

```
select
employee_id,
salary - t.next_lowest_salary as diff_salary from
(
select
employee_id,
salary,
job_id,
lead(salary) OVER() as next_lowest_salary
from
employees
where department_id = 30
)t;
```

```
select
employee_id,
first_name,
job_id,
salary,
department_id,
nth_value(salary,7) OVER(partition by department_id order by salary desc
RANGE BETWEEN unbounded preceding and unbounded following) as 7th_highest
from employees;
```

```
select
employee_id,
first_name,
job_id,
salary,
department_id,
nth_value(salary,12) OVER(partition by department_id order by salary desc
RANGE BETWEEN unbounded preceding and unbounded following) as 7th_highest
from employees;
```

```
select
employee_id,
first_name,
job_id,
salary,
department_id,
nth_value(salary,11) OVER(partition by department_id order by salary desc
ROWS BETWEEN unbounded preceding and unbounded following) as 7th_highest
from employees;
```

```
select count(*) from  
employees;
```

```
select  
employee_id,  
first_name,  
job_id,  
salary,  
department_id,  
first_value(salary) OVER(partition by department_id order by salary desc  
RANGE BETWEEN unbounded preceding and unbounded following) as 7th_highest  
from employees;
```

```
select * from  
(  
select  
employee_id,  
first_name,  
salary,  
department_id,  
NTILE(4) OVER(partition by department_id order by salary) as quartile  
from  
employees  
)t  
where t.quartile = 1;
```

####difference between dense_rank and nth_value

```
select  
employee_id,  
first_name,  
salary,  
department_id,  
dense_rank() OVER(partition by department_id order by salary desc) as d_rnk,  
nth_value(salary,1) OVER(partition by department_id order by salary desc) as nth_rnk  
from  
employees;
```

#####pratik

```
select  
employee_id,  
first_name,
```

```
salary,  
department_id,  
rank() OVER(order by salary desc)  
from  
employees;
```

###shahnawaz

```
select  
employee_id,  
first_name,  
job_id,  
salary,  
department_id,  
nth_value(salary,7) OVER(partition by department_id) as 7th_highest  
from employees;
```

###rituraj

```
select  
employee_id,  
first_name,  
job_id,  
salary,  
department_id,  
nth_value(salary,7) OVER(partition by department_id order by salary desc) as 7th_highest  
from employees;
```

###merlin

```
select *,  
sum(sale) OVER(order by date ROWS BETWEEN unbounded preceding and current row) as  
agg_demo,  
rank() OVER(order by date ROWS BETWEEN unbounded preceding and current row) as  
rnk_demo,  
row_number() OVER(order by date) as rw_demo  
from sales;
```

krunal

```
select  
employee_id,  
first_name,  
job_id,
```

```
salary,  
department_id,  
nth_value(salary,11) OVER(partition by department_id  
order by salary desc) as 7th_highest  
from employees;
```