Oracle Database 11g: Advanced PL/SQL

Duration: 3 Days

What you will learn

In this course, students learn to use the advanced features of PL/SQL in order to design and tune PL/SQL to interface with the database and other applications in the most efficient manner. Using advanced features of program design, packages, cursors, extended interface methods, and collections, students learn to write powerful PL/SQL programs. Programming efficiency, use of external C and Java routines, PL/SQL server pages, and fine-grained access are covered.

Students learn to write PL/SQL routines that analyze the PL/SQL applications and caching techniques that can improve performance. Students are also introduced to Virtual Private Database (VPD) to implement security policies and they also learn techniques and tools to strengthen their applications against SQL injection attacks.

Learn To:

PL/SQL designing best practices
Create PL/SQL applications that use collections
Implement a virtual private database with fine-grained access control
Write code to interface with external C and Java applications
Write code to interface with large objects and use SecureFile LOBs
Write and tune PL/SQL code effectively to maximize performance

Audience

Application Developers
Database Administrators
PL/SQL Developer

Prerequisites

Required Prerequisites

Knowledge of SQL

PL/SQL Programming experience

Suggested Prerequisites

Oracle Database: Introduction to SQL

Course Objectives

Design PL/SQL packages and program units that execute efficiently
Write code to interface with external applications and the operating system
Create PL/SQL applications that use collections
Write and tune PL/SQL code effectively to maximize performance
Implement a virtual private database with fine-grained access control
Write code to interface with large objects and use SecureFile LOBs

Course Topics

Introduction
Course objectives
Course agenda
Tables and data used for this course
Overview of the development environments: SQL Developer, SQL Plus

PL/SQL Programming Concepts Review
Identify PL/SQL block structure
Create procedures
Create functions
List restrictions and guidelines on calling functions from SQL expressions
Create packages
Review of implicit and explicit cursors
List exception syntax
Identify the Oracle supplied packages

Designing PL/SQL Code
Describe the predefined data types
Create subtypes based on existing types for an application
List the different guidelines for cursor design
Cursor variables

Using Collections
Overview of collections
Use Associative arrays
Use Nested tables
Use VARRAYs
Compare nested tables and VARRAYs
Write PL/SQL programs that use collections
Use Collections effectively

Manipulating Large Objects
Describe a LOB object
Use BFILEs
Use DBMS_LOB.READ and DBMS_LOB.WRITE to manipulate LOBs
Create a temporary LOB programmatically with the DBMS_LOB package
Introduction to SecureFile LOBs
Use SecureFile LOBs to store documents
Convert BasicFile LOBs to SecureFile LOB format
Enable reduplication and compression

Using Advanced Interface Methods
Calling External Procedures from PL/SQL
Benefits of External Procedures
C advanced interface methods
Java advanced interface methods

Performance and Tuning
Understand and influence the compiler
Tune PL/SQL code

Enable intra unit inlining
Identify and tune memory issues
Recognize network issues

Improving Performance with Caching
Describe result caching
Use SQL query result cache’
PL/SQL function cache ’
Review PL/SQL function cache considerations

Analyzing PL/SQL Code
Finding Coding Information
Using DBMS_DESCRIBE
Using ALL_ARGUMENTS
Using DBMS_UTILITY.FORMAT_CALL_STACK
Collecting PL/Scope Data
The USER/ALL/DBA_IDENTIFIERS Catalog View
DBMS_METADATA Package

Profiling and Tracing PL/SQL Code
Tracing PL/SQL Execution
Tracing PL/SQL: Steps

Implementing VPD with Fine-Grained Access Control
Understand how fine-grained access control works overall
Describe the features of fine-grained access control
Describe an application context
Create an application context
Set an application context
List the DBMS_RLS procedures
Implement a policy
Query the dictionary views holding information on fine-grained access

Safeguarding Your Code Against SQL Injection Attacks
SQL Injection Overview
Reducing the Attack Surface
Avoiding Dynamic SQL
Using Bind Arguments
Filtering Input with DBMS_ASSERT
Designing Code Immune to SQL Injections
Testing Code for SQL Injection Flaws