## 一．Redis基本命令及常识

①：连接服务端：

./redis-cli -h 127.0.0.1 -p 6379

②：Redis默认是有16个数据库的（0~15）通过select命令来切换数据库

select 1 -- 连接到第 2 个数据库 0开始计算

③：往数据库设置string类型值

set name zhangsan

④：查看数据库中key的数量

dbsize

⑤：查看刚才添加的key的值

get name

⑥：查看所有key的值

keys \*

⑦：清空全部数据库和清空当前库

flushall（清空全部库） flushdb（清空当前库）

⑧：删除添加的name key键

del name

## 二．List（集合）类型命令

创建两个集合：（后面会说list集合命令）

lpush listNumber 8.4 13 14 10.5 4 19.6 10 14 5.2 10 3 2.5 7 4.7 10 11.2 8 2.2 15.7 20.9

lpush listString remini Momen Pledg Memo Tende Biode Revie silen Romanti AusL Simpl Promis Romanti Bautifu smil Initiall sunse lemo firs Chaffere

查看添加的命令：

lrange listNumber 0 -1

lrange listString 0 -1

基本使用：

sort listNumber -- 默认排序是按照双精度浮点数类型从小到大升序

sort listNumber desc -- 按照从降序排列

sort listString alpha -- 如果排序中包含字符串类型则必须使用 alpha 修饰；则会使用字节排序

sort listString limit 2 5 alpha -- 排序并分页，分页从0开始数到2，再从2的位置往后数5个元素

sort listString alpha limit 2 5 store newliststring -- 排序分页，并把排序好的数据存放到另外一个集合中

by属性的使用（通过外部key排序）：

lpush mylist 20 15 18

set n\_20 b

set n\_15 a

set n\_18 c

sort mylist by n\_\* alpha

1) "15"

3) "20"

5) "18"

如果使用了by属性，那么sort就会先取出mylist的全部值，也就是20，15，18，然后再去跟by里设置的pattern进行

组合（n\_\*，里面的\*必须要和mylist的值有个一一对应关系，要不无法组合）；所以就变为了n\_20,n\_15,n\_18 ,

再按照这些拼装出的键里面的值进行排序（默认升序，从小到大）；这些键排完序规则为

n\_15 a

n\_20 b

n\_18 c

对应好之后，程序就会按照键后面的权重值来排序，所以最终排序a,b,c起作用，那么反过来推算

n\_15代表mylist的15，n\_20代表mylist的20，n\_18代表mylist的18，

所以15对应排序的权重a；20对应排序的权重b；18对应排序的权重c；

练习：现在有zhangsan（salary=5200），lisi（salary=3000），wangwu（salary=4500），mazi（salary=3200）四人，

但是他们想比较谁的工资高，该如何设计排序

lpush names zhangsan lisi wangwu mazi

set name\_zhangsan 5200

set name\_lisi 3000

set name\_wangwu 4500

set name\_mazi 3200

sort names by name\_\* desc

get是使用则是先加载出mylist，并且再拼装到n\_\*（一一对应）然会直接输出那个键的权重（根据mylist里面排序）

get属性使用（获取外部key）：

lpush mylist 20 15 18

set n\_20 b

set n\_15 a

set n\_18 c

1：获取外部key

127.0.0.1:6379> sort mylist get n\_\*

1) "a"

2) "c"

3) "b"

2：获取外部key并获取当前结合的集合值

127.0.0.1:6379> sort mylist get n\_\* get #

1) "a"

2) "15"

3) "c"

4) "18"

5) "b"

6) "20"

3：如果有多个get也是可以的，会依次匹对返回

127.0.0.1:6379> sort mylist get n\_\* get # get nn\_\* get #

set name zhangsan

getrange name 2 5 -- 获取范围值，最终返回 'angs'

getrange name 3 -2 -- 获取范围值，最终返回 'ngsa'

## 三．String（字符串）类型命令

语法：set key value [ex seconds|px milliseconds|exat timestamp|pxat milliseconds-timestamp|keepttl] [nx|xx] [get]

说明：设置string类型的键值，如果key已经保存了一个值，那么这个操作会直接覆盖原来的值，并且忽略原始类型

key：存储的key名称

value：存储的value数据

[ex seconds|px milliseconds|exat timestamp|pxat milliseconds-timestamp|keepttl]：

ex seconds：设置键key的过期时间，单位时秒

px milliseconds：设置键key的过期时间，单位时毫秒

exat timestamp：设置键过期的指定Unix时间，以秒为单位

pxat milliseconds-timestamp：设置键过期的指定Unix时间，以毫秒为单位

keepttl：获取key的过期时间

[nx|xx]：

nx：只有键key不存在的时候才会设置key的值

xx：只有键key存在的时候才会设置key的值

[get]：返回 key 存储的值，如果 key 不存在返回空

set name zhangsan -- 设置一个最基本的键值

set name zhangsan ex 60 -- 设置一个键值，并指定过期时间秒，ttl可以查看过期时间

set name zhangsan nx -- 设置一个键值，但是加上nx 代表只能更新已经存在的，如不存在name键则无法添加

set name zhangsan xx -- 生长一个键值，但是xx和nx相反

语法：setnx key value

说明：设置键值，存在此键则返回0不覆盖，否则正常设置

setnx name zhangsan -- 设置name为键，并赋值

语法：setrange key offset value

说明：偏移量offset>=0开始， 用value参数覆盖键key储存的字符串值。不存在的键key当作空白字符串处理。

set name zhangsan --创建原始键值

setrange name 5 ' yu xiao'

-- 把原有的 zhangsan 从第五位之后更改（0下标）；最终变为 "zhang yu xiao"

setrange name 14 out

-- 超出偏移则使用空格 '\x00' 代替一个空格；最终变为 "zhang yu xiao\x00out"

setrange address 2 anhui

-- 如果设置的键不存在则会新建，但是偏移量会以空格代替；最终变为 "\x00\x00anhui"

语法：strlen key

说明：获取指定key所储存的字符串值的长度。当key储存的不是字符串类型时，返回错误。

语法：append key value

说明：用于为指定的key追加值，成功后返回当前键里面的字符串全部长度（如果追加有空格需要使用 ''）

append name 'good good boy' -- 追加有空格的，并且成功后返回当前key的全部长度

语法：get key

说明：如果键key不存在，那么返回特殊值nil；否则返回键key的值。

get name -- 获取name键的值

语法：getdel key

说明：先获取到指定的key后，再删除获取的那个key；最终返回被删除的值

语法：getset key value

说明：设置更新key值，设置前先把原有的值返回出来，并设置新的值，如果key不存在时使用getset则返回nil，并设置新值

语法：getrange key start end

说明：获取指定的范围值，start（从0开始）end（从0开始）

set name zhangsan

getrange name 2 5 -- 获取范围值，最终返回 'angs'

getrange name 3 -2 -- 获取范围值，最终返回 'ngsa'

注：若使用getrange name 0 -1 (其中-1代表从后往前数)

语法：getex key [ex seconds|px milliseconds|exat timestamp|pxat milliseconds-timestamp|persist]

说明：获取指定的key值，并且获取后可以对当前key设置超时时间或者清除超时时间

[ex seconds|px milliseconds|exat timestamp|pxat milliseconds-timestamp|persist]：

ex seconds：设置键key的过期时间，单位时秒

px milliseconds：设置键key的过期时间，单位时毫秒

exat timestamp：设置键过期的指定Unix时间，以秒为单位

pxat milliseconds-timestamp：设置键过期的指定Unix时间，以毫秒为单位

persist：清除超时时间

语法：mset key value [key value ...]

说明：和set命令差不多，但是这个是批量设置，如果设置键存在则覆盖，不存在则添加

mset name zhangsn age 22 address anhui -- 批量设置name和age和address

语法：mget key [key ...]

说明：批量获取键的值，如果获取的某个不存在则返回（nil），其它正常返回

mget name aaa -- 批量获取name和aaa的值（aaa键不存在则返回nil）

语法：setex key seconds value

说明：将键key的值设置为value ，并将键key的过期时间设置为seconds秒钟，如果key存在则覆盖原有值

setex name 60 zhangsan -- 设置key为name，并且设置60秒过期时间

语法：psetex key milliseconds value

说明：将键key的值设置为value ，并将键key的过期时间设置为milliseconds毫秒，如果key存在则覆盖原有值

psetex name 70000 zhangsan -- 设置key为name，并且设置70秒过期时间

语法：msetnx key value [key value ...]

说明：当且仅当所有给定键都不存在时，为所有给定键设置值（如果添加的其中键在当前数据库存在则都不成功）

msetnx是一个原子性(atomic)操作，所有给定键要么就全部都被设置，要么就全部都不设置

msetnx name zhangsan age 22 -- 设置name和age两个键值

语法：incr key

说明：将key中储存的数字值增一，并返回增加后的值（只能用在整型，字符串啥的会报错）

语法：incrby key increment

说明：将key中储存的数字值增加指定步长increment，并返回增加后的值（只能用在整型，字符串啥的会报错）

语法：incrbyfloat key increment

说明：将key中储存的数字值增加指定步长increment，并返回增加后的值（只能用在浮点型，字符串啥的会报错）

incrbyfloat salary 333.33 -- 对salary添加步长333.33

语法：decr key

语法：将key中储存的数字值减一，并返回减后的值（只能用在整型，字符串啥的会报错）

语法：decrby key decrement

说明：将key中储存的数字值减指定步长increment，并返回减后的值（只能用在整型，字符串啥的会报错）

## 四．Set（无序集合）类型命令

语法：sadd key member [member ...]

说明：将一个或多个元素加入到集合中，添加已存在的集合元素将被忽略（不会添加上），返回添加成功的元素个数

127.0.0.1:6379> sadd mysetA zhangsan lisi wangwu mazi zhangsan

(integer) 4

-- 成功添加四个元素，其中一个为重复的，无法添加

语法：srem key member [member ...]

说明：删除指定的元素；如果指定的元素不是集合成员则被忽略，返回被删除元素个数，不含不存在的元素

语法：spop key [count]

说明：从集合key中删除一个或多个随机元素，并返回删除的元素

语法：srandmember key [count]

说明：随机返回集合key中的一个或多个随机元素，若返回个数的count大于集合总数则返回全部

注：srandmember随机返回不删除原集合，spop返回并删除原集合返回的元素

语法：sdiff key [key ...]

说明：返回第一个集合与其它集合之间的差异；说白就是第一个集合的某个元素在其它集合都不存在则这个元素会被返回，

key1 = {a,b,c,d}

key2 = {c}

key3 = {a,c,e}

SDIFF key1 key2 key3 = {b,d}

注：如果只携带一个key比较则会返回当前集合全部元素

举例：

127.0.0.1:6379> sadd mysetA zhangsan lisi wangwu mazi zhangsan

(integer) 4

127.0.0.1:6379> sadd mysetB anhui shanghai zhangsan mazi

(integer) 4

127.0.0.1:6379> sdiff mysetA mysetB

1) "wangwu"

2) "lisi"

语法：sdiffstore destination key [key ...]

说明：此命令和sdiff功能差不多，不同的是它将结果保存到destination集合，并返回成功添加到新集合上的个数。

127.0.0.1:6379> sdiffstore newmyset mysetA mysetB

(integer) 2

语法：sinter key [key ...]

说明：返回第一个集合与其它集合之间的交集；说白就是第一个集合的某个元素在其它集合都存在则这个元素会被返回，

key1 = {a,b,c,d}

key2 = {c}

key3 = {a,c,e}

SINTER key1 key2 key3 = {c}

注：如果只携带一个key比较则会返回当前集合全部元素

举例：

127.0.0.1:6379> sadd mysetA zhangsan lisi wangwu mazi zhangsan

(integer) 4

127.0.0.1:6379> sadd mysetB anhui shanghai zhangsan mazi

(integer) 4

127.0.0.1:6379> sinter mysetA mysetB

1) "zhangsan"

2) "mazi"

语法：sinterstore destination key [key ...]

说明：此命令和sinter功能差不多，不同的是它将结果保存到destination集合，并返回成功添加到新集合上的个数。

127.0.0.1:6379> sinterstore newmyset mysetA mysetB

(integer) 2

语法：sunion key [key ...]

说明：用于返回所有给定集合的并集

key1 = {a,b,c,d}

key2 = {c}

key3 = {a,c,e}

sunion key1 key2 key3 = {a,b,c,d,e}

注：如果只携带一个key比较则会返回当前集合全部元素

举例：

127.0.0.1:6379> sadd mysetA zhangsan lisi wangwu mazi zhangsan

(integer) 4

127.0.0.1:6379> sadd mysetB anhui shanghai zhangsan mazi

(integer) 4

127.0.0.1:6379> sunion mysetA mysetB

1) "mazi"

2) "zhangsan"

3) "lisi"

4) "anhui"

5) "wangwu"

6) "shanghai"

语法：sunionstore destination key [key ...]

说明：类似于sunion，不同的是不返回结果集，而是把返回存储在destination集合中。

127.0.0.1:6379> sunionstore newset mysetA mysetB

(integer) 6

语法：scard key

说明：返回集合中元素的数量（整型值）

语法：smembers key

说明：返回存储在key中的集合的所有的成员，此命令可以使用携带单个key的sdiff、sinter、sunion命令替代

语法：sismember key member

说明：判断元素member是否是集合key的成员，是返回1，否则0

127.0.0.1:6379> sismember mysetA zhangsan

(integer) 1

语法：smismember key member [member ...]

说明：批量判断元素members是否是集合key的成员，是返回1，否则0

127.0.0.1:6379> smismember mysetA zhangsan lisi

1) (integer) 1

2) (integer) 1

语法：smove source destination member

说明：从集合source中移动成员member到集合destination；

注：返回1代表成功，返回0代表移动element元素在source不存在

注：移动成功后会把element元素在原集合上删除

注：若被移动的元素在两个集合都存在，则会覆盖移动，再删除原集合上的元素

smove mysetA mysetB zhangsan -- 把元素zhangsan从集合mysetA移动到mysetB上

语法：sscan key cursor [match pattern] [count count]

说明：用于遍历集合中键的元素，sscan继承自scan，具体可以参考scan，上面第三章有说明

key：迭代指定元素

cursor：游标（告诉迭代器从哪开始迭代）

[match pattern]：过滤筛选条件

[count count]：迭代的个数

举例：每次迭代两个

127.0.0.1:6379> sscan mysetA 0 match \* count 2

1) "1"

2) 1) "zhangsan"

2) "lisi"

127.0.0.1:6379> sscan mysetA 1 match \* count 2

1) "0"

2) 1) "wangwu"

2) "mazi"

补充Redis7.0.0即将提供

语法：sintercard key [key ...]