



Prática 7 – Procedimentos e funções

Entrega: 14/10 – 23:59h – turma de 5ª. feira

15/10 – 23:59h – turma de 6ª. feira

Em caso de falta na aula da prática: entrega individual na data da turma

OBS. IMPORTANTES PARA TODOS OS RELATÓRIOS (LEIAM!!!!):

- Deve ser entregue um relatório, em formato DOC (somente doc!) com a seguinte estrutura principal:
 - Identificação da dupla: nome e nro USP
 - Nro da Prática
 - Exercício X:

Explicações, justificativas e comentários sobre a resolução do exercício.

Código SQL a ser compilado/executado (incluir **todos** os comandos executados). Na correção, será feito Copy/Paste do código para compilação. **Se o código não compilar, o exercício recebe nota zero até que o grupo, na revisão da correção, identifique o erro de compilação e indique como corrigi-lo.**

Saída: resultados das consultas ou de procedimentos, funções, triggers, etc...

OBS: o quadro para separar as partes da resolução do exercício é opcional.

- A prática será entregue via sistema Tidia. **Apenas um** membro da dupla deve fazer o *upload* do arquivo.
 - O arquivo deve ser identificado da seguinte maneira:
PraticaX_Nome1Sobrenome1_Nome2Sobrenome2, onde **Nome** e **Sobrenome** são respectivamente Primeiro e Último nomes dos membros da dupla (ex:
Pratica2_ElaineSousa_FulanoDeTal.doc).
- Será considerado o português do relatório! Então, cuidado na hora de escrever: seja coerente, claro, organize bem as idéias e escreva corretamente.

1. (3.0) No esquema do sistema de E-Learning, os principais atores são definidos em uma generalização: um vínculo usp pode ser um aluno, um funcionário ou um professor, cada qual com seus atributos.

a) Escreva um procedimento para a inserção dos dados de um vínculo USP.

```
CREATE OR REPLACE PROCEDURE insere_vinculo_usp(  
    p_nrousp LBD01_VINCULO_USP.NROUSP%TYPE,  
    p_tipovinc LBD01_VINCULO_USP.TIPOVINC%TYPE,  
    p_nome LBD01_VINCULO_USP.NOME%TYPE,  
  
    p_dataingresso LBD01_VINCULO_USP.DATAINGRESSO%TYPE DEFAULT SYSDATE,  
    p_datanascimento LBD01_VINCULO_USP.DATANASCIMENTO%TYPE DEFAULT NULL,  
    p_ativo LBD01_VINCULO_USP.ATIVO%TYPE DEFAULT 'y'  
)
```

b) Escreva um procedimento para a inserção dos dados de professor, e que usa o procedimento do item a).

```
CREATE OR REPLACE PROCEDURE insere_professor(  
    p_nrousp LBD01_VINCULO_USP.NROUSP%TYPE,  
    p_tipovinc LBD01_VINCULO_USP.TIPOVINC%TYPE,  
    p_nome LBD01_VINCULO_USP.NOME%TYPE,  
    p_predio LBD02_PROFESSOR.PREDIO%TYPE,  
    p_sala LBD02_PROFESSOR.SALA%TYPE,  
  
    p_dataingresso LBD01_VINCULO_USP.DATAINGRESSO%TYPE DEFAULT SYSDATE,  
    p_datanascimento LBD01_VINCULO_USP.DATANASCIMENTO%TYPE DEFAULT NULL,  
    p_ativo LBD01_VINCULO_USP.ATIVO%TYPE DEFAULT 'y',  
    p_titulacao LBD02_PROFESSOR.TITULACAO%TYPE DEFAULT 'Doutorado'  
)
```

Obs.:

- As exceções não devem ser tratadas definitivamente dentro dos procedimentos. Em ambos os procedimentos deve-se interceptar as possíveis exceções, re-editá-las e jogá-las novamente com um texto que explica o problema ocorrido;
- Deve ser tratado o caso em que é possível inserir um vínculo USP, mas não um professor para ele. Nesse caso, o banco de dados deve retornar ao seu estado anterior à tentativa de inserção de professor.

2. (3.0) No Sistema de E-Learning as entidades turma, grupo, trabalho e mensagem possuem a chave composta por sequencias que não são globais, mas determinadas em função de quais dados estão sendo inseridos. Escreva uma **única** função que consulta qualquer uma das tabelas destas entidades e retorna o próximo número da sequencia. Esta função deve receber o nome da tabela, e o nome e o valor de cada um dos atributos necessários para a consulta.

Observe que o número de parâmetros necessários não é igual para todas as tabelas, e que os atributos das tabelas têm nomes e tipos diferentes. Para resolver estes problemas use parâmetros do tipo varchar2 e também inicialização de parâmetros com o cláusula DEFAULT.

Escreva também um bloco anônimo que execute esta função para cada uma das tabelas – usando os dados necessários – e imprime o resultado. Como no exemplo:

```
DECLARE
BEGIN
    dbms_output.put_line(proximo_valor_geral('LBD07_TURMA', 'ANO', '2008', 'CODDISC', 'SCC0261'));
    dbms_output.put_line(proximo_valor_geral('LBD11_GRUPO', 'CODCURSO', '8'));
EXCEPTION
    WHEN OTHERS
        THEN dbms_output.put_line('Erro nro: ' || SQLCODE || '. Mensagem: ' || SQLERRM );
END;
```

3. (4.0) No Modelo Entidade-Relacionamento do Sistema de E-Learning há um ciclo determinado pelas entidades Professor, Disciplina e Turma. Este ciclo requer um cuidado especial durante operações de escrita e atualização, pois do contrário é possível atribuir um professor a uma turma de uma disciplina sem que este professor esteja cadastrado para isso (relacionamento leciona).
- a) Escreva um procedimento que recebe parâmetros de professor, e de disciplina e que joga (raise_application_error) a seguinte exceção:
- O professor x não está habilitado a ministrar a disciplina y.
- b) Escreva um procedimento que insere uma nova turma.
- c) Escreva um procedimento que atualiza o professor de uma turma já existente. Este procedimento deve jogar (raise_application_error) o seguinte erro:
- Não é possível atualizar a turma, pois ela não existe.

Ambos os procedimentos devem garantir que o professor que ministrará as aulas está cadastrado como professor que leciona para a respectiva disciplina. Para isso, use o procedimento criado no item a).

- d) Escreva um procedimento que recebe parâmetros de professor e disciplina, e que remove um cadastro professor-disciplina da tabela leciona apenas se não houver nenhuma turma associada aquele professor para aquela disciplina. Caso não seja possível fazer a remoção o procedimento deve jogar (raise_application_error) o seguinte erro:
- Não é possível remover o cadastro do professor x para a disciplina y, pois há turmas desta disciplina atribuídas a ele.

Escreva um bloco anônimo que ilustra o funcionamento dos procedimentos a), b) e c) assim como no exemplo:

```
DECLARE
  ano NUMBER := 2009;
  disc VARCHAR2(7) := 'SCC0261';
  prof NUMBER := 31;
  nroturma NUMBER;
BEGIN
  nroturma := proximo_valor_geral('LBD07_TURMA','ANO',ano,'CODDISC',disc);
  safe_insert_turma(nroturma,ano,disc,prof);

  prof := 40;
  safe_update_turma(nroturma,ano,disc,prof);

  prof := 40;
  safe_update_turma(nroturma,ano,disc,prof);

  disc := 'SCC0261';
  prof := 40;
  safe_delete_leciona(prof, disc);

EXCEPTION
  WHEN OTHERS
  THEN dbms_output.put_line('Erro nro: ' || SQLCODE || '. Mensagem: ' || SQLERRM);
END;
```

- e) Suponha um usuário denominado *Cliente* que inicialmente possui permissões de escrita, atualização e remoção nas tabelas turma e leciona. Usando permissões de acesso, como você poderia usar os procedimentos dos itens b), c) e d) para garantir que o usuário *Cliente* não comprometa a integridade dos dados do banco? Liste os comandos necessários.