## ■ Articles > 615. Average Salary: Departments VS Company ▼

June 16, 2017 | 11.7K views

\* \* \* \* \* \*

Given two tables as below, write a query to display the comparison result (higher/lower/same) of the average salary of employees in a department to the company's average salary.

615. Average Salary: Departments VS Company

Table: salary

id	employee_id	amount	pay_date
1	1	9000	2017-03-31
2	2	6000	2017-03-31
3	3	10000	2017-03-31
4	1	7000	2017-02-28
5	2	6000	2017-02-28
6	3	8000	2017-02-28

| employee\_id | department\_id |

The employee\_id column refers to the employee\_id in the following table employee.

pay\_month | department\_id | comparison

#### The average salary for department '1' is 9000, which is the salary of **employee\_id** '1' since there is only one

employee in this department. So the comparison result is 'higher' since 9000 > 8333.33 obviously.

department '1' and '2' have the same average salary with the company, which is 7000.

In March, the company's average salary is (9000+6000+10000)/3 = 8333.33...

and '3'. So the comparison result is 'lower' since 8000 < 8333.33.

With he same formula for the average salary comparison in February, the result is 'same' since both the

The average salary of department '2' is (6000 + 10000)/2 = 8000, which is the average of employee\_id '2'

1.Calculate the company's average salary in every month MySQL has the built-in function avg() to get the

#### Intuition

Solution

Algorithm

Approach: Using avg() and case...when... [Accepted]

### average of a list of numbers. Also, we need to format the *pay\_date* column for future use.

company\_avg

7000.0000

8333.3333

from salary

1

1

2

select avg(amount) as company\_avg, date\_format(pay\_date, '%Y-%m') as pay\_month

Solve this problem by 3 steps as below.

from salary
group by date\_format(pay\_date, '%Y-%m')

2.Calculate the each department's average salary in every month To do this, we need to join the **salary** table with the **employee** table using condition **salary.employee\_id** = **employee.id**. **select** department\_id, **avg**(amount) **as** department\_avg, date\_format(pay\_date, '%Y-%m') **as** 

pay\_month

2017-02

2017-03

2017-02

2017-03

2017-02

7000.0000

9000.0000

7000.0000

select department\_salary.pay\_month, department\_id,

when department\_avg>company\_avg then 'higher'
when department\_avg<company\_avg then 'lower'</pre>

computer. It works right. Can you guys help?

SHOW 1 REPLY

select left(t1.pay\_date,7) as pay\_month, t2.department\_id,

case when avg(t1.amount) > t3.value then "higher"

select department\_salary.pay\_month, department\_id,

department\_id,

with dataset as (

select

SELECT pay\_date,

tyumneva 🛊 83 🧿 July 18, 2018 2:49 AM

date\_format(s.pay\_date,'%Y-%m') pay\_month,

3.Compare the previous numbers and display the result This step might be the hardest if you have no idea on how to use MySQL flow control statement case...when...

MySQL, like other programming languages, also has its flow control. Click this link to learn it.

At last, combine the above two query and join them on department\_salary.pay\_month = company\_salary.pay\_month.

# select department\_id, avg(amount) as department\_avg, date\_format(pay\_date, '%Y-%m') from salary join employee on salary.employee\_id = employee.employee\_id group by department\_id, pay\_month

join

from

else 'same'
end as comparison

) as department\_salary

MySQL

```
select avg(amount) as company_avg, date_format(pay_date, '%Y-%m') as pay_month from
) as company_salary
on department_salary.pay_month = company_salary.pay_month
;

Rate this article: ** * * * *

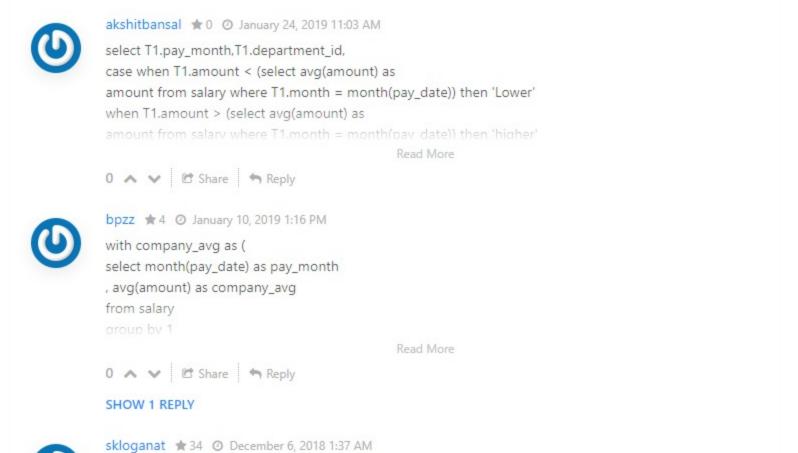
O Previous

Next O

Type comment here... (Markdown is supported)

Post

wjch7722 * 5 O February 10, 2019 4:53 AM
I submitted my solution which has problem with the test. However, I built the dataset in my own
```



Read More

when department_avg>company_avg then 'higher'	
when department_avg <company_avg 'lower'<="" th="" then=""><th></th></company_avg>	
else 'same'	
Read More	
0 ∧ ∨ ☑ Share ★ Reply	
ravinfy ★ 10 ② October 7, 2018 11:01 PM	
Using Analytical Functions in Oracle	
Solution was not implemented to test ALL use cases	
select	
distinct a now data a dant id	

Read More

Read More

	pachiappan ★ 12 ② May 24, 2018 12:08 PM				
	with c as(				
	select to_char(s.pay_date,'YYYY-MM') pay_date, avg(amount) amt				
	from salary s				
	group by to_char(s.pay_date,'YYYY-MM'))				
	select to char(s.pav date.'YYYY-MM') pav month. e.department id.				
	Read More				
	0 A V Share Reply				

	avg(amount) over (partition by trunc(pay_date,'mm') ) - avg(amount) over (partition by					
	trunc(pay_date, 'mm'), department_id ) as salary_ind,					
	department id. trunc(pay date.'mm") as pay month					
	Read More					
	0 A V C Share					
	margol ★ 0 ② January 31, 2018 6:08 AM					

iamkiranps@gmail.com 🛊 1 🗿 April 16, 2018 10:43 PM

You could write CTEs If you are using PostgreSQL:

WITH company\_salary AS (

with company\_avg as (

	0 ∧ ∨ © Share	
<b>©</b>	npras ★4 ② January 16, 2018 4:39 PM	
	If you are using PostgreSQL, you could write the solution mentioned above in a more readable way using CTEs (common table expressions):	

Read More

```
Read More

O A V Share Reply
```