

**Departamento de Ciências de Computação – SCC  
Instituto de Ciências Matemáticas e de Computação – ICMC  
Universidade de São Paulo – USP**

**Laboratório de Bases de Dados  
Docente Responsável: Profa. Dra. Cristina Dutra de Aguiar Ciferri**

## **QUARTO TRABALHO PRÁTICO**

### PREPARAÇÃO PARA O TRABALHO PRÁTICO 4

1. Para o trabalho prático 4, os alunos devem corrigir os erros do trabalho prático 1, tanto os erros do *scriptCria* quanto os erros do *scriptInsere*. Esses dois *scripts* corrigidos devem ser entregues novamente juntamente com o trabalho prático 4, em um único *script* chamado *scriptCriaInsere*.

#### Observações:

- Todas as restrições definidas no trabalho prático 1 também devem ser cumpridas nas correções do *scriptCria* e do *scriptInsere*.
- Antes dos comandos de criação das tabelas, devem ser especificados comandos de remoção das tabelas e de remoção de sequências.
- Antes dos comandos de criação das tabelas, devem ser especificados o comando para o funcionamento correto de sequências e o comando que possibilita o uso de caracteres especiais, como aqueles caracteres acentuados.
- Novas tuplas podem ser inseridas nas tabelas para o melhor funcionamento das consultas. No entanto, isso não acrescentará pontos na nota do trabalho prático 1.

### ESPECIFICAÇÃO DO TRABALHO PRÁTICO 4

2. Preparar um *scriptSubprograma* contendo problemas de PL-SQL com procedimentos e funções. Os problemas devem ser respondidos utilizando não somente os comandos PL/SQL exigidos nos exercícios práticos de sala de aula, mas também outros comandos PL/SQL vistos na aula teórica. Deve ser proposto 1 problema da seguinte forma:

- Criar um procedimento que receba parâmetros de entrada e que atue como “gerador de relatórios”, ou seja, que realize uma listagem com base nos parâmetros passados. O procedimento deve obrigatoriamente fazer chamada para duas funções criadas especificamente para este exercício. Cada consulta contida no procedimento deve utilizar duas ou mais tabelas do banco de dados e deve conter uma ou mais funções de agrupamento. O procedimento deve realizar o tratamento das devidas exceções. Na linha de comando, crie 3 diferentes chamadas para o procedimento.
3. Preparar um *scriptGatilho* contendo problemas de PL-SQL com gatilhos. Os problemas devem ser respondidos utilizando não somente os comandos PL/SQL exigidos nos exercícios práticos de sala de aula, mas também outros comandos PL/SQL vistos na aula teórica. Devem ser propostos 2 problemas da seguinte forma:
- Criar um gatilho que tem como objetivo armazenar contadores relacionados à aplicação. Por exemplo, considerando a aplicação “Campeonatos de Futebol” utilizada nos exercícios desenvolvidos em sala de aula, este exercício poderia armazenar, para cada jogador, o número de partidas jogadas, o número de gols pró marcados, o número de gols contra marcados, o número de cartões amarelos recebidos, o número de cartões vermelhos recebidos, o número de faltas sofridas. Na linha de comando, crie diferentes chamadas que disparem o gatilho e mostre como as ações disparadas pelo gatilho podem ser verificadas.
    - i. Deve ser indicado, passo a passo, quando os gatilhos devem ser criados e como eles devem ser disparados. Por exemplo, é possível indicar a seguinte sequência de passos: 1- criação de tabelas; 2- criação dos gatilhos; 3- inserção de dados; 4- diferentes chamadas que disparem o gatilho. Outra sequência de passos é: 1- criação de tabelas; 2- inserção de dados; 3- execução de procedimento; 4- diferentes chamadas que disparem o gatilho.
  - Criar um gatilho que sofra do problema de tabela mutante. Resolva o problema do gatilho, eliminando o problema de tabela mutante (deve ser usada a solução com gatilhos compostos - compound triggers). Explique a solução apresentada e também defina o que é tabela mutante.
    - i. Criar um gatilho que sofra do problema de tabela mutante.
    - ii. Explicar porque houve o problema de tabela mutante.
    - iii. Resolver o problema de tabela mutante.

iv. Explicar porque o problema de tabela mutante foi resolvido.

Os scripts *scriptSubprograma* e *scriptGatilho* devem descrever cada problema da seguinte forma:

- Descrição textual do problema. Colocar a descrição textual entre comentários.
- Programa PL/SQL documentado.
- Listagem dos resultados obtidos. Colocar a listagem dos resultados obtidos entre comentários.

Observações:

- Os problemas PL-SQL devem realizar o tratamento adequado das devidas exceções.
- Os scripts devem ser executados na íntegra (ou seja, devem ser executados como um *script*)
- Devem ser utilizadas cláusulas de formatação das respostas para os usuários, por exemplo, que concatenem strings, que renomeiem os nomes das colunas, que representem valores monetários de forma legível (e.g., R\$10.000,00).
- As colunas exibidas devem ser formatadas apropriadamente, justificando-se os resultados à direita ou à esquerda quando necessário.
- Além de realizar o tratamento adequado das exceções e formatar adequadamente as respostas exibidas, os problemas PL-SQL devem ser criativos. Por criatividade entende-se, por exemplo, o uso de diferentes comandos PL/SQL, a definição de consultas complexas usando várias tabelas do banco de dados e a importância (significado) da consulta para a aplicação de banco de dados.
- A organização do projeto é fundamental para a correção do mesmo, e portanto será levada em consideração.

## OBSERVAÇÕES

### Entrega do Trabalho Prático

- Data de entrega: até as 23h59m da data especificada na página da disciplina. Será descontado 1,0 ponto por dia de atraso.
- Forma de entrega. O grupo deve enviar os *scripts* por e-mail. Todos os *scripts* devem ser enviados no mesmo e-mail.
- Recebimento dos e-mails. Os e-mails devem ser identificados com o assunto “[LabBD Trabalho4 DD-MM-AA] número do grupo”. A ausência deste assunto implicará no descarte do material, sem qualquer análise do trabalho prático. A validação do e-mail será feita por uma resposta ‘recebido’ enviada para o grupo. Essa validação indicará que o trabalho foi recebido e que o arquivo não está corrompido. O trabalho não será analisado em dois casos: (i) quando ele não for entregue; e (ii) quando o arquivo estiver corrompido e não for enviado um novo arquivo no prazo. É responsabilidade do grupo verificar o recebimento do e-mail de validação. Além disto, o horário e a data de entrega do projeto são os que constam do e-mail que contém a versão correta.
- Identificação dos *scripts*. Cada *script* deve possuir um cabeçalho contendo as seguintes informações: número do grupo, nome e número USP de cada aluno do grupo, nome da disciplina, turma, “Trabalho Prático 4” e tipo do *script* (i.e., criação, inserção, consulta). A ausência deste cabeçalho implicará no descarte do material, sem qualquer análise do trabalho prático.