

Exercício 4

```
1  SET SERVEROUTPUT ON
2
3  -- funcao auxiliar para ver se o ano e bissexto
4  CREATE OR REPLACE FUNCTION ano_bissexto(anoB IN NUMBER)
5      RETURN BOOLEAN
6      IS
7      ano_bissexto1 BOOLEAN := FALSE;
8  BEGIN
9      IF (MOD(anoB, 4) = 0)
10         AND ((MOD(anoB, 100) <> 0) OR (MOD(anoB, 400) = 0))
11      THEN
12         ano_bissexto1 := TRUE;
13      END IF;
14
15      RETURN ano_bissexto1;
16  END;
17
```

```
18 --funcao auxiliar para calcular o numero de dias num mes
19 CREATE OR REPLACE FUNCTION days_in_a_month(mes1 IN NUMBER, anoB IN NUMBER)
20     RETURN NUMBER
21     IS
22     days_in_month NUMBER(2);
23  BEGIN
24     CASE -- verificar os meses que tem 31 dias
25     WHEN (mes1 = 1
26         OR mes1 = 3
27         OR mes1 = 5
28         OR mes1 = 7
29         OR mes1 = 8
30         OR mes1 = 10
31         OR mes1 = 12
32         )
33     THEN
34         days_in_month := 31;
35     WHEN mes1 = 2 -- se for fevereiro teremos de ver se o ano e bissexto
36     THEN
37         IF (ano_bissexto(anoB))
38         THEN
39             days_in_month := 29; -- e bissexto logo tem 29 dias
40         ELSE
41             days_in_month := 28; -- e comum logo tem 28 dias
42         END IF;
43     ELSE
44         days_in_month := 30; -- se nao for nem fevereiro nem os restantes meses com 31 dias entao tem 30 dias
45     END CASE;
46
47     RETURN days_in_month;
48  END;
```

```

50 CREATE OR REPLACE FUNCTION fncObterRegistoMensalCamareira
51 (mes IN INTEGER, ano IN INTEGER DEFAULT (EXTRACT(YEAR FROM sysdate) - 1)) RETURN SYS_REFCURSOR --retorna o cursor
52 IS
53 c_camareira SYS_REFCURSOR; -- cria o cursor
54 Mes_Invalido EXCEPTION; -- variavel para as excecoes
55 Ano_Invalido EXCEPTION; -- variavel para as excecoes
56 BEGIN
57 -- verificar se o ano e maior que 2020
58 IF ano>extract(year from sysdate) THEN
59 RAISE Ano_Invalido;
60 END IF;
61 -- verificar se o mes está entre 1 e 12
62 IF mes>12 OR mes<1 THEN
63 RAISE Mes_Invalido;
64 END IF;
65
66 --identificador único da camareira, nome da camareira, valor total dos consumos registados, data do primeiro registo
67 -- de consumo e data do último registo de consumo e quantidade de dias em que não foram registados quaisquer consumos
68 OPEN c_camareira FOR
69 SELECT f.Id, f.nome "Nome", SUM(lcc.Quantidade*lcc.Preco_Unitario) "Valor_total_consumos", MIN(lcc.Data_Registo) "Primeiro_registo",
70 MAX(lcc.Data_Registo) "Ultimo_registo", days_in_a_month(mes, ano) - COUNT(DISTINCT lcc.Data_Registo)
71 FROM Funcionario f,Linha_Conta_Consumo lcc
72 WHERE lcc.Id_Camareira=f.Id
73 AND mes = EXTRACT(Month FROM lcc.Data_Registo)
74 AND ano = EXTRACT(Year FROM lcc.Data_Registo)
75 GROUP BY f.Id, f.nome
76 ORDER BY f.Id ASC;
77 RETURN c_camareira;
78
79 EXCEPTION
80 WHEN Ano_Invalido THEN
81 RAISE_APPLICATION_ERROR(-20000, 'Ano invalido. Tente novamente');
82 WHEN Mes_Invalido THEN
83 RAISE_APPLICATION_ERROR(-20000, 'Mes invalido. Tente novamente');
84
85 END;
86
87 -- testes
88 DECLARE
89 cursor_camareira SYS_REFCURSOR;
90 v_id_camareira Linha_Conta_Consumo.Id_Camareira%TYPE;
91 v_nome_camareira Funcionario.nome%TYPE;
92 v_total_consumos INTEGER;
93 v_data_primeiro_registo DATE;
94 v_data_ultimo_registo DATE;
95 v_dias_sem_consumos INTEGER;
96 BEGIN
97 cursor_camareira := fncObterRegistoMensalCamareira(6,2021);
98 LOOP
99 FETCH cursor_camareira INTO v_id_camareira, v_nome_camareira, v_total_consumos, v_data_primeiro_registo, v_data_ultimo_registo, v_dias_sem_consumos;
100 EXIT WHEN cursor_camareira%NOTFOUND;
101 dbms_output.put_line('ID da camareira: ' || v_id_camareira || ' | Nome da camareira: ' || v_nome_camareira || ' | Valor total dos consumos: ' || v_total_consumos ||
102 ' | Data do primeiro registo: ' || v_data_primeiro_registo || ' | Data do ultimo registo: ' ||
103 v_data_ultimo_registo || ' | Dias sem qualquer consumo: ' || v_dias_sem_consumos);
104 END LOOP;
105 END;

```

Este exercício tem como objetivo implementar uma função que retorne um cursor com informações relativas aos consumos registados pelas camareiras. Esta recebe o mês e o ano (opcional) como parâmetro. O cursor retornado tem as seguintes informações: identificador único da camareira, nome da camareira, valor total dos consumos registados, data do primeiro registo de consumo, data do último registo de consumo e quantidade de dias em que não foram registados quaisquer consumos. De notar que, caso o valor do ano não seja passado por parâmetro teremos de considerar o ano anterior ao ano atual do sistema.

De facto, como o exercício pede para contar os dias em que não houve consumos, foi necessário implementar uma função auxiliar, “days_in_a_month”, que calculasse o número de dias num mês. Contudo, e apesar da maior parte dos meses se dividir entre 30 e 31 dias, o mês fevereiro divide-se em 28 ou 29 dias consoante o ano: comum ou bissexto, respetivamente.

Assim, de forma a solucionar este problema foi necessário implementar outra função auxiliar, “ano_bissext”, que verificasse se o ano tinha 365 ou 366 dias. Após isto tudo implementou-se a função pedida no enunciado.

Como referido no enunciado, caso o ano não seja passado por parâmetro deve-se considerar o ano como o anterior ao do sistema: 2019. Assim, e para o mês de junho os resultados foram:

```

92 BEGIN
93   cursor_camareira := fncObterRegistoMensualCamareira(6);
94   LOOP
95     FETCH cursor_camareira INTO v_id_camareira, v_nome_camareira, v_total_consumos, v_data_primeiro_registo, v_data_ultimo_registo, v_dias_sem_consumos;
96     EXIT WHEN cursor_camareira%NOTFOUND;
97     dbms_output.put_line('ID da camareira: ' || v_id_camareira || ' | Nome da camareira: ' || v_nome_camareira || ' | Valor total dos consumos: ' || v_total_consumos ||
98                           ' | Data do primeiro registo: ' || v_data_primeiro_registo || ' | Data do ultimo registo: ' ||
99                           v_data_ultimo_registo || ' | Dias sem qualquer consumo: ' || v_dias_sem_consumos);
100   END LOOP;
101 END;
102

```

Script Output x

Task completed in 1,024 seconds

ID da camareira: 11	Nome da camareira: Camareira 1	Valor total dos consumos: 186	Data do primeiro registo: 19.06.01	Data do ultimo registo: 19.06.29	Dias sem qualquer consumo: 11
ID da camareira: 12	Nome da camareira: Camareira 2	Valor total dos consumos: 187	Data do primeiro registo: 19.06.01	Data do ultimo registo: 19.06.30	Dias sem qualquer consumo: 8
ID da camareira: 13	Nome da camareira: Camareira 3	Valor total dos consumos: 117	Data do primeiro registo: 19.06.03	Data do ultimo registo: 19.06.29	Dias sem qualquer consumo: 10
ID da camareira: 14	Nome da camareira: Camareira 4	Valor total dos consumos: 102	Data do primeiro registo: 19.06.02	Data do ultimo registo: 19.06.30	Dias sem qualquer consumo: 8
ID da camareira: 15	Nome da camareira: Camareira 5	Valor total dos consumos: 74	Data do primeiro registo: 19.06.02	Data do ultimo registo: 19.06.30	Dias sem qualquer consumo: 12
ID da camareira: 16	Nome da camareira: Camareira 6	Valor total dos consumos: 148	Data do primeiro registo: 19.06.01	Data do ultimo registo: 19.06.30	Dias sem qualquer consumo: 12
ID da camareira: 17	Nome da camareira: Camareira 7	Valor total dos consumos: 158	Data do primeiro registo: 19.06.02	Data do ultimo registo: 19.06.30	Dias sem qualquer consumo: 13
ID da camareira: 18	Nome da camareira: Camareira 8	Valor total dos consumos: 203	Data do primeiro registo: 19.06.01	Data do ultimo registo: 19.06.29	Dias sem qualquer consumo: 12
ID da camareira: 19	Nome da camareira: Camareira 9	Valor total dos consumos: 154	Data do primeiro registo: 19.06.01	Data do ultimo registo: 19.06.30	Dias sem qualquer consumo: 10
ID da camareira: 20	Nome da camareira: Camareira 10	Valor total dos consumos: 57	Data do primeiro registo: 19.06.01	Data do ultimo registo: 19.06.30	Dias sem qualquer consumo: 9

PL/SQL procedure successfully completed.

Como se verifica, o ano considerado foi 2019. Para o output ser este foi necessário alterar os dados dos “inserts” e acrescentar dados na tabela “Epoca” e “Reserva” pois estas só consideravam o ano de 2020. Assim, na linha 337 foi colocada a data de “2019-01-01”

```

319 --reservas (N por cada dia, de 1-1-2019 até 31-12-2020)
320 DECLARE
321     K_Count          INT          := 10;
322     V_Count          INT;
323     V_Label          VARCHAR(50) := 'Reserva';
324     V_Di             DATE;
325     V_Df             DATE;
326     V_D              DATE;
327     V_Id_Cliente     Cliente.Id%TYPE;
328     V_Step           INT          := 7; --de K em K reserv
329     V_Id_Reserva     INT;
330     V_Nr_Pessoas     INT;
331     V_Id_Tipo_Quarto INT;
332     V_Dias           INT;
333     V_Lag_Dias       INT;
334     V_Preco          NUMBER;
335 BEGIN
336     DELETE FROM Reserva;
337     V_Di := TO_DATE('2019-01-01', 'yyyy-mm-dd');
338     V_Df := TO_DATE('2020-12-31', 'yyyy-mm-dd');

```


Na tabela “Epoca” foram adicionadas épocas em 2019.

```

279 --içáspocas 2020
280 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
281 VALUES (1, 'içáspoca 1', TO_DATE('2020-01-01', 'yyyy-mm-dd'), TO_DATE('2020-03-31', 'yyyy-mm-dd'));
282 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
283 VALUES (2, 'içáspoca 2', TO_DATE('2020-04-01', 'yyyy-mm-dd'), TO_DATE('2020-06-30', 'yyyy-mm-dd'));
284 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
285 VALUES (3, 'içáspoca 3', TO_DATE('2020-07-01', 'yyyy-mm-dd'), TO_DATE('2020-09-30', 'yyyy-mm-dd'));
286 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
287 VALUES (4, 'içáspoca 4', TO_DATE('2020-09-01', 'yyyy-mm-dd'), TO_DATE('2020-12-31', 'yyyy-mm-dd'));
288
289 --epocas 2019
290 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
291 VALUES (5, 'Epoca 5', TO_DATE('2019-01-01', 'yyyy-mm-dd'), TO_DATE('2019-03-31', 'yyyy-mm-dd'));
292 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
293 VALUES (6, 'Epoca 6', TO_DATE('2019-04-01', 'yyyy-mm-dd'), TO_DATE('2019-06-30', 'yyyy-mm-dd'));
294 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
295 VALUES (7, 'Epoca 7', TO_DATE('2019-07-01', 'yyyy-mm-dd'), TO_DATE('2019-09-30', 'yyyy-mm-dd'));
296 INSERT INTO Epoca(Id, Nome, Data_Ini, Data_Fim)
297 VALUES (8, 'Epoca 8', TO_DATE('2019-09-01', 'yyyy-mm-dd'), TO_DATE('2019-12-31', 'yyyy-mm-dd'));
---
```

De seguida testou-se a função como o mês de junho e o ano de 2020 e o output foi o seguinte:

```

93 cursor_camareira := fncObterRegistoMensalCamareira(6,2020);
94 LOOP
95     FETCH cursor_camareira INTO v_id_camareira, v_nome_camareira, v_total_consumos, v_data_primeiro_registo, v_data_ultimo_registo, v_dias_sem_consumos;
96     EXIT WHEN cursor_camareira%NOTFOUND;
97     dbms_output.put_line('ID da camareira: ' || v_id_camareira || ' | Nome da camareira: ' || v_nome_camareira || ' | Valor total dos consumos: ' || v_total_consumos ||
98         ' | Data do primeiro registo: ' || v_data_primeiro_registo || ' | Data do ultimo registo: ' ||
99         v_data_ultimo_registo || ' | Dias sem qualquer consumo: ' || v_dias_sem_consumos);
100 END LOOP;
101 END;
102
```

ID da camareira	Nome da camareira	Valor total dos consumos	Data do primeiro registo	Data do ultimo registo	Dias sem qualquer consumo
11	Camareira 1	153	20.06.01	20.06.29	10
12	Camareira 2	131	20.06.02	20.06.30	10
13	Camareira 3	119	20.06.03	20.06.30	12
14	Camareira 4	107	20.06.02	20.06.30	11
15	Camareira 5	109	20.06.01	20.06.30	9
16	Camareira 6	170	20.06.01	20.06.29	10
17	Camareira 7	175	20.06.02	20.06.30	10
18	Camareira 8	184	20.06.01	20.06.30	8
19	Camareira 9	164	20.06.01	20.06.30	9
20	Camareira 10	70	20.06.01	20.06.30	8

PL/SQL procedure successfully completed.

De facto, na função “fncObterRegistoMensalCamareira” foram tratadas duas exceções: ano inválido e mês inválido. Assim, testou-se esta mesma função para o ano 2021 e o resultado foi:

```
Error starting at line : 88 in command -
DECLARE
    cursor_camareira SYS_REFCURSOR;
    v_id_camareira Linha_Conta_Consumo.Id_Camareira%TYPE;
    v_nome_camareira Funcionario.nome%TYPE;
    v_total_consumos INTEGER;
    v_data_primeiro_registo DATE;
    v_data_ultimo_registo DATE;
    v_dias_sem_consumos INTEGER;
BEGIN
    cursor_camareira := fncObterRegistoMensalCamareira(6,2021);
    LOOP
        FETCH cursor_camareira INTO v_id_camareira, v_nome_camareira, v_total_consumos, v_data_primeiro_registo, v_data_ultimo_registo, v_dias_sem_consumos;
        EXIT WHEN cursor_camareira%NOTFOUND;
        dbms_output.put_line('ID da camareira: ' || v_id_camareira || ' | Nome da camareira: ' || v_nome_camareira || ' | Valor total dos consumos: ' || v_total_consumos ||
            ' | Data do primeiro registo: ' || v_data_primeiro_registo || ' | Data do ultimo registo: ' ||
            v_data_ultimo_registo || ' | Dias sem qualquer consumo: ' || v_dias_sem_consumos);
    END LOOP;
END;
Error report -
ORA-20000: Ano invalido. Tente novamente
```

Ainda assim testou-se também para um mês inválido e o output foi:

```
Error starting at line : 88 in command -
DECLARE
    cursor_camareira SYS_REFCURSOR;
    v_id_camareira Linha_Conta_Consumo.Id_Camareira%TYPE;
    v_nome_camareira Funcionario.nome%TYPE;
    v_total_consumos INTEGER;
    v_data_primeiro_registo DATE;
    v_data_ultimo_registo DATE;
    v_dias_sem_consumos INTEGER;
BEGIN
    cursor_camareira := fncObterRegistoMensalCamareira(36,2020);
    LOOP
        FETCH cursor_camareira INTO v_id_camareira, v_nome_camareira, v_total_consumos, v_data_primeiro_registo, v_data_ultimo_registo, v_dias_sem_consumos;
        EXIT WHEN cursor_camareira%NOTFOUND;
        dbms_output.put_line('ID da camareira: ' || v_id_camareira || ' | Nome da camareira: ' || v_nome_camareira || ' | Valor total dos consumos: ' || v_total_consumos ||
            ' | Data do primeiro registo: ' || v_data_primeiro_registo || ' | Data do ultimo registo: ' ||
            v_data_ultimo_registo || ' | Dias sem qualquer consumo: ' || v_dias_sem_consumos);
    END LOOP;
END;
Error report -
ORA-20000: Mes invalido. Tente novamente
```