Previous Next

185. Department Top Three Salaries

July 11, 2017 | 79.4K views

★★★★☆
Average Rating: 4.16 (38 votes)

The `Employee` table holds all employees. Every employee has an Id, and there is also a column for the department Id.

```
+----+-------+--------+--------------+
| Id | Name  | Salary | DepartmentId |
+----+-------+--------+--------------+
| 1  | Joe   | 85000  | 1            |
| 2  | Henry | 80000  | 2            |
| 3  | Sam   | 60000  | 2            |
| 4  | Max   | 90000  | 1            |
| 5  | Janet | 69000  | 1            |
| 6  | Randy | 85000  | 1            |
| 7  | Will  | 70000  | 1            |
+----+-------+--------+--------------+
```

The `Department` table holds all departments of the company.

```
+----+----------+
| Id | Name     |
+----+----------+
| 1  | IT       |
| 2  | Sales    |
+----+----------+
```

Write a SQL query to find employees who earn the top three salaries in each of the department. For the above tables, your SQL query should return the following rows (order of rows does not matter).

```
+------------+----------+--------+
| Department | Employee | Salary |
+------------+----------+--------+
| IT         | Max      | 90000  |
| IT         | Randy    | 85000  |
| IT         | Joe      | 85000  |
| IT         | Will     | 70000  |
| Sales      | Henry    | 80000  |
| Sales      | Sam      | 60000  |
+------------+----------+--------+
```

Explanation:

In IT department, Max earns the highest salary, both Randy and Joe earn the second highest salary, and Will earns the third highest salary. There are only two employees in the Sales department, Henry earns the highest salary while Sam earns the second highest salary.

Solution

Approach: Using `JOIN` and sub-query [Accepted]

Algorithm

A top 3 salary in this company means there is no more than 3 salary bigger than itself in the company.

```
select e1.Name as 'Employee', e1.Salary
from Employee e1
where 3 >
(
    select count(distinct e2.Salary)
    from Employee e2
    where e2.Salary > e1.Salary
)
;
```

In this code, we count the salary number of which is bigger than e1.Salary. So the output is as below for the sample data.

```
| Employee | Salary |
|----------|--------|
| Henry    | 80000  |
| Max      | 90000  |
| Randy    | 85000  |
```

Then, we need to join the **Employee** table with **Department** in order to retrieve the department information.

MySQL

```
SELECT
    d.Name AS 'Department', e1.Name AS 'Employee', e1.Salary
FROM
    Employee e1
        JOIN
    Department d ON e1.DepartmentId = d.Id
WHERE
    3 > (SELECT
            COUNT(DISTINCT e2.Salary)
        FROM
            Employee e2
        WHERE
            e2.Salary > e1.Salary
                AND e1.DepartmentId = e2.DepartmentId
        )
;
```

```
| Department | Employee | Salary |
|------------|----------|--------|
| IT         | Joe      | 70000  |
| Sales      | Henry    | 80000  |
| Sales      | Sam      | 60000  |
| IT         | Max      | 90000  |
| IT         | Randy    | 85000  |
```

Rate this article: ★★★★☆

Previous Next

Comments: 77 Sort By

Type comment here... (Markdown is supported)

Preview Post

rongy2018 ★ 184 January 13, 2019 1:52 AM

If you have access to Dense_Rank (), then the following query will work.

```
select d.Name as Department, a. Name as Employee, a. Salary
from (
select e.*, dense_rank() over (partition by DepartmentId order by Salary desc) as
```

Read More

73 Share Reply

SHOW 6 REPLIES

kkzhao ★ 23 December 18, 2018 8:39 PM

The best mySQL solution should be like the one below, without sub-query in where condition. This is more aligned with how SQL works, as we are using a big batch operation instead of many small batches.

select tD.Name as 'Department', tE1.Name as 'Employee', tE1.Salary from Employee as tE1

Read More

23 Share Reply

SHOW 5 REPLIES

gauravtiwari505 ★ 20 August 6, 2018 6:15 PM

Accepted Solution

```
SELECT a.Department, a.Employee, a.Salary FROM
(
Select b.Name as Department, a.Name as Employee, a.salary As Salary
```

Read More

20 Share Reply

SHOW 5 REPLIES

sgv ★ 16 October 11, 2018 2:31 AM

```
Triple join more straight forward

  SELECT
        d.name        as "Department"
```

Read More

10 Share Reply

SHOW 2 REPLIES

satybald ★ 24 April 14, 2018 1:52 PM

Solution based on window functions. Tested on Oracle SQL.

```
with agg as (
select DepartmentId, Name, Salary, dense_rank() over (partition by DepartmentId o
rder by Salary desc) as rank
```

Read More

9 Share Reply

tongzeyud ★ 7 April 20, 2019 11:53 PM

dense_rank() window function is much easier to handle problems like this.
select a.Department,a.Employee,a.Salary
from
(select e.Name as Employee,e.Salary,d.Name as Department, dense_rank() over(partition by d.Name order by Salary desc) as rank

Read More

7 Share Reply

tyumneva ★ 96 July 17, 2018 1:43 AM

select d.Name Department,
e.Name Employee,
e.Salary
from (select
Id,

Read More

7 Share Reply

deepthi93 ★ 12 September 18, 2018 12:34 AM

```
select d.name  "Department",
       e.name "Employee",
       e.Salary,
       row_number() over (partition by departmentid order by salary desc) "rw"
```

Read More

10 Share Reply

SHOW 9 REPLIES

felix-citycs ★ 4 June 11, 2018 10:14 AM

I think there's a problem on the judge engineer...For this test case below it returns top FOUR salaries from ONE department. Please Leetcode take a look into this question?

Requirement:
Write a SQL query to find employees who earn the ***top three salaries in each of the department***. For

Read More

4 Share Reply

SHOW 1 REPLY

venkateshgovind ★ 2 July 6, 2018 4:58 AM

For some reason WITH TIES is not yielding the correct results in MSSQLServer.
/* Write your T-SQL query statement below */
select d.Name as Department, i.Name as Employee, i.Salary
from
Department d

Read More

2 Share Reply

‹ 1 2 3 4 5 6 7 8 ›