



Programação de Banco de Dados com SQL

10-4

Subconsultas Correlacionadas



Objetivos

Esta lição abrange os seguintes objetivos:

- Identificar quando subconsultas correlacionadas são necessárias.
- Construir e executar subconsultas correlacionadas.
- Criar uma consulta usando os operadores EXISTS e NOT EXISTS para testar as linhas que retornam da subconsulta
- Construir e executar subconsultas nomeadas usando a cláusula WITH.

Finalidade

- Às vezes, você precisa responder a mais de uma pergunta em uma frase.
- Um amigo seu pode lhe perguntar se você tem dinheiro para ir ao cinema e comprar pipoca e um refrigerante.
- Antes de dar a resposta, você precisa saber o preço do ingresso, da pipoca e do refrigerante.
- Você também precisa ver quanto dinheiro tem no bolso.
- Então, na verdade, o que parecia ser uma pergunta fácil se torna quatro perguntas para as quais você precisa de respostas antes de poder dizer "Sim" ou "Não".

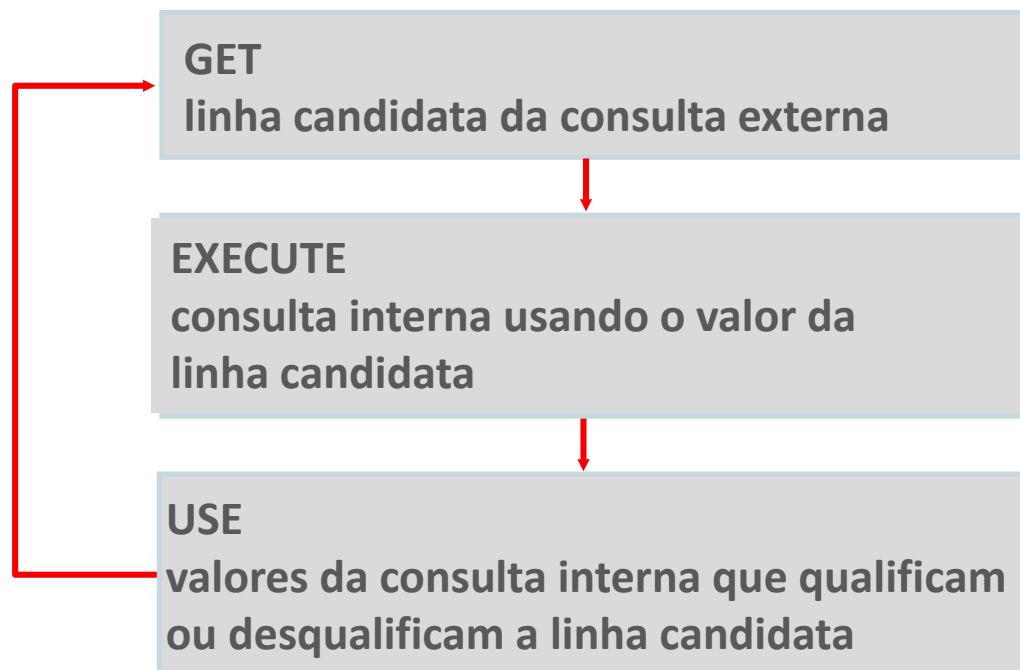
Finalidade

- Em um ambiente de negócios, é possível que lhe peçam para fazer um relatório com todos os funcionários que ganham mais do que a média salarial de seus departamentos.
- Então, primeiro, você precisa calcular a média salarial por departamento e, em seguida, comparar o salário de cada funcionário à média do departamento dele.



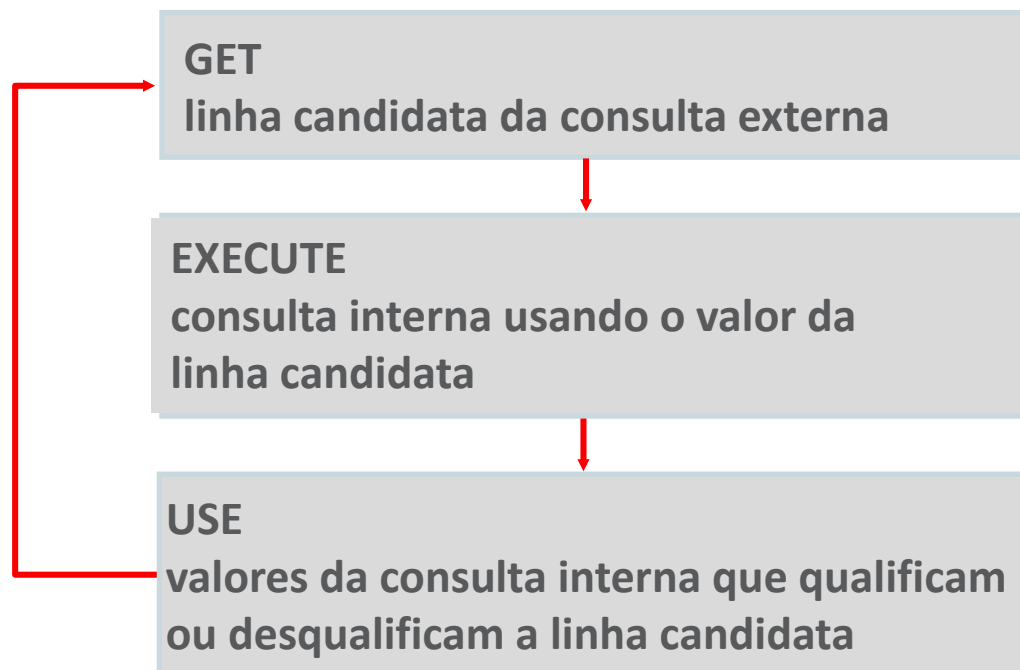
Subconsultas Correlacionadas

- O servidor Oracle faz uma subconsulta correlacionada quando a subconsulta referencia uma coluna de uma tabela referenciada na instrução pai.



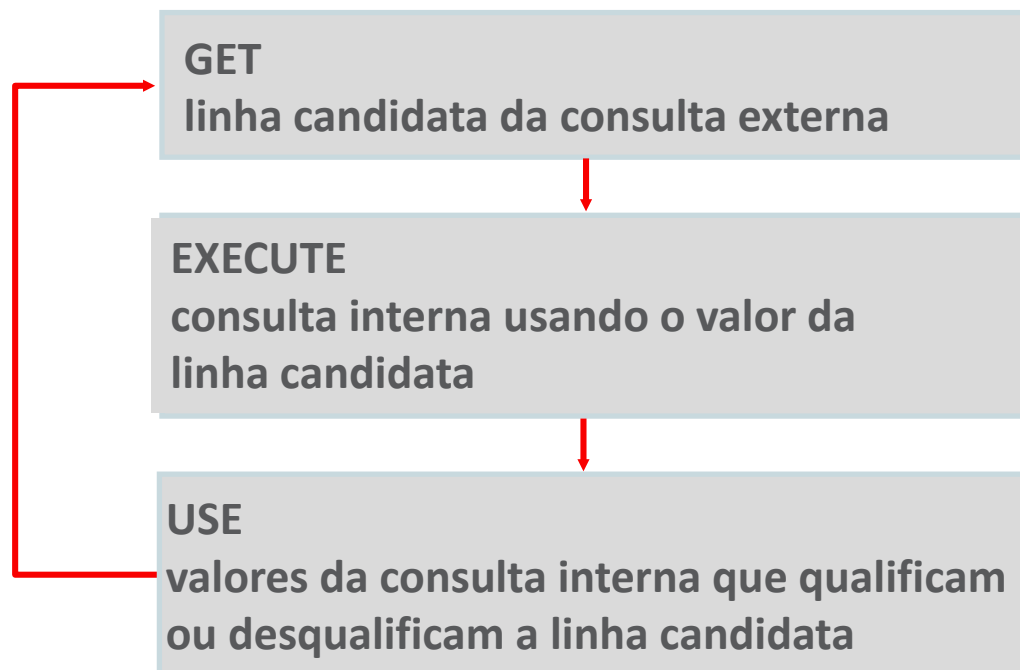
Subconsultas Correlacionadas

- Uma subconsulta correlacionada é avaliada uma vez para cada linha processada pela instrução pai.



Subconsultas Correlacionadas

- A instrução pai pode ser do tipo SELECT, UPDATE ou DELETE.



Exemplo de Subconsulta Correlacionada

- Quem tem um salário maior que a média salarial do próprio departamento?
- Para responder a essa pergunta, precisamos gravar uma subconsulta correlacionada.
- As subconsultas correlacionadas são usadas para processamento linha a linha.

```
SELECT  o.first_name,  
        o.last_name,  
        o.salary  
FROM employees o  
WHERE o.salary >  
      (SELECT AVG(i.salary)  
       FROM employees i  
       WHERE i.department_id =  
             o.department_id);
```

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Shelley	Higgins	12000
Eleni	Zlotkey	10500
Ellen	Abel	11000
Kevin	Mourgos	5800
Alexander	Hunold	9000
Michael	Hartstein	13000

Exemplo de Subconsulta Correlacionada

- Cada subconsulta é executada uma vez para cada linha da consulta externa.
- Em uma subconsulta normal, a consulta SELECT interna é executada primeiro e uma única vez, retornando os valores a serem usados pela consulta externa.

```
SELECT  o.first_name,  
        o.last_name,  
        o.salary  
FROM employees o  
WHERE o.salary >  
      (SELECT AVG(i.salary)  
       FROM employees i  
       WHERE i.department_id =  
             o.department_id);
```

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Shelley	Higgins	12000
Eleni	Zlotkey	10500
Ellen	Abel	11000
Kevin	Mourgos	5800
Alexander	Hunold	9000
Michael	Hartstein	13000

Exemplo de Subconsulta Correlacionada

- No entanto, uma subconsulta correlacionada é executada uma vez para cada linha considerada pela consulta externa.
- Em outras palavras, a consulta interna é impulsionada pela consulta externa.
- Neste exemplo, a subconsulta correlacionada está marcada em vermelho.

```
SELECT  o.first_name,  
        o.last_name,  
        o.salary  
FROM employees o  
WHERE o.salary >  
      (SELECT AVG(i.salary)  
       FROM employees i  
       WHERE i.department_id =  
             o.department_id);
```

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Shelley	Higgins	12000
Eleni	Zlotkey	10500
Ellen	Abel	11000
Kevin	Mourgos	5800
Alexander	Hunold	9000
Michael	Hartstein	13000

EXISTS e NOT EXISTS em Subconsultas

- EXISTS e seu oposto, NOT EXISTS, são duas cláusulas que podem ser usadas para testar equivalências em subconsultas.
- EXISTS faz o teste para encontrar um valor TRUE, ou um resultado equivalente na subconsulta.
- Para responder à pergunta: "Quais funcionários não são gerentes?"
 - Você primeiro precisa perguntar: "Quem são os gerentes?"
 - Depois, precisa saber "Quem NÃO ESTÁ na lista de gerentes?"

EXISTS e NOT EXISTS em Subconsultas

- Neste exemplo, a subconsulta seleciona os funcionários que são gerentes.
- A consulta externa retorna então as linhas da tabela de funcionários que **NÃO EXISTEM** na subconsulta.

```
SELECT last_name AS "Not a Manager"
FROM employees emp
WHERE NOT EXISTS
  (SELECT *
   FROM employees mgr
   WHERE mgr.manager_id = emp.employee_id);
```

Not a Manager
Whalen
Gietz
Abel
Taylor
Grant
Rajs
Davies
Matos
Vargas
Ernst
...

EXISTS e NOT EXISTS em Subconsultas

- Se a mesma consulta for executada com NOT IN, em vez de NOT EXISTS, o resultado será bem diferente.
- O resultado dessa consulta sugere que não há funcionários que também não são gerentes, portanto, todos os funcionários são gerentes, o que já sabemos que não é verdade.

```
SELECT last_name AS "Not a Manager"  
FROM employees emp  
WHERE emp.employee_id NOT IN  
  (SELECT mgr.manager_id  
   FROM employees mgr);
```

nenhum dado encontrado

EXISTS e NOT EXISTS em Subconsultas

- O motivo do resultado estranho é o valor nulo retornado pela subconsulta.
- Uma das linhas na tabela de funcionários não tem um gerente, e isso deixa todo o resultado errado.
- As subconsultas podem retornar três valores: TRUE, FALSE e UNKNOWN.
- Um valor nulo no conjunto de resultados da subconsulta retornará um valor UNKNOWN, que o Oracle não consegue avaliar e, portanto, não avalia.

```
SELECT last_name AS "Not a Manager"  
FROM employees emp  
WHERE emp.employee_id NOT IN  
      (SELECT mgr.manager_id  
       FROM employees mgr);
```

nenhum dado encontrado



EXISTS e NOT EXISTS em Subconsultas

- TENHA CUIDADO com valores nulos nas subconsultas quando estiver usando IN ou NOT IN.
- Se não tiver certeza se uma subconsulta incluirá ou não um valor nulo, elimine o valor nulo usando IS NOT NULL em uma cláusula WHERE ou use NOT EXISTS.
- Por exemplo: WHERE emp.manager_id IS NOT NULL, ou use NOT EXISTS, por via das dúvidas.

Cláusula WITH

- Se precisar gravar uma consulta muito complexa, com junções e agregações usadas muitas vezes, você poderá gravar as diferentes partes da instrução como blocos de consultas e, em seguida, usar esses mesmos blocos em uma instrução SELECT.
- O Oracle permite gravar subconsultas nomeadas em uma instrução, desde que ela comece com a palavra-chave WITH.
- A cláusula WITH recupera os resultados de um ou mais blocos de consulta e armazena-os para o usuário que executa a consulta.

Cláusula WITH

- A cláusula WITH melhora o desempenho.
- A cláusula WITH torna a consulta mais fácil de ser lida.
- A sintaxe da cláusula WITH é a seguinte:

```
WITH nome-subconsulta AS (subconsulta),  
     nome-subconsulta AS (subconsulta)  
SELECT lista-coluna  
FROM   {tabela | nome-subconsulta | exibição}  
WHERE  condição é verdadeira;
```

Cláusula WITH

- Grave a consulta do seguinte requisito:
 - Exiba uma lista de sobrenomes de funcionários que não são gerentes.
- Para construir essa consulta, você primeiro precisará obter uma lista de `manager_ids` da tabela de funcionários e, em seguida, retornar os nomes dos funcionários cuja id de funcionário não esteja na lista de gerentes.
- Podemos criar uma subconsulta nomeada usando a cláusula `WITH` para recuperar o `manager_id` da tabela de funcionários. Depois, a consulta externa retornará os funcionários que não aparecem nessa lista.

Cláusula WITH

```
WITH managers AS
  (SELECT DISTINCT manager_id
   FROM employees
   WHERE manager_id IS NOT NULL)

SELECT last_name AS "Not a manager"
FROM employees
WHERE employee_id NOT IN
  (SELECT *
   FROM managers);
```

Not a manager
Whalen
Gietz
Abel
Taylor
Grant
Rajs
Davies
Vargas
Ernst
...

Resumo

Nesta lição, você deverá ter aprendido a:

- Identificar quando subconsultas correlacionadas são necessárias.
- Construir e executar subconsultas correlacionadas.
- Criar uma consulta usando os operadores EXISTS e NOT EXISTS para testar as linhas que retornam da subconsulta
- Construir e executar subconsultas nomeadas usando a cláusula WITH.

