

分布式数据库原理与应用实验手册

实验四：MongoDB数据库的基本概念及数据库基本操作

【实验名称】 实验四：MongoDB数据库的基本概念及数据库基本操作

【实验目的】

1. 掌握 MongoDB 数据库的基本概念
2. 掌握 MongoDB 创建数据库
3. 掌握 MongoDB 删除数据库

【实验原理】

MongoDB 是一个开源文档数据库，提供高性能，高可用性和自动扩展，旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。

MongoDB 中的记录是一个文档，它是由字段和值(key=>value)对组成的数据结构。 MongoDB 文档与 JSON 对象相似。 字段的值可能包括其他文档，数组和文档数组。

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

在 mongodb 中基本的概念是文档(document)、集合(collection)、数据库(database)。

SQL 术语/概念	MongoDB 术语/概念	解释/说明
database	database	数据库
table	collection	数据库表/集合
row	document	数据记录行/文档
column	field	数据字段/域
index	index	索引

table joins		表连接,MongoDB 不支持
primary key	primary key	主键,MongoDB 自动将_id 字段设置为主键

例如：

id	user_name	email	age	city
1	Mark Hanks	mark@abc.com	25	Los Angeles
2	Richard Peter	richard@abc.com	31	Dallas



```

{
  "_id": ObjectId("5146bb52d8524270060001f3"),
  "age": 25,
  "city": "Los Angeles",
  "email": "mark@abc.com",
  "user_name": "Mark Hanks"
}
{
  "_id": ObjectId("5146bb52d8524270060001f2"),
  "age": 31,
  "city": "Dallas",
  "email": "richard@abc.com",
  "user_name": "Richard Peter"
}

```

【实验环境】

OS : Ubuntu16.04

MongoDB : v3.6 from I9000

【实验步骤】

题目0：进入mongodb shell

新建一个终端，在终端中输入以下命令：

```

cd ~
mkdir -p ./data/db
mongod --dbpath=./data/db

```

```

File Edit View Terminal Tabs Help
ua01@desktop:~/Desktop$ cd ~
ua01@desktop:~$ mkdir -p ./data/db
ua01@desktop:~$ mongod --dbpath=./data/db

```

命令执行完毕后，再新建一个终端，在第二个终端中输入以下命令，

mongo

```

Termin
File Edit View Terminal Tabs Help
ua01@desktop:~/Desktop$ mongo

```

执行命令后，进入mongodb shell（不要关闭第一个终端）

```
2018-08-04T10:22:36.643+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-08-04T10:22:36.643+0000 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestricted.
2018-08-04T10:22:36.643+0000 I CONTROL [initandlisten]
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten]
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten] ** WARNING: You are running on a NUMA machine.
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten] **           We suggest launching mongod like this to avoid performance problems:
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten] **           numactl --interleave=all mongod [other options]
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten]
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten] **           We suggest setting it to 'never'
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten]
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten] **           We suggest setting it to 'never'
2018-08-04T10:22:36.644+0000 I CONTROL [initandlisten]
>
```

注意：在mongodb程序执行过程中，第一个终端不要关闭！

题目一：数据库

一个 mongodb 中可以建立多个数据库。

MongoDB 的默认数据库为"db"，该数据库存储在 data 目录中。

MongoDB 的单个实例可以容纳多个独立的数据库，每一个都有自己的集合和权限，不同的数据库也放置在不同的文件中。

1. "show dbs" 命令可以显示所有数据的列表：

```
> show dbs
```

```
admin    0.000GB
```

```
config   0.000GB
```

```
local    0.000GB
```

2. 执行 "**db**" 命令可以显示当前数据库对象或集合：

```
> db
```

```
test
```

```
>
```

3. 运行"use"命令，可以连接到一个指定的数据库：

```
> use local
```

```
switched to db local
```

```
> db
```

```
local
```

```
>
```

以上示例命令中，"local" 是要链接的数据库。

数据库也通过名字来标识。数据库名可以是满足以下条件的任意 UTF-8 字符串。

✚ 不能是空字符串 ("")。

✚ 不得含有' ' (空格)、.、\$、/、\和\0 (空字符)。

✚ 应全部小写。

✚ 最多 64 字节。

有一些数据库名是保留的，可以直接访问这些有特殊作用的数据库。

admin：从权限的角度来看，这是"root"数据库。要是将一个用户添加到这个数据库，这个用户自动继承所有数据库的权限。一些特定的服务器端命令也只能从这个数据库运行，比如列出所有的数据库或者关闭服务器。

local: 这个数据永远不会被复制，可以用来存储限于本地单台服务器的任意集合

config: 当 Mongo 用于分片设置时，config 数据库在内部使用，用于保存分片的相关信息。

题目二：文档

文档是一组键值(key-value)对(即 BSON)。MongoDB 的文档不需要设置相同的字段，并且相同的字段不需要相同的数据类型，这与关系型数据库有很大的区别，也是 MongoDB 非常突出的特点。

一个简单的文档例子如下：

```
{"site": "translate.google.cn", "name": "google 翻译"}
```

下表列出了 RDBMS 与 MongoDB 对应的术语：

RDBMS	MongoDB
数据库	数据库
表格	集合
行	文档
列	字段
表联合	嵌入文档

主键	主键 (MongoDB 提供了 key 为 _id)
----	-----------------------------

数据库服务和客户端	
Mysqld/Oracle	mongod
mysql/sqlplus	mongo

需要注意的是：

- ①. 文档中的键/值对是有序的。
- ②. 文档中的值不仅可以是在双引号里面的字符串，还可以是其他几种数据类型（甚至可以是整个嵌入的文档）。
- ③. MongoDB 区分类型和大小写。
- ④. MongoDB 的文档不能有重复的键。
- ⑤. 文档的键是字符串。除了少数例外情况，键可以使用任意 UTF-8 字符。

文档键命名规范：

- ✚ 键不能含有\0 (空字符)。这个字符用来表示键的结尾。
- ✚ .和\$有特别的意义，只有在特定环境下才能使用。
- ✚ 以下划线"_"开头的键是保留的(不是严格要求的)。

题目三：集合

集合就是 MongoDB 文档组，类似于 RDBMS（关系数据库管理系统：Relational Database Management System)中的表格。

集合存在于数据库中，集合没有固定的结构，这意味着在对集合可以插入不同格式和类型的数据，但通常情况下插入集合的数据都会有一定的关联性。

比如，可以将以下不同数据结构的文档插入到集合中：。

```
{"site": "www.baidu.com"}  
{"site": "translate.google.cn", "name": "Google"}  
{"site": "docs.mongodb.com ", "name": "MongoDB 文档", "num": 5}
```

当第一个文档插入时，集合就会被创建。

合法的集合名

✚ 集合名不能是空字符串""。

✚ 集合名不能含有\0 字符（空字符），这个字符表示集合名的结尾。

✚ 集合名不能以"system."开头，这是为系统集合保留的前缀。

✚ 用户创建的集合名字不能含有保留字符。有些驱动程序的确支持在集合名里面包含，这是因为某些系统生成的集合中包含该字符。除非要访问这种系统创建的集合，否则千万不要在名字里出现\$。

如下示例：

```
db.col.findOne()
```

capped collections

Capped collections 就是固定大小的 collection。

它有很高的性能以及队列过期的特性(过期按照插入的顺序). 有点和 "RRD" 概念类似。

Capped collections 是高性能自动的维护对象的插入顺序。它非常适合类似记录日志的功能 和标准的 collection 不同，必须要显式的创建一个 capped collection，指定一个 collection 的大小，单位是字节。collection 的数据存储空间值提前分配的。

要注意的是指定的存储大小包含了数据库的头信息。

```
db.createCollection("mycoll", {capped:true, size:100000})
```

```
> db.createCollection("mycoll", {capped:true, size:100000})  
{ "ok" : 1 }
```

在 capped collection 中，能添加新的对象。

能进行更新，然而，对象不会增加存储空间。如果增加，更新就会失败。

数据库不允许进行删除。使用 drop()方法删除 collection 所有的行。

注意: 删除之后，必须显式的重新创建这个 collection。

在 32bit 机器中，capped collection 最大存储为 1e9(1X109)个字节。

删除 MongoDB 数据库和日志文件。

题目四：元数据

数据库的信息是存储在集合中。它们使用了系统的命名空间：

```
dbname.system.*
```

在 MongoDB 数据库中名字空间 <dbname>.system.* 是包含多种系统信息的特殊集合(Collection)，如下：

集合命名空间	描述
dbname.system.namespaces	列出所有名字空间。
dbname.system.indexes	列出所有索引。
dbname.system.profile	包含数据库概要(profile)信息。
dbname.system.users	列出所有可访问数据库的用户。
dbname.local.sources	包含复制对端 (slave) 的服务器信息和状态。

对于修改系统集合中的对象有如下限制。

在{{system.indexes}}插入数据，可以创建索引。但除此之外该表信息是不可变

的(特殊的 drop index 命令将自动更新相关信息)。

{{system.users}}是可修改的。 {{system.profile}}是可删除的。

题目五：MongoDB 数据类型

下表为 MongoDB 中常用的几种数据类型：

数据类型	描述
String	字符串。存储数据常用的数据类型。在 MongoDB 中，UTF-8 编码的字符串才是合法的。
Integer	整型数值。用于存储数值。根据所采用的服务器，可分为 32 位或 64 位。
Boolean	布尔值。用于存储布尔值（真/假）。
Double	双精度浮点值。用于存储浮点值。
Min/Max keys	将一个值与 BSON（二进制的 JSON）元素的最低值和最高值相对比。
Array	用于将数组或列表或多个值存储为一个键。
Timestamp	时间戳。记录文档修改或添加的具体时间。
Object	用于内嵌文档。
Null	用于创建空值。
Symbol	符号。该数据类型基本上等同于字符串类型，但不同的是，它一般用于采用特殊符号类型的语言。
Date	日期时间。用 UNIX 时间格式来存储当前日期或时间。可以指定自己的日期时间：创建 Date 对象，传入年月日信息。
Object ID	对象 ID。用于创建文档的 ID。
Binary Data	二进制数据。用于存储二进制数据。
Code	代码类型。用于在文档中存储 JavaScript 代码。
Regular expression	正则表达式类型。用于存储正则表达式。

题目六：MongoDB 创建数据库

MongoDB 创建数据库的语法格式如下：

```
use DATABASE_NAME
```

如果数据库不存在，则创建数据库，否则切换到指定数据库。

例如：

以下示例创建了数据库 Employee:

```
> use Employee  
  
switched to db Employee  
  
> db  
  
Employee  
  
>
```

如果想查看所有数据库，可以使用 show dbs 命令：

```
> show dbs  
  
admin    0.000GB  
config   0.000GB  
local    0.000GB  
  
>
```

可以看到，刚创建的数据库 Employee 并不在数据库的列表中，要显示它，需要向 Employee 数据库插入一些数据。

```
> db.Employee.insert({"name":"google"})  
  
WriteResult({ "nInserted" : 1 })  
  
> show dbs  
  
Employee 0.000GB  
admin     0.000GB  
config    0.000GB  
local     0.000GB  
  
>
```

MongoDB 中默认的数据库为 test，如果没有创建新的数据库，集合将存放在

test 数据库中。

题目七：MongoDB 删除数据库

MongoDB 删除数据库的语法格式如下：

```
db.dropDatabase()
```

删除当前数据库，默认为 test，可以使用 db 命令查看当前数据库名。

首先，查看所有数据库：

```
> show dbs
```

接下来切换到数据库 Employee：

```
> use Employee
switched to db Employee
>
```

执行删除命令：

```
> db.dropDatabase()
{ "dropped" : "Employee", "ok" : 1 }
>
```

最后，再通过 show dbs 命令数据库是否删除成功：

```
> show dbs
admin    0.000GB
config   0.000GB
local    0.000GB
>
```

