

Prática

Gerando trace

1. Antes, vamos criar o usuário para monitorar

```
Create user TESTETRACE identified by TESTE995TRACE;  
Grant connect to TESTETRACE;  
Grant resource to TESTETRACE;
```

```
Grant select on HR.COUNTRIES to TESTETRACE;  
Grant select on HR.REGIONS to TESTETRACE;  
Grant select on HR.LOCATIONS to TESTETRACE;  
Grant select on HR.DEPARTMENTS to TESTETRACE;  
Grant select on HR.JOBS to TESTETRACE;  
Grant select on HR.EMPLOYEES to TESTETRACE;  
Grant select on HR.JOB_HISTORY to TESTETRACE;
```

Logar no ambiente

```
sqlplus TESTETRACE/TESTE995TRACE@PDB1
```

Identificando a sessão

```
select S.sid, S.serial#, s.program, S.osuser, p.spid,  
S.machine, S.logon_time  
from v$session s, v$process p  
where s.paddr = p.addr  
and S.username='TESTETRACE';
```

```
set pages 80
```

```
set lines 300
col machine format a30
Col osuser format a10
alter session set nls_date_format='dd/mm/yyyy
hh24:mi';
```

```
select S.sid, S.serial#, s.program, S.osuser, p.spid,
S.machine, S.logon_time
  from v$session s, v$process p
 where s.paddr = p.addr
    and S.machine like '%&maquina%'
    order by s.program
/
```

```
set pages 80
set lines 180
col machine format a30
```

```
alter session set nls_date_format='dd/mm/yyyy
hh24:mi';
```

```
select S.sid, S.serial#, s.program, S.osuser, p.spid,
S.machine, S.logon_time
  from v$session s, v$process p
 where s.paddr = p.addr
    and S.program like '%&rotina%'
/
```

Ativando o monitoramento do trace

```
EXEC DBMS_MONITOR.session_trace_enable(session_id  
=>&SID, serial_num=>&SERIAL, waits=>TRUE,  
binds=>FALSE);
```

SQL de exemplo para execução

```
SELECT e.employee_id, e.first_name, e.last_name,  
d.department_name  
FROM hr.employees e  
JOIN hr.departments d ON e.department_id =  
d.department_id;
```

```
SELECT AVG(salary) AS average_salary  
FROM hr.employees;
```

```
SELECT employee_id, first_name, last_name, salary  
FROM hr.employees  
WHERE salary > (SELECT AVG(salary) FROM  
hr.employees);
```

```
SELECT employee_id, first_name, last_name, salary  
FROM hr.employees  
ORDER BY salary DESC;
```

```
SELECT d.department_name, COUNT(e.employee_id)  
AS employee_count  
FROM hr.employees e  
JOIN hr.departments d ON e.department_id =  
d.department_id  
GROUP BY d.department_name;
```

TKPROF

Tkprof arquivo.trc nometrace.txt sys=no

SORT: Essa opção permite que você especifique a ordem de classificação das instruções SQL no relatório.

Alguns valores comuns são:

- exeela: Classifica por tempo de execução crescente.
- exedsc: Classifica por tempo de execução decrescente.
- prsela: Classifica por número de vezes que a instrução foi executada (contagem de execução) crescente.
- prsdsc: Classifica por número de vezes que a instrução foi executada (contagem de execução) decrescente.
- fchela: Classifica por número de vezes que os dados foram buscados (contagem de busca) crescente.
- fchdsc: Classifica por número de vezes que os dados foram buscados (contagem de busca) decrescente.

Por exemplo, para classificar o relatório por tempo de execução decrescente, você pode usar a opção **SORT=exedsc**.

SYS: Essa opção controla a inclusão ou exclusão de informações do usuário no relatório. Quando **SYS=NO**, apenas informações do sistema são

exibidas, enquanto SYS=YES exibe informações do usuário também.

WAIT: Essa opção controla a inclusão ou exclusão de informações de espera de eventos no relatório. Quando WAIT=YES, informações sobre eventos de espera são exibidas, o que pode ser útil para identificar gargalos de desempenho. Por padrão, WAIT=NO.