MongoDB 基本使用

作者：elijah
原文链接：https://ld246.com/article/1469023027733
来源网站：链滴

成功启动MongoDB后，再打开一个命令行窗口输入mongo，就可以进行数库的一些操作。
输入help可以看到基本操作命令：
show dbs:显示数据库列表
show collections：显示当前数据库中的集合（类似系数据库中的表）
show users：显示用户
use <db name>：切换当前数据库，这和MS-SQL里面的意思一样
db.help(：显示数据库操作命令，里面有很多的命令
db.foo.help()：显示集合操作命令，同样很多的命令，foo指的是当前数据库下，一个叫foo的集合，并非真正意义上的命令
db.oo.find()：对于当前数据库中的foo集合进行数据查找（由于没有条件，会列出所有数据）
db.foo.find({ a : 1 })：对于当前数据库中的foo集合进行查找，条件是数据中有一个属性叫a，且的值为1
MongoDB没有创建数据库的命令，但有类似的命令。
如：如果你想创建一个“myTest”的数据库，先运行use myTest命令，后就做一些操作（如：db.createCollection('user')）,这样就可以创建一个名叫“myTest”的数据库。
数据库常用命令
1、Help查看命令提示
 help
  db.help();
  db.yourColl.help();
  db.youColl.find().help();
  rs.help();
2、切换/创建数据库
 use yourDB; 当创建一个集合(table)时候会自动创建当前数据库
3、查询所有数据库
 show dbs;
4、删除当前使用数据库
 db.dropDatabase();
5、从指定主机上克隆数据库
 db.cloneDatabase(“127.0.0.1”); 将指定机器上的数库的数据克隆到当前数据库
6、从指定的机器上复制指定数据库数据到某个数据库
 db.copyDatabase("mydb", "temp", "127.0.0.1");将本机的mydb的据复制到temp数据库中
7、修复当前数据库
 db.repairDatabase();
8、查看当前使用的数据库
 db.getName();
 db; db和getName方法是一样的效果，都可以查询当前使用的数据库
9、显示当前db状态
 db.stats();
10、当前db版本
 db.version();
11、查看当前db的链接机器地址
 db.getMongo();
Collection聚集集合
1、创建一个聚集集合（table）
 db.createCollection(“collName”, {size: 20, capped: 5 max: 100});
2、得到指定名称的聚集集合（table）
 db.getCollection("account");
3、得到当前db的所有聚集集合

 db.getCollectionNames();
4、显示当前db所有聚集索引的状态
 db.printCollectionStats();
 用户相关
1、添加一个用户
 db.addUser("name");
 db.addUser("userName", "pwd123", true); 添加用户、设置密码、否只读
2、数据库认证、安全模式
 db.auth("userName", "123123");
3、显示当前所有用户
 show users;
4、删除用户
 db.removeUser("userName");
其他
1、查询之前的错误信息
 db.getPrevError();
2、清除错误记录
 db.resetError();

查看聚集集合基本信息

<pre>1、查看帮助 db.yourColl.help();</pre>
<pre>2、查询当前集合的数据条数 db.yourColl.count();</pre>
<pre>3、查看数据空间大小 db.userInfo.dataSize();</pre>
<pre>4、得到当前聚集集合所在的db db.userInfo.getDB();</pre>
<pre>5、得到当前聚集的状态 db.userInfo.stats();</pre>
<pre>6、得到聚集集合总大小 db.userInfo.totalSize();</pre>
<pre>7、聚集集合储存空间大小 db.userInfo.storageSize();</pre>
<pre>8、Shard版本信息 db.userInfo.getShardVersion()</pre>
<pre>9、聚集集合重命名 db.userInfo.renameCollection("users"); 将userInfo重命名为users</pre>
<pre>10、删除当前聚集集合 db.userInfo.drop();</pre>
聚集集合查询

<pre>1、查询所有记录</pre>
<pre>db.userInfo.find();</pre>
<pre>相当于：select* from userInfo;</pre>
<pre>默认每页显示20条记录，当显示不下的情况下，可以用it迭代命令查询下一页数据。注意：键入t命令不能带“；”</pre>
<pre>但是你可以设置每页显示数据的大小，用DBQuery.shellBatchSize= 50;这样每页就显示50条记录了。</pre>
<pre> </pre>
<pre>2、查询去掉后的当前聚集集合中的某列的重复数据</pre>
<pre>db.userInfo.distinct("name");</pre>
<pre>会过滤掉name中的相同数据</pre>
<pre>相当于：select distict name from userInfo;</pre>
<pre> </pre>
<pre>3、查询age = 22的记录</pre>
<pre>db.userInfo.find({"age": 22});</pre>
<pre>相当于： select * from userInfo where age = 22;</pre>

<pre> </pre>
<pre>4、查询age > 22的记录</pre>

<pre>db.userInfo.find({age: {$gt: 22}});</pre>
<pre>相当于：select * from userInfo where age >
2;</pre>
<pre> </pre>
<pre>5、查询age < 22的记录</pre>
<pre>db.userInfo.find({age: {$lt: 22}});</pre>
<pre>相当于：select * from userInfo where age <
2;</pre>
<pre> </pre>
<pre>6、查询age >= 25的记录</pre>
<pre>db.userInfo.find({age: {$gte: 25}});</pre>
<pre>相当于：select * from userInfo where age ><span
= 25;</pre>
<pre> </pre>
<pre>7、查询age <= 25的记录</pre>
<pre>db.userInfo.find({age: {$lte: 25}});</pre>
<pre> </pre>
<pre>8、查询age >= 23 并且 age <= 26</pre>
<pre>db.userInfo.find({age: {$gte: 23, $lte: 26}});</pre>
<pre> </pre>
<pre>9、查询name中包含 mongo的数据</pre>
<pre>db.userInfo.find({name: /mongo/});</pre>
<pre>//相当于%%</pre>
<pre>select * from userInfo where name like ‘%mongo%’;</p
e>
<pre> </pre>
<pre>10、查询name中以mongo开头的</pre>
<pre>db.userInfo.find({name: /^mongo/});</pre>
<pre>select * from userInfo where name like ‘mongo%’;</pre

<pre> </pre>
<pre>11、查询指定列name、age数据</pre>
<pre>db.userInfo.find({}, {name: 1, age: 1});</pre>
<pre>相当于：select name, age from userInfo;</pre>
<pre>当然name也可以用true或false,当用ture的情况下河name:1效果一样，如果用false就是排除n
me，显示name以外的列信息。</pre>
<pre> </pre>
<pre>12、查询指定列name、age数据, age > 25</pre>
<pre>db.userInfo.find({age: {$gt: 25}}, {name: 1, age: 1});</p
e>
<pre>相当于：select name, age from userInfo where age ></spa
>25;</pre>
<pre> </pre>
<pre>13、按照年龄排序</pre>
<pre>升序：db.userInfo.find().sort({age: 1});
/pre>
<pre>降序：db.userInfo.find().sort({age: -1})
</pre>
<pre> </pre>
<pre>14、查询name = zhangsan, age = 22的数据</pre>
<pre>db.userInfo.find({name: 'zhangsan', age: 22});</pre>
<pre>相当于：select * from userInfo where name = ‘zhangsan’
and age = ‘22’;</pre>
<pre> </pre>

<pre>15、查询前5条数据</pre>
<pre>db.userInfo.find().limit(5);</pre>
<pre>相当于：selecttop 5 * from userInfo;<pre>
<pre> </pre>
<pre>16、查询10条以后的数据</pre>
<pre>db.userInfo.find().skip(10);</pre>
<pre>相当于：select * from userInfo where id not in (</pre>
<pre>selecttop 10 * from userInfo</pre>
<pre>);</pre>
<pre> </pre>
<pre>17、查询在5-10之间的数据</pre>
<pre>db.userInfo.find().limit(10).ski(5);</pre>
<pre>可用于分页，limit是pageSize，skip是第几页*pageSize</pre>
<pre> </pre>
<pre>18、or与 查询</pre>
<pre>db.userInfo.find({$or: [{age: 22}, {age: 25}]});</pre>
<pre>相当于：select * from userInfo where age = 22 or age = 25;</pre>
<pre> </pre>
<pre>19、查询第一条数据</pre>
<pre>db.userInfo.findOne();</pre>
<pre>相当于：selecttop 1 * from userInfo;<pre>
<pre>db.userInfo.find().limit(1);</pre>
<pre> </pre>
<pre>20、查询某个结果集的记录条数</pre>
<pre>db.userInfo.find({age: {$gte: 25}}).count();</pre>
<pre>相当于：select count(*) from userInfo where age >span>= 20;</pre>
<pre> </pre>
<pre>21、按照某列进行排序</pre>
<pre>db.userInfo.find({sex: {$exists: true}}).count();</pre>
<pre>相当于：select count(sex) from userInfo;</pre>

索引

<pre>1、创建索引</pre>
<pre>db.userInfo.ensureIndex({name: 1});</pre>
<pre>db.userInfo.ensureIndex({name: 1, ts: -1});</pre>
<pre> </pre>
<pre>2、查询当前聚集集合所有索引</pre>
<pre>db.userInfo.getIndexes();</pre>
<pre> </pre>
<pre>3、查看总索引记录大小</pre>
<pre>db.userInfo.totalIndexSize();</pre>
<pre> </pre>
<pre>4、读取当前集合的所有index信息</pre>
<pre>db.users.reIndex();</pre>

<pre> </pre>
<pre>5、删除指定索引</pre>
<pre>db.users.dropIndex("name_1");</pre>
<pre> </pre>
<pre>6、删除所有索引索引</pre>
<pre>db.users.dropIndexes();</pre>

 修改、添加、删除集合数据

<pre>1、添加</pre>
<pre>db.users.save({name: ‘zhangsan’, age: 2, sex: true});</pre>
<pre>添加的数据的数据列，没有固定，根据添加的数据为准</pre>
<pre> </pre>
<pre>2、修改</pre>
<pre>db.users.update({age: 25}, {$set: {name: 'changeName'}, false, true);</pre>
<pre>相当于：update users set name = ‘changeName’ where age = 25</pre>
<pre> </pre>
<pre>db.users.update({name: 'Lisi'}, {$inc: {age: 50}}, false, tre);</pre>
<pre>相当于：update users set age = age + 50 where name = ‘Lisi’;</pre>
<pre> </pre>
<pre>db.users.update({name: 'Lisi'}, {$inc: {age: 50}, $set: {name: 'hoho'}}, false, true);</pre>
<pre>相当于：update users set age = age + 50, name = ‘hoho’ where name = ‘Lisi’;</pre>
<pre> </pre>
<pre>3、删除</pre>
<pre>db.users.remove({age: 132});</pre>
<pre> </pre>
<pre>4、查询修改删除</pre>
<pre>db.users.findAndModify({</pre>
<pre>   query: {age: {$gte: 25}}, </pre>
<pre>   sort: {age: -1}, </pre>
<pre>   update: {$set: {name: 'a2'}, $inc: {age: 2}},</pre>
<pre>   remove: true</pre>
<pre>});</pre>
<pre> </pre>
<pre>db.runCommand({ findandmodify : "users", </pre>
<pre>   query: {age: {$gte: 25}}, </pre>
<pre>   sort: {age: -1}, </pre>
<pre>   update: {$set: {name: 'a2'}, $inc: {age: 2}},</pre>
<pre>   remove: true</pre>
<pre>});</pre>
update 或 remove 其中一个是必须的参数; 其他参数可选。
<table border="1" cellspacing="0" cellpadding="2">
<tbody>
<tr>
<td valign="top">
参数

```
</td>
<td valign="top">
<p><strong>详解</strong></p>
</td>
<td valign="top">
<p><strong>默认值</strong></p>
</td>
</tr>
<tr>
<td valign="top">
<p><em>query</em></p>
</td>
<td valign="top">
<p>查询过滤条件</p>
</td>
<td valign="top">
<p>{}</p>
</td>
</tr>
<tr>
<td valign="top">
<p><em>sort</em></p>
</td>
<td valign="top">
<p>如果多个文档符合查询过滤条件，将以该参数指定的排列方式选择出排在首位的对象，该对象将
操作</p>
</td>
<td valign="top">
<p>{}</p>
</td>
</tr>
<tr>
<td valign="top">
<p><em>remove</em></p>
</td>
<td valign="top">
<p>若为true，被选中对象将在返回前被删除</p>
</td>
<td valign="top">
<p>N/A</p>
</td>
</tr>
<tr>
<td valign="top">
<p><em>update</em></p>
</td>
<td valign="top">
<p>一个 <a href="http://www.mongodb.org/display/DOCS/Updating">修改器对象</a
</p>
</td>
<td valign="top">
<p>N/A</p>
</td>
</tr>
```

<tr>
<td valign="top">
new
</td>
<td valign="top">
若为true，将返回修改后的对象而不是原始对象。在删除操作中，该参数被忽略。
</td>
<td valign="top">
false
</td>
</tr>
<tr>
<td valign="top">
fields
</td>
<td valign="top">
参见<a href="http://www.mongodb.org/display/DOCS/Retrieving+a+Subset+of+Fields"
Retrieving a Subset of Fields (1.5.0+)
</td>
<td valign="top">
All fields
</td>
</tr>
<tr>
<td valign="top">
upsert
</td>
<td valign="top">
创建新对象若查询结果为空。 <a href="http://github.com/mongodb/mongo/blob/
aster/jstests/find_and_modify4.js">示例 (1.5.4+)
</td>
<td valign="top">
false
</td>
</tr>
</tbody>
</table>

语句块操作

<pre>1、简单Hello World</pre>
<pre>print("Hello World!");</pre>
<pre>这种写法调用了print函数，和直接写入"Hello World!"的效果是一样的；</pre>
<pre> </pre>
<pre>2、将一个对象转换成json</pre>
<pre>tojson(new Object());</pre>
<pre>tojson(new Object('a'));</pre>
<pre> </pre>
<pre>3、循环添加数据</pre>
<pre>> for (vari= 0;<sp
n>i<30;i++) {</pre>
<pre>... db.users.save({name: "u_" + i, age:

2 + i, sex: i % 2});</pre>
<pre>... };</pre>
<pre>这样就循环添加了30条数据，同样也可以省略括号的写法</pre>
<pre>> for (vari= 0;<sp
n>i<30;i++) db.users</s
an>.save({name: "u_" + i, age: 22 + i, sex: <
pan>i % 2});</pre>
<pre>也是可以的，当你用db.users.find()查询的时候，显示多
数据而无法一页显示的情况下，可以用it查看下一页的信息； </pre>
<pre> </pre>
<pre>4、find 游标查询</pre>
<pre>>varcursor = db.users</sp
n>.find();</pre>
<pre>> while (cursor.hasNext()) { </pre>
<pre> printjson(cursor.next()); </pre>
<pre>}</pre>
<pre>这样就查询所有的users信息，同样可以这样写</pre>
<pre>varcursor = db.users.find</sp
n>();</pre>
<pre>while (cursor.hasNext()) { printjson(cursor</sp
n>.next); }</pre>
<pre>同样可以省略{}号</pre>
<pre></pre>
<pre>5、forEach迭代循环</pre>
<pre>db.users.find().forEach(printjson);</pr
>
<pre>forEach中必须传递一个函数来处理每条迭代的数据信息</pre>
<pre> </pre>
<pre>6、将find游标当数组处理</pre>
<pre>varcursor = db.users.find</sp
n>();</pre>
<pre>cursor[4];</pre>
<pre>取得下标索引为4的那条数据</pre>
<pre>既然可以当做数组处理，那么就可以获得它的长度： cursor.length<
span>();或者cursor.count();</pre>
<pre>那样我们也可以用循环显示数据</pre>
<pre>for (vari = 0, len = c.length();
<len;i++) printjson(c[i</
pan>]);</pre>
<pre> </pre>
<pre>7、将find游标转换成数组</pre>
<pre>>var arr = db.users.find</
pan>().toArray();</pre>
<pre>> printjson(arr[2]);</pre>
<pre>用toArray方法将其转换为数组</pre>
<pre> </pre>
<pre>8、定制我们自己的查询结果</pre>
<pre>只显示age <= 28的并且只显示age这列数据</pre>
<pre>db.users.find({age: {$lte: 28}}, {age: 1}).forEach
/span>(printjson);</pre>
<pre>db.users.find({age: {$lte: 28}}, {age: true}).forEa
h(printjson);</pre>
<pre>排除age的列</pre>
<pre>db.users.find({age: {$lte: 28}}, {age: false}).forE

ch(printjson);</pre>
<pre> </pre>
<pre>9、forEach传递函数显示信息</pre>
<pre>db.things.find({x:4}).forEach(function(
) {print(tojson(x));});</pre>