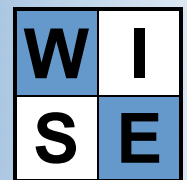


- COURSE CODE:** ORAAPS
- COURSE TITLE:** Advanced PL/SQL & Performance Workshop
- CURRENCY:** 10g & 11g
- AUDIENCE:** PL/SQL developers, DBAs and production DBAs.
- PREREQUISITES:** Completion of our course 'PL/SQL Application Workshop' (ORAPSW) or equivalent knowledge.
- DURATION:** 4 days
- SUMMARY:** This lab-oriented PL/SQL presentation focuses on the advanced features of Oracle's PL/SQL and SQL features; additionally, all of the attendant performance and tuning issues are elaborated on.
- Among the featured topics are:
- Collections
 - SPARSE collections
 - Table functions
 - BULK COLLECT
 - Cursor variables
 - Dynamic SQL
 - Packages
 - PL/SQL optimisation
 - Locking & concurrency
 - Performance and good practices
- OBJECTIVES:** Upon completion of this course, the participant should be able to develop and tune advanced PL/SQL applications.



1. REVIEW OF PL/SQL FUNDAMENTALS

- Client-server architecture
- SQL*Plus
- Programming conventions
- Inclusion of SQL
- Conditional controls
- Error/exception handling
- Iterative controls
- Cursors

2. USER-DEFINED TYPES

- Definitions
- Object types
- REFs
- Varrays (collections)
- Nested tables (collections)
- CREATE TYPE ... semantics

3. COLLECTIONS CONCEPTS

- Definitions
- When to use
- Advantages
- Storage considerations
- Performance: varrays vs. nested tables
- Design considerations
- Multi-level collections

4. PL/SQL USE OF COLLECTIONS

- Nested tables
- Varrays
- Associative arrays (i.e., index by table)
- Collection types definition
- Collection variables declaration
- Collection initialisation and reference
- Collection access via SQL (e.g., INSERT, UPDATE)
- Multi-level collection
- Collection methods (e.g., FIRST/LAST)
- Assignment via set operators (e.g., MULTISSET)
- Exception handling
- Comparing collections

5. PL/SQL RECORDS

- Definition
- When to use
- Advantages
- Creation
- Records as parameters and return values
- Record initialisation
- Record comparisons
- SQL manipulation of records

6. TABLE FUNCTIONS

- Definitions
- When to use
- REF cursor
- Fetching from result sets exception handling
- Passing data via cursor variables
- Performing DML on
- Performing DML inside
- Pipelining
- Querying
- Returning result sets
- Data transformation

7. CURSOR OPERATIONS

- Implicit vs. explicit cursors
- FETCH VIA BULK COLLECT
- Non-correlated vs. correlated subqueries
- Cursor attributes
- REF CURSORS
- Fetching into collections
- Cursor expressions

8. TRANSACTION PROCESSING

- Review of Oracle's locking and concurrency options
- What is and Oracle transaction?
- Autonomous transactions
- COMMITs
- SAVEPOINTs
- ROLLBACKs



9. DYNAMIC SQL

- Definition
- When to use
- EXECUTE IMMEDIATE
- Coding queries
- Bulk dynamic SQL
- Bulk dynamic binds

10. PL/SQL PACKAGES

- Definition
- When to use
- Advantages
- CREATE PACKAGE semantics
- Package specifications and structure
- Package overloading
- Private vs. public variables
- Overloading subprograms
- DBMS_ALERT
- DBMS_OUTPUT
- DBMS_PIPE
- DBMS_PROFILER
- UTL_FILE
- UTL_HTTP

11. LOCKING & CONCURRENCY

- Purpose
- Lock types
- Releasing locks
- Modes (e.g., exclusive)
- Isolation levels (e.g., serializable)
- Data lock conversion vs. lock escalation
- Deadlocks
- How to control deadlocks
- Referential integrity (RI) locks
- Distributed transaction locks
- Latches
- Parse locks

12. PL/SQL TUNING TOOLS & BEST PRACTICES

- Tuning issues
- Explain, traces and other tools (e.g., TKPROF)
- Overview of Oracle's Cost-Based Optimizer (CBO)
- Statistics of Oracle DB objects (e.g., DBMS_STATS)
- PL_SQL_OPTIMIZE_LEVEL options
- Controlling CPU utilisation
- Choice of indexes
- Efficient SQL (e.g., SELECTs)
- Efficient SQL predicates (e.g., local vs. join)
- Efficient joins
- Optimum data types (e.g., *PLS_INTEGER*)
- Optimise VARCHAR2
- Function calls
- Efficient loops (e.g., FORALL)
- Bulk fetching (e.g., BULK COLLECT)
- Dynamic SQL and cursor variables
- SQL PROCEDURES ... NOCOPY
- Native PL/SQL compilation
- Table functions (optimising multiple calls)
- Dual alternative
- Reusable SQL
- Using hints
- Using optimum IF statements
- Temporary table usage
- Triggers
- Lock optimisation

13. PL/SQL DEBUGGING

- DBMS_DEBUG
- ADP_DEBUG
- How to debug a PL/SQL block
- How to start the 'debuggee'
- How to start the 'debugger'
- Breakpoints
- Executing the debugger
- 'Stepping' the debugger
- ADP package definition
- ADP debugger body
- XXX.SQL strings