# 1.Mongodb启动为服务

I:\MongoDB\bin>mongod.exe --bind\_ip 127.0.0.1 --logpath "f:\data\dbConf\mongodb.log" --logappend --dbpath "f:\data\db" --port 3721 --serviceName "My\_MongodbService" --serviceDisplayName "My\_MongodbService" –install

|  |  |
| --- | --- |
| **参数** | **描述** |
| --bind\_ip | 绑定服务IP，若绑定127.0.0.1，则只能本机访问，不指定默认本地所有IP |
| --logpath | 定MongoDB日志文件，注意是指定文件不是目录 |
| --logappend | 使用追加的方式写日志 |
| --dbpath | 指定数据库路径 |
| --port | 指定服务端口号，默认端口27017 |
| --serviceName | 指定服务名称 |
| --serviceDisplayName | 指定服务名称，有多个mongodb服务时执行。 |
| --install | 指定作为一个Windows服务安装。 |

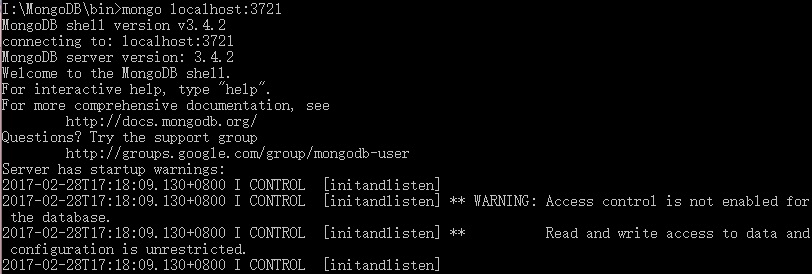
注意: f:\data\dbConf这个文件夹需要先建好,否则执行命令的时候会报错

2017-02-28T17:17:17.483+0800 F CONTROL [main] Failed global initialization: FileNotOpen: Failed to open "f:\data\dbConf\mongodb.log"

# 2.基本操作

## 连接mongo

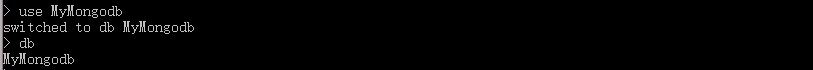
I:\MongoDB\bin>mongo localhost:3721 (3721是上一步启动mongodb时用的端口)



## 创建数据库

use DatabaseName

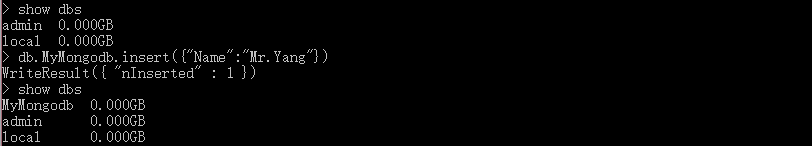
(这个写你想创建的数据库名字即可.如果,数据库不存在,则创建,如果已经存在,那么切换到这个数据库)



db 命令可以显示当前操作的是那一个数据库

show dbs 可以查看系统有多少个数据库

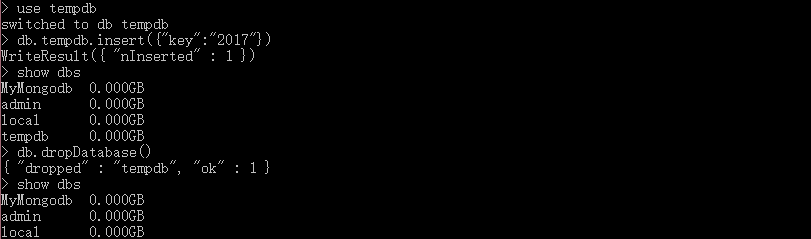
当使用use MyMongodb创建数据库后,使用show dbs 时是不能看到该新建的数据库的。只有当MyMongodb中数据时，使用show dbs时才能看到该数据库



## 删除数据库

use databaseName 切换到要删除的数据库

db.dropDatabase() 删除数据库



这里是先新建一个tempdb的数据库，然后，再把它删除掉。我们在删除的时候可以先用db命令来查看当前操作的是那一个数据库，避免误删除。

## 增

1.

db.col.insert({

title:'Mr.Yang',

location:'chengdu',

tags:['asp.net','b/s']

})

2.

jsondata=({

title:'Miss.Yang',

country:’china’

})

db.col.insert(jsondata)



## 删

db.collection.remove(

<query>,

{

justOne: <boolean>,

writeConcern: <document>

}

)

参数说明：

query :（可选）删除的文档的条件。

justOne : （可选）如果设为 true 或 1，则只删除一个文档。

writeConcern :（可选）抛出异常的级别。

## 改

db.collection.update(

<query>,

<update>,

{

upsert: <boolean>,

multi: <boolean>,

writeConcern: <document>

}

)

参数说明：

query : update的查询条件，类似sql update查询内where后面的。

update : update的对象和一些更新的操作符（如$,$inc...）等，也可以理解为sql update查询内set后面的

upsert : 可选，这个参数的意思是，如果不存在update的记录，是否插入objNew,true为插入，默认是false，不插入。

multi : 可选，mongodb 默认是false,只更新找到的第一条记录，如果这个参数为true,就把按条件查出来多条记录全部更新。

writeConcern :可选，抛出异常的级别。

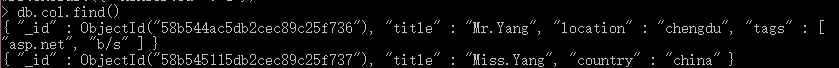
例如：

db.col.update({'title':'MongoDB 教程'},{$set:{'title':'MongoDB'}},{multi:true})

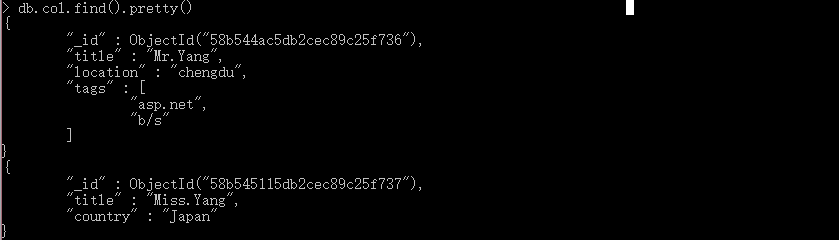
## 查

注意：查询条件的value区分大小写

db.col.find()



db.col.find().pretty() pretty 按格式化方法显示数据

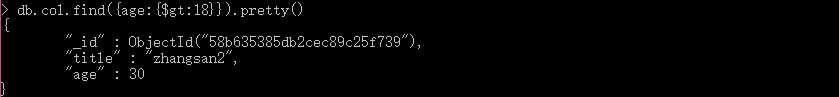


## MongoDB 与 RDBMS Where 语句比较

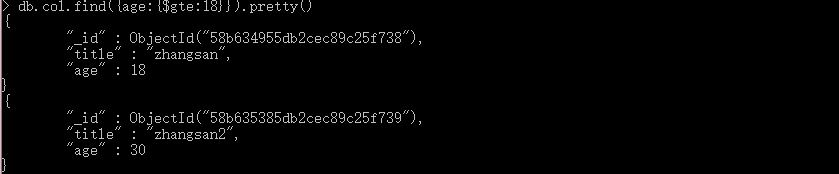
如果你熟悉常规的 SQL 数据，通过下表可以更好的理解 MongoDB 的条件语句查询：

| **操作** | **格式** | **范例** | **RDBMS中的类似语句** |
| --- | --- | --- | --- |
| 等于 | {<key>:<value>} | db.col.find({"title":"Mr.Yang"}).pretty() | where title = ' Mr.Yang ' |
| 小于 | {<key>:{$lt:<value>}} | db.col.find({"likes":{$lt:50}}).pretty() | where likes < 50 |
| 小于或等于 | {<key>:{$lte:<value>}} | db.col.find({"likes":{$lte:50}}).pretty() | where likes <= 50 |
| 大于 | {<key>:{$gt:<value>}} | db.col.find({"likes":{$gt:50}}).pretty() | where likes > 50 |
| 大于或等于 | {<key>:{$gte:<value>}} | db.col.find({"likes":{$gte:50}}).pretty() | where likes >= 50 |
| 不等于 | {<key>:{$ne:<value>}} | db.col.find({"likes":{$ne:50}}).pretty() | where likes != 50 |

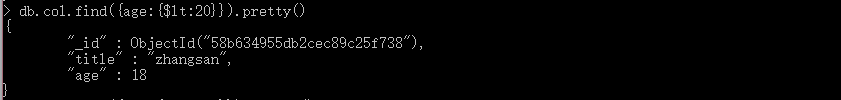
age >18



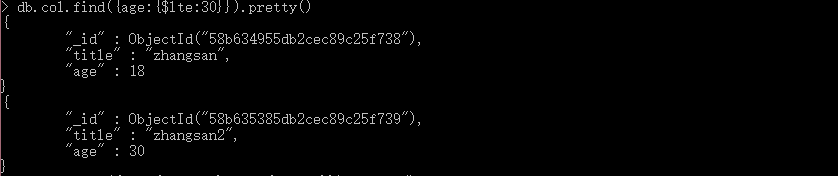
Age>=18



Age<20



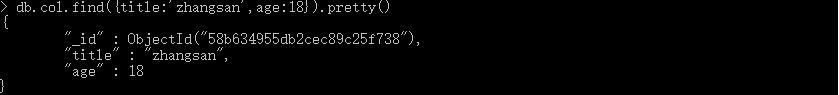
Age<=30



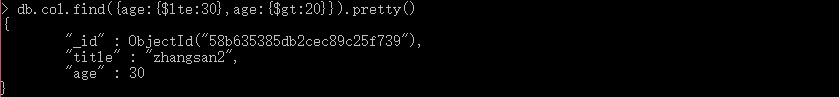
## MongoDB AND 条件

db.col.find({key1:'value1',key2:values}).pretty()

db.col.find({title:'zhangsan',age:18}).pretty()

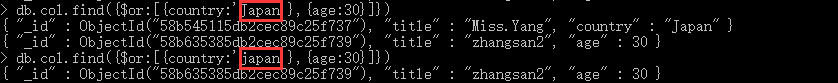


20<age<=30



## MongoDB OR 条件

db.col.find( { $or: [ {key1: value1}, {key2:value2} ] }).pretty()



## AND 和 OR 联合使用

db.col.find({title:'Mr.Yang',$or:[{location:'chengdu'}]})



## Like

**where name like %m%**

db.users.find({"name": /.\*m.\*/})

db.users.find({"name": /m/})

**where name like 'm%'**

db.users.find({name: /^m/})

**where name like ‘%m’**

db.users.find({name: /m$/})