

Az instancia elindítása és leállítása (startup, shutdown)

Ezeket a műveleteket csak rendszeradminisztrátor végezheti, így először connect-álni kell ilyen jogosultságokkal. Ez a következő módok valamelyikével tehető meg.

1. Az op. rendszer ad a usernek ilyen jogot (op. rendszer szinten dba group)
2. Password file használatával és SYSDBA vagy SYSOPER jogosultság adásával
3. INTERNAL-hoz tartozó jelszó megadásával. (régebbi verziókban)

A SYSDBA és SYSOPER rendszer szintű jogosultságok (lásd: system_privilege_map), a DBA pedig egy role (lásd: dba_roles) !!!

Ha az op. rendszer azonosít. (-> REMOTE_LOGIN_PASSWORDFILE=NONE)
CONNECT / AS SYSOPER (vagy CONNECT / AS SYSDBA)

Ha password file-t használunk.

1. Password file létrehozása: orapwd file=<f> password=<pw> entries=<n>
(a kezdeti jelszót és az adminisztrátorok maximális számát kell megadnunk)
 2. REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE
 3. GRANT SYSDBA TO SCOTT (hozzáadja a passw file-hoz scott-ot)
 4. CONNECT scott/tiger@dblink AS SYSDBA
(ez utóbbi esetben belépés után scott alapértelmezett sémája a SYS lesz)
- Ha később újabb usereknek adjuk meg a SYSDBA (v. SYSOPER) jogot, ők is bekerülnek a password fájlba. A password fájlba felvett userek -> V\$PWFILERS

Az adatbázis elindítása (megfelelő jogosultsággal való kapcsolódás után)

1. instancia elindítása (NOMOUNT)
paraméter file beolvasása, SGA lefoglalása, háttérprocesszek elindítása
(az instancia indítható restricted módban is, ilyenkor csak azok tudnak bejelentkezni, akik rendelkeznek a RESTRICTED_MODE jogosultsággal)
2. adatbázis mountolása (MOUNT: az adatbázis hozzárendelése az instanciához)
control file megnyitása
ilyenkor az adatbázis még zárva van, csak a rendszergazda érheti el
3. adatbázis megnyitása (OPEN)
adatfájlok és redo log fájlok megnyitása

Az egyes lépéseket külön-külön is végre lehet hajtani

```
STARTUP ... [ NOMOUNT | MOUNT | OPEN ]  
ALTER DATABASE { MOUNT | OPEN }
```

Az instancia az elindulásakor beolvassa a paraméter állományt, ami többek között a következő információkat tartalmazza:

Adatbázis neve
Memóriaterületek mérete
Kontroll fájl neve és helye
Visszaállítási szegmensek információi
Archiválással kapcsolatos beállítások ... stb.

Kétfajta paraméterállomány létezik, statikus (PFILE) és dinamikus (SPFILE)

PFILE: szerkeszthető szöveges állomány, általában a kliens oldalon van (ha a szerver hozza létre, vagy ebből hoz létre SPFILE-t, akkor persze a szerver oldalon van), csak az instancia elindulásakor érvényesülnek a beállításai.

SPFILE: bináris állomány, a szerver tartja karban, így csak a szerver oldalon lehet, a menet közbeni változtatások (ALTER SYSTEM) is megőrződnek benne.

Mindkét fajta paraméterállományból létre lehet hozni a másik fajtát:

```
CREATE SPFILE [= 'file'] FROM PFILE [= 'file']
```

```
CREATE PFILE [= 'file'] FROM SPFILE [= 'file']
```

Az instancia először az SPFILE-t keresi, utána a PFILE-t, de ez felülbírálható a STARTUP parancsban (PFILE= ...). Ha az SPFILE nem az alapértelmezett helyen van, azt is a STARTUP-ban kell megadni (SPFILE=<file>)

Néhány paraméter menet közben dinamikusan is módosítható. ALTER SESSION, ALTER SYSTEM

```
ALTER SYSTEM SET param = érték [SCOPE = memory | spfile | both] [DEFERRED]
```

SCOPE: változás csak a példányban, csak a fájlban, vagy mindkettőben

DEFERRED: csak az ezután induló session-ökben történik meg a változás

A paraméterek aktuális értékeit az alábbi dinamikus nézetekből nézhetjük meg
v\$parameter az aktuális munkamenet (session) értékeit mutatja
v\$system_parameter az aktuális rendszerbeli értékeket mutatja

Dinamikus nézetek: V\$

A szerver folyamatosan karbantartja őket, mindenféle változást, eseményt rögzítenek. Az adatbázis működése közben folyamatosan változik a tartalmuk. Általában a DBA használja a rendszer figyeléséhez, hangolásához.

Adatbázis létrehozása

Kell hozzá egy hitelesített felhasználó (op. rendszer vagy jelszó állomány által)

Lépések:

Példány és AB nevének meghatározása

Op. rendszer változók beállítása (ORACLE_SID, ORACLE_HOME ...)

Paraméterállomány létrehozása

 PFILE szerkesztése, DB_NAME paraméter megadása kötelező

 SPFILE létrehozása (opcionális)

Jelszó állomány létrehozása (opcionális)

Példány indítás

 SYSDBA-ként bejelentkezve, NOMOUNT-tal indítani

CREATE DATABASE utasítás kiadása (név -> DB_NAME legyen)

Példa egy létrehozó parancsra:

```

CREATE DATABASE oradb
  USER SYS IDENTIFIED BY sys
  USER SYSTEM IDENTIFIED BY system
LOGFILE
  GROUP 1 '/big/oracle/oradata/oradb/redo01.log' SIZE 50M,
  GROUP 2 '/big/oracle/oradata/oradb/redo02.log' SIZE 50M,
  GROUP 3 '/big/oracle/oradata/oradb/redo03.log' SIZE 50M
MAXLOGFILES 6
MAXLOGMEMBERS 5
MAXLOGHISTORY 2
NOARCHIVELOG
MAXDATAFILES 40
MAXINSTANCES 2
CHARACTER SET AL32UTF8
NATIONAL CHARACTER SET AL16UTF16
DATAFILE
  '/big/oracle/oradata/oradb/system01.dbf' SIZE 400M AUTOEXTEND ON
  EXTENT MANAGEMENT LOCAL
SYSAUX DATAFILE
  '/big/oracle/oradata/oradb/sysaux01.dbf' SIZE 400M AUTOEXTEND ON
DEFAULT TABLESPACE users DATAFILE
  '/big/oracle/oradata/oradb/users01.dbf' SIZE 400M AUTOEXTEND ON
DEFAULT TEMPORARY TABLESPACE temp TEMPFILE
  '/big/oracle/oradata/oradb/temp01.dbf' SIZE 400M AUTOEXTEND ON
UNDO TABLESPACE undotbs1 DATAFILE
  '/big/oracle/oradata/oradb/undotbs01.dbf' SIZE 400M AUTOEXTEND ON
SET TIME_ZONE = '+02:00'
;

```

A létrehozott adatbázis tulajdonságait a DATABASE_PROPERTIES nevű adatszótárban nézhetjük meg.

További scriptek futtatása (catalog.sql, catproc.sql)

Az sql scriptek elnevezési szabályai:

cat*.sql: katalógus és adatszótár infók

dbms*.sql: package-ek specifikációi

prvt*.plb: kódolt package törzsek

utl*.sql: táblák és nézetek a segédprogramokhoz (pl. utlexcpt.sql)