

## Lista de Exercícios – nº 2

### PL/SQL

Não esquecer de ligar: **Set serveroutput on**

- Transparências das páginas 18 até 28:

#### **8. ACESSAR NA TABELA FORNECEDOR O ENDEREÇO E AS VENDAS DO FORNECEDOR INFORMADO PELO USUÁRIO.**

*Criar um bloco PL-SQL para acessar um determinado fornecedor na tabela FORNECEDOR, cujo código do fornecedor deverá ser informado em tempo de execução. Apresente na tela o seu código, nome, endereço e o valor de suas vendas.*

#### **9. Acessar a tabela “CONTA” pesquisando registro e em função do resultado da pesquisa, inserir nova linha na tabela A\_FAZER.**

Criar um bloco PL-SQL para recuperar, na tabela CONTA, o numero da conta com saldo inferior a R\$ 10.000 em uma variável e caso encontre um registro, inserir na tabela A\_FAZER uma nova tarefa nos campos abaixo:

FEITO = 'N'

DATA = data do sistema

OBSERV = 'Mensagem para CC nº xxxx – saldo baixo'

Para evitar problemas com o SELECT retornando mais de uma linha ou não encontrando registros, acrescente na programação o controle de erros, utilizando para isso, a seção EXCEPTION.

#### **10. ACESSAR TABELA FORNECEDOR, CUJO CÓDIGO DEVERÁ SER INFORMADO NA TELA E ALTERAR O VALOR DA COMISSÃO, NAS CONDIÇÕES ABAIXO.**

*Criar um bloco PL-SQL para acessar, na tabela FORNECEDOR, o fornecedor informado na tela e executar as seguintes ações:*

- *Caso não encontre dar a mensagem 'FORNECEDOR NÃO INFORMADO'*
- *Caso encontre deverá fazer a seguinte atualização na comissão:*
  - *Se as vendas forem maiores que 6000 → a comissão deverá ser 15% do valor das vendas*
  - *Caso contrário → a comissão deverá ser 5% do valor das vendas*
- *No final, atualizar o valor da comissão do fornecedor e dar uma mensagem do valor calculado da comissão.*

**11. Criar um bloco PL/SQL para atualizar um campo numa tabela de acordo com o resultado de uma pesquisa feita no registro dessa tabela.**

Antes de criar o bloco para atualização dos dados, execute os comandos abaixo para criação e inserção de dados na tabela ALUNO.

- **Criar uma tabela conforme segue:**

```
CREATE TABLE ALUNO_TI  
( RA NUMBER(9) PRIMARY KEY, NOME VARCHAR(50) NOT NULL,  
  DISCIPLINA VARCHAR(30), MEDIA NUMBER(3,1), CARGA_HORA NUMBER(2),  
  FALTAS NUMBER(2), RESULTADO VARCHAR(10)  
);
```

- **Inserir registros na tabela ALUNO deixando a coluna RESULTADO em branco.**

```
INSERT INTO ALUNO_TI VALUES (100, 'Luis', 'Modelagem Dados',7.5,80,14,null);  
INSERT INTO ALUNO_TI VALUES (200, 'Carla', 'Modelagem Dados',4.5,80,16,null);  
INSERT INTO ALUNO_TI VALUES (300, 'Pedro', 'Modelagem Dados',6.0,80,24,null);
```

Criar um bloco PL/SQL para pesquisar um nº de RA fornecido em tempo de execução e, encontrando o registro, atualizar o campo RESULTADO, de acordo com as regras abaixo:

- ✓ Se o aluno obteve **média igual ou maior que 7.0** e suas **faltas não ultrapassarem 25% da carga horária** da disciplina o resultado será: **APROVADO**.
- ✓ Se o aluno obteve **média inferior a 7.0** e suas **faltas não ultrapassarem 25% da carga horária** da disciplina o resultado será: **EXAME**.
- ✓ Para os **demais casos** o resultado será: **REPROVADO**.