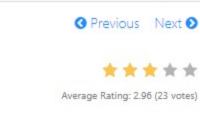
569. Median Employee Salary 🗗

May 13, 2017 | 20.6K views

any built-in SQL functions.



The Employee table holds all employees. The employee table has three columns: Employee Id, Company Name, and Salary.

```
Salary
       Company
  1
        Α
                      2341
  2
                      341
  |3
                    15
  4
                     15314
  5
                     451
  6
                    513
  7
                    15
         В
  8
         В
                     13
  9
        В
                    1154
                    1345
  11
                    1221
  12
       В
                      234
  13
       C
                     2345
  14
                      2645
  15
       C
                      2645
  16
       C
                      2652
  17
       C
                      65
Write a SQL query to find the median salary of each company. Bonus points if you can solve it without using
```

Company Salary

9 B 1154
14 C 2645

Approach #1: Using the definition of median [Accepted]

Solution

By the definition of median, the count of the bigger numbers than itself should be equal to the count of the smaller ones in an odd array.

Intuition

Algorithm Take array [1,3,2] for example, is the first number 1 the median? No, because this array only have 3 elements but there are 2 of them (3, 2) are greater than 1. To continue, we know 3 is not the median as well since there

are 2 elements smaller. But for the last element 2, there are equal amount of bigger and smaller numbers. So it is the median in this array!

What if an array has even amount of distinct values, the median is the average of the middle two elements next to each other after sorting this array. It is not hard to understand that for either of these two elements, the difference (absolute value) of its bigger and smaller number than itself in this array is 1, which is the exactly frequency of a element in the distinct array.

So in general, the median's frequency should be equal or grater than the absolute difference of its bigger elements and small ones in an array no matter whether it has odd or even amount of numbers and whether

they are distinct. This rule is the key, and it is represented as the following code.

SUM(CASE WHEN Employee.Salary = alias.Salary THEN 1 ELSE 0 END) >= ABS(SUM(SIGN(Employee.Salary - alias.Salary)))

```
Thus, this approach is as following in MySQL code.
MySQL
  SELECT
```

Employee.Id, Employee.Company, Employee.Salary FROM

Employee,

```
Employee alias
WHERE
    Employee.Company = alias.Company
GROUP BY Employee.Company , Employee.Salary
HAVING SUM(CASE
    WHEN Employee. Salary = alias. Salary THEN 1
END) >= ABS(SUM(SIGN(Employee.Salary - alias.Salary)))
ORDER BY Employee.Id
   Note: In MySQL 5.6, this code runs fine, but if you are using MySQL 5.7+, please use
   ANY_VALUE(Employee.Id) instead of Employee.Id in the SELECT statement. Otherwise, you may
   encouter the following error message: Error Code: 1055. Expression #1 of SELECT list is not in GROUP
   BY clause and contains nonaggregated column 'Employee.Id' which is not functionally dependent on
```

Approach #2: Sort and then select the *median* [Accepted] Intuition In general, we can just pick the middle one(s) to get the median if the records are ranked by salary. But how can we get them sorted particularly MySQL does not have the build-in rank function, and these are several

columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by. For more

By adding a virtual column to simulate the ranking, we can sort these records by **salary** and pick up the middle one(s). Here we need to use the session variable to achieve this goal.

details on how to user ANY_VALUE(arg), please refer to this link.

FROM

(SELECT

e.Id,

e.Salary, e.Company,

Preview

Algorithm

companies in this case.

table. SELECT Id, Company, Salary

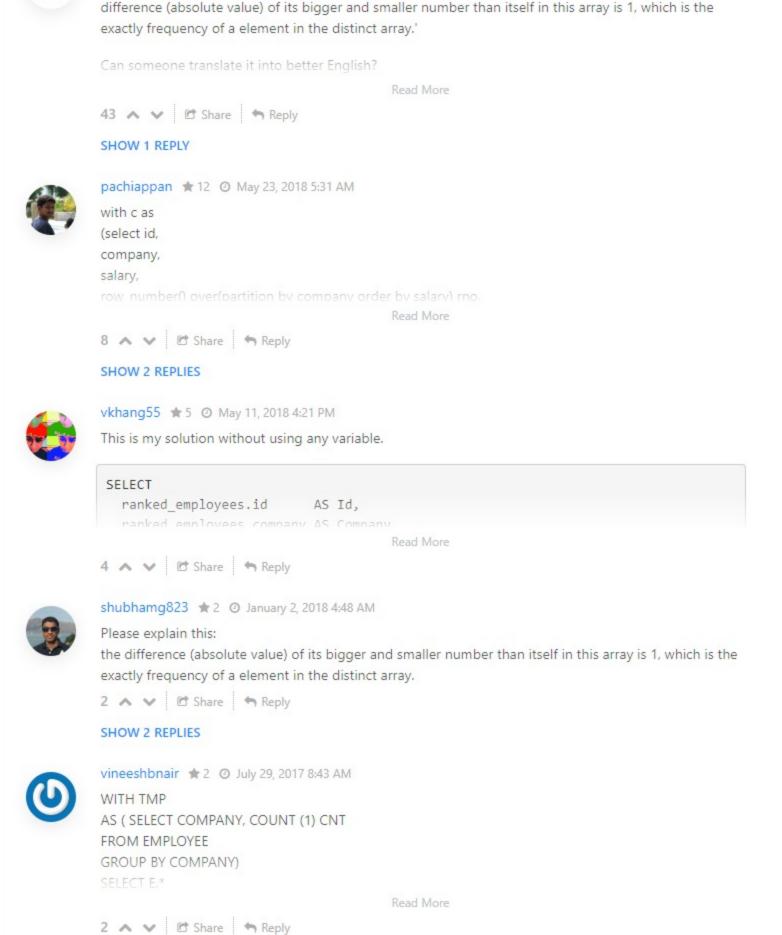
IF(@prev = e.Company, @Rank:=@Rank + 1, @Rank:=1) AS rank,

This approach is more efficient than the first one since it does not calculate all the salary one by one in the

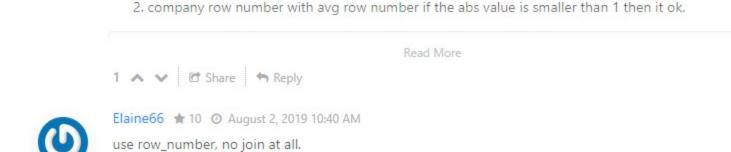
```
@prev:=e.Company
      FROM
          Employee e, (SELECT @Rank:=0, @prev:=0) AS temp
      ORDER BY e.Company , e.Salary , e.Id) Ranking
          INNER JOIN
      (SELECT
          COUNT(*) AS totalcount, Company AS name
      FROM
          Employee e2
      GROUP BY e2. Company) companycount ON companycount.name = Ranking.Company
  WHERE
      Rank = FLOOR((totalcount + 1) / 2)
          OR Rank = FLOOR((totalcount + 2) / 2)
Rate this article: * * * * *
 O Previous
                                                                                    Next 0
Comments: 35
                                                                                   Sort By ▼
           Type comment here... (Markdown is supported)
```

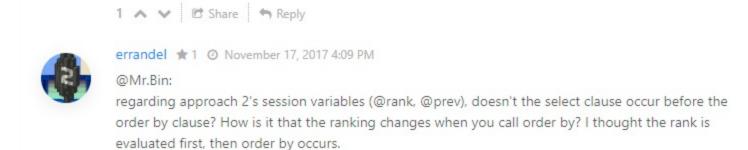
I can't understand this statement 'It is not hard to understand that for either of these two elements, the

Post









Read More



Read More

(1234)

1 A V C Share Reply

My solution:

rank and have row number.

select id, company, salary

(select Employee,,