1. Sorted sets是将set中的元素增加了一个权重参数score，使得集合中的元素能够按score进行有序排列
2. 添加：
   1. AddItemToSet 只添加setid和value
   2. AddItemToSortedSet方法添加key和value都相同的数据时，后者会覆盖前者的值（利用此方法添加的时候） 当不写score值的时候，默认值自增（需要在value前加上1.、2.等）
   3. IncrementItemInSortedSet 方法添加key和vlaue都相同的数据时，后者会在前者的基础上进行计算，然后得出最终的值（利用此方法添加的时候）
   4. 两种方法中，如果都没有找到相同的key和value，那么就进行添加

public static void Add()

{

client.AddItemToSortedSet("2年级", "zxl\_l", 1);

//for (int i = 2; i < 11; i++)

//{

// client.AddItemToSortedSet("一年级", "zxl\_" + i, i);

//}

//会覆盖前面的score

client.AddItemToSortedSet("2年级", "zxl\_l", 200);

//1.1 如果没有这个value 会默认添加进去

//var dd1 = client.IncrementItemInSortedSet("一年级", "zx9", 100);

//var dd2 = client.IncrementItemInSortedSet("一年级", "zx10", -50);

* 1. }

1. 删除

public static void Delete()

{

//删除setid集合中删除最高分

//client.PopItemWithHighestScoreFromSortedSet("一年级");

//setid集合中删除setid中指定value

//client.RemoveItemFromSortedSet("一年级","zxl\_4");

//setid集合中删除下标从1到2的数据

client.RemoveRangeFromSortedSet("一年级",1,2);

//setid集合中删除分数从fromscore到toscore的setid集合数据

// client.RemoveRangeFromSortedSetByScore("一年级",9,9);

1. }
2. 查询

/// <summary>

/// 升序获取

/// </summary>

public static void Select()

{

//升序获取setid为一年级的value值

var dd3 = client.GetAllItemsFromSortedSet("一年级");

//降序获取

var dd4 = client.GetAllItemsFromSortedSetDesc("一年级");

//获取排名前4的value值

var dd5 = client.GetRangeFromSortedSet("一年级",0,3); //不带分数

var dd6 = client.GetRangeWithScoresFromSortedSet("一年级", 0, 3);//带分数

//降序获取排名前4的值

var dd7 = client.GetRangeFromSortedSetDesc("一年级", 0, 3);//不带分数

var dd8 = client.GetRangeWithScoresFromSortedSetDesc("一年级", 0, 3);//带分数

//降序获取分数在【2,6】之内的值

var dd9 = client.GetRangeFromSortedSetByHighestScore("一年级",2,6);

var dd10 = client.GetRangeFromSortedSetByLowestScore("一年级",2,6);

//获取指定setid集合中zxl\_2值的分数

var dd11 = client.GetItemScoreInSortedSet("一年级","zxl\_2");

//获取集合的数量

var dd12 = client.GetSortedSetCount("一年级");

* 1. }
  2. 交差并补

//获取交集（两个集合中共有的元素）

HashSet<string> data1=client.GetIntersectFromSets("刘备", "张飞");

//获取并集（两个集合中全部的元素）

HashSet<string> data2 = client.GetUnionFromSets("刘备", "张飞");

//获取差集（两个集合没有的）

HashSet<string> data3 = client.GetDifferencesFromSet("刘备","张飞");

//补集（父子集合中，父有，子没有）

client.StoreDifferencesFromSet("关羽", "刘备", "张飞");

HashSet<string> data4 = client.GetAllItemsFromSet("关羽");

1. 修改
   1. //var dd1 = client.IncrementItemInSortedSet("一年级", "zx9", 100);
   2. Setid和value相同 然后修改值