

Informe del Trabajo de la Semana-06

RESUMEN

Consultas multitable (**Resumen, Subconsultas:Operadores, ALL, ANY, IN, NOT IN**)

6. Resumen:

1. **SELECT SUM(presupuesto) AS TotalPresupuesto**
FROM departamento;

- Obtiene la suma total de los presupuestos asignados a todos los departamentos.

	TotalPresupuesto
1	6210000.00

2. **SELECT AVG(presupuesto) AS MediaPresupuesto**
FROM departamento;

- Calcula el promedio del presupuesto entre todos los departamentos, proporcionando una idea del gasto medio por departamento.

	MediaPresupuesto
1	147857.142857

3. **SELECT MIN(presupuesto) AS MinPresupuesto**
FROM departamento;

- Encuentra el presupuesto más bajo entre todos los departamentos, lo que puede ayudar a identificar áreas con menos recursos.

	MinPresupuesto
1	0.00

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

- Muestra el nombre y el presupuesto del departamento que tiene el menor presupuesto, destacando posibles limitaciones financieras.

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00
3	Proyectos	0.00
4	Publicidad	0.00
5	Proyectos	0.00
6	Publicidad	0.00
7	Proyectos	0.00
8	Publicidad	0.00
9	Proyectos	0.00
10	Publicidad	0.00
11	Proyectos	0.00
12	Publicidad	0.00

```
5. SELECT MAX(presupuesto) AS MaxPresupuesto
FROM departamento;
```

- Encuentra el presupuesto más alto entre los departamentos, lo que puede indicar áreas con mayores inversiones.

	MaxPresupuesto
1	375000.00

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

- Muestra el nombre y el presupuesto del departamento que tiene el mayor presupuesto, lo que puede señalar la importancia estratégica de ese departamento.

	nombre	presupuesto
1	I+D	375000.00
2	I+D	375000.00
3	I+D	375000.00
4	I+D	375000.00
5	I+D	375000.00
6	I+D	375000.00

7. **SELECT COUNT(*) AS TotalEmpleados**
FROM empleado;

- Cuenta cuántos empleados hay en la tabla, proporcionando un número total de la fuerza laboral de la organización.

	TotalEmpleados
1	15

8. **SELECT COUNT(*) AS EmpleadosConSegundoApellido**
FROM empleado
WHERE apeMaterno IS NOT NULL;

- Cuenta cuántos empleados tienen un segundo apellido registrado, ayudando a entender la completitud de los datos.

	EmpleadosConSegundoApellido
1	13

9. **SELECT d.nombre AS Departamento, COUNT(e.codigo) AS**
TotalEmpleados
FROM departamento d
LEFT JOIN empleado e ON d.codigo = e.codigo_departamento
GROUP BY d.nombre;

- Devuelve una lista que muestra el número de empleados en cada departamento, lo que puede ayudar en la evaluación de la distribución de recursos humanos.

	Departamento	TotalEmpleados
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 AS Departamento,**
COUNT(e.codigo) AS TotalEmpleados
FROM departamento d
JOIN empleado e ON d.codigo = e.codigo_departamento
GROUP BY d.nombre
HAVING COUNT(e.codigo) > 2;

- Muestra aquellos departamentos que tienen más de dos empleados, lo que puede ser útil para identificar áreas con mayor actividad laboral.

	Departamento	TotalEmpleados
1	Desarrollo	3
2	Sistemas	3

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

- Presenta una lista de todos los departamentos junto con el número de empleados asignados a cada uno, incluyendo aquellos que no tienen empleados, lo que proporciona una visión completa de la estructura organizativa.

	Departamento	TotalEmpleados
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  
TotalEmpleados  
FROM departamento d  
LEFT JOIN empleado e ON d.codigo = e.codigo_departamento  
WHERE d.presupuesto > 200000  
GROUP BY d.nombre;
```

- Obtiene una lista con el nombre del departamento y el número de empleados en cada uno de ellos que cumplen con el criterio de presupuesto.

	Departamento	TotalEmpleados
1	I+D	2
2	Recursos Humanos	2

7. Subconsultas

7.1 Con operadores básicos de comparación:

1. **SELECT * FROM empleado**

**WHERE codigo_departamento = (SELECT codigo FROM departamento
WHERE nombre = 'Sistemas');**

- Esta consulta muestra todos los empleados que pertenecen al departamento de Sistemas, utilizando una subconsulta para identificar el departamento.

	codigo	dni	nombre	apePatemo	apeMatemo	codigo_departamento
1	2	Y5575632D	Adela	Salas	Díaz	2
2	7	80576669X	Pilar	Ruiz	NULL	2
3	9	56399183D	Juan	Gómez	López	2

2. **SELECT nombre, presupuesto**

FROM departamento

**WHERE presupuesto = (SELECT MAX(presupuesto) FROM
departamento);**

- Identifica el departamento con el mayor presupuesto y presenta su nombre y cantidad, utilizando una subconsulta que evita funciones agregadas.

	nombre	presupuesto
1	I+D	375000.00
2	I+D	375000.00
3	I+D	375000.00
4	I+D	375000.00
5	I+D	375000.00
6	I+D	375000.00

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

- Similar al anterior, esta consulta muestra el departamento con el menor presupuesto, evitando el uso de funciones como MIN.

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00
3	Proyectos	0.00
4	Publicidad	0.00
5	Proyectos	0.00
6	Publicidad	0.00
7	Proyectos	0.00
8	Publicidad	0.00
9	Proyectos	0.00
10	Publicidad	0.00
11	Proyectos	0.00
12	Publicidad	0.00
13	Proyectos	0.00
14	Publicidad	0.00

7.2 Subconsultas con ALL y ANY:

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

- Encuentra el departamento con el mayor presupuesto usando comparaciones, sin utilizar funciones de agregación.

	nombre	presupuesto
1	I+D	375000.00
2	I+D	375000.00
3	I+D	375000.00
4	I+D	375000.00
5	I+D	375000.00
6	I+D	375000.00
7	I+D	375000.00

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

- Identifica el departamento con el menor presupuesto sin usar funciones como MIN, basándose en comparaciones.

	nombre	presupuesto
1	Proyectos	0.00
2	Publicidad	0.00
3	Proyectos	0.00
4	Publicidad	0.00
5	Proyectos	0.00
6	Publicidad	0.00
7	Proyectos	0.00
8	Publicidad	0.00
9	Proyectos	0.00
10	Publicidad	0.00
11	Proyectos	0.00
12	Publicidad	0.00
13	Proyectos	0.00
14	Publicidad	0.00

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

- Devuelve los nombres de los departamentos que tienen empleados, utilizando condiciones de comparación para evaluar la existencia de registros.

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

```
7. SELECT nombre
FROM departamento
WHERE codigo NOT IN (SELECT codigo_departamento FROM
empleado) ;
```

- Muestra los departamentos que no tienen empleados asociados, utilizando condiciones de comparación que aseguran que no existan registros en la tabla de empleados.

	nombre
1	Proyectos
2	Publicidad
3	Marketing
4	Logística
5	IT Support
6	Departamento A
7	Departamento B

7.3 Subconsultas con IN y NOT IN:

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

- Devuelve los nombres de los departamentos que tienen empleados utilizando la cláusula IN para filtrar aquellos que aparecen en la lista.

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

```
9. SELECT nombre
FROM departamento
WHERE codigo NOT IN (SELECT codigo_departamento FROM
empleado) ;
```

- Muestra los departamentos que no tienen empleados asociados, utilizando la cláusula NOT IN para identificar aquellos que no están presentes en la lista.

	nombre
1	Proyectos
2	Publicidad
3	Marketing
4	Logística
5	IT Support
6	Departamento A
7	Departamento B

7.4 Subconsultas con EXISTS Y NOT EXISTS:

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

- Identifica los departamentos que tienen empleados, utilizando la cláusula EXISTS para verificar si existen registros relacionados en la tabla de empleados.

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

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

- Esta consulta devuelve los nombres de los departamentos que tienen al menos un empleado asociado, utilizando EXISTS para comprobar la existencia de registros en la tabla de empleados.

	nombre
1	Proyectos
2	Publicidad
3	Marketing
4	Logística
5	IT Support
6	Departamento A
7	Departamento B