

Aplicações com Banco de Dados

Aula 02

14/08/2024

Conteúdo da aula

- Subconsultas

(obs.: todos os exemplos da aula, utilizam o database Aventuras)

Subconsultas

Uma **subconsulta** ou **subquery** é uma instrução Select (consulta) dentro (aninhada) de outra instrução SELECT, INSERT, UPDATE OU DELETE.

As subqueries podem ser codificadas ou mesmo introduzidas em cláusulas where, having, from, ou mesmo um select de uma outra instrução select.

Subconsulta na cláusula Select

```
SELECT nomecontato, (SELECT tipoCliente FROM [Clientes-Tipo]  
WHERE Clientes.idtipo = [clientes-Tipo].idTipo) AS Tipo_Cliente  
FROM Clientes
```

Neste exemplo, a subquery está sendo utilizada na cláusula Select, no lugar da especificação de uma coluna, por isso, uma **subquery deve retornar um único valor**. Na maioria dos casos, as subconsultas utilizadas na cláusula select, são subconsultas correlacionadas (são consultas que dependem e fazem referência as colunas de consultas externas a qual estão contidas).

Subconsulta na cláusula Select

Na maioria dos casos, é possível substituir a subquery por um Join, para facilitar a leitura e executar mais rapidamente.

```
SELECT c.nomecontato, tc.tipocliente  
FROM Clientes c  
inner join [Clientes-tipo] tc on c.idtipo = tc.idtipo
```

Subconsulta na cláusula Select

Quando uma subconsulta retorna um único valor, ela pode ser utilizada em qualquer lugar, substituindo uma expressão. Mas uma subconsulta também pode retornar um conjunto de resultados, e neste caso ela pode ser utilizada no lugar de uma lista de valores, como a lista para um operador **IN**.

SELECT nomecontato

FROM Clientes

where idTipo **in** (SELECT idTipo FROM [Clientes-Tipo] where
tipoCliente = 'Cliente Ouro' or tipoCliente = 'Cliente Prata')

Subconsulta na cláusula Select

Também é possível codificar uma subconsulta dentro de outra subconsulta, as chamadas subconsultas aninhadas (difícil leitura e baixo desempenho)

```
select cidadeDestinatario
```

```
from Ordens
```

```
where IDCliente in (select idcliente from clientes where idtipo in  
(SELECT idTipo FROM [Clientes-Tipo] where tipoCliente = 'Cliente  
Ouro'))
```

Prática: Faça você mesmo

Utilizando subconsulta, elabore uma consulta que retorne o nome do produto e o nome do contato do fornecedor deste produto

Subconsulta na cláusula Select

Existem três tipos de subqueries:

- a) Subquery single-row (subconsulta de uma única linha)
- b) Subquery multiple-row (subconsulta de várias linhas)
- c) Subquery multiple columns (subconsulta de múltiplas colunas)

Subquery Single-row

Este tipo de subconsulta pode retornar zero ou uma linha para a instrução SQL exterior. Neste tipo de consulta podemos utilizar os operadores necessários para retornar as informações solicitadas em uma única linha.

SELECT nomecontato

FROM Clientes **where** idTipo = (SELECT idTipo FROM [Clientes-Tipo] **where** tipoCliente = 'Cliente Premiun')

(No exemplo acima, foi utilizado o operador de igualdade (=), mas também é possível utilizar os operadores : maior que (>), menor que (<), menor ou igual (<=), diferente (<>)

Subquery Single-row

Subquery na cláusula From

```
select * from (select idordem from ordens where frete > 50) as  
teste
```

Subquery Single-row **

Subquery com Having

Having é para filtrar grupos de linhas de registros (group by). Isto permite filtrar grupos de linhas com base no resultado retornado por uma subquery

```
select IDCliente, Pais, count(*) total
from Clientes
group by IDCliente, pais
having IDCliente = (select IDCliente from Ordens where
DATEDIFF(day,DataOrdem,GETDATE()) > 8802)
```

Subquery Single-row

se colocássemos:

```
select IDCliente, Pais, count(*) total
```

```
from Clientes
```

```
group by IDCliente, pais
```

```
having IDCliente = (select IDCliente from Ordens where  
DATEDIFF(day,DataOrdem,GETDATE()) > 1000)
```

Ocorreria o erro abaixo, pois a subquery retorna mais de um valor:

Mensagem 512, Nível 16, Estado 1, Linha 2 A subconsulta retornou mais de 1 valor. Isso não é permitido quando a subconsulta segue um =, !=, <= , >, >= ou quando ela é usada como uma expressão.

Subquery Multiple row

Quando é necessário realizar consultas onde o resultado seja retornado em várias linhas, pode-se utilizar os operadores **IN** ou **ANY** nas consultas externas para lidar com uma subconsulta que retorne várias linhas

Subquery Multiple row

O operador IN verifica um valor dentro de um conjunto de valores (executando o exemplo anterior que apresentou erro, e substituindo o operador (=) por IN, o erro não ocorre mais)

```
select IDCliente, Pais, count(*) total
from Clientes
group by IDCliente, pais
having IDCliente IN (select IDCliente from Ordens where
DATEDIFF(day,DataOrdem,GETDATE()) > 1000)
```

Subquery Multiple row

Se usarmos o NOT IN, os resultados serão os valores que não estejam contidos na subquery

```
select IDCliente, Pais, count(*) total
from Clientes
group by IDCliente, pais
having IDCliente NOT IN (select IDCliente from Ordens where
DATEDIFF(day,DataOrdem,GETDATE()) > 1000)
```


Subquery Multiple row

Exemplo com **ANY**

```
select IDCliente, Pais, count(*) total
from Clientes
group by IDCliente, pais
having IDCliente = any (select IDCliente from Ordens where
DATEDIFF(day,DataOrdem,GETDATE()) > 1000)
```

ANY: retorna um valor booleano “True” ou “False”. Os valores para comparação sempre devem ser baseados em uma única coluna

Subquery Multiple Column

Consulta que será realizada utilizando-se de duas ou mais colunas. No exemplo , usamos o Exists como um teste de existência. A clausula where presente na consulta externa realiza um teste que verifica se as linhas retornadas pela subconsulta existem.

SELECT nomecontato

FROM Clientes

where exists (SELECT * FROM [Clientes-Tipo] where tipoCliente in ('Cliente Ouro', 'Cliente Prata'))

Exemplos de uso de subquery

Exemplo 1

```
SELECT nomeContato FROM clientes c WHERE c.idcliente IN  
(SELECT IDCliente FROM Ordens o  
INNER JOIN [Ordens-Detalhes] od ON o.IDOrdem=od.IDOrdem  
WHERE od.Desconto>0)
```

Exemplos de uso de subquery

Exemplo 2

```
insert into [Clientes-Tipo] values (4, 'Cliente teste')
```

```
DELETE FROM [Clientes-Tipo]  
WHERE idtipo not in  
(SELECT idTipo FROM Clientes)
```

Exemplos de uso de subquery

Exemplo 3

```
insert into [Clientes-Tipo] values (4, 'Cliente teste')

update Clientes set idTipo = 4
WHERE IDCliente in
(SELECT IDCliente FROM Ordens o
INNER JOIN [Ordens-Detalhes] od ON o.IDOrdem=od.IDOrdem
WHERE od.Desconto=0)
```

Exemplos de uso de subquery

Exemplo 4

```
create table [clientes-tipo 2] (  
  idtipo int,  
  TipoCliente varchar(50)  
)
```

```
insert into [clientes-tipo 2] select * from [Clientes-Tipo]
```

Exercícios

- 1) Liste o nome do funcionário mais velho da região Eastern
- 2) Liste a descrição do tipo de cliente com o total de compras maior que a média de todas as vendas realizadas
- 3) Liste apenas funcionários (nome e sobrenome) que já realizaram mais de 15 vendas
- 4) Exclua todos os clientes que tenham a classificação de “Cliente bronze”

Exercícios

- 1) Liste o nome do funcionário mais velho da região Eastern
- 2) Liste a descrição do tipo de cliente com o total de compras maior que a média de todas as vendas realizadas
- 3) Liste apenas funcionários (nome e sobrenome) que já realizaram mais de 15 vendas
- 4) Exclua todos os clientes que tenham a classificação de “Cliente bronze”

Referências

ELMASRI, Ramez. Sistemas de banco de dados. Tradução Marília Guimarães Pinheiro. 4. ed. São Paulo: Pearson Addison Wesley, 2006.