

Тестовое задание по графовым базам данных от MindSet

Дарья Минина, 2023

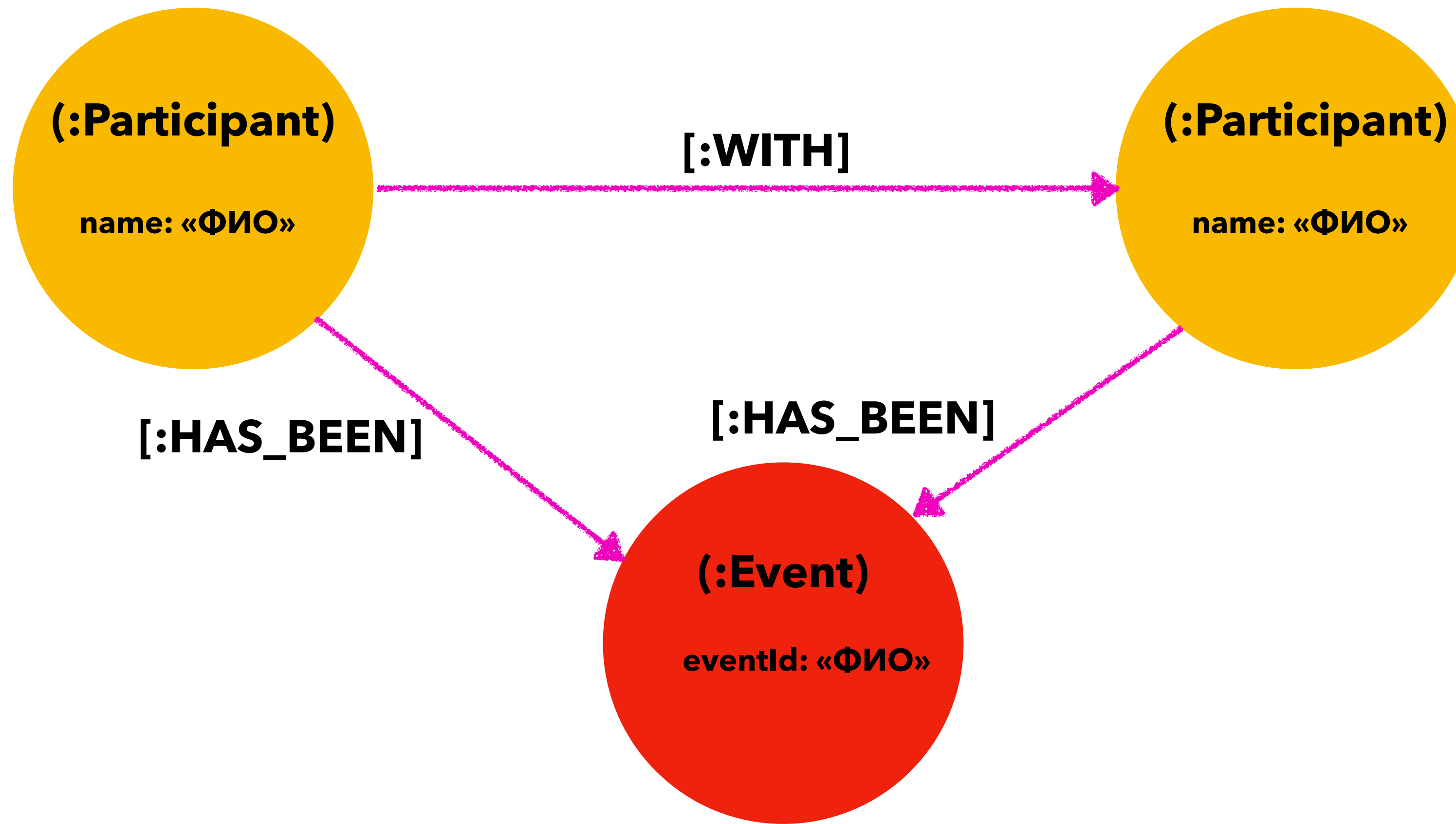
Задание

1. Установить 2 графовые базы из списка DB-Engines Ranking (Предпочтительные - neo4j, nebula, arangodb), предпочтительный язык запросов cypher
2. Создать ipynb ноутбук в котором:
3. Считать данные из источника Yandex
4. Внести данные из таблицы в графовую БД
5. Построить графовое представление в БД, осуществить несколько запросов на языке запросов к графовой БД
6. Найти взаимосвязи визуально и с помощью алгоритмов (алгоритмы на ваше усмотрение)
7. Написать rest сервис на python к графовой БД в котором на вход поступает ФИО, на выходе graphml или json
8. Результаты представить на гитхаб и в виде кода + небольшой презентации
9. Прислать ссылку на решение и резюме в телеграм @frankshikhaliev
10. Также надо будет заполнить форму
11. Срок выполнения задания - около 10 дней, если вы не успеваете можете взять больше времени

Используемые графовые БД:
Neo4j, Memgraph

Rest сервис написан в Jupiter Notebook

Модель графа



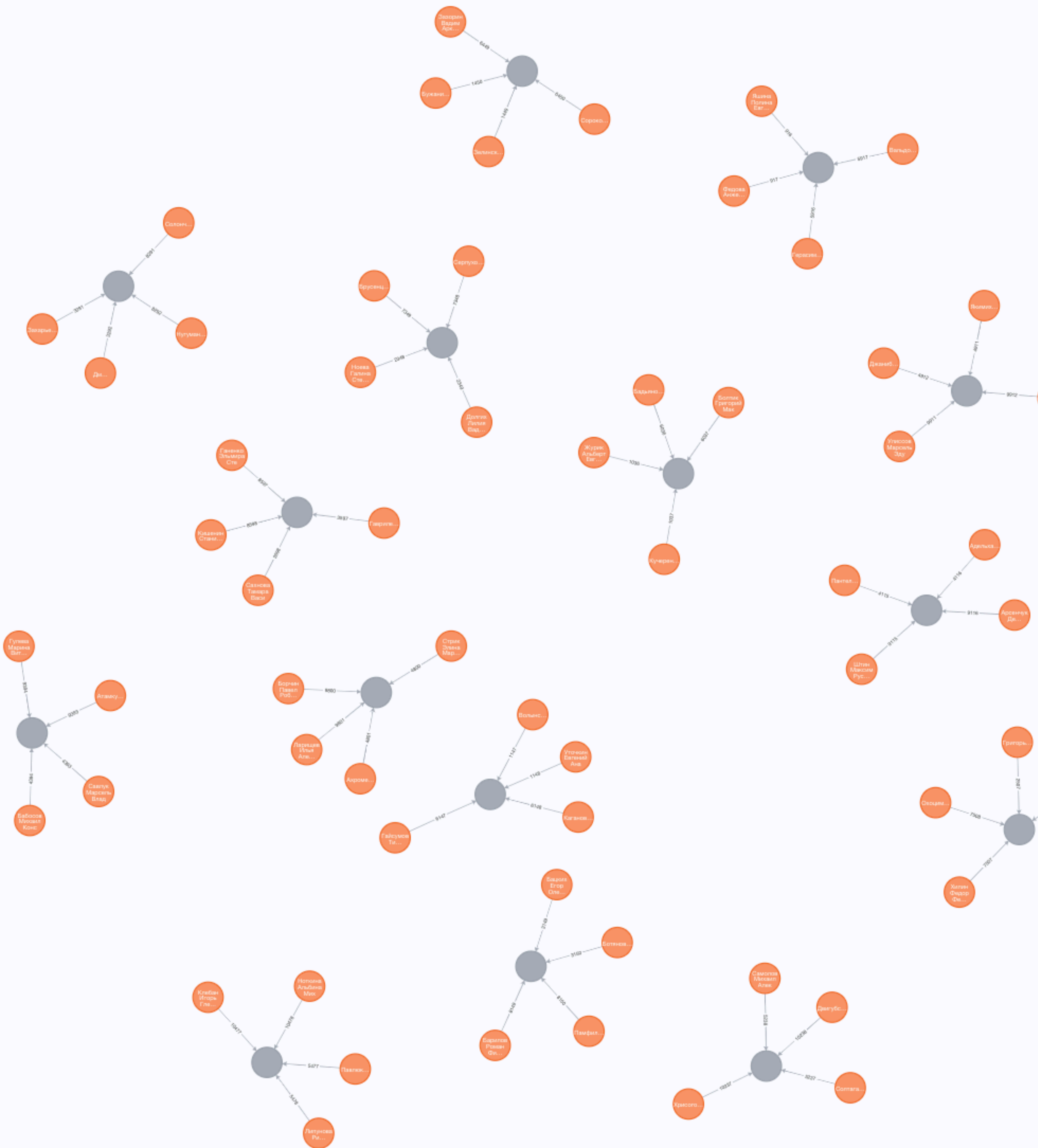
Выполнение запросов к БД язык Cypher

```
LOAD CSV WITH HEADERS FROM 'file:///data_test_com.csv'  
AS row  
WITH row WHERE row.`id события` IS NOT NULL  
MERGE (c:Event {eventId: row.`id события`});
```

Рис. 1. Пример импорта данных из файла

```
MATCH (n:Participant) - [r:HAS_BEEN] -> (e:Event)  
RETURN e, COLLECT(n) as p  
ORDER BY SIZE(p) DESC LIMIT 15
```

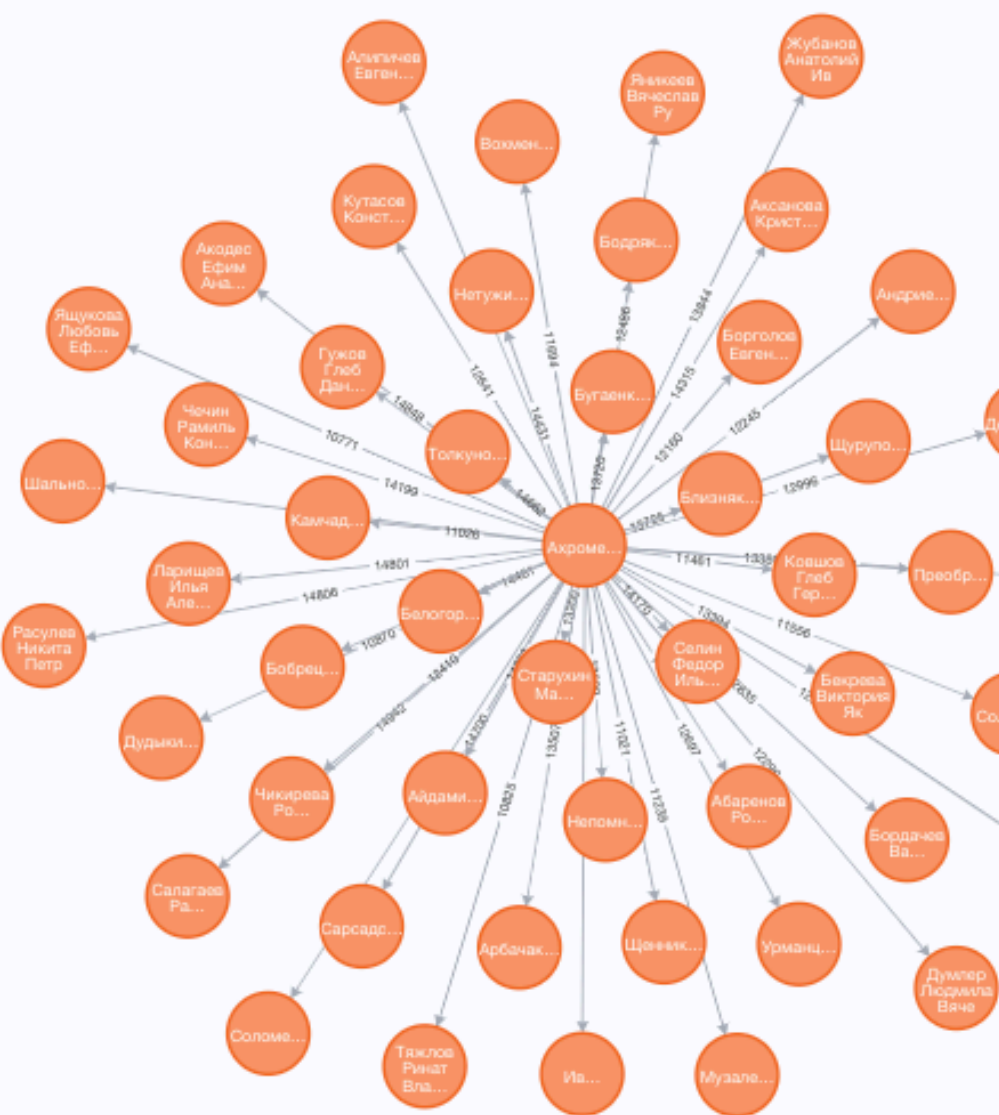
Рис. 2. Выведем события с наибольшим числом участников (4)



Поиск ТОП-5 самых «популярных» людей из первого и второго столбцов.

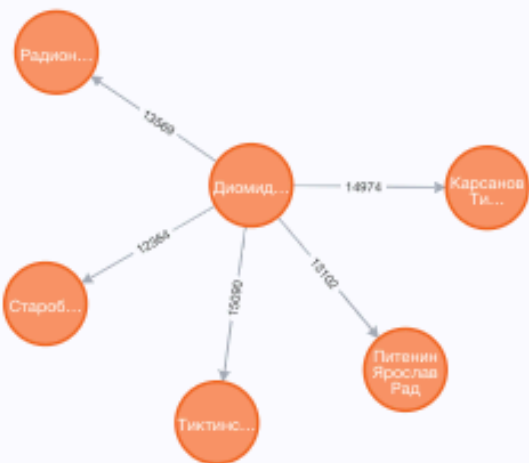
```
MATCH (p1:Participant) - [r:WITH] -> (p2:Participant)
RETURN p1, COLLECT(p2) as p
ORDER BY SIZE(p) DESC LIMIT 5
```

Рис. 3. Из первого столбца



```
MATCH (p1:Participant) - [r:WITH] -> (p2:Participant)
RETURN p2, COLLECT(p1) as p
ORDER BY SIZE(p) DESC LIMIT 5
```

Рис. 4. Из второго столбца



REST API на базе Jupiter Notebook

```
# POST /graph
req = json.loads(REQUEST)
body = json.dumps(req['body'])

if "name" not in body:
    print(json.dumps({"name": None}))
else:
    name = str(json.loads(body)['name'])
    events = []
    with GraphDatabase.driver(URI, auth=AUTH) as driver:
        with driver.session() as session:
            events = session.execute_read(get_events, name)
    print(json.dumps({
        "headers": {
            "Content-Type": "application/json"
        },
        "status": 200,
        "body": {"id событий": events},
    })))
```

Данные запроса: имя участника события в формате ФИО
Данные ответа: id событий, в которых участвовал человек

http://127.0.0.1:8889/graph

POST

http://127.0.0.1:8889/graph

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSC

1

{

2

... "name": "Шолохов Игорь Робертович"

3

}

Body

Cookies

Headers (4)

Test Results

Pretty

Raw

Preview

Visualize

JSON

↺

1

{

2

"headers": {

3

"Content-Type": "application/json"

4

},

5

"status": 200,

6

"body": {

7

"id событий": [

8

"750824",

9

"829652",

10

"875321",

11

"218462"

12

]

13

}

14

}