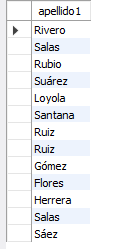
Consultas sobre la tabla empleado y la tabla departamento

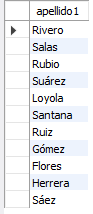
1. Lista el primer apellido de todos los empleados.

* select apellido1 from empleado;



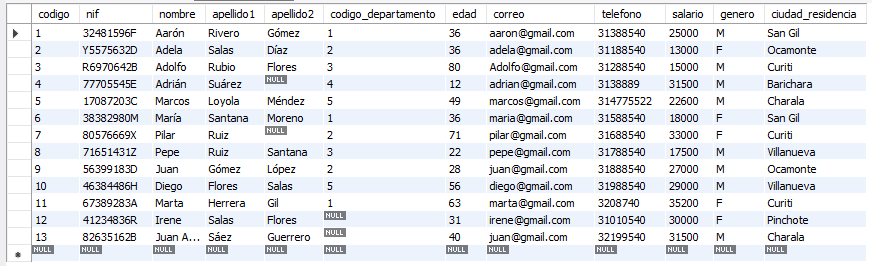
1. Lista el primer apellido de los empleados eliminando los apellidos que estén repetidos.

* SELECT DISTINCT apellido1 FROM empleado;



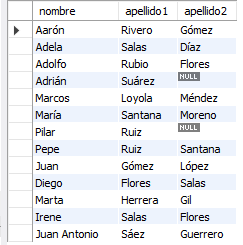
1. Lista todas las columnas de la tabla empleado.

* select \* from empleado;



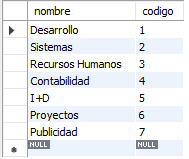
1. Lista el nombre y los apellidos de todos los empleados.

* select nombre,apellido1,apellido2 from empleado;



1. Lista el nombre y el código de los departamentos de los empleados que aparecen en la tabla empleado.

* select nombre,codigo from departamento;



1. Lista el nombre y el código de los departamentos de los empleados que aparecen en la tabla empleado, pero no mostrar los valores nulos en el campo código departamento

* Forma 1:
* select nombre,codigo from empleado where codigo\_departamento<>0;
* Forma 2:
* SELECT nombre,código FROM empleado WHERE codigo\_departamento IS NOT NULL;



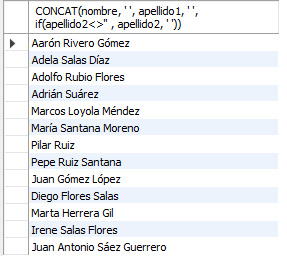
1. Lista el código de los departamentos de los empleados que aparecen en la tabla empleado, eliminando los códigos que aparecen repetidos.

* SELECT DISTINCT codigo\_departamento FROM empleado



1. Lista el nombre y apellidos de los empleados en una única columna.

* SELECT CONCAT(nombre, ' ', apellido1, ' ', if(apellido2<>'' , apellido2, ' ')) FROM empleado;



1. Lista el nombre y apellidos de los empleados en una única columna con el alias “nombre completo”

* SELECT CONCAT(nombre, ' ', apellido1, ' ', if(apellido2<>'' , apellido2, ' ')) AS nombre\_completo FROM empleado;



1. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula.

* Select UCASE(CONCAT(nombre, ' ', apellido1,'' , if(apellido2<>'' , apellido2, ' ')))nombre\_completo FROM empleado ORDER BY nombre\_completo;



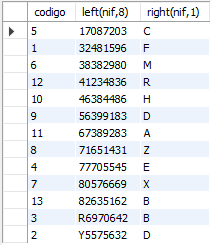
1. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en minúscula.

* Select lower(CONCAT(nombre, ' ', apellido1,'' , if(apellido2<>'' , apellido2, ' ')))nombre\_completo FROM empleado ORDER BY nombre\_completo;



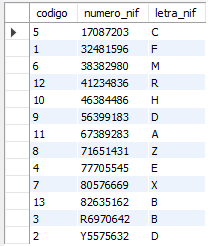
1. Lista el código de los empleados junto al nif, pero el nif deberá aparecer en dos columnas, una mostrará únicamente los dígitos del nif y la otra la letra.

* SELECT codigo,left(nif,8), right(nif,1) from empleado;



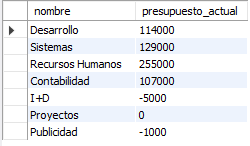
1. Lista el código de los empleados junto al nif, pero el nif deberá aparecer en dos columnas, una columna llamada “numero\_nif” mostrará únicamente los dígitos del nif y la otra columna se llamará “letra nif” donde solo muestre la letra del nif.

* SELECT codigo,left(nif,8) as numero\_nif, right(nif,1) as letra\_nif from empleado;



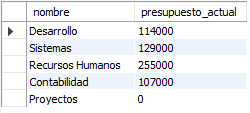
1. Lista el nombre de cada departamento y el valor del presupuesto actual del que dispone. Para calcular este dato tendrá que restar al valor del presupuesto inicial (columna presupuesto) los gastos que se han generado (columna gastos). Tenga en cuenta que en algunos casos pueden existir valores negativos. Utilice un alias apropiado para la nueva columna que está calculando.

* SELECT nombre, (presupuesto-gastos) as presupuesto\_actual from departamento;



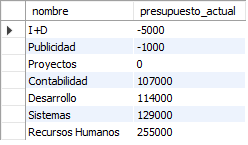
1. Lista el nombre de cada departamento y el valor del presupuesto actual del que dispone, **solo mostrar los presupuestos actuales positivos** Para calcular este dato tendrá que restar al valor del presupuesto inicial (columna presupuesto) los gastos que se han generado (columna gastos). Tenga en cuenta que en algunos casos pueden existir valores negativos. Utilice un alias apropiado para la nueva columna columna que está calculando.

* SELECT nombre,(presupuesto-gastos) as presupuesto\_actual from departamento WHERE (presupuesto-gastos) IS NOT NULL and (presupuesto-gastos)>=0;



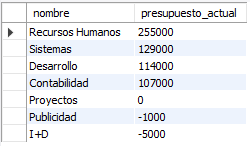
1. Lista el nombre de los departamentos y el valor del presupuesto actual ordenado de forma ascendente.

* SELECT nombre, (presupuesto-gastos) as presupuesto\_actual from departamento ORDER BY presupuesto\_actual ASC;



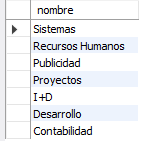
1. Lista el nombre de todos los departamentos ordenados de forma ascendente.

* SELECT nombre, (presupuesto-gastos) as presupuesto\_actual from departamento ORDER BY presupuesto\_actual desc;



1. Lista el nombre de todos los departamentos ordenados de forma descendente.

* select nombre from departamento ORDER BY nombre desc;



1. Lista los apellidos y el nombre de todos los empleados, ordenados de forma alfabética teniendo en cuenta en primer lugar sus apellidos y luego su nombre.

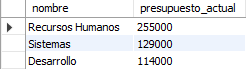
* Select CONCAT(apellido1,' ' , if(apellido2<>'' , apellido2, ' '), ' ',nombre)nombre\_completo FROM empleado ORDER BY

nombre\_completo;



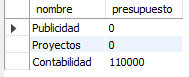
1. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen mayor presupuesto.

* SELECT nombre, presupuesto from departamento ORDER BY presupuesto asc limit 3;



1. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen menor presupuesto.

* SELECT nombre, presupuesto from departamento ORDER BY presupuesto asc limit 3;



1. Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen mayor gasto.

* SELECT nombre, gastos from departamento ORDER BY gastos desc limit 2;



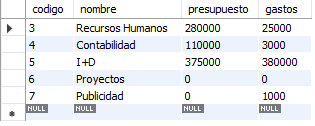
1. Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen menor gasto.

* SELECT nombre, gastos from departamento ORDER BY gastos asc limit 2;



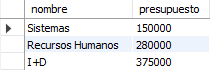
1. Devuelve una lista con 5 filas a partir de la tercera fila de la tabla empleado. La tercera fila se debe incluir en la respuesta. La respuesta debe incluir todas las columnas de la tabla empleado.

* SELECT \* from departamento limit 2,5;



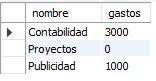
1. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto mayor o igual a 150000 euros.

* SELECT nombre, presupuesto from departamento where presupuesto>=150000;



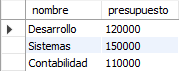
1. Devuelve una lista con el nombre de los departamentos y el gasto, de aquellos que tienen menos de 5000 euros de gastos.

* SELECT nombre, gastos from departamento where gastos<5000;



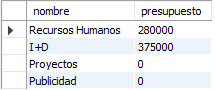
1. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto entre 100000 y 200000 euros. Sin utilizar el operador BETWEEN.

* SELECT nombre, presupuesto from departamento where presupuesto<200000 and presupuesto>100000;



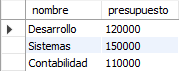
1. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre 100000 y 200000 euros. Sin utilizar el operador BETWEEN

* SELECT nombre, presupuesto from departamento where presupuesto<=100000 or presupuesto>=200000;



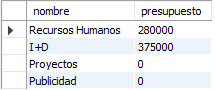
1. Devuelve una lista con el nombre de los departamentos que tienen un presupuesto entre 100000 y 200000 euros. Utilizando el operador BETWEEN.

* SELECT nombre, presupuesto from departamento where presupuesto between 100000 and 200000;



1. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre 100000 y 200000 euros. Utilizando el operador BETWEEN.

* SELECT nombre, presupuesto from departamento where presupuesto not between 100000 and 200000;



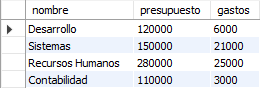
1. Devuelve una lista con el nombre de los departamentos, gastos y presupuesto, de aquellos departamentos donde los gastos sean mayores que el presupuesto del que disponen.

* SELECT nombre, presupuesto, gastos from departamento where gastos>presupuesto;



1. Devuelve una lista con el nombre de los departamentos, gastos y presupuesto, de aquellos departamentos donde los gastos sean menores que el presupuesto del que disponen.

* SELECT nombre, presupuesto, gastos from departamento where gastos<presupuesto;



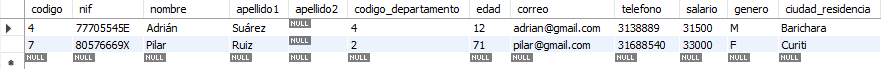
1. Devuelve una lista con el nombre de los departamentos, gastos y presupuesto, de aquellos departamentos donde los gastos sean iguales al presupuesto del que disponen.

* SELECT nombre, presupuesto, gastos from departamento where gastos=presupuesto;



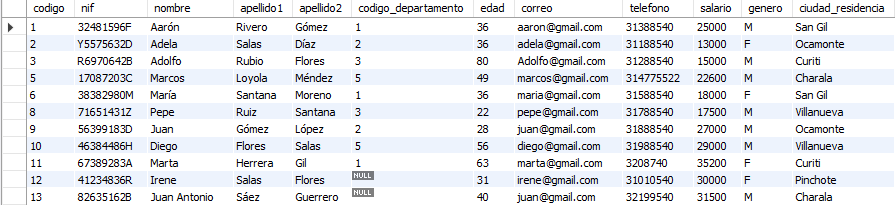
1. Lista todos los datos de los empleados cuyo segundo apellido sea NULL.

* SELECT \* from empleado where apellido2 is null;



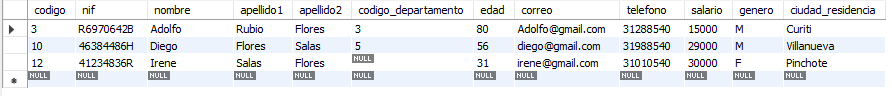
1. Lista todos los datos de los empleados cuyo segundo apellido no sea NULL.

* SELECT \* from empleado where apellido2 is not null;



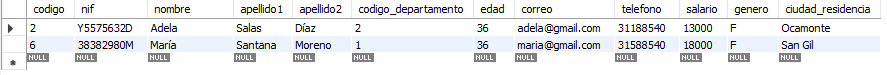
1. Lista todos los datos de los empleados cuyo segundo apellido sea López.

* SELECT \* from empleado where apellido1 = "flores" or apellido2 = "flores";



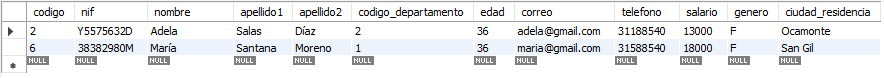
1. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Sin utilizar el operador IN.

* SELECT \* from empleado where apellido2 = "diaz" or apellido2 ="moreno";



1. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Utilizando el operador IN.

* SELECT \* FROM empleado WHERE apellido2 IN ("diaz", "moreno");



1. Lista los nombres, apellidos y nif de los empleados que trabajan en el departamento 3.

* select nombre, CONCAT(apellido1,' ' , if(apellido2<>'' , apellido2, ' '))apellidos,nif from empleado where codigo\_departamento = 3;



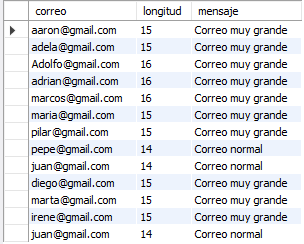
1. Lista los nombres, apellidos y nif de los empleados que trabajan en los departamentos 2, 4 o 5.

* select nombre, CONCAT(apellido1,' ' , if(apellido2<>'' , apellido2, ' '))apellidos,nif from empleado where codigo\_departamento = 3 or codigo\_departamento = 4 or codigo\_departamento = 5;



1. Seleccionar tres campos campo 1 el correo de la persona, campo 2 número de caracteres del campo correo, campo 3 si el numero de caracteres excede de 14 enviar un mensaje en la tercera columna que diga “Correo muy grande” y sino “Correo normal” de la tabla empleado.

* select correo, length(correo), if(length(correo)>14, 'Correo muy grande' , 'Correo normal') as mensaje from empleado;



1. Contar cuantas personas tienen el apellido 2 en nulo

* select count(apellido2 is null) from empleado;



1. Cuántas mujeres hay residentes en Curiti mayores a 40 años

* select count(\*) from empleado where genero = "F" AND ciudad\_residencia = "Curiti" AND edad>40;



1. Cual son los nombres de las mujeres residentes en Curiti y mayores a 40 años

* select nombre from empleado where genero = "F" AND ciudad\_residencia = "Curiti" AND edad>40;



1. Cuál es el sueldo máximo que hay entre las mujeres

* select max(salario) from empleado where genero = "F" ;



1. Cuantas personas hay que termine su apellido 2 en la letra “z”

* select count(\*) from empleado where apellido2 like "%z";



Consultas multi-tabla

Consultas multitabla (Composición interna)

1. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno.

* Select CONCAT(empleado.nombre, ' ', empleado.apellido1) as nombre\_empleado,

departamento.nombre,

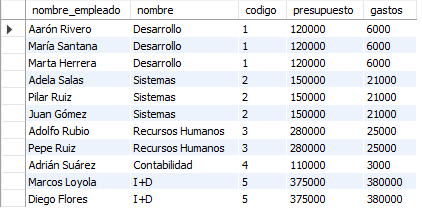
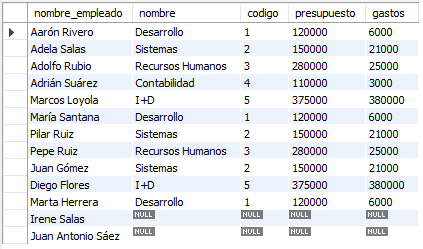
departamento.codigo,

departamento.presupuesto,

departamento.gastos

FROM empleado

inner JOIN Departamento ON empleado.codigo\_departamento = departamento.codigo;



1. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno. Ordena el resultado, en primer lugar por el nombre del departamento (en orden alfabético) y en segundo lugar por los apellidos y el nombre de los empleados.

* Select CONCAT(empleado.nombre, ' ', empleado.apellido1) as nombre\_empleado,

departamento.nombre,

departamento.codigo,

departamento.presupuesto,

departamento.gastos

FROM empleado

left JOIN Departamento ON empleado.codigo\_departamento = departamento.codigo

ORDER BY departamento.nombre ASC , nombre\_empleado asc;

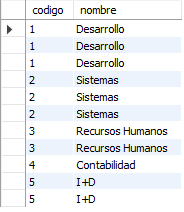


1. Devuelve un listado con el código y el nombre del departamento, solamente de aquellos departamentos que tienen empleados.

* select departamento.codigo, departamento.nombre

from empleado

inner JOIN Departamento ON empleado.codigo\_departamento = departamento.codigo



1. Devuelve un listado con el código, el nombre del departamento y el valor del presupuesto actual del que dispone, solamente de aquellos departamentos que tienen empleados. El valor del presupuesto actual lo puede calcular restando al valor del presupuesto inicial (columna presupuesto) el valor de los gastos que ha generado (columna gastos).

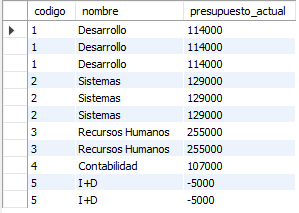
* select departamento.codigo,

departamento.nombre,

(presupuesto-gastos) as presupuesto\_actual

from empleado

inner join Departamento ON empleado.codigo\_departamento = departamento.codigo;



1. Devuelve el nombre del departamento donde trabaja el empleado que tiene el nif 38382980M.

* select departamento.nombre

from empleado

inner join Departamento ON empleado.codigo\_departamento = departamento.codigo

where empleado.nif="38382980M";



1. Devuelve el nombre del departamento donde trabaja el empleado Pepe Ruiz Santana.

* select departamento.nombre

from empleado

inner join Departamento ON empleado.codigo\_departamento = departamento.codigo

where empleado.nombre="Pepe" and empleado.apellido1="Ruiz" and empleado.apellido2="Santana";



1. Devuelve un listado con los datos de los empleados que trabajan en el departamento de I+D. Ordena el resultado alfabéticamente.

* select \*

from empleado

inner join Departamento ON empleado.codigo\_departamento = departamento.codigo

where Departamento.nombre="I+D" order by empleado.nombre asc;



1. Devuelve un listado con los datos de los empleados que trabajan en el departamento de Sistemas, Contabilidad o I+D. Ordena el resultado alfabéticamente.

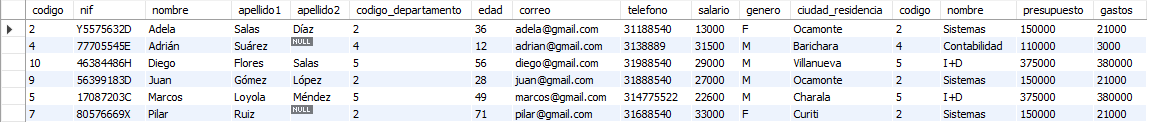
* select \*

from empleado

inner join Departamento ON empleado.codigo\_departamento = departamento.codigo

where Departamento.nombre in ("Sistemas", "Contabilidad", "I+D")

order by empleado.nombre asc;



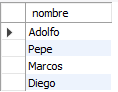
1. Devuelve una lista con el nombre de los empleados que tienen los departamentos que no tienen un presupuesto entre 100000 y 200000 euros.

* select empleado.nombre

from empleado

inner join Departamento ON empleado.codigo\_departamento = departamento.codigo

where Departamento.presupuesto not between 100000 and 200000;



1. Devuelve un listado con el nombre de los departamentos donde existe algún empleado cuyo segundo apellido sea NULL. Tenga en cuenta que no debe mostrar nombres de departamentos que estén repetidos.

* select departamento.nombre

from empleado

inner join Departamento ON empleado.codigo\_departamento = departamento.codigo

where empleado.apellido2 is null;



1.2.5 Consultas multitabla (Composición externa)

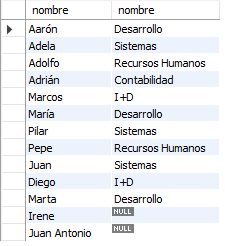
Resuelva todas las consultas utilizando las cláusulas LEFT JOIN y RIGHT JOIN.

1. Devuelve un listado con todos los empleados junto con los datos de los departamentos donde trabajan. Este listado también debe incluir los empleados que no tienen ningún departamento asociado.

* select empleado.nombre, departamento.nombre

from empleado

left join Departamento ON empleado.codigo\_departamento = departamento.codigo;



1. Devuelve un listado donde sólo aparezcan aquellos empleados que no tienen ningún departamento asociado.

* select empleado.nombre, departamento.nombre

from empleado

left join Departamento ON empleado.codigo\_departamento = departamento.codigo

where departamento.nombre is null;



1. Devuelve un listado donde sólo aparezcan aquellos departamentos que no tienen ningún empleado asociado.

* select departamento.nombre

from departamento

left join empleado ON empleado.codigo\_departamento = departamento.codigo

where empleado.nombre is null;



1. Devuelve un listado con todos los empleados junto con los datos de los departamentos donde trabajan. El listado debe incluir los empleados que no tienen ningún departamento asociado y los departamentos que no tienen ningún empleado asociado. Ordene el listado alfabéticamente por el nombre del departamento.

* SELECT CONCAT(empleado.apellido1,' ' , if(empleado.apellido2<>'' , empleado.apellido2, ' '), ' ',empleado.nombre)nombre\_completo, departamento.nombre as nombre\_departamento

FROM departamento

left JOIN empleado

ON empleado.codigo\_departamento = departamento.codigo

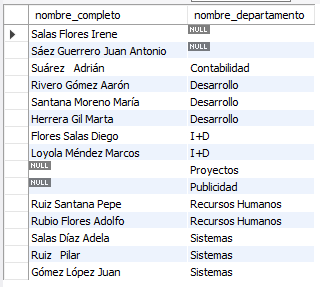
union

SELECT CONCAT(empleado.apellido1,' ' , if(empleado.apellido2<>'' , empleado.apellido2, ' '), ' ',empleado.nombre)nombre\_completo, departamento.nombre as nombre\_departamento

FROM empleado

left JOIN departamento

ON empleado.codigo\_departamento = departamento.codigo order by nombre\_departamento;



1. Devuelve un listado con los empleados que no tienen ningún departamento asociado y los departamentos que no tienen ningún empleado asociado. Ordene el listado alfabéticamente por el nombre del departamento.

* SELECT empleado.nombre, departamento.nombre as nombre\_departamento

FROM departamento

left JOIN empleado

ON empleado.codigo\_departamento = departamento.codigo

where empleado.nombre is null

union

SELECT CONCAT(empleado.apellido1,' ' , if(empleado.apellido2<>'' , empleado.apellido2, ' '), ' ',empleado.nombre) as nombre\_completo, departamento.nombre as nombre\_departamento

FROM empleado

left JOIN departamento

ON empleado.codigo\_departamento = departamento.codigo

where departamento.nombre is null order by nombre\_departamento;;

