

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 5
по дисциплине «Постреляционные базы данных»

Тема: «Создание графовой базы данных и работа с ней на примере СУБД
Neo4j»

ИСПОЛНИТЕЛЬ:
группа ИУ5-24М

Журавлев Н.В.
ФИО

подпись

"30" марта 2024 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В.
ФИО

подпись

" " _____ 202_ г.

Москва - 2024

Цель работы

- Изучить модель представления данных и способы работы с графовыми БД;
- Освоить методы создания графовой БД и языки запросов к ней;
- Получить навыки работы с графовой СУБД Neo4j.

Задание

1. Создать в среде Neo4j базу данных по теме, выданной преподавателем. Определить набор узлов, задать их свойства и метки.
2. Продемонстрировать (вывести на экран) содержимое БД (узлы и их свойства), используя команды Match/Where/Return.
3. Создать набор узлов БД. Определить для них несколько меток и свойств. Создать отношение между новыми узлами. Создать отношение между существующими узлами.
4. Продемонстрировать удаление узлов и связей. Продемонстрировать удаление и изменение свойств и меток.
5. Продемонстрировать содержимое БД (фильтрация по узлам, отношениям, меткам и связям).
6. Выполнить запросы к базе данных на языке Cypher:
 - a. с условием NOT NULL
 - b. операторами AND, OR
 - c. с сортировкой
 - d. с условием на направление отношения
 - e. с параметрами отношения
7. Продемонстрировать работу команд LIMIT, SKIP
8. Продемонстрировать работу команд UNION, MERGE.
9. Выполнить запросы к базе данных на языке Cypher:
 - a. с агрегированием,
 - b. с встроенными функциями (строковые или иные),
 - c. с указанием начального узла (точка входа),
 - d. с шаблонами отношений,

е. с удалением дубликатов

Ход работы

Для создания в среде Neo4j базу данных по выбранной теме, определения наборов узлов, задания их свойства и метки, использовались следующие команды:

```
CREATE (a1: account {login: 'login', password: 'password', weight:80, height:180, age:18, special: 'больной'});  
CREATE (a2: account {login: 'log', password: 'pass', weight:90, height:190, age:19, special: 'здоровый'});  
CREATE (a3: account {login: 'logg', password: 'passw', weight:70, height:170, age:17, special: 'здоровый'});  
CREATE (d1: diet {name: 'name diet', description: 'description', weight:80, height:180, age:18, special: 'здоровый'});  
CREATE (d2: diet {name: 'name', description: 'desc', weight:80, height:180, age:18, special: 'здоровый'});  
CREATE (d3: diet {name: 'nn', description: 'des', weight:90, height:190, age:19, special: 'больной'});  
CREATE (ds1: dish {name: 'name dish', description: 'description dish'});  
CREATE (ds2: dish {name: 'name', description: 'desc'});  
CREATE (ds3: dish {name: 'nn', description: 'des'});
```

Можно вывести содержимое БД, используя Match/Where/Return. Пример такой команды:

```
MATCH (a:account) WHERE a.login='login' RETURN a.password;
```

Для создания отношения между новыми узлами для этого используется следующее строение команды:

```
CREATE (d: diet {name: 'name diet', description: 'description', weight:60, height:160, age:16, special: 'здоровый'})-[r:-CONSIDER{full:'Yes'}]->(ds: dish {name: 'name dish', description: 'description'});
```

Для создания отношения между существующими узлами, используется:

```
MATCH (d: diet), (ds: dish) WHERE d.name='name' AND ds.name='name dish' CREATE (d)-[r:CONSIDER]->(ds) RETURN d,ds;
```

Для удаления узлов и связей, используются следующие команды:

```
MATCH (a:account {height: 180}) DETACH DELETE a;  
MATCH (d:diet) - [rel: CONSIDER] -> () DELETE rel;
```

Для удаления и изменение свойств и меток, можно использоваться следующие команды:

```
MATCH (a:account) WHERE a.special='больной' REMOVE a.special RETURN a;  
MATCH (a:account) WHERE a.special='здоровый' REMOVE a: account RETURN a;  
MATCH (d:diet {name: 'name diet'}) SET d.name='diet name' RETURN d.name;  
MATCH (ds: dish) SET ds: dishes RETURN ds;
```

Для фильтрации по узлам, отношениям, меткам и связям, используются следующего вида команды:

```
MATCH (d)-[rel:CONSIDER]-(ds) RETURN d, ds;  
MATCH (a:account) WHERE a.login = "login" RETURN a;  
MATCH (a:account) WHERE a:account RETURN a;  
MATCH ((d)-->(ds)) RETURN d,ds;
```

Для выполнения запросов к базе данных с условием NOT NULL, необходимо добавить эту метку к команде:

```
MATCH (a:account) WHERE a.age IS NOT NULL RETURN a;
```

Для выполнения запросов к базе данных операторами AND, OR, необходимо добавить их в условие:

```
MATCH (a:account) WHERE a.age=18 AND a.weight=80 AND a:account RETURN a;  
MATCH (a:account) WHERE (a.age=18 OR a.age=19) AND a:account RETURN a;
```

Для выполнения запросов к базе данных с сортировкой, необходимо добавить ORDER BY:

```
MATCH (d:diet) RETURN d.age, d.name ORDER BY d.age;
```

Для выполнения запросов к базе данных с условием на направление отношения необходима команда следующего вида:

```
MATCH (d:diet)-->(ds:dish) RETURN d, ds;
```

Для выполнения запросов к базе данных с параметрами отношения, необходимо дополнить предыдущую команду следующим:

```
MATCH (d)-[rel:CONSIDER]->(ds) WHERE rel.full='Yes' RETURN d, ds;
```

Для работы команд LIMIT, SKIP необходимо добавить их в конец запроса с указанием параметров:

```
MATCH (n) RETURN n LIMIT 2;  
MATCH (n) RETURN n SKIP 2;
```

Для работы команд UNION, MERGE, необходимо команды следующего вида:

```
MATCH (d:diet) WHERE d.age=18 RETURN d UNION MATCH (d:diet) WHERE d.age > 18  
RETURN d;  
MATCH (d:diet{name: 'name'}), (ds:dish{name:'name'}) MERGE (d)-[rel: CONSIDER]->(ds)  
RETURN d,ds;
```

Для выполнения запросов к базе данных с агрегированием, нужно добавить агрегирующую функцию в RETURN:

```
MATCH (a: account) RETURN min(a.age);
```

Пример выполнения запросов к базе данных со встроенными функциями:

```
MATCH (a: account) RETURN substring(login, 2, 2);
```

Пример выполнения запросов к базе данных с шаблонами отношений:

```
MATCH ((d)-->(ds)) RETURN d,ds;
```

Для выполнения запроса с удалением дубликатов необходимо использовать DISTINCT:

```
MATCH (a:account) RETURN DISTINCT a.special;
```

Вывод

В результате выполнения работы были изучены модель представления

данных и способ работы с графовыми БД, так же освоены методы создания графовой БД и языки запросов к ней. Так же были получены навыки работы с графовой СУБД Neo4j.

Список используемой литературы

1. Neo4j – Текст. Изображение: электронные // Neo4j: [сайт]. – URL: Дистрибутив и документация – <https://neo4j.com/> (дата обращения: 15.12.2018)
2. Виноградов В.И., Виноградова М.В. Постреляционные модели данных и языки запросов: Учебное пособие. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2017. — 100с. - ISBN 978-5-7038-4283-6. URL: <http://ebooks.bmstu.ru/catalog/254/book1615.html> (дата обращения: 26.04.2022). - Текст. Изображение : электронные.
3. Маркин, А. В. Системы графовых баз данных. Neo4j : учебное пособие для вузов / А. В. Маркин. — Москва : Издательство Юрайт, 2021. — 303 с. — (Высшее образование). — ISBN 978-5-534-13996-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/467452> (дата обращения: 19.05.2022).
4. Робинсон Ян, Вебер Джим, Эйфрем Эмиль. Графовые базы данных: новые возможности для работы со связанными данными / пер. с англ. Р.Н. Рагимова; науч. Ред. А.Н. Кисилев. — 2-е изд. — М.: LVR Пресс, 2016. — 256 с.: ил. ISBN 978-5-97060- 201-0 5. OpenSypher – Текст. Изображение: электронные //
5. OpenSypher: [сайт]. – URL: <https://opencypher.org/cips/> (дата обращения: 01.05.2022)
6. Фаулер, Мартин, Садаладж, Прамодкумар Дж. NoSQL: новая методология разработки нереляционных баз данных. : Пер. с англ. - М.: ООО "И.Д. Вильямс", 2013г.
7. Neo4j Documentation: macOS installation – Текст. Изображение: электронные // Neo4j Docs: [сайт]. – URL: <https://neo4j.com/docs/operations-manual/current/installation/osx/> (дата

обращения: 01.05.2022)

8. Neo4j Documentation: CREATE – Текст. Изображение: электронные // Neo4j Docs: [сайт]. – URL: <https://neo4j.com/docs/cypher-manual/current/clauses/create/> (дата обращения: 01.05.2022)
9. Neo4j Documentation: SET – Текст. Изображение: электронные // Neo4j Docs: [сайт]. – URL: <https://neo4j.com/docs/cypher-manual/current/clauses/set/> (дата обращения: 01.05.2022)
10. Neo4j Documentation: REMOVE – Текст. Изображение: электронные // Neo4j Docs: [сайт]. – URL: <https://neo4j.com/docs/cypher-manual/current/clauses/remove/> (дата обращения: 01.05.2022)