13

Workshop

Objectives

After completing this lesson, you should be able to do the following:

- Identify some common problems in writing SQL statement
- Determine the common causes and observations
- Understand the different tuning approaches
- Understand the behavior of the optimizer
- Understand the effects of functions on index usage
- Tune sort operations for `ORDER BY` clause
- Tune high parse time using parse time reduction strategy
- Maintain stable execution plans over time

Overview

- This lesson is made up of nine workshops based on the SQL tuning techniques covered in the previous lessons.
- All workshops are based on problem solving.
- You have to find ways to enhance performance through several tasks in each workshop.
- You also review concepts learned in each workshop through quizzes.
- Finally, by resolving the problems, you reinforce your learning.

Workshop 1

Introduction:

- In this workshop, you have to find a workaround to enhance performance.
- You analyze a poorly-written SQL statement and perform additional tasks such as creating a function-based index, redesigning a simple table, and rewriting the SQL statement.

Workshop 2

Introduction:

- This workshop is to review the *execution steps of a SQL statement* section.
- You review the SQL statement that uses a bind variable in the indexed column.
- You analyze two execution plans of the SQL statement using the `EXPLAIN PLAN` command and V$ view.
- You should be able to understand why the two execution plans are different for the same SQL statement.

Workshop 3

Introduction:

- This workshop shows the possible usage of an index in the `ORDER BY` clause to tune a sort operation.
- You perform several tasks to be able to use the indexed `ORDER BY` column.
- Once all tasks are done, you verify if the index always produces a better plan cost.

Workshop 4

Introduction:

- In this workshop, you assume that you have identified a poorly-running SQL statement.
- You already noticed that a root cause of the issue is the table design issue. However, recreating the table is not an option at this point. The table is already in use.
- Try to tune the SQL statement by rewriting it.

Workshop 5

Introduction:

- In this workshop, you write several SQL statements that return the same output to see which SQL statement is more efficient.
- You also examine the effects of changing column order in a composite index.

Workshop 6 & 7

Introduction:

- In these workshops, you review the execution plan and several sections in an event 10053 trace file.
- You interpret the information to understand the optimizer's decision.
- There are several questions on cost model, selectivity, and cardinality.
- Finally, you learn how to use the information in the 10053 file to tune the SQL statement.

Workshop 8

Introduction:

- You have identified the slow-running SQL statement. It was run 534 times in a certain time period. You noticed that a different literal was used in each execution. This caused the system to parse the same statement 534 times with the high parse time, which is not efficient.
- You perform several tasks to tune the high parse time.
- What tuning strategy could be used?

Workshop 9

Introduction:

- Oracle has introduced several methods to maintain the stable execution plans of SQL statements such as hints, stored outlines, and SQL profile. In Oracle 11*g*, we can use a better way to manage the execution plan called SQL Plan Baseline. This feature allows the optimizer to make the right decision in choosing a verified or tested execution plan in SQL Plan Baseline even if the optimizer generates a new plan during hard parse.
- In this workshop, you learn how to use this feature to associate a hinted execution plan with a hard-coded SQL statement.

Summary

In this lesson, you should have learned how to:

- Identify some common problems in writing SQL statement
- Determine the common causes and observations
- Understand the different tuning approaches
- Understand the behavior of the optimizer
- Understand the effects of functions on index usage
- Tune sort operations for `ORDER BY` clause
- Tune high parse time using parse time reduction strategy
- Maintain stable execution plans over time