

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

Отчет по практической работе № 1
Работа с официальной документации различных СУБД

Выполнил:
Студент группы ЗИП-11-19
Войлоков А.А.
Проверил:
Преподаватель Басыров С. А.

ВВЕДЕНИЕ

Цель – получить актуальные знания по различным СУБД.

Актуальность – важность 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 в СУБД.

Предмет исследования – исследование СУБД.

Объект исследования – Neo4j ,Cassandra, MySQL.

ОСНОВНАЯ ЧАСТЬ

1. Neo4j

1.1 Анализ

Neo4j – графовая система управления базами данных с открытым исходным кодом, реализованная на Java. Данные хранит в собственном формате, специализированно приспособленном для представления графовой информации, такой подход в сравнении с моделированием графовой базы данных средствами реляционной СУБД позволяет применять дополнительную оптимизацию в случае данных с более сложной структурой. Также утверждается о наличии специальных оптимизаций для SSD-накопителей, при этом для обработки графа не требуется его помещение целиком в оперативную память вычислительного узла, таким образом, возможна обработка достаточно больших графов.

1.2 Основные возможности

Данная СУБД имеет следующие возможности:

- Гибкая модель данных – Neo4j предоставляет гибкую простую и вместе с тем мощную модель данных, которую можно легко изменять в зависимости от приложений и отраслей;
- Анализ в реальном времени – Neo4j предоставляет результаты на основе данных в реальном времени;
- Высокая доступность – Neo4j отлично доступен для крупных корпоративных приложений реального времени с транзакционными гарантиями;
- Простой поиск – Используя Neo4j, вы можете не только представлять, но и легко извлекать (перемещаться / перемещаться) связанные данные быстрее по сравнению с другими базами данных;
- Нет объединений – используя Neo4j, он НЕ требует сложных объединений для извлечения связанных / связанных данных, так

как очень легко получить сведения о соседнем узле или взаимосвязи без объединений или индексов.

1.3 Типы данных

Типы недвижимости:

- Number, абстрактный тип, который имеет подтипы *Integer* и *Float*;
- Нить;
- Логический;
- Пространственный тип *Point*;
- Временные типы: *Date*, *Time*, *LocalTime*, *DateTime*, *LocalDateTime* и *Duration*.
- Структурные типы:
- Узлы (идентификатор, этикетка (и), карта (собственности));
- Отношения (идентификатор, тип, карта (собственности), идентификатор начального и конечного узлов);
- Пути (чередующаяся последовательность узлов и отношений).
- Составные типы:
- Списки – представляют собой разнородные упорядоченные наборы значений, каждое из которых имеет какое-либо свойство, структурный или составной тип.
- Карты – представляют собой разнородные неупорядоченные коллекции пар (ключ, значение).

1.4 Язык запроса

В СУБД используется собственный язык запросов – Cypher, но запросы можно делать и другими способами, например, напрямую через Java API и на языке Gremlin, созданном в проекте с открытым исходным кодом TinkerPop. Cypher является не только языком запросов, но и языком манипулирования данными, так как предоставляет функции CRUD для графового хранилища.

2 Cassandra

2.1 Анализ

Apache Cassandra – это не реляционная отказоустойчивая распределенная СУБД, рассчитанная на создание высоко масштабируемых и надёжных хранилищ огромных массивов данных, представленных в виде хэша. Проект был разработан на языке Java в корпорации Facebook в 2008 году, и передан фонду Apache Software Foundation в 2009. Эта СУБД относится к гибридным NoSQL-решениям, поскольку она сочетает модель хранения данных на базе семейства столбцов (ColumnFamily) с концепцией key-value (ключ-значение).

2.2 Основные возможности

Данная СУБД имеет следующие возможности:

- 1) Эластичная масштабируемость – Cassandra отлично масштабируется; это позволяет добавить больше оборудования, чтобы разместить больше клиентов и больше данных согласно требованию;
- 2) Всегда на архитектуре – Cassandra не имеет единой точки отказа, и она постоянно доступна для критически важных для бизнеса приложений, которые не могут допустить сбоя;
- 3) Высокая производительность в линейном масштабе – Cassandra линейно масштабируется, т. Е. Увеличивает пропускную способность при увеличении количества узлов в кластере. Поэтому он поддерживает быстрое время отклика;
- 4) Простое распространение данных – Cassandra обеспечивает гибкость в распределении данных там, где вам нужно, путем репликации данных между несколькими центрами обработки данных;
- 5) Поддержка транзакций – Cassandra поддерживает такие свойства, как атомарность, согласованность, изоляция и долговечность (ACID).

2.3 Типы данных

- 1) ByteType: любые байтовые строки (без валидации);
- 2) ByteType: любые байтовые строки (без валидации);
- 3) UTF8Type: UTF-8 строка;
- 4) IntegerType: число с произвольным размером;

- 5) Int32Type: 4-байтовое число;
- 6) LongType: 8-байтовое число;
- 7) TimeUUIDType: UUID 1-ого типа;
- 8) BooleanType: два значения: true = 1 или false = 0;
- 9) DoubleType: 8-байтовое число с плавающей запятой;
- 10) FloatType: 4-байтовое число с плавающей запятой;
- 11) DecimalType: число с произвольным размером и плавающей запятой.

2.4 Языки запроса

Пользователи могут получить доступ к Cassandra через его узлы, используя Cassandra Query Language (CQL). CQL рассматривает базу данных (**Keyspace**) как контейнер таблиц. Программисты используют **cqlsh**: приглашение работать с CQL или отдельными драйверами языка приложения.

Клиенты обращаются к любому из узлов за своими операциями чтения-записи. Этот узел (координатор) воспроизводит прокси между клиентом и узлами, содержащими данные.

3 MySQL

3.1 Анализ

MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

3.2 Основные возможности

MySQL имеет следующие основные возможности:

- 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) Очень быстрые объединения, использующие оптимизированное однопроходное объединение многих таблиц;
- 5) Полная поддержка операторов и функций в частях запроса SELECT и WHERE;
- 6) Поддержка LEFT OUTER JOIN и RIGHT OUTER JOIN с синтаксисами ANSI SQL и ODBC;
- 7) Привилегии и система паролей, которая является очень гибкой и безопасной, и позволяет проверку, основанную на имени хоста.

3.3 Типы данных

- 1) CHAR: представляет строку фиксированной длины;
- 2) VARCHAR: представляет строку переменной длины;
- 3) TINYTEXT: представляет текст длиной до 255 байт;
- 4) INT: представляет целые числа от -2147483648 до 2147483647, занимает 4 байта;
- 5) DECIMAL: хранит числа с фиксированной точностью. Данный тип может принимать два параметра precision и scale: DECIMAL (precision, scale);
- 6) FLOAT: хранит дробные числа с плавающей точкой одинарной точности;
- 7) DOUBLE: хранит дробные числа с плавающей точкой двойной точности.

3.4 Язык запроса

Общая структура запросов в MySQL выглядит следующим образом.

SELECT ('столбцы или * для выбора всех столбцов; обязательно');

FROM ('таблица; обязательно');

WHERE ('условие/фильтрация, например, city = 'Moscow'; необязательно');

GROUP BY ('столбец, по которому хотим сгруппировать данные;
необязательно');

HAVING ('условие/фильтрация на уровне сгруппированных данных;
необязательно');

ORDER BY ('столбец, по которому хотим отсортировать вывод;
необязательно');

ЗАКЛЮЧЕНИЕ

В процессе выполнения практической работы были получены знания по различным СУБД. Был проведен анализ их характеристик, возможностей, типов данных, а также языков запроса.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) <https://neo4j.com/docs/>
- 2) <https://coderlessons.com/tutorials/bazy-dannykh/uznaite-neo4j/neo4j-kratkoe-rukovodstvo>
- 3) <https://coderlessons.com/tutorials/bazy-dannykh/uznaite-neo4j/neo4j-kratkoe-rukovodstvo>
- 4) <https://www.bigdataschool.ru/wiki/cassandra>
- 5) <https://docs.microsoft.com/ru-ru/azure/cosmos-db/cassandra/cassandra-support>
- 6) <https://dev.mysql.com/doc/refman/8.0/en/tutorial.html>
- 7) <https://metanit.com/sql/mysql/>