

Oracle Database 12c: Advanced PL/SQL Programming & Tuning (5 Days)

Description

The PL/SQL programming language is at the core of most Oracle database applications. First, we will explore the advanced features of the language which allow powerful and effective database applications to be built. Next, we will discuss performance tuning techniques which allows these applications to run efficiently. Finally, we will consider critical security measures which should be implemented to counter hacker attacks and other security threats.

Audience

The target audience for this course is senior application developers. Developers who will be building, debugging and tuning PL/SQL program units will benefit from this course.

Prerequisites

Knowledge and working experience with Oracle 12c SQL and PL/SQL are required.

Course Objectives

The objectives for this course are as follows:

- Among the specific topics contained within these training materials are:
- Invoking external procedures and integrating these into PL/SQL applications including external Java classes using the JDBC interface and external C programs contained within DLL libraries.
- Using dynamic SQL to extend the functionality and flexibility of database programs, including the DBMS_SQL() system-supplied package for maximum flexibility.
- Identifying SQL injection attack vulnerabilities within an application and applying countermeasures to address security risks and protect against hacking.
- Incorporate collections into application logic to increase efficiency and execution speed.
- Working with LOBs, including piece-wise data manipulation and dynamic modification of SecureFiles storage options.
- Expanding database application functionality with advanced system-supplied database utility packages, integrating ones applications with external mail systems, database internals and other facilities.
- Tuning with the DBMS_PROFILER() system-supplied package and debugging with the DBMS_TRACE() system-supplied package.
- Writing efficient PL/SQL code and avoiding common coding mistakes.
- Enabling native compilation and execution of all database-resident program units.
- Controlling and managing PL/SQL compilation for high-efficiency execution.
- Analyze PL/SQL code structure by means of the PL/Scope facility.
- Tune bottlenecks using the PL/SQL Hierarchical Profiler.
- Implementing fine-grained security mechanisms as part of an advanced security model using application contexts and the Oracle virtual private database (VPD).
- Dynamic partitioning and DML parallelization using the system-supplied package DBMS_PARALLEL_EXECUTE().
- Using the wrap utility to hide the source code of database-resident programs, even from the owner or authorized users of the programs.

Course Outline

Advanced Programming: Why Needed & PL/SQL Execution Internals

- Why Advanced Programming?
- SQL & PL/SQL Execution Internals
- SQL & PL/SQL PGA Internals

Advanced Programming: Dynamic SQL

- Advantages Of Dynamic SQL
- Native Dynamic SQL
- Dynamic SQL Using DBMS_SQL()

Advanced Programming: Using Collections

- About Collections
- Bulk Bind Using Collections
- About SQL%BULK_ROWCOUNT()
- About SQL%BULK_EXCEPTIONS()
- Collection Methods
- More About RETURNING Clause
- Advanced Collection Features
- IN INDICES OF Clause
- IN VALUES OF Clause

Advanced Programming: Java & C Interface Methods

- Advanced Program Interfaces
- Calling Java Classes
- Calling C Programs

System-Supplied Packages: DBMS_METADATA() – Part I

- Why Retrieve Object Definitions?
- Retrieving Default Metadata
- Retrieving Customized Metadata
- Using SET_COUNT()
- Using ADD_TRANSFORM()
- Using FETCH DDL()
- Calling FETCH_DDL()

System-Supplied Packages: DBMS_METADATA() – Part II

- SET_TRANSFORM_PARAM()
- GET_QUERY()

System-Supplied Packages: DBMS_METADATA() – Part III

- FETCH CLOB()
- SET_FILTER() Dependent Objects
- SET_PARSE_ITEM()
- Primary & Dependent Object DDL

System-Supplied Packages: DBMS_REDEFINITION()

- About Table Redefinition
- Using DBMS_REDEFINITION()

- DBA_REDEFINITION_ERRORS
- CAN_REDEF_TABLE()
- START_REDEF_TABLE()
- FINISH_REDEF_TABLE()
- ABORT_REDEF_TABLE()
- COPY_TABLE_DEPENDENTS()
- SYNC_INTERIM_TABLE()

System-Supplied Packages: DBMS_LOB()

- Working With External BFILES
- Working With Internal LOBS
- SUBSTR()
- INSTR()
- Dynamic SECUREFILE Options

High-Performance: Advanced System-Supplied Packages

- COMPRESSION & UTL_COMPRESS()
- LZ_COMPRESS()
- LZ_UNCOMPRESS()
- DBMS_DESCRIBE()
- UTL_MAIL()
- DBMS_UTILITY()
- COMPILE_SCHEMA()
- DB_VERSION()
- GET_PARAMETER_VALUE()
- WAIT_ON_PENDING_DML()
- GET_TIME()
- GET_ENDIANNESNESS()
- DBMS_FILE_TRANSFER()

High Performance: Programming & Coding Techniques

- Autonomous Transactions
- Using NOCOPY FOR Parameters
- Choosing The Optimum Data Type
- About NOT NULL
- Useful PL/SQL Coding Techniques
- Handling String Literals
- User-Defined SQL Functions

High Performance: Influencing Oracle PL/SQL Compilation

- PL/SQL Compiler Optimization
- PLSQL_OPTIMIZE_LEVEL
- Controlling Compilation Messages
- PL/SQL Native Execution
- Wrapping Source Code

High Performance: Dynamic Partitioning & Parallelization

- Dynamic Partitioning (Chunks)
- Creating & Processing Chunks
- CREATE_TASK()

- CREATE_CHUNKS_BY_ROWID()
- CREATE_CHUNKS_BY_NUMBER_COL()
- EXECUTE_RUN_TASK()
- TASK_STATUS()
- DROP_TASK()
- Monitoring Chunk Processing

High Performance: Using PL/SCOPE For Code Analysis

- Configuring PL/SCOPE
- PLSCOPE_SETTINGS
- Using PL/SCOPE Data

High Performance: Tuning With The Hierarchical Profiler

- What Is The Hierarchical Profiler?
- Configuring The Profiler
- Managing Profiler Runs
- Analyzing Profiler Data
- Interpreting The Results
- DBMSHP_RUNS
- DBMSHP_FUNCTION_INFO
- DBMSHP_PARENT_CHILD_INFO

High Performance: Debugging With DBMS_TRACE()

- Using The Trace Facility
- DBMS_TRACE() To Manage Runs
- Examining The Trace Data
- EVENT_KIND Values

Application Security: SQL Injection Attacks

- Understanding The Threat
- Applying Countermeasures

Application Security: Virtual Private Databases

- Understanding VPDS
- Preparing For A VPD
- Configuring A VPD
- Managing Application Contexts
- Managing Policies & Security Rules