mongo 启动

mongo 127.0.0.1:27017 连接远程mongodb

cls 清屏

# 查

show dbs 查看所有数据库

show collections 查看所有集合(mongodb里面不叫集合)

db.user.find() 查找user集合中所有数据

## 条件查询

db.user.find({"age":20}) 查找user集合中age**等于**20的数据

db.user.find({"age":{$gt:20}}) 查找user集合中**大于**20的数据

db.user.find({"age":{$gte:20}}) 查找user集合中**大于等于**20的数据

db.user.find({"age":{$lt:30}}) 查找user集合中**小于**30的数据

db.user.find({"age":{$lte:30}}) 查找user集合中**小于等于**30的数据

db.user.find({"age":{$gt:20,$lt:30}}) 查找user集合中**大于20并且小于30**的数据

查找user集合中name等于张三的数据

db.user.find({"name":"zhangsan"})

查找user集合中name等于张三并且age等于22的数据

db.user.find({"name":"zhangsan","age":22})

db.user.find({$or:[{"age":20},{"age":30}]}) 查询age等于20或age等于3的数据

查询shop集合中title为shop1或shop2的数据

db.shop.find({"title":{$in:["shop1","shop2"]}})

## 模糊查询

db.user.find({"name":/zhang/}) 查找user集合中名字**包含**zhang的数据

db.user.find({"name":/^w/}) 查找user集合中**以w开头**的数据

## 显示特定字段

如：select name from user

db.user.find({},{name:1}) 查找user集合中name列

db.user.find({},{name:1,”\_id”:0}) 不显示\_id字段,只有\_id默认是1

db.user.find({"age":{$gt:20}},{name:1}) 查找user集合中age大于20的name列

## 查询结果排序

db.user.find({}).sort({"age":1}) 按age升序排序

db.user.find({}).sort({"age":-1}) 按age降序排序

## 查询结果限制显示行数

db.user.find({}).limit(2) 只显示前2条数据

db.user.find().skip(2).limit(1) 跳过前两条数据，只显示后一条数据

如果要做分页功能可以这样设计

db.user.find().skip(0).limit(5)

db.user.find().skip(5).limit(5)

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

db.user.findOne() 查找第一条数据

## 统计查询数量

db.user.find().count() 统计查询结果总数

db.user.find({"age":{$gt:24}}).count() 查询age大于24的结果的总数

# 增

将数据存入集合并建好数据库

use test 选择数据库(就算没有这个数据库也没关系)

db.user.insert({"name":"zhangsan","age":20}) 将数据存入集合,mongodb会自动建好集合和库

db.user.insert({"name":"zhangsan","age":20,"sex":"男"}) 字段可以不一致

向shop集合插入100条数据

for(var i=0;i<100;i++){ db.shop.insert({"title":"shop"+i}) }

将json对象插入集合中

person={"name":"aaaaa"}

db.person.insert(person)

save操作

save操作和insert操作区别在于当遇到\_id相同的情况下，save更新数据，insert报错

向person集合中插入一条数据

db.person.save({name:"lisi",age:22})

# 删

db.user.drop() 删除user集合

db.dropDatabase() 删除当前数据库 （当库里的集合全删完后自动删库）

db.user.remove({"age":44}) 删除user集合中age等于44的数据

删除user集合中name等于zhangsan的第一条数据

db.user.remove({"name":"zhangsan"},{justOne:true})

# 改

修改user集合中wangwu为wangwu666

db.user.update({"name":"wangwu"},{$set:{"name":"wangwu666"}})

修改user集合中list的age为44，name为lisi

db.user.update({"name":"list"},{$set:{"age":44,"name":"lisi"}})

注意，如果没加$set关键字会只保留设置的字段（**完整替换**）

db.user.update({"name":"lisi"},{"age":44}) 这样运行后只剩"age":44字段

# 索引

创建索引的命令：

db.user.ensureIndex({"name":1})

创建复合索引

db.user.ensureIndex({"name":1,”age”:-1})

1集合示name键的索引按升序存储，-1集合示降序存储

该索引被创建后，基于name和age的查询将会用到该索引，或者是基于name的查询也会用到该索引，**但是只是基于age的查询将不会用到该复合索引。因此可以说，如果向用到复合索引，必须在查询条件中包含复合索引中的前N个索引列。**如果查询条件中的键值顺序和复合索引中的创建顺序不一致的化，MongoDB可以智能的帮助我们调整该顺序。

获取当前集合的索引：

db.user.getIndexes()

删除索引的命令：

db.user.dropIndex({"name":1})

查看语句执行情况

db.user.find({}).explain("executionStats")

for(var i=0;i<2000000;i++){ db.user.insert({"name":"zh"+i,"score":i%100}) }

db.user.find({"score":99}).explain("executionStats")

没建立索引的查询情况：



db.user.ensureIndex({"score":1})

建立索引后的查询情况：



唯一索引：

db.test.ensureIndex({"userid":1},{"unique":true})

