

MongoDB常用操作命令大全

MongoDB常用操作命令大全

一、数据库常用命令

- 1、Help查看命令提示
- 2、切换/创建数据库
- 3、查询所有数据库
- 4、删除当前使用数据库
- 5、从指定主机上克隆数据库
- 6、从指定的机器上复制指定数据库数据到某个数据库
- 7、修复当前数据库
- 8、查看当前使用的数据库
- 9、显示当前db状态
- 10、当前db版本
- 11、查看当前db的链接机器地址

二、Collection聚集集合

- 1、创建一个聚集集合 (table)
- 2、得到指定名称的聚集集合 (table)
- 3、得到当前db的所有聚集集合
- 4、显示当前db所有聚集索引的状态

三、用户相关

- 1、添加一个用户
- 2、数据库认证安全模式
- 3、显示当前所有用户
- 4、删除用户

四、聚集集合查询

- 1、查询所有记录
- 2、查询去掉后的当前聚集集合中的某列的重复数据
- 3、查询age = 22的记录
- 4、查询age > 22的记录
- 5、查询age < 22的记录
- 6、查询age >= 25的记录
- 7、查询age <= 25的记录
- 8、查询age >= 23 并且 age <= 26
- 9、查询name中包含 mongo的数据
- 10、查询name中以mongo开头的
- 11、查询指定列name、age数据
- 12、查询指定列name, age数据,age > 25
- 13、按照年龄排序
- 14、查询name = zhangsan, age = 22的数据
- 15、查询前5条数据
- 16、查询10条以后的数据
- 17、查询在5-10之间的数据
- 18、or与 查询
- 19、查询第一条数据
- 20、查询某个结果集的记录条数
- 21、按照某列进行排序

五、索引

- 1、创建索引
- 2、查询当前聚集集合所有索引
- 3、查看总索引记录大小
- 4、读取当前集合的所有index信息
- 5、删除指定索引

6、删除所有索引索引

六、修改、添加、删除集合数据

- 1、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
- 3、删除
- 4、查询修改删除

七、语句块操作

- 1、简单Hello World
- 2、将一个对象转换成json
- 3、循环添加数据
- 4、find 游标查询
- 5、forEach迭代循环
- 6、将find游标当数组处理
- 7、将find游标转换成数组
- 8、定制我们自己的查询结果
- 9、forEach传递函数显示信息

八、其他

一. 数据库常用命令

1、Help查看命令提示

```
helpdb.help();  
db.yourColl.help();  
db.yourColl.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();
```

二、Collection聚集集合

1、创建一个聚集集合（table）

```
db.createCollection("collName", {size: 20, capped: 5, max: 100}); //创建成功会显示{"ok":1}
db.collName.isCapped(); //判断集合是否为定容量
```

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();
```

三、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");
```

四、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命令不能带分号，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 distinct 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/});  
相当于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;`
`name`可以用`true`或`false`,当用`true`的情况下`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: select * from userInfo 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;
```

五、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();
```

六、修改、添加、删除集合数据

1、添加的数据的数据列没有固定,根据添加的数据为准

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

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  
  }  
);
```

七、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

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；

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

4、find 游标查询

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());  
}
```

5、forEach迭代循环

`forEach`中必须传递一个函数来处理每条迭代的数据信息

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

6、将find游标当数组处理

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()`；

```
var cursor = db.users.find();  
cursor[4];
```

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

用`toArray`方法将其转换为数组

```
var arr = db.users.find().toArray();  
printjson(arr[2]);
```

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);
```

或

```
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));  
  }  
);
```

八、其他

```
db.getPrevError(); //查询之前的错误信息

db.resetError(); //清除错误记录

db.yourColl.help(); //查看帮助

db.yourColl.count(); //查询当前集合的数据条数

db.userInfo.dataSize(); //查看数据空间大小

db.userInfo.getDB(); //得到当前聚集集合所在的db

db.userInfo.stats(); //得到当前聚集的状态

db.userInfo.totalSize(); //得到聚集集合总大小

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

db.userInfo.getShardVersion(); //Shard版本信息

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

db.userInfo.drop(); //删除当前聚集集合

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
```