```
with agg as (

select dpid, name, salary, dense_rank() over (partition by dpid order by salary desc) as rank from
employee)


select a.salary as sl

from agg a

inner join dpt d on a.dpid = d.id

where a.rank <4
```

window function, dense_rank vs rank

how they handle identical values,

rank will skip the next available ranking values

dense_rank would still use the next chronological ranking value.

```
create function name (n int) returns int

begin

set n = n  - 1;

        return (select distinct salary from employee

order by salary desc

limit 1 offerset n

);

end
```

```sql
select email
from person
group by email
having count(*) > 1
```

```sql
with cte as (
select email, row_number() over (partition by email order by email)
as rn from person)
select email
from cte
where rn >1;
```

```sql
select name as customers
from customers
left join orders
on customer.id = orders.customerid
where orders.customerid is null;
```

```sql
select name as customers
from customres
where id not in (select customerid from orders);
```

```
seelect dp.name as dptname
, employe.name as empl
, salary
from employee em
inner join department as dp
on em.deptid = dp.id
where (dpid, salary) in (select departmentid, max(salary) as salary
from employee
group by dpartmentid
);
```

```
delete p2
from person p1
join person p2
on p1.email = p2.email
and p1.id < p2.id
```

```
select t.id
from weather t
join weather y
```

```
on datediff(t.data, y.date) = 1
and t.tmp > y.tmp
```

```
select t.id
from weather as t,
weather as y
where datediff(t., y.) = 1
and t.t > y.t;
```

```
SELECT Request_at AS Day,
ROUND(SUM(IF(Status<>"completed", 1, 0))/COUNT(Status),2) AS "Cancellation Rate"
FROM Trips
WHERE Request_at BETWEEN "2013-10-01" AND "2013-10-03"
AND Client_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Yes')
AND Driver_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Yes')
GROUP BY Request_at;
```

```
SELECT
```

```
    a.id,
    a.month,
    SUM(b.salary) Salary
FROM
    Employee a JOIN Employee b ON
    a.id = b.id AND
    a.month - b.month >= 0 AND
    a.month - b.month < 3
GROUP BY
    a.id, a.month
HAVING
    (a.id, a.month) NOT IN (SELECT id, MAX(month) FROM Employee GROUP BY id)
ORDER BY
    a.id, a.month DESC
```