

## Oracle Database 10g: Programação com PL/SQL NOVO

Duração: 5 Dias

### Objetivos do Curso

Este curso apresenta aos alunos os fundamentos de PL/SQL e as vantagens dessa avançada linguagem de programação. Os alunos aprenderão a criar blocos PL/SQL no código da aplicação, os quais podem ser compartilhados por vários forms, relatórios e aplicações de gerenciamento de dados. Eles aprenderão também a criar blocos PL/SQL anônimos e conhecerão funções e procedures armazenados. Além disso, verão como declarar variáveis, interceptar exceções e declarar e controlar cursores. Será ensinado como desenvolver, executar e gerenciar unidades de programa PL/SQL armazenadas, como procedures, funções, packages e triggers de banco de dados. Os alunos aprenderão a gerenciar dependências de objetos e recompilação de objetos inválidos. Este curso também descreve as características e os métodos de manipulação de LOBs (Large Objects), além das formas de utilização de alguns dos packages fornecidos pelo Oracle.

Learn To:

- Criar seção executável e estruturas de controle
- Criar e gerenciar procedures, funções, packages e triggers
- Trabalhar com tipos de dados compostos e cursores
- Utilizar os packages fornecidos pelo Oracle no desenvolvimento de aplicações
- Gerenciar dependências e LOBs (Large Objects)

### Público

Administrador de Banco de Dados  
Consultor Técnico  
Database Administrators  
Database Designers  
Desenvolvedor em PL/SQL  
Forms Developer  
PL/SQL Developer  
Technical Consultant

### Pré-requisitos

*Pré-requisitos Recomendados*

Oracle Database 10g: Introduction to SQL

### Objetivos do Curso

- Criar códigos PL/SQL para estabelecer interface com o banco de dados
- Desenvolver unidades de programa PL/SQL executadas de forma eficiente
- Usar estruturas de programação PL/SQL e instruções de controle condicional
- Tratar erros de runtime
- Descrever funções e procedures armazenados
- Criar instruções SQL dinâmicas a fim de obter maior flexibilidade de codificação
- Desenvolver código PL/SQL para tipos de dados predefinidos, subprogramas locais, programas adicionais, exceções e c
- Usar a infra-estrutura de advertências do compilador

- Manipular LOBs (Large Objects)
- Criar triggers para resolver desafios de negócios
- Gerenciar dependências entre subprogramas PL/SQL
- Programar jobs PL/SQL para serem executados de forma independente
- Criar funções e procederes armazenados
- Desenvolver packages PL/SQL para agrupar e conter estruturas relacionadas
- Criar subprogramas para packages com overloading a fim de obter maior flexibilidade
- Categorizar os packages PL/SQL fornecidos pelo Oracle

## **Tópicos do Curso**

### **Introdução ao Código PL/SQL**

- O que é PL/SQL
- Ambiente PL/SQL
- Vantagens do código PL/SQL
- Visão geral dos tipos de blocos PL/SQL
- Criando e executando um bloco anônimo simples
- Gerando saída a partir de um bloco PL/SQL
- iSQL\*Plus como ambiente de programação PL/SQL

### **Declarando Identificadores PL/SQL**

- Fazendo a distinção entre os diferentes tipos de identificadores em um subprograma PL/SQL
- Usando a seção declarativa para definir identificadores
- Listando os usos de variáveis
- Armazenando dados em variáveis
- Declarando variáveis PL/SQL

### **Criando Instruções Executáveis**

- Descrevendo diretrizes básicas de sintaxe de bloco
- Usando literais em códigos PL/SQL
- Personalizando designações de identificador com funções SQL
- Usando blocos aninhados como instruções
- Fazendo referência a um valor de identificador em um bloco aninhado
- Qualificando um identificador com um label
- Usando operadores em códigos PL/SQL
- Usando as diretrizes e a sintaxe de bloco PL/SQL adequadas

### **Interagindo com o Oracle Server**

- Identificando as instruções SQL que podem ser usadas em códigos PL/SQL
- Incluindo instruções SELECT em códigos PL/SQL
- Recuperando dados em códigos PL/SQL com a instrução SELECT
- Usando convenções de nomeação para evitar erros ao criar instruções DML e realizar operações de recuperação
- Manipulando dados no servidor com códigos PL/SQL
- O conceito de cursor SQL
- Usando atributos do cursor SQL para obter feedback de códigos DML
- Salvando e descartando transações

### **Criando Estruturas de Controle**

- Controlando o fluxo de execução de códigos PL/SQL
- Processamento condicional com instruções IF
- Processamento condicional com instruções CASE

Tratando valores nulos para evitar erros comuns  
Criando condições booleanas com operadores lógicos  
Usando controle iterativo com instruções em loop

### **Trabalhando com Tipos de Dados Compostos**

Conhecendo os tipos de dados compostos de tabelas e registros PL/SQL  
Usando registros PL/SQL para armazenar valores de diferentes tipos  
Inserindo e atualizando com registros PL/SQL  
Usando tabelas INDEX BY para armazenar valores do mesmo tipo de dados

### **Usando Cursores Explícitos**

Loops FOR de cursor usando subconsultas  
Aumentando a flexibilidade de cursores com parâmetros  
Usando a cláusula FOR UPDATE para bloquear linhas  
Usando a cláusula WHERE CURRENT para fazer referência à linha atual  
Usando cursores explícitos para processar linhas  
Atributos de cursores explícitos  
Cursores e registros

### **Tratando Exceções**

Tratando exceções com PL/SQL  
Exceções predefinidas  
Interceptando erros não predefinidos do Oracle Server  
Funções que retornam informações sobre as exceções encontradas  
Detectando exceções definidas pelo usuário  
Propagando exceções  
Usando o procedure RAISE\_APPLICATION\_ERROR para reportar erros em aplicações

### **Criando Procedures Armazenados**

Descrevendo a estrutura de blocos usada em procedures PL/SQL armazenados  
Chamando funções/procedures armazenados a partir de diferentes ferramentas  
Chamando um procedure armazenado com variáveis de host a partir de Java, C, iSQL\*Plus, Forms etc.  
Chamando um procedure armazenado a partir de um bloco anônimo ou de outro procedure armazenado  
Listando a sintaxe CREATE OR REPLACE PROCEDURE  
Identificando as etapas de desenvolvimento para a criação de um procedure armazenado  
Usando o comando SHOW ERRORS  
Verificando o código-fonte na view de dicionário USER\_SOURCE

### **Criando Funções Armazenadas**

Descrevendo funções armazenadas  
Listando a sintaxe CREATE OR REPLACE FUNCTION  
Identificando as etapas para criar uma função armazenada  
Executando uma função armazenada  
Identificando as vantagens de utilizar funções armazenadas em instruções SQL  
Identificando as restrições para chamar funções a partir de instruções SQL  
Removendo uma função

### **Criando Packages**

Listando as vantagens dos packages  
Descrevendo packages  
Mostrando os componentes de um package. Mostrando as estruturas que formam um package  
Desenvolvendo um package

- Criando a especificação do package
- Declarando estruturas públicas
- Criando o package body

### **Usando Mais Conceitos de Package**

- Listando as vantagens do recurso de overloading
- Mostrando um exemplo de overloading
- Usando declarações Forward em packages
- Criando um procedure que será usado uma única vez (inicialização do código do package)
- Listando as restrições para funções de packages usadas em códigos SQL
- Encapsulando código em uma demonstração de package
- Chamando funções de packages definidas pelo usuário em uma instrução SQL
- Utilizando o estado persistente de variáveis de package

### **Utilizando os Packages Fornecidos pelo Oracle no Desenvolvimento de Aplicações**

- Listando os diversos usos dos packages fornecidos pelo Oracle
- Reutilizando o código incluído no package para concluir diversas tarefas do desenvolvedor e do DBA.
- Usando o comando DESCRIBE para verificar especificações de package e ocorrências de overloading
- Explicando como funciona o package DBMS\_OUTPUT (em conjunto com SET SERVEROUTPUT ON)
- Interagindo com arquivos do sistema operacional usando o package UTL\_MAIL
- Descrevendo o processamento de arquivos com o package UTL\_FILE
- Verificando exceções e rotinas do package UTL\_FILE
- Usando o package UTL\_FILE para gerar um relatório em um arquivo

### **Código SQL Dinâmico e Metadados**

- Descrevendo o uso do código SQL dinâmico nativo
- Descrevendo o fluxo de execução de instruções SQL
- Mostrando a sintaxe da instrução EXECUTE IMMEDIATE para código SQL dinâmico nativo
- Criando um procedure para gerar código SQL dinâmico nativo usando a instrução EXECUTE IMMEDIATE para deletar I
- Descrevendo o package DBMS\_SQL
- Fornecendo um exemplo de DBMS\_SQL
- Listando as vantagens da utilização do código SQL dinâmico nativo em vez do package DBMS\_SQL

### **Considerações de Projeto para Códigos PL/SQL**

- Padronizando constantes com um package de constantes
- Padronizando exceções com um package de tratamento de exceções
- Apresentando subprogramas locais
- Usando subprogramas locais
- Rastreamento de erros de runtime com um package de exceções
- Descrevendo a hint de compilador NOCOPY
- Usando a hint de compilador NOCOPY
- Explicando os efeitos da hint NOCOPY

### **Gerenciando Dependências**

- Definindo objetos dependentes e referenciados
- Diagramando dependências com código, views, procedures e tabelas
- Gerenciando dependências locais entre um procedure, uma view e uma tabela
- Analisando um cenário de dependências locais
- Exibindo dependências diretas com a view USER\_DEPENDENCIES
- Executando o script UTL\_DTREE.SQL para criar objetos que permitem verificar dependências diretas e indiretas
- Prevenção dos efeitos de alterações em objetos dependentes

## **Manipulando LOBs (Large Objects)**

- Descrevendo um objeto LOB
- Diagramando os componentes de um LOB
- Gerenciando e listando as características de LOBs internos
- Descrevendo, gerenciando e protegendo BFILES
- Criando e usando o objeto DIRECTORY para acessar e usar BFILES
- Preparando BFILES para uso
- Usando a função BFILENAME para carregar BFILES
- Descrevendo o package DBMS\_LOB

## **Criando Triggers**

- Descrevendo os diversos tipos de triggers e as formas de executá-los
- Listando as vantagens dos triggers de banco de dados e as diretrizes para sua utilização
- Utilizando um exemplo de trigger de banco de dados básico para mostrar como os triggers são executados
- Criando triggers DML, mostrando sua sintaxe e listando seus componentes
- Explicando a seqüência de acionamento de triggers
- Criando uma instrução DML e triggers em nível de linha
- Usando os qualificadores OLD e NEW para fazer referência a valores de colunas
- Usando predicados condicionais com triggers

## **Aplicações de Triggers**

- Criando triggers para os seguintes eventos DDL: CREATE, ALTER e DROP
- Criando triggers para os seguintes eventos de sistema: SERVERERROR, STARTUP, SHUTDOWN, LOGON e LOGOFF
- Definindo uma tabela mutante
- Descrevendo cenários de aplicações de negócios para implementar triggers
- Descrevendo os privilégios necessários para gerenciar triggers

## **Compreendendo e Influenciando o Compilador PL/SQL**

- Listando as características da compilação nativa
- Descrevendo as características do compilador PL/SQL no Oracle Database 10g
- Identificando os três parâmetros usados para influenciar a compilação (PLSQL\_CODE\_TYPE, PLSQL\_DEBUG, PLSQL\_
- Mostrando como definir os parâmetros
- Descrevendo a view de dicionário usada para ver a compilação do código (USER\_PLSQL\_OBJECTS)
- Alterando as definições de parâmetros, recompilando o código e verificando os resultados
- Descrevendo a infra-estrutura de advertências do compilador no Oracle Database 10g
- Listando as etapas a serem seguidas para definir os níveis de advertência do compilador