

# Tarea 4

BASES DE DATOS

IRENE CALDELAS FERNÁNDEZ

## Tarea 4 BD – Irene Caldelas Fernández

The screenshot shows a web browser window with the URL `https://fpdistancia.educa.jcyl.es/mod/assign/view.php?id=84761`. The page displays a course titled "Bases de Datos" and a list of participants. A Windows command prompt window is open, showing the following commands and output:

```
C:\Windows\system32>conn sys as sysdba
"conn" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Windows\system32>sqlplus conn sys as sysdba

SQL*Plus: Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Utilice SQL*Plus para ejecutar sentencias SQL, PL/SQL y SQL*Plus.

Sintaxis 1: sqlplus -H | -V

-H          Muestra la versión de SQL*Plus y
            ayuda de sintaxis.
-V          Muestra la versión de SQL*Plus.

Sintaxis 2: sqlplus [ [<option>] [{logon | /nolog}] [<start>] ]
<opción> es: [-AC] [-C <versión>] [-F] [-L] [-M "<opciones>"] [-NOLOGINTIME]
```

The command prompt window is then closed, and the browser window shows the course content. The course content includes a list of topics and a list of participants. The participants list shows the name "IRENE CALDELAS FERNÁNDEZ" and a profile picture. The course content also includes a list of topics and a list of participants.

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

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

```
C:\Windows\system32>sqlplus sys as sysdba

SQL*Plus: Release 18.0.0.0.0 - Production on Jue Mar 24 01:46:57 2022
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Introduzca la contraseña:

Conectado a:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> create user c##dpto identified by dpto default tablespace users;

Usuario creado.

SQL>
```

## Tarea 4 BD – Irene Caldelas Fernández

The screenshot shows a web browser window displaying a course page for 'Tarea 4 BD' on the 'fpdistancia.educa.jcyl.es' website. The page includes a sidebar with 'Bases de Datos' and 'Participantes' sections. The main content area shows a list of topics and a description of the task. Overlaid on the browser is a terminal window titled 'Administrador: Símbolo del sistema - sqlplus sys as sysdba'. The terminal shows the following commands and output:

```
C:\Windows\system32\sqlplus sys as sysdba
SQL*Plus: Release 18.0.0.0.0 - Production on Jue Mar 24 01:46:57 2022
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Introduzca la contraseña:

Conectado a:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> create user c##dpto identified by dpto default tablespace users;
Usuario creado.

SQL> grant connect, resource, DBA to c##dpto;
Concesión terminada correctamente.

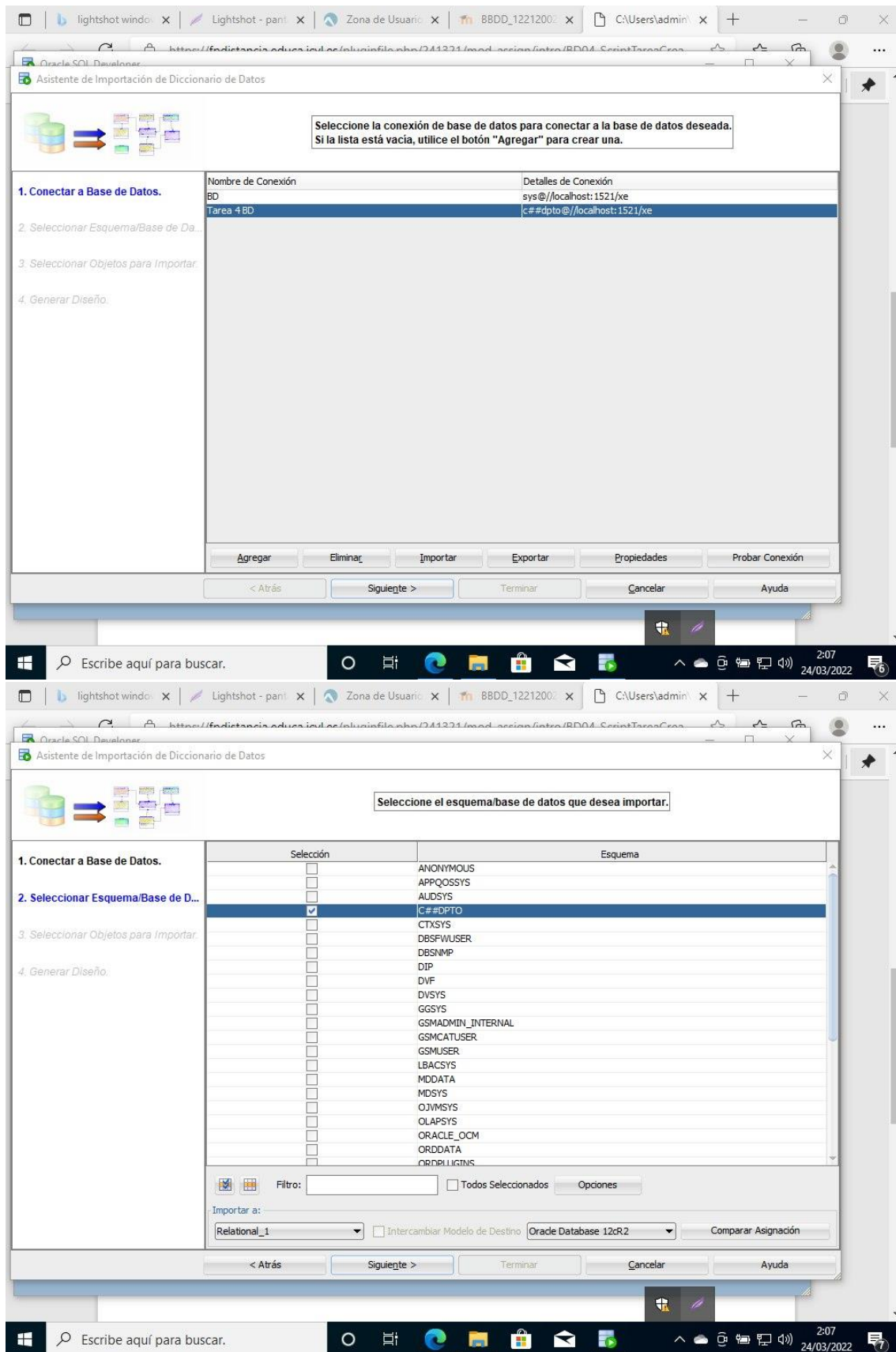
SQL> CONNECT c##dpto/dpto;
Conectado.
SQL>
```

Below the terminal window, the 'Oracle SQL Developer' application is open, showing the 'Nueva / Seleccionar Conexión a Base de Datos' dialog box. The dialog is configured with the following details:

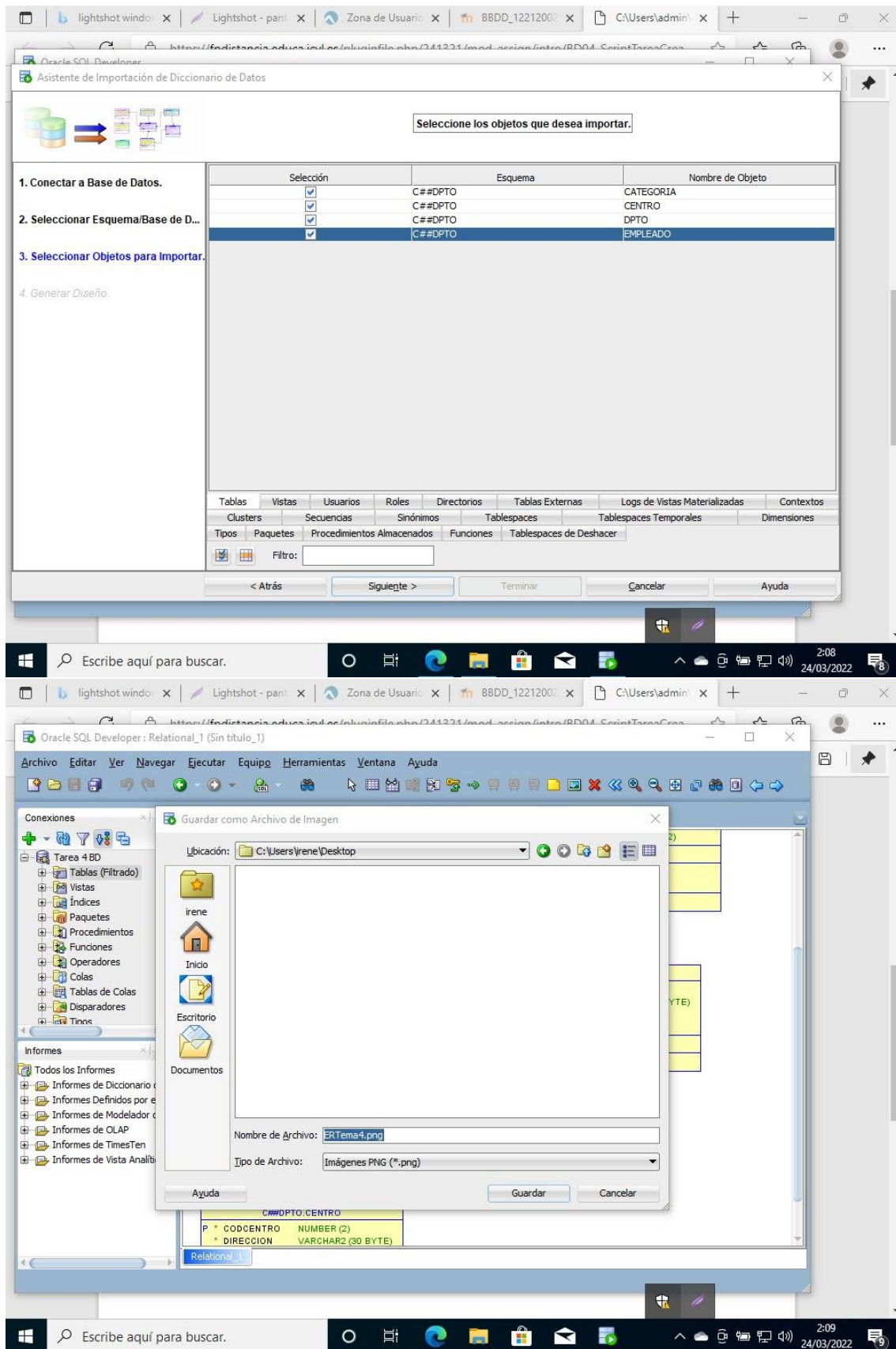
- Name: Tarea 4 BD
- Tipo de Base de Datos: Oracle
- Información de usuario: Usuario de Proxy
- Tipo de autenticación: Por defecto
- Usuario: c##dpto
- Rol: valor por defecto
- Contraseña: \*\*\*\*
- Tipo de Conexión: Básico
- Detalles: Avanzado
- Nombre del Host: localhost
- Puerto: 1521
- SID: xe
- Nombre del Servicio: (empty)

The dialog box also shows the 'Estado: Correcto' and buttons for 'Ayuda', 'Guardar', 'Borrar', 'Probar', 'Conectar', and 'Cancelar'.

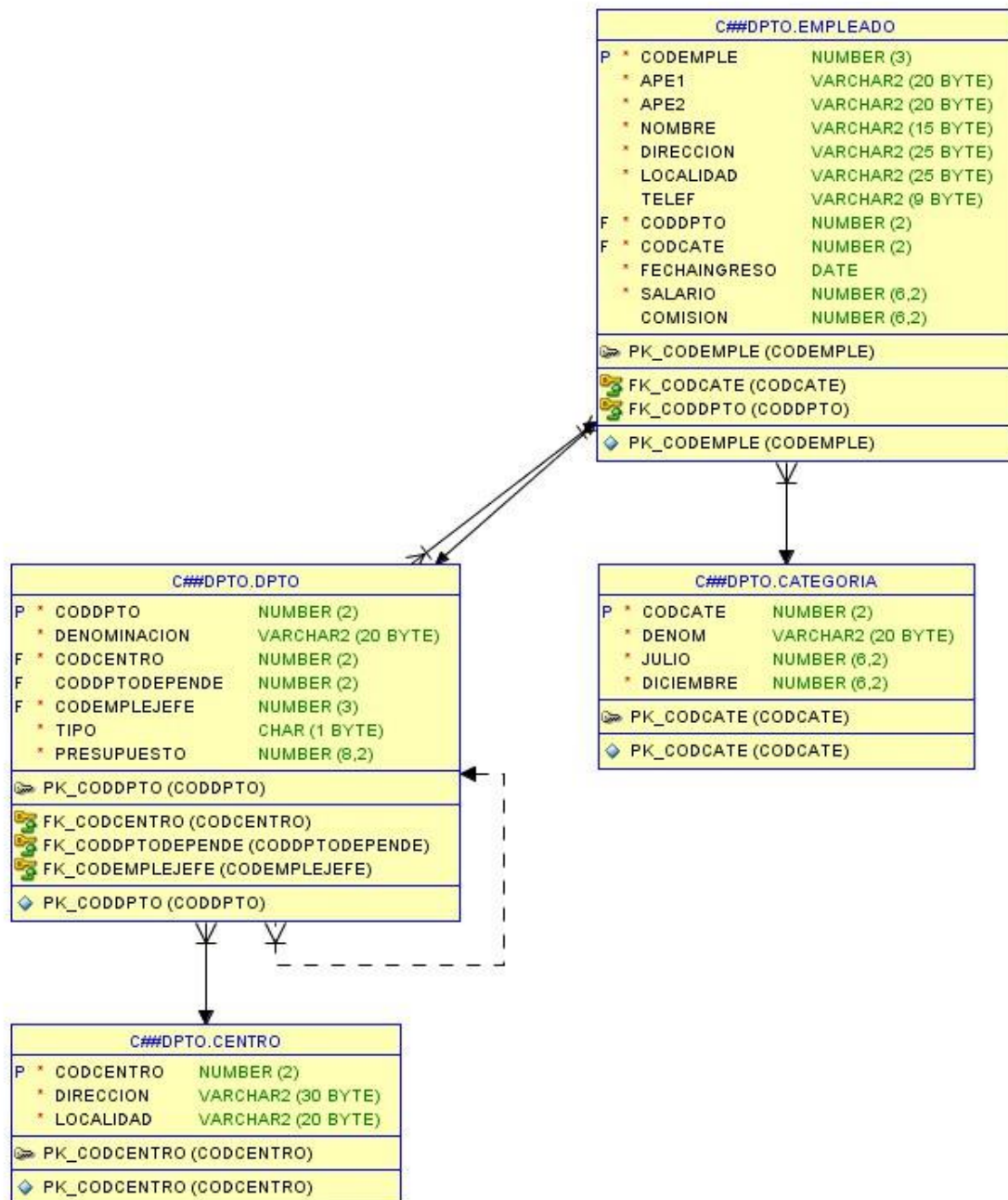
## Tarea 4 BD – Irene Caldelas Fernández



## Tarea 4 BD – Irene Caldelas Fernández







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 salario ASC;
```

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

```
SELECT nombre 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 codemple, coddpto, nombre, salario*166.3860 from empleado WHERE salario+comision = 1800 ORDER BY coddpto, nombre ASC;
```

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

5. Obtener una 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, EXTRACT(YEAR FROM SYSDATE)-EXTRACT(YEAR FROM fechaingreso) FROM empleado ORDER BY fechaingreso;
```

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 DISTINCT e.nombre FROM empleado e  
INNER JOIN dpto d ON d.coddpto = e.coddpto WHERE d.presupuesto > 5000;
```

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 > ANY(SELECT 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 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 e.nombre FROM empleado e  
INNER JOIN dpto d ON d.coddpto = e.coddpto WHERE d.codemplejefe = 1;
```

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

```
SELECT nombre 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 e.nombre FROM empleado e  
INNER JOIN dpto d ON d.coddpto = e.coddpto  
INNER JOIN centro c ON c.codcentro = d.codcentro;
```

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

```
SELECT e.nombre FROM empleado e  
  
INNER JOIN dpto d ON d.coddpto = e.coddpto  
  
WHERE salario = (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 salario, nombre FROM empleado WHERE salario > (SELECT  
MAX(salario)*0.6 FROM empleado) ORDER BY nombre ASC;
```



15. Obtener en cuántas ciudades distintas viven los empleados.

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

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

```
SELECT nombre, ape1 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(*) 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 d.coddpto, COUNT(*), SUM(e.salario), SUM(e.comision) FROM dpto d
INNER JOIN empleado e ON e.coddpto = d.coddpto WHERE salario > ANY
(SELECT salario FROM empleado WHERE salario > 1700) GROUP BY d.coddpto;
```

19. Obtener el departamento que más empleados tiene.

```
SELECT d.coddpto as departamento, COUNT(*) as numEmpleados
FROM dpto d
INNER JOIN empleado e ON e.coddpto = d.coddpto
GROUP BY d.coddpto
HAVING COUNT(*) =
(SELECT MAX(Count(*))
FROM dpto d
INNER JOIN empleado e ON e.coddpto = d.coddpto
GROUP BY d.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.localidad, d.denominacion FROM centro c
```

```
INNER JOIN dpto d ON d.codcentro = c.codcentro;
```

21. 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;
```

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

```
SELECT d.denominacion, e.nombre FROM dpto d  
  
INNER JOIN empleado e ON e.coddpto = d.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.

```
SELECT nombre, 'No tiene comision' as comision  
  
FROM empleado  
  
WHERE comision IS null  
  
UNION  
  
SELECT nombre, 'Tiene comision' as comision  
  
FROM empleado  
  
WHERE comision IS NOT null  
  
ORDER BY nombre ASC;
```

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

```
SELECT DISTINCT localidad  
  
FROM centro  
  
WHERE localidad NOT IN
```

```
(SELECT DISTINCT c.localidad  
  
FROM centro c  
  
INNER JOIN dpto d ON d.codcentro = c.codcentro  
  
INNER JOIN empleado e ON e.coddpto = d.coddpto);
```

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

```
SELECT DISTINCT c.localidad  
  
FROM centro c  
  
INNER JOIN dpto d ON d.codcentro = c.codcentro  
  
INNER JOIN empleado e ON e.coddpto = d.coddpto
```

27. Esta cuestión puntúa doble. Se da 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

```
SELECT e.nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM  
e.fechaingreso)) AS antigüedad, c.diciembre+100 as gratificacion  
  
FROM empleado e  
  
INNER JOIN categoria c ON c.codcate = e.codcate  
  
WHERE ((EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM  
fechaingreso)) BETWEEN 1 AND 5);  
  
UNION  
  
SELECT e.nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM  
e.fechaingreso)) AS antigüedad, (((EXTRACT(YEAR FROM SYSDATE) -  
EXTRACT(YEAR FROM fechaingreso))*50)+c.diciembre) as gratificacion  
  
FROM empleado e  
  
INNER JOIN categoria c ON c.codcate = e.codcate
```

```

WHERE ((EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
fechaingreso)) BETWEEN 6 AND 10) ;

UNION

SELECT e.nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
e.fechaingreso)) AS antiguedad, (((EXTRACT(YEAR FROM SYSDATE) -
EXTRACT(YEAR FROM fechaingreso))*70)+c.diciembre) AS gratificacion

FROM empleado e

INNER JOIN categoria c ON c.codcate = e.codcate

WHERE ((EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
fechaingreso)) BETWEEN 11 AND 20);

UNION

SELECT e.nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
e.fechaingreso)) AS antiguedad, (((EXTRACT(YEAR FROM SYSDATE) -
EXTRACT(YEAR FROM fechaingreso))*100)+c.diciembre) as gratificacion

FROM empleado e

INNER JOIN categoria c ON c.codcate = e.codcate

WHERE (EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
fechaingreso)) >= 21

```

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

```

SELECT nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
fechaingreso)) AS antiguedad, 100 as gratificacion

FROM empleado

WHERE ((EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM fechaingreso))
BETWEEN 1 AND 5)

UNION ALL

SELECT nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
fechaingreso)) AS antiguedad, (((EXTRACT(YEAR FROM SYSDATE) -
EXTRACT(YEAR FROM fechaingreso))*50)) as gratificacion

FROM empleado

```

```
WHERE ((EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM fechaingreso))
BETWEEN 6 AND 10)

UNION ALL

SELECT nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
fechaingreso)) AS antiguedad, (((EXTRACT(YEAR FROM SYSDATE) -
EXTRACT(YEAR FROM fechaingreso))*70)) AS gratificacion

FROM empleado

WHERE ((EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM fechaingreso))
BETWEEN 11 AND 20)

UNION ALL

SELECT nombre,(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
fechaingreso)) AS antiguedad, (((EXTRACT(YEAR FROM SYSDATE) -
EXTRACT(YEAR FROM fechaingreso))*100)) as gratificacion

FROM empleado

WHERE (EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM fechaingreso))
> 20

ORDER BY nombre ASC;
```

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

```
SELECT e.nombre, e.ape1, e.ape2 FROM empleado e

INNER JOIN dpto d ON d.coddpto = e.coddpto

WHERE d.codemplejefe != e.codemple;
```