“Slow SQL Queries ⟶ Slow Application Response”

– A Beginner-Friendly Guide to AWR

When your application is slow, it’s often because of a slow SQL query

To fix it, you’ll need to *read and understand the AWR report*. This is a guide to help you get comfortable reading AWR reports.

What is AWR ❓

AWR stands for ***Automated Workload Repository***

It is a powerful diagnostic tool that captures snapshots of database activity over time. Think of AWR as your database's "black box recorder" - it collects critical performance data that helps you identify bottlenecks and troubleshoot issues.

By focusing on just a few key tables in the AWR report, you'll learn to spot performance problems and provide specific, actionable information to your database administrators or developers. Let's get started with the basics!

The First 4 Tables You Need to Understand

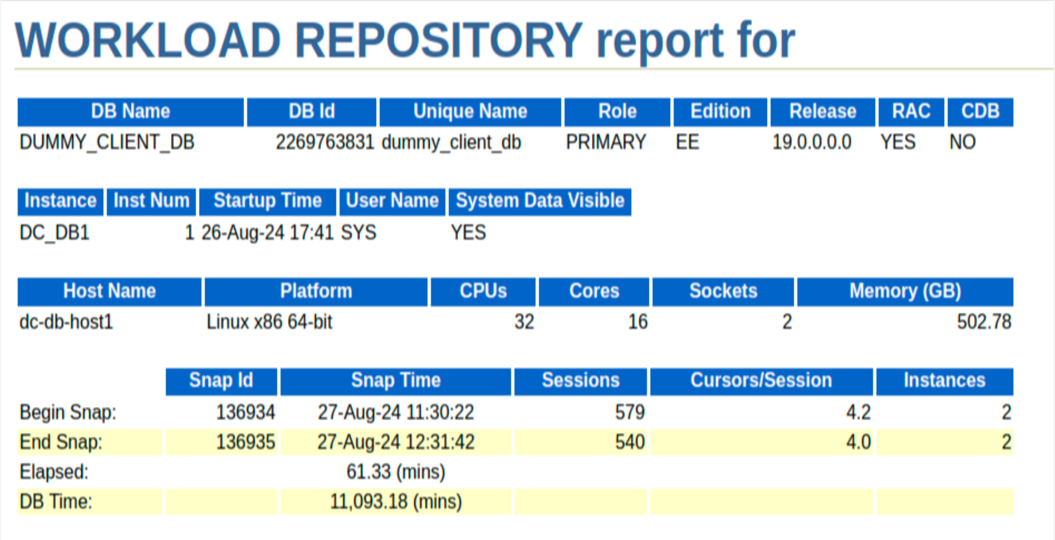
First 4 Tables on the AWR report introduce you to the Database, Database Instance, Host machine running the instance, and the time frame of the report.

1: DB Info ➡️

2. Instance Info ➡️

3. Host Info ➡️

4. Snap Info ➡️



WORKLOAD REPOSITORY report for

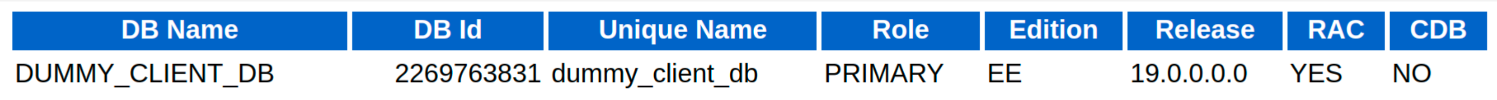
DB Name	DB Id	Unique Name	Role	Edition	Release	RAC	CDB
DUMMY_CLIENT_DB	2269763831	dummy_client_db	PRIMARY	EE	19.0.0.0.0	YES	NO

Instance	Inst Num	Startup Time	User Name	System Data Visible
DC_DB1	1	26-Aug-24 17:41	SYS	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
dc-db-host1	Linux x86 64-bit	32	16	2	502.78

	Snap Id	Snap Time	Sessions	Cursors/Session	Instances
Begin Snap:	136934	27-Aug-24 11:30:22	579	4.2	2
End Snap:	136935	27-Aug-24 12:31:42	540	4.0	2
Elapsed:		61.33 (mins)			
DB Time:		11,093.18 (mins)			

Table 1: Database Info



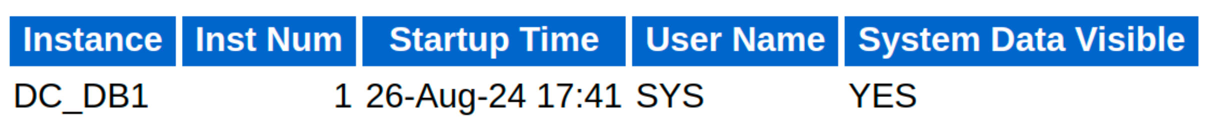
DB Name	DB Id	Unique Name	Role	Edition	Release	RAC	CDB
DUMMY_CLIENT_DB	2269763831	dummy_client_db	PRIMARY	EE	19.0.0.0.0	YES	NO

Columns	🔍 What to Look For
DB Name, DB Id, Unique Name	Helps identify the database — useful to make sure you are analyzing the right one
Role	Is it a Primary or Standby? Only Primary matters for app-level performance.
Edition, Release	Knowing the version helps when searching for known bugs or tuning advice.
RAC, CDB	Indicates if the DB is clustered or containerized — performance tuning may differ.

RAC - Stands for Real Application Cluster. If the value for RAC is Yes, there will be other instances too. Check the ***Instance ID*** in the next table, and ask for AWR reports of other instances as well.

CDB - Stands for Container DB.

Table 2: Instance Info



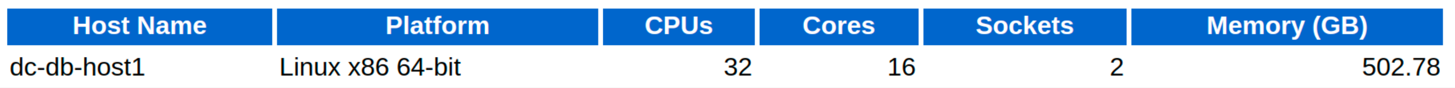
Instance	Inst Num	Startup Time	User Name	System Data Visible
DC_DB1	1	26-Aug-24 17:41	SYS	YES

Columns	What to Look For
Instance, Inst Num, Startup Time	Shows when the DB was restarted. A recent restart may reset trends.
User Name	Most of the time, you'll see it as **SYS**, because that's the default high-privileged user used for generating AWR reports.
System Data Visible	This field tells you whether the **AWR report includes system-level stats** like: ● OS statistics (CPU, memory usage, etc.) ● Background wait events ● Host-level metrics This is typically set to: ● **YES** – Full system data is included ● **NO** – The report is missing system data (maybe due to lack of permissions or some config being off) 💡 You generally want this to be **YES**, so that you get a complete picture — not just SQL stats, but also server-level resource utilization.

Table 3: Host Info

This Machine runs your Database

This table gives you a quick glance at the **hardware** and **OS environment** where the Oracle Database is running — very important for performance context.



Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
dc-db-host1	Linux x86 64-bit	32	16	2	502.78

Column	What It Means
Host Name	The **server name** (hostname) where this Oracle instance is running. Helps you identify the physical or virtual machine in your environment.
Platform	The **operating system and architecture**, e.g., `Linux x86 64-bit`, `Windows x86 64-bit`, etc. Useful for understanding the environment and compatibility.
CPUs*	Total number of **logical CPUs** (includes cores × threads). Affects how many processes Oracle can run in parallel.(See note below)
Cores*	Total number of **physical CPU cores**. More cores = better parallel query performance. (See note below)
Sockets	Number of **CPU sockets** on the motherboard. Usually, servers with multiple CPUs will have more sockets.
Memory (GB)	Amount of **physical RAM** available on the host. This impacts how much memory Oracle can use for caching, sorting, and buffer pools.

…Continued

🔍 *Note on CPUs and Cores in AWR Reports*

In AWR reports:

- ***Cores** = Physical processor cores*
- ***CPUs** = Logical threads (also called vCPUs)*

*→The number of **logical threads per core** depends on the processor architecture:*

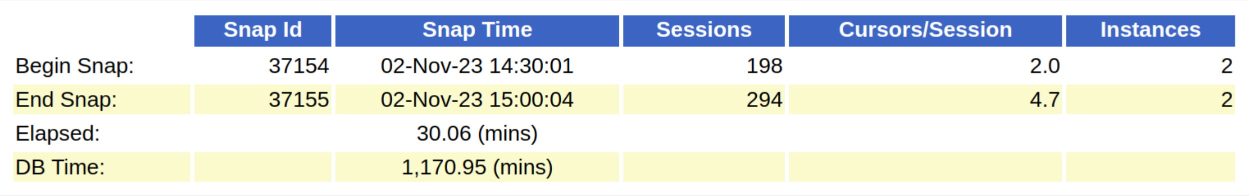
- *On **Intel/AMD**: 1 core → 2 threads (via Hyper-Threading / SMT)*
- *On **PowerPC/SPARC**: 1 core → up to 8 threads (via Simultaneous Multi Threading – SMT)*

This info helps you understand how much parallel processing your DB host is capable of.

Table 4: Snapshot Info

Time Window of the Report

Oracle takes a snapshot (think of a photo) of the current state of its operation.



	Snap Id	Snap Time	Sessions	Cursors/Session	Instances
Begin Snap:	37154	02-Nov-23 14:30:01	198	2.0	2
End Snap:	37155	02-Nov-23 15:00:04	294	4.7	2
Elapsed:		30.06 (mins)			
DB Time:		1,170.95 (mins)			

Key points of this table:

- **Begin Snapshot -** Starting snapshot of the report
- **End Snapshot -** Ending snapshot of the report
- **Elapsed time - Duration** of the AWR Report
- **The 4th row is DB time -** The total time the database spent actively processing requests (CPU + wait time). We won't dive into using it just yet, but know that many tables in AWR report reference % of DB Time — so it's worth getting familiar with the term.
- **Snap id** is the numeric id of the snapshot. Typically they are in sequence. However if the difference in them is more than 1, you can get another AWR report for a finer timeframe.

…Continued

- **Snap Time** is the time when that snapshot was taken.
- **Sessions, Cursors/Sessions & Instances** columns show the respective values at the beginning & End time of the AWR Report. The numbers might have increased or decreased between these AWR snapshots but that information is not shown in the AWR report. These define the **time range** (aka reporting interval) for the entire AWR report.

ℹ️ By default every 1 hour, but it is configurable.

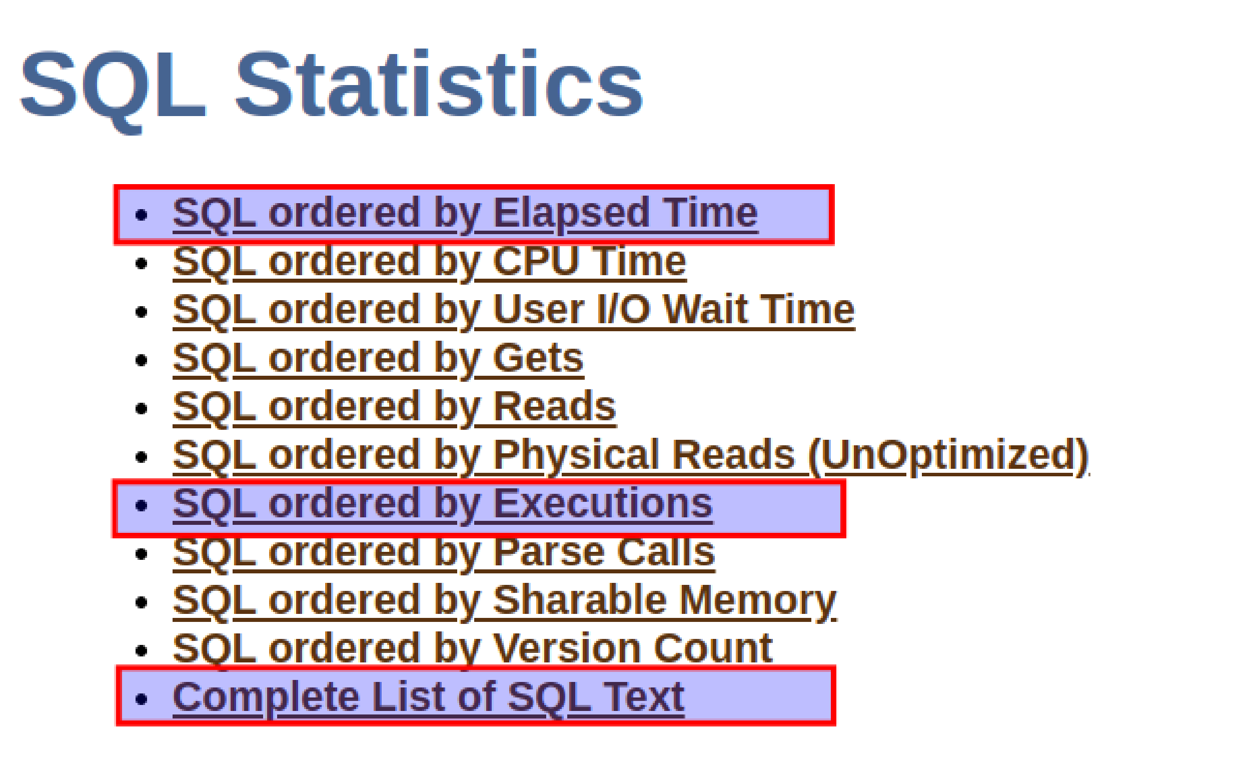
👉 For 30 mins of load tests, it is advised to change the duration of snapshot interval to 15 mins, and generate at least 3 AWR reports of 15 mins each to completely cover the load test duration. This helps you compare different phases of your load test.

👉 If you are analyzing for production, 1 hour gap in snap is perfectly fine.

The **important information** for you at this time are **Snap times, Snap Id & Elapsed time**. The rest you can learn later when you acquire advanced knowledge about Oracle Database tuning.

Next chapter is about **Identifying Slow Queries**

Next, find **SQL Statistics** Section in the AWR report.
(Tip: You can press **Ctrl+F** in your browser to search with **"SQL Statistics"** string in AWR Report's Web Page)



SQL Statistics

- SQL ordered by Elapsed Time
- SQL ordered by CPU Time
- SQL ordered by User I/O Wait Time
- SQL ordered by Gets
- SQL ordered by Reads
- SQL ordered by Physical Reads (UnOptimized)
- SQL ordered by Executions
- SQL ordered by Parse Calls
- SQL ordered by Sharable Memory
- SQL ordered by Version Count
- Complete List of SQL Text

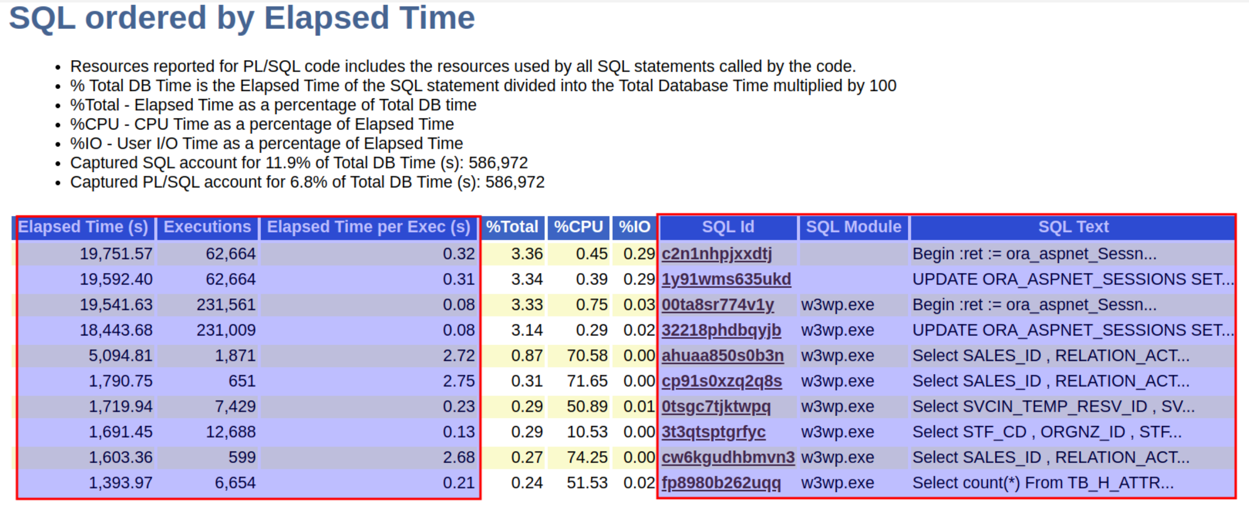
The 3 key subsections here are

- ***SQL ordered by Elapsed Time***
- ***SQL ordered by Executions***
- ***Complete List of SQL Text***

Click on "***SQL ordered by Elapsed Time***"

Table: SQL Ordered by Elapsed Time

Main table of slow queries



SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 11.9% of Total DB Time (s): 586,972
- Captured PL/SQL account for 6.8% of Total DB Time (s): 586,972

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
19,751.57	62,664	0.32	3.36	0.45	0.29	c2n1nhpjxxdtj		Begin :ret := ora_aspnet_Sessn...
19,592.40	62,664	0.31	3.34	0.39	0.29	1y91wms635ukd		UPDATE ORA_ASPNET_SESSIONS SET...
19,541.63	231,561	0.08	3.33	0.75	0.03	00ta8sr774v1y	w3wp.exe	Begin :ret := ora_aspnet_Sessn...
18,443.68	231,009	0.08	3.14	0.29	0.02	32218phdbqyjb	w3wp.exe	UPDATE ORA_ASPNET_SESSIONS SET...
5,094.81	1,871	2.72	0.87	70.58	0.00	ahuaa850s0b3n	w3wp.exe	Select SALES_ID , RELATION_ACT...
1,790.75	651	2.75	0.31	71.65	0.00	cp91s0xzq2q8s	w3wp.exe	Select SALES_ID , RELATION_ACT...
1,719.94	7,429	0.23	0.29	50.89	0.01	0tsgc7tjktwpq	w3wp.exe	Select SVCIN_TEMP_RESV_ID , SV...
1,691.45	12,688	0.13	0.29	10.53	0.00	3t3qtsptgrfyc	w3wp.exe	Select STF_CD , ORGNZ_ID , STF...
1,603.36	599	2.68	0.27	74.25	0.00	cw6kgudhbmvn3	w3wp.exe	Select SALES_ID , RELATION_ACT...
1,393.97	6,654	0.21	0.24	51.53	0.02	fp8980b262uqq	w3wp.exe	Select count(*) From TB_H_ATTR...

The key columns are (not in order of appearance)

- ***SQL Text*** - Actual text of the SQL query
- ***SQL Id*** - alphanumeric unique id generated for the given ***SQL Text***
- ***Elapsed Time per Exec (s)*** - Average time taken per execution by *SQL Id*
- ***Executions*** - Total number of executions of that query during Elapsed time of the AWR report
- ***Elapsed Time (s)*** - *Executions* x *Elapsed Time per Exec (s)*

✅ *"Tip: You've seen two types of Elapsed Time — they mean different things:*

- In *Snapshot section*: it's the duration of the AWR report
- In other *SQL sections*: it's the total time the query took to execute across all runs"

🚩 *How to Spot A Slow Query*

Case 1: Very High `Elapsed Time` and `Elapsed time per Exec (s)`

- These are **expensive queries** that take too long.
- Even a **single execution** may be hurting performance.

Case 2: Many Executions + Moderate Elapsed Time

- These queries may not be slow per execution, but because they run **too often**, they add up.
- These are **high-impact frequent queries**.

Tips to Investigate Further

- Click the **SQL Id** to go to the **full SQL Text** later in AWR.
- Once you find a slow query, ensure that it belongs to the slow transaction/API you are concerned with. → Sometimes the identified slow query may be related to a background job or another application and not related to your transaction/API.
- Compare the ***Elapsed Time (s)***, ***Executions***, ***Rows per Exec*** for the same SQL Id across multiple AWR reports. For example, compare the first half of the load test with the second. Peak and Non Peak, etc. → Sometimes, the query runs slow only for one instance or one interval only.

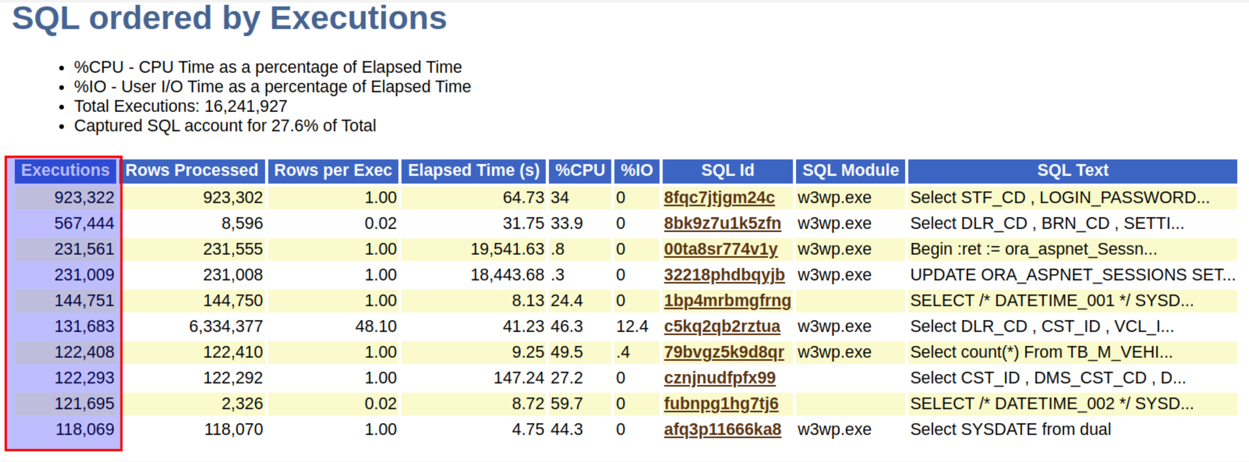
🔄 *Watch Out for Over-Executed Queries (Chatty APIs)*

Sometimes, your app is not slow because of **a few slow queries**, but because of **many fast queries** being executed **too frequently**. This is called **“chatty” behavior** — where the database is overwhelmed by sheer volume.

You can refer to 2 tables for this case.

Table 1: ***SQL ordered by Elapsed Time*** table → Executions Column (Refer previous section)

Table 2: ***SQL Ordered by Executions*** (See below table)



SQL ordered by Executions

- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Total Executions: 16,241,927
- Captured SQL account for 27.6% of Total

Executions	Rows Processed	Rows per Exec	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
923,322	923,302	1.00	64.73	34	0	8fqc7jtjgm24c	w3wp.exe	Select STF_CD , LOGIN_PASSWORD...
567,444	8,596	0.02	31.75	33.9	0	8bk9z7u1k5zfn	w3wp.exe	Select DLR_CD , BRN_CD , SETTI...
231,561	231,555	1.00	19,541.63	.8	0	00ta8sr774v1y	w3wp.exe	Begin :ret := ora_aspnet_Sessn...
231,009	231,008	1.00	18,443.68	.3	0	32218phdbqyjb	w3wp.exe	UPDATE ORA_ASPNET_SESSIONS SET...
144,751	144,750	1.00	8.13	24.4	0	1bp4mrbmgfrng		SELECT /* DATETIME_001 */ SYSD...
131,683	6,334,377	48.10	41.23	46.3	12.4	c5kq2qb2rztua	w3wp.exe	Select DLR_CD , CST_ID , VCL_I...
122,408	122,410	1.00	9.25	49.5	.4	79bvgz5k9d8qr	w3wp.exe	Select count(*) From TB_M_VEHI...
122,293	122,292	1.00	147.24	27.2	0	cznjnudfpfx99		Select CST_ID , DMS_CST_CD , D...
121,695	2,326	0.02	8.72	59.7	0	fubnpg1hg7tj6		SELECT /* DATETIME_002 */ SYSD...
118,069	118,070	1.00	4.75	44.3	0	afq3p11666ka8	w3wp.exe	Select SYSDATE from dual

Pro Tip: Look at the ***Rows per Exec*** count with respect to ***Executions*** Count, High ***Rows per execution***, as well as very low ***Rows per Exec*** but high ***Executions*** count can cause high response time.

📌 *Key Takeaways*

- Use **SQL ordered by Elapsed Time** to find slow queries
- Watch out for **chatty APIs** with frequent executions
- Use **SQL Id** to trace the query in other sections or instance of AWR report
- Focus on **Snap Time, Elapsed Time**, and **Executions** to isolate issues

Pro Tip: AWR report comes in Text & HTML format. Whenever possible, use HTML for easy navigation.

What Next? Ask for Plans, Compare Across Reports

1. Ask for a ***Query Execution Plan*** or ***SQL plan*** for the given ***Query ID***.
2. Check other areas or instance of the AWR using the ***SQL Id***
3. Coordinate with DBAs or Performance Engineers

🙌 Feeling Confident with AWR Now? Let Me Know!