

CONSULTAS SQL

CREACIÓN DE ARCHIVOS

```
use master
go
create database GestionEmpleado
on primary
(
    name = GestionEmpleado_Data,
    filename = 'C:\BD2024\GestionEmpleado_Data.mdf',
    size = 5,
    maxsize = 20,
    filegrowth = 5
)
log on
(
    name = GestionEmpleado_Log,
    filename = 'C:\BD2024\GestionEmpleado_Log.ldf',
    size = 1,
    maxsize = 4,
    filegrowth = 1
)
go
```

CREACIÓN DE TABLAS

```
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
```

CARGA DE DATOS

```
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('32481596F', 'Aarón', 'Rivero', 'Gómez', 1);
INSERT INTO empleado VALUES('Y5575632D', 'Adela', 'Salas', 'Díaz', 2);
INSERT INTO empleado VALUES('R6970642A', 'Adolfo', 'Rubio', 'López', 3);
INSERT INTO empleado VALUES('7778545E', 'Adrián', 'Suárez', NULL, 4);
INSERT INTO empleado VALUES('3883298M', 'Marcos', 'Loyola', 'Méndez', 5);
INSERT INTO empleado VALUES('38382980M', 'María', 'Santana', 'Moreno', 1);
INSERT INTO empleado VALUES('8057669K', 'Pilar', 'Ruiz', NULL, 2);
INSERT INTO empleado VALUES('71651431Z', 'Pepe', 'Ruiz', 'Santana', 3);
INSERT INTO empleado VALUES('5633914B', 'Juan', 'Gómez', 'López', 2);
INSERT INTO empleado VALUES('5693459C', 'Diego', 'Flores', 'Salas', 5);
INSERT INTO empleado VALUES('6738894N', 'Marta', 'Herrera', 'Gil', 1);
INSERT INTO empleado VALUES('4123483R', 'Irene', 'Salas', 'Flores', NULL);
INSERT INTO empleado VALUES('82635162B', 'JuanAntonio', 'Sáez', 'Guerrero', NULL);
```

CONSULTAS:

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

```
SELECT apePaterno
FROM empleado;
```

	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 estén repetidos

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

	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
```

```
SELECT *  
FROM empleado;
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	1	32481596F	Aarón	Rivero	Gómez	1
2	2	Y5575632D	Adela	Salas	Díaz	2
3	3	R6970642A	Adolfo	Rubio	López	3
4	4	7778545E	Adrián	Suárez	NULL	4
5	5	3883298M	Marcos	Loyola	Méndez	5
6	6	38382980M	María	Santana	Moreno	1
7	7	8057669K	Pilar	Ruiz	NULL	2
8	8	71651431Z	Pepe	Ruiz	Santana	3
9	9	5633914B	Juan	Gómez	López	2
10	10	5693459C	Diego	Flores	Salas	5
11	11	6738894N	Marta	Herrera	Gil	1
12	12	4123483R	Irene	Salas	Flores	NULL
13	13	82635162B	JuanAntonio	Sáez	Guerrero	NULL

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

```
SELECT nombre, apePaterno, apeMaterno  
FROM empleado;
```

	nombre	apePaterno	apeMaterno
1	Aarón	Rivero	Gómez
2	Adela	Salas	Díaz
3	Adolfo	Rubio	López
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	JuanAntonio	Sáez	Guerrero

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

```
SELECT codigo_departamento  
FROM empleado;
```

	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
```

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

	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 CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno)
      AS NombreCompleto
FROM empleado;
```

	NombreCompleto
1	Aarón Rivero Gómez
2	Adela Salas Díaz
3	Adolfo Rubio López
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	JuanAntonio Sáez Guerrero

```
-- 8. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula
SELECT UPPER(CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno))
      AS NombreCompleto
FROM empleado;
```

	NombreCompleto
1	AARÓN RIVERO GÓMEZ
2	ADELA SALAS DÍAZ
3	ADOLFO RUBIO LÓPEZ
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	JUANANTONIO SÁEZ GUERRERO

```
-- 9. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en minúscula
SELECT LOWER(CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno))
      AS NombreCompleto
FROM empleado;
```

	NombreCompleto
1	aarón rivero gómez
2	adela salas díaz
3	adolfo rubio lópez
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	juanantonio 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, LEFT(dni, LEN(dni)-1)
    AS DigitosNIF, RIGHT(dni, 1)
    AS LetraNIF
FROM empleado;
```

	codigo	DigitosNIF	LetraNIF
1	1	32481596	F
2	6	38382980	M
3	5	3883298	M
4	12	4123483	R
5	9	5633914	B
6	10	5693459	C
7	11	6738894	N
8	8	71651431	Z
9	4	7778545	E
10	7	8057669	K
11	13	82635162	B
12	3	R6970642	A
13	2	Y5575632	D

```
-- 11. Lista el nombre de cada departamento y el valor del presupuesto actual del que dispone
SELECT nombre, (presupuesto - gastos)
    AS PresupuestoActual
FROM departamento;
```

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

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

	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

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

	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

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

	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	López	Adolfo
7	Ruiz	NULL	Pilar
8	Ruiz	Santana	Pepe

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

```
SELECT TOP 3 nombre, presupuesto
FROM departamento
ORDER BY presupuesto DESC;
```

	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
SELECT TOP 3 nombre, presupuesto
FROM departamento
ORDER BY presupuesto ASC;
```

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00
3	Contabili...	110000.00

```
-- 18. Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen mayor gasto
SELECT TOP 2 nombre, gastos
FROM departamento
ORDER BY gastos DESC;
```

	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
SELECT TOP 2 nombre, gastos
FROM departamento
ORDER BY gastos ASC;
```

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

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	3	R6970642A	Adolfo	Rubio	López	3
2	4	7778545E	Adrián	Suárez	NULL	4
3	5	3883298M	Marcos	Loyola	Méndez	5
4	6	38382980M	María	Santana	Moreno	1
5	7	8057669K	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
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto >= 150000;
```

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

	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

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

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	Contabili...	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, presupuesto
FROM departamento
WHERE presupuesto < 100000 OR presupuesto > 200000;
```

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

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

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	Contabili...	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

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

	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.

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

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

	nombre	gastos	presupuesto
1	Desarrollo	6000.00	120000.00
2	Sistemas	21000.00	150000.00
3	Recurso...	25000.00	280000.00
4	Contabili...	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
FROM departamento
WHERE gastos = presupuesto;
```

	nombre	gastos	presupuesto
1	Proyectos	0.00	0.00

```
-- 30. Lista todos los datos de los empleados cuyo segundo apellido (apeMaterno) sea NULL.
```

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

	codigo	dni	nombre	apePatemo	apeMaterno	codigo_departamento
1	4	7778545E	Adrián	Suárez	NULL	4
2	7	8057669K	Pilar	Ruiz	NULL	2

```
-- 31. Lista todos los datos de los empleados cuyo segundo apellido (apeMaterno) no sea NULL.
```

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

	codigo	dni	nombre	apePatemo	apeMaterno	codigo_departamento
1	1	32481596F	Aarón	Rivero	Gómez	1
2	2	Y5575632D	Adela	Salas	Díaz	2
3	3	R6970642A	Adolfo	Rubio	López	3
4	5	3883298M	Marcos	Loyola	Méndez	5
5	6	38382980M	María	Santana	Moreno	1
6	8	71651431Z	Pepe	Ruiz	Santana	3
7	9	5633914B	Juan	Gómez	López	2
8	10	5693459C	Diego	Flores	Salas	5

```
-- 32. Lista todos los datos de los empleados cuyo segundo apellido (apeMaterno) sea López.
```

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

	codigo	dni	nombre	apePatemo	apeMaterno	codigo_departamento
1	3	R6970642A	Adolfo	Rubio	López	3
2	9	5633914B	Juan	Gómez	López	2

```
-- 33. Lista todos los datos de los empleados cuyo segundo apellido (apeMaterno) sea Díaz o Moreno, sin utilizar el operador IN.
```

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

	codigo	dni	nombre	apePatemo	apeMaterno	codigo_departamento
1	2	Y5575632D	Adela	Salas	Díaz	2
2	6	38382980M	María	Santana	Moreno	1

```
-- 34. Lista todos los datos de los empleados cuyo segundo apellido (apeMaterno) sea Díaz o Moreno, utilizando el operador IN.
SELECT *
FROM empleado
WHERE apeMaterno IN ('Díaz', 'Moreno');
```

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

```
-- 35. Lista los nombres, apellidos y DNI de los empleados que trabajan en el departamento 3.
SELECT nombre, apePaterno, apeMaterno, dni
FROM empleado
WHERE codigo_departamento = 3;
```

	nombre	apePaterno	apeMaterno	dni
1	Adolfo	Rubio	López	R6970642A
2	Pepe	Ruiz	Santana	71651431Z

```
-- 36. Lista los nombres, apellidos y DNI 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);
```

	nombre	apePaterno	apeMaterno	dni
1	Adela	Salas	Díaz	Y5575632D
2	Adrián	Suárez	NULL	7778545E
3	Marcos	Loyola	Méndez	3883298M
4	Pilar	Ruiz	NULL	8057669K
5	Juan	Gómez	López	5633914B
6	Diego	Flores	Salas	5693459C

CONSULTAS MULTITABLA(COMPOSICIÓN INTERNA)

```
-- 1. Listado con los empleados (nombre completo) y el nombre del departamento donde trabaja cada uno
SELECT CONCAT(empleado.nombre, ' ', empleado.apePaterno, ' ', empleado.apeMaterno) AS NombreCompleto,
departamento.nombre AS Departamento
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo;
```

	NombreCompleto	Departamento
1	Aarón Rivero Gómez	Desarrollo
2	Adela Salas Díaz	Sistemas
3	Adolfo Rubio López	Recursos Humanos
4	Adrián Suárez	Contabilidad
5	Marcos Loyola Méndez	I+D
6	María Santana Moreno	Desarrollo
7	Pilar Ruiz	Sistemas
8	Pepe Ruiz Santana	Recursos Humanos

```
-- 2. Listado con los empleados (nombre completo) y los datos de los departamentos donde trabaja cada uno,
-- ordenado por departamento y luego por apellidos y nombre
SELECT CONCAT(empleado.apePaterno, ' ', empleado.apeMaterno, ' ', empleado.nombre) AS NombreCompleto,
departamento.nombre AS Departamento
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo
ORDER BY departamento.nombre ASC, empleado.apePaterno ASC, empleado.apeMaterno ASC, empleado.nombre ASC;
```

	NombreCompleto	Departamento
1	Suárez Adrián	Contabilidad
2	Herrera Gil Marta	Desarrollo
3	Rivero Gómez Aarón	Desarrollo
4	Santana Moreno María	Desarrollo
5	Flores Salas Diego	I+D
6	Loyola Méndez Marcos	I+D
7	Rubio López Adolfo	Recursos H...
8	Ruiz Santana Pepe	Recursos H...

```
-- 3. Listado con el código y el nombre del departamento, solamente de aquellos departamentos que tienen empleados
SELECT DISTINCT departamento.codigo, departamento.nombre
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo;
```

	codigo	nombre
1	1	Desarrollo
2	2	Sistemas
3	3	Recurs...
4	4	Contabil...
5	5	I+D

```
-- 4. Listado con el código, el nombre del departamento y el valor del presupuesto actual, de aquellos departamentos que tienen empleados
SELECT departamento.codigo, departamento.nombre,
(departamento.presupuesto - departamento.gastos) AS PresupuestoActual
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo;
```

	codigo	nombre	PresupuestoActual
1	1	Desarrollo	114000.00
2	2	Sistemas	129000.00
3	3	Recurs...	255000.00
4	4	Contabil...	107000.00
5	5	I+D	-5000.00
6	1	Desarrollo	114000.00
7	2	Sistemas	129000.00
8	3	Recurs...	255000.00

```
-- 5. Nombre del departamento donde trabaja el empleado que tiene el nif '3883298M'
SELECT departamento.nombre AS Departamento
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo
WHERE empleado.dni = '3883298M';
```

	Departamento
1	I+D

```
-- 6. Nombre del departamento donde trabaja el empleado 'Pepe Ruiz Santana'
SELECT departamento.nombre AS Departamento
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo
WHERE empleado.nombre = 'Pepe' AND empleado.apePaterno = 'Ruiz' AND empleado.apeMaterno = 'Santana';
```

	Departamento
1	Recursos Humanos

```
-- 7. Listado de los empleados (nombre completo) que trabajan en el departamento de 'I+D', ordenado alfabéticamente
SELECT CONCAT(empleado.nombre, ' ', empleado.apePaterno, ' ', empleado.apeMaterno) AS NombreCompleto
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo
WHERE departamento.nombre = 'I+D'
ORDER BY empleado.apePaterno ASC, empleado.apeMaterno ASC, empleado.nombre ASC;
```

NombreCompleto
Diego Flores Salas
Marcos Loyola Méndez

```
-- 8. Listado de los empleados (nombre completo) que trabajan en los departamentos de 'Sistemas', 'Contabilidad' o 'I+D', ordenado alfabéticamente
SELECT CONCAT(empleado.nombre, ' ', empleado.apePaterno, ' ', empleado.apeMaterno) AS NombreCompleto
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo
WHERE departamento.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
ORDER BY empleado.apePaterno ASC, empleado.apeMaterno ASC, empleado.nombre ASC;
```

	NombreCompleto
1	Diego Flores Salas
2	Juan Gómez López
3	Marcos Loyola M...
4	Pilar Ruiz
5	Adela Salas Díaz
6	Adrián Suárez

```
-- 9. Lista con el nombre de los empleados (nombre completo) que tienen los departamentos que no tienen un
-- presupuesto entre 100000 y 200000 euros
SELECT CONCAT(empleado.nombre, ' ', empleado.apePaterno, ' ', empleado.apeMaterno) AS NombreCompleto
FROM empleado
JOIN departamento ON empleado.codigo_departamento = departamento.codigo
WHERE departamento.presupuesto NOT BETWEEN 100000 AND 200000;
```

	NombreCompleto
1	Adolfo Rubio López
2	Marcos Loyola Méndez
3	Pepe Ruiz Santana
4	Diego Flores Salas

```
-- 10. Devuelve un listado con el nombre de los departamentos donde existe algún empleado cuyo segundo apellido sea NULL.
-- Ten en cuenta que no debe mostrar nombres de departamentos que estén repetidos.
SELECT DISTINCT d.nombre AS Departamento
FROM empleado e
JOIN departamento d
ON e.codigo_departamento = d.codigo
WHERE e.apeMaterno IS NULL;
```

	Departamento
1	Contabilidad
2	Sistemas

CONSULTAS MULTITABLA(COMPOSICIÓN EXTERNA)

```
-- 1. Devuelve un listado con todos los empleados junto con los datos de los departamentos donde trabajan.
-- Este listado también debe incluir los empleados que no tienen ningún departamento asociado.
SELECT CONCAT(e.nombre, ' ', e.apePaterno, ' ', e.apeMaterno) AS NombreCompleto, d.nombre AS Departamento, d.presupuesto, d.gastos
FROM empleado e
LEFT JOIN departamento d
ON e.codigo_departamento = d.codigo;
```

	NombreCompleto	Departamento	presupuesto	gastos
1	Aarón Rivero Gómez	Desarrollo	120000.00	6000.00
2	Adela Salas Díaz	Sistemas	150000.00	21000.00
3	Adolfo Rubio López	Recursos Humanos	280000.00	25000.00
4	Adrián Suárez	Contabilidad	110000.00	3000.00
5	Marcos Loyola Méndez	I+D	375000.00	380000.00
6	María Santana Moreno	Desarrollo	120000.00	6000.00
7	Pilar Ruiz	Sistemas	150000.00	21000.00
8	Pepe Ruiz Santana	Recursos Humanos	280000.00	25000.00

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

```
SELECT CONCAT(e.nombre, ' ', e.apePaterno, ' ', e.apeMaterno) AS NombreCompleto
FROM empleado e
LEFT JOIN departamento d
ON e.codigo_departamento = d.codigo
WHERE e.codigo_departamento IS NULL;
```

	NombreCompleto
1	Irene Salas Flores
2	JuanAntonio Sáez Guerrero

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

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

	Departamento
1	Proyectos
2	Publicidad

-- 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.

```
SELECT CONCAT(e.nombre, ' ', e.apePaterno, ' ', e.apeMaterno) AS NombreCompleto, d.nombre AS Departamento, d.presupuesto, d.gastos
FROM empleado e
FULL OUTER JOIN departamento d
ON e.codigo_departamento = d.codigo
ORDER BY d.nombre;
```

	NombreCompleto	Departamento	presupuesto	gastos
1	Irene Salas Flores	NULL	NULL	NULL
2	JuanAntonio Sáez Guerrero	NULL	NULL	NULL
3	Adrián Suárez	Contabilidad	110000.00	300...
4	Aarón Rivero Gómez	Desarrollo	120000.00	600...
5	María Santana Moreno	Desarrollo	120000.00	600...
6	Marta Herrera Gil	Desarrollo	120000.00	600...
7	Diego Flores Salas	I+D	375000.00	380...
8	Marcos Loyola Méndez	I+D	375000.00	380...

-- 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.

```
SELECT CONCAT(e.nombre, ' ', e.apePaterno, ' ', e.apeMaterno) AS NombreCompleto, d.nombre AS 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 d.nombre;
```

	NombreCompleto	Departamento
1	Irene Salas Flores	NULL
2	JuanAntonio Sáez Guerrero	NULL
3		Proyectos
4		Publicidad

SUBCONSULTAS

-- CON OPERADORES BÁSICOS DE COMPARACIÓN

```
-- 1. Devuelve un listado con todos los empleados que tiene el departamento de Sistemas. (Sin utilizar INNER JOIN)
SELECT CONCAT(e.nombre, ' ', e.apaPaterno, ' ', e.apaMaterno) AS NombreCompleto
FROM empleado e, departamento d
WHERE e.codigo_departamento = d.codigo AND d.nombre = 'Sistemas';
```

	NombreCompleto
1	Adela Salas Díaz
2	Pilar Ruiz
3	Juan Gómez López

```
-- 2. Devuelve el nombre del departamento con mayor presupuesto y la cantidad que tiene asignada.
SELECT d.nombre AS Departamento, d.presupuesto
FROM departamento d
WHERE d.presupuesto = (SELECT MAX(presupuesto) FROM departamento);
```

	Departamento	presupuesto
1	I+D	375000.00

```
-- 3. Devuelve el nombre del departamento con menor presupuesto y la cantidad que tiene asignada.
SELECT d.nombre AS Departamento, d.presupuesto
FROM departamento d
WHERE d.presupuesto = (SELECT MIN(presupuesto) FROM departamento);
```

	Departamento	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00

-- SUBCONSULTAS CON ALL Y ANY

```
-- 1. Devuelve el nombre del departamento con mayor presupuesto y la cantidad que tiene asignada. Sin hacer uso de MAX, ORDER BY ni LIMIT.
SELECT d.nombre AS Departamento, d.presupuesto
FROM departamento d
WHERE d.presupuesto >= ALL (SELECT presupuesto FROM departamento);
```

	Departamento	presupuesto
1	I+D	375000.00

```
-- 2. Devuelve el nombre del departamento con menor presupuesto y la cantidad que tiene asignada. Sin hacer uso de MIN, ORDER BY ni LIMIT.
SELECT d.nombre AS Departamento, d.presupuesto
FROM departamento d
WHERE d.presupuesto <= ALL (SELECT presupuesto FROM departamento);
```

	Departamento	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00

```
-- 3. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando ALL o ANY).
SELECT DISTINCT d.nombre AS Departamento
FROM departamento d
WHERE d.codigo = ANY (SELECT e.codigo_departamento FROM empleado e);
```


	Departamento
1	Contabilidad
2	Desarrollo
3	I+D
4	Recursos H...
5	Sistemas

```
-- 4. Devuelve los nombres de los departamentos que no tienen empleados asociados. (Utilizando ALL o ANY).
SELECT DISTINCT d.nombre AS Departamento
FROM departamento d
WHERE d.codigo != ALL (SELECT e.codigo_departamento FROM empleado e);
```

	Departamento
--	--------------

```
-- SUBCONSULTAS CON IN Y NOT IN
```

```
-- 1. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando IN o NOT IN).
SELECT DISTINCT d.nombre AS Departamento
FROM departamento d
WHERE d.codigo IN (SELECT e.codigo_departamento FROM empleado e);
```

	Departamento
1	Contabilidad
2	Desarrollo
3	I+D
4	Recursos H...
5	Sistemas

```
-- 2. Devuelve los nombres de los departamentos que no tienen empleados asociados. (Utilizando IN o NOT IN).
SELECT DISTINCT d.nombre AS Departamento
FROM departamento d
WHERE d.codigo NOT IN (SELECT e.codigo_departamento FROM empleado e);
```

	Departamento
--	--------------

```
-- SUBCONSULTAS CON EXISTS Y NOT EXISTS
```

```
-- 1. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando EXISTS o NOT EXISTS).
SELECT DISTINCT d.nombre AS Departamento
FROM departamento d
WHERE EXISTS (SELECT 1 FROM empleado e WHERE e.codigo_departamento = d.codigo);
```

	Departamento
1	Contabilidad
2	Desarrollo
3	I+D
4	Recursos H...
5	Sistemas

```
-- 2. Devuelve los nombres de los departamentos que no tienen empleados asociados. (Utilizando EXISTS o NOT EXISTS).
SELECT DISTINCT d.nombre AS Departamento
FROM departamento d
WHERE NOT EXISTS (SELECT 1 FROM empleado e WHERE e.codigo_departamento = d.codigo);
```

	Departamento
1	Proyectos
2	Publicidad