mongoDB基本命令

show dbs

查看数据库列表

use 数据库名称

切换到制定的数据库 (如果没有会新建)但是不会马上新建,需要插入一条数据

db

查看当前操作数据库

show collections

显示当前db的集合

插入数据:

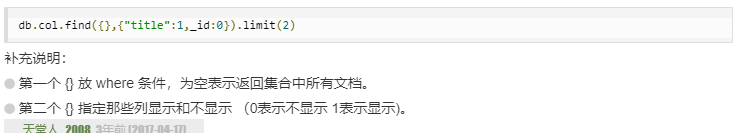
db.student.insertOne({“name”:”Jack”})

db.student.insert ({“name”:”Jack”})

db.COLLECTION\_NAME.insert(document) // doucment为对象

查找所有数据:

db.student.find() // 参数为对象{name:”xxx”|| /正则/} 返回为数组



查找单个数据:

db.student.findOne().pretty() // pertty为格式化输出 返回为对象

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）。

删除数据:

db.student.deleteMany({}) // 删除所有匹配到的数据

db.student.deleteOne({}) // 删除第一个匹配到的数据

更新数据:

db.student.update()

db.collection.update(

<query>,

<update>,

{

upsert: <boolean>,

multi: <boolean>,

writeConcern: <document>

}

)

截取:

db.student.find().limit(number)

跳过:

db.student.find().limit(number),skip(number)

//以上两个可以结合做分页

排序:

db.student..find().sort({key: 1 || 0})

// 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。







在node中如何操作mogodb

1. 使用官方的MongoDB包来使用 npm i mongodb

<https://www.npmjs.com/package/mongodb>

1. 使用第三方mongoose来操作mongodb

基于MongoDB官方的mongodb 二次封装

<https://mongoosejs.net/>

实际使用

db.dogs.insert({name:"哈士奇"})

db.dogs.update({name:"哈士奇"},{like:"拆家"})

db.dogs.update({like:"拆家"},{name:"哈士奇",like:"拆家"})

db.dogs.find({name:"哈士奇"})

db.dogs.deleteOne({like:"拆家"})

使用nodejs操作MySQL数据库

安装: npm i mysql