解锁SQL无限可能｜如何利用SQL求解力扣难题接雨水问题?

原创 江月明203 会飞的一十六 2024年08月27日 21:37 重庆

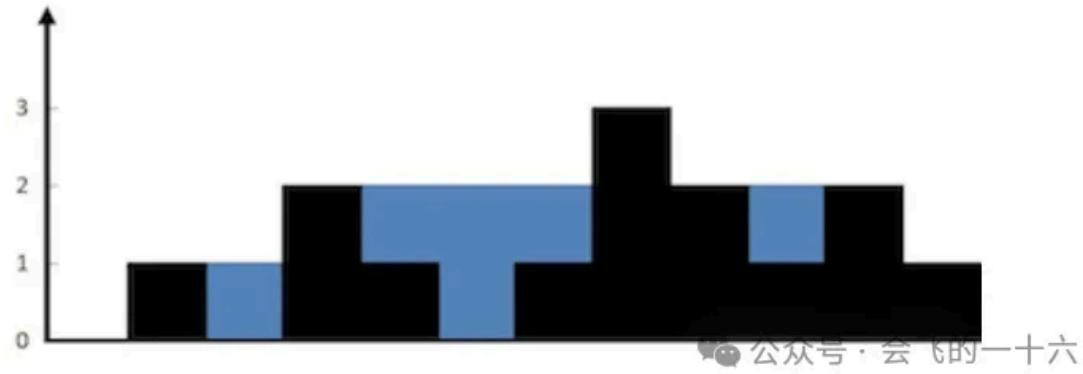
“本问题由oracle大神苏旭辉老师提出，其oracle解法及链接如下
趣味SQL题：力扣 42 之接雨水 – Oracle开发 – ITPUB论坛－专业的IT技术社区。该题目属于趣味题目，感兴趣的同学可以研究一下”

01

题目描述

力扣原文链接：42. 接雨水 – 力扣（LeetCode）
给定 n 个非负整数表示每个宽度为 1 的柱子的高度图，计算按此排列的柱子，下雨之后能接多少雨水。

示例 1：



输入：height = [0,1,0,2,1,0,1,3,2,1,2,1]
输出：6
解释：上面是由数组 [0,1,0,2,1,0,1,3,2,1,2,1] 表示的高度图，在这种情况下，可以接 6 个单位的雨水（蓝色部分表示雨水）。
示例 2：

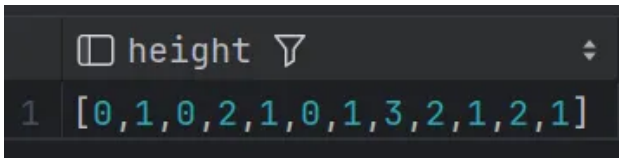
输入：height = [4,2,0,3,2,5]
输出：9

02

数据准备

```
1 with rain_data as
```

```
(
  select array(0,1,0,2,1,0,1,3,2,1,2,1) height
)
select * from rain_data;
```



03

问题分析

本文为力扣第42题，属于hard级别。分析该问题的关键点在于"找规律，抓特征"，这也是我专栏里一直提的核心技巧，语言不重要，重要的是思维方法，是算法逻辑，其余的工作只是利用语言去翻译，这一点对SQL语言尤为重要，因为语法简单的语言在实现上一定要重思维，重逻辑。

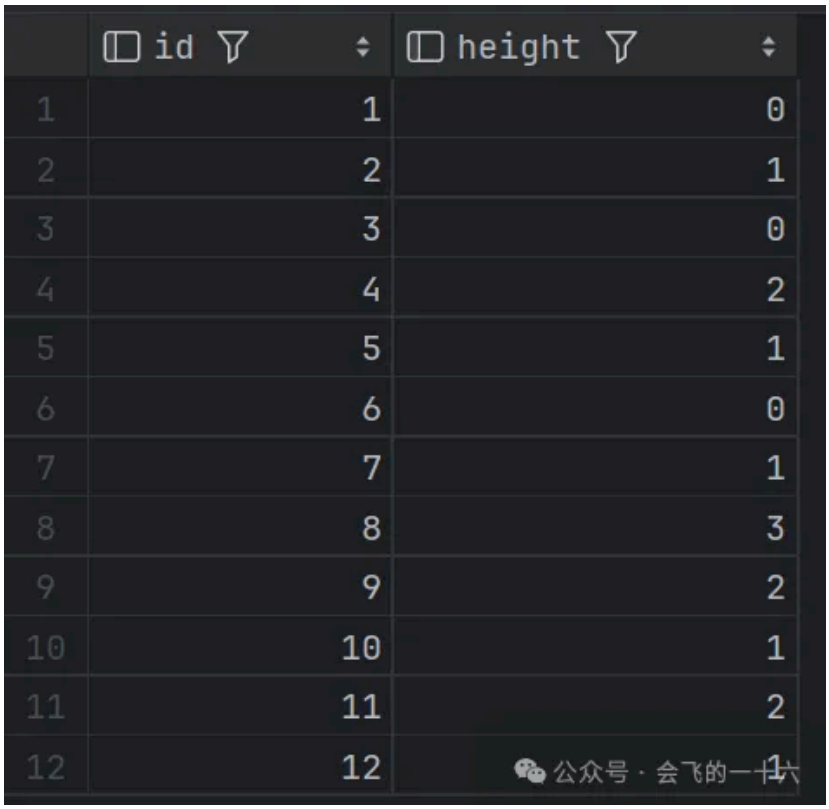
这道题其实就是“木桶原理”，简单的说就是一个柱子能接多少水，取决于它两边“较短的板”。另外一个前提条件就是，两边的柱子高度都要比所要装水的柱子的高度要高，否则肯定是无法装水的。因此我们只需要关注左边最高的木板和右边最高的模板中较矮的一个就够了，那么存储的水，等于两边木板的较小值减去当前高度的值。用公式表示如下：

```
res[i] = min(l_max[i], r_max[i]) - height[i];
```

我们将上述公式翻译成SQL语言即可：

第一步：先将数组展开成行

```
select tmp.pos + 1 id
          , tmp.height  height
     from rain_data
              lateral view posexplode(height) tmp as pos, height
```



	id	height
1	1	0
2	2	1
3	3	0
4	4	2
5	5	1
6	6	0
7	7	1
8	8	3
9	9	2
10	10	1
11	11	2
12	12	1

公众号 · 会飞的一十六

第二步：求截止当前位置处，左最大与右最大

```sql
select  id
    , height
    , max(height) over(order by id) l_max
    , max(height) over(order by id desc) r_max
from
    (select tmp.pos + 1 id
          , tmp.height  height
     from rain_data
              lateral view posexplode(height) tmp as pos, height
     )
        t;
```

第三步：利用公式进行标记

```sql
select id
      ,height
      ,least(l_max,r_max) - height     as rain_flg
from
(select id
      , height
      , max(height) over (order by id)      l_max
      , max(height) over (order by id desc) r_max
 from (select tmp.pos + 1 id
            , tmp.height  height
       from rain_data
```

```
                lateral view posexplode(height) tmp as pos, height) t
order by id
```

	id	height	rain_flg
1	1	0	0
2	2	1	0
3	3	0	1
4	4	2	0
5	5	1	1
6	6	0	2
7	7	1	1
8	8	3	0
9	9	2	0
10	10	1	1
11	11	2	0
12	12	1	0

公众号 · 会飞的一十六

第四步：求出最终结果值。完整的SQL如下：

```
select sum(rain_flg)
from
(select least(max(tmp.height) over (order by id), max(tmp.height) ove
 from rain_data r
          lateral view posexplode(r.height) tmp as id, height
 ) t
```

	_c0
1	6

04
小结

关于本题，作者进行了多次探索，最终能想到利用SQL解决的即文中所述，若采用代码语言上述的方法属于暴力求解，需要优化其两边搜所的效率，因此就会有双指针和动态规划的求解思路。但对于本文一开始而言找出”木桶原理”的特征（**res[i] = min(l_max[i], r_max[i]) - height[i]**），也不是一件容易得事情，需要一定问题的分析能力。有兴趣的读者可以尝试是否可以利用单调栈的方法，利用SQL语言求解出本题。