

Oracle Database 12c : Program with PL/SQL

(40 hours)

Prerequisites

- Oracle Database: Introduction to SQL
- Previous programming experience

Course contents:

Introduction to PL/SQL

Identify the benefits of PL/SQL Subprograms
Create a Simple Anonymous Block
Overview of PL/SQL
How to generate output from a PL/SQL Block?
Overview of the types of PL/SQL blocks

Declare PL/SQL Variables

Sequences in PL/SQL Expressions
The %TYPE Attribute
Use variables to store data
List the different Types of Identifiers in a PL/SQL subprogram
What are Bind Variables?
Identify Scalar Data Types
Usage of the Declarative Section to Define Identifiers

Write Anonymous PL/SQL Blocks

Describe Nested Blocks
Describe Basic PL/SQL Block Syntax Guidelines
How to convert Data Types?
Learn to Comment the Code
Identify the Operators in PL/SQL
Deployment of SQL Functions in PL/SQL

SQL Statements in a PL/SQL block

Retrieve Data in PL/SQL
Data Manipulation in the Server using PL/SQL
Avoid Errors by using Naming Conventions when using Retrieval and DML Statements
Understand the SQL Cursor concept
Invoke SELECT Statements in PL/SQL
Use SQL Cursor Attributes to Obtain Feedback on DML
Save and Discard Transactions
SQL Cursor concept

Control Structures

Conditional processing using IF Statements
Conditional processing using CASE Statements
Use the Continue Statement

Describe While Loop Statement
Describe simple Loop Statement
Describe For Loop Statement

Composite Data Types

The %ROWTYPE Attribute
Use PL/SQL Records
Insert and Update with PL/SQL Records
Examine INDEX BY Table Methods
INDEX BY Tables
Use INDEX BY Table of Records

Explicit Cursors

Describe the FOR UPDATE Clause and WHERE CURRENT Clause
Declare the Cursor
What are Explicit Cursors?
Fetch data from the Cursor
Cursor FOR loop
Close the Cursor
The %NOTFOUND and %ROWCOUNT Attributes
Open the Cursor

Exception Handling

Trap User-Defined Exceptions
Trap Non-Predefined Oracle Server Errors
Understand Exceptions
RAISE_APPLICATION_ERROR Procedure
Propagate Exceptions
Trap Predefined Oracle Server Errors
Handle Exceptions with PL/SQL

Stored Procedures

List the benefits of using PL/SQL Subprograms
List the differences between Anonymous Blocks and Subprograms
Create a Modularized and Layered Subprogram Design
Implement Procedures Parameters and Parameters Modes
Create, Call, and Remove Stored Procedures
Modularize Development With PL/SQL Blocks
Understand the PL/SQL Execution Environment
View Procedure Information

Stored Functions

Identify the steps to create a stored function
Control side effects when calling Functions
Create, Call, and Remove a Stored Function
View Functions Information
Restrictions when calling Functions
Identify the advantages of using Stored Functions
Invoke User-Defined Functions in SQL Statements

Debugging Subprograms

Debugging through SQL Developer
How to debug Functions and Procedures?

Packages

Describe Packages
What are the components of a Package?
Develop a Package
Invoke the Package Constructs
View the PL/SQL Source Code using the Data Dictionary
How to enable visibility of a Packages Components?
Listing the advantages of Packages
Create the Package Specification and Body using the SQL CREATE Statement and SQL Developer

Deploying Packages

Persistent State of a Package Cursor
Overloading Subprograms in PL/SQL
Use Forward Declarations to solve Illegal Procedure Reference
Control side effects of PL/SQL Subprograms
Use the STANDARD Package
Persistent State of Packages
Invoke PL/SQL Tables of Records in Packages
Implement Package Functions in SQL and Restrictions

Implement Oracle-Supplied Packages in Application Development

Examples of some of the Oracle-Supplied Packages
Invoke the UTL_MAIL Package
Use the UTL_FILE Package to Interact with Operating System Files
How does the DBMS_OUTPUT Package work?
What are Oracle-Supplied Packages?
Write UTL_MAIL Subprograms

Dynamic SQL

The Execution Flow of SQL
Declare Cursor Variables
Configure Native Dynamic SQL to Compile PL/SQL Code
Dynamic SQL Functional Completeness
What is Dynamic SQL?
Dynamically Executing a PL/SQL Block
How to invoke DBMS_SQL Package?
Implement DBMS_SQL with a Parameterized DML Statement

Design Considerations for PL/SQL Code

Understand Local Subprograms
Implement the NOCOPY Compiler Hint
The Cross-Session PL/SQL Function Result Cache
Usage of Bulk Binding to Improve Performance
Standardize Constants and Exceptions
The DETERMINISTIC Clause with Functions

Write Autonomous Transactions
Invoke the PARALLEL_ENABLE Hint

Triggers

Identify the Trigger Event Types and Body
How to Manage, Test and Remove Triggers?
Identify the Trigger Event Types, Body, and Firing (Timing)
Differences between Statement Level Triggers and Row Level Triggers
Create DML Triggers using the CREATE TRIGGER Statement and SQL Developer
Describe Triggers
Business Application Scenarios for Implementing Triggers
Create Instead of and Disabled Triggers

Creating Compound, DDL, and Event Database Triggers

Implement a Compound Trigger to Resolve the Mutating Table Error
Identify the Timing-Point Sections of a Table Compound Trigger
Comparison of Database Triggers to Stored Procedures
What are Compound Triggers?
Create Database-Event and System-Events Triggers
Understand the Compound Trigger Structure for Tables and Views
System Privileges Required to Manage Triggers
Create Triggers on DDL Statements

PL/SQL Compiler

Overview of PL/SQL Compile Time Warnings for Subprograms
List 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
View Compiler Warnings: Using SQL Developer, SQL*Plus, or the Data Dictionary Views
List the benefits of Compiler Warnings
What is the PL/SQL Compiler?
Describe the Initialization Parameters for PL/SQL Compilation
List the new PL/SQL Compile Time Warnings

Manage Dependencies

Query Direct Object Dependencies using the USER_DEPENDENCIES View
Overview of Schema Object Dependencies
Invalidation of Dependent Objects
Recompile a PL/SQL Program Unit
Understand Remote Dependencies
Query an Objects Status
Fine-Grained Dependency Management in Oracle Database 12c
Display the Direct and Indirect Dependencies