🚀 🚀 Quick & Fast AWR Report Analysis Checklist 🚀 🚀

When analyzing an **AWR report**, focus on the **most critical sections** first. Below is a **step-by-step, quick analysis guide** to identify performance bottlenecks efficiently.

---

1. Key Summary Metrics (Top of AWR Report)

📍 **Location:** "Report Summary"
📍 **Purpose:** Get a **quick overview** of system workload.

✅ **Key Metrics to Check:**

Metric	Ideal Value	Issue If...
DB Time vs. **Elapsed Time**	Should be similar	If **DB Time > 3x Elapsed Time**, system is overloaded.
DB CPU %	< 70% of DB Time	If high, CPU-intensive queries exist.
Hard Parse %	< 20%	If high, inefficient SQL execution (use bind variables).
Logons per Sec	< 5	If high, possible connection pool misconfiguration.

🔎 **Quick Check:**

- **High DB Time?** → Look for high CPU/Wait Events.
- **High Hard Parses?** → Check for SQL parsing issues.
- **High Logons?** → Use **connection pooling**.

---

2. Load Profile Analysis

📍 **Location:** "Load Profile"
📍 **Purpose:** Understand database activity.

✅ **Focus Areas:**

Metric	Issue If...	Impact
Transactions Per Sec (TPS)	Sudden drop	System may be slow under load.
User Calls Per Sec	Unusually high	Too many DB interactions per request.

Metric	Issue If...	Impact
Physical Reads/Writes	Higher than normal	Possible **I/O bottleneck**.
Redo Size	Excessively high	**Too many commits?** Check **log file sync** waits.

🔎 **Quick Check:**

- **High User Calls?** → Too many short transactions; optimize batching.
- **High Physical Reads?** → **Missing indexes?**
- **High Redo Size?** → **Too many commits per transaction?**

---

3. Top Timed Events (Bottleneck Identification)

📍 **Location:** "Top Timed Events"
📍 **Purpose:** Identify **top performance bottlenecks**.

✅ **Key Events & Their Fixes:**

Wait Event	Issue	Fix
DB CPU	High CPU usage	Optimize SQL, check indexing.
log file sync	Slow commits	Reduce commit frequency, optimize redo logs.
db file sequential read	Slow single-block reads (Index Lookups)	Check indexing, optimize queries.
db file scattered read	Slow multi-block reads (Full Table Scans)	Avoid FTS, use indexes, partitioning.
latch free	Contention for shared resources	Increase shared_pool_size, reduce parsing.
buffer busy waits	Buffer cache contention	Optimize memory, indexing, and tablespaces.

🔎 **Quick Check:**

- Focus on **top 3-5 wait events**.
- **DB CPU > 50%?** → CPU-intensive queries.
- **IO Wait Events High?** → Possible **storage issue**.

4. SQL Statements Analysis

📍 **Location:** "SQL Ordered by ..."
📍 **Purpose:** Find **slow and expensive queries**.

✅ **Key Reports to Check:**

Section	Focus
SQL Ordered by Elapsed Time	Queries taking the most time.
SQL Ordered by CPU Time	Queries consuming excessive CPU.
SQL Ordered by Buffer Gets	Queries with high logical I/O.
SQL Ordered by Physical Reads	Queries causing high disk I/O.

🔎 **Quick Check:**

- **Find queries with high execution time** and **optimize execution plans** (EXPLAIN PLAN).
- **Reduce physical reads** using **indexes**.
- **Use bind variables** to avoid parsing overhead.

---

5. Tablespace & I/O Bottlenecks

📍 **Location:** "Tablespace I/O Statistics"
📍 **Purpose:** Identify **slow I/O operations**.

✅ **Key Areas to Check:**

Metric	Issue If...	Fix
Tablespace Read Latency	> 10 ms	Disk I/O bottleneck. Tune storage.
Tablespace Write Latency	> 5 ms	Consider SSDs or redo log tuning.
Datafile Reads/Writes	High on single file	Balance I/O across files.

🔎 **Quick Check:**

- If **high read latency** → **Tune queries, partition tables**.
- If **high redo log waits** → **Increase redo log size**.

---

6. Memory & PGA Analysis

📍 **Location:** "Memory Statistics"
📍 **Purpose:** Check for memory bottlenecks.

✅ **Key Metrics:**

Metric	Issue If...	Fix
Buffer Cache Hit Ratio	< 90%	Increase DB_CACHE_SIZE or optimize queries.
Library Cache Hit Ratio	< 95%	Reduce hard parsing, use bind variables.
PGA Memory Usage	Near max limit	Increase pga_aggregate_target.

🔎 **Quick Check:**

- **Low Buffer Cache Hit Ratio?** → **Insufficient memory or inefficient SQL**.
- **High Library Cache Misses?** → **Too many new SQL statements**.

---

7. Redo & Undo Bottlenecks

📍 **Location:** "Redo Log Statistics" & "Undo Statistics"
📍 **Purpose:** Identify issues with redo logging and undo tablespace.

✅ **Key Metrics:**

Metric	Issue If...	Fix
Redo Log Space Requests	> 0	Increase redo log size.
Log File Sync Waits	High	Check disk performance, optimize commits.
Undo Tablespace Usage	Near 100%	Increase undo tablespace.

🔎 **Quick Check:**

- **Frequent "log file sync" waits?** → **Optimize commit frequency**.
- **Undo tablespace full?** → Increase **undo tablespace** or tune **retention**.

---

8. Latch & Mutex Contention

📍 **Location:** "Latch Statistics"
📍 **Purpose:** Identify contention in shared memory structures.

✅ **Common Contention Issues & Fixes:**

Latch	Issue	Fix
Library Cache Latch	High parsing	Use bind variables.
Shared Pool Latch	High contention	Increase shared_pool_size.
Redo Allocation Latch	High redo contention	Optimize commits, increase redo logs.

🔎 **Quick Check:**

- **High latch waits?** → Reduce **SQL parsing & optimize memory**.

🚀 **Quick Summary Table**

Issue Type	Key Check	Fix
High CPU	DB CPU > 50%	Optimize SQL, indexing.
Slow Queries	Top SQL by Elapsed Time	Tune execution plans.
I/O Bottlenecks	db file sequential/scattered read	Optimize storage, indexing.
High GC Activity	Excessive undo/redo logs	Optimize undo retention.
Connection Issues	High Logons/sec	Use connection pooling.

🔥 **Final Takeaways for Quick AWR Analysis**

✓ Start with the Summary (DB Time, CPU, Transactions).
✓ Check Load Profile (User Calls, Hard Parses, Physical Reads).
✓ Analyze Top Wait Events (DB CPU, IO, Log Sync, Contention).
✓ Investigate Top SQL Queries (CPU, Elapsed Time, Buffer Gets).
✓ Check Tablespace I/O and Memory (Buffer Cache, PGA Usage).
✓ Look at Redo & Undo (Log Sync, Undo Tablespace Usage).
✓ Monitor Latch & Mutex (Contention on shared resources).