npm install express 本地安装

npm install express -g 全局安装

npm uninstall express 卸载模块

npm update express 更新模块

npm search express 搜索模块

1. 启动mongo：在终端输入：mongo
2. 创建数据库：use databaseName，如果数据库不存在，则创建数据库，否则切换到指定数据库。
3. 查看所有数据库：show dbs
4. 向数据库新增数据：db.databaseName.insert({"name":"W3Cschool教程"})
5. 删除当前数据库：db.dropDatabase()
6. 查看数据库的数据：db.databaseName.find()；

db. databaseName.find().pretty()：会以格式化后的形式显示数据

1. 更新、插入数据：db.databaseName.update(search,data,isInsert,isAll)

search:表示要更新的数据的查询条件

data:表示要更新的新数据

isInsert:一个布尔值，表示如果不存在update的记录，是否插入objNew,true为插入，默认是false，不插入。即可用来更新，又可用来插入数据

isAll: mongodb默认是false,只更新找到的第一条记录，如果这个参数为true,就把按条件查出来多条记录全部更新。

例：

只更新第一条记录：db.test0.update( { "count" : { $gt : 1 } } , { $set : { "test2" : "OK"} } );

全部更新：db.test0.update( { "count" : { $gt : 3 } } , { $set : { "test2" : "OK"} },false,true );

1. 删除某一条数据：db. databaseName.remove(search,isOne)；

search:要删除的数据的查询条件；

可以添加多个key:value表示多个查询条件：db.col.find({"by":"w3cschool", "title":"MongoDB 教程"})；

若search是一个空对象，则会删除所有的数据：db. databaseName.remove({})

若添加$or数组，表示数组内的条件为或的关系：

db.col.find({$or:[{"by":"w3cschool"},{"title": "MongoDB 教程"}]})

AND 和 OR 联合使用：likes>50 and (by = 'w3cschool' or title = 'MongoDB 教程')

db.col.find({"likes": {$gt:50}, $or: [{"by": "w3cschool"},{"title": "MongoDB 教程"}]})

isOne:默认是false，可以不写，表示删除所有符合search条件的数据，如果isOne=true，表示只删除第一条符合search条件的数据

查询条件具有比较性时的对比符号：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 等于 | {key:value} | db.col.find({"by":"w3cschool"}) | by= w3cschool |  |  |
| 小于 | {key:{$lt:value}} | db.col.find({"by":{$lt:50}}) | by<50 |  |  |
| 小于或等于 | {key:{$lte:value}} | db.col.find({"by":{$lte:50}}) | by<=50 |  |  |
| 大于 | {key:{$gt:value}} | db.col.find({"by":{$gt:50}}) | by>50 |  |  |
| 大于或等于 | {key:{$gte:value}} | db.col.find({"by":{$gte:50}}) | by>=50 |  |  |
| 不等于 | {key:{$ne:value}} | db.col.find({"by":{$ne:50}}) | by!=50 |  |  |

1. 限制查询数量：db.databaseName.find().limit(num)

num：数字，表示限量多少

1. 跳过多少数量：db.databaseName.find().skip(num)

num：数字，表示跳过前面多少条

1. 排序：.sort(num)

num: 使用 1 和 -1 来指定排序的方式，其中 1 为升序排列，而-1是用于降序排列。

按字段 title 的降序排列：db.mycol.find({},{"title":1 }).sort({"title":-1})

如果没有指定sort()方法的排序方式，默认按照文档的升序排列

1. 创建索引：db.mycol.ensureIndex({key:1})

语法中 key 值为你要创建的索引字段，1为指定按升序创建索引，如果你想按降序来创建索引指定为-1即可。

db. mycol.ensureIndex({open: 1, close: 1}, {background: true})

通过在创建索引时加background:true 的选项，让创建工作在后台执行，防止阻塞

* 集合中索引不能超过64个
* 索引名的长度不能超过125个字符
* 一个复合索引最多可以有31个字段

1. 数据统计：db. mycol.aggregate(options)；

options为统计计算类型：

|  |  |  |
| --- | --- | --- |
| $sum | 计算总和 |  |
| $avg | 计算平均值 |  |
| $min | 获取集合中所有文档对应值得最小值 |  |
| $max | 获取集合中所有文档对应值得最大值 |  |
| $push | 在结果文档中插入值到一个数组中 |  |
| $addToSet | 在结果文档中插入值到一个数组中，但不创建副本 |  |
| $first | 根据资源文档的排序获取第一个文档数据 |  |
| $last | 根据资源文档的排序获取最后一个文档数据 |  |

例：

通过字段name对数据进行分组，并计算age字段相同值的总和，将统计结果赋给kk属性，\_id必须有。

db.lipeng.aggregate([{$group : {\_id : "$name", kk : {$sum : "$age"}}}])

通过字段name对数据进行分组，并计算age字段相同值的最大值，将统计结果赋给kk属性

db.lipeng.aggregate([{$group : {\_id : "$name", kk : {$max : "$age"}}}])

1. 管道符：

$project：修改输入文档的结构。可以用来重命名、增加或删除域，也可以用于创建计算结果以及嵌套文档。

$match：用于过滤数据，只输出符合条件的文档。$match使用MongoDB的标准查询操作。

$limit：用来限制MongoDB聚合管道返回的文档数。

$skip：在聚合管道中跳过指定数量的文档，并返回余下的文档。

$unwind：将文档中的某一个数组类型字段拆分成多条，每条包含数组中的一个值。

$group：将集合中的文档分组，可用于统计结果。

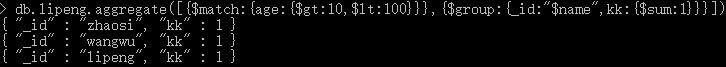
$sort：将输入文档排序后输出。

$geoNear：输出接近某一地理位置的有序文档。

例：

查询年龄大于10小于100：db.lipeng.aggregate([{$match:{age:{$gt:10,$lt:100}}}])

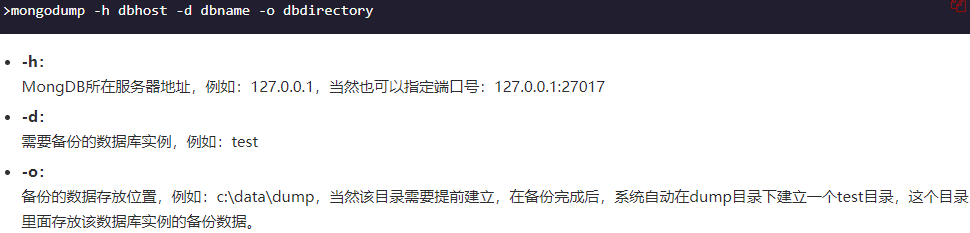
查询年龄大于10小于100，name作为分组，统计不同名字的各有多少个，将统计结果赋给kk：

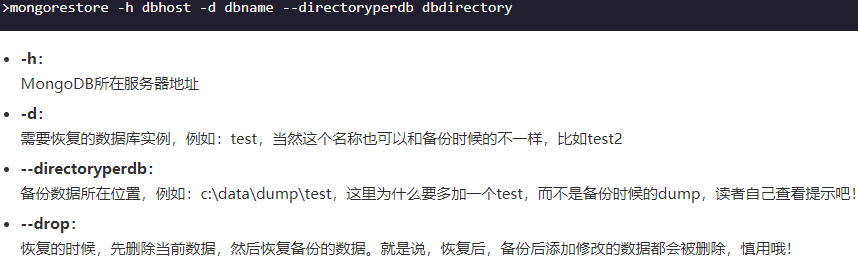
db.lipeng.aggregate([{$match:{age:{$gt:10,$lt:100}}},{$group:{\_id:"$name",kk:{$sum:1}}}])

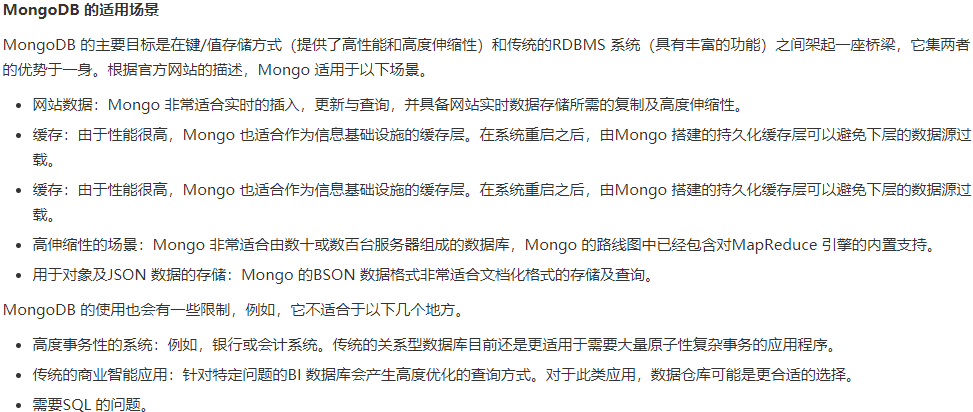
跳过前五个：db.article.aggregate({ $skip : 5 });

MongoDB的副本集与我们常见的主从有所不同，主从在主机宕机后所有服务将停止，而副本集在主机宕机后，副本会接管主节点成为主节点，不会出现宕机的情况。

当MongoDB存储海量的数据时，一台机器可能不足以存储数据也足以提供可接受的读写吞吐量。这时，我们就可以通过在多台机器上分割数据，使得数据库系统能存储和处理更多的数据。

在Mongodb中我们使用mongodump命令来备份MongoDB数据

mongodb使用 mongorerstore 命令来恢复备份的数据。

mongostat是mongodb自带的状态检测工具，在命令行下使用。它会间隔固定时间获取mongodb的当前运行状态，并输出。如果你发现数据库突然变慢或者有其他问题的话，你第一手的操作就考虑采用mongostat来查看mongo的状态。

使用正则表达式查找包含 w3cschool.cn 字符串的，且不区分大小写：

db.posts.find({post\_text:{$regex:"w3cschool.cn", $options:"$i"}})

以上查询也可以写为：db.posts.find({post\_text:/w3cschool.cn/i})