

# CONSULTAS EMPLEADOS CON JOIN

Iñaki Borrego Bau

1D.A.W

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

```
SELECT empleado.nombre AS nombre_empleado, empleado.apellido1,  
empleado.apellido2, departamento.nombre AS nombre_departamento
```

```
FROM empleado
```

```
JOIN departamento ON empleado.id_departamento = departamento.id;
```

	nombre_empleado character varying (100) 🔒	apellido1 character varying (100) 🔒	apellido2 character varying (100) 🔒	nombre_departamento character varying (100) 🔒
1	Marta	Herrera	Gil	Desarrollo
2	María	Santana	Moreno	Desarrollo
3	Aarón	Rivero	Gómez	Desarrollo
4	Juan	Gómez	López	Sistemas
5	Pilar	Ruiz	[null]	Sistemas
6	Adela	Salas	Díaz	Sistemas
7	Pepe	Ruiz	Santana	Recursos Humanos
8	Adolfo	Rubio	Flores	Recursos Humanos
9	Adrián	Suárez	[null]	Contabilidad
10	Diego	Flores	Salas	I+D
11	Marcos	Loyola	Méndez	I+D

2. 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 empleado.nombre AS nombre_empleado, empleado.apellido1,  
empleado.apellido2, departamento.nombre AS nombre_departamento
```

```
FROM empleado
```

```
JOIN departamento ON empleado.id_departamento = departamento.id
```

```
ORDER BY departamento.nombre, empleado.apellido1, empleado.apellido2,  
empleado.nombre;
```

	nombre_empleado character varying (100) 🔒	apellido1 character varying (100) 🔒	apellido2 character varying (100) 🔒	nombre_departamento character varying (100) 🔒
1	Adrián	Suárez	[null]	Contabilidad
2	Marta	Herrera	Gil	Desarrollo
3	Aarón	Rivero	Gómez	Desarrollo
4	María	Santana	Moreno	Desarrollo
5	Diego	Flores	Salas	I+D
6	Marcos	Loyola	Méndez	I+D
7	Adolfo	Rubio	Flores	Recursos Humanos
8	Pepe	Ruiz	Santana	Recursos Humanos
9	Juan	Gómez	López	Sistemas
10	Pilar	Ruiz	[null]	Sistemas
11	Adela	Salas	Díaz	Sistemas

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

```
SELECT departamento.id, departamento.nombre
```

```
FROM departamento
```

```
WHERE EXISTS (SELECT 1 FROM empleado WHERE empleado.id_departamento =  
departamento.id);
```

	id [PK] integer	nombre character varying (100)
1	1	Desarrollo
2	2	Sistemas
3	3	Recursos Humanos
4	4	Contabilidad
5	5	I+D

4. Devuelve un listado con el identificador, 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.id, departamento.nombre, (departamento.presupuesto -  
departamento.gastos) AS presupuesto_actual
```


```
FROM departamento
```

```
WHERE EXISTS (SELECT 1 FROM empleado WHERE empleado.id_departamento =  
departamento.id);
```

	id [PK] integer	nombre character varying (100)	presupuesto_actual numeric
1	1	Desarrollo	114000
2	2	Sistemas	129000
3	3	Recursos Humanos	255000
4	4	Contabilidad	107000
5	5	I+D	-5000


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

```
SELECT departamento.nombre  
FROM departamento  
JOIN empleado ON empleado.id_departamento = departamento.id  
WHERE empleado.nif = '38382980M';
```

	nombre character varying (100) 
1	Desarrollo

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

```
SELECT departamento.nombre  
FROM departamento  
JOIN empleado ON empleado.id_departamento = departamento.id  
WHERE empleado.nombre = 'Pepe' AND empleado.apellido1 = 'Ruiz' AND  
empleado.apellido2 = 'Santana';
```

	nombre character varying (100) 
1	Recursos Humanos

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

```
SELECT empleado.id, empleado.nombre, empleado.apellido1, empleado.apellido2  
FROM empleado  
JOIN departamento ON empleado.id_departamento = departamento.id  
WHERE departamento.nombre = 'I+D'  
ORDER BY empleado.nombre, empleado.apellido1, empleado.apellido2;
```

	id [PK] integer	nombre character varying (100)	apellido1 character varying (100)	apellido2 character varying (100)
1	10	Diego	Flores	Salas
2	5	Marcos	Loyola	Méndez

8. 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 empleado.id, empleado.nombre, empleado.apellido1, empleado.apellido2
FROM empleado
JOIN departamento ON empleado.id_departamento = departamento.id
WHERE departamento.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
ORDER BY empleado.nombre, empleado.apellido1, empleado.apellido2;
```

	id [PK] integer	nombre character varying (100)	apellido1 character varying (100)	apellido2 character varying (100)
1	2	Adela	Salas	Díaz
2	4	Adrián	Suárez	[null]
3	10	Diego	Flores	Salas
4	9	Juan	Gómez	López
5	5	Marcos	Loyola	Méndez
6	7	Pilar	Ruiz	[null]


9. 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
JOIN departamento ON empleado.id_departamento = departamento.id
WHERE departamento.presupuesto < 100000 OR departamento.presupuesto > 200000;
```

	nombre character varying (100)
1	Adolfo
2	Marcos
3	Pepe
4	Diego

10. 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 DISTINCT departamento.nombre  
FROM departamento  
JOIN empleado ON empleado.id_departamento = departamento.id  
WHERE empleado.apellido2 IS NULL;
```

	<b>nombre</b> character varying (100) 
1	Contabilidad
2	Sistemas