02

Trabalhando com valores nulos

Transcrição

Temos os dados da tabela `funcionarios` .

```
SQL> select * from funcionarios;

        ID NOME                 SOBRENOME          SALARIO PCT_COMISSAO
---------- -------------------- --------------- --------- ------------
         1 Kauan                Silva                1100           ,3
         2 Jose                 Souza                1200           ,3
         3 Eduarda              Oliveira             1200           ,2
         4 Leonardo             Amaral               3700
         5 Thais                Silveira             6700
         6 Nicole               Moraes               1700           ,3
         7 Leticia              Moreira                             ,9
         8 Joao                 Silva                6700           ,3
         9 Igor                 Pires                1500            0
        10 Diogo                Paizano              1500           ,2
        11 Karlos               Pereira              1500

        ID NOME                 SOBRENOME          SALARIO PCT_COMISSAO
---------- -------------------- --------------- --------- ------------
        12 Stevan               Rodrigues            1500
        13 Otavio               Martez               1500
```

Agora, quero incluir uma regra diferente. Quero calcular o salário do funcionário somado ao salário multiplicado pela porcentagem da comissão. Também quero calcular, que nos casos casos em que a comissão não seja definida, quero retornar o salário somado a um bônus de R$100. E na hipótese do funcionário não ter um salário e a comissão definida, que o valor recebido por ele, seja de R$800. Como podemos fazer isto? Para calcular o valor do salário bruto, iremos usar o `case` para adicionar a primeira condição. Na terceira linha, temos certeza que ele receberá um salário e a comissão não é nula, então (then) faremos o calculo citado anteriormente. O terceiro caso será quando tanto o salário como a comissão não estarão definidos.

```
SQL> select nome, salario, (
  2 case when salario is null then 800
  3 when pct_comissao is not null the salario + salario * pct_comissao
  4 else salario+100 end) as salarioFinal from funcionarios;
```

Ele irá retornar a seguinte tabela:

```
NOME                 SALARIO SALARIOFINAL
-------------------- ---------- ------------
Kauan                    1100         1430
Jose                     1200         1560
Eduarda                  1200         1440
Leonardo                 3700         3800
Thais                    6700         6800
Nicole                   1700         2210
```

```
Leticia                                800
Joao                                   800
Igor                      6700         6700
Diogo                     1500         1800
Karlos                    1500         1600

NOME                   SALARIO SALARIOFINAL
------------------- ---------- ------------
Stevan                    1500         1600
Otavio                    1500         1600

13 linhas selecionadas.
```

Vemos que os valores do `salarioFinal` ficaram correto, então, conseguimos fazer os cálculos usando `case` . No entanto, com estas três condições, nossa consulta ficou confusa. Além disso, queríamos que ele retornasse primeiramente os valores que não fossem nulos. Para isto, seria mais simples usar a função `coalesce()` e especificar quais valores queremos que ela retorne:

```
coalesce(salario + salario * pct_comissao, salario+100,800)
```

Se os valores estão definidos, quero que a função retorne o resultado da conta. Veremos que obteremos o mesmo resultado que a tabela anterior.

```
SQL> select nome,salario, coalesce(salario + salario * pct_comissao,salario+100,800) as salarioFinal

NOME                   SALARIO SALARIOFINAL
------------------- ---------- ------------
Kauan                     1100         1430
Jose                      1200         1560
Eduarda                   1200         1440
Leonardo                  3700         3800
Thais                     6700         6800
Nicole                    1700         2210
Leticia                                800
Joao                                   800
Igor                      6700         6700
Diogo                     1500         1800
Karlos                    1500         1600

NOME                   SALARIO SALARIOFINAL
------------------- ---------- ------------
Stevan                    1500         1600
Otavio                    1500         1600

13 linhas selecionadas.
```

Porém, a leitura da *query* ficou mais simples. É bastante comum querermos retornar o primeiro valor não nulo de uma lista. Com `coalesce()` , nós já vimos as três principais funções para trabalharmos com valores nulos dentro do Oracle. Porém, precisamos ficar atentos durante a prova, porque estes temas não serão cobrados separadamente. Não espere questões como "qual é a utilidade da função `nvl()` ?". Em geral, o enunciado virá semelhante à última *query* usada, na qual temos algumas

condições que podem retornar um valor nulo e você deve responder como será a saída. Ou talvez, seja pedido que você defina qual a *query* para retornar os dados em determinadas condições.

Por isso, **fique atento**! Qualquer cálculo que possa receber um campo com valor nulo poderá retornar um resultado igualmente nulo.

Em seguida, iremos aprender a trabalhar com ***strings***, ou seja, com textos.