



Programação de Banco de Dados com SQL

10-3

Subconsultas Multilinha



Objetivos

Esta lição abrange os seguintes objetivos:

- Usar corretamente os operadores de comparação IN, ANY e ALL em subconsultas multilinha
- Construir e executar uma subconsulta multilinha na cláusula WHERE ou HAVING
- Descobrir o que acontece se uma subconsulta multilinha retorna um valor nulo
- Compreender quando as subconsultas multilinha devem ser usadas e quando é seguro utilizar uma subconsulta de linha única
- Distinguir entre subconsultas do tipo pair-wise e non-pair-wise



Finalidade

- A subconsulta foi feita para encontrar as informações que você não sabe para descobrir as informações que você quer saber.
- No entanto, as subconsultas de linha única conseguem retornar apenas uma linha. E se você precisar encontrar informações baseadas em várias linhas e valores?
- A subconsulta terá de retornar várias linhas.
- Conseguimos isso usando subconsultas multilinha e os três operadores de comparação: IN, ANY e ALL.

Comparação de Consultas

- Quem recebe um salário semelhante ao de um funcionário no departamento 20?
- O exemplo abaixo retorna um erro porque existe mais de um funcionário no departamento 20. A subconsulta retorna várias linhas.
- Chamamos isso de subconsulta multilinha.

LAST_NAME	DEPT_ID	SALARY
Hartstein	20	13000
Fay	20	6000

```
SELECT first_name, last_name
FROM employees
WHERE salary =
  (SELECT salary
   FROM employees
   WHERE department_id = 20);
```



ORA-01427: single-row subquery returns more than one row
(a subconsulta de linha única retornou mais de uma linha)

Comparação de Consultas

- O problema é o sinal de igual (=) na cláusula WHERE da consulta externa.
- Como um valor pode ser igual a (ou diferente de) mais de um valor ao mesmo tempo?
- É uma pergunta boba, não é?

```
SELECT first_name, last_name
FROM employees
WHERE salary =
  (SELECT salary
   FROM employees
   WHERE department_id = 20);
```



ORA-01427: single-row subquery returns more than one row
(a subconsulta de linha única retornou mais de uma linha)

IN, ANY e ALL

- As subconsultas que retornam mais de um valor são chamadas de multilinha.
- Como não podemos usar os operadores de comparação de linha única (=, < etc.), precisamos de operadores de comparação diferentes para as subconsultas multilinha.
- Os operadores multilinha são:
 - IN
 - ANY
 - ALL
- O operador NOT pode ser usado com qualquer um desses três operadores.

IN

- O operador IN é usado na cláusula WHERE da consulta externa para selecionar somente as linhas que estejam na lista de valores que retornaram da consulta interna.
- Por exemplo, estamos interessados em todos os funcionários que foram contratados no mesmo ano que um funcionário no departamento 90.

```
SELECT last_name, hire_date
FROM employees
WHERE EXTRACT(YEAR FROM hire_date) IN
      (SELECT EXTRACT(YEAR FROM hire_date)
       FROM employees
       WHERE department_id=90);
```

LAST_NAME	HIRE_DATE
King	17-Jun-1987
Kochhar	21-Sep-1989
De Haan	13-Jan-1993
Whalen	17-Sep-1987

IN

- A consulta interna retornará uma lista dos anos em que os funcionários do departamento 90 foram contratados.
- Em seguida, a consulta externa retornará os funcionários contratados no mesmo ano que qualquer um presente na lista da consulta interna.

```
SELECT last_name, hire_date
FROM employees
WHERE EXTRACT(YEAR FROM hire_date) IN
      (SELECT EXTRACT(YEAR FROM hire_date)
       FROM employees
       WHERE department_id=90);
```

LAST_NAME	HIRE_DATE
King	17-Jun-1987
Kochhar	21-Sep-1989
De Haan	13-Jan-1993
Whalen	17-Sep-1987

ANY

- Usamos o operador ANY quando queremos que a cláusula WHERE da consulta externa selecione as linhas que correspondem aos critérios (<, >, = etc.) de pelo menos um valor no conjunto de resultados da subconsulta.
- O exemplo mostrado retornará qualquer funcionário cujo ano de contratação é inferior a pelo menos um ano de contratação dos funcionários no departamento 90.



Year Hired
1987
1989
1993

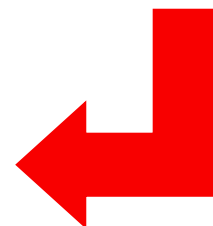
```
SELECT last_name, hire_date
FROM employees
WHERE EXTRACT(YEAR FROM hire_date) < ANY
      (SELECT EXTRACT(YEAR FROM hire_date)
       FROM employees
       WHERE department_id=90);
```

LAST_NAME	HIRE_DATE
King	17-Jun-1987
Kochhar	21-Sep-1989
Whalen	17-Sep-1987
Hunold	03-Jan-1990
Ernst	21-May-1991

ALL

- Usamos o operador ALL quando queremos que a cláusula WHERE da consulta externa selecione as linhas que correspondem aos critérios (<, >, = etc.) de todos os valores no conjunto de resultados da subconsulta.
- O operador ALL compara um valor a todos os valores retornados pela consulta interna.
- Como nenhum funcionário foi contratado antes de 1987, nenhuma linha é retornada.

```
SELECT last_name, hire_date
FROM employees
WHERE EXTRACT(YEAR FROM hire_date) < ALL
      (SELECT EXTRACT(YEAR FROM hire_date)
       FROM employees
       WHERE department_id=90);
```



Year Hired
1987
1989
1993

nenhum dado encontrado

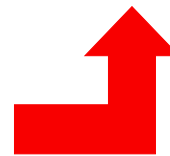
Valores Nulos

- Vamos supor que um dos valores retornados por uma subconsulta multilinha seja nulo, mas os outros não sejam.
- Se IN ou ANY forem usados, a consulta externa retornará linhas que correspondam aos valores não nulos.

```
SELECT last_name, employee_id
FROM employees
WHERE employee_id IN
  (SELECT manager_id
   FROM employees);
```

MANAGER_ID
-
100
100
101
101
205
100
...

LAST_NAME	EMPLOYEE_ID
King	100
Kochhar	101
De Haan	102
Higgins	205
...	



Resultado da subconsulta

Valores Nulos

- Se ALL for usado, a consulta externa não retornará linhas porque ALL compara a linha da consulta externa com todos os valores retornados pela subconsulta, incluindo o valor nulo.
- E a comparação de qualquer elemento com resultados nulos é nula.

```
SELECT last_name, employee_id
FROM employees
WHERE employee_id <= ALL
  (SELECT manager_id
   FROM employees);
```

nenhum dado encontrado

GROUP BY e HAVING

- Como você já deve estar pensando, as cláusulas GROUP BY e HAVING também podem ser usadas com subconsultas multilinha.
- E se você quisesse encontrar os departamentos cujo mínimo salário fosse inferior ao salário de qualquer funcionário que trabalha no departamento 10 ou 20?

LAST_NAME	DEPT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000

DEPARTMENT_ID	MIN(SALARY)
10	4400
20	6000
50	2500
60	4200
80	8600
110	8300
(nulo)	7000

GROUP BY e HAVING

- Precisamos de uma subconsulta multilinha que retorne os salários dos funcionários nos departamentos 10 e 20.
- A consulta externa usará uma função de grupo (MIN). Portanto, precisamos agrupar a consulta externa por department_id.

LAST_NAME	DEPT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000

DEPARTMENT_ID	MIN(SALARY)
10	4400
20	6000
50	2500
60	4200
80	8600
110	8300
(nulo)	7000

GROUP BY e HAVING

- Eis a instrução SQL:

```
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) < ANY
      (SELECT salary
       FROM employees
       WHERE department_id IN (10,20))
ORDER BY department_id;
```

LAST_NAME	DEPT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000

Resultado da subconsulta

DEPARTMENT_ID	MIN(SALARY)
10	4400
20	6000
50	2500
60	4200
80	8600
110	8300
(nulo)	7000

Subconsultas de Várias Colunas

- As subconsultas podem usar uma ou mais colunas.
- Se usam mais de uma coluna, elas são chamadas de subconsultas de várias colunas.
- Uma subconsulta de várias colunas podem ser comparações do tipo pair-wise e non-pair-wise.

```
SELECT employee_id, manager_id, department_id
FROM employees
WHERE(manager_id,department_id) IN
  (SELECT manager_id,department_id
   FROM employees
   WHERE employee_id IN (149,174))
AND employee_id NOT IN (149,174)
```

EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
176	149	80

Subconsultas de Várias Colunas

- O exemplo a seguir mostra uma subconsulta pair-wise de várias colunas realçada em vermelho e o resultado na tabela logo abaixo.
- A consulta lista os funcionários cujos gerente e departamentos são os mesmos que os gerente e departamentos dos funcionários 149 ou 174.

```
SELECT employee_id, manager_id, department_id
FROM employees
WHERE(manager_id,department_id) IN
  (SELECT manager_id,department_id
    FROM employees
    WHERE employee_id IN (149,174))
AND employee_id NOT IN (149,174)
```

EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
176	149	80

Subconsultas de Várias Colunas

- Uma subconsulta de várias colunas non-pair-wise também usa mais de uma coluna, mas as compara separadamente, portanto, as comparações acontecem em subconsultas diferentes.

```
SELECT  employee_id,  
        manager_id,  
        department_id  
FROM    employees  
WHERE   manager_id IN  
        (SELECT  manager_id  
         FROM    employees  
         WHERE   employee_id IN  
                (149,174))  
AND     department_id IN  
        (SELECT  department_id  
         FROM    employees  
         WHERE   employee_id IN  
                (149,174))  
AND     employee_id NOT IN(149,174);
```

EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
176	149	80

Subconsultas de Várias Colunas

- Você precisará gravar uma subconsulta por coluna que quiser comparar quando estiver executando subconsultas de várias colunas non-pair-wise.
- O exemplo à direita mostra uma subconsulta non-pair-wise de várias colunas com as subconsultas realçadas em vermelho.

```
SELECT  employee_id,  
        manager_id,  
        department_id  
FROM    employees  
WHERE   manager_id IN  
        (SELECT  manager_id  
         FROM    employees  
         WHERE   employee_id IN  
                (149,174))  
AND     department_id IN  
        (SELECT  department_id  
         FROM    employees  
         WHERE   employee_id IN  
                (149,174))  
AND     employee_id NOT IN(149,174);
```

EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
176	149	80

Subconsultas de Várias Colunas

- Esta consulta lista os funcionários que têm o mesmo `manager_id` e `department_id` que os funcionários 149 ou 174.

Resultado da primeira subconsulta

MANAGER_ID
100
149

Resultado da segunda subconsulta

DEPARTMENT_ID
80
80

```
SELECT  employee_id,  
        manager_id,  
        department_id  
FROM    employees  
WHERE   manager_id IN  
        (SELECT  manager_id  
         FROM    employees  
         WHERE   employee_id IN  
                (149,174))  
AND     department_id IN  
        (SELECT  department_id  
         FROM    employees  
         WHERE   employee_id IN  
                (149,174))  
AND     employee_id NOT IN(149,174);
```

EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
176	149	80

Uma Última Consideração sobre Subconsultas

- Algumas subconsultas podem retornar uma ou várias linhas, dependendo dos valores de dados nas linhas.
- Se houver a menor possibilidade de que sejam retornadas várias linhas, certifique-se de gravar uma subconsulta multilinha.

```
SELECT first_name, last_name, job_id
FROM employees
WHERE job_id =
  (SELECT job_id
   FROM employees
   WHERE last_name = 'Ernst');
```

FIRST_NAME	LAST_NAME	JOB_ID
Alexander	Hunold	IT_PROG
Bruce	Ernst	IT_PROG
Diana	Lorentz	IT_PROG

FIRST_NAME	LAST_NAME	JOB_ID
Bruce	Ernst	IT_PROG

Resultado da subconsulta

Uma Última Consideração sobre Subconsultas

- Por exemplo: Quem tem o mesmo job_id que Ernst?
- A subconsulta de linha única funciona corretamente porque há apenas um Ernst na tabela.
- Mas e se depois a empresa contratar uma nova funcionária chamada Susan Ernst?

```
SELECT first_name, last_name, job_id
FROM employees
WHERE job_id =
  (SELECT job_id
   FROM employees
   WHERE last_name = 'Ernst');
```

FIRST_NAME	LAST_NAME	JOB_ID
Alexander	Hunold	IT_PROG
Bruce	Ernst	IT_PROG
Diana	Lorentz	IT_PROG

FIRST_NAME	LAST_NAME	JOB_ID
Bruce	Ernst	IT_PROG

Resultado da subconsulta

Uma Última Consideração sobre Subconsultas

- Seria melhor gravar uma subconsulta multilinha.
- A sintaxe da subconsulta multilinha funcionará mesmo que a subconsulta retorne uma única linha.
- Se estiver na dúvida, grave uma subconsulta multilinha!

```
SELECT first_name, last_name, job_id
FROM employees
WHERE job_id IN
  (SELECT job_id
   FROM employees
   WHERE last_name = 'Ernst');
```

FIRST_NAME	LAST_NAME	JOB_ID
Alexander	Hunold	IT_PROG
Bruce	Ernst	IT_PROG
Diana	Lorentz	IT_PROG

FIRST_NAME	LAST_NAME	JOB_ID
Bruce	Ernst	IT_PROG

Resultado da subconsulta
Há duas pessoas com o sobrenome "Ernst"

Terminologia

Estes são os principais termos usados nesta lição:

- Subconsulta de várias colunas non-pair-wise
- Subconsulta de várias colunas pair-wise

Resumo

Nesta lição, você deverá ter aprendido a:

- Usar corretamente os operadores de comparação IN, ANY e ALL em subconsultas multilinha
- Construir e executar uma subconsulta multilinha na cláusula WHERE ou HAVING
- Descobrir o que acontece se uma subconsulta multilinha retorna um valor nulo
- Compreender quando as subconsultas multilinha devem ser usadas e quando é seguro utilizar uma subconsulta de linha única
- Distinguir entre subconsultas do tipo pair-wise e non-pair-wise

ORACLE®

Academy

DPS10L3
Subconsultas Multilinha

Copyright © 2019, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

26

