

LeetCode 178

题目

Write a SQL query to rank scores. If there is a tie between two scores, both should have the same ranking. Note that after a tie, the next ranking number should be the next consecutive integer value. In other words, there should be no "holes" between ranks.

```
+-----+-----+
| Id | Score |
+-----+-----+
| 1 | 3.50 |
| 2 | 3.65 |
| 3 | 4.00 |
| 4 | 3.85 |
| 5 | 4.00 |
| 6 | 3.65 |
+-----+-----+
```

For example, given the above Scores table, your query should generate the following report (order by highest score):

```
+-----+-----+
| Score | Rank |
+-----+-----+
| 4.00 | 1 |
| 4.00 | 1 |
| 3.85 | 2 |
| 3.65 | 3 |
| 3.65 | 3 |
| 3.50 | 4 |
+-----+-----+
```

题目大意

编写一个SQL查询来评分。如果两个分数之间有一个平分，那么两个分数应该是相同的。请注意，平局之后，下一个排名数应该是下一个连续的整数值。换句话说，行列之间不应该有“漏洞”。

解题思路

用查询编写一个成绩单

如何生成这个rank字段是本题的重点

换一个角度,其实rank也就是这个之前大于等于本成绩的成绩个数

```
SELECT Score,  
(SELECT COUNT(DISTINCT Score) FROM Scores WHERE Score >= s.Score) Rank  
FROM Scores s ORDER BY Score DESC;
```

做这题的时候卡在这里挺长时间,于是搜集了一下网上能找到的所有答案

答案1

```
SELECT Score,  
(SELECT COUNT(*) FROM (SELECT DISTINCT Score s FROM Scores) t WHERE s >= Score) Rank  
FROM Scores ORDER BY Score DESC;
```

与最上面的解法差别不是很大,这里用到了一个去重之后的分数临时表来进行查询,相比最开始的每一行都查询本身表并去重速度得到了提升

答案2

```
SELECT s.Score, COUNT(DISTINCT t.Score) Rank  
FROM Scores s JOIN Scores t ON s.Score <= t.Score  
GROUP BY s.Id ORDER BY s.Score DESC;
```

下面这种解法使用了内交, Join是Inner Join的简写形式, 自己和自己内交, 条件是右表的分数大于等于左表, 然后群组起来根据分数的降序排列, 十分巧妙的解法

SQL 用户定义变量

若是使用编程语言解此题很容易想到使用循环变量不断判断大小并自家1的形式.来解决此题.但是如何在一个sql语句保存赋值一个自定义的变量呢?

```
SELECT Score,  
@rank := @rank + (@pre <> (@pre := Score)) Rank  
FROM Scores, (SELECT @rank := 0, @pre := -1) INIT  
ORDER BY Score DESC;
```

下面这种解法跟上面三种的画风就不太一样了, 这里用了两个变量, 变量使用时其前面需要加@, 这里的:= 是赋值的意思, 如果前面有Set关键字, 则可以直接用=号来赋值, 如果没有, 则必须要使用:=来赋值, 两个变量rank和pre, 其中rank表示当前的排名, pre表示之前的分数, 下面代码中的<>表示不等于, 如果左右两边不相等, 则返回true或1, 若相等, 则返回false或0。初始化rank为0, pre为-1, 然后按降序排列分数, 对于分数4来说, pre赋为4, 和之前的pre值-1不同, 所以rank要加1, 那么分数4的rank就为1, 下面一个分数还是4, 那么pre赋值为4和之前的4相同, 所以rank要加0, 所以这个分数4的rank也是1, 以此类推就可以计算出所有分数的rank了。

sql中 (SELECT @rank := 0, @pre := -1) INIT 是为这两个变量赋初值

