

**“Año de la recuperación y consolidación de la economía peruana  
“**

**UNIVERSIDAD PERUANA LOS ANDES**

**FACULTAD DE INGENIERÍA”**

**ESCUELA PROFESIONAL “INGENIERÍA DE SISTEMAS Y  
COMPUTACIÓN”**

**Cátedra: Base de datos II**

**Catedrático: Fernandez Bejarano Raul Enrique**

**Estudiante: Egoavil Machado Jesus Miguel**

**HUANCAYO PERÚ 2025**



## CREACIÓN DE LA BASE DE DATOS:

```
-- 1. Crear la base de datos
USE master;
GO

CREATE DATABASE Empleados
ON
(
    NAME = Empleados_data,
    FILENAME = 'C:\BaseDatos2024\Empleados_data.mdf',
    SIZE = 5MB,
    MAXSIZE = 20MB,
    FILEGROWTH = 4MB
)
LOG ON
(
    NAME = Empleados_log,
    FILENAME = 'C:\BaseDatos2024\Empleados_log.ldf',
    SIZE = 2MB,
    MAXSIZE = 8MB,
    FILEGROWTH = 2MB
);
GO
```

## CREACIÓN DE LAS TABLAS:

```
-- 3. Crear la tabla 'departamento'
CREATE TABLE departamento (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nombre NVARCHAR(100) NOT NULL,
    presupuesto FLOAT NOT NULL CHECK (presupuesto >= 0),
    gastos FLOAT NOT NULL CHECK (gastos >= 0) -- Cambiado >0 por >=0 para permitir 0
);
GO

-- 4. Crear la tabla 'empleado'
CREATE TABLE empleado (
    id INT IDENTITY(1,1) PRIMARY KEY,
    DNI NVARCHAR(9) NOT NULL UNIQUE,
    nombre NVARCHAR(100) NOT NULL,
    apellidoPaterno NVARCHAR(100) NOT NULL,
    apellidoMaterno NVARCHAR(100),
    id_departamento INT,
    FOREIGN KEY (id_departamento) REFERENCES departamento(id)
);
GO
```

## INGRESAR DATOS:

```

-- 5. Insertar datos en 'departamento'
SET IDENTITY_INSERT departamento ON;
INSERT INTO departamento (id, nombre, presupuesto, gastos) VALUES
(1, 'Desarrollo', 120000, 6000),
(2, 'Sistemas', 150000, 21000),
(3, 'Recursos Humanos', 280000, 25000),
(4, 'Contabilidad', 110000, 3000),
(5, 'ID', 375000, 380000),
(6, 'Proyectos', 0, 0),
(7, 'Publicidad', 0, 1000);
SET IDENTITY_INSERT departamento OFF;
GO

-- 6. Insertar datos en 'empleado'
SET IDENTITY_INSERT empleado ON;
INSERT INTO empleado (id, DNI, nombre, apellidoPaterno, apellidoMaterno, id_departamento) VALUES
(1, '32481596F', 'Aarón', 'Rivero', 'Gómez', 1),
(2, 'Y55756320', 'Adela', 'Salas', 'Díaz', 2),
(3, 'R69706428', 'Adolfo', 'Rubio', 'Flores', 3),
(4, '77705545E', 'Adrián', 'Suárez', NULL, 4),
(5, '170872030', 'Marcos', 'Loyola', 'Méndez', 5),
(6, '38382988M', 'Maria', 'Santana', 'Moreno', 1),
(7, '88576669X', 'Pilar', 'Ruiz', NULL, 2),
(8, '716514312', 'Pepe', 'Ruiz', 'Santana', 3),
(9, '563991830', 'Juan', 'Gomez', 'López', 2),
(10, '46384486H', 'Diego', 'Flores', 'Salas', 5),
(11, '67389283A', 'Marta', 'Herrera', 'Gil', 1),
(12, '41234836R', 'Irene', 'Salas', 'Flores', NULL),
(13, '82635162B', 'Juan Antonio', 'Sáez', 'Guerrero', NULL);
SET IDENTITY_INSERT empleado OFF;
GO

```

## 5. CONSULTAS SOBRE UNA TABLA

### 1. Lista el primer apellido de todos los empleados.

```

--1
SELECT apellidoPaterno FROM empleado;
--2

```

100 %

	apellidoPaterno
1	Rivero
2	Salas
3	Rubio
4	Suárez
5	Loyola
6	Santana
7	Ruiz
8	Ruiz
9	Gomez
10	Flores
11	Herrera
12	Salas
13	Sáez

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

```
--2
SELECT DISTINCT apellidoPaterno FROM empleado;
```

100 %

Results Messages

	apellidoPaterno
1	Flores
2	Gomez
3	Herrera
4	Loyola
5	Rivero
6	Rubio
7	Ruiz
8	Sáez
9	Salas
10	Santana
11	Suárez

3. Lista todas las columnas de la tabla empleado.

```
--3
SELECT * FROM empleado;
--4
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	1	32481596F	Aarón	Rivero	Gómez	1
2	2	Y55756320	Adela	Salas	Díaz	2
3	3	R69706428	Adolfo	Rubio	Flores	3
4	4	77705545E	Adrián	Suárez	NULL	4
5	5	170872030	Marcos	Loyola	Méndez	5
6	6	38382988M	Maria	Santana	Moreno	1
7	7	88576669X	Pilar	Ruiz	NULL	2
8	8	716514312	Pepe	Ruiz	Santana	3
9	9	563991830	Juan	Gomez	López	2
10	10	46384486H	Diego	Flores	Salas	5
11	11	67389283A	Marta	Herrera	Gil	1
12	12	41234836R	Irene	Salas	Flores	NULL
13	13	82635162B	Juan Antonio	Sáez	Guerrero	NULL

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

```
--4
SELECT nombre, apellidoPaterno, apellidoMaterno FROM empleado;
--5
```

100 %

Results Messages

	nombre	apellidoPaterno	apellidoMaterno
1	Aarón	Rivero	Gómez
2	Adela	Salas	Díaz
3	Adolfo	Rubio	Flores
4	Adrián	Suárez	NULL
5	Marcos	Loyola	Méndez
6	Maria	Santana	Moreno
7	Pilar	Ruiz	NULL
8	Pepe	Ruiz	Santana
9	Juan	Gomez	López
10	Diego	Flores	Salas
11	Marta	Herrera	Gil
12	Irene	Salas	Flores
13	Juan Antonio	Sáez	Guerrero

5. Lista el identificador de los departamentos de los empleados que aparecen en la tabla empleado.

```
--5
SELECT id_departamento FROM empleado;
--6
```

100 %

Results Messages

	id_departamento
1	1
2	2
3	3
4	4
5	5
6	1
7	2
8	3
9	2
10	5
11	1
12	NULL
13	NULL

6. Lista el identificador de los departamentos de los empleados que aparecen en la tabla empleado, eliminando los identificadores que aparecen repetidos.

```
--6
SELECT DISTINCT id_departamento FROM empleado;

--7
```

100 %

	id_departamento
1	NULL
2	1
3	2
4	3
5	4
6	5

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

```
--7
SELECT CONCAT(nombre, ' ', apellidoPaterno, ' ', apellidoMaterno) AS nombre_completo FROM empleado;

--8
```

100 %

	nombre_completo
1	Aarón Rivero Gómez
2	Adela Salas Díaz
3	Adolfo Rubio Flores
4	Adrián Suárez
5	Marcos Loyola Méndez
6	Maria Santana Moreno
7	Pilar Ruiz
8	Pepe Ruiz Santana
9	Juan Gomez López
10	Diego Flores Salas
11	Marta Herrera Gil
12	Irene Salas Flores
13	Juan Antonio Sáez Guerrero

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

```
--8
SELECT UPPER(CONCAT(nombre, ' ', apellidoPaterno, ' ', apellidoMaterno)) AS nombre_completo FROM empleado;
```

nombre_completo
1 AARÓN RIVERO GÓMEZ
2 ADELA SALAS DIAZ
3 ADOLFO RUBIO FLORES
4 ADRIÁN SUÁREZ
5 MARCOS LOYOLA MÉNDEZ
6 MARIA SANTANA MORENO
7 PILAR RUIZ
8 PEPE RUIZ SANTANA
9 JUAN GOMEZ LÓPEZ
10 DIEGO FLORES SALAS
11 MARTA HERRERA GIL
12 IRENE SALAS FLORES
13 JUAN ANTONIO SÁEZ GUERRERO

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

```
--9
SELECT LOWER(CONCAT(nombre, ' ', apellidoPaterno, ' ', apellidoMaterno)) AS nombre_completo FROM empleado;

--10
SELECT id,
LEFT(DNI, LEN(DNI) - 1) AS digitos_nif,
SUBSTR(DNI, 4, 1) AS letra_nif
```

nombre_completo
1 aarón rivero gómez
2 adela salas diaz
3 adolfo rubio flores
4 adrián suárez
5 marcos loyola méndez
6 maria santana moreno
7 pilar ruiz
8 pepe ruiz santana
9 juan gomez lópez
10 diego flores salas
11 marta herrera gil
12 irene salas flores
13 juan antonio sáez guerrero

10. Lista el identificador 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.



```
--10
SELECT id,
       LEFT(DNI, LEN(DNI) - 1) AS digitos_nif,
       RIGHT(DNI, 1) AS letra_nif
FROM empleado;
```

100 %

Results Messages

	id	digitos_nif	letra_nif
1	5	17087203	0
2	1	32481596	F
3	6	38382988	M
4	12	41234836	R
5	10	46384486	H
6	9	56399183	0
7	11	67389283	A
8	8	71651431	2
9	4	77705545	E
10	13	82635162	B
11	7	88576669	X
12	3	R6970642	8
13	2	Y5575632	0

11. 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

```
--11
SELECT nombre, (presupuesto - gastos) AS presupuesto_actual
FROM departamento;

--12
SELECT nombre, (presupuesto - gastos) AS presupuesto_actual
```

100 %

Results Messages

	nombre	presupuesto_actual
1	Desarrollo	114000
2	Sistemas	129000
3	Recursos Humanos	255000
4	Contabilidad	107000
5	ID	-5000
6	Proyectos	0
7	Publicidad	-1000

columna que está calculando.

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

```
--12
SELECT nombre, (presupuesto - gastos) AS presupuesto_actual
FROM departamento
ORDER BY presupuesto_actual ASC;
```

100 %

	nombre	presupuesto_actual
1	ID	-5000
2	Publicidad	-1000
3	Proyectos	0
4	Contabilidad	107000
5	Desarrollo	114000
6	Sistemas	129000
7	Recursos Humanos	255000

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

```
--13
SELECT nombre FROM departamento ORDER BY nombre ASC;
```

100 %

	nombre
1	Contabilidad
2	Desarrollo
3	ID
4	Proyectos
5	Publicidad
6	Recursos Humanos
7	Sistemas

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

```
--14
SELECT nombre FROM departamento ORDER BY nombre DESC;
```

100 %	
Results	Messages
nombre	
1	Sistemas
2	Recursos Humanos
3	Publicidad
4	Proyectos
5	ID
6	Desarrollo
7	Contabilidad

15. 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.

```
--15
SELECT apellidoPaterno, apellidoMaterno, nombre
FROM empleado
ORDER BY apellidoPaterno, apellidoMaterno, nombre;
--16
```

100 %

Results Messages

	apellidoPaterno	apellidoMaterno	nombre
1	Flores	Salas	Diego
2	Gomez	López	Juan
3	Herrera	Gil	Marta
4	Loyola	Méndez	Marcos
5	Rivero	Gómez	Aarón
6	Rubio	Flores	Adolfo
7	Ruiz	NULL	Pilar
8	Ruiz	Santana	Pepe
9	Sáez	Guerrero	Juan Antonio
10	Salas	Díaz	Adela
11	Salas	Flores	Irene
12	Santana	Moreno	Maria
13	Suárez	NULL	Adrián

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

```
--16
SELECT nombre, presupuesto
FROM departamento
ORDER BY presupuesto DESC
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;

--17
```

100 %

	nombre	presupuesto
1	ID	375000
2	Recursos Humanos	280000
3	Sistemas	150000

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

```
--17
SELECT nombre, presupuesto
FROM departamento
ORDER BY presupuesto ASC
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;

--18
```

100 %

	nombre	presupuesto
1	Proyectos	0
2	Publicidad	0
3	Contabilidad	110000

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

```
--18
SELECT nombre, gastos
FROM departamento
ORDER BY gastos DESC
OFFSET 0 ROWS FETCH NEXT 2 ROWS ONLY;

--19
```

100 %

	nombre	gastos
1	ID	380000
2	Recursos Humanos	25000

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

```
--19
SELECT nombre, gastos
FROM departamento
ORDER BY gastos ASC
OFFSET 0 ROWS FETCH NEXT 2 ROWS ONLY;

--20
```

	nombre	gastos
1	Proyectos	0
2	Publicidad	1000

20. 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.

```
--20
SELECT *
FROM empleado
ORDER BY id
OFFSET 2 ROWS FETCH NEXT 5 ROWS ONLY;

--21
```

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	3	R69706428	Adolfo	Rubio	Flores	3
2	4	77705545E	Adrián	Suárez	NULL	4
3	5	170872030	Marcos	Loyola	Méndez	5
4	6	38382988M	Maria	Santana	Moreno	1
5	7	88576669X	Pilar	Ruiz	NULL	2

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

```
--21
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto >= 150000;
```

100 %

Results Messages

	nombre	presupuesto
1	Sistemas	150000
2	Recursos Humanos	280000
3	ID	375000

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

```
--22
SELECT nombre, gastos
FROM departamento
WHERE gastos < 5000;
```

100 %

Results Messages

	nombre	gastos
1	Contabilidad	3000
2	Proyectos	0
3	Publicidad	1000

23. 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.

```
--23
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto >= 100000 AND presupuesto <= 200000;
```

```
--24
```

100 %

Results Messages

	nombre	presupuesto
1	Desarrollo	120000
2	Sistemas	150000
3	Contabilidad	110000

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

```
--24
SELECT nombre
FROM departamento
WHERE presupuesto < 100000 OR presupuesto > 200000;
```

100 %

Results Messages

	nombre
1	Recursos Humanos
2	ID
3	Proyectos
4	Publicidad

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

```
--25
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto BETWEEN 100000 AND 200000;
```

100 %

Results Messages

	nombre	presupuesto
1	Desarrollo	120000
2	Sistemas	150000
3	Contabilidad	110000

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

```
--26
SELECT nombre
FROM departamento
WHERE presupuesto NOT BETWEEN 100000 AND 200000;
```

```
--27
```

100 %

Results Messages

	nombre
1	Recursos Humanos
2	ID
3	Proyectos
4	Publicidad

27. 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.

```
--27
SELECT nombre, gastos, presupuesto
FROM departamento
WHERE gastos > presupuesto;

--28
SELECT nombre, gastos, presupuesto
```

100 %

Results Messages

	nombre	gastos	presupuesto
1	ID	380000	375000
2	Publicidad	1000	0

28. 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

```
--28
SELECT nombre, gastos, presupuesto
FROM departamento
WHERE gastos < presupuesto;
```

100 %

Results Messages

	nombre	gastos	presupuesto
1	Desarrollo	6000	120000
2	Sistemas	21000	150000
3	Recursos Humanos	25000	280000
4	Contabilidad	3000	110000

disponen.

29. 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.

```
--29
SELECT nombre, gastos, presupuesto
FROM departamento
WHERE gastos = presupuesto;

--30
```

100 %

Results Messages

	nombre	gastos	presupuesto
1	Proyectos	0	0

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



```
--30
SELECT * FROM empleado
WHERE apellidoMaterno IS NULL;
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	4	77705545E	Adrián	Suárez	NULL	4
2	7	88576669X	Pilar	Ruiz	NULL	2

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

```
--31
SELECT * FROM empleado
WHERE apellidoMaterno IS NOT NULL;

--32
SELECT * FROM empleado
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	1	32481596F	Aarón	Rivero	Gómez	1
2	2	Y55756320	Adela	Salas	Díaz	2
3	3	R69706428	Adolfo	Rubio	Flores	3
4	5	170872030	Marcos	Loyola	Méndez	5
5	6	38382988M	Maria	Santana	Moreno	1
6	8	716514312	Pepe	Ruiz	Santana	3
7	9	563991830	Juan	Gomez	López	2
8	10	46384486H	Diego	Flores	Salas	5
9	11	67389283A	Marta	Herrera	Gil	1
10	12	41234836R	Irene	Salas	Flores	NULL
11	13	82635162B	Juan Antonio	Sáez	Guerrero	NULL

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

```
--32
SELECT * FROM empleado
WHERE apellidoMaterno = 'López';

--33
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	9	563991830	Juan	Gomez	López	2

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

```
--33
SELECT * FROM empleado
WHERE apellidoMaterno = 'Diaz' OR apellidoMaterno = 'Moreno';
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	2	Y55756320	Adela	Salas	Diaz	2
2	6	38382988M	Maria	Santana	Moreno	1

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

```
--34
SELECT * FROM empleado
WHERE apellidoMaterno IN ('Diaz', 'Moreno');
```

```
--35
SELECT nombre, apellidoPaterno, apellidoMaterno, DNI
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	2	Y55756320	Adela	Salas	Diaz	2
2	6	38382988M	Maria	Santana	Moreno	1

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

```
--35
SELECT nombre, apellidoPaterno, apellidoMaterno, DNI
FROM empleado
WHERE id_departamento = 3;
```

100 %

Results Messages

	nombre	apellidoPaterno	apellidoMaterno	DNI
1	Adolfo	Rubio	Flores	R69706428
2	Pepe	Ruiz	Santana	716514312

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

```
--36
SELECT nombre, apellidoPaterno, apellidoMaterno, DNI
FROM empleado
WHERE id_departamento IN (2, 4, 5);

--Composicion interna
```

100 %

Results Messages

	nombre	apellidoPaterno	apellidoMaterno	DNI
1	Adela	Salas	Diaz	Y55756320
2	Adrián	Suárez	NULL	77705545E
3	Marcos	Loyola	Méndez	170872030
4	Pilar	Ruiz	NULL	88576669X
5	Juan	Gomez	López	563991830
6	Diego	Flores	Salas	46384486H

### CONSULTAS MULTITABLA (COMPOSICIÓN INTERNA)

Resuelva todas las consultas utilizando la sintaxis de SQL1 y SQL2.

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

--Composicion interna

--1

SELECT e.\*, d.\*

FROM empleado e, departamento d

WHERE e.id\_departamento = d.id;

SELECT e.\*, d.\*

FROM empleado e

INNER JOIN departamento d ON e.id\_departamento = d.id;

100 %

Results

Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento	id	nombre	presupuesto	gastos
1	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000	6000
2	2	Y55756320	Adela	Salas	Díaz	2	2	Sistemas	150000	21000
3	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000	25000
4	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000	3000
5	5	170872030	Marcos	Loyola	Méndez	5	5	ID	375000	380000
6	6	38382988M	Maria	Santana	Moreno	1	1	Desarrollo	120000	6000
7	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000	21000
8	8	716514312	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000	25000
9	9	563991830	Juan	Gomez	López	2	2	Sistemas	150000	21000
10	10	46384486H	Diego	Flores	Salas	5	5	ID	375000	380000
11	11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120000	6000

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento	id	nombre	presupuesto	gastos
1	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000	6000
2	2	Y55756320	Adela	Salas	Díaz	2	2	Sistemas	150000	21000
3	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000	25000
4	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000	3000
5	5	170872030	Marcos	Loyola	Méndez	5	5	ID	375000	380000
6	6	38382988M	Maria	Santana	Moreno	1	1	Desarrollo	120000	6000
7	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000	21000
8	8	716514312	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000	25000
9	9	563991830	Juan	Gomez	López	2	2	Sistemas	150000	21000
10	10	46384486H	Diego	Flores	Salas	5	5	ID	375000	380000
11	11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120000	6000

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.

```
--2
SELECT e.*, d.*
FROM empleado e, departamento d
WHERE e.id_departamento = d.id
ORDER BY d.nombre, e.apellidoPaterno, e.apellidoMaterno, e.nombre;

SELECT e.*, d.*
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
ORDER BY d.nombre, e.apellidoPaterno, e.apellidoMaterno, e.nombre;

--3
```

100 %

Results

Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento	id	nombre	presupuesto	gastos
1	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000	3000
2	11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120000	6000
3	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000	6000
4	6	38382988M	Maria	Santana	Moreno	1	1	Desarrollo	120000	6000
5	10	46384486H	Diego	Flores	Salas	5	5	ID	375000	380000
6	5	170872030	Marcos	Loyola	Méndez	5	5	ID	375000	380000
7	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000	25000
8	8	716514312	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000	25000
9	9	563991830	Juan	Gomez	López	2	2	Sistemas	150000	21000
10	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000	21000
11	2	Y55756320	Adela	Salas	Diaz	2	2	Sistemas	150000	21000

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento	id	nombre	presupuesto	gastos
1	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000	3000
2	11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120000	6000
3	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000	6000
4	6	38382988M	Maria	Santana	Moreno	1	1	Desarrollo	120000	6000
5	10	46384486H	Diego	Flores	Salas	5	5	ID	375000	380000
6	5	170872030	Marcos	Loyola	Méndez	5	5	ID	375000	380000
7	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000	25000
8	8	716514312	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000	25000
9	9	563991830	Juan	Gomez	López	2	2	Sistemas	150000	21000
10	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000	21000
11	2	Y55756320	Adela	Salas	Diaz	2	2	Sistemas	150000	21000

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

```
--3
SELECT DISTINCT d.id, d.nombre
FROM empleado e, departamento d
WHERE e.id_departamento = d.id;

--4
SELECT DISTINCT d.id, d.nombre
FROM departamento d
INNER JOIN empleado e ON e.id_departamento = d.id;
```

100 %

Results Messages

	id	nombre
1	1	Desarrollo
2	2	Sistemas
3	3	Recursos Humanos
4	4	Contabilidad
5	5	ID

  

	id	nombre
1	1	Desarrollo
2	2	Sistemas
3	3	Recursos Humanos
4	4	Contabilidad
5	5	ID

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).

```
--4
SELECT DISTINCT d.id, d.nombre, (d.presupuesto - d.gastos) AS presupuesto_actual
FROM empleado e, departamento d
WHERE e.id_departamento = d.id;

--5
SELECT DISTINCT d.id, d.nombre, (d.presupuesto - d.gastos) AS presupuesto_actual
FROM departamento d
INNER JOIN empleado e ON e.id_departamento = d.id;
```

100 %

Results Messages

	id	nombre	presupuesto_actual
1	1	Desarrollo	114000
2	2	Sistemas	129000
3	3	Recursos Humanos	255000
4	4	Contabilidad	107000
5	5	ID	-5000

  

	id	nombre	presupuesto_actual
1	1	Desarrollo	114000
2	2	Sistemas	129000
3	3	Recursos Humanos	255000
4	4	Contabilidad	107000
5	5	ID	-5000

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

```
--5
SELECT d.nombre
FROM empleado e, departamento d
WHERE e.id_departamento = d.id
AND e.DNI = '38382988M';

SELECT d.nombre
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE e.DNI = '38382988M';

--6
SELECT
```

100 %

Results Messages

	nombre
1	Desarrollo

  

	nombre
1	Desarrollo

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



```
--6
SELECT d.nombre
FROM empleado e, departamento d
WHERE e.id_departamento = d.id
AND e.nombre = 'Pepe'
AND e.apellidoPaterno = 'Ruiz'
AND e.apellidoMaterno = 'Santana';

SELECT d.nombre
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE e.nombre = 'Pepe' AND e.apellidoPaterno = 'Ruiz' AND e.apellidoMaterno = 'Santana';
```

100 %

Results Messages

	nombre
1	Recursos Humanos

  

	nombre
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.

```
--7
SELECT e.*
FROM empleado e, departamento d
WHERE e.id_departamento = d.id AND d.nombre = 'ID'
ORDER BY e.apellidoPaterno, e.apellidoMaterno, e.nombre;

SELECT e.*
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE d.nombre = 'ID'
ORDER BY e.apellidoPaterno, e.apellidoMaterno, e.nombre;
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	10	46384486H	Diego	Flores	Salas	5
2	5	170872030	Marcos	Loyola	Méndez	5

  

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	10	46384486H	Diego	Flores	Salas	5
2	5	170872030	Marcos	Loyola	Méndez	5

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.

```
--8
SELECT e.*
FROM empleado e, departamento d
WHERE e.id_departamento = d.id
AND d.nombre IN ('Sistemas', 'Contabilidad', 'ID')
ORDER BY e.apellidoPaterno, e.apellidoMaterno, e.nombre;

SELECT e.*
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE d.nombre IN ('Sistemas', 'Contabilidad', 'ID')
ORDER BY e.apellidoPaterno, e.apellidoMaterno, e.nombre;
```

100 %

Results Messages

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	10	46384486H	Diego	Flores	Salas	5
2	9	563991830	Juan	Gomez	López	2
3	5	170872030	Marcos	Loyola	Méndez	5
4	7	88576669X	Pilar	Ruiz	NULL	2
5	2	Y55756320	Adela	Salas	Diaz	2
6	4	77705545E	Adrián	Suárez	NULL	4

  

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	10	46384486H	Diego	Flores	Salas	5
2	9	563991830	Juan	Gomez	López	2
3	5	170872030	Marcos	Loyola	Méndez	5
4	7	88576669X	Pilar	Ruiz	NULL	2
5	2	Y55756320	Adela	Salas	Diaz	2
6	4	77705545E	Adrián	Suárez	NULL	4

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

```
--9|
SELECT e.nombre
FROM empleado e, departamento d
WHERE e.id_departamento = d.id
AND (d.presupuesto < 100000 OR d.presupuesto > 200000);

SELECT e.nombre
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE d.presupuesto NOT BETWEEN 100000 AND 200000;
```

100 %

Results Messages

	nombre
1	Adolfo
2	Marcos
3	Pepe
4	Diego

	nombre
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.

```
--10
SELECT DISTINCT d.nombre
FROM empleado e, departamento d
WHERE e.id_departamento = d.id
AND e.apellidoMaterno IS NULL;

SELECT DISTINCT d.nombre
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE e.apellidoMaterno IS NULL;
```

100 %

Results Messages

	nombre
1	Contabilidad
2	Sistemas

  

	nombre
1	Contabilidad
2	Sistemas

### 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.

```
--1
SELECT e.*, d.*
FROM empleado e
LEFT JOIN departamento d ON e.id_departamento = d.id;
```

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento	id	nombre	presupuesto	gastos
1	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000	6000
2	2	Y55756320	Adela	Salas	Díaz	2	2	Sistemas	150000	21000
3	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000	25000
4	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000	3000
5	5	170872030	Marcos	Loyola	Méndez	5	5	ID	375000	380000
6	6	38382988M	Maria	Santana	Moreno	1	1	Desarrollo	120000	6000
7	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000	21000
8	8	716514312	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000	25000
9	9	563991830	Juan	Gomez	López	2	2	Sistemas	150000	21000
10	10	46384486H	Diego	Flores	Salas	5	5	ID	375000	380000
11	11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120000	6000
12	12	41234836R	Irene	Salas	Flores	NULL	NULL	NULL	NULL	NULL
13	13	82635162B	Juan Antonio	Sáez	Guerrero	NULL	NULL	NULL	NULL	NULL

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

```
--2
SELECT e.*
FROM empleado e
LEFT JOIN departamento d ON e.id_departamento = d.id
WHERE d.id IS NULL;
```

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	12	41234836R	Irene	Salas	Flores	NULL
2	13	82635162B	Juan Antonio	Sáez	Guerrero	NULL

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

```
--3
SELECT d.*
FROM departamento d
LEFT JOIN empleado e ON d.id = e.id_departamento
WHERE e.id IS NULL;

SELECT d.*
FROM empleado e
RIGHT JOIN departamento d ON e.id_departamento = d.id
WHERE e.id IS NULL;

--4
SELECT * FROM departamento
```

100 %

Results Messages

	id	nombre	presupuesto	gastos
1	6	Proyectos	0	0
2	7	Publicidad	0	1000

---

	id	nombre	presupuesto	gastos
1	6	Proyectos	0	0
2	7	Publicidad	0	1000

4. 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.

```
--4
SELECT e.*, d.*
FROM empleado e
FULL OUTER JOIN departamento d ON e.id_departamento = d.id
ORDER BY d.nombre;

--5
SELECT * FROM empleado e
FULL OUTER JOIN departamento d ON e.id_departamento = d.id
ORDER BY d.nombre;
```

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento	id	nombre	presupuesto	gastos
1	12	41234836R	Irene	Salas	Flores	NULL	NULL	NULL	NULL	NULL
2	13	82635162B	Juan Antonio	Sáez	Guerrero	NULL	NULL	NULL	NULL	NULL
3	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000	3000
4	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000	6000
5	6	38382988M	Maria	Santana	Moreno	1	1	Desarrollo	120000	6000
6	11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120000	6000
7	10	46384486H	Diego	Flores	Salas	5	5	ID	375000	380000
8	5	170872030	Marcos	Loyola	Méndez	5	5	ID	375000	380000
9	NULL	NULL	NULL	NULL	NULL	NULL	6	Proyectos	0	0
10	NULL	NULL	NULL	NULL	NULL	NULL	7	Publicidad	0	1000
11	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000	25000
12	8	716514312	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000	25000
13	9	563991830	Juan	Gomez	López	2	2	Sistemas	150000	21000
14	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000	21000
15	2	Y55756320	Adela	Salas	Diaz	2	2	Sistemas	150000	21000

5. 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.

```
--5
SELECT e.*, d.*
FROM empleado e
FULL OUTER JOIN departamento d ON e.id_departamento = d.id
WHERE e.id_departamento IS NULL OR d.id IS NULL
ORDER BY d.nombre;
```

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento	id	nombre	presupuesto	gastos
1	12	41234836R	Irene	Salas	Flores	NULL	NULL	NULL	NULL	NULL
2	13	82635162B	Juan Antonio	Sáez	Guerrero	NULL	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL	NULL	NULL	6	Proyectos	0	0
4	NULL	NULL	NULL	NULL	NULL	NULL	7	Publicidad	0	1000

## CONSULTAS RESUMEN

1. Calcula la suma del presupuesto de todos los departamentos.



```
--1
SELECT SUM(presupuesto) AS suma_presupuestos
FROM departamento;
```

100 %	
Results Messages	
	suma_presupuestos
1	1035000

2. Calcula la media del presupuesto de todos los departamentos.

```
--2
SELECT AVG(presupuesto) AS media_presupuesto
FROM departamento;
```

100 %	
Results Messages	
	media_presupuesto
1	147857.142857143

3. Calcula el valor mínimo del presupuesto de todos los departamentos.

```
--3
SELECT MIN(presupuesto) AS presupuesto_minimo
FROM departamento;
```

```
--4
SELECT
```

100 %	
Results Messages	
	presupuesto_minimo
1	0

4. Calcula el nombre del departamento y el presupuesto que tiene asignado, del departamento con menor presupuesto.

```
--4
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto = (SELECT MIN(presupuesto) FROM departamento);
```

100 %

Results Messages

	nombre	presupuesto
1	Proyectos	0
2	Publicidad	0

5. Calcula el valor máximo del presupuesto de todos los departamentos.

```
--5
SELECT MAX(presupuesto) AS presupuesto_maximo
FROM departamento;

--6
```

presupuesto_maximo	
1	375000

6. Calcula el nombre del departamento y el presupuesto que tiene asignado, del departamento con mayor presupuesto.

```
--6
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto = (SELECT MAX(presupuesto) FROM departamento);
```

	nombre	presupuesto
1	ID	375000

7. Calcula el número total de empleados que hay en la tabla empleado.

```
--7
SELECT COUNT(*) AS total_empleados
FROM empleado;
```

total_empleados	
1	13

8. Calcula el número de empleados que no tienen NULL en su segundo apellido.

```
--8
SELECT COUNT(*) AS empleados_con_apellido_materno
FROM empleado
WHERE apellidoMaterno IS NOT NULL;

--9
```

empleados_con_apellido_materno	
1	11

9. Calcula el número de empleados que hay en cada departamento. Tienes que devolver dos columnas, una con el nombre del departamento y otra con el número de empleados que tiene asignados.

```
--9
SELECT d.nombre AS nombre_departamento, COUNT(e.id) AS numero_empleados
FROM departamento d
JOIN empleado e ON d.id = e.id_departamento
GROUP BY d.nombre;
--10
```

100 %

Results Messages

	nombre_departamento	numero_empleados
1	Contabilidad	1
2	Desarrollo	3
3	ID	2
4	Recursos Humanos	2
5	Sistemas	3

10. Calcula el nombre de los departamentos que tienen más de 2 empleados. El resultado debe tener dos columnas, una con el nombre del departamento y otra con el número de empleados que tiene asignados.

```
--10
SELECT d.nombre AS nombre_departamento, COUNT(e.id) AS numero_empleados
FROM departamento d
JOIN empleado e ON d.id = e.id_departamento
GROUP BY d.nombre
HAVING COUNT(e.id) > 2;
```

100 %

Results Messages

	nombre_departamento	numero_empleados
1	Desarrollo	3
2	Sistemas	3

11. Calcula el número de empleados que trabajan en cada uno de los departamentos. El resultado de esta consulta también tiene que incluir aquellos departamentos que no tienen ningún empleado asociado.

```
--11
SELECT d.nombre AS nombre_departamento, COUNT(e.id) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.id = e.id_departamento
GROUP BY d.nombre;
```

	nombre_departamento	numero_empleados
1	Contabilidad	1
2	Desarrollo	3
3	ID	2
4	Proyectos	0
5	Publicidad	0
6	Recursos Humanos	2
7	Sistemas	3

12. Calcula el número de empleados que trabajan en cada uno de los departamentos que tienen un presupuesto mayor a 200000 euros.

```
--12
SELECT d.nombre AS nombre_departamento, COUNT(e.id) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.id = e.id_departamento
WHERE d.presupuesto > 200000
GROUP BY d.nombre;

--Subconsulta
--1
```

	nombre_departamento	numero_empleados
1	ID	2
2	Recursos Humanos	2

## SUBCONSULTAS

### CON OPERADORES BÁSICOS DE COMPARACIÓN

1. Devuelve un listado con todos los empleados que tiene el departamento de Sistemas. (Sin utilizar INNER JOIN).

```
--Subconsulta
--1
SELECT *
FROM empleado
WHERE id_departamento = (
    SELECT id
    FROM departamento
    WHERE nombre = 'Sistemas'
);
--2
```

100 %

	id	DNI	nombre	apellidoPaterno	apellidoMaterno	id_departamento
1	2	Y55756320	Adela	Salas	Diaz	2
2	7	88576669X	Pilar	Ruiz	NULL	2
3	9	563991830	Juan	Gomez	López	2

2. Devuelve el nombre del departamento con mayor presupuesto y la cantidad que tiene asignada.

```
--2
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto = (
    SELECT MAX(presupuesto)
    FROM departamento
);
```

100 %

	nombre	presupuesto
1	ID	375000

3. Devuelve el nombre del departamento con menor presupuesto y la cantidad que tiene asignada. SUBCONSULTAS CON ALL Y ANY

```
--3
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto = (
    SELECT MIN(presupuesto)
    FROM departamento
);
```

100 %

Results Messages

	nombre	presupuesto
1	Proyectos	0
2	Publicidad	0

4. Devuelve el nombre del departamento con mayor presupuesto y la cantidad que tiene asignada. Sin hacer uso de MAX, ORDER BY ni LIMIT.

```
--4 Sin max, order by, limit
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto > ALL (
    SELECT presupuesto
    FROM departamento
    WHERE presupuesto <> presupuesto
);

SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto >= ALL (
    SELECT presupuesto
    FROM departamento
);
```

100 %

Results Messages

	nombre	presupuesto
1	Desarrollo	120000
2	Sistemas	150000
3	Recursos Humanos	280000
4	Contabilidad	110000
5	ID	375000
6	Proyectos	0
7	Publicidad	0

  

	nombre	presupuesto
1	ID	375000

5. Devuelve el nombre del departamento con menor presupuesto y la cantidad que tiene asignada. Sin hacer uso de MIN, ORDER BY ni LIMIT.

```
--5 Sin order by, limit
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto <= ALL (
    SELECT presupuesto
    FROM departamento
);

--6 Any
SELECT nombre
```

100 %

Results Messages

	nombre	presupuesto
1	Proyectos	0
2	Publicidad	0

6. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando ALL O ANY).

```
--6 Any
SELECT nombre
FROM departamento
WHERE id = ANY (
    SELECT id_departamento
    FROM empleado
    WHERE id_departamento IS NOT NULL
);

--7 All
SELECT nombre
```

100 %

Results Messages

	nombre
1	Desarrollo
2	Sistemas
3	Recursos Humanos
4	Contabilidad
5	ID

7. Devuelve los nombres de los departamentos que no tienen empleados asociados. (Utilizando ALL o ANY).

```
--7 All
SELECT nombre
FROM departamento
WHERE id <> ALL (
    SELECT id_departamento
    FROM empleado
    WHERE id_departamento IS NOT NULL
);
```

100 %

Results Messages

	nombre
1	Proyectos
2	Publicidad

## SUBCONSULTAS CON IN Y NOT IN

8. Devuelve los nombres de los departamentos que tienen empleados asociados.  
(Utilizando IN o NOT IN).

```
--8 In
SELECT nombre
FROM departamento
WHERE id IN (
    SELECT id_departamento
    FROM empleado
    WHERE id_departamento IS NOT NULL
);
```

100 %

Results Messages

	nombre
1	Desarrollo
2	Sistemas
3	Recursos Humanos
4	Contabilidad
5	ID

9. Devuelve los nombres de los departamentos que no tienen empleados asociados.  
(Utilizando IN O NOT IN).



```
--9 Not in
SELECT nombre
FROM departamento
WHERE id NOT IN (
    SELECT id_departamento
    FROM empleado
    WHERE id_departamento IS NOT NULL
);
```

100 %

Results Messages

	nombre
1	Proyectos
2	Publicidad

### SUBCONSULTAS CON EXISTS Y NOT EXISTS

10. Devuelve los nombres de los departamentos que tienen empleados asociados.  
(Utilizando EXISTS O NOT EXISTS).

```
--10 Exists
SELECT d.nombre
FROM departamento d
WHERE EXISTS (
    SELECT 1
    FROM empleado e
    WHERE e.id_departamento = d.id
);
```

100 %

Results Messages

	nombre
1	Desarrollo
2	Sistemas
3	Recursos Humanos
4	Contabilidad
5	ID

11. Devuelve los nombres de los departamentos que tienen empleados asociados.  
(Utilizando EXISTS O NOT EXISTS).

```
--11 Not exists  
SELECT d.nombre  
FROM departamento d  
WHERE NOT EXISTS (  
    SELECT 1  
    FROM empleado e  
    WHERE e.id_departamento = d.id  
);
```

100 %

Results Messages

	nombre
1	Proyectos
2	Publicidad