INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA - IFPB
Unidade Acadêmica de Informação e Comunicação
CST em Sistemas para Internet

Disciplina: Banco de Dados II
Professora: Damires
Aluno: _____ Matrícula: _____

07 - Procedimentos e Funções Armazenados - Exercícios

1. Transforme o bloco anônimo "*atualiza_status_estoque.sql"* (material 11-PL/SQL- INTRO) em um procedimento armazenado que receba o **código do produto** como parâmetro. Execute o procedimento. Verifique o resultado: onde?

2. Você obteve algum "erro" no exercício anterior? Refaça este último procedimento, incluindo o tratamento de exceções *no_data_found* e *too_many_rows*. Caso aconteça uma exceção, **mostre uma mensagem na tela** indicando o que ocorreu.

** Teste com produtos que existam e também com códigos não cadastrados. O que ocorre nesses testes?

3. Verifique se possui a tabela **Aluno** (esquema mostrado a seguir). Caso não a tenha, crie-a. Crie uma **sequência** a ser usada na inserção para geração da matrícula. Em seguida, faça um procedimento que permita a inserção de registros nesta tabela, passando os parâmetros necessários (SEM a matrícula que é a PK). Depois execute o procedimento e insira 3 alunos. Consulte a tabela e veja como ficaram os registros.

Tipo de Objeto **TABLE** Objeto **ALUNO**

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável	Default	Comentário
ALUNO	MATRICULAALU	Number	-	-	-	1	-	-	-
	NOMEALU	Varchar2	30	-	-	-	✓	-	-
	DATA_NASC	Date	7	-	-	-	✓	-	-
	CURSO	Varchar2	25	-	-	-	✓	-	-
	APELIDO	Varchar2	20	-	-	-	✓	-	-
									1 - 5

4. Faça um procedimento que mostre todos os alunos cadastrados. Apresente seus nomes e cursos. Use um cursor. Verifique sua execução.

5. Faça um procedimento que mostre todos os artistas que estão associados a uma determinada categoria de filme (por exemplo, '**Ação**'). Passe a categoria como parâmetro e construa o **cursor** com base nela. Apresente o nome e o país de cada artista. Verifique sua execução. Teste, em seguida, com a categoria 'Aventura'.

6. Analise e execute a função seguinte. O que ela faz?

```
create or replace function getDeptoSalario(dno number) return number is
  sal_comp number;
Begin
    Sal_comp := 0;
    for func in (select salario from empregado where coddepto = dno and salario is not
null)    loop
        sal_comp := sal_comp + func.salario;
    end loop;
    return sal_comp;
end getDeptoSalario;
```

6.1 Crie outra versão desta function, usando a função **SUM** pre-definida. Mostre sua execução.

7. Crie uma função que, passado o código do artista, retorne sua idade (e.g., use: round((months_between(sysdate,data_nasc)/12)) ou EXTRACT(YEAR FROM sysdate) - EXTRACT(YEAR FROM data)).

8. Crie outra função que calcule a **média** de idade de todos os artistas cadastrados.

9. Analise e execute o procedimento seguinte. O que ele faz? Como são definidos os parâmetros? Qual a exceção definida?

```
Create or Replace PROCEDURE calcula_bonus (emp_id IN INTEGER, bonus OUT
   REAL) IS
   hiredate  DATE;
   bonus_exc EXCEPTION;
BEGIN
   SELECT salary * 0.10, hire_date INTO bonus, hiredate FROM empregado2
      WHERE employee_id = emp_id;
   IF bonus IS NULL THEN
      RAISE bonus_exc;
   END IF;
   IF MONTHS_BETWEEN(SYSDATE, hiredate) > 30 THEN
      bonus := bonus + 500;
   END IF;
   EXCEPTION
      WHEN bonus_exc THEN dbms_output.put_line('bonus inexistente');
END calcula_bonus;
```

Execute-o usando o bloco seguinte:

```
declare bonus real;
begin
  calcula_bonus (100, bonus);
  dbms_output.Put_line('Bonus ' || bonus);
end;
```