

Exercício 5

```
1 SET SERVEROUTPUT ON
2
3 CREATE OR REPLACE PROCEDURE prcAtualizarBonusCamareiras
4 (mes IN NUMBER, ano IN NUMBER DEFAULT (EXTRACT(YEAR FROM sysdate)))
5 IS
6 Mes_Invalido EXCEPTION; -- variavel para as excecoes
7 Ano_Invalido EXCEPTION; -- variavel para as excecoes
8 v_id_camareira Linha_Conta_Consumo.Id_Camareira%type;
9 v_consumo INTEGER;
10 v_camareira Camareira.Id%type;
11 CURSOR cursor_consumo IS -- cursor com o if da camareira
12     SELECT Id_Camareira
13     FROM Linha_Conta_Consumo,
14     Camareira
15     WHERE Linha_Conta_Consumo.Id_Camareira = Camareira.Id
16     GROUP BY Id_Camareira;
17 BEGIN
18     -- verificar se o ano e maior que 2020
19     IF ano > extract(year from sysdate) THEN
20         RAISE Ano_Invalido;
21     END IF;
22     -- verificar se o mes está entre 1 e 12
23     IF mes > 12 OR mes < 1 THEN
24         RAISE Mes_Invalido;
25     END IF;
26
27     OPEN cursor_consumo;
28 LOOP
29     FETCH cursor_consumo INTO v_id_camareira;
30     EXIT WHEN cursor_consumo%NOTFOUND;
31
32     SELECT Id_Camareira, SUM(Quantidade * Preco_Unitario) -- atribuir o id da camareira a variavel v_camareira e o consumo a variavel v_consumo
33     INTO v_camareira, v_consumo
34     FROM Linha_Conta_Consumo
35     WHERE Linha_Conta_Consumo.Id_Camareira = v_id_camareira
36     AND EXTRACT(YEAR FROM Data_Registo) = ano
37     AND EXTRACT(MONTH FROM Data_Registo) = mes
38     GROUP BY id_Camareira;
39
40     -- aplicar criterios do bonus: >1000-15%/ <= 1000 e >=500-10%/<500 e >100-5%/ <=100 sem bônus
41 CASE WHEN v_consumo <= 100 THEN
42     UPDATE Camareira
43     SET Bonus = 0
44     WHERE Camareira.Id = v_camareira;
45
46     WHEN v_consumo < 500 AND v_consumo > 100 THEN
47     UPDATE Camareira
48     SET Bonus = v_consumo * 0.05
49     WHERE CAMAREIRA.ID = v_camareira;
50
51     WHEN v_consumo <= 1000 AND v_consumo >= 500 THEN
52     UPDATE Camareira
53     SET Bonus = v_consumo * 0.10
54     WHERE CAMAREIRA.ID = v_camareira;
55
56     WHEN v_consumo > 1000 THEN
57     UPDATE Camareira
58     SET Bonus = v_consumo * 0.15
59     WHERE CAMAREIRA.ID = v_camareira;
60
61     END CASE;
62 END LOOP;
63 EXCEPTION
64 WHEN Ano_Invalido THEN
65     RAISE_APPLICATION_ERROR(-20000, 'Ano invalido. Tente novamente');
66 WHEN Mes_Invalido THEN
67     RAISE_APPLICATION_ERROR(-20000, 'Mes invalido. Tente novamente');
68 END;
```

```

70  -- testes
71  DECLARE
72  v_bonus Camareira.Bonus%TYPE;
73  v_id_camareira Camareira.Id%TYPE;
74  BEGIN
75      prcAtualizarBonusCamareiras(4,2020);
76  END;

```

Este exercício tem como objetivo implementar um procedimento que permita atualizar o valor de um bônus às camareiras mediante o valor do total dos consumos registados num dado mês de um dado ano. O mês é passado por parâmetro e o ano é opcional. Na ausência deste último deve ser considerado o ano atual do sistema. O valor do bônus segue os seguintes critérios:

Critério	Bónus
>1000	15% do valor
<= 1000 e >=500	10% do valor
<500 e >100	5% do valor
<=100	sem bónus

De facto, para atualizarmos os bónus da Camareira tivemos que adicionar um atributo a esta mesma tabela “Camareira” denominado de “Bonus” que se encontra atualizado no Modelo Relacional.

Antes de se executar o procedimento verificou-se a tabela “Camareira” e esta não apresentava valores de bónus:

	BONUS	ID
1	{null}	11
2	{null}	12
3	{null}	13
4	{null}	14
5	{null}	15
6	{null}	16
7	{null}	17
8	{null}	18
9	{null}	19
10	{null}	20

Após a verificação testou-se o procedimento e verificaram-se as exceções utilizadas. Em primeiro lugar utilizou-se um mês não existente:

```

Error starting at line : 71 in command -
DECLARE
v_bonus Camareira.Bonus%TYPE;
v_id_camareira Camareira.Id%TYPE;
BEGIN
    prcAtualizarBonusCamareiras(34,2020);
END;
Error report -
ORA-20000: Mes invalido. Tente novamente

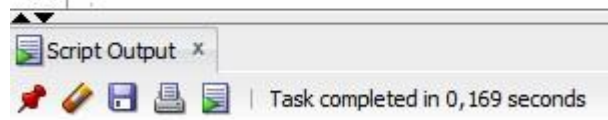
```

De seguida, utilizou-se o ano de 2021:

```
Error starting at line : 71 in command -  
DECLARE  
v_bonus Camareira.Bonus%TYPE;  
v_id_camareira Camareira.Id%TYPE;  
BEGIN  
    prcAtualizarBonusCamareiras(4,2021);  
END;  
Error report -  
ORA-20000: Ano invalido. Tente novamente
```

Após isto, colocou-se apenas o mês sendo que o ano considerado foi o atual do sistema:

```
70 -- testes  
71 DECLARE  
72 v_bonus Camareira.Bonus%TYPE;  
73 v_id_camareira Camareira.Id%TYPE;  
74 BEGIN  
75     prcAtualizarBonusCamareiras(4);  
76 END;  
77  
78  
79
```



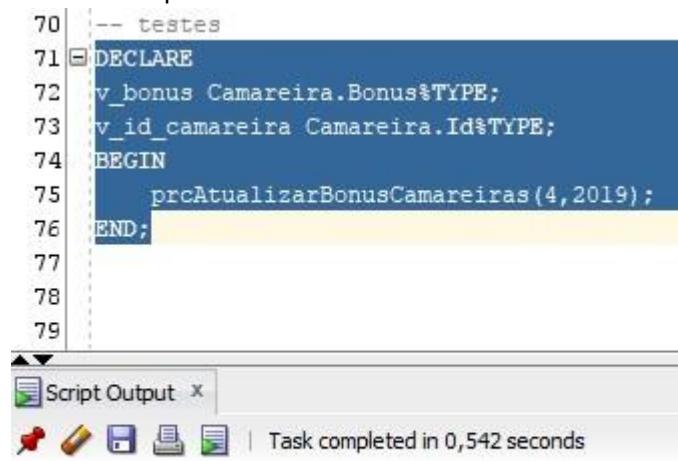
PL/SQL procedure successfully completed.

Consultou-se novamente a tabela “Camareira” e verificou-se que o atributo “Bonus” foi atualizado para o mês de abril do ano de 2020:

	BONUS	ID
1	10,2	11
2	8,45	12
3	6,55	13
4	0	14
5	6,5	15
6	9,9	16
7	10,85	17
8	11,6	18
9	9,6	19
10	0	20

Por fim, colocou-se como parâmetros o mês de abril e o ano de 2019 e o resultado foi:

```
70  -- testes
71  DECLARE
72  v_bonus Camareira.Bonus%TYPE;
73  v_id_camareira Camareira.Id%TYPE;
74  BEGIN
75      prcAtualizarBonusCamareiras(4,2019);
76  END;
```



PL/SQL procedure successfully completed.

Consultou-se novamente a tabela “Camareira” e verificou-se que o atributo “Bonus” foi atualizado para o mês de abril do ano de 2019:

	BONUS	ID
1	9,45	11
2	7,65	12
3	6,1	13
4	5,55	14
5	5,9	15
6	9,25	16
7	10,5	17
8	10,65	18
9	6,9	19
10	0	20