

TAREA BD04 – BEATRIZ GARCÍA HERRERO

Como tarea realiza los siguientes requerimientos en SQL:

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

```
SELECT nombre, ape1, ape2, salario
FROM empleado
WHERE salario>1000
ORDER BY nombre, ape1, ape2;
```

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

```
SELECT nombre, ape1, ape2
FROM empleado
WHERE comision >= salario*0.20;
```

```
SELECT nombre, ape1, ape2
FROM EMPLEADO
WHERE comision*100/salario >=20;
```

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 codemple, coddpto, nombre, ape1, ape2, salario*166.386 SUELDO_PESETAS
FROM empleado
WHERE (salario+comision)>1800
ORDER BY coddpto, nombre, ape1, ape2;
```

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, ape1, ape2
FROM empleado
WHERE salario >= (SELECT salario*1.05 FROM empleado WHERE nombre ='MARIA'
AND ape1='JAZMIN')
ORDER BY nombre, ape1, ape2;
```

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 FLOOR((SYSDATE-fechaingreso)/365) "ANTIGUEDAD", nombre, ape1, ape2
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, ape1, ape2
FROM empleado NATURAL JOIN dpto
WHERE presupuesto > 50000;
```

```
SELECT nombre, ape1, ape2
FROM empleado
WHERE coddpto IN (SELECT coddpto FROM dpto WHERE presupuesto > 50000);
```

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

```
SELECT nombre, ape1, ape2
FROM empleado
WHERE salario+comision IN (SELECT MAX(salario+comision) 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, ape1, ape2
FROM empleado
WHERE salario < (SELECT MIN(SALARIO) FROM empleado WHERE coddpto=1)
ORDER BY nombre, ape1, ape2;
```

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

```
SELECT nombre, ape1, ape2
FROM empleado
WHERE coddpto =(SELECT coddpto FROM dpto WHERE codemplejefe=1);
```

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

```
SELECT nombre, ape1, ape2
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, ape1, ape2
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, ape1, ape2
FROM empleado, centro
WHERE UPPER(empleado.localidad) = UPPER(centro.localidad);
```

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

```
SELECT nombre, ape1, ape2, salario
FROM empleado e, dpto d
WHERE e.codemple=d.codemplejefe AND salario IN (SELECT MAX(salario) FROM
empleado WHERE e.codemple=d.codemplejefe);
```

- 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, ape1, ape2, salario
FROM empleado
WHERE salario > (SELECT MAX(salario*0.6) FROM empleado)
ORDER BY nombre, ape1, ape2;
```

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

```
SELECT COUNT(DISTINCT localidad) "NUM_LOCALIDADES"
FROM empleado;
```

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

```
SELECT nombre, ape1, ape2
FROM empleado
WHERE salario IN (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(codemple)
FROM empleado
GROUP BY localidad
HAVING COUNT(codemple) > 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(codemple), SUM(salario), SUM(comision)
FROM empleado
WHERE coddpto IN (SELECT coddpto FROM empleado WHERE salario > 1700)
GROUP BY coddpto;
```

19. Obtener el departamento que más empleados tiene.

No logro dar con la solución, consigo contar el departamento con más empleados, pero no consigo nombrar dicho departamento.

```
SELECT MAX(COUNT(codemple))  
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 coddpto, denominacion, codcentro, coddptodepende  
FROM dpto;
```

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.

```
SELECT denominacion, nombre, ape1, ape2  
FROM dpto d JOIN empleado e ON (d.coddpto = e.coddpto)
```

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

```
SELECT coddpto, denominacion, coddptodepende  
FROM dpto;
```

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.

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

```
SELECT UPPER(localidad)  
FROM centro  
MINUS  
SELECT UPPER(localidad)  
FROM empleado;
```

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

```
SELECT UPPER(localidad)
FROM centro
INTERSECT
SELECT UPPER(localidad)
FROM empleado;
```

- 27. 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

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

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

```
SELECT nombre, ape1, ape2, codemple
FROM empleado NATURAL JOIN dpto
WHERE empleado.codemple NOT IN codemplejefe;
```