

2

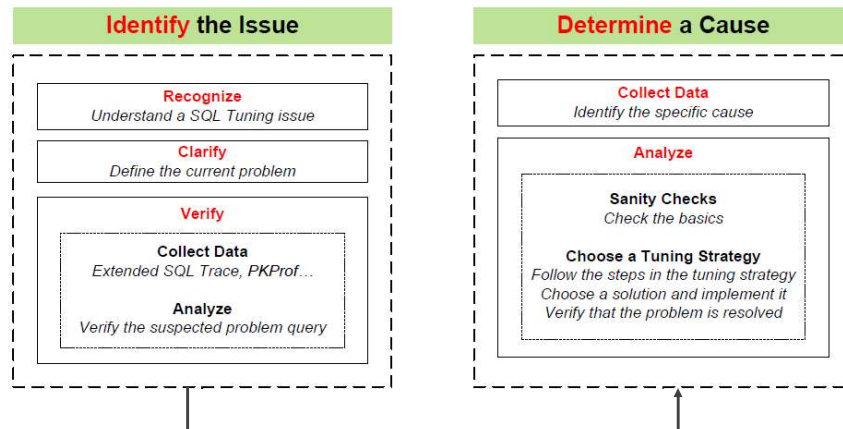
Introduction to SQL Tuning

Objectives

After completing this lesson, you should be able to:

- Describe SQL tuning
- Explain SQL tuning strategies
- Describe Oracle SQL Developer
- Describe `SQLTXPLAIN` (SQLT)

SQL Tuning Session



Recognize: What Is Bad SQL?

- Bad SQL uses more resources than necessary.
- Bad SQL has the following characteristics:
 - Excessive parse time
 - Excessive I/O (physical reads and writes)
 - Excessive CPU time
 - Excessive waits

Clarify: Understand the Current Issue

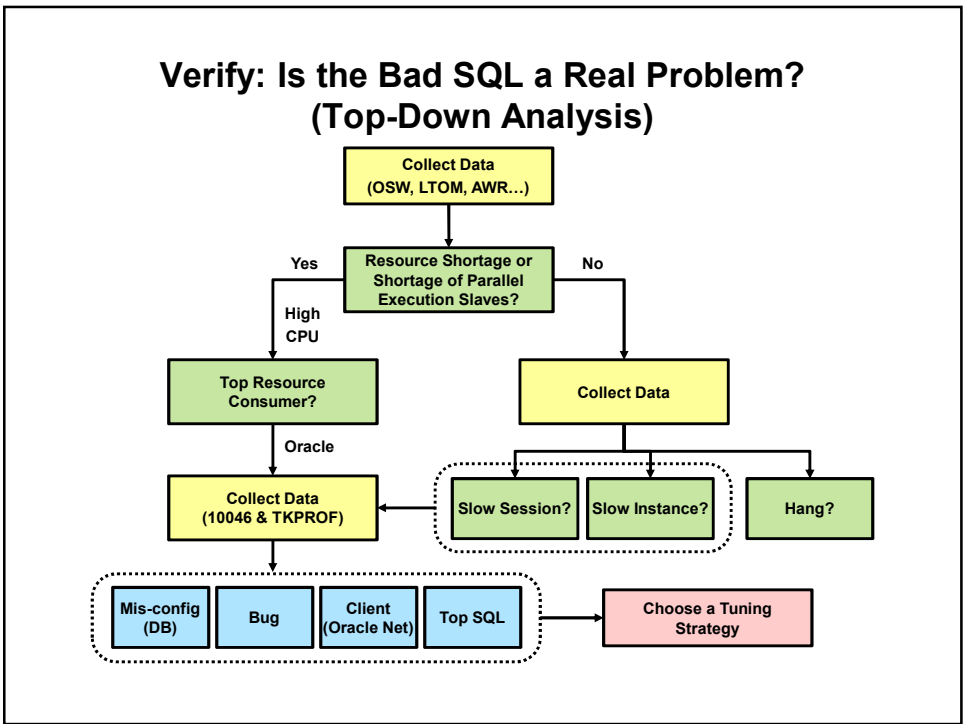
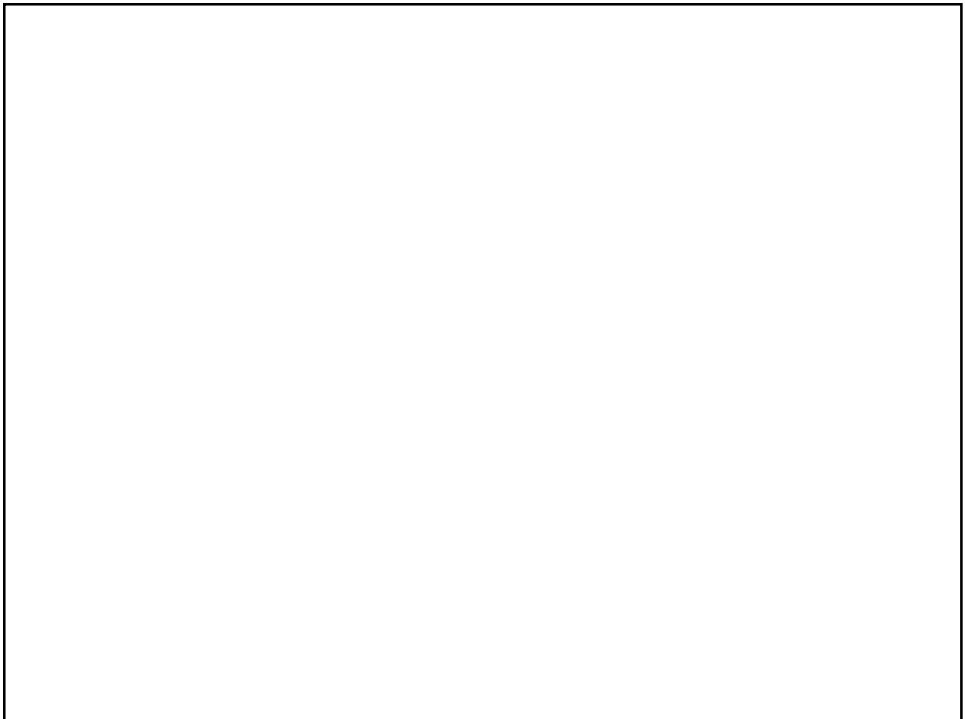
Changes that might trigger the issue:

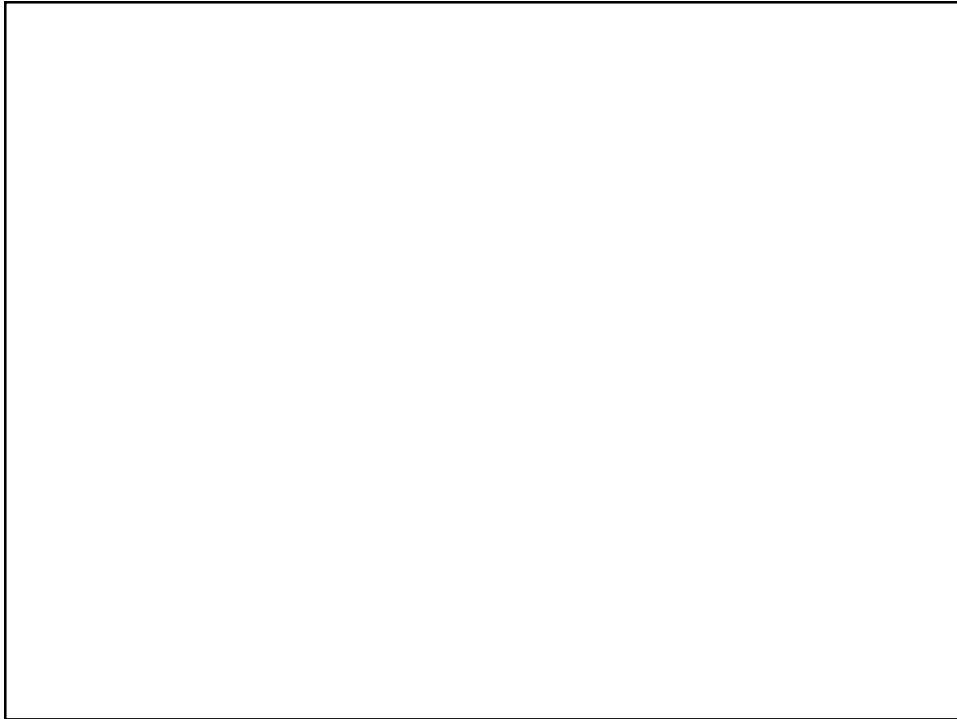
- Database upgraded
- Statistics gathered
- Schema changed
- Database parameter changed
- Application changed
- Operating system (OS) and hardware changed
- Data volume changed by more active users

Verify: Collect Data

These tools are available to identify bad SQL statements:

- Client Reports
- OS Statistics
- SQL Trace Facility
- Trace Analyzer (TRCANLZR)
- `SQLTXPLAIN` (SQLT)
- SQL Performance Analyzer (SPA)
- Automatic Workload Repository (AWR) Report
- Active Session History (ASH) Report
- Top SQL Report
- Automatic Database Diagnostic Monitor (ADDM)





SQL Tuning Strategies: Overview

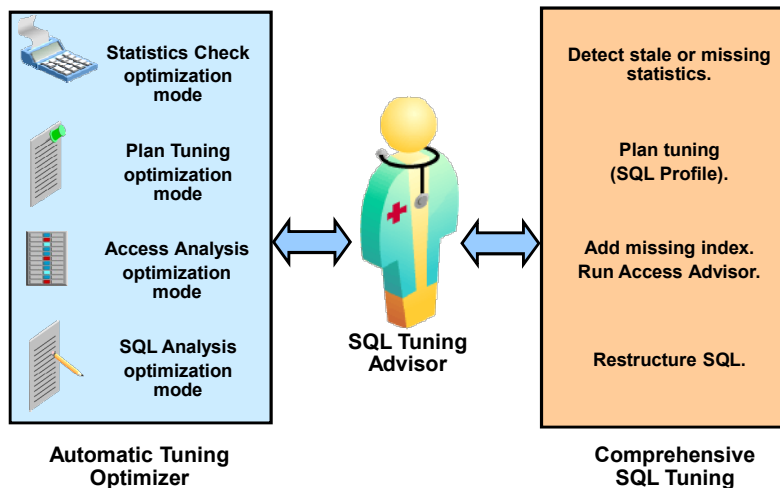
- Sanity Checks
- Advanced SQL tuning analysis
- Parse time reduction
- Plan comparison
- Quick solution
- Query analysis

Checking the Basics

Always check the basics first.

- Ensure that up-to-date statistics are collected properly.
- Ensure that reasonable initialization parameters are set.
- Ensure that the proper optimizer mode is set.
- Ensure that appropriate and valid hints are used.

Advanced SQL Tuning Analysis



Parse Time Reduction Strategy

- Example 1: Inefficient SQL statements

SELECT * FROM

call	count	cpu	elapsed	disk	query	current	rows
Parse	555	100.09	300.83	0	0	0	0
Execute	555	0.42	0.78	0	0	0	0
Fetch	555	14.04	85.03	513	1448514	0	11724
total	1665	114.55	386.65	513	1448514	0	11724

- Example 2: Cursor sharing
- Example 3: Connection management

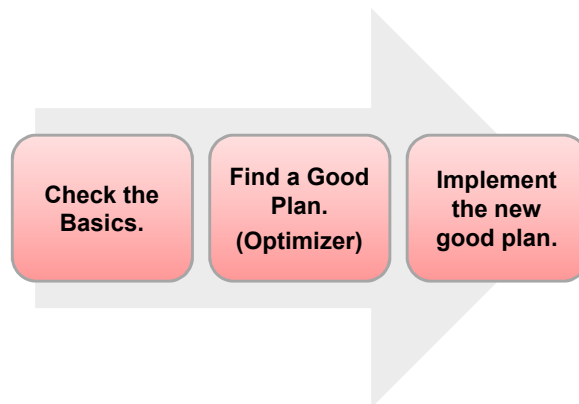
Plan Comparison Strategy

ID	Exec Ord	Operation	Time 1
0	11	SELECT STATEMENT	
1	10	SORT GROUP BY	
2	9	HASH JOIN	
3	1	TABLE ACCESS FULL PRODUCTS	
4	8	PARTITION RANGE ALL	
5	7	HASH JOIN	
6	4	TABLE ACCESS	
7	3	BITMAP CON	
8	2	BITMAP IN	
9	6	BITMAP CON	
10	5	BITMAP IN	

ID	Exec Ord	Operation	Time 2
0	16	SELECT STATEMENT	
1	15	HASH GROUP BY	
2	14	HASH JOIN	
3	7	JOIN FILTER CREATE :BF0000	
4	6	HASH JOIN	
5	1	TABLE ACCESS FULL PRODUCTS	
6	5	PARTITION RANGE ALL	
7	4	TABLE ACCESS BY LO	
8	3	BITMAP CONVERSION	
9	2	BITMAP INDEX SIN	
10	13	VIEW VW_GSF_3	
11	12	HASH GROUP BY	
12	11	JOIN FILTER USE :BF	
13	10	PARTITION RANGE AL	
14	9	BITMAP CONVERSION	
15	8	BITMAP INDEX FUL	

ID	Exec Ord	Operation	Time 3
0	16	SELECT STATEMENT	
1	15	HASH GROUP BY	
2	14	HASH JOIN	
3	7	JOIN FILTER CREATE :BF0000	
4	6	HASH JOIN	
5	1	TABLE ACCESS FULL PRODUCTS	
6	5	PARTITION RANGE ALL	
7	4	TABLE ACCESS BY LOCAL INDEX ROWID SALES	
8	3	BITMAP CONVERSION TO ROWIDS	
9	2	BITMAP INDEX SINGLE VALUE SALES_CUST_BIX	
10	13	VIEW VW_GSF_3	
11	12	HASH GROUP BY	
12	11	JOIN FILTER USE :BF0000	
13	10	PARTITION RANGE ALL	
14	9	BITMAP CONVERSION TO ROWIDS	
15	8	BITMAP INDEX FULL SCAN SALES_TIME_BIX	

Quick Solution Strategy



Find a Good Plan

- Use SQL Tuning Advisor (if you can).
- Get a good test case.
- Leverage the optimizer by changing settings to obtain the better plan (SESSION only):
 - `OPTIMIZER_MODE (FIRST_ROWS_N, ALL_ROWS)`
 - `OPTIMIZER_FEATURES_ENABLE`
 - `OPTIMIZER_INDEX_COST_ADJ`
 - `OPTIMIZER_INDEX_CACHING`
- Try dynamic sampling at high levels.
- Try appropriate hints.

Implement the New Good Plan

- Use a stored outline.
- Accept a profile from STA.
- Use a SQLT profile that was generated from a good plan.
- Use SQL Plan Management (11g feature).

Query Analysis Strategy: Overview

- Use this strategy when:
 - A good plan is not available
 - An urgent solution is not required
 - You want to determine an underlying cause
 - The query may be modified
- This strategy focuses on:
 - Statistics and parameters
 - SQL statement structure
 - Data access paths
 - Join orders and join methods
 - Other operations, such as parallelism or partition pruning

Query Analysis Strategy: Collecting Data

- Collect data for the query:
 - Execution plans
 - Information about each table or view
 - Object statistics and system statistics
 - Histograms
 - Parameter settings
- Tools available to collect data:
 - SQLT
 - Extended SQL trace and `tkprof`
 - `SPREPSQL.SQL`
 - `AWRSQRPT.SQL`
 - And more

Query Analysis Strategy: Examining SQL Statements

Look at the query for common mistakes:

- Understand the volume of resulting data.
- Ensure that no join predicates are missing.
- Look for unusual predicates.
- Look for constructs known to cause problems such as large `IN` list / `OR` statement, outer joins, hierarchy queries, views, inline views, and subqueries.

Query Analysis Strategy: Analyzing Execution Plans

Analyze the execution plan to find areas that need to be tuned:

- Examine data access paths, such as full table scan, index full scan, and index fast full scan.
- Examine join order and join types, such as nested loop join, hash join, and sort merge join.
- Review the actual number of rows and the estimated number of rows that are returned by the query.
- Look for plan steps where there is a large discrepancy between the actual and estimated rows.
- Look for plan steps where cost and logical read differ significantly.

Query Analysis Strategy: Finding Execution Plans

Use the following tools to find execution plans:

- V\$SQL_PLAN (Library Cache)
- V\$SQL_PLAN_MONITOR (11g)
- DBA_HIST_SQL_PLAN (AWR)
- SQL management base (SQL plan baselines)
- SQL tuning set
- Trace files generated by DBMS_MONITOR
- Event 10053 trace file
- SQLTXPLAIN report
- SQL Trace
- Extended SQL trace and tkprof
- SPREPSQL.SQL
- AWRSQRPT.SQL

Query Analysis Strategy: Reviewing Common Observations and Causes

Review common observations and causes:

- Poorly written SQL
- Index used / not used
- Lack of an index
- Wrong join order
- Wrong type
- Predicates not pushed, views not merged
- Transformation improperly costed
- Other problems

Query Analysis Strategy: Determining Solutions

Consider possible solutions:

- Gather statistics properly.
- Create a new index or re-create an existing index.
- Use the SQL Advisors with the Tuning Pack option.
- Use hints and stored outline to get the preferred plan.
- Use SQL Plan Management.
- Use dynamic sampling to obtain accurate selectivity estimates.
- Eliminate implicit data type conversions.
- Create a function-based index.
- Rewrite the query to permit the use of an existing index.
- Load the data in the key order.

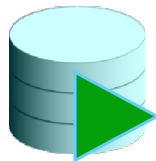
Query Analysis Strategy: Determining Solutions

Consider possible solutions:

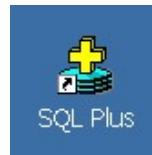
- Use an Index-Organized Table (IOT).
- Remove hints that influence the choice of index.
- Correct common problems with hints.
- Use the correct optimizer mode.
- Add the appropriate join predicate for the query.
- Review the intent of the query and ensure that a predicate is not missing.
- Use parallel execution / parallel data manipulation language (DML).
- Ensure that array processing is used.
- Use materialized views and query rewrite.

Development Environments: Overview

- Oracle SQL Developer
- Oracle SQL*Plus

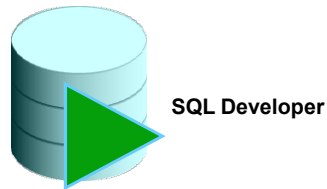


SQL Developer




What Is Oracle SQL Developer?

- Oracle SQL Developer is a free graphical tool that improves productivity and simplifies database development tasks.
- You can connect to any target Oracle database schema by using standard Oracle database authentication.
- You use Oracle SQL Developer in this course.
- Appendix C contains details for using Oracle SQL Developer.



Coding PL/SQL in Oracle SQL*Plus

A small blue square icon with a white terminal symbol (a rectangle with a cross) and the word 'Terminal' written below it.

```
oracle@EDRSR13P1:~/demo
File Edit View Terminal Tabs Help
[oracle@EDRSR13P1-orcl ~]$ pwd
/home/oracle
[oracle@EDRSR13P1-orcl ~]$ cd demo
[oracle@EDRSR13P1-orcl demo]$ sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Mon May 7 07:06:14 2012
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter user-name: sqlt
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> set serveroutput on
SQL> create or replace procedure hello is
2 begin
3 dbms_output.put_line('Hello Class!');
4 end;
5 /

Procedure created.

SQL> execute hello
Hello Class!

PL/SQL procedure successfully completed.

SQL>
```

SQLTXPLAIN (SQLT) Diagnostic Tool

Global	Plans	Tables
<ul style="list-style-type: none">• Observations• SQL Text• SQL Identification• Environment• CBO Environment• Fix Control• CBO System Statistics• DBMS_STATS Setup• Initialization Parameters• Tool Configuration Parameters	<ul style="list-style-type: none">• Summary• Performance Statistics• Execution Plans	<ul style="list-style-type: none">• Tables• Statistics• Statistics Versions• Properties• Physical Properties• Constraints• Columns• Indexed Columns• Histograms• Partitions• Indexes
Cursor Sharing and Binds	Plan Control	Objects
<ul style="list-style-type: none">• Cursor Sharing• Peeked Binds• Captured Binds	<ul style="list-style-type: none">• Stored Outlines• SQL Profiles• SQL Plan Baselines	<ul style="list-style-type: none">• Objects• Dependencies• Fixed Objects• Fixed Object Columns• Tablespaces• Metadata
SQL Execution	SQL Tuning Advisor	
<ul style="list-style-type: none">• Segment Statistics• Session Statistics• Session Events	<ul style="list-style-type: none">• STA Report• STA Script	
	SQL	
	<ul style="list-style-type: none">• SQL Statistics• Monitor Statistics• Monitor HTML Report• Monitor TEXT Report	

Quiz

Although views provide clean programming interfaces, they should be used carefully because they can cause suboptimal, resource-intensive queries when nested too deeply.

- a. True
- b. False

Quiz

Identify the characteristics that must be supported by an application that is designed for SQL execution efficiency.

- a. Use concurrent connections to the database.
- b. Use cursors so that SQL statements are parsed once and executed multiple times.
- c. For data-warehousing queries, cursor sharing is always important to get the best plan.

Summary

In this lesson, you should have learned how to:

- Describe SQL tuning
- Explain SQL tuning strategies
- Describe Oracle SQL Developer
- Describe `SQLTXPLAIN` (SQLT)

Practice 2: Overview

This practice covers using SQL Developer.