

“UNIVERSIDAD PERUANA LOS ANDES”
FACULTAD DE INGENIERÍA DE SISTEMAS Y
COMPUTACIÓN



CURSO:

Base de Datos II

DOCENTE:

Mg.Raúl Fernandez Bejarano

CICLO:

V

AULA:

A1

INTEGRANTES:

Luis Ramirez Brayan Esnyder

HUANCAYO – PERU

2024

1-2.-Base de Datos SQLSERVER:

```
USE master;
GO

CREATE DATABASE GestionEmpleado
ON PRIMARY
(
    NAME = GestionEmpleado_Data,
    FILENAME = 'D:\BaseDatos2024\GestionEmpleado_Data.mdf',
    SIZE = 5MB,
    MAXSIZE = 20MB,
    FILEGROWTH = 5MB
)
LOG ON
(
    NAME = GestionEmpleado_Log,
    FILENAME = 'D:\BaseDatos2024\GestionEmpleado_Log.ldf',
    SIZE = 1MB,
    MAXSIZE = 4MB,
    FILEGROWTH = 1MB
);
GO
```

```
SQLQuery2.sql - D...3GC\USUARIO (63))* X SQLQuery1.sql - D...3GC\USUARIO (56))*
use GestionEmpleado
go

create table departamento (
    codigo INT IDENTITY(1,1) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    presupuesto numeric(10,2) NOT NULL,
    gastos numeric(10,2) NOT NULL
)
CREATE TABLE empleado (
    codigo INT IDENTITY(1,1) PRIMARY KEY,
    dni VARCHAR(9) NOT NULL UNIQUE,
    nombre VARCHAR(100) NOT NULL,
    apePaterno VARCHAR(100) NOT NULL,
    apeMaterno VARCHAR(100),
    codigo_departamento INT,
    FOREIGN KEY (codigo_departamento) REFERENCES departamento(codigo)
)
go
```

```
USE GestionEmpleado;  
GO
```

```
INSERT INTO departamento VALUES ('Desarrollo', 120000, 6000);  
INSERT INTO departamento VALUES ('Sistemas', 150000, 21000);  
INSERT INTO departamento VALUES ('Recursos Humanos', 280000, 25000);  
INSERT INTO departamento VALUES ('Contabilidad', 110000, 3000);  
INSERT INTO departamento VALUES ('I+D', 375000, 380000);  
INSERT INTO departamento VALUES ('Proyectos', 0, 0);  
INSERT INTO departamento VALUES ('Publicidad', 0, 1000);  
  
INSERT INTO empleado VALUES ('32481596', 'Aarón', 'Rivero', 'Gómez', 1);  
INSERT INTO empleado VALUES ('Y5575632', 'Adela', 'Salas', 'Díaz', 2);  
INSERT INTO empleado VALUES ('R6970642', 'Adolfo', 'Rubio', 'Flores', 3);  
INSERT INTO empleado VALUES ('77705545', 'Adrián', 'Suárez', NULL, 4);  
INSERT INTO empleado VALUES ('17087203', 'Marcos', 'Loyola', 'Méndez', 5);  
INSERT INTO empleado VALUES ('38382980', 'María', 'Santana', 'Moreno', 1);  
INSERT INTO empleado VALUES ('80576669', 'Pilar', 'Ruiz', NULL, 2);  
INSERT INTO empleado VALUES ('71651431', 'Pepe', 'Ruiz', 'Santana', 3);  
INSERT INTO empleado VALUES ('56399183', 'Juan', 'Gómez', 'López', 2);  
INSERT INTO empleado VALUES ('46384486', 'Diego', 'Flores', 'Salas', 5);  
INSERT INTO empleado VALUES ('67389283', 'Marta', 'Herrera', 'Gil', 1);  
INSERT INTO empleado VALUES ('41234836', 'Irene', 'Salas', 'Flores', NULL);  
INSERT INTO empleado VALUES ('82635162', 'Juan Antonio', 'Sáez', 'Guerrero', NULL);
```

3-. Consultas sobre una tabla

1. Lista el primer apellido de todos los empleados.

```
SELECT apePaterno  
FROM empleado;
```

100 %

Results Messages

	apePaterno
1	Rivero
2	Salas
3	Rubio
4	Suárez
5	Loyola
6	Santana
7	Ruiz
8	Ruiz
9	Gómez
10	Flores
11	Herrera
12	Salas
13	Sáez
14	Rivero
15	Salas
16	Rubio
17	Suárez
18	Loyola
19	Santana
20	Ruiz
21	Ruiz
22	Gómez
23	Flores
24	Herrera
25	Salas
26	Sáez

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

```
SELECT DISTINCT apePaterno  
FROM empleado;
```

%

Results Messages

apePaterno
Flores
Gómez
Herrera
Loyola
Rivero
Rubio
Ruiz
Sáez
Salas
Santana
Suárez

- Lista todas las columnas de la tabla empleado.

SELECT *
FROM empleado;

10 %

Results Messages

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	1	32481596F	Aarón	Rivero	Gómez	1
2	2	Y5575632D	Adela	Salas	Díaz	2
3	3	R6970642B	Adolfo	Rubio	Flores	3
4	4	77705545E	Adrián	Suárez	NULL	4
5	5	17087203C	Marcos	Loyola	Méndez	5
6	6	38382980M	María	Santana	Moreno	1
7	7	80576669X	Pilar	Ruiz	NULL	2
8	8	71651431Z	Pepe	Ruiz	Santana	3
9	9	56399183D	Juan	Gómez	López	2
0	10	46384486H	Diego	Flores	Salas	5
1	11	67389283A	Marta	Herrera	Gil	1
2	12	41234836R	Irene	Salas	Flores	NULL
3	13	82635162B	Juan Antonio	Sáez	Guerrero	NULL
4	14	32481596	Aarón	Rivero	Gómez	1
5	15	Y5575632	Adela	Salas	Díaz	2
6	16	R6970642	Adolfo	Rubio	Flores	3
7	17	77705545	Adrián	Suárez	NULL	4
8	18	17087203	Marcos	Loyola	Méndez	5
9	19	38382980	María	Santana	Moreno	1
0	20	80576669	Pilar	Ruiz	NULL	2
1	21	71651431	Pepe	Ruiz	Santana	3
2	22	56399183	Juan	Gómez	López	2
3	23	46384486	Diego	Flores	Salas	5
4	24	67389283	Marta	Herrera	Gil	1
5	25	41234836	Irene	Salas	Flores	NULL
6	26	82635162	Juan Antonio	Sáez	Guerrero	NULL

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

SELECT nombre, apePaterno, apeMaterno
FROM empleado;

00 %

Results Messages

	nombre	apePaterno	apeMaterno
1	Aarón	Rivero	Gómez
2	Adela	Salas	Díaz
3	Adolfo	Rubio	Flores
4	Adrián	Suárez	NULL
5	Marcos	Loyola	Méndez
6	María	Santana	Moreno
7	Pilar	Ruiz	NULL
8	Pepe	Ruiz	Santana
9	Juan	Gómez	López
10	Diego	Flores	Salas
11	Marta	Herrera	Gil
12	Irene	Salas	Flores
13	Juan Antonio	Sáez	Guerrero
14	Aarón	Rivero	Gómez
15	Adela	Salas	Díaz
16	Adolfo	Rubio	Flores
17	Adrián	Suárez	NULL
18	Marcos	Loyola	Méndez
19	María	Santana	Moreno
20	Pilar	Ruiz	NULL
21	Pepe	Ruiz	Santana
22	Juan	Gómez	López
23	Diego	Flores	Salas
24	Marta	Herrera	Gil
25	Irene	Salas	Flores
26	Juan Antonio	Sáez	Guerrero

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

```
SELECT DISTINCT codigo_departamento
FROM empleado;
```

00 %

Results Messages

	codigo_departamento
1	NULL
2	1
3	2
4	3
5	4
6	5

6. 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;
```

00 %

Results Messages

	codigo_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.

```
SELECT
nombre + ' ' + apePaterno + ' ' + COALESCE(apeMaterno, '') AS nombre_completo
FROM empleado;
```

00 %

Results Messages

	nombre_completo
1	Aarón Rivero Gómez
2	Adela Salas Díaz
3	Adolfo Rubio Flores
4	Adrián Suárez
5	Marcos Loyola Méndez
6	María Santana Moreno
7	Pilar Ruiz
8	Pepe Ruiz Santana
9	Juan Gómez López
10	Diego Flores Salas
11	Marta Herrera Gil
12	Irene Salas Flores
13	Juan Antonio Sáez Guerrero
14	Aarón Rivero Gómez
15	Adela Salas Díaz
16	Adolfo Rubio Flores
17	Adrián Suárez
18	Marcos Loyola Méndez
19	María Santana Moreno
20	Pilar Ruiz
21	Pepe Ruiz Santana
22	Juan Gómez López
23	Diego Flores Salas
24	Marta Herrera Gil
25	Irene Salas Flores
26	Juan Antonio Sáez Guerrero

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

```
SELECT  
    UPPER(nombre + ' ' + apePaterno + ' ' + COALESCE(apeMaterno, '')) AS nombre_completo  
FROM empleado;
```

100 %

	nombre_completo
1	AARÓN RIVERO GÓMEZ
2	ADELA SALAS DÍAZ
3	ADOLFO RUBIO FLORES
4	ADRIÁN SUÁREZ
5	MARCOS LOYOLA MÉNDEZ
6	MARÍA SANTANA MORENO
7	PILAR RUIZ
8	PEPE RUIZ SANTANA
9	JUAN GÓMEZ LÓPEZ
10	DIEGO FLORES SALAS
11	MARTA HERRERA GIL
12	IRENE SALAS FLORES
13	JUAN ANTONIO SÁEZ GUERRERO
14	AARÓN RIVERO GÓMEZ
15	ADELA SALAS DÍAZ
16	ADOLFO RUBIO FLORES
17	ADRIÁN SUÁREZ
18	MARCOS LOYOLA MÉNDEZ
19	MARÍA SANTANA MORENO
20	PILAR RUIZ
21	PEPE RUIZ SANTANA
22	JUAN GÓMEZ LÓPEZ
23	DIEGO FLORES SALAS
24	MARTA HERRERA GIL
25	IRENE SALAS FLORES
26	JUAN ANTONIO SÁEZ GUERRERO

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

```
SELECT  
    LOWER(nombre + ' ' + apePaterno + ' ' + COALESCE(apeMaterno, '')) AS nombre_completo  
FROM empleado;
```

0 %

	nombre_completo
	aarón rivero gómez
	adela salas díaz
	adolfo rubio flores
	adrián suárez
	marcos loyola méndez
	maría santana moreno
	pilar ruiz
	pepe ruiz santana
	juan gómez lópez
	diego flores salas
	marta herrera gil
!	irene salas flores
!	juan antonio sáez guerrero
!	aarón rivero gómez
!	adela salas díaz
!	adolfo rubio flores
!	adrián suárez
!	marcos loyola méndez
!	maría santana moreno
!	pilar ruiz
!	pepe ruiz santana
!	juan gómez lópez
!	diego flores salas
!	marta herrera gil
!	irene salas flores
!	juan antonio sáez guerrero

10. 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,
    SUBSTRING(dni, 1, LEN(dni) - 1) AS nif_digitos,
    RIGHT(dni, 1) AS nif_letra
FROM empleado;
```

100 %

Results Messages

	codigo	nif_digitos	nif_letra
1	18	1708720	3
2	5	17087203	C
3	14	3248159	6
4	1	32481596	F
5	19	3838298	0
6	6	38382980	M
7	25	4123483	6
8	12	41234836	R
9	23	4638448	6
10	10	46384486	H
11	22	5639918	3
12	9	56399183	D
13	24	6738928	3
14	11	67389283	A
15	21	7165143	1
16	8	71651431	Z
17	17	7770554	5
18	4	77705545	E
19	20	8057666	9
20	7	80576669	X
21	26	8263516	2
22	13	82635162	B
23	16	R697064	2
24	3	R6970642	B
25	15	Y557563	2
26	2	Y5575632	D

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 columna columna que está calculando.

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

00 %

Results Messages

	nombre	presupuesto_actual
1	Desarrollo	114000.00
2	Sistemas	129000.00
3	Recursos Humanos	255000.00
4	Contabilidad	107000.00
5	I+D	-5000.00
6	Proyectos	0.00
7	Publicidad	-1000.00
8	Desarrollo	114000.00
9	Sistemas	129000.00
10	Recursos Humanos	255000.00
11	Contabilidad	107000.00
12	I+D	-5000.00
13	Proyectos	0.00
14	Publicidad	-1000.00

12. 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;
```

nombre	presupuesto_actual
I+D	-5000.00
I+D	-5000.00
Publicidad	-1000.00
Publicidad	-1000.00
Proyectos	0.00
Proyectos	0.00
Contabilidad	107000.00
Contabilidad	107000.00
Desarrollo	114000.00
Desarrollo	114000.00
Sistemas	129000.00
Sistemas	129000.00
Recursos Humanos	255000.00
Recursos Humanos	255000.00

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

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

nombre
1 Contabilidad
2 Contabilidad
3 Desarrollo
4 Desarrollo
5 I+D
6 I+D
7 Proyectos
8 Proyectos
9 Publicidad
10 Publicidad
11 Recursos Humanos
12 Recursos Humanos
13 Sistemas
14 Sistemas

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

<pre> SELECT nombre FROM departamento ORDER BY nombre DESC; </pre>	
100 %	Results Messages
nombre	
1	Sistemas
2	Sistemas
3	Recursos Humanos
4	Recursos Humanos
5	Publicidad
6	Publicidad
7	Proyectos
8	Proyectos
9	I+D
10	I+D
11	Desarrollo
12	Desarrollo
13	Contabilidad
14	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.

```
SELECT
    apePaterno,
    apeMaterno,
    nombre
FROM empleado
ORDER BY apePaterno ASC, apeMaterno ASC, nombre ASC;
```

%

Results Messages

apePaterno	apeMaterno	nombre
Flores	Salas	Diego
Flores	Salas	Diego
Gómez	López	Juan
Gómez	López	Juan
Herrera	Gil	Marta
Herrera	Gil	Marta
Loyola	Méndez	Marcos
Loyola	Méndez	Marcos
Rivero	Gómez	Aarón
Rivero	Gómez	Aarón
Rubio	Flores	Adolfo
Rubio	Flores	Adolfo
Ruiz	NULL	Pilar
Ruiz	NULL	Pilar
Ruiz	Santana	Pepe
Ruiz	Santana	Pepe
Sáez	Guerrero	Juan Antonio
Sáez	Guerrero	Juan Antonio
Salas	Díaz	Adela
Salas	Díaz	Adela
Salas	Flores	Irene
Salas	Flores	Irene
Santana	Moreno	María
Santana	Moreno	María
Suárez	NULL	Adrián
Suárez	NULL	Adrián

16. 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 DESC
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;
```

nombre	presupuesto
I+D	375000.00
I+D	375000.00
Recursos Humanos	280000.00

17. 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
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;
```

nombre	presupuesto
Proyectos	0.00
Publicidad	0.00
Proyectos	0.00

18. 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
OFFSET 0 ROWS FETCH NEXT 2 ROWS ONLY;
```

nombre	gastos
I+D	380000.00
I+D	380000.00

19. 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
OFFSET 0 ROWS FETCH NEXT 2 ROWS ONLY;
```

%

Results Messages

nombre	gastos
Proyectos	0.00
Proyectos	0.00

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.

```
SELECT *
FROM empleado
ORDER BY codigo
OFFSET 2 ROWS FETCH NEXT 5 ROWS ONLY;
```

00 %

Results Messages

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	3	R6970642B	Adolfo	Rubio	Flores	3
2	4	77705545E	Adrián	Suárez	NULL	4
3	5	17087203C	Marcos	Loyola	Méndez	5
4	6	38382980M	Maria	Santana	Moreno	1
5	7	80576669X	Pilar	Ruiz	NULL	2

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

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

00 %

Results Messages

	nombre	presupuesto
1	Sistemas	150000.00
2	Recursos Humanos	280000.00
3	I+D	375000.00
4	Sistemas	150000.00
5	Recursos Humanos	280000.00
6	I+D	375000.00
7	Sistemas	150000.00
8	Recursos Humanos	280000.00
9	I+D	375000.00

22. 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;
```

	nombre	gastos
1	Contabilidad	3000.00
2	Proyectos	0.00
3	Publicidad	1000.00
4	Contabilidad	3000.00
5	Proyectos	0.00
6	Publicidad	1000.00
7	Contabilidad	3000.00
8	Proyectos	0.00
9	Publicidad	1000.00

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

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

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	Contabilidad	110000.00
4	Desarrollo	120000.00
5	Sistemas	150000.00
6	Contabilidad	110000.00
7	Desarrollo	120000.00
8	Sistemas	150000.00
9	Contabilidad	110000.00

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.

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

	nombre
1	Recursos Humanos
2	I+D
3	Proyectos
4	Publicidad
5	Recursos Humanos
6	I+D
7	Proyectos
8	Publicidad
9	Recursos Humanos
10	I+D
11	Proyectos
12	Publicidad

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

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

100 %

Results Messages

	nombre
1	Desarrollo
2	Sistemas
3	Contabilidad
4	Desarrollo
5	Sistemas
6	Contabilidad
7	Desarrollo
8	Sistemas
9	Contabilidad

26. 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
FROM departamento
WHERE presupuesto NOT BETWEEN 100000 AND 200000;
```

100 %

Results Messages

	nombre
1	Recursos Humanos
2	I+D
3	Proyectos
4	Publicidad
5	Recursos Humanos
6	I+D
7	Proyectos
8	Publicidad
9	Recursos Humanos
10	I+D
11	Proyectos
12	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.

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

100 %

Results Messages

	nombre	gastos	presupuesto
1	I+D	380000.00	375000.00
2	Publicidad	1000.00	0.00
3	I+D	380000.00	375000.00
4	Publicidad	1000.00	0.00
5	I+D	380000.00	375000.00
6	Publicidad	1000.00	0.00
7	I+D	380000.00	375000.00
8	Publicidad	1000.00	0.00

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 disponen.

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

100 %

Results Messages

	nombre	gastos	presupuesto
1	Desarrollo	6000.00	120000.00
2	Sistemas	21000.00	150000.00
3	Recursos Humanos	25000.00	280000.00
4	Contabilidad	3000.00	110000.00
5	Desarrollo	6000.00	120000.00
6	Sistemas	21000.00	150000.00
7	Recursos Humanos	25000.00	280000.00
8	Contabilidad	3000.00	110000.00
9	Desarrollo	6000.00	120000.00
10	Sistemas	21000.00	150000.00
11	Recursos Humanos	25000.00	280000.00
12	Contabilidad	3000.00	110000.00
13	Desarrollo	6000.00	120000.00
14	Sistemas	21000.00	150000.00

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.

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

100 %

Results Messages

	nombre	gastos	presupuesto
1	Proyectos	0.00	0.00
2	Proyectos	0.00	0.00
3	Proyectos	0.00	0.00
4	Proyectos	0.00	0.00

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

```
SELECT *
FROM empleado
WHERE apeMaterno IS NULL;
```

100 %

Results Messages

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	4	77705545E	Adrián	Suárez	NULL	4
2	7	80576669X	Pilar	Ruiz	NULL	2
3	17	77705545	Adrián	Suárez	NULL	4
4	20	80576669	Pilar	Ruiz	NULL	2

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

```
SELECT *
FROM empleado
WHERE apeMaterno IS NOT NULL;
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	1	32481596F	Aarón	Rivero	Gómez	1
2	2	Y5575632D	Adela	Salas	Díaz	2
3	3	R6970642B	Adolfo	Rubio	Flores	3
4	5	17087203C	Marcos	Loyola	Méndez	5
5	6	38382980M	María	Santana	Moreno	1
6	8	71651431Z	Pepe	Ruiz	Santana	3
7	9	56399183D	Juan	Gómez	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
12	14	32481596	Aarón	Rivero	Gómez	1
13	15	Y5575632	Adela	Salas	Díaz	2
14	16	R6970642	Adolfo	Rubio	Flores	3

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

```
SELECT *
FROM empleado
WHERE apeMaterno = 'López';
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	9	56399183D	Juan	Gómez	López	2
2	22	56399183	Juan	Gómez	López	2

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

```
SELECT *
FROM empleado
WHERE apeMaterno = 'Díaz' OR apeMaterno = 'Moreno';
```

% ▾ ◀

Results Messages

codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
2	Y5575632D	Adela	Salas	Díaz	2
6	38382980M	María	Santana	Moreno	1
15	Y5575632	Adela	Salas	Díaz	2
19	38382980	María	Santana	Moreno	1

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

```
SELECT *
FROM empleado
WHERE apeMaterno IN ('Díaz', 'Moreno');
```

100 %

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	2	Y5575632D	Adela	Salas	Díaz	2
2	6	38382980M	Maria	Santana	Moreno	1
3	15	Y5575632	Adela	Salas	Díaz	2
4	19	38382980	Maria	Santana	Moreno	1

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

```
SELECT
    nombre,
    apePaterno,
    apeMaterno,
    dni
FROM empleado
WHERE codigo_departamento = 3;
```

100 %

	nombre	apePaterno	apeMaterno	dni
1	Adolfo	Rubio	Flores	R6970642B
2	Pepe	Ruiz	Santana	71651431Z
3	Adolfo	Rubio	Flores	R6970642
4	Pepe	Ruiz	Santana	71651431

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

```
SELECT
    nombre,
    apePaterno,
    apeMaterno,
    dni
FROM empleado
WHERE codigo_departamento IN (2, 4, 5);
```

100 %

	nombre	apePaterno	apeMaterno	dni
1	Adela	Salas	Díaz	Y5575632D
2	Adrián	Suárez	NULL	77705545E
3	Marcos	Loyola	Méndez	17087203C
4	Pilar	Ruiz	NULL	80576669X
5	Juan	Gómez	López	56399183D
6	Diego	Flores	Salas	46384486H

4.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.

```
SELECT
    e.codigo AS codigo_empleado,
    e.dni,
    e.nombre AS nombre_empleado,
    e.apePaterno,
    e.apeMaterno,
    d.codigo AS codigo_departamento,
    d.nombre AS nombre_departamento,
    d.presupuesto,
    d.gastos
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo;
```

	codigo_empleado	dni	nombre_empleado	apePaterno	apeMaterno	codigo_departamento	nombre_departamento	presupuesto	gastos
1	1	32481596	Aarón	Rivero	Gómez	1	Desarrollo	120000.00	6000.00
2	2	Y5575632	Adela	Salas	Díaz	2	Sistemas	150000.00	21000.00
3	3	R6970642	Adolfo	Rubio	Flores	3	Recursos Humanos	280000.00	25000.00
4	4	77705545	Adrián	Suárez	NULL	4	Contabilidad	110000.00	3000.00
5	5	17087203	Marcos	Loyola	Méndez	5	I+D	375000.00	380000.00
6	6	38382980	María	Santana	Moreno	1	Desarrollo	120000.00	6000.00
7	7	80576669	Pilar	Ruiz	NULL	2	Sistemas	150000.00	21000.00
8	8	71651431	Pepe	Ruiz	Santana	3	Recursos Humanos	280000.00	25000.00
9	9	56399183	Juan	Gómez	López	2	Sistemas	150000.00	21000.00
10	10	46384486	Diego	Flores	Salas	5	I+D	375000.00	380000.00
11	11	67389283	Marta	Herrera	Gil	1	Desarrollo	120000.00	6000.00

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
    e.codigo AS codigo_empleado,
    e.dni,
    e.nombre AS nombre_empleado,
    e.apePaterno,
    e.apeMaterno,
    d.codigo AS codigo_departamento,
    d.nombre AS nombre_departamento,
    d.presupuesto,
    d.gastos
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
ORDER BY
    d.nombre,          -- Ordenar por nombre del departamento
    e.apePaterno,      -- Luego por primer apellido del empleado
    e.apeMaterno,      -- Luego por segundo apellido del empleado
    e.nombre;          -- Finalmente por nombre del empleado
```

	codigo_empleado	dni	nombre_empleado	apePaterno	apeMaterno	codigo_departamento	nombre_departamento	presupuesto	gastos
1	4	77705545	Adrián	Suárez	NULL	4	Contabilidad	110000.00	3000.00
2	11	67389283	Marta	Herrera	Gil	1	Desarrollo	120000.00	6000.00
3	1	32481596	Aarón	Rivero	Gómez	1	Desarrollo	120000.00	6000.00
4	6	38382980	María	Santana	Moreno	1	Desarrollo	120000.00	6000.00
5	10	46384486	Diego	Flores	Salas	5	I+D	375000.00	380000.00
6	5	17087203	Marcos	Loyola	Méndez	5	I+D	375000.00	380000.00
7	3	R6970642	Adolfo	Rubio	Flores	3	Recursos Humanos	280000.00	25000.00
8	8	71651431	Pepe	Ruiz	Santana	3	Recursos Humanos	280000.00	25000.00
9	9	56399183	Juan	Gómez	López	2	Sistemas	150000.00	21000.00
10	7	80576669	Pilar	Ruiz	NULL	2	Sistemas	150000.00	21000.00

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

```
SELECT DISTINCT
    d.codigo AS codigo_departamento,
    d.nombre AS nombre_departamento
FROM departamento d
JOIN empleado e ON d.codigo = e.codigo_departamento;
```

100 %

Results Messages

	codigo_departamento	nombre_departamento
1	1	Desarrollo
2	2	Sistemas
3	3	Recursos Humanos
4	4	Contabilidad
5	5	I+D

4. 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 presupuesta) el valor de los gastos que ha generado (columna gastos).

```
SELECT
    d.codigo AS codigo_departamento,
    d.nombre AS nombre_departamento,
    (d.presupuesto - d.gastos) AS presupuesto_actual
FROM departamento d
JOIN empleado e ON d.codigo = e.codigo_departamento;
```

100 %

Results Messages

	codigo_departamento	nombre_departamento	presupuesto_actual
1	1	Desarrollo	114000.00
2	2	Sistemas	129000.00
3	3	Recursos Humanos	255000.00
4	4	Contabilidad	107000.00
5	5	I+D	-5000.00
6	1	Desarrollo	114000.00
7	2	Sistemas	129000.00
8	3	Recursos Humanos	255000.00
9	2	Sistemas	129000.00
10	5	I+D	-5000.00
11	1	Desarrollo	114000.00

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

```
SELECT
    d.nombre AS nombre_departamento
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE e.dni = '38382980';
```

100 %

Results Messages

	nombre_departamento
1	Desarrollo

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

```
SQLQuery3.sql - D...3GC\USUARIO (66))* SQLQuery1.sql - D...3GC\USUARIO (51))* SQLQuery2.sql
SELECT
    d.nombre AS nombre_departamento
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE e.nombre = 'Pepe' AND e.apePaterno = 'Ruiz' AND e.apeMaterno = 'Santana';
```

100 %

Results Messages

	nombre_departamento
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.

```
SQLQuery3.sql - D...3GC\USUARIO (66))* SQLQuery1.sql - D...3GC\USUARIO (51))
SELECT
    e.codigo AS codigo_empleado,
    e.dni,
    e.nombre,
    e.apePaterno,
    e.apeMaterno,
    e.codigo_departamento
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE d.nombre = 'I+D'
ORDER BY e.nombre, e.apePaterno, e.apeMaterno;
```

100 %

Results Messages

	codigo_empleado	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	10	46384486	Diego	Flores	Salas	5
2	5	17087203	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.

SQLQuery3.sql - D...3GC\USUARIO (66))* SQLQuery1.sql - D...3GC\USUARIO (51))*

```
SELECT
    e.codigo AS codigo_empleado,
    e.dni,
    e.nombre,
    e.apePaterno,
    e.apeMaterno,
    e.codigo_departamento
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE d.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
ORDER BY e.nombre, e.apePaterno, e.apeMaterno;
```

100 %

Results Messages

	codigo_empleado	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	2	Y5575632	Adela	Salas	Díaz	2
2	4	77705545	Adrián	Suárez	NULL	4
3	10	46384486	Diego	Flores	Salas	5
4	9	56399183	Juan	Gómez	López	2
5	5	17087203	Marcos	Loyola	Méndez	5
6	7	80576669	Pilar	Ruiz	NULL	2

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

SQLQuery3.sql - D...3GC\USUARIO (66))* SQLQuery1.sql - D...3GC\USUARIO (51))*

```
SELECT
    e.nombre AS nombre_empleado
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE d.presupuesto < 100000 OR d.presupuesto > 200000;
```

100 %

Results Messages

	nombre_empleado
1	Adolfo
2	Marcos
3	Pepe
4	Diego

- 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
    d.nombre AS nombre_departamento
FROM departamento d
JOIN empleado e ON d.codigo = e.codigo_departamento
WHERE e.apeMaterno IS NULL;

```

100 %

Results Messages

	nombre_departamento
1	Contabilidad
2	Sistemas

5.-Consultas multitabla (Composición externa)

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

- 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
    e.codigo AS codigo_empleado,
    e.dni,
    e.nombre,
    e.apePaterno,
    e.apeMaterno,
    d.codigo AS codigo_departamento,
    d.nombre AS nombre_departamento,
    d.presupuesto,
    d.gastos
FROM empleado e
LEFT JOIN departamento d ON e.codigo_departamento = d.codigo;

```

100 %

Results Messages

	codigo_empleado	dni	nombre	apePaterno	apeMaterno	codigo_departamento	nombre_departamento	presupuesto	gastos
1	1	32481596	Aarón	Rivero	Gómez	1	Desarrollo	120000.00	6000.00
2	2	Y5575632	Adela	Salas	Díaz	2	Sistemas	150000.00	21000.00
3	3	R6970642	Adolfo	Rubio	Flores	3	Recursos Humanos	280000.00	25000.00
4	4	77705545	Adrián	Suárez	NULL	4	Contabilidad	110000.00	3000.00
5	5	17087203	Marcos	Loyola	Méndez	5	I+D	375000.00	380000.00
6	6	38382980	María	Santana	Moreno	1	Desarrollo	120000.00	6000.00
7	7	80576669	Pilar	Ruiz	NULL	2	Sistemas	150000.00	21000.00
8	8	71651431	Pepe	Ruiz	Santana	3	Recursos Humanos	280000.00	25000.00
9	9	56399183	Juan	Gómez	López	2	Sistemas	150000.00	21000.00
10	10	46384486	Diego	Flores	Salas	5	I+D	375000.00	380000.00
11	11	67389283	Marta	Herrera	Gil	1	Desarrollo	120000.00	6000.00
12	12	41234836	Irene	Salas	Flores	NULL	NULL	NULL	NULL
13	13	82635162	Juan Antonio	Sáez	Guerrero	NULL	NULL	NULL	NULL

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

```
SELECT
    e.codigo AS codigo_Empleado,
    e.dni,
    e.nombre,
    e.apePaterno,
    e.apeMaterno
FROM empleado e
WHERE e.codigo_departamento IS NULL;
```

100 %

Results Messages

	codigo_Empleado	dni	nombre	apePaterno	apeMaterno
1	12	41234836	Irene	Salas	Flores
2	13	82635162	Juan Antonio	Sáez	Guerrero

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

```
SELECT
    d.codigo AS codigo_departamento,
    d.nombre AS nombre_departamento,
    d.presupuesto,
    d.gastos
FROM departamento d
LEFT JOIN empleado e ON d.codigo = e.codigo_departamento
WHERE e.codigo_departamento IS NULL;
```

100 %

Results Messages

	codigo_departamento	nombre_departamento	presupuesto	gastos
1	6	Proyectos	0.00	0.00
2	7	Publicidad	0.00	1000.00
3	8	Desarrollo	120000.00	6000.00
4	9	Sistemas	150000.00	21000.00
5	10	Recursos Humanos	280000.00	25000.00
6	11	Contabilidad	110000.00	3000.00
7	12	I+D	375000.00	380000.00
8	13	Proyectos	0.00	0.00
9	14	Publicidad	0.00	1000.00
10	15	Desarrollo	120000.00	6000.00
11	16	Sistemas	150000.00	21000.00
12	17	Recursos Humanos	280000.00	25000.00
13	18	Contabilidad	110000.00	3000.00
14	19	I+D	375000.00	380000.00
15	20	Proyectos	0.00	0.00
16	21	Publicidad	0.00	1000.00
17	22	Desarrollo	120000.00	6000.00
18	23	Sistemas	150000.00	21000.00

- ```

SELECT
 e.codigo AS codigo_empleado,
 e.dni,
 e.nombre,
 e.apePaterno,
 e.apeMaterno,
 d.codigo AS codigo_departamento,
 d.nombre AS nombre_departamento,
 d.presupuesto,
 d.gastos
FROM empleado e
FULL OUTER JOIN departamento d ON e.codigo_departamento = d.codigo
ORDER BY COALESCE(d.nombre, e.nombre);

```
- 100 %
- Results Messages
- |    | codigo_empleado | dni      | nombre | apePaterno | apeMaterno | codigo_departamento | nombre_departamento | presupuesto | gastos    |
|----|-----------------|----------|--------|------------|------------|---------------------|---------------------|-------------|-----------|
| 1  | 4               | 77705545 | Adrián | Suárez     | NULL       | 4                   | Contabilidad        | 110000.00   | 3000.00   |
| 2  | NULL            | NULL     | NULL   | NULL       | NULL       | 11                  | Contabilidad        | 110000.00   | 3000.00   |
| 3  | NULL            | NULL     | NULL   | NULL       | NULL       | 18                  | Contabilidad        | 110000.00   | 3000.00   |
| 4  | NULL            | NULL     | NULL   | NULL       | NULL       | 25                  | Contabilidad        | 110000.00   | 3000.00   |
| 5  | NULL            | NULL     | NULL   | NULL       | NULL       | 32                  | Contabilidad        | 110000.00   | 3000.00   |
| 6  | NULL            | NULL     | NULL   | NULL       | NULL       | 39                  | Contabilidad        | 110000.00   | 3000.00   |
| 7  | NULL            | NULL     | NULL   | NULL       | NULL       | 36                  | Desarrollo          | 120000.00   | 6000.00   |
| 8  | NULL            | NULL     | NULL   | NULL       | NULL       | 22                  | Desarrollo          | 120000.00   | 6000.00   |
| 9  | NULL            | NULL     | NULL   | NULL       | NULL       | 29                  | Desarrollo          | 120000.00   | 6000.00   |
| 10 | NULL            | NULL     | NULL   | NULL       | NULL       | 15                  | Desarrollo          | 120000.00   | 6000.00   |
| 11 | 1               | 32481596 | Aarón  | Rivero     | Gómez      | 1                   | Desarrollo          | 120000.00   | 6000.00   |
| 12 | 6               | 38382980 | María  | Santana    | Moreno     | 1                   | Desarrollo          | 120000.00   | 6000.00   |
| 13 | 11              | 67389283 | Marta  | Herrera    | Gil        | 1                   | Desarrollo          | 120000.00   | 6000.00   |
| 14 | NULL            | NULL     | NULL   | NULL       | NULL       | 8                   | Desarrollo          | 120000.00   | 6000.00   |
| 15 | 10              | 46384486 | Diego  | Flores     | Salas      | 5                   | I+D                 | 375000.00   | 380000.00 |
- Query executed successfully.
- DESKTOP-MOTM3GC (16.0 RTM) DESKTOP-MOTM3GC (16.0 RTM)

- ```

SQLQuery3.sql - D...3GC\USUARIO (66))*
SQLQuery1.sql - D...3GC\USUARIO (51))*
SQLQuery2.sql

SELECT
    e.codigo AS codigo_empleado,
    e.dni,
    e.nombre AS nombre_empleado,
    e.apePaterno,
    e.apeMaterno,
    d.codigo AS codigo_departamento,
    d.nombre AS nombre_departamento
FROM empleado e
FULL OUTER JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE e.codigo_departamento IS NULL OR d.codigo IS NULL
ORDER BY COALESCE(d.nombre, e.nombre);

```


6.- Consultas resumen

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

```
SELECT SUM(presupuesto) AS suma_presupuesto_total
FROM departamento;
```

100 %

Results Messages

	suma_presupuesto_total
1	6210000.00

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

```
SELECT AVG(presupuesto) AS media_presupuesto_total
FROM departamento;
```

100 %

Results Messages

	media_presupuesto_total
1	147857.142857

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

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

100 %

Results Messages

	presupuesto_minimo
1	0.00

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

```
SELECT TOP 1 nombre, presupuesto
FROM departamento
ORDER BY presupuesto ASC;
```

100 %

Results Messages

	nombre	presupuesto
1	Proyectos	0.00

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

SQLQuery3.sql - D...3GC\USUARIO (66))* SQLQuery1.sql - D.

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

100 %

Results Messages

	presupuesto_maximo
1	375000.00

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

```
SELECT TOP 1 nombre, presupuesto
FROM departamento
ORDER BY presupuesto DESC;
```

100 %

Results Messages

	nombre	presupuesto
1	I+D	375000.00

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

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

100 %

Results Messages

	total_empleados
1	13

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

```
SELECT COUNT(*) AS empleados_con_segundo_apellido
FROM empleado
WHERE apeMaterno IS NOT NULL;
```

100 %

Results Messages

	empleados_con_segundo_apellido
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.

```
SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.codigo = e.codigo_departamento
GROUP BY d.nombre;
```

100 %

Results Messages

	nombre_departamento	numero_empleados
1	Contabilidad	1
2	Desarrollo	3
3	I+D	2
4	Proyectos	0
5	Publicidad	0
6	Recursos Humanos	2
7	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.

```
SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.codigo = e.codigo_departamento
GROUP BY d.nombre
HAVING COUNT(e.codigo) > 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.

```
SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.codigo = e.codigo_departamento
GROUP BY d.nombre;
```

100 %

Results Messages

	nombre_departamento	numero_empleados
1	Contabilidad	1
2	Desarrollo	3
3	I+D	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.

```
SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
FROM departamento d
JOIN empleado e ON d.codigo = e.codigo_departamento
WHERE d.presupuesto > 200000
GROUP BY d.nombre;
```

100 %

Results Messages

	nombre_departamento	numero_empleados
1	I+D	2
2	Recursos Humanos	2

7.- Subconsultas

7.1 Con operadores básicos de comparación:

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

```
SELECT *
FROM empleado
WHERE codigo_departamento = (SELECT TOP 1 codigo FROM departamento WHERE nombre = 'Sistemas')
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	2	Y5575632	Adela	Salas	Díaz	2
2	7	80576669	Pilar	Ruiz	NULL	2
3	9	56399183	Juan	Gómez	López	2

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

```
SELECT *
FROM empleado
WHERE codigo_departamento IN (SELECT codigo FROM departamento WHERE nombre = 'Sistemas');
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	2	Y5575632	Adela	Salas	Díaz	2
2	7	80576669	Pilar	Ruiz	NULL	2
3	9	56399183	Juan	Gómez	López	2

3. Devuelve el nombre del departamento con menor presupuesto y la cantidad que tiene asignada.

```
SELECT TOP 1 nombre, presupuesto
FROM departamento
ORDER BY presupuesto ASC;
```

	nombre	presupuesto
1	Proyectos	0.00

7.2 Subconsultas con ALL Y ANY:

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

```
SELECT nombre, presupuesto
FROM departamento d1
WHERE presupuesto >= ALL (SELECT presupuesto FROM departamento);
```

	nombre	presupuesto
1	I+D	375000.00
2	I+D	375000.00
3	I+D	375000.00
4	I+D	375000.00
5	I+D	375000.00
6	I+D	375000.00

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

```
SELECT nombre, presupuesto
FROM departamento d1
WHERE presupuesto <= ALL (SELECT presupuesto FROM departamento);
```

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00
3	Proyectos	0.00
4	Publicidad	0.00
5	Proyectos	0.00
6	Publicidad	0.00
7	Proyectos	0.00
8	Publicidad	0.00
9	Proyectos	0.00
10	Publicidad	0.00
11	Proyectos	0.00
12	Publicidad	0.00

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

```
SELECT nombre
FROM departamento d
WHERE codigo = ANY (SELECT codigo_departamento FROM empleado WHERE codigo_departamento IS NOT NULL);
```

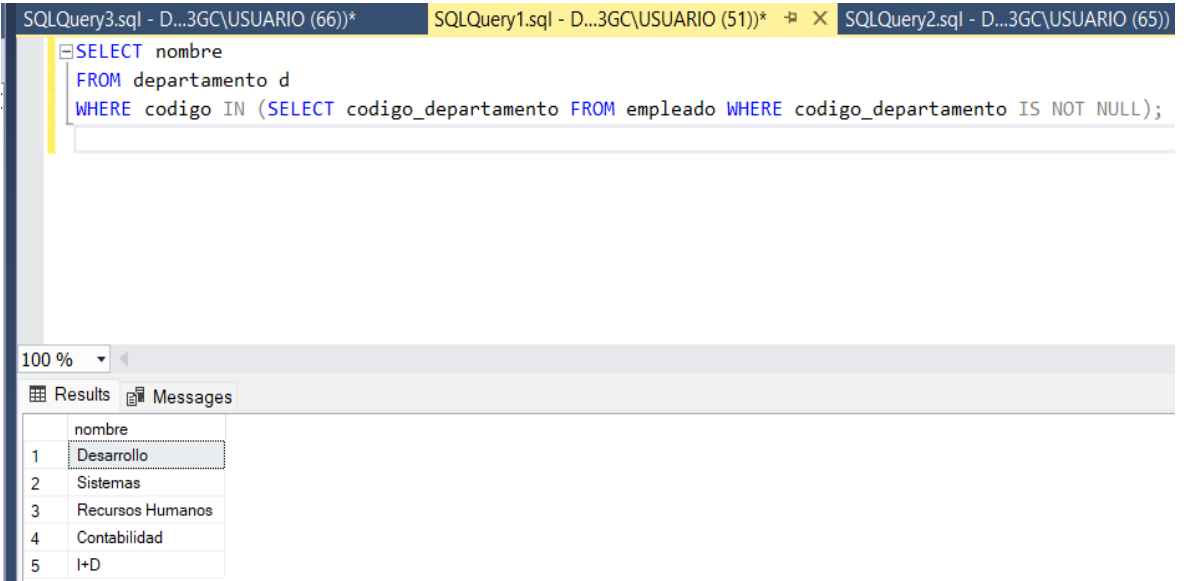
	nombre
1	Desarrollo
2	Sistemas
3	Recursos Humanos
4	Contabilidad
5	I+D

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

```
SELECT nombre
FROM departamento d
WHERE codigo <> ALL (SELECT codigo_departamento FROM empleado);
```

7.3 Subconsultas con IN y NOT IN

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



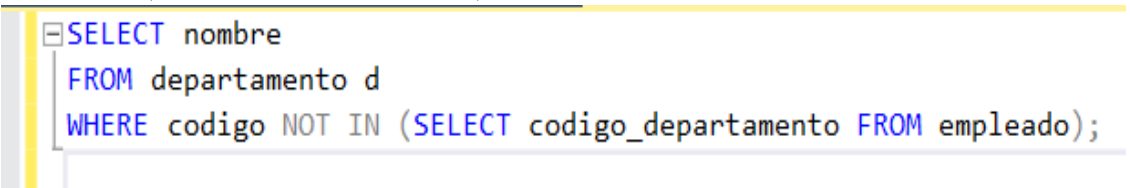
The screenshot shows the SQL Server Enterprise Manager interface. At the top, there are three tabs: 'SQLQuery3.sql - D...3GC\USUARIO (66))*', 'SQLQuery1.sql - D...3GC\USUARIO (51))*', and 'SQLQuery2.sql - D...3GC\USUARIO (65))'. The active tab is 'SQLQuery3.sql'. The query editor shows the following SQL code:

```
SELECT nombre
FROM departamento d
WHERE codigo IN (SELECT codigo_departamento FROM empleado WHERE codigo_departamento IS NOT NULL);
```

Below the query editor, there is a 'Results' tab and a 'Messages' tab. The 'Results' tab is active, showing a table with the following data:

	nombre
1	Desarrollo
2	Sistemas
3	Recursos Humanos
4	Contabilidad
5	I+D

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

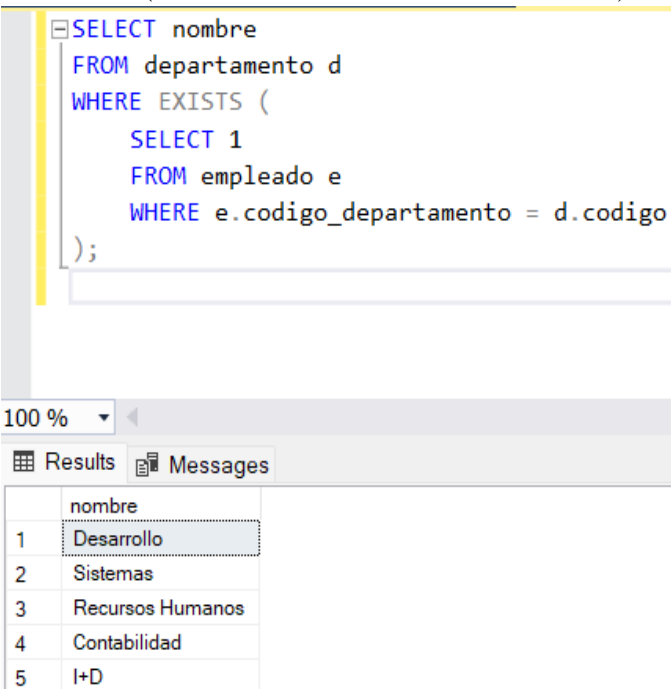


The screenshot shows the SQL Server Enterprise Manager interface. The query editor shows the following SQL code:

```
SELECT nombre
FROM departamento d
WHERE codigo NOT IN (SELECT codigo_departamento FROM empleado);
```

7.4 Subconsultas con EXISTS Y NOT EXISTS

10. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando EXISTS ONOT EXISTS).



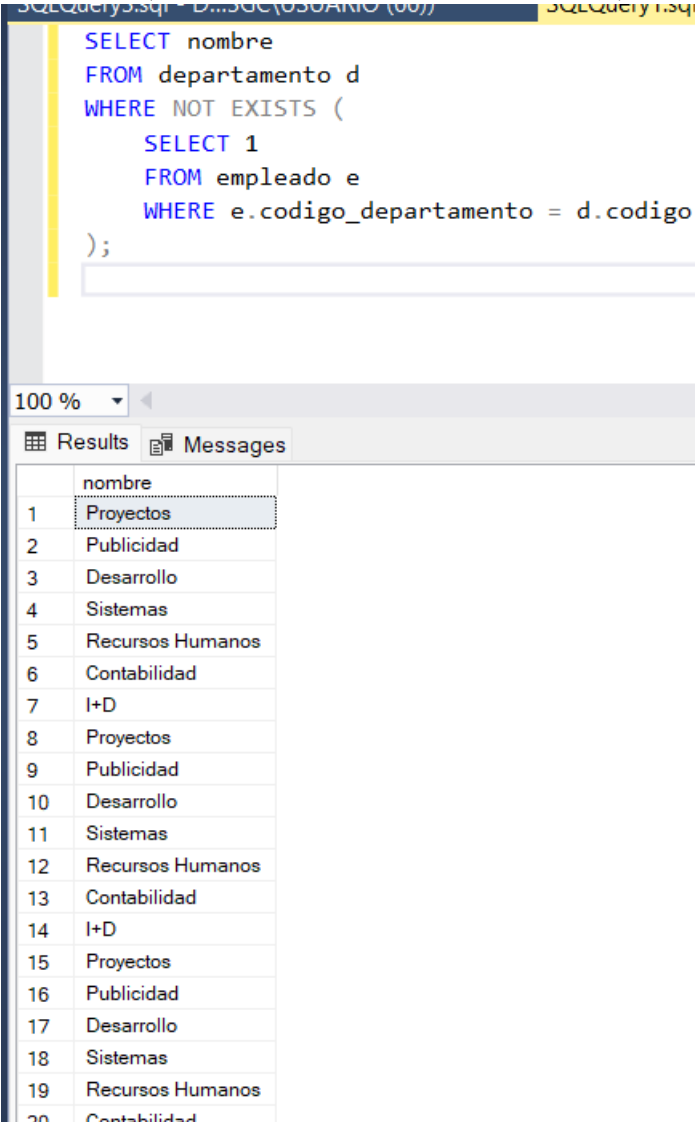
The screenshot shows the SQL Server Enterprise Manager interface. The query editor shows the following SQL code:

```
SELECT nombre
FROM departamento d
WHERE EXISTS (
    SELECT 1
    FROM empleado e
    WHERE e.codigo_departamento = d.codigo
);
```

Below the query editor, there is a 'Results' tab and a 'Messages' tab. The 'Results' tab is active, showing a table with the following data:

	nombre
1	Desarrollo
2	Sistemas
3	Recursos Humanos
4	Contabilidad
5	I+D

11. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando EXISTS ONOT EXISTS).



The screenshot shows a SQL query window with the following code:

```
SELECT nombre
FROM departamento d
WHERE NOT EXISTS (
    SELECT 1
    FROM empleado e
    WHERE e.codigo_departamento = d.codigo
);
```

Below the query window, the 'Results' tab is active, displaying a table with 20 rows and 1 column named 'nombre'. The results are as follows:

	nombre
1	Proyectos
2	Publicidad
3	Desarrollo
4	Sistemas
5	Recursos Humanos
6	Contabilidad
7	I+D
8	Proyectos
9	Publicidad
10	Desarrollo
11	Sistemas
12	Recursos Humanos
13	Contabilidad
14	I+D
15	Proyectos
16	Publicidad
17	Desarrollo
18	Sistemas
19	Recursos Humanos
20	Contabilidad