BASE DE DATOS DE EMPLEADOS Y CONSULTAS

06/05/2024

Creado por: Pedro José Riquelme Guerrero

INDICE

1.2.3 Consultas sobre una tabla

1. Lista el primer apellido de todos los empleados.
```
SELECT apellido1 FROM empleado;
```

2. Lista el primer apellido de los empleados eliminando los apellidos que estén repetidos.
```
SELECT DISTINCT apellido1 FROM empleado;
```

3. Lista todas las columnas de la tabla empleado.
```
SELECT * FROM empleado;
```
4. Lista el nombre y los apellidos de todos los empleados.

```
SELECT nombre, apellido1, apellido2 FROM empleado;
```

5. Lista el identificador de los departamentos de los empleados que aparecen en la tabla empleado.
```
SELECT id_departamento FROM empleado;
```

6. Lista el identificador de los departamentos de los empleados que aparecen en la tabla empleado, eliminando los identificadores que aparecen repetidos.
```
SELECT DISTINCT id_departamento FROM empleado;
```

7. Lista el nombre y apellidos de los empleados en una única columna.
```
SELECT CONCAT(nombre, ' ', apellido1, ' ', COALESCE(apellido2, '')) AS nombre_completo
FROM empleado;
```

8. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula.
```
SELECT UPPER(CONCAT(nombre, ' ', apellido1, ' ', COALESCE(apellido2, ''))) AS
nombre_completo FROM empleado;
```

9. Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en minúscula.
```
SELECT LOWER(CONCAT(nombre, ' ', apellido1, ' ', COALESCE(apellido2, ''))) AS
nombre_completo FROM empleado;
```

10. Lista el identificador 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 id, LEFT(nif, 8) AS nif_numerico, RIGHT(nif, 1) AS nif_letra FROM empleado;
```

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).
-- Utilice un alias apropiado para la nueva columna que está calculando.
```
SELECT nombre, presupuesto - gastos AS presupuesto_actual FROM departamento;
```

12. Lista el nombre de los departamentos y el valor del presupuesto actual ordenado de forma ascendente.
```
SELECT nombre, presupuesto - gastos AS presupuesto_actual FROM departamento ORDER
BY presupuesto_actual ASC;
```

13. Lista el nombre de todos los departamentos ordenados de forma ascendente.
```
SELECT nombre FROM departamento ORDER BY nombre ASC;
```

14. Lista el nombre de todos los departamentos ordenados de forma descendente.
```
SELECT nombre FROM departamento ORDER BY nombre DESC;Ç
```

15. Lista los apellidos y el nombre de todos los empleados, ordenados de forma alfabética teniendo en cuenta en primer lugar sus apellidos y luego su nombre.
```
SELECT apellido1, COALESCE(apellido2, ''), nombre FROM empleado ORDER BY apellido1, apellido2, nombre;
```

16. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen mayor presupuesto.
```
SELECT nombre, presupuesto FROM departamento ORDER BY presupuesto DESC LIMIT 3;
```

17. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen menor presupuesto.
```
SELECT nombre, presupuesto FROM departamento ORDER BY presupuesto ASC LIMIT 3;
```

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

19. Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen menor gasto.
```
SELECT nombre, gastos FROM departamento ORDER BY gastos ASC LIMIT 2;
```

20. Devuelve una lista con 5 filas a partir de la tercera fila de la tabla empleado. La tercera fila se debe incluir en la respuesta. La respuesta debe incluir todas las columnas de la tabla empleado.
```
SELECT * FROM empleado LIMIT 2, 5;
```

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;
```

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;
```

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;
```

24. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre 100000 y 200000 euros. Sin utilizar el operador BETWEEN.
```
SELECT nombre FROM departamento WHERE presupuesto < 100000 OR presupuesto > 200000;
```

25. Devuelve una lista con el nombre de los departamentos que tienen un presupuesto entre 100000 y 200000 euros. Utilizando el operador BETWEEN.
```
SELECT nombre FROM departamento WHERE presupuesto BETWEEN 100000 AND 200000;
```

26. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre 100000 y 200000 euros. Utilizando el operador BETWEEN.
```
SELECT nombre FROM departamento WHERE NOT presupuesto BETWEEN 100000 AND 200000;
```

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;
```

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;
```

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;
```

30. Lista todos los datos de los empleados cuyo segundo apellido sea NULL.
```
SELECT * FROM empleado WHERE apellido2 IS NULL;
```
CONSULTAS RESUMEN

31. Lista todos los datos de los empleados cuyo segundo apellido no sea NULL.
```
SELECT * FROM empleado WHERE apellido2 IS NOT NULL;
```

32. Lista todos los datos de los empleados cuyo segundo apellido sea López.
```
SELECT * FROM empleado WHERE apellido2 = 'López';
```

33. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Sin utilizar el operador IN.
```
SELECT * FROM empleado WHERE apellido2 = 'Díaz' OR apellido2 = 'Moreno';
```

34. Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Utilizando el operador IN.
```
SELECT * FROM empleado WHERE apellido2 IN ('Díaz', 'Moreno');
```

35. Lista los nombres, apellidos y nif de los empleados que trabajan en el departamento 3.
```
SELECT nombre, apellido1, apellido2, nif FROM empleado WHERE id_departamento = 3;
```

36. Lista los nombres, apellidos y nif de los empleados que trabajan en los departamentos 2, 4 o 5.
```
SELECT nombre, apellido1, apellido2, nif FROM empleado WHERE id_departamento IN (2, 4, 5);
```

1.2.4 Consultas multitabla (Composición interna)

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

```
SELECT e.*, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id;
```

2. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno. Ordena el resultado, en primer lugar por el nombre del departamento (en orden alfabético) y en segundo lugar por los apellidos y el nombre de los empleados.

```
SELECT e.*, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
ORDER BY nombre_departamento ASC, apellido1 ASC, apellido2 ASC, nombre ASC;
```

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

```
SELECT d.id, d.nombre
FROM departamento d
INNER JOIN empleado e ON d.id = e.id_departamento
GROUP BY d.id, d.nombre;
```

4. Devuelve un listado con el identificador, 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).

```
SELECT d.id, Título 1d.nombre, (d.presupuesto - d.gastos) AS presupuesto_actual
FROM departamento d
INNER JOIN empleado e ON d.id = e.id_departamento
GROUP BY d.id, d.nombre, d.presupuesto, d.gastos;
```

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

```
SELECT d.nombre AS nombre_departamento
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE e.nif = '38382980M';
```

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

```
SELECT d.nombre AS nombre_departamento
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE e.nombre = 'Pepe' AND e.apellido1 = 'Ruiz' AND e.apellido2 = 'Santana';
```

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

```
SELECT e.*
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE d.nombre = 'I+D'
ORDER BY e.apellido1, e.apellido2, e.nombre;
```

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.

```
SELECT e.*
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE d.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
ORDER BY e.apellido1, e.apellido2, e.nombre;
```

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

```
SELECT e.nombre
FROM empleado e
LEFT JOIN departamento d ON e.id_departamento = d.id
WHERE d.presupuesto NOT BETWEEN 100000 AND 200000;
```

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.

```
SELECT DISTINCT d.nombre
FROM empleado e
INNER JOIN departamento d ON e.id_departamento = d.id
WHERE e.apellido2 IS NULL;
```

1.2.5 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 e.*, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
FROM empleado e
LEFT JOIN departamento d ON e.id_departamento = d.id;
```

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

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

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

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

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 e.*, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
FROM empleado e
LEFT JOIN departamento d ON e.id_departamento = d.id

UNION ALL

SELECT e.*, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
FROM empleado e
RIGHT JOIN departamento d ON e.id_departamento = d.id
WHERE e.id IS NULL
ORDER BY nombre_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.

```
SELECT e.*, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
FROM empleado e
LEFT JOIN departamento d ON e.id_departamento = d.id
WHERE e.id_departamento IS NULL

UNION ALL

SELECT e.*, d.nombre AS nombre_departamento, d.presupuesto, d.gastos
FROM empleado e
RIGHT JOIN departamento d ON e.id_departamento = d.id
WHERE e.id IS NULL
ORDER BY nombre_departamento;
```

1.2.6 Consultas resumen

1. Calcula la suma del presupuesto de todos los departamentos.
```
SELECT SUM(presupuesto) AS suma_presupuesto FROM departamento;
```

2. Calcula la media del presupuesto de todos los departamentos.
```
SELECT AVG(presupuesto) AS media_presupuesto FROM departamento;
```

3. Calcula el valor mínimo del presupuesto de todos los departamentos.
```
SELECT MIN(presupuesto) AS min_presupuesto FROM departamento;
```

4. Calcula el nombre del departamento y el presupuesto que tiene asignado, del departamento con menor presupuesto.
```
SELECT nombre, presupuesto
FROM departamento
WHERE presupuesto = (SELECT MIN(presupuesto) FROM departamento);
```

5. Calcula el valor máximo del presupuesto de todos los departamentos.
```
SELECT MAX(presupuesto) AS max_presupuesto FROM departamento;
```

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

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

7. Calcula el número total de empleados que hay en la tabla empleado.
```
SELECT COUNT(*) AS total_empleados FROM empleado;
```

8. Calcula el número de empleados que no tienen NULL en su segundo apellido.
```
SELECT COUNT(*) AS empleados_con_segundo_apellido
FROM empleado
WHERE apellido2 IS NOT NULL;
```

9. Calcula el número de empleados que hay en cada departamento. Tienes que devolver dos columnas, una con el nombre del departamento y otra con el número de empleados que tiene asignados.
```
SELECT d.nombre AS nombre_departamento, COUNT(e.id) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.id = e.id_departamento
GROUP BY d.nombre;
```

10. Calcula el nombre de los departamentos que tienen más de 2 empleados. El resultado debe tener dos columnas, una con el nombre del departamento y otra con el número de empleados que tiene asignados.
```
SELECT d.nombre AS nombre_departamento, COUNT(e.id) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.id = e.id_departamento
GROUP BY d.nombre
HAVING COUNT(e.id) > 2;
```

11. Calcula el número de empleados que trabajan en cada uno de los departamentos. El resultado de esta consulta también tiene que incluir aquellos departamentos que no tienen ningún empleado asociado.

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

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

```
SELECT d.nombre AS nombre_departamento, COUNT(e.id) AS numero_empleados
FROM departamento d
LEFT JOIN empleado e ON d.id = e.id_departamento
WHERE d.presupuesto > 200000
GROUP BY d.nombre;12 pt
```

1.2.7 Subconsultas

1.2.7.1 Con operadores básicos de comparación

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

```
SELECT *
FROM empleado
WHERE id_departamento = (SELECT id FROM departamento WHERE nombre = 'Sistemas');
```

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

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

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

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

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

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

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

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

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

```
SELECT nombre
FROM departamento
WHERE id IN (SELECT id_departamento FROM empleado);
```

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

```
SELECT nombre
FROM departamento
WHERE id NOT IN (SELECT id_departamento FROM empleado);
```