**Лабораторная работа №1**

**Знакомство с Mongo**

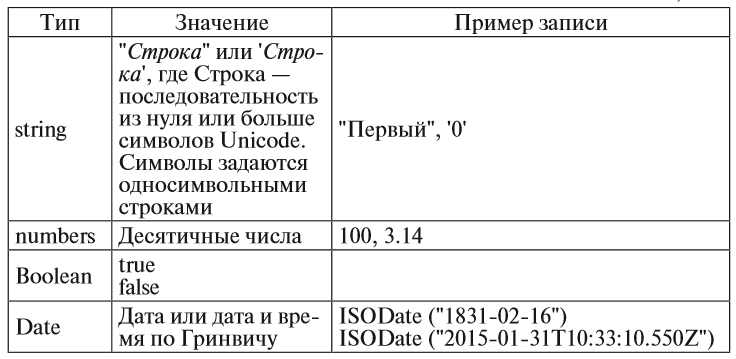
Для работы с БД в Mongo Shell необходимо запустить файл mongo.exe, находящийся в папке <Домашний каталог>\bin, например D:\mongodb\mongo.exe.

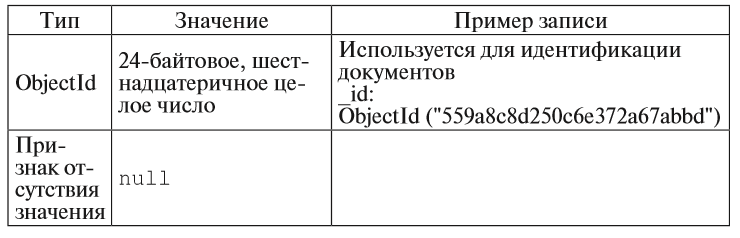
Происходит подключение к серверу MongoDB и устанавливается умалчиваемая база test, к которой адресуются последующие запросы. Для работы с БД может быть запущено несколько таких приложений. Далее в ответ на приглашение «>» в командной строке вводятся команды и обращения к методам управления данными, а также дополнительные команды для получения справок и выполнения функций операционной системы.

Можно сказать упрощенно, что MongoDB состоит из ``баз данных'', которые состоят из ``коллекций''. ``Коллекции'' состоят из ``документов''. Каждый ``документ'' состоит из ``полей''. ``Коллекции'' могут быть проиндексированы, что улучшает производительность выборки и сортировки. И наконец, получение данных из MongoDB сводится к получению``курсора'', который отдает эти данные по мере надобности.

Основной единицей хранения, доступа и обработки в базе MongoDB является документ. Документ имеет набор свойств — атрибутов. Каждый атрибут в документе задается парой <Имя атрибута> : <Значение атрибута>. Наборы документов хранятся в коллекциях. Хотя в одну коллекцию обычно помещаются документы, представляющие однотипные объекты, документы в одной коллекции не обязаны иметь одинаковые наборы атрибутов. Также не требуется совпадения типов значений одноименных атрибутов в разных документах коллекции. Для идентификации каждый документ имеет атрибут с именем \_id, содержащий уникальный ключ и набор заданных пользователем произвольных атрибутов. Если при включении документа ключ не задан, то он генерируется автоматически. Автоматически созданный ключ соответствует 24-разрядному шестнадцатеричному числу. В целом для задания документа используется формат JSON (JavaScript Object Notation).

Документ — неупорядоченный набор пар <Имя (ключ) атрибута> : <Значение атрибута>. Документ начинается с открывающей фигурной скобки и заканчивается закрывающей фигурной скобкой. Имя и значение атрибута разделяет двоеточие. Пары <Имя атрибута> : <Значение атрибута> в документе разделяются запятой. Значение атрибута может иметь простой предопределенный в системе тип. Простые (атомарные) типы данных представлены в табл.





Кроме простого типа значение атрибута может быть представлено массивом значений или другим документом. Таким образом, атрибуты в документах могут образовывать вложенные структуры. Массив — упорядоченный набор значений . Массив начинается с открывающей квадратной скобки и заканчивается закрывающей квадратной скобкой. Элементы массива разделены запятой и индексируются целыми числами, начиная с нуля. Вложенные массивы не допускаются. Однако если элемент массива является документом, то его атрибуты также могут быть заданы массивами.

Справочная команда help выводит сведения о назначении других команд и обращения к справочникам по методам отдельных объектов MongoDB. Например, информацию о методах базы данных, имеющей псевдоним db, можно получить обращением к методу db.help (), а выполнение метода db.<Имя коллекции>.help () предоставит информацию о методах коллекции. Команды, необходимые для начала работы с MongoDB, представлены в табл

|  |  |
| --- | --- |
| Команда | Назначение |
| Команды и методы для работы с БД в MongoDB Shell | |
| help | Вывод списка команд MongoDB Shell и справочных методов для БД |
| db.help () | Справочник по методам базы данных |
| db. <Имя коллекции>help () | Справочник по методам коллекции |
| show dbs | Вывод имен, имеющихся на сервере БД |
| use <Имя БД> | Установка текущей БД, с которой далее ведется работа. Установить можно и несуществующую базу. Создание первой коллекции приведет к созданию базы. Обращение к установленной базе выполняется под псевдонимом >db. Например, >db. getCollectionNames () выполняет метод БД, возвращающий имена коллекций в установленной базе |
| db.getName () | Вывод имени текущей базы |
| show collections | Вывод имен коллекций в установленной базе (команда приложения mongo shell) |
| show logs | Вывод имен используемых журналов. По умолчанию имя журнала global |
| db.dropDatabase () | Удаление базы |
| exit | Выход из mongo shell |
| Команды создания и обработки коллекции в БД | |
| db.createCollection (<Имя коллекции>. [,{size:… [, capped: …, max: …}]]) | Создание новой коллекции |
| db.getCollectionNames () | Вывод списка имен коллекций в виде массива. |
| db.<Имя коллекции>.find () | Вывод документов из коллекции |
| db.<Имя существ. коллекции>. renameCollection ("<Новое имя коллекции>") | Переименование коллекции |
| db.<Имя коллекции>.drop () | Удаление коллекции |
| db.<Имя коллекции>.find () | Вывод всех документов из заданной коллекции |

В документе между любыми лексемами можно задавать произвольное число пробелов. Пример документа, содержащего сведения об авторе Джеймсе Олдридже и заданных в массиве Books двух его книгах:

{Au\_id: "000–11–0001", Au\_lname: "Олдридж, Джеймс", Years: [1918, 2015], Contract: false,

Books: [{Title\_id: "BB1111", Title: "Морской орел"}, {Title\_id: "BB2222", Title: "Последний дюйм"}]}.

При добавлении документа в базу указывается коллекция, в которую включается и где хранится документ. Коллекции в БД должны иметь уникальные имена. Управление документами выполняется функциями (методами) коллекции. Для поиска, включения, удаления или изменения документа необходимо вызвать соответствующий метод коллекции. Коллекции являются объектами БД и имеют полный набор методов для доступа и обработки документов.

**Создание и удаление коллекций**

Далее следует выполнять команды, выделенные **жирным шрифтом.**

Создание новой коллекции выполняет метод createCollection для БД >db.createCollection (<Имя коллекции> [, {size:<Размер базы в байтах> [, capped: true|false, max:<Число документов>}]])

Например, создание коллекции authors

**>db.createCollection ("authors");**

приводит к созданию определения коллекции authors. Фактическое создание коллекции происходит при включении в нее первого документа.

Выполнение метода

**>db.createCollection (authors, {size: 3000000})**

создает коллекцию authors, для которой предварительно резервируется ~3 Мб в каталоге для баз данных, заданным параметром dbpath в конфигурационном файле mongod.cfg

Количественные характеристики коллекции выводит ее метод stats (): >db.<Имя коллекции>.stats (); Например, метод

**db.authors.stats ();**

выводит статистику коллекции authors в базе Authors (размеры букв в именах имеют значения):

"ns": "Authors. authors",

"count": 0,

"size": 0,

"storageSize": 3002368,

"numExtents": 1,

"nindexes": 1,

"lastExtentSize": 3002368,

"paddingFactor": 1,

"systemFlags": 1,

"userFlags": 0,

"totalIndexSize": 8176,

"indexSizes": { "\_id\_": 8176 },

"ok": 1

В строке "storageSize": 3002368 указан размер зарезервированной памяти

Необязательный параметр capped: true метода db.createCollection используется при создании ограниченных коллекций, в которых общее число документов не может превышать значение, заданное параметром max:<Число документов>. При добавлении в ограниченную коллекцию лишних документов происходит удаление наиболее старых документов. Ограниченные коллекции требуют обязательного указания размера резервируемой памяти (size) и не допускают добавления в существующие документы новых атрибутов.

Вывод списка имеющихся в базе коллекций выполняет метод **db.getCollectionNames ().**

Для полного удаления коллекции со всеми имеющимися в ней документами предназначен метод самой коллекции drop. Например, >db.authors.drop () удалит коллекцию authors из текущей БД.

Переименование коллекции выполняется методом renameCollection: db.<Имя существующей коллекции>.renameCollection ("<Новое имя коллекции>").

Создадим коллекцию OldCollection

**> db.createCollection("OldCollection");**

Убедимся в том, что она создана

**>db.getCollectionNames()**

Переименуем ее , дав ей имя NewCollection

**> db.OldCollection.renameCollection("NewCollection")**

Убедимся в том, что коллекция получила новое имя

**> db.getCollectionNames()**

Удалим коллекцию NewCollection

**> db.NewCollection.drop()**

Убедимся в том, что коллекция удалена

**> db.getCollectionNames()**

**Включение документов в коллекцию**

Добавление нового документа в коллекцию выполняет метод Insert (): db.<Имя коллекции>.insert (<Документ в формате JSON>)

или db.<Имя коллекции>.insert ({<Имя атрибута>:<Значение> [,..});

Например, включение в коллекцию авторов Николая Лескова может выполнить метод:

**> db.authors.insert ({"Au\_lname":"Лесков", "Au\_fname": "Николай", "birthday": ISODate ("1831-02-16")});**

При этом атрибут "birthday" (день рождения) имеет тип Date и поэтому задается строкой в аргументе функции ISODate, которая преобразует строку в значение типа дата-время в формате GMT. В примере время в течение суток не задано, поэтому будет доопределено нулем (началом суток).

Вывод документов из заданной коллекции

**> db.authors.find()**

Возвратит введенный документ

{ "\_id" : ObjectId("58695c396b9421d84b9efb93"), "Au\_lname" : "Лесков", "Au\_fname" : "Николай", "birthday" : ISODate("1831-02-16T00:00:00Z") }

В момент включения документа сервер MongoDB автоматически создал для него ключевой атрибут \_id, имеющий специальный тип ObjectId с уникальным значением ObjectId ("58695c396b9421d84b9efb93"). Ключ документа может использоваться в условии запроса для получения конкретного документа.

Запрос **> db.authors.find ({\_id: ObjectId("58695c396b9421d84b9efb93")})**

возвратит тот же документ

Пользователь при добавлении документа методом insert может задать свое уникальное в коллекции значение атрибута \_id. Например,

**db.authors.insert ({\_id: "000–11–1111", "Au\_ lname": "Пелевин", " Au\_fname ": "Виˊктор", "year\_of\_birth": 1962, Contract: true});**

По заданному пользователем значению \_id также возможен поиск документа: **db.authors.find ({\_id: "000–11– 1111"});**

Метод insert позволяет добавлять в коллекцию документы из переменной java script:

**var doc={"Au\_lname":"Н. В. Гоголь", "year\_of\_ birth":1809};**

**db.authors.insert (doc);**

Добавим в коллекцию также документ

**> var dsc={"\_id":"01","Фамилия":"Калинин","Имя":"Юрий", "Книги":["Букварь","Чтение"]};**

**db.authors.insert (dsc);**

и документ

**> db.authors.insert ({\_id: "02", "Автор": "Сумкин Н", "Книги ": ["Азбука"]})**

**Задание для самостоятельной работы**

Создать базу данных с коллекцией, содержащей результаты последней экзаме-национной сессии, в которой была бы информация о дисциплинах изученных Вами, преподавателях, которым Вы сдавали зачеты и экзамены, о форме отчет-ности (зачет или экзамен) и полученных оценках. Для ввода документов используйте разные способы. Выведете документы, содержащие информацию о сданных экзаменах

**Вопросы для защиты работы**

1.Перечислите типы данных и укажите способы их записи

2.Перечислите команды и методы для работы с БД в MongoDB

3. Перечислите команды создания и обработки коллекции в БД

4. Как задается атрибут в документе ?

5. Что такое массив и как он задается?

**Содержание отчета**

1.Номер и название лабораторной работы

2. Экранные формы, показывающие порядок выполнения лабораторной

работы, и результаты, полученные в ходе её выполнения.

3.Постановка задачи для самостоятельной работы

4. Экранные формы, показывающие порядок выполнения задания для

Самостоятельной работы с соответствующими пояснениями, и результаты, полученные в ходе её выполнения.

5.Ответы на вопросы для защиты