

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

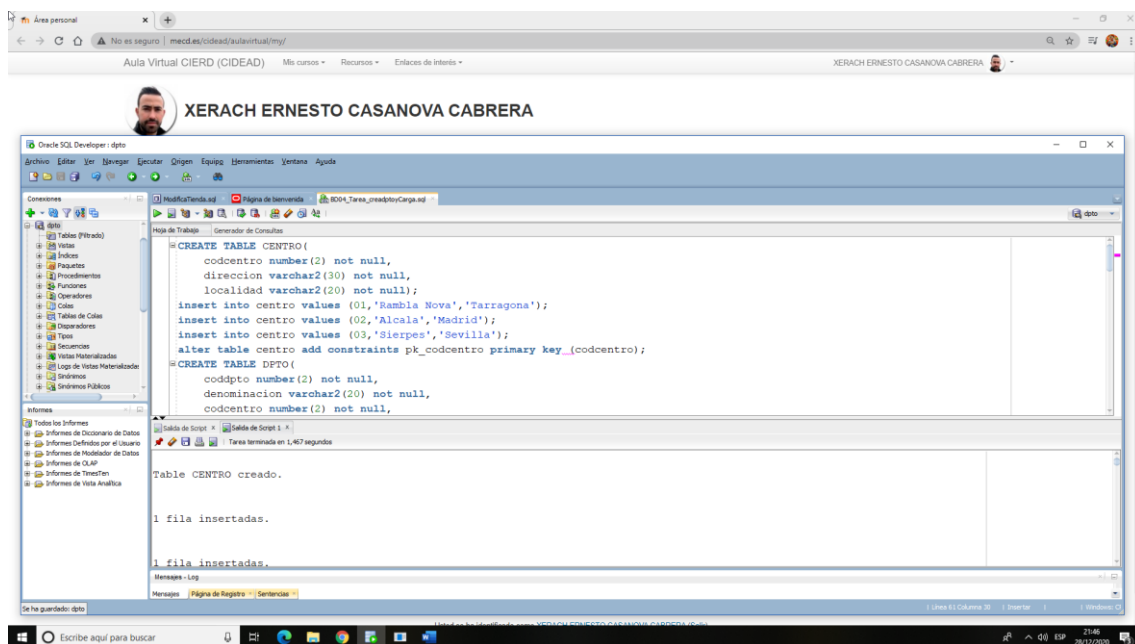
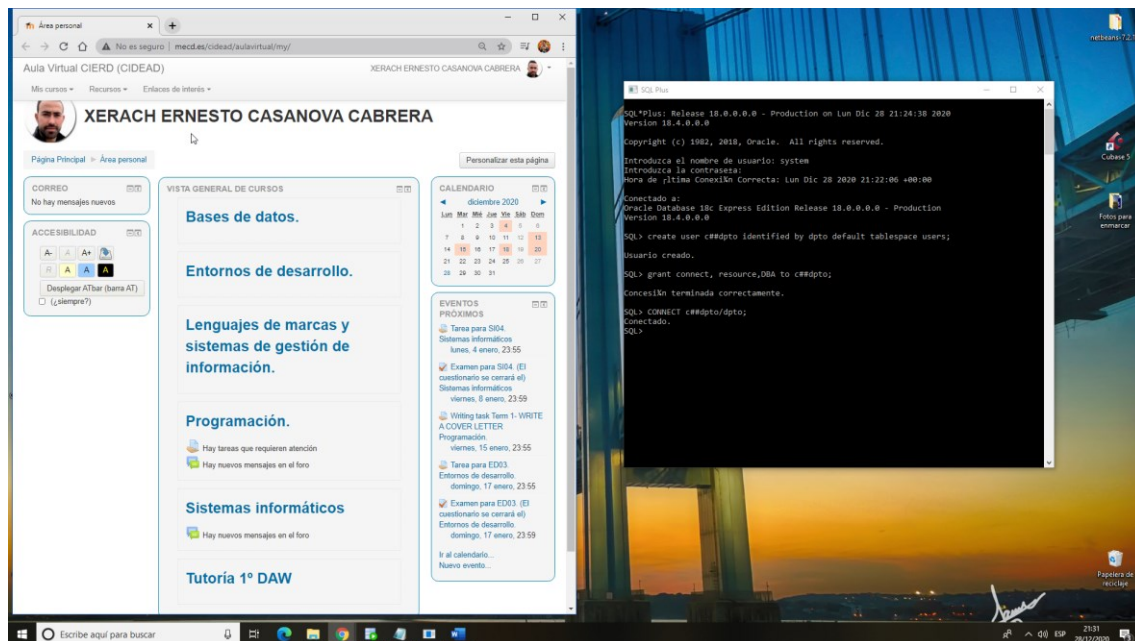
Una empresa está organizada en Departamentos.

- 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, Direccion y localidad.
- De los Departamentos interesa almacenar el código, que es único para cada dpto, y la siguiente información: Denominacion, Centro en el que está ubicado, departamento del que depende, en el caso de dependa e alugno, 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, direccion, localidad, telef, Departamento en el que trabaja, categoría que tiene, comision (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.

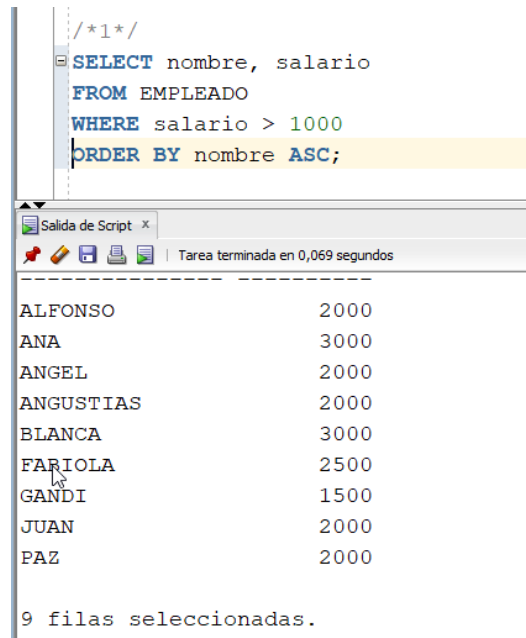


Obtén el diagrama del modelo relacional siguiendo los pasos del enlace [Obtención del modelo entidad relación a partir de las tablas con SQLDeveloper](#) para entender bien el esquema, sus tablas y la relación entre ellas.

Como tarea realiza los siguientes requerimientos en SQL

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 ASC;
```



```
/*1*/
SELECT nombre, salario
FROM EMPLEADO
WHERE salario > 1000
ORDER BY nombre ASC;
```

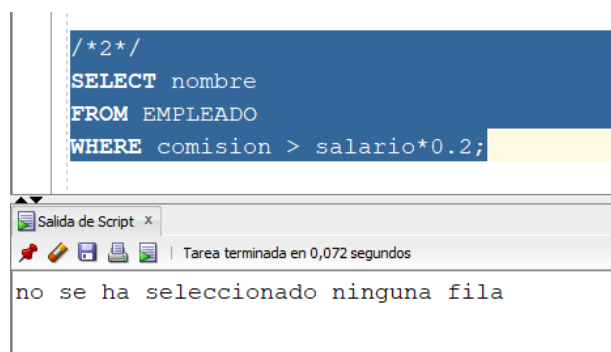
Salida de Script x | Tarea terminada en 0,069 segundos

ALFONSO	2000
ANA	3000
ANGEL	2000
ANGUSTIAS	2000
BLANCA	3000
FARIOLA	2500
GANDI	1500
JUAN	2000
PAZ	2000

9 filas seleccionadas.

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

```
SELECT nombre
FROM EMPLEADO
WHERE comision > salario*0.2;
```



```
/*2*/
SELECT nombre
FROM EMPLEADO
WHERE comision > salario*0.2;
```

Salida de Script x | Tarea terminada en 0,072 segundos

no se ha seleccionado ninguna fila

(Nota: solo un empleado cobra el 20% de su salario en comisiones, pero el ejercicio pide que la comisión sea SUPERIOR al 20% y no IGUAL O SUPERIOR).

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 E.codemple, D.coddpto, nombre, (salario+ NVL(comision, 0))*166.386 || ' ptas.' AS "salario en ptas."
FROM EMPLEADO E JOIN DPTO D ON E.coddpto = D.coddpto
WHERE salario + NVL(comision, 0) > 1800
ORDER BY D.coddpto ASC, nombre ASC;
```

```

/*3*/
SELECT E.codemple, D.coddpto, nombre, (salario+ NVL(comision, 0))*166.386 || ' ptas.' AS "salario en ptas."
FROM EMPLEADO E JOIN DPTO D ON E.coddpto = D.coddpto
WHERE salario + NVL(comision, 0) > 1800
ORDER BY D.coddpto ASC, nombre ASC;

```

Salida de Script x

Tarea terminada en 0,035 segundos

CODEMPLE	CODDPTO	NOMBRE	salario en ptas.
1	1	ANA	499158 ptas.
2	1	JUAN	332772 ptas.
3	2	ANGEL	332772 ptas.
4	3	FABIOLA	415965 ptas.
5	3	PAZ	354402,18 ptas.
6	5	ANGUSTIAS	332772 ptas.
7	6	BLANCA	520788,18 ptas.
8	7	ALFONSO	332772 ptas.

8 filas seleccionadas.

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*1.05
   FROM empleado
   WHERE UPPER(nombre) = 'MARIA' AND UPPER(APE1) = 'JAZMIN')
ORDER BY NOMBRE ASC;

```

```

/*4*/
SELECT nombre
FROM EMPLEADO
WHERE salario >=
  (SELECT salario*1.05
   FROM empleado
   WHERE UPPER(nombre) = 'MARIA' AND UPPER(APE1) = 'JAZMIN')
ORDER BY NOMBRE ASC;

```

Salida de Script x

Tarea terminada en 0,073 segundos

NOMBRE

ALFONSO

ANA

ANGEL

ANGUSTIAS

BLANCA

FABIOLA

GANDI

JUAN

PAZ

9 filas seleccionadas.

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 AS "NOMBRE",
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' as "Antigüedad"
FROM EMPLEADO

```

ORDER BY fechaingreso DESC;

```
/*5*/
SELECT nombre || ' ' || ape1 || ' ' || ape2 AS "NOMBRE", TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' as "Antigüedad"
FROM EMPLEADO
ORDER BY fechaingreso DESC;
/*6*/
```

NOMBRE	Antigüedad
ANGEL CORTES LOPEZ	17 años
ALFONSO ROS SANTON	17 años
JUAN FERNANDEZ MORON	18 años
FABIOLA SANCHEZ LUZ	19 años
MARIA JAZMIN EXPOSITO	19 años
PAZ RAJOY AZNAR	20 años
BLANCA FLOR LUZ	20 años
ANA LOPEZ GARCIA	20 años
ANGUSTIAS ZAPATERO GALLARDON	20 años
GANDI LOPEZ ITURRIALDE	22 años

10 filas seleccionadas.

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);
```

```
/*6*/
SELECT nombre
FROM EMPLEADO
WHERE coddpto = ANY(SELECT coddpto FROM dpto WHERE presupuesto > 50000);
```

NOMBRE
ANA
JUAN
ANGUSTIAS
GANDI
MARIA

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

```
SELECT nombre || ' ' || ape1 || ' ' || ape2 AS NOMBRE
FROM EMPLEADO
WHERE salario + comision = (SELECT MAX(salario+NVL(comision,0)) FROM empleado);
```

```
/*7*/
SELECT nombre || ' ' || ape1 || ' ' || ape2 AS NOMBRE
FROM EMPLEADO
WHERE salario + NVL(comision,0) = (SELECT MAX(salario+NVL(comision,0)) FROM empleado);
```

NOMBRE
BLANCA FLOR LUZ

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;

```

```

/*8*/
SELECT nombre
FROM EMPLEADO
WHERE salario <
      (SELECT MIN(salario)
       FROM EMPLEADO
       WHERE coddpto = 1)
ORDER BY nombre;

```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,052 segundos

NOMBRE
GANDI
MARIA

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

```

SELECT nombre
FROM EMPLEADO E JOIN DPTO D ON E.coddpto = D.coddpto
WHERE codemplejefe = 1;

```

```

/*9*/
SELECT nombre
FROM EMPLEADO E JOIN DPTO D ON E.coddpto = D.coddpto
WHERE codemplejefe = 1;

```

Salida de Script x

Tarea terminada en 0,07 segundos

NOMBRE
ANA
JUAN

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

```

SELECT nombre || ' ' || ape1 || ' ' || ape2 AS "NOMBRE"
FROM EMPLEADO
WHERE SUBSTR(UPPER(ape1),0,1) IN ('P', 'Q', 'R', 'S');

```

```

/*10*/
SELECT nombre || ' ' || ape1 || ' ' || ape2 AS "NOMBRE"
FROM EMPLEADO
WHERE SUBSTR(UPPER(ape1),0,1) IN ('P', 'Q', 'R', 'S');

```

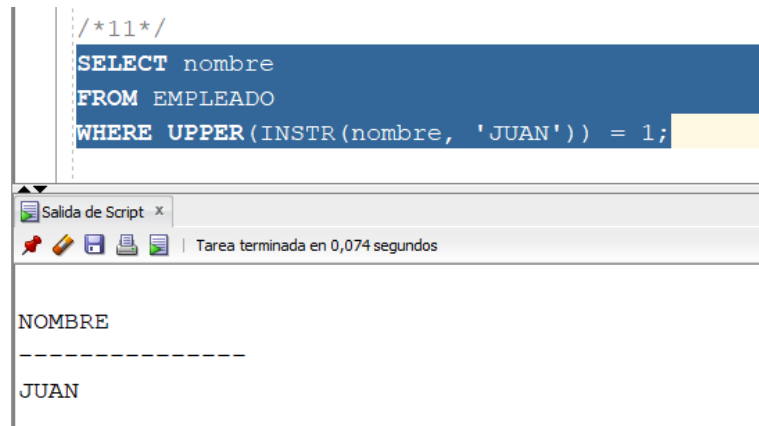
Salida de Script x

Tarea terminada en 0,07 segundos

NOMBRE
FABIOLA SANCHEZ LUZ
PAZ RAJOY AZNAR
ALFONSO ROS SANTON

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

```
SELECT nombre
FROM EMPLEADO
WHERE UPPER(INSTR(nombre, 'JUAN')) = 1;
```



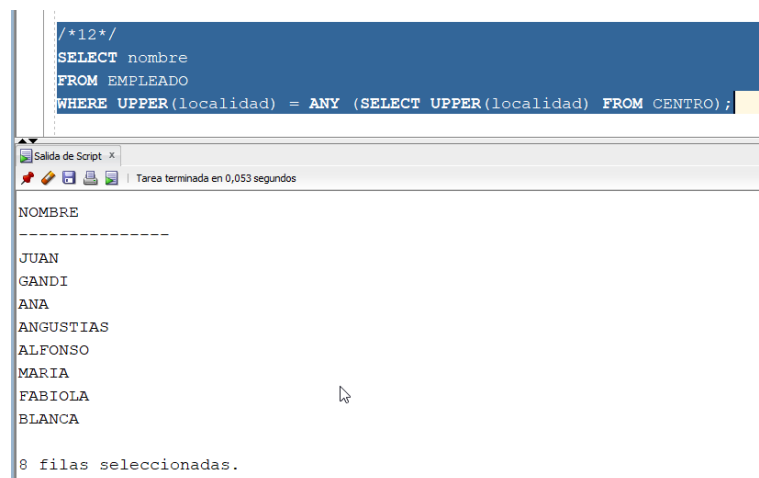
```
/*11*/
SELECT nombre
FROM EMPLEADO
WHERE UPPER(INSTR(nombre, 'JUAN')) = 1;
```

Salida de Script x
Tarea terminada en 0,074 segundos

NOMBRE
JUAN

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);
```



```
/*12*/
SELECT nombre
FROM EMPLEADO
WHERE UPPER(localidad) = ANY (SELECT UPPER(localidad) FROM CENTRO);
```

Salida de Script x
Tarea terminada en 0,053 segundos

NOMBRE
JUAN
GANDI
ANA
ANGUSTIAS
ALFONSO
MARIA
FABIOLA
BLANCA

8 filas seleccionadas.

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

```
SELECT nombre
FROM EMPLEADO
WHERE codemple IN (SELECT codemplejefe FROM DPTO) AND salario = (SELECT MAX(salario) FROM EMPLEADO);
```



```

/*13*/
SELECT nombre
FROM EMPLEADO
WHERE codemple IN (SELECT codemplejefe FROM DPTO) AND salario = (SELECT MAX(salario) FROM EMPLEADO);

/*14*/

```

Salida de Script x
Tarea terminada en 0,073 segundos

NOMBRE

ANA
BLANCA

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 AS "NOMBRE", salario
FROM EMPLEADO
WHERE salario > (SELECT MAX(salario)*0.6 FROM EMPLEADO)
ORDER BY nombre ASC, ape1 ASC, ape2 ASC;

```

```

/*14*/
SELECT nombre || ' ' || ape1 || ' ' || ape2 AS "NOMBRE", salario
FROM EMPLEADO
WHERE salario > (SELECT MAX(salario)*0.6 FROM EMPLEADO)
ORDER BY nombre ASC, ape1 ASC, ape2 ASC;

```

Salida de Script x
Tarea terminada en 0,086 segundos

NOMBRE	SALARIO
ALFONSO ROS SANTON	2000
ANA LOPEZ GARCIA	3000
ANGEL CORTES LOPEZ	2000
ANGUSTIAS ZAPATERO GALLARDON	2000
BLANCA FLOR LUZ	3000
FABIOLA SANCHEZ LUZ	2500
JUAN FERNANDEZ MORON	2000
PAZ RAJOY AZNAR	2000

8 filas seleccionadas.

Obtener en cuántas ciudades distintas viven los empleados

```

SELECT COUNT(DISTINCT localidad) AS "LOCALIDADES DE EMPLEADOS"
FROM EMPLEADO;

```

```

/*15*/
SELECT COUNT(DISTINCT localidad) AS "LOCALIDADES DE EMPLEADOS"
FROM EMPLEADO;

```

Salida de Script x
Tarea terminada en 0,075 segundos

LOCALIDADES DE EMPLEADOS

5

El nombre y apellidos del empleado que más salario cobra

```

SELECT nombre || ' ' || ape1 || ' ' || ape2 AS "NOMBRE"
FROM EMPLEADO
WHERE salario = (SELECT MAX(salario) FROM EMPLEADO);

```

```

/*16*/
SELECT nombre || ' ' || ape1 || ' ' || ape2 AS "NOMBRE"
FROM EMPLEADO
WHERE salario = (SELECT MAX(salario) FROM EMPLEADO);

```

Salida de Script x

Tarea terminada en 0,052 segundos

```

NOMBRE
-----
ANA LOPEZ GARCIA
BLANCA FLOR LUZ

```

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

```

SELECT localidad, COUNT(codemple) as "Nº DE EMPLEADOS"
FROM EMPLEADO
GROUP BY localidad
HAVING COUNT(codemple) > 3;

```

```

/*17*/
SELECT localidad, COUNT(codemple) as "Nº DE EMPLEADOS"
FROM EMPLEADO
GROUP BY localidad
HAVING COUNT(codemple) > 3;

```

Salida de Script x

Tarea terminada en 0,07 segundos

LOCALIDAD	Nº DE EMPLEADOS
MADRID	4

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 "Nº DE EMPLEADOS", SUM(salario) AS "SUMA DE SALARIOS", SUM(ALL comision) AS "SUMA DE COMISIONES"
FROM DPTO D JOIN EMPLEADO E ON D.coddpto = E.coddpto
WHERE E.coddpto IN (SELECT coddpto FROM EMPLEADO WHERE salario > 1700)
GROUP BY denominacion;

```

```

/*18*/
SELECT denominacion, COUNT(codemple) AS "Nº DE EMPLEADOS", SUM(salario) AS "SUMA DE SALARIOS", SUM(ALL comision) AS "SUMA DE COMISIONES"
FROM DPTO D JOIN EMPLEADO E ON D.coddpto = E.coddpto
WHERE E.coddpto IN (SELECT coddpto FROM EMPLEADO WHERE salario > 1700)
GROUP BY denominacion;

```

Salida de Script x

Resultado de la Consulta x

Tarea terminada en 0,047 segundos

*Action:

DENOMINACION	Nº DE EMPLEADOS	SUMA DE SALARIOS	SUMA DE COMISIONES
DIRECCIÓN	2	5000	
RECURSOS HUMANOS	2	4500	130
CENTRAL COMERCIAL	3	4500	410
COMERCIAL CENTRO	1	3000	130
COMERCIAL SUR	1	2000	
ADMINISTRACION	1	2000	

6 filas seleccionadas.

Obtener el departamento que más empleados tiene

```
SELECT denominacion
FROM DPTO D JOIN EMPLEADO E ON D.coddpto = E.coddpto
GROUP BY denominacion
HAVING COUNT(D.coddpto) = (SELECT MAX(COUNT(coddpto)) FROM EMPLEADO GROUP BY coddpto)
```

```
/*19*/
SELECT denominacion
FROM DPTO D JOIN EMPLEADO E ON D.coddpto = E.coddpto
GROUP BY denominacion
HAVING COUNT(D.coddpto) = (SELECT MAX(COUNT(coddpto)) FROM EMPLEADO GROUP BY coddpto);
```

Salida de Script x Resultado de la Consulta x
Tarea terminada en 0,064 segundos

DENOMINACION
CENTRAL COMERCIAL

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

```
/*20*/
SELECT C.codcentro, D.denominacion
FROM CENTRO C RIGHT OUTER JOIN DPTO D ON C.codcentro = D.codcentro;
```

Salida de Script x Resultado de la Consulta x
Tarea terminada en 0,077 segundos

CODCENTRO	DENOMINACION
1	DIRECCIÓN
1	ADMINISTRACION
1	RECURSOS HUMANOS
1	CENTRAL COMERCIAL
2	COMERCIAL CENTRO
3	COMERCIAL SUR

6 filas seleccionadas.

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;
```

```
/*21*/
SELECT denominacion
FROM DPTO
WHERE coddptodepende IS NULL;
```

Salida de Script x Resultado de la Consulta x
Tarea terminada en 0,164 segundos

DENOMINACION
DIRECCIÓN

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

```
SELECT nombre, denominacion
FROM DPTO D LEFT OUTER JOIN EMPLEADO E ON D.coddpto = E.coddpto
```

```
/*22*/
SELECT nombre, denominacion
FROM DPTO D LEFT OUTER JOIN EMPLEADO E ON D.coddpto = E.coddpto;
```

Salida de Script x

Tarea terminada en 0,061 segundos

NOMBRE	DENOMINACION
ANA	DIRECCIÓN
JUAN	DIRECCIÓN
ANGEL	ADMINISTRACION
FABIOLA	RECURSOS HUMANOS
PAZ	RECURSOS HUMANOS
ANGUSTIAS	CENTRAL COMERCIAL
GANDI	CENTRAL COMERCIAL
MARIA	CENTRAL COMERCIAL
BLANCA	COMERCIAL CENTRO
ALFONSO	COMERCIAL SUR

10 filas seleccionadas.

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

```
SELECT DDEPENDE.Denominacion AS "DEPARTAMENTO", D.Denominacion AS "DPTO SUPERIOR"
FROM DPTO DDEPENDE LEFT OUTER JOIN DPTO D ON DDEPENDE.coddptodepende = D.coddpto;
```

```
/*23*/
SELECT DDEPENDE.Denominacion AS "DEPARTAMENTO", D.Denominacion AS "DPTO SUPERIOR"
FROM DPTO DDEPENDE LEFT OUTER JOIN DPTO D ON DDEPENDE.coddptodepende = D.coddpto;
```

Salida de Script x

Tarea terminada en 0,044 segundos

6 filas seleccionadas.

DEPARTAMENTO	DPTO SUPERIOR
ADMINISTRACION	DIRECCIÓN
RECURSOS HUMANOS	DIRECCIÓN
CENTRAL COMERCIAL	DIRECCIÓN
COMERCIAL CENTRO	CENTRAL COMERCIAL
COMERCIAL SUR	CENTRAL COMERCIAL
DIRECCIÓN	

6 filas seleccionadas.

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, DECODE(comision, NULL, 'No tiene comisión', 'Tiene comisión') AS "COMISIÓN"
FROM EMPLEADO
ORDER BY NOMBRE;
```

```

/*24*/
SELECT nombre, DECODE(comision, NULL, 'No tiene comisión', 'Tiene comisión') AS "COMISIÓN"
FROM EMPLEADO
ORDER BY NOMBRE;

```

Salida de Script x | Tarea terminada en 0,058 segundos

NOMBRE	COMISIÓN
ALFONSO	No tiene comisión
ANA	No tiene comisión
ANGEL	No tiene comisión
ANGUSTIAS	No tiene comisión
BLANCA	Tiene comisión
FABIOLA	No tiene comisión
GANDI	Tiene comisión
JUAN	No tiene comisión
MARIA	Tiene comisión
PAZ	Tiene comisión

10 filas seleccionadas.

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)
ORDER BY localidad;

```

```

/*25*/
SELECT localidad
FROM CENTRO
WHERE UPPER(localidad) NOT IN (SELECT UPPER(localidad) FROM empleado)
ORDER BY localidad;

```

Salida de Script x | Tarea terminada en 0,056 segundos

no se ha seleccionado ninguna fila

Nota: todos los centros están ubicados en localidades donde viven empleados, por eso no devuelve ninguna fila.

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) = ANY (SELECT UPPER(localidad) FROM empleado)
ORDER BY localidad;

```

```

/*26*/
SELECT localidad
FROM CENTRO
WHERE UPPER(localidad) = ANY (SELECT UPPER(localidad) FROM empleado)
ORDER BY localidad;

```

Salida de Script x | Tarea terminada en 0,058 segundos

LOCALIDAD
Madrid
Sevilla
Tarragona

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,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
(CASE
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 1 AND 5
THEN '100 Euros'
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 6 AND 10
THEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 50 || ' Euros '
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 11 AND 20
THEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 70 || ' Euros '
ELSE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 100 || ' Euros '
END) AS "GRATIFICACIÓN"
FROM EMPLEADO
ORDER BY nombre;
```

```
/*27/28 - 1*/
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
(CASE
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 1 AND 5
THEN '100 Euros'
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 6 AND 10
THEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 50 || ' Euros '
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 11 AND 20
THEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 70 || ' Euros '
ELSE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 100 || ' Euros '
END) AS "GRATIFICACIÓN"
FROM EMPLEADO
ORDER BY nombre;
```

Salida de Script: x

Tarea terminada en 0,059 segundos

ANA	20 años	1400 Euros
ANGEL	17 años	1190 Euros
ANGUSTIAS	20 años	1400 Euros
BLANCA	20 años	1400 Euros
FABIOLA	19 años	1330 Euros
GANDI	22 años	2200 Euros
JUAN	18 años	1260 Euros
MARIA	19 años	1330 Euros
PAZ	20 años	1400 Euros

10 filas seleccionadas.

Entendiendo que esta manera de implementación no está en los apuntes, también he optado por hacerlo con UNION. El resultado es el mismo:

```

/*27*/
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
(CASE
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 1 AND 5
THEN '100 Euros'
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 6 AND 10
THEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 50 || ' Euros '
WHEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 11 AND 20
THEN TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 70 || ' Euros '
ELSE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 100 || ' Euros '
END) AS "GRATIFICACIÓN"
FROM EMPLEADO;

```

Salida de Script x

Tarea terminada en 0,06 segundos

NOMBRE	ANTIGÜEDAD	GRATIFICACIÓN
ANA	20 años	1400 Euros
JUAN	18 años	1260 Euros
ANGEL	17 años	1190 Euros
FABIOLA	19 años	1330 Euros
PAZ	20 años	1400 Euros
ANGUSTIAS	20 años	1400 Euros
BLANCA	20 años	1400 Euros
ALFONSO	17 años	1190 Euros
GANDI	22 años	2200 Euros
MARIA	19 años	1330 Euros

```

SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
'100 Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 1 AND 5
UNION
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 50 || ' Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 6 AND 10
UNION
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 70 || ' Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 11 AND 20
UNION
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 100 || ' Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) > 20
ORDER BY nombre;

```

```
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
'100 Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 1 AND 5
UNION
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 50 || ' Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 6 AND 10
UNION
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 70 || ' Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) BETWEEN 11 AND 20
UNION
SELECT nombre,
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) || ' años' AS "ANTIGÜEDAD",
TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) * 100 || ' Euros' AS "GRATIFICACION"
FROM EMPLEADO
WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, fechaingreso)/12, 0) > 20
ORDER BY nombre;
```

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

```
SELECT nombre || ' ' || ape1 || ' ' || ape2 as "NOMBRE"
FROM EMPLEADO E JOIN DPTO D ON E.coddpto = D.coddpto
WHERE E.codemple NOT IN (SELECT codemplejefe FROM DPTO);
```

```
SELECT nombre || ' ' || ape1 || ' ' || ape2 as "NOMBRE"
FROM EMPLEADO E JOIN DPTO D ON E.coddpto = D.coddpto
WHERE E.codemple NOT IN (SELECT codemplejefe FROM DPTO);
```

Salida de Script x

Tarea terminada en 0,058 segundos

NOMBRE

ANGUSTIAS ZAPATERO GALLARDON

ALFONSO ROS SANTON

GANDI LOPEZ ITURRIALDE

MARIA JAZMIN EXPOSITO