When dealing with long-running SQL queries in Oracle 19c, several useful commands and tools can help you monitor and diagnose performance issues. Here’s a list of key commands:

1. View Current Sessions

To see currently running queries and their status:

SELECT \* FROM v$session WHERE status = 'ACTIVE';

2. Find Long-Running Queries

To identify queries that are running for a long time:

SELECT s.sid, s.serial#, s.status, q.sql\_id, q.seconds\_in\_wait, q.wait\_time, q.cpu\_time FROM v$session s

JOIN v$sql q ON s.sql\_id = q.sql\_id

WHERE s.status = 'ACTIVE' AND q.seconds\_in\_wait > 60; -- Adjust the threshold as needed

3. Check Execution Plan

To see the execution plan of a specific SQL query:

EXPLAIN PLAN FOR your\_query;

SELECT \* FROM table(dbms\_xplan.display);

4. View SQL Statistics

To get detailed statistics about a specific SQL statement:

SELECT \* FROM v$sql WHERE sql\_id = 'your\_sql\_id';

5. Active Session History (ASH)

To view active session history and see what sessions are doing:

SELECT \* FROM v$active\_session\_history WHERE session\_id = :sid; -- Replace :sid with the session ID

6. Automatic Workload Repository (AWR) Report

Generate an AWR report for a detailed analysis of performance over time:

-- You can use the following to generate an AWR report for specific time intervals

-- Make sure to adjust the time range accordingly

SELECT \* FROM table(dbms\_workload\_repository.awr\_report\_html(:dbid, :inst\_id, :begin\_snap\_id, :end\_snap\_id));

7. SQL Monitoring

If you're using SQL Monitoring, you can check the status of long-running queries:

SELECT \* FROM v$sql\_monitor WHERE sql\_id = 'your\_sql\_id';

8. Check for Blocking Sessions

To find if a session is being blocked:

SELECT a.sid, a.serial#, b.sid AS blocking\_sid

FROM v$session a

JOIN v$session b ON a.blocking\_session = b.sid

WHERE a.blocking\_session IS NOT NULL;

9. Session Wait Events

To identify what a session is waiting for:

SELECT event, wait\_time, seconds\_in\_wait

FROM v$session

WHERE sid = :sid; -- Replace :sid with the session ID

10. Kill a Session

If you need to terminate a long-running session:

ALTER SYSTEM KILL SESSION 'sid,serial#'; -- Replace with the appropriate values

11. Gather Statistics

To gather fresh statistics for better query optimization:

EXEC DBMS\_STATS.GATHER\_TABLE\_STATS('schema\_name', 'table\_name');

12. Monitor Resource Usage

To check CPU and I/O usage for active sessions:

SELECT \* FROM v$session\_longops WHERE totalwork > 0;

**13. SQL Execution History**

To see the execution history of SQL statements:

SELECT sql\_id, child\_number, executions, elapsed\_time, cpu\_time, disk\_reads, buffer\_gets

FROM v$sql

ORDER BY elapsed\_time DESC;

**14. Detailed SQL Execution Stats**

To get detailed execution statistics for a specific SQL statement

SELECT \* FROM v$sql\_plan WHERE sql\_id = 'your\_sql\_id';

**15. Find Wait Events**

To analyze the wait events for a specific session:

SELECT event, wait\_time, time\_waited

FROM v$session\_wait

WHERE sid = :sid; -- Replace :sid with the session ID

**16. Identify Resource Intensive Queries**

To identify SQL statements consuming high resources:

SELECT sql\_id, cpu\_time, elapsed\_time, disk\_reads

FROM v$sql

ORDER BY cpu\_time DESC;

**17. View Locks**

To check for locks that might be causing delays:

SELECT \* FROM v$locked\_object;

**18. Monitoring Resource Usage by SQL ID**

To check resource usage for a specific SQL statement:

SELECT \* FROM v$sql\_monitor

WHERE sql\_id = 'your\_sql\_id';

**19. Historical Session Activity**

To see historical session activity:

SELECT \* FROM DBA\_HIST\_ACTIVE\_SESS\_HISTORY

WHERE session\_id = :sid; -- Replace :sid with the session ID

**20. View Session Wait Events Over Time**

To see wait events for a specific session over time:

SELECT \* FROM v$session\_longops WHERE sid = :sid; -- Replace :sid with the session ID

**21. Analyze Index Usage**

To find out if indexes are being used effectively:

SELECT index\_name, column\_name, key\_columns

FROM user\_ind\_columns

WHERE table\_name = 'your\_table\_name'; -- Replace with your table name

**22. Check for Unused Indexes**

To identify indexes that may not be useful:

SELECT index\_name, table\_name, num\_rows

FROM user\_indexes

WHERE status = 'UNUSABLE';

**23. Check for Bind Variable Peeking**

To analyze if bind variable peeking is affecting performance:

SELECT sql\_id, child\_number, last\_active\_time

FROM v$sql

WHERE sql\_text LIKE '%:variable\_name%'; -- Replace with your variable

**24. Use SQL Tuning Advisor**

To get tuning recommendations for a specific SQL statement:

EXEC DBMS\_SQLTUNE.CREATE\_TUNING\_TASK(sql\_id => 'your\_sql\_id');

EXEC DBMS\_SQLTUNE.EXECUTE\_TUNING\_TASK(task\_name => 'your\_task\_name');

EXEC DBMS\_SQLTUNE.REPORT\_TUNING\_TASK(task\_name => 'your\_task\_name');

**25. View System Resource Usage**

To check system-wide resource usage:

SELECT \* FROM v$system\_event;

**26. Query Resource Manager Plan**

To see if a Resource Manager plan is affecting query performance:

SELECT plan\_name, active

FROM dba\_rsrc\_plan;

**27. Identify Skewed Data**

To check for skewed data in tables that could be causing performance issues:

SELECT column\_name, num\_distinct, avg\_col\_len

FROM user\_tab\_columns

WHERE table\_name = 'your\_table\_name'; -- Replace with your table name

**28. View Session Resource Consumption**

To see resource consumption per session:

SELECT sid, serial#, cpu\_time, elapsed\_time FROM v$session WHERE status = 'ACTIVE';

**29. Identify and Resolve Fragmentation**

To check for fragmentation in tables:

SELECT table\_name, partition\_name, round((blocks \* 8) / 1024, 2) AS size\_mb

FROM user\_segments WHERE segment\_type = 'TABLE' ORDER BY size\_mb DESC;

**30. Monitor CPU and Memory Usage**

To monitor overall CPU and memory usage of the database:

SELECT \* FROM v$osstat;

31. Check SQL Execution Time

To find out the execution time of the most recent SQL statement:

SELECT sql\_id, elapsed\_time, cpu\_time, disk\_reads, buffer\_gets

FROM v$sql

ORDER BY last\_active\_time DESC;

32. View Active SQL Statements

To see currently active SQL statements:

SELECT sql\_id, sql\_text FROM v$sql

WHERE sql\_id IN (SELECT sql\_id FROM v$session WHERE status = 'ACTIVE');

33. Session Resource Limits

To check resource limits applied to a specific session:

SELECT sid, cpu\_per\_session, cpu\_per\_call, logical\_reads\_per\_session, logical\_reads\_per\_call

FROM v$resource\_limit;

34. Identify High Load SQL Statements

To find SQL statements that contribute significantly to load:

SELECT sql\_id, executions, elapsed\_time, cpu\_time, buffer\_gets

FROM v$sql ORDER BY elapsed\_time DESC;

35. Analyze Wait Event History

To see historical wait events for sessions:

SELECT \* FROM DBA\_HIST\_WAITCLASS;

36. View Query Performance with SQL Monitoring

To monitor a specific SQL statement in detail:

SELECT \* FROM v$sql\_monitor WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

37. Check for Resource Contention

To identify contention on a specific resource:

SELECT event, wait\_time, time\_waited

FROM v$system\_event;

38. Session Stats for Performance

To get session statistics that help in diagnosing issues:

SELECT \* FROM v$sesstat WHERE sid = :sid; -- Replace with your session ID

39. Monitor I/O Statistics

To monitor I/O statistics for the database:

SELECT \* FROM v$filestat;

40. Identify Unused Tables and Indexes

To identify tables and indexes that may no longer be needed:

SELECT table\_name, num\_rows FROM user\_tables WHERE num\_rows = 0;

41. Session History and Activity

To track historical session activity:

SELECT \* FROM DBA\_HIST\_ACTIVE\_SESS\_HISTORY

WHERE session\_id = :sid; -- Replace with your session ID

42. Identify Long-Running Queries

To find queries running longer than a specific threshold:

SELECT sql\_id, elapsed\_time, sql\_text FROM v$sql

WHERE elapsed\_time > 60000; -- Queries running longer than 60 seconds

43. Memory Usage Analysis

To analyze memory usage in the database:

SELECT name, value FROM v$parameter WHERE name LIKE '%memory%';

44. Check for SQL Plan Baselines

To view existing SQL plan baselines that might help in execution:

SELECT \* FROM dba\_sql\_plan\_baselines;

45. SQL Profiles

To see SQL profiles that might improve execution:

SELECT \* FROM dba\_sql\_profiles;

46. Analyze Index Fragmentation

To check index fragmentation that might slow down performance:

SELECT index\_name, blevel, leaf\_blocks, distinct\_keys, num\_rows

FROM user\_indexes WHERE table\_name = 'your\_table\_name'; -- Replace with your table name

47. Real-Time SQL Monitoring

To enable real-time monitoring for a running query:

SELECT sql\_id, sql\_text FROM v$sql\_monitor WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

48. Identify SQL Execution Plans in Cache

To find execution plans for cached SQL:

SELECT \* FROM v$sql\_plan WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

49. Session Blocking Information

To check which sessions are blocking others:

SELECT b.sid AS blocking\_sid, a.sid AS blocked\_sid FROM v$session a

JOIN v$session b ON a.blocking\_session = b.sid;

50. Database Alerts and Logs

To check alert logs for any database errors or warnings that might affect performance:

SHOW PARAMETER background\_dump\_dest; -- Find the alert log location

51. Identify Large Segments

To find large segments consuming space:

SELECT segment\_name, segment\_type, bytes / (1024 \* 1024) AS size\_mb

FROM dba\_segments ORDER BY bytes DESC;

52. Check Table and Index Sizes

To check the size of tables and their indexes:

SELECT table\_name, SUM(bytes) / (1024 \* 1024) AS size\_mb FROM dba\_segments

WHERE segment\_type IN ('TABLE', 'INDEX') GROUP BY table\_name ORDER BY size\_mb DESC;

53. Review Data Skewness

To analyze data distribution which might affect performance:

SELECT column\_name, num\_distinct, density FROM user\_tab\_col\_statistics

WHERE table\_name = 'your\_table\_name'; -- Replace with your table name

54. Tuning Advisor Reports

To generate a tuning report for a specific SQL statement:

EXEC DBMS\_SQLTUNE.CREATE\_TUNING\_TASK(sql\_id => 'your\_sql\_id');

EXEC DBMS\_SQLTUNE.EXECUTE\_TUNING\_TASK(task\_name => 'your\_task\_name');

SELECT DBMS\_SQLTUNE.REPORT\_TUNING\_TASK('your\_task\_name') FROM dual;

55. Review Table Locks

To see if any tables are being locked:

SELECT object\_name, object\_type FROM dba\_objects WHERE object\_id IN (SELECT object\_id FROM v$locked\_object);

56. Check System Resource Usage

To monitor overall system performance metrics:

SELECT \* FROM v$system\_usage;

57. Profile Active Sessions

To get a profile of active sessions including wait events:

SELECT a.sid, a.serial#, a.status, b.event, b.wait\_time, b.seconds\_in\_wait

FROM v$session a JOIN v$session\_wait b ON a.sid = b.sid WHERE a.status = 'ACTIVE';

58. View Database Wait Events

To see wait events affecting the database:

SELECT event, total\_waits, time\_waited, average\_wait FROM v$system\_event ORDER BY time\_waited DESC;

59. Check Parallel Execution Settings

To view settings related to parallel execution:

SELECT \* FROM v$parameter WHERE name LIKE '%parallel%';

60. Identify Top CPU Consuming SQL Statements

To find the top SQL statements consuming CPU:

SELECT sql\_id, cpu\_time, elapsed\_time, buffer\_gets FROM v$sql ORDER BY cpu\_time DESC

FETCH FIRST 10 ROWS ONLY;

61. Session History for Performance Monitoring

To track historical performance metrics for sessions:

SELECT \* FROM DBA\_HIST\_ACTIVE\_SESS\_HISTORY WHERE session\_id = :sid; -- Replace with the session ID

62. Monitor Wait Classes

To analyze wait classes affecting performance:

SELECT wait\_class, total\_waits, time\_waited FROM v$waitclass;

63. Analyze SQL Execution with Real-Time Monitoring

To enable real-time monitoring for a specific SQL statement:

SELECT sql\_id, sql\_text FROM v$sql\_monitor WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

64. Check for Missing Indexes

To find SQL statements that may benefit from missing indexes:

SELECT \* FROM dba\_sql\_tuning\_advisor WHERE recommendation\_type = 'INDEX';

65. Identify SQL Statements with High Buffer Gets

To find SQL statements with high buffer gets that could indicate performance issues:

SELECT sql\_id, buffer\_gets FROM v$sql ORDER BY buffer\_gets DESC FETCH FIRST 10 ROWS ONLY;

66. Gather Session Statistics

To gather statistics for a specific session:

SELECT \* FROM v$sessstat WHERE sid = :sid; -- Replace with the session ID

67. Monitor Temporary Tablespace Usage

To check usage of temporary tablespaces:

SELECT tablespace\_name, SUM(bytes\_used) / (1024 \* 1024) AS used\_mb FROM v$tempseg\_usage

GROUP BY tablespace\_name;

68. Track SQL Statement Performance Over Time

To analyze performance trends for specific SQL statements:

SELECT sql\_id, executions, elapsed\_time, cpu\_time, buffer\_gets FROM dba\_hist\_sqlstat

WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

69. Examine Query Execution Plans for Changes

To check if execution plans have changed for a specific SQL:

SELECT \* FROM dba\_hist\_sql\_plan WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

70. Analyze Blocking Sessions

To find sessions that are blocking others:

SELECT b.sid AS blocking\_sid, a.sid AS blocked\_sid FROM v$session a

JOIN v$session b ON a.blocking\_session = b.sid;

71. Resource Allocation by SQL ID

To view resource allocation for a specific SQL ID:

SELECT sql\_id, cpu\_time, elapsed\_time, disk\_reads, buffer\_gets

FROM v$sql WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

72. View Tablespace Usage

To check how much space each tablespace is using:

SELECT tablespace\_name, SUM(bytes) / (1024 \* 1024) AS size\_mb FROM dba\_data\_files GROUP BY tablespace\_name;

73. Analyze Schema for Unused Objects

To identify unused objects within a schema:

SELECT object\_name, object\_type FROM user\_objects WHERE status = 'VALID'

AND last\_ddl\_time < SYSDATE - 365; -- Objects not modified in the last year

74. Check for SQL Execution Errors

To find SQL statements that encountered errors:

SELECT \* FROM v$sql

WHERE parsing\_schema\_name = 'your\_schema\_name' AND sql\_text LIKE '%ERROR%'; -- Replace with your schema

75. Resource Manager Status

To check the status of the Oracle Resource Manager:

SELECT \* FROM dba\_rsrc\_plan;

76. Monitoring Memory Allocation

To monitor memory allocation for the instance:

SELECT \* FROM v$memory\_target;

77. Examine System Wait Statistics

To see overall system wait statistics:

SELECT \* FROM v$system\_wait\_stats;

78. Identify Query Execution Variability

To check variability in execution times for a SQL statement:

SELECT sql\_id, executions, avg\_elapsed\_time, min\_elapsed\_time, max\_elapsed\_time

FROM dba\_hist\_sqlstat WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID

79. Table Compression Analysis

To analyze table compression that might improve performance:

SELECT table\_name, compression FROM user\_tables WHERE compression IS NOT NULL;

80. Execution Plan Evolution

To monitor changes in execution plans over time:

SELECT sql\_id, plan\_hash\_value, execution\_count, last\_active\_time FROM dba\_hist\_sql\_plan

WHERE sql\_id = 'your\_sql\_id'; -- Replace with your SQL ID