* 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;

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

SELECT NOMBRE

FROM EMPLEADO

WHERE COMISION >= SALARIO \* 0.2;

* 1. 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 codemple, coddpto, NOMBRE, SALARIO \* 166.386 AS SALARIO\_PESETAS

FROM EMPLEADO

WHERE (SALARIO + COMISION) > 1800

ORDER BY coddpto, nombre;

* 1. 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

FROM empleado

WHERE (salario \* 1.05) >= (SELECT salario

FROM empleado

WHERE UPPER(nombre)='MARIA'

AND UPPER(ape1)='JAZMIN')

ORDER BY nombre;

* 1. 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, ape1 || ‘ ’ || ape2 as apellidos, floor(months\_between(SYSDATE, fechaingreso) /12) as antiguedad

FROM empleado

ORDER BY antiguedad;

* 1. 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

FROM empleado

WHERE coddpto = ANY(SELECT coddpto FROM DPTO WHERE presupuesto > 50000);

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

SELECT nombre, ape1 || ' ' || ape2 as apellidos

FROM empleado

WHERE (salario + comision) = (SELECT MAX(salario + comision) FROM empleado);

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

SELECT nombre

FROM empleado

WHERE salario < (SELECT MIN(salario) FROM empleado WHERE coddpto = 1)

order by nombre;

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

SELECT nombre

FROM empleado

WHERE coddpto = (SELECT coddpto FROM dpto WHERE CODEMPLEJEFE = 1)

and codemple != 1;

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

SELECT nombre, ape1

FROM empleado

WHERE lower(ape1) LIKE 'p%' or lower(ape1) LIKE 'q%' or lower(ape1) LIKE 'r%' or lower(ape1) LIKE 's%';

* 1. Obtener los empleados cuyo nombre de pila contenga el nombre JUAN.

SELECT nombre

FROM empleado

WHERE upper(nombre) LIKE '%JUAN%';

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

SELECT nombre

FROM empleado

WHERE UPPER(localidad) = any(SELECT upper(localidad) FROM centro);

* 1. 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 codemple in (SELECT codemplejefe

FROM DPTO)

)

and codemple in (SELECT codemplejefe FROM DPTO);

* 1. 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;

* 1. Obtener en cuántas ciudades distintas viven los empleados.

SELECT COUNT(DISTINCT localidad)

FROM empleado;

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

SELECt nombre, ape1 || ' ' || ape2 as apellidos

FROM empleado

WHERE salario = (SELECT MAX(salario) from empleado)

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

SELECT localidad, count(\*) as num\_empleados

FROM empleado

GROUP BY localidad

having count(\*) > 3;

* 1. 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 num\_empleados, sum(salario) as suma\_salario, sum(nvl(comision, 0)) as suma\_comisiones

FROM empleado E

WHERE coddpto = any(SELECT coddpto from empleado where salario > 1700 and e.coddpto = coddpto)

GROUP BY coddpto

* 1. Obtener el departamento que más empleados tiene.

SELECT t.denominacion, t.num\_empleados

FROM (

SELECT d.denominacion, count(\*) as num\_empleados

FROM empleado e, dpto d

WHERE d.coddpto = e.coddpto

GROUP BY d.denominacion

) T

WHERE t.num\_empleados = (SELECT max(t2.num\_empleados) FROM (

SELECT d.denominacion, count(\*) as num\_empleados

FROM empleado e, dpto d

WHERE d.coddpto = e.coddpto

GROUP BY d.denominacion) T2

);

* 1. 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 direccion, count(\*)

FROM DPTO D LEFT JOIN CENTRO C ON c.codcentro = d.codcentro

GROUP BY direccion;

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

SELECT denominacion

FROM DPTO

WHERE coddptodepende is null;

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

SELECT denominacion, count(e.coddpto) as num\_empleados

FROM dpto d left join empleado e on e.coddpto = d.coddpto

GROUP BY denominacion;

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

SELECT d2.denominacion as dpto, nvl(d1.denominacion, d2.denominacion) as depende

FROM dpto d1 right join dpto d2 on d1.coddpto = d2.coddptodepende;

* 1. 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.

SELECT CASE WHEN

comision is null

then nombre || ' no tiene comision'

else nombre || ' tiene comision' end as info\_emp

FROM empleado

order by nombre

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

SELECT c.localidad

FROM centro c left join empleado e on upper(c.localidad) = upper(e.localidad)

GROUP BY c.localidad

having count(\*) = 0

Order by c.localidad;

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

SELECT c.localidad

FROM centro c, empleado e

WHERE upper(c.localidad) = upper(e.localidad)

GROUP BY c.localidad

having count(\*) > 1

Order by c.localidad;

* 1. Esta cuestión puntúa doble. Se desea dar una gratificación por navidades en función de la antigüedad en la empresa siguiendo estas pautas:
     + Si lleva entre 1 y 5 años, se le dará 100 euros
     + Si lleva entre 6 y 10 años, se le dará 50 euros por año
     + Si lleva entre 11 y 20 años, se le dará 70 euros por año
     + Si lleva más de 21 años, se le dará 100 euros por año

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

SELECT nombre, floor(months\_between(SYSDATE, fechaingreso)/12) as antiguedad,

case when floor(months\_between(SYSDATE, fechaingreso)/12) >= 1 and floor(months\_between(SYSDATE, fechaingreso)/12) <= 5 THEN 100

when floor(months\_between(SYSDATE, fechaingreso)/12) >= 6 and floor(months\_between(SYSDATe, fechaingreso)/12) <= 10 THEN 50

when floor(months\_between(SYSDATE, fechaingreso)/12) >= 11 and floor(months\_between(SYSDATe, fechaingreso)/12) <= 20 THEN 70

else 100 end as gratificacion

FROM empleado

order by nombre

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

SELECT nombre, ape1 || ' ' || ape2 as apellidos

FROM empleado

WHERE codemple not in (SELECT codemplejefe from dpto);