

**Цель работы**

Получить и отработать навыки работы с MongoDB.

**Теоретическое введение**

MongoDB — документоориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Написана на языке C++. Применяется в веб-разработке, в частности, в рамках JavaScript-ориентированного стека MEAN.

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 (англ. sharding) 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 (функция Grid File System; поставляется вместе с драйверами MongoDB). Предоставляются программные средства для работы с файлами и их содержимым. GridFS используется в плагинах для Nginx и lighttpd. GridFS разделяет файл на части и хранит каждую часть как отдельный документ.

Может работать в соответствии с парадигмой MapReduce. Во фреймворке для агрегации есть аналог SQL-выражения GROUP BY. Операторы агрегации могут быть связаны в конвейер подобно UNIX-конвейрам. Фреймворк так же имеет оператор $lookup для связки документов при выгрузке и статистические операции такие как среднеквадратическое отклонение.

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

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

В июне 2018 года (в версии 4.0) добавлена поддержка транзакций, удовлетворяющих требованиям ACID.

**Установка**

Для установки MongoDB я использовал виртуальную машину с ос Ubuntu 20.04. Выполняем следующие команды в консоли:

sudo apt update

sudo apt install -y mongodb

sudo systemctl status mongodb

mongo

**Запуск**

Запуск mongoDB производится командой mongo.



Рисунок 1 – Запуск.

**Взаимодействие с MongoDB**

Для создание базы надо использовать команду use, после которой идет название базы данных. При этом не важно, существует ли такая бд или нет. Если ее нет, то MongoDB автоматически создаст ее при добавлении в нее данных.

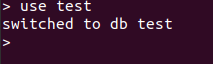


Рисунок 2 – Переключение на бд test.

Для создания коллекций в MongoDB используется следующая команда: db.createCollection и для просмотра всех коллекций используется команда: show collections. В MongoDB при вставке документа коллекция создаётся автоматически.

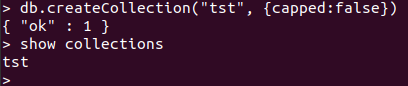


Рисунок 3 – Создание коллекции.

Для добавления в коллекцию могут использоваться методы:

insertOne: добавляет один документ

insertMany: добавляет несколько документов

insert: может добавлять, как один, так и несколько документов

Для поиска по коллекции, используется метод find.

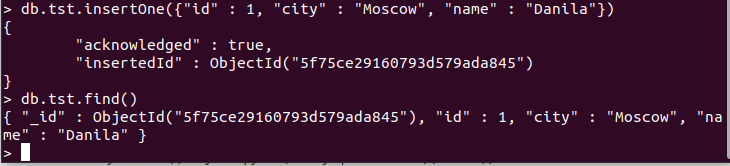


Рисунок 4 – Добавление в базу и поиск методом find.

Условные операторы задают условие, которому должно соответствовать

значение поля документа:

* $eq (равно)
* $ne (не равно)
* $gt (больше чем)
* $lt (меньше чем)
* $gte (больше или равно)
* $lte (меньше или равно)
* $in определяет массив значений, одно из которых должно иметь поле документа
* $nin определяет массив значений, которые не должно иметь поле документа

Найдем пользователя с id 1:

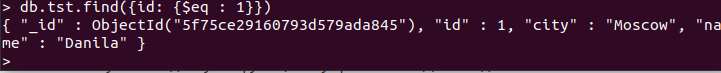


Рисунок 5 – Поиск пользователя.

**Выводы:**

В ходе данной работы я получил навыки работы с mongoDB и успешно отработал типовые задачи.

**Библиографический список**:

1. руководство по MongoDB [Электронный ресурс]. URL: https://metanit.com/nosql/mongodb/
2. УСТАНОВКА MONGODB В UBUNTU 20.04 [Электронный ресурс]. URL: https://losst.ru/kak-[ustanovit](https://losst.ru/kak-ustanovit-mongodb-v-ubuntu)-mongodb-v-ubuntu