SQL query optimization for Oracle Database 11g/12c

REFERENCE	COT
INTENDED PARTICIPANTS	This course is intended for database developers, database administrators and SQL developers to identify and resolve inefficient SQL code. It deals with study methods and analysis that allow access to different levels of detail about how the Oracle database executes SQL statements. Proficiency in SQL language is required.
LEARNING OBJECTIVES	Define the effect that statistics for the optimizer can have on the performance of SQL statements. Identify poorly performing SQL statements. Identify possible data access methods and the different methods of included joints. Edit a SQL statement in order to optimize performance. Generate an application trace in the different levels of the application architecture. Understand decision-making by the instruction optimizer regarding access to data.
DURATION	3 days
DELIVERY PRICE	Contact Us

Learn to:

- Identify inefficient SQL statements using Oracle tools
- Write more efficient SQL instructions
- Use the Automatic SQL Tuning function
- Monitor resource-intensive SQL statements and generate the corresponding track
- Use the SQL monitoring function in real time
- Manage the optimizer statistics for database objects

COURSE CONTENT

EXPLORING ORACLE DATABASE ARCHITECTURE

Overview of the Oracle database server architecture
Connecting to a database instance
Physical Structure
Presentation of the Oracle database memory structures
Automatic management of Shared Memory
Automatic management of the execution memory SQL code
Storage architecture and logical and physical structures of the database
Segments, extents, blocks and SYSTEM and SYSAUX tablespaces

INTRODUCTION TO SQL TUNING

Possible causes of poor performance of SQL statements
Performance Monitoring Solutions
Presentation of tools for monitoring and control
Setting dimensions of CPU time and wait time
Impact of the design, implementation and configuration of applications on scalability
Current errors appearing in client systems and proactive adjustment methodology
Simplicity in application design
Data modelling, design tables, design indexes, using views, efficient instruction execution

OVERVIEW OF THE OPTIMIZER

SQL language
Presentation of SQL statements analysis
Why use an optimizer?
Optimization during a complete analysis operation

Transformer and evaluator
Cost-based optimizer
Plan generator

INTERPRETING EXECUTION PLANS

What is an execution plan, where to find and view execution plans
PLAN_TABLE and AUTOTRACE
Using the V$SQLPLAN view
AWR (Automatic Workload Repository)
SQL Monitoring Overview
Interpreting an implementation plan
Reading more complex execution plans and reviewing the execution plan

TRACING AN APPLICATION

The end to end trace function of an application
Diagnosis trace location
Services: Overview - Using services with client applications and generating a trace at service level
Example of trace generation at session level
trcsess utility and SQL trace file content
Running the TKPROF utility and TKPROF command output
TKPROF output with and without index: Example

OPERATIONS ON TABLES AND THE INDEXES VIEWED BY THE OPTIMIZER

Operation on row sources, main structures and paths
Full table scan
Index presentation (B*-tree index and NULL value)
Using indexes: Consideration of columns allowing NULL values
Index-organized tables
Bitmap indexes, operations on bitmap indexes and bitmap join indexes
Composite indexes and invisible indexes
Rules for index management and to address cases of non-use of indexes

OPTIMIZER JOINING METHODS

Nested loop join
Nested loop join implemented from version 11g
Sort-Merge join type
Hash join and Cartesian join
Equijoins and non-equijoins
Outer joins
Semijoins
Antijoins

OTHER OPTIMIZER OPERATORS

When to use clusters
Sorting operators and operator SORT BUFFER
MIN/MAX and FIRST_ROW operators, and other degree N operations
Filtering and concatenation operations
UNION [ALL], INTERSECT and MINUS operations
RESULT CACHE operator

PROCESSING STAR CASES

Star schema model and flake schema model
Star transformation
Retrieving rows from the fact table for one dimension and all dimensions
Performing the join between the set of intermediate results and dimensions

Examples of a star transformation plan
Star transformation tips
Using bitmap join indexes
Bitmap join indexes: Join Models 1-4

STATISTICS FOR THE OPTIMIZER

Types of statistics for the optimizer
Statistics on tables, indexes and columns
Grouping factors in the index
Histograms, frequency histograms and comments on histograms
Introducing multi column statistics and statistics on expressions
Collecting system statistics and statistical preferences
Manual statistics collection
Locking, importing/exporting and defining statistics

USING BIND VARIABLES

Cursor sharing and different literal values
Cursor sharing and attached variables
Review of attached variables
Cursor sharing enhancements
CURSOR_SHARING parameter
Forcing cursor sharing
Adaptive Cursor Sharing functionality
Interacting with Adaptive Cursor Sharing functionality

USING THE STA ADVISORY FUNCTION

Automatic tuning of SQL statements
Challenges of setting an application
STA advisory function: Presentation
Outdated or missing statistics on objects and profiling SQL statements
Adjusting plan flow and SQL Profile Creation
SQL tuning loop, path analysis and SQL structure analysis
Implementing recommendations

USING THE SAA ADVISORY FUNCTION

Overview of the SAA advisory function
Possible recommendations

USING AUTOMATIC SQL TUNING

SQL control loop
Automatic tuning of SQL statements
Automatic setting process
Configuring the Automatic SQL Tuning task

SQL PERFORMANCE MANAGEMENT

Maintaining SQL code and SQL Plan Management performance: Overview
SQL Plan Baseline Architecture
Important attributes of SQL Plan Baselines
Selecting SQL Plans
Sample SQL Plan Management Scenarios
SQL Performance Analyzer and SQL Plan Baseline scenario
Automatically loading an SQL Plan Baseline and purge policy of the SQL Management Base (SMB)

PRACTICAL WORK