简介：MongoDB[1]  是一个基于分布式文件存储的数据库。由[C++](http://baike.baidu.com/view/824.htm)语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。

mongoDB

MongoDB[2]  是一个介于[关系数据库](http://baike.baidu.com/view/68348.htm)和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似[json](http://baike.baidu.com/view/136475.htm)的[bson](http://baike.baidu.com/view/3463259.htm)格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立[索引](http://baike.baidu.com/view/262241.htm)。

[分布式文件系统](http://baike.baidu.com/view/771589.htm)（Distributed File System）是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连。分布式文件系统的设计基于客户机/服务器模 式。一个典型的网络可能包括多个供多用户访问的服务器。另外，对等特性允许一些系统扮演客户机和服务器的双重角色。

[HBase](http://baike.baidu.com/view/1993870.htm)是一个分布式的、面向列的开源数据库，该技术来源于 Fay Chang 所撰写的Google论文“Bigtable：一个结构化数据的[分布式存储系统](http://baike.baidu.com/view/1911305.htm)。

Yonghong Data Mart是基于自有技术研发的一款数据存储、数据处理的软件。Yonghong Data Mart的分布式文件存储系统 (ZDFS)是在Hadoop HDFS基础上进行的改造和扩展，将服务器集群内所有节点上存储的文件统一管理和存储。

以上是对mongoDB的简单介绍，通过介绍要使用MongoDB除了了解他的基本数据存储，还要了解分布式系统如何部署，以及如何解决负载均衡问题。

1. 基本使用以命令方式
2. 数据库创建

use DATABASE\_NAME

1. 查看数据库

show dbsshow dbs

1. 删除数据库

db.dropDatabase()

1. 插入文档

db.user.insert({\_id:1,name:’mybo’})

插入语句使用类似json的bson

1. 更新已存在文档

db.collection.update(

<query>,

<update>,

{

upsert: <boolean>,

multi: <boolean>,

writeConcern: <document>

}

)

**参数说明：**

* **query** : update的查询条件，类似sql update查询内where后面的。
* **update** : update的对象和一些更新的操作符（如$,$inc...）等，也可以理解为sql update查询内set后面的
* **upsert** : 可选，这个参数的意思是，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。
* **multi** : 可选，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。
* **writeConcern** :可选，抛出异常的级别。

Exmple: db.col.update({'title':'MongoDB 教程'},{$set:{'title':'MongoDB'}},{multi:true})

1. 替换文档

db.col.save({

"\_id" : ObjectId("56064f89ade2f21f36b03136"),

"title" : "MongoDB",

"description" : "MongoDB 是一个 Nosql 数据库",

"by" : "Runoob",

"url" : "http://www.runoob.com",

"tags" : [

"mongodb",

"NoSQL"

],

"likes" : 110

})

1. 删除文档

db.col.remove({'title':'MongoDB 教程'})

删除第一条：db.COLLECTION\_NAME.remove(DELETION\_CRITERIA,1)

1. 删所有文档
2. db.col.remove({})
3. 文档查询

db.col.find()//查询所有

1. 条件查询例子

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

1. 条件操作符

* (>) 大于 - $gt
* (<) 小于 - $lt
* (>=) 大于等于 - $gte
* (<= ) 小于等于 - $lte

1. 类型操作

$type操作符是基于BSON类型来检索集合中匹配的数据类型，并返回结果。

1. limit 限制两条数据

db.col.find({},{"title":1,\_id:0}).limit(2)

{ "title" : "PHP 教程" }

{ "title" : "Java 教程" }

1. 我们除了可以使用limit()方法来读取指定数量的数据外，还可以使用skip()方法来跳过指定数量的数据，skip方法同样接受一个数字参数作为跳过的记录条数。

db.col.find({},{"title":1,\_id:0}).limit(1).skip(1)

{ "title" : "Java 教程" }

1. 排序

db.COLLECTION\_NAME.find().sort({KEY:1})//1 升序 2 降序

1. 建立索引

b.col.ensureIndex({"title":1})。

18. 聚合

MongoDB中聚合(aggregate)主要用于处理数据(诸如统计平均值,求和等)，并返回计算后的数据结果。有点类似sql语句中的 count(\*)。

db.mycol.aggregate([{$group : {\_id : "$by\_user", num\_tutorial : {$sum : 1}}}])

{

"result" : [

{

"\_id" : "w3cschool.cc",

"num\_tutorial" : 2

},

{

"\_id" : "Neo4j",

"num\_tutorial" : 1

}

],

"ok" : 1

}

* 1. 正则表达式

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

以上是对mongoDB的命令行操作，但是对于程序员来说一些简单命令操作可以方便我们使用，但是对于作为一个从事java开发人员来说我希望是熟练l了解在java如何调用。