

# 6

## Subconsultas

## Objetivos deste Capítulo

- Ao concluir este capítulo, você poderá:
  - Criar subconsultas
  - Descrever quais as aplicações das subconsultas
  - Citar os tipos de subconsultas
  - Criar subconsultas de uma única linha, de várias linhas e de várias colunas
  - Criar subconsultas correlacionadas
  - Atualizar e deletar linhas usando subconsultas correlacionadas
  - Usar os operadores `EXISTS` e `NOT EXISTS`
  - Usar a cláusula `WITH`

6-2

### Objetivos deste Capítulo

Neste capítulo, você verá que é possível criar subconsultas na cláusula `WHERE` de um comando SQL para obter valores baseados em um valor condicional desconhecido. Este capítulo trata de subconsultas de uma única linha, de várias linhas e de várias colunas.

Serão abordadas também subconsultas na cláusula `FROM` de um comando `SELECT`, subconsultas correlacionadas e a cláusula `WITH`.

## Problema a Ser Resolvido

**Quais funcionários têm um salário maior que o salário de Paula?**

**Consulta interna ou subconsulta:**

**Qual é o salário de Paula?**

6-3

### Problema a Ser Resolvido

Suponha que você queira criar uma consulta para saber quem ganha um salário maior que o salário da funcionária Paula.

Para solucionar esse problema, são necessárias *duas* consultas: uma para saber quanto Paula ganha e outra para saber quem ganha uma quantia maior.

Você pode solucionar esse problema com a combinação das duas consultas, *inserindo* uma na outra.

A consulta interna (ou *subconsulta*) retorna um valor usado pela consulta externa (ou *consulta principal*). Usar uma subconsulta é o mesmo que executar duas consultas em sequência e utilizar o resultado da primeira como o valor de pesquisa na segunda.

# Sintaxe da Subconsulta

```
SELECT  lista_do_select
FROM    tabela
WHERE   expressao operador
              (SELECT  lista_do_select
                FROM    tabela);
```

- A subconsulta (consulta interna) é executada uma vez antes da consulta principal (consulta externa).
- O resultado da subconsulta é usado pela consulta principal.

6-4

## Sintaxe da Subconsulta

Uma subconsulta é um comando `SELECT` incorporado a uma cláusula de outro comando `SELECT`. É possível criar comandos complexos a partir de comandos simples usando subconsultas. Elas poderão ser muito úteis quando for necessário selecionar linhas de uma tabela com uma condição que dependa dos dados da própria tabela.

É possível inserir a subconsulta em várias cláusulas SQL, inclusive nestas:

- Cláusula `WHERE`
- Cláusula `HAVING`
- Cláusula `FROM`

Na sintaxe: *operador* inclui uma condição de comparação, como `>`, `=` ou `IN`

As condições de comparação estão incluídas em duas classes: operadores de uma única linha (`>`, `=`, `>=`, `<`, `<>`, `<=`) e de várias linhas (`IN`, `ANY`, `ALL`).

A subconsulta normalmente é executada primeiro e o seu resultado é usado para concluir a condição da consulta principal (ou externa).

## Exemplo de Subconsulta

```
SELECT nome
FROM   funcionario 10500
WHERE  salario >
      (SELECT salario
       FROM   funcionario
       WHERE  nome = 'Paula');
```

NOME
Roberto
Nair
Leonardo
Pamela
William
Sheila

6 linhas retornadas

6-5

### Exemplo de Subconsulta

O exemplo do slide, a consulta interna determina o salário da funcionária Paula. A consulta externa recebe o resultado da consulta interna e o utiliza para exibir todos os funcionários que ganham mais que esse valor.

## Diretrizes do Uso de Subconsultas

- As subconsultas devem ser delimitadas por parênteses.
- Por clareza, posicione subconsultas à direita da condição de comparação.
- Não é necessário o uso da cláusula `ORDER BY` em subconsultas, a não ser que se deseje realizar uma análise Top-N.
- Use operadores de uma única linha com subconsultas de uma única linha e operadores de várias linhas com subconsultas de várias linhas.

6-6

### Diretrizes de Uso de Subconsultas

Uma subconsulta deve ser delimitada por parênteses.

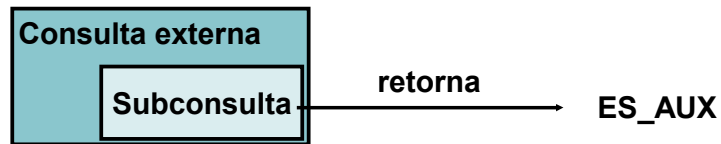
Posicione a subconsulta à direita da condição de comparação para fins de legibilidade.

Uma cláusula `ORDER BY` pode ser usada na subconsulta quando se deseja executar uma análise do tipo Top-N.

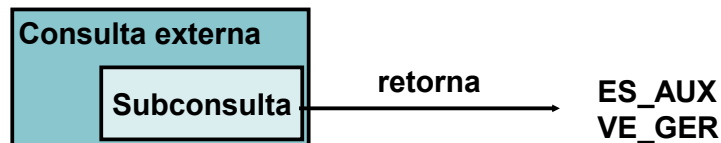
Duas classes de condições de comparação são usadas em subconsultas: operadores de uma única linha e operadores de várias linhas.

# Tipos de Subconsultas

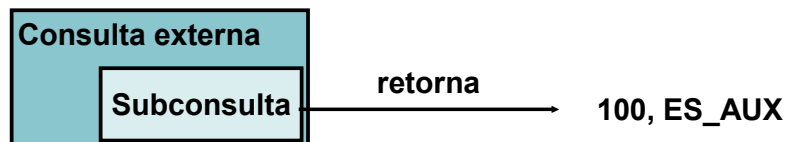
- Subconsulta de uma única linha



- Subconsulta de várias linhas



- Subconsulta de várias colunas



6-7

## Tipos de Subconsultas

Subconsultas de uma única linha: comando `SELECT` interno que retorna apenas uma única linha.

Subconsultas de várias linhas: comando `SELECT` interno que retorna mais de uma linha.

Subconsultas de várias colunas: comando `SELECT` interno que retorna mais de uma coluna.

## Subconsultas de uma Única Linha

- Retornam somente uma linha
- Usam operadores de comparação de uma única linha

Operador	Significado
=	Igual a
>	Maior que
>=	Maior que ou igual a
<	Menor que
<=	Menor que ou igual a
<> ou !=	Diferente de

6-8

### Subconsultas de uma Única Linha

Uma subconsulta de uma única linha retorna uma linha do comando `SELECT` interno. Esse tipo de subconsulta usa um operador de uma única linha. O slide exibe uma lista de operadores de uma única linha.

#### Exemplo

Exiba os funcionários cujo código do cargo é igual ao do funcionário 141:

```
SELECT sobrenome, cod_cargo
FROM   funcionario
WHERE  cod_cargo =
      (SELECT cod_cargo
       FROM   funcionario
       WHERE  cod_funcionario = 141);
```

SOBRENOME	COD_CARGO
Lopes	ES_AUX
Cabral	ES_AUX
Miranda	ES_AUX
Chaves	ES_AUX

4 linhas retornadas em 0,05 segundos



## Subconsultas de uma Única Linha

```
SELECT sobrenome, cod_cargo, salario
FROM funcionario
WHERE cod_cargo = (SELECT cod_cargo
                   FROM funcionario
                   WHERE cod_funcionario = 141)
AND salario > (SELECT salario
               FROM funcionario
               WHERE cod_funcionario = 143);
```

SOBRENOME	COD_CARGO	SALARIO
Lopes	ES_AUX	3500
Cabral	ES_AUX	3100

2 linhas retornadas em 0,03 segundos

6-9

### Subconsultas de uma Única Linha

O exemplo do slide mostra os funcionários cujo código do cargo é igual ao do funcionário 141 e cujo salário é maior que o do funcionário 143.

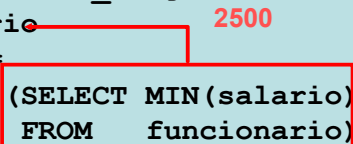
O exemplo consiste em três blocos de consulta: a consulta externa e duas consultas internas. Os blocos de consulta interna são executados primeiro, produzindo os resultados ES\_AUX e 2600, respectivamente. O bloco de consulta externa é processado depois e usa os valores retornados pelas consultas internas para realizar suas condições de pesquisa.

As duas consultas internas retornam valores únicos (ES\_AUX e 2600, respectivamente); por isso, essas subconsultas são denominadas subconsultas de uma única linha.

As consultas internas e externas podem obter dados de tabelas diferentes.

# Funções de Agrupamento em Subconsultas

```
SELECT sobrenome, cod_cargo, salario
FROM funcionario
WHERE salario =
    (SELECT MIN(salario)
     FROM funcionario);
```



SOBRENOME	COD_CARGO	SALARIO
Chaves	ES_AUX	2500

1 linhas retornadas em 0,03 segundos

6-10

## Funções de Agrupamento em Subconsultas

Você poderá usar uma função de agrupamento numa subconsulta para retornar uma única linha.

O exemplo do slide mostra o sobrenome, o código do cargo e o salário de todos os funcionários cujo salário é igual ao menor salário. A função de agrupamento `MIN` retorna um único valor (2500) que será utilizado pela consulta externa.

## A Cláusula HAVING com Subconsultas

- A subconsulta é executada primeiro e retorna o resultado que será usado pela cláusula HAVING da consulta externa.

```
SELECT  cod_departamento, MIN(salario)
FROM    funcionario
GROUP BY cod_departamento
HAVING  MIN(salario) > (SELECT MIN(salario)
                        FROM    funcionario
                        WHERE cod_departamento = 50);
```

Diagram illustrating the execution of the SQL query. A red box highlights the `HAVING MIN(salario)` clause, and another red box highlights the subquery `(SELECT MIN(salario) FROM funcionario WHERE cod_departamento = 50);`. A red arrow points from the subquery result `2500` to the comparison operator `>` in the `HAVING` clause.

6-11

### A Cláusula HAVING com Subconsultas

É possível usar subconsultas tanto na cláusula `WHERE` como na cláusula `HAVING`. A subconsulta é executada e o resultado é retornado para a cláusula `HAVING` da consulta principal.

O comando SQL do slide mostra todos os departamentos cujo salário mínimo é maior que o do departamento 50.

COD_DEPARTAMENTO	MIN(SALARIO)
-	7000
90	17000
20	6000
110	8300
80	8600
60	4200
10	4400

Exemplo : Localize o cargo com o menor salário médio.

```
SELECT  cod_cargo, AVG(salario)
FROM    funcionario
GROUP BY cod_cargo
HAVING  AVG(salario) = (SELECT  MIN(AVG(salario))
                        FROM    funcionario
                        GROUP BY cod_cargo);
```

# Erros com Subconsultas

```
SELECT cod_funcionario, sobrenome
FROM   funcionario
WHERE  salario =
        (SELECT   MIN(salario)
         FROM     funcionario
         GROUP BY cod_departamento);
```

```
(SELECT   MIN(salario)
      *
ERRO na linha 4:
ORA-01427: a subconsulta de uma única linha retorna
mais de uma linha
```

6-12

## Erros com Subconsultas

Um erro comum com subconsultas ocorre quando mais de uma linha é retornada por uma subconsulta quando foi usado um operador de Única Linha na consulta externa.

No comando SQL do slide, a subconsulta contém uma cláusula `GROUP BY`, o que significa que ela retornará várias linhas, uma para cada grupo localizado. Nesse caso, os resultados da subconsulta são 4400, 6000, 2500, 4200, 7000, 17000 e 8300.

A consulta externa recebe esses resultados e os utiliza em sua cláusula `WHERE`. A cláusula `WHERE` contém um operador de igual a (`=`), que é um operador de comparação de uma única linha que espera apenas um valor. Esse operador não pode aceitar mais de um valor da subconsulta e, portanto, gera o erro.

Para corrigir esse erro, altere o operador `=` para `IN`.

## Subconsultas de Várias Linhas

- Retornam mais de uma linha
- Usam operadores de comparação de várias linhas

Operador	Significado
IN	Igual a qualquer elemento da lista
ANY ou SOME	Compara o valor com cada valor retornado pela subconsulta
ALL	Compara o valor com todos os valores retornados pela subconsulta

6-13

### Subconsultas de Várias Linhas

As subconsultas que retornam mais de uma linha são denominadas subconsultas de várias linhas. Com uma subconsulta de várias linhas, você deve usar um operador de várias linhas, ao invés de um operador de uma única linha, pois o operador de várias linhas espera um ou mais valores. Por exemplo, localize os funcionários cujo salário é igual ao menor salário em cada departamento:

```
SELECT sobrenome, salario, cod_departamento
FROM   funcionario
WHERE  salario IN (SELECT   MIN(salario)
                   FROM     funcionario
                   GROUP BY cod_departamento);
```

A consulta interna é executada primeiro e produz um resultado. A consulta externa é processada depois e usa os valores retornados pela consulta interna para realizar sua pesquisa. Após o processamento da consulta interna, a consulta externa aparece para o Oracle da seguinte forma:

```
SELECT sobrenome, salario, cod_departamento
FROM   funcionario
WHERE  salario IN (2500, 4200, 4400, 6000, 7000, 8300,
                  8600, 17000);
```

# Operador ANY

```
SELECT cod_funcionario, sobrenome, cod_cargo, salario
FROM   funcionario 9000, 6000, 4200
WHERE  salario < ANY
      (SELECT salario
       FROM   funcionario
       WHERE  cod_cargo = 'IT_PROG')
AND    cod_cargo <> 'IT_PROG';
```

COD_FUNCIONARIO	SOBRENOME	COD_CARGO	SALARIO
124	Brunni	ES_GER	5800
141	Lopes	ES_AUX	3500
142	Cabral	ES_AUX	3100
143	Miranda	ES_AUX	2600
144	Chaves	ES_AUX	2500
176	Voorhees	VE_REP	8600
178	Yamamura	VE_REP	7000
200	Trajano	AD_ASST	4400
202	Kramer	MK_ANA	6000
206	Nascimento	CTPUB_GER	8300

10 linhas retornadas em 0,04 segundos

6-14

## Operador ANY

O operador **ANY** (e seu sinônimo, o operador **SOME**) faz com que um operador de uma única linha possa ser usado em uma subconsulta de várias linhas. O operador **ANY** verifica se a condição de pesquisa é verdadeira para pelo menos um dos membros da lista de retorno da subconsulta.

O exemplo do slide mostra os funcionários que não são programadores e cujo salário é menor que o de qualquer programador. Na tabela **FUNCIONARIO**, o maior salário de um programador é R\$ 9.000,00.

<ANY significa menor que o máximo.

>ANY significa maior que o mínimo.

=ANY equivale a **IN**.

# Operador ALL

```
SELECT cod_funcionario, sobrenome, cod_cargo, salario
FROM funcionario 9000, 6000, 4200
WHERE salario < ALL (SELECT salario
                     FROM funcionario
                     WHERE cod_cargo = 'IT_PROG')
AND cod_cargo <> 'IT_PROG';
```

COD_FUNCIONARIO	SOBRENOME	COD_CARGO	SALARIO
141	Lopes	ES_AUX	3500
142	Cabral	ES_AUX	3100
143	Miranda	ES_AUX	2600
144	Chaves	ES_AUX	2500

4 linhas retornadas em 0,00 segundos

6-15

## Operador ALL

O operador ALL também faz com que um operador de uma única linha seja aplicado a uma subconsulta de várias linhas. Porém, no caso do operador ALL a condição de pesquisa deve ser verdadeira para todos os membros da lista de retorno da subconsulta.

O exemplo do slide mostra os funcionários que não são programadores e cujo salário é menor que o de todos os programadores. Na tabela FUNCIONARIO, o menor salário de um programador é R\$ 4.200.

>ALL significa maior que o máximo.

<ALL significa menor que o mínimo.

É possível usar o operador NOT com os operadores IN, ANY e ALL.

## Subconsulta com Valores Nulos

```
SELECT func.sobrenome
FROM   funcionario func
WHERE  func.cod_funcionario NOT IN
                                   (SELECT ger.cod_gerente
                                   FROM   funcionario ger);

não há linhas selecionadas
```

6-16

### Subconsulta com Valores Nulos

O comando SQL do slide tenta exibir todos os funcionários que não são gerentes. A intenção era que esse comando retornasse 12 linhas, porém ele não retorna nenhuma. Isso ocorre porque um dos valores retornados pela consulta interna é um valor nulo e, por isso, a consulta externa não retorna nenhuma linha.

O motivo é que todas as condições que comparam com um valor nulo resultam em um valor nulo. Dessa forma, sempre que houver a possibilidade de valores nulos integrarem o conjunto de resultados de uma subconsulta, não use o operador `NOT IN`. Esse operador corresponde a `<> ALL`.

Observe que o valor nulo como parte do conjunto de resultados de uma subconsulta não representa um problema se você usa o operador `IN`. Esse operador corresponde a `=ANY`. Por exemplo, para exibir os funcionários que são gerentes, o comando SQL abaixo pode ser perfeitamente utilizado:

```
SELECT func.sobrenome
FROM   funcionario func
WHERE  func.cod_funcionario IN
                                   (SELECT ger.cod_gerente
                                   FROM   funcionario ger);
```



### Subconsulta com Valores Nulos (continuação)

Para resolver o problema do valor nulo retornado pela subconsulta que foi usada com o operador NOT IN, você pode incluir um filtro na subconsulta que elimine os valores nulos:

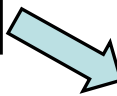
```
SELECT func.sobrenome
FROM   funcionario func
WHERE  func.cod_funcionario NOT IN
      (SELECT ger.cod_gerente
       FROM   funcionario ger
       WHERE  ger.cod_gerente IS NOT NULL);
```

## Subconsultas de Várias Colunas

- Para cada linha da consulta principal, o grupo colunas na cláusula WHERE é comparado aos valores do grupo de colunas retornados pela subconsulta.

### Consulta principal:

WHERE (cod\_gerente, cod\_departamento) IN



### Subconsulta:

100	90
102	60
124	50

6-18

### Subconsultas de Várias Colunas

Quando você precisar realizar comparação de duas ou mais colunas, crie uma cláusula WHERE composta e use operadores lógicos para comparar esse grupo de colunas com o resultado de uma subconsultas que, além de retornar várias linhas, retorna várias colunas.

A sintaxe para criação de uma subconsulta de várias colunas é a seguinte:

```
SELECT  coluna1, coluna2, ...
FROM    tabela
WHERE   (coluna1, coluna2, ...) IN
                                   (SELECT coluna1, coluna2, ...
                                    FROM    tabela
                                    WHERE   condição);
```

## Subconsultas de Várias Colunas

- Consulte os funcionários que estão subordinados ao mesmo gerente e que trabalham no mesmo departamento que os funcionários de código 199 ou 174

```
SELECT cod_funcionario, cod_gerente, cod_departamento
FROM funcionario
WHERE (cod_gerente, cod_departamento) IN
      (SELECT cod_gerente, cod_departamento
       FROM funcionario
       WHERE cod_funcionario IN (199,174))
AND   cod_funcionario NOT IN (199,174);
```

6-19

### Subconsultas de Várias Colunas (continuação)

O exemplo do slide exibe uma subconsulta de várias colunas, isto é, uma subconsulta que retorna mais de uma coluna. Ela compara os valores das colunas `COD_GERENTE` e `COD_DEPARTAMENTO` de cada linha da tabela `FUNCIONARIO` com os valores das mesmas colunas para os funcionários com `COD_FUNCIONARIO` 199 ou 174.

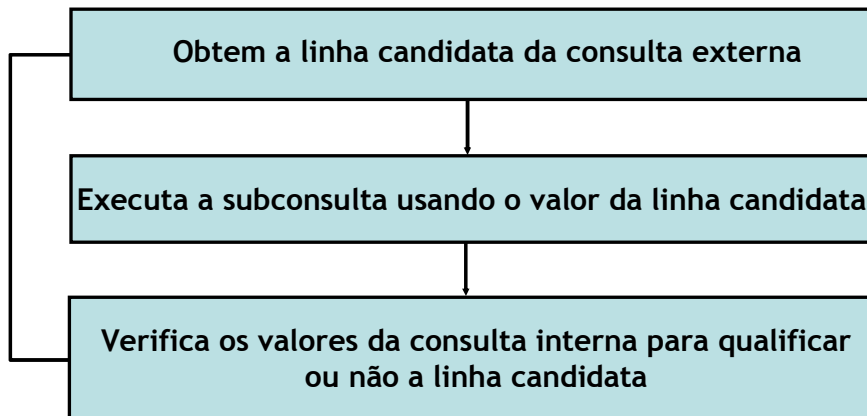
Primeiro, a subconsulta para recuperar os valores de `COD_GERENTE` e `COD_DEPARTAMENTO` para os funcionários com `COD_FUNCIONARIO` 199 ou 174 é executada. Esses valores são comparados com as colunas `COD_GERENTE` e `COD_DEPARTAMENTO` de cada linha da tabela `FUNCIONARIO`. Se houver correspondência entre os valores, a linha será exibida. Na saída, os registros dos funcionários com `COD_FUNCIONARIO` 199 ou 174 não serão exibidos. A saída da consulta no slide é a seguinte:

COD_FUNCIONARIO	COD_GERENTE	COD_DEPARTAMENTO
176	149	80

1 linhas retornadas em 0,17 segundos

## Subconsultas Correlacionadas

- As subconsultas correlacionadas são usadas para processar linha por linha da consulta externa. A subconsulta é executada uma vez para cada linha da consulta externa.



6-20

### Subconsultas Correlacionadas

Uma subconsulta correlacionada é quando uma subconsulta faz referência a uma coluna de uma tabela da consulta externa. A subconsulta correlacionada é executada uma vez para cada linha processada pela consulta externa. A consulta externa pode ser uma instrução `SELECT`, `UPDATE` ou `DELETE`.

Uma subconsulta aninhada normal é executada primeiro e apenas uma vez, retornando os valores a serem usados pela consulta externa. No entanto, uma subconsulta correlacionada é executada uma vez para cada linha candidata considerada pela consulta externa. Em outras palavras, a consulta interna é orientada pela consulta externa.

O Oracle executa uma subconsulta correlacionada seguindo o seguinte fluxo:

- Obtem uma linha candidata (resultado apenas da consulta externa).
- Executa a consulta interna usando o valor da linha candidata.
- Usa os valores resultantes da consulta interna para qualificar ou desqualificar a linha candidata.
- Repete até que não existam mais linhas candidatas.

## Subconsultas Correlacionadas

- A subconsulta correlacionada faz referência a uma coluna de uma tabela da consulta externa.

```
SELECT coluna1, coluna2, ...  
FROM   tabela1 externa  
WHERE  coluna1 operador  
        (SELECT  coluna1, coluna2  
          FROM    tabela2  
          WHERE    expressao1 =  
                  externa.expressao2) ;
```

6-21

### Subconsultas Correlacionadas (continuação)

Uma subconsulta correlacionada é uma forma de ler todas as linhas de uma tabela e comparar os valores de cada linha com os dados relacionados da subconsulta. Ela é usada sempre que uma subconsulta precisa retornar um resultado ou um conjunto de resultados diferente para cada linha candidata considerada pela consulta externa. Em outras palavras, use uma subconsulta correlacionada para responder a uma pergunta com várias partes cuja resposta depende do valor de cada linha processada pela consulta externa.

O Oracle executa uma subconsulta correlacionada quando essa subconsulta faz referência a uma coluna de uma tabela na consulta externa.

Você pode usar os operadores `ANY` e `ALL` em uma subconsulta correlacionada.

## Subconsultas Correlacionadas

- Liste todos os funcionários que recebem mais que o salário médio nos seus respectivos departamentos.

```
SELECT sobrenome, salario, cod_departamento
FROM funcionario externa
WHERE salario >
      (SELECT AVG(salario)
       FROM funcionario
       WHERE cod_departamento =
         externa.cod_departamento);
```

Sempre que uma linha da consulta externa for processada, a consulta interna será avaliada.

6-22

### Subconsultas Correlacionadas (continuação)

O exemplo do slide determina quais funcionários recebem mais que o salário médio dos seus respectivos departamentos. Nesse caso, a subconsulta correlacionada calcula especificamente o salário médio do departamento de cada funcionário pesquisado.

Como as consultas externa e interna usam a tabela `FUNCIONARIO` na cláusula `FROM`, é fornecido um apelido a essa tabela na consulta externa para maior clareza. Além do apelido tornar todo o comando mais legível, sem ele a consulta não funcionará corretamente, porque a subconsulta não conseguirá diferenciar a coluna da tabela interna da coluna da tabela externa.

SOBRENOME	SALARIO	COD_DEPARTAMENTO
Carlos	24000	90
Thacker	13000	20
Almeida	12000	110
Brunni	5800	50
Schultz	10500	80
Sue	11000	80
Honorato	9000	60

7 linhas retornadas em 0,05 segundos

## Subconsultas Correlacionadas

- Consulte os detalhes dos funcionários que mudaram de cargo pelo menos duas vezes.

```
SELECT f.cod_funcionario, f.sobrenome, f.cod_cargo
FROM   funcionario f
WHERE  2 <= (SELECT COUNT(*)
             FROM   historico_cargo
             WHERE  cod_funcionario = f.cod_funcionario);
```

COD_FUNCIONARIO	SOBRENOME	COD_CARGO
101	Martins	AD_VP
176	Voorhees	VE_REP
200	Trajano	AD_ASST

3 linhas retornadas em 0,08 segundos

6-23

### Subconsultas Correlacionadas (continuação)

O exemplo do slide exibe os detalhes dos funcionários que mudaram de cargo pelo menos duas vezes.

## Operador EXISTS

- O operador `EXISTS` verifica a existência de linhas no conjunto de resultados da subconsulta que sejam correlacionadas com a linha candidata da consulta externa.
- Se for localizado uma linha na subconsulta a linha candidata externa é retornada no resultado final.
- Se não for localizado um valor de linha na subconsulta a linha externa é desqualificada.

6-24

### Operador EXISTS

Esse operador é frequentemente usado com subconsultas correlacionadas para verificar se uma linha recuperada pela consulta externa resulta em alguma linha recuperada pela consulta interna. Se a subconsulta retornar, pelo menos, uma linha, o operador retornará `TRUE`, qualificando a linha externa. Se a subconsulta não retornar nenhuma linha, o operador retornará `FALSE`, desqualificando a linha externa. Da mesma forma, `NOT EXISTS` verifica se uma linha recuperada pela consulta externa não faz recuperar registros pela consulta interna.



# Operador EXISTS

- Liste os funcionários que possuem subordinados:

```
SELECT cod_funcionario, sobrenome, cod_cargo, cod_departamento
FROM   funcionario externa
WHERE  EXISTS ( SELECT 'X'
                  FROM   funcionario
                  WHERE  cod_gerente =
                        externa.cod_funcionario);
```

COD_FUNCIONARIO	SOBRENOME	COD_CARGO	COD_DEPARTAMENTO
100	Carlos	AD_PRES	90
101	Martins	AD_VP	90
102	da Silva	AD_VP	90
103	Honorato	IT_PROG	60
124	Brunni	ES_GER	50
149	Schultz	VE_GER	80
201	Thacker	MK_GER	20
205	Almeida	CT_GER	110

8 linhas retornadas em 0,03 segundos

6-25

## Operador EXISTS (continuação)

O exemplo do slide consulta os funcionários que possuem subordinados.

O Oracle processa o comando da seguinte forma:

- Para cada funcionário (linha externa), é executada a subconsulta que utiliza como cláusula WHERE a condição:

`cod_gerente = externa.cod_funcionario`

- Se for encontrado pelo menos um registro onde o funcionário da consulta externa corresponde ao gerente de alguém (consulta interna), o processamento da subconsulta é interrompido retornando TRUE e qualificando a linha externa.
- Se não for encontrado nenhum registro onde o funcionário da consulta externa corresponde ao gerente de qualquer outro funcionário da consulta interna, então o retorno é FALSE e a linha externa é desqualificada.

Observe que a consulta interna não precisa retornar um valor específico, pois a cláusula EXISTS verifica apenas a existência de algum resultado e não o valor do resultado. Portanto, será possível selecionar uma constante.

# Operador NOT EXISTS

- Localize todos os departamentos sem funcionários:

```
SELECT cod_departamento, nome_departamento
FROM departamento d
WHERE NOT EXISTS (SELECT 'X'
                   FROM   funcionario
                   WHERE  cod_departamento
                        = d.cod_departamento);
```

COD_DEPARTAMENTO	NOME_DEPARTAMENTO
190	Contratos

1 linhas retornadas em 0,05 segundos

6-26

## Operador NOT EXISTS

O exemplo do slide consulta todos os departamentos que não possuem funcionários.

É possível usar o operador NOT IN como uma alternativa para um operador NOT EXISTS, como mostra o seguinte exemplo:

```
SELECT cod_departamento, nome_departamento
FROM   departamento
WHERE  cod_departamento NOT IN (SELECT cod_departamento
                                FROM   funcionario);
```

não há linhas selecionadas

No entanto, NOT IN será avaliado como FALSE se um elemento do conjunto for um valor NULL. Assim, a consulta não retornará linhas mesmo se houver linhas na tabela de departamentos que atendam à condição WHERE.

## Subconsulta na Lista SELECT

- Uma subconsulta pode ser usada na lista `SELECT` de uma consulta principal apenas se ela retornar um ou nenhum registro.

```
SELECT    cod_funcionario,  
          nome,  
          sobrenome,  
          (SELECT nome_departamento  
           FROM    departamento d  
           WHERE   f.cod_departamento =  
                  d.cod_departamento ) departamento  
FROM      funcionario f;
```

6-27

### Subconsulta na Lista SELECT

O resultado da consulta do slide é o seguinte:

COD_FUNCIONARIO	NOME	SOBRENOME	DEPARTAMENTO
100	Roberto	Carlos	Executivo
101	Nair	Martins	Executivo
102	Leonardo	da Silva	Executivo
103	Alexandre	Honorato	Informática
104	Pedro	Osterno	Informática
107	Dilma	Barata	Informática
124	Manuela	Brunni	Logística
141	Tomé	Lopes	Logística
142	Cristóvão	Cabral	Logística
143	Rafael	Miranda	Logística
144	Pedro	Chaves	Logística
149	Paula	Schultz	Vendas
174	Pamela	Sue	Vendas
176	Elias	Voorhees	Vendas
178	Sadako	Yamamura	-
200	Luciana	Trajano	Administração
201	William	Thacker	Marketing
202	Cosmo	Kramer	Marketing
205	Sheila	Almeida	Contabilidade
206	Roberto	Nascimento	Contabilidade

20 linhas retornadas em 0,00 segundos

## A Cláusula WITH

- Permite usar o mesmo bloco de consulta em um comando `SELECT` quando ele ocorre mais de uma vez em uma consulta complexa.
- A cláusula `WITH` recupera os resultados de um bloco de consulta e os armazena no tablespace temporário do usuário.
- A cláusula `WITH` melhora o desempenho.

6-28

### A Cláusula WITH

Usando a cláusula `WITH`, é possível definir um bloco de consulta antes de usá-lo em uma consulta. Essa cláusula permite reutilizar o mesmo bloco de consulta em um comando `SELECT` quando ele ocorre mais de uma vez em uma consulta complexa. Ela é especialmente útil quando uma consulta faz diversas referências ao mesmo bloco de consulta e existem joins e agregações.

Usando a cláusula `WITH`, o Oracle recupera os resultados de um bloco de consulta e os armazena no tablespace temporário do usuário. Essa característica pode melhorar o desempenho.

#### Vantagens da Cláusula WITH

- Facilita a leitura da consulta
- Avalia uma cláusula apenas uma vez, mesmo quando ela aparece diversas vezes na consulta
- Na maioria dos casos, pode melhorar o desempenho para consultas grandes

## Cláusula WITH: Exemplo

- Crie uma consulta com a cláusula `WITH` para exibir o nome e o salário total dos departamentos cujo salário total é maior que o salário médio de todos os departamentos.

6-29

### Cláusula `WITH`: Exemplo

O problema do slide exigirá os seguintes cálculos intermediários:

1. Calcule o salário total de cada departamento e armazene o resultado usando uma cláusula `WITH`.
2. Calcule o salário médio dos departamentos e armazene o resultado usando uma cláusula `WITH`.
3. Compare o salário total calculado na primeira etapa com o salário médio calculado na segunda etapa. Se o salário total de um departamento específico for maior que o salário médio dos departamentos, exiba o nome do departamento e o salário total correspondente.

## Cláusula WITH: Exemplo

```
WITH
custo_dept AS (
  SELECT d.nome_departamento, SUM(f.salario) AS total_dept
  FROM funcionario f, departamento d
  WHERE f.cod_departamento = d.cod_departamento
  GROUP BY d.nome_departamento),
custo_medio AS (
  SELECT SUM(total_dept)/COUNT(*) AS media_dept
  FROM custo_dept)
SELECT *
FROM custo_dept
WHERE total_dept >
  (SELECT media_dept
   FROM custo_medio)
ORDER BY nome_departamento;
```

6-30

### Cláusula WITH: Exemplo (continuação)

O exemplo do slide é um exemplo de situação na qual é possível melhorar o desempenho e criar código SQL mais simples usando a cláusula WITH. A consulta cria as consultas de nomes CUSTO\_DEPT e CUSTO\_MEDIO e as utiliza no corpo da consulta principal. Internamente, a cláusula WITH é resolvida como uma visão inline ou uma tabela temporária.

A saída gerada pelo código SQL do slide será esta:

NOME_DEPARTAMENTO	TOTAL_DEPT
Executivo	58000
Vendas	30100

2 linhas retornadas em 0,05 segundos

### Diretrizes de Uso da Cláusula WITH

- É usada apenas com comandos SELECT.
- Um nome de consulta pode ser visto por todos os blocos de consulta de elemento WITH definidos depois dele.
- Quando o nome da consulta é igual ao de uma tabela existente o nome do bloco de consulta tem precedência sobre o nome da tabela.
- A cláusula WITH pode conter mais de uma consulta. Nesse caso, cada consulta é separada por uma vírgula.

# Exercício 6

6-31

## Exercício 6

1. Crie uma consulta que solicite ao usuário o sobrenome de um funcionário. A consulta exibe o sobrenome e a data de admissão de todos os funcionários no mesmo departamento do funcionário cujo nome foi fornecido (excluindo esse funcionário). Por exemplo, se o usuário informar Schultz, serão exibidos todos os funcionários que trabalham com Schultz (excluindo ele próprio).

SOBRENOME	DATA_ADMISSAO
Sue	11/05/06
Voorhees	24/03/98

2 linhas retornadas em 0,03 segundos

2. Crie um relatório que exiba o código e o sobrenome de todos os funcionários cujo salário é maior que o salário médio. Ordene os resultados em ordem crescente de salário.

COD_FUNCIONARIO	SOBRENOME	SALARIO
103	Honorato	9000
149	Schultz	10500
174	Sue	11000
205	Almeida	12000
201	Thacker	13000
101	Martins	17000
102	da Silva	17000
100	Carlos	24000

8 linhas retornadas em 0,03 segundos

3. Crie uma consulta que exiba o código e o sobrenome de todos os funcionários que trabalham em um departamento com funcionários cujos sobrenomes contêm a letra *u*. Salve o comando SQL no arquivo de texto `cap_06_03.sql` e o execute.

COD_FUNCIONARIO	SOBRENOME
124	Brunni
141	Lopes
142	Cabral
143	Miranda
144	Chaves
149	Schultz
174	Sue
176	Voorhees

8 linhas retornadas em 0,01 segundos



### Exercício 6 (continuação)

4. a) Crie um relatório que exiba o sobrenome, o código do departamento e o código do cargo de todos os funcionários cujo código de localidade do departamento é 1700.

SOBRENOME	COD_DEPARTAMENTO	COD_CARGO
Trajano	10	AD_ASST
Carlos	90	AD_PRES
Martins	90	AD_VP
da Silva	90	AD_VP
Almeida	110	CT_GER
Nascimento	110	CTPUB_GER

6 linhas retornadas em 0,06 segundos

- b) Modifique a consulta para que um código de localidade seja solicitado ao usuário. Salve-a no arquivo `cap_06_04.sql`.
5. Crie um relatório que exiba o sobrenome e o salário de todos os funcionários subordinados ao gerente de sobrenome Carlos.

SOBRENOME	SALARIO
Martins	17000
da Silva	17000
Brunni	5800
Schultz	10500
Thacker	13000

5 linhas retornadas em 0,03 segundos

6. Crie um relatório que exiba o código do departamento, o sobrenome e o código do cargo de todos os funcionários no departamento Executivo.

COD_DEPARTAMENTO	SOBRENOME	COD_CARGO
90	Carlos	AD_PRES
90	Martins	AD_VP
90	da Silva	AD_VP

3 linhas retornadas em 0,00 segundos

7. Modifique a consulta em `cap_06_03.sql` para exibir o código, o sobrenome, bem como o salário de todos os funcionários que ganham mais que o salário médio e trabalham em um departamento com funcionários cujos sobrenomes contêm a letra *u*. Salve n como `cap_06_07.sql` e execute.

COD_FUNCIONARIO	SOBRENOME	SALARIO
149	Schultz	10500
174	Sue	11000

2 linhas retornadas em 0,04 segundos

### Exercício 6 (continuação)

8. Crie uma consulta para exibir o sobrenome, o código do departamento e o salário de qualquer funcionário cujo código do departamento e salário coincidam com o código do departamento e salário de qualquer funcionário que receba comissão.

SOBRENOME	COD_DEPARTAMENTO	SALARIO
Schultz	80	10500
Sue	80	11000
Voorhees	80	8600

3 linhas retornadas em 0.00 segundos

9. Exiba o sobrenome, o nome do departamento e o salário de qualquer funcionário cujo salário e comissão coincidam com o salário e a comissão de qualquer funcionário que esteja na localização de código 1700.

SOBRENOME	NOME_DEPARTAMENTO	SALARIO
Trajano	Administração	4400
Carlos	Executivo	24000
da Silva	Executivo	17000
Martins	Executivo	17000
Almeida	Contabilidade	12000
Nascimento	Contabilidade	8300

6 linhas retornadas em 0,05 segundos

10. Crie uma consulta para exibir o sobrenome, a data de admissão e o salário de todos os funcionários que tenham a mesma comissão da funcionária de sobrenome Schultz e ganhem R\$ 1.900,00 a menos que Schultz.

SOBRENOME	DATA_ADMISSAO	SALARIO
Voorhees	24/03/98	8600

1 linhas retornadas em 0,00 segundos

11. Crie uma consulta para exibir sobrenome, código do cargo e salário dos funcionários que recebem um salário mais alto do que o salário dos representantes de vendas (`cod_cargo = 'VE_REP'`). Ordene os resultados dos salários do maior para o menor.

SOBRENOME	COD_CARGO	SALARIO
Carlos	AD_PRES	24000
Martins	AD_VP	17000
da Silva	AD_VP	17000
Thacker	MK_GER	13000
Almeida	CT_GER	12000

5 linhas retornadas em 0,03 segundos

### Exercício 6 (continuação)

12. Exiba código, sobrenome e código do departamento dos funcionários que vivem em cidades que começam com a letra *T*.

COD_FUNCIONARIO	SOBRENOME	COD_DEPARTAMENTO
201	Thacker	20
202	Kramer	20

13. Crie uma consulta para localizar todos os funcionários que recebem mais que o salário médio nos respectivos departamentos. Exiba o sobrenome, o salário, o código do departamento e o salário médio do departamento. Ordene pelo salário médio. Use nomes para as colunas recuperadas pela consulta, conforme indicado no exemplo de saída.

[illegible]

7 linhas retornadas em 0,05 segundos

14. Liste os nomes e sobrenomes dos funcionários que não são supervisores.

- a) Usando o operador NOT EXISTS.

NOME	SOBRENOME
Pedro	Osterno
Dilma	Barata
Tomé	Lopes
Cristóvão	Cabral
Rafael	Miranda
Pedro	Chaves
Pamela	Sue
Elias	Voorhees
Sadako	Yamamura
Luciana	Trajano
Cosmo	Kramer
Roberto	Nascimento

- b) Usando o operador NOT IN

12 linhas retornadas em 0,00 segundos

### Exercício 6 (continuação)

15. Crie uma consulta para exibir os nomes e sobrenomes dos funcionários que recebem menos que o salário médio nos respectivos departamentos.

NOME	SOBRENOME
Nair	Martins
Leonardo	da Silva
Cosmo	Kramer
Roberto	Nascimento
Cristóvão	Cabral
Rafael	Miranda
Pedro	Chaves
Elias	Voorhees
Pedro	Osterno
Dilma	Barata

10 linhas retornadas em 0,04 segundos

16. Crie uma consulta para exibir os nomes e sobrenomes dos funcionários que têm um ou mais colegas de trabalho nos respectivos departamentos com datas de admissão posteriores, mas com salários mais altos.

NOME	SOBRENOME
Tomé	Lopes
Cristóvão	Cabral
Rafael	Miranda
Pedro	Chaves
Paula	Schultz
Elias	Voorhees

6 linhas retornadas em 0,00 segundos

17. Crie uma consulta para exibir os nomes dos departamentos cujo custo total de salário está acima de 1/8 do custo total de salário de toda a empresa. Use a cláusula `WITH` para criar esta consulta. Nomeie a consulta `RESUMO`.

NOME_DEPARTAMENTO	TOTAL_DEPT
Executivo	58000
Vendas	30100

2 linhas retornadas em 0,08 segundos