大数据库系统

3.3 Redis数据类型及操作

3.3 Redis数据类型及操作

- 主要内容

3.3.1 通用命令

3.3.2 STRING（字符串）

3.3.3 LIST（列表）

3.3.4 SET（无序集合）

3.3.5 ZSET（有序集合）

3.3.6 HASH（哈希表）

3.3.5 ZSET（有序集合）

- Sorted Set类型及相关命令
  - 有序集合（Sorted Set）也是String类型的集合。
  - 有序集合中不存在重复的元素，每个集合元素都有一个对应的double类型的分数。
  - 通过分数来为集合元素进行从小到大的排序，分数值不唯一，可以重复

3.3.5 ZSET（有序集合）

- ZADD命令

ZADD key score member [score member] [score member] …

将一个或多个member元素及它对应的score（分数）值加入有序集合key中

注意：如果有序集合key中已经存在某个member元素，那么只更新这个member元素的score值，然后重新插入member元素，以此来确保member元素在正确的位置上

```
redis 127.0.0.1:6379> zadd class 12 lily 13 lucy 18 lilei 6 poly
(integer) 4
redis 127.0.0.1:6379>
```

注意：分数要在member之前

3.3.5 ZSET（有序集合）

- ZRANGE 命令

ZRANGE key start stop [WITHSCORES]

- 返回有序集合key中指定区间内的元素。返回的元素按照score值从小到大的顺序排序，具有相同score值的元素会按照字典序排序
- 可以使用WITHSCORES同时返回集合元素和这些元素所对应的score值，返回的格式是：value1, score1, …, valueN, scoreN
- start 和stop中 0表示有序集合key中的第一个元素，1表示有序集合key中的第二个元素，以此类推，使用-1表示有序集合key中的最后一个元素，使用-2表示有序集合key中的倒数第二个元素，以此类推

3.3.5 ZSET（有序集合）

例1：根据分数按顺序显示class集合中的元素

```
redis 127.0.0.1:6379> zadd class 12 lily 13 lucy 18 lilei 6 poly
(integer) 4
redis 127.0.0.1:6379> zrank class 0 3
(error) ERR wrong number of arguments for 'zrank' command
redis 127.0.0.1:6379> zrange class 0 3
1) "poly"
2) "lily"
3) "lucy"
4) "lilei"
redis 127.0.0.1:6379>
```

升序排列，poly的分数最低，lilei的分数最高

例2：根据分数取class集合中的第2名到第4名，并显示各自分数

```
redis 127.0.0.1:6379> zrange class 1 3 withscores
1) "lily"
2) "12"
3) "lucy"
4) "13"
5) "lilei"
6) "18"
redis 127.0.0.1:6379>
```

3.3.5 ZSET（有序集合）

- ZRANGEBYSCORE命令

ZRANGEBYSCORE key min max [WITHSCORES] [LIMIT offset count]

- 返回有序集合key中，所有score值介于min和max之间（包含等于min和max）的元素
- 当不知道min和max参数的具体值时，可以使用-inf来表示min值，使用+inf来表示max值
- 在默认情况下，min与max区间是闭区间（小于等于或大于等于），也可以在参数前面添加“(”符号来使用可选的开区间
- LIMIT可以指定跳过多少个member，count指定跳过多少member后取count个元素

- 例1：取集合class中分数范围为[13，18]的元素

```
redis 127.0.0.1:6379> zadd class 12 lily 13 lucy 18 lilei 6 poly
(integer) 4
redis 127.0.0.1:6379> zrangebyscore class 13 18
1) "lucy"
2) "lilei"
redis 127.0.0.1:6379>
```

- 例2：取集合class中分数范围为[1，20]，并跳过第一个成员后取之后的两个成员

```
redis 127.0.0.1:6379> zrangebyscore class 1 20 limit 1 2
1) "lily"
2) "lucy"
redis 127.0.0.1:6379>
```

- 例3：取集合class中分数范围为[10，20]的成员，并显示各自分数

```
redis 127.0.0.1:6379> zrangebyscore class 10 20 withscores
1) "lily"
2) "12"
3) "lucy"
4) "13"
5) "lilei"
6) "18"
redis 127.0.0.1:6379>
```

3.3.5 ZSET（有序集合）

◆ ZRANK命令

ZRANK key member

获取有序集合key中member元素的排名，按照score值从小到大的顺序排序，排名以0为底

- 例：查询class中lily、 poly、lilei所在的名次，class按分数升序排列

```
redis 127.0.0.1:6379> zadd class 12 lily 13 lucy 18 lilei 6 poly
(integer) 4
redis 127.0.0.1:6379> zrank class lily
(integer) 1
redis 127.0.0.1:6379> zrank class poly
(integer) 0
redis 127.0.0.1:6379> zrank class lilei
(integer) 3
redis 127.0.0.1:6379> zrank class zhangfei
(nil)
redis 127.0.0.1:6379>
```

- zhangfei不存在所以返回nil

3.3.5 ZSET（有序集合）

◆ ZREVRANK命令

ZREVRANK key member

ZREVRANK命令用于返回有序集合key中member元素的排名，其中，有序集合元素按照score值从大到小的顺序排序，排名以0为底

- 例：查询class中lily、 poly、lilei所在的名次，class按分数降序排列

```
redis 127.0.0.1:6379> zadd class 12 lily 13 lucy 18 lilei 6 poly
(integer) 4
```

```
redis 127.0.0.1:6379> zrevrank class lilei
(integer) 0
redis 127.0.0.1:6379> zrevrank class poly
(integer) 3
redis 127.0.0.1:6379>
```

3.3.5 ZSET（有序集合）

- ZREMRANGEBYSCORE命令

ZREMRANGEBYSCORE key min max

删除有序集合key中，所有score值介于min和max之间（包含等于min或max）的元素，返回被删除元素的数量

例：删除集合class中所有分数范围在[10，15]的元素

```
redis 127.0.0.1:6379> zrange class 0 -1 withscores
1) "poly"
2) "6"
3) "lily"
4) "12"
5) "lucy"
6) "13"
7) "lilei"
8) "18"
redis 127.0.0.1:6379> zremrangebyscore class 10 15
(integer) 2
redis 127.0.0.1:6379> zrange class 0 -1 withscores
1) "poly"
2) "6"
3) "lilei"
4) "18"
redis 127.0.0.1:6379>
```

3.3.5 ZSET（有序集合）

- ZREMRANGEBYRANK命令

ZREMRANGEBYRANK key start stop

删除有序集合key在指定排名（rank）区间内的所有元素。区间范围由下标参数start和stop给出，包含start和stop在内。返回被删除元素的数量

例：删除集合class中的第一名和第二名，class按分数升序排列

```
redis 127.0.0.1:6379> zrange class 0 -1 withscores
1) "poly"
2) "6"
3) "lily"
4) "12"
5) "lucy"
6) "13"
7) "lilei"
8) "18"
redis 127.0.0.1:6379> zremrangebyrank class 0 1
(integer) 2
redis 127.0.0.1:6379> zrange class 0 -1 withscores
1) "lucy"
2) "13"
3) "lilei"
4) "18"
redis 127.0.0.1:6379>
```

3.3.5 ZSET（有序集合）

◆ ZREM命令

ZREM key member [member ...]

ZREM命令用于删除有序集合key中的一个或多个元素，不存在的元素会被忽略，返回被删除元素的数量，不包括被忽略的元素

- 例：删除集合class中的lucy

```
redis 127.0.0.1:6379> zrange class 0 -1 withscores
1) "lucy"
2) "13"
3) "lilei"
4) "18"
redis 127.0.0.1:6379> zrem class lucy
(integer) 1
redis 127.0.0.1:6379> zrange class 0 -1 withscores
1) "lilei"
2) "18"
```

3.3.5 ZSET（有序集合）

- ZCARD命令

ZCARD key

获取有序集合key中的元素数量

```
redis 127.0.0.1:6379> zadd ty 25 zhang 27 guan 28 liubei
(integer) 3
redis 127.0.0.1:6379> zcard ty
(integer) 3
redis 127.0.0.1:6379> zadd ty 23 zhao
(integer) 1
redis 127.0.0.1:6379> zcard ty
(integer) 4
redis 127.0.0.1:6379>
```

3.3.5 ZSET（有序集合）

- ZCOUNT命令

ZCOUNT key min max

获取有序集合key中，score值在min和max之间（默认包含score值等于min或max）的元素数量

```
redis 127.0.0.1:6379> zadd ty 25 zhang 27 guan 28 liubei
(integer) 3
redis 127.0.0.1:6379> zcard ty
(integer) 3
redis 127.0.0.1:6379> zadd ty 23 zhao
(integer) 1
redis 127.0.0.1:6379> zcard ty
(integer) 4
redis 127.0.0.1:6379> zcount ty 25 30
(integer) 3
redis 127.0.0.1:6379>
```

3.3.5 ZSET（有序集合）

- ZINTERSTORE命令

ZINTERSTORE destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM | MIN | MAX]

计算给定的一个或多个有序集合key的交集，其中给定key的数量必须和numkeys相等，并将该交集存储到destination中

- [WEIGHTS weight [weight ...]]表示权重，先将key的各个元素分数与相应集合的权重相乘，不指定权重都为1再进行交集和聚合运算

- 过程：将两个集合个元素乘以权重得到新的score，之后元素作交集，然后将两者相同元素的score作聚合运算，如求和sum，最大最小max，min运算

- 例1：集合lisi和wang进行交集运算，相同元素分数相加

```
redis 127.0.0.1:6379> zadd lisi 3 cat 5 dog 6 horse
(integer) 3
redis 127.0.0.1:6379> zadd wang 2 cat 6 dog 8 horse 1 donkey
(integer) 4
redis 127.0.0.1:6379> zinterstore result 2 lisi wang
(integer) 3
redis 127.0.0.1:6379> zrange result 0 -1 withscores
1) "cat"
2) "5"
3) "dog"
4) "11"
5) "horse"
6) "14"
```

- 例2：集合lisi和wang进行交集运算，相同元素取最小分数

```
redis 127.0.0.1:6379> zinterstore result 2 lisi wang aggregate min
(integer) 3
redis 127.0.0.1:6379> zrange result 0 -1 withscores
1) "cat"
2) "2"
3) "dog"
4) "5"
5) "horse"
6) "6"
redis 127.0.0.1:6379>
```

- 例3：集合lisi和wang进行交集运算，相同元素取最大分数

```
redis 127.0.0.1:6379> zadd lisi 3 cat 5 dog 6 horse
(integer) 3
redis 127.0.0.1:6379> zadd wang 2 cat 6 dog 8 horse 1 donkey
(integer) 4
redis 127.0.0.1:6379> zinterstore result 2 lisi wang aggregate max
(integer) 3
redis 127.0.0.1:6379> zrange result 0 -1 withscores
1) "cat"
2) "3"
3) "dog"
4) "6"
5) "horse"
6) "8"
```

- 例4：集合lisi和wang进行交集运算，相同元素求和以及取最大分数，lisi的权重为2，wang的权重为1

```
redis 127.0.0.1:6379> zinterstore result 2 lisi wang weights 2 1 aggregate max
(integer) 3
redis 127.0.0.1:6379> zrange result 0 -1 withscores
1) "cat"
2) "6"
3) "dog"
4) "10"
5) "horse"
6) "12"
redis 127.0.0.1:6379> zinterstore result 2 lisi wang weights 2 1 aggregate sum
(integer) 3
redis 127.0.0.1:6379> zrange result 0 -1 withscores
1) "cat"
2) "8"
3) "dog"
4) "16"
5) "horse"
6) "20"
redis 127.0.0.1:6379>
```

- 先将两个集合个元素的分数乘以权重，即lisi中cat分数变为6，dog变为10，horse变为12，而wang里元素的分数均不变

3.3.5 ZSET（有序集合）

- ZUNIONSTORE命令

ZUNIONSTORE destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM | MIN | MAX]

用于计算给定的一个或多个有序集合key的并集，其中给定key的数量必须等于numkeys，并将计算的并集结果存入destination中

- 使用WEIGHTS选项来为每个给定的有序集合分别指定一个乘数，每个给定的有序集合中的所有元素的score值在传递给聚合函数之前都会乘以这个乘数（weight）
- 如果没有指定WEIGHTS选项，则乘数默认设置为1

3.3.5 ZSET（有序集合）

- Sorted Set相关命令

ZADD
ZREM
ZREMRANGEBYSCORE
ZREMRANGEBYRANK
ZRANK
ZREVRANK
ZRANGE
ZREVRANGE
ZRANGEBYSCORE
ZCARD
ZCOUNT
ZINTERSTORE/
ZUNIONSTORE

3.3 Redis数据类型及操作

- 主要内容

3.3.1 通用命令

3.3.2 STRING（字符串）

3.3.3 LIST（列表）

3.3.4 SET（无序集合）

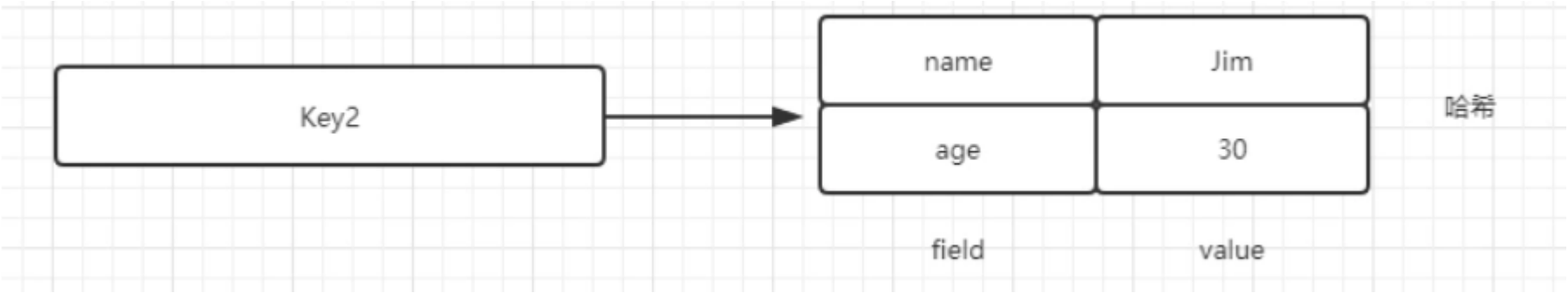
3.3.5 ZSET（有序集合）

3.3.6 HASH（哈希表）

3.3.6 HASH（哈希表）

◆ HASH类型的相关命令

Hash类型是一个String类型的域（field）和值（value）的映射表，Hash数据类型常常用来存储对象信息



3.3.6 HASH（哈希表）

- HSET命令

HSET key field value

使用HSET命令将哈希表key中的field（域）的值设置为value

key不存在时，将会创建一个新的哈希表并进行HSET操作

- 操作成功了，则将会返回1
- 已经存在field，那么在新值覆盖旧值后，将会返回0
  - 例：创建一个哈希表user1，其中，域name为lisi，age为28，height为175

```
redis 127.0.0.1:6379> flushdb
OK
redis 127.0.0.1:6379> hset user1 name lisi
(integer) 1
redis 127.0.0.1:6379> hset user1 age 28
(integer) 1
redis 127.0.0.1:6379> hset user1 height 175
(integer) 1
redis 127.0.0.1:6379>
```

3.3.6 HASH（哈希表）

- HGETALL命令

HGETALL key

取哈希表key中所有的field和value

```
redis 127.0.0.1:6379> hset user1 name lisi
(integer) 1
redis 127.0.0.1:6379> hset user1 age 28
(integer) 1
redis 127.0.0.1:6379> hset user1 height 175
(integer) 1
redis 127.0.0.1:6379> hgetall user1
1) "name"
2) "lisi"
3) "age"
4) "28"
5) "height"
6) "175"
```

3.3.6 HASH（哈希表）

- HMSET命令

HMSET key field value [field value ...]

将一个或多个域-值（field-value）对设置到哈希表key中，执行该命令后，将会覆盖哈希表key中原有的域。

```
redis 127.0.0.1:6379> hmset user2 name wang age 10 height 100
OK
redis 127.0.0.1:6379> hgetall user2
1) "name"
2) "wang"
3) "age"
4) "10"
5) "height"
6) "100"
redis 127.0.0.1:6379>
```

3.3.6 HASH（哈希表）

- HGET命令

HGET key field

获取哈希表key中field的值

```
redis 127.0.0.1:6379> hgetall user1
1) "name"
2) "lisi"
3) "age"
4) "28"
5) "height"
6) "175"
redis 127.0.0.1:6379> hmset user2 name wang age 10 height 100
OK
redis 127.0.0.1:6379> hgetall user2
1) "name"
2) "wang"
3) "age"
4) "10"
5) "height"
6) "100"
redis 127.0.0.1:6379> hget user1 name
"lisi"
redis 127.0.0.1:6379> hget user2 age
"10"
redis 127.0.0.1:6379>
```

3.3.6 HASH（哈希表）

- HMGET命令

HMGET key field [field ...]

获取哈希表key中一个或多个field的值，返回一个包含多个指定域（field）的关联值的表，表中值的顺序与给定域参数的请求顺序保持一致

```
redis 127.0.0.1:6379> hgetall user1
1) "name"
2) "lisi"
3) "age"
4) "28"
5) "height"
6) "175"
redis 127.0.0.1:6379> hmset user2 name wang age 10 height 100
OK
redis 127.0.0.1:6379> hgetall user2
1) "name"
2) "wang"
3) "age"
4) "10"
5) "height"
6) "100"
redis 127.0.0.1:6379> hmget user1 name height
1) "lisi"
2) "175"
```

3.3.6 HASH（哈希表）

- HDEL命令

HDEL key field [field ...]

HDEL命令用于删除哈希表key中的一个或多个指定域（field），它会忽略不存在的域，返回被删除的域的数量

例：删除user1中的域age

```
redis 127.0.0.1:6379> hgetall user1
1) "name"
2) "lisi"
3) "age"
4) "28"
5) "height"
6) "175"
redis 127.0.0.1:6379> hdel user1 age
(integer) 1
redis 127.0.0.1:6379> hgetall user1
1) "name"
2) "lisi"
3) "height"
4) "175"
redis 127.0.0.1:6379>
```

3.3.6 HASH（哈希表）

- HLEN命令

HLEN key

统计哈希表key中域的数量

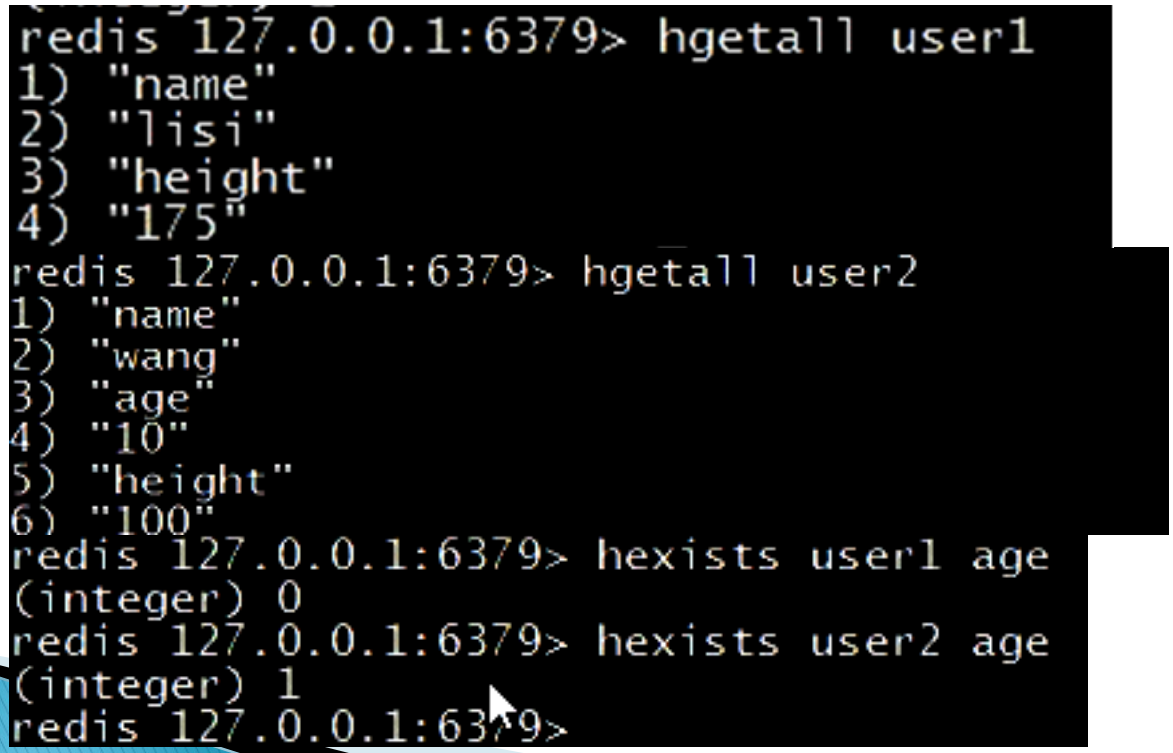
```
redis 127.0.0.1:6379> hgetall user1
1) "name"
2) "lisi"
3) "height"
4) "175"
redis 127.0.0.1:6379> hlen user1
(integer) 2
```

3.3.6 HASH（哈希表）

- HEXISTS命令

HEXISTS key field

判断哈希表key中的field是否存在



3.3.6 HASH（哈希表）

- HINCRBY命令

  HINRBY key field value

  把key中的field域的值增长整型值value

- HINCRBY FLOAT命令

  HINRBY FLOAT key field value

  是把key中的field域的值增长浮点值value

- 例1：将user2中的age域增长1

```
redis 127.0.0.1:6379> hgetall user2
1) "name"
2) "wang"
3) "age"
4) "10"
5) "height"
6) "100"
redis 127.0.0.1:6379> hincrby user2 age 1
(integer) 11
redis 127.0.0.1:6379> hgetall user2
1) "name"
2) "wang"
3) "age"
4) "11"
5) "height"
6) "100"
```

- 例2：将user2中的age域增长0.5

```
redis 127.0.0.1:6379> hgetall user2
1) "name"
2) "wang"
3) "age"
4) "11"
5) "height"
6) "100"
redis 127.0.0.1:6379> hget user2 age
"11"
redis 127.0.0.1:6379> hincrby user2 age 0.5
(error) ERR value is not an integer or out of range
redis 127.0.0.1:6379> hincrbyfloat user2 age 0.5
"11.5"
redis 127.0.0.1:6379>
```

- 0.5不是整数型，所以用hincerby会报错

3.3.6 HASH（哈希表）

- HKEYS命令

HKEYS key

获取哈希表key中的所有域（field）

例：返回user2所有域名

```
redis 127.0.0.1:6379> hkeys user2
1) "name"
2) "age"
3) "height"
redis 127.0.0.1:6379>
```

3.3.6 HASH（哈希表）

- HVALS 命令.

HVALS key

返回哈希表key中所有域的值

例：返回user2所有域的值

```
redis 127.0.0.1:6379> hgetall user2
1) "name"
2) "wang"
3) "age"
4) "11"
5) "height"
6) "100"
redis 127.0.0.1:6379> hget user2 age
"11"
redis 127.0.0.1:6379> hincrby user2 age 0.5
(error) ERR value is not an integer or out of range
redis 127.0.0.1:6379> hincrbyfloat user2 age 0.5
"11.5"
redis 127.0.0.1:6379>
redis 127.0.0.1:6379> hvals user2
1) "wang"
2) "11.5"
3) "100"
redis 127.0.0.1:6379>
```

总结

- Sorted Set相关命令

ZADD
ZREM
ZREMRANGEBYSCORE
ZREMRANGEBYRANK
ZRANK
ZREVRANK
ZRANGE
ZREVRANGE
ZRANGEBYSCORE
ZCARD
ZCOUNT
ZINTERSTORE/
ZUNIONSTORE

- HASH类型的相关命令

HSET
HMSET
HGET
HMGET
HGETALL
HDEL
HLEN
HEXISTS
HINCRBY
HINCRBY FLOAT
HKEYS
HVALS