

MongoDB 数据库创建删除、表（集合） 创建删除、数据增删改查

主讲教师：（大地）

合作网站：www.itying.com

目录

一、 数据库使用	1
二、 创建数据库	2
三、 插入（增加）数据	3
四、 查找数据	3
四、 修改数据	7
五、 删除数据	7

一、 数据库使用

开启 **mongodb** 服务：要管理数据库，必须先开启服务，开启服务使用 **mongod --dbpath c:\mongodb**

```
C:\Users\Administrator>mongod --dbpath C:\mongodb
2017-07-09T13:23:31.028+0800 I CONTROL [initandlisten] MongoDB starting : pid=13376 port=27017 dbpath=C:\mongodb 64-bit
host=MICROSO-JR3EE06
2017-07-09T13:23:31.032+0800 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2017-07-09T13:23:31.034+0800 I CONTROL [initandlisten] db version v3.4.1
2017-07-09T13:23:31.036+0800 I CONTROL [initandlisten] git version: 5e103c4f5583e2566a45d740225dc250baacfb7
2017-07-09T13:23:31.038+0800 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1t-fips 3 May 2016
2017-07-09T13:23:31.040+0800 I CONTROL [initandlisten] allocator: tcmalloc
2017-07-09T13:23:31.041+0800 I CONTROL [initandlisten] modules: none
2017-07-09T13:23:31.043+0800 I CONTROL [initandlisten] build environment:
2017-07-09T13:23:31.044+0800 I CONTROL [initandlisten] distmod: 2008plus-ssl
2017-07-09T13:23:31.045+0800 I CONTROL [initandlisten] distarch: x86_64
```

管理 **mongodb** 数据库：mongo （一定要在新的 cmd 中输入）

```
C:\Windows\system32\cmd.exe - mongo

C:\Users\Administrator>mongo
MongoDB shell version v3.4.1
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.1
Server has startup warnings:
2017-07-09T13:23:33.826+0800 I CONTROL [initandlisten]
2017-07-09T13:23:33.828+0800 I CONTROL [initandlisten] ** WARNING: Access control
2017-07-09T13:23:33.830+0800 I CONTROL [initandlisten] **      Read and write
restricted.
2017-07-09T13:23:33.833+0800 I CONTROL [initandlisten]
>
```

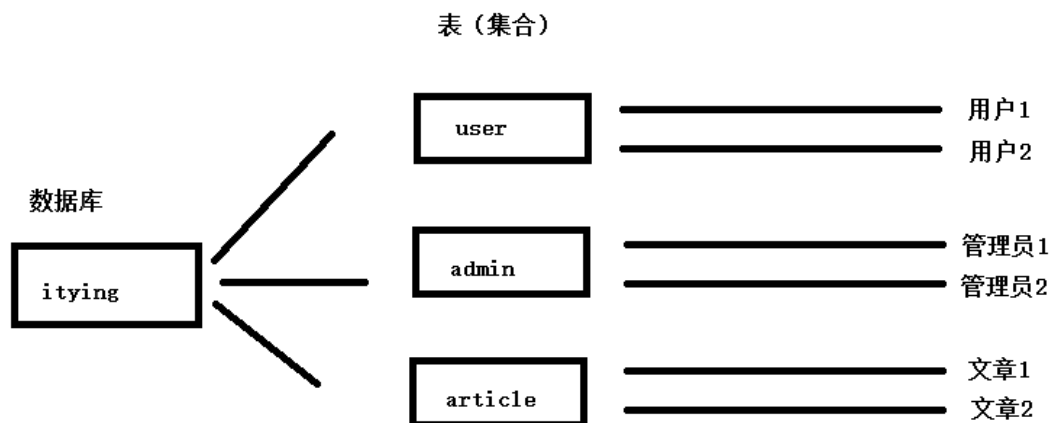
清屏:

```
cls
```

查看所有数据库列表

```
show dbs
```

二、 创建数据库



使用数据库、创建数据库

```
use student
```

如果真的想把这个数据库创建成功，那么必须插入一个数据。

数据库中不能直接插入数据，只能往集合(collections)中插入数据。不需要专门创建集合，只需要写点语法插入数据就会创建集合：

```
db.student.insert({"name":"xiaoming"});
```

db.student 系统发现 student 是一个陌生的集合名字，所以就自动创建了集合。

显示当前的数据集合（mysql 中叫表）

```
show collections
```

删除数据库，删除当前所在的数据库

```
db.dropDatabase();
```

删除集合，删除指定的集合 删除表

```
删除集合 db.COLLECTION_NAME.drop()
```

```
db.user.drop()
```

三、 插入（增加）数据

插入数据，随着数据的插入，数据库创建成功了，集合也创建成功了。

```
db.表名.insert({"name":"zhangsan"}); student 集合名称（表）
```

四、 查找数据

1、查询所有记录

```
db.userInfo.find();
```

相当于：select* from userInfo;

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 注意书写格式

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

9、查询 name 中包含 mongo 的数据 模糊查询用于搜索

```
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 true 的情况下和 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 1 升序 -1 降序

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: `select top 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 (`

`select top 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、findOne 查询第一条数据

```
db.userInfo.findOne();
```

相当于: `select top 1 * from userInfo;`

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

20、查询某个结果集的记录条数 统计数量

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

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

如果要返回限制之后的记录数量, 要使用 `count(true)` 或者 `count(非 0)`

```
db.users.find().skip(10).limit(5).count(true);
```

四、修改数据

修改里面还有查询条件。你要改谁, 要告诉 mongo。

查找名字叫做小明的, 把年龄更改为 16 岁:

```
1 db.student.update({"name":"小明"},{$set:{ "age":16 }});
```

查找数学成绩是 70, 把年龄更改为 33 岁:

```
1 db.student.update({"score.shuxue":70},{$set:{ "age":33 }});
```

更改所有匹配项目: "

By default, the `update()` method updates a single document. To update multiple documents, use the `multi` option in the `update()` method.

```
1 db.student.update({"sex":"男"},{$set:{ "age":33 }},{multi: true});
```

完整替换, 不出现 `$set` 关键字了: 注意

```
1 db.student.update({"name":"小明"},{"name":"大明","age":16});
```

```
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' ;`

五、删除数据

```
db.collectionsNames.remove( { "borough": "Manhattan" } )
```

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

By default, the `remove()` method removes all documents that match the remove condition. Use the `justOne` option to limit the remove operation to only one of the matching documents.

```
db.restaurants.remove( { "borough": "Queens" }, { justOne: true } )
```