**部署mongodb服务**

mongodb概述

mongodb介绍

介于关系数据库和非关系数据库之间的产品

一款基于分布式文件存储的数据库，旨在为WEB应用提供可扩展的高性能数据存储解决方案

将数据存储为一个文档（类似于JSON对象，mysql里的记录），数据结构由键值（key=>values）对组成

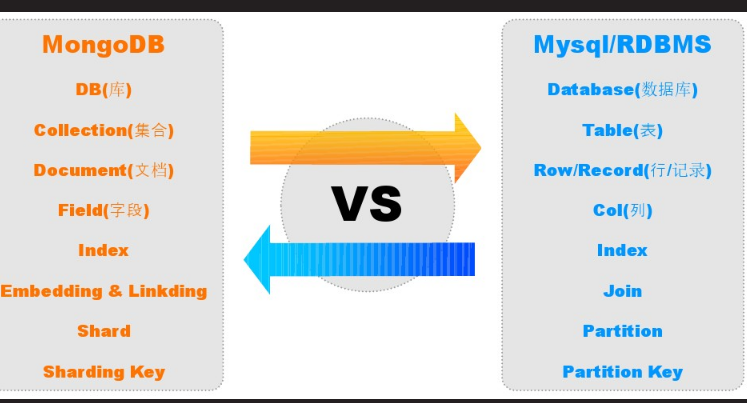
支持丰富的查询表达，可以设置任何属性的索引

支持副本集（相当于mysql里面的主从同步），分片

直接存储到硬盘

里头命令严格区分大小写

相关概念



搭建MDB服务器

装包

免安装，解压后即可使用

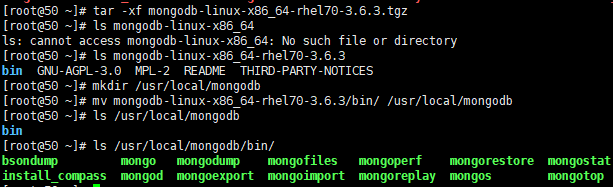
mkdir /usr/local/mongodb

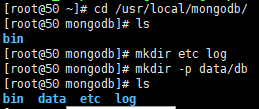
tar -xf mongodb-linux-x86\_64-rhel70-3.63.tgz

cp -r mongodb-linux-x86\_64-rhel70-3.63/bin /usr/local/mongodb

cd /usr/local/mongodb/

mkdir -p etc log data/db

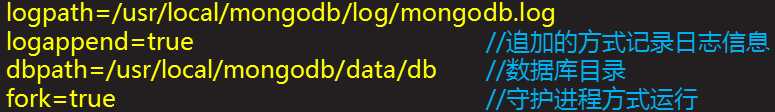




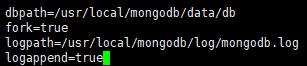
创建配置文件

手动创建服务主配置文件

vim mongodb.conf









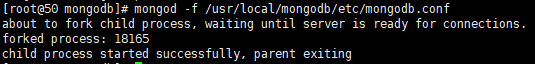


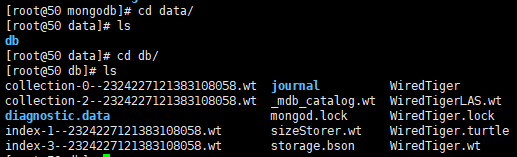


启动服务

启动服务

./bin/mongod -f /usr/local/mongodb/etc/mongodb.conf





查看进程

ps -C mongod



查看端口

netstat -tunlp | grep 27017



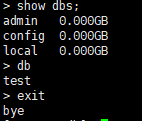
连接服务

本地连接，默认没有密码

/usr/local/mongodb/bin/mongo







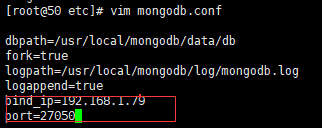
指定服务的IP地址和端口号

停止服务





修改配置文件

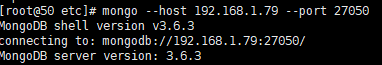
不指为默认

启动服务





连接



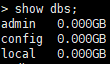
**mongodb基本使用**

常用管理命令

数据库管理

查看、创建、切换、删除库

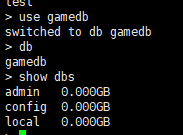
show dbs //查看已有库



db //显示当前所在的库



use 库名//切换库，若库不存在延时创建库

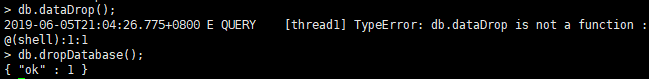


show collections或show tables//查看库下已有集合



db.dropdatabase()//删除当前所在的库





数据库名称规范

不能是空字符串（“”）

不能含有’’（空格）、.、$、/、\和\0（空字符）

应全部小写

最多64字节

集合管理

查看、创建、删除集合

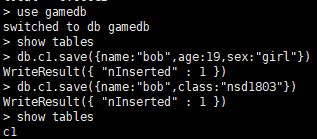
show collections 或show tables //查看集合



db.集合名.drop() //删除集合



db.集合名.save({“.”}) //创建集合，集合不存在时，创建并添加文档



集合命名规范

不能是空字符串””

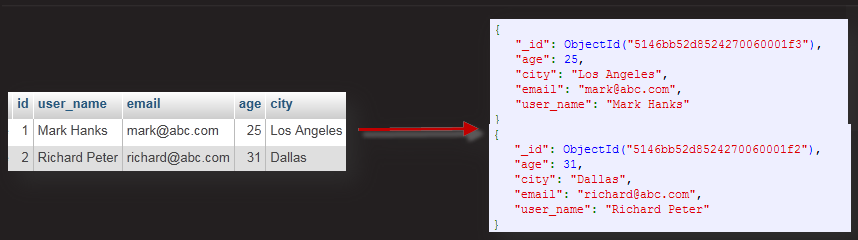
不能含有\0字符（空字符），此字符表示集合的结尾

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

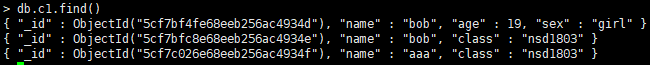
文档基本管理

文档：类似于mysql表里的记录



查看、统计、添加、删除文档

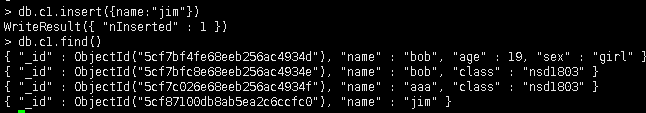
db.集合名.find()



db.集合名.count()



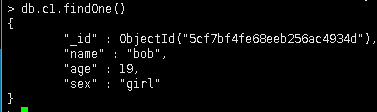
db.集合名.insert({“name”:”jim”})



db.集合名.find({条件})



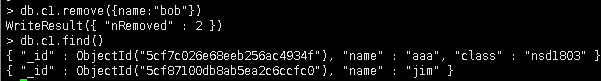
db.集合名.findOne() //返回一条文档



db.集合名.remove({}) //删除所有文档



db.集合名.remove({条件}) //删除匹配的所有文档



插入记录



基本数据类型

字符string/布尔bool/空null

字符串string

UTF-8字符串都可以表示为字符串类型的数据

{name:”张三”}或{school:”aaa”}

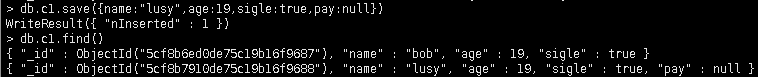
布尔值bool

布尔类型有两个值true和false,{x:true}



空null

用于表示空值或者不存在的字段，{x:null}



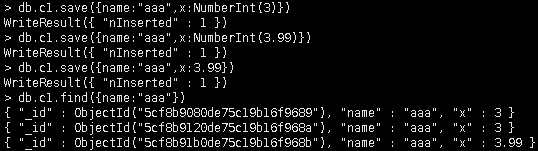
数值/数组array

数值

shell默认使用64位浮点型数值。{x:3.14}或{x:3}

numberint（4字节整数）{x:numberint(3)}

numberlong（8字节整数）{x:numberlong(3)}



数组array

数据列表或数据集可以表示为数组

{x:[“a”,”b”,”c”]}

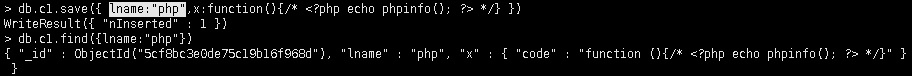


代码/日期/对象

代码

查询和文档中可以包含任何javascript代码

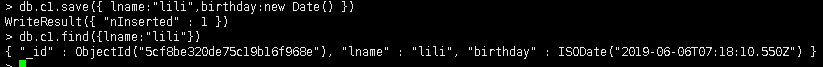
{x:function(){/\*代码\*/}}



日期

日期被存储为自新纪元以来净经过的毫秒数，不含时区

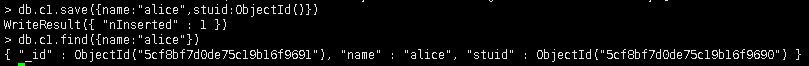
{x:new date()}



对象（如果不指定，默认生成，相当于mysql主键）

对象id是一个12字节的字符串，是文档的唯一标识

{x:ObjectId()}



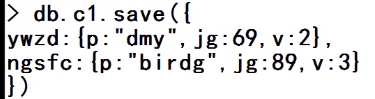
内嵌/正则表达式

内嵌

文档可以嵌套其他文档，被嵌套的文档作为值来处理

{aaa:

{address:”beijing”,trl:”8888888”,person:”hansy”}}



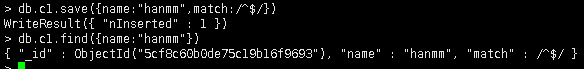




正则表达式

查询时，使用正则表达式作为限定条件

{x:/正则表达式/}



数据导入导出

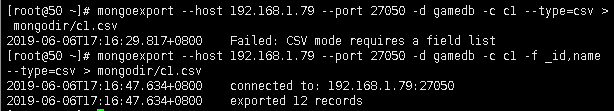
csv适合导出指定列的

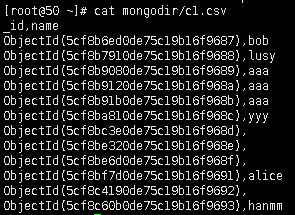
json适合导出所有列的

数据导出

语法格式1：

mongoexport [--host ip地址 --port 端口] -d 库名 -c 集合名 -f 字段名1,字段2 --type=csv > 目录名/文件名.csv





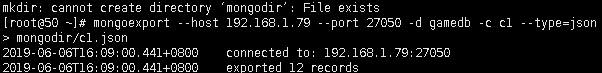
语法格式2：

mongoexport --host ip地址 --port 端口 -库名 -c 集合名 -q ‘{条件}’ -f 字段名1,字段名2 --type=csv > 目录名/文件名.csv

注意：导出为csv格式必须使用-f指定字段名列表

语法格式3：

mongoexport [--host ip地址 --port 端口] -d 库名 -c 集合名 [-q ‘{条件}’ -f 字段列表] --type=json > 目录名/文件名.json



目录必须创建，文件如果存在则会覆盖

数据导入（事先可以没有库和集合，会延时创建，json格式的重复导入记录会报错，csv格式的会追加导入）

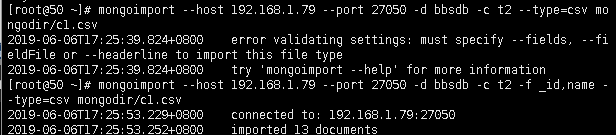
语法格式1：

mongoimport --host ip地址 --port 端口 -d 库名 -c 集合名 --type=json 目录名/文件名.json

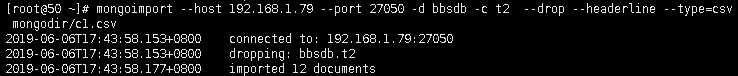


语法格式2：

mongoimport --host ip地址 --port 端口 -d 库名 -c 集合名 --type=csv [--hearderline] [--drop] 目录名/文件名.csv

必须指定列名否则会报错，默认会把标题也当数据导入

1. 导入数据时，若库和集合不存在，则先创建库和集合后再导入数据；
2. 若库和集合已经存在，则以追加的方式导入数据到集合里；
3. 使用--drop选项可以删除原数据后导入新数据，--headerline 忽而略标题





数据的备份恢复

数据备份

备份数据所有库到当前目录下的dump目录下

mongodump [--host ip地址 --port 端口]

备份时指定备份的库和备份目录

mongodump [--host ip地址 --port 端口] -d 数据库名 -c 集合名 -o 目录

查看bson文件内容

bsondump ./dump/bbs/t1.bson

数据恢复

语法格式

mongorestore --host ip地址 --port 端口 -d 数据库名 [-c 集合名] 备份目录名

总结

数据类型

字符串类型

数值类型、布尔类型

空/正则/代码

数值、数组、日期

对象

内嵌