# 字典

字典，又称为符号表（ symbol table ）、关联数组（ associative array ）或映射（ map ），是一种用于保存键值对（ key-value pair ）的抽象数据结构。在字典中，一个键（ key ）可以和一个值（ value ）进行关联（或者说将键映射为值），这些关联的键和值就称为键值对。

字典中的每个键都是独一无二的，程序可以在字典中根据键查找与之关联的值，或者通过键来更新值，又或者根据键来删除整个键值对，等等。

字典经常作为一种数据结构内置在很多高级编程语言里面，但Redis 所使用的C 语言并没有内置这种数据结构，因此Redis 构建了自己的字典实现。

字典在Redis 中的应用相当广泛，比如Redis 的数据库就是使用字典来作为底层实现的，对数据库的增、删、查、政操作也是构建在对字典的操作之上的。

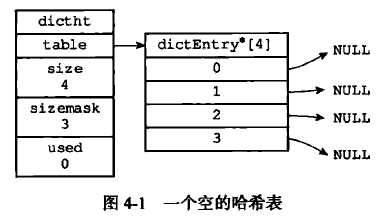
除了用来表示数据库之外，字典还是晗希键的底层实现之一，当一个晗希键包含的键值对比较多，又或者键值对中的元素都是比较长的字符串时， Redis 就会使用字典作为晗希键的底层实现。

## 字典的实现

Redis 字典所使用的晗希表由dict.h/dictht 结构定义：

|  |
| --- |
| typedef struct dictht {  // 哈希表数组  dictEntry \*\*table;  // 哈希表大小  unsigned long size;  // 哈希表大小掩蜀，用于计算索引值  // 总是等于size-1  unsigned long sizemask;  // 该哈希表已有节点的数量  unsigned long used;  } dictht; |

table 属性是一个数组，数组中的每个元素都是一个指向dict.h/dictEntry 结构的指针，每个dictEntry 结构保存着一个键值对。size 属性记录了晗希表的大小，也即是table 数组的大小，而used 属性则记录了哈希表目前已有节点（键值对）的数量。sizemask 属性的值总是等于size-1 ，这个属性和晗希值一起决定一个键应该被放到table 数组的哪个索引上面。图4-1 展示了一个大小为4 的空哈希表（没有包含任何键值对）。



## 哈希表节点