579. Find Cumulative Salary of an Employee

July 10, 2017 | 20.5K views

Average Rating: 4.20 (15 votes)

The **Employee** table holds the salary information in a year.

Write a SQL to get the cumulative sum of an employee's salary over a period of 3 months but exclude the most recent month.

The result should be displayed by 'Id' ascending, and then by 'Month' descending.

Example
Input

```
| Id | Month | Salary |
|----|-------|--------|
| 1  | 1     | 20     |
| 2  | 1     | 20     |
| 1  | 2     | 30     |
| 2  | 2     | 30     |
| 3  | 2     | 40     |
| 1  | 3     | 40     |
| 3  | 3     | 60     |
| 1  | 4     | 60     |
| 3  | 4     | 70     |
```

Output

```
| Id | Month | Salary |
|----|-------|--------|
| 1  | 3     | 90     |
| 1  | 2     | 50     |
| 1  | 1     | 20     |
| 2  | 1     | 20     |
| 3  | 3     | 100    |
| 3  | 2     | 40     |
```

Explanation
Employee '1' has 3 salary records for the following 3 months except the most recent month '4': salary 40 for month '3', 30 for month '2' and 20 for month '1'
So the cumulative sum of salary of this employee over 3 months is 90(40+30+20), 50(30+20) and 20 respectively.

```
| Id | Month | Salary |
|----|-------|--------|
| 1  | 3     | 90     |
| 1  | 2     | 50     |
| 1  | 1     | 20     |
```

Employee '2' only has one salary record (month '1') except its most recent month '2'.

```
| Id | Month | Salary |
|----|-------|--------|
| 2  | 1     | 20     |
```

Employ '3' has two salary records except its most recent pay month '4': month '3' with 60 and month '2' with 40. So the cumulative salary is as following.

```
| Id | Month | Salary |
|----|-------|--------|
| 3  | 3     | 100    |
| 3  | 2     | 40     |
```

Solution

Approach: Using `OUTER JOIN` and temporary tables [Accepted]

Intuition

Solve this issue by two steps. The first one is to get the cumulative sum of an employee's salary over a period of 3 months, and then exclude the most recent month from the result.

Algorithm

If you feel hard to work out how to get the cumulative sum of an employee's salary over a period of 3 months, think about 2 months as a start. By joining this **Employee** table with itself, you can get salary information for one more month.

```sql
SELECT *
FROM
    Employee E1
        LEFT JOIN
    Employee E2 ON (E2.id = E1.id
        AND E2.month = E1.month - 1)
ORDER BY E1.id ASC , E1. month DESC
```

Id	Month	Salary	Id	Month	Salary
1	4	60	1	3	40
1	3	40	1	2	30
1	2	30	1	1	20
1	1	20			
2	2	30	2	1	20
2	1	20			
3	4	70	3	3	60
3	3	60	3	2	40
3	2	40			
>Note:					
> - The blank value in the output is actually `NULL` in the database.					
> - The first three columns are from E1, and the rest ones are from E2.					

Then we can add the salary to get the cumulative sum for 2 months.

```sql
SELECT
    E1.id,
    E1.month,
    (IFNULL(E1.salary, 0) + IFNULL(E2.salary, 0)) AS Salary
FROM
    Employee E1
        LEFT JOIN
    Employee E2 ON (E2.id = E1.id
        AND E2.month = E1.month - 1)
ORDER BY E1.id ASC , E1.month DESC
```

```
| id | month | Salary |
|----|-------|--------|
| 1  | 4     | 100    |
| 1  | 3     | 70     |
| 1  | 2     | 50     |
| 1  | 1     | 20     |
| 2  | 2     | 50     |
| 2  | 1     | 20     |
| 3  | 4     | 130    |
| 3  | 3     | 100    |
| 3  | 2     | 40     |
```

Similarly, you can join this table one more time to get the cumulative sum for 3 months.

```sql
SELECT
    E1.id,
    E1.month,
    (IFNULL(E1.salary, 0) + IFNULL(E2.salary, 0) + IFNULL(E3.salary, 0)) AS Salary
FROM
    Employee E1
        LEFT JOIN
    Employee E2 ON (E2.id = E1.id
        AND E2.month = E1.month - 1)
        LEFT JOIN
    Employee E3 ON (E3.id = E1.id
        AND E3.month = E1.month - 2)
ORDER BY E1.id ASC , E1.month DESC
;
```

```
| id | month | Salary |
|----|-------|--------|
| 1  | 4     | 130    |
| 1  | 3     | 90     |
| 1  | 2     | 50     |
| 1  | 1     | 20     |
| 2  | 2     | 50     |
| 2  | 1     | 20     |
| 3  | 4     | 170    |
| 3  | 3     | 100    |
| 3  | 2     | 40     |
```

In addition, we have to exclude the most recent month as required. If we have a temp table including every id and most recent month like below, then we can easily opt out these months by join it with the above table.

```
| id | month |
|----|-------|
| 1  | 4     |
| 2  | 2     |
| 3  | 4     |
```

Here is the code to generate this table.

```sql
SELECT
    id, MAX(month) AS month
FROM
    Employee
GROUP BY id
HAVING COUNT(*) > 1
;
```

At last, we can join them together and get the desired cumulative sum of an employee's salary over a period of 3 months excluding the most recent one.

MySQL

```sql
SELECT
    E1.id,
    E1.month,
    (IFNULL(E1.salary, 0) + IFNULL(E2.salary, 0) + IFNULL(E3.salary, 0)) AS Salary
FROM
    (SELECT
        id, MAX(month) AS month
    FROM
        Employee
    GROUP BY id
    HAVING COUNT(*) > 1) AS maxmonth
        LEFT JOIN
    Employee E1 ON (maxmonth.id = E1.id
        AND maxmonth.month > E1.month)
        LEFT JOIN
    Employee E2 ON (E2.id = E1.id
        AND E2.month = E1.month - 1)
        LEFT JOIN
    Employee E3 ON (E3.id = E1.id
        AND E3.month = E1.month - 2)
ORDER BY id ASC , month DESC
;
```

id	month	Salary
1	3	90
1	2	50
1	1	20
2	1	20
3	3	100
3	2	40

Note: Thank @xiaxin for providing this elegant solution.

Rate this article:

Previous Next

Comments: 50 Sort By

Type comment here... (Markdown is supported)

Preview Post

AnandhaPrakash ★ 21 June 16, 2018 5:36 AM

SELECT A.EMPLOYEEID,A.MONTH
,SUM(SALARY) over (PARTITION BY A.EMPLOYEEID order by salary ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) CUMULATIVESALARY
FROM EMPLOYEE_INPUT A
WHERE (A.EMPLOYEEID,A.MONTH)

Read More

21 Share Reply

rongy2018 ★ 166 December 29, 2018 2:32 AM

```
SELECT
    id, MAX(month) AS month
FROM
    Employee
```

Read More

14 Share Reply

SHOW 1 REPLY

mojavesolutions ★ 12 April 22, 2018 3:39 AM

I accidentally had an extra item on the WHERE clause... here is the corrected version:

```
SELECT e.id, e.month,
    SUM(e2.Salary) as Salary
FROM Employee e
```

Read More

8 Share Reply

SHOW 2 REPLIES

Supriya_29 ★ 8 January 17, 2019 7:17 PM

One more way of doing it using ROW_NUMBER:

WITH s AS
(SELECT Id,Month,Salary,
Sum(Salary) OVER (PARTITION BY Id ORDER BY Month) as SumSal,

Read More

7 Share Reply

SHOW 1 REPLY

justpracticing123 ★ 6 September 7, 2018 3:12 AM

If I understand this properly, the solution relies on a user having information for all months they worked consecutively (2nd to last month = last month -1). But this is not guaranteed?

6 Share Reply

tyumneva ★ 83 July 25, 2018 10:25 PM

The solution on leetcode only counts up to 3 months of running total.
This query gives the correct answer for any amount of months:
Select
e1.id,
e2.month,

Read More

5 Share Reply

SHOW 1 REPLY

jkim65537 ★ 4 March 8, 2019 10:24 AM

`select a.id, a.month, sum(b.salary) from employee a, employee b where a.id = b.id and a.month >= b.month and a.month < (select max(month) from employee c where a.id = c.id) group by a.id, a.month`

4 Share Reply

tyumneva ★ 83 July 17, 2018 11:42 PM

select e.id,
e.month,
(select sum(salary) from employee e2 where e2.id = e.id and e2.month <= e.month) as salary
from employee e
where (e.id, e.month) not in (select id, max(month) month from Employee group by id)

Read More

4 Share Reply

SHOW 1 REPLY

mojavesolutions ★ 12 April 22, 2018 3:36 AM

Simple solution that just does a self-JOIN to get the last 2 months (excluding the latest / MAX month) and then just SUMming the Salary for those JOINed months.

```
SELECT e.id, e.month,
    SUM(e2.Salary) as Salary
```

Read More

4 Share Reply

SHOW 1 REPLY

npras ★ 4 January 15, 2018 4:46 PM

A much simpler solution using window functions RANK() and SUM() if you are using PostgreSQL:

```
select
    employeeid,
    month,
```

Read More

4 Share Reply

SHOW 4 REPLIES

‹ 1 2 3 4 5 ›