LAS CONSULTAS SUBACUÁTICAS

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 EmployeeID a, FirstName, (SELECT MIN(orderdate) FROM orders WHERE EmployeeId = a)

FROM Employees;

En este ejemplo la consulta principal es ***SELECT... FROM employees***.

La subconsulta es ***(SELECT MIN(orderdate) FROM orders EmployeeId = a)***.

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

¿Qué pasa cuando se ejecuta la consulta principal?

- Se coge el primer empleado y se calcula la subconsulta sustituyendo employeeID por el valor que tiene en el primer empleado. La subconsulta obtiene la fecha más antigua en los pedidos del empleado.
- Se coge el segundo empleado y se calcula la subconsulta con el employeeID 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 employeeID, FirstName

FROM employees

WHERE employeeID = (SELECT employeeID FROM orders WHERE customerID = (SELECT customerID FROM customers WHERE companyName = 'nombre empresa'))

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).

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.

Muchos SQLs 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 employees.employeeID, FirstName, MIN(orderdate)

FROM employees LEFT JOIN orders ON employees.employeeID = orders. employeeID

GROUP BY employees.employeeID, FirstName

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 vimos 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 employeeID, FirstName

FROM employees

WHERE hiredate = (SELECT MIN(orderdate) FROM orders)

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 de selección con subconsultas

Las condiciones de selección son las condiciones que pueden aparecer en la cláusula WHERE o HAVING.

En SQL tenemos cuatro nuevas condiciones:

- El test de comparación con subconsulta
- El test de comparación cuantificada
- El test de pertenencia a un conjunto
- El test de existencia

En todos los tests estudiados a continuación *expresión* puede ser cualquier nombre de columna de la consulta principal o una expresión válida.

El test de comparación con subconsulta.

Es el equivalente al test de comparación simple. Se utiliza para comparar un valor de la fila que se está examinado con un único valor producido por la subconsulta. La subconsulta debe devolver una única columna, sino se produce un error.

Si la subconsulta no produce ninguna fila o devuelve el valor nulo, el test devuelve el valor nulo, si la subconsulta produce varias filas, SQL devuelve una condición de error.

La sintaxis es la siguiente:

SELECT productID, productName
FROM products
WHERE supplierID = (
SELECT supplierID
FROM suppliers
WHERE suppliers.CompanyName = 'Exotic Liquids')

Lista los productos cuyo proveedor sea de Exotic Liquids

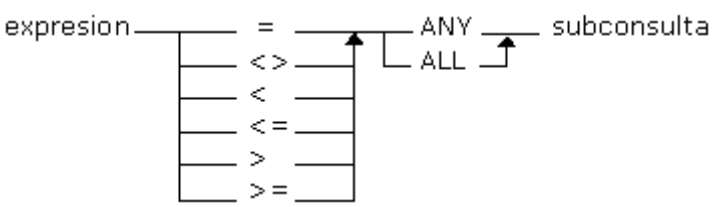
En este caso la subconsulta devuelve una única columna y una única fila (es un consulta de resumen sin GROUP BY)

El test de comparación cuantificada.

Este test es una extensión del test de comparación y del test de conjunto. Compara el valor de la expresión con cada uno de los valores producidos por la subconsulta. La subconsulta debe devolver una única columna sino se produce un error.

Tenemos el test ANY (algún, alguno en inglés) y el test ALL (todos en inglés).

La sintaxis es la siguiente:



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.

SELECT oficina, ciudad

FROM oficinas

WHERE objetivo > ANY (SELECT SUM(cuota) FROM empleados GROUP BY oficina)

En este caso la subconsulta devuelve una única columna con las sumas de las cuotas de los empleados de cada oficina.

Lista las oficinas cuyo objetivo sea superior a alguna de las sumas obtenidas.

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. (¡Ojo con esto!)

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)

En este caso se listan las oficinas cuyo objetivo sea superior a todas las sumas.

Test de pertenencia a conjunto (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 employees.employeeID, FirstName, territoryID

FROM employees JOIN EmployeeTerritories ON employees.EmployeeID=EmployeeTerritories.EmployeeID

WHERE EmployeeTerritories.territoryID IN (SELECT territoryID FROM territories WHERE territoryDescription = 'Atlanta')

Con la subconsulta se obtiene la lista de los territorios cuya descripción se Atlanta (En caso de que existiese más de uno) y la consulta principal obtiene los empleados cuyo territorio sea uno de los que tiene como descripción Atlanta

Por lo tanto lista los empleados de Atlanta

El test de existencia 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)

La sintaxis es la siguiente:

SELECT numemp, nombre, oficina

FROM empleados

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

Observa que delante de EXISTS no va ningún nombre de columna.

En la subconsulta se pueden poner las columnas que queramos en la lista de selección (hemos utilizado el *).

Hemos añadido una condición adicional al WHERE, la de la referencia externa para que la oficina que se compare sea la oficina del empleado.

NOTA. Cuando se trabaja con tablas muy voluminosas el test EXISTS suele dar mejor rendimiento que el test IN.