## 1、字符串string的操作

下表列出了常用的 redis 字符串命令

|  |  |  |
| --- | --- | --- |
| **CRUD** | **功能** | **示例** |
| 增 | 添加字符串 | **set** hello world |
| 设置多个KV | **mset** AA AAA BB BBB CC CCC |
| 设置过期时间 | **setex** hello1 5 word1 |
| key不存在时设置 | **setnx** hello2 word22 |
| 查 | 查询字符串 | **get** hello |
| 查询部分 | **getrange** hello 0 2 |
| 查询多key | **mget** hello hello2 hello3 |
| 查询长度 | **strlen** hello |
| 改 | 修改字符串 | **set** hello world2 |
| 追加 | **append** hello 333 |
| 删 | 删除字符串 | **del** hello |
| 其他 | 现获取后设置 | **getset** hello word222 |
| 替换 | **setrange** hello22 2 AA |
| 累加1 | **incr** number |
| 累加N | **incrby** number 10 |
| 累加小数 | **incrbyfloat** number 0.5 |
| 减一（只对integer有效） | **decr** number |
| 减N | **decrby** number 3 |

## 2、对hash列表的操作

Redis hash 是一个string类型的field和value的映射表，hash特别适合用于存储对象。

Redis 中每个 hash 可以存储 232 - 1 键值对（40多亿）

下表列出了 redis hash 基本的相关命令：

|  |  |  |
| --- | --- | --- |
| **CRUD** | **功能** | **示例** |
| 增 | 添加一个hash列表 | **hset** hkey1 name zhangsan |
| 不存在时添加 | **hsetnx** hkey2 name lisi |
| 一次设置多个值 | **hmset** hkey3 name wangwu age 20 sex 1 |
| 查 | 查看数据 | **hget** hkey1 name |
| 查看是否存在 | **hexists** hkey3 name |
| 查询key下所有值 | **hgetall** hkey3 |
| 查询key内所有的字段 | **hkeys** hkey3 |
| 查询key的数据量 | **hlen** hkey3 |
| 查询某一key内多字段的值 | **hmget** hkey3 name age |
| 查询key中所有的值 | **hincrbyfloat** hkey number1 0.5 |
| 改 | 修改数据 | **hset** hkey1 name zhangsan2 |
| 删 | 删除key中的数据 | **hdel** hkey3 sex |
| 其他 | 数据累加（1-N） | **hincrby** hkey number1 10 |
| 累加小数 | **hincrbyfloat** hkey number1 0.5 |

## 3、对list列表的操作

Redis列表是简单的字符串列表，按照插入顺序排序。你可以添加一个元素到列表的头部（左边）或者尾部（右边）

一个列表最多可以包含 232 - 1 个元素 (4294967295, 每个列表超过40亿个元素)。

下表列出了列表相关的基本命令

|  |  |  |
| --- | --- | --- |
| **CRUD** | **功能** | **示例** |
| 增 | 添加一个或多个值到列表 | **lpush** lkey1 zhangsan lisi |
| 将一个值插入到已存在的列表头部 | **lpushx** lkey1 wangwuxxx |
| 向已经存在的列表中插入数据(右边插入) | **rpush** lkey1 AA BB |
| 将一个值插入到已存在的列表尾部 | **rpushx** lkey1 ccc |
| 在某一数据之前插入 | **linsert** lkey1 before lisi lisibefore |
| 在某一数据之后插入 | **linsert** lkey1 after lisi lisiafter |
| 查 | 查询list中所有数据 （-1表示所有） | **lrange** lkey1 0 -1 |
| 通过索引查询数据 | **lindex** lkey1 0 |
| 查询列表长度 | **llen** lkey1 |
| 改 | 通过索引修改数据 | **lset** lkey1 5 eee |
| 删 | 数据修剪 | **ltrim** lkey1 4 6 |
| 删除指定key数据 | **del** lkey1 |
| 其他 | 移出并获取列表的第一个元素 | **lpop** lkey1 |
| 移出并获取列表的最后一个元素 | **rpop** lkey1 |
| 移除列表的最后一个元素，并将该元素添加到另一个列表并返回 | **rpoplpush** lkey1 lkey2 |

## 4、操作set集合

redis 的 Set 是 String 类型的无序集合。集合成员是唯一的，这就意味着集合中不能出现重复的数据。

Redis 中集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是 O(1)。

集合中最大的成员数为 232 - 1 (4294967295, 每个集合可存储40多亿个成员)。

下表列出了 Redis 集合基本命令：

|  |  |  |
| --- | --- | --- |
| **CRUD** | **功能** | **示例** |
| 增 | 向集合添加一个或多个数据 | **sadd** skey1 zhangsan lisi  **sadd** skey2 wangwu lisi |
| 查 | 查询集合所有数据 | **smembers** skey1 |
| 查询集合内数据总量 | **scard** skey1 |
| 查询两个集合的差值 | **sdiff** skey1 skey2 |
| 查询差值并将结果写入新集合 | **sdiffstore** skey3 skey1 skey2 |
| 查询两个集合交集 | **sinter** skey1 skey2 |
| 查询两个集合交集结果写入新集合 | **sinterstore** skey4 skey1 skey2 |
| 查询某一数据是否包含在集合中 | **sismember** skey1 zhangsan |
| 返回集合内随机的N个数据 | **srandmember** skey2 2 |
| 查询并集 | **sunion** skey1 skey2 |
| 查询并集结果写入新集合 | **sunionstore** skeyn skey1 skey2 |
| 改 | 将数据在一个集合移动到另一个集合 | **smove** skey1 skey2 zhangsan |
| 删 | 移除结果中的数据个数据 | **srem** skey2 zhangsan |

## 5、对key的操作

下表给出了与 Redis 键相关的基本命令：

|  |  |  |
| --- | --- | --- |
| **CRUD** | **功能** | **示例** |
| 查 | 查询所有符合模式的key | **Keys** \* |
| 查询剩余过期时间（毫秒) | **pttl** hkey1 |
| 查询剩余过期时间（秒) | **ttl** hkey1 |
| 在数据库中随机回返一个key | **randomkey** |
| 查询key对应数据的类型 | **Type** hkey1 |
| 改 | 对key进行重命名 | **Rename** hkey1 hkey10 |
| 删 | 删除存在的key | **del** BB |
| 其他 | 判断key是否存在 | **exists** AA |
| 为key 设置超时时间 | **expire** AA 10 |
| 为key 设置超时时间(毫秒) | **pexpire** CC 10000 |
| 移除key的过期时间 | **persist** hkey1 |

## 6、redis当中的数据库

Redis支持多个数据库，并且每个数据库的数据是隔离的不能共享，并且基于单机才有，如果是集群就没有数据库的概念。

Redis是一个字典结构的存储服务器，而实际上一个Redis实例提供了多个用来存储数据的字典，客户端可以指定将数据存储在哪个字典中。这与我们熟知的在一个关系数据库实例中可以创建多个数据库类似，所以可以将其中的每个字典都理解成一个独立的数据库。

每个数据库对外都是一个从0开始的递增数字命名，Redis默认支持16个数据库（可以通过配置文件支持更多，无上限），可以通过配置databases来修改这一数字。客户端与Redis建立连接后会自动选择0号数据库，不过可以随时使用SELECT命令更换数据库，如要选择1号数据库：

Redis支持多个数据库，并且每个数据库的数据是隔离的不能共享，并且基于单机才有，如果是集群就没有数据库的概念。

Redis是一个字典结构的存储服务器，而实际上一个Redis实例提供了多个用来存储数据的字典，客户端可以指定将数据存储在哪个字典中。这与我们熟知的在一个关系数据库实例中可以创建多个数据库类似，所以可以将其中的每个字典都理解成一个独立的数据库。

每个数据库对外都是一个从0开始的递增数字命名，Redis默认支持16个数据库（可以通过配置文件支持更多，无上限），可以通过配置databases来修改这一数字。客户端与Redis建立连接后会自动选择0号数据库，不过可以随时使用SELECT命令更换数据库，如要选择1号数据库：

node01:6379> select 1

node01:6379[1]> keys \*

清空redis数据库当中的数据

node01:6379> flushdb

清空所有redis数据库当中的数据

node01:6379> flushall