1 基本概念

1.1 NoSQL

NoSQL(NoSQL = Not Only SQL)，意即"不仅仅是 SQL"，非关系型的数据库。

1.2 MongoDB

ongoDB 是由 C++语言编写的，是一个基于分布式文件存储的开源数据库系统。

1.3 架构特性

- 无数据结构限制
- 完全的索引支持
- 方便的冗余与扩展
- 良好的支持

1.4 要素特性

- 文档的“_id”在集合中是唯一的。
- 文档中的键值对是有序的；
- 键不能包含\0，不能为.和$，不能_开头；

2 安装配置

2.1 基础配置

1. 创建文件夹，包含文件夹 data、log、bin、conf

2. 在 conf 中创建 mongod.conf 配置如下：

```
port = 27017

dbpath = data

logpath = log/mongod.log
```

```
fork = true //linux 下有效
```

3. mongod -f conf/mongod.conf

2.2 创建服务

1. 在 conf 中创建 mongod.cfg 配置如下:

```
systemLog:
  destination: file
  path: D:\server\MongoDB\Server\3.4\log\mongod.log
  logAppend: true
storage:
  journal:
    enabled: true
  dbPath: D:\server\MongoDB\Server\3.4\data
net:
  port: 27017
```

2. 安装服务

```
mongod --config "...\mongod.cfg" --install
```

2.3 连接与关闭

```
mongo 127.0.0.1:27017
mongo 127.0.0.1:27017 -u user -p password
db.shutdownServer()
```

3 基本使用

3.1 基本操作

1. 查看当前数据库

```
db
```

2. 查看数据库

```
show dbs
```

3. 切换数据库（不存在时会自动创建）

```
use dbname
```

4. 删除数据库

```
db.dropDatabase()
```

3.2 写入数据

```
db.COLLECTION_NAME.insert(document)
```

```
db.rain_collection.insert({x:1})

db.rain_collection.insert({x:4,_id:4})

for(i=3;i<100;i++)db.rain_collection.insert({x:i})
```

3.3 查找数据

```
db.collection.find(query, projection)
```

- `query` ：可选，使用查询操作符指定查询条件
- `projection` ：可选，使用投影操作符指定返回的键。查询时返回文档中所有键值， 只需省略该参数即可（默认省略）。

```
db.col.find().pretty()
```

- `pretty()` 方法以格式化的方式来显示所有文档。

```
db.rain_collection.find()

db.rain_collection.find({x:1})

db.rain_collection.find().count()

db.rain_collection.find().skip(3).limit(2).sort({x:1})
```

3.3.1 模糊查找

/keyword/

3.3.2 查找实例

1. 删除不包含“军”的全部记录

```
db.baike.remove({category:{$nin:[/军/]}})
```

3.4 更新数据

```
db.collection.update(
   <query>,
   <update>,
   {
     upsert: <boolean>,
     multi: <boolean>,
     writeConcern: <document>
   }
)
```

- 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。

```
db.rain_collection.update({x:1},{x:999})

//局部更新

db.rain_collection.update({z:100},{$set:{y:99}})

//更新不存在时创建

db.rain_collection.update({y:1},{y:999},true)

//更新多条数据

db.rain_collection.update({c:1},{$set:{c:2}},false,true)
```

3.5 删除数据

```
db.collection.remove(
   <query>,
   {
     justOne: <boolean>,
     writeConcern: <document>
   }
)
```

- query :（可选）删除的文档的条件。
- justOne : （可选）如果设为 true 或 1，则只删除一个文档。
- writeConcern :（可选）抛出异常的级别。

```
db.collection.drop()
```

- 删除整个集合

```
db.rain_collection.remove({x:123})
```

4 索引

4.1 索引基础

4.1.1 获取索引

```
db.rain_collection.getIndexes()
```

4.1.2 创建索引

```
db.COLLECTION_NAME.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.rain_collection.ensureIndex({x:1})
```

4.2 _id 索引

默认，自动创建且唯一

4.3 单键索引

最普通

4.4 多键索引

方式同单键，但值为多个记录，如数组

4.5 复合索引

查询条件不只一个时

```
db.rain_collection.ensureIndex({x:1},{y:1})
```

4.6 过期索引

一段时间后会过期删除，必须是 ISODate 或 ISODate 数组，若是数组则按最小时间删除，不能是复合索引，删除时间不精确（每 60s 执行一次删除）。

```
db.collection.ensureIndex({time:1},{expireAfterSeconds:5})
```

4.7 全文索引

对字符串与字符串数组创建全文可搜索的索引。

4.7.1 建立索引

```
db.article.ensureIndex({key:”text”})

db.article.ensureIndex({key1:”text”, key2:”text”})

db.article.ensureIndex({”$**”:”text”})
```

4.7.2 查询

```
//单关键字

db.article.find({$text:{$search:”coffe”}})

//多关键字或

db.article.find({$text:{$search:”aa bb cc”}})

//多关键字与

db.article.find({$text:{$search:”\”aa\” \”bb\” \”cc\””}})

//不包含关键字

db.article.find({$text:{$search:”aa bb -cc”}})
```

4.7.3 相似度

```
//score 可自定义

db.article.find({$text:{$search:”coffe”}},{score:{$meta:”textScore”}})

//score 可用于排序

db.article.find({$text:{$search:”coffe”}},{score:{$meta:”textScore”}}).sort({score:{$meta:”textScore”}})
```

4.7.4 使用限制

- 每次只能指定一个$text

- 不能出现在$nor 查询中
- hint 失效

4.8 索引属性

4.8.1 name

1. 默认：

```
key1_order1_key2_order2
```

2. 定义：

```
ensureIndex({KEY:1},{name:”index_name”})
```

4.8.2 unique

```
ensureIndex({KEY:1},{unique:true/false})
```

4.8.3 sparse

```
ensureIndex({KEY:1},{sparse:true/false})//默认不稀疏
```

稀疏不必为不存在的字段创建索引

4.8.4 expireAfterSeconds

```
ensureIndex({time:1},{expireAfterSeconds:5})
```

4.9 地理位置索引

4.9.1 基础

4.9.1.1 类型

2d 索引，存储和查找平面上的点；

2dsphere 索引，存储和查找球面上的点。

4.9.1.2 查找方式

查找距离某个点一定距离内的点

查找包含在某区域内的点

4.9.2 2D 索引

4.9.2.1 创建

```
ensureIndex({KEY:”2d”})
```

4.9.2.2 位置表示

[经度,纬度]//经度[-180,180]，纬度[-90,90]，经度超出会提示，纬度不会

4.9.2.3 距离查询

1. near

```
find( {key:{$near:[1,1], $maxDistance:10}} )//默认 100 条
```

2. geoNear

```
db.runCommand({

geoNear:<collection>,

near:[x,y],

minDistance:(对 2d 索引无效)

maxDistance:

num:

...})
```

4.9.2.4 形状查询

矩形：{$box:[[<x1>,<y1>],[<x2>,<y2>]]}

圆形：{$center:[[<x1>,<y1>], r]}

多边形：{$polygon:[[<x1>,<y1>],[<x2>,<y2>],[<x3>,<y3>]]}

```
find( {key:{$geoWithin: {$box:[[<x1>,<y1>],[<x2>,<y2>]]}}} )

find( {key:{$geoWithin: {$center:[[<x1>,<y1>], r]}}} )

find({key:{$geoWithin:
{$polygon:[ [<x1>,<y1>],[<x2>,<y2>],[<x3>,<y3>]]}}})
```

4.9.3 2dsphere

4.9.3.1 创建

```
ensureIndex({KEY:"2d"})
```

4.9.3.2 位置表示

GeoJSON:{type:"", coordinates:[< coordinates >]}

4.9.4 索引构建情况分析

4.9.4.1 mongostat

mongostat -h 127.0.0.1:27017

idx miss//查询没有使用索引的情况

4.9.4.2 profile

测试观察阶段，生产环境不推荐使用。

db.getProfilingLevel()

db.setProfilingLevel()//0 关闭、1 记录超时(默认>100ms)、2 全部

db.getProfilingStatus()

db.system.profile.find()

4.9.4.3 日志

mongod.cfg 配置

verbose=vvvvv//1 到 5 个 v，越多越详细

4.9.4.4 explain

在查询操作后使用显示查询的详细信息

db.collection.find().explain()

5 安全

5.1 防护方式

物理隔离 > 网络隔离 > 防火墙 > 用户识别

5.2 开启权限认证

5.2.1 auth 开启

mongod.cfg 配置

auth=true

5.2.2 keyfile 开启

5.3 用户管理

5.3.1 创建用户

db.createUser(...)

{user:”<name>”, pwd:”<password>”, customData:<information>, roles:[{role:”<role>”, db:”<database>”}]}

5.3.2 内建用户角色

数据库角色：read, readWrite, dbAdmin, dbOwner, userAdmin

集群角色：clusterAdmin, clusterManager

备份角色：backup,restore

其他特殊权限：DBAdminAnyDatabase

5.3.3 创建用户角色

create role

6 维护

6.1 数据导入

6.1.1 JSON 文件

```
mongoimport --host <host> --port <port> --db <database-name> --collection <collection-name> --file input.json
```

6.2 数据导出

6.2.1 JSON 文件

```
mongoexport --host <host> --port <port> --db <database-name> --collection <collection-name> --out output.json
```

7 Tips

1. Shell 中单击 Tab 可自动补全，双击 Tab 可列出全部提示。

2.

8 参考资料

1. 官网

www.mongodb.org

2. 中国官网

www.mongoing.com

3. Github

github.com/mongodb

4. 慕课网

http://www.imooc.com/learn/295

5. 菜鸟教程

http://www.runoob.com/mongodb/mongodb-tutorial.html

《MongoDB 权威指南》