```
select sum(salary)
from employees;

select department_id,sum(salary)
from employees
group by department_id;

select department_id,
(select sum(salary) from employees group by department_id)
from employees;

######8.0#################
select
employee_id,
first_name,
salary,
department_id,
job_id,
MIN(salary) OVER() as min_salary
from
employees;


#######
select
employee_id,
first_name,
salary,
department_id,
job_id,
avg(salary) OVER(PARTITION BY department_id,job_id) as avg_salary_dept_wise
from
employees;

select
employee_id,
first_name,
salary,
department_id,
job_id,
avg(salary) OVER(PARTITION BY department_id) as avg_salary_dept_wise
from
employees;


###################
select
```

```
*,
sum(sale) OVER(partition by employee)
from sales;

select
*,
sum(sale) OVER(ORDER by date ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) as orderbydate
from sales;


select
*,
count(sale) OVER(ORDER by date ROWS BETWEEN
1 PRECEDING AND 1 FOLLOWING) as orderbydate
from sales;


######

select
*,
sum(sale) OVER(partition by employee order by date)
from sales;

select * from sales;

select
*,
row_number() OVER(order by date) as rw_number
from sales;



select * from employees;

select * from
(
select
employee_id,
first_name,
salary,
department_id,
job_id,
DENSE_RANK() OVER(PARTITION BY department_id order by salary desc) as
avg_salary_dept_wise
from
```

```
employees
) t
where t.avg_salary_dept_wise = 5
order by
```

##########Doubt Clearing Session#########

```
select
        e.department_id,
   d.department_name,
   avg(e.salary) as Average_salary
from employees e LEFT JOIN departments d
using(department_id)
where e.department_id is not null
group by e.department_id
order by department_id;
```

```
select * from
(
select
employee_id,
first_name,
salary,
department_id,
job_id,
DENSE_RANK() OVER(PARTITION BY department_id order by salary desc) as
avg_salary_dept_wise
from
employees
) t
where t.avg_salary_dept_wise = 2
order by first_name;
```