mongodb学习

# 1.mongodb安装，这里安装的是zip版本

|  |
| --- |
| **安装MongoDB-windows-4.0.25**  **1.下载：https://fastdl.mongodb.org/win32/mongodb-win32-x86\_64-2008plus-ssl-4.0.25.zip**  **2.将zip文件解压缩到E:\mongodbw-4.0（可以根据实际修改路径）**  **3.将E:\mongodbw-4.0\bin路径添加到环境变量path中。**  **4.在E:\mongodbw-4.0\目录下面创建一个叫data的目录，在data目录里面创建一个db目录和一个log目录**  **5.在E:\mongodbw-4.0\bin\下面创建一个叫mongod.cfg文件，内容如下（路径可以根据需要修改）：**  **systemLog:**  **destination: file**  **path: E:\mongodbw-4.0\data\log\mongod.log**  **logAppend: true**  **storage:**  **journal:**  **enabled: true**  **dbPath: E:\mongodbw-4.0\data\db**  **net:**  **#bindIpAll: true**  **bindIp: 0.0.0.0**  **port: 27017**  **#security:**  **#authorization: enabled**    **6.安装MongoDB服务：以管理员身份运行cmd，在打开的窗口中输入：**  **mongod.exe --config “E:\mongodbw-4.0\bin\mongod.cfg” --install**    **7.开启mongodb服务：net start mongodb**  **看到**  **MongoDB 服务正在启动 .**  **MongoDB 服务已经启动成功。**  **说明安装成功** |

# 2、核心概念

|  |
| --- |
|  |
|  |

# 3.mongodb常用数命令

# 1.数据库操作

## a.显示所有数据库：show dbs;

## b.新建数据库，使用use 数据库名称；回车，如果有这个数据库，就切换到这个数据库，如果没有，就新建一个数据库

## c.删除数据库，需要分2步走

### 1）先切换到需要删除的数据库，使用 use 数据库名称；回车

### 如：我们需要删除students数据库：use students

|  |
| --- |
|  |

### 2）输入db.dropDatabase()；,然后就会删除

|  |
| --- |
|  |

## c.复制数据库,最好使用一些数据库工具

### db.copyDatabase("原来名称","副本名称"); # 注意：这个命令在mongodb4.20以上已经过时不能用了，这里是4.025,仍然可以使用

## d.数据库改名，最后分3步走

**db.copyDatabase('old\_name', 'new\_name');**

**use old\_name**

**db.dropDatabase();**

**d.切换数据库：use 数据库名称;**

## e.查看数据库名称db;回车，一般默认数据库是test

|  |
| --- |
|  |

## f.显示数据库相关信息：db.stats()

|  |
| --- |
|  |

# 2.集合操作

## a.新建集合，很简单，直接使用db.需要新建的集合名称.insert({对象数据})就可以了

#### 如：我们想添一个staffs集合

|  |
| --- |
|  |

#### 当然还可以使用db.createCollection(集合名称)来创建

## b.删除集合，分3步

### 1）切换到需要操作的数据库，use 数据库名称；回车

|  |
| --- |
|  |

### 2）查看该数据库的集合show collections;回车

|  |
| --- |
|  |

### 3）db.集合名称.drop()

|  |
| --- |
|  |

## c.给集合改名：db.旧集合名称.renameCollection(新集合名称)

|  |
| --- |
|  |

# 3.数据操作

## a.新增数据：db.集合名称.insert({})

## b.查询数据

### 1）查询全部，db.集合名称.find()

### 2)查询符合条件的数据db.集合名称.find({key:value})

## c.更新（修改）数据：db.集合名称.update({key:value},{$set,{key,value}})

### 不要这么写：db.集合名称.update({key:value},{key,value}) 不好用，会覆盖，需要使用上面的写法

#### 关于update方法下面有文档参考

### 如：db.staffs.update({"id":100},{$set:{name: "Jason"}});

|  |
| --- |
|  |

#### 解析：

// 1、如果第二个参数是一个对象，后边两个参数无效

// 2、如果第二个参数是通过set设置的话， 后两个参数才有效

// 3、后两个参数的第一个参数：true/如果数据查询不到，就创建 false/如果数据查询不到，就什么都不做

// 4、后两个参数第第二个参数：true/更新多条，false/更新一条

db.users.update({username: 'gp145'}, {set: {username: 'yl'}}, true, true)

// 5、如果使用updateMany, 就不需要传递后两个参数第二个了

db.users.updateMany({username: 'yl'}, {$set: {username: 'yangli'}})

## d.删除数据：db.集合名称.remove({key:value})

### 如：我们删除emp数据库里面的staffs集合里面id为101的记录：db.staffs.remove({id:101});

# mongodb参考文档

## Mongodb命令行 更新命令（update、save）

### 1.1update命令

update命令格式：

db.collection.update(criteria,objNew,upsert,multi)

参数说明：

criteria：查询条件

objNew：update对象和一些更新操作符

upsert：如果不存在update的记录，是否插入objNew这个新的文档，true为插入，默认为false，不插入。

multi：默认是false，只更新找到的第一条记录。如果为true，把按条件查询出来的记录全部更新。

示例：

1. > db.classes.insert({"name":"c1","count":30})
2. > db.classes.insert({"name":"c2","count":30})
3. > db.classes.find()
4. { "\_id" : ObjectId("5030f3a3721e16c4ab180cd9"), "name" : "c1", "count" : 30 }
5. { "\_id" : ObjectId("5030f3ab721e16c4ab180cda"), "name" : "c2", "count" : 30 }
6. >

示例1：把count大于20的class name修改为c3

1. > db.classes.update({"count":{$gt:20}},{$set:{"name":"c3"}})
2. > db.classes.find()
3. { "\_id" : ObjectId("5030f3a3721e16c4ab180cd9"), "name" : "c3", "count" : 30 }
4. { "\_id" : ObjectId("5030f3ab721e16c4ab180cda"), "name" : "c2", "count" : 30 }
5. >

由于没有指定upsert和multi的值，所以全部默认为false，由结果可以看出，只修改了第一条符合条件的记录。

示例2：把count大于20的class name修改为c4，设置multi为true

1. > db.classes.update({"count":{$gt:20}},{$set:{"name":"c4"}},false,true)
2. > db.classes.find()
3. { "\_id" : ObjectId("5030f3a3721e16c4ab180cd9"), "name" : "c4", "count" : 30 }
4. { "\_id" : ObjectId("5030f3ab721e16c4ab180cda"), "name" : "c4", "count" : 30 }
5. >

由于指定了multi为true，所以对两条符合条件的记录都进行了更新。

示例3： 把count大于50的class name修改为c5，设置upsert为true

1. > db.classes.update({"count":{$gt:50}},{$set:{"name":"c5"}},true,false)
2. > db.classes.find()
3. { "\_id" : ObjectId("5030f3a3721e16c4ab180cd9"), "name" : "c4", "count" : 30 }
4. { "\_id" : ObjectId("5030f3ab721e16c4ab180cda"), "name" : "c4", "count" : 30 }
5. { "\_id" : ObjectId("5030f589ce8fa8884e6cd441"), "name" : "c5" }
6. >

在集合中没有count大于50的记录，但是由于指定了upsert为true，如果找不到则会插入一条新记录。

### 1.2save命令

Mongodb另一个更新命令是save，格式如下：

db.collection.save(obj)

obj代表需要更新的对象，如果集合内部已经存在一个和obj相同的"\_id"的记录，Mongodb会把obj对象替换集合内已存在的记录，如果不存在，则会插入obj对象。

这条命令比较简单，示例就省略了。

## 二、数据更新操作符

### 1.$inc

用法：{$inc:{field:value}}

作用：对一个数字字段的某个field增加value

示例：将name为chenzhou的学生的age增加5

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "name" : "chenzhou", "age" : 22 }
3. #查询结果显示年龄为22
4. > db.students.update({name:"chenzhou"},{$inc:{age:5}})
5. #执行修改，把age增加5
6. > db.students.find()
7. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "name" : "chenzhou", "age" : 27 }
8. >
9. #查询结果显示年龄为27，修改成功

### 2.$set

用法：{$set:{field:value}}

作用：把文档中某个字段field的值设为value

示例： 把chenzhou的年龄设为23岁

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "name" : "chenzhou", "age" : 27 }
3. > db.students.update({name:"chenzhou"},{$set:{age:23}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "name" : "chenzhou", "age" : 23 }
6. >

从结果可以看到，更新后年龄从27变成了23

### 3.$unset

用法：{$unset:{field:1}}

作用：删除某个字段field

示例： 将chenzhou的年龄字段删除

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "name" : "chenzhou", "age" : 23 }
3. > db.students.update({name:"chenzhou"},{$unset:{age:1}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "name" : "chenzhou" }
6. >

### 4.$push

用法：{$push:{field:value}}

作用：把value追加到field里。注：field只能是数组类型，如果field不存在，会自动插入一个数组类型

示例：给chenzhou添加别名"michael"

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$push:{"ailas":"Michael"}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "Michael" ], "name" : "chenzhou" }
6. >

由结果可以看到，记录中追加了一个数组类型字段alias，且字段有一个为"Michael"的值

### 5.pushAll

用法：{$pushAll:{field:value\_array}}

作用：用法同$push一样，只是$pushAll可以一次追加多个值到一个数组字段内。

示例：给chenzhou追加别名A1，A2

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "Michael" ], "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$pushAll:{"ailas":["A1","A2"]}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "Michael", "A1", "A2" ], "name" : "chenzhou" }
6. >

### 6.$addToSet

用法：{$addToSet:{field:value}}

作用：加一个值到数组内，而且只有当这个值在数组中不存在时才增加。

示例：往chenzhou的别名字段里添加两个别名A3、A4

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "Michael", "A1", "A2" ], "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$addToSet:{"ailas":["A3","A4"]}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "Michael", "A1", "A2", [ "A3", "A4" ] ], "name" : "chenzhou" }
6. >

由结果可以看出，更新后ailas字段里多了一个对象，这个对象里包含2个数据，分别是A3、A4

### 7.$pop

用法：删除数组内第一个值：{$pop:{field:-1}}、删除数组内最后一个值：{$pop:{field:1}}

作用：用于删除数组内的一个值

示例： 删除chenzhou记录中alias字段中第一个别名

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "Michael", "A1", "A2", [ "A3", "A4" ] ], "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$pop:{"ailas":-1}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "A1", "A2", [ "A3", "A4" ] ], "name" : "chenzhou" }
6. >

由结果可以看书，第一个别名Michael已经被删除了。

我们再使用命令删除最后一个别名：

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "A1", "A2", [ "A3", "A4" ] ], "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$pop:{"ailas":1}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "A1", "A2" ], "name" : "chenzhou" }
6. >

由结果可以看出，alias字段中最后一个别名["A3","A4"]被删除了。

### 8.$pull

用法：{$pull:{field:\_value}}

作用：从数组field内删除一个等于\_value的值

示例：删除chenzhou记录中的别名A1

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "A1", "A2" ], "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$pull:{"ailas":"A1"}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "A2" ], "name" : "chenzhou" }
6. >

### 9.$pullAll

用法：{$pullAll:value\_array}

作用：用法同$pull一样，可以一次性删除数组内的多个值。

示例： 删除chenzhou记录内的所有别名

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ "A1", "A2" ], "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$pullAll:{"ailas":["A1","A2"]}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ ], "name" : "chenzhou" }
6. >

可以看到A1和A2已经全部被删除了

### 10.$rename

用法：{$rename:{old\_field\_name:new\_field\_name}}

作用：对字段进行重命名

示例：把chenzhou记录的name字段重命名为sname

1. > db.students.find()
2. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ ], "name" : "chenzhou" }
3. > db.students.update({name:"chenzhou"},{$rename:{"name":"sname"}})
4. > db.students.find()
5. { "\_id" : ObjectId("5030f7ac721e16c4ab180cdb"), "ailas" : [ ], "sname" : "chenzhou" }
6. >

由结果可以看出name字段已经被更新为sname了。

# 关于mongodb数据库的复制

## 使用 mongodump 和 mongorestore 工具在 MongoDB 中复制/克隆数据库

要在早期版本的 MongoDB 中复制数据库，您可以使用 copyDB 命令或其辅助方法 db.copyDatabase()。 但是，MongoDB 随后弃用了这些。

此外，从 4.2 版开始，MongoDB 删除了 copydb 命令和 db.copyDatabase() 方法，因此如果您使用 MongoDB 4.2 或更高版本，您将无法使用它们。 但是，有一种不同的方法来复制 MongoDB 数据库。

MongoDB 数据库工具可用于在 MongoDB 中克隆数据库。 您可以使用命令 mongodump 和 mongorestore。

MongoDB Database Tools 是一组用于处理 MongoDB 的命令行工具。 如果不确定是否安装了 MongoDB 数据库工具，请使用终端或命令提示符检查：

mongodump --version

mongorestore --version

这段代码主要是查找mongodump和mongorestore的版本。 如果您还没有它们，您可以按照 MongoDB 网站上的安装说明在您的 PC 上安装 MongoDB 数据库工具。

您必须使用系统的命令行来运行 mongodump 和 mongorestore（例如，新的终端或命令提示符窗口）。 它不应该从 mongo shell 中执行。

这是数据库克隆代码的示例：

mongodump --archive --db=CatHotel | mongorestore --archive --nsFrom='CatHotel.\*' --nsTo='CatHouse.\*'

在这种情况下，我们备份 CatHotel 数据库，然后将其所有集合恢复到名为 CatHouse 的数据库中。 换句话说，我们将 CatHotel 数据库克隆为 CatHouse。

这使用 mongodump 生成数据库备份文件，然后使用 mongorestore 以新名称恢复数据库。 首先，数据库被转储到标准输出流中并通过管道传输到 mongorestore 中。

以下是每个参数的作用：

| **参数** | **说明** |
| --- | --- |
| archive | 如果未指定存档文件，它将输出写入指定的存档文件或标准输出（stdout）。 在您的情况下，未指定存档文件，因此它将写入标准输出。 |
| db | 它将指定要备份的数据库。 在这种情况下，您将备份 CatHotel 数据库。 |
| nsFrom | 会指定转储文件中的集合。 星号通配符 (\*) 表示所有集合。 |
| nsTo | 指定恢复数据库中需要使用的集合名称。 |

您还可以使用 mongodump 转储所有数据库并在不带任何参数的情况下运行它。 但是，当您这样做时，本地和配置数据库不会包含在转储中。

1. 这是一种高速方法。
2. 它可以将集合克隆到另一个数据库和服务器。

## 使用 mongoexport 和 mongoimport 工具在 MongoDB 中复制/克隆数据库

MongoDB 工具包包括 mongoexport 和 mongoimport 工具。 该工具包可从 MongoDB 下载中心下载。

例如，在命令行中运行以下命令：

mongoexport.exe /host:<host> /port:<port> /db:test /collection:collection1 /out:collection1.json

mongoimport.exe /host:<host> /port:<port> /db:test /collection:collection2 /file:collection1.json

1. 这是一种快速的方法。
2. 它可以将集合克隆到不同的数据库和服务器。
3. 它不复制集合属性和索引。

## NoSQL Manager for MongoDB 的重复收集工具

重复收藏是一项专业功能。 它允许在同一个数据库中非常快速地复制集合。

右键单击 DB Explorer 中的 collection1 集合，然后在弹出菜单中选择 Duplicate collection1 Collection 项。

指定目标集合名称、复制参数，然后单击“复制”。

1. 这是一种高速方法。
2. 它复制集合属性和索引。
3. 它只能将集合克隆到同一个数据库。

## 将集合复制到 NoSQL Manager for MongoDB 的另一个数据库工具

将集合复制到另一个数据库是 NoSQL Manager for MongoDB Pro 的专业功能。 它允许在数据库和服务器之间复制一个或多个集合。

在 DB Explorer 中右键单击 collection1，然后在弹出菜单中选择 Copy collection1 Collection to another Database 项。

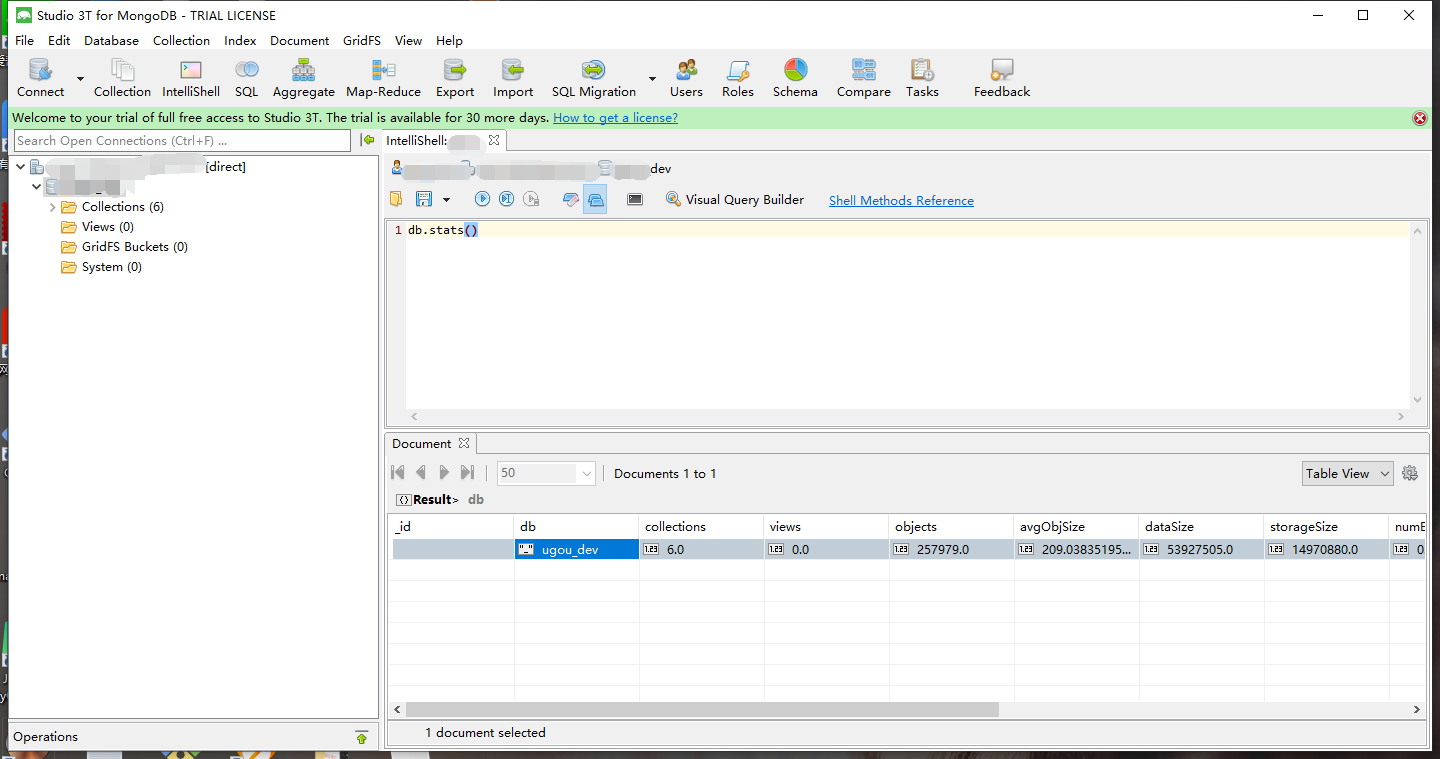
指定目标数据库和其他参数，然后单击执行。

1. 这是一种快速的方法。
2. 它复制集合属性和索引。
3. 它可以将集合复制到另一个数据库和服务器。
4. 它可以一次复制多个集合。
5. 它不能重命名集合。

# 小tips：如何永久免费使用studio 3T？

### Studio 3T

收费的，但是可以免费试用30天。支持导入导出，支持 sql 查询，支持语法填充，功能丰富。支持 MongoDB 4.0。支持在三大平台 Windows 、Mac 、Linux 上运行。 [下载链接](https://studio3t.com/download/)



该工具只能试用30天，可通过购买进行使用，如果你需要长期保持试用的功能，跟我做就好了：

新建一个文件，后缀为 .bat,并将以下内容写入。

**@echo off**

**ECHO 重置Studio 3T的使用日期......**

**FOR /f "tokens=1,2,\* " %%i IN ('reg query "HKEY\_CURRENT\_USER\Software\JavaSoft\Prefs\3t\mongochef\enterprise" ^| find /V "installation" ^| find /V "HKEY"') DO ECHO yes | reg add "HKEY\_CURRENT\_USER\Software\JavaSoft\Prefs\3t\mongochef\enterprise" /v %%i /t REG\_SZ /d ""**

**ECHO 重置完成, 按任意键退出......**

**pause>nul**

**exit**

然后我们运行该文件，即可重新开启试用30天，30天后我们也可以再次运行该脚本。或者我们可以将该脚本放到开机启动中，或者放到定时任务中。

上面脚本仅供学习使用，不能用于商业用途。

# 