting when the user join in a chat session
chat_leave_team_chat.csv	userid	the user id assigned to the user
	teamchatsessionid	a unique id for the chat session
	timestamp	a timestamp denoting when the user leave a chat session
chat_mention_team_chat.csv	ChatItemId	the id of the ChatItem
	userid	the user id assigned to the user
	timeStamp	a timestamp denoting when the user mentioned by a chat item
chat_respond_team_chat.csv	chatid1	the id of the chat post 1
	chatid2	the id of the chat post 2
	timestamp	a timestamp denoting when the chat post 1 responds to the chat post 2

Graph Analytics





Graph Analytics

```
// Question 1-a
// Find the longest conversation chain in the chat data using the
"ResponseTo" edge label. This question has two parts
// How many chats are involved in it?
MATCH p=(start:ChatItem)-[:ResponseTo*]→(end:ChatItem)
RETURN p, length(p) AS longestChainLength
ORDER BY longestChainLength DESC
LIMIT 1;
```

Graph | Table | Text | Code

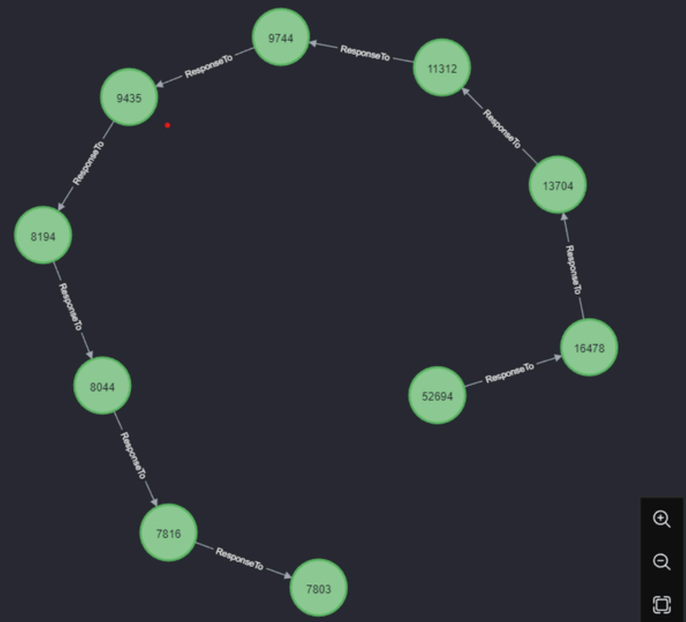
p	longestChainLength
1	9


```
{
  "start": {
    "identity": 35255,
    "labels": [
      "ChatItem"
    ],
    "properties": {
      "id": 52694
    },
    "elementId": "4:87c785a9-239c-46ad-9c38-352ec785cd86:35255"
  },
  "end": {
    "identity": 963,
    "labels": [
      "ChatItem"
    ],
```


Started streaming 1 records after 8 ms and completed after 96 ms.


```
// Question 1-a
// Find the longest conversation chain in the chat data using the
"ResponseTo" edge label. This question has two parts
// How many chats are involved in it?
MATCH p=(start:ChatItem)-[:ResponseTo*]→(end:ChatItem)
RETURN p, length(p) AS longestChainLength
ORDER BY longestChainLength DESC
LIMIT 1;
```


Graph | Table | Text | Code



Overview

Node labels

* (10) | ChatItem (10)

Relationship types

* (9) | ResponseTo (9)

Displaying 10 nodes, 9 relationships.

Graph Analytics

Graph Analytics

Xác định xem những người chat nhiều nhất có tham gia team tạo nhiều chat nhất không?

```
// Q2-b

// Chattiest Teams

// Query to find the top 10 chattiest teams
MATCH (ci:ChatItem)-[:PartOf]→(tcs:TeamChatSession)-[:OwnedBy]→
(t:Team)
RETURN t.id AS teamId, count(ci) AS chatCount
ORDER BY chatCount DESC
LIMIT 10;
```

Table | Text | Code

	teamId	chatCount
1	82	1324
2	185	1036
3	112	957
4	18	844
5	194	836
6	129	814

```
// Chattiest Users

// Query to find the top 10 chattiest users
MATCH (u:User)-[:CreatesChat]→(c:ChatItem)
RETURN u.id AS userId, count(c) AS chatCount
ORDER BY chatCount DESC
LIMIT 10;
```

Table | Text | Code

	userId	chatCount
1	394	115
2	2067	111
3	1087	109
4	209	109
5	554	107
6	1627	105

Started streaming 10 records after 7 ms and completed after 40 ms.

Graph Analytics

Xác định xem những người chat nhiều nhất có tham gia team tạo nhiều chat nhất không?

Graph Analytics

```
1
2  MATCH
3  (u:User)-[c:CreatesChat]→()
4  WITH u, COUNT(c) as Chats
5  ORDER BY Chats DESC LIMIT 10 WITH [u] as ChattiestUsers
6  //Getting the neighbours of all Users and the count
7  MATCH (u1:User)-[:InteractsWith]→(u2:User)
8  WHERE u1 in ChattiestUsers
9  WITH u1.id AS UserID, COLLECT(DISTINCT u2.id) AS Neighbours RETURN
   UserID, Neighbours, SIZE(Neighbours) AS k
```

Table | Text | Code

	UserID	Neighbours	k
1	394	[1012, 2011, 1997, 1782]	4
2	2067	[63, 516, 1265, 1672, 209, 1627]	6
3	1087	[1311, 426, 929, 772, 1879, 1098]	6
4	209	[63, 516, 1627, 2067, 1672]	5
5	554	[2018, 1959, 1687, 1096, 1010]	5
6	1627	[516, 2067, 209, 63, 1672, 1265]	6
7			

Started streaming 10 records after 11 ms and completed after 74 ms.

Xác định những người hoạt động tích cực nhất

Những khó khăn gặp phải

- Vấn đề khi chạy trên WSL
- Lạ lẫm với Cypher
- Gặp nhiều vấn đề khi tích hợp với Spark

Dự định

- Tiếp tục tìm cách xử lý Neo4j với Spark
- Có thể sẽ sử dụng Docker

Demo

- Các câu lệnh đơn giản
- Graph Analytics of Catch the Pink Flamingo Chat Data Using Neo4j

QA

Xin chân thành cảm ơn