

Trabajo con Subconsultas

Contenido

Contenido

Introducción	1
Introducción a las subconsultas	2
Uso de una subconsulta como una tabla derivada	4
Uso de una subconsulta como una expresión	5
Uso de una subconsulta para correlacionar datos	6
Uso de las cláusulas EXISTS y NOT EXISTS	13

Notas para el instructor

Este módulo presenta técnicas avanzadas de consulta, que incluyen subconsultas anidadas y correlacionadas. En él se describe cuándo y cómo utilizar una subconsulta y el modo de usar subconsultas para dividir y realizar consultas complejas.

Al terminar este módulo, los alumnos serán capaces de:

- Describir cuándo y cómo utilizar subconsultas.
- Utilizar subconsultas para dividir y realizar consultas complejas.

Introducción

Objetivos de la diapositiva

Exponer un breve resumen de los temas tratados en este módulo.

Explicación previa

En este módulo aprenderá acerca de las técnicas avanzadas de consulta.

- **Introducción a las subconsultas**
- **Uso de una subconsulta como una tabla derivada**
- **Uso de una subconsulta como una expresión**
- **Uso de una subconsulta para correlacionar datos**
- **Uso de las cláusulas EXISTS y NOT EXISTS**

Este módulo presenta técnicas avanzadas de consulta, lo que incluye las subconsultas anidadas y correlacionadas, y cómo se pueden utilizar para modificar datos. En él se describe cuándo y cómo utilizar una subconsulta y el modo de usar subconsultas para dividir y realizar consultas complejas.

Al terminar este módulo, el alumno será capaz de:

- Describir cuándo y cómo utilizar subconsultas.
- Utilizar subconsultas para dividir y realizar consultas complejas.

Introducción a las subconsultas

Objetivos de la diapositiva

Comentar si se deben utilizar subconsultas.

Explicación previa

Las subconsultas son un conjunto de instrucciones SELECT. A menudo, las subconsultas se pueden escribir como combinaciones.

- **Por qué utilizar subconsultas**
 - Para dividir una consulta compleja en varios pasos lógicos
 - Para responder una consulta que depende de los resultados de otra consulta
- **Por qué utilizar combinaciones en lugar de subconsultas**
 - SQL Server ejecuta combinaciones más rápidas que la subconsultas
- **Cómo utilizar subconsultas**

Una *subconsulta* es una instrucción SELECT anidada en una instrucción SELECT, INSERT, UPDATE o DELETE, o en otra subconsulta. A menudo puede escribir las subconsultas como combinaciones y utilizarlas en lugar de una expresión.

Una *expresión* es una combinación de identificadores, valores y operadores que evalúa SQL Server para obtener un resultado.

Por qué utilizar subconsultas

Las subconsultas se utilizan para dividir una consulta compleja en varios pasos lógicos y, como resultado, resolver un problema con una única instrucción. Las subconsultas son útiles cuando la consulta depende de los resultados de otra consulta.

Por qué utilizar combinaciones en lugar de subconsultas

A menudo, una consulta que contiene subconsultas se puede escribir como una combinación. En general, el rendimiento de una consulta puede ser similar con una combinación y con una subconsulta. El optimizador de consultas optimiza habitualmente subconsultas mediante el uso del plan de ejecución de ejemplo que utilizaría una combinación semánticamente equivalente. La diferencia consiste en que la subconsulta puede requerir que el optimizador de consultas realice pasos adicionales, como ordenar, lo que puede influir en la estrategia del proceso.

Normalmente, utilizar combinaciones permite al optimizador de consultas recuperar datos de forma más eficiente. Si una consulta no requiere varios pasos, puede que no sea necesario utilizar una subconsulta.

Sugerencia

Revise cada hecho e instrucción para determinar cuándo usar subconsultas.

Cómo utilizar subconsultas

Cuando decida utilizar subconsultas, tenga en cuenta los siguientes hechos e instrucciones:

- Las subconsultas se deben incluir entre paréntesis.
- Se pueden utilizar subconsultas en lugar de una expresión siempre y cuando se devuelva un solo valor o una lista de valores. Se pueden utilizar subconsultas que devuelvan un conjunto de registros de varias columnas en lugar de una tabla o para realizar la misma función que una combinación.
- No se pueden utilizar subconsultas que recuperen columnas con tipos de datos **text** e **image**.
- Puede tener subconsultas dentro de subconsultas, con una anidación de hasta 32 niveles. El límite varía según la cantidad de memoria disponible y la complejidad de las otras expresiones de la consulta. Las consultas individuales pueden no admitir una anidación de hasta 32 niveles.

Uso de una subconsulta como una tabla derivada

Objetivos de la diapositiva

Describir cómo utilizar una subconsulta como una tabla derivada.

Explicación previa

Una tabla derivada se crea al utilizar una subconsulta en lugar de una tabla en una cláusula FROM:

- Es un conjunto de registros dentro de una consulta que funciona como una tabla
- Ocupa el lugar de la tabla en la cláusula FROM
- Se optimiza con el resto de la consulta

```
USE northwind
SELECT T.orderid, T.customerid
FROM ( SELECT orderid, customerid
        FROM orders ) AS T
GO
```

Una *tabla derivada* se crea al utilizar una subconsulta en lugar de una tabla en una cláusula FROM. Una tabla derivada es un uso especial de una subconsulta en una cláusula FROM a la que hace referencia un alias o nombre especificado por el usuario. El conjunto de resultados de la subconsulta en la cláusula FROM crea una tabla que utiliza la instrucción SELECT externa.

Ejemplo

En este ejemplo se utiliza una subconsulta para crear una tabla derivada en la parte interna de la consulta que consulta la parte externa. La tabla derivada propiamente dicha es funcionalmente equivalente a la consulta entera, pero aparece separada con propósitos ilustrativos.

```
USE northwind
SELECT T.orderid, T.customerid
FROM ( SELECT orderid, customerid
        FROM orders ) AS T
GO
```

Cuando se utilice como tabla derivada, tenga en cuenta que una subconsulta:

- Es un conjunto de registros dentro de una consulta que funciona como una tabla.
- Ocupa el lugar de la tabla en la cláusula FROM.
- Se optimiza con el resto de la consulta.

Uso de una subconsulta como una expresión

Objetivos de la diapositiva

Describir cómo usar una subconsulta como una expresión.

Explicación previa

Puede sustituir una subconsulta donde utilice una expresión en las instrucciones SELECT, UPDATE, INSERT y DELETE.

- Se evalúa y trata como una expresión
- Se ejecuta una vez para la instrucción entera

```
USE pubs
SELECT title, price
      ,( SELECT AVG(price) FROM titles) AS average
      ,price-(SELECT AVG(price) FROM titles) AS difference
FROM titles
WHERE type='popular_comp'
GO
```

Sugerencia

Señale que las subconsultas que devuelven una lista de valores sustituyen a una expresión en una cláusula WHERE que contiene la palabra clave IN.

En Transact-SQL, puede sustituir una subconsulta donde utilice una expresión. La subconsulta debe producir un valor escalar o una lista de valores de una sola columna. Las subconsultas que devuelven una lista de valores sustituyen a una expresión en una cláusula WHERE que contiene la palabra clave IN.

Si se utiliza como expresión, tenga en cuenta que una subconsulta:

- Se evalúa y trata como una expresión. Con frecuencia, el optimizador de consultas evalúa una expresión como equivalente a una combinación que conecta con una tabla que tiene una fila.
- Se ejecuta una vez para la instrucción entera.

Ejemplo

En este ejemplo se devuelve el precio de un popular libro de informática, el promedio del precio de todos los libros y la diferencia entre el precio del libro y el promedio del precio de todos los libros.

```
USE pubs
SELECT title, price
      ,(SELECT AVG(price) FROM titles) AS average
      ,price-(SELECT AVG(price) FROM titles) AS difference
FROM titles
WHERE type='popular_comp'
GO
```

◆ Uso de una subconsulta para correlacionar datos

Objetivos de la diapositiva

Describir cómo usar una subconsulta para correlacionar datos.

Explicación previa

Es posible utilizar una subconsulta correlacionada como una expresión dinámica que cambia en cada fila de una consulta externa.

- Evaluación de una subconsulta correlacionada
- Simulación de una cláusula JOIN
- Simulación de una cláusula HAVING

Puede utilizar una subconsulta correlacionada como una expresión dinámica que cambia en cada fila de una consulta externa.

El procesador de consultas realiza la subconsulta para cada fila de la consulta externa, una fila a la vez, que a su vez se evalúa como una expresión para esa fila y se pasa a la consulta externa. La subconsulta correlacionada es, de hecho, una COMBINACIÓN entre la subconsulta ejecutada dinámicamente y la fila de la consulta externa.

Normalmente, puede escribir una consulta de varias maneras y aun así obtener los mismos resultados. Las subconsultas correlacionadas dividen consultas complejas en dos o más consultas simples relacionadas.

Sugerencia Las subconsultas correlacionadas se pueden reconocer fácilmente. La columna de una tabla dentro de la subconsulta se compara con la columna de una tabla externa a la subconsulta.

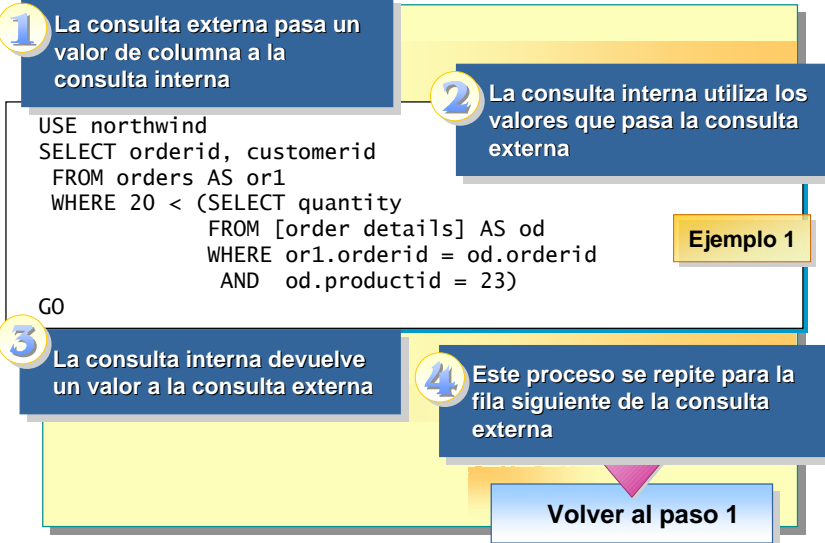
Evaluación de una subconsulta correlacionada

Objetivos de la diapositiva

Describir cómo se procesan las subconsultas correlacionadas.

Explicación previa

Cuando crea una subconsulta correlacionada, las subconsultas internas se evalúan repetidamente, una vez por cada fila de la consulta externa.



Sugerencia

Compare las subconsultas correlacionadas con las subconsultas anidadas.

Cuando crea una subconsulta correlacionada, las subconsultas internas se evalúan repetidamente, una vez por cada fila de la consulta externa:

- SQL Server ejecuta la consulta interna por cada fila que selecciona la consulta externa.
- SQL Server compara los resultados de la subconsulta con los resultados externos a ella.

Ejemplo 1

En este ejemplo se devuelve una lista de clientes que han pedido más de 20 unidades del producto número 23.

```
USE northwind
SELECT orderid, customerid
FROM orders AS or1
WHERE 20 < (SELECT quantity
            FROM [order details] AS od
            WHERE or1.orderid = od.orderid
            AND od.productid = 23)
GO
```

Resultado

Orderid	customerid
10337	FRANK
10348	WANDK
10396	FRANK
10402	ERNSH
10462	CONSH
.	.
.	.
.	.
(11 filas afectadas)	

Las subconsultas correlacionadas devuelven un único valor o una lista de valores por cada fila especificada en la cláusula FROM de la consulta externa. Los pasos siguientes describen cómo se evalúa la subconsulta correlacionada del ejemplo 1:

1. La consulta externa pasa un valor de columna a la consulta interna.

El valor de columna que la consulta externa pasa a la consulta interna es **orderid**. La consulta externa pasa el primer **orderid** de la tabla **orders** a la consulta interna.

2. La consulta interna utiliza los valores que pasa la consulta externa.

Cada **orderid** de la tabla **orders** se evalúa para determinar si existe un **orderid** idéntico en la tabla **order details**. Si el primer **orderid** coincide con un **orderid** de la tabla **order details** y ese **orderid** ha adquirido el producto número 23, la consulta interna devuelve ese **orderid** a la consulta externa.

3. La consulta interna devuelve un valor a la consulta externa.

La cláusula WHERE de la consulta externa evalúa posteriormente el **orderid** que adquirió el producto número 23 para determinar si la cantidad pedida es mayor de 20.

4. Este proceso se repite para la fila siguiente de la consulta externa.

La consulta externa pasa el segundo **orderid** de la tabla **orders** a la consulta interna y SQL Server repite el proceso de evaluación para esa fila.

Ejemplo 2

En este ejemplo se devuelve una lista de productos y el pedido mayor realizado hasta la fecha de cada producto de la tabla **order details**. Observe que esta subconsulta correlacionada hace referencia a la misma tabla que la consulta externa; generalmente, el optimizador tratará esto como una autocombinación.

```
USE northwind
SELECT DISTINCT productid, quantity
FROM [order details] AS ord1
WHERE quantity = ( SELECT MAX(quantity)
                  FROM [order details] AS ord2
                  WHERE ord1.productid = ord2.productid)
GO
```

Resultado

productid	quantity
50	40
67	40
4	50
9	50
11	50
.	
.	
.	

(77 filas afectadas)

Simulación de una cláusula JOIN

Objetivos de la diapositiva

Describir cómo usar una subconsulta correlacionada para simular una combinación.

Explicación previa

Puede utilizar una subconsulta correlacionada para producir los mismos resultados que una combinación.

- Las subconsultas correlacionadas pueden producir el mismo resultado que una cláusula JOIN
- Las combinaciones permiten al optimizador de consultas determinar la manera de correlacionar los datos de la forma más eficiente

Ejemplo 1

```
USE pubs
SELECT DISTINCT t1.type
  FROM titles AS t1
 WHERE t1.type IN
       (SELECT t2.type
        FROM titles AS t2
        WHERE t1.pub_id <> t2.pub_id)
GO
```

Sugerencia

La clave para comprender la sintaxis de las subconsultas correlacionadas es entender el uso de los alias de tabla. Los alias de tabla muestran las tablas que están correlacionadas.

Puede utilizar una subconsulta correlacionada para producir los mismos resultados que una combinación, por ejemplo, seleccionar datos de una tabla a la que se hizo referencia en la consulta externa.

Nota Normalmente, las subconsultas correlacionadas se pueden escribir como combinaciones. El uso de combinaciones en lugar de subconsultas correlacionadas permite al optimizador de consultas determinar la manera de correlacionar los datos de la forma más eficiente.

Ejemplo 1

En este ejemplo se utiliza una subconsulta correlacionada para buscar los tipos de libros publicados por más de un editor. Para evitar ambigüedades, se necesitan alias para distinguir las dos funciones diferentes en las que aparece la tabla **titles**.

```
USE pubs
SELECT DISTINCT t1.type
  FROM titles AS t1
 WHERE t1.type IN
       (SELECT t2.type
        FROM titles AS t2
        WHERE t1.pub_id <> t2.pub_id)
GO
```

Resultado

Type

business
psychology

(2 filas afectadas)

Ejemplo 2

En este ejemplo se devuelven los mismos resultados que en el ejemplo 1 mediante el uso de una autocombinación en lugar de una subconsulta correlacionada.

```
USE pubs
SELECT DISTINCT t1.type
  FROM titles AS t1
  INNER JOIN titles AS t2
    ON t1.type = t2.type
 WHERE t1.pub_id <> t2.pub_id
GO
```

Sugerencia

Utilice el Analizador de consultas SQL para ejecutar ambos ejemplos de combinación y mostrar los distintos planes de ejecución.

Simulación de una cláusula HAVING

Objetivos de la diapositiva

Describir cómo simular una cláusula HAVING.

Explicación previa

Puede utilizar una subconsulta correlacionada para producir los mismos resultados que una consulta que utiliza la cláusula HAVING.

■ Subconsulta con el mismo resultado que una cláusula HAVING

```
USE pubs
SELECT t1.type, t1.title, t1.price
FROM titles AS t1
WHERE t1.price > ( SELECT AVG(t2.price) FROM titles AS t2
                  WHERE t1.type = t2.type )
GO
```

Ejemplo 1

■ Uso de una cláusula HAVING sin una subconsulta

```
USE pubs
SELECT t1.type, t1.title, t1.price
FROM titles AS t1
INNER JOIN titles AS t2 ON t1.type = t2.type
GROUP BY t1.type, t1.title, t1.price
HAVING t1.price > AVG(t2.price)
GO
```

Ejemplo 2

Puede utilizar una subconsulta correlacionada para producir los mismos resultados que una consulta que utiliza la cláusula HAVING.

Ejemplo 1

En este ejemplo se buscan todos los títulos con un precio mayor que el precio promedio para libros del mismo tipo. Por cada valor posible de **t1**, SQL Server evalúa la subconsulta e incluye la fila en los resultados si el precio de dicha fila es mayor que el promedio calculado. No es necesario agrupar explícitamente por tipo, puesto que las filas cuyo precio promedio se calcula están restringidas por la cláusula WHERE de la subconsulta.

Sugerencia

Utilice el Analizador de consultas SQL para ejecutar ambos ejemplos y comprobar que producen los mismos resultados.

```
USE pubs
SELECT t1.type, t1.title, t1.price
FROM titles AS t1
WHERE t1.price > ( SELECT AVG(t2.price)
                  FROM titles AS t2
                  WHERE t1.type = t2.type )
GO
```

Resulttype	title
Business	The Busy Executive's Database Guide
Business	Straight Talk About Computers
mod_cook	Silicon Valley Gastronomic Treats
popular_comp	But Is It User Friendly?
Psychology	Computer Phobic AND Non-Phobic Individuals: Behavior Variations
Psychology	Prolonged Data Deprivation: Four Case Studies
trad_cook	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean

(7 filas afectadas)

Ejemplo 2

Este ejemplo produce el mismo conjunto de resultados que el ejemplo 1, pero utiliza una autocombinación con las cláusulas GROUP BY y HAVING.

```
USE pubs
SELECT t1.type, t1.title, t1.price
  FROM titles AS t1
  INNER JOIN titles AS t2
    ON t1.type = t2.type
  GROUP BY t1.type, t1.title, t1.price
  HAVING t1.price > AVG(t2.price)
GO
```

Nota Puede escribir subconsultas correlacionadas que produzcan los mismos resultados que una cláusula JOIN o HAVING, pero es posible que el procesador de consultas no las implemente de la misma manera.

Uso de las cláusulas EXISTS y NOT EXISTS

Objetivos de la diapositiva

Describir cómo se utilizan los operadores EXISTS y NOT EXISTS con las subconsultas correlacionadas.

Explicación previa

Los operadores EXISTS y NOT EXISTS se pueden utilizar para determinar si hay datos en una lista de valores.

- Uso con subconsultas correlacionadas
- Determinar si hay datos en una lista de valores
- Proceso de SQL Server
 - La consulta externa prueba la existencia de las filas
 - La consulta interna devuelve TRUE o FALSE
 - No se produce ningún dato

Ejemplo 1

```
USE northwind
SELECT lastname, employeeid
FROM employees AS e
WHERE EXISTS (SELECT * FROM orders AS o
              WHERE e.employeeid = o.employeeid
              AND o.orderdate = '5/9/1997')
GO
```

Los operadores EXISTS y NOT EXISTS se pueden utilizar para determinar si hay datos en una lista de valores.

Uso con subconsultas correlacionadas

Puede utilizar los operadores EXISTS y NOT EXISTS con subconsultas correlacionadas para restringir el conjunto de resultados de una consulta externa a las filas que cumplen la subconsulta. Los operadores EXISTS y NOT EXISTS devuelven TRUE o FALSE, en función de si las subconsultas devuelven filas o no.

Determinar si hay datos en una lista de valores

Cuando se introduce una subconsulta con el operador EXISTS, SQL Server comprueba si hay datos que coincidan con la subconsulta. No se recupera realmente ninguna fila. SQL Server termina la recuperación de filas cuando ha comprobado que al menos una fila cumple la condición WHERE en la subconsulta.

Proceso de SQL Server

Cuando SQL Server procesa subconsultas que utilizan el operador EXISTS o NOT EXISTS:

- La consulta externa prueba que existen las filas que devuelve la subconsulta.
- La subconsulta devuelve un valor TRUE o FALSE en función de la condición dada en la consulta.
- La subconsulta no produce ningún dato.

Sintaxis parcial

WHERE [NOT] EXISTS (*subconsulta*)

Ejemplo 1

En este ejemplo se utiliza una subconsulta correlacionada con un operador EXISTS en la cláusula WHERE para devolver una lista de empleados que anotaron pedidos el día 5/9/1997.

Sugerencia

Ejecute estos dos ejemplos con STATISTICS TIME establecido en ON para comparar el tiempo de proceso.

```
USE northwind
SELECT lastname, employeeid
FROM employees AS e
WHERE EXISTS ( SELECT * FROM orders AS o
              WHERE e.employeeid = o.employeeid
                AND o.orderdate = '5/9/1997' )
GO
```

Resultado

lastname	employeeid
Peacock	4
King	7

(2 filas afectadas)

Ejemplo 2

Este ejemplo devuelve el mismo conjunto de resultados del ejemplo 1 e ilustra que podría utilizar una operación de combinación en lugar de una subconsulta correlacionada. Observe que la consulta necesita la palabra clave DISTINCT para devolver únicamente una sola fila por cada empleado.

```
USE northwind
SELECT DISTINCT lastname, e.employeeid
FROM orders AS o
INNER JOIN employees AS e
ON o.employeeid = e.employeeid
WHERE o.orderdate = '5/9/1997'
GO
```

Resultado

lastname	employeeid
Peacock	4
King	7

(2 filas afectadas)