

# ОТЧЕТ

Лабораторная работа 2.1

Выполнил студент группы АДЭУ-221,

Бойко Константин

## Введение

Цель работы.

Изучить методы хранения данных на основе NoSQL. (neo4j)

Краткое описание используемых данных и их структуры.

### Узлы:

Company (Компания)

Employee (Сотрудник)

Project (Проект)

Client (Клиент)

### Отношения:

WORKS\_AT (Сотрудник работает в компании)

MANAGES (Сотрудник управляет проектом)

PARTICIPATES\_IN (Сотрудник участвует в проекте)

HAS\_PROJECT (Компания имеет проект)

IS\_CLIENT\_OF (Клиент связан с проектом)

### Типы данных:

Company {id: 1, name: "ABC Corp", industry: "IT"}

Employee {id: 1, name: "John Doe", position: "Developer", company\_id: 1}

Project {id: 1, name: "Project X", start\_date: "2023-01-01", end\_date: "2023-06-01", company\_id: 1}

Client {id: 1, name: "Client A", project\_id: 1}

Создадим все узлы и отношения через bash mobaexterm, войдя в докер с neo4j.

```
neo4j@neo4j> CREATE (:Company {id: 1, name: "ABC Corp", industry: "IT"}),
                  (:Company {id: 2, name: "XYZ Inc", industry: "Finance"}),
                  (:Company {id: 3, name: "DEF Ltd", industry: "Retail"}),
                  (:Company {id: 4, name: "GHI Group", industry: "Manufacturing"}),
                  (:Company {id: 5, name: "JKL Co", industry: "Healthcare"});
0 rows
ready to start consuming query after 3320 ms, results consumed after another 0 ms
Added 5 nodes, Set 15 properties, Added 5 labels
```

**1 Задание.** Найдите все компании, имеющие более 7 сотрудников.

**Решение:**

```
MATCH (c:Company)-[:WORKS_AT]-(e:Employee)
```

```
WITH c, count(e) AS employeeCount
```

```
WHERE employeeCount > 7
```

```
RETURN c.name AS CompanyName, employeeCount;
```

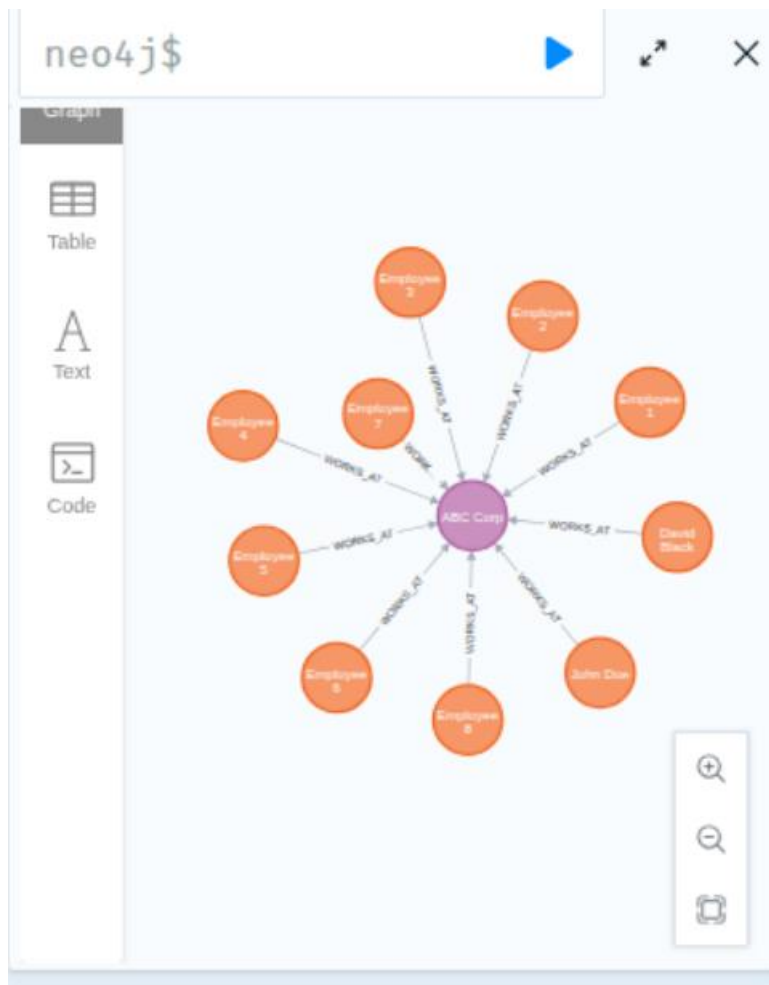
**Результат:**

```
neo4j@neo4j> MATCH (c:Company)-[:WORKS_AT]-(e:Employee)
              WITH c, count(e) AS employeeCount
              WHERE employeeCount > 7
              RETURN c.name AS CompanyName, employeeCount;
+-----+
| CompanyName | employeeCount |
+-----+
|              |                |
```

Таких компаний в БД просто нет.

Можем в этом убедиться, отправив запрос, отображающий компании в которых более 1 сотрудника:

```
neo4j@neo4j> MATCH (c:Company)-[:WORKS_AT]-(e:Employee)
              WITH c, count(e) AS employeeCount
              WHERE employeeCount > 1
              RETURN c.name AS CompanyName, employeeCount;
+-----+
| CompanyName | employeeCount |
+-----+
| "ABC Corp"  | 2             |
| "XYZ Inc"   | 2             |
| "DEF Ltd"   | 2             |
| "GHI Group" | 2             |
| "JKL Co"    | 2             |
+-----+
```

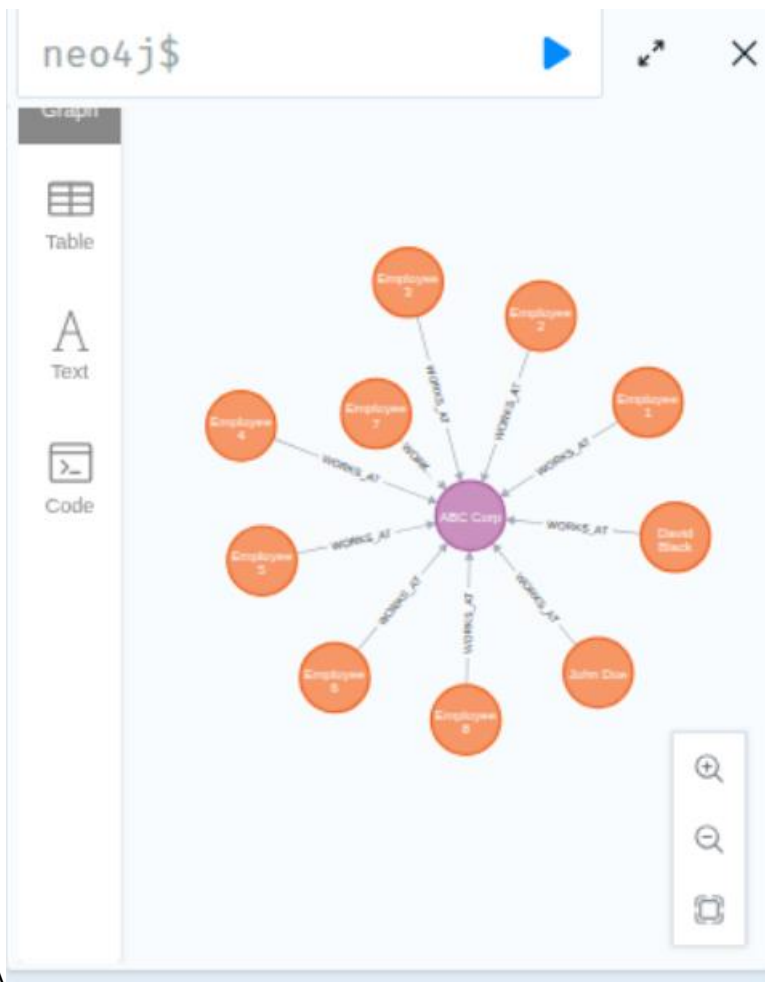


Граф в neo4j browser подтверждает истинность результата запроса:



CREATE

```
(e1)-[:WORKS_AT]->(c),  
(e2)-[:WORKS_AT]->(c),  
(e3)-[:WORKS_AT]->(c),  
(e4)-[:WORKS_AT]->(c),  
(e5)-[:WORKS_AT]->(c),  
(e6)-[:WORKS_AT]->(c),  
(e7)-[:WORKS_AT]->(c),  
(e8)-[:WORKS_AT]->(c);
```



**2 Задание.** Найдите всех сотрудников, работающих в компании "DEF Ltd".

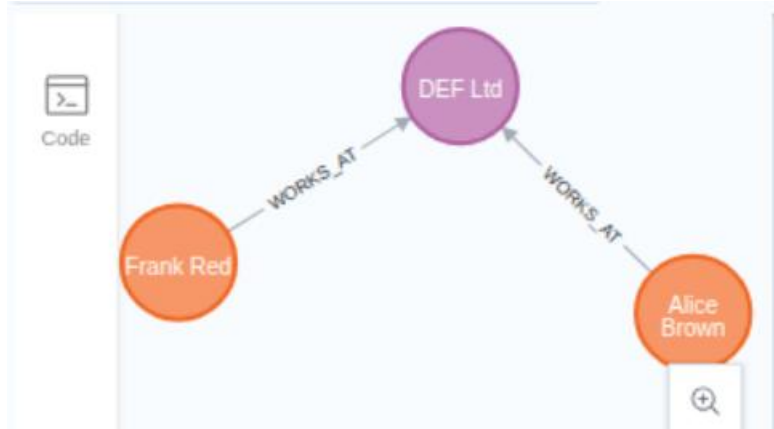
**Решение:**

```
MATCH (e:Employee)-[:WORKS_AT]->(c:Company {name: "DEF Ltd"})
```

```
RETURN e.name AS EmployeeName, e.position AS Position;
```

**Результат:**

```
+-----+
| EmployeeName | Position |
+-----+
| "Frank Red"  | "Analyst" |
| "Alice Brown" | "Analyst" |
+-----+
```



**3 Задание.** Найдите все проекты, начавшиеся в июле 2023 года.

**Решение:**

```
MATCH (p:Project) WHERE p.start_date >= "2023-07-01" AND p.start_date < "2023-08-01"
RETURN p.name AS ProjectName, p.start_date AS StartDate;
```

**Результат:**

```
+-----+
| ProjectName | StartDate |
+-----+
| "Project D" | "2023-07-01" |
+-----+
```

Граф визуально неинформативен в этом запросе.

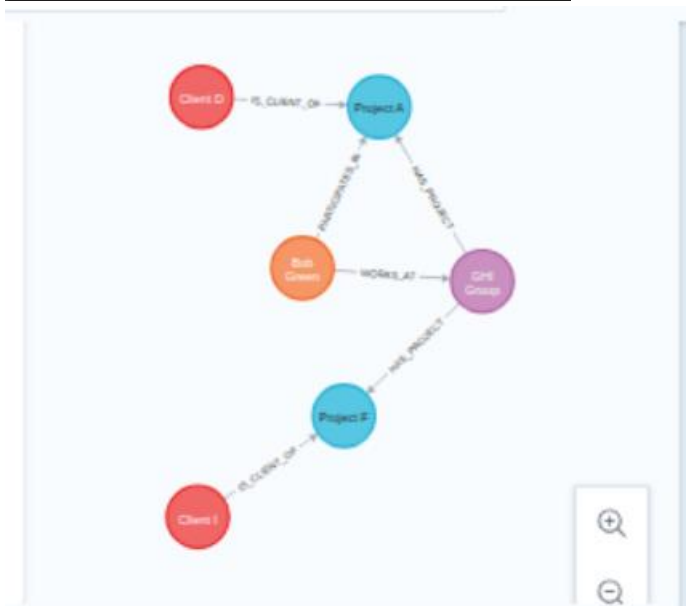
**4 Задание.** Найдите всех клиентов, связанных с проектами компании "GHI Group"

**Решение:**

```
MATCH (c:Company {name: "GHI Group"})-[:HAS_PROJECT]->(p:Project)<-[:IS_CLIENT_OF]-
(cl:Client) RETURN cl.name AS ClientName, p.name AS ProjectName;
```

**Результат:**

ClientName	ProjectName
"Client I"	"Project F"
"Client D"	"Project A"



**5 Задание.** Найдите все проекты, в которых сотрудник "Frank Red" является аналитиком.

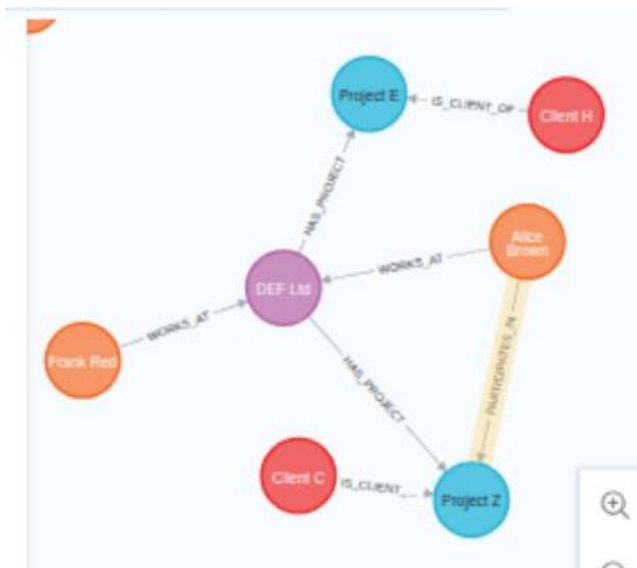
**Решение:**

```
MATCH (e:Employee {name: "Frank Red", position: "Analyst"})-[r:PARTICIPATES_IN]->(p:Project) RETURN p.name AS ProjectName, p.start_date, p.end_date;
```

**Результат:**

ProjectName	p.start_date	p.end_date

Frank не участвует как аналитик ни в одном проекте своей компании.



## Выводы

Что было изучено?

Я вспомнил как осуществляется подключение к виртуальному окружению Ubuntu с помощью `mobaxterm`, познакомился с графовой СУБД `neo4j`, создал в ней несколько узлов и отношений между ними. Сформулировал несколько простых `Cypher` запросов для поиска взаимоотношений данных в узлах. Проверил их на графовой визуализации.

Какие сложности возникли и как они были решены?

Я не сразу создал узел с клиентами, но исправил это во время выполнения задания. Во время вайб кодинга возникли спорные результаты запросов, и мне пришлось аналитически исправлять запросы погружаясь в структуру предложенных в работе данных.

Какие навыки были получены?

Навык взаимодействия с `neo4j browser` и `cypher` запросами.