

```

select
    employee_id,first_name,hire_date,job_id,department_id,salary,
    sum(salary) over(order by hire_date
        ROWS BETWEEN UNBOUNDED preceding and current row) as tot_salary
from employees;

```

```

select *,
sum(sale) OVER (order by mkt_date
    ROWS BETWEEN UNBOUNDED preceding and current row) as tot_sale
from sales;

```

```

#200 600 900 1200 1600 2100

```

```

# get me the running sum of sales for last 3 days including today?
select *,
    sum(sale) over(order by mkt_date
    ROWS between 3 preceding and current row)
from sales;

```

```

select *,
    sum(sale) over(order by mkt_date
    ROWS between 1 preceding and 1 following)
from sales;

```

```

select *,
    sum(sale) over(order by mkt_date
    ROWS between unbounded preceding and unbounded following)
from sales;

```

```

select *,
    sum(sale) over(partition by employee order by mkt_date desc
    RANGE BETWEEN unbounded preceding and current row)
from sales;

```

```

select *,
    avg(sale) over(partition by employee order by mkt_date desc
    ROWS BETWEEN unbounded preceding and current row)
from sales;

```

```

## get me the next highest salary for every employee
#and then the difference between them

```

```

select count(*) from
(
select
    employee_id,first_name,hire_date,job_id,department_id,salary,
    lead(salary,1) over(order by salary) as next_highest_salary,

```

```
        #lead(salary,1) over(order by salary) - salary  as promotion,
        lag(salary,1) over(order by salary desc) as nxt_3rd_highest
from employees
) t
where t.promotion = 0;
```

```
#DSC
select
    employee_id,first_name,hire_date,job_id,department_id,salary,
    lead(salary,1) over(order by salary) as next_highest_salary
from employees
```