



Mongo DB 简介

2018.10



Mongo DB 是什么

- 由C++编写，是一个基于分布式文件存储的开源数据库系统
- 旨在为WEB应用提供可扩展的高性能数据存储解决方案
- 在高负载的情况下，可以添加更多的节点来保证服务器性能
- MongoDB 将数据存储为一个文档，数据结构由键值 (key=>value)对组成。MongoDB 文档类似于 JSON 对象。
字段值可以包含其他文档，数组及文档数组



Mongo DB 特性

- 层级 Database-Collection-Document
- 灵活的类JSON数据存储，每条文档的字段可以完全不同
- 方便的即席查询（ad hoc queries）、索引（indexing）和实时聚合（aggregation）
- 使用update()命令可以实现替换完成的文档（数据）或者一些指定的数据字段
- MongoDB允许在服务端执行脚本



Mongo DB 下载和安装

- 可以在mongodb官网下载安装包，地址为：

<https://www.mongodb.com/download-center#community>

- 也可以直接用 curl 命令下载，以 64 位 Linux 平台为例：

```
$ curl -O https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-3.4.9.tgz
```

```
$ tar -zxvf mongodb-linux-x86_64-3.4.9.tgz
```

- 解压完成之后，将 bin 目录加入环境变量 PATH 中
- `$./mongod` 启动 mongo 服务，默认 dbpath 为 `/data/db`



Mongo DB 主要概念

SQL术语/概念	MongoDB术语/概念	解释/说明
database	database	数据库
table	collection	数据库表/集合
row	document	数据记录行/文档
column	field	数据字段/域
index	index	索引
primary key	primary key	MongoDB自动将_id字段设置为主键



Mongo DB 与 MySQL 数据对比

id	user_name	email	age	city
1	Mark Hanks	mark@abc.com	25	Los Angeles
2	Richard Peter	richard@abc.com	31	Dallas



```
{
  "_id": ObjectId("5146bb52d8524270060001f3"),
  "age": 25,
  "city": "Los Angeles",
  "email": "mark@abc.com",
  "user_name": "Mark Hanks"
}
{
  "_id": ObjectId("5146bb52d8524270060001f2"),
  "age": 31,
  "city": "Dallas",
  "email": "richard@abc.com",
  "user_name": "Richard Peter"
}
```



Mongo DB 连接

- 标准 URI 连接语法:

`mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]`

- `mongodb://` 这是固定的格式，必须要指定。
- `username:password@` 可选项，用户名/密码。
- `host1` 必须指定至少一个host, `host1` 是这个URI唯一必须要填写的，它指定了要连接服务器的地址。如果要连接复制集，需要指定多个主机地址。
- `portN` 可选的指定端口，如果不填，默认为27017。
- `/database` 如果指定`username:password@`，连接并验证登陆指定数据库。若不指定，默认打开`test` 数据库。
- `?options` 连接选项。

- 示例 `$./mongo mongodb://admin:123456@localhost/test`



数据库 (database)

- 显示当前数据库

> db

- 查看所有数据库

> show dbs

- 新建数据库/连接到指定数据库

> use DATABASE_NAME

- 删除数据库

> db.dropDatabase()



集合 (collection)

- 新建集合

> db.createCollection(name, options)

示例: > db.createCollection("mycol", { capped : true, autoIndexId : true, size : 6142800, max : 10000 })

- 删除集合

> db. COLLECTION_NAME.drop()

- 在 MongoDB 中, 通常不需要专门创建集合; 当你插入一些文档时, MongoDB 会自动创建集合。



文档 (document)

- 文档操作 —— 插入文档 (Create, C)

> db.COLLECTION_NAME.insert(document)

这里的 document 是一个文档对象, 如:

```
> document=({name: 'iPhone', category: 'cellphone',  
value: 5000 });
```



文档操作 —— 删除文档 (Delete, D)

```
> db.COLLECTION_NAME.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_NAME.remove({})
```



文档操作 —— 更新文档 (Update, U)

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

- query : update的查询条件
 - update : update的对象和一些更新的操作符 (如 \$set, \$inc)
 - upsert : 可选, 这个参数的意思是, 如果不存在update的记录, 是否插入; 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。
- 示例:

```
> db.col.update({'name':'iPhone'},{$set:{'value': 6000}},{multi:true})  
  
> db.COLLECTION_NAME.save( <document>, { writeConcern: <document> } )
```



文档操作 —— 查询 (Retrieve, R)

> db.COLLECTION_NAME.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)

- 示例

> db.COLLECTION_NAME.find({"name": "iPhone"}, {"name": 1, _id: 0})



查询中的 AND 和 OR 条件

- MongoDB 的 find() 方法可以传入多个键(key), 每个键(key)以逗号隔开, 即常规 SQL 的 AND 条件

```
> db.COLLECTION_NAME.find({key1:value1, key2:value2})
```

- MongoDB OR 条件语句使用了关键字 **\$or**

```
> db.COLLECTION_NAME.find( { $or: [ {key1: value1},  
  
{key2:value2} ] } )
```

- 联合使用示例:

```
> db.col.find({"value": {$gt:2000}, $or: [{"category":  
  
"cellphone"}, {"name": "iPhone"}]}).pretty()
```



排序 (sort) 和索引 (index)

排序 (sort)

- 在 MongoDB 中使用 sort() 方法对数据进行排序，可以通过参数指定排序的字段，并使用 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

```
> db.COLLECTION_NAME.find().sort({KEY:1})
```

索引 (index)

- MongoDB使用 createIndex() 方法来创建索引

```
> db. COLLECTION_NAME.createIndex( keys, options )
```

Key 值为要创建索引的字段，options 取值 1 按升序创建索引，-1 为降序

- 索引示例

```
> db.col.createIndex({"title":1,"description":-1})
```



Q & A



尚硅谷

