

SQL

1. Mostra os postos de traballo que hai en cada departamento (código de dept e nome do posto de traballo). Non deben aparecer repetidos.

```
select distinct deptno, job
from emp;
```

2. Mostra os códigos de empregados que son xefes. No resultado non deben aparecer filas con nulos.

```
select mgr
from emp
where mgr is not null;
```

3. Mostra as cidades onde se executan proxectos controlados polo departamento 30. Non deben aparecer repetidos no resultado.

```
select distinct loc
from dept
where deptno=30;
```

4. Mostra empregados que non teñen xefe.

```
select empno, ename
from emp
where mgr is null;
```

5. Mostra empregados que teñan xefe e que gañen (incluíndo salario e comisión) máis de 2500.

```
select empno, ename
from emp
where mgr is not null and coalesce(sal+comm, sal)>2500;
```

6. Mostra os empregados cuxo nome comeza por 'S'.

```
select empno, ename
from emp
where ename like 'S%';
```

7. Mostra os empregados que gañan (incluíndo salario e comisión) entre 1500 e 2500 euros.

```
select empno, ename
from emp
where coalesce(sal, sal+comm) between 1500 and 2500;
```

8. Mostra os empregados que son 'CLERK', 'SALESMAN' ou 'ANALYST' e gañan (incluíndo salario e comisión) máis de 1250.

```
select empno, ename
from emp
where coalesce(sal, sal+comm)>1250 and job in ('CLERK', 'SALESMAN',
'ANALYST');
```

9. Cantos empregados hai en cada departamento, cantos teñen comisión, cantos non teñen comisión e cales son os ingresos medios (incluíndo salario e comisión).

```
select count(empno), count(comm), count(empno)-count(comm),
avg(coalesce(sal+comm, sal))
from emp
group by deptno;
```

10. Mostra os departamentos que teñen empregados con comisión. Non pode haber valores repetidos.

```
select distinct deptno
from emp
where comm is not null;
```

11. Para cada departamento mostra a comisión media; se non ten empregados con comisión, débese indicar cun 0.

```
select deptno, coalesce(avg(comm), 0)
from emp
group by deptno;
```

-
12. Para cada departamento mostra cantos postos de traballo distintos desempeñan os seus traballadores.

```
select deptno, count(distinct job)
from emp
group by deptno;
```

13. Para cada departamento mostra cantos empregados hai de cada posto de traballo.

```
select deptno, job, count(empno)
from emp
group by deptno, job;
```

14. Mostra cantos empregados teñen uns ingresos superiores a 2500 € en cada departamento.

```
select deptno, count(empno)
from emp
where coalesce(sal+comm, sal)>2500
group by deptno;
```

15. Para cada departamento mostra cantos empregados teñen uns ingresos (sal+comm) superiores a 2500 €.

```
select deptno, count(empno)
from emp
where coalesce(sal+comm, sal)>2500
group by deptno;
```

16. Mostra os departamentos cuns ingresos medios superiores aos 2500 €. Mostra para cada un, cantos empregados teñen.

```
select deptno, count(empno)
from emp
group by deptno
having avg(coalesce(sal+comm, sal))>2500;
```

17. Departamentos con, polo menos, dous 'MANAGER'.

```
select deptno
from emp
where job='MANAGER'
group by deptno
having count(*)>=2;
```

18. Departamentos con, polo menos, dous empregados con comisión. Para cada departamento mostra cantos empregados ten (en total) e cantos con comisión.

```
select deptno, count(empno), count(comm)
from emp
group by deptno
having count(comm)>=2;
```

19. Departamentos con, polo menos, dous empregados co mesmo posto de traballo. Non poden aparecer repetidos.

```
select distinct deptno
from emp
group by deptno, job
having count(job)>1;
```

20. Para cada proxecto mostra o seu nome e o nome do departamento que os controla.

```
select p.pname, d.dname
from pro p join dept d on p.deptno=d.deptno;
```

21. Para cada empregado mostra o seu nome e os códigos de proxectos para os que traballa.

```
select e.ename, ep.prono
from emp e join emppro ep on e.empno=ep.empno;
```

22. Para cada empregado mostra o seu nome e os códigos de proxectos para os que traballa. Se hai empregados que non traballan en proxectos, estes deben aparecer co código de proxecto a nulo.

```
select e.ename, ep.prono
from emp e left join emppro ep on e.empno=ep.empno;
```

23. Para cada empregado mostra o nome do seu xefe; se non ten xefe, mostra un nulo no nome do xefe.

```
select e.ename, j.ename  
from emp e left join emp j on e.empno=j.empno;
```

24. Para cada empregado mostra o seu nome, o nome do seu xefe e o departamento para o que traballa o seu xefe.

```
select e.ename, j.ename, j.deptno  
from emp e join emp j on e.empno=j.empno;
```

25. Devolve os empregados que teñen un salario máis alto que o seu xefe.

```
select e.ename, e.sal, j.ename, j.sal  
from emp e join emp j on e.mgr=j.empno and e.sal>j.sal;
```

26. Para cada empregado mostra o seu nome e cantas horas traballou en proxectos.

```
select e.ename, ep.hours  
from emp e join empproj ep on e.empno=ep.empno;
```

27. Para cada departamento, mostra o seu nome e cantos empregados ten.

```
select d.dname, count(empno)  
from emp e join dept d on e.deptno=d.deptno  
group by d.dname;
```

28. Para cada xefe, mostra o seu nome e cantos subordinados ten.

```
select j.ename, count(e.empno)  
from emp e join emp j on e.mgr=j.empno  
group by j.ename;
```

29. Mostra o nome de proxectos onde se traballou (en total, todos os empregados) máis de 15 horas.

```
select p.pname
from emppro ep join pro p on ep.prono=ep.prono
group by p.pname
having sum(hours)>15;
```

30. Mostra os departamentos (nome) que controlan máis de dous proxectos.

```
select d.dname
from dept d join pro p on d.deptno=p.deptno
group by d.dname
having count(p.prono)>2;
```

31. Mostra os departamentos (nome) onde hai polo menos dous empregados co mesmo posto de traballo. Non deben aparecer repetidos.

```
select distinct d.dname
from emp e join dept d on e.deptno=d.deptno
group by d.dname, e.job
having count(e.job)>=2;
```

32. Para cada departamento mostra o seu nome e cantos empregados ten; se non ten ningún, hai que indicalo cun 0.

```
select d.dname, count(e.empno)
from emp e right join dept d on e.deptno=d.deptno
group by d.dname;
```

33. Para cada empregado mostra as horas que traballou en proxectos; se non traballou en ningún, hai que indicalo cun 0.

```
select e.ename, coalesce(sum(hours),0)
from emp e left join emppro ep on e.empno=ep.empno
group by e.ename;
```

34. Para cada xefe, cantos subordinados gañan máis ca el; se non o gaña ningún, hai que indicalo cun 0.

```
select j.ename, count(e.empno)
from emp e right join emp j on e.mgr=j.empno and e.sal>j.sal
group by j.ename;
```

35. Empleados que teñen un salario maior ao salario medio da empresa.

```
select e.ename
from emp e
where sal >
(
select avg(sal)
from emp
);
```

36. Para cada departamento, mostra cantos empregados ten que gañen máis do salario medio da empresa. Mostra o nome do departamento.

```
select deptno, count(empno)
from emp
where sal >
(
select avg(sal)
from emp
)
group by deptno;
```

37. Empleados que son xefe. Mostra o seu nome

```
select ename
from emp
where empno in
(
select mgr
from emp
);
```

38. Empleados que non son xefe. Mostra o seu nome.

```
select ename
from emp
where empno not in
(
select mgr
```

```
from emp
where mgr is not null
);
```

39. Mostra o(s) empregado(s) (nome) co salario máis alto.

```
select ename
from emp
where sal=
(
select max(sal)
from emp
);
```

40. Mostra o departamento (nome) coa suma de salarios máis alta.

```
select d.dname
from emp e join dept d on e.deptno=d.deptno
group by e.deptno, d.dname
having sum(sal)=
(
select max(sum(sal))
from emp
group by deptno
);
```

41. Para os departamentos que teñen empregados con comisión, mostra cantos empregados teñen comisión, e cantos non. Mostra o nome do departamento.

```
select d.dname, count(comm), count(*)-count(comm)
from emp e join dept d on e.deptno=d.deptno
where d.deptno in
(
select distinct deptno
from emp
where comm is not null
)
group by d.dname;
```

42. Mostra o(s) empregado(s) co salario máis alto de cada departamento.


```

select ename, deptno
from emp e
where sal >= all
(
select sal
from emp
where deptno=e.deptno
)
group by ename, deptno;

```

43. Mostra o código do(s) empregado(s) que máis horas traballa(n) en cada proxecto.

```

select e.empno, ep.prono
from emp e join emppro ep on e.empno=ep.empno
where hours=
(
select max(hours)
from emppro
where ep.prono=pronon
);

```

44. Mostra o nome do(s) empregado(s) que máis horas traballa(n) en cada proxecto.

```

select e.ename, ep.prono
from emp e join emppro ep on e.empno=ep.empno
where hours=
(
select max(hours)
from emppro
where ep.prono=pronon
);

```

45. Mostra o nome do(s) empregado(s) que máis horas traballa(n) en cada proxecto. Mostra tamén o nome do proxecto.

```

select e.ename, ep.prono, p.pname
from emp e join emppro ep on e.empno=ep.empno
join pro p on ep.prono=p.prono
where hours=

```

```
(  
select max(hours)  
from emppro  
where ep.prono=prono  
);
```

46. Para cada departamento mostra o seu nome e cantos empregados dese departamento teñen un salario maior ao salario medio do seu departamento.

```
select d.dname, count(e.empno)  
from emp e join dept d on e.deptno=d.deptno  
where e.sal >  
(  
select avg(sal)  
from emp  
where deptno=e.deptno  
)  
group by d.dname;
```

47. Para cada departamento mostra o seu nome e cantos empregados gañan máis que o seu xefe.

```
select d.dname, count(e.empno)  
from emp e join dept d on e.deptno=d.deptno  
where e.sal >  
(  
select sal  
from emp  
where empno=e.mgr  
)  
group by d.dname;
```

48. Para cada empleado muestra su nombre y cuántas horas trabajó en proyectos.

```
select ename, sum(hours)  
from emp e join emppro ep on e.empno=ep.empno  
group by ename;
```

49. Para cada empleado muestra su nombre, el nombre de su jefe, y el departamento para el que trabaja su jefe.

```
select e.ename, j.ename, j.deptno
from emp e join emp j on e.mgr=j.empno;
```

50. Para cada jefe, muestra su nombre y cuántos subordinados tiene.

```
select j.ename, count(e.ename)
from emp e join emp j on e.mgr=j.empno
group by j.ename;
```

51. Muestra el nombre de proyectos donde se ha trabajado (en total, todos los empleados) más de 15 horas.

```
select pname, sum(hours)
from pro p join emppro ep on p.prono=ep.prono
group by pname
having sum(hours)>15;
```

52. Muestra los departamentos (nombre) donde hay por lo menos dos empleados con el mismo puesto de trabajo. No debe aparecer repetidos.

```
select distinct dname
from dept d join emp e on d.deptno=e.deptno
group by dname, job
having count(job)>=2;
```

53. Para cada proyecto muestra el empleado que ha trabajado más horas. Muestra los nombres del proyecto y empleado.

```
select p.pname, e.ename
from emp e join emppro ep on e.empno=ep.empno
join pro p on ep.prono=p.prono
where ep.hours=
(
select max(hours)
from emppro a
where a.prono=ep.prono
)
group by pname, ename;
```

54. Muestra el jefe tal que la suma de los salarios de sus subordinados es la más alta. Muestra su nombre y la suma de salarios de sus subordinados.

```
select b.ename, sum(a.sal)
from emp a join emp b on a.mgr=b.empno
group by b.ename
having sum(a.sal)=
(
select max(sum(e.sal))
from emp e join emp j on e.mgr=j.empno
group by j.ename
);
```

55. Muestra para cada departamento el proyecto controlado por él donde se han trabajado más horas.

```
select p.deptno, p.prono, sum(hours)
from emppro ep join pro p on ep.prono=p.prono
group by p.deptno, p.prono
having sum(hours)=
(
select max(sum(hours))
from emppro a join pro b on a.prono=b.prono
where b.deptno=p.deptno
group by b.prono
);
```

56. Para cada departamento mostrar su nombre y cuántos empleados tiene, si no tiene ninguno, indicarlo con un 0.

```
select dname, count(empno)
from emp natural join dept
group by dname;
```

57. Para cada empleado mostrar las horas que trabajó en proyectos, si no trabajó en ninguno, indicarlo con un 0.

```
select ename, coalesce(sum(hours), 0)
from emp e left join emppro ep on e.empno=ep.empno
group by ename;
```

58. Para cada empleado muestra su nombre, el nombre de su jefe, y el departamento para el que trabaja su jefe. Si el empleado no tiene jefe, debe aparecer con nulos en los datos del jefe.

```
select e.ename, j.ename, j.deptno
from emp e left join emp j on e.mgr=j.empno;
```

59. Para cada empleado muestra en cuántas ciudades distintas ha trabajado. Si no ha trabajado en ninguna, debe mostrar un cero. Muestra el nombre del empleado.

```
select ename, count(p.loc)
from emp e left join emppro ep on e.empno=ep.empno
left join pro p on ep.prono=p.prono
group by ename;
```

60. Considerando los proyectos controlados por los departamentos 30 y 40, muestra cuántos empleados distintos han trabajado en cada departamento. Si no han trabajado empleados, debe mostrar un cero. Muestra el nombre del departamento.

```
select d.deptno, d.dname, count(distinct ep.empno)
from pro p right join emppro ep on ep.prono=p.prono
right join dept d on p.deptno=d.deptno
where d.deptno=30 or d.deptno=40
group by d.deptno, d.dname;
```

61. Muestra cuántos empleados subordinados tiene cada jefe, que fueran contratados el mismo año que él. Si no tiene ninguno debe mostrar un cero.

```
select j.ename, count(e.empno)
from emp e right join emp j on e.mgr=j.empno
and to_char(e.hiredate, 'YYYY')=to_char(j.hiredate, 'YYYY')
where j.empno in
(
select mgr
from emp
)
group by j.ename;
```

62. Muestra los empleados que han trabajado en proyectos ubicados en la misma localidad donde está su departamento.

```
select distinct empno, d.loc
from emppro ep join pro p on ep.prono=p.prono
join dept d on d.deptno=p.deptno
where p.loc=d.loc;
```

63. Para cada empleado mostrar cuántas veces trabajó en proyectos ubicados en la misma ciudad donde está su departamento.

```
select empno, count(p.prono)
from emppro ep join pro p on ep.prono=p.prono
join dept d on d.deptno=p.deptno
where p.loc=d.loc
group by empno;
```

64. Idem anterior, pero mostrando un cero cuando nunca ocurrió.

```
select e.empno, count(p.prono)
from emp e left join emppro ep on e.empno=ep.empno
left join pro p on ep.prono=p.prono
left join dept d on d.deptno=p.deptno and p.loc=d.loc
group by e.empno;
```

65. Muestra los empleados que ganan más del salario medio de la empresa.
Muestra también dicho salario medio.

```
select ename, (select avg(sal) from emp) as media
from emp
where sal>(select avg(sal) from emp);
```

66. Muestra los empleados que ganan más del salario medio de su departamento.
Muestra también dicho salario medio.

```
select e.ename, e.deptno, media
from emp e join
(
select deptno, avg(sal) media
from emp
```

```
group by deptno
) f
on e.deptno=f.deptno
where sal>media;
```

67. Muestra de cada proyecto, el empleado que más horas trabaja.

```
select pname, empno
from emppro ep join pro p on ep.prono=p.prono
group by pname, empno
having sum(hours)=
(
select max(sum(hours))
from emppro a join pro b on a.prono=b.prono
group by pname, empno
having pname=p.pname
);
```

68. El empleado que ha trabajado más horas en cada ciudad. Muestra los nombres del empleado y de la ciudad.

```
select ename, loc, sum(hours)
from emp e join emppro ep on e.empno=ep.empno
join pro p on ep.prono=p.prono
group by ename, loc
having sum(hours)>=all
(
select sum(hours)
from emppro ep1 join pro p1 on ep1.prono=p1.prono
where p1.loc=p.loc
group by ep1.empno
);
```

69. Muestra la media de horas dedicadas a proyectos (sumando las horas de todos los empleados que trabajan en el proyecto).

```
select avg(sum(hours))
from emppro ep
group by prono;
```

70. Muestra los proyectos cuya suma de horas es mayor que la cantidad de horas media dedicadas a proyectos. Debes mostrar dicha suma.

```
select p.pname, suma
from pro p join
(
select prono, sum(hours) as suma
from emppro
group by prono
) q on p.prono=q.prono
where suma>
(
select avg(hours)
from emppro
); //está mal, no me convence el enunciado
```

71. Para cada empleado muestra su nombre y cuál es la diferencia de sus ingresos con respecto a los ingresos del empleado que más gana de su departamento.

```
select ename, maximo-coalesce(sal+comm, sal)
from emp e join
(
select deptno, max(coalesce(sal+comm, sal)) as maximo
from emp
group by deptno
) f on e.deptno=f.deptno;
```

72. Mostrar para cada departamento la diferencia entre la suma de los salarios de todos sus empleados y la del departamento con la suma más alta.

```
select deptno,
(
select max(sum(sal))
from emp
group by deptno
) -sum(sal)
from emp
group by deptno;
```

73. Muestra para cada empleado en qué ciudades trabajó.


```
select distinct empno, loc
from emppro ep join pro p on ep.prono=p.prono;
```

74. De entre los empleados que han trabajado en proyectos, muestra para cada uno en qué ciudades no trabajó.

```
select distinct empno, loc
from emppro ep cross join pro p
where (empno,loc) not in
(
select distinct empno, loc
from emppro ep join pro p on ep.prono=p.prono
);
```

75. Muestra para cada empleado de la empresa en qué ciudades no trabajó. Muestra el nombre del empleado.

```
select distinct empno, loc
from emp e cross join pro p
where (empno,loc) not in
(
select distinct empno, loc
from emppro ep join pro p on ep.prono=p.prono
);
```

76. Muestra las ciudades donde trabajó un empleado con un texto 'SI', y aquellas ciudades donde no trabajó con un texto 'NO'. Ten en cuenta todos los empleados (los que están en emp y no en emppro también).

```
select distinct empno, loc, 'SI'
from emp e cross join pro p
where (empno,loc) in
(
select distinct empno, loc
from emppro ep join pro p on ep.prono=p.prono
)
union
select distinct empno, loc, 'NO'
from emp e cross join pro p
where (empno,loc) not in
```

```
(  
select distinct empno, loc  
from emppro ep join pro p on ep.prono=p.prono  
)  
order by empno;
```

77. Muestra parejas de empleados que tienen relación jefe-empleado directa.

```
select e.empno, j.empno  
from emp e join emp j on e.mgr=j.empno;
```

78. Muestra las parejas de empleados que no tienen relación jefe-empleado directa.

```
select e.empno, j.empno  
from emp e cross join emp j  
where (e.empno, j.empno) not in  
(  
select a.empno, b.empno  
from emp a join emp b on a.mgr=b.empno  
union  
select b.empno, a.empno  
from emp a join emp b on a.mgr=b.empno  
)  
and e.empno<>j.empno;
```