

Mongodb练习题

一、Insert语句

db.collection.insertOne() 插入一个文档

db.collection.insertMany([]) 插入多个文档

1. 插入一个文档 (test集合)

```
db.test.insertOne({  
  name: "Joe",  
  gender: "m"  
})
```

2. 插入一个文档 (带有嵌入式文档)

```
db.test.insertOne({  
  name: "Kate",  
  birthdate: { "day": 25, "month": 7, "year": 1998 }  
})
```

3. 插入多个文档 (带有嵌入式文档和数组)

```
db.test.insertMany([  
  {  
    name: "Jason",  
    birthdate: { "day": 3, "month": 3, "year": 1996 },  
    hobby: ["basketball", "running", "traveling"]  
  },  
  {  
    name: "Jason",  
    birthdate: { "day": 3, "month": 3, "year": 1996 },  
    hobby: ["basketball", "running", "traveling"]  
  }  
])
```

二、Query查询语句

db.collection.findOne() 查找到一个文档即返回

db.collection.find() 查找所有文档

1. 清空集合，并准备数据

```
db.student.deleteMany({})
```

```
db.student.insertMany([
```

```
  { name: "Joe", gender: "m", age: 23, birthdate: { "day": 15, "month": 3, "year": 1997 }, hobby: ["football", "basketball", "reading"], city: "Beijing", time: [9, 18] },
  { name: "Kate", gender: "f", age: 22, birthdate: { "day": 25, "month": 7, "year": 1998 }, hobby: ["reading", "piano"], city: "Hangzhou", time: [8, 17] },
  { name: "Rose", gender: "f", age: 24, birthdate: { "day": 3, "month": 3, "year": 1996 }, hobby: ["basketball", "running", "traveling"], city: "Shanghai", time: [9, 19] },
  { name: "Jason", gender: "m", age: 21, birthdate: { "day": 17, "month": 12, "year": 1999 }, hobby: ["cooking", "photography"], city: "Chengdu", time: [8, 20] },
  { name: "Grace", gender: "f", age: 22, birthdate: { "day": 10, "month": 6, "year": 1998 }, hobby: ["photography", "cooking", "drama"], city: "Nanjing", time: [9, 18] },
  { name: "Jessica", gender: "f", age: 22, birthdate: { "day": 21, "month": 3, "year": 1998 }, hobby: ["cooking", "piano"], city: "Shanghai", time: [10, 19] }
])
```

2. 查找所有学生

```
db.student.find()
```

db.student.find().pretty() pretty()方法：文档结构化展示

3. 查找所有的女学生

```
db.student.find({ gender: "f" })
```

Output: Kate, Rose, Grace, Jessica

4. 查找城市在上海、杭州、成都之中的学生

```
db.student.find({ city: { $in: ["Shanghai", "Hangzhou", "Chengdu"] } })
```

Output: Kate, Rose, Jason, Jessica

5. 查找年龄小于24岁的女学生

```
db.student.find({ gender: "f", age : { $lt: 24 } })
```

Output: Kate, Grace, Jessica

6. 查找城市在上海或者年龄小于23岁的学生

```
db.student.find({ $or: [ { city: "Shanghai" }, { age: { $lt: 23 } } ] })
```

Output: Kate, Rose, Jason, Grace, Jessica

7. 查找城市在上海或者年龄大于21岁的名字以J为开头的学生

(/[^]J/为正则表达式)

```
db.student.find( {  
  name: /^J/,  
  $or: [ { city: "Shanghai" }, { age: { $gt: 23 } } ]  
})
```

Output: Jessica

8. 查找1998年出生的学生 (嵌入文档的查找)

```
db.student.find({ "birthdate.year": 1998 })
```

不能写作db.student.find({ birthdate: {year: 1998 } })，这个语句是要对文档进行精确匹配，birthdate必须要完全等于{year: 1998}，包括文档的顺序也要相同

Output: Kate, Grace, Jessica

9. 查找喜欢钢琴的学生(数组内的查找)

```
db.student.find( { hobby: "piano" } )
```

Output: Kate, Jessica

10. 查找喜欢摄影和烹饪的学生

```
db.student.find( { hobby: { $all: [ "photography", "cooking" ] } } )
```

注意：如果用db.student.find({ hobby: ["photography", "cooking"] })，意思是要匹配hobby值完全等于["photography", "cooking"]的文档，并且数组里的顺序不能变。用\$all则表示数组内只要包含这些元素即可。

Output: Jason, Grace

11. 查找开始工作时间在9点以及9点之后的学生 (数组元素标号从零开始)

```
db.student.find( { "time.0": { $gte: 9 } } )
```

Output: Joe, Rose, Grace, Jessica

12. 查找time数组里包含小于9并且包含大于18的数字的学生 (如果数组中有一个数字同时满足两个条件的也可以)

```
db.student.find( { time: { $lt: 9, $gt: 18 } } )
```

Output: Jason

13. 查找time数组里包含大于17且小于19的数字的学生

查找数组中的元素，且该元素需要匹配多个条件时用\$elemMatch（即单个元素符合多个条件）

```
db.student.find( { time: { $elemMatch: { $gt: 17, $lt: 19 } } } )
```

Output: Joe, Grace

14. 查找正好有三个爱好的学生

```
db.student.find( {hobby: { $size: 3 } } )
```

Output: Joe, Rose, Grace

15. 准备新数据（嵌入式文档数组的练习）

```
db.tutor.deleteMany({})
```

```
db.tutor.insertMany( [
  { name: "Jack", students:[ { name: "Joe", age: 23 }, { name: "Kate", age: 22 } ] },
  { name: "Max", students:[ { name: "Rose", age: 24 } ] },
  { name: "Jace", students:[ { name: "Joe", age: 23 }, { name: "Jason", age: 21 } ] },
] )
```

16. 查找有学生年龄在22岁以下的导师

```
db.tutor.find( { 'students.age': { $lt: 22 } } )
```

Output: Jace

17. 查找有学生年龄在23岁以下并且该学生名字开头为J的导师

查找数组中的元素，且该元素需要匹配多个条件时用\$elemMatch

```
db.tutor.find( { "students": { $elemMatch: { age: { $lt: 22 }, name: /^J/ } } } )
```

注意下面这个语句是学生数组里有一个人满足两个条件或者两个人分别满足一个条件就会被选择出来

```
db.tutor.find( { "students.age": { $lt: 22 }, "students.name": /^J/ } )
```

为了说明区别，可以插入下面一条记录

```
db.tutor.insertOne({name: "Amy", students: [{name: "Joe", age: 23}, {name: "Katty", age: 21}]})
```

再分别输入两条查询语句查看结果

Output: Jace

18. 查找所有男生的名字（以下用student集合）

db.student.find({ gender: "m" }, { name: 1 }) (默认包含_id)

db.student.find({ gender: "m" }, { name: 1, _id: 0 }) (不包含_id)

19. 查找所有男生除了生日和爱好之外的信息

db.student.find({ gender: "m" }, { birthdate: 0, hobby: 0 })

20. 查找工作时间在9点之后的学生的名字和出生年份

db.student.find (

{ "time.0": { \$gt: 9 } },

{ name: 1, "birthdate.year": 1 }

)

三、Update语句

db.collection.updateOne() 或 db.collection.update() 只更新符合条件的第一个文档

db.collection.updateMany() 更新所有符合条件的文档

1. 将所有男生的城市改为上海（\$set）

db.student.updateMany (

{gender: "m"},

{"\$set": {city: "Shanghai"}}

)

2. 将所有9月之前出生的学生年龄加一岁(\$inc)

db.student.updateMany (

{ "birthdate.month": { \$lt: 9 } },

{ "\$inc": {age: 1} }

)

四、Delete语句

db.collection.deleteOne({}) 删除第一个

db.collection.deleteMany({}) 删除所有

1. 删除所有年龄为23岁的学生

db.student.deleteMany({ age: 23 })

2. 删除出生月份在6月之前的第一个学生

```
db.student.deleteOne( { "birthdate.month": { $lt: 6 } } )
```