Oracle Performance Tuning Interview Questions & Answers

Level 4 / Senior DBA Edition

This PDF is designed for senior Oracle DBA interviews. It focuses on production troubleshooting, AWR/ASH interpretation, wait events, SQL tuning, RAC performance, memory, I/O, TEMP/UNDO, and scenario-based answers.

1. Core Performance Tuning Concepts

1. What is Oracle performance tuning?

Oracle performance tuning is the process of identifying and resolving database performance problems by analyzing SQL execution plans, wait events, AWR and ASH reports, memory usage, I/O performance, CPU usage, locking, optimizer statistics, and application workload patterns. A senior DBA should not tune blindly. First identify the bottleneck, then apply the fix.

2. What is the basic performance tuning approach?

```
1. Understand the issue.
2. Confirm the scope: database-wide or SQL-specific.
3. Check the exact time window of the issue.
4. Review AWR and ASH for that period.
5. Identify top wait events and top SQL.
6. Check execution plans and row estimates.
7. Validate statistics, indexes, CPU, memory, and I/O.
8. Apply the least risky fix.
9. Compare before and after metrics.
```

3. What is the difference between database tuning and SQL tuning?

Database tuning	SQL tuning
Focuses on whole database performance	Focuses on individual SQL
Checks CPU, memory, I/O, waits	Checks execution plan, joins, indexes
Uses AWR, ADDM, ASH	Uses SQL Monitor, DBMS_XPLAN, SQL Trace
Example: high DB CPU	Example: one query doing full table scan

4. What are the main tools used for Oracle performance tuning?

Tool	Purpose
AWR	Historical workload report
ASH	Session-level active history
ADDM	Oracle performance recommendations
SQL Monitor	Real-time SQL execution monitoring
SQL Trace / TKPROF	Detailed SQL execution statistics
DBMS_XPLAN	Execution plan display
OEM	GUI monitoring
V$ views	Real-time troubleshooting

2. AWR Interview Questions

5. What is AWR?

AWR means Automatic Workload Repository. It stores database performance statistics such as wait events, top SQL, DB time, CPU usage, I/O statistics, instance activity, segment statistics, and memory statistics. AWR is mainly used to analyze historical database performance issues.

6. What is DB Time?

DB Time is the total time spent by foreground sessions inside the database.

```
DB Time = DB CPU + Non-idle wait time
```

If DB Time is very high compared to elapsed time, it means many active sessions were consuming database resources.

7. What is DB CPU?

DB CPU is the time spent by Oracle foreground sessions actively using CPU. If DB CPU is the top event, it usually indicates SQL consuming too much CPU, too many logical reads, inefficient execution plans, high parsing, excessive PL/SQL processing, or application workload increase.

8. What is the difference between DB Time and elapsed time?

Elapsed time is wall-clock time. DB Time is cumulative database session time. For example, if elapsed time is 1 hour and average active sessions are 10, DB Time is about 10 hours.

9. How do you read an AWR report?

Check snapshot interval, DB Time, elapsed time, load profile, instance efficiency, top timed events, host CPU, top SQL by elapsed time, CPU, buffer gets and physical reads, wait event details, segment statistics, instance activity statistics, and advisory sections. The most important sections are usually Top Timed Events and SQL ordered by Elapsed Time / CPU / Gets / Reads.

10. What are Top Timed Events in AWR?

Top Timed Events show where database time was spent.

Wait event	Meaning
DB CPU	CPU usage
db file sequential read	Single block read, often index lookup
db file scattered read	Multiblock read, often full table scan
log file sync	Commit wait
log file parallel write	Redo write wait
enq: TX - row lock contention	Row locking
library cache lock	Shared pool/library cache issue
gc buffer busy acquire	RAC block contention
direct path read	Direct reads, often parallel/full scan

3. ASH Interview Questions

11. What is ASH?

ASH means Active Session History. It samples active sessions every second and stores SQL ID, session ID, wait event, blocking session, module, program, object, machine, and execution plan line. ASH is useful when the issue happened in the past but is not happening now.

12. Difference between AWR and ASH?

AWR	ASH
Aggregated historical report	Session-level sampled data
Good for overall workload	Good for specific session/SQL/wait analysis
Snapshot-based	Sample-based
Usually every 1 hour	Every second active session sampling

AWR	ASH
Used for trend and summary	Used for deep drill-down

13. How do you find what a session was waiting for yesterday?

```
SELECT sample_time,
       session_id,
       session_serial#,
       sql_id,
       event,
       blocking_session,
       wait_class
FROM dba_hist_active_sess_history
WHERE sample_time BETWEEN TO_DATE('2026-06-13 10:00','YYYY-MM-DD HH24:MI')
                      AND TO_DATE('2026-06-13 11:00','YYYY-MM-DD HH24:MI')
ORDER BY sample_time;
```

4. Wait Event Questions

14. What is a wait event?

A wait event shows what a database session is waiting for, such as disk read, row lock, redo write, CPU, RAC block transfer, latch, or mutex. Performance tuning is mostly about finding the biggest wait and reducing it.

15. What is db file sequential read?

It means Oracle is reading single blocks from disk. Common causes are index range scans, unique scans, rowid table access, nested loop joins, poor clustering factor, missing data in buffer cache, or slow storage. It is not always bad, but it becomes a problem when average wait time or number of waits is very high.

16. How do you troubleshoot high db file sequential read?

Check SQL causing the wait, execution plan, index usage, buffer gets, physical reads, object statistics, clustering factor, storage latency, and cache behavior.

```
SELECT sql_id, event, COUNT(*)
FROM v$active_session_history
WHERE event = 'db file sequential read'
GROUP BY sql_id, event
ORDER BY COUNT(*) DESC;
```

17. What is db file scattered read?

It means Oracle is doing multiblock reads. It is commonly seen during full table scans, fast full index scans, large reports, missing indexes, parallel query, or poor execution plans.

18. Difference between sequential read and scattered read?

Event	Meaning	Common operation
db file sequential read	Single block read	Index scan
db file scattered read	Multiblock read	Full table scan

19. What is log file sync?

log file sync occurs when a user session commits and waits for LGWR to flush redo to disk. High log file sync usually indicates commit performance issues. Causes include frequent commits, slow redo storage, LGWR bottleneck, high concurrent commits, Data Guard synchronous transport, or CPU starvation.

20. How do you troubleshoot high log file sync?

Check average wait time for log file sync, log file parallel write, redo disk latency, commit frequency, application commit pattern, LGWR CPU usage, Data Guard sync mode, redo log size, and switching frequency.

```
SELECT event, total_waits, time_waited_micro/1000000 seconds_waited
FROM v$system_event
WHERE event IN ('log file sync','log file parallel write');
```

If log file parallel write is also high, storage may be the issue. If log file sync is high but log file parallel write is low, check CPU scheduling, commit frequency, or LGWR contention.

21. What is log file parallel write?

It is the wait event for LGWR writing redo entries to redo log files. If high, it indicates redo write latency. Fixes include moving redo logs to faster storage, avoiding overloaded disks, reducing unnecessary redo generation, and checking ASM/storage/multipathing.

22. What is enq: TX - row lock contention?

It occurs when a session waits for a row locked by another transaction. Causes include uncommitted transactions, application not committing, batch jobs locking rows, foreign key without index, duplicate key conflict, or hot rows.

23. How do you find blocking sessions?

```
SELECT s.sid,
       s.serial#,
       s.username,
       s.blocking_session,
       s.event,
       s.sql_id,
       s.seconds_in_wait
FROM v$session s
WHERE s.blocking_session IS NOT NULL;
```

24. What is library cache lock?

It occurs when sessions wait to access objects in the library cache. Causes include DDL on active objects, package recompilation during business hours, invalid objects, hard parsing, shared pool pressure, or object dependency locks.

25. What is cursor: pin S wait on X?

This is a mutex wait related to cursor sharing and parsing. Causes include high hard parsing, many sessions parsing the same SQL, bind variable issues, frequent invalidation, shared pool pressure, or version-specific bugs. Actions include using bind variables, reducing hard parse, reviewing child cursors, checking v$sql_shared_cursor, and applying patches if needed.

5. SQL Tuning Questions

26. How do you tune a slow SQL query?

```
1. Get SQL_ID.
2. Check execution plan.
3. Compare actual rows vs estimated rows.
4. Check wait events.
5. Check buffer gets and physical reads.
6. Verify optimizer statistics.
7. Check indexes and predicates.
8. Check joins and join order.
9. Check bind variable peeking/adaptive plans.
10. Test rewrite, index, baseline, profile, or hint if needed.
11. Compare before and after metrics.
```

27. How do you find SQL execution plan?

For cached SQL:

```
SELECT *
FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR('sql_id_here', NULL, 'ALLSTATS LAST +PEEKED_BINDS'));
```

For historical SQL:

```
SELECT *
FROM TABLE(DBMS_XPLAN.DISPLAY_AWR('sql_id_here'));
```

For production tuning, DISPLAY_CURSOR with actual statistics is more useful than plain EXPLAIN PLAN.

28. Difference between explain plan and actual execution plan?

Explain Plan	Actual Execution Plan
Estimated plan before execution	Real plan used during execution
Does not show actual rows	Can show actual rows
May differ due to binds/adaptive plans	More reliable
Uses EXPLAIN PLAN	Uses DBMS_XPLAN.DISPLAY_CURSOR

29. What is cardinality?

Cardinality is the optimizer's estimated number of rows returned from an operation. Bad cardinality estimates can cause bad execution plans, such as a nested loop join chosen for millions of rows.

30. What causes wrong cardinality estimates?

Stale statistics, missing histograms, skewed data, function-based predicates, correlated columns, bind variable peeking, complex predicates, missing extended statistics, and insufficient dynamic sampling.

31. What is selectivity?

Selectivity means how many rows a predicate returns compared to total rows. High selectivity usually benefits from an index. Low selectivity may be better with a full scan.

32. What is a full table scan? Is it always bad?

A full table scan reads all or most blocks of a table. It is not always bad. It can be good when a query returns a large percentage of the table, the table is small, parallel query is used, or index access would require too many table lookups.

33. What is index range scan?

Index range scan is used when Oracle scans a range of values from an index, such as salary between two values or hire_date greater than a date.

34. What is index unique scan?

Index unique scan occurs when Oracle uses a unique index to fetch one row, usually by primary key or unique key.

35. What is index fast full scan?

Index fast full scan reads the entire index using multiblock reads. It may happen when all required columns are available in the index.

36. What is clustering factor?

Clustering factor tells how ordered table rows are compared to the index order. A low clustering factor is good. A high clustering factor means table rows are scattered, causing more table block visits during index access.

37. Why is my index not being used?

Common reasons include query returns large percentage of rows, function on indexed column, data type mismatch, stale statistics, low selectivity, optimizer chooses full scan as cheaper, invisible/unusable index, predicate not matching leading

column, or bind variable peeking issue.

```
-- Bad for normal index on order_date
WHERE TO_CHAR(order_date,'YYYY') = '2026'

-- Better
WHERE order_date >= DATE '2026-01-01'
AND   order_date <  DATE '2027-01-01'
```

38. What is a function-based index?

A function-based index is created on an expression.

```
CREATE INDEX emp_upper_name_idx ON emp (UPPER(ename));
```

Useful for queries such as:

```
WHERE UPPER(ename) = 'SMITH';
```

39. What is composite index?

A composite index is an index on multiple columns. Column order matters and should be based on filtering, join predicates, and workload.

```
CREATE INDEX orders_idx1 ON orders (customer_id, order_date, status);
```

40. What is the leading column rule?

For a normal B-tree composite index, Oracle is most efficient when predicates use the leading column. Oracle may still use index skip scan in some cases, but it is not always efficient.

6. Join Method Questions

41. What are Oracle join methods?

Join method	Best for
Nested Loop Join	Small result set, indexed lookup
Hash Join	Large data sets
Sort Merge Join	Sorted data, inequality joins
Cartesian Join	Usually bad unless intentional

42. What is nested loop join?

Nested loop join reads rows from one table and probes another table repeatedly. It is good when the outer table returns few rows and the inner table has a good index. It is bad when the outer table returns millions of rows.

43. What is hash join?

Hash join builds a hash table from one input and probes it with another input. It is good for large joins and data warehouse queries. It needs enough PGA memory, otherwise it may spill to TEMP.

44. What is sort merge join?

Sort merge join sorts both data sets and merges them. It is useful when inputs are already sorted, inequality joins are used, or hash join is not suitable.

45. How do you decide if join method is wrong?

Check actual rows versus estimated rows. If the optimizer estimated 10 rows but actual rows are 5 million and selected nested loop, a hash join may be better.

```
SELECT *
FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR('sql_id', NULL, 'ALLSTATS LAST'));
```

7. Optimizer Statistics Questions

46. What are optimizer statistics?

Optimizer statistics describe table, index, and column data distribution. They include number of rows, blocks, distinct values, density, histograms, clustering factor, and partition statistics. The optimizer uses these to choose execution plans.

47. How do you gather table statistics?

```
BEGIN
  DBMS_STATS.GATHER_TABLE_STATS(
    ownname => 'HR',
    tabname => 'EMPLOYEES',
    cascade => TRUE
  );
END;
/
```

48. How do you gather schema statistics?

```
BEGIN
  DBMS_STATS.GATHER_SCHEMA_STATS(
    ownname => 'HR',
    cascade => TRUE
  );
END;
/
```

49. What is histogram?

A histogram stores data distribution details for a column. It helps when data is skewed, such as STATUS where ACTIVE is 95%, CLOSED is 4%, and FAILED is 1%.

50. When are histograms useful?

Histograms are useful when data is skewed, the column is used in WHERE clauses, different values return very different row counts, and the optimizer needs better selectivity estimates. They are not useful for every column and can cause plan instability if overused.

51. What are stale statistics?

Statistics are stale when table data changed significantly after stats were gathered. Oracle tracks modifications in DBA_TAB_MODIFICATIONS.

```
SELECT table_name, inserts, updates, deletes, timestamp
FROM dba_tab_modifications
WHERE table_owner = 'HR';
```

52. How do you check last analyzed date?

```
SELECT owner, table_name, last_analyzed, num_rows
FROM dba_tables
WHERE owner = 'HR'
AND table_name = 'EMPLOYEES';
```

53. What are extended statistics?

Extended statistics help the optimizer understand relationships between columns, such as STATE and CITY.

```
SELECT DBMS_STATS.CREATE_EXTENDED_STATS(
  ownname   => 'HR',
  tabname   => 'CUSTOMERS',
```

```
  extension => '(STATE, CITY)'
)
FROM dual;
```

8. Execution Plan Questions

54. What do you check in an execution plan?

Access path, join method, join order, estimated rows, actual rows, cost, predicate information, partition pruning, parallel execution, TEMP usage, index usage, and adaptive plan notes.

55. What are access predicates and filter predicates?

Predicate type	Meaning
Access Predicate	Used to access data through index/table access
Filter Predicate	Applied after rows are fetched

Access predicates are usually more efficient.

56. What is cost in execution plan?

Cost is an optimizer estimate of resource usage. It is not time in seconds. It is based on CPU, I/O, cardinality, object statistics, and system statistics. A lower-cost plan is not always faster if estimates are wrong.

57. What is partition pruning?

Partition pruning means Oracle scans only required partitions instead of the whole table.

```
WHERE sales_date >= DATE '2026-06-01'
AND   sales_date <  DATE '2026-07-01'
```

58. What is bind peeking?

Bind peeking means optimizer looks at bind variable value during hard parse and creates a plan based on that value. If later bind values have very different selectivity, the plan may be bad.

59. What is adaptive cursor sharing?

Adaptive cursor sharing allows Oracle to create multiple child cursors for the same SQL when different bind values need different execution plans.

60. What is SQL Plan Baseline?

SQL Plan Baseline preserves accepted execution plans and prevents bad plan changes. It helps with plan stability when SQL suddenly starts using a bad plan.

9. SQL Trace and TKPROF Questions

61. What is SQL Trace?

SQL Trace captures detailed execution information for SQL statements: parse count, execute count, fetch count, CPU time, elapsed time, logical reads, physical reads, wait events, and bind values if enabled.

62. How do you enable trace for a session?

```
BEGIN
  DBMS_MONITOR.SESSION_TRACE_ENABLE(
```

```
    session_id => 123,
    serial_num => 456,
    waits      => TRUE,
    binds      => TRUE
  );
END;
/
```

Disable:

```
BEGIN
  DBMS_MONITOR.SESSION_TRACE_DISABLE(
    session_id => 123,
    serial_num => 456
  );
END;
/
```

63. What is TKPROF?

TKPROF formats raw SQL trace files into readable reports.

```
tkprof input.trc output.prf sort=exeela,fchela
```

64. What do you check in TKPROF?

SQL elapsed time, CPU time, parse/execute/fetch counts, consistent gets, physical reads, rows processed, wait events, execution plan, and bind values if captured.

10. Memory Tuning Questions

65. What are the main Oracle memory areas?

Memory area	Purpose
SGA	Shared memory for database instance
Buffer Cache	Caches data blocks
Shared Pool	SQL, PL/SQL, dictionary cache
Redo Log Buffer	Redo entries before LGWR writes
Large Pool	RMAN, parallel query, shared server
Java Pool	Java execution
PGA	Private memory for sessions
TEMP	Used when PGA is insufficient for sort/hash

66. What is SGA?

SGA is shared memory used by all database sessions. It contains database buffer cache, shared pool, redo log buffer, large pool, Java pool, and Streams pool.

67. What is PGA?

PGA is private memory allocated to server processes. It is used for sorting, hash joins, bitmap operations, session memory, and cursor state. If PGA is insufficient, operations spill to TEMP.

68. How do you check PGA issues?

```
SELECT name, value
FROM v$pgastat;

SELECT name, value
FROM v$sysstat
```

```
WHERE name LIKE 'workarea executions%';
```

High multipass workarea executions indicate insufficient PGA or very large operations.

69. What is shared pool tuning?

Shared pool tuning focuses on reducing hard parsing, library cache waits, cursor invalidations, shared pool fragmentation, and excessive literal SQL. Fixes include bind variables, reducing hard parse, avoiding frequent DDL, and checking child cursor issues.

70. What causes high hard parsing?

SQL without bind variables, literal values, dynamic SQL, frequent invalidations, small shared pool, different optimizer environments, and cursor sharing issues.

```
SELECT sql_text, parse_calls, executions
FROM v$sql
WHERE parse_calls > 1000
ORDER BY parse_calls DESC;
```

11. I/O Tuning Questions

71. How do you identify I/O bottleneck?

Check AWR wait events, file I/O statistics, ASM disk latency, OS tools, storage metrics, and top SQL by physical reads.

```
SELECT file#, singleblkrds, singleblkrdtim, phyblkrd
FROM v$filestat;
```

72. What is average read latency?

Average read latency is the average time taken to complete disk read. High latency indicates slow or overloaded storage.

73. How do you reduce physical reads?

Tune SQL, add proper indexes, improve execution plans, increase buffer cache only if needed, partition large tables, avoid unnecessary full scans, use materialized views for reports, archive/purge old data, and compress historical data where suitable.

12. TEMP Tuning Questions

74. What causes high TEMP usage?

Large sorts, hash joins spilling to disk, global temporary tables, index creation, parallel query, large ORDER BY, large GROUP BY, poor SQL plans, and insufficient PGA.

75. How do you find sessions using TEMP?

```
SELECT s.sid,
       s.serial#,
       s.username,
       u.tablespace,
       u.blocks * t.block_size / 1024 / 1024 AS temp_mb,
       s.sql_id
FROM v$tempseg_usage u
JOIN v$session s ON u.session_addr = s.saddr
JOIN dba_tablespaces t ON u.tablespace = t.tablespace_name
ORDER BY temp_mb DESC;
```

76. What will you do if TEMP is full?

Identify top TEMP sessions, check SQL_ID, confirm business impact, kill runaway session only if approved, add tempfile if needed, and tune SQL permanently.

```
ALTER TABLESPACE temp ADD TEMPFILE '+DATA' SIZE 20G AUTOEXTEND ON NEXT 1G MAXSIZE 100G;
```

13. Undo Tuning Questions

77. What causes ORA-01555 snapshot too old?

Long-running query, undo overwritten before query completes, small undo tablespace, very high DML activity, frequent commits in batch job, or poorly tuned query running too long.

78. How do you fix ORA-01555?

Tune the long query, increase undo tablespace, increase undo_retention, avoid frequent commits, schedule reports during low DML periods, and use flashback/materialized snapshot if needed.

79. How do you check undo usage?

```
SELECT begin_time,
       end_time,
       undoblks,
       txncount,
       maxquerylen,
       ssolderrcnt
FROM v$undostat
ORDER BY begin_time DESC;
```

14. Locking and Blocking Questions

80. How do you troubleshoot blocking sessions?

Identify blocked sessions, identify blocker, check SQL of blocker and waiter, check transaction start time, check locked object, contact application owner, and kill blocker only after approval.

```
SELECT blocking_session,
       sid,
       serial#,
       wait_class,
       event,
       seconds_in_wait
FROM v$session
WHERE blocking_session IS NOT NULL;
```

81. How do you find locked objects?

```
SELECT s.sid,
       s.serial#,
       s.username,
       o.owner,
       o.object_name,
       o.object_type,
       l.locked_mode
FROM v$locked_object l
JOIN dba_objects o ON l.object_id = o.object_id
JOIN v$session s ON l.session_id = s.sid;
```

82. What is deadlock?

Deadlock occurs when two or more sessions wait for each other's locked resources. Oracle automatically detects deadlocks and rolls back one statement. Error: ORA-00060. Deadlock details are written to a trace file.

15. RAC Performance Questions

83. What are common RAC wait events?

RAC wait event	Meaning
gc cr request	Consistent read block request
gc current request	Current block request
gc buffer busy acquire	Block busy in cluster
gc buffer busy release	Waiting for block transfer
gc cr block busy	CR block contention
gc current block busy	Current block contention
gc block lost	Interconnect/block issue

84. What is Global Cache Service?

In RAC, Global Cache Service manages block sharing between instances. If one instance needs a block in another instance's buffer cache, the block is transferred through the interconnect instead of reading from disk.

85. What causes high gc buffer busy acquire?

Hot blocks, multiple instances modifying same blocks, poor application partitioning, sequence/index right-hand growth, frequent updates to same rows, insert contention, and inefficient SQL touching same objects.

86. How do you troubleshoot RAC performance issue?

Check global AWR, per-instance AWR, top RAC waits, interconnect latency, hot blocks/segments, SQL causing global cache waits, instance affinity, services placement, and cross-instance blocking.

```
SELECT inst_id, event, total_waits, time_waited
FROM gv$system_event
WHERE event LIKE 'gc%'
ORDER BY time_waited DESC;
```

87. How do you reduce RAC contention?

Use service-based workload separation, partition application workload, reduce hot block contention, use reverse key index or scalable sequences where suitable, partition large tables/indexes, tune SQL, and avoid unnecessary cross-instance updates.

16. Parallel Query and Partitioning

88. What is parallel query?

Parallel query divides a SQL operation among multiple parallel execution servers. It is useful for large table scans, data warehouse queries, index creation, large aggregations, and data loading. It is not always suitable for OLTP.

89. What are risks of parallel query?

High CPU usage, high TEMP usage, more I/O pressure, resource starvation, too many parallel servers, and poor performance if used for small queries.

90. How do you check parallel execution?

```
SELECT sid,
       qcsid,
       qcserial#,
```

```
       degree,
       req_degree
FROM v$px_session;
```

91. How does partitioning improve performance?

Partitioning improves performance through partition pruning, faster maintenance, partition-wise joins, local indexes, and better manageability for large tables.

92. What is local index and global index?

Local Index	Global Index
Partitioned same as table	Independent index structure
Easier partition maintenance	Can be better for unique/global access
Less maintenance overhead	Can become unusable after partition operations
Good for data warehouse	Good for OLTP lookup patterns

93. What is partition pruning issue?

Partition pruning fails when predicate does not match partition key properly.

```
-- Bad
WHERE TO_CHAR(order_date,'YYYY-MM') = '2026-06'

-- Good
WHERE order_date >= DATE '2026-06-01'
AND   order_date <  DATE '2026-07-01'
```

17. Real Production Scenario Questions

Scenario 1: Database is slow. What will you check first?

Clarify what is slow, since when, all users or specific module, recent changes, and whether it is happening now. Then check active sessions by wait event, CPU, top sessions, blocking, top SQL, and AWR/ASH for the issue time.

```
SELECT event, COUNT(*)
FROM v$session
WHERE status = 'ACTIVE'
GROUP BY event
ORDER BY COUNT(*) DESC;
```

Scenario 2: CPU usage is 95%. What will you do?

Check whether CPU is consumed by Oracle or OS process. Inside DB, identify SQL on CPU and top SQL by CPU time.

```
SELECT sql_id, COUNT(*)
FROM v$active_session_history
WHERE session_state = 'ON CPU'
GROUP BY sql_id
ORDER BY COUNT(*) DESC;
```

Possible causes include bad SQL plan, high logical reads, excessive parsing, workload spike, parallel query overload, PL/SQL loops, or statistics changes.

Scenario 3: A SQL was running in 5 seconds yesterday, today 30 minutes. What will you check?

Check SQL_ID, current execution plan, previous plan from AWR, statistics changes, object changes, index status, bind values, cardinality estimates, plan hash value, and SQL baseline/profile availability.

Scenario 4: Application is hanging. What will you check?

First check blocking, then active sessions. Possible causes include locks, deadlocks, connection pool exhaustion, slow SQL, log file sync, CPU starvation, database link wait, or network issue.

```
SELECT sid, serial#, blocking_session, event, seconds_in_wait, sql_id
FROM v$session
WHERE blocking_session IS NOT NULL;
```

Scenario 5: Insert performance is slow. What can be the reasons?

Too many indexes, triggers, foreign key checks, sequence contention, redo generation, undo pressure, ITL waits, block contention, storage latency, commit frequency, and RAC hot block contention.

Scenario 6: Commit is slow. What will you check?

Commit slowness usually appears as log file sync. Check log file sync average wait, log file parallel write, redo storage latency, commit frequency, Data Guard sync transport, LGWR CPU scheduling, and redo size per second.

Scenario 7: Query is using full table scan. Should you create an index?

Not immediately. Check table size, rows returned, predicate selectivity, existing indexes, column statistics, functions on columns, execution frequency, and DML overhead of the new index. Full scan can be best when many rows are returned.

Scenario 8: TEMP tablespace is growing rapidly. What will you do?

Find top TEMP consumers, review SQL plan, tune joins/sorts, increase PGA if needed, add indexes, rewrite SQL, or add TEMP space temporarily.

Scenario 9: AWR shows high buffer busy waits. What does it mean?

Sessions are waiting for access to a block in buffer cache. Causes include hot blocks, concurrent inserts, index block contention, segment header contention, freelist issues in old tablespaces, or ITL waits.

```
SELECT owner, object_name, statistic_name, value
FROM v$segment_statistics
WHERE statistic_name LIKE '%buffer busy%'
ORDER BY value DESC;
```

Scenario 10: High parse time elapsed. What will you check?

Check hard parse count, SQL using literals, shared pool size, library cache waits, child cursor count, bind variable usage, and invalidations.

```
SELECT name, value
FROM v$sysstat
WHERE name IN ('parse count (hard)', 'parse count (total)');
```

18. Advanced Level 4 Questions

94. What is SQL Profile?

SQL Profile provides additional optimizer information to improve cardinality estimates. It does not force a fixed plan like a baseline. It helps the optimizer choose a better plan.

95. Difference between SQL Profile and SQL Plan Baseline?

SQL Profile	SQL Plan Baseline
Improves optimizer estimates	Controls accepted plans
Does not force exact plan	Helps preserve plan stability
Created by SQL Tuning Advisor	Created/captured by SPM

SQL Profile	SQL Plan Baseline
Good for bad estimates	Good for plan regression

96. What is SQL Patch?

SQL Patch can inject hints into a SQL statement without changing application code. It is useful when application SQL cannot be modified or an emergency production hint is required.

97. What is adaptive plan?

Adaptive plan allows Oracle to change parts of the execution plan during runtime based on actual row counts, such as switching join methods.

98. What are dynamic sampling and dynamic statistics?

Dynamic sampling allows the optimizer to sample data during parse time when statistics are missing or insufficient. It helps improve estimates for complex queries or missing stats.

99. What is result cache?

Result cache stores query results in memory. It is useful for repeated queries on mostly static data, but not good for frequently changing tables.

100. What is In-Memory Column Store?

Oracle In-Memory stores data in columnar format in memory and improves analytic queries by scanning compressed columnar data faster. It is useful for reporting and analytics, not always necessary for OLTP.

19. Level 4 DBA Case Study Answers

Case Study 1: AWR shows DB CPU as top event

DB CPU means sessions are actively using CPU. Check host CPU utilization, top SQL by CPU time, top SQL by buffer gets, execution plans, hard parse count, SQL doing excessive logical reads, PL/SQL procedures, and parallel queries.

```
SELECT sql_id,
       executions,
       cpu_time/1000000 cpu_seconds,
       buffer_gets,
       rows_processed
FROM v$sql
ORDER BY cpu_time DESC
FETCH FIRST 10 ROWS ONLY;
```

Possible fixes include tuning high CPU SQL, adding/changing indexes, fixing execution plans, reducing logical reads, using bind variables, limiting unnecessary parallelism, and scaling CPU only after SQL tuning.

Case Study 2: AWR shows high log file sync

log file sync means sessions are waiting for commit confirmation. Check average wait time, log file parallel write, commit count, user commits per second, redo write latency, Data Guard sync configuration, and LGWR performance. If log file parallel write is high, storage is likely. If commits are extremely high, application may be committing too frequently.

Case Study 3: AWR shows high db file sequential read

This means many single-block reads. Check SQL ordered by physical reads, SQL ordered by elapsed time, execution plans, index access paths, table access by rowid, clustering factor, and storage latency. Causes include bad nested loop plan, index lookup returning many rows, poor clustering factor, cold cache, or slow storage.

Case Study 4: Query plan changed after stats gather

Compare old and new plan hash value, check stats history, histograms created/dropped, cardinality estimates, restore previous stats if urgent, create SQL Plan Baseline if old plan was better, and regather stats with proper method_opt if needed.

```
BEGIN
  DBMS_STATS.RESTORE_TABLE_STATS(
    ownname => 'HR',
    tabname => 'EMPLOYEES',
    as_of_timestamp => SYSTIMESTAMP - INTERVAL '1' DAY
  );
END;
/
```

Case Study 5: RAC database has high interconnect waits

Check which GC wait event is high, interconnect latency, hot blocks/segments, SQL causing global cache waits, services distribution, and whether same objects are modified from multiple instances. Fixes include service affinity, workload partitioning, tuning hot SQL, reducing cross-instance block transfers, partitioning hot tables/indexes, and checking network/interconnect health.

20. Important Queries for Level 4 DBA

Active sessions by wait event

```
SELECT event, COUNT(*)
FROM v$session
WHERE status = 'ACTIVE'
GROUP BY event
ORDER BY COUNT(*) DESC;
```

Top SQL by elapsed time

```
SELECT sql_id,
       executions,
       elapsed_time/1000000 elapsed_sec,
       cpu_time/1000000 cpu_sec,
       buffer_gets,
       disk_reads
FROM v$sql
ORDER BY elapsed_time DESC
FETCH FIRST 10 ROWS ONLY;
```

Top SQL by buffer gets

```
SELECT sql_id,
       executions,
       buffer_gets,
       disk_reads,
       rows_processed
FROM v$sql
ORDER BY buffer_gets DESC
FETCH FIRST 10 ROWS ONLY;
```

Get SQL text

```
SELECT sql_fulltext
FROM v$sql
WHERE sql_id = 'your_sql_id';
```

Get execution plan

```
SELECT *
FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR('your_sql_id', NULL, 'ALLSTATS LAST +PEEKED_BINDS'));
```

Blocking sessions

```
SELECT sid,
       serial#,
       username,
       blocking_session,
       event,
       sql_id,
       seconds_in_wait
FROM v$session
WHERE blocking_session IS NOT NULL;
```

Kill session

```
ALTER SYSTEM KILL SESSION 'sid,serial#' IMMEDIATE;
```

Find object from file and block

```
SELECT owner, segment_name, segment_type
FROM dba_extents
WHERE file_id = &file_id
AND &block_id BETWEEN block_id AND block_id + blocks - 1;
```

21. Quick Interview One-Liners

Question	Strong answer
What is DB Time?	DB CPU plus non-idle wait time
What is AWR used for?	Historical workload and performance analysis
What is ASH used for?	Session-level active performance analysis
What is SQL tuning?	Reducing resource usage and improving SQL execution time
What is cardinality?	Estimated number of rows returned
What causes bad plans?	Wrong stats, skew, bind peeking, missing indexes
What is log file sync?	Commit wait for redo flush
What is sequential read?	Single-block read, often index access
What is scattered read?	Multiblock read, often full table scan
What is SQL baseline?	Accepted plan control mechanism
What is SQL profile?	Optimizer correction information
What is clustering factor?	Index order compared with table row order
What is bind peeking?	Optimizer peeks bind value during hard parse
What is hard parse?	Full SQL parse requiring optimization
What causes TEMP usage?	Sorts, hash joins, parallel query, large aggregations
What causes ORA-01555?	Undo overwritten before query completes
RAC GC waits?	Inter-instance block transfer waits

22. Most Expected Level 4 Scenario Questions

1. Database is slow for all users. How do you troubleshoot?
2. One SQL suddenly became slow. How do you find root cause?
3. AWR shows high DB CPU. What is your action plan?
4. AWR shows high log file sync. What does it mean?
5. AWR shows high db file sequential read. How do you analyze?

6. TEMP tablespace is full. What do you do?
7. Undo tablespace is full. What do you do?
8. Application is hanging due to locks. How do you resolve?
9. RAC database has high GC waits. How do you troubleshoot?
10. SQL plan changed after stats gather. How do you fix?
11. Index is not being used. Why?
12. Query is doing full table scan. Is it bad?
13. CPU is 100% on DB server. How do you identify culprit?
14. Commit performance is slow. What do you check?
15. Batch job is taking longer than usual. What do you analyze?
16. Data warehouse query consuming huge TEMP. How do you tune?
17. High library cache waits. What is the cause?
18. High hard parsing. How do you fix?
19. Parallel query is slowing down database. What do you do?
20. How do you prove performance improvement after tuning?

23. Best Final Interview Answer Pattern

For senior DBA interviews, answer every performance question like this:

```
1. I will not assume the issue.
2. I will identify whether it is SQL, database, OS, storage, or application.
3. I will check AWR/ASH for the exact issue window.
4. I will identify top wait events and top SQL.
5. I will compare current and previous execution plans if SQL-specific.
6. I will validate statistics, indexes, and row estimates.
7. I will apply the least risky fix first.
8. I will verify improvement using before/after metrics.
```

A strong Level 4 DBA answer is always based on evidence, wait events, SQL ID, execution plan, and measurable improvement.