

Advanced Oracle SQL Tuning

Duration: 3 Days · **Training format:** Virtual Classroom / Online

Description

This Advanced SQL tuning course is designed for practicing Oracle DBAs and Systems Administrators, professionals who have basic experience with SQL statements. Prior experience with Oracle is not required, but experience using SQL with a relational database is highly desirable.

The goal of the SQL tuning class is to provide a comprehensive toolkit to allow the DBA to quickly locate and tune a database workload.

By the end of this SQL tuning course the student will be able to tune advanced SQL queries including correlated subqueries and outer joins. The student will also become familiar with all of the major SQL tuning techniques for Oracle, including global parameter and statistics changes, adding missing indexes and adjusting optimizer statistics. The student will also see the internals of the Oracle optimizers, and see proven techniques for tuning Oracle SQL statements for optimal performance.

Outline

Introduction to SQL Tuning

- Intro to the class
- History of SQL
- Evolution of SQL
- Exercise - declarative SQL

Internal processing of SQL statements

- Parsing SQL Syntax
- SQL Semantic Analysis
- Generating the execution plan
- Using optimizer plan stability
- Using the v\$sql view
- Using the v\$sql_plan view
- Exercise - Query the library cache

Optimizer Statistics

- Purpose of statistics
- Types of statistics (table, column, system)

- Histogram statistics
- Dynamic sampling
- using dbms_stats
- Exporting/importing statistics
- Statistics management
- Exercise – gather system stats

Optimizer modes and goals

- Management issues with system-wide optimization
- Different modes of SQL optimization
- Bi-modal databases
- Rule-based optimization
- Cost-based optimization
- All rows optimization
- First_rows optimization
- Exercise – display and change optimizer_mode
- 1-5 - Table joining internals
- Sort-merge joins
- Nested Loop joins
- Hash joins
- STAR joins
- Bitmap joins
- Exercise – Change table join techniques & evaluate performance
- 1-6 – SQL Tuning and full-table scans
- Basics of file I/O
- Sequential reads vs. scattered reads
- When full scans are best
- RAM caching in the SGA
- Automating table caching
- Solid State Disks
- Tracking full-scans over time with AWR
- Exercise – Query v\$sql

Oracle parallel query and parallel DML

- Parallel and SMP processing
- Parallel query optimal degree
- Parallel query management (system, session, statement)
- Parallel DML
- Parallel parallelism

- Evaluating Large-table, full-table scans
- Index Usage Analysis
- Reports on system-wide SQL execution
- Exercise – run autotrace options
- 2-3 - Altering SQL execution plans
- Using hints
- Changing the system-wide optimizer mode
- Changing optimizer mode for specific statements
- Re-writing SQL queries
- Table join order evaluation
- Using the ordered hint
- Exercise – Optimizer costing models

Tuning SQL with hints

- Optimizer directives
- Scope of hints (session-level, statement level)
- Broad hints (optimizer mode) vs. narrow hints
- Good Hints vs. bad hints
- Using the ORDERED and LEADING HINTS
- Forcing index usage
- Using hints in subqueries

Oracle Index Optimization

- B-Tree indexes
- Bitmap Indexes
- Function-based Indexes
- Clustered indexes
- Index-only tables
- Exercise – Create a function-based index

Tuning Oracle sorting

- When a sort is invoked (order by, group by, etc.)
- Sorting with indexes vs internal sorting
- Detecting disk sorts
- Sorted Hash Clusters
- Exercise: Force two sort methods

Monitoring SQL Performance

- Measuring end-to-end response time

- Measuring SQL throughput
- Using v\$sqlsession_longops
- Optional Exercise – Run plan9i.sql

Oracle DML Tuning

- DML Tuning is not for neophytes
- Oracle DML tuning
- Optimizing Oracle SQL insert performance
- High Impact insert Tuning Techniques
- Tips for batch inserts
- Low-impact insert techniques (% and % faster)
- Tuning insert speed with the nologging option
- Reverse key indexes and insert performance
- Blocksize and insert performance
- Oracle Delete & Update Tuning
- High impact update techniques (over % faster):
- Low-impact techniques (between % and % faster)
- Using bulk binds for PL/SQL updates
- Oracle subquery factoring (with clause) for DML

Tuning with materialized views and temporary objects

- Materialized views
- Global temporary tables
- Using scalar subqueries (WITH clause)
- Simplifying complex SQL with temporary objects

Tuning subqueries

- Subquery Tuning and SQL
- Types of SQL Subqueries
- Tuning Guidelines for Subqueries
- Avoiding SQL Subqueries
- Subqueries in the where Clause
- In vs. exists Subqueries
- Same Results, Different Syntax and Plans
- Non-correlated subquery:
- Outer Join:
- Correlated Subquery:
- Tuning Scalar Subqueries
- Scalar Subquery Performance

- Removing Subqueries for Fast SQL Performance
- Internals of Temporary Tables
- Correlated vs. Non-correlated Subqueries
- Tuning Correlated Subqueries
- Automatic Rewriting not exists Subqueries
- Automatic Rewriting exists Subqueries
- Rewriting Non-equality Correlated Subqueries
- Rewriting exists Subqueries with the rank Function
- Subquery Hint Tuning
- Subquery Tuning with Index Hints
- Tuning Subqueries With the push_subq Hint
- Table Anti-Join Hints
- The merge_aj Hint
- SQL Tuning With the hash_aj Hint

Troubleshooting bad SQL

- Troubleshooting Problem SQL
- The Holistic Approach to SQL Tuning
- Troubleshooting Oracle SQL Bugs
- What is Bad SQL?
- Identifying Problem SQL
- Troubleshooting with v\$sql_plan
- SQL Troubleshooting with v\$sql_plan_statistics
- Finding indexing opportunities

Advanced Optimizer Statistics

- Histograms
- Exporting/importing statistics
- Statistics management

Prerequisites

There are no prerequisites