Московское бюджетное профессиональное образовательное учреждение
города Москвы
«Московский государственный колледж электромеханики и
информационных технологий»
(ГБПОУ МГКЭИТ)

ОТЧЕТ

по практической работе №1

«Работа с официальной документацией СУБД»

Выполнила:
Студентка группы 3ИП-11-19
Мельничук Виктория Вячеславовна

Проверил:
Преподаватель
Басыров Сергей Амирович

Москва 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель работы заключается в исследовании официальной документации трех СУБД.

Актуальность работы обусловлена важностью выбора СУБД при разработке базы данных.

Задачи:

1) Анализ назначения СУБД;
2) Анализ основных возможностей СУБД;
3) Анализ типов данных в СУБД;
4) Анализ языка запроса СУБД.

Предмет исследования – исследование СУБД.

Объект исследования – исследование Redis, Neo4j, MySQL.

ОСНОВНАЯ ЧАСТЬ

1 Neo4J

1.1 Анализ назначения СУБД

Neo4j – лидирующая графическая база данных. Архитектура разработана для оптимального управления, хранения и обхода узлов и отношений. Графическая база данных применяется для моделирования данных в форме графиков. Узлы графа изображают сущности, отношения отображают ассоциацию этих узлов.

1.2 Анализ основных возможностей СУБД

Neo4j отличается следующими особенностями:

1) DBMS – Система управления графическими базами данных – способна хранить, обрабатывать графы, находящиеся в базе данных.

2) Модель данных/граф (гибкая схема) - Neo4j следует модели данных, называемой моделью графа собственных свойств. Здесь граф содержит узлы (сущности), и эти узлы связаны друг с другом (изображены отношениями). Узлы и отношения хранят данные в парах ключ-значение, известных как свойства;

3) Свойства ACID – Neo4j поддерживает полные правила ACID (атомарность, согласованность, изоляция и долговечность);

4) Cypher Query Language – мощный декларативный язык запросов, использующий ASCII-арт для изображения графиков. Может быть использован для создания и извлечения отношений без Join;

5) Встроенное веб-приложение – Neo4j Browser;

6) Драйверы – REST API (для работы с Java, Spring, Scala), Java Script – для работы с MVC (Node JS);

7) Поддержка Java API: Cypher API, Native Java API;

8) Индексирование – поддержка индексов с помощью Apache Lucence.

1.3 Анализ языка запроса в СУБД

Neo4j CQL – декларативный язык запросов для Neo4j Graph Database.

Структура языка взята из SQL. Язык состоит из предложений, например, предложения чтения графов: MATCH – шаблон для сопоставления графика; WHERE – создает ограничения в шаблоне Match; RETURN – что требуется вернуть.

Основные особенности:

1) CQL предоставляет возможность либо читать и сопоставлять графы, либо их обновлять, но не одновременно;

2) Транзакции. Запрос на чтение всегда выполнится; запрос на обновление может либо выполниться, либо отмениться;

3) Путь сопоставления Cypher. Neo4j использует изоморфизм отношений для сопоставления путей и является очень эффективным способом уменьшения размера набора результатов и предотвращение бесконечного обхода;

4) Гомоморфизм – нет ограничений для путей сопоставления;

5) Узловой изоморфизм – один и тот же узел не может быть возвращен более одного раза в каждой записи пути сопоставления;

6) Изоморфизм отношений. Одно и то же отношение не может быть возвращено более одного раза в каждой записи пути сопоставления.

1.4 Анализ типов данных в СУБД

Cypher поддерживает следующие типы данных:

Типы свойств: Integer, Float, String, Boolean, Point, Date, Time, LocalTime, DateTime, LocalDateTime, Duration. Могут быть возвращены с запросами Cypher, могут быть использованы как параметры, храниться как свойства, быть сконструированы с Cypher literals. Выражаются как: Number (Integer, Float), String, Boolean, Point, Date, Time, LocalTime, DateTime, LocalDateTime, Duration

Структурные типы: Node, Relationship, Path. Могут быть возвращены с запросами Cypher.

Составные типы: List, Map. Могут быть возвращены с запросами Cypher, могут быть использованы, быть сконструированы с Cypher literals.

1.5 Анализ синтаксиса CQL в СУБД:

Пример правильного синтаксиса отображен на рисунке 1 и рисунке 2.

```
// Создаем узел n, помечаем как Person, с двумя атрибутами name и рожд
CREATE (n:Person { name:'Tom Hanks', born:1956 }) RETURN n

// Создание отношения между a и b как НАПРАВЛЕННОЕ и возвращение отнош
MATCH (a:Person),(b:Movie)
WHERE a.name = 'Robert Zemeckis' AND b.title = 'Forrest Gump'
CREATE (a)-[r:DIRECTED]->(b)
RETURN r, id(r), type(r)

// Создание отношения между a и b как ACTED_IN, отношение имеет роли с
MATCH (a:Person), (b:Person)
WHERE a.name='Tom Hanks' AND b.title='Forrest Gump'
CREATE (a)-[r:ACTED_IN { roles:['Forrest'] }]->(b)
RETURN r, id(r), type(r)
```

Рисунок 1 – Синтаксис CREATE

```
// Запрашиваем всю базу данных графа
match(n) return n

// Запрашиваем узлы, чей родовой атрибут меньше 1952
MATCH (n) WHERE n.born < 1952 RETURN n

// Запрос узла с указанной меткой
MATCH (n:Movie) RETURN n, labels(n)

// Запрос всех узлов, связанных с тегами Movie
match(n)--(m:Movie)
return n

// Метка запроса - Person, а имя - это все узлы, на которые указал Том
MATCH (:Person { name: 'Tom Hanks' })-->(movie) RETURN movie;
```

Рисунок 2 – Синтаксис MATCH

2 MySQL

2.1 Анализ назначения СУБД

MySQL это система управления реляционными базами данных с открытым исходным кодом (СУРБД) с моделью клиент-сервер. СУРБД — это программное обеспечение или служба, используемая для создания и управления базами данных на основе реляционной модели.

2.2 Анализ основных возможностей СУБД

Следующий перечень описывает основные возможности MySQL:

1) Полностью многопоточное использование ядерных нитей. Это означает, что пакет может легко использовать много CPUs, если они есть;

2) Интерфейсы для языков C, C++, Eiffel, Java, Perl, PHP, Python и Tcl;

3) Работает на многих различных платформах;

4) Много типов столбцов: целые со знаком или без него длиной 1, 2, 3, 4 и 8 байт, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM;

5) Очень быстрые объединения, использующие оптимизированное однопроходное объединение многих таблиц.

6) SQL-функции выполнены через хорошо оптимизированную библиотеку классов и должны выполняться с такой скоростью, с какой только возможно! Обычно не имеется никакого распределения памяти вообще после инициализации запроса.;

7) Полная поддержка предложений SQL GROUP BY, ORDER BY;

8) Поддержка LEFT OUTER JOIN, RIGHT OUTER JOIN;

9) Привилегии и система паролей, которая является очень гибкой и безопасной, и позволяет проверку, основанную на имени хоста. Пароли безопасны потому, что вся передача пароля шифрована, когда Вы соединяетесь с сервером.

2.3 Анализ типов данных в СУБД

Основные типы данных:

— CHAR: представляет строку фиксированной длины;

— VARCHAR: представляет строку переменной длины;

— TINYTEXT: представляет текст длиной до 255 байт;

— TEXT: представляет текст длиной до 65 КБ;

— MEDIUMTEXT: представляет текст длиной до 16 МБ;

— LARGETEXT: представляет текст длиной до 4 ГБ;

— TINYINT: представляет целые числа от -128 до 127, занимает 1 байт;

— BOOL: фактически не представляет отдельный тип, а является лишь псевдонимом для типа TINYINT и может хранить два значения 0 и 1;

— INT: представляет целые числа от -2147483648 до 2147483647, занимает 4 байта;

— INT UNSIGNED: представляет целые числа от 0 до 4294967295, занимает 4 байта;

— BIGINT: представляет целые числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, занимает 8 байт;

— BIGINT UNSIGNED: представляет целые числа от 0 до 18 446 744 073 709 551 615, занимает 8 байт;

— DECIMAL: хранит числа с фиксированной точностью. Данный тип может принимать два параметра precision и scale: DECIMAL(precision, scale);

— FLOAT: хранит дробные числа с плавающей точкой одинарной точности от -3.4028 * 1038 до 3.4028 * 1038, занимает 4 байта;

— DOUBLE: хранит дробные числа с плавающей точкой двойной точности от -1.7976 * 10308 до 1.7976 * 10308, занимает 8 байт;

— DATE: хранит даты с 1 января 1000 года до 31 деабря 9999 года (с "1000-01-01" до "9999-12-31"). По умолчанию для хранения используется формат yyyy-mm-dd. Занимает 3 байта;

— TIME: хранит время от -838:59:59 до 838:59:59. По умолчанию для хранения времени применяется формат "hh:mm:ss". Занимает 3 байта;

— DATETIME: объединяет время и дату, диапазон дат и времени - с 1 января 1000 года по 31 декабря 9999 года (с "1000-01-01 00:00:00" до "9999-12-31 23:59:59"). Для хранения по умолчанию используется формат "yyyy-mm-dd hh:mm:ss". Занимает 8 байт;

— TIMESTAMP: также хранит дату и время, но в другом диапазоне: от "1970-01-01 00:00:01" UTC до "2038-01-19 03:14:07" UTC. Занимает 4 байта;

— YEAR: хранит год в виде 4 цифр. Диапазон доступных значений от 1901 до 2155. Занимает 1 байт;

— ENUM: хранит одно значение из списка допустимых значений. Занимает 1-2 байта;

— SET: может хранить несколько значений (до 64 значений) из некоторого списка допустимых значений. Занимает 1-8 байт;

— TINYBLOB: хранит бинарные данные в виде строки длиной до 255 байт;

— BLOB: хранит бинарные данные в виде строки длиной до 65 КБ;

— MEDIUMBLOB: хранит бинарные данные в виде строки длиной до 16 МБ;

— LARGEBLOB: хранит бинарные данные в виде строки длиной до 4 ГБ.

2.4 Анализ языка запроса в СУБД

В MySQL можно создавать следующие запросы:

1. Самые простые MySQL запросы;
2. Простые SELECT (выбрать) запросы;
3. Простые INSERT (новая запись) запросы;
4. Простые UPDATE (перезаписать, дописать) запросы;
5. Простые DELETE (удалить запись) запросы;
6. Простые DROP (удалить таблицу) запросы;
7. Сложные MySQL запросы;
8. MySQL запросы и переменные PHP.

Используются операторы WHERE, ORDER BY, GROUP BY при создании запросов с целью создания ограничений.

3 Redis

3.1 Анализ назначения СУБД

Redis (расшифровывается как Remote Dictionary Server) – это быстрое хранилище данных типа «ключ-значение» в памяти с открытым исходным кодом. Проект возник, когда Сальваторе Санфилиппо, первоначальный разработчик Redis, захотел улучшить масштабируемость стартапа в Италии. Он создал хранилище Redis, которое теперь используется в качестве базы данных, кэша, брокера сообщений и очереди.

Redis обеспечивает время отклика на уровне долей миллисекунды и позволяет приложениям, работающим в режиме реального времени, выполнять миллионы запросов в секунду. Такие приложения востребованы в сферах игр, рекламных технологий, финансовых сервисов, здравоохранения и IoT.

3.2 Анализ основных возможностей СУБД

Особенности Redis:

1) В первую очередь, Redis умеет сохранять данные на диск. Можно настроить Redis так, чтобы данные вообще не сохранялись, сохранялись периодически по принципу copy-on-write, или сохранялись периодически b писались в журнал (binlog). Таким образом, всегда можно добиться требуемого баланса между производительностью и надежностью;

2) Redis, в отличие от Memcached, позволяет хранить не только строки, но и массивы (которые могут использоваться в качестве очередей или стеков), словари, множества без повторов, большие массивы бит (bitmaps), а также множества, отсортированные по некой величине. Разумеется, можно работать с отдельными элементами списков, словарей и множеств. Как и Memcached, Redis позволяет указать время жизни данных (двумя способами — «удалить тогда-то» и «удалить через …»). По умолчанию все данные хранятся вечно;

3) Интересная особенность Redis заключается в том, что это — однопоточный сервер. Такое решение сильно упрощает поддержку кода, обеспечивает атомарность операций и позволяет запустить по одному процессу Redis на каждое ядро процессора. Разумеется, каждый процесс будет прослушивать свой порт. Решение нетипичное, но вполне оправданное, так как на выполнение одной операции Redis тратит очень небольшое количество времени;

4) В Redis есть репликация. Репликация с несколькими главными серверами не поддерживается. Каждый подчиненный сервер может выступать в роли главного для других. Репликация в Redis не приводит к блокировкам ни на главном сервере, ни на подчиненных. На репликах разрешена операция записи. Когда главный и подчиненный сервер восстанавливают соединение после разрыва, происходит полная синхронизация (resync);

5) Также Redis поддерживает транзакции (будут последовательно выполнены либо все операции, либо ни одной) и пакетную обработку команд (выполняем пачку команд, затем получаем пачку результатов). Притом ничто не мешает использовать их совместно;

6) Еще одна особенность Redis — поддержка механизма publish/subscribe. С его помощью приложения могут создавать каналы, подписываться на них и помещать в каналы сообщения, которые будут получены всеми подписчиками.

3.3 Анализ типов данных в СУБД

Типы данных Redis:

1) Строки (strings). Базовый тип данных Redis. Строки в Redis бинарно-безопасны, могут использоваться так же как числа, ограничены размером 512 Мб;

2) Списки (lists). Классические списки строк, упорядоченные в порядке вставки, которая возможна как со стороны головы, так и со стороны хвоста списка. Максимальное количество элементов — $2^{32} - 1$;

3) Множества (sets). Множества строк в математическом понимании: не упорядочены, поддерживают операции вставки, проверки вхождения элемента, пересечения и разницы множеств. Максимальное количество элементов — $2^{32} – 1$;

4) Хеш-таблицы(hashes). Классические хеш-таблицы или ассоциативные массивы. Максимальное количество пар «ключ-значение» — $2^{32}- 1$;

5) Упорядоченные множества (sorted sets). Упорядоченное множество отличается от обычного тем, что его элементы упорядочены по особому параметру «score».

3.4 Анализ языка запроса в СУБД

Строки Redis (команды):

1) SET “student” “john” – добавление строки;
2) GET “student” – получение строки;
3) DEL “student” – удаление строки.

Списки содержат строки, отсортированные по порядку их вставки. Можно добавлять элементы в начало или конец списка, что полезно для заданий очереди. Более срочные задания можно поставить перед менее приоритетными. Для вставки элемента в начале строки (слева) мы используем команду LPUSH, а для вставки в конце строки (справа) — команду RPUSH.

Множества Redis — полезный тип данных, поддерживающий мощные операции, такие как пересечения и объединения. Они никак не упорядочены и обычно используются, когда вы хотите выполнить аудит и увидеть взаимосвязи между различными переменными.

Операции с множествами выполняются достаточно быстро. Независимо от количества сохраненных элементов, добавление или удаление элементов в множество занимает одно и то же время.

Кроме того, множества не допускают повторения ключей или объектов, поэтому многократное добавление ключа в множество просто проигнорируется. Это определяется функцией SADD, которая позволяет избежать дублирования нескольких похожих записей. Атрибут SADD можно

использовать для проверки уникальных значений, а также планирования заданий в фоновом режиме, включая задания cron, которые представляют собой автоматизированные скрипты.

Множества особенно полезны для анализа поведения клиентов в режиме реального времени. Представим магазин одежды. Здесь упорядоченные множества Redis с помощью различных сопоставлений дадут точную картину поведения клиентов. Вы можете определить паттерны покупок в зависимости от пола и других характеристик покупателя, узнать, какие модели одежды продаются лучше, в какие часы зафиксированы максимальные продажи каждого товара.

Примеры использования языка запроса Redis со множествами отображен на рисунке 3.

```
127.0.0.1:6379> SADD objects obj1
(integer) 1
127.0.0.1:6379> SADD objects obj2
(integer) 1
127.0.0.1:6379> SADD objects obj3
(integer) 1
127.0.0.1:6379> SADD objects obj1
(integer) 0
127.0.0.1:6379> SMEMBERS objects
1) "obj2"
2) "obj3"
3) "obj1"
```

Рисунок 3 – Множества Redis

Пример использования транзакций в Redis отображен на рисунке 4.

```
127.0.0.1:6379> MULTI
OK
127.0.0.1:6379> SET obj1 value1
QUEUED
127.0.0.1:6379> SET obj2 value2
QUEUED
127.0.0.1:6379> EXEC
1) OK
2) OK
127.0.0.1:6379> GET obj1
"value1"
```

Рисунок 4 – Транзакции Redis

ЗАКЛЮЧЕНИЕ

В процессе выполнения практической работы были приобретены знания по основным характеристикам таких СУБД, как Redis, Neo4j, MySQL; были проанализированы их назначение, отличительные особенности, типы данных, синтаксис языка запроса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. MySQL 8.0 Reference Manual [Электронный ресурс]: MySQL. The world’s most popular open source database. - Режим доступа : https://dev.mysql.com/ (дата обращения: 17.11.2021);

2. Справочное руководство по MySQL [Электронный ресурс]: MySQL. Одобрено лучшими российскими программистами. - Режим доступа : http://www.mysql.ru/docs/man/Reference.html (дата обращения: 17.11.21);

3. Шпаргалка по Redis [Электронный ресурс]: Хабр. - Режим доступа : https://habr.com/ru/post/204354/ (дата обращения: 17.11.2021);

4. Getting Started Guide v4.3 [Электронный ресурс]: Neo4j. - Режим доступа : https://neo4j.com/docs/ (дата обращения: 17.11.21).