

# 简单使用

不错的博客

## 目录

- 简单使用
  - 目录
  - 连接数据库
    - 使用
  - 使用数据库
    - 数据库操作
    - 集合操作(表操作)
      - 创建
      - 查看
      - 删除
    - 数据操作
      - 新增数据
      - 查看数据
      - 文档操作(插入数据)
  - 查询数据
    - 关系的表达式

## 连接数据库

```
mongo 127.0.0.1/config
```

## 使用

1. mongo 127.0.0.1:27017/config #连接到任何数据库config
2. mongo --nodb #不连接到任何数据库
3. 启动之后，在需要时运行new Mongo(hostname)命令就可以连接到想要的mongod了：

```
conn=new Mongo('127.0.0.1:27017')
connection to 127.0.0.1:27017
db=conn.getDB('admin')
admin
```

4. help查看帮助

5. mongo时一个简化的JavaScript shell，是可以执行JavaScript脚本的

## 使用数据库

### 数据库操作

```
# 进入test，如果没有就会自动创建
use test
```

```
# dbs 是databases的缩写
# 如果没有数据，那就看不到库
show dbs
```

### 集合操作(表操作)

db相当于是个全局变量，你在哪个库中，db就是谁

```
# 打印当前在那(通过use)
db
```

### 创建

```
# 当我们往表中插入数据，表就自动创建了
db.table1.insert({'name':'psclly', 'age':18})
```

因为没写id，所以它会自动创个id(\_id)

[关于自动创建的id](#)

### 查看

```
show tables
```

### 删除

```
db.table2.drop()    # table2是表名
```

## 数据操作

### 新增数据

```
db.table1.insert({"name":"pscly", "age":18})
```

### 查看数据

```
db.table1.find()
```

### 文档操作(插入数据)

```
# 先定义一个对象(字典), 然后再插入
user1 = {
    "name": "pscly1",
    "age": 18
}
db.table1.insert(user1)
```

如果插入多条

```
db.table1.insertMany([user2,user3])
```

## 查询数据

优点：速度比mysql快

缺陷：因为是非关系型，所以没有那些约束

# SQL: =, !=, >, <, >=, <=

# MongoDB: {key:value}代表什么等于什么,"\$ne","\$gt","\$lt","gte","lte",其中"\$ne"能用于所有数据类

#1、select \* from db1.user where name = "alex";  
db.user.find({'name':'alex'})

#2、select \* from db1.user where name != "alex";  
db.user.find({'name':{'\$ne':'alex'}})

#3、select \* from db1.user where id > 2;  
db.user.find({'\_id':{'\$gt':2}})

#4、select \* from db1.user where id < 3;  
db.user.find({'\_id':{'\$lt':3}})

#5、select \* from db1.user where id >= 2;  
db.user.find({'\_id':{'\$gte':2}})

#6、select \* from db1.user where id <= 2;  
db.user.find({'\_id':{'\$lte':2}})

## 关系的表达式

# 不等于, 大于, 小于  
\$ne \$gt \$lt  
# 大于等于, 小于等于  
\$gte \$lte