

03

## Gerando relatório

### Transcrição

Vamos voltar a nossa tabela `funcionarios` :

```
SQL> desc funcionarios;
Nome                      Nulo?    Tipo
-----                  -----
ID                         NOT NULL NUMBER(11)
NOME                      NOT NULL VARCHAR(20)
SOBRENOME                  NOT NULL VARCHAR(20)
SALARIO                     NUMBER(10,2)
PCT_COMISSAO                NUMBER(10,2)
```

Nós queremos gerar um relatório com o `NOME` , `SOBRENOME` e `SALARIO` .

```
SQL> select nome,sobrenome,salario from funcionarios;
```

NOME	SOBRENOME	SALARIO
Kauan	Silva	1100
Jose	Souza	1200
Eduarda	Oliveira	1200
Leonardo	Amaral	3700
Thais	Silveira	6700
Nicole	Moraes	1700
Leticia	Moreira	
Joao	Silva	
Igor	Pires	6700
Diogo	Paizano	1500
Karlos	Pereira	1500
Stevan	Rodrigues	1500
Otavio	Martez	1500

13 linhas selecionadas.

Neste caso específico, quero que nome e sobrenome estejam em uma única coluna. No Oracle, faremos isto com a função `concat()` e chamaremos a coluna de `NomeCompleto` . Queremos continuar exibindo a coluna `salario` .

```
SQL> desc funcionarios;
Nome                      Nulo?    Tipo
-----                  -----
ID                         NOT NULL NUMBER(11)
NOME                      NOT NULL VARCHAR(20)
SOBRENOME                  NOT NULL VARCHAR(20)
```

SALARIO	NUMBER(10,2)
PCT_COMISSAO	NUMBER(10,2)

Nós queremos gerar um relatório com o NOME, SOBRENOME e SALARIO.

```
SQL> select concat(nome,sobrenome) as nomeCompleto, salario from funcionarios;
```

NOMECOMPLETO	SALARIO
KauanSilva	1100
JoseSouza	1200
EduardaOliveira	1200
LeonardoAmaral	3700
ThaissSilveira	6700
NicoleMoraes	1700
LeticiaMoreira	
JoaoSilva	
IgorPires	6700
DiogoPaizano	1500
KarlosPereira	1500
NOMECOMPLETO	SALARIO
StevanRodrigues	1500
OtavioMartez	1500

13 linhas selecionadas.

Observe que o Nome e Sobrenome ficaram exibidos juntos, sem um espaço separando um do outro. Isto é um comportamento normal da função concat(), que apenas concatena. Mas podemos tentar adicionar um espaço na condição. Será que a query irá funcionar?

```
SQL> select concat(nome, ' ',sobrenome) as nomeCompleto, salario from funcionarios;
select concat(nome, ' ',sobrenome) as nomeCompleto, salario from funcionarios
```

ERRO na linha 1:

ORA-00909: numero de argumentos invalido

O Oracle vai exibir a mensagem "numero de argumentos invalidos", porque a função CONCAT só aceita dois argumentos.

Teremos que encontrar outra solução. As **Single-Row Functions** nos permitem usar uma função dentro da outra. Então, antes de concatenar as colunas nome e sobrenome, iremos concatenar nome e ' ' (espaço).

```
SQL> select concat(concat(nome, ' '), sobrenome) as nomeCompleto, salario from funcionarios;
```

Agora, ele irá executar primeiro concat(nome, ' ') e depois, o restante.

```
SQL> select concat(concat(nome, ' '), sobrenome) as nomeCompleto, salario from funcionarios;
```

NOME COMPLETO	SALARIO
KauanSilva	1100
JoseSouza	1200
EduardaOliveira	1200
LeonardoAmaral	3700
ThaissSilveira	6700
NicoleMoraes	1700
LeticiaMoreira	
JoaoSilva	
IgorPires	6700
DiogoPaizano	1500
KarlosPereira	1500
NOME COMPLETO	SALARIO
StevanRodrigues	1500
OtavioMartez	1500

Kuan Silva	1100
Jose Souza	1200
Eduarda Oliveira	1200
Leonardo Amaral	3700
Thais Silveira	6700
Nicole Moraes	1700
Leticia Moreira	
Joao Silva	
Igor Pires	6700
Diogo Paizano	1500
Karlos Pereira	1500

NOMECOMPLETO	SALARIO
-----	-----
Stevan Rodrigues	1500
Otavio Martez	1500

13 linhas selecionadas.

É importante saber que podemos usar funções que trabalham em uma única linha, dentro uma da outra.

Também é válido conhecer o comportamento quando estas funções têm o argumento nulo.

```
SQL> select concat(null,sobrenome) as nomeCompleto,salario from funcionarios;
```

NOMECOMPLETO	SALARIO
-----	-----
Silva	1100
Souza	1200
Oliveira	1200
Amaral	3700
Silveira	6700
Moraes	1700
Moreira	
Silva	
Pires	6700
Paizano	1500
Pereira	1500

NOMECOMPLETO	SALARIO
-----	-----
Rodrigues	1500
Martez	1500

13 linhas selecionadas.

Caso o parâmetro seja `null`, ele retornará apenas o valor do outro. Mas e se os dois parâmetros da função `concat()` forem nulos?

```
SQL> select concat(null,null) as nomeCompleto,salario from funcionarios;
```

N	SALARIO
-----	-----
	1100

```
1200
1200
3700
6700
1700
```

```
6700
1500
1500
```

```
N    SALARIO
- -----
1    1500
1    1500
```

13 linhas selecionadas.

A função irá retornar o valor nulo, apenas quando os dois parâmetros forem `null`. Ela também é feita para trabalhar com texto. Caso seja adicionado um número como parâmetro, ela irá transformá-lo em texto também.

```
SQL> select concat(null,null) as nomeCompleto,salario from funcionarios;
```

```
N    SALARIO
- -----
1    1100
1    1200
1    1200
1    3700
1    6700
1    1700
1
1
1    6700
1    1500
1    1500
```

```
N    SALARIO
- -----
1    1500
1    1500
```

13 linhas selecionadas.

Existe também a possibilidade de quando formos fazer uma pesquisa, cometermos um engano e digitarmos uma letra em caixa alta, sem querer. Com a função `initcap()`, podemos pesquisar pela palavra `reNaN` e o Oracle irá retornar `Renan`, desconsiderando os caracteres maiúsculos no meio da palavra.

```
SQL> select initcap('reNaN') from dual;
```

```
INITC
-----
Renan
```

A função `INITC` irá retornar o termo com o primeiro caractere em caixa alta e os demais em caixa baixa. E se testarmos a função com o nome composto `isabel cristina`, com todas as letras digitadas em minúsculo.

```
SQL> select initcap('isabel cristina') from dual;  
  
INITC('ISABEL  
-----  
Isabel Cristina
```

O Oracle soube separar as palavras `isabel` e `cristina`. Porém, se por engano tivéssemos acrescentado um número entre as palavras, por exemplo `isabel1cristina`, ele ficaria confuso.

```
SQL> select initcap('isabel1cristina') from dual;  
  
INITC('ISABEL  
-----  
isabel1cristina
```

Mas se tivéssemos digitado `*` entre as palavras, o Oracle teria reconhecido as duas.

```
SQL> select initcap('isabel*cristina') from dual;  
  
INITC('ISABEL  
-----  
Isabel*Cristina
```

Como o Oracle consegue identificar que temos duas palavras no texto? Ele identifica qual é o primeiro caractere não alfa numérico e quebra as palavras. Então, quando ele encontrou espaço ou asterisco, a função `initcap()` soube quebrar as palavras e adicionou uma letra maiúscula na primeira letra de cada nome.

E vamos repetir o teste de comportamento da função com um dos parâmetros nulo.

```
SQL> select initcap(null) from dual;  
  
I  
-
```

Ele irá retornar um valor nulo, caindo na regrinha das funções para `string`. E se passarmos o número `1`, por exemplo, ele irá retornar `1`. Isto acontece porque ele tem o mesmo comportamento da `concat()` e transformará o número em uma `string`.

Vimos já duas funções para trabalhar com texto: `CONCAT` e `INITCAP`. A `concat()` tem o mesmo comportamento de outro operador que temos nativos no Oracle: `||`. Para concatenar as colunas `nome` e `sobrenome` da tabela `funcionarios`, poderíamos usar a seguinte `query`:

```
SQL> select nome || sobrenome as nomeCompleto from funcionarios;  
  
NOMECOMPLETO  
-----  
SALARIO  
-----
```

KauanSilva	1100
JoseSouza	1200
EduardaOliveira	1200
LeonardoAmaral	3700
ThaissSilveira	6700
NicoleMoraes	1700
LeticiaMoreira	
JoaoSilva	
IgorPires	6700
DiogoPaizano	1500
KarlosPereira	1500

NOMECOMPLETO	SALARIO
-----	-----
StevanRodrigues	1500
OtavioMartez	1500

13 linhas selecionadas.

Então, equivalente ao `concat()` , temos o operador de concatenação ( `||` ).

