MongoDB 命令

帮助help 命令

查看当前MongoDB 版本

version()

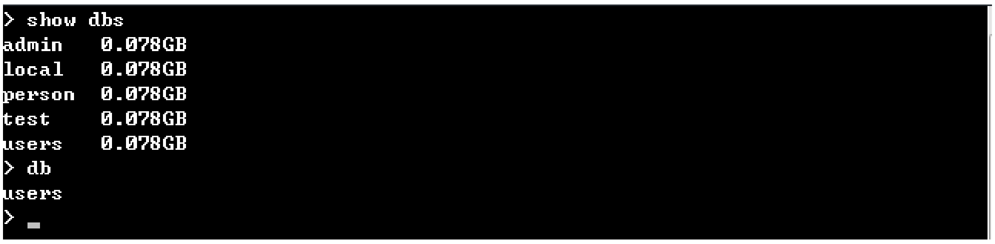
创建数据库 & 切换数据库

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。

查看当前数据库

db

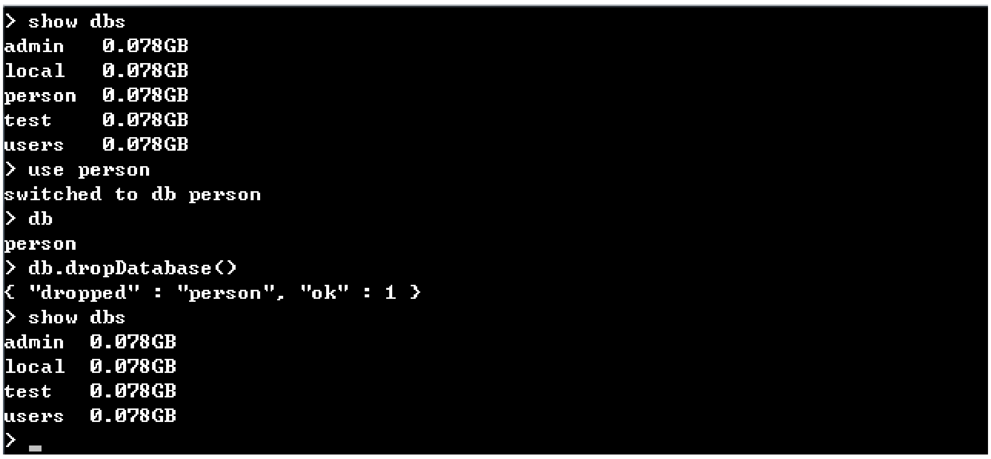


```
> show dbs
admin    0.078GB
local    0.078GB
person   0.078GB
test     0.078GB
users    0.078GB
> db
users
> _
```

删除数据库

db.dropDatabase()

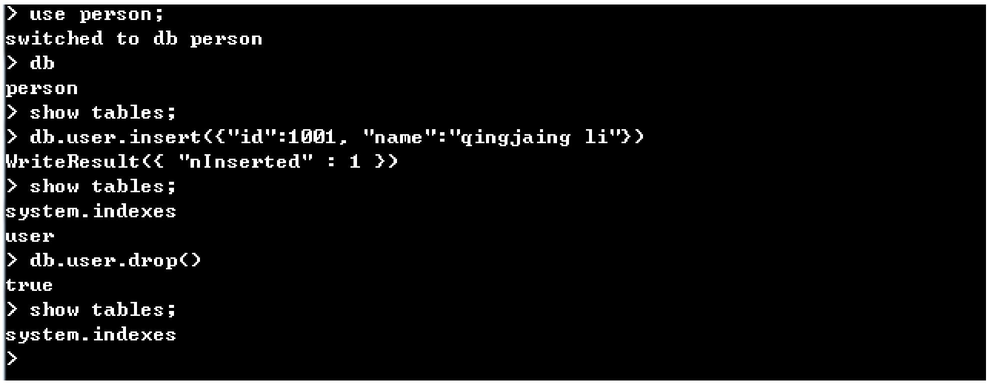
删除当前数据库



```
> show dbs
admin    0.078GB
local    0.078GB
person   0.078GB
test     0.078GB
users    0.078GB
> use person
switched to db person
> db
person
> db.dropDatabase()
{ "dropped" : "person", "ok" : 1 }
> show dbs
admin  0.078GB
local  0.078GB
test   0.078GB
users  0.078GB
> _
```

删除集合

db.collection.drop()

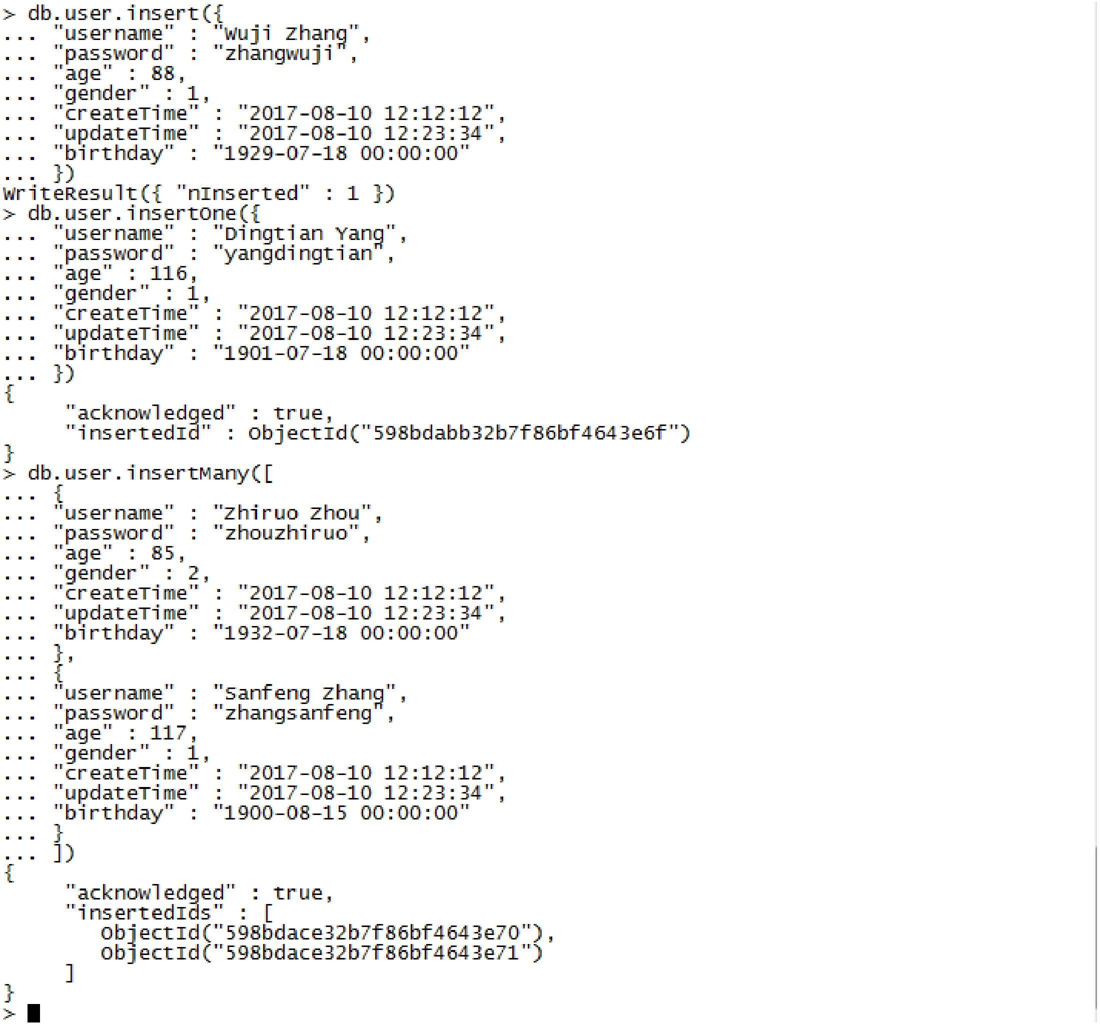


```
> use person;
switched to db person
> db
person
> show tables;
> db.user.insert({"id":1001, "name":"qingjaing li"})
WriteResult({ "nInserted" : 1 })
> show tables;
system.indexes
user
> db.user.drop()
true
> show tables;
system.indexes
>
```

插入文档

3.2 版本后还有以下几种语法可用于插入文档:

- db.collection.insertOne():向指定集合中插入一条文档数据
- db.collection.insertMany():向指定集合中插入多条文档数据



```
> db.user.insert({
... "username" : "Wuji Zhang",
... "password" : "zhangwuji",
... "age" : 88,
... "gender" : 1,
... "createTime" : "2017-08-10 12:12:12",
... "updateTime" : "2017-08-10 12:23:34",
... "birthday" : "1929-07-18 00:00:00"
... })
WriteResult({ "nInserted" : 1 })
> db.user.insertOne({
... "username" : "Dingtian Yang",
... "password" : "yangdingtian",
... "age" : 116,
... "gender" : 1,
... "createTime" : "2017-08-10 12:12:12",
... "updateTime" : "2017-08-10 12:23:34",
... "birthday" : "1901-07-18 00:00:00"
... })
{
        "acknowledged" : true,
        "insertedId" : ObjectId("598bdabb32b7f86bf4643e6f")
}
> db.user.insertMany([
... {
... "username" : "Zhiruo Zhou",
... "password" : "zhouzhiruo",
... "age" : 85,
... "gender" : 2,
... "createTime" : "2017-08-10 12:12:12",
... "updateTime" : "2017-08-10 12:23:34",
... "birthday" : "1932-07-18 00:00:00"
... },
... {
... "username" : "Sanfeng Zhang",
... "password" : "zhangsanfeng",
... "age" : 117,
... "gender" : 1,
... "createTime" : "2017-08-10 12:12:12",
... "updateTime" : "2017-08-10 12:23:34",
... "birthday" : "1900-08-15 00:00:00"
... }
... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("598bdace32b7f86bf4643e70"),
                ObjectId("598bdace32b7f86bf4643e71")
        ]
}
> █
```

删除文档

在执行remove()函数前先执行find()命令来判断执行的条件是否正确，这是一个比较好的习惯。

remove() 方法的基本语法格式:

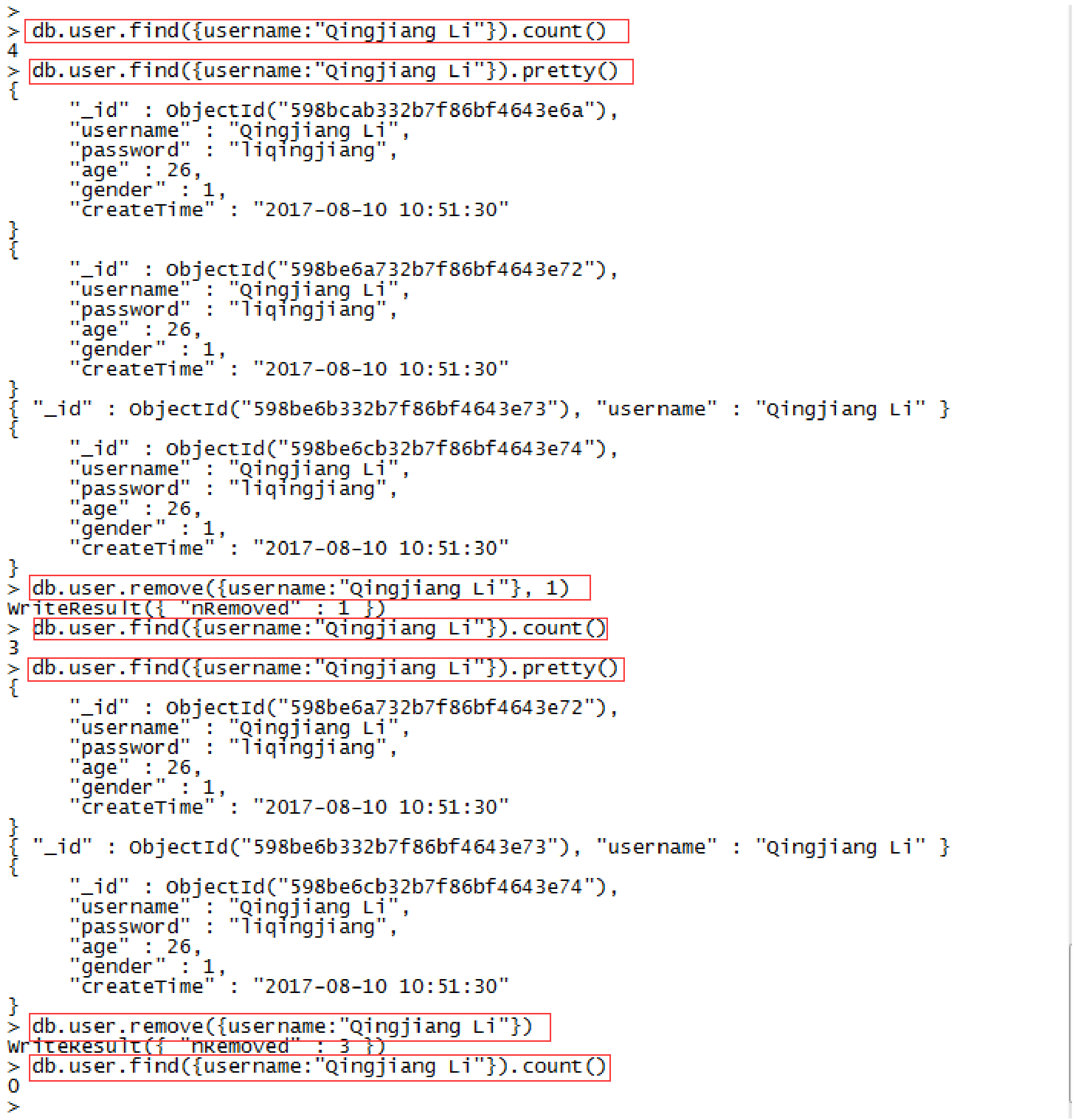
```
db.collection.remove(
   <query>,
   <justOne>
)
```

MongoDB 2.6 版本以后的语法格式如下：

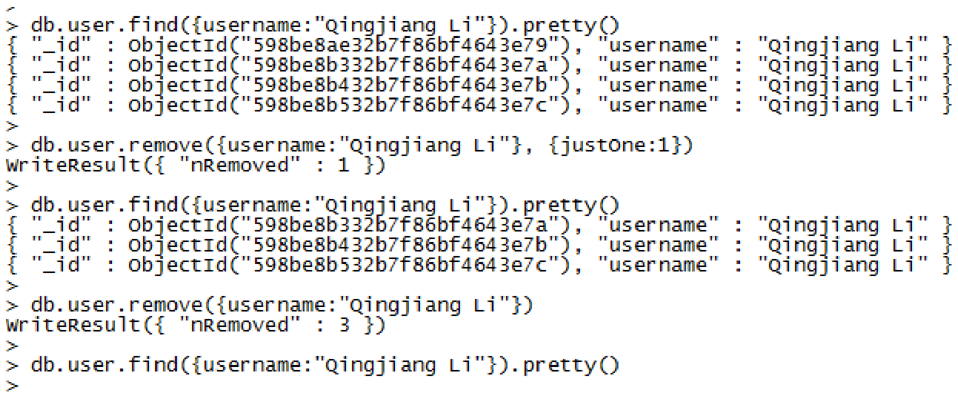
```
db.collection.remove(
   <query>,
   {
     justOne: <boolean>,
     writeConcern: <document>
   }
)
```

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。



```
>
> db.user.find({username:"Qingjiang Li"}).count()
4
> db.user.find({username:"Qingjiang Li"}).pretty()
{
        "_id" : ObjectId("598bcab332b7f86bf4643e6a"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 26,
        "gender" : 1,
        "createTime" : "2017-08-10 10:51:30"
}
{
        "_id" : ObjectId("598be6a732b7f86bf4643e72"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 26,
        "gender" : 1,
        "createTime" : "2017-08-10 10:51:30"
}
{ "_id" : ObjectId("598be6b332b7f86bf4643e73"), "username" : "Qingjiang Li" }
{
        "_id" : ObjectId("598be6cb32b7f86bf4643e74"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 26,
        "gender" : 1,
        "createTime" : "2017-08-10 10:51:30"
}
> db.user.remove({username:"Qingjiang Li"}, 1)
WriteResult({ "nRemoved" : 1 })
> db.user.find({username:"Qingjiang Li"}).count()
3
> db.user.find({username:"Qingjiang Li"}).pretty()
{
        "_id" : ObjectId("598be6a732b7f86bf4643e72"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 26,
        "gender" : 1,
        "createTime" : "2017-08-10 10:51:30"
}
{ "_id" : ObjectId("598be6b332b7f86bf4643e73"), "username" : "Qingjiang Li" }
{
        "_id" : ObjectId("598be6cb32b7f86bf4643e74"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 26,
        "gender" : 1,
        "createTime" : "2017-08-10 10:51:30"
}
> db.user.remove({username:"Qingjiang Li"})
WriteResult({ "nRemoved" : 3 })
> db.user.find({username:"Qingjiang Li"}).count()
0
>
```



```
> db.user.find({username:"Qingjiang Li"}).pretty()
{ "_id" : ObjectId("598be8ae32b7f86bf4643e79"), "username" : "Qingjiang Li" }
{ "_id" : ObjectId("598be8b332b7f86bf4643e7a"), "username" : "Qingjiang Li" }
{ "_id" : ObjectId("598be8b432b7f86bf4643e7b"), "username" : "Qingjiang Li" }
{ "_id" : ObjectId("598be8b532b7f86bf4643e7c"), "username" : "Qingjiang Li" }
>
> db.user.remove({username:"Qingjiang Li"}, {justOne:1})
WriteResult({ "nRemoved" : 1 })
>
> db.user.find({username:"Qingjiang Li"}).pretty()
{ "_id" : ObjectId("598be8b332b7f86bf4643e7a"), "username" : "Qingjiang Li" }
{ "_id" : ObjectId("598be8b432b7f86bf4643e7b"), "username" : "Qingjiang Li" }
{ "_id" : ObjectId("598be8b532b7f86bf4643e7c"), "username" : "Qingjiang Li" }
>
> db.user.remove({username:"Qingjiang Li"})
WriteResult({ "nRemoved" : 3 })
>
> db.user.find({username:"Qingjiang Li"}).pretty()
>
```

更新文档

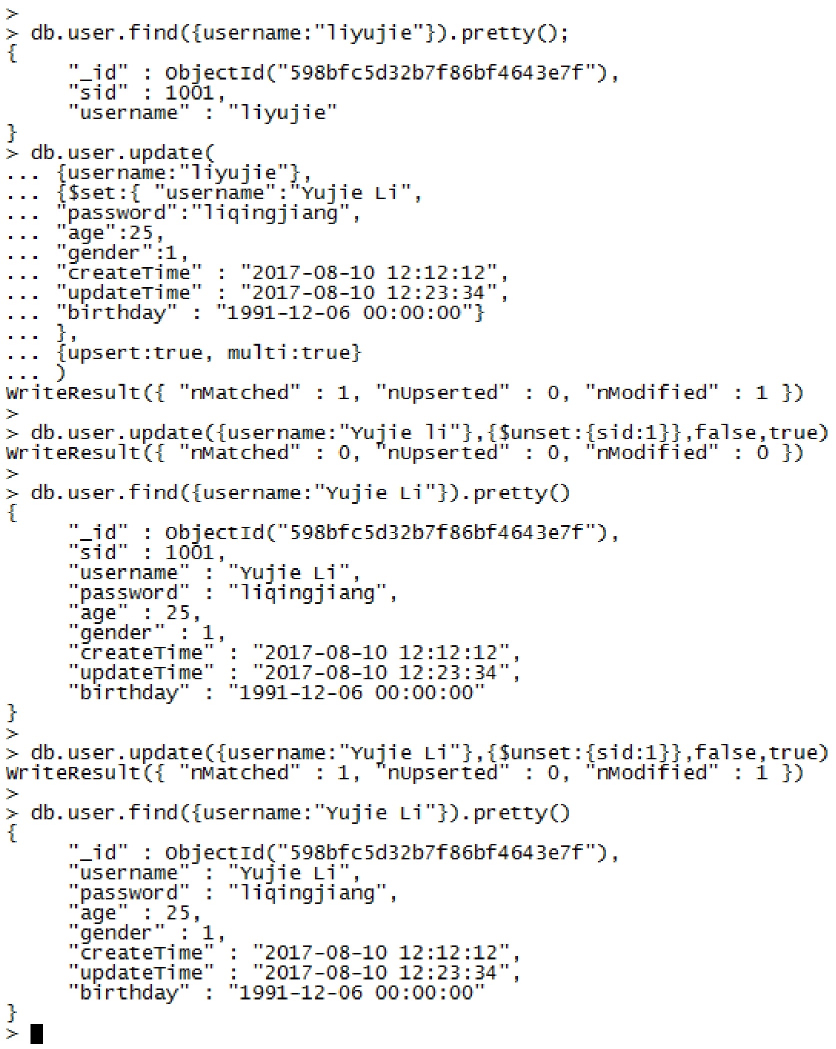
MongoDB 使用 **update()** 和 **save()** 方法来更新集合中的文档。

update() 方法用于更新已存在的文档。语法格式如下：

```
db.collection.update(
   <query>,
   <update>,
   {
     upsert: <boolean>,
     multi: <boolean>,
     writeConcern: <document>
   }
)
```

参数说明：

- **query** : update的查询条件
- **update** : update的对象和一些更新的操作符（如$,$inc...）等
- **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。



```
>
> db.user.find({username:"liyujie"}).pretty();
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "sid" : 1001,
        "username" : "liyujie"
}
> db.user.update(
... {username:"liyujie"},
... {$set:{ "username":"Yujie Li",
... "password":"liqingjiang",
... "age":25,
... "gender":1,
... "createTime" : "2017-08-10 12:12:12",
... "updateTime" : "2017-08-10 12:23:34",
... "birthday" : "1991-12-06 00:00:00"}
... },
... {upsert:true, multi:true}
... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.user.update({username:"Yujie li"},{$unset:{sid:1}},false,true)
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
>
> db.user.find({username:"Yujie Li"}).pretty()
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "sid" : 1001,
        "username" : "Yujie Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1991-12-06 00:00:00"
}
>
> db.user.update({username:"Yujie Li"},{$unset:{sid:1}},false,true)
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.user.find({username:"Yujie Li"}).pretty()
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "username" : "Yujie Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1991-12-06 00:00:00"
}
> █
```

save() 方法通过传入的文档来替换已有文档。语法格式如下：

```
db.collection.save(
   <document>,
   {
     writeConcern: <document>
   }
)
```

参数说明：

- **document** : 文档数据。
- **writeConcern** :可选，抛出异常的级别。

查询文档

MongoDB 查询文档使用 find() 方法。

find() 方法以非结构化的方式来显示所有文档。

MongoDB 查询数据的语法格式如下：

db.collection.find(query, projection)

- query ：可选，使用查询操作符指定查询条件
- projection ：可选，使用投影操作符指定返回的键。查询时返回文档中所有键值， 只需省略该参数即可（默认省略）。

如果你需要以易读的方式来读取数据，可以使用 pretty() 方法，语法格式如下：

>db.col.find().pretty()

pretty() 方法以格式化的方式来显示所有文档。

除了 find() 方法之外，还有一个 findOne() 方法，它只返回一个文档。

MongoDB 与 RDBMS Where 语句比较

大于，小于，等于及其组合

操作	格式	范例	RDBMS中的类似语句
等于	{<key>:<value>}	db.col.find({"name":"jiang"}).pretty()	where name = 'jiang'
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

and 条件

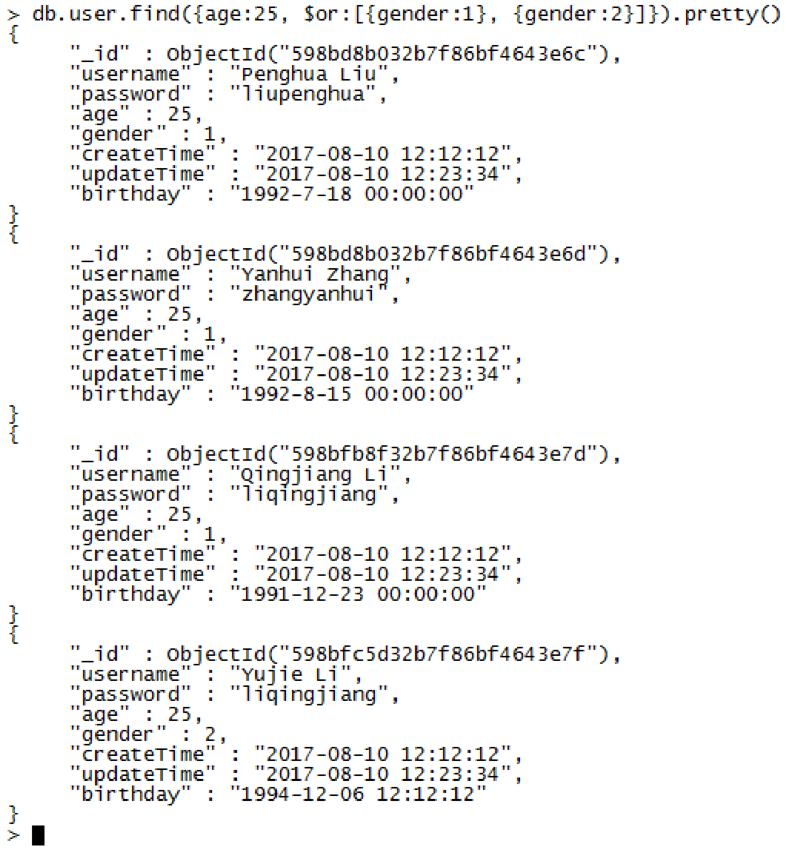
```
db.col.find({key1:value1, key2:value2}).pretty()
```

or 条件

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

and & or

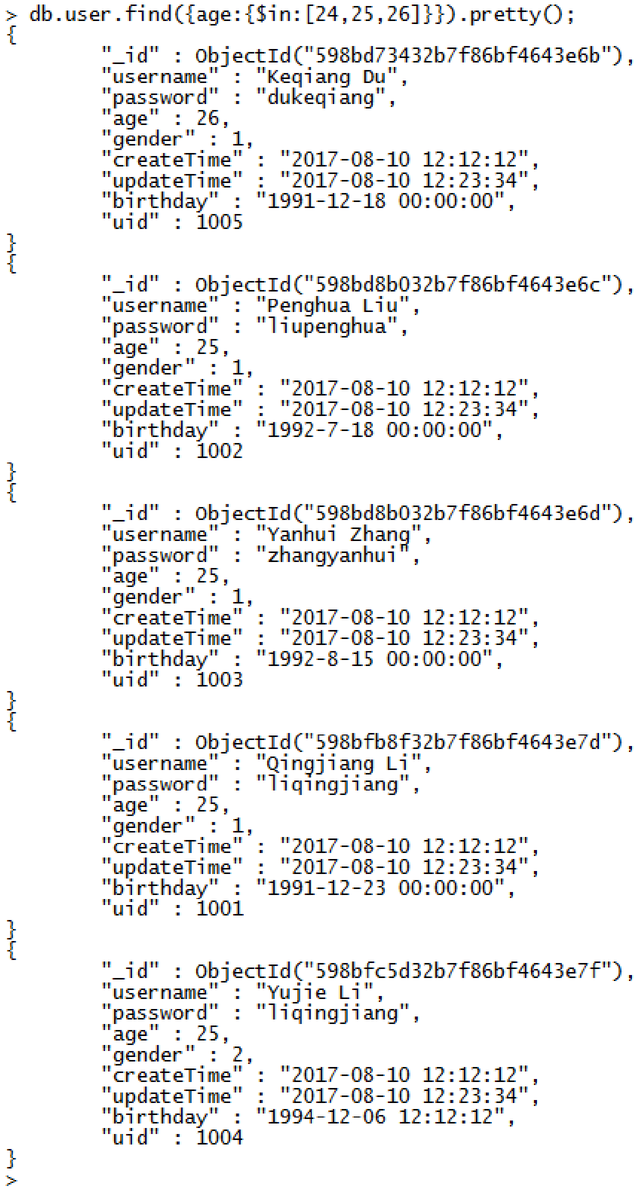
eg:



```
> db.user.find({age:25, $or:[{gender:1}, {gender:2}]}).pretty()
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6c"),
        "username" : "Penghua Liu",
        "password" : "liupenghua",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-7-18 00:00:00"
}
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6d"),
        "username" : "Yanhui Zhang",
        "password" : "zhangyanhui",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-8-15 00:00:00"
}
{
        "_id" : ObjectId("598bfb8f32b7f86bf4643e7d"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1991-12-23 00:00:00"
}
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "username" : "Yujie Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 2,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1994-12-06 12:12:12"
}
> 
```

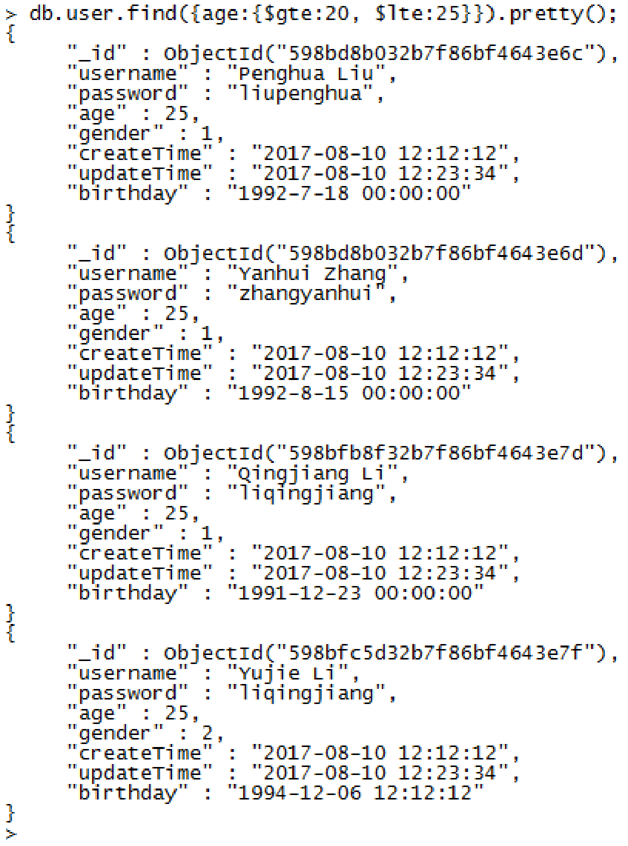
$in 条件

```
db.user.find({age:{$in:[24,25,26]}}).pretty();
```



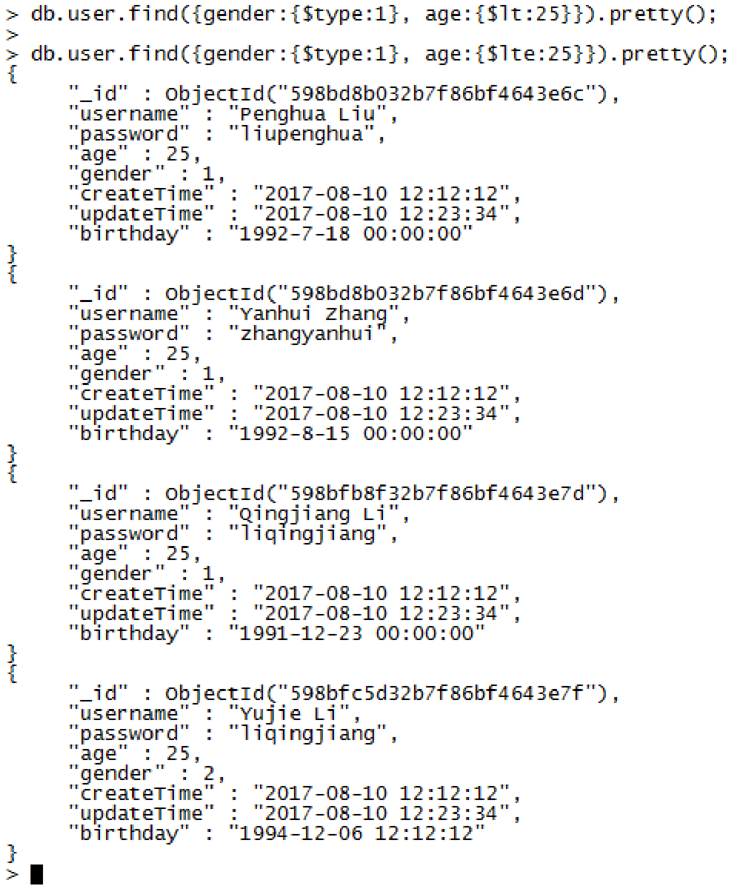
```
> db.user.find({age:{$in:[24,25,26]}}).pretty();
{
        "_id" : ObjectId("598bd73432b7f86bf4643e6b"),
        "username" : "Keqiang Du",
        "password" : "dukeqiang",
        "age" : 26,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1991-12-18 00:00:00",
        "uid" : 1005
}
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6c"),
        "username" : "Penghua Liu",
        "password" : "liupenghua",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-7-18 00:00:00",
        "uid" : 1002
}
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6d"),
        "username" : "Yanhui Zhang",
        "password" : "zhangyanhui",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-8-15 00:00:00",
        "uid" : 1003
}
{
        "_id" : ObjectId("598bfb8f32b7f86bf4643e7d"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1991-12-23 00:00:00",
        "uid" : 1001
}
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "username" : "Yujie Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 2,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1994-12-06 12:12:12",
        "uid" : 1004
}
>
```

在两个数之间



```
> db.user.find({age:{$gte:20, $lte:25}}).pretty();
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6c"),
        "username" : "Penghua Liu",
        "password" : "liupenghua",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-7-18 00:00:00"
}
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6d"),
        "username" : "Yanhui Zhang",
        "password" : "zhangyanhui",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-8-15 00:00:00"
}
{
        "_id" : ObjectId("598bfb8f32b7f86bf4643e7d"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1991-12-23 00:00:00"
}
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "username" : "Yujie Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 2,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1994-12-06 12:12:12"
}
>
```

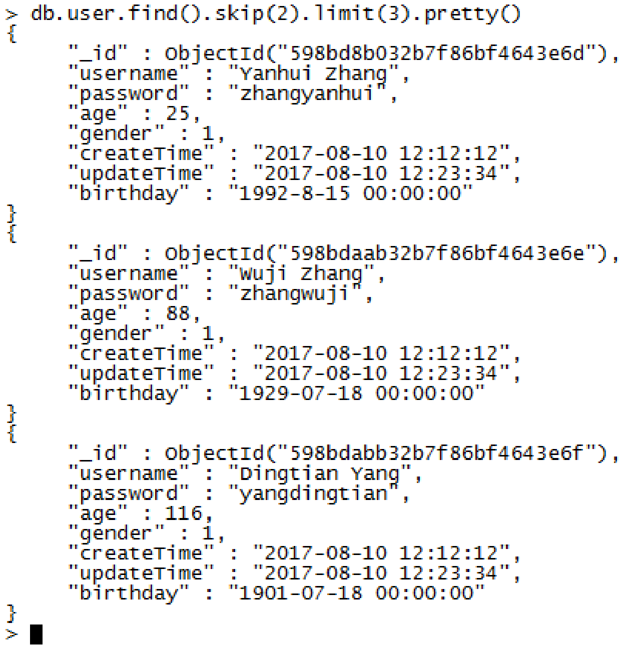
数据类型匹配查询



```
> db.user.find({gender:{$type:1}, age:{$lt:25}}).pretty();
>
> db.user.find({gender:{$type:1}, age:{$lte:25}}).pretty();
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6c"),
        "username" : "Penghua Liu",
        "password" : "liupenghua",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-7-18 00:00:00"
}
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6d"),
        "username" : "Yanhui Zhang",
        "password" : "zhangyanhui",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-8-15 00:00:00"
}
{
        "_id" : ObjectId("598bfb8f32b7f86bf4643e7d"),
        "username" : "Qingjiang Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1991-12-23 00:00:00"
}
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "username" : "Yujie Li",
        "password" : "liqingjiang",
        "age" : 25,
        "gender" : 2,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1994-12-06 12:12:12"
}
>
```

分页查询

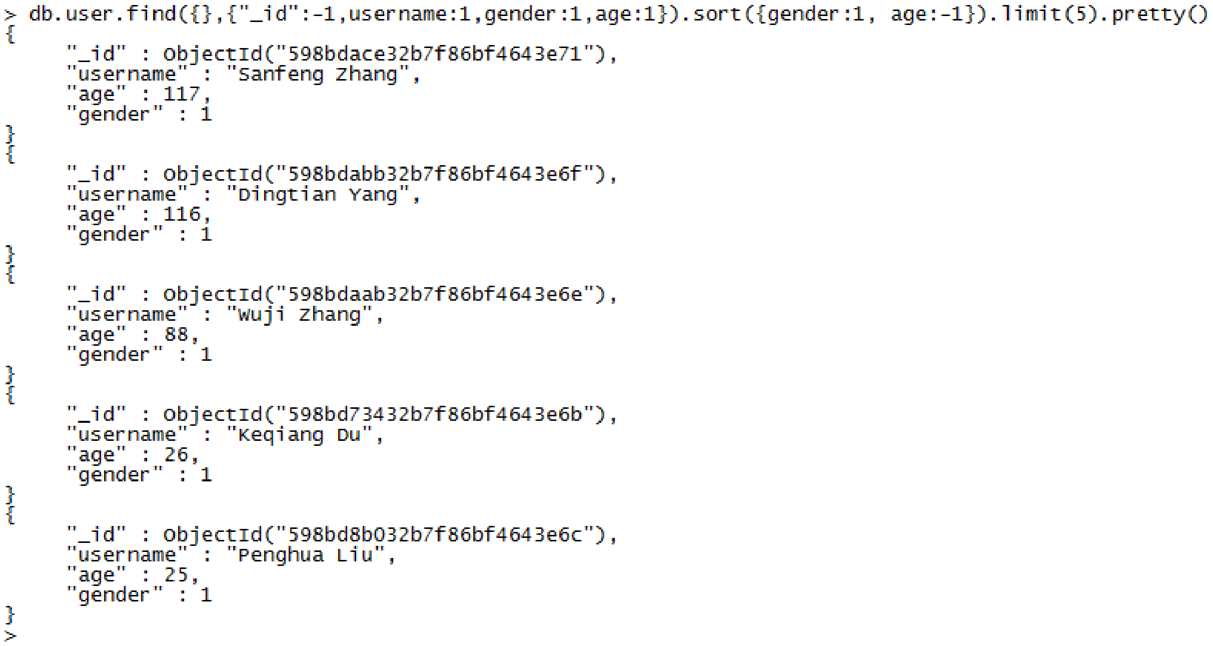
skip() , limit()



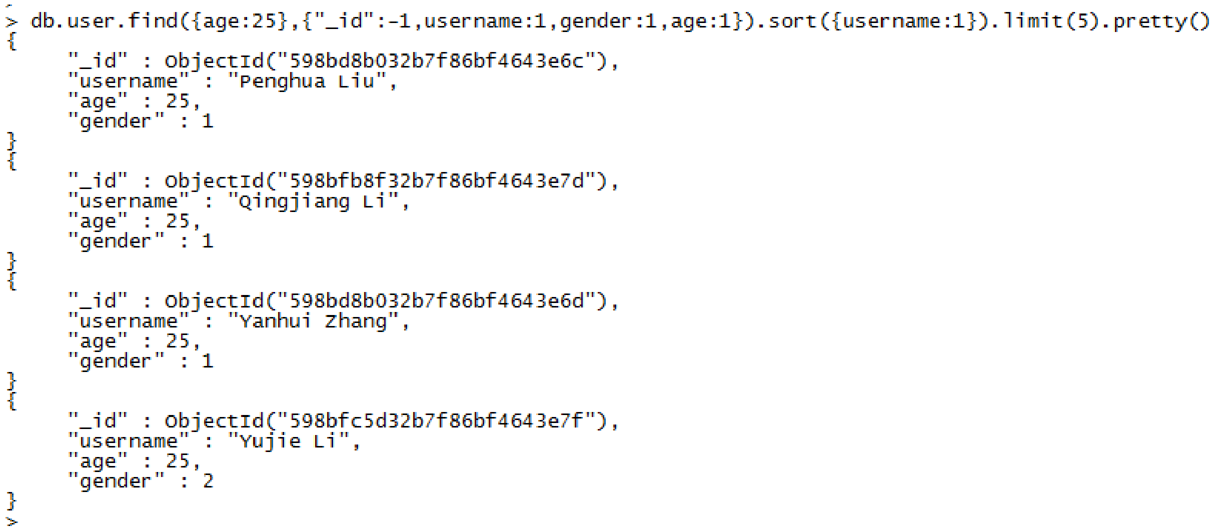
```
> db.user.find().skip(2).limit(3).pretty()
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6d"),
        "username" : "Yanhui Zhang",
        "password" : "zhangyanhui",
        "age" : 25,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1992-8-15 00:00:00"
}
{
        "_id" : ObjectId("598bdaab32b7f86bf4643e6e"),
        "username" : "Wuji Zhang",
        "password" : "zhangwuji",
        "age" : 88,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1929-07-18 00:00:00"
}
{
        "_id" : ObjectId("598bdabb32b7f86bf4643e6f"),
        "username" : "Dingtian Yang",
        "password" : "yangdingtian",
        "age" : 116,
        "gender" : 1,
        "createTime" : "2017-08-10 12:12:12",
        "updateTime" : "2017-08-10 12:23:34",
        "birthday" : "1901-07-18 00:00:00"
}
> 
```

当查询时同时使用sort,skip,limit，无论位置先后，最先执行顺序 sort再skip再limit。

排序



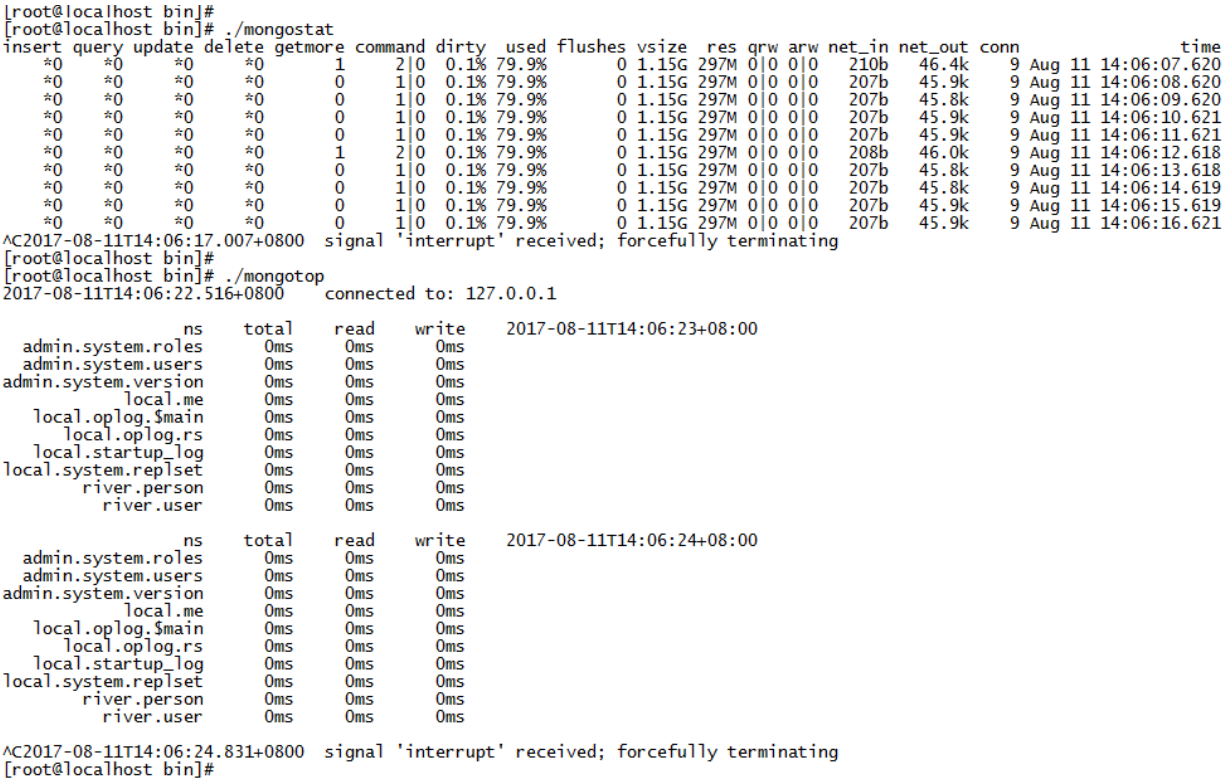
```
> db.user.find({},{"_id":-1,username:1,gender:1,age:1}).sort({gender:1, age:-1}).limit(5).pretty()
{
        "_id" : ObjectId("598bdace32b7f86bf4643e71"),
        "username" : "Sanfeng Zhang",
        "age" : 117,
        "gender" : 1
}
{
        "_id" : ObjectId("598bdabb32b7f86bf4643e6f"),
        "username" : "Dingtian Yang",
        "age" : 116,
        "gender" : 1
}
{
        "_id" : ObjectId("598bdaab32b7f86bf4643e6e"),
        "username" : "Wuji Zhang",
        "age" : 88,
        "gender" : 1
}
{
        "_id" : ObjectId("598bd73432b7f86bf4643e6b"),
        "username" : "Keqiang Du",
        "age" : 26,
        "gender" : 1
}
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6c"),
        "username" : "Penghua Liu",
        "age" : 25,
        "gender" : 1
}
>
```



```
> db.user.find({age:25},{"_id":-1,username:1,gender:1,age:1}).sort({username:1}).limit(5).pretty()
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6c"),
        "username" : "Penghua Liu",
        "age" : 25,
        "gender" : 1
}
{
        "_id" : ObjectId("598bfb8f32b7f86bf4643e7d"),
        "username" : "Qingjiang Li",
        "age" : 25,
        "gender" : 1
}
{
        "_id" : ObjectId("598bd8b032b7f86bf4643e6d"),
        "username" : "Yanhui Zhang",
        "age" : 25,
        "gender" : 1
}
{
        "_id" : ObjectId("598bfc5d32b7f86bf4643e7f"),
        "username" : "Yujie Li",
        "age" : 25,
        "gender" : 2
}
>
```

MongoDB　监控

MongoDB中提供了mongostat 和 mongotop 两个命令来监控MongoDB的运行情况。



```
[root@localhost bin]#
[root@localhost bin]# ./mongostat
insert query update delete getmore command dirty  used flushes vsize  res qrw arw net_in net_out conn                time
    *0    *0     *0     *0       1     2|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   210b   46.4k    9 Aug 11 14:06:07.620
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.9k    9 Aug 11 14:06:08.620
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.8k    9 Aug 11 14:06:09.620
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.9k    9 Aug 11 14:06:10.621
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.9k    9 Aug 11 14:06:11.621
    *0    *0     *0     *0       1     2|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   208b   46.0k    9 Aug 11 14:06:12.618
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.8k    9 Aug 11 14:06:13.618
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.8k    9 Aug 11 14:06:14.619
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.9k    9 Aug 11 14:06:15.619
    *0    *0     *0     *0       0     1|0  0.1% 79.9%       0 1.15G 297M 0|0 0|0   207b   45.9k    9 Aug 11 14:06:16.621
^C2017-08-11T14:06:17.007+0800  signal 'interrupt' received; forcefully terminating
[root@localhost bin]#
[root@localhost bin]# ./mongotop
2017-08-11T14:06:22.516+0800    connected to: 127.0.0.1

                  ns    total    read    write    2017-08-11T14:06:23+08:00
  admin.system.roles      0ms     0ms      0ms
  admin.system.users      0ms     0ms      0ms
admin.system.version      0ms     0ms      0ms
            local.me      0ms     0ms      0ms
   local.oplog.$main      0ms     0ms      0ms
      local.oplog.rs      0ms     0ms      0ms
   local.startup_log      0ms     0ms      0ms
local.system.replset      0ms     0ms      0ms
        river.person      0ms     0ms      0ms
          river.user      0ms     0ms      0ms

                  ns    total    read    write    2017-08-11T14:06:24+08:00
  admin.system.roles      0ms     0ms      0ms
  admin.system.users      0ms     0ms      0ms
admin.system.version      0ms     0ms      0ms
            local.me      0ms     0ms      0ms
   local.oplog.$main      0ms     0ms      0ms
      local.oplog.rs      0ms     0ms      0ms
   local.startup_log      0ms     0ms      0ms
local.system.replset      0ms     0ms      0ms
        river.person      0ms     0ms      0ms
          river.user      0ms     0ms      0ms

^C2017-08-11T14:06:24.831+0800  signal 'interrupt' received; forcefully terminating
[root@localhost bin]#
```