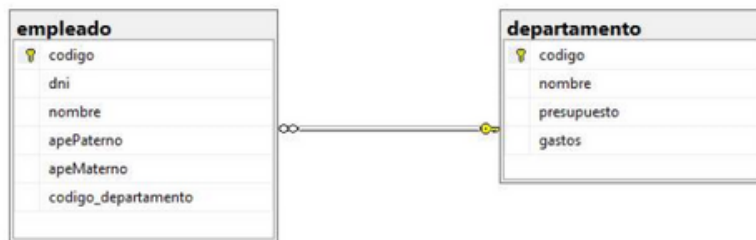


INFORME DE LABORATORIO

Desarrollo y Ejecución de Consultas SQL para la Gestión de Bases de Datos

DB Gestionempleados

Gestión de empleados



Procedimiento

```
SQLQuery2.sql - DE...Brayan Vargas (69))* -> X
USE master;
GO
CREATE DATABASE GestionEmpleado
ON PRIMARY
(
    NAME = GestionEmpleado_Data,
    FILENAME = 'C:\BaseDatos2024\GestionEmpleado_Data.mdf',
    SIZE = 5MB,
    MAXSIZE = 20MB,
    FILEGROWTH = 5MB
)
LOG ON
(
    NAME = GestionEmpleado_Log,
    FILENAME = 'C:\BaseDatos2024\GestionEmpleado_Log.ldf',
    SIZE = 1MB,
    MAXSIZE = 4MB,
    FILEGROWTH = 1MB
);
GO

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
```

1. Lista el primer apellido de todos los empleados.

```
SQLQuery2.sql - DE...Brayan Vargas (69))*  
SELECT apePaterno FROM empleado;
```

Conclusión

	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

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT DISTINCT apePaterno FROM empleado;
```

Conclusión

	apePaterno
1	Flores
2	Gómez
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.

SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ✕
`SELECT * FROM empleado;`

Conclusión

Results

Messages

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_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	38382980M	Maria	Santana	Moreno	1
7	7	88576669X	Pilar	Ruiz	NULL	2
8	8	71651431Z	Pepe	Ruiz	Santana	3
9	9	56399183D	Juan	Gómez	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	826351628	JuanAntonio	Sáez	Guerrero	NULL

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

SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ✕
`SELECT nombre, apePaterno, apeMaterno FROM empleado;`

Conclusión

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	Maria	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	JuanAntonio	Sáez	Guerrero

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ✕  
SELECT codigo_departamento FROM empleado;
```

Conclusión

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ✕  
SELECT DISTINCT codigo_departamento FROM empleado;
```

Conclusión

	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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* X  
SELECT CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno) AS nombre_completo FROM empleado;
```

Conclusión

	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 Gómez López
10	Diego Flores Salas
11	Marta Herrera Gil
12	Irene Salas Flores
13	JuanAntonio Sáez Guerrero

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* X  
SELECT UPPER(CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno)) AS nombre_completo_mayuscula FROM empleado;
```

Conclusión

	nombre_completo_mayuscula
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 GÓMEZ LÓPEZ
10	DIEGO FLORES SALAS
11	MARTA HERRERA GIL
12	IRENE SALAS FLORES
13	JUANANTONIO SÁEZ GUERRERO

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ×  
SELECT LOWER(CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno)) AS nombre_completo_minuscula FROM empleado;
```

Conclusión

	Results	Messages
	nombre_completo_minuscula	
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 gómez lópez	
10	diego flores salas	
11	marta herrera gil	
12	irene salas flores	
13	juanantonio sáez guemero	

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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ×  
SELECT codigo, LEFT(dni, LEN(dni)-1) AS nif_digitos, RIGHT(dni, 1) AS nif_letra FROM empleado;
```

Conclusión

Results		Messages	
	codigo	nif_digitos	nif_letra
1	5	17087203	0
2	1	32481596	F
3	6	38382980	M
4	12	41234836	R
5	10	46384486	H
6	9	56399183	D
7	11	67389283	A
8	8	71651431	Z
9	4	77705545	E
10	13	82635162	8
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. Restar al presupuesto los gastos.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* -p X  
SELECT nombre, (presupuesto - gastos) AS presupuesto_actual FROM departamento;
```

Conclusión

	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

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* -p X  
SELECT nombre, (presupuesto - gastos) AS presupuesto_actual FROM departamento ORDER BY presupuesto_actual ASC;
```

Conclusión

	nombre	presupuesto_actual
1	I+D	-5000.00
2	Publicidad	-1000.00
3	Proyectos	0.00
4	Contabilidad	107000.00
5	Desarrollo	114000.00
6	Sistemas	129000.00
7	Recursos Humanos	255000.00

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* X  
SELECT nombre FROM departamento ORDER BY nombre ASC;
```

Conclusión

	nombre
1	Contabilidad
2	Desarrollo
3	I+D
4	Proyectos
5	Publicidad
6	Recursos Humanos
7	Sistemas

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* X  
SELECT nombre FROM departamento ORDER BY nombre DESC;
```

Conclusión

	nombre
1	Sistemas
2	Recursos Humanos
3	Publicidad
4	Proyectos
5	I+D
6	Desarrollo
7	Contabilidad

15. Lista los apellidos y el nombre de todos los empleados, ordenados de forma alfabética teniendo en cuenta primero los apellidos y luego el nombre.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT apePaterno, apeMaterno, nombre FROM empleado ORDER BY apePaterno, apeMaterno, nombre;
```

Conclusión

	apePaterno	apeMaterno	nombre
1	Flores	Salas	Diego
2	Gómez	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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT nombre, presupuesto  
FROM departamento  
ORDER BY presupuesto DESC  
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;
```

Conclusión

	nombre	presupuesto
1	I+D	375000.00
2	Recursos Humanos	280000.00
3	Sistemas	150000.00

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* -# X  
SELECT nombre, presupuesto  
FROM departamento  
ORDER BY presupuesto ASC  
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;
```

Conclusión

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00
3	Contabilidad	110000.00

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* -# X  
SELECT TOP 2 nombre, gastos  
FROM departamento  
ORDER BY gastos DESC;
```

Conclusión

	nombre	gastos
1	I+D	380000.00
2	Recursos Humanos	25000.00

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT TOP 2 nombre, gastos  
FROM departamento  
ORDER BY gastos ASC;
```

Conclusión

	nombre	gastos
1	Proyectos	0.00
2	Publicidad	1000.00

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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT * FROM empleado ORDER BY codigo OFFSET 2 ROWS FETCH NEXT 5 ROWS ONLY;
```

Conclusión

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	3	R69706428	Adolfo	Rubio	Flores	3
2	4	77705545E	Adrián	Suárez	NULL	4
3	5	170872030	Marcos	Loyola	Méndez	5
4	6	38382980M	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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* -> X  
SELECT nombre, presupuesto FROM departamento WHERE presupuesto >= 150000;
```

Conclusión

	nombre	presupuesto
1	Sistemas	150000.00
2	Recursos Humanos	280000.00
3	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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* -> X  
SELECT nombre, gastos FROM departamento WHERE gastos < 5000;
```

Conclusión

	nombre	gastos
1	Contabilidad	3000.00
2	Proyectos	0.00
3	Publicidad	1000.00

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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* -> X  
SELECT nombre, presupuesto FROM departamento WHERE presupuesto >= 100000 AND presupuesto <= 200000;
```

Conclusión

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ✕  
SELECT nombre, presupuesto FROM departamento WHERE presupuesto < 100000 OR presupuesto > 200000;
```

Conclusión

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	Contabilidad	110000.00

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ✕  
SELECT nombre, presupuesto FROM departamento WHERE presupuesto BETWEEN 100000 AND 200000;
```

Conclusión

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	Contabilidad	110000.00

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))* ✕  
SELECT nombre, presupuesto FROM departamento WHERE presupuesto NOT BETWEEN 100000 AND 200000;
```

Conclusión

	nombre	presupuesto
1	Recursos Humanos	280000.00
2	I+D	375000.00
3	Proyectos	0.00
4	Publicidad	0.00

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.

```
SQLQuery2.sql - DE...Brayan Vargas (69)))* SQLQuery3.sql - DE...Brayan Vargas (57)))*
SELECT nombre, gastos, presupuesto
FROM departamento
WHERE gastos > presupuesto;
```

Conclusión

	nombre	gastos	presupuesto
1	I+D	380000.00	375000.00
2	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.

```
SQLQuery2.sql - DE...Brayan Vargas (69)))* SQLQuery3.sql - DE...Brayan Vargas (57)))*
SELECT nombre, gastos, presupuesto
FROM departamento
WHERE gastos < presupuesto;
```

Conclusión

	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

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

```
SQLQuery2.sql - DE...Brayan Vargas (69)))* SQLQuery3.sql - DE...Brayan Vargas (57)))*
SELECT nombre, gastos, presupuesto
FROM departamento
WHERE gastos = presupuesto;
```

Conclusión

	nombre	gastos	presupuesto
1	Proyectos	0.00	0.00

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

```
SQLQuery2.sql - DE...Brayan Vargas (69)))* SQLQuery3.sql - DE...Brayan Vargas (57)))*
SELECT *
FROM empleado
WHERE apeMaterno IS NULL;
```

Conclusión

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	4	77705545E	Adrián	Suárez	NULL	4
2	7	88576669X	Pilar	Ruiz	NULL	2

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*
SELECT e.*, d.*
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo;
```

Conclusión

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento	codigo	nombre	presupuesto	gastos
1	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000.00	6000.00
2	2	Y55756320	Adela	Salas	Díaz	2	2	Sistemas	150000.00	21000.00
3	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000.00	25000.00
4	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000.00	3000.00
5	5	170872030	Marcos	Loyola	Méndez	5	5	I+D	375000.00	380000.00
6	6	38382980M	Maria	Santana	Moreno	1	1	Desarrollo	120000.00	6000.00
7	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000.00	21000.00
8	8	71651431Z	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000.00	25000.00
9	9	56399183D	Juan	Gómez	López	2	2	Sistemas	150000.00	21000.00
10	10	46384486H	Diego	Flores	Salas	5	5	I+D	375000.00	380000.00
11	11	67389283A	Marta	Herrera	Gil	1	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 y en segundo lugar por los apellidos y el nombre de los empleados.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*
SELECT e.*, d.*
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
ORDER BY d.nombre, e.apePaterno, e.apeMaterno, e.nombre;
```

Conclusión

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento	codigo	nombre	presupuesto	gastos
1	4	77705545E	Adrián	Suárez	NULL	4	4	Contabilidad	110000.00	3000.00
2	11	67389283A	Marta	Herrera	Gil	1	1	Desarrollo	120000.00	6000.00
3	1	32481596F	Aarón	Rivero	Gómez	1	1	Desarrollo	120000.00	6000.00
4	6	38382980M	Maria	Santana	Moreno	1	1	Desarrollo	120000.00	6000.00
5	10	46384486H	Diego	Flores	Salas	5	5	I+D	375000.00	380000.00
6	5	170872030	Marcos	Loyola	Méndez	5	5	I+D	375000.00	380000.00
7	3	R69706428	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000.00	25000.00
8	8	71651431Z	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000.00	25000.00
9	9	56399183D	Juan	Gómez	López	2	2	Sistemas	150000.00	21000.00
10	7	88576669X	Pilar	Ruiz	NULL	2	2	Sistemas	150000.00	21000.00
11	2	Y55756320	Adela	Salas	Díaz	2	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.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT DISTINCT d.codigo, d.nombre  
FROM empleado e  
JOIN departamento d ON e.codigo_departamento = d.codigo;
```

Conclusión

	codigo	nombre
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, solamente de aquellos departamentos que tienen empleados.

Conclusión

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

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT d.nombre  
FROM empleado e  
JOIN departamento d ON e.codigo_departamento = d.codigo  
WHERE e.nombre = 'Pepe' AND e.apePaterno = 'Ruiz' AND e.apeMaterno = 'Santana';
```

Conclusión

	nombre
1	Recursos Humanos

7. Devuelve un listado con los datos de los empleados que trabajan en el departamento 10. Ordena el resultado alfabéticamente.

```
SQLQuery2.sql - DE...Brayan Vargas (69))* SQLQuery3.sql - DE...Brayan Vargas (57))*  
SELECT *  
FROM empleado  
WHERE codigo_departamento = 10  
ORDER BY apePaterno, apeMaterno, nombre;
```

Conclusión

Results					
Messages					
Client Statistics					
codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento