Oracle Database 11g: Develop PL/SQL Program Units

What you will learn:

This course introduces students to PL/SQL and explains the benefits of this powerful programming language. Students learn to create PL/SQL blocks of application code that can be shared by multiple forms, reports, and data management applications. Students also learn to create anonymous PL/SQL blocks and are introduced to stored procedures and functions. Students learn about declaring identifiers and trapping exceptions. Demonstrations and hands-on practice reinforce the fundamental concepts.

Students use Oracle SQL Developer to develop these program units. SQL*Plus and JDeveloper are introduced as optional tools.After the basic PL/SQL and SQL language skills, students learn to develop, execute, and manage PL/SQL stored program units such as procedures, functions, packages, and database triggers. Students also learn to manage, PL/SQL subprograms and triggers. Students are introduced to the utilization of some of the Oracle-supplied packages.

Additionally students learn to use Dynamic SQL, understand design considerations when coding using PL/SQL, understand and influence the PL/SQL compiler and manage dependencies.

In this course, students use Oracle SQL Developer as the main tool and SQL*Plus is introduced as an optional tool.

This course counts towards the Hands-on course requirement for the Oracle Database 11g Administrator Certification. Only instructor-led inclass or instructor-led online formats of this course will meet the Certification Hands-on Requirement.

Course Topics:

1: Introduction

- Course Objectives and Agenda
- Describing the Human Resources (HR) Schema
- Identifying the Appendices Used in this Course
- Listing the PL/SQL Development Environments Available in this Course
- Introduction to SQL Developer
- Using SQL Developer
- Reviewing the Online Oracle 11g SQL and PL/SQL Documentation and the Additional Available Resources

1: Introduction to PL/SQL

- PL/SQL Overview
- Benefits of PL/SQL Subprograms
- Overview of the Types of PL/SQL blocks
- Creating and Executing a Simple Anonymous Block
- Generating Output from a PL/SQL Block

1: Declaring PL/SQL Identifiers

- Different Types of Identifiers in a PL/SQL subprogram
- Using the Declarative Section to Define Identifiers
- Storing Data in Variables
- Scalar Data Types
- %TYPE Attribute
- Bind Variables
- Using Sequences in PL/SQL Expressions

1: Writing Executable Statements

- Describing Basic PL/SQL Block Syntax Guidelines
- Commenting Code
- SQL Functions in PL/SQL
- Data Type Conversion
- Nested Blocks
- Operators in PL/SQL

2: Interacting with the Oracle Server

- Including SELECT Statements in PL/SQL to Retrieve data
- Manipulating Data in the Server Using PL/SQL
- The SQL Cursor concept
- Using SQL Cursor Attributes to Obtain Feedback on DML
- Saving and Discarding Transactions

2: Writing Control Structures

- Conditional processing Using IF Statements
- Conditional processing Using CASE Statements
- Simple Loop Statement
- While Loop Statement
- For Loop Statement
- The Continue Statement

2: Using Explicit Cursors

- Understanding Explicit Cursors
- Declaring the Cursor
- Opening the Cursor
- Fetching data from the Cursor
- Closing the Cursor
- Cursor FOR loop
- Explicit Cursor Attributes
- FOR UPDATE Clause and WHERE CURRENT Clause

3: Handling Exceptions

- Understanding Exceptions
- Handling Exceptions with PL/SQL
- Trapping Predefined Oracle Server Errors
- Trapping Non-Predefined Oracle Server Errors
- Trapping User-Defined Exceptions
- Propagate Exceptions
- RAISE_APPLICATION_ERROR Procedure

3: Creating Stored Procedures

- Creating a Modularize and Layered Subprogram Design
- Modularizing Development With PL/SQL Blocks
- Understanding the PL/SQL Execution Environment
- The Benefits of Using PL/SQL Subprograms
- The Differences Between Anonymous Blocks and Subprograms
- Creating, Calling, and Removing Stored Procedures Using the CREATE Command and SQL Developer
- Using Procedures Parameters and Parameters Modes
- Viewing Procedures Information Using the Data Dictionary Views and SQL Developer

3: Creating Stored Functions

- Creating, Calling, and Removing a Stored Function Using the CREATE Command and SQL Developer
- Identifying the Advantages of Using Stored Functions in SQL Statements
- Identify the steps to create a stored function
- Using User-Defined Functions in SQL Statements
- Restrictions When Calling Functions from SQL statements

- Controlling Side Effects When Calling Functions from SQL Expressions
- Viewing Functions Information

3: Creating Triggers

- Working With Triggers
- Identifying the Trigger Event Types and Body
- Business Application Scenarios for Implementing Triggers
- Creating DML Triggers Using the CREATE TRIGGER Statement and SQL Developer
- Identifying the Trigger Event Types, Body, and Firing (Timing)
- Statement Level Triggers Versus Row Level Triggers
- Creating Instead of and Disabled Triggers
- Managing, Testing, and Removing Triggers

3: Using the PL/SQL Compiler

- Using the PL/SQL Compiler
- Using the Initialization Parameters for PL/SQL Compilation
- Using the New PL/SQL Compile Time Warnings
- Overview of PL/SQL Compile Time Warnings for Subprograms
- The Benefits of Compiler Warnings
- The PL/SQL Compile Time Warning Messages Categories
- Setting the Warning Messages Levels: Using SQL Developer, PLSQL_WARNINGS Initialization Parameter, and the DBMS_WARNING Package Subprograms
- Viewing the Compiler Warnings: Using SQL Developer, SQL*Plus, or the Data Dictionary Views