June 19

1 MongoDB

1.1 数据存储阶段

优点：

数据组织结构化，降低了冗余；

提高了增、删、改、查的效率；

方便扩展；

方便程序调用，做自动化处理；

缺点：

数据库的使用往往需要特定的语句或者命令进行操作，相对比较复杂；

1.2 文件管理阶段（.txt .doc .xls）

优点：

数据可以长期保存；

存储大量的数据；

使用简单；

缺点：

数据的一致性差；

数据的查找修改方便；

数据的冗余性大；

2 数据库的概念

数据：

能够输入到计算机中，并被识别处理的信息的集合

数据结构：

研究一个数据集合中，数据之间关系的学科

数据库：

按照数据结构，存储管理数据的仓库。数据库是在数据库管理系统管理和控制下在一定的节会上数据集合。

数据库管理系统：

数据库管理软件，用于建立维护数据库

数据库系统：

由数据库和数据库管理系统等开发工具组成的集合

3 数据库的分类

3.1 关系型数据库

采用关系模型来组织数据结构的数据库（二维表）

Oracle DB2 SQLServer MySQL SqLite

优点：

容易理解，逻辑类似常见表格；

使用方便，都使用sql语句，sql语句非常成熟；

数据的一致性高，冗余低，完整性好；

技术成熟，可以使用外部连接等复杂操作；

缺点：

每次都需要sql语句的解析，消耗较大；

不能很好的满足并发需求，特别是海量数据的爆发，关系型数据库读写能力差；

关系型数据库每一步都要加锁的操作，以保证操作的原子性，增加了数据库的负担；

数据的一致性有时会导致大量的空间浪费；

3.2 非关系型数据库（Nosql）

not only sql

优点：

高并发，读写能力强；

普遍比关系型数据库容易扩展；

弱化了数据结构，降低了数据的一致性；

缺点：

通用性差，没有sql一样的一致化的操作；

操作灵活，容易混乱；

没有join等操作，有的数据库不支持关系型数据库中类似事物等操作，版本更新快；

3.3 非关系型数据库和关系型数据库的比较

1.不是以关系模型构数据结构，结构比较自由不保证数据的一致性；

2.非关系型数据库弥补了关系型数据库的一些不足，能够在处理高并发，海量数据上体现优势；

3.非关系型数据库的个性化，使其可以在节省空间，提高效率方面发挥作用；

4 Nosql的使用情况

1.数据一致性要求低；

2.数据库处理海量并发存储；

3.数据库大小不容易估算，需要分部扩展

4.给定的数据比较容易建立Nosql模型

5 Nosql的分类

键值型数据库：Redis

列存储数据库

文档型数据库：MongoDB

图形数据库

6 MongoDB数据库（非关系型数据库）

文档型数据库

1.由C++编写的数据库管理系统

2.支持丰富的增删改查操作

3.支持丰富的数据类型

4.支持众多的编程语言接口（python php C++ C#）

5.使用方便，便于部署。在非关系型数据库中属于比较成熟的数据库

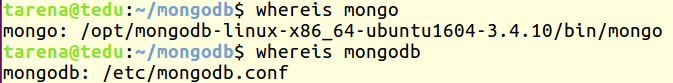
7 MongoDB的安装

自动安装 sudo apt-get install mongdb

默认安装位置 /var/lib/mongodb

配置文件位置 /var/mongodb.conf

cookie 查询配置文件位置



命令集/usr/bin /usr/local/bin

手动安装：

1.下载MongoDB（开源）

www.mongodb.com----》get MongoDB----》community server选择想要的版本下载

2.选择合适的位置解压（/usr/local）

tar解压后得到MongoDB文件夹

3.将Mongodb文件夹中的bin文件夹变为环境变量

PATH = $PATH:/opt/mong…/bin

export PATH

将以上两句写入/etc/re.local

4.重启系统

8 MongoDB命令

设置数据库的存储位置

mongod --dbpath 目录

eg：mongod --dbpath dbs

设置数据库端口

mongod --port 8080

\*如果不设置默认为27017

mongo

进入Mongo shell界面：MongoDB的交互界面，用来通过mongo语句操作

组织结构：键值对--》文档--》集合--》数据库

ID | name | age

-------------------------------------------------------------

1 | Lily | 17

--------------------------------------------------------------

2 | Lucy | 18

--------------------------------------------------------------

{

‘\_id’:Objectid(‘asdfsadfasdf’),’name’:’Lily’,’age’:17

},q{

‘\_id’:Objectid(‘asdfsadfasdf’),’name’:’Lucy’,’age’:18

}

8.1 mysql和MongoDB概念比较

|  |  |  |
| --- | --- | --- |
| MySQL | MongoDB | 含义 |
| database | database | 数据库 |
| table | collection | 表/集合 |
| colunn | field | 字段/域 |
| row | document | 记录/文档 |
| index | index | 索引 |

系统数据库：

admin：存储用户及其权限

local：存储本地数据

config：存储分片信息的

数据库的命令规则：

1.使用utf-8字符

2.不能含有空格，点 / \ ‘\0’字符

3.长度不能超过64字节

4.不能和系统数据库重名

5.习惯使用小写字母，表达数据库功能

db：mongo系统全局变量，代表你当前正在使用的数据库

\*当不用use选择任何数据库时，db表示test。此时插入数据则创建test数据库

集合的命名规则

1.使用utf-8字符串

2.不能有‘\0’

3.不能以system.开头，因为这是系统的保留前缀

4.不能和关键字重复

8.2 MongoD命令

创建数据库

use databsesName

创建一个名字为stu的数据库

eg:use stu

\*use实际上是表示选择使用哪个数据库，如果这个数据库不存在，则表示同时创建这个数据库

\*使用use后数据库不会被马上创建，而是需要写入数据时才会创建

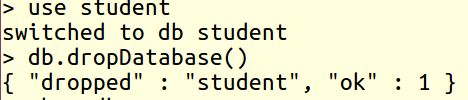
查看当前系统下的数据库

show dbs

删除数据库

db.dropDatabase()

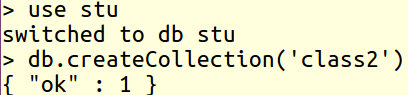
删除db当前所代表的数据库



创建集合

db.createCollection(collection\_name)

创建一个集合名字为class2



创建集合2

当向一个集合中插入数据时，如果这个集合不存在则会自动创建

db.collectionName.insert(…)



删除集合

db.collectionName.drop()



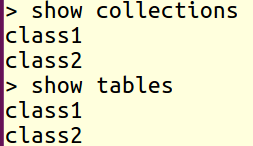
集合重命名

db.collectionName.renameCollection(‘new\_name’)



查看集合

show collections/show tables



8.3 MongoD的备份和恢复

备份：mongodump -h dbhost -d dbname -o dbdir



将‘127.0.0.1’主机上stu数据库备份入bak文件夹

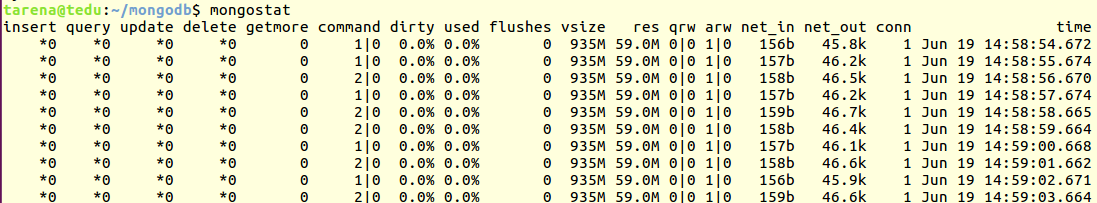
恢复：mongorestore -h dbhost:port -d dbname path(备份文件夹的位置)



将bak文件夹下备份的stu数据库恢复到本机的student数据库中。student不存在则会自动创建

8.4 数据库的监测

mongostat



|  |  |
| --- | --- |
| insert query update delete | 每秒增删改查的次数 |
| command | 每秒运行命令次数 |
| flushes | 每秒和磁盘交互的次数 |
| vsize | 使用虚拟内存 |

mongotop

监测每个数据库的读写时长

ns total read write

数据集合 总时长 读时长 写时长

9 文档

MongoDB中数据的组织形式---》文档

mongo文档以键值对的形式组成的类似于字典的数据描述形式

键：即文档的域

键的命名规则：

1.utf-8格式字符串

2.不使用‘\0’通常不会使用‘.’和‘$’

3.一个文档中的键不可以重复

\*文档中的键值对是有序的

\*MongoDB严格的区分大小写

值：即文档存储的数据 支持bson类型

|  |  |
| --- | --- |
| 类型 | 值 |
| 整型 | 整数 |
| 布尔类型 | True False |
| 浮点型 | 小数 |
| Array | 数组 |
| Timestamp | 时间戳 |
| Date | 时间日期 |
| Object | 内部文档 |
| Null | 空值null |
| Symbol | 特殊字符 |
| String | 字符串 |
| Binary data | 二进制字符串 |
| code | 代码 |
| regex | 正则表达式 |
| Objectid | Objectid子串 |

ObjectId

{"\_id" : ObjectId("5b28a031a9d1d286af40aed5")}

\_id：当在MongoDB中插入文档时，如果不指定\_id则会自动添加这个域。值是一个ObjectId类型数据

24位16进制数-----》用来保证\_id值的唯一性

8位文档创建时间6位机器id 4位进程id 6位计数器

集合中的文档

1.集合中的文档不一定都有相同的域

2.集合中文档域的个数也不一定相同

集合的设计

1.集合中的文档尽可能描述同一数据

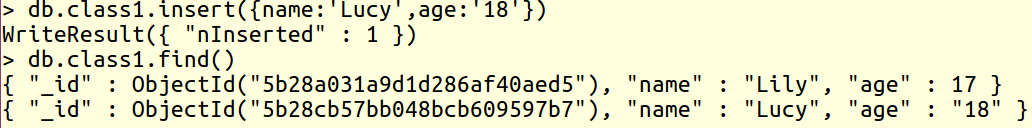
2.同一类数据不要过多的分散集合存放

3.集合中文档的层次不要包含太多

9.1 插入文档

db.collectionName.insert()

插入单个文档



\*插入操作时，键可以不加引号

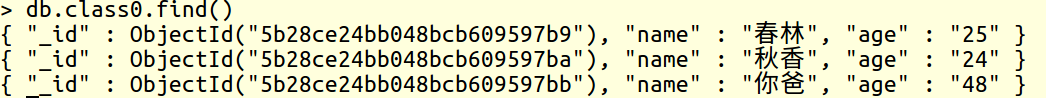
\*查看插入结果db.class1.find()

\*\_id为系统自动添加主键，如果自己写\_id则会使用自己的值，但是仍然不能重复

插入多个文档

db.collectionName.insert([{},{},{}])





save插入文档

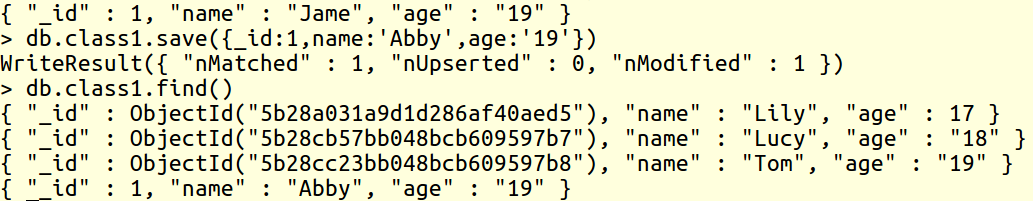
db.collectionName.save()



save和insert的区别：

\*如果不加\_id选项时save和insert相同

\*如果加\_id选项，则此\_id值存在save表示修改该文档内容，如果不存在则正常插入



作业：

1.关系型数据库和非关系型数据库有什么区别？

2.复习mysql增删改查操作

3.练习mongo数据库的创建、删除，集合的创建、删除，文档插入

June 20

1 获取集合对象

1.1 数据存储阶段



1.2 查找操作

mysql:select……from table where ……

mongo:db.collectionName.find(query,field)

查找所有内容

db.collectionName.find()--------->select \* from table

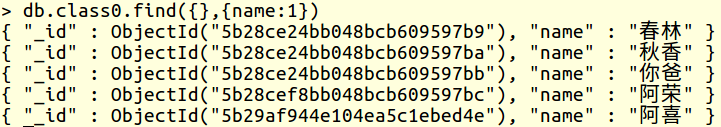
|  |  |
| --- | --- |
| find(query,field) | |
| 功能 | 查找数据 |
| 参数 | query 筛选条件，相当于where  field 选定要展示的域 |
| 返回值 | 返回查找到的文档 |

query以键值对的形式给出筛选条件

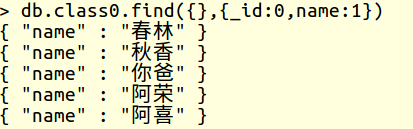


filed以键值对的形式给出要展示（不展示）的域；

域名为键，值为1则表示展示，0表示不展示。



除了id之外，0和1不能够同时出现



\*如果某个域设置为0则表示不展示该域，其他域均显示；

\*如果某个域设置为1则表示展示该域，其他域均不显示；

\*如果不写field参数则表示所有内容都显示。

只显示姓名为Abby，和年龄其他不显示



只显示年龄为18岁的，不显示id



|  |  |
| --- | --- |
| findOne(query,field) | |
| 功能 | 查找符合条件的第一条文档数据 |
| 参数 | query 筛选条件，相当于where  field 选定要展示的域 |
| 返回值 | 返回查找到的文档 |

2 query更多的筛选功能

操作符：使用$符号注明的特殊子串表达特殊含义

例如：$lt表示小于

比较操作符

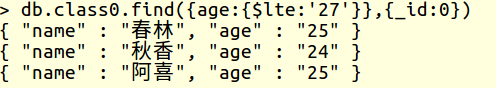
$eq 等于



$lt 小于



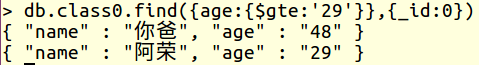
$lte 小于等于



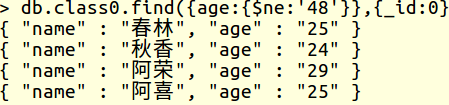
$gt 大于



$gte 大于等于

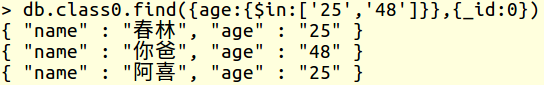


$ne 不等于

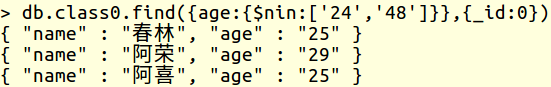


\*如果一个文档不存在某个域，则业认为是不等于

$in 包含



$nin 不包含



年龄大于25小于48之间的人，不包含ID号



3 逻辑操作符

and

1.query多个条件为并列关系



2.$and 逻辑与



查找，年龄等于19姓名小于Tom不带id号的



$or 逻辑或

年龄等于25的或者名字叫阿喜



$not 逻辑非

$nor 既不也不

年龄既不是17岁也不是男性

db.class1.find({$nor:[{age:17},{sex:’m’}]},{\_id:0})

年龄大于24并且叫阿喜 或者 叫阿荣



4 数组值

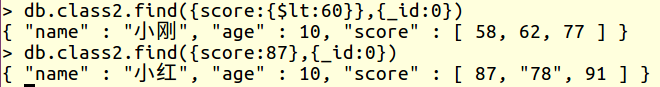
[1,2,3,4]

\*数值类型可以混合

\*是有序的

4.1 数组的查找

查看数组中，是否包含某一项



\*只要数组中有任意一项符合查找条件即可被查找

$all

查找数组中同时包含多项的文档

查找数组中同时包含65 88的文档



$size

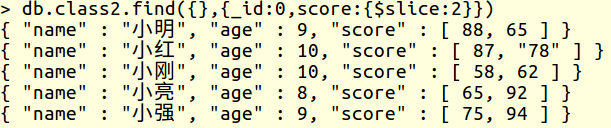
通过数组中元素的个数查找



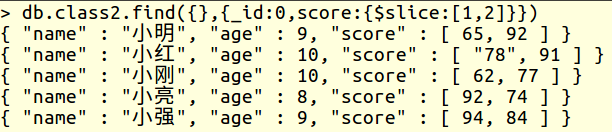
$slice

取数组的部分进行显示，放在field中

显示数组中的前几项



跳过第一项，显示后两项



$exists

判断一个域是否存在

查找存在sex域的文档（如果是False则表示查找无这个域的文档）



$mod

余数查找

查找年龄为单数的文档

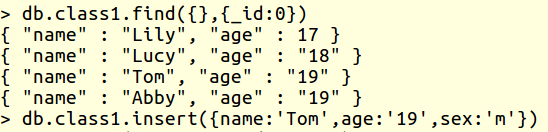


除2余1

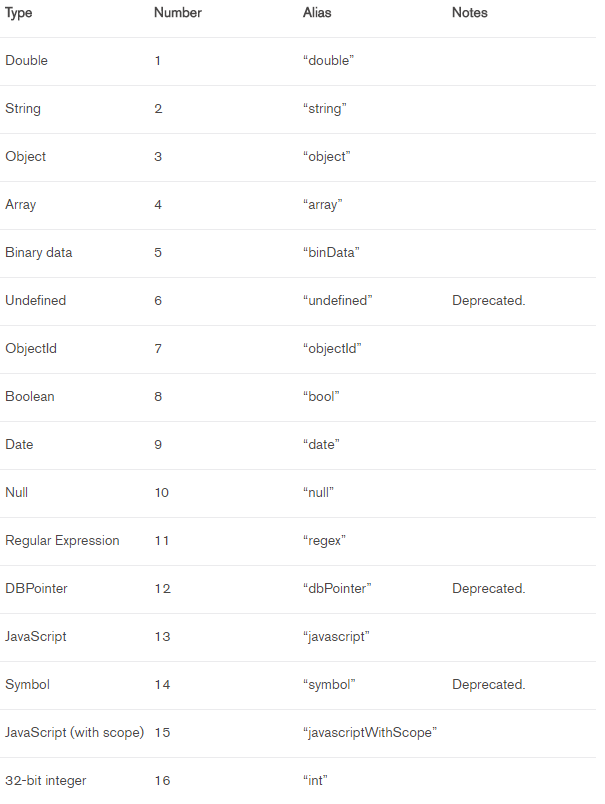
$type

查找指定数据类型的文档

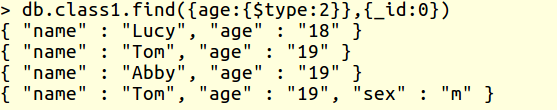
查找类型为整数的











age的值为字符串的

\*如果对数据进行查找则表示查找数组值对应的类型

5 查找结果处理函数

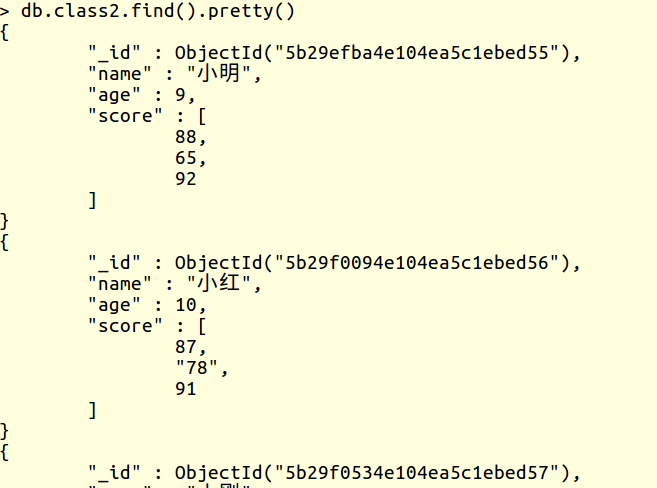
distinct()

功能：查看一个集合中某个域的值的范围



pretty()

功能：将查询结果格式化显示



limit(n)

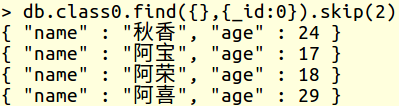
功能：显示查找结果的前几条

显示查找结果的前两条



skip(n)

功能：显示查找结果跳过前n条



count()

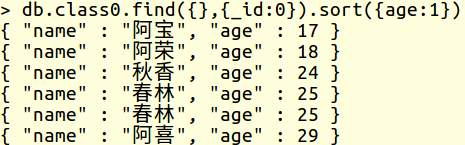
功能：查找结果计数



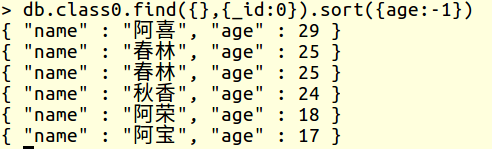
sort()

功能：对查找结果排序显示

按年龄升序

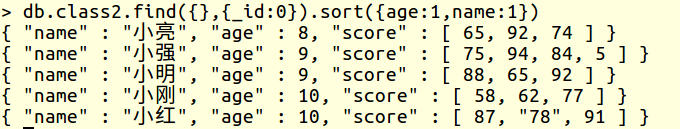


按年龄降序

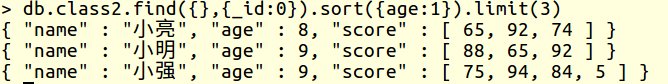


复合排序

按年龄升序排序，当年龄相同按姓名升序排序



显示出年龄最小的三个学生



6 文档删除操作

mysql：delete from table where ….

MongoDB：db.collectionName.remove(query,justOne)

功能：删除文档

参数：query筛选要删除的文档，相当于where

justOne布尔值，默认为false，表示删除所有符合条件的文档。如果赋值为true则只删除第一条符合条件的文档

删除有sex域的文档



只删除第一条年龄为17的文档



删除集合中所有文档

db.collectionName.remove({})

删除class0里所有的文档



练习：

1.创建数据库名字grade

2.数据库中创建集合名字class

3.集合中插入若干文档，文档格式如下

{name:’zhang’,age:10,sex:’m’,hobby:[‘a’,’b’]}

hobby可以是:draw sing dance baskerball football pingpong computer

4.查找练习

查看班级所有人信息

查看班级中年龄为8岁的学生信息

查看年龄大于10岁的学生信息

查看年龄在8-11岁之间的学生信息

找到年龄为9岁且为男生的学生

找到年龄小于7岁或者大于11岁的学生

找到年龄是8岁或者11岁的学生

找到有两项兴趣爱好的学生

找到兴趣爱好中有draw的学生

找到喜欢画画和跳舞的学生

统计兴趣有三项的学生人数

找出本班年龄第二大的学生

查看学生的兴趣范围

找到班级中年龄最大的三个学生

5.删除所有年龄大于12或者小于6岁的学生

June 21

1 修改数据

mysql： updata table set … where…

mongodb: db.collection.update(query,update,upset,multi)

功能:修改文档

参数:

|  |  |
| --- | --- |
| query | 筛选要修改的文档，相当于where子句，用法同查找 |
| update | 将筛选的文档修改什么内容，相当于set。需要配合修改操作符使用 |
| upsert | bool值默认为false表示如果query不存在则无法修改  true表示如果筛选的文档不存在则根据修改内容插入文档 |
| multi | bool值默认为false表示如果有多条文档符合筛选条件，只修改第一条  true表示修改所有 |

将Lucy的年龄设置为20



修改Lily的年龄（插入一条）



将所有年龄为20的人的年龄修改为17



1.1 修改操作符

$set

功能：修改一个域的值，或者增加一个域

给每个人都增加性别



可以修改多个值



$unset

功能：删除一个域

删除Abby的sex域



$rename

功能：修改域名

将sex域名修改为gender



$setOnInsert

功能：如果第三个参数为true，插入新的文档作为插入补充内容

插入姓名为Jame，年龄为18，插入电话和性别



如果掺入了新的文档则将setOnInsert中的内容也插入到文档中

$inc

功能：加减修改器

\*操作可以是正数、负数、小数

给Abby的年龄加1



$mul

功能：乘法修改器

\*操作可以是正数、负数、小数

Jam的年龄乘以2



$min

功能：设置值的下限。如果筛选的文档指定的值小于min则不修改；大于则修改为min值

将女性未满19岁的修改为19岁



$max

功能：设置值的上限，如果筛选到的文档指定域的值大于max的值不变，小于max的值修改为max值

1.2 数组修改器

$push

功能：向数组中添加一项





$pushAll

功能：向数组中添加多项



$pull

功能：从数组中删除一项



$pullAll

功能：从数组中删除多项



$each

功能：对[]内的每一项操作



$position

功能：指定插入位置（需要搭配each使用，插入到指定位置）

将小红的78分插入到87分后





$sort

功能：对数组进行排序

将小明的成绩不再添加，进行升序排序



$pop

功能：弹出一项

弹出第一项

-1第一项，1为最后一项







$addToSet

功能：向数组中添加一项，不能和其他的项重复



2 时间类型

mongo中存储时间格式：ISODate()

方法1：

new Date()自动生成当前时间





方法2：

ISODate()生成当前时间



方法3：

Date()将当前生成的时间变为字符串（系统时间）



ISODate()

功能：生成mongo时间存储类型

参数：参数为指定时间

‘2018-1-1 12:10:34’或‘20180621’



2.1 时间戳

valueof()

功能：将当前时间生成为时间戳





3 null

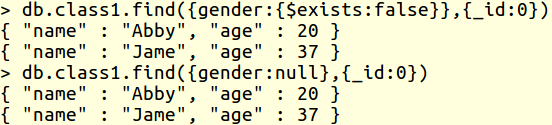
1.如果某个域存在却没有值，可以设置为null





2.某个域不存在，可以使用null来匹配

当gender域不存在时，可以找到这个文档

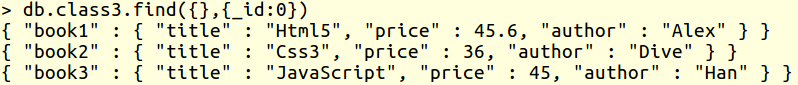


4 Object内部文档

内部文档：

文档内某个域的值还是一个文档，则这个文档为内部文档

当需要使用内部文档某个域时，可以使用外部文档，引用内部文档的方式。但是注意在使用时需要加引号





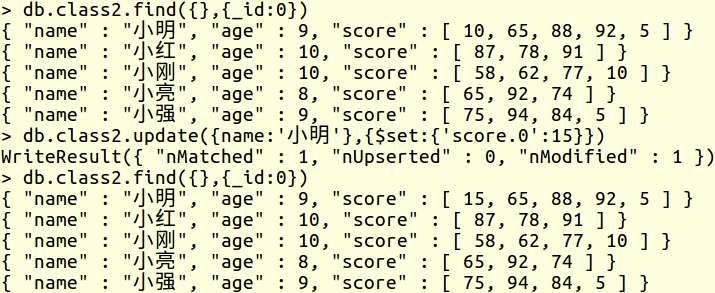
将book3的售价修改为38.8



5 数组的下标

使用数组时，可以使用数组域 . 数组序列下标的方式使用数组中的具体某一元素

以下标的方式修改数组的数据

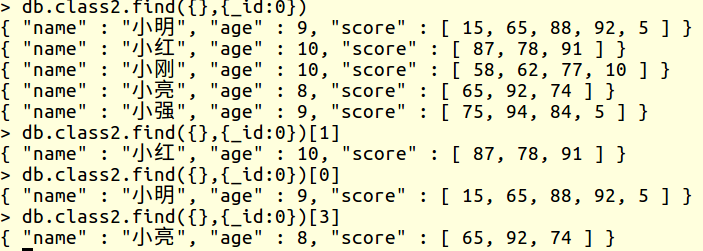


查找score第一项是15的文档



6 查找结果的有序性

既可以对find查找结果使用[ ]引用序列下标的方式获取查找结果的第几项



练习：

使用之前的grade数据库

1.将年龄修改为8岁，兴趣爱好变为，跳舞 画画

2.追加xx的兴趣爱好，唱歌

3.xx兴趣爱好多了 吹牛，打篮球

4.xx兴趣多了 跑步和唱歌，但是要确保不和之前的兴趣重复

5.该班所有成员年龄加1

6.删除小明的sex属性

7.删除小李的兴趣中的第一项

8.删除小红兴趣中画画和唱歌

7 索引

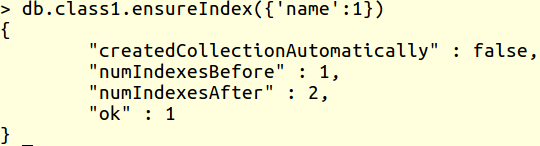
指建立指定键值及所在文档中存储的位置的对照清单。是用索引可以方便我们进行快速查找，减少遍历次数，提高查找效率。

7.1 创建索引

ensureIndex()

功能：创建索引

参数：索引域，索引类别，索引选项

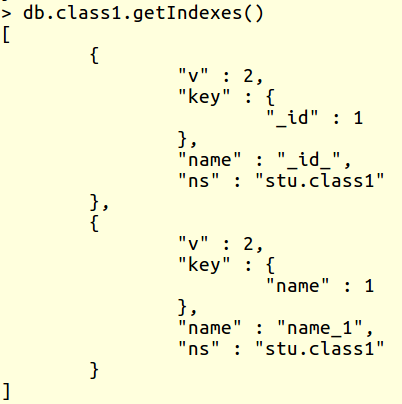


\*1表示正向索引

\*-1表示逆向索引

7.2 查看集合索引

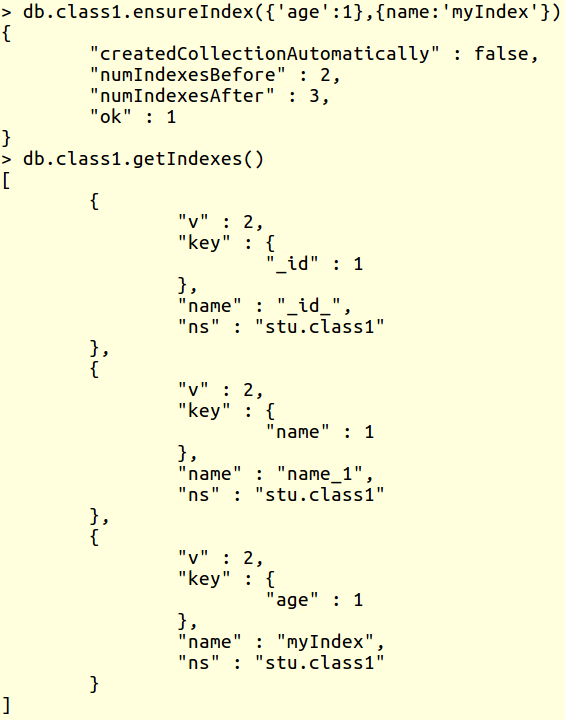
db.class1.getIndexes()



\*\_id域是由系统自动为集合创建的索引

自定义索引名称

通过第二个参数中name选项

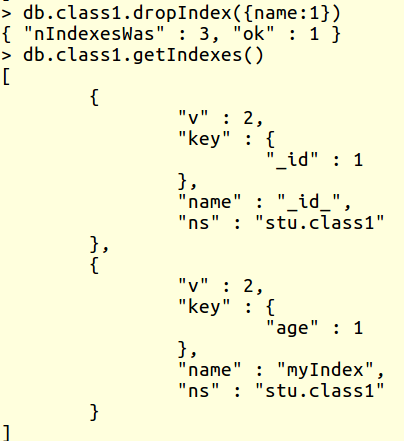


7.3 删除索引

dropIndex()

功能：删除索引

参数：要删除的索引，可以使索引名或者索引键值对

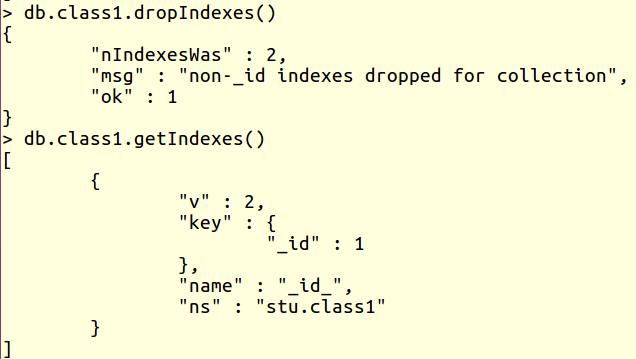


也可以通过名称删除



dropIndexes()

功能：删除所有索引

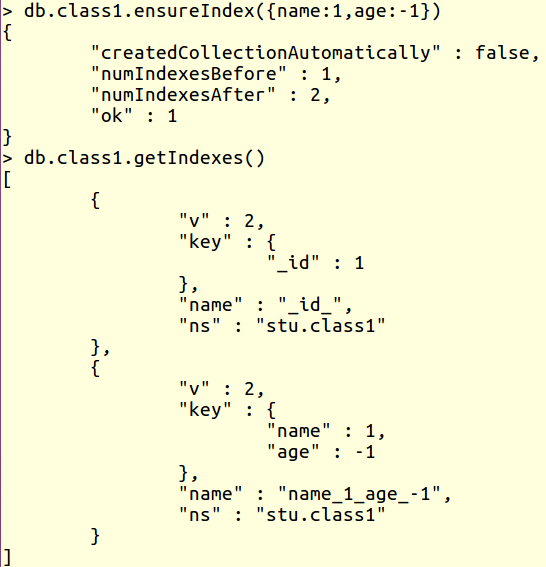


删除所有索引但是不会删除\_id索引

7.4索引类型

符合索引

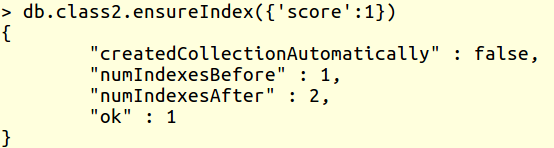
根据多个域创建一个索引



7.5 数组和子文档索引

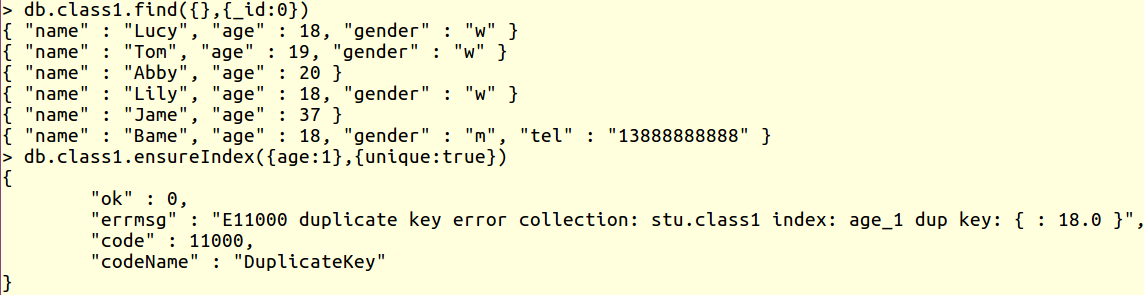
如果对某个域的值为数组或者子文档的域创建索引，则通过数组或者子文档进行查找时也是索引查找

如果对‘score’创建了索引，则该查找也为索引查找



7.6 唯一索引

创建索引时，希望集合中创建索引的域的值均布重复



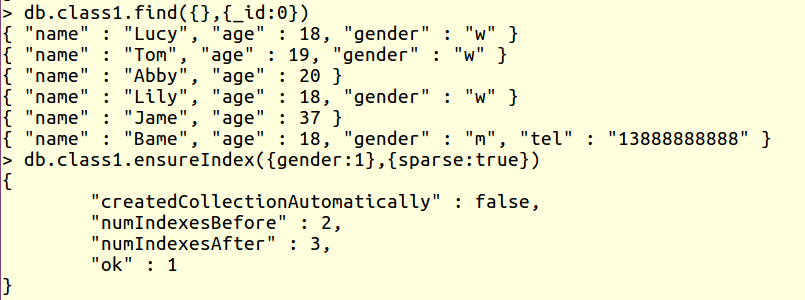
由于age中有相同的值，所以不能创建唯一索引

因此，可以给name创建唯一索引

\*当对某个域创建了唯一索引，则无法插入重复值

7.6 稀疏索引

只针对有指定id域的文档创建索引表，如果某个文档没有该域则不会在索引表中插入内容



eg：对没有gender域的文档不做索引处理

7.7 索引约束

1.当数据发生更新，索引表也要随之更新，影响插入，删除修改操作效率

2.索引表也需要占有一定的磁盘空间

综上：

1.当对数据库进行大量的查找操作，而不是修改操作，适合创建索引。

2.集合中，文档的数据量较大的时候，适合创建索引。

June 23

1 聚合操作

对文档的筛选结果进行整理统计（更高级的查找）

db.collection.aggregate()

功能：聚合函数，完成聚合操作

参数：聚合条件---》聚合操作符

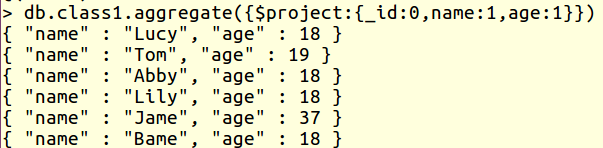
1.1 聚合操作符

$group分组聚合

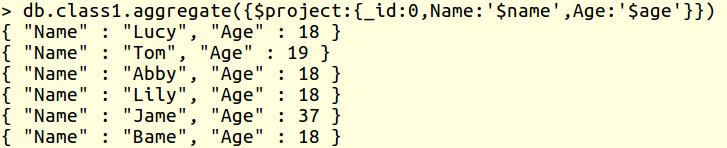
需要配合具体的分组操作符使用

$project修饰文档的显示效果

显示age和name



显示自定义的域名

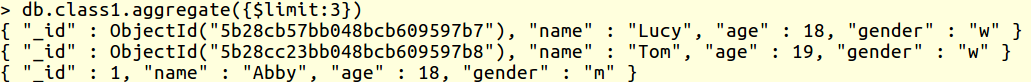


$match过滤数据

只显示年龄大于18岁的人



显示前三条文档

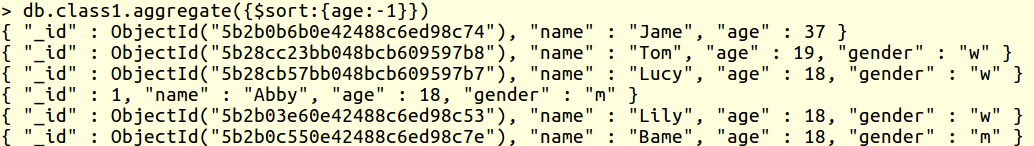


$skip跳过几条文档



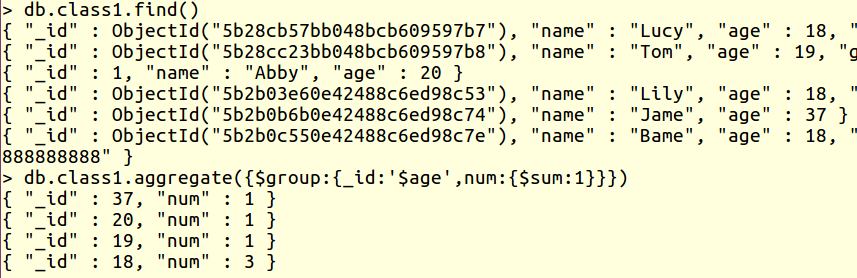
$sort排序

按年龄降序排序

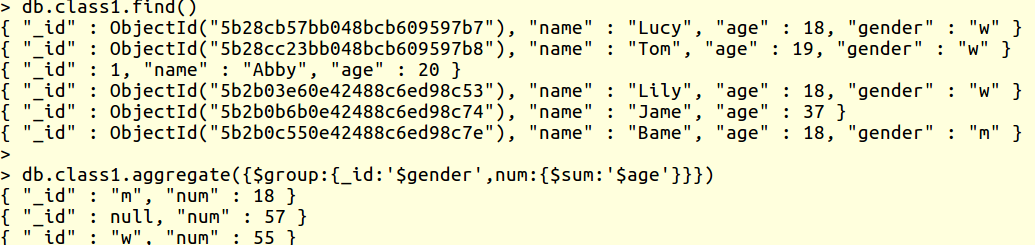


1.2 分组操作符

$sum求和

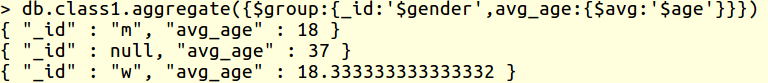


按照gender分组，给分组后取名为num，sum为按照age相加



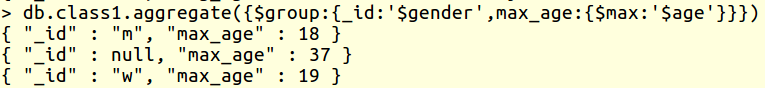
$avg求平均数

求每个性别的平均年龄



$max求最大值

统计每个性别中的最大值



$min求最小值

2 聚合管道

将前一个聚合操作得到的结果，给后一个聚合操作继续使用

db,collectionNmae.aggregate([聚合1，聚合2……])





找出重名的人



3 固定集合

MongoDB中可以创建大小固定的集合，称之为固定集合。

特点：

插入速度快，顺序查找速度快

能够淘汰早期的数据

可控制集合的空间大小

使用：

临时缓存

日志处理

创建：

db.createCollection(collectionName,{capped:true,size:10000,max:10000})

|  |  |
| --- | --- |
| capped:true | 表示创建的集合是固定集合 |
| size | 表示集合的大小（kb） |
| max | 表示最多存放的文档个数 |



当继续插入超过设置最大文档个数时，第一条插入的就会被删除

4 文件存储

文件存储数据库的方式

1.存储路径

将文件在本地的路径以字符串的形式存储到数据库

优点：节省数据空间

缺点：当数据库或者文件位置发生变化时，需要修改数据库内容

2.存储文件本身

将文件以二进制（数据库指定编码）的形式存入到数据库中

优点：数据库在，文件就在，随着数据库的迁移，不会丢失

缺点：占用数据库的空间大；文件较大时，由于数据库都是结构化的存储数据，所以对大文件处理效率较低

MongoDB存储大文件的方案：GridFS存储大文件（超过16MB的文件）

GridFS方案解释

4.1 GridFS方案解释

在MongoDB一个数据库中创建两个集合共同完成文件的存储

fs.files:存储文件的相关信息（文件名，文件类型）

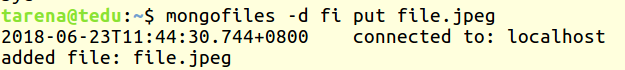
fs.chunks:分块存储文件的实际内容

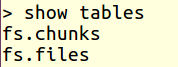
4.2 如何存储

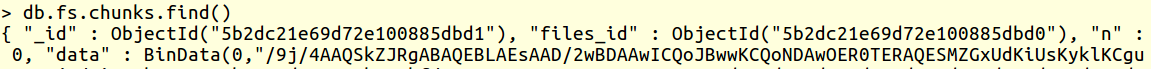
mongofiles -d dbname put file

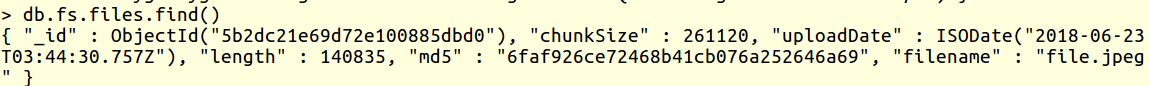
数据库 要存储的文件

\*如果数据库不存在，则会自动创建









\*对于同一个文件fs.files的\_id值等于fs.chunks中的files\_id域的值

提取文件

mongofiles -d dbname get file



Grid的优缺点

优点：存储方便，提供了较好的存储命令，方便移植

缺点：读写效率低下（可能慢于硬盘的速度）

5 游标

通过获取操作数据库的返回结果，得到返回结果的对象。通过该游标对象科技进一步得到数据库的内容。

var cursor = db.class1.find()

cursor.next()

cursor.hanNext()

通过python操作MongoDB

5.1 cursor属性函数

next()

limit()

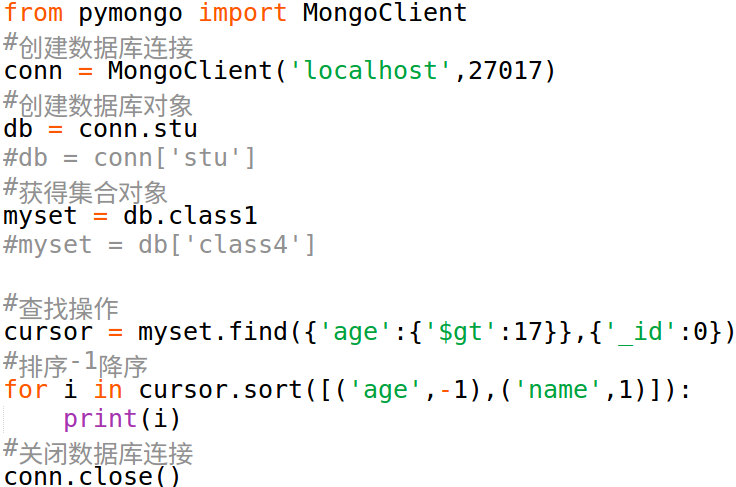
skip()

count()

sort()

pymongo：cursor.sort([('age',-1),('name',1)])

mongoshell：sort({‘age’:-1,’name’:1})



\*当游标使用了next或for取值后就不能在进行limit，skip或者sort操作了

pymongo模块 第三方模块

安装 sudo pip3 install pymongo

操作步骤

1.创建mongo数据库连接对象

conn = pymongo.MongoClient(‘localhost’,’监听的端口号(27017)’)

2.生成要操作的数据库对象

db = conn.stu或者db = conn[‘stu’]

3.获取集合对象

myset = db.class0 或 myset = db[‘class0’]

4.通过集合对象操作MongoDB数据库

增、删、改、查、索引、聚合文件操作

5.关闭数据库连接

conn.close()



5.2 插入数据

insert()

参数用法同mongoshell中的insert

insert\_many()

insert\_one()

save()



5.3 查找操作

find()

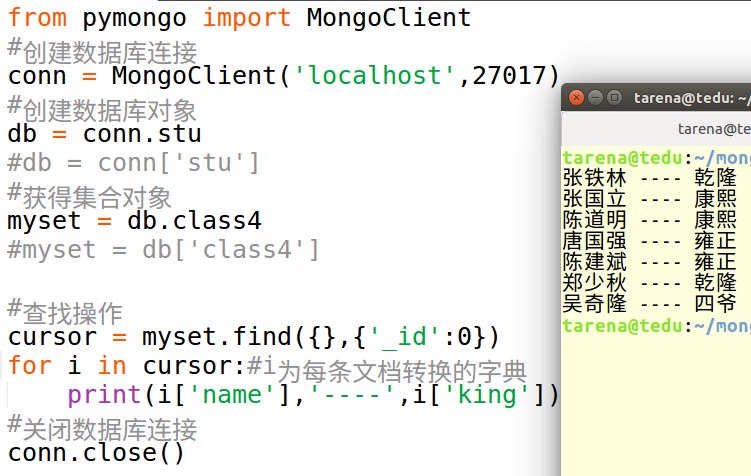
功能：查找数据库中的内容

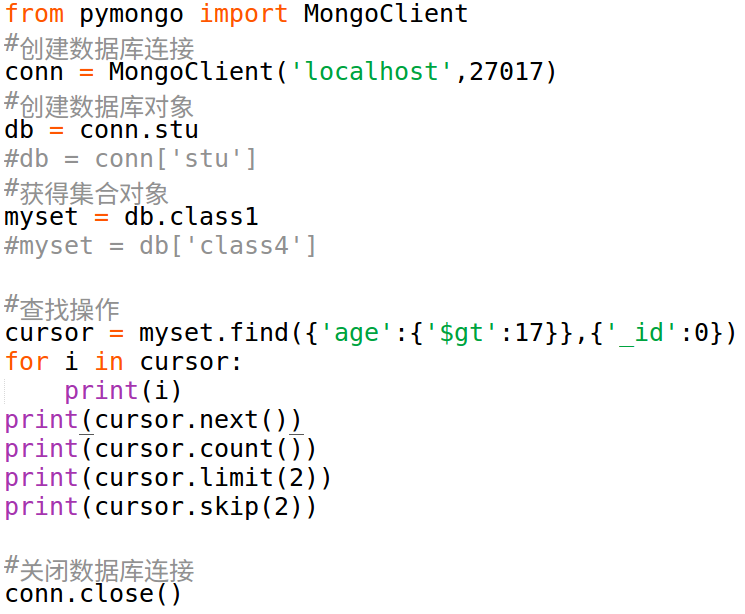
参数：用mongo shell find()

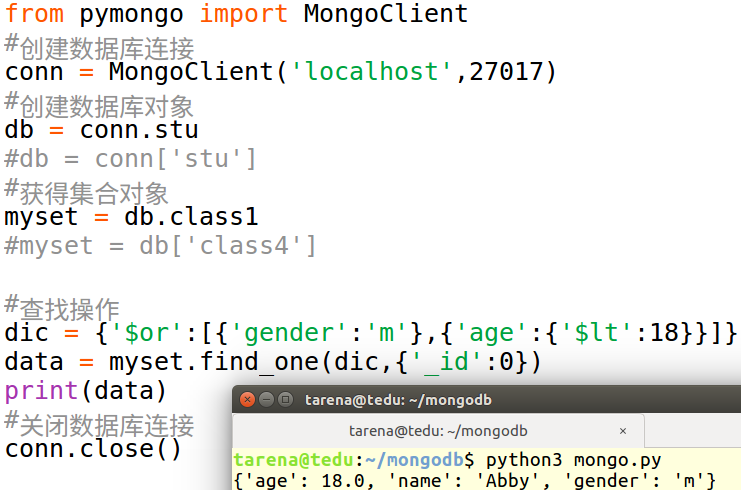
返回值：返回一个结果游标

\*在pymongo中使用操作符的方法和在mongoshell中一样，只需要加引号以字符串的方式给出

find\_one()







find\_one()返回一个字典

5.4 修改操作

update(query,update,upset,multi)

update\_many()

update\_one()

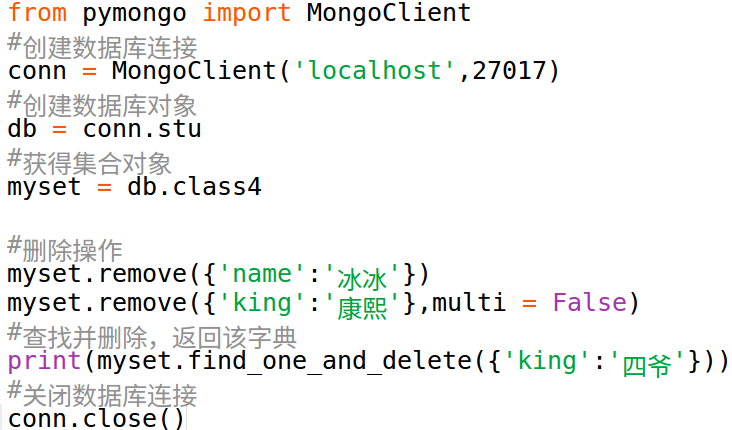


5.5 删除操作

remove(query,multi = True)

multi默认为True表示删除所有符合条件的数据

设置为False表示只删除一条



6 索引操作

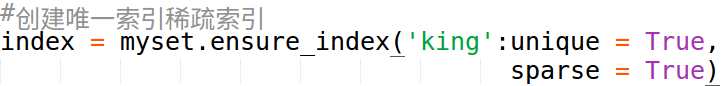
索引

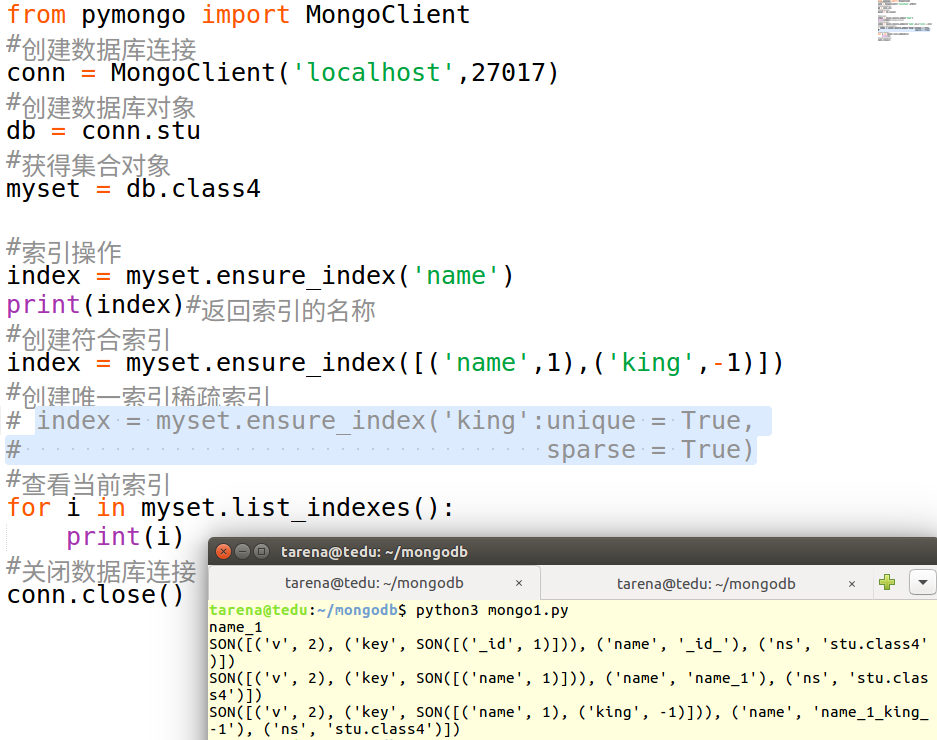
ensure\_index()

list\_indexes()



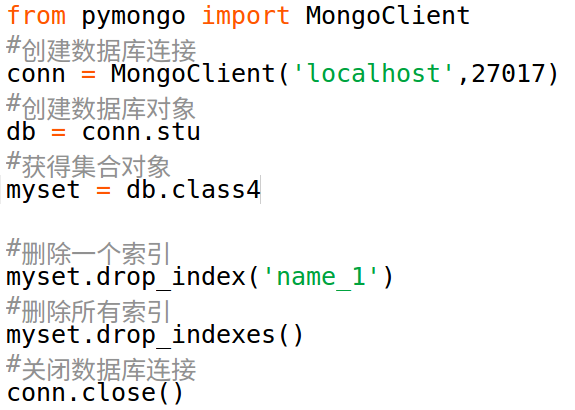






drop\_index()

drop\_indexes()

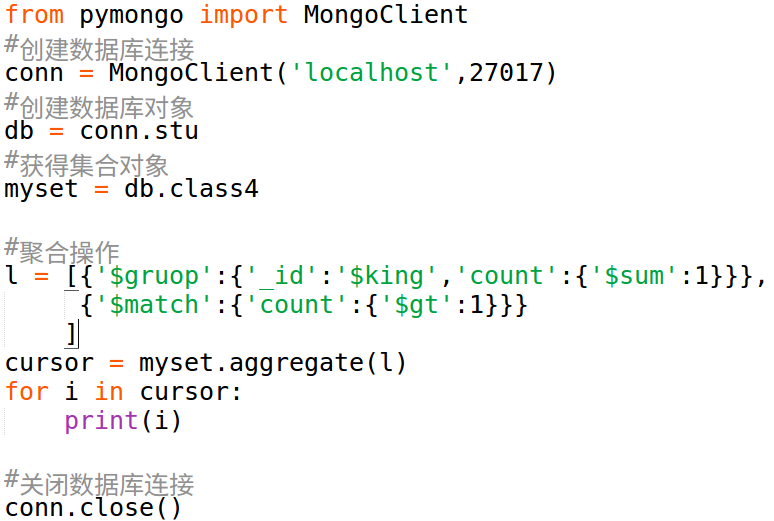


聚合

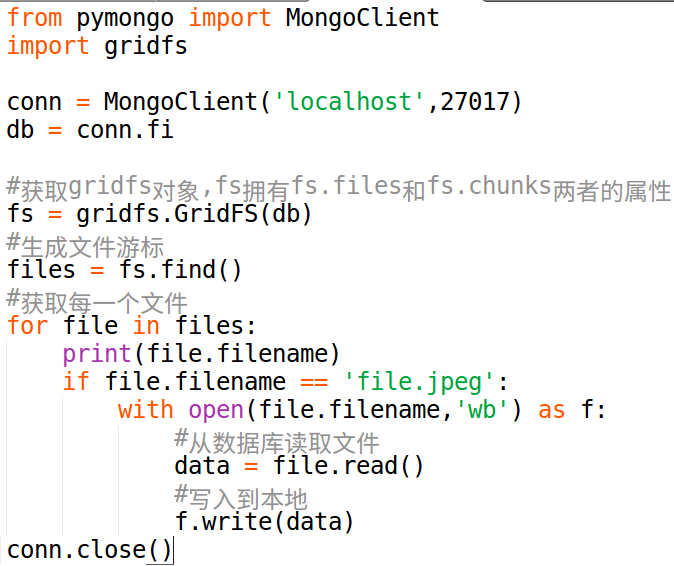
aggregate([])

参数：和mongoshell中写法一致

返回值：



文件读取



写文件



