**Государственное бюджетное профессиональное образовательное учреждение города Москвы  
Московский государственный колледж электромеханики и информационных технологий**

Практическая работа №1

по теме

разработка, администрирование и защита баз данных

специальность 09.02.07 Информационные системы и программирование

студента группы 2ИП-11-19

Дмитриева Александра Олеговича

преподаватель: Басыров Сергей Амирович

**ВВЕДЕНИЕ**

**Цель работы** – сравнение трёх систем управления базами данных.

**Актуальность работы** обусловлена неочевидностью выбора системы управления базами данных по тому или иному критерию.

**Задачи:**

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

**Объект исследования** – системы управления базами данных, их различия и особенности.

**Предмет исследования** – анализ систем управления базами данных MongDB, Cassandra, MySQL.

**ОСНОВНАЯ ЧАСТЬ**

**1 MongoDB**

**1.1 Назначение**

Данная СУБД применяется в web-разработке, является документоориентированной и подходит для следующих применений:

* хранение и регистрация событий;
* системы управления документами и контентом;
* электронная коммерция;
* игры;
* данные мониторинга, датчиков;
* мобильные приложения;
* хранилище операционных данных веб-страниц (например, хранение комментариев, рейтингов, профилей пользователей, сеансы пользователей).

**1.2 Возможности**

Система поддерживает ad-hoc-запросы: они могут возвращать конкретные поля документов и пользовательские JavaScript-функции. Поддерживается поиск по регулярным выражениям. Также можно настроить запрос на возвращение случайного набора результатов.

Имеется поддержка индексов.

Система может работать с набором реплик, то есть содержать две или более копии данных на различных узлах. Каждый экземпляр набора реплик может в любой момент выступать в роли основной или вспомогательной реплики. Все операции записи и чтения по умолчанию осуществляются с основной репликой. Вспомогательные реплики поддерживают в актуальном состоянии копии данных. В случае, когда основная реплика дает сбой, набор реплик проводит выбор, которая из реплик должна стать основной. Второстепенные реплики могут дополнительно являться источником для операций чтения.

Система масштабируется горизонтально, используя технику сегментирования объектов баз данных — распределение их частей по различным узлам кластера. Администратор выбирает ключ сегментирования, который определяет, по какому критерию данные будут разнесены по узлам (в зависимости от значений хэша ключа сегментирования). Благодаря тому, что каждый узел кластера может принимать запросы, обеспечивается балансировка нагрузки.

Система может быть использована в качестве файлового хранилища с балансировкой нагрузки и репликацией данных.

Может работать в соответствии с парадигмой MapReduce. Для агрегации данных предусмотрен аналог SQL-выражения GROUP BY; операторы агрегации могут быть связаны в цепи подобно Unix-конвейерам. Фреймворк также имеет оператор $lookup для связки документов при выгрузке и статистические операции такие как среднеквадратическое отклонение.

Поддерживается JavaScript в запросах, функциях агрегации (например, в MapReduce).

Поддерживаются коллекции с фиксированным размером. Такие коллекции сохраняют порядок вставки и по достижении заданного размера ведут себя как кольцевой буфер.

**1.3 Типы данных**

В MongoDB используются следующие типы данных:

Integer – используется для хранения целочисленных значений. В зависимости от сервера может быть как 32-битным, так и 64-битным;

Double – используется для хранения значений с плавающей точко;

Boolean – используется для хранения логических (true / false) значений;

String – используется для хранения символьных строк UTF-8;

Arrays – данный тип данных используется для хранения массивов значений по одному ключу;

Object – используется для встроенных документов;

Symbol – используется так же, как и String, но, обычно, резервируется для языков, которые используют специальные символы;

Null – используется для хранения значения Null;

Timestamp – используется для хранения даты и времени;

Min / Max – используется для сравнения значений с наибольшим и наименьшим BSON (Binary JSON) элементом;

Object ID – используется для хранения ID документа;

Regular Expression – используется для хранения регулярных выражений;

Code – используется для хранения JavaScript кода в документе;

Binary data – данный тип данных позволяет хранить бинарные данные;

Date – используется для хранения текущей даты или времени в UNIX формате.

**1.4 Язык запросов**

В MongoDB в качестве язык запросов используется JavaScript и JSON-структуры. Выбор столь нехарактерного языка запроса объясняется тем, что эта документ-ориентированная СУБД использует JSON-формат для представления документов и вывода результатов.

**2 Cassandra**

**2.1 Назначение**

Apache Cassandra – это распределённая система управления базами данных, относящаяся к классу NoSQL-систем и рассчитанная на создание высокомасштабируемых и надёжных хранилищ огромных массивов данных, представленных в виде хэша.

**2.2 Возможности**

Apache Cassandra - это распределенное, отказоустойчивое, масштабируемое, колонко-ориетированное хранилище данных. Хранилище само позаботится о проблемах наличия единой точки отказа (single point of failure), отказа серверов и о распределении данных между узлами кластера (cluster node). При чем, как в случае размещения серверов в одном центре обработки данных (data center), так и в конфигурации со многими центрами обработки данных, разделенных расстояниями и, соответственно, сетевыми задержками.

**2.3 Типы данных**

* ascii – хранит строки (ascii);
* bigint – хранит большие целые числа (64-битные);
* blob – хранит байты;
* boolean – хранит логические значения;
* counter– хранит целые числа (счётчик);
* decimal – хранит целые числа, точные числа с плавающей точкой;
* double – хранит целые числа, числа с плавающей точкой (64-битные числа IEEE-754);
* float – хранит целые числа, числа с плавающей точкой (32-битные числа IEEE-754);
* inet – хранит строки (IP адреса, IPv4 или IPv6);
* int – хранит 32-битные знаковые целые числа;
* text – хранит строки в кодировке UTF-8;
* timestamp – хранит целые числа, строки (время);
* timeuuid – хранит уникальные идентификаторы UUID тип 1;
* uuid – хранит уникальные идентификаторы UUID тип 1 или 4;
* varchar – хранит строки в кодировке UTF-8;
* varint – хранит точные целые числа.

**2.4 Язык запроса**

В качестве языка запросов используется CQL (Cassandra Query Language), который в какой-то степени сходен с SQL, но существенно урезан по функциональным возможностям. Например, можно выполнять только простейшие запросы SELECT с выборкой по определённому условию. Добавление и обновление осуществляется через единое выражение UPDATE, операция INSERT отсутствует (если записи нет, при выполнении UPDATE она создаётся — используется семантика SQL-оператора MERGE). Из отличительных возможностей — поддержка пространств имён и семейств столбцов, создание индексов через выражение «CREATE INDEX».

**3 MySQL**

**3.1 Назначение**

MySQL является свободной реляционной системой управления базами данных.

Обычно используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

**3.2 Возможности**

Главная особенность MySQL в том, что он поддерживает язык запросов SQL в стандарте ANSI 92, и кроме этого имеет множество расширений к этому стандарту, которых нет ни в одной другой системе управления базами данных.

Перечень возможностей MySQL:

* поддерживается эффективная работа неограниченного количества пользователей, одновременно работающих с базой данных;
* количество строк в таблицах может достигать 50 млн;
* максимально быстрое выполнение команд;
* простая и эффективная система безопасности.

Существует мнение, что MySQL – самый быстрый сервер из существующих.

**3.3 Типы данных**

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп:

**3.3.2 Символьные типы**

* CHAR: представляет строку фиксированной длины. Длина хранимой строки указываются в скобках, например, CHAR(10) - строка из десяти символов. И если в таблицу в данный столбец сохраняется строка из 6 символов (то есть меньше установленной длины в 10 символов), то строка дополняется 4 пробелами и в итоге все равно будет занимать 10 символов. Тип CHAR может хранить до 255 байт;
* VARCHAR: представляет строку переменной длины. Длина хранимой строки также указывается в скобках, например, VARCHAR(10). Однако в отличие от CHAR хранимая строка будет занимать именно столько места, сколько необходимо. Например, если определенная длина в 10 символов, но в столбец сохраняется строка в 6 символов, то хранимая строка так и будет занимать 6 символов плюс дополнительный байт, который хранит длину строки. Тип VARCHAR может хранить до 65535 байт.

Начиная с MySQL 5.6 типы CHAR и VARCHAR по умолчанию используют кодировку UTF-8, которая позволяет использовать до 3 байт для хранения символа в зависимости от языка.

Ряд дополнительных типов данных представляют текст неопределенной 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.

**3.3.2 Числовые типы**

* TINYINT – представляет целые числа от -128 до 127, занимает 1 байт;
* BOOL – фактически не представляет отдельный тип, а является лишь псевдонимом для типа TINYINT(1) и может хранить два значения: 0 и 1. Однако данный тип может также в качестве значения принимать встроенные константы TRUE (представляет число 1) и FALSE (предоставляет число 0).Также имеет псевдоним BOOLEAN;
* TINYINT UNSIGNED: представляет целые числа от 0 до 255, занимает 1 байт;
* SMALLINT: представляет целые числа от -32768 до 32767, занимает 2 байтa;
* SMALLINT UNSIGNED: представляет целые числа от 0 до 65535, занимает 2 байтa;
* MEDIUMINT: представляет целые числа от -8388608 до 8388607, занимает 3 байта;
* MEDIUMINT UNSIGNED: представляет целые числа от 0 до 16777215, занимает 3 байта;
* 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). Параметр precision представляет максимальное количество цифр, которые может хранить число. Это значение должно находиться в диапазоне от 1 до 65. Параметр scale представляет максимальное количество цифр, которые может содержать число после запятой. Это значение должно находиться в диапазоне от 0 до значения параметра precision. По умолчанию оно равно 0;
* FLOAT: хранит дробные числа с плавающей точкой одинарной точности от -3.4028 \* 1038 до 3.4028 \* 1038, занимает 4 байта. Может принимать форму FLOAT(M,D), где M - общее количество цифр, а D - количество цифр после запятой;
* DOUBLE: хранит дробные числа с плавающей точкой двойной точности от -1.7976 \* 10308 до 1.7976 \* 10308, занимает 8 байт. Также может принимать форму DOUBLE(M,D), где M - общее количество цифр, а D - количество цифр после запятой. Данный тип также имеет псевдонимы REAL и DOUBLE PRECISION, которые можно использовать вместо DOUBLE.

**3.3.3 Типы для работы с датой и временем**

* DATE: хранит даты с 1 января 1000 года до 31 деабря 9999 года (c "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. Тип Date может принимать даты в различных форматах, однако непосредственно для хранения в самой бд даты приводятся к формату "yyyy-mm-dd". Некоторые из принимаемых форматов:
  + - yyyy-mm-dd - 2018-05-25;
    - yyyy-m-dd - 2018-5-25;
    - yy-m-dd - 18-05-25.

В таких форматах двузначные числа от 00 до 69 воспринимаются как даты в диапазоне 2000-2069. А числа от 70 до 99 как диапазон чисел 1970 - 1999. Другие форматы:

* + - yyyymmdd – 20180525;
    - yyyy.mm.dd - 2018.05.25.

Для времени тип Time использует 24-часовой формат. Он может принимать время в различных форматах:

* hh:mi - 3:21 (хранимое значение 03:21:00);
* hh:mi:ss - 19:21:34;
* hhmiss – 192134.

Примеры значений для типов DATETIME и TIMESTAMP:

* 2018-05-25 19:21:34;
* 2018-05-25 (хранимое значение 2018-05-25 00:00:00).

**3.3.4 Составные типы**

* 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.

Бинарные типы

* 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.

**3.4 Язык запроса**

Данная СУБД использует язык запросов SQL. Structured Query Language (SQL) — язык структурированных запросов, с помощью него пишутся специальные запросы (SQL инструкции) к базе данных с целью получения этих данных из базы и для управления этими данными.

**ЗАКЛЮЧЕНИЕ**

В результате анализа систем управления базами данных MongoDB, Cassandra и MySQL стало возможно выявить из них наиболее подходящую для каждой конкретной цели или ситуации.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. <https://ru.wikipedia.org/wiki/MongoDB> (17.11.2021)
2. <https://proselyte.net/tutorials/mongodb/data_types/> (17.11.2021)
3. <https://salatpower.ru/?p=5> (17.11.2021)
4. <https://ru.wikipedia.org/wiki/Apache_Cassandra> (17.11.2021)
5. <https://proselyte.net/tutorials/cassandra/cql-data-types/> (17.11.2021)
6. <https://ru.wikipedia.org/wiki/MySQL> (17.11.2021)
7. <https://metanit.com/sql/mysql/2.3.php> (17.11.2021)