

Subconsultas

Sitio: Agencia de Habilidades para el Futuro
Curso: Administración de Base de Datos 1° D
Libro: Subconsultas

Imprimido por: RODRIGO PINTO
Día: miércoles, 27 de noviembre de 2024, 09:07

Tabla de contenidos

- 1. ¿Qué es una subconsulta?**
- 2. Subconsultas en el "select"**
- 3. Subconsulta en el "from"**
- 4. Subconsulta en el "where"**
- 4.1. Errores frecuentes**

¿Qué es una subconsulta?



Una **subconsulta** es una instrucción select - from - where que está anidada dentro de otra consulta.

Características y usos de las subconsultas

La **subconsulta** se ejecuta **una vez antes de ejecutarse la consulta principal**.

Los resultados de la subconsulta son utilizados en la consulta principal.

Si estamos seguros de que la subconsulta devuelve un único valor, es posible utilizar los operadores de comparación habituales =, < >, <=, <, >=, >

Para subconsultas que devuelven múltiples valores se pueden utilizar los operadores:

- **In / not in**
- Los operadores de comparación habituales junto a los cuantificadores **any**, **some** y **all**. El predicado de comparación cuantificado con **any** o **some** se evalúa **a cierto** si lo es para alguna fila de la expresión de tabla. El predicado de comparación cuantificado con **all** se evalúa a cierto si lo es para **todas las filas de la expresión** de tabla.



el **where**.

La ubicación de la **subconsulta** en la consulta principal puede ser en el **select**, en el **from** o en

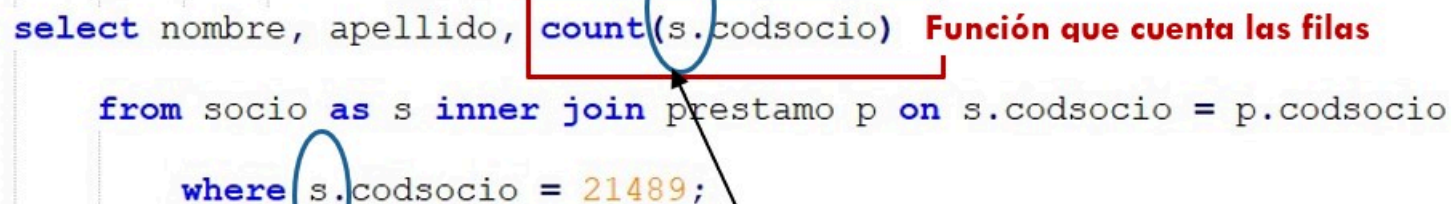
Caso 1: subconsultas en el "select"

Cuando la subconsulta está en el **select** debe arrojar **un único** valor, es decir, el resultado tiene una fila y una columna.

Supongamos que queremos saber el nombre / apellido y cuántos préstamos solicitó el socio con código 21489. Las tablas que intervienen son **socio**, para mostrar el nombre, y **préstamo**, para contar la cantidad de veces que aparece en esa entidad.

Seguramente la consulta que pensamos es:

```
select nombre, apellido, count(s.codsocio)
from socio as s inner join prestamo p on s.codsocio = p.codsocio
where s.codsocio = 21489;
```



Como la columna en las tablas tiene el mismo nombre se indica uno de ellas



Veamos otro ejemplo

Transformemos la consulta agregando una subconsulta en el **select**, esto es posible ya que el nombre y apellido es **único** para el código de socio 21489, pero para mostrarlo deben estar ambos en una sola columna, ya que como dijimos antes debe tener una sola fila y una sola columna.

Veamos cómo queda:

Función que concatena las columnas(para que no queden los datos pegados se agregan espacios en blanco)

Subconsulta

```
select (select concat(nombre, ' ', apellido) from socio where codsocio = 21489), count(codsocio)
from prestamo
where codsocio = 21489;
```


Caso 2: Subconsulta en el "From"



Cuando **la subconsulta está en el from** se utiliza como si fuese una **tabla**, es decir las columnas que proyecta la consulta se consideran como los atributos de una tabla ficticia a la que le asignamos un nombre, por lo tanto, **puede tener múltiples columnas y filas**.

Supongamos que queremos conocer los autores de los libros de la editorial McGraw-Hill y de la editorial MP Ediciones. Las tablas que intervienen son **libro**, porque tiene la editorial, **libroautor** para conocer el código de autor y a través de la **tabla autor** saber quiénes son.

Seguramente la consulta que pensamos es:

```
select nombre, apellido
  from autor as a inner join libroautor as la on a.codautor = la.codautor
    inner join libro as l on la.codlibro = l.codlibro
  where editorial = 'McGraw-Hill' or editorial = 'MP Ediciones';
```

Condición



Veamos otro ejemplo

Transformemos la consulta colocando una **subconsulta** en el **from**.

La propuesta es armar una consulta que proyecte los códigos de libro de las editoriales en cuestión y utilizarla como tabla en el **from**.

Veamos cómo es:

```
select nombre, apellido
```

Esta parte es igual a la anterior

```
from autor as a inner join libroautor as la on a.codautor = la.codautor
```

```
inner join (select codlibro from libro
```

```
where editorial = 'McGraw-Hill' or editorial = 'MP Ediciones') T1
```

```
on la.codlibro = T1.codlibro;
```

La SubConsulta colocada en el FROM de la consulta principal se transforma en una Tabla. Se le dio el nombre de T1

Observa que el codLibro ahora esta en la tabla T1

Caso 3: Subconsulta en el where



Cuando la **subconsulta** está en el **where** puede devolver uno o varios valores, es decir, tiene **una o varias filas**, pero solamente **una columna**. Dependiendo de los valores se utilizarán unos operandos u otro.



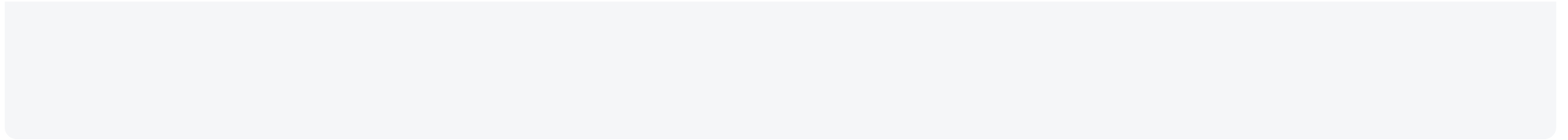
Veamos un ejemplo

Tomemos la consulta del caso 2 (**subconsulta en el from**) "*conocer los autores de los libros de la editorial McGraw-Hill y de la editorial MP Ediciones*".

```
select nombre, apellido
from autor as a inner join libroautor as la on a.codautor = la.codautor
where codlibro IN (select codlibro from libro
                  where editorial = 'McGraw-Hill' or editorial = 'MP Ediciones');
```

Se usa el operador IN porque la subconsulta puede tener en el resultado más de una fila

Subconsulta en el WHERE



Errores Frecuentes a nivel del where



La subconsulta en el “where” indica que estamos buscando para un **atributo determinado un valor o varios valores**, los cuales son el resultado de una subconsulta.

Si buscamos un valor quiere decir que o es **igual**, o es **mayor**, o es **menor** o las **posibles combinaciones**.

Si buscamos varios valores quiere decir que puede tomar más de un valor del conjunto de dominios arrojados por la subconsulta; los operadores a utilizar son **in**, **not in**, **any**, **some**, y **all**.



Pero si es tan simple, ¿dónde está el error?

El error se presenta en los siguientes casos:

- La subconsulta tiene **más de una columna**.
- La subconsulta **devuelve varias filas** y el operador utilizado es un **igual, mayor, menor o las combinaciones**.

Estos errores ocurren porque **las consultas y subconsultas** se razonan en función de los **dominios almacenados** en lugar de razonarlas en función del requerimiento y la regla de negocios. Se piensa que los datos posibles son **solamente los existentes** y se olvida que **las tablas son flexibles**.