

SUBCONSULTAS EN LA LISTA DEL SELECT

Existen dos tipos de subconsultas: subconsultas simples o básicas y las correlacionadas.

Una subconsulta básica es una declaración SELECT independiente, funciona por sí sola, que está incrustada dentro de otra consulta SQL.

Mientras que, una subconsulta correlacionada también es una declaración SELECT dentro de otra consulta, pero en este caso hay una dependencia. La subconsulta correlacionada no se puede ejecutar independientemente de la declaración en la que está incrustada.

Tanto una subconsulta básica como una correlacionada se pueden utilizar en cualquier lugar de la declaración SELECT donde podamos usar una expresión.

Dependiendo de cómo se utilice la consulta interna (subconsulta) en relación con la consulta externa, se determinará cuantas columnas y valores puede devolver una subconsulta. Por ejemplo, cuando se utiliza una subconsulta en la cláusula WHERE que contiene un operador de comparación (=, !=, <, > o >=) la subconsulta debe devolver un valor de una sola columna.

Si se usa un operador de comparación que admite múltiples valores como una expresión IN entonces la subconsulta podrá devolver múltiples valores. También puede devolver varias columnas cuando se usa una expresión EXISTS.

Los dos tipos de subconsultas se pueden usar en la lista de columnas del SELECT cuando devuelven un solo valor.

Ejemplo con una consulta básica:

```
SELECT nombre FROM empleado WHERE id='2';
```

Vamos a utilizar esta subconsulta básica en una consulta.

```
SELECT id, nombre, gastos, (SELECT nombre FROM empleado  
WHERE id='2') as empleado FROM departamento WHERE id = '3';
```

En cambio, las consultas correlacionadas hacen referencia a columnas de la tabla de la consulta externa. La subconsulta correlacionada se ejecuta una vez por cada fila devuelta por la consulta externa. No olvidemos que la subconsulta correlacionada no puede ejecutarse por sí sola.

```
SELECT nombre, nif, (SELECT SUM(presupuesto)as total FROM  
departamento
```

```
WHERE departamento.id = empleado.id_departamento)
```

```
FROM empleado ORDER BY total DESC;
```

Podemos usar alias para los nombres de las tablas:

```
SELECT nombre, nif, (SELECT SUM(presupuesto)as total FROM  
departamento d WHERE d.id = e.id_departamento) FROM empleado e ORDER  
BY total DESC;
```