



1. Obtener los nombres y salarios de los empleados con más de 1000 euros de salario por orden alfabético.

SELECT nombre, salario

FROM empleado

WHERE salario > 1000

ORDER BY nombre;

2. Obtener el nombre de los empleados cuya comisión es superior al 20% de su salario.

```
SELECT nombre, salario, comision
FROM empleado
WHERE comision >= salario*0.2;
```

3. Obtener el código de empleado, código de departamento, nombre y sueldo total en pesetas, de aquellos empleados cuyo sueldo total (salario más comisión) supera los 1800 euros. Presentarlos ordenados por código de departamento y, dentro de éstos, por orden alfabético.

```
SELECT codempleado, coddpto, nombre, salario*166.38 AS salario_ptas
FROM empleado
WHERE salario > 1800
ORDER BY coddpto, nombre;
```

4. Obtener, por orden alfabético, los nombres de empleados cuyo salario igualen o superen en más de un 5% al salario de la empleada 'MARIA JAZMIN'.

```
SELECT nombre, salario
FROM empleado
WHERE salario >= ((SELECT salario FROM empleado WHERE nombre = 'MARIA') * 1.05)
ORDER BY nombre;
```

5. Obtener un listado ordenado por años en la empresa con los nombres, y apellidos de los empleados, y los años de antigüedad en la empresa.

```
SELECT nombre, TRUNC (MONTHS_BETWEEN(sysdate, fechaingreso)/12) AS antigüedad
FROM empleado
ORDER BY antigüedad;
```

6. Obtener el nombre de los empleados que trabajan en un departamento con presupuesto superior a 50.000 euros. Hay que usar predicado cuantificado.

```
SELECT nombre, dpto.presupuesto, empleado.coddpto
FROM empleado, dpto
WHERE dpto.coddpto = empleado.coddpto AND dpto.presupuesto > 50000
ORDER BY nombre;
```

7. Obtener los nombres y apellidos de empleados que más cobran en la empresa. Considerar el salario más la comisión.

```
SELECT nombre, ape1 || ' ' || ape2 apellidos, (salario + NVL(comision, 0)) as Total_Salario
FROM empleado
WHERE salario + NVL(comision, 0) =
  (SELECT max(salario + NVL(comision, 0))
   FROM empleado);
```

8. Obtener en orden alfabético los nombres de empleado cuyo salario es inferior al mínimo de los empleados del departamento 1.

```
SELECT nombre, salario
FROM empleado
WHERE salario <
  (SELECT MIN(salario)
   FROM empleado WHERE coddpto = 1);
```

9. Obtener los nombres de empleados que trabajan en el departamento del cuál es jefe el empleado con código 1.

```
SELECT nombre, dpto.coddpto, dpto.dptodepende
FROM empleado, dpto
```

WHERE empleado.coddpto = dpto.coddpto AND dpto.codemplejefe = 1;

10. Obtener los nombres de los empleados cuyo primer apellido empiece por las letras p, q, r, s.

```
SELECT nombre, ape1
FROM empleado
WHERE ape1 LIKE 'P%' OR ape1 LIKE 'Q%' OR ape1 LIKE 'R%' OR ape1 LIKE 'S%';
```

11. Obtener los empleados cuyo nombre de pila contenga el nombre JUAN.

```
SELECT nombre
FROM empleado
WHERE nombre LIKE '%JUAN%';
```

12. Obtener los nombres de los empleados que viven en ciudades en las que hay algún centro de trabajo.

```
SELECT nombre, empleado.localidad
FROM empleado
WHERE LOWER(localidad) =
ANY (SELECT LOWER(localidad) from centro);
```

13. Obtener el nombre del jefe de departamento que tiene mayor salario de entre los jefes de departamento.

```
SELECT nombre
FROM empleado
WHERE salario =
  (SELECT max(salario)
   FROM empleado
   WHERE codemplejefe = 1);
```

```
WHERE codemple IN  
(SELECT codemplejefe  
FROM dpto));
```

- 14. Obtener en orden alfabético los salarios y nombres de los empleados cuyo salario sea superior al 60% del máximo salario de la empresa.**

```
SELECT nombre, salario  
FROM empleado  
WHERE salario > (SELECT max(salario) * 0.6 FROM empleado)  
ORDER BY nombre;
```

- 15. Obtener en cuántas ciudades distintas viven los empleados.**

```
SELECT COUNT (DISTINCT localidad) AS localidades_distintas  
FROM empleado;
```

- 16. El nombre y apellidos del empleado que más salario cobra.**

```
SELECT nombre, ape1 || ' ' || ape2 apellidos, salario  
FROM empleado  
WHERE salario =  
(SELECT MAX(salario) FROM empleado);
```

- 17. Obtener las localidades y número de empleados de aquellas en las que viven más de 3 empleados.**

```
SELECT localidad, COUNT(*) AS numempleados  
FROM empleado  
GROUP BY localidad  
HAVING COUNT(*) > 3;
```

18. Obtener, para cada departamento, cuántos empleados trabajan, la suma de sus salarios y la suma de sus comisiones, para aquellos departamentos en los que hay algún empleado cuyo salario es superior a 1700 euros.

```
SELECT coddpto, COUNT(*) AS numempleados, SUM(salario) AS total_salarios,  
SUM(NVL(comision, 0)) AS total_comisiones  
FROM empleado  
WHERE coddpto = ANY (SELECT coddpto FROM empleado WHERE salario > 1700)  
GROUP BY coddpto;
```

19. Obtener el departamento que más empleados tiene.

```
/* OPCION 1: */  
SELECT t.coddpto, t.numempleados  
FROM (SELECT d.coddpto, COUNT(*) AS numempleados  
      FROM empleado e, dpto d  
      WHERE e.coddpto = d.coddpto  
      GROUP BY d.coddpto) T  
WHERE t.numempleados = (SELECT MAX(t2.numempleados)  
                        FROM (SELECT d.coddpto, COUNT(*) AS numempleados  
                              FROM empleado e, dpto d  
                              WHERE e.coddpto = d.coddpto  
                              GROUP BY d.coddpto) T2);
```

```
/* OPCION 2: */  
SELECT coddpto, COUNT(*) AS num_empleados  
FROM empleado  
GROUP BY coddpto  
HAVING COUNT(*) = (SELECT MAX(empleados_numero)  
                  FROM (SELECT COUNT(*) AS empleados_numero  
                        FROM empleado
```

```
GROUP BY coddpto));
```

20. Obtener los nombres de todos los centros y los departamentos que se ubican en cada uno, así como aquellos centros que no tienen departamentos.

```
SELECT c.codcentro, COUNT(*) AS num_departamentos
FROM centro c LEFT JOIN dpto d ON c.codcentro = d.codcentro
GROUP BY c.codcentro;
```

21. Obtener el nombre del departamento de más alto nivel, es decir, aquel que no depende de ningún otro.

```
SELECT coddpto, denominacion
FROM dpto
WHERE coddptodepende IS NULL;
```

22. Obtener todos los departamentos existentes en la empresa y los empleados (si los tiene) que pertenecen a él.

```
/* OPCION 1: */
SELECT e.nombre, d.denominacion
FROM empleado e LEFT JOIN dpto d ON e.coddpto = d.coddpto
ORDER BY denominacion;
```

```
/* OPCION 2: */
SELECT d.denominacion, COUNT(*) as num_empleados
FROM empleado e LEFT JOIN dpto d ON e.coddpto = d.coddpto
GROUP BY d.denominacion;
```

23. Obtener un listado en el que aparezcan todos los departamentos existentes y el departamento del cual depende, si depende de alguno.

```
/* OPCION 1: */  
SELECT coddpto, denominacion, coddptodepende  
FROM dpto  
WHERE coddptodepende IS NOT NULL;
```

```
/* OPCION 2: */  
SELECT d1.coddpto, d1.denominacion, d2.denominacion AS dpto_depends  
FROM dpto d1, dpto d2  
WHERE d1.coddptodepende = d2.coddpto;
```

24. Obtener un listado, ordenado alfabéticamente, donde aparezcan los nombres de los empleados y a continuación el literal "tiene comisión" si la tiene, y "no tiene comisión" si no la tiene.

```
/* OPCION 1: */  
SELECT nombre, DECODE(comision, NULL, 'No tiene comision', 'Tiene comision') AS  
estado_comision1  
FROM empleado  
ORDER BY nombre;
```

```
/* OPCION 2: */  
SELECT nombre,  
CASE WHEN comision IS NULL THEN 'No tiene comision' ELSE 'Tiene comision' END AS  
estado_comision2  
FROM empleado  
ORDER BY NOMBRE;
```


25. Obtener un listado de las localidades en las que hay centros y no vive ningún empleado, ordenado alfabéticamente.

```
SELECT c.localidad, count(*)AS num_empleados
FROM centro c LEFT JOIN empleado e ON LOWER(c.localidad) = LOWER(e.localidad)
GROUP BY c.localidad
HAVING COUNT(*) = 0;
```

26. Obtener un listado de las localidades en las que hay centros y además vive al menos un empleado, ordenado alfabéticamente.

```
SELECT c.localidad, count(*)AS num_empleados
FROM centro c LEFT JOIN empleado e ON UPPER(c.localidad) = UPPER(e.localidad)
GROUP BY c.localidad
HAVING COUNT(*) >= 1;
```

27. Esta cuestión puntúa doble. Se desea dar una gratificación por navidades en función de la antigüedad en la empresa siguiendo estas pautas:

- o Si lleva entre 1 y 5 años, se le dará 100 euros**
- o Si lleva entre 6 y 10 años, se le dará 50 euros por año**
- o Si lleva entre 11 y 20 años, se le dará 70 euros por año**
- o Si lleva más de 21 años, se le dará 100 euros por año**

28. Obtener un listado de los empleados, ordenado alfabéticamente, indicando cuánto le corresponde de gratificación.

```
SELECT nombre, t.antigüedad,
CASE WHEN t.antigüedad >= 1 and t.antigüedad <=5 THEN 100
WHEN t.antigüedad >= 6 and t.antigüedad <=10 THEN 50
WHEN t.antigüedad >= 11 and t.antigüedad <=20 THEN 70
WHEN t.antigüedad > 20 THEN 100
ELSE 0
```

```
END AS gratificacion
FROM (SELECT nombre, FLOOR(MONTHS_BETWEEN(sysdate, fechaingreso)/12) AS antiguedad
FROM empleado)T
ORDER BY nombre;
```

**29. Obtener los nombres y apellidos de los empleados que no son jefes
de departamento.**

```
SELECT nombre, ape1 || ' ' || ape2 apellidos, codemple
FROM empleado
WHERE codemple NOT IN (SELECT codemplejefe FROM dpto)
ORDER BY nombre;
```