Programming with PL/SQL

Introduction

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.

Course Objectives

- Design PL/SQL anonymous blocks that execute efficiently
- Write PL/SQL code to interface with the database
- Describe the features and syntax of PL/SQL
- Use PL/SQL programming constructs and conditionally control code flow (loops, control structures, and explicit cursors)
- Handle runtime errors
- Create simple procedures and functions
- Design PL/SQL packages to group and contain related constructs
- Schedule PL/SQL jobs to run independently
- Create triggers to solve business challenges

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 Subqueries
- 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 Nonpredefined 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 PL/SQL blocks and subprograms
- Describe the uses of procedures
- Create procedures
- Differentiate between formal and actual parameters
- List the features of different parameter modes
- Create procedures with parameters and invoke a procedure
- Handle exceptions in procedures

Creating Stored Functions

- Describe stored functions
- List the CREATE OR REPLACE FUNCTION syntax
- Identify the steps to create a stored function
- Create a stored function in SQL Developer and execute a stored function
- Identify the advantages of using stored functions in SQL statements
- Identify the restrictions of calling functions from SQL statements
- Describe how procedures and functions differ

Creating Packages

- List the benefits or using PL/SQL packages
- Differentiate between a package specification and a package body
- Create packages
- Include public and private constructs in a package
- Call public and private constructs in a package
- Remove packages

Creating Triggers

- Describe different types of triggers
- Describe database triggers and their use
- Create database triggers
- Describe database trigger firing rules
- Remove database triggers

Understanding and Influencing the PL/SQL Compiler

- Describe native compilation and interpreted compilation
- List the features of native compilation
- Switch between native and interpreted compilation for compiled PL/SQL code
- Set the parameters to control aspects of PL/SQL compilation
- Explain the compiler warning mechanism
- List the steps to use the compiler warnings
- Use DBMS_WARNING to implement compiler warnings