# 存储sortedeset

## 概述

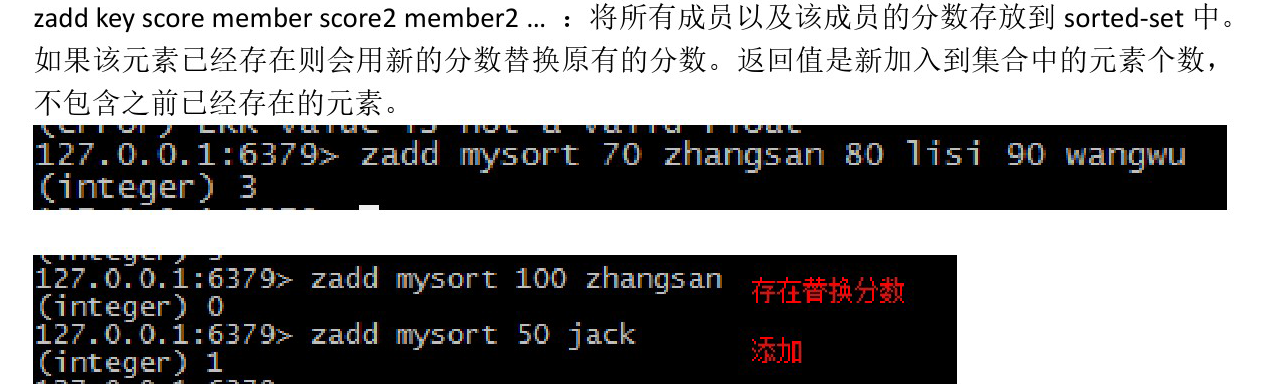
Sorted-Sets和Sets类型极为相似，它们都是字符串的集合，都不允许重复的成员出现在一个Set中。它们之间的主要差别是Sorted-Sets中的每一个成员都会有一个分 数(score)与之关联，Redis正是通过分数来为集合中的成员进行从小到大的排序。然 而需要额外指出的是，尽管Sorted-Sets中的成员必须是唯一的，但是分数(score) 却是可以重复的。

在Sorted-Set中添加、删除或更新一个成员都是非常快速的操作，其时间复杂度为 集合中成员数量的对数。由于Sorted-Sets中的成员在集合中的位置是有序的，因此，即便是访问位于集合中部的成员也仍然是非常高效的。事实上，Redis所具有的这一特征在很多其它类型的数据库中是很难实现的，换句话说，在该点上要想达到和Redis同样的高效，在其它数据库中进行建模是非常困难的。

例如：游戏排名、微博热点话题等使用场景。

## 常用命令

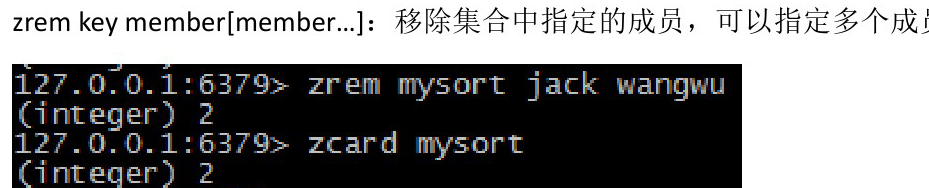
### 添加元素(zadd)



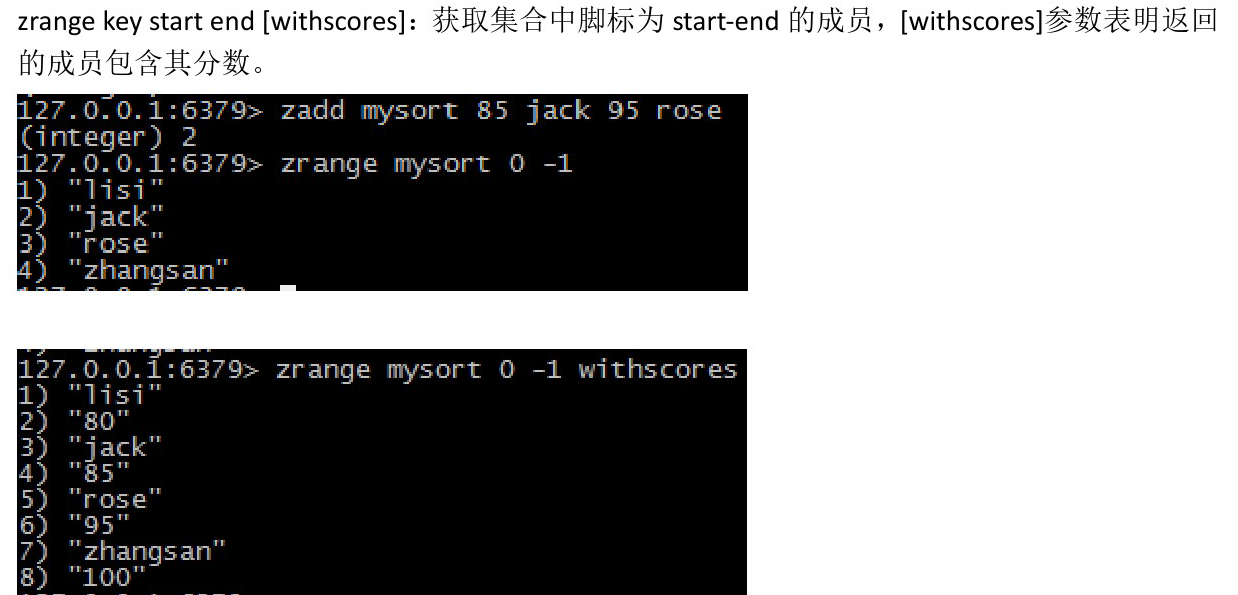
### 获得元素(zscore,zcard)

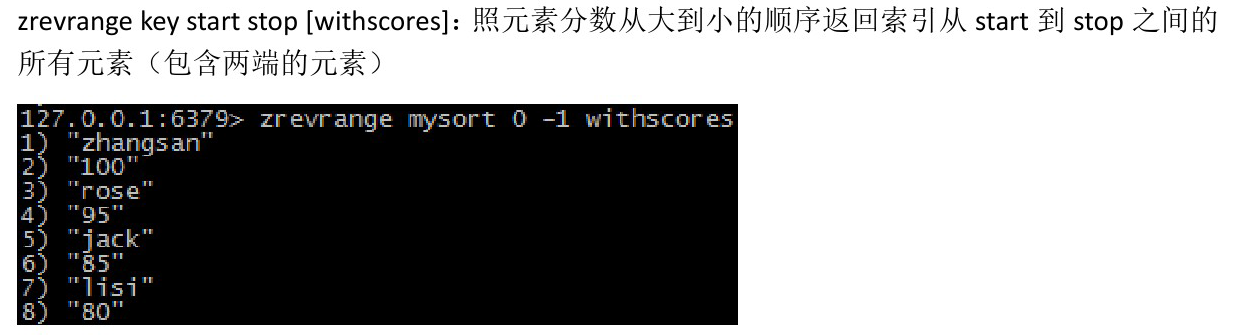


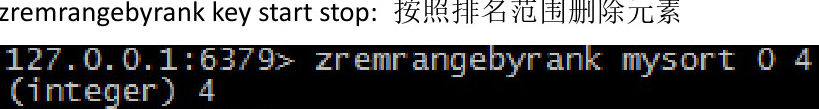
### 删除元素(zrem)



### 范围查询(zrange,zrevrange,zremrangebyrank, zremrangebyscope)



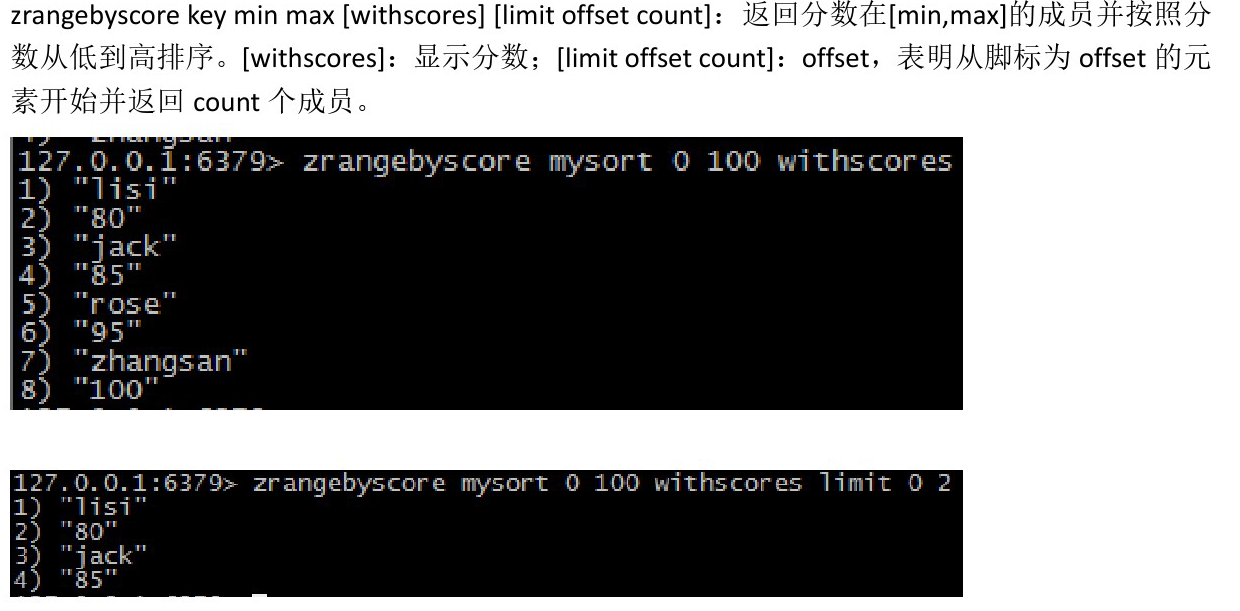




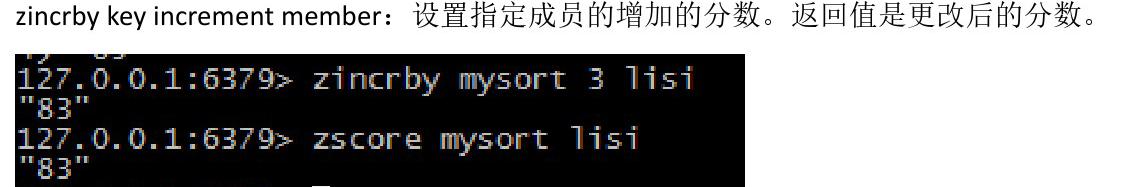


## 扩展命令

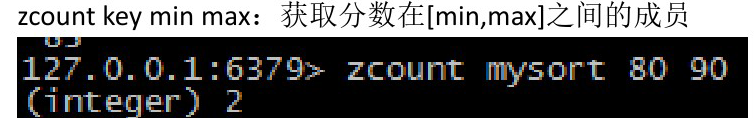
### 分数排序(zrangebyscope)



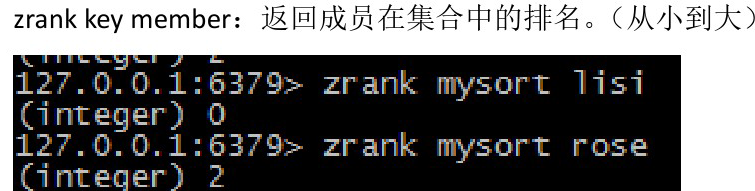
### zincrby



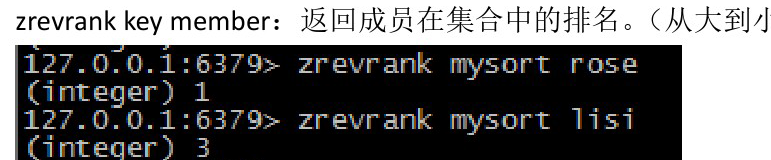
### zcount



### zrank



### zrevrank



# 使用场景

