ProTech "Charting the Course ...

protechtraining.com

... to Your Success!"

Oracle Database 12c: PL/SQL III – Advanced Programming & Tuning

Course Summary

Description

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

Objectives

By the end of this course, students will be able to:

- Invoke external procedures and integrating these into PL/SQL applications. These include external Java classes using the JDBC interface and external C programs contained within DLL libraries.
- Use dynamic SQL to extend the functionality and flexibility of database programs, including the DBMS_SQL() system-supplied package for maximum flexibility.
- Identify SQL injection attack vulnerabilities within an application and applying countermeasures to address security risks and protect against hacking.
- Incorporate collections and other advanced types into application logic to increase efficiency and execution speed.
- Work with LOBs, including piece-wise data manipulation and dynamic modification of SecureFiles storage options.
- Expand database application functionality with advanced system-supplied database utility packages, integrating ones applications with external mail systems, database internals and other facilities.
- Tune with the DBMS_PROFILER() system-supplied package and debugging with the DBMS_TRACE() system-supplied package.
- Write efficient PL/SQL code and avoiding common coding mistakes.
- Enable native compilation and execution of all database-resident program units.
- Control and manage PL/SQL compilation for high-efficiency execution.
- Analyze PL/SQL code structure by means of the PL/Scope facility.
- Analyze PL/SQL application performance and tune bottlenecks using the PL/SQL Hierarchical Profiler.
- Implement 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().
- Use the wrap utility to hide the source code of database-resident programs, even from the owner or authorized users of the programs.

Oracle Database 12c: PL/SQL III – Advanced Programming & Tuning

Course Summary (con't)

Topics

- Execution internals
- Advanced programming: using collections
- Advanced programming: Java & C interface methods
- System-supplied packages: DBMS_METADATA() – Part I
- System-supplied packages: DBMS_METADATA() – Part II
- System-supplied packages: DBMS_METADATA() – Part III
- System-supplied packages: DBMS_REDEFINITION()
- High-performance: advanced system-supplied packages
- High performance: programming & coding techniques
- High performance: influencing Oracle PL/SQL compilation
- High performance: dynamic partitioning & parallelization
- High performance: using PL/SCOPE for code analysis
- High performance: tuning with the hierarchical profiler
- High performance: debugging with DBMS_TRACE()
- Application security: SQL injection attacks
- Application security: virtual private databases

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 textbook.

Prerequisites

Oracle Database 12c: SQL Fundamentals (Levels I & II)
Oracle Database 12c: PL/SQL Fundamentals (Levels I & II)

Duration

Five days

ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Oracle Database 12c: PL/SQL III – Advanced Programming & Tuning

Course Outline

I. Execution Internals
- A. Why Advanced Programming?
- B. SQL & PL/SQL Execution Internals
- C. SQL & PL/SQL PGA Internals Advanced Programming: Dynamic SQL
- D. Advantages Of Dynamic SQL
- E. Native Dynamic SQL
- F. Dynamic SQL Using DBMS_SQL()

II. Advanced Programming: Using Collections
- A. About Collections
- B. Bulk Bind Using Collections
- C. About SQL%BULK_ROWCOUNT()
- D. About SQL%BULK_EXCEPTIONS()
- E. Collection Methods
- F. More About Returning Clause
- G. Advanced Collection Features
- H. In Indices Of Clause
- I. In Values Of Clause

III. Advanced Programming: Java & C Interface Methods
- A. Advanced Program Interfaces
- B. Calling Java Classes
- C. Calling C Programs

IV. System-Supplied Packages: DBMS_METADATA() – Part I
- A. Why Retrieve Object Definitions?
- B. Retrieving Default Metadata
- C. Retrieving Customized Metadata
- D. Using SET_COUNT()
- E. Using ADD_TRANSFORM()
- F. Using FETCH DDL()
- G. Calling FETCH_DDL()

V. System-Supplied Packages: DBMS_METADATA() – Part II
- A. SET_TRANSFORM_PARAM()
- B. GET_QUERY()

VI. System-Supplied Packages: DBMS_METADATA() – Part III
- A. Fetch CLOB()
- B. SET_FILTER() Dependent Objects
- C. SET_PARSE_ITEM()
- D. PRIMARY & DEPENDENT Object DDL

VII. System-Supplied Packages: DBMS_REDEFINITION()
- A. About Table Redefinition
- B. Using DBMS_REDEFINITION()
- C. DBA_REDEFINITION_ERRORS
- D. CAN_REDEF_TABLE()
- E. START_REDEF_TABLE()
- F. FINISH_REDEF_TABLE()
- G. ABORT_REDEF_TABLE()
- H. COPY_TABLE_DEPENDENTS()
- I. SYNC_INTERIM_TABLE()

VIII. System-Supplied Packages: DBMS_LOB()
- A. Working With External BFILES
- B. Working With Internal LOBs
- C. SUBSTR()
- D. INSTR()
- E. Dynamic SecureFile options

IX. High-Performance: Advanced System-Supplied Packages
- A. COMPRESSION & UTL_COMPRESS()
- B. LZ_COMPRESS()
- C. LZ_UNCOMPRESS()
- D. DBMS_DESCRIBE()
- E. UTL_MAIL()
- F. DBMS_UTILITY()
- G. COMPILE_SCHEMA()
- H. DB_VERSION()
- I. GET_PARAMETER_VALUE()
- J. WAIT_ON_PENDING_DML()
- K. GET_TIME()
- L. GET_ENDIANNESS()
- M. DBMS_FILE_TRANSFER()

Oracle Database 12c: PL/SQL III – Advanced Programming & Tuning

Course Outline (con't)

X. High Performance: Programming & Coding Techniques
- A. Autonomous Transactions
- B. Using NOCOPY For Parameters
- C. Choosing The Optimum Data Type
- D. About NOT NULL
- E. Useful PL/SQL Coding Techniques
- F. Handling String Literals
- G. User-Defined SQL Functions

XI. High Performance: Influencing Oracle PL/SQL Compilation
- A. PL/SQL Compiler Optimization
- B. PLSQL_OPTIMIZE_LEVEL
- C. Controlling Compilation Messages
- D. PL/SQL Native Execution
- E. Wrapping Source Code

XII. High Performance: Dynamic Partitioning & Parallelization
- A. Dynamic Partitioning (CHUNKS)
- B. Creating & Processing CHUNKS
- C. CREATE_TASK()
- D. CREATE_CHUNKS_BY_ROWID()
- E. CREATE_CHUNKS_BY_NUMBER_COL()
- F. EXECUTE_RUN_TASK()
- G. TASK_STATUS()
- H. DROP_TASK()
- I. Monitoring CHUNK Processing

XIII. High Performance: Using PL/SCOPE For Code Analysis
- A. DBMSHP_RUNS
- B. DBMSHP_FUNCTION_INFO
- C. DBMSHP_PARENT_CHILD_INFO

XIV. High Performance: Debugging With DBMS_TRACE()
- A. Using The Trace Facility
- B. DBMS_TRACE() To Manage Runs
- C. Examining The Trace Data
- D. EVENT_KIND Values

XV. Application Security: SQL Injection Attacks
- A. Understanding The Threat
- B. Applying Countermeasures

XVI. Application Security: Virtual Private Databases
- A. Understanding VPD's
- B. Preparing For A VPD
- C. Configuring A VPD
- D. Managing Application Contexts
- E. Managing Policies & Security Rules
- F. Configuring PL/SCOPE
- G. PLSCOPE_SETTINGS
- H. Using PL/SCOPE Data

XVII. High Performance: Tuning With The Hierarchical Profiler
- A. What Is The Hierarchical Profiler?
- B. Configuring The Profiler
- C. Managing Profiler Runs
- D. Analyzing Profiler Data
- E. Interpreting The Results