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, large objects, and collections, students learn to write powerful PL/SQL programs. Students also learn programming efficiency, use of external C and Java routines, fine-grained access and safeguarding code against SQL Injection attacks.

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. Self Study CD-Rom and Knowledge Center courses DO NOT meet the Hands-on Requirement.

Learn To:

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

Audience

Application Developers
Developer
PL/SQL Developer

Prerequisites

Required Prerequisites
Oracle Database 11g: Develop PL/SQL Program Units

Oracle Database 11g: PL/SQL Fundamentals

Oracle Database 11g: SQL Fundamentals II

Oracle Database 11g: SQL Fundamentals I

Suggested Prerequisites
Experience with SQL and PL/SQL required
Familiarity with the Oracle Database
Oracle Database 11g: Introduction to SQL

Oracle Database 11g: Program with PL/SQL

Course Objectives

Tune PL/SQL code
Categorize and explain various types of SQL injection attacks
Apply coding standards to eliminate SQL injection vulnerabilities
Create subtypes based on existing types for an application
Create and use collections
Execute external C programs from PL/SQL
Execute Java programs from PL/SQL
Describe the process of fine-grained access control
Create and maintain LOB data types
Use the DBMS_LOB PL/SQL package to control LOBs
Describe SecureFile LOB features
Identify guidelines for cursor design
Enable SecureFile LOB deduplication, compression, and encryption
Improve memory usage by caching SQL result sets
Set up PL/SQL functions to use PL/SQL result caching
Profile PL/SQL applications

Course Topics

Overview of the Development Environments

SQL Developer
SQL*Plus

Design Considerations

Describe the predefined data types
Create subtypes based on existing types for an application
List the different guidelines for cursor design
Use cursor variables
Pass cursor variables as program parameters
Compare cursor variables to static cursors

Using Collections

Overview of collections
Use Associative arrays
Use Nested tables
Use Varrays
Write PL/SQL programs that use collections
Use Collections effectively

Using Advanced Interface Methods

Calling C from PL/SQL
Calling Java from PL/SQL

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

Manipulating Large Objects
Describe a LOB object
Manage internal LOBs
Describe BFILEs
Create and use the DIRECTORY object to access and use BFILEs
Describe the DBMS_LOB package
Remove LOBs
Create a temporary LOB programmatically with the DBMS_LOB package

Administering SecureFile LOBs
Introduction to SecureFile LOBs
Enable the environment for SecureFile LOBs
Use SecureFile LOBs to store documents
Convert BasicFile LOBs to SecureFile LOB format
Examine the performance of SecureFile LOBs
Enable deduplication and compression
Enable encryption

Tuning and Performance
Understand and influence the compiler
Tune PL/SQL code
Enable intra unit inlining
Identify and tune memory issues

Improving Performance with SQL and PL/SQL Caching
Describe result caching
Use SQL query result cache
PL/SQL function cache

Analyzing PL/SQL Code
Use the supplied packages and dictionary views to find coding information
Determine identifier types and usages with PL/Scope
Use the DBMS_METADATA package to obtain metadata from the data dictionary as XML or creation DDL that can be us

Profiling and Tracing PL/SQL Code
Trace PL/SQL program execution
Profile PL/SQL applications

Safeguarding Your Code Against SQL Injection Attacks
Describe SQL injections
Reduce attack surfaces
Use DBMS_ASSERT
Design immune code
Test code for SQL injection flaws