

ACTIVIDAD 1.

La tarea consiste en realizar consultas en SQL sobre el Sistema de Información descrito a continuación:

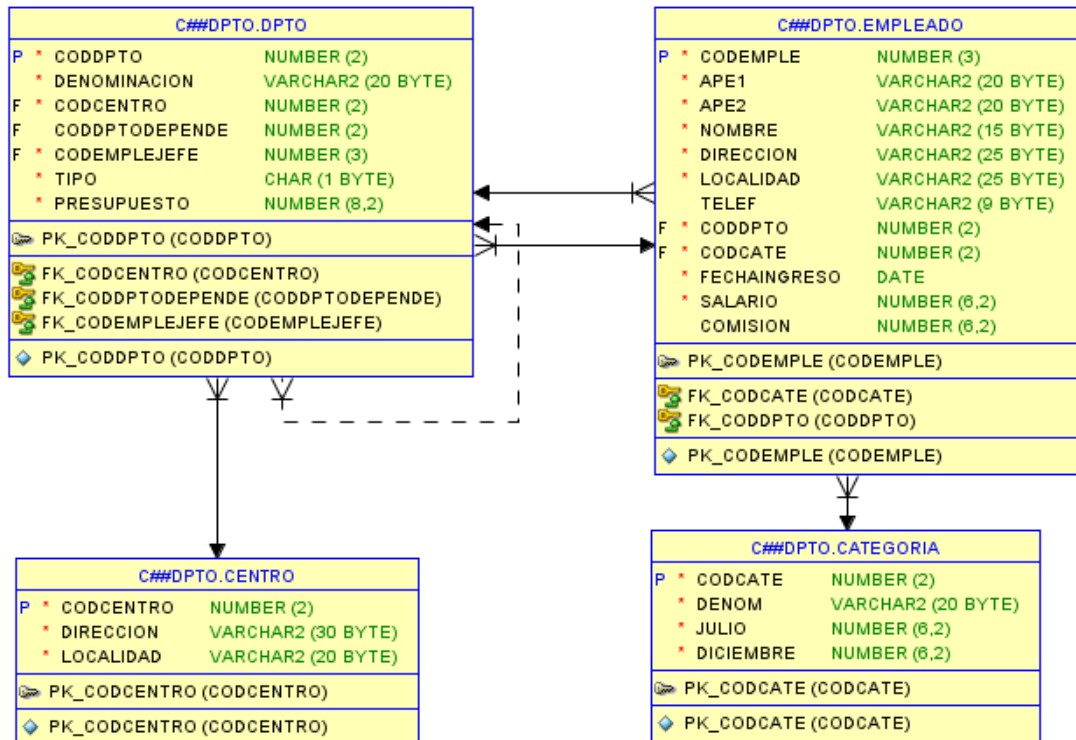
- Una empresa está organizada en Departamentos.
- Cada departamento se ubica físicamente en un Centro de trabajo, pudiendo contener un Centro más de un Departamento.
- Cada departamento tiene un director que es un empleado de la empresa y puede desempeñar el cargo en Propiedad o en Funciones.
- Cada departamento depende de otro departamento, excepto el de más alto nivel, que no depende de ninguno.
- Cada departamento tiene a su cargo a varios departamentos, excepto los departamentos de más bajo nivel, de los cuales no depende ninguno.
- Cada empleado tiene una categoría que lleva asociada un complemento de Julio y otro de Navidad, y pertenece a un departamento.
- De los Centros interesa almacenar el código, que es único para cada Centro, Dirección y Localidad.
- De los Departamentos interesa almacenar el código, que es único para cada dpto., y la siguiente información: Denominación, Centro en el que está ubicado, departamento del que depende, en el caso de dependa de alguno, Empleado jefe, Tipo de empleado jefe (si es en propiedad o en funciones, y presupuesto asignado).
- De los Empleados interesa almacenar el código, que es único para cada empleado, y la siguiente información nombre, ap1, ape2, dirección, localidad, teléf., Departamento en el que trabaja, categoría que tiene, comisión (solo la tendrán algunos empleados, otros tendrán nulos), y salario
- De las Categorías interesa almacenar el código, que es único para cada Categoría, la denominación de categoría y el importe de paga extra de julio y el importe de septiembre.

En el script BD04_Tarea_creadptoyCarga.sql, contenido en el PDF del enlace, tienes las sentencias para crear el usuario c##dpto con clave dpto y las tablas y añadir datos para probar las sentencias SQL que se piden en la tarea.

[PDF con sentencias script creadptoyCarga.sql](#)

Crea un script con las sentencias del pdf y ejecútalo. El script conecta como administrador con "sys as sysdba", crea el usuario c##dpto, las tablas e inserta contenido en ellas.

Obtén el diagrama del modelo relacional siguiendo los pasos del enlace [Obtención del modelo entidad relación a partir de las tablas con SQLDeveloper](#) para entender bien el esquema, sus tablas y la relación entre ellas.



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 AS Apellidos, SALARIO
FROM EMPLEADO
WHERE SALARIO > 1000
ORDER BY NOMBRE;
```

	NOMBRE	APELLIDOS	SALARIO
1	ALFONSO	ROS SANTON	2000
2	ANA	LOPEZ GARCIA	3000
3	ANGEL	CORTES LOPEZ	2000
4	ANGUSTIAS	ZAPATERO GALLARDON	2000
5	BLANCA	FLOR LUZ	3000
6	FABIOLA	SANCHEZ LUZ	2500
7	GANDI	LOPEZ ITURRIALDE	1500
8	JUAN	FERNANDEZ MORON	2000
9	PAZ	RAJOY AZNAR	2000

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

```
SELECT NOMBRE, APE1 || ' ' || APE2 AS Apellidos
FROM EMPLEADO
WHERE COMISION > 0.2 * SALARIO;
```

NOMBRE	APELLIDOS
--------	-----------

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 AS Codigo_Empleado, CODDPTO AS Codigo_Departamento, NOMBRE, APE1
|| ' ' || APE2 AS Apellidos, (SALARIO + NVL(COMISION, 0)) * 166 AS Sueldo_Total_Ptas
FROM EMPLEADO
WHERE (SALARIO + NVL(COMISION, 0)) > 1800
ORDER BY CODDPTO, NOMBRE;
```

	CODIGO_EMPLEADO	CODIGO_DEPARTAMENTO	NOMBRE	APELLIDOS	SUELDO_TOTAL_PTAS
1	1	1	ANA	LOPEZ GARCIA	498000
2	2	1	JUAN	FERNANDEZ MORON	332000
3	3	2	ANGEL	CORTES LOPEZ	332000
4	4	3	FABIOLA	SANCHEZ LUZ	415000
5	5	3	PAZ	RAJOY AZNAR	353580
6	6	5	ANGUSTIAS	ZAPATERO GALLARDON	332000
7	7	6	BLANCA	FLOR LUZ	519580
8	8	7	ALFONSO	ROS SANTON	332000

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 AS Apellidos
FROM EMPLEADO
WHERE SALARIO >= (SELECT SALARIO FROM EMPLEADO WHERE NOMBRE = 'MARIA' AND APE1
= 'JAZMIN') * 1.05
ORDER BY NOMBRE;
```

	NOMBRE	APELLIDOS
1	ALFONSO	ROS SANTON
2	ANA	LOPEZ GARCIA
3	ANGEL	CORTES LOPEZ
4	ANGUSTIAS	ZAPATERO GALLARDON
5	BLANCA	FLOR LUZ
6	FABIOLA	SANCHEZ LUZ
7	GANDI	LOPEZ ITURRIALDE
8	JUAN	FERNANDEZ MORON
9	PAZ	RAJOY AZNAR

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, APE1 || ' ' || APE2 AS Apellidos, TRUNC(MONTHS_BETWEEN(SYSDATE,
FECHAINGRESO)/12, 0) AS Antigüedad
FROM EMPLEADO
ORDER BY Antigüedad DESC;
```

	NOMBRE	APELLIDOS	ANTIGÜEDAD
1	GANDI	LOPEZ ITURRIALDE	23
2	ANA	LOPEZ GARCIA	21
3	PAZ	RAJOY AZNAR	21
4	BLANCA	FLOR LUZ	21
5	ANGUSTIAS	ZAPATERO GALLARDON	21
6	MARIA	JAZMIN EXPOSITO	20
7	FABIOLA	SANCHEZ LUZ	20
8	JUAN	FERNANDEZ MORON	19
9	ALFONSO	ROS SANTON	18
10	ANGEL	CORTES LOPEZ	18

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 AS Apellidos
FROM EMPLEADO
WHERE CODDPTO = SOME(SELECT CODDPTO FROM DPTO WHERE PRESUPUESTO > 50000);
```

	NOMBRE	APELLIDOS
1	ANA	LOPEZ GARCIA
2	JUAN	FERNANDEZ MORON
3	ANGUSTIAS	ZAPATERO GALLARDON
4	GANDI	LOPEZ ITURRIALDE
5	MARIA	JAZMIN EXPOSITO

7. 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 + NVL(COMISION, 0)) = (SELECT MAX(SALARIO + NVL(COMISION, 0)) FROM EMPLEADO);
```

	NOMBRE	APELLIDOS
1	BLANCA	FLOR LUZ

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 AS Apellidos
FROM EMPLEADO
WHERE SALARIO < (SELECT MIN(SALARIO) FROM EMPLEADO WHERE CODDPTO = 1)
ORDER BY NOMBRE;
```

	NOMBRE	APELLIDOS
1	GANDI	LOPEZ ITURRIALDE
2	MARIA	JAZMIN EXPOSITO

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

```
SELECT NOMBRE, APE1 || ' ' || APE2 AS Apellidos
FROM EMPLEADO
WHERE CODDPTO = (SELECT CODDPTO FROM DPTO WHERE CODEMPLEJEFE=1);
```

	NOMBRE	APELLIDOS
1	ANA	LOPEZ GARCIA
2	JUAN	FERNANDEZ MORON

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

```
SELECT NOMBRE, APE1 || ' ' || APE2 AS Apellidos
FROM EMPLEADO
WHERE LOWER(APE1) LIKE 'p%' OR LOWER(APE1) LIKE 'q%' OR LOWER(APE1) LIKE 'r%' OR LOWER(APE1) LIKE 's%';
```

	NOMBRE	APELLIDOS
1	FABIOLA	SANCHEZ LUZ
2	PAZ	RAJOY AZNAR
3	ALFONSO	ROS SANTON

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

```
SELECT NOMBRE, APE1 || ' ' || APE2 AS Apellidos
FROM EMPLEADO
WHERE UPPER(NOMBRE) LIKE '%JUAN%';
```

	NOMBRE	APELLIDOS
1	JUAN	FERNANDEZ MORON

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

```
SELECT NOMBRE, APE1 || ' ' || APE2 AS Apellidos
FROM EMPLEADO
WHERE UPPER(LOCALIDAD) IN(SELECT UPPER(LOCALIDAD) FROM CENTRO);
```

	NOMBRE	APELLIDOS
1	JUAN	FERNANDEZ MORON
2	GANDI	LOPEZ ITURRIALDE
3	ANA	LOPEZ GARCIA
4	ANGUSTIAS	ZAPATERO GALLARDON
5	ALFONSO	ROS SANTON
6	MARIA	JAZMIN EXPOSITO
7	FABIOLA	SANCHEZ LUZ
8	BLANCA	FLOR LUZ

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

```
SELECT NOMBRE, APE1 || ' ' || APE2 AS Apellidos
FROM EMPLEADO JOIN DPTO ON (EMPLEADO.CODEMPLE = DPTO.CODEMPLEJEFE)
WHERE EMPLEADO.CODEMPLE = CODEMPLEJEFE
AND SALARIO = (SELECT MAX(SALARIO) FROM EMPLEADO);
```

	NOMBRE	APELLIDOS
1	ANA	LOPEZ GARCIA
2	BLANCA	FLOR LUZ

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, APE1 || ' ' || APE2 AS Apellidos
```

```
FROM EMPLEADO
WHERE SALARIO > (SELECT MAX(SALARIO)*0.6 FROM EMPLEADO)
ORDER BY NOMBRE;
```

	SALARIO	NOMBRE	APELLIDOS
1	2000	ALFONSO	ROS SANTON
2	3000	ANA	LOPEZ GARCIA
3	2000	ANGEL	CORTES LOPEZ
4	2000	ANGUSTIAS	ZAPATERO GALLARDON
5	3000	BLANCA	FLOR LUZ
6	2500	FABIOLA	SANCHEZ LUZ
7	2000	JUAN	FERNANDEZ MORON
8	2000	PAZ	RAJOY AZNAR

15. Obtener en cuántas ciudades distintas viven los empleados.

```
SELECT LOCALIDAD
FROM EMPLEADO
GROUP BY LOCALIDAD;
```

	LOCALIDAD
1	MADRID
2	SEVILLA
3	TARRAGONA
4	BARACALDO
5	JAEN

16. 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);
```

	NOMBRE	APELLIDOS
1	ANA	LOPEZ GARCIA
2	BLANCA	FLOR LUZ

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

```
SELECT LOCALIDAD, COUNT(CODEMPLE) AS Num_Empleados
FROM EMPLEADO
GROUP BY LOCALIDAD
HAVING COUNT(LOCALIDAD)>3;
```

	LOCALIDAD	NUM_EMPLEADOS
1	MADRID	4

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 DENOMINACION, COUNT(CODEMPLE) AS Num_Empleados, SUM(SALARIO) AS Total_Salarios, SUM(NVL(COMISION, 0)) AS Total_Comisiones
FROM EMPLEADO JOIN DPTO ON (EMPLEADO.CODDPTO = DPTO.CODDPTO)
WHERE DPTO.CODDPTO = SOME(SELECT CODDPTO FROM EMPLEADO WHERE SALARIO > 1700)
GROUP BY DENOMINACION;
```

	DENOMINACION	NUM_EMPLEADOS	TOTAL_SALARIOS	TOTAL_COMISIONES
1	DIRECCIÓN	2	5000	0
2	RECURSOS HUMANOS	2	4500	130
3	CENTRAL COMERCIAL	3	4500	410
4	COMERCIAL CENTRO	1	3000	130
5	COMERCIAL SUR	1	2000	0
6	ADMINISTRACION	1	2000	0

19. Obtener el departamento que más empleados tiene.

```
SELECT DENOMINACION AS Departamento
FROM DPTO NATURAL JOIN (SELECT CODDPTO, Num_Empleados
FROM (SELECT CODDPTO, COUNT(CODEMPLE) AS Num_Empleados
FROM EMPLEADO
GROUP BY CODDPTO))
WHERE Num_Empleados = (SELECT MAX(Num_Empleados) FROM (SELECT CODDPTO,
COUNT(CODEMPLE) AS Num_Empleados
FROM EMPLEADO
GROUP BY CODDPTO));
```

	DEPARTAMENTO
1	CENTRAL COMERCIAL

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 CENTRO.CODCENTRO, LOCALIDAD, DENOMINACION AS Departamento
FROM CENTRO LEFT OUTER JOIN DPTO ON (CENTRO.CODCENTRO = DPTO.CODCENTRO);
```

	CODCENTRO	LOCALIDAD	DEPARTAMENTO
1	1	Tarragona	DIRECCIÓN
2	1	Tarragona	ADMINISTRACION
3	1	Tarragona	RECURSOS HUMANOS
4	1	Tarragona	CENTRAL COMERCIAL
5	2	Madrid	COMERCIAL CENTRO
6	3	Sevilla	COMERCIAL SUR

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

```
SELECT DENOMINACION Departamento
FROM DPTO
WHERE CODDPTODEPENDE IS NULL;
```

	DEPARTAMENTO
1	DIRECCIÓN

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

```
SELECT DENOMINACION Departamento, NOMBRE, APE1 || ' ' || APE2 Apellidos
FROM EMPLEADO RIGHT OUTER JOIN DPTO ON (EMPLEADO.CODDPTO = DPTO.CODDPTO);
```

	DEPARTAMENTO	NOMBRE	APELLIDOS
1	DIRECCIÓN	ANA	LOPEZ GARCIA
2	DIRECCIÓN	JUAN	FERNANDEZ MORON
3	ADMINISTRACION	ANGEL	CORTES LOPEZ
4	RECURSOS HUMANOS	FABIOLA	SANCHEZ LUZ
5	RECURSOS HUMANOS	PAZ	RAJOY AZNAR
6	CENTRAL COMERCIAL	ANGUSTIAS	ZAPATERO GALLARDON
7	CENTRAL COMERCIAL	GANDI	LOPEZ ITURRIALDE
8	CENTRAL COMERCIAL	MARIA	JAZMIN EXPOSITO
9	COMERCIAL CENTRO	BLANCA	FLOR LUZ
10	COMERCIAL SUR	ALFONSO	ROS SANTON

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

```
SELECT DPTO.DENOMINACION Departamento, DPTODEPENDE.DENOMINACION Depende_de
FROM DPTO LEFT JOIN DPTO DPTODEPENDE ON (DPTO.CODDPTODEPENDE = DPTODEPENDE.CODDPTO);
```

	DEPARTAMENTO	DEPENDE_DE
1	ADMINISTRACION	DIRECCIÓN
2	RECURSOS HUMANOS	DIRECCIÓN
3	CENTRAL COMERCIAL	DIRECCIÓN
4	COMERCIAL CENTRO	CENTRAL COMERCIAL
5	COMERCIAL SUR	CENTRAL COMERCIAL
6	DIRECCIÓN	(null)

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 || ' ' || APE1 || ' ' || APE2 "Nombre completo", DECODE(NVL(COMISION, 0),
0, 'No tiene comisión', 'Tiene comisión') Comisión
FROM EMPLEADO
ORDER BY NOMBRE;
```

	Nombre completo	COMISIÓN
1	ALFONSO ROS SANTON	No tiene comisión
2	ANA LOPEZ GARCIA	No tiene comisión
3	ANGEL CORTES LOPEZ	No tiene comisión
4	ANGUSTIAS ZAPATERO GALLARDON	No tiene comisión
5	BLANCA FLOR LUZ	Tiene comisión
6	FABIOLA SANCHEZ LUZ	No tiene comisión
7	GANDI LOPEZ ITURRIALDE	Tiene comisión
8	JUAN FERNANDEZ MORON	No tiene comisión
9	MARIA JAZMIN EXPOSITO	Tiene comisión
10	PAZ RAJOY AZNAR	Tiene comisión

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

```
SELECT LOCALIDAD
FROM CENTRO
WHERE UPPER(LOCALIDAD) NOT IN (SELECT UPPER(LOCALIDAD) FROM EMPLEADO GROUP BY
LOCALIDAD)
ORDER BY LOCALIDAD;
```

LOCALIDAD

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

```
SELECT LOCALIDAD
FROM CENTRO
WHERE UPPER(LOCALIDAD) IN (SELECT UPPER(LOCALIDAD) FROM EMPLEADO GROUP BY
LOCALIDAD)
ORDER BY LOCALIDAD;
```

LOCALIDAD
1 Madrid
2 Sevilla
3 Tarragona

27. Esta cuestión puntúa por 2. 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 || ' ' || APE1 || ' ' || APE2 "Nombre completo", Gratificacion
FROM EMPLEADO NATURAL JOIN (SELECT CODEMPLE, NOMBRE || ' ' || APE1 || ' ' || APE2
"Nombre completo", Antigüedad, 0 Gratificacion
FROM EMPLEADO NATURAL JOIN (SELECT CODEMPLE,
TRUNC(MONTHS_BETWEEN(SYSDATE, FECHAINGRESO)/12, 0) Antigüedad
FROM EMPLEADO)
WHERE Antigüedad <1
UNION
SELECT CODEMPLE, NOMBRE || ' ' || APE1 || ' ' || APE2 "Nombre completo",
Antigüedad, 100 Gratificacion
FROM EMPLEADO NATURAL JOIN (SELECT CODEMPLE,
TRUNC(MONTHS_BETWEEN(SYSDATE, FECHAINGRESO)/12, 0) Antigüedad
FROM EMPLEADO)
WHERE Antigüedad BETWEEN 1 AND 5
UNION
SELECT CODEMPLE, NOMBRE || ' ' || APE1 || ' ' || APE2 "Nombre completo",
Antigüedad, Antigüedad*50 Gratificacion
FROM EMPLEADO NATURAL JOIN (SELECT CODEMPLE,
TRUNC(MONTHS_BETWEEN(SYSDATE, FECHAINGRESO)/12, 0) Antigüedad
FROM EMPLEADO)
WHERE Antigüedad BETWEEN 6 AND 10
UNION
SELECT CODEMPLE, NOMBRE || ' ' || APE1 || ' ' || APE2 "Nombre completo",
Antigüedad, Antigüedad*70 Gratificacion
FROM EMPLEADO NATURAL JOIN (SELECT CODEMPLE,
TRUNC(MONTHS_BETWEEN(SYSDATE, FECHAINGRESO)/12, 0) Antigüedad
FROM EMPLEADO)
WHERE Antigüedad BETWEEN 11 AND 20
UNION
SELECT CODEMPLE, NOMBRE || ' ' || APE1 || ' ' || APE2 "Nombre completo",
Antigüedad, Antigüedad*100 Gratificacion
FROM EMPLEADO NATURAL JOIN (SELECT CODEMPLE,
TRUNC(MONTHS_BETWEEN(SYSDATE, FECHAINGRESO)/12, 0) Antigüedad
FROM EMPLEADO)
WHERE Antigüedad >= 21)
ORDER BY NOMBRE;
```

	Nombre completo	GRATIFICACION
1	ALFONSO ROS SANTON	1260
2	ANA LOPEZ GARCIA	2100
3	ANGEL CORTES LOPEZ	1260
4	ANGUSTIAS ZAPATERO GALLARDON	2100
5	BLANCA FLOR LUZ	2100
6	FABIOLA SANCHEZ LUZ	1400
7	GANDI LOPEZ ITURRIALDE	2300
8	JUAN FERNANDEZ MORON	1330
9	MARIA JAZMIN EXPOSITO	1400
10	PAZ RAJOY AZNAR	2100

28. Obtener a los nombres, apellidos de los empleados que no son jefes de departamento.

```
SELECT NOMBRE || ' ' || APE1 || ' ' || APE2 "Nombre completo"
FROM EMPLEADO
WHERE CODEMPLE IN (SELECT CODEMPLE FROM EMPLEADO
MINUS
SELECT CODEMPLEJEFE FROM DPTO);
```

	Nombre completo
1	ANGUSTIAS ZAPATERO GALLARDON
2	ALFONSO ROS SANTON
3	GANDI LOPEZ ITURRIALDE
4	MARIA JAZMIN EXPOSITO