https://www.cnblogs.com/mushroom/p/4659206.html

[Redis系列(六)-SortedSets设计技巧](https://www.cnblogs.com/mushroom/p/4659206.html)

**阅读目录:**

1. [介绍](https://www.cnblogs.com/mushroom/p/4659206.html#one)
2. [Score占位](https://www.cnblogs.com/mushroom/p/4659206.html#two)
3. [更多位信息](https://www.cnblogs.com/mushroom/p/4659206.html#three)
4. [总结](https://www.cnblogs.com/mushroom/p/4659206.html#FOUR)

**介绍**

Redis Sorted Sets是类似Redis Sets数据结构，不允许重复项的String集合。不同的是Sorted Sets中的每个成员都分配了一个分数值(score)，它用于在Sorted Sets中进行成员排序，从最小值到最大值。Sorted Sets中所有的成员都是唯一的，其分数(score)是可以重复的，即是说一个分数可能会对应多个值。

用Sorted Sets可以非常快的进行添加、删除、或更新成员，其复杂度是O(m\*log(n))，m是添加或查询的成员数量。因为成员是按照顺序添加的，所以可以非常快的通过score或者索引进行范围查询。访问Sorted Sets中间的元素也是非常快的，因此可以用sort sets作为一个不重复的小型有序列表。 通过Sorted Sets可以快速操作任何你想做的事情：排序的成员，判断成员是否在集合中，快速访问集合中间的成员。

如果多次添加相同的值到Sorted Sets上，redis会以最后一次的值分数为准。

总的来说，在其他数据库比较难完成的任务，用Sorted Sets可以更快更优性能的完成。

更多Sorted Sets的用法可以查看官方文档。

**Score占位**

Sorted Sets中的Score是个64位整数类型，其范围在-9007199254740992到9007199254740992之间，这是一个非常有用的关键点。

我们通常可以用它进行简单的范围查询，比如把年龄当分数，通过ZRANGEBYSCORE查询某个年龄段的所有用户。  
查询18-35年龄的用户：

ZRANGEBYSCORE user 18 35

如果有更复杂的需求，比如通过日期，用户类型等来查询，就比较困难了。 如果用其他数据结构的话，想查询多条时就要多次往返操作，有性能上的损耗。 这里可以用Score整数类型的长度特性，来存储查询的条件信息。

比如有些数据需要通过日期范围查询，这是个很常见的场景。

使用Sorted Sets当存储的数据结构，添加key时把日期进行简单的转码。

一年有365天，在score中预留三位来存储天份，通过DayOfYear计算出一年第多少天：

var time1=DateTime.Now.DayOfYear

因为第100天前的数据不满三位，需要进行用0补全，写个方法：

[复制代码](javascript:void(0);)

string AutoCompletion(int length, int num)

{

string str = num.ToString();

if (str.Length > length)

throw new Exception("max length");

var real = length - str.Length;

var sb = new StringBuilder(length);

for (int j = 0; j < real; j++)

{

sb.Append((char) 48);

}

sb.Append(str);

return sb.ToString();

}

[复制代码](javascript:void(0);)

因为数据的日期可能跨年，在天位前预留2位，存储年份。现在长度是5位，如15200表示的是15年19号(第200天)。

通过年份+天数计算出当前数据项score的值。

var score= long.Parse(dateTime.Year.ToString().Substring(2, 2) + AutoCompletion(3, dateTime.DayOfYear));

把实际值和分数值存储到redis里面：

ZADD test socre value

查询时把需要查询的日期范围转换到分数,这里查询前10天的数据:

[复制代码](javascript:void(0);)

int time1 = DateTime.Now.DayOfYear;

var score1 = AutoCompletion(time1, 2) + AutoCompletion(DateTime.Now.DayOfYear, 3);

int time2 = DateTime.Now.DayOfYear;

var score2 = AutoCompletion(time1, 2) + AutoCompletion(DateTime.Now.AddDays(-10).ToLocalTime().DayOfYear, 3);

[复制代码](javascript:void(0);)

在redis中查询：

ZRANGEBYSCORE user score1 score2

查询当天(分数)下的信息：

ZRANGEBYSCORE user score2 score2

**更多位信息**

r64位整数类型有16个位置，意味着还可以存储更多维度信息。这里以日期为主，在增加个类型维度，日期位置左移4位，预留4位来存储类型维度。

如152003456，其3456四位是存储类型的，可以根据类型种类的多少可以增加位或减少位占用。

如果单纯以日期来查询时，类型不用传，后面4位在查询时补千位最小值和最大值即可。

查询前10天的数据：

ZRANGEBYSCORE user 151900000 152009999

查询15年19号3578类型的数据：

ZRANGEBYSCORE user 152003578 152003578

查询15年19号1000-4000类型的数据：

ZRANGEBYSCORE user 152001000 152004000

关于类型往往不是简单的数据类型那么简单，可以做个简单的mapping表。

abcd 1000

dcda 1001

以此类推...

**在添加时或查询时注意转换。**

**总结**

需要注意的是score是整数类型按从小到大排序存储的，按照上面的设计是不能在指定类型的情况下，进行日期范围查找的，因为我们定义日期作为主维度。

通常在数据库中有很多层级关系的表，比如大分类、子分类、小分类等一级一级递减。 使用Sorted Sets我们可以模拟这种关系去设计Score值，主维度->次维度->低维度，这样就可以更快的进行范围分类信息的查询，有个前提是64位整数的位置够用。

如果没有类似的需求，可以使用时间戳作为值的分数值，这样就不需要额外的转换了。

简单分享了Sorted Sets存储设计的经验，希望对大家有所帮助。