

UNIVERSIDAD PERUANA LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS
Y COMPUTACIÓN



Ejercicios

Estudiante:

Enero Rojas Cristhian Enrique

Docente:

Fernandez Bejarano Raul

Curso:

Base de datos II

HUANCAYO – PERÚ

2024

1. Crear las tablas departamento y empleado

```
1 CREATE TABLE departamento (  
2     codigo INT IDENTITY (1,1) PRIMARY KEY,  
3     nombre VARCHAR(100) NOT NULL,  
4     presupuesto NUMERIC(10, 2) NOT NULL,  
5     gastos NUMERIC(10, 2) NOT NULL  
6 );  
7  
8 CREATE TABLE empleado (  
9     codigo INT IDENTITY(1,1) PRIMARY KEY,  
10    dni VARCHAR(9) NOT NULL UNIQUE,  
11    nombre VARCHAR(100) NOT NULL,  
12    apePaterno VARCHAR(100) NOT NULL,  
13    apeMaterno VARCHAR(100),  
14    codigo_departamento INT,  
15    FOREIGN KEY (codigo_departamento) REFERENCES departamento (codigo)  
16 );  
17 GO
```

2. Insertar los datos en la tabla

```
19 INSERT INTO departamento VALUES ('Desarrollo', 120000, 6000);  
20 INSERT INTO departamento VALUES ('Sistemas', 150000, 21000);  
21 INSERT INTO departamento VALUES ('Recursos Humanos', 280000, 25000);  
22 INSERT INTO departamento VALUES ('Contabilidad', 110000, 3000);  
23 INSERT INTO departamento VALUES ('I+D', 375000, 380000);  
24 INSERT INTO departamento VALUES ('Proyectos', 0, 0);  
25 INSERT INTO departamento VALUES ('Publicidad', 0, 1000);  
26  
27 INSERT INTO empleado VALUES ('32481596F', 'Aarón', 'Rivero', 'Gómez', 1);  
28 INSERT INTO empleado VALUES ('Y5575632D', 'Adela', 'Salas', 'Díaz', 2);  
29 INSERT INTO empleado VALUES ('R6970642B', 'Adolfo', 'Rubio', 'Flores', 3);  
30 INSERT INTO empleado VALUES ('77705545E', 'Adrián', 'Suárez', NULL, 4);  
31 INSERT INTO empleado VALUES ('17087203C', 'Marcos', 'Loyola', 'Méndez', 5);  
32 INSERT INTO empleado VALUES ('38382980M', 'María', 'Santana', 'Moreno', 1);  
33 INSERT INTO empleado VALUES ('80576669X', 'Pilar', 'Ruiz', NULL, 2);  
34 INSERT INTO empleado VALUES ('71651431Z', 'Pepé', 'Ruiz', 'Santana', 3);  
35 INSERT INTO empleado VALUES ('56399183D', 'Juan', 'Gómez', 'López', 2);  
36 INSERT INTO empleado VALUES ('46384486H', 'Diego', 'Flores', 'Salas', 5);  
37 INSERT INTO empleado VALUES ('67389283A', 'Marta', 'Herrera', 'Gil', 1);  
38 INSERT INTO empleado VALUES ('41234836R', 'Irene', 'Salas', 'Flores', NULL);  
39 INSERT INTO empleado VALUES ('82635162B', 'JuanAntonio', 'Sáez', 'Guerrero', NULL);  
40
```

3.

1. Listar el primer apellido de todos los empleados.

41 SELECT apePaterno FROM empleado;

Ejecutar < Compartir

Resultados Mensajes

apePaterno
Rivero
Salas
Rubio
Suárez
Loyola

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

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

apePaterno
Flores
Gómez
Herrera
Loyola
Rivero

3. Listar todas las columnas de la tabla empleado.

```
SELECT * FROM empleado;
```

código	DNI	nombre	monoPaterno	monoMaterno	codigo_departamento
1	32481596F	Aarón	Rivero	Gómez	1
2	Y5575632D	Adela	Salas	Díaz	2
3	R6970642B	Adolfo	Rubio	Flores	3
4	77705545E	Adrián	Suárez	NULO	4
5	17087203C	Marcos	Loyola	Méndez	5

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

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

nombre	monoPaterno	monoMaterno
Aarón	Rivero	Gómez
Adela	Salas	Díaz
Adolfo	Rubio	Flores
Adrián	Suárez	NULO
Marcos	Loyola	Méndez

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

```
SELECT codigo_departamento FROM empleado;
```

codigo_departamento
1
2
3

6. Listar 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
NULO
1

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

```
41 SELECT CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno) AS nombre_completo
42 FROM empleado;
43
```

Execute Share SQL Server

Results Messages

nombre_completo
Aarón Rivero Gómez
Adela Salas Díaz
Adolfo Rubio Flores

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

```
44 SELECT UPPER(CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno)) AS nombre_completo
45 FROM empleado;
46
```

Execute Share SQL Server

nombre_completo
AARÓN RIVERO GÓMEZ
ADELA SALAS DÍAZ
ADOLFO RUBIO FLORES

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

```
47 SELECT LOWER(CONCAT(nombre, ' ', apePaterno, ' ', apeMaterno)) AS nombre_completo
48 FROM empleado;
49
```

Execute Share SQL Server

nombre_completo
aarón rivero gómez
adela salas díaz
adolfo rubio flores

10. Listar 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.

```
50 SELECT
51     codigo,
52     dni,
53     LEFT(dni, LEN(dni) - 1) AS nif_digitos,
54     RIGHT(dni, 1) AS nif_letra
55 FROM empleado;
56
```

Execute Share SQL Server

codigo	dni	nif_digitos	nif_letra
5	17087203C	17087203	C
1	32481596F	32481596	F
6	38382980M	38382980	M

11. Listar 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.

```
57 SELECT
58     nombre AS nombre_departamento,
59     (presupuesto - gastos) AS presupuesto_actual
60 FROM departamento;
```

Execute [Share](#) SQL Server

nombre_departamento	presupuesto_actual
Desarrollo	114000.00
Sistemas	129000.00
Recursos Humanos	255000.00

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

```
62 SELECT
63     nombre AS nombre_departamento,
64     (presupuesto - gastos) AS presupuesto_actual
65 FROM departamento
66 ORDER BY presupuesto_actual ASC;
```

Execute [Share](#) SQL Server

nombre_departamento	presupuesto_actual
I+D	-5000.00
Publicidad	-1000.00
Proyectos	.00
Contabilidad	107000.00

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

```
41 SELECT nombre AS nombre_departamento
42 FROM departamento
43 ORDER BY nombre ASC;
```

Execute [Share](#) SQL Server

nombre_departamento
Contabilidad
Desarrollo
I+D
Proyectos

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

```
45 SELECT nombre AS nombre_departamento
46 FROM departamento
47 ORDER BY nombre DESC;
48
```

Execute Share SQL Server

nombre_departamento
Sistemas
Recursos Humanos
Publicidad
Proyectos
I+D

15. Listar 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.

```
49 SELECT
50     apePaterno,
51     apeMaterno,
52     nombre
53 FROM empleado
54 ORDER BY apePaterno ASC, apeMaterno ASC, nombre ASC;
55
56
```

Execute Share SQL Server

apePaterno	apeMaterno	nombre
Flores	Salas	Diego
Gómez	López	Juan
Herrera	Gil	Marta
Loyola	Méndez	Marcos

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

```
41 SELECT TOP 3
42     nombre AS nombre_departamento,
43     presupuesto
44 FROM departamento
45 ORDER BY presupuesto DESC;
46
```

Execute Share SQL Server

Results Messages

nombre_departamento	presupuesto
I+D	375000.00
Recursos Humanos	280000.00
Sistemas	150000.00

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

```

47 SELECT TOP 3
48     nombre AS nombre_departamento,
49     presupuesto
50 FROM departamento
51 ORDER BY presupuesto ASC;
52

```

Execute Share SQL Server

nombre_departamento	presupuesto
Proyectos	.00
Publicidad	.00
Contabilidad	110000.00

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

```

53 SELECT TOP 2
54     nombre AS nombre_departamento,
55     gastos
56 FROM departamento
57 ORDER BY gastos DESC;
58

```

Execute Share SQL Server

nombre_departamento	gastos
I+D	380000.00
Recursos Humanos	25000.00

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

```

59 SELECT TOP 2
60     nombre AS nombre_departamento,
61     gastos
62 FROM departamento
63 ORDER BY gastos ASC;
64

```

Execute Share SQL Server

nombre_departamento	gastos
Proyectos	.00
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. La respuesta debe incluir todas las columnas de la tabla empleado.

```

65 SELECT *
66 FROM empleado
67 ORDER BY codigo
68 OFFSET 2 ROWS
69 FETCH NEXT 5 ROWS ONLY;
70

```

Execute Share SQL Server

codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
3	R6970642B	Adolfo	Rubio	Flores	3
4	77705545E	Adrián	Suárez	NULL	4
5	17087203C	Marcos	Loyola	Méndez	5
6	38383838Y	Marcos	Barrios	Méndez	1

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.

```
71 SELECT
72     nombre AS nombre_departamento,
73     presupuesto
74 FROM departamento
75 WHERE presupuesto >= 150000;
76
```

Execute [Share](#) SQL Server

nombre_departamento	presupuesto
Sistemas	150000.00
Recursos Humanos	280000.00
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.

```
41 SELECT
42     nombre AS nombre_departamento,
43     gastos
44 FROM departamento
45 WHERE gastos < 5000;
46
```

Execute [Share](#) SQL Server

Results Messages

nombre_departamento	gastos
Contabilidad	3000.00
Proyectos	.00
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.

```
47 SELECT
48     nombre AS nombre_departamento,
49     presupuesto
50 FROM departamento
51 WHERE presupuesto >= 100000 AND presupuesto <= 200000;
52
```

Execute [Share](#) SQL Server

nombre_departamento	presupuesto
Desarrollo	120000.00
Sistemas	150000.00
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.


```

53 SELECT
54     nombre AS nombre_departamento
55 FROM departamento
56 WHERE presupuesto < 100000 OR presupuesto > 200000;
57

```

Execute

Share

SQL Server

nombre_departamento
Recursos Humanos
I+D
Proyectos
Publicidad

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

```

58 SELECT
59     nombre AS nombre_departamento
60 FROM departamento
61 WHERE presupuesto BETWEEN 100000 AND 200000;
62

```

Execute

Share

SQL Server

nombre_departamento
Desarrollo
Sistemas
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.

```

63 SELECT
64     nombre AS nombre_departamento
65 FROM departamento
66 WHERE presupuesto NOT BETWEEN 100000 AND 200000;
67

```

Execute

Share

SQL Server

nombre_departamento
Recursos Humanos
I+D
Proyectos
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.

```
68 SELECT
69     nombre AS nombre_departamento,
70     gastos,
71     presupuesto
72 FROM departamento
73 WHERE gastos > presupuesto;
```

Execute Share SQL Server

nombre_departamento	gastos	presupuesto
I+D	380000.00	375000.00
Publicidad	1000.00	.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.

```
75 SELECT
76     nombre AS nombre_departamento,
77     gastos,
78     presupuesto
79 FROM departamento
80 WHERE gastos < presupuesto;
```

Execute Share SQL Server

nombre_departamento	gastos	presupuesto
Desarrollo	6000.00	120000.00
Sistemas	21000.00	150000.00
Recursos Humanos	25000.00	280000.00
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.

```
41 SELECT
42     nombre AS nombre_departamento,
43     gastos,
44     presupuesto
45 FROM departamento
46 WHERE gastos = presupuesto;
```

Execute Share SQL Server

Results Messages

nombre_departamento	gastos	presupuesto
Proyectos	.00	.00

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

```

48| SELECT *
49| FROM empleado
50| WHERE apeMaterno IS NULL;
51|

```

Execute Share SQL Server

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

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

```

52| SELECT *
53| FROM empleado
54| WHERE apeMaterno IS NOT NULL;
55|

```

Execute Share SQL Server

codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
1	32481596F	Aarón	Rivero	Gómez	1
2	Y5575632D	Adela	Salas	Díaz	2
3	R6970642B	Adolfo	Rubio	Flores	3

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

```

56| SELECT *
57| FROM empleado
58| WHERE apeMaterno = 'López';
59|

```

Execute Share SQL Server

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

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

```

60| SELECT *
61| FROM empleado
62| WHERE apeMaterno = 'Díaz' OR apeMaterno = 'Moreno';
63|

```

Execute Share SQL Server

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

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

```

64 SELECT *
65 FROM empleado
66 WHERE apeMaterno IN ('Díaz', 'Moreno');
67

```

Execute Share SQL Server

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

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

```

68 SELECT
69     nombre,
70     apePaterno,
71     apeMaterno,
72     dni AS NIF
73 FROM empleado
74 WHERE codigo_departamento = 3;
75

```

Execute Share SQL Server

nombre	apePaterno	apeMaterno	NIF
Adolfo	Rubio	Flores	R6970642B
Pepe	Ruiz	Santana	71651431Z

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

```

76 SELECT
77     nombre,
78     apePaterno,
79     apeMaterno,
80     dni AS NIF
81 FROM empleado
82 WHERE codigo_departamento IN (2, 4, 5);

```

Execute Share SQL Server

nombre	apePaterno	apeMaterno	NIF
Adela	Salas	Díaz	Y5575632D
Adrián	Suárez	NULL	77705545E
Marcos	Loyola	Méndez	17087203C

4. Consultar Multitablas (Composición interna)

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

```

41 SELECT e.nombre AS nombre_empleado, e.apePaterno, e.apeMaterno, d.nombre AS nombre_departamento, d.presu
42 FROM empleado e
43 JOIN departamento d ON e.codigo_departamento = d.codigo;
44

```

Pepe	Ruiz	Santana	Recursos Humanos	280000.00	25000.00
Juan	Gómez	López	Sistemas	150000.00	21000.00
Diego	Flores	Salas	I+D	375000.00	380000.00
Marta	Herrera	Gil	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.

```

41 SELECT e.nombre AS nombre_empleado, e.apePaterno, e.apeMaterno, d.nombre AS nombre_departamento, d.presu
42 FROM empleado e
43 JOIN departamento d ON e.codigo_departamento = d.codigo
44 ORDER BY d.nombre ASC, e.apePaterno ASC, e.apeMaterno ASC, e.nombre ASC;
45
46

```

Pepe	Ruiz	Santana	Recursos Humanos	280000.00	25000.00
Juan	Gómez	López	Sistemas	150000.00	21000.00
Pilar	Ruiz	NULL	Sistemas	150000.00	21000.00
Adela	Salas	Díaz	Sistemas	150000.00	21000.00

```

41 SELECT empleado.nombre AS nombre_empleado, empleado.apePaterno, empleado.apeMaterno, departamento.nombre
42 FROM empleado
43 INNER JOIN departamento ON empleado.codigo_departamento = departamento.codigo
44 ORDER BY departamento.nombre, empleado.apePaterno, empleado.apeMaterno, empleado.nombre;
45
46
47

```

Pepe	Ruiz	Santana	Recursos Humanos	280000.00	25000.00
Juan	Gómez	López	Sistemas	150000.00	21000.00
Pilar	Ruiz	NULL	Sistemas	150000.00	21000.00
Adela	Salas	Díaz	Sistemas	150000.00	21000.00

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

```

41 SELECT d.codigo, d.nombre
42 FROM departamento d
43 WHERE EXISTS (SELECT 1 FROM empleado e WHERE e.codigo_departamento = d.codigo);
44
45
46
47

```

Execute

Share

SQL Server

2	Sistemas
3	Recursos Humanos
4	Contabilidad
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 presupuesto) el valor de los gastos que ha generado (columna gastos).

```

41 SELECT d.codigo, d.nombre, (d.presupuesto - d.gastos) AS presupuesto_actual
42 FROM departamento d
43 WHERE EXISTS (SELECT 1 FROM empleado e WHERE e.codigo_departamento = d.codigo);
44
45
46
47

```

Execute

Share

SQL Server

2	Sistemas	129000.00
3	Recursos Humanos	255000.00
4	Contabilidad	107000.00
5	I+D	-5000.00

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

```

41 SELECT d.nombre AS nombre_departamento
42 FROM departamento d
43 JOIN empleado e ON e.codigo_departamento = d.codigo
44 WHERE e.dni = '38382980M';
45

```

Execute

Share

SQL Server

Results Messages

nombre_departamento
Desarrollo

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

```

41 SELECT d.nombre AS nombre_departamento
42 FROM departamento d
43 JOIN empleado e ON e.codigo_departamento = d.codigo
44 WHERE e.nombre = 'Pepe' AND e.apePaterno = 'Ruiz' AND e.apeMaterno = 'Santana';
45

```

Wait 2 seconds... [Share](#) SQL Server

Results [Messages](#)

nombre_departamento
Recursos Humanos

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

```

41 SELECT e.*
42 FROM empleado e
43 JOIN departamento d ON e.codigo_departamento = d.codigo
44 WHERE d.nombre = 'I+D'
45 ORDER BY e.nombre ASC;
46

```

Wait 3 seconds... [Share](#) SQL Server

Results [Messages](#)

codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
10	46384486H	Diego	Flores	Salas	5
5	17087203C	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.

```

41 SELECT e.*
42 FROM empleado e
43 JOIN departamento d ON e.codigo_departamento = d.codigo
44 WHERE d.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
45 ORDER BY e.nombre ASC;
46

```

Execute [Share](#) SQL Server

codigo	dni	nombre	apePaterno	apeMaterno	codigo_departamento
2	Y5575632D	Adela	Salas	Díaz	2
4	77705545E	Adrián	Suárez	NULL	4
10	46384486H	Diego	Flores	Salas	5
9	56399183D	Juan	Gómez	López	2

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

```

41 SELECT e.nombre
42 FROM empleado e
43 JOIN departamento d ON e.codigo_departamento = d.codigo
44 WHERE d.presupuesto NOT BETWEEN 100000 AND 200000;
45

```

Execute Share SQL Server

nombre
Adolfo
Marcos
Pepe
Diego

10. Devuelve un listado con el nombre de los departamentos donde existe algún empleado cuyo segundo apellido sea NULL. Tenga en cuenta que no debe mostrar nombres de departamentos que estén repetidos.

```

41 SELECT DISTINCT d.nombre AS nombre_departamento
42 FROM departamento d
43 JOIN empleado e ON e.codigo_departamento = d.codigo
44 WHERE e.apeMaterno IS NULL;
45

```

Execute Share SQL Server

nombre_departamento
Contabilidad
Sistemas

5. Consultas Multitablas (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.

```

41 SELECT e.nombre AS nombre_empleado, e.apePaterno, e.apeMaterno, d.nombre AS nombre_departamento, d.presu
42 FROM empleado e
43 LEFT JOIN departamento d ON e.codigo_departamento = d.codigo;
44
45

```

Execute Share SQL Server

nombre_empleado	apePaterno	apeMaterno	nombre_departamento	presupuesto	gastos
Aarón	Rivero	Gómez	Desarrollo	120000.00	6000.00
Adela	Salas	Díaz	Sistemas	150000.00	21000.00
Adolfo	Rubio	Flores	Recursos Humanos	280000.00	25000.00
Adrián	Suárez	NULL	Contabilidad	110000.00	3000.00

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


```

41 SELECT e.nombre AS nombre_empleado, e.apePaterno, e.apeMaterno
42 FROM empleado e
43 LEFT JOIN departamento d ON e.codigo_departamento = d.codigo
44 WHERE d.codigo IS NULL;
45

```

Execute

Share

SQL Server

nombre_empleado	apePaterno	apeMaterno
Irene	Salas	Flores
JuanAntonio	Sáez	Guerrero

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

```

41 SELECT d.codigo, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
42 FROM departamento d
43 LEFT JOIN empleado e ON e.codigo_departamento = d.codigo
44 WHERE e.codigo IS NULL;
45

```

Execute

Share

SQL Server

codigo	nombre_departamento	presupuesto	gastos
6	Proyectos	.00	.00
7	Publicidad	.00	1000.00

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.

```

41 SELECT e.nombre AS nombre_empleado, e.apePaterno, e.apeMaterno, d.nombre AS nombre_departamento, d.presu
42 FROM empleado e
43 FULL JOIN departamento d ON e.codigo_departamento = d.codigo
44 ORDER BY d.nombre ASC;
45

```

Execute

Share

SQL Server

nombre_empleado	apePaterno	apeMaterno	nombre_departamento	presupuesto	gastos
Irene	Salas	Flores	NULL	NULL	NULL
JuanAntonio	Sáez	Guerrero	NULL	NULL	NULL
Adrián	Suárez	NULL	Contabilidad	110000.00	3000.00
Aarón	Rivero	Gómez	Desarrollo	120000.00	6000.00

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.

```

41 SELECT e.nombre AS nombre_empleado, e.apePaterno, e.apeMaterno, d.nombre AS nombre_departamento, d.presu
42 FROM empleado e
43 FULL JOIN departamento d ON e.codigo_departamento = d.codigo
44 WHERE e.codigo_departamento IS NULL OR d.codigo IS NULL
45 ORDER BY d.nombre ASC;
46
47

```

nombre_empleado	apePaterno	apeMaterno	nombre_departamento	presupuesto	gastos
Irene	Salas	Flores	NULL	NULL	NULL
JuanAntonio	Sáez	Guerrero	NULL	NULL	NULL
NULL	NULL	NULL	Proyectos	.00	.00
NULL	NULL	NULL	Publicidad	.00	1000.00

6. Consultas resumen

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

```

41 SELECT SUM(presupuesto) AS suma_presupuesto
42 FROM departamento;
43

```

suma_presupuesto
1035000.00

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

```

41 SELECT AVG(presupuesto) AS media_presupuesto
42 FROM departamento;
43

```

media_presupuesto
147857.142857

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

```

41 SELECT MIN(presupuesto) AS minimo_presupuesto
42 FROM departamento;
43

```

minimo_presupuesto
.00

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

```
41 SELECT nombre AS nombre_departamento, presupuesto
42 FROM departamento
43 WHERE presupuesto = (SELECT MIN(presupuesto) FROM departamento);
44
```

nombre_departamento	presupuesto
Proyectos	.00
Publicidad	.00

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

```
41 SELECT MAX(presupuesto) AS maximo_presupuesto
42 FROM departamento;
43
```

maximo_presupuesto
375000.00

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

```
41 SELECT nombre AS nombre_departamento, presupuesto
42 FROM departamento
43 WHERE presupuesto = (SELECT MAX(presupuesto) FROM departamento);
44
```

nombre_departamento	presupuesto
I+D	375000.00

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

```
41 SELECT COUNT(*) AS total_empleados
42 FROM empleado;
43
```

total_empleados
13

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

```
41 SELECT COUNT(*) AS empleados_con_apeMaterno
42 FROM empleado
43 WHERE apeMaterno IS NOT NULL;
44
```

Execute

Share

SQL Server

Results

Messages

empleados_con_apeMaterno
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.

```
41 SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
42 FROM departamento d
43 LEFT JOIN empleado e ON e.codigo_departamento = d.codigo
44 GROUP BY d.nombre;
45
```

Execute

Share

SQL Server

Results

Messages

nombre_departamento	numero_empleados
Contabilidad	1
Desarrollo	3
I+D	2

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.

```
41 SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
42 FROM departamento d
43 JOIN empleado e ON e.codigo_departamento = d.codigo
44 GROUP BY d.nombre
45 HAVING COUNT(e.codigo) > 2;
46
```

Execute

Share

SQL Server

Results

Messages

nombre_departamento	numero_empleados
Desarrollo	3
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.

```

41 SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
42 FROM departamento d
43 LEFT JOIN empleado e ON e.codigo_departamento = d.codigo
44 GROUP BY d.nombre;
45

```

nombre_departamento	numero_empleados
Contabilidad	1
Desarrollo	3
I+D	2

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

```

41 SELECT d.nombre AS nombre_departamento, COUNT(e.codigo) AS numero_empleados
42 FROM departamento d
43 LEFT JOIN empleado e ON e.codigo_departamento = d.codigo
44 WHERE d.presupuesto > 200000
45 GROUP BY d.nombre;
46

```

nombre_departamento	numero_empleados
I+D	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).

```

41 SELECT nombre, apePaterno, apeMaterno
42 FROM empleado
43 WHERE codigo_departamento = (SELECT codigo FROM departamento WHERE nombre = 'Sistemas');
44

```

nombre	apePaterno	apeMaterno
Adela	Salas	Díaz
Pilar	Ruiz	NULL
Juan	Gómez	López

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

```

41 SELECT nombre, presupuesto
42 FROM departamento
43 WHERE presupuesto = (SELECT MAX(presupuesto) FROM departamento);
44

```

Execute

Share

SQL Server

Results

Messages

nombre	presupuesto
I+D	375000.00

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

```

41 SELECT nombre, presupuesto
42 FROM departamento
43 WHERE presupuesto = (SELECT MIN(presupuesto) FROM departamento);
44

```

Execute

Share

SQL Server

Results

Messages

nombre	presupuesto
Proyectos	.00
Publicidad	.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.

```

41 SELECT nombre, presupuesto
42 FROM departamento
43 WHERE presupuesto >= ALL (SELECT presupuesto FROM departamento);
44

```

Execute

Share

SQL Server

Results

Messages

nombre	presupuesto
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.

```

41 SELECT nombre, presupuesto
42 FROM departamento
43 WHERE presupuesto <= ALL (SELECT presupuesto FROM departamento);
44

```

Execute

Share

SQL Server

Results

Messages

nombre	presupuesto
Proyectos	.00
Publicidad	.00

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

```
41 SELECT nombre
42 FROM departamento
43 WHERE codigo = ANY (SELECT codigo_departamento FROM empleado);
```

Execute Share SQL Server

Results Messages

nombre
Desarrollo
Sistemas
Recursos Humanos

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

```
41 SELECT nombre
42 FROM departamento
43 WHERE codigo <> ANY (SELECT codigo_departamento FROM empleado);
44
```

Execute Share SQL Server

Results Messages

nombre
Desarrollo
Sistemas
Recursos Humanos

7.3 Subconsultas con IN y NOT IN

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

```
41 SELECT nombre
42 FROM departamento
43 WHERE codigo IN (SELECT codigo_departamento FROM empleado);
44
```

Execute Share SQL Server

nombre
Desarrollo
Sistemas
Recursos Humanos
Contabilidad

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

```

41 SELECT nombre
42 FROM departamento
43 WHERE codigo NOT IN (SELECT codigo_departamento FROM empleado);
44

```

Execute

Share

SQL Server

Results

Messages

nombre

7.4 Subconsultas con EXISTS Y NOT EXISTS

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

```

41 SELECT nombre
42 FROM departamento d
43 WHERE EXISTS (SELECT 1 FROM empleado e WHERE e.codigo_departamento = d.codigo);
44

```

Execute

Share

SQL Server

nombre

Desarrollo

Sistemas

Recursos Humanos

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

```

41 SELECT nombre
42 FROM departamento d
43 WHERE NOT EXISTS (SELECT 1 FROM empleado e WHERE e.codigo_departamento = d.codigo);
44

```

Wait 1 seconds...

Share

SQL Server

nombre

Proyectos

Publicidad