

## Exercício 2

```

CREATE OR REPLACE PROCEDURE Prc_Check_Out(Row_Reserva Reserva%ROWTYPE)
IS
  -- cursores e respectivas variáveis iterativas
  CURSOR C_Contas_Consumo IS
    SELECT Id
    FROM Conta_Consumo
    WHERE Id_Reserva = Row_Reserva.Id;
  V_Id_Conta_Consumo_Iteracao Conta_Consumo.Id%TYPE;
  --
  CURSOR C_Linhas_Conta_Consumo
    (V_Id_Conta_Consumo Conta_Consumo.Id%TYPE) IS
    SELECT *
    FROM Linha_Conta_Consumo
    WHERE Id_Conta_Consumo = V_Id_Conta_Consumo;
  Row_Linha_Conta_Consumo      Linha_Conta_Consumo%ROWTYPE;
  --
  CURSOR C_Linhas_Fatura
    (V_Id_Fatura Fatura.Id%TYPE) IS
    SELECT Valor_Consumo
    FROM Linha_Fatura
    WHERE Id_Fatura = V_Id_Fatura;
  V_Valor_Consumo              Linha_Fatura.Valor_Consumo%TYPE;

  -- outras variáveis
  V_Id_Fatura                   Fatura.Id%TYPE;
  V_Preco_Artigo                Artigo_Consumo.Preco%TYPE;
  Dummy                         INTEGER;
  Nao_Tem_Checkin EXCEPTION;
  Ja_Tem_Fatura EXCEPTION ;
BEGIN
  -- verificar se a reserva ainda não fez o checkin
  SELECT COUNT(*)
  INTO Dummy
  FROM Checkin
  WHERE Id_Reserva = Row_Reserva.Id;
  IF Dummy != 1 THEN
    RAISE Nao_Tem_Checkin;
  END IF;
  -- verificar se a reserva já tem uma fatura associada
  SELECT COUNT(*)
  INTO Dummy
  FROM Fatura
  WHERE Id_Reserva = 51;
  IF Dummy = 1 THEN
    RAISE Ja_Tem_Fatura;
  END IF;

```

```
-- ## criar nova fatura
V_Id_Fatura := Seq_Fatura.Nextval;
INSERT INTO Fatura(Id, Numero, Data, Id_Cliente,
                  Id_Reserva, Valor_Faturado_Reserva)
VALUES (V_Id_Fatura, V_Id_Fatura, SYSDATE,
        Row_Reserva.Id_Cliente, Row_Reserva.Id, Row_Reserva.Preco);

-- ## para linha-conta-consumo de cada conta-consumo da reserva,
-- criar uma linha-fatura ##
OPEN C_Contas_Consumo;
LOOP
    FETCH C_Contas_Consumo INTO V_Id_Conta_Consumo_Iteracao;

    -- a linhas associadas à conta
    OPEN C_Linhas_Conta_Consumo(V_Id_Conta_Consumo_Iteracao);
    LOOP
        FETCH C_Linhas_Conta_Consumo INTO Row_Linha_Conta_Consumo;

        -- o custo do artigo da linha
        SELECT Preco
        INTO V_Preco_Artigo
        FROM Artigo_Consumo
        WHERE Id = Row_Linha_Conta_Consumo.Id_Artigo_Consumo;

        INSERT INTO Linha_Fatura(Id_Fatura, Linha, Id_Conta_Consumo,
                                Linha_Conta_Consumo, Valor_Consumo)
        VALUES (V_Id_Fatura, Seq_Linha_Fatura.Nextval, V_Id_Conta_Consumo_Iteracao,
                Row_Linha_Conta_Consumo.Linha, V_Preco_Artigo * Row_Linha_Conta_Consumo.Quantidade);

        EXIT WHEN C_Linhas_Conta_Consumo%NOTFOUND;
    END LOOP;
    CLOSE C_Linhas_Conta_Consumo;

    EXIT WHEN C_Contas_Consumo%NOTFOUND;
END LOOP;
CLOSE C_Contas_Consumo;

-- ## colocar o valor dos consumos na fatura
OPEN C_Linhas_Fatura(V_Id_Fatura);
LOOP
    FETCH C_Linhas_Fatura INTO V_Valor_Consumo;

    UPDATE Fatura
    SET Valor_Faturado_Consumo = V_Valor_Consumo
    WHERE Id = V_Id_Fatura;
```

```

        EXIT WHEN C_Linhas_Fatura%NOTFOUND;
    END LOOP;
    CLOSE C_Linhas_Fatura;

    -- alterar estado da reserva
    UPDATE Reserva
    SET Id_Estado_Reserva = (
        SELECT Id
        FROM Estado_Reserva
        WHERE Nome LIKE 'Check%out'
    )
    WHERE Id = Row_Reserva.Id;

    -- criar checkout
    INSERT INTO Checkout(Id_Reserva, Data)
    VALUES (Row_Reserva.Id, SYSDATE);

    RETURN;

EXCEPTION
    WHEN Nao_Tem_Checkin THEN
        RAISE_APPLICATION_ERROR(-20000, 'A reserva dada como parâmetro não tem checkin efetuado.');
```

```

    WHEN Ja_Tem_Fatura THEN
        RAISE_APPLICATION_ERROR(-20000, 'A reserva dada como parâmetro já tem uma fatura associada.');
```

```

END;
```

Para executarmos o procedimento, precisamos de uma reserva que tenham efetuado check-in mas não o check-out. Nos dados existentes não existe nenhuma reserva nesta situação, por isso vamos fazer com que duas fiquem:

```

800AD_DH2> DELETE
            FROM Checkout
            WHERE Id_Reserva IN (51, 52)
[2020-12-18 10:33:24] completed in 64 ms
800AD_DH2> DELETE
            FROM Linha_Fatura
            WHERE Id_Fatura IN (
                SELECT Id
                FROM Fatura
                WHERE Id_Reserva IN (51, 52)
            )
[2020-12-18 10:33:24] completed in 78 ms
800AD_DH2> DELETE
            FROM Fatura
            WHERE Id_Reserva IN (51, 52)
[2020-12-18 10:33:25] completed in 58 ms
```

Resultado da query que busca todas as reservas com checkin efetuado, mas não checkout:

ID	ID_CLIENTE	ID_TIPO_QUARTO	data entrada	data saída	NR_PESSOAS	PRECO	ID_ESTADO_RESERVA
51	41	1	27/01/2019	28/01/2019	4	40.00	2
52	32	2	28/01/2019	30/01/2019	1	60.00	2

Todos os checkouts, respetiva fatura, e respetivas linhas de fatura foram apagados.

Esta query permite-nos visualizar todos os consumos (e respetivas quantidades) de todas as linhas de conta de consumo de todas as contas de consumo da reserva 51:

id artigo	PREÇO	QUANTIDADE	LINHA
51	10.00	1	1

Execução do procedimento através de um bloco anónimo:

```

BDDAD_DH2> DECLARE
    Row_Reserva Reserva%ROWTYPE;
BEGIN
    SELECT *
    INTO Row_Reserva
    FROM Reserva
    WHERE Id = 51;
    Prc_Check_Out(Row_Reserva);
END;
[2020-12-18 10:40:13] completed in 1 s 13 ms

```

Visualização de todas as faturas existentes:

Id	Numero	Data	Id_Cliente	Id_Reserva	Valor_Faturado_Reserva	Valor_Faturado_Consumo
1	1	2020-12-18 10:40:08	41	51	40.00	10.00

O valor faturado em consumos está de acordo com o previsto. O valor faturado na reserva foi copiado da reserva para a fatura, e também está de acordo com o que seria previsível.

Além disso, o estado da reserva também deverá ter sido atualizado:

"ID reserva"	"nome do estado"	"ID estado"
51	Check-out	3

E deverá ter sido adicionada uma linha à tabela checkout:

Id_Reserva	Data	Observacoes	Valor_Extra
51	2020-12-18 14:11:30	<null>	<null>

Tentemos correr o mesmo bloco anónimo outra vez, dando ao procedimento a reserva 51 novamente. Deverá provocar exceção, já que não deve ser possível criar duas faturas para uma reserva.

```

BDDAD_DH2> DECLARE
    Row_Reserva Reserva%ROWTYPE;
BEGIN
    SELECT *
    INTO Row_Reserva
    FROM Reserva
    WHERE Id = 51;
    Prc_Check_Out(Row_Reserva);
END;
[2020-12-18 10:47:47] [72000][20000]
[2020-12-18 10:47:47] ORA-20000: A reserva dada como parâmetro já tem uma fatura associada.
[2020-12-18 10:47:47] ORA-06512: na "BDDAD_DH2.PRC_CHECK_OUT", linha 42
[2020-12-18 10:47:47] ORA-06512: na linha 8
[2020-12-18 10:47:47] Position: 0

```

É o resultado que se esperava.

Testemos também o comportamento do procedimento se for passada por parâmetro uma reserva que não tenha efetuado o checkin:

```

BDDAD_DH2> DECLARE
    Row_Reserva Reserva%ROWTYPE;
BEGIN
    SELECT *
    INTO Row_Reserva
    FROM Reserva
    WHERE Id = 131;--reserva não fez o checkin
    Prc_Check_Out(Row_Reserva);--exceção levantada
END;
[2020-12-18 10:45:01] [72000][20000]
[2020-12-18 10:45:01] ORA-20000: A reserva dada como parâmetro não tem checkin efetuado.
[2020-12-18 10:45:01] ORA-06512: na "BDDAD_DH2.PRC_CHECK_OUT", linha 34
[2020-12-18 10:45:01] ORA-06512: na linha 8
[2020-12-18 10:45:01] Position: 0

```

O procedimento levantou a exceção, e a nova fatura não foi criada:

	Id	Numero	Data	Id_Cliente	Id_Reserva
1	1	1	2020-12-18 10:40:08	41	51

A única fatura que existe continua a ser a da reserva 51.