

## 【実行完了した SQL 文の実行統計および実行経過の経過記録】

## 【現在実行中の負荷が重い SQL 文の実行統計および実行経過】

実行中の SQL 文に関して、以下のような実行統計を調査することが出来る

調査可能項目： CPU 時間、待機時間、処理の開始時間、読取りブロック数

対象オブジェクト： V\$SESSION

比較) V\$SESSION 動的ビューは、現在処理を行っている SQL 文にのみ実行統計情報が出力される

実行中の SQL 文の処理が終了すると、その実行統計情報が削除される

注意) V\$SESSION はセッションが終了するとその情報は削除されるが、新たなセッションが作成された時に、その中の SID 項目には値が使い回してセットされるので、下の統計情報の SID と Where 条件で一致させても意味はない

実行済み SQL 文に関して、過去に遡ってその情報を調査することが出来る

(セッションが終了しても、その記録は残っている)

調査可能項目： CPU 時間、待機時間、処理の開始時間、読取りブロック数

対象オブジェクト： V\$ACTIVE\_SESSION\_HISTORY、  
DBA\_HIST\_ACTIVE\_SESS\_HISTORY、  
V\$SESSION\_LONGOPS、  
V\$SQL\_MONITOR ← ※ 終了後 1 分間程度のみしか残っていない

【補 足】： 現在実行中の SQL 文を対象にする場合は、V\$SESSION ビューの SID、SERIAL#、SQL\_ID、SQL\_EXEC\_ID、USERNAME の 5 つが一致するレコードのみを抜き取れば良い

## 【 調査対象ビューの**利用目的**と特徴 】

目 的：

現在実行されている SQL 文について調査する時に使用する

現在の待機状態や待機状態継続時間、現在実行している SQL 文の CPU 使用時間についてを調査する

調査対象ビュー	特 徴
V\$SESSION	<p>現在実行中のセッションの待機状況や待機させているブロッカーを確認できる</p> <p>実行計画や実行統計（CPU 時間、Disk アクセス時間等）を見るためには、v\$sql_plan、v\$sqlarea を SQL_ID 一致条件で、Select する</p> <p>SQL 処理が終了すると、その実行統計情報も削除されてしまう</p>

目 的：

ライブラリ・キャッシュに残っている**実行計画**について調査する時に使用する

調査対象ビュー	特 徴
V\$SQL_PLAN_MONITOR	SQL 内のオペレーションに対する詳細な実行統計が確認できます

※ V\$ACTIVE\_SESSION\_HISTORY と DBA\_HIST\_ACTIVE\_SESS\_HISTORY には、Diagnostic Pack ライセンスが必要

※ V\$SQL\_MONITOR には、Tuning Pack ライセンスが必要

目的：

START\_TIME と LAST\_UPDATE\_TIME を使用して、指定時間に負荷が大きかった SQL 文を特定するときに使用する

なお、調査に使える情報としては少なく、START\_TIME、ELAPSED\_SECONDS と SQL\_PLAN\_HASH\_VALUE くらいしかない

ただし、V\$SQL\_MONITOR と比較して、SGA メモリに残っている時間は長いので、SQL 終了後に見つかる可能性が高い

調査対象ビュー	特 徴
V\$SESSION_LONGOPS	<p>実行に 6 秒以上（絶対時間）を要しているセッションの実行時間や終了推定時間が提供される 処理終了後のセッション情報も残っている</p> <p>実行計画や実行統計（CPU 時間、Disk アクセス時間等）を見るためには、v\$sql_plan、v\$sqlarea を SQL_ID 一致条件で、Select する必要がある</p>

V\$SESSION\_LONGOPS と V\$SESSION を組み合わせることによって、現在実行中で実行時間が長い SQL 文を洗い出すことが出来る

目的：

SQL\_EXEC\_START と LAST\_REFRESH\_TIME を使用して、指定時間に負荷が大きかった SQL 文の特定とリソースの使用状況などの SQL の実行統計情報を調査するときに使用する

出力される SQL の実行統計情報に関する項目は、実行時間（CPU 使用＋待機状態）、CPU 使用時間、バッファ使用量、ディスク読込み量、SQL\_ID、SQL\_TEXT など多数用意されている

また、V\$SQL\_MONITOR（リアルタイムに更新される）と V\$SESSION を組み合わせることによって、現在実行中の SQL 文の実行状況を監視することが出来る

ただし、SGA メモリに残っている時間は 1 分間しか保証されていないので、SQL 終了後に調査する場合は、情報が残っている可能性が低い

調査対象ビュー	特 徴
V\$SQL_MONITOR	<p>リアルタイムで SQL 文が消費した CPU 時間や読込みブロック数（バッファ数）などの実行統計情報が提供される</p> <p>※ V\$SQL_MONITOR は、SQL 文の 1 回の実行ごとにレコードを作成するので、現在の SQL 文の実行統計値に限定して情報を取得することができる</p> <p>注意）レコードが記録されている期間は、SQL 文の処理が終了してから 1 分程度</p> <p>―― 記録される処理の条件 ――</p> <ul style="list-style-type: none"><li>・ 5 秒以上 CPU 時間が消費されている SQL</li><li>・ 5 秒以上 I/O 時間が消費されている SQL</li><li>・ パラレル実行されている SQL</li><li>・ /*+ MONITOR */ ヒント句が指定されている SQL 文</li></ul> <p>―― ビューの中で直接保持している統計情報 ――</p> <p>ELAPSED_TIME（CPU 時間＋待機時間）</p> <p>CPU_TIME（待機時間）</p> <p>FETCHES（フェッチ回数）</p> <p>BUFFER_GETS（バッファ読込みバイト数）</p> <p>DISK_READS（ディスク読込みブロック数）</p> <p>DIRECT_WRITES（ダイレクトディスク書込ブロック）</p> <p>現在実行中の SQL 文のみを対象にする場合は、V\$SESSION との Where 条件指定が必要</p> <p>永続的に保存される AWR（DBA_HIST_****）のテーブルは、存在しない</p>

目的：

過去に問題が発生していた**時間帯**の**全セッション**の SQL の状態を調べる時に使用する

このオブジェクトに対し、Where 条件で SAMPLE\_TIME を Between で時刻（1 秒間）を指定し、調査したい時点での SQL 文の EVENT を確認すれば、待機状態についての調査が出来る

また、SQL\_ID や USER を Where 条件に指定することによって、実行していた SQL 文の時間別の実行状態遷移が確認できる

時刻ごとの EVENT 発生件数の集計を取ることによって、発生イベントの遷移状況が確認できる

調査対象ビュー	特 徴
V\$ACTIVE_SESSION_HISTORY =ASH	<b>V\$SESSION</b> の情報を自動的に <b>1 秒に 1 回取得させたセッション情報の履歴</b> セッションが終了しても、保存されている （待機状況や待機させているブロッカー）  実行計画や実行統計（CPU 時間、Disk アクセス時間等）を見るためには、v\$sql_plan、v\$sqlarea を SQL_ID 一致条件で、Select する  記録されている内容は、V\$SESSION と同等である ※ <b>SQL 処理終了後も</b> 、過去に遡ってのセッションの遷移状態が記録されている
DBA_HIST_ACTIVE_SESS_HISTORY	<b>V\$ACTIVE_SESSION_HISTORY</b> のメモリ上では保持しきれない分の情報を、テーブルに書き出して保存したもの （待機状況や待機させているブロッカー）  実行計画や実行統計（CPU 時間、Disk アクセス時間等）を見るためには、v\$sql_plan、v\$sqlarea を SQL_ID 一致条件で、Select する  ただし、情報記録時間間隔が、10 秒と粗くされてしまう ※ <b>SQL 処理終了後も</b> 、過去に遡ってのセッションの遷移状態が記録されている

【 動的パフォーマンス・ビューが保持している主な情報（列名） 】

列 名	内 容
他のビューとの関連キー項目  ..¥黄 King ファイル¥p016.docx 参照	
SAMPLE_TIME	実行統計情報を取得した時刻
SID	セッション識別子
SERIAL#	セッション・シリアル番号
SQL_ID	SQL 文の識別子（10 g）
SQL_EXEC_ID	同一の SQL が複数回実行された場合に、それぞれの SQL 実行を区別できる識別子（11 g）
SQL_CHILD_NUMBER	現在実行されている SQL 文の実行計画を決定づけるための識別番号（10 g） <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px; margin-top: 5px;">                     ライブラリ・キャッシュには、1 つの SQL 文に対して、                      複数の実行計画を確保することが可能である                      その場合の実行計画を決定するための識別番号                 </div>
SQL_ADDRESS	SQL 文の識別子（9 g）
SQL_HASH_VALUE	SQL 文の識別子（9 g）
実行統計情報	
CPU_SECONDS	CPU 時間（秒）
CPU_TIME	CPU 時間
ELAPSED_SECONDS	実行時間（秒） 待機時間 = 実行時間 - CPU 時間
ELAPSED_TIME	実行時間
START_TIME	処理の開始時刻
SQL_EXEC_START	〃
DISK_READ	読取りブロック数
TIME_REMAINING	SQL 処理が終了するまでの残りの推定秒数 （ V\$SESSION_LONGOPS ）
EVENT	発生待機イベント
OPNAME	SQL 文における処理の操作内容の説明

## 【 調査用 SQL 】

### V\$SESSION

#### 実行中のセッションの待機状況と実行統計情報の調査

使用例 )

```
col sql_text format a40
select sid, serial#, substr(SA.sql_fulltext, 0, 40) sql_text, SA.cpu_time,
       (sysdate - SE.sql_exec_start) * 86400 AS ELAPSED_TIME_SEC,
       SA.executions, SA.disk_reads, SA.buffer_gets,
       SA.user_io_wait_time, SE.sql_id, SE.sql_exec_id, SE.event,
       SE.wait_time_micro, SE.seconds_in_wait, SE.program
from   V$SESSION SE, V$SQLAREA SA ※ 86400 秒 =
       24 時間 × 60 分 × 60 秒
where  SE.sql_id = SA.sql_id
       and SE.event Not like 'SQL*Net%'
       and SE.program Not like 'ORACLE.EXE%' ←
```

バックグラウンド・プロセスを除外するために必要

SID	SERIAL#	SQL_TEXT
106	2341	select SID, SERIAL#, SUBSTR(SA.SQL_T
127	605	select substr(SA.SQL_TEXT, 0, 40), S

ELAPSED_TIME_SEC		DISK_READS		BUFFER_GETS
CPU_TIME	↓	EXECUTIONS	↓	
15600	359	1	17788	480
31200	105	2	3911	870

SQL_ID	SQL_EXEC_ID	EVENT	WAIT_TIME_MICRO
kchy583mk45es	21348962	enq: TX - row lock contention	3120
6hm9tp1gwbtuh	16777221	enq: TX - row lock contention	4530

単位: 1/1,000,000 秒

SECONDS_IN_WAIT
458
2

↑

同一の待機イベントでずっと待機させられている時間が、SECONDS\_IN\_WAIT  
待機イベントが一度でも解消されれば、リセットされる

## V\$SESSION\_LONGOPS

### 6 秒以上かかった SQL の実行時間と作業内容調査

(指定時間帯に動作していた SQL 文が対象)

```
col sid format 99999
col serial# format 999999
col sql_id format a15
col sql_exec_id format 99999999999
col start_time format a19
col last_time format a19
col elapsed_seconds format 99999999999
col opname format a25
col username format a15
col sql_text format a300

select SSL.sid, SSL.serial#, SSL.sql_id, sql_exec_id,
       to_char( SSL.start_time, 'YYYY-MM-DD HH24:MI:SS' ) START_TIME,
       to_char( SSL.last_update_time, 'YYYY-MM-DD HH24:MI:SS' )
       LAST_TIME,
       SSL.elapsed_seconds, SSL.opname, SSL.username, SL.sql_text
from   V$SESSION_LONGOPS SSL, DBA_HIST_SQLTEXT SL
where  SSL.sql_id = SL.sql_id(+)
       and TO_DATE( '<指定日時>', 'YYYY-MM-DD HH24:MI:SS' )
       BETWEEN start_time and last_update_time
order by START_TIME ;
```

<指定日時> : '2017-11-28 16:46:00'

※ SQL 文の実行が最近の場合、AWR スナップショットが採取されていない  
と SQL 文が記録されていないので、SQL\_TEXT 列は空白となる

このことを回避するためには、Select 対象テーブルを v\$sql ビューに変更  
するか、もしくは、AWR スナップショットを手動で採取する

```
execute DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT
        ('TYPICAL') ;
```

SID	SERIAL#	SQL_ID	SQL_EXEC_ID	START_TIME
110	3621	6hm9tp1gwbtuh	16777221	2017-11-28 16:45:23
ELAPSED_TIME_SEC		OPNAME	SQL_TEXT	
43		Table Scan	SELECT * FROM . . .	

単位 : 秒



## V\$SQL\_MONITOR

現在実行中で長時間処理中の SQL 文の洗い出し

```
col sid          format 99999
col sql_id       format a14
col username     format a14
col cpu_time_s   format 999999999
col elapsed_time_s format 999999999
col program      format a30
col sql_text     format a300
```

86300=60×60×24

```
Select SE.sid, SE.sql_id, SE.username, cpu_time/1000000 CPU_TIME_S,
       elapsed_time/1000000 ELAPSED_TIME_S,
       round((sysdate - SE.sql_exec_start) * 86300, 0) ELAPSED_SED,
       . . . , . . . , . . . , . . . , . . . , . . . , . . . ,
       SE.program, SM.sql_text
```

```
From V$SQL_MONITOR SM, V$SESSION SE
```

```
Where SM.sid          = SE.sid
      and SM.session_serial# = SE.serial#
      and SM.sql_id     = SE.sql_id
```

```
      and SM.sql_exec_id = SE.sql_exec_id
```

```
      and SM.username   = SE.username
```

```
;
```

バックグラウンド・プロセスを除外するために必要

同一 SQL 文の複数回実行の時に、終了した SQL 実行を除外するために必要

SID	SQL_ID	USERNAME	CPU_TIME_S	ELAPSED_TIME_S
147	9x7tn5rdxkcjd	KOZUE	2560	3478

PROGRAM	SQL_TEXT
---------	----------

sqlplus.exe	select /*+ MONITOR */ * from emp f
-------------	------------------------------------

## 負荷の高い SQL の実行時間と実行統計の調査

注意) レコードが記録されている期間は、SQL 文の処理が終了してから 1 分程度

(指定時間帯に動作していた SQL 文が対象)

```
col sid format 99999
col session_serial# format 999999
col sql_id format a15
col sql_exec_id format 99999999999
col start_time format a19
col last_time format a19
col elapsed_seconds format 99999999999
col opname format a25
col username format a15
col sql_text format a300

select SM.sid, SM.session_serial#, SM.username, SM.sql_id,
       SM.sql_exec_id, to_char( SM.sql_exec_start, 'YYYY-MM-DD
       HH24:MI:SS' ) START_TIME, SM.elapsed_time, SM.cpu_time,
       SM.disk_reads, SM.sql_text, SM.program
from V$SQL_MONITOR SM
where TO_DATE('<指定日時>', 'YYYY-MM-DD HH24:MI:SS')
      BETWEEN sql_exec_start AND last_refresh_time
order by SM.sql_exec_start ;
```

<指定日時> : '2017-11-28 16:46:00'

SID	SESSION_SERIAL#	SQL_ID	SQL_EXEC_ID	SQL_EXEC_START
110	3621	6hm9tp1gwbtuh	16777221	2017-11-28 16:45:23

ELAPSED_TIME	CPU_TIME	DISK_READ	SQL_TEXT
194540899	40477899	17799	SELECT * FROM . . .

↑ 単位 : 1/1000,000 秒

SQL 文に対する CPU 時間と **ディスク Read** などの実行統計で、負荷が高い SQL 文の特定

## V\$ACTIVE\_SESSION\_HISTORY (=ASH)

## DBA\_HIST\_ACTIVE\_SESS\_HISTORY

このビューの項目には、CPU\_TIME ~~や ELAPSED\_TIME~~ が無いので、この処理  
ELAPSED\_TIME = sample\_time - sql\_exec\_start と仮定  
が「どれ位の時間実行されている」のかや「待機されていた合計時間」について  
は、判断が出来ないので注意すること

〔 特定 SQL 文処理に対しての実行進捗には、V\$SQL\_MONITOR で調査する  
もしくは、このビューの複数時間レコードを確認して状況を把握する 〕

### 1 秒間隔の実行中 SQL 文の状態遷移と実行時間の調査

時刻ごとの EVENT 発生件数の集計

時刻	イベント名	発生回数
2019/02/10 11 : 00 : 00	イベント A	13
	イベント B	6
	イベント C	2
	イベント D	0
2019/02/10 11 : 00 : 01	イベント A	25
	イベント B	8
	イベント C	0
	イベント D	0
2019/02/10 11 : 00 : 01	イベント A	30
	イベント B	7
	イベント C	0
	イベント D	5

EVENT 別発生回数

時刻 \ 発生回数	イベント A	イベント B	イベント C	イベント D
2019/02/10 11:00:00	13	6	2	0
2019/02/10 11:00:01	25	8	0	0
2019/02/10 11:00:02	30	7	0	5

## 過去についての状況を調査する場合

※ 過去の SQL の実行状況についてを調査する場合には、  
DBA\_HIST\_ACTIVE\_SESS\_HISTORY と DBA\_HIST\_SQLTEXT、  
~~DBA\_LOCK~~を使用する

使用例)

```
col SAMPLE_TIME format a25
col EVENT format a30
col ELAPSED_TIME_SEC format a25
col NAME format a15
```

```
select  ASH.sample_time ,  B.name ,  ASH.session_id ,
        ASH.session_serial# ,      ASH.sql_id ,      ASH.sql_exec_id ,
        ASH.session_state ,  ASH.event ,  ASH.time_waited ,
        substr( to_char( ( ASH.sample_time - ash.sql_exec_start ) , 'YYYY-MM-DD
        HH24:MI:SS' ) ,11 ,9)  AS  ELAPSED_TIME_SEC ,
        substr( SL.sql_text ,  0 ,  40 ) ,  ASH.program
```

```
from    V$ACTIVE_SESSION_HISTORY  ASH,  V$SQLTEXT  SL ,
        sys.user$  B
where   session_type = 'BACKGROUND'
       and ASH.sql_id = SL.sql_id(+)
       and SL.piece = 0      ← V$SQLTEXT の SQL 文は、分割されている
       and ASH.user_id = B.user#(+)      最初の部分だけ表示させる
       and ASH.sample_time Between '2001/01/01 00:00:00'
                                   and '2020/12/31 23:59:59'

order by ASH.sample_time ;
```

注意)

ORA-01031: 権限が不足しています

とエラーが表示された場合、SYSTEM ユーザーでは sys.user\$オブジェクトに対して、Select 権限が付与されていないことに原因がありました

SYS ユーザーにて、grant Select on sys.user\$ to system; を実行のこと

実行結果)

SAMPLE_TIME	SESSION_ID		SQL_ID		SQL_EXEC_ID
	↓	SESSION_SERIAL#	↓		
-----	-----	-----	-----	-----	-----
17-11-30 12:34:05.000	125	1079	47ktpzzg4du31		16777217
17-11-30 12:34:05.000	109	1644	4xcvf44gp3174		36778516
17-11-30 12:34:05.000	125	1079	47ktpzzg4du31		16777217
17-11-30 12:34:05.000	109	1644	4xcvf44gp3174		36778516
17-11-30 12:34:06.000	125	1079	47ktpzzg4du31		16777217
17-11-30 12:34:06.000	109	1644	4xcvf44gp3174		36778516

SESSION_STATE	EVENT	TIME_WAITED	ELAPSED_TIME_SEC		SQL_TEXT
			↓	↓	
WAITING	enq: TX - row lock contention	0	02:10:00		select * from...
ON CPU		0	01:00:05		select * from...
WAITING	enq: TX - row lock contention	0	02:10:01		select * from...
ON CPU		0	01:00:06		select * from...
WAITING	enq: TX - row lock contention	0	02:10:02		select * from...
ON CPU		0	01:00:07		select * from...

- ※ 1つのセッションで実行している SQL 文に対し、V\$SQLTEXT が複数行に分割して保存されているために、同一セッションの行が複数出力された  
V\$SQLTEXT の piece 列が、0、1、2 となっている

## 特定端末（特定セッション）での SQL 文の経過記録

### = ロック元で実行していた SQL 文の調査方法

V\$ACTIVE\_SESSION\_HISTORY (=ASH)

- ※ 過去の SQL の実行状況についてを調査する場合には、  
DBA\_HIST\_ACTIVE\_SESS\_HISTORY と DBA\_HIST\_SQLTEXT、  
~~DBA\_LOCK~~を使用する

【端末名を指定して実行 SQL 文の状況を調査する場合】

V\$ACTIVE\_SESSION\_HISTORY と DBA\_HIST\_ACTIVE\_SESS\_HISTORY には  
端末名は、保持されていない  
代わりに、ユーザー名で対応させる  
前ページの SQL 文の Where 条件に、user\_id を加える

調査したいユーザー名を指定

↓

and B.name = 'KOZUE'

SAMPLE_TIME	SESSION_ID		SQL_ID	SQL_TEXT
	↓	↓		
	SESSION_SERIAL#			
17-11-30 12:34:05.000	125	1079	47ktpzzg4du31	Select A From . .
17-11-30 12:34:06.000	125	1079	47ktpzzg4du31	Select A From . .
17-11-30 12:34:07.000	125	1079	47ktpzzg4du31	Update K Set No = . .

↑

- ※ ロック元の SQL 文を決定するには、調査で表示された SQL 文を個別に  
ロックされた側の SQL 文と関係があるかの比較検討を行う  
なお、DBA\_HIST\_SQLTEXT では、SQL 文は分割されていない

注意)

- サンプリング時に実行されていた SQL 文しか出力されてこない  
サンプリングの間の短時間で処理されたものには、出力されてこない
- ・ V\$ACTIVE\_SESSION\_HISTORY (=ASH) 1 秒／サンプリング
  - ・ DBA\_HIST\_ACTIVE\_SESS\_HISTORY 10 秒／サンプリング

このビューには、CPU\_TIME が無いので、処理が実行されているのか、待機されたままなのかの判断が出来ない ので、注意すること

特定 SQL 文処理に対しての実行進捗には、V\$SQL\_MONITOR を使って調査できる

## 【ユーザー名を指定して実行している SQL 文の状況を調査する】

※ 過去の SQL の実行状況についてを調査する場合には、  
DBA\_HIST\_ACTIVE\_SESS\_HISTORY と DBA\_HIST\_SQLTEXT、  
~~DBA\_LOCK~~を使用する

使用例)

```
select  ASH.sample_time ,  substr( SL.sql_text ,  0 ,  100 ) ,  
        B.name ,  ASH.session_id ,  ASH.session_serial# ,  ASH.sql_id ,  
        ASH.sql_exec_id ,  ASH.session_state ,  ASH.event ,  ASH.time_waited ,  
        substr( to_char( ( ASH.sample_time  -  ASH.sql_exec_start ), 'YYYY-  
MM-DD HH24:MI:SS' ) ,11 ,9)      AS  ELAPSED_TIME_SEC ,  
        ASH.program  
from    DBA_HIST_ACTIVE_SESS_HISTORY  ASH,  
        DBA_HIST_SQLTEXT  SL ,  sys.user$ B  
where   session_type = 'FOREGROUND'  
and     ASH.sql_id = SL.sql_id(+)  
and     ASH.user_id = B.user#(+)  
and     ASH.sample_time  Between  '2018/04/06 14:00:00'  and '2018/04/06  
        15:00:00'  
and     B.name = 'KOZUE'  
order by ASH.sample_time  DESC ;
```

注意)

ORA-01031: 権限が不足しています

とエラーが表示された場合、SYSTEM ユーザーでは sys.user\$オブジェクトに対して、Select 権限が付与されていないことに原因がありました

SYS ユーザーにて、grant Select on sys.user\$ to system; を実行のこと

## 【セッションを指定して実行 SQL 文の状況を調査する】

※ 過去の SQL の実行状況についてを調査する場合には、  
DBA\_HIST\_ACTIVE\_SESS\_HISTORY と DBA\_HIST\_SQLTEXT、  
~~DBA\_LOCK~~を使用する

前ページの SQL 文の Where 条件に、session\_id を加える

調査したロック元のセッション ID を指定

↓

and ASH.session\_id = 125

使用例)

```
select  ASH.sample_time , substr( SL.sql_text , 0 , 100 ) ,
        B.name , ASH.session_id , ASH.session_serial# , ASH.sql_id ,
        ASH.sql_exec_id , ASH.session_state , ASH.event , ASH.time_waited ,
        substr( to_char( ( ASH.sample_time - ASH.sql_exec_start ) , 'YYYY-
        MM-DD HH24:MI:SS' ) ,11 ,9) AS ELAPSED_TIME_SEC ,
        ASH.program
from    DBA_HIST_ACTIVE_SESS_HISTORY ASH, DBA_HIST_SQLTEXT
        SL , sys.user$ B
where   session_type = 'FOREGROUND'
and     ASH.sql_id = SL.sql_id(+)
and     ASH.user_id = B.user#(+)
and     ASH.sample_time Between '2018/04/06 14:00:00' and '2018/04/06
        15:00:00'
and     ASH.SESSION_ID = 119
order by ASH.sample_time DESC ;
```

注意)

ORA-01031: 権限が不足しています

とエラーが表示された場合、SYSTEM ユーザーでは sys.user\$オブジェクトに対して、Select 権限が付与されていないことに原因がありました

SYS ユーザーにて、grant Select on sys.user\$ to system; を実行のこと



## その他

### V\$SQL\_MONITOR と V\$SQL\_PLAN\_MONITOR をつなぐ SELECT 文

```
select    VSM.key ,      VSM.sid ,      VSM.session_serial# ,      VSM.sql_id ,
          VSM.sql_exec_id ,      to_char( VSM.sql_exec_start , 'YYYY-MM-DD
          HH24:MI:SS' ) ,      VSM.elapsed_time ,      VSM.cpu_time ,
          VSM.disk_reads , VSM.buffer_gets , SL.sql_text
from      V$SQL_MONITOR  VSM , V$SQL_PLAN_MONITOR  VSPM,
          V$SQL  SL
where     VSM.key = VSPM.key
          and VSM.sql_id = SL.sql_id(+)
          and VSM.sid = <SID 値>
          and VSM.session_serial# = <セッション・シリアル番号> ;
```

### V\$SQL\_MONITOR と V\$SQL\_PLAN\_MONITOR の主な構成項目 (列)

V\$SQL_MONITOR		V\$SQL_PLAN_MONITOR	
Key	結合するための結合キー	Key	結合するための結合キー
Sid	セッション識別子	Sid	セッション識別子
session_serial#	セッション・シリアル番号		
sql_id	SQL 識別子	sql_id	SQL 識別子
sql_exec_id	複数実行の SQL 文を一意に識別する値	sql_exec_id	複数実行の SQL 文を一意に識別する値
sql_exec_start	実行の開始時刻	sql_exec_start	実行の開始時刻
elapsed_time	経過時間(マイクロ秒)		
cpu_time	CPU 時間(マイクロ秒)		
disk_reads	ディスク読取り数		
buffer_gets	バッファ取得操作数		