



INGENIERÍA DE SISTEMAS Y COMPUTACIÓN



BASE DE DATOS II **Enunciado 02**

Realizado por:
Edwin >>>>>
Ramírez Yarasca

Catedrático:
Mg. Ing. Raúl >>>
Fernández Bejarano

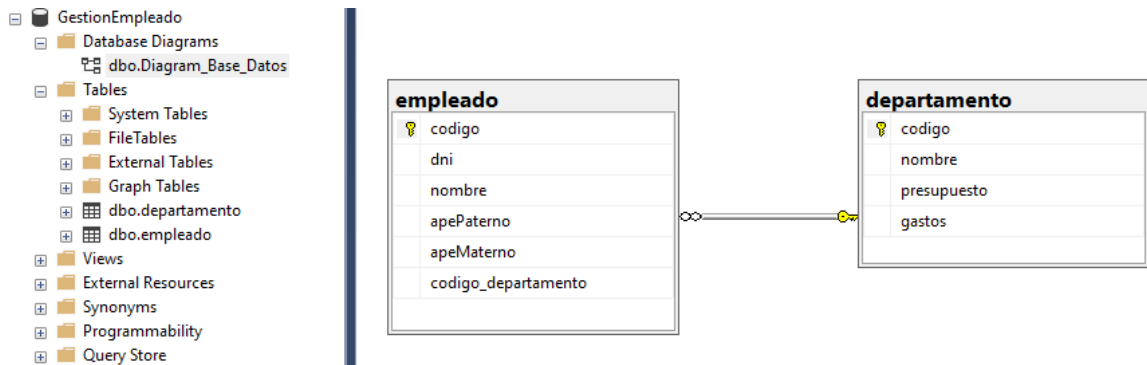


Microsoft®
SQL Server
Managemer



Enunciado 02:

Del siguiente modelo de base de datos Gestión de Empleados realizar las siguientes consultas:

**CREACIÓN DE BASE DE DATOS en SQL SERVER**

```
use master
go
Create database GestionEmpleado
on primary
    (name = GestionEmpleado_Data,
    filename= 'F:\BaseDatos2024\GestionEmpleado_Data.mdf',
    size = 5,
    maxsize = 20,
    filegrowth = 5)
log on
    (name = GestionEmpleado_Log,
    filename = 'F:\BaseDatos2024\GestionEmpleado_Log.ldf',
    size = 1,
    maxsize = 4,
    filegrowth = 1)
go
```

CREACIÓN DE LAS TABLAS EMPLEADO Y DEPARTAMENTO

```
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 (8) 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
```

INSERTAR DATOS A LAS TABLAS

```
USE GestionEmpleado;
GO
```

```
-- Insertar valores en la tabla departamento
INSERT INTO departamento (nombre, presupuesto, gastos) VALUES ('Desarrollo', 120000, 6000);
INSERT INTO departamento (nombre, presupuesto, gastos) VALUES ('Sistemas', 150000, 21000);
INSERT INTO departamento (nombre, presupuesto, gastos) VALUES ('Recursos Humanos', 280000, 25000);
INSERT INTO departamento (nombre, presupuesto, gastos) VALUES ('Contabilidad', 110000, 3000);
INSERT INTO departamento (nombre, presupuesto, gastos) VALUES ('I+D', 375000, 380000);
INSERT INTO departamento (nombre, presupuesto, gastos) VALUES ('Proyectos', 0, 0);
INSERT INTO departamento (nombre, presupuesto, gastos) VALUES ('Publicidad', 0, 1000);
```

```
-- Insertar valores en la tabla empleado
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('32481596', 'Aarón', 'Rivero', 'Gómez', 1);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('35575632', 'Adela', 'Salas', 'Díaz', 2);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('46970642', 'Adolfo', 'Rubio', 'Flores', 3);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('77705545', 'Adrian', 'Suárez', NULL, 4);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('17087203', 'Marcos', 'Loyola', 'Méndez', 5);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('38382980', 'Maria', 'Santana', 'Moreno', 1);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('80576669', 'Pilar', 'Ruiz', NULL, 2);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('71651431', 'Pepe', 'Ruiz', 'Santana', 3);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('56399183', 'Juan', 'Gomez', 'López', 2);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('46384486', 'Diego', 'Flores', 'Salas', 5);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('67389283', 'Marta', 'Herrera', 'Gil', 1);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('41234836', 'Irene', 'Salas', 'Flores', NULL);
INSERT INTO empleado (dni, nombre, apePaterno, apeMaterno, codigo_departamento)
VALUES ('82635162', 'JuanAntonio', 'Sáez', 'Guerrero', NULL);
```

CONSULTAS

A. Consultas sobre una tabla

1. Lista el primer apellido de todos los empleados.
2. Lista el primer apellido de los empleados eliminando los apellidos que estén repetidos.
3. Lista todas las columnas de la tabla empleado.
4. Lista el nombre y los apellidos de todos los empleados.
5. Lista el código de los departamentos de los empleados que aparecen en la tabla empleado.
6. Lista el código de los departamentos de los empleados que aparecen en la tabla empleado, eliminando los códigos que aparecen repetidos.

7. Lista el nombre y apellidos de los empleados en una única columna.
8. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula.
9. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en minúscula.
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.
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 que está calculando.
12. Lista el nombre de los departamentos y el valor del presupuesto actual ordenado de forma ascendente.
13. Lista el nombre de todos los departamentos ordenados de forma ascendente.
14. Lista el nombre de todos los departamentos ordenados de forma descendente.
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.
16. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen mayor presupuesto.
17. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen menor presupuesto.
18. Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen mayor gasto.
19. Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen menor gasto.
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.
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.
22. Devuelve una lista con el nombre de los departamentos y el gasto, de aquellos que tienen menos de 5000 euros de gastos.
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.
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.
25. Devuelve una lista con el nombre de los departamentos que tienen un presupuesto entre 100000 y 200000 euros. Utilizando el operador BETWEEN.
26. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre 100000 y 200000 euros. Utilizando el operador BETWEEN.
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.
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.
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.
30. Lista todos los datos de los empleados cuyo segundo apellido sea NULL.
31. Lista todos los datos de los empleados cuyo segundo apellido no sea NULL.
32. Lista todos los datos de los empleados cuyo segundo apellido sea López.
33. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Sin utilizar el operador IN.

34. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Utilizando el operador IN.
35. Lista los nombres, apellidos y nif de los empleados que trabajan en el departamento 3.
36. Lista los nombres, apellidos y nif de los empleados que trabajan en los departamentos 2, 4 o 5.

B. Consultas multitable (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.
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.
3. Devuelve un listado con el código y el nombre del departamento, solamente de aquellos departamentos que tienen empleados.
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 presupuesto) el valor de los gastos que ha generado (columna gastos).
5. Devuelve el nombre del departamento donde trabaja el empleado que tiene el nif 38382980.
6. Devuelve el nombre del departamento donde trabaja el empleado Pepe Ruiz Santana.
7. Devuelve un listado con los datos de los empleados que trabajan en el departamento de I+D. Ordena el resultado alfabéticamente.
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.
9. Devuelve una lista con el nombre de los empleados que tienen los departamentos que no tienen un presupuesto entre 100000 y 200000 euros.
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.

C. Consultas multitable (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.
2. Devuelve un listado donde sólo aparezcan aquellos empleados que no tienen ningún departamento asociado.
3. Devuelve un listado donde sólo aparezcan aquellos departamentos que no tienen ningún empleado asociado.
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.
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.

D. Consultas resumen

1. Calcula la suma del presupuesto de todos los departamentos.
2. Calcula la media del presupuesto de todos los departamentos.
3. Calcula el valor mínimo del presupuesto de todos los departamentos.
4. Calcula el nombre del departamento y el presupuesto que tiene asignado, del departamento con menor presupuesto.
5. Calcula el valor máximo del presupuesto de todos los departamentos.
6. Calcula el nombre del departamento y el presupuesto que tiene asignado, del departamento con mayor presupuesto.
7. Calcula el número total de empleados que hay en la tabla empleado.
8. Calcula el número de empleados que no tienen NULL en su segundo apellido.
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.
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.
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.
12. Calcula el número de empleados que trabajan en cada uno de los departamentos que tienen un presupuesto mayor a 200000 euros.

E. Subconsultas**Con operadores básicos de comparación**

1. Devuelve un listado con todos los empleados que tiene el departamento de Sistemas. (Sin utilizar INNER JOIN).
2. Devuelve el nombre del departamento con mayor presupuesto y la cantidad que tiene asignada.
3. Devuelve el nombre del departamento con menor presupuesto y la cantidad que tiene asignada.

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.
5. Devuelve el nombre del departamento con menor presupuesto y la cantidad que tiene asignada. Sin hacer uso de MIN, ORDER BY ni LIMIT.
6. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando ALL O ANY).
7. Devuelve los nombres de los departamentos que no tienen empleados asociados. (Utilizando ALL O ANY).

Subconsultas con IN Y NOT IN

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

Subconsultas con EXISTS Y NOT EXISTS

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

SOLUCIÓN

```
use GestionEmpleado  
go
```

```
--1--
```

```
select apePaterno from empleado;
```

	apePaterno
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--
```

```
select distinct apePaterno from empleado;
```

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


```
--3--
```

```
SELECT * FROM empleado;
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	1	32481596	Aarón	Rivero	Gómez	1
2	2	35575632	Adela	Salas	Díaz	2
3	3	46970642	Adolfo	Rubio	Flores	3
4	4	77705545	Adrian	Suárez	NULL	4
5	5	17087203	Marcos	Loyola	Méndez	5
6	6	38382980	Maria	Santana	Moreno	1
7	7	80576669	Pilar	Ruiz	NULL	2
8	8	71651431	Pepe	Ruiz	Santana	3
9	9	56399183	Juan	Gomez	López	2
10	10	46384486	Diego	Flores	Salas	5
11	11	67389283	Marta	Herrera	Gil	1
12	12	41234836	Irene	Salas	Flores	NULL
13	13	82635162	JuanAntonio	Sáez	Guerrero	NULL

```
--4--
```

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

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

```
--5--
```

```
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--

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

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

--7--

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

	nombre_completo
1	Aarón Rivero Gómez
2	Adela Salas Diaz
3	Adolfo Rubio Flores
4	Adrian 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	JuanAntonio Sáez Guerrero

--8--

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

	nombre_completo
1	AARÓN RIVERO GÓMEZ
2	ADELA SALAS DIAZ
3	ADOLFO RUBIO FLORES
4	ADRIAN 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	JUANANTONIO SÁEZ GUERRERO

Click to select the whole row

--9--

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

Results Messages

	nombre_completo
1	aarón rivero gómez
2	adela salas diaz
3	adolfo rubio flores
4	adrian 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	juanantonio sáez guerrero

--10--

```
SELECT codigo,
       LEFT(dni, LEN(dni) - 1) AS nif_digitos,
       RIGHT(dni, 1) AS nif_letra
FROM empleado;
```

Results Messages

	codigo	nif_digitos	nif_letra
1	5	1708720	3
2	1	3248159	6
3	2	3557563	2
4	6	3838298	0
5	12	4123483	6
6	10	4638448	6
7	3	4697064	2
8	9	5639918	3
9	11	6738928	3
10	8	7165143	1
11	4	7770554	5
12	7	8057666	9
13	13	8263516	2

--11--

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

	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--

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

	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--

```
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--

```
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--

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

	apePaterno	apeMaterno	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	Adrian

--16--

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

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

--17--

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

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00
3	Contabilidad	110000.00

--18--

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

	nombre	gastos
1	I+D	380000.00
2	Recursos Humanos	25000.00

--19--

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

Go

	nombre	gastos
1	Proyectos	0.00
2	Publicidad	1000.00

--20--

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

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	3	46970642	Adolfo	Rubio	Flores	3
2	4	77705545	Adrian	Suárez	NULL	4
3	5	17087203	Marcos	Loyola	Méndez	5
4	6	38382980	Maria	Santana	Moreno	1
5	7	80576669	Pilar	Ruiz	NULL	2

Go

--21--

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

Go

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

--22--

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

Go

	nombre	gastos
1	Contabilidad	3000.00
2	Proyectos	0.00
3	Publicidad	1000.00

--23--

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

Go

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	Contabilidad	110000.00

--24--

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

Go

	nombre
1	Recursos Humanos
2	I+D
3	Proyectos
4	Publicidad

--25--

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

go

	nombre	presupuesto
1	Desarrollo	120000.00
2	Sistemas	150000.00
3	Contabilidad	110000.00

--26--

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

go

	nombre
1	Recursos Humanos
2	I+D
3	Proyectos
4	Publicidad

--27--

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

go

	nombre	gastos	presupuesto
1	I+D	380000.00	375000.00
2	Publicidad	1000.00	0.00

--28--

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

go

	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--

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

go

	nombre	gastos	presupuesto
1	Proyectos	0.00	0.00

```
--30--
SELECT *
  FROM empleado
 WHERE apeMaterno IS NULL;
go
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	4	77705545	Adrian	Suárez	NULL	4
2	7	80576669	Pilar	Ruiz	NULL	2

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

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	1	32481596	Aarón	Rivero	Gómez	1
2	2	35575632	Adela	Salas	Díaz	2
3	3	46970642	Adolfo	Rubio	Flores	3
4	5	17087203	Marcos	Loyola	Méndez	5
5	6	38382980	Maria	Santana	Moreno	1
6	8	71651431	Pepe	Ruiz	Santana	3
7	9	56399183	Juan	Gomez	López	2
8	10	46384486	Diego	Flores	Salas	5
9	11	67389283	Marta	Herrera	Gil	1
10	12	41234836	Irene	Salas	Flores	NULL
11	13	82635162	JuanAntonio	Sáez	Guerrero	NULL

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

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	9	56399183	Juan	Gomez	López	2

```
--33--
SELECT *
  FROM empleado
 WHERE apeMaterno = 'Díaz' OR apeMaterno = 'Moreno';
go
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	6	38382980	Maria	Santana	Moreno	1

```
--34--
SELECT *
  FROM empleado
 WHERE apeMaterno IN ('Díaz', 'Moreno');
go
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	6	38382980	Maria	Santana	Moreno	1

```
--35--
SELECT nombre, apePaterno, apeMaterno, dni
  FROM empleado
 WHERE codigo_departamento = 3;
go
```

	nombre	apePaterno	apeMaterno	dni
1	Adolfo	Rubio	Flores	46970642
2	Pepe	Ruiz	Santana	71651431


```
--36--
SELECT nombre, apePaterno, apeMaterno, dni
FROM empleado
WHERE codigo_departamento IN (2, 4, 5);
go
```

	nombre	apePaterno	apeMaterno	dni
1	Adela	Salas	Diaz	35575632
2	Adrian	Suárez	NULL	77705545
3	Marcos	Loyola	Méndez	17087203
4	Pilar	Ruiz	NULL	80576669
5	Juan	Gomez	López	56399183
6	Diego	Flores	Salas	46384486

```
--MULTITABLA--
--1--
SELECT
    CONCAT(empleado.nombre, ' ', empleado.apePaterno, ' ', empleado.apeMaterno) AS
nombre_completo,
    departamento.nombre
FROM
    empleado
JOIN
    departamento ON empleado.codigo_departamento = departamento.codigo;
go
```

	nombre_completo	nombre
1	Aarón Rivero Gómez	Desarrollo
2	Adela Salas Diaz	Sistemas
3	Adolfo Rubio Flores	Recursos Humanos
4	Adrian Suárez	Contabilidad
5	Marcos Loyola Méndez	I+D
6	Maria Santana Moreno	Desarrollo
7	Pilar Ruiz	Sistemas
8	Pepe Ruiz Santana	Recursos Humanos
9	Juan Gomez López	Sistemas
10	Diego Flores Salas	I+D
11	Marta Herrera Gil	Desarrollo

```
--2--
SELECT CONCAT(empleado.nombre, ' ', empleado.apePaterno, ' ', empleado.apeMaterno) AS
nombre_completo,
    departamento.nombre
FROM empleado
INNER JOIN departamento ON empleado.codigo_departamento = departamento.codigo
ORDER BY departamento.nombre, empleado.apePaterno, empleado.nombre;
```

	nombre_completo	nombre
1	Adrian Suárez	Contabilidad
2	Marta Herrera Gil	Desarrollo
3	Aarón Rivero Gómez	Desarrollo
4	Maria Santana Moreno	Desarrollo
5	Diego Flores Salas	I+D
6	Marcos Loyola Méndez	I+D
7	Adolfo Rubio Flores	Recursos Humanos
8	Pepe Ruiz Santana	Recursos Humanos
9	Juan Gomez López	Sistemas
10	Pilar Ruiz	Sistemas
11	Adela Salas Diaz	Sistemas

--3--

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

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

--4--

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

	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	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--

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

	nombre
1	Desarrollo

--6--

```
SELECT d.nombre
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE e.nombre = 'Pepe' AND e.apPaterno = 'Ruiz' AND e.apMaterno = 'Santana';
```

	nombre
1	Recursos Humanos

--7--

```
SELECT e.*
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE d.nombre = 'I+D'
ORDER BY e.apePaterno, e.nombre;
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	10	46384486	Diego	Flores	Salas	5
2	5	17087203	Marcos	Loyola	Méndez	5

--8--

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

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	10	46384486	Diego	Flores	Salas	5
2	9	56399183	Juan	Gomez	López	2
3	5	17087203	Marcos	Loyola	Méndez	5
4	7	80576669	Pilar	Ruiz	NULL	2
5	2	35575632	Adela	Salas	Diaz	2
6	4	77705545	Adrian	Suárez	NULL	4

--9--

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

	nombre	apePaterno	apeMaterno
1	Adolfo	Rubio	Flores
2	Marcos	Loyola	Méndez
3	Pepe	Ruiz	Santana
4	Diego	Flores	Salas

--10--

```
SELECT DISTINCT d.nombre
FROM empleado e
JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE e.apeMaterno IS NULL;
```

	nombre
1	Contabilidad
2	Sistemas

--Consulta Multitablas--

--1--

```
SELECT e.*, d.*
FROM empleado e
LEFT JOIN departamento d ON e.codigo_departamento = d.codigo;
```

codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento	codigo	nombre	presupuesto	gastos
1	32481596	Aarón	Rivero	Gómez	1	1	Desarrollo	120000.00	6000.00
2	35575632	Adela	Salas	Díaz	2	2	Sistemas	150000.00	21000.00
3	46970642	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000.00	25000.00
4	77705545	Adrian	Suárez	NULL	4	4	Contabilidad	110000.00	3000.00
5	17087203	Marcos	Loyola	Méndez	5	5	I+D	375000.00	380000.00
6	38382980	Maria	Santana	Moreno	1	1	Desarrollo	120000.00	6000.00
7	80576669	Pilar	Ruiz	NULL	2	2	Sistemas	150000.00	21000.00
8	71651431	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000.00	25000.00
9	56399183	Juan	Gomez	López	2	2	Sistemas	150000.00	21000.00
10	46384486	Diego	Flores	Salas	5	5	I+D	375000.00	380000.00
11	67389283	Marta	Herrera	Gil	1	1	Desarrollo	120000.00	6000.00
12	41234836	Irene	Salas	Flores	NULL	NULL	NULL	NULL	NULL
13	82635162	JuanAntonio	Sáez	Guerrero	NULL	NULL	NULL	NULL	NULL

--2--

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

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	12	41234836	Irene	Salas	Flores	NULL
2	13	82635162	JuanAntonio	Sáez	Guerrero	NULL

--3--

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

	codigo	nombre	presupuesto	gastos
1	6	Proyectos	0.00	0.00
2	7	Publicidad	0.00	1000.00

--4--

```
SELECT e.*, d.*
FROM empleado e
FULL OUTER JOIN departamento d ON e.codigo_departamento = d.codigo
ORDER BY d.nombre ASC;
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento	codigo	nombre	presupuesto	gastos
1	12	41234836	Irene	Salas	Flores	NULL	NULL	NULL	NULL	NULL
2	13	82635162	JuanAntonio	Sáez	Guerrero	NULL	NULL	NULL	NULL	NULL
3	4	77705545	Adrian	Suárez	NULL	4	4	Contabilidad	110000.00	3000.00
4	1	32481596	Aarón	Rivero	Gómez	1	1	Desarrollo	120000.00	6000.00
5	6	38382980	Maria	Santana	Moreno	1	1	Desarrollo	120000.00	6000.00
6	11	67389283	Marta	Herrera	Gil	1	1	Desarrollo	120000.00	6000.00
7	10	46384486	Diego	Flores	Salas	5	5	I+D	375000.00	380000.00
8	5	17087203	Marcos	Loyola	Méndez	5	5	I+D	375000.00	380000.00
9	NULL	NULL	NULL	NULL	NULL	NULL	6	Proyectos	0.00	0.00
10	NULL	NULL	NULL	NULL	NULL	NULL	7	Publicidad	0.00	1000.00
11	3	46970642	Adolfo	Rubio	Flores	3	3	Recursos Humanos	280000.00	25000.00
12	8	71651431	Pepe	Ruiz	Santana	3	3	Recursos Humanos	280000.00	25000.00
13	9	56399183	Juan	Gomez	López	2	2	Sistemas	150000.00	21000.00
14	7	80576669	Pilar	Ruiz	NULL	2	2	Sistemas	150000.00	21000.00
15	2	35575632	Adela	Salas	Díaz	2	2	Sistemas	150000.00	21000.00

--5--

```
SELECT e.*, d.*
FROM empleado e
FULL OUTER JOIN departamento d ON e.codigo_departamento = d.codigo
WHERE e.codigo_departamento IS NULL OR d.codigo NOT IN (SELECT codigo_departamento
FROM empleado)
ORDER BY d.nombre ASC;
```

	codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento	codigo	nombre	presupuesto	gastos
1	12	41234836	Irene	Salas	Flores	NULL	NULL	NULL	NULL	NULL
2	13	82635162	Juan Antonio	Sáez	Guerrero	NULL	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL	NULL	NULL	6	Proyectos	0.00	0.00
4	NULL	NULL	NULL	NULL	NULL	NULL	7	Publicidad	0.00	1000.00

--Consulta resumen--

--1--

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

	total_presupuesto
1	1035000.00

--2--

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

	promedio_presupuesto
1	147857.142857

--3--

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

	presupuesto_minimo
1	0.00

--4--

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

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00

--5--

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

	presupuesto_maximo
1	375000.00

--6--

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

	nombre	presupuesto
1	I+D	375000.00

--7--

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

	total_empleados
1	13

--8--

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

	empleados_con_apMaterno
1	11

--9--

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

	nombre	num_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--

```
SELECT d.nombre, COUNT(e.codigo) AS num_empleados
FROM departamento d
LEFT JOIN empleado e ON d.codigo = e.codigo_departamento
GROUP BY d.nombre
HAVING COUNT(e.codigo) > 2;
```

	nombre	num_empleados
1	Desarrollo	3
2	Sistemas	3

--11--

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

	nombre	num_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--

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

	departamento	num_empleados
1	I+D	2
2	Recursos Humanos	2

```
--Subconsulta--
-- Con operadores básicos de comparación--
--1--
SELECT nombre, apePaterno, apeMaterno
FROM empleado
WHERE codigo_departamento = (SELECT codigo FROM departamento WHERE nombre =
'Sistemas');
GO
```

	nombre	apePaterno	apeMaterno
1	Adela	Salas	Diaz
2	Pilar	Ruiz	NULL
3	Juan	Gomez	López

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

	nombre	presupuesto
1	I+D	375000.00

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

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00

```
--Sub consultas con ALL y ANY--
--4--
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto >= ALL (SELECT presupuesto FROM departamento);
GO
```

	nombre	presupuesto
1	I+D	375000.00

```
--5--
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto <= ALL (SELECT presupuesto FROM departamento);
GO
```

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00

```
--6--
SELECT nombre
FROM departamento
WHERE codigo = ANY (SELECT codigo_departamento FROM empleado);
GO
```

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

--7--

```
SELECT nombre
FROM departamento
WHERE codigo != ANY (SELECT codigo_departamento FROM empleado);
GO
```

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

--Sub consultas con IN y NOT IN--

--8--

```
SELECT nombre
FROM departamento
WHERE codigo IN (SELECT codigo_departamento FROM empleado);
GO
```

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

--9--

```
SELECT nombre
FROM departamento
WHERE codigo IN (SELECT codigo_departamento FROM empleado);
GO
```

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

--Sub consultas con EXIST y NOT EXISTS--

--10--

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

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

--11--

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

	nombre
1	Proyectos
2	Publicidad