数据库NOSQL day6

副本集介绍 也成为MONGODB复制

-指在多个服务器上存储数据副本，并实现数据同步

-提高数据可用性，安全性，方便数据故障恢复

MongoDB复制原理

-副本集工作过程

-知道需要两个节点。其中一个是主节点，负责处理客户端请求，其余是从节点，负责复制主节点数据

-常见搭配方式：一主一从；一主多从

-主节点记录所有操作oplog，从节点定期轮询节点获取这些操作，然后对自己的数据副本执行这些操作，从而保证从节点的数据与主节点一致

副本集实现方式

Master-Slave主从复制

-启动一台服务器时加上‘-master’参数，作为主节点

-启动其他服务器时加上‘-Slave’ 和‘-source’作为从节点

主从复制优点

-从节点可以提供数据查询，降低主节点的访问压力

-由从节点执行备份，避免锁定主节点数据

-当主节点故障时，可快速切换到从节点实现高可用

配置replica sets

启动服务时先在主配置文件里加入

replSet=集群名 #每台都要写

启动服务

在想要哪台服务器做主节点就在哪台写

集合名 = {

\_id:”集群名”，

members[

{\_id,host:”IP地址:端口号”} #有多少就加多少个

{\_id,host:”IP地址:端口号”}

{\_id,host:”IP地址:端口号”}]};

#注意错了整个就费了，定义过别的集合名从新创

出时化Replica Set环境

执行命令

Rs.initiate(集合名)

查看信息

Rs.status()

查看是否是master库

rs.isMaster()

验证副本集配置

同步数据验证，允许从库查看数据

db.getMongo().setSlaveOk()

自动切换主库验证

rs.isMaster()

Mongodb文档管理

插入文档

Save()

格式

Db.集合名.save({key:”值”，key:”值”})

注意

-集合不存在时创建集合，然后插入记录

-\_id字段值已存在时，修改文档字段值

-\_id字段值不存在时，插入文档

Insert()

格式

Db.集合名.insert({key:“值”,key:”值”})

插入多条记录

Db.集合名.insertMany([{name:”xixi”,age:”19”},{name:”tom”,age:”18”}])

注意

-集合不存在时创建集合，然后插入记录

-\_id字段值已存在时，停止插入

-\_id字段值不存在时，插入文档

查询语法

显示所以行(默认输出20行，输入it可显示后续行)

Db.集合名.find()

显示第一行

Db.集合名。findOne()

指定显示条件并指定显示的字段

Db.集合名.find({条件}，{定义显示的字段})

例子：db.user.find({},\_id:0,name:1) #0为不显示，1为显示

行数显示限制

Limit(数字)

Db.集合名.find().limit(3) #显示前几行

Skip(数字)

Db.集合名.find().skip(2) #跳过前几行

Sort(字段名)

-db.集合名.find().sort(age:1|-1) #1升序 -1降序

匹配条件

简单条件

-db.集合名.find({key:”值”})

-db.集合名.find({key:”值”，keyname:”值”})

匹配条件

范围比较

-$in #在。。。里

-$nin #不再。。里

-$or #或

正则匹配

-db.user.find({name:/^a/})

数值比较

-$lt $lte $gt $gte $ne

< <= > >= !=

匹配null,也可以匹配没有的字段

更新文档

Update()

格式

-db.集合名.update({条件}，{修改的字段})

#以上会把除了修改的字段其他字段都删除且只匹配第一行

多文档更新

格式：默认只更新与条件匹配的第一行

Db.user.update({条件},{$set{修改字段}}，false,true)

$set/$unset

$set 条件匹配时，修改指定的字段值

-db.集合名.update({条件},$set:{修改的字段})

$unset 删除与条件不匹配文档的字段

-db.集合名.update({条件},{$unset:{key:“值”}})

$inc

$inc 条件匹配时，字段值自加或自减

-db.集合名.update({条件},{$inc:{字段名:数字}})

#+数字是自增 -数字自减

$push/$addToSet

$push 向数字中添加新元素

-db.集合.update({条件},{$push:{数字组名:”值”}})

$addToSet 避免重复添加

-db.集合.update({条件},{$addToSet:{数字组名:”值”}})

$pop/$pull

$pop从数组头部删除一个元素

-db.集合.update({条件},{$pop:{数字组名:”值”}})

$pull删除数组指定元素

-db.集合.update({条件},{$pull:{数字组名:”值”}})

#1 是删除数组尾部元素 -1是删除头部元素

删除文档

$drop/$remove

$drop 删除集合的同时删除索引

-db.集合名.drop()

$remove() 删除文档时不删除索引

-db.集合名.remove({}) #删除所有文档

-db.集合名.remove({条件}) #删除与条件匹配的文档