[**MongoDB基本命令用**](http://www.cnblogs.com/xusir/archive/2012/12/24/2830957.html)

成功启动MongoDB后，再打开一个命令行窗口输入mongo，就可以进行数据库的一些操作。

输入help可以看到基本操作命令：

show dbs:显示数据库列表   
show collections：显示当前数据库中的集合（类似关系数据库中的表）   
show users：显示用户

use <db name>：切换当前数据库，这和MS-SQL里面的意思一样   
db.help()：显示数据库操作命令，里面有很多的命令   
db.foo.help()：显示集合操作命令，同样有很多的命令，foo指的是当前数据库下，一个叫foo的集合，并非真正意义上的命令   
db.foo.find()：对于当前数据库中的foo集合进行数据查找（由于没有条件，会列出所有数据）   
db.foo.find( { a : 1 } )：对于当前数据库中的foo集合进行查找，条件是数据中有一个属性叫a，且a的值为1

MongoDB没有创建数据库的命令，但有类似的命令。

如：如果你想创建一个“myTest”的数据库，先运行use myTest命令，之后就做一些操作（如：db.createCollection('user')）,这样就可以创建一个名叫“myTest”的数据库。

数据库常用命令

1、Help查看命令提示

 help

  db.help();

  db.yourColl.help();

  db.youColl.find().help();

  rs.help();

2、切换/创建数据库

 use yourDB;  当创建一个集合(table)的时候会自动创建当前数据库

3、查询所有数据库

 show dbs;

4、删除当前使用数据库

 db.dropDatabase();

5、从指定主机上克隆数据库

 db.cloneDatabase(“127.0.0.1”); 将指定机器上的数据库的数据克隆到当前数据库

6、从指定的机器上复制指定数据库数据到某个数据库

 db.copyDatabase("mydb", "temp", "127.0.0.1");将本机的mydb的数据复制到temp数据库中

7、修复当前数据库

 db.repairDatabase();

8、查看当前使用的数据库

 db.getName();

 db; db和getName方法是一样的效果，都可以查询当前使用的数据库

9、显示当前db状态

 db.stats();

10、当前db版本

 db.version();

11、查看当前db的链接机器地址

 db.getMongo();

Collection聚集集合

1、创建一个聚集集合（table）

 db.createCollection(“collName”, {size: 20, capped: 5, max: 100});

2、得到指定名称的聚集集合（table）

 db.getCollection("account");

3、得到当前db的所有聚集集合

 db.getCollectionNames();

4、显示当前db所有聚集索引的状态

 db.printCollectionStats();

 用户相关

1、添加一个用户

 db.addUser("name");

 db.addUser("userName", "pwd123", true); 添加用户、设置密码、是否只读

2、数据库认证、安全模式

 db.auth("userName", "123123");

3、显示当前所有用户

 show users;

4、删除用户

 db.removeUser("userName");

其他

1、查询之前的错误信息

 db.getPrevError();

2、清除错误记录

 db.resetError();

查看聚集集合基本信息

1、查看帮助 db.yourColl.help();

2、查询当前集合的数据条数 db.yourColl.count();

3、查看数据空间大小 db.userInfo.dataSize();

4、得到当前聚集集合所在的db db.userInfo.getDB();

5、得到当前聚集的状态 db.userInfo.stats();

6、得到聚集集合总大小 db.userInfo.totalSize();

7、聚集集合储存空间大小 db.userInfo.storageSize();

8、Shard版本信息 db.userInfo.getShardVersion()

9、聚集集合重命名 db.userInfo.renameCollection("users"); 将userInfo重命名为users

10、删除当前聚集集合 db.userInfo.drop();

聚集集合查询

1、查询所有记录

db.userInfo.find();

相当于：select\* from userInfo;

默认每页显示20条记录，当显示不下的情况下，可以用it迭代命令查询下一页数据。注意：键入it命令不能带“；”

但是你可以设置每页显示数据的大小，用DBQuery.shellBatchSize= 50;这样每页就显示50条记录了。

2、查询去掉后的当前聚集集合中的某列的重复数据

db.userInfo.distinct("name");

会过滤掉name中的相同数据

相当于：select distict name from userInfo;

3、查询age = 22的记录

db.userInfo.find({"age": 22});

相当于： select \* from userInfo where age = 22;

4、查询age > 22的记录

db.userInfo.find({age: {$gt: 22}});

相当于：select \* from userInfo where age >22;

5、查询age < 22的记录

db.userInfo.find({age: {$lt: 22}});

相当于：select \* from userInfo where age <22;

6、查询age >= 25的记录

db.userInfo.find({age: {$gte: 25}});

相当于：select \* from userInfo where age >= 25;

7、查询age <= 25的记录

db.userInfo.find({age: {$lte: 25}});

8、查询age >= 23 并且 age <= 26

db.userInfo.find({age: {$gte: 23, $lte: 26}});

9、查询name中包含 mongo的数据

db.userInfo.find({name: /mongo/});

//相当于%%

select \* from userInfo where name like ‘%mongo%’;

10、查询name中以mongo开头的

db.userInfo.find({name: /^mongo/});

select \* from userInfo where name like ‘mongo%’;

11、查询指定列name、age数据

db.userInfo.find({}, {name: 1, age: 1});

相当于：select name, age from userInfo;

当然name也可以用true或false,当用ture的情况下河name:1效果一样，如果用false就是排除name，显示name以外的列信息。

12、查询指定列name、age数据, age > 25

db.userInfo.find({age: {$gt: 25}}, {name: 1, age: 1});

相当于：select name, age from userInfo where age >25;

13、按照年龄排序

升序：db.userInfo.find().sort({age: 1});

降序：db.userInfo.find().sort({age: -1});

14、查询name = zhangsan, age = 22的数据

db.userInfo.find({name: 'zhangsan', age: 22});

相当于：select \* from userInfo where name = ‘zhangsan’ and age = ‘22’;

15、查询前5条数据

db.userInfo.find().limit(5);

相当于：selecttop 5 \* from userInfo;

16、查询10条以后的数据

db.userInfo.find().skip(10);

相当于：select \* from userInfo where id not in (

selecttop 10 \* from userInfo

);

17、查询在5-10之间的数据

db.userInfo.find().limit(10).skip(5);

可用于分页，limit是pageSize，skip是第几页\*pageSize

18、or与 查询

db.userInfo.find({$or: [{age: 22}, {age: 25}]});

相当于：select \* from userInfo where age = 22 or age = 25;

19、查询第一条数据

db.userInfo.findOne();

相当于：selecttop 1 \* from userInfo;

db.userInfo.find().limit(1);

20、查询某个结果集的记录条数

db.userInfo.find({age: {$gte: 25}}).count();

相当于：select count(\*) from userInfo where age >= 20;

21、按照某列进行排序

db.userInfo.find({sex: {$exists: true}}).count();

相当于：select count(sex) from userInfo;

索引

1、创建索引

db.userInfo.ensureIndex({name: 1});

db.userInfo.ensureIndex({name: 1, ts: -1});

2、查询当前聚集集合所有索引

db.userInfo.getIndexes();

3、查看总索引记录大小

db.userInfo.totalIndexSize();

4、读取当前集合的所有index信息

db.users.reIndex();

5、删除指定索引

db.users.dropIndex("name\_1");

6、删除所有索引索引

db.users.dropIndexes();

 修改、添加、删除集合数据

1、添加

db.users.save({name: ‘zhangsan’, age: 25, sex: true});

添加的数据的数据列，没有固定，根据添加的数据为准

2、修改

db.users.update({age: 25}, {$set: {name: 'changeName'}}, false, true);

相当于：update users set name = ‘changeName’ where age = 25;

db.users.update({name: 'Lisi'}, {$inc: {age: 50}}, false, true);

相当于：update users set age = age + 50 where name = ‘Lisi’;

db.users.update({name: 'Lisi'}, {$inc: {age: 50}, $set: {name: 'hoho'}}, false, true);

相当于：update users set age = age + 50, name = ‘hoho’ where name = ‘Lisi’;

3、删除

db.users.remove({age: 132});

4、查询修改删除

db.users.findAndModify({

query: {age: {$gte: 25}},

sort: {age: -1},

update: {$set: {name: 'a2'}, $inc: {age: 2}},

remove: true

});

db.runCommand({ findandmodify : "users",

query: {age: {$gte: 25}},

sort: {age: -1},

update: {$set: {name: 'a2'}, $inc: {age: 2}},

remove: true

});

*update* 或 *remove* 其中一个是必须的参数; 其他参数可选。

|  |  |  |
| --- | --- | --- |
| **参数** | **详解** | **默认值** |
| *query* | 查询过滤条件 | {} |
| *sort* | 如果多个文档符合查询过滤条件，将以该参数指定的排列方式选择出排在首位的对象，该对象将被操作 | {} |
| *remove* | 若为true，被选中对象将在返回前被删除 | N/A |
| *update* | 一个 [修改器对象](http://www.mongodb.org/display/DOCS/Updating) | N/A |
| *new* | 若为true，将返回修改后的对象而不是原始对象。在删除操作中，该参数被忽略。 | false |
| *fields* | 参见[Retrieving a Subset of Fields](http://www.mongodb.org/display/DOCS/Retrieving+a+Subset+of+Fields) (1.5.0+) | All fields |
| *upsert* | 创建新对象若查询结果为空。 [示例](http://github.com/mongodb/mongo/blob/master/jstests/find_and_modify4.js) (1.5.4+) | false |

语句块操作

1、简单Hello World

print("Hello World!");

这种写法调用了print函数，和直接写入"Hello World!"的效果是一样的；

2、将一个对象转换成json

tojson(new Object());

tojson(new Object('a'));

3、循环添加数据

> for (var i = 0; i < 30; i++) {

... db.users.save({name: "u\_" + i, age: 22 + i, sex: i % 2});

... };

这样就循环添加了30条数据，同样也可以省略括号的写法

> for (var i = 0; i < 30; i++) db.users.save({name: "u\_" + i, age: 22 + i, sex: i % 2});

也是可以的，当你用db.users.find()查询的时候，显示多条数据而无法一页显示的情况下，可以用it查看下一页的信息；

4、find 游标查询

>var cursor = db.users.find();

> while (cursor.hasNext()) {

printjson(cursor.next());

}

这样就查询所有的users信息，同样可以这样写

var cursor = db.users.find();

while (cursor.hasNext()) { printjson(cursor.next); }

同样可以省略{}号

5、forEach迭代循环

db.users.find().forEach(printjson);

forEach中必须传递一个函数来处理每条迭代的数据信息

6、将find游标当数组处理

var cursor = db.users.find();

cursor[4];

取得下标索引为4的那条数据

既然可以当做数组处理，那么就可以获得它的长度：cursor.length();或者cursor.count();

那样我们也可以用循环显示数据

for (var i = 0, len = c.length(); i < len; i++) printjson(c[i]);

7、将find游标转换成数组

> var arr = db.users.find().toArray();

> printjson(arr[2]);

用toArray方法将其转换为数组

8、定制我们自己的查询结果

只显示age <= 28的并且只显示age这列数据

db.users.find({age: {$lte: 28}}, {age: 1}).forEach(printjson);

db.users.find({age: {$lte: 28}}, {age: true}).forEach(printjson);

排除age的列

db.users.find({age: {$lte: 28}}, {age: false}).forEach(printjson);

9、forEach传递函数显示信息

db.things.find({x:4}).forEach(function(x) {print(tojson(x));});

**Mongodb数据备份恢复**

* |
* 浏览：483
* |
* 更新：2013-10-10 10:14

一、mongodb界面客户端工具

MongoVUE -一个windows下的客户端管理工具

下载地址：http://pan.baidu.com/share/link?shareid=369097&uk=4095176268

MongoHUB - Mac下的 MongoDB 客户端。

Server Density -是一个商业的监控服务提供商

rock\_mongo - PHP写的一个web工具

二、MongoDB文档导入导出操作

1、文档导出

    MongoDB提供了mongoexport工具，可以把一个collection导出成json格式或csv格式的文件。可以指定导出哪些数据项，也可以根据给定的条件导出数据。

-h [ --host ]

-u [ --username ] arg username

-p [ --password ] arg password

-d [ --db ] arg database to use

-c [ --collection ] arg collection to use (some commands)

-f [ --fields ]如 -f name,age

-q [ --query ] arg query filter, as a JSON string

--csv export to csv instead of json

-o [ --out ] arg output file

如下命令：

导出格式分别为：dat、json、csv

       mongoexport -h 127.0.0.1 -d twang -c adservice -o D:/logoadservice.dat

       mongoexport -h 127.0.0.1 -d twang -c adservice -o D:/logoadservice.json

       mongoexport -h 127.0.0.1 -d twang -c adservice -o D:/logoadservice.csv

2、文档导入

【注意：】导入前请删除原有的文件

    mongoimport工具，可以把一个特定格式文件中的内容导入到某张collection中。

-h [ --host ] arg mongo host to connect to ( /s1,s2 for sets)

-u [ --username ] arg username

-p [ --password ] arg password

-d [ --db ] arg database to use

-c [ --collection ] arg collection to use (some commands)

-f [ --fields ]

如下命令：

导出格式分别为：dat、json、csv

       mongoimport -h 127.0.0.1 -d twang -c adservice D:/logoadservice.dat

       mongoimport -h 127.0.0.1 -d twang -c adservice D:/logoadservice.json

       mongoimport -h 127.0.0.1 -d twang -c adservice D:/logoadservice.csv

二、MongoDB数据库导入导出操作

1、导出数据库 twangback为备份的文件夹

命令：

    mongodump -h 127.0.0.1【服务器IP】 -d adview 【数据库名】-o  D:Mongodbdata wangback【自己定义备份存储文件夹】

例如：

    mongodump -h 127.0.0.1 -d twang-o twangback

-h：MongDB所在服务器地址，例如：127.0.0.1，当然也可以指定端口号：127.0.0.1:27017

-d：需要备份的数据库实例，例如：twang

-o：备份的数据存放位置，例如：D:Mongodbdata，当然该目录需要提前建立，在备份完成后，系统自动在data目录下建立一个twangback目录，这个目录里面存放该数据库实例的备份数据。

2、导入数据库从文件夹twangback

【注意：】导入前请删除原有的文件

命令：

    mongorestore -h localhost -d twang【数据库名】

    mondodbback wang wangback wang【备份的路径】

例如：

mongorestore-hlocalhost-dtwangD:Mongodbdata wangback wang

-h：MongoDB所在服务器地址

-d：需要恢复的数据库实例，例如：twang，当然这个名称也可以和备份时候的不一样，比如test2

--directoryperdb：备份数据所在位置，例如：D:Mongodbdata wangback wang

**备份前的检查**  
> show dbs  
MyDB 0.0625GB  
admin (empty)  
bruce 0.0625GB  
local (empty)  
test 0.0625GB  
> use MyDB  
switched to db MyDB  
> db.users.find()  
{ "\_id" : ObjectId("4e290aa39a1945747b28f1ee"), "a" : 1, "b" : 1 }  
{ "\_id" : ObjectId("4e2cd2182a65c81f21566318"), "a" : 3, "b" : 5 }  
>

**整库备份：**  
mongodump -h dbhost -d dbname -o dbdirectory  
-h：MongDB所在服务器地址，例如：127.0.0.1，当然也可以指定端口号：127.0.0.1:27017  
-d：需要备份的数据库实例，例如：test  
-o：备份的数据存放位置，例如：c:\data\dump，当然该目录需要提前建立，在备份完成后，系统自动在dump目录下建立一个test目录，这个目录里面存放该数据库实例的备份数据。

mongodump的官方说明(可通过mongodump --help查看)：

options:

--help produce help message

-v [ --verbose ] be more verbose (include multiple times for more

verbosity e.g. -vvvvv)

--version print the program's version and exit

-h [ --host ] arg mongo host to connect to ( /s1,s2 for

sets)

--port arg server port. Can also use --host hostname:port

--ipv6 enable IPv6 support (disabled by default)

-u [ --username ] arg username

-p [ --password ] arg password

--dbpath arg directly access mongod database files in the given

path, instead of connecting to a mongod server -

needs to lock the data directory, so cannot be used

if a mongod is currently accessing the same path

--directoryperdb if dbpath specified, each db is in a separate

directory

--journal enable journaling

-d [ --db ] arg database to use

-c [ --collection ] arg collection to use (some commands)

-o [ --out ] arg (=dump) output directory or "-" for stdout

-q [ --query ] arg json query

--oplog Use oplog for point-in-time snapshotting

--repair try to recover a crashed database

--forceTableScan force a table scan (do not use $snapshot)

**整库恢复：**  
mongorestore -h dbhost -d dbname –directoryperdb dbdirectory  
-h：MongoDB所在服务器地址  
-d：需要恢复的数据库实例，例如：test，当然这个名称也可以和备份时候的不一样，比如test2  
–directoryperdb：备份数据所在位置，例如：c:\data\dump\test，这里为什么要多加一个test，而不是备份时候的dump，读者自己查看提示吧！  
–drop：恢复的时候，先删除当前数据，然后恢复备份的数据。就是说，恢复后，备份后添加修改的数据都会被删除，慎用哦！

mongorestore的官方说明(可通过mongorestore --help查看)：

options:

--help produce help message

-v [ --verbose ] be more verbose (include multiple times for more

verbosity e.g. -vvvvv)

--version print the program's version and exit

-h [ --host ] arg mongo host to connect to ( /s1,s2 for sets)

--port arg server port. Can also use --host hostname:port

--ipv6 enable IPv6 support (disabled by default)

-u [ --username ] arg username

-p [ --password ] arg password

--dbpath arg directly access mongod database files in the given

path, instead of connecting to a mongod server -

needs to lock the data directory, so cannot be used

if a mongod is currently accessing the same path

--directoryperdb if dbpath specified, each db is in a separate

directory

--journal enable journaling

-d [ --db ] arg database to use

-c [ --collection ] arg collection to use (some commands)

--objcheck validate object before inserting

--filter arg filter to apply before inserting

--drop drop each collection before import

--oplogReplay replay oplog for point-in-time restore

--oplogLimit arg exclude oplog entries newer than provided timestamp

(epoch[:ordinal])

--keepIndexVersion don't upgrade indexes to newest version

--noOptionsRestore don't restore collection options

--noIndexRestore don't restore indexes

--w arg (=1) minimum number of replicas per write

**单个collection备份：**  
mongoexport -h dbhost -d dbname -c collectionname -f collectionKey -o dbdirectory  
-h: MongoDB所在服务器地址  
-d: 需要恢复的数据库实例  
-c: 需要恢复的集合  
-f: 需要导出的字段(省略为所有字段)  
-o: 表示导出的文件名

mongoexport的官方说明(可通过mongoexport --help查看)：

--help produce help message

-v [ --verbose ] be more verbose (include multiple times for more

verbosity e.g. -vvvvv)

--version print the program's version and exit

-h [ --host ] arg mongo host to connect to ( /s1,s2 for

sets)

--port arg server port. Can also use --host hostname:port

--ipv6 enable IPv6 support (disabled by default)

-u [ --username ] arg username

-p [ --password ] arg password

--dbpath arg directly access mongod database files in the given

path, instead of connecting to a mongod server -

needs to lock the data directory, so cannot be used

if a mongod is currently accessing the same path

--directoryperdb if dbpath specified, each db is in a separate

directory

--journal enable journaling

-d [ --db ] arg database to use

-c [ --collection ] arg collection to use (some commands)

-f [ --fields ] arg comma separated list of field names e.g. -f

name,age

--fieldFile arg file with fields names - 1 per line

-q [ --query ] arg query filter, as a JSON string

--csv export to csv instead of json

-o [ --out ] arg output file; if not specified, stdout is used

--jsonArray output to a json array rather than one object per

line

-k [ --slaveOk ] arg (=1) use secondaries for export if available, default

true

--forceTableScan force a table scan (do not use $snapshot)

**单个collection恢复：**  
mongoimport -d dbhost -c collectionname –type csv –headerline –file  
-type: 指明要导入的文件格式  
-headerline: 批明不导入第一行，因为第一行是列名  
-file: 指明要导入的文件路径

mongoimport的官方说明(可通过mongoimport --help查看)：

--help produce help message

-v [ --verbose ] be more verbose (include multiple times for more

verbosity e.g. -vvvvv)

--version print the program's version and exit

-h [ --host ] arg mongo host to connect to ( /s1,s2 for sets)

--port arg server port. Can also use --host hostname:port

--ipv6 enable IPv6 support (disabled by default)

-u [ --username ] arg username

-p [ --password ] arg password

--dbpath arg directly access mongod database files in the given

path, instead of connecting to a mongod server -

needs to lock the data directory, so cannot be used

if a mongod is currently accessing the same path

--directoryperdb if dbpath specified, each db is in a separate

directory

--journal enable journaling

-d [ --db ] arg database to use

-c [ --collection ] arg collection to use (some commands)

-f [ --fields ] arg comma separated list of field names e.g. -f name,age

--fieldFile arg file with fields names - 1 per line

--ignoreBlanks if given, empty fields in csv and tsv will be ignored

--type arg type of file to import. default: json (json,csv,tsv)

--file arg file to import from; if not specified stdin is used

--drop drop collection first

--headerline CSV,TSV only - use first line as headers

--upsert insert or update objects that already exist

--upsertFields arg comma-separated fields for the query part of the

upsert. You should make sure this is indexed

--stopOnError stop importing at first error rather than continuing

--jsonArray load a json array, not one item per line. Currently

limited to 16MB.

**其他导入与导出操作：**

1. mongoimport -d my\_mongodb -c user user.dat  
  
参数说明:  
  
-d 指明使用的库, 本例中为” my\_mongodb”  
  
-c 指明要导出的表, 本例中为”user”  
  
可以看到导入数据的时候会隐式创建表结构

2. mongoexport -d my\_mongodb -c user -o user.dat  
  
参数说明:  
  
-d 指明使用的库, 本例中为” my\_mongodb”  
  
-c 指明要导出的表, 本例中为”user”  
  
-o 指明要导出的文件名, 本例中为”user.dat”  
  
从上面可以看到导出的方式使用的是JSON 的样式.