Course Outline

Oracle SQL Statement Tuning

Course Summary

Description

This course is designed to give students a foundation in SQL statement tuning. They are provided the necessary knowledge and skills to effectively tune SQL statements against the Oracle18c server. Course content is applicable to Oracle 12c, Oracle 18c & Oracle 19c. Students will focus on gaining an understanding of the behavior of the Oracle Cost-Based Optimizer (CBO) and how to accomplish their performance tuning goals. The students learn to use the Oracle diagnostic tools and facilities: EXPLAIN, AUTOTRACE and other tools. Gathering and utilizing object and system statistics is covered in detail. In addition, the participants also learn to influence the behavior of the CBO by hinting and modifying physical database objects.

Topics

- Oracle's SQL tool
- The Petsaver database
- Introduction to tuning
- SQL statement processing
- Optimizing SHARED_POOL usage
- Effective Indexing
- Using the EXPLAIN PLAN utility
- SQL*Plus tuning tools
- Collecting Statistics
- Using TKPROF
- Choosing the driving table in JOINS
- Using Hints to influence the Optimizer
- Optimizing sorts
- Advanced Indexes
- Optimizing PL/SQL

Audience

This course is designed for experienced Oracle developers and application development support DBAs

Prerequisite

Knowledge of SQL join syntax and 3 to 6 months of experience writing SQL in Oracle version 11g or higher are require for this course.

Duration

Three Days

Course Outline

Oracle SQL Statement Tuning

Course Outline

I. Oracle's SQL tool
A. SQL*Plus
B. SQL Developer

II. The Petsaver database
A. Understanding the sample database

III. Introduction to tuning
A. Kinds of performance problems
B. Methods to measure performance
C. Techniques to improve SQL performance

IV. SQL statement processing
A. Understanding SQL statement processing steps
B. The Oracle optimizer
C. The cost-based optimizer (CBO)
D. Version specific optimization (OPTIMIZER_FEATURES_ENABLE)

V. Optimizing SHARED_POOL usage
A. Identifying ways to minimize parsing
B. Using bind variables
C. Using PL/SQL packages

VI. Effective Indexing
A. Creating B*-Tree indexes
B. Utilizing "super" indexes and partial index utilization
C. Indexes in the data dictionary
D. Monitoring index usage

VII. Using the EXPLAIN PLAN utility
A. Creating a PLAN table
B. Using the EXPLAIN PLAN command
C. Interpreting EXPLAIN PLAN output
D. Understanding row access methods

VIII. SQL*Plus tuning tools
A. Using statement TIMING
B. Invoking the SQL Autorace Facility
C. Interpreting AUTOTRACE Statistics

IX. Collecting Statistics
A. Using the ANALYZE Command
B. Using the DBMS_STATS Package
C. Understanding histograms
D. Verifying histogram usage

X. Using TKPROF
A. Prerequisites for TKPROF
B. Formatting trace files with TKPROF
C. Interpreting TKPROF output

XI. Choosing the driving table in JOINS
A. Creating the query diagram
B. Rules to choose the driving table

XII. Using Hints to influence the Optimizer
A. Using Hints
B. Optimizer goal hints
C. Access method hints
D. Join hints
E. Other hints

XIII. Optimizing sorts
A. When does Oracle sort
B. Sorting efficiently

XIV. Advanced Indexes
A. Creating bitmap indexes
B. Creating function-based Indexes
C. Creating reverse key Indexes

XV. Optimizing PL/SQL
A. The RETURN clause
B. Using DBMS_PROFILER to measure PL/SQL execution
C. Using PL/SQL hinting
D. BULK operations
E. Dynamic SQL
F. Native PL/SQL compilation
G. Using DBMS_SHARED_POOL