1. 复习mongodb的聚合操作命令

1.首先新建数据库，数据库名称为：jobs

2.在jobs中新建一个collection名称为 jobdesc

3.在jobdesc中添加下列数据

jobname workyears tags needpople age

python 3 [python,scrapy,mongodb] 3 20

iOS 3 [iOS,mysql,mongodb] 2 25

java 2 [java, mysql] 1 23

python2 5 [python,scrapy] 5 24

python 2 [scrapy,mongodb] 2 30

1. 使用$group 根据jobname 进行分组，统计每种职位的总数

db.jbdesc.aggregate([{$group:{\_id:'$jobname',conunt:{$sum:1}}}])

2. 使用$group 根据jobname 进行分组，统计每种职位需要的总人数

db.jbdesc.aggregate([{$group:{\_id:'$jobname',needpople:{$sum:'$needpople'}}}])

3. 使用$group将集合中的所有文档分为一组，统计职位总数，算出平均年纪age

db.jbdesc.aggregate([{$group:{\_id:'null',needpople:{$sum:'$needpople'},age:{$avg:'$age'}}}])

4. 使用$project修改文档的结构，只展示jobname和needpople 两个文档字段

db.jbdesc.find({},{\_id:0,jobname:1,needpople:1})

5. 使用$match函数筛选年纪在20～23岁的职位数量包含20和23，并且只返回jobname和age

db.jbdesc.aggregate([{$match:{age:{$gte:20,$lte:23}}},{$project:{\_id:0,jobname:1,age:1}}])

6. 使用$match函数筛选年纪在20～23岁的职位数量包含20和23，并且只返回jobname和age,做降序排列

db.jbdesc.aggregate([{$match:{age:{$gte:20,$lte:23}}},{$project:{\_id:0,jobname:1,age:1}},{$sort:{age:-1}}])

7. 使用$match函数筛选年纪在20～23岁的职位数量包含20和23后，通过jobname分组，并且显示每种职位的平均年龄

db.jbdesc.aggregate([{$match:{age:{$gte:20,$lte:23}}},{$project:{\_id:0,jobname:1,age:1}},{$group:{\_id:'$jobname',age:{$avg:'$age'}}}])

8. 使用limit和skip结合使用，返回集合中第二条开始的后2条数据

db.jbdesc.aggregate([{$skip:2},{$limit:2}])

9. 使用$ｍａｔｃｈ筛选出集合中职位年龄要求在23～30的职位，并根据职位需要的人数做降序排列

db.jbdesc.aggregate([{$match:{age:{$gte:23,$lte:30}}},{$sort:{needpople:-1}}])

10. 使用$unwind 将文档中的tags字段拆分成多条后，返回拆分后的各职位的总数量。

db.jbdesc.aggregate([{$unwind:'$tas'},{$group:{\_id:'null',count:{$sum:1}}}])

a