

Oracle Database 11g: Advanced PL/SQL (40 hrs.)

Prerequisites:

Oracle Database 11g PL/SQL Fundamentals+Program Units

Suggested Prerequisites:

Oracle PL/SQL course or equivalent experience

Course Topics:

Introduction

- Course Objectives
- Lesson Agenda
- Course Agenda
- Appendixes Used in This Course
- Lesson Agenda
- Development Environments: Overview
- What Is Oracle SQL Developer?
- Coding PL/SQL in SQL*Plus 0
- Lesson Agenda
- Tables Used in This Course
- The Order Entry Schema
- The Human Resources Schema
- Oracle g SQL and PL/SQL Documentation

PL/SQL Programming Concepts: Review

- Objectives
- Lesson Agenda
- PL/SQL Block Structure
- Naming Conventions
- Procedures
- Procedure: Example
- Functions
- Function: Example
- Ways to Execute Functions 0
- Lesson Agenda
- Restrictions on Calling Functions from SQL Expressions
- Lesson Agenda
- PL/SQL Packages: Review
- Components of a PL/SQL Package
- Creating the Package Specification
- Creating the Package Body
- Lesson Agenda
- Cursor 0
- Processing Explicit Cursors
- Explicit Cursor Attributes
- Cursor FOR Loops
- Cursor: Example
- Lesson Agenda
- Handling Exceptions

- Exceptions: Example
- Predefined Oracle Server Errors 0
- Trapping Non-Predefined Oracle Server Errors
 - Trapping User-Defined Exceptions
 - Lesson Agenda
 - The RAISE_APPLICATION_ERROR Procedure
 - Lesson Agenda
 - Dependencies
 - Displaying Direct and Indirect Dependencies
 - Lesson Agenda
 - Using Oracle-Supplied Packages
 - Some of the Oracle-Supplied Packages
 - DBMS_OUTPUT Package
 - UTL_FILE Package

Designing PL/SQL Code

- Objectives
- Lesson Agenda
- Guidelines for Cursor Design
- Lesson Agenda
- Cursor Variables: Overview 0
- Working with Cursor Variables
- Strong Versus Weak REF CURSOR Variables
- Step 1: Defining a REF CURSOR Type
- Step 1: Declaring a Cursor Variable
- Step 1: Declaring a REF CURSOR Return Type
- Step 2: Opening a Cursor Variable
- Step 3: Fetching from a Cursor Variable
- Step 4: Closing a Cursor Variable
- Passing Cursor Variables as Arguments 0
- Using the SYS_REFCURSOR Predefined Type
- Rules for Cursor Variables
- Comparing Cursor Variables with Static Cursors
- Lesson Agenda
- Predefined PL/SQL Data Types
- Subtypes: Overview
- Benefits of Subtypes
- Declaring Subtypes
- Using Subtypes
- Subtype Compatibility

Working with Collections

- Objectives
- Lesson Agenda
- Understanding Collections
- Collection Types
- Lesson Agenda
- Using Associative Arrays
- Creating the Array 0
- Traversing the Array
- Lesson Agenda
- Using Nested Tables
- Nested Table Storage

- Creating Nested Tables
- Declaring Collections: Nested Table
- Using Nested Tables
- Referencing Collection Elements 0
- Using Nested Tables in PL/SQL
- Lesson Agenda
- Understanding Varrays
- Declaring Collections: Varray
- Using Varrays
- Lesson Agenda
- Working with Collections in PL/SQL
- Initializing Collections
- Referencing Collection Elements
- Using Collection Methods
- Manipulating Individual Elements
- Lesson Agenda
- Avoiding Collection Exceptions
- Avoiding Collection Exceptions: Example
- Lesson Agenda
- Listing Characteristics for Collections
- Guidelines for Using Collections Effectively

Manipulating Large Objects

- Objectives
- Lesson Agenda
- What Is a LOB?
- Components of a LOB
- Internal LOBs
- Managing Internal LOBs
- Lesson Agenda
- What Are BFILES? 0
- Securing BFILES
- What Is a DIRECTORY?
- Using the DBMS_LOB Package
- DBMS_LOB.READ and DBMS_LOB.WRITE
- Managing BFILES
- Preparing to Use BFILES
- Populating BFILE Columns with SQL
- Populating a BFILE Column with PL/SQL
- Using DBMS_LOB Routines with BFILES
- Lesson Agenda 0
- Initializing LOB Columns Added to a Table
- Populating LOB Columns
- Writing Data to a LOB
- Reading LOBs from the Table
- Updating LOB by Using DBMS_LOB in PL/SQL 0
- Checking the Space Usage of a LOB Table
- Selecting CLOB Values by Using SQL
- Selecting CLOB Values by Using DBMS_LOB
- Selecting CLOB Values in PL/SQL
- Removing LOBs
- Lesson Agenda 0

- Temporary LOBs
- Creating a Temporary LOB
- Lesson Agenda
- SecureFile LOBs
- Storage of SecureFile LOBs
- Creating a SecureFile LOB
- Comparing Performance
- Enabling Deduplication and Compression
- Enabling Deduplication and Compression: Example
- Step 1: Checking Space Usage 0
- Step 1: Checking Space Usage
- Step 2: Enabling Deduplication and Compression
- Step 3: Rechecking LOB Space Usage
- Step 4: Reclaiming the Free Space
- Using Encryption
- Using Encryption: Example
- Migrating from BasicFile to SecureFile Format

Using Advanced Interface Methods

- Objectives
- Calling External Procedures from PL/SQL
- Benefits of External Procedures
- External C Procedure Components
- How PL/SQL Calls a C External Procedure
- The extproc Process
- Development Steps for External C Procedures
- The Call Specification
- Publishing an External C Routine
- Executing the External Procedure
- Java: Overview
- Calling a Java Class Method by Using PL/SQL
- Development Steps for Java Class Methods
- Loading Java Class Methods 0
- Publishing a Java Class Method
- Executing the Java Routine
- Creating Packages for Java Class Methods

Performance and Tuning

- Objectives
- Lesson Agenda
- Native and Interpreted Compilation
- Deciding on a Compilation Method
- Setting the Compilation Method
- Viewing the Compilation Settings
- Setting Up a Database for Native Compilation 0
- Compiling a Program Unit for Native Compilation
- Lesson Agenda
- Tuning PL/SQL Code
- Avoiding Implicit Data Type Conversion
- Understanding the NOT NULL Constraint
- Using the PLS_INTEGER Data Type for Integers
- Using the SIMPLE_INTEGER Data Type
- Modularizing Your Code

- Comparing SQL with PL/SQL
- Using Bulk Binding
- Using SAVE EXCEPTIONS
- Handling FORALL Exceptions
- Rephrasing Conditional Control Statements 0
- Passing Data Between PL/SQL Programs
- Lesson Agenda
- Introducing Intraunit Inlining
- Using Inlining
- Inlining Concepts
- Inlining: Example
- Inlining: Guidelines

Improving Performance with Caching

- Objectives
- Lesson Agenda
- What Is Result Caching?
- Increasing Result Cache Memory Size
- Setting Result_Cache_Max_Size
- Enabling Query Result Cache
- Using the DBMS_RESULT_CACHE Package
- Lesson Agenda
- SQL Query Result Cache 0
- Clearing the Shared Pool and Result Cache
- Examining the Memory Cache
- Examining the Execution Plan for a Query
- Examining Another Execution Plan
- Executing Both Queries
- Viewing Cache Results Created
- Re-Executing Both Queries
- Viewing Cache Results Found 0
- Lesson Agenda
- PL/SQL Function Result Cache
- Marking PL/SQL Function Results to Be Cached
- Clearing the Shared Pool and Result Cache
- Lesson Agenda
- Creating a PL/SQL Function Using the RESULT_CACHE Clause
- Lesson Agenda
- Calling the PL/SQL Function Inside a Query
- Verifying Memory Allocation
- Viewing Cache Results Created 0
- Calling the PL/SQL Function Again
- Viewing Cache Results Found
- Confirming That the Cached Result Was Used

Analyzing PL/SQL Code

- Objectives
- Lesson Agenda
- Finding Coding Information
- Using SQL Developer to Find Coding Information
- Using DBMS_DESCRIBE
- Using ALL_ARGUMENTS
- Using SQL Developer to Report on Arguments

- Using DBMS_UTILITY.FORMAT_CALL_STACK
- Finding Error Information 0
- Lesson Agenda
- PL/Scope Concepts
- Collecting PL/Scope Data
- Using PL/Scope
- The USER/ALL/DBA_IDENTIFIERS Catalog View
- Sample Data for PL/Scope 0
- Collecting Identifiers
- Viewing Identifier Information
- Performing a Basic Identifier Search
- Using USER_IDENTIFIERS to Find All Local Variables
- Finding Identifier Actions
- Lesson Agenda
- DBMS_METADATA Package 0
- Metadata API
- Subprograms in DBMS_METADATA
- FETCH_xxx Subprograms
- SET_FILTER Procedure
- Filters
- Examples of Setting Filters
- Programmatic Use: Example
- Programmatic Use: Example
- Browsing APIs
- Browsing APIs: Examples

Profiling and Tracing PL/SQL Code

- Objectives
- Lesson Agenda
- Tracing PL/SQL Execution
- Tracing PL/SQL: Steps
- Step 1: Enable Specific Subprograms
- Steps 2 and 3: Identify a Trace Level and Start Tracing
- Step 4: Turn Off Tracing 0
- Step 5: Examine the Trace Information
- plsql_trace_runs and plsql_trace_events
- Lesson Agenda
- Hierarchical Profiling Concepts
- Using the PL/SQL Profiler
- Understanding Raw Profiler Data
- Using the Hierarchical Profiler Tables
- Using DBMS_HPROF.ANALYZE
- Using DBMS_HPROF.ANALYZE to Write to Hierarchical Profiler Tables
- Analyzer Output from the DBMSHP_RUNS Table
- Analyzer Output from the DBMSHP_FUNCTION_INFO Table
- plshprof: A Simple HTML Report Generator
- Using plshprof
- Using the HTML Reports

Implementing Fine-Grained Access Control for VPD

- Objectives
- Lesson Agenda
- Fine-Grained Access Control: Overview

- Identifying Fine-Grained Access Features
- How Fine-Grained Access Works
- Why Use Fine-Grained Access?
- Lesson Agenda
- Using an Application Context 0
- Creating an Application Context
- Setting a Context
- Implementing a Policy
- Step 2: Creating the Package
- Step 3: Defining the Policy
- Step 4: Setting Up a Logon Trigger
- Example Results
- Data Dictionary Views
- Using the ALL_CONTEXT Dictionary View
- Policy Groups
- More About Policies

Safeguarding Your Code Against SQL Injection Attacks

- Objectives
- Lesson Agenda
- Understanding SQL Injection
- Identifying Types of SQL Injection Attacks
- SQL Injection: Example
- Assessing Vulnerability
- Avoidance Strategies Against SQL Injection
- Protecting Against SQL Injection: Example
- Lesson Agenda 0
- Reducing the Attack Surface
- Expose the Database Only Via PL/SQL API
- Using Invoker's Rights
- Reducing Arbitrary Inputs
- Strengthen Database Security
- Lesson Agenda
- Using Static SQL
- Using Dynamic SQL
- Lesson Agenda
- Using Bind Arguments with Dynamic SQL
- Using Bind Arguments with Dynamic PL/SQL
- What If You Cannot Use Bind Arguments?
- Lesson Agenda
- Understanding DBMS_ASSERT
- Formatting Oracle Identifiers
- Working with Identifiers in Dynamic SQL
- Choosing a Verification Route 0
- Avoiding Injection by Using DBMS_ASSERT.ENQUOTE_LITERAL
- Avoiding Injection by Using DBMS_ASSERT.SIMPLE_SQL_NAME
- DBMS_ASSERT Guidelines
- Lesson Agenda
- Using Bind Arguments 0
- Avoiding Privilege Escalation
- Beware of Filter Parameters
- Trapping and Handling Exceptions

- Lesson Agenda
- Coding Review and Testing Strategy
- Reviewing Code
- Running Static Code Analysis
- Testing with Fuzzing Tools
- Generating Test Cases