



Técnico en
< DESARROLLO DE SOFTWARE >

Bases de Datos II

(CC BY-NC-ND 4.0)
International

Attribution-NonCommercial-NoDerivatives 4.0



Atribución

Usted debe reconocer el crédito de una obra de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace.



No Comercial

Usted no puede hacer uso del material con fines comerciales.



Sin obra derivada

Si usted mezcla, transforma o crea un nuevo material a partir de esta obra, no puede distribuir el material modificado.

No hay restricciones adicionales - Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



Bases de Datos II

Unidad II

Sub consultas anidadas / Vistas

1. Subconsultas en MySQL

Una subconsulta es una sentencia SELECT que aparece dentro de otra sentencia SELECT que llamaremos consulta principal, se puede encontrar en la lista de selección, en la cláusula WHERE o en la cláusula HAVING de la consulta principal.

Una subconsulta tiene la misma sintaxis que una sentencia SELECT normal exceptuando que aparece encerrada entre paréntesis, no puede contener la cláusula ORDER BY, ni puede ser la UNION de varias sentencias SELECT, además tiene algunas restricciones en cuanto a número de columnas según el lugar donde aparece en la consulta principal. Estas restricciones las iremos describiendo en cada caso.

Cuando se ejecuta una consulta que contiene una subconsulta, la subconsulta se ejecuta por cada fila de la consulta principal, se aconseja no utilizar campos calculados en las subconsultas, ralentizan la consulta.

Las consultas que utilizan subconsultas suelen ser más fáciles de interpretar por el usuario.

Referencias externas

A menudo, es necesario, dentro del cuerpo de una subconsulta, hacer referencia al valor de una columna en la fila actual de la consulta principal, ese nombre de columna se denomina referencia externa.

Una referencia externa es un nombre de columna que estando en la subconsulta, no se refiere a ninguna columna de las tablas designadas en la FROM de la subconsulta sino a una columna de las tablas designadas en la FROM de la consulta principal. Como la subconsulta se ejecuta por cada fila de la consulta principal, el valor de la referencia externa irá cambiando.

Ejemplo:

```
SELECT num_empleado, nombre, (SELECT MIN(fecha_pedido) FROM pedidos
WHERE rep = num_empleado) FROM empleados;
```

En este ejemplo la consulta principal es SELECT... FROM empleados.

La subconsulta es (SELECT MIN(fechapedido) FROM pedidos WHERE rep = num_empleado).

En esta subconsulta tenemos una referencia externa (num_empleado) es un campo de la tabla empleados (origen de la consulta principal).

¿Qué pasa cuando se ejecuta la consulta principal?

Se selecciona el primer empleado y se calcula la subconsulta sustituyendo num_empleado por el valor que tiene en el primer empleado. La subconsulta obtiene la fecha más antigua en los pedidos del rep = 101.

Se selecciona el segundo empleado y se calcula la subconsulta con num_empleado = 102 (num_empleado del segundo empleado)... y así sucesivamente hasta llegar al último empleado.

Al final obtenemos una lista con el número, nombre y fecha del primer pedido de cada empleado.

Si quitamos la cláusula WHERE de la subconsulta obtenemos la fecha del primer pedido de todos los pedidos no del empleado correspondiente.

Anidar subconsultas

Las subconsultas pueden anidarse de forma que una subconsulta aparezca en la cláusula WHERE (por ejemplo) de otra subconsulta que a su vez forma parte de otra consulta principal. En la práctica, una consulta consume mucho más tiempo y memoria cuando se incrementa el número de niveles de anidamiento. La consulta resulta también más difícil de leer, comprender y mantener cuando contiene más de uno o dos niveles de subconsultas.

Ejemplo:

```
SELECT num_empleado, nombre FROM empleados WHERE num_empleado =  
(SELECT rep FROM pedidos WHERE cliente = (SELECT num_cliente FROM clientes  
WHERE nombre = 'Julia Antequera'))
```

En este ejemplo, por cada línea de pedido se calcula la subconsulta de clientes, y esto se repite por cada empleado, en el caso de tener 10 filas de empleados y 200 filas de pedidos (tablas realmente pequeñas), la subconsulta más interna se ejecutaría 2000 veces (10 x 200).

2. Subconsulta en la lista de selección

Cuando la subconsulta aparece en la lista de selección de la consulta principal, en este caso la subconsulta, no puede devolver varias filas ni varias columnas, de lo contrario se da un mensaje de error.

Muchas sentencias SQL no permiten que una subconsulta aparezca en la lista de selección de la consulta principal pero eso no es ningún problema ya que normalmente se puede obtener lo mismo utilizando como origen de datos las dos tablas. El ejemplo anterior se puede obtener de la siguiente forma:

```
SELECT num_empleado, nombre, MIN(fechapedido) FROM empleados LEFT JOIN  
pedidos ON empleados.num_empleado = pedidos.rep GROUP BY num_empleado,  
nombre
```

En la cláusula FROM

En la cláusula FROM se puede encontrar una sentencia SELECT encerrada entre paréntesis, pero más que subconsulta sería una consulta ya que no se ejecuta para cada fila de la tabla origen, sino que se ejecuta una sola vez al principio, su resultado se combina con las filas de la otra tabla para formar las filas origen de la SELECT primera y no admite referencias externas.

En la cláusula FROM notamos que se podía poner un nombre de tabla o un nombre de consulta, pues en vez de poner un nombre de consulta se puede poner directamente la sentencia SELECT correspondiente a esa consulta encerrada entre paréntesis.

Subconsulta en las cláusulas WHERE y HAVING

Se suele utilizar subconsultas en las cláusulas WHERE o HAVING cuando los datos que queremos visualizar están en una tabla, pero para seleccionar las filas de esa tabla necesitamos un dato que está en otra tabla.

Ejemplo:

```
SELECT num_empleado, nombre FROM empleados WHERE contrato = (SELECT  
MIN(fechapedido) FROM pedidos)
```

En este ejemplo listamos el número y nombre de los empleados cuya fecha de contrato sea igual a la primera fecha de todos los pedidos de la empresa.

En una cláusula WHERE / HAVING tenemos siempre una condición y la subconsulta actúa de operando dentro de esa condición.

En el ejemplo anterior se compara contrato con el resultado de la subconsulta. Hasta ahora las condiciones estudiadas tenían como operandos valores simples (el valor contenido en una columna de una fila de la tabla, el resultado de una operación aritmética...) ahora la subconsulta puede devolver una columna entera por lo que es necesario definir otro tipo de condiciones especiales para cuando se utilizan con subconsultas.

Condiciones asociadas a las subconsultas son las siguientes:

El test ANY

La subconsulta debe devolver una única columna sino se produce un error, se evalúa la comparación con cada valor devuelto por la subconsulta.

Si alguna de las comparaciones individuales produce el resultado verdadero, el test ANY devuelve el resultado verdadero.

Si la subconsulta no devuelve ningún valor, el test ANY devuelve falso.

Si el test de comparación es falso para todos los valores de la columna, ANY devuelve falso.

Si el test de comparación no es verdadero para ningún valor de la columna, y es nulo para al menos alguno de los valores, ANY devuelve nulo.

Ejemplo:


```
SELECT oficina, ciudad FROM oficinas WHERE objetivo > ANY (SELECT SUM(cuota)
FROM empleados GROUP BY oficina)
```

El test ALL

La subconsulta debe devolver una única columna sino se produce un error, se evalúa la comparación con cada valor devuelto por la subconsulta.

Si todas las comparaciones individuales, producen un resultado verdadero, el test devuelve el valor verdadero.

Si la subconsulta no devuelve ningún valor el test ALL devuelve el valor verdadero.

Si el test de comparación es falso para algún valor de la columna, el resultado es falso.

Si el test de comparación no es falso para ningún valor de la columna, pero es nulo para alguno de esos valores, el test ALL devuelve valor nulo.

```
SELECT oficina, ciudad FROM oficinas WHERE objetivo > ALL (SELECT SUM(cuota)
FROM empleados GROUP BY oficina)
```

El test IN

Examina si el valor de la expresión es uno de los valores incluidos en la lista de valores producida por la subconsulta. La subconsulta debe generar una única columna y las filas que sean.

Si la subconsulta no produce ninguna fila, el test da falso.

Tiene la siguiente sintaxis:

```
SELECT num_empleado, nombre, oficina FROM empleados WHERE oficina IN  
(SELECT oficina FROM oficinas WHERE region = 'sur')
```

Sentencia con NOT IN

```
SELECT oficina, ciudad FROM oficinas WHERE oficina NOT IN (SELECT oficina FROM  
empleados)
```

```
SELECT oficina, ciudad FROM oficinas WHERE oficina NOT IN (SELECT oficina FROM  
empleados WHERE oficina IS NOT NULL)
```

El test EXISTS

Examina si la subconsulta produce alguna fila de resultados.

Si la subconsulta contiene filas, el test adopta el valor verdadero, si la subconsulta no contiene ninguna fila, el test toma el valor falso, nunca puede tomar el valor nulo.

Con este test la subconsulta puede tener varias columnas, no importa ya que el test se fija no en los valores devueltos sino en si hay o no fila en la tabla resultado de la subconsulta.

Cuando se utiliza el test de existencia en la mayoría de los casos habrá que utilizar una referencia externa. Si no se utiliza una referencia externa la subconsulta devuelta siempre será la misma para todas las filas de la consulta principal y en este caso se seleccionan todas las filas de la consulta principal (si la subconsulta genera filas) o ninguna (si la subconsulta no devuelve ninguna fila)

```
SELECT num_empleado, nombre, oficina FROM empleados WHERE EXISTS (SELECT  
* FROM oficinas WHERE region = 'este' AND empleados.oficina = oficinas.oficina)
```

3. Vistas en MySQL

Las vistas en MySQL es una tabla virtual con una estructura que nosotros definimos, pero sin datos. ¿En que nos pueden ayudar las vistas? Las respuestas son varias, y seguro que hay muchas, pero las siguientes aclaran el objetivo de utilizarlas.

Se almacenan en el servidor con lo que el consumo de recursos y eficacia siempre serán más óptimos.

En temas de seguridad siempre es mejor utilizar vistas en lugar de permitir a nadie acceder directamente a los datos, nosotros mostramos al resto de desarrolladores los datos que queremos.

Podemos llamarlas de forma sencilla en una consulta y utilizar clausulas contra las mismas, ahora veremos unos ejemplos.

Puede haber desarrolladores con poca experiencia y con dificultades por hacer complejas consultas, podemos darles la opción de simplemente llamar a una vista para así obtener los datos.

Una vista es un camino simple para guardar complejas consultas de selección en nuestra base de datos.

Una diferencia entre vistas y procedimientos almacenados es que las primeras no aceptan parámetros, no siendo así con los procedimientos almacenados, que si los aceptan.

Un procedimiento almacenado suele utilizarse cuando no es suficiente una simple consulta SQL. Los procedimientos almacenados contienen variables, bucles y llamadas a otros procedimientos almacenados.

Sintaxis para crear una vista.

```
1 CREATE VIEW
2 usuarios_comentarios AS
3 SELECT usuarios.id,nombre, rango, titulo, comentario
4 FROM usuarios
5 INNER JOIN comentarios
6 ON usuarios.id = comentarios.usuario_id;
```

Sintaxis para ejecutar la vista

```
1 SELECT * FROM usuarios_comentarios;
```

Descargo de responsabilidad

La información contenida en este documento descargable en formato PDF o PPT es un reflejo del material virtual presentado en la versión online del curso. Por lo tanto, su contenido, gráficos, links de consulta, acotaciones y comentarios son responsabilidad exclusiva de su(s) respectivo(s) autor(es) por lo que su contenido no compromete al área de e-Learning del Departamento GES o al programa académico al que pertenece.

El área de e-Learning no asume ninguna responsabilidad por la actualidad, exactitud, obligaciones de derechos de autor, integridad o calidad de los contenidos proporcionados y se aclara que la utilización de este descargable se encuentra limitada de manera expresa para los propósitos educacionales del curso.

