INSTANȚA ORACLE

Suport Laborator Saptămâna 03

FEAA Master SIA/SDBIS

Cuprins

Instanța Oracle

La laboratorul de data trecută am pregătit mediul de lucru. Ne amintim că am instalat o nouă bază de date folosind "Database Configuration Assistant". Ăsta nu-i puțin lucru, mai ales că instalarea de noi servere Oracle e o activitate frecventă în fișa postului unui DBA Oracle.

Arhitectura serverului de baze de date Oracle

Din exterior, la nivelul sistemului de operare, baza de date Oracle ne apare sub forma unei colecţii de fişiere: fisiere de control, fişiere de date, redolog-uri etc. Da, nu greşim spunând că acele fişiere reprezintă de fapt baza noastră de date. Totuşi, datele stocate în acele fişiere nu ar putea fi accesate optim, partajat şi în condiţii de siguranţă fără o serie de procese şi structuri de memorie specializate, toate reunite sub denumirea de instanţă Oracle. În general, există o relaţie de unu la unu între o instanţă Oracle şi o bază de date asociată, deşi în configuraţiile te tip cluster sau, odată cu introducerea conceptului de bază de date de tip container (CDB), în versiunea 12c, această relaţie nu mai poate fi considerată biunivocă. La laborator vom acoperi cazul cel mai simplu, cel în care există o singură instanţă Oracle asociată bazei de date.

Pornirea și oprirea instanței

Instanţa Oracle poate fi pornită în diferite stadii şi, evident, poate fi şi oprită în moduri diferite. Oprirea sau pornirea instanţei nu se poate face, totuşi, decât folosind un utilizator cu drepturi de SYSDBA sau SYSOPER. De obicei, folosim utilizatorul SYS. Vom începe prin a opri instanţa Oracle:

```
[oracle@orasrv ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.2.0.1.0 Production on Tue Mar 12 07:33:43 2019

Copyright (c) 1982, 2016, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SYS@SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

Ce face clauza "immediate" a comenzii "shutdown"? Ce alte moduri de oprire a instanţei cunoaşteţi?

Pentru a porni instanţa Oracle folosim comanda "*startup*". O putem porni în modul "*nomount*" sau "*mount*" sau cu deschiderea efectivă a bazei de date.

```
SYS@SQL> startup nomount
ORACLE instance started.

Total System Global Area  838860800 bytes
Fixed Size                  3051184 bytes
Variable Size             599785808 bytes
Database Buffers          230686720 bytes
Redo Buffers                5337088 bytes
```

Încercați să deschideți o conexiune nouă cu utilizatorul SYSTEM. Ce se întâmplă? Explicați!

Pentru a pune instanța în module "*mount*", va trebui să dăm următoarea comandă (doar fișierele de control vor fi citite):

```
SYS@SQL> alter database mount;

Database altered.
```

Există și comanda "*startup mount*" pe care o putem folosi, dar instanța nu trebuie să fi fost pornită în "*nomount*" așa cum am facut noi anterior. Altfel, Oracle protestează cu un mesaj de eroare cât se poate de explicit: "ORA-01081: cannot start already-running ORACLE - shut it down first".

*Interogați view-ul system **DBA_DATA_FILES** pentru a afla câte fisiere de date sunt în componența bazei de date. Ce obțineți? Cum explicați?*

În sfârșit, pentru a deschide fișierele bazei de date și a o face accesibilă va trebui să dăm următoarea comandă:

```
SYS@SQL> alter database open;

Database altered.
```

Desigur, dacă nu vrem să trecem prin pașii anteriori și dorim să deschidem direct baza de date e suficient să dăm comanda "*startup*".

Procese ale instanței Oracle

Atunci când pornim instanța bazei de date, câteva procese foarte importante sunt pornite. Aceste procese nu sunt altceva decât programe/module/bucăți de cod software specializat, care comunică între ele, transparent, pentru a asigura buna funcționare a bazei de date Oracle.

Pe sistemele Windows, aceste procese sunt implementate ca fire de execuție (thread-uri) în executabilul "*oracle.exe*". Pe sistemele bazate pe UNIX, le putem vedea ca procese de sine stătătoare.

```
[oracle@orasrv ~]$ ps x | grep ora_
 2888 ?        Ss     0:00 ora_pmon_mydb
 2890 ?        Ss     0:00 ora_clmn_mydb
 2892 ?        Ss     0:00 ora_psp0_mydb
 2895 ?        Ss     0:02 ora_vktm_mydb
 2899 ?        Ss     0:00 ora_gen0_mydb
 2903 ?        Ss     0:00 ora_mman_mydb
 2905 ?        Ssl    0:00 ora_gen1_mydb
 2909 ?        Ss     0:00 ora_diag_mydb
 2911 ?        Ssl    0:00 ora_ofsd_mydb
 2915 ?        Ss     0:00 ora_dbrm_mydb
 2917 ?        Ss     0:00 ora_vkrm_mydb
 2919 ?        Ss     0:00 ora_svcb_mydb
 2921 ?        Ss     0:00 ora_pman_mydb
 2923 ?        Ss     0:00 ora_dia0_mydb
 2925 ?        Ss     0:00 ora_dbw0_mydb
 2927 ?        Ss     0:00 ora_lgwr_mydb
 2929 ?        Ss     0:00 ora_ckpt_mydb
 2931 ?        Ss     0:00 ora_smon_mydb
 2933 ?        Ss     0:00 ora_smco_mydb
 2935 ?        Ss     0:00 ora_w000_mydb
 2937 ?        Ss     0:00 ora_reco_mydb
 2939 ?        Ss     0:00 ora_w001_mydb
 2941 ?        Ss     0:00 ora_lreg_mydb
 2943 ?        Ss     0:00 ora_pxmn_mydb
 2947 ?        Ss     0:02 ora_mmon_mydb
 2949 ?        Ss     0:00 ora_mmnl_mydb
 2951 ?        Ss     0:00 ora_d000_mydb
 2953 ?        Ss     0:00 ora_s000_mydb
 2955 ?        Ss     0:00 ora_tmon_mydb
 2995 ?        Ss     0:00 ora_tt00_mydb
 2997 ?        Ss     0:00 ora_tt01_mydb
 2999 ?        Ss     0:00 ora_tt02_mydb
 3001 ?        Ss     0:00 ora_aqpc_mydb
 3006 ?        Ss     0:00 ora_p000_mydb
 3008 ?        Ss     0:00 ora_p001_mydb
 3010 ?        Ss     0:00 ora_p002_mydb
 3012 ?        Ss     0:00 ora_p003_mydb
 3015 ?        Ss     0:00 ora_cjq0_mydb
 3197 ?        Ss     0:00 ora_qm02_mydb
 3201 ?        Ss     0:00 ora_q002_mydb
 3203 ?        Ss     0:00 ora_q003_mydb
 3323 ?        Ss     0:00 ora_w002_mydb
```

Totuși, să nu uităm că avem întotdeauna la dispoziție dicționarul bazei de date unde putem găsi informații și despre procesele Oracle.

```
SYS@SQL> select program, background from v$process;

PROGRAM                                                          B
---------------------------------------------------------------- -
PSEUDO
```

```
ORACLE.EXE (PMON)                                                1
ORACLE.EXE (PSP0)                                                1
ORACLE.EXE (VKTM)                                                1
ORACLE.EXE (GEN0)                                                1
ORACLE.EXE (MMAN)                                                1
ORACLE.EXE (QM02)                                                1
ORACLE.EXE (DIAG)                                                1
ORACLE.EXE (DBRM)                                                1
ORACLE.EXE (VKRM)                                                1
ORACLE.EXE (DIA0)                                                1

PROGRAM                                                          B
---------------------------------------------------------------- -
ORACLE.EXE (DBW0)                                                1
ORACLE.EXE (LGWR)                                                1
ORACLE.EXE (CKPT)                                                1
ORACLE.EXE (SMON)                                                1
ORACLE.EXE (RECO)                                                1
ORACLE.EXE (LREG)                                                1
ORACLE.EXE (PXMN)                                                1
ORACLE.EXE (MMON)                                                1
ORACLE.EXE (MMNL)                                                1

...
```

Ce înseamnă proces de "background"? Care este rolul proceselor: LGWR, PMON, DBW0, SMON şi CKPT?

Mai departe putem să verificăm câte procese "*background*" şi câte "*foreground*" rulează.

```
SYS@SQL> select decode(background, '', 'FOREGROUND', 'BACKGROUND') tip_proces, count(*)
from v$process group by background;

TIP_PROCES   COUNT(*)
---------- ----------
FOREGROUND          9
BACKGROUND         28
```

Deschideţi o conexiune nouă din SqlPlus folosind utilizatorul SYSTEM şi executaţi din nou interogarea precedentă. Câte procese "foreground" şi câte "background" veţi avea acum? Cum explicaţi?

În general, fiecărui proces din *V$PROCESS* îi corespunde o sesiune Oracle care reprezintă, de fapt, o conexiune la baza de date. Există, desigur, şi excepţii, dar nu intrăm în detalii acum. Fiecare sesiune poate fi identificată în mod unic prin intermediul valorii coloanelor *SID* şi *SERIAL#*. Spre exemplu, putem identifica sesiunea pe care am facut-o anterior cu utilizatorul *SYSTEM*, folosind:

```
SYS@SQL> column program format a20
SYS@SQL> select sid, serial#, status, program from v$session where username = 'SYSTEM';

       SID    SERIAL# STATUS   PROGRAM
---------- ---------- -------- --------------------
        28      14892 INACTIVE sqlplus.exe
```

Un DBA poate decide să închidă necondiționat o sesiune folosind o comandă "ALTER SYSTEM KILL SESSION...", asa cum este exemplificat mai jos. Vom "omorî" sesiunea pe care am identificat-o anterior.

```
SYS@SQL> alter system kill session '28,14892';

System altered.
```

- *Ce s-a întâmplat cu sesiunea de SqlPlus făcută cu utilizatorul SYSTEM? Ce se întâmplă dacă dați o comandă SQL din acea sesiune? Spre exemplu: SELECT 'merge?' from DUAL;*
- *Dați exemple de situații când DBA-ul ar putea interveni pentru a da "kill" la anumite sesiuni.*

Structuri de memorie

La pornirea instanței Oracle, pe langă crearea proceselor despre care am discutat deja, are loc și alocarea de zone de memorie specializate, folosite de serverul Oracle în diferite scopuri. Pe de o parte este creată așa-numita zonă de memorie SGA (System Global Area) care este partajată între procesele instanței și o zona de memorie denumită PGA (Program Global Area) care este privată și asociată unui singur proces. De fapt, global, PGA-ul este suma zonelor de memorie privată a fiecărui proces Oracle.

O primă dovadă că la pornirea instanței Oracle sunt alocate zone de memorie o dă și SqlPlus-ul prin mesajele pe care le afișează la început:

```
ORACLE instance started.

Total System Global Area  838860800 bytes
Fixed Size                  3051184 bytes
Variable Size             599785808 bytes
Database Buffers          230686720 bytes
Redo Buffers                5337088 bytes
```

În primul rând, putem observa că SqlPlus ne spune câtă memoria totală putem aloca pentru SGA. Este de fapt valoarea parametrului SGA_MAX_SIZE.

```
SYS@SQL> show parameter sga_max_size

NAME            TYPE        VALUE
--------------- ----------- -----
sga_max_size    big integer 800M
```

Ca să obţineţi valoarea în bytes va trebui să înmulţiţi de două ori cu 1024.

Unde şi cum este setat parametrul SGA_MAX_SIZE?

Celelalte valori pot fi deduse interogând view-ul sistem *V$SGASTAT*, astfel:

```
SYS@SQL> select bytes from v$sgastat where name = 'fixed_sga';

     BYTES
----------
   3051184

SYS@SQL> select bytes from v$sgastat where name = 'buffer_cache';

     BYTES
----------
 322961408

SYS@SQL> select bytes from v$sgastat where name = 'log_buffer';

     BYTES
----------
   5337088
```

- *Comparaţi valorile obţinute cu cele afişate de SqlPlus. Cum comentaţi?*
- *Care este rolul zonei de memorie "buffer_cache"? Dar a "log buffer"-ului?*

Interogarea dicționarului

Toate informațiile cu privire la structura fizică a bazei de date, alături de alte multe metadate, se regăsesc în dicționarul bazei de date. De altfel, EM Express sau alte utilitare de administrare/monitorizare a bazei de date își trag mare parte din datele afișate, tot din dicționarul bazei de date. Toate aceste metadate sunt fie stocate fizic în tablespace-ul SYSTEM, fiind deținute de utilizatorul SYS, fie sunt ținute doar în memorie, în structuri transiente care se pierd la fiecare oprire a instanței bazei de date.

Nu modificați niciodată obiecte din dicționar! Nu ștergeți, nu adăugați nu scrieți nimic în schema SYS! Citiți de-acolo și nimic mai mult! Alfel, e posibil să compromiteți definitiv respectiva bază de date!

Mai departe vom folosi o consolă deprimantă de CMD în care vom rula un SQLPLUS, la fel de bacovian. SQLPLUS este un utilitar cu care ne putem conecta la baza de date, rula SQL-uri și din care putem administra instanța Oracle, toate la linia de comandă. Deși arată urât, SQLPLUS este un utilitar indispensabil pentru administrarea bazei de date. Nu prea poți spune că ești un Oracle DBA dacă nu știi să folosești SQLPLUS. Așa că vom lucra puțin cu acest utilitar pentru a ne familiariza cu el. Dacă vi se face cumva greață sau simțiți că începe să vă doară capul puteți trece la SQL Developer.

Mai întâi, să vedem cum ne conectăm.

```
C:\Users\talek>sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Feb 19 13:50:56 2017

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect / as sysdba
Connected.

SQL> show user
USER is "SYS"

SQL> connect sys/Test1234@TESTDB as sysdba
Connected.

SQL> show user
USER is "SYS"

SQL> connect system/Test1234@localhost:1521/TESTDB
Connected.

SQL> show user
USER is "SYSTEM"
```

Ce e scris cu litere îngroșate ar trebui să tastați la consolă. Câteva explicații lămuritoare sunt necesare. Prima comandă, "sqlplus /nolog", va lansa utilitarul SQLPLUS dar nu va încerca să se

conecteze la vreo bază de date. De asta i-am şi spus "/nolog", adică "no login". Asta pentru că vrem să experimentăm comanda "connect" din sqlplus. Prima commandă "connect" arată ciudat, în sensul că nu i-am specificat nici utilizatorul, nici parola. Asta pentru că folosim aşa-numita autentificare prin sistemul de operare. Ce înseamnă asta? Înseamnă că, odată autentificaţi în sistemul de operare (în cazul de faţă printr-un cont de Windows), Oracle va presupune că suntem un utilizator valid, deja autentificat şi nu ne va mai cere nici o parolă. Bine, lucrurile sunt puţin mai complicate: utilizatorul acesta de sistem de operare trebuie să fie intr-un anumit grup, etc. etc. Nu intrăm în detalii acum. Sintaxa "as sysdba" este obligatorie ori de câte ori vrem să ne conectăm cu SYS sau cu un alt utilizator de tip SYSDBA. Altfel, Oracle ne va arunca o eroare. Important de reţinut este că autentificarea prin sistemul de operare nu se poate face decât de pe server-ul unde baza de date Oracle rulează. În plus, pentru acest tip de access nu este necesar LISTENER-ul. De fapt, reţeaua este ocolită cu totul.

A doua comandă de conectare foloseşte o autentificare clasică de tip utilizator/parolă. Sintaxa '@TESTDB' indică la ce bază de date vrem să ne conectăm şi este de fapt, ceea ce numim noi, pompos, un descriptor TNS. De obicei, găsiţi acest descriptor în fişierul *ORACLE_HOME/network/admin/tnsnames.ora*. În cazul nostru, acest descriptor arată aşa:

```
TESTDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = TESTDB)
    )
  )
```

Nebunie! Cert este că SQLPLUS va citi respectivul TNS descriptor, va extrage adresa IP şi portul pe care se presupune că ascultă listener-ul, va extrage şi numele bazei de date, după care va iniţia o conexiune.

Dacă nu vrem să ne batem capul cu astfel de descriptori TNS, putem să folosim a treia variantă de conectare, denumită "EASY Connect". "EASY" e un fel de-a spune... Vouă cum vi se pare? Observaţi că mare parte din informaţiile din descriptor (IP, port, nume bază de date) sunt furnizate direct la conectare. Observaţi de asemenea că am folosit utilizatorul SYSTEM pentru care nu este necesară specificarea sintaxei "as sysdba".

Mai departe, haideţi să răspundem la câteva întrebări punctuale, interogând dicţionarul.

Prima întrebare: *care sunt fişierele de control ale bazei de date?*

```
SQL> select name from v$controlfile;

NAME
--------------------------------------------------------------------
C:\APP\TALEK\ORADATA\TESTDB\CONTROL01.CTL
C:\APP\TALEK\FAST_RECOVERY_AREA\TESTDB\CONTROL02.CTL
```

Tabelele care încep cu V$ sunt acele structuri tranziente care-şi ţin datele doar în memorie. Aceste tabele/view-uri pot fi accesate chiar dacă baza de date nu a fost deschisă. Vom discuta mai încolo despre modurile în care o instanţă Oracle poate fi pornită.

Date despre locaţia fişierelor de control putem găsi şi prin alte locuri în dicţionar. De altfel, există multă informaţie redundantă în dicţionarul bazei de date: aceeaşi informaţie stocată în mai multe locuri, dar prezentată diferit (uneori chiar inconsistent). Iată, putem găsi locaţia fişierelor de control folosind şi această interogare:

```
SQL> select value from v$parameter where name = 'control_files';

VALUE
------------------------------------------------------------------------
C:\APP\TALEK\ORADATA\TESTDB\CONTROL01.CTL, C:\APP\TALEK\FAST_RECOVERY_AREA\TESTD
B\CONTROL02.CTL
```

Întrebarea 2: *Unde se află fişierele de date?*

```
SQL> select file_name from dba_data_files;

FILE_NAME
-----------------------------------------------
C:\APP\TALEK\ORADATA\TESTDB\SYSTEM01.DBF
C:\APP\TALEK\ORADATA\TESTDB\SYSAUX01.DBF
C:\APP\TALEK\ORADATA\TESTDB\UNDOTBS01.DBF
C:\APP\TALEK\ORADATA\TESTDB\USERS01.DBF
```

Sau:

```
SQL> select name from v$datafile;

NAME
-------------------------------------------------
C:\APP\TALEK\ORADATA\TESTDB\SYSTEM01.DBF
C:\APP\TALEK\ORADATA\TESTDB\SYSAUX01.DBF
C:\APP\TALEK\ORADATA\TESTDB\UNDOTBS01.DBF
C:\APP\TALEK\ORADATA\TESTDB\USERS01.DBF
```

Întrebarea 3: *Unde se află fişierele redo-log?*

```
SQL> select member from v$logfile;

MEMBER
-----------------------------------------
C:\APP\TALEK\ORADATA\TESTDB\REDO03.LOG
C:\APP\TALEK\ORADATA\TESTDB\REDO02.LOG
C:\APP\TALEK\ORADATA\TESTDB\REDO01.LOG
```

Întrebarea 4: *Care sunt tablespace-urile disponibile în baza de date şi care e tipul acestora?*

```
SQL> column tablespace_name format a15
SQL> select tablespace_name, contents from dba_tablespaces;

TABLESPACE_NAME CONTENTS
--------------- --------------------------
```

```
SYSTEM           PERMANENT
SYSAUX           PERMANENT
UNDOTBS1         UNDO
TEMP             TEMPORARY
USERS            PERMANENT
```

Ce reprezintă un tablespace UNDO?
Dar unul TEMPORARY?

Întrebarea 5: *Ce fișiere de date conține tablespace-ul SYSTEM? Care e dimensiunea acestor fișiere?*

```
SQL> column file_name format a40
SQL> select file_name, bytes/1024/1024 size_mb from dba_data_files where tablespace_name
= 'SYSTEM';

FILE_NAME                                SIZE_MB
---------------------------------------- ----------
C:\APP\TALEK\ORADATA\TESTDB\SYSTEM01.DBF         790
```

Întrebarea 6: *Cât spațiu liber și cât spațiu ocupat se află în tablespace-ul SYSTEM?*

```
SQL> select sum(bytes)/1024/1024 "free space in MB" from dba_free_space where
tablespace_name = 'SYSTEM';

free space in MB
----------------
          5.8125

SQL> select sum(bytes)/1024/1024 "used space in MB" from dba_segments where
tablespace_name = 'SYSTEM';

used space in MB
----------------
        783.1875
```

Verificați dacă cifrele obținute sunt conforme cu ce afișează EM Express. Ar trebui să ne alarmăm că mai sunt doar 5MB liberi în tablespace-ul SYSTEM?

Observați că am folosit de data aceasta niște view-uri sistem care încep cu DBA_*. Rețineți următoarele:

- *toate view-urile sistem care incep cu **DBA_** sunt accesibile administratorilor și contin toate obiectele din baza de date, desemnate de acel view. Spre exemplu, DBA_SEGMENTS conține toate segmentele din baza de date.*
- *view-urile sistem care încep cu **ALL_** conțin toate obiectele la care utilizatorul care interoghează respectivul view are acces. Spre exemplu, ALL_SEGMENTS conține toate segmentele pentru care cel care interoghează ALL_SEGMENTS are drepturi/privilegii.*
- *view-urile sistem care incep cu **USER_** conțin toate obiectele deținute de utilizatorul care interoghează view-ul USER_*.*

În plus, rețineți că nu puteți interoga aceste view-uri dacă baza de date nu este deschisă!