

## Oracle Database 10g: Program with PL/SQL

**Duração:** 5 Dias

### Objetivos do Curso

This course introduces students to PL/SQL and helps them understand the benefits of this powerful programming language. In the class, students learn to create PL/SQL blocks of application code that can be shared by multiple forms, reports, and data management applications. Students learn to create anonymous PL/SQL blocks and are introduced to stored procedures and functions. They learn about declaring variables, trapping exceptions and they also learn to declare and control cursors. In class students learn to develop, execute, and manage PL\SQL stored program units like procedures, functions, packages, and database triggers. Student also learns to manage object dependencies and recompilation of invalid objects. This course also describes the characteristics and ways of manipulation of large objects. Students are introduced to the utilization of some of the Oracle-supplied packages. Learn To: Create Executable Section and write Control Structures Create and manage Procedures, Functions, Packages, and Triggers Work with Composite Data Types and cursors Utilize Oracle-Supplied Packages in Application Development Include Exception Handling Manage Dependencies and 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

### Objetivos do Curso

Schedule PL/SQL jobs to run independently  
Manage dependencies between PL/SQL subprograms  
Create stored procedures and functions  
Design PL/SQL packages to group and contain related constructs  
Create overloaded package subprograms for more flexibility  
Categorize the Oracle supplied PL/SQL packages  
Design PL/SQL program units that execute efficiently  
Use PL/SQL programming constructs and conditional control statements  
Handle run-time errors  
Describe stored procedures and functions  
Write dynamic SQL for more coding flexibility  
Design PL/SQL code for predefined data types, local subprograms, additional programs and standardized constants and  
Use the compiler warnings infrastructure  
Write PL/SQL code to interface with the database  
Manipulate large objects  
Create triggers to solve business challenges

### Tópicos do Curso

## **Introduction to PL/SQL**

What is PL/SQL

PL/SQL Environment

Benefits of PL/SQL

Overview of the Types of PL/SQL blocks

Create and Execute a Simple Anonymous Block

Generate Output from a PL/SQL Block

SQL Developer as PL/SQL Programming Environment

## **Declaring PL/SQL Identifiers**

Identify the Different Types of Identifiers in a PL/SQL subprogram

Use the Declarative Section to Define Identifiers

List the Uses for Variables

Store Data in Variables

Declare PL/SQL Variables

## **Writing Executable Statements**

Describe Basic Block Syntax Guidelines

Use Literals in PL/SQL

Customize Identifier Assignments with SQL Functions

Use Nested Blocks as Statements

Reference an Identifier Value in a Nested Block

Qualify an Identifier with a Label

Use Operators in PL/SQL

Use Proper PL/SQL Block Syntax and Guidelines

## **Interacting with the Oracle Server**

Identify the SQL Statements You Can Use in PL/SQL

Include SELECT Statements in PL/SQL

Retrieve Data in PL/SQL with the SELECT Statement

Avoid Errors by Using Naming Conventions When Using Retrieval and DML Statements

Manipulate Data in the Server Using PL/SQL

The SQL Cursor concept

Use SQL Cursor Attributes to Obtain Feedback on DML

Save and Discard Transactions

## **Writing Control Structures**

Control PL/SQL Flow of Execution

Conditional processing Using IF Statements

Conditional Processing CASE Statements

Handle Nulls to Avoid Common Mistakes

Build Boolean Conditions with Logical Operators

Use Iterative Control with Looping Statements

## **Working with Composite Data Types**

Learn the Composite Data Types of PL/SQL Records and Tables

Use PL/SQL Records to Hold Multiple Values of Different Types

Inserting and Updating with PL/SQL Records

Use INDEX BY Tables to Hold Multiple Values of the Same Data Type

## **Using Explicit Cursors**

Cursor FOR Loops Using Sub-queries

- Increase the Flexibility of Cursors By Using Parameters
- Use the FOR UPDATE Clause to Lock Rows
- Use the WHERE CURRENT Clause to Reference the Current Row
- Use Explicit Cursors to Process Rows
- Explicit Cursor Attributes
- Cursors and Records

### **Handling Exceptions**

- Handling Exceptions with PL/SQL
- Predefined Exceptions
- Trapping Non-predefined Oracle Server Errors
- Functions that Return Information on Encountered Exceptions
- Trapping User-Defined Exceptions
- Propagate Exceptions
- Use The RAISE\_APPLICATION\_ERROR Procedure To Report Errors To Applications

### **Creating Stored Procedures**

- Describe the block structure for PL/SQL stored procedures
- Invoke a stored procedure/function from different tools
- Call a stored procedure with host variables from SQL\*Plus, Forms, Java, C, etc
- Invoke a stored procedure from an anonymous block or another stored procedure
- List the CREATE OR REPLACE PROCEDURE syntax
- Identify the development steps for creating a stored procedure
- Use the SHOW ERRORS command
- View source code in the USER\_SOURCE dictionary view

### **Creating Stored Functions**

- Describe stored functions
- List the CREATE OR REPLACE FUNCTION syntax
- Identify the steps to create a stored function
- Execute a stored function
- Identify the advantages of using stored functions in SQL statements
- Identify the restrictions of calling functions from SQL statements
- Remove a function

### **Creating Packages**

- List the advantages of packages
- Describe packages
- Show the components of a package Diagram the visibility of constructs within a package
- Develop a package
- Create the package specification
- Declare public constructs
- Create the package body

### **Using More Package Concepts**

- List the benefits of overloading
- Show overloading example
- Use forward declarations in packages
- Create a one-time only procedure (package code initialization)
- List the restrictions on package functions used in SQL
- Encapsulate code in a package demonstration
- Invoke a user-defined package function from a SQL statement

Utilize the persistent state of package variables

### **Utilizing Oracle Supplied Packages in Application Development**

List the various uses for the Oracle supplied packages

Reuse pre-packaged code to complete various tasks from developer to DBA purposes

Use the DESCRIBE command to view the package specifications and overloading

Explain how DBMS\_OUTPUT works (in conjunction with SET SERVEROUTPUT ON)

Interact with operating system files with UTL\_MAIL

Describe file processing with UTL\_FILE

Review UTL\_FILE routines and exceptions

Use UTL\_FILE to generate a report to a file

### **Dynamic SQL and Metadata**

Describe using native dynamic SQL

List the execution flow of SQL

Show the syntax for the EXECUTE IMMEDIATE statement for native dynamic SQL

Create a procedure to generate native dynamic SQL using EXECUTE IMMEDIATE to delete rows from a table

Describe the DBMS\_SQL package

Provide an example of DBMS\_SQL

List the advantages of using Native Dynamic SQL Over the DBMS\_SQL package

### **Design Considerations for PL/SQL Code**

Standardize constants with a constant package

Standardize exceptions with an exception handling package

Introduce local sub-programs

Use local sub-programs

Track run time errors with an exception package

Describe the NOCOPY compiler hint

Use the NOCOPY compiler hint

Explain the effects of NOCOPY

### **Managing Dependencies**

Define dependent and referenced objects

Diagram dependencies with code, views, procedures, and tables

Manage local dependencies between a procedure, view, and a table

Analyze a scenario of local dependencies

Display direct dependencies using the USER\_DEPENDENCIES view

Run the UTL\_DTREE.SQL script to create objects that enable you to view direct and indirect dependencies

Predict the effects of changes on dependent objects

### **Manipulating Large Objects**

Describe a LOB object

Diagram the anatomy of a LOB

Manage and list the features on internal LOBs

Describe, manage, and secure BFILEs

Create and use the DIRECTORY object to access and use BFILEs

Prepare BFILEs for usage

Use the BFILENAME function to load BFILEs

Describe the DBMS\_LOB package

### **Creating Triggers**

Describe the different types of triggers and how they execute

List the benefits and guidelines of using database triggers  
Show how triggers are executed with a basic database trigger example  
Show syntax and create DML triggers, and list the DML trigger components  
Explain the firing sequence of triggers  
Create a DML statement and row level triggers  
Use the OLD and NEW qualifiers to reference column values  
Use conditional predicates with triggers

### **Applications for Triggers**

Create triggers for DDL events of CREATE, ALTER, and DROP  
Create triggers for system events of SERVERERROR, STARTUP, SHUTDOWN, LOGON and LOGOFF  
Define a mutating table  
Describe business application scenarios for implementing with triggers  
Describe the privileges required to manage triggers

### **Understanding and Influencing the PL/SQL Compiler**

List the features of native compilation  
Describe the features of the PL/SQL compiler in Oracle Database 10g  
Identify the 3 parameters used to influence compilation (PLSQL\_CODE\_TYPE, PLSQL\_DEBUG, PLSQL\_OPTIMIZE\_LE  
Show how to set the parameters  
Describe the dictionary view used to see how code is compiled (USER\_PLSQL\_OBJECTS)  
Change the parameter settings, recompile code, and view the results  
Describe the compiler warning infrastructure in Oracle Database 10g  
List the steps used in setting compiler warning levels

### **Cursos Equivalentes**

Oracle Database 10g: PL/SQL Fundamentals Self-Study CD Course