OR5475 - Troubleshooting, Debugging and Tuning Oracle PL/SQL Programs

Course ID#: 0370-242-DB-W

Hours: 14

Course Content

Course Description:

This fast-paced, hands-on course helps the student identify performance issues and problem areas within their PL/SQL code using various techniques and packages. Some of the topics covered include Profiling and Tracing, Debugging, Tuning and Exception Handling. These topics include details on the PL/SQL Optimizer, Subprogram Inlining, Subprogram Invocations, Bulk Binding, Pipelined Table Functions, Helpful Pragmas and Native Compilation and Execution of PL/SQL code. A number of useful Oracle supplied packages are covered including DBMS_PROFILER, DBMS_TRACE, DBMS_HPROF, DBMS_WARNING and DBMS_PARALLEL_EXECUTE. Oracle versions 11g and 12c are covered..

Audience:

Developers and DBAs

Prerequisites:

PL/SQL experience with stored procedures, functions and packages along with embedded SQL, explicit cursors, loop structures and basic exception handling.

Topics

Profiling, Tracing, Debugging Using Oracle Supplied Packages

- DBMS_PROFILER
  - Analyze each program statement
  - Collect runtime statistics
- DBMS_TRACE
  - Trace program and subprogram execution steps
  - Collect runtime statistics
- DBMS_HPROF
  - Hierarchical Profiler
  - Analyze SQL and PL/SQL statements separately
  - Generate HTML Reports
- DBMS_DEBUG
  - Debug server side code
  - The PLSQL_DEBUG parameter
  - Using the COMPILE DEBUG option
  - Setting breakpoints
  - Analyzing variables

PL/SQL Error and Exception Handling

- Compile Time Warnings
  - Informational
  - Performance
  - Severe
  - Using the PLSQL_WARNINGS compilation parameter
  - The DBMS_WARNING Package
- Exception Categories
  - Predefined
  - User Defined
  - Steps (DECLARE, RAISE, HANDLE)
  - Internally Defined (ORA-n errors)

OR5475 - Troubleshooting, Debugging and Tuning Oracle PL/SQL Programs

Course ID#: 0370-242-DB-W

Hours: 14

  - – Using PRAGMA EXCEPTION_INIT
- Raising Exceptions Explicitly
- Reraising the current exception
- Propagation of Exceptions
- Error Code and Error Message Retrieval
  - – Using the DBMS_Utility Package
  - – SQLCODE, SQLERRM Functions

Understanding the PL/SQL Optimizer

  - – PL/SQL Optimizer Overview
  - – PL/SQL Optimizer Parameters
    - ▪ PLSQL_OPTIMIZE_LEVEL
- Subprogram Inlining
  - – Performance Benefits and Considerations
  - – Using the PRAGMA INLINE Directive

PL/SQL Performance Considerations

- Code to Tune
  - – SQL Code
  - – Function Calls
  - – Passing Parameters and the NOCOPY option
  - – Loop Considerations
  - – Do data types matter?
  - – Implicit Conversions
  - – Supplied Functions
  - – Conditional Tests Order
- Subprogram Invocations
- Table Functions
- Pipelined Table Functions
  - – Options
  - – PIPELINED
  - – PARALLEL_ENABLE
  - – DETERMINISTIC
  - – Fetching Results
- The DBMS_PARALLEL_EXECUTE Package
  - – Updating large tables in parallel

Using Bulk Binding for Performance

- Performance Benefits
- Review of PL/SQL Collections
- Using the FORALL Statement
  - – INSERT, UPDATE, DELETE Clauses
  - – FORALL Clauses
  - – INDICES OF
  - – VALUES OF
  - – Handling FORALL Exceptions
  - – The SAVE EXCEPTIONS Clause
  - – SQL%BULK_EXCEPTIONS Associative Array
  - – Accessing the ERROR_INDEX Attribute
  - – Accessing the ERROR_CODE Attribute
  - – Using the SQL%BULK_EXCEPTIONS.count Method
  - – SQL%BULK_ROWCOUNT Attribute Usage
  - – The RETURNING BULK COLLECT INTO Clause
- The BULK COLLECT Clause
  - – Usages
  - – The SELECT INTO Statement
  - – A FETCH Statement
  - – In a RETURNING INTO Clause
  - – Setting Row Limits
  - – Using the FETCH ... LIMIT Clause

PL/SQL Compilation Options

- Native Compilation and Execution of PL/SQL Code
  - – Benefits of Native Compilation
  - – When to use Native Compilation
  - – The PLSQL_CODE_TYPE Compilation Parameter