# V$ Dynamic Performance Views

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| V$SYSSTAT | A table containing a collection of general database performance statistics. |
| V$SESSION | Contains one row per connected session. |
| V$SESSTAT | Contains the statistics from V$SYSSTAT for each session. |
| V$PROCESS | Contains one row per server process dedicated server, shared server, background, process and so on. |
| V$SQL, V$SQL\_PLAN, V$SQL\_PLAN\_STATISTICS | Contains details of cached SQL |
| V$SYSTEM\_EVENT | Records totals of waits since database startup. |
| V$SESSION\_EVENT | Records totals of waits for individual sessions. |

Wait interface alone has never given complete picture of Oracle Performance.

The time model accurately records total elapsed time, CPU time, and time spend on various interesting activities that involve both CPU and wait times.

**V$SYS\_TIME\_MODEL** for time model data across whole database and

**V$SESS\_TIME\_MODEL** for time model for individual session.

The following query provides the high-level summary of the waits experienced in a database:

SQL> ***SELECT wait\_class, event, total\_waits AS waits,***

***ROUND(time\_waited\_micro / 1000, 2) AS total\_ms,***

***ROUND(time\_waited\_micro \* 100 / SUM(time\_waited\_micro) OVER (), 2) AS pct\_time,***

***ROUND((time\_waited\_micro / total\_waits) / 1000, 2) AS avg\_ms***

***FROM V$SYSTEM\_EVENT***

***WHERE wait\_class <> 'Idle'***

***ORDER BY time\_waited\_micro DESC;***

SQL with wait interfaces and time model joined together.

> ***SELECT wait\_class, event, total\_waits AS waits,***

***ROUND(time\_waited\_micro / 1000, 2) AS total\_ms,***

***ROUND(time\_waited\_micro \* 100 / SUM(time\_waited\_micro) OVER (), 2) AS pct\_time,***

***ROUND((time\_waited\_micro / total\_waits) / 1000, 2) AS avg\_ms***

***FROM V$SYSTEM\_EVENT***

***WHERE wait\_class <> 'Idle'***

***UNION***

***SELECT stat\_name, NULL, VALUE***

***FROM V$SYS\_TIME\_MODEL***

***WHERE stat\_name IN ('DB CPU', 'background cpu time')***

***ORDER BY 3 DESC;***