

Employees			
employee_id	name	department	salary
unique			

projects		
project_id	project_name	department
unique		

### Assignments

Assignment Id	employee_id	Project Id	hours_worked
unique			

with cte as

(1) (select department, name, salary, Dense rank  
 ( ) over (partition by salary, department  
 order by salary desc) as rank  
 from employees) +  
 select \* from cte where rank = 1;

Date: \_\_\_\_\_

② select E.employee-Id, E.name,  
 Count(~~P~~A.project-Id) from  
 Employees E inner Join  
 Assignments ~~Project~~ ~~E~~ department  
 on E.department = P.department  
 inner Join Assignments A  
 on P.project-Id = A.project-Id  
 group by ~~E~~ E.employee-Id, E.name  
 Having Count(A.project-Id) > 3;

② select E.employee-Id, E.name, Count(A.project-  
 Id) as total\_projects from  
 Employees E  
 inner Join Assignments A  
 on E.employee-Id = A.employee-Id  
 group by E.employee-Id, E.name  
 Having Count(A.project-Id) > 3;

③ select  
 P.department, ~~A.project-Id~~ Sum(A.hours\_worked) as  
 Total\_hours from Assignments A Join  
 Projects P  
 on A.project-Id = P.project-Id  
 group by P.department

④ Underpaid but Overworked :-

```

Select E.employee-Id, E.name, E.department,
Sum(E.salary), Sum(A.hours-worked)
as total-hours
from Employees E
Inner Join
Assignment A
on E.employee-Id = A.employee-Id
group by E.employee-Id, E.name, E.department
Having Sum(Salary) < Avg(Salary)
And A.hours-worked > 100 hours

```

```

Select E.employee-Id, E.name, E.department,
E.salary, Sum(A.hours-worked) as
total-hours
from employees E
Inner Join Assignments A
on E.employee-Id = A.employee-Id
Having
group by 1, 2, 3
Having
e.salary < (select avg(salary) from
employees E where department = E.department)
and Sum(a.hours-worked) > 100;

```

⑤ Select ~~projectId~~ project name,  
Count (department) from projects  
group by ~~projectId~~ project name  
having Count (department) > 1;