

MongoDB 基本命令

目 录

1. MongoDB 基本概念.....	1
1.1. 文档 (Document)	1
1.2. 集合 (Collection)	2
1.3. 数据库.....	2
2. 简单 CRUD 操作.....	2
2.1. 创建数据库	3
2.2. 显示所有数据库	3
2.3. 显示所有集合	3
2.4. 创建 collection 并 insert 数据	3
2.5. 查询 collection 中数据	3
2.6. update 数据.....	3
2.7. remove 数据.....	4
2.8. 删除 Collection.....	4
2.9. 删除当前数据库.....	4

1. MongoDB 基本概念

在关系型数据库里，基本的概念有表，字段，记录。同样在 MongoDB 中也有相关的概念，这些概念有：

- (1) 集合 (Collection)：可以看成没有 Schema (模式) 的表。
- (2) 文档 (Document)：是 MongoDB 中数据的基本单元,类似于关系型数据库的行(但比行复杂的多)。
- (3) 每一个文档都有一个特殊的键"_id",它在文档所处的集合中是唯一的。

1.1. 文档 (Document)

文档是 MongoDB 的核心概念，多个值放置在一起便是文档。在关系型数据库中叫记录。比如有一条员工记录，姓名 (name) 为 zhangsan，年龄 (age) 为 20。

在 MongoDB 中表示为:

```
{"name":"zhangsan","age":20}
```

说明:

- (1) 使用{}表示一个文档, 或者叫一个对象, 在关系型数据库中叫记录。
- (2) 一个对象有多个属性, 每个属性有属性名和属性值, **name** 和 **age** 就是属性名, **zhangsan** 和 **20** 为属性对应的值。属性与属性之间使用逗号“,”隔开, 属性名与属性值用冒号“:”隔开。属性名两端需要加双引号。属性值, 如果是字符型, 则两端需要加双引号, 如果是数字, 则不需要加双引号。一个文档中不能有重复属性名的多个属性。

再例如一条客户资料文档, 客户名称(**customerName**)为新浪科技, 客户地址(**address**)为中关村, 客户联系电话(**telephone**)为 010-82842203。可表示为:

```
{"customerName":"新浪科技","address":"中关村","telephone":"010-82842203"}
```

1.2. 集合 (Collection)

集合是一组文档, 文档相当于关系型数据库中的行记录, 集合相当于表。在关系型数据库中, 表的字段一旦创建了, 就固定下来了, 添加数据是只能添加有字段的值, 不能添加字段外的值, 而且每条数据都有相同的字段。集合是无模式的, 这意味着一个集合里面的文档可以是各种各样的, 下面两个文档可以存在同一个集合中:

```
{"name":"zhangsan","age":20}
```

```
{"customerName":"新浪科技","address":"中关村","telephone":"010-82842203"}
```

1.3. 数据库

MongoDB 中多个文档组成集合, 同样多个集合组成数据库。一个 MongoDB 实例可以有多个数据库, 它们之间可视为完全独立的。每个数据库都有独立的权限控制, 即便是在磁盘上, 不同的数据库也放置在不同的文件中。将一个应用的所有数据存储在一个数据库中。

2. 简单 CRUD 操作

在使用命令时注意, 命令后没有分号“;”。

2.1. 创建数据库

```
> use demo
```

use 命令用来切换当前数据库，如果该数据库不存在，则会先新建一个。此处的“demo”是数据库名。

2.2. 显示所有数据库

```
> show dbs: 显示所有数据库。
```

2.3. 显示所有集合

```
> show collections: 显示当前数据库下的所有集合。
```

2.4. 创建 collection 并 insert 数据

```
> db.users.insert({'name':'Tom','pwd':'123'})
```

db.users.insert()命令向 users 集合中插入一条数据，准确的讲，应该是插入一个文档。如果集合 users 不存在，则系统会先自动创建集合 users，然后再插入数据，参数以 JSON 格式传入。这个和关系型数据库不一样，关系型数据库中，在插入数据前，表必须是存在的，而在 MongoDB 中，插入数据时，集合可以不存在，如果不存在，系统会自动创建。

2.5. 查询 collection 中数据

```
> db.users.find()
```

db.users.find(): 显示 users 集合下的所有数据文档。

如果明确的知道自己当前要查询的 collection，直接使用 db.collection 名.find() 查询所有数据文档（记录）。

2.6. update 数据

我们修改 Tom 的密码为 111。

```
> db.users.update({'name':'Tom'}, {'$set':{'pwd':'111'}},upset=true,multi=false)
```

然后通过如下语句查询数据是否被修改：

```
> db.users.find()
```

说明: `db.users.update()` 的第一个参数代表查询的条件, 第二个代表更新的字段, 第三个代表如果不存在则插入, 第四个代表是否允许修改多条记录。

2.7.remove 数据

删除 name 为 Tom 的数据文档:

```
> db.users.remove({'name':'Tom'})
```

删除所有数据文档, 但保留 collection。

```
> db.users.remove()
```

2.8.删除 Collection

```
> db.users.drop()
```

删除 users 集合。

2.9.删除当前数据库

```
> db.dropDatabase()
```

删除当前数据库。