P5 文档中的键值对是有序的

P6 MongoDB区分类型和大小写，文档不能有重复的键

P6 集合是一组文档，集合是无模式的

P8 数据库名必须小写

P9 把数据库的名字放到集合名前面，得到的就是集合的完全限定名，称命名空间

P9 MongoDB默认端口27017

P10 MongoDB会启动一个HTTP服务器，端口比主端口高1000，即28017，可通过访问http://localhost:28017

P12 insert函数添加一个文档到集合里

find查看（默认显示20个文档）；find One查看一个文档

P14 db.getCollection()访问集合函数

P17 MongoDB有3种数字类型：32位整数，64位整数，64位浮点数；默认情况下，shell中的数字都被当作32位整数

P18 创建一个新的Date对象会调用new Date()

P19 数组可以包含不同的数据类型元素，MongoDB可以使用原子更新修改数组内容

P20 文档必须有一个"\_id"键，默认是objectID对象；在集合内，其值唯一

P24 db.users.remove()会删除users集合中所有文档，但不会删除集合及索引

P25 update有两个参数，一个是查询文档，另一个是修改器文档

db.users.update（查询文档，修改器文档）

P28 使用修改器时"\_id"的值不能改，其他包括唯一索引的键，都可以改

P28 "$set"用来修改一个键的值，如果该键不存在就创建它

db.user.update(条件,{"$set":{字段,值}})

P29 "$unset"将键完全删除

P30 "$inc"增加或减少一个键的值，如果该键不存在就创建它

P31 "$inc"只能用于且其键也必须是整数、长整数、双精度浮点数（正或负）

P31 数组修改器只能用在值为数据的键上

"$push"向已有的数值末尾加入一个元素，如果键不存在就创建一个

\*.update({条件},{$push:{field:{values}}})

$addToSet 如果一个值不在数组里面就把它加进去,存在则不加

\*.update({条件},{$addToSet:{fields:{values}}})

P33 $addToSet配合$each，可以添加多个不同的值

\*.update({条件},{$addToSet:{fields:{$each:[values1,values2,...]}}})

P34 {$pop:{key:1}}从数组末尾删除一个元素

{$pop:{key:-1}}从数组头部删除一个元素

$pull按条件删除元素 \*.update({},{#pull:{key:value}})

$pull会将所有匹配的元素删除

P35 数组的定位修改器

\*.update({条件},{$inc:{field1.0.field2:value}}) 修改数组field1的第一个元素的field2键的值

\*.update({条件},{$inc:{field1.$.field2:value}}) 修改符合条件的第一个匹配的元素

P36 upsert是一种特殊的更新，如果没有文档符合更新条件，就会以这个条件和更新文档为基础创建一个新的文档，update第三个参数表示这个是upsert

\*.update({条件},{$inc:{field:value}},true)

P37 save是一个Shell函数，可以在文档不存在时插入，存在时更新，它只有一个参数：文档。要是这个文档含有”\_id”键，save会调用upsert，否则调用插入

P38 \*.update({条件},{$set:{field:value}},false,true) update第四个参数为true，表示更新所有符合条件的文档

Db.runCommand({getLastError:1}) getLastError命令查看更新了多少文档

{“err”:null,

“updatedExisting”:true, //为true，说明对已有文档进行更新

“n”:5, //5个文档被更新

“ok”:true}

P40 返回已更新的文档：findAndModify (一次只能处理一个文档)

db.runCommand({“findAndModify”:”processes”, //集合名

“query”:{“status”:”READY”}, //检索文档的条件

“sort”:{“priority”:-1}, //排序条件

“update”:{“$set”:{“status”:”RUNNING”}}}).value //更新

还可有remove：删除文档，与update有且只能有一个，如无匹配文档，返回错误

P45 find的第一个参数决定了要返回哪些文档，其形式也是一个文档。如不指定查询文档，默认为{},匹配集合的所有内容

P46 find的第二个参数指定返回哪些键，其中”\_id”总是被返回；也可用{field:0}指定不返回的键；查询文档的值必须是常量

P47 $lt:< $lte:<= $gt:> $gte:>= $ne:≠

P48 \*.find({field:{$in:[value1,value2,value3]}}) 查询一个键的多个值

$in可以指定不同类型的条件和值，即value1-3可以是不同类型

$nin返回数组中所有条件都不匹配的文档

$or接受一个包含所有可能条件的数组作为参数

P49 \*.find({field:{$mod:[num1,num2]}})

$mod将查询的值除以num1，若余数等于num2则返回结果（配合find使用情况下）

$not返回非给定条件的文档

P49 条件名是内层文档的键，修改器是外层文档的键；一个键可以有多个条件，但不能对应多个修改器

P50 find时值为null，不仅匹配值为null的文档，还会返回缺少这个键的文档，如仅仅想匹配键值为null的文档，可$exists条件判定键值已存在

\*.find({“z”:{$in:[null],”$exists”:true}})

P51 数组查询

\*.find({“fruit”:”apple”}) 查找数组中有apple元素的文档

\*.find({“fruit”:{$all:[”apple”,”banana”]}}) 用$all查找含多元素的文档（AND关系）

\*.find({“fruit.2”:”peach”}) 查找数组指定位置的元素，数组从0开始

\*.find({field:{$size:3}}) $size查询指定长度的数组，不能与比较符同用

P52 $slice返回数组的一个子集合，默认返回文档中所有键

\*.findOne({条件},{field:{$slice:±10}}) 返回数组前10（10）或后10（-10）条

\*.findOne({条件},{field:{$slice:[23,3]}}) 返回第24-26个元素

P54 查询内嵌文档

\*.find ({name:{first:Joe,last:Sch}}) 整个文档完全匹配

\*.find ({name.first:A,name.last:B}) 点表示法，匹配A orB

\*.find ({name:{$elemMatch:{first:A,last:{$gt:5}}}}) $elemMatch指定一组条件，仅当需要对一个内嵌文档的多个键操作时才会用到

P56 cursor.hasNext()检查是否有下一个 cursor. Next()获得下一个

P57 \*.find().limit(value) 只返回value个结果

\*.find().skip(value) 略过value个结果

\*.find().sort({field: ±1}) 1：升序 -1：降序

P65 \*.ensureIndex({field: ±1}) 创建索引 1：升序 -1：降序

P68 \*.ensureIndex({”comments.date”:1}) 内嵌文档键也可创建索引

P69 field1\_1\_field2\_-1\_...fieldN\_1 索引默认名称

\*.ensureIndex({a:1,b:1…},{name:abcd}) 为索引指定名称

P69 \*.ensureIndex({name:1},{unique:true}) 建立唯一索引，\_id索引默认创建，如果没有对应的键，索引会将其作为null存储，插入多个缺少索引键文档会报错

P70 \*.ensureIndex({name:1},{unique:true,dropDups:true}) 创建索引时删除除第一个的重复文档

P70 复合唯一索引，创建索引时，多个键共同唯一索引

P75 system.indexs集合包含每个索引的详细信息；system.namespaces保存所有集合的信息

P76 \*.ensureIndex({name:1},{backgroud:true }) 后台添加索引

P76 db.runCommand({dropIndexs:foo,index:alphabet})删除索引

P77 \*.ensureIndex({gps:[0,100]}) 创建地理空间索引

\*.find({gps:{$near:[40,-73]}}).limit(10) 查找10个距点最近的文档

db.runCommand({geoNear:map,near:[40,-73],num:10}) 查找10个距点最近的文档

P79 \*.count({条件})返回文档总数，使用时可省略条件

db.runCommand({distinct:集合名,key:键名})返回给定键所有不同的值

P81 group:condition条件

P82 group:{$exists:true}去掉没有键的一组

P83 group:finalize可修改传递的参数的返回新值

P84 group:$keyf定义分组函数

P84 MapReduce

P93 常用命令列表

P95 固定集合：大小固定，空间不足自动覆盖最旧文档，默认情况下无索引