

MongoDB 数据库创建、删除、表（集合） 创建删除、数据的增、删、改、查

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

目录

一、 连接数据库.....	1
二、 创建数据库、查看、删除数据库.....	2
三、 插入（增加）数据.....	3
四、 查找数据.....	3
四、 修改数据.....	6
五、 删除数据.....	7

一、 连接数据库

1、连接数据库

```
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]
>
```

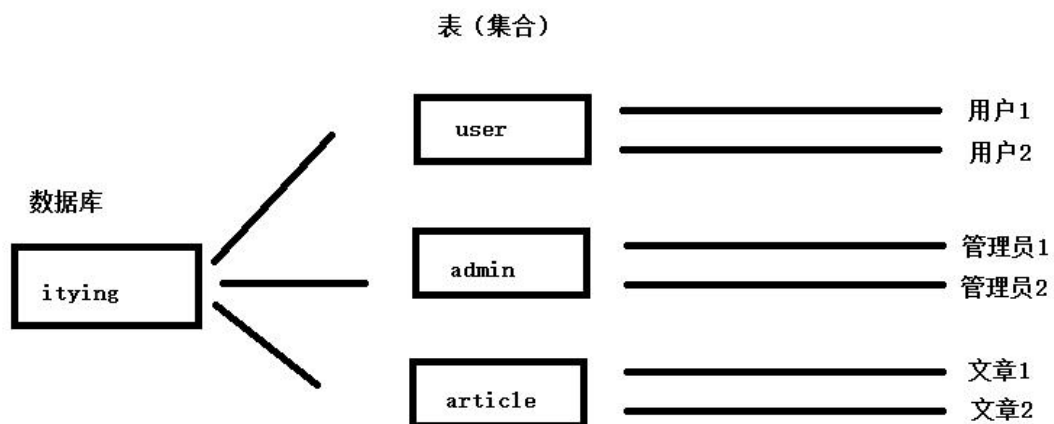
清屏：

```
cls
```

查看所有数据库列表

```
show dbs
```

二、 创建数据库、查看、删除数据库



1、使用数据库、创建数据库

```
use itying
```

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。下面命令表示给 itying 数据库的 user 表中插入数据。

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

2、查看数据库

```
show dbs
```

3、显示当前的数据集合 (mysql 中叫表)

```
show collections
```

4、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()

5、删除数据库，删除当前所在的数据库

db.dropDatabase();

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", "age":20});

四、 查找数据

1、查询所有记录

```
db.user.find();
```

相当于：select* from user;

2、查询去掉后的当前聚集集合中的某列的重复数据

```
db.user.distinct("name");
```

会过滤掉 name 中的相同数据

相当于：select distict name from user;

3、查询 age = 22 的记录

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

相当于： select * from user where age = 22;

4、查询 age > 22 的记录

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

相当于: select * from user where age >22;

5、查询 age < 22 的记录

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

相当于: select * from user where age <22;

6、查询 age >= 25 的记录

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

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

7、查询 age <= 25 的记录

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

8、查询 age >= 23 并且 age <= 26 注意书写格式

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

//相当于%%

select * from user where name like '%mongo%';

10、查询 name 中以 mongo 开头的

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

select * from user where name like 'mongo%';

11、查询指定列 name、age 数据

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

相当于: select name, age from user;

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.user.find({age: {$gt: 25}}, {name: 1, age: 1});
```

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

13、按照年龄排序 1 升序 -1 降序

升序:

```
db.user.find().sort({age: 1});
```

降序:

```
db.user.find().sort({age: -1});
```

14、查询 name = zhangsan, age = 22 的数据

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

相当于: select * from user where name = 'zhangsan' and age = '22' ;

15、查询前 5 条数据

```
db.user.find().limit(5);
```

相当于: select top 5 * from user;

16、查询 10 条以后的数据

```
db.user.find().skip(10);
```

17、查询在 5-10 之间的数据

```
db.user.find().limit(10).skip(5);
```

可用于分页，limit 是 pageSize，skip 是 (page-1)*pageSize

18、or 与 查询

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

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

19、findOne 查询第一条数据

```
db.user.findOne();
```

相当于：select top 1 * from user;

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

20、查询某个结果集的记录条数 统计数量

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

相当于：select count(*) from user 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 } )
```