

Oracle

1Z0-084

Oracle Database 19c: Performance Management and Tuning

- Up to Date products, reliable and verified.
- Questions and Answers in PDF Format.

Full Version Features:

- 90 Days Free Updates
- 30 Days Money Back Guarantee
- Instant Download Once Purchased
- 24 Hours Live Chat Support

For More Information:

<https://www.testsexpert.com/>

- Product Version

Latest Version: 6.0

Question: 1

You must produce a consolidated formatted trace file by combining all trace files generated by all clients for a single service.

Which combination of utilities does this?

- A. Trace Analyzer and Trcsess
- B. Trcsess and TKPROF
- C. Autotrace and TKPROF
- D. TKPROF and Trace Analyzer

Answer: B

Explanation:

To produce a consolidated formatted trace file from multiple trace files generated by all clients for a single service, the combination of trcsess and TKPROF utilities is used. The trcsess utility consolidates trace files based on specified criteria such as session, client identifier, or service name. This results in a single trace file that combines the desired tracing information. Next, TKPROF is used to format the output of the trace file generated by trcsess, providing a readable summary of the trace, including execution counts, execution times, and SQL statement text along with execution plans.

Steps:

Use trcsess to combine trace files:

Command: `trcsess output=consolidated.trc service=your_service_name *.trc`

Use TKPROF to format the consolidated trace file:

Command: `tkprof consolidated.trc output.txt explain=user/password sys=no sort=prsela,fchela`

Reference:

Oracle Database Performance Tuning Guide, 19c

Oracle Database Utilities, 19c

Question: 2

Which two statements are true about cursor sharing?

- A. Setting `Cursor_sharing` to `FORCE` can result in a plan that is suboptimal for the majority of values bound to a bind variable when executing a cursor with one or more bind variables.
- B. Adaptive Cursor Sharing guarantees that a suboptimal plan will never be used on any execution of a SQL statement.
- C. Setting `optimizer_capture_sql_plan_baselines` to `TRUE` loads all adaptive plans for the same statement into the cursor cache.
- D. Setting `cursor_sharing` to `EXACT` prevents Adaptive Cursor Sharing from being used.
- E. Adaptive Cursor Sharing requires histograms on filtered columns, used in equality predicates, to allow

different execution plans to be generated for statements whose bound values would normally generate different plans at hard parse time.

Answer: AD

Explanation:

A) When `Cursor_sharing` is set to `FORCE`, Oracle tries to avoid hard parses by replacing literals in SQL statements with bind variables, even if the original statement didn't include bind variables. This can lead to the use of a single execution plan for multiple executions of a statement with different literal values, which might not be optimal for all executions.

D) Setting `cursor_sharing` to `EXACT` ensures that SQL statements must match exactly for them to share a cursor. This setting prevents the use of Adaptive Cursor Sharing (ACS) since ACS relies on the ability to share cursors among similar statements that differ only in their literal values. With `EXACT`, there's no cursor sharing for statements with different literals, hence no opportunity for ACS to operate.

Reference:

Oracle Database SQL Tuning Guide, 19c

Oracle Database Reference, 19c

Question: 3

Examine this statement and output:

```
SQL> select sid,
           state,
           wait_time/100 as "WAIT TIME IN SECONDS",
           event
       from v$session
       where username='OE';
```

SID	STATE	WAIT TIME IN SECONDS	EVENT
9857	WAITED KNOWN TIME	1354	rdbms ipc message
9822	WAITING	0	enq: TX - row lock contention
8779	WAITING	0	SQL*Net message from client

Which three statements are true?

- A. Both 9822 and 8779 sessions are waiting for operating system resources.
- B. Session 8779 may be waiting due to a network problem.
- C. Session 9857 waited 1354 seconds for another process, which was also waiting for a transaction to end.
- D. Session 9857 is not waiting.
- E. Session 8779 may be waiting for a user or application response.
- F. Session 9822 will always stop waiting if the session that owns the TX enqueue issues a COMMIT statement as session 9822 is the first session in the transaction queue.

Answer: BEF

Explanation:

For this SQL statement and output, we can analyze the EVENT column to understand the type of wait:

B) The event "SQL*Net message from client" typically indicates that the session is waiting for a response from the client. This can be due to a network issue, user response, or an application processing delay.

E) The event "SQL*Net message from client" also implies that the session is idle waiting for the client (a user or an application) to send a request to the server. This event usually indicates that the session is not actively working but is instead waiting for the next command.

F) The wait event "enq: TX - row lock contention" suggests that session 9822 is waiting for a row-level lock held by another session. If the holding session issues a COMMIT or ROLLBACK, the lock will be released, and session 9822 will stop waiting. Since this session is experiencing row lock contention, it implies it's waiting for a specific transaction to complete.

Reference:

Oracle Database Reference, 19c

Oracle Wait Events Documentation

Question: 4

Which two statements are true about session wait information contained in v\$session or v\$session_wait?

- A. Rows for sessions displaying WAITED UNKNOWN TIME in the STATE column indicate that the session is still waiting.
- B. Rows for sessions that are currently waiting have a wait time of 0.
- C. Rows for sessions that are not waiting might contain the actual wait time for the last event for which they waited.
- D. Rows for sessions that are currently waiting have their wait time incremented every microsecond.
- E. Rows for sessions that are not waiting always contain the total wait time since the session started.

Answer: BC

Explanation:

In the V\$SESSION view, Oracle provides information about the session waits:

B) When the WAIT_TIME column has a value of 0, it signifies that the session is currently waiting for a resource. This column represents the duration of the current or last wait.

C) If the session is not actively waiting, the WAIT_TIME column shows the time the session spent waiting for the last wait event. If the STATE column is showing "WAITED KNOWN TIME", it means the session is not currently waiting, but it indicates the time for which it had waited.

Reference:

Oracle Database Reference, 19c

Oracle Database Performance Tuning Guide, 19c

Question: 5

For which two actions can SQL Performance Analyzer be used to assess the impact of changes to SQL

performance?

- A. storage, network, and interconnect changes
- B. operating system upgrades
- C. changes to database initialization parameters
- D. database consolidation for pluggable databases (PDBs)
- E. operating system and hardware migrations

Answer: CD

Explanation:

SQL Performance Analyzer (SPA) can be used to assess the impact of different types of changes on SQL performance. These changes can include database initialization parameters, which can significantly affect how SQL statements are executed and therefore their performance. SPA allows you to capture a workload before and after the change and compare the performance of each SQL statement.

Database consolidation, including moving to pluggable databases (PDBs), can also affect SQL performance. SPA can analyze the SQL workload to see how consolidation impacts performance, by comparing metrics such as elapsed time and CPU time before and after the consolidation.

Reference:

Oracle Database SQL Tuning Guide, 19c

Oracle Database Performance Tuning Guide, 19c

For More Information – Visit link below:
<https://www.testsexpert.com/>

16\$ Discount Coupon: **9M2GK4NW**

Features:

■ Money Back Guarantee.....



■ 100% Course Coverage.....



■ 90 Days Free Updates.....



■ Instant Email Delivery after Order.....

