|  |  |
| --- | --- |
| **Subconsultas con in** |  |

Vimos que una subconsulta puede reemplazar una expresión. Dicha subconsulta debe devolver un valor escalar o una lista de valores de un campo; las subconsultas que retornan una lista de valores reemplazan a una expresión en una cláusula "where" que contiene la palabra clave "in".

El resultado de una subconsulta con "in" (o "not in") es una lista. Luego que la subconsulta retorna el resultados, la consulta exterior los usa.

La sintaxis básica es la siguiente:

...where EXPRESION in (SUBCONSULTA);

Este ejemplo muestra los nombres de las editoriales que han publicado libros de un determinado autor:

select nombre

from editoriales

where codigo in

(select codigoeditorial

from libros

where autor='Richard Bach');

La subconsulta (consulta interna) retorna una lista de valores de un solo campo (codigo) que la consulta exterior luego emplea al recuperar los datos.

Podemos reemplazar por un "join" la consulta anterior:

select distinct nombre

from editoriales as e

join libros

on codigoeditorial=e.codigo

where autor='Richard Bach';

Una combinación (join) siempre puede ser expresada como una subconsulta; pero una subconsulta no siempre puede reemplazarse por una combinación que retorne el mismo resultado. Si es posible, es aconsejable emplear combinaciones en lugar de subconsultas, son más eficientes.

Se recomienda probar las subconsultas antes de incluirlas en una consulta exterior, así puede verificar que retorna lo necesario, porque a veces resulta difícil verlo en consultas anidadas.

También podemos buscar valores No coincidentes con una lista de valores que retorna una subconsulta; por ejemplo, las editoriales que no han publicado libros de un autor específico:

select nombre

from editoriales

where codigo not in

(select codigoeditorial

from libros

where autor='Richard Bach');

|  |  |
| --- | --- |
| **Subconsultas (Exists y No Exists)** |  |

Los operadores "exists" y "not exists" se emplean para determinar si hay o no datos en una lista de valores.

Estos operadores pueden emplearse con subconsultas correlacionadas para restringir el resultado de una consulta exterior a los registros que cumplen la subconsulta (consulta interior). Estos operadores retornan "true" (si las subconsultas retornan registros) o "false" (si las subconsultas no retornan registros).

Cuando se coloca en una subconsulta el operador "exists", MySQL analiza si hay datos que coinciden con la subconsulta, no se devuelve ningún registro, es como un test de existencia; MySQL termina la recuperación de registros cuando por lo menos un registro cumple la condición "where" de la subconsulta.

La sintaxis básica es la siguiente:

... where exists (SUBCONSULTA);

En este ejemplo se usa una subconsulta correlacionada con un operador "exists" en la cláusula "where" para devolver una lista de clientes que compraron el artículo "lapiz":

select cliente,numero

from facturas as f

where exists

(select \* from detalles as d

where f.numero=d.numerofactura

and d.articulo='lapiz');

Puede obtener el mismo resultado empleando una combinación.

Podemos buscar los clientes que no han adquirido el artículo "lapiz" empleando "if not exists":

select cliente,numero

from facturas as f

where not exists

(select \* from detalles as d

where f.numero=d.numerofactura

and d.articulo='lapiz');