**ЛАБОРАТОРНАЯ РАБОТА №8**

**ИССЛЕДОВАНИЕ ПРИНЦИПОВ РАБОТЫ С ГРАФОВЫМИ БД**

**8.1 Цель работы**

Рассмотреть основные принципы моделирования структуры данных с использованием графовой модели, познакомиться с наиболее популярными сферами деятельности, где возможно более эффективное её применение по сравнению с реляционной моделью, выполнить настройку программной среды для работы с СУБД Neo4j.

**8.2 Порядок выполнения работы**

8.2.1 Выполнить инсталляцию сервера базы данных Neo4j и инструменты разработчика, создать новый проект и базу данных.

8.2.2 Разработать базу данных для предметной области по вариантам. Не менее 5 типов узлов и 5 типов связей. Написать код, генерирующий узлы и связи между узлами.

8.2.3 Используя полученную базу данных, выполнить и описать результаты запросов, представленных в рамках данной лабораторной работы (поиск, сортировка, фильтрация, изменение данных, удаление данных, использование агрегатных функций).

8.2.4 Измерить скорость выполнения запросов для следующего количества узлов: 100, 1000, 10 000.

8.2.5 Оформить отчет по работе.

**8.3 Индивидуальный вариант №3**

Предметная область: Социальная сеть.

**8.4 Ход работы**

8.4.1 Был установлен Neo4j Desktop, создано новое подключение и база данных (рисунок 8.1).

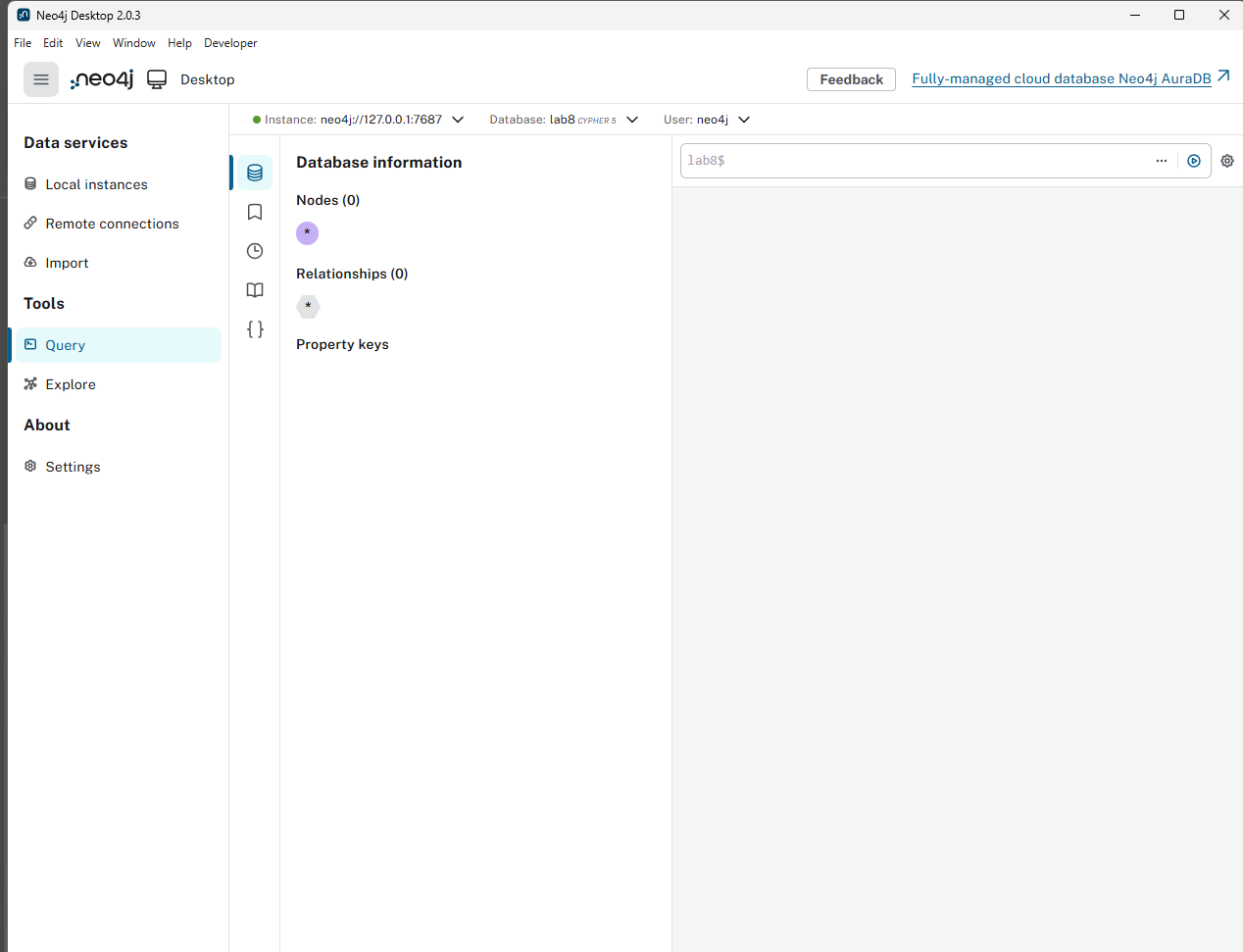


Рисунок 8.1 – Окно Neo4j Desktop

8.4.2 Были определены типы узлов и связей для базы данных. Типы были отражены в таблице (таблицы 8.1 – 8.2)

Таблица 8.1 – Типы узлов

|  |  |  |
| --- | --- | --- |
| Тип узла | Описание | Свойства |
| User | Пользователь | Id, name, email, registered\_at, avatar\_url |
| Post | Пост в соц. сети | Id, content, created\_at, media\_url |
| Comment | Комментарий, оставленный пользователем | Id, text, created\_at |
| Group | Сообщество | Id, name, created\_at, desctiption, logo\_url |
| Story | История | Id, created\_at, content\_url, expires\_at |

Таблица 8.2 – Типы связей

|  |  |
| --- | --- |
| Типы связей | Описание |
| FOLLOWS | Пользователь подписан на другого пользователя |
| FRIEND | Пользователи друзья |
| LIKED | Пользователь лайкнул пост, комментарий или историю |
| CREATED | Пользователь опубликовал пост, комментарий или историю |
| MEMBER\_OF | Пользователь состоит в сообществе |
| COMMENTED\_ON | Пользователь оставил комментарий под постом |

8.4.3 Был написан код для генерации тестовых данных (листинги 8.1-).

Листинг 8.1 – Код запросов для генерации пользователей

WITH range(1, 100) AS ids

UNWIND ids AS id

CREATE (:User {

id: id,

name: "User\_" + id,

email: "user\_" + id + "@example.com",

registered\_at: date() - duration({days: id}),

avatar\_url: "avatar\_" + id + ".png"

});

В листинге 8.1 генерируются 100 записей, для генерации большего количества узлов, число заменяется на соответствующее.

Листинг 8.2 – Код запроса для генерации постов и связей с пользователями

MATCH (u:User)

WITH u, rand() AS r

WHERE r < 0.7

WITH u, toInteger(rand() \* 5) AS postCount

UNWIND range(1, postCount) AS i

CREATE (p:Post {

id: u.id \* 10 + i,

content: "Post #" + u.id + "\_" + i,

created\_at: datetime() - duration({minutes: i}),

media\_url: "media\_" + u.id + "\_" + i + ".jpg"

}),

(u)-[:CREATED]->(p);

Здесь учтено, что не все пользователи публикуют посты (для примера взяли 70%). Также у одного пользователя может быть несколько постов, поэтому, при генерации, на каждого пользователя из 70% может быть от 1 до 4 постов.

Листинг 8.3 – Код запроса для генерации комментариев к постам

MATCH (p:Post)

WITH p, toInteger(rand() \* 4) AS commentCount

UNWIND range(1, commentCount) AS i

WITH p, i

MATCH (u:User)

WITH p, u, i, rand() AS r

WHERE r < 0.05

CREATE (c:Comment {

id: p.id \* 100 + i,

text: "Comment on post " + p.id + " by user " + u.id,

created\_at: datetime() - duration({minutes: i})

}),

(u)-[:CREATED]->(c),

(c)-[:COMMENTED\_ON]->(p);

Здесь для каждого поста совершается по 3 попытки на каждого пользователя оставить комментарий. Шанс, что пользователь оставит комментарий 5%. Таким образом, от каждого пользователя на каждом посте может быть от 0 до 3 комментариев. Шанс того, что комментарии будут под каждым постом в данном запросе достаточно низок.

Листинг 8.4 – Код запроса для генерации историй и связей с пользователями

MATCH (u:User)

WITH u, rand() AS r

WHERE r < 0.3

WITH u, toInteger(rand() \* 4) AS storyCount

UNWIND range(1, storyCount) AS i

CREATE (s:Story {

id: u.id \* 10 + i,

created\_at: datetime() - duration({hours: i}),

content\_url: "story\_" + (u.id \* 10 + i) + ".mp4",

expires\_at: datetime() + duration({hours: 24})

}),

(u)-[:CREATED]->(s);

Способ генерации аналогичен генерации постов

Листинг 8.5 – Код запроса для генерации сообществ и связей с пользователями

WITH range(1, 40) AS groupIds

UNWIND groupIds AS groupId

CREATE (:Group {

id: groupId,

name: "Group\_" + groupId,

created\_at: date() - duration({days: groupId}),

description: "Описание сообщества #" + groupId,

logo\_url: "group\_" + groupId + ".png"

});

MATCH (u:User), (g:Group)

WITH u, g, rand() AS r

WHERE r < 0.1

CREATE (u)-[:MEMBER\_OF]->(g);

Листинг 8.6 – Код запроса для генерации постов сообществ

MATCH (g:Group)

WITH g, g.id AS groupId, rand() AS r

WHERE r < 0.8

WITH g, groupId, toInteger(rand() \* 7) + 1 AS postCount

UNWIND range(1, postCount) AS i

CREATE (p:Post {

id: groupId \* 1000 + i,

content: "Group Post #" + groupId + "\_" + i,

created\_at: datetime() - duration({hours: i}),

media\_url: "group\_media\_" + groupId + "\_" + i + ".jpg"

}),

(g)-[:CREATED]->(p);

Способ генерации постов сообществ аналогичен генерации постов пользователей. На посты сообществ пользователи также могут оставлять комментарии.

Листинг 8.7 – Код запроса для генерации подписчиков и друзей пользователя

MATCH (u1:User), (u2:User)

WHERE u1.id < u2.id

WITH u1, u2, rand() AS r

FOREACH (\_ IN CASE WHEN r < 0.3 THEN [1] ELSE [] END |

CREATE (u1)-[:FRIEND]->(u2),

(u2)-[:FRIEND]->(u1)

)

FOREACH (\_ IN CASE WHEN r >= 0.3 AND r < 0.8 THEN [1] ELSE [] END |

CREATE (u1)-[:FOLLOWS]->(u2)

)

В данном запросе 50% пользователей подписаны на кого-то, а 30% дружат.

Листинг 8.8 – Код запроса для генерации лайков на истории, посты или комментарии

MATCH (u:User), (p:Post)

WHERE NOT (u)-[:CREATED]->(p)

WITH u, p, rand() AS r

WHERE r < 0.05

CREATE (u)-[:LIKED]->(p);

MATCH (u:User), (c:Comment)

WHERE NOT (u)-[:CREATED]->(c)

WITH u, c, rand() AS r

WHERE r < 0.03

CREATE (u)-[:LIKED]->(c);

MATCH (u:User), (s:Story)

WHERE NOT (u)-[:CREATED]->(s)

WITH u, s, rand() AS r

WHERE r < 0.04

CREATE (u)-[:LIKED]->(s);