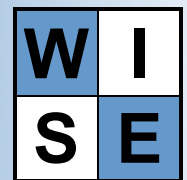


- COURSE CODE:** ORAASE
- COURSE TITLE:** Oracle Advanced SQL via **EXPLAIN PLAN**
- CURRENCY:** 10g & 11g
- AUDIENCE:** SQL*Plus application developers, DBAs and system administrators.
- PREREQUISITES:** Minimum of 6-months of SQL*Plus and PL/SQL developer experience or its equivalent.
- DURATION:** 4 days
- SUMMARY:** This course focuses on best practices and performance issues as these relate to Oracle SQL and PL/SQL. To this end, the Oracle EXPLAIN PLAN and related monitoring tools are central. As the EXPLAIN PLAN facility assumes firm grounding in Oracle's:
- Indexing options
 - Cost-Based Optimizer (CBO) statistics
 - I/O types (access paths)
 - Partitioning options
 - PL/SQL good practices
 - Etc.
- these are covered in much detail.
- OBJECTIVES:** Upon completion of this intensive hands-on course, the delegate should feel comfortable when working with the following 'Core Topics', especially SQL query optimisation.



1. A QUICK REVIEW OF ORACLE'S LOGICAL COMPONENTS

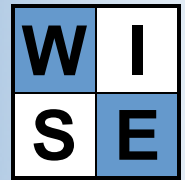
- Database
- Table space
- Data blocks
- Extents
- Segments
- Table
- Indexes
- Data dictionary
- Instance
- Partition
- Data integrity
- Schema
- Views
- Clusters
- Synonyms
- Data types
- Oracle application limits

2. HANDLING 'NULLs'

- Definition
- When to use
- In arithmetic operations
- In comparisons
- In conditions
- In indexes
- With column functions (e.g., SUM, AVG)
- In ORDER BY
- Primary key vs. foreign key considerations

3. ADVANCED COMMON SELECTs

- SELECT * and performance/maintainability
- DISTINCT vs. GROUP BY
- Partition clause
- With READ ONLY
- FOR UPDATE
- Selecting a sample (statistics)
- Selecting sequence values
- Hands-on workshop



4. GROUPING QUERIES

- Definition
- When to use
- Basic GROUP BY
- GROUP BY HAVING
- GROUP BY CUBE
- GROUPING SETS

5. JOINS

- Definition
- When to use
- Join methods
- Self join
- INNER, OUTER & CROSS joins
- NATURAL joins
- Importance of local predicates
- Partitioned outer joins
- Star join transformation option

6. IN-LINE VIEWS

- Definition
- When to use
- Subquery factoring (i.e., common table expressions)
- UPDATE using complex nested table expressions

7. COMPOUND QUERIES

- Definition
- When to use
- UNION vs. UNION ALL
- INTERSECT
- MINUS
- MULTISSET
- MULTISSET EXCEPT
- MULTISSET INTERSECT
- MULTISSET UNION
- Hands-on workshop

8. CASE EXPRESSIONS

- Definition
- When to use
- Simple CASE
- Searched CASE
- Alternative to UNION
- Importance of local predicates
- Hands-on workshop



9. DATE/TIME ARITHMETIC

- Oracle date/time description
- Date and time arithmetic
- Hands-on workshop

10. SUBQUERIES

- Definition
- When to use
- Non-correlated
- Correlated
- IN vs. EXISTS (and NOT IN vs. NOT EXISTS)
- Subqueries vs. joins
- UPDATE via subquery
- Correlated updates
- Hands-on workshop

11. MERGE STATEMENT

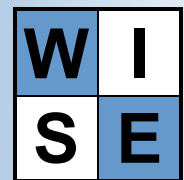
- Definition
- When to use
- INSERT vs. UPDATE
- Hands-on workshop

12. INDEX DESIGN & USAGE

- Advantages
- Disadvantages
- How to avoid sorts
- How to select and arrange keys in indexes
- How to use the SQLAccess Advisor
- When to use composite indexes
- Function-based indexes – advantages

13. I/O TYPES (i.e., ACCESS PATHS)

- Definition
- Full table space scan
- ROWID scans
- Index scans
- Block vs. row access
- Index unique scans
- Index range scans
- Index skip scans
- Full scans
- Index joins
- Bitmap index
- Hash access



14. TRANSACTION PROCESSING

- SQL execution process
- What is a transaction?
- Repeatable Read with Read-only transactions
- Using cursors
- User locks
- Serializable transactions
- Autonomous transactions
- LOCK table
- COMMIT
- ROLLBACK
- Savepoints
- Use of temporary tables

15. OPTIMIZER STATISTICS

- Understanding
- Automatic gathering
- Manual gathering
- System statistics
- Managing statistics
- How to view

16. THE EXPLAIN PLAN, USE & INTERPRETATION

- Definition
- Output tables
- Displaying
- Interpreting
- Hands-on workshop (selects, joins, subqueries, unions, CASE expressions, indexes, etc.)

17. EFFICIENT SQL BEST PRACTICES

- Optimizer statistics maintenance
- Access plan
- Predicate choices
- Control number of SQL statements
- Sort controls
- Optimizer hints
- Choice of indexes
- Choice of optimizer goal

18. PL/SQL BEST PRACTICES

- Control number of SQL statements
- CASE vs. other SQL expressions
- SQL vs. procedural language
- How to improve CURSOR performance
- When to use SQL functions
- Temporary table usage
- How to keep the optimizer informed