

CUPRINS

| | |
|--|----|
| 12. Gestiunea bazei de date | 2 |
| 12.1. Crearea unei baze de date | 2 |
| 12.2. Pornirea unei baze de date | 8 |
| 12.3. Închiderea unei baze de date | 9 |
| 12.4. Starea de repaus a unei baze de date | 11 |
| Bibliografie | 12 |

12. Gestiunea bazei de date

- O bază de date *Oracle* este disponibilă utilizatorilor din momentul deschiderii și până la închiderea ei.
- Un utilizator *Oracle* este o persoană care posedă un cont înregistrat în *server*-ul *Oracle* și pentru care au fost acordate privilegiile corespunzătoare de acces la date.
- Fiecare bază de date are un administrator. Acesta este un utilizator cu privilegii speciale care trebuie:
 - să evalueze modul de folosire a resurselor mașinii pe care va fi instalat sistemul *Oracle* și aplicațiile sale (discuri disponibile pentru sistem și baze de date, *driver*-e dedicate, memoria necesară pentru instanțe);
 - să instaleze sistemul;
 - să creeze planul bazei de date, adică structurile logice de stocare, modelul general al bazei și strategiile de restaurare a acestora (planificarea modului în care structurile logice de stocare ale bazei de date vor afecta funcționarea mașinii pe care rulează sistemul, performanțele bazei în timpul operațiilor de acces la date și eficiența procedurilor de recuperare și restaurare a datelor);
 - să creeze și să deschidă baza de date;
 - să asigure strategii de recuperare a bazei de date în cazul pierderilor;
 - să înregistreze utilizatorii bazei de date și să definească domenii de securitate pentru aceștia;
 - să controleze și să monitorizeze accesul utilizatorilor la baza de date;
 - să optimizeze performanțele bazei de date;
 - să modifice structura bazei, la cererea dezvoltatorilor de aplicații etc.
- Administratorul bazei de date sau utilizatorii posesori ai unui privilegiu *DBA* (*Database Administrator*) pot să pornească sau să oprească baza de date.

12.1. Crearea unei baze de date

- Crearea unei baze de date constă în definirea și implementarea celor două tipuri de structuri (logică și fizică) ce compun baza.
- O bază de date poate fi creată nu numai în momentul instalării sistemului, ci și ulterior.

- Modalitățile de creare a unei baze de date sunt următoarele:
 - în mod automat, la instalarea lui *Oracle Database Server*, utilizând *Oracle Universal Installer*;
 - folosind comanda *CREATE DATABASE* din *SQL*;
 - utilizând instrumentul *Oracle Database Configuration Assistant*;
 - în timpul migrării de la o versiune anterioară a bazei deja existente, folosind *Oracle Data Migration Assistant* etc.
- În continuare sunt prezentați succint pașii care trebuie urmați atunci când baza de date este creată manual.

Pas 1

- Se alege un identificator unic (*SID*) al instanței.
 - În felul acesta, sunt evitate confuziile cu instanțele care vor fi create ulterior și vor rula concurrent cu aceasta.
 - Comanda prin care se setează identificatorul instanței este următoarea:
`%identificator_instanță ORACLE_SID nume_bază;`

Pas 2

- Se stabilesc metodele de autentificare pentru administratorul bazei de date.

Pas 3

- Se creează fișierul parametrilor de inițializare.
 - Acest fișier este necesar pentru definirea caracteristicilor instanței.
 - Denumirea și locația fișierului depinde de platforma pe care rezidă *server-ul Oracle*.
 - De exemplu, în mediul *WINDOWS*, denumirea fișierului parametrilor de inițializare este *init\$ORACLE_SID.ora* și este localizat în directorul *\$ORACLE_HOME\database*.

Pas 4

- Se realizează conectarea la o instanță.
 - De exemplu, aceasta presupune pornirea utilitarului *SQL*Plus* și conectarea la instanța *Oracle* ca *SYSDBA*:
`CONNECT SYS/parola AS SYSDBA;`

Pas 5

- Se pornește instanța fără a monta baza de date.
 - În acest moment este definită zona de memorie *SGA* și sunt pornite procesele *background* necesare creării bazei de date.
 - Comanda are următoarea sintaxă:
`STARTUP NOMOUNT;`

Pas 6

- Se selectează numele global al bazei de date, care reprezintă numele și domeniul acesteia.
 - Numele bazei de date este precizat prin intermediul parametrului de inițializare *DB_NAME*.
 - Valoarea acestui parametru trebuie să fie aceeași cu cea a parametrului *SID*.
- Domeniul bazei de date reprezintă locația acesteia în structura rețelei, fiind specificat prin intermediul parametrului de inițializare *DB DOMAIN*.

Pas 7

- Se creează efectiv baza de date.
 - Aceasta presupune definirea fișierelor de date, de control și de reluare, a spațiului tabel *SYSTEM* și a segmentului de revenire *SYSTEM*, a dicționarului datelor, alegerea setului de caractere pentru stocarea datelor de tip text în baza de date, setarea zonei de timp (*timezone*) pentru bază.
 - Comanda are următoarea sintaxă:

```
CREATE DATABASE [nume_bază]
[CONTROLFILE REUSE]
[LOGFILE GROUP întreg specificație_fișier
                [, specificație_fișier...]
                [, GROUP întreg specificație_fișier
                [, specificație_fișier...]... ] ]
[MAXLOGFILES întreg]
[MAXLOGMEMBERS întreg]
[MAXLOGHISTORY întreg]
[MAXDATAFILES întreg]
[MAXINSTANCES întreg]
[ {ARCHIVELOG | NOARCHIVELOG} ]
[CHARACTER SET set caractere]
```

```
[NATIONAL CHARACTER SET set_caractere]
[DATAFILE specificație_fișier [clauza_autoextend]
      [, specificație_fișier [clauza_autoextend]...] ]
[specificație_spațiu_tabel_implicit]
[specificație_spațiu_tabel_de_revenire]
[SET TIME_ZONE = '{ [+ | -] hh : mi | timp_regiune}' ];
```

- Numele bazei de date poate avea lungimea de cel mult 8 octeți și conține doar caractere *ASCII*. Sistemul scrie acest nume în fișierul de control. Atunci când numele bazei nu este specificat, se folosește parametrul de inițializare *DB_NAME*.
- Clauza *CONTROLFILE REUSE* permite reutilizarea fișierelor de control identificate prin parametrul de inițializare *CONTROL_FILES*. În general, clauza este folosită doar dacă baza de date este creată din nou.
- Specificarea fișierelor de reluare se realizează prin clauza *LOGFILE GROUP*. Fiecărui fișier de reluare declarat prin *specificație_fișier* i se asociază un grup corespunzător, care poate conține unul sau mai mulți membri (copii ale fișierului de reluare respectiv). O bază de date trebuie să conțină cel puțin două grupuri de fișiere de reluare.
- Clauza *MAXLOGFILES* precizează numărul maxim de grupuri de fișiere de reluare care pot fi create pentru baza de date. Valorile implicite, minime și maxime ale acestui parametru depind de sistemul de operare.
- Prin clauza *MAXLOGMEMBERS* se specifică numărul maxim de membri ai unui grup.
- Parametrul *MAXDATAFILES* precizează în fișierul de control, dimensiunea inițială (la momentul creării bazei de date sau a fișierului de control) a zonei alocate pentru fișierele de date.
- Parametrul *MAXINSTANCES* specifică numărul maxim de instanțe simultane corespunzătoare aceleiași baze de date (care este montată și deschisă). Valoarea minimă a acestui parametru este 1, iar valorile maxime și implicite depind de sistemul de operare pe care rulează *Oracle*.
- Atunci când se dorește menținerea unor copii *offline* ale fișierelor de reluare, înainte ca acestea să fie reutilizate, se folosește opțiunea *ARCHIVELOG*. În caz contrar, se precizează opțiunea *NOARCHIVELOG*.

- Parametrul *MAXLOGHISTORY* este util doar dacă baza de date este folosită cu *Real Application Clusters*, în mod *ARCHIVELOG*. Acest parametru specifică numărul maxim de fișiere de reluare arhivate.
- Prin clauza *CHARACTER SET* se precizează setul de caractere *Unicode* folosit de baza de date (*AL32UTF8*, *UTF8*, *UTFE* etc.).
- Clauza *NATIONAL CHARACTER SET* precizează setul de caractere *Unicode* (*AL16UTF16*, *UTF8* etc.) folosit pentru a stoca datele din coloanele de tip *NCHAR*, *NCLOB* sau *NVARCHAR2*.
- Clauza *DATAFILE* specifică fișierele de date din spațiul tabel *SYSTEM*.
- Clauza *SET TIME_ZONE* stabilește zona de timp pentru baza de date. Dacă această clauză este omisă, sistemul *Oracle* încearcă să preia zona de timp a sistemului de operare gazdă.
- Intervalul folosit pentru *hh:mi* este de la -12:00 la +14:00. Specificațiile posibile pentru *timp_regiune* se găsesc în coloana *TZNAME* a vizualizării *V\$TIMEZONE_NAMES* din dicționarul datelor.

Exemplul 12.1

```
CREATE DATABASE baza_depozit
CONTROLFILE REUSE
LOGFILE
    GROUP 1 ('C:/baza_depozit/redolog1.log',
            'D:/baza_depozit/redolog1.log') SIZE 25K
    GROUP 2 ('C:/baza_depozit/redolog2.log',
            'D:/baza_depozit/redolog2.log') SIZE 25K
MAXLOGFILES    6
MAXLOGMEMBERS  2
MAXDATAFILES   50
MAXINSTANCES   1
CHARACTER SET  US7ASCII
DATAFILE 'C:/baza_depozit/datafile1.dbf' AUTOEXTEND ON
UNDO TABLESPACE undo_tbs
DEFAULT TEMPORARY TABLESPACE temp_tbs;
```

Pas 8

- Se creează spații tabel adiționale pentru utilizatori, astfel încât baza de date să devină funcțională.

Exemplul 12.2

Se creează spațiul tabel *users_tbs*, având un fișier de date reutilizabil de dimensiune 200MB autoextensibilă. Dimensiunea următorului spațiu de pe disc ce va fi alocat pentru

extinderea fișierului de date este 1000KB, iar dimensiunea maximă a spațiului de pe disc care poate fi alocat este nelimitată.

```
CREATE TABLESPACE users_tbs  
  DATAFILE 'C:/baza_depozit/users1.dbf'  
  SIZE 200M REUSE  
  AUTOEXTEND ON NEXT 1000K  
  MAXSIZE UNLIMITED;
```

Pas 9

- Se execută *script*-urile de generare a vizualizărilor dicționarului datelor și de instalare a obiectelor folosite de *PL/SQL*.

```
$ORACLE_HOME/rdbms/admin/catalog.sql;
```

```
$ORACLE_HOME/rdbms/admin/catproc.sql;
```

Pas 10

- Se creează fișierul de parametri *server* pentru a putea gestiona dinamic parametrii de inițializare.
 - Acesta este definit pe baza fișierului parametrilor de inițializare.

Exemplul 12.3

```
CREATE SPFILE='/oracle/dbs/spfile_depozit.ora'  
FROM PFILE='/oracle/admin/baza_depozit/pfile/init.ora';
```

Pas 11

- Se dezvoltă strategiile de recuperare a bazei de date.



- ❖ Baza de date poate fi utilizată doar după ce a fost creată și pornită.
- ❖ Orice utilizator cu privilegii de administrare poate opri sau porni baza de date. În general, doar administratorii au aceste privilegii.
- ❖ De asemenea, aceștia sunt singurii utilizatori care pot accesa baza de date atunci când ea este închisă.

12.2. Pornirea unei baze de date

- Presupune crearea unei instanțe asociate și alegerea modului în care baza de date va rula.
- Există mai multe metode de a porni o bază de date, în funcție de utilitarul folosit (*SQL*Plus*, *RMAN*, *OEM*).
- Pornirea unei instanțe presupune pregătirea contextului necesar pentru utilizarea bazei de date (alocarea spațiului de memorie *SGA* și crearea proceselor *background*, acestea nefiind încă asociate bazei).
- O instanță poate fi pornită în mai multe moduri:
 - fără montarea bazei de date (*STARTUP NOMOUNT*)
 - accesul la baza de date nu este permis;
 - acest procedeu se realizează în vederea creării bazei de date sau pentru recrearea fișierelor de control;
 - cu montarea bazei de date, dar fără deschiderea acesteia (*STARTUP MOUNT*)
 - se permite accesul la baza de date numai pentru anumite operații de administrare (recuperarea bazei de date, adăugarea, ștergerea sau redenumirea fișierelor de reluare, redenumirea fișierelor de date etc.);
 - cu montarea și deschiderea bazei de date
 - se permite accesul la date;
 - dacă deschiderea bazei se realizează în mod restrictiv (*STARTUP RESTRICT*), atunci aceasta este disponibilă doar pentru administrator;
 - accesul tuturor utilizatorilor este permis numai dacă baza de date este deschisă în mod nerrestrictiv.
- Pornirea unei instanțe include următoarele etape:
 - citirea fișierului parametrilor de inițializare;
 - alocarea memoriei *SGA*;
 - pornirea proceselor *background*;
 - deschiderea fișierelor istorice (*trace file* și *alert file*).
- Montarea bazei de date include următorii pași:
 - asocierea bazei de date la instanța pornită anterior;
 - localizarea și deschiderea fișierelor de control specificate în fișierul parametrilor de inițializare;

- consultarea fișierelor de control pentru a obține numele și starea fișierelor de date și de reluare (nu se verifică existența acestor fișiere).
- Procesul de deschidere al bazei de date presupune deschiderea fișierelor de date și a fișierelor de reluare. Dacă unul dintre aceste fișiere nu există, atunci sistemul va returna un mesaj de eroare.
 - *Server-ul Oracle* verifică dacă toate fișierele de date și cele de reluare pot fi deschise și, apoi, controlează starea bazei de date. Dacă este necesar, procesul *background SMON* inițiază recuperarea instanței.
 - Deschiderea bazei de date se face cu ajutorul comenzii *STARTUP*.
 - Atunci când instanța este deja pornită, se folosește comanda *ALTER DATABASE OPEN*.
 - Baza de date poate fi deschisă astfel încât să permită doar consultarea datelor (*ALTER DATABASE OPEN READ ONLY*) sau atât citirea, cât și scrierea acestora (*ALTER DATABASE OPEN READ WRITE*).
 - Baza de date poate fi încărcată în mod dedicat sau partajat. Sistemul *Oracle* permite mai multor instanțe să fie asociate concurent unei singure baze de date. Prin setarea parametrului de inițializare *CLUSTER_DATABASE*, baza de date devine disponibilă mai multor instanțe.
 - Comanda pentru pornirea unei instanțe are următoarea sintaxă simplificată:

```
STARTUP  [FORCE]  [RESTRICT]
        [ {OPEN [RECOVER] [bază_de_date]  | MOUNT | NOMOUNT} ] ;
```

 - Clauza *FORCE* permite abandonarea forțată a unei instanțe pornite.
 - Clauza *RECOVER* asigură desfășurarea procesului de recuperare înainte de pornirea bazei.

12.3. Închiderea unei baze de date

- Pentru a iniția operația de închidere a bazei de date se folosește comanda *SHUTDOWN*.
 - Este necesară conectarea ca *SYSDBA* sau *SYSOPER*.
- Utilizatorii care încearcă să se conecteze la baza de date în timp ce închiderea acesteia este în derulare, vor primi mesajul de eroare „*ORA-01090: shutdown in progress - connection is not permitted*“.

- Procesul de închidere a unei baze de date presupune trei etape:
 - închiderea bazei de date
 - datele din *SGA* sunt înregistrate în fișierele de date și în cele de reluire care apoi vor fi închise, iar fișierele de control rămân deschise;
 - demontarea bazei de date
 - baza de date este disociată de instanță și sunt închise fișierele de control;
 - oprirea instanței
 - se eliberează memoria utilizată de *SGA* și se opresc toate procesele *Oracle*, apoi se închid fișierele istorice.
- Închiderea unei baze de date se poate realiza în mai multe moduri:
 - normal;
 - imediat;
 - tranzacțional;
 - renunțare (*abort*).

Închiderea bazei de date în modul normal

- Este opțiune implicită.
- Se realizează prin comanda *SHUTDOWN NORMAL*.
- Aceasta presupune că nu mai sunt permise conexiuni noi.
- Sistemul așteaptă ca toți utilizatorii curenți să se deconecteze, apoi demarează procesul de închidere a bazei de date.
- Baza de date și fișierele de reluire sunt scrise pe disc.
- Repornirea bazei nu necesită recuperarea instanței.

Închiderea bazei de date în modul imediat

- Se realizează prin comanda *SHUTDOWN IMMEDIATE*.
- Se folosește pentru a iniția o operație automată de recuperare a bazei, atunci când este anunțată o întrerupere imediată de curent sau atunci când baza de date, respectiv una dintre aplicațiile sale funcționează anormal și utilizatorii nu se pot deconecta.
- Presupune că nu sunt permise conexiuni sau tranzacții noi.
- Tranzacțiile care nu sunt permanentizate se derulează înapoi (*rollback*).
- Sistemul deconectează automat toți utilizatorii și începe procesul de oprire a bazei de date. Următoarea pornire a bazei nu va necesita recuperarea instanței.

Închiderea bazei de date în modul tranzacțional

- Se realizează prin comanda *SHUTDOWN TRANSACTIONAL*.
- Este utilizată pentru a permite finalizarea tranzacțiilor active, înainte de a se activa procesul de închidere a bazei.
 - Prin urmare, utilizatorul va fi deconectat numai după ce se termină tranzacția curentă a acestuia cu baza de date.
- Nu sunt permise conexiuni sau tranzacții noi.
- Nici în acest caz, următoarea pornire a bazei de date nu va necesita recuperarea instanței.

Închiderea bazei de date în modul renunțare

- Se realizează prin comanda *SHUTDOWN ABORT*.
- Presupune că baza de date se închide instantaneu, prin renunțarea la instanță.
- Nu sunt permise conexiuni sau tranzacții noi.
- Comenzile *SQL* curente nu sunt procesate.
- Tranzacțiile care nu sunt permanentizate, nu se derulează înapoi (*rollback*).
- Sistemul deconectează automat toți utilizatorii.
- Următoarea pornire a bazei de date necesită operația de recuperare a instanței, care se realizează în mod automat.

12.4. Starea de repaus a unei baze de date

- Administratorii bazei trebuie să izoleze uneori anumite acțiuni de cele concurente realizate de utilizatorii obișnuiți (tranzacții, interogări sau comenzi *PL/SQL*).
 - De exemplu, în timpul modificării structurii unui tabel poate fi necesară interzicerea tranzacțiilor concurente care accesează datele acelui tabel.
- Procesul de izolare a bazei de date presupune închiderea acesteia și redeschiderea sa în mod restrictiv (*STARTUP RESTRICT*). În cazul sistemelor care trebuie să funcționeze continuu, oprirea temporară a bazei de date poate cauza probleme importante.
- *Oracle9i* a introdus posibilitatea de izolare a bazei de date prin trecerea sa în stare de repaus, fără a deconecta utilizatorii. În acest fel, administratorul poate întreprinde în siguranță acele acțiuni a căror execuție cere izolare față de utilizatorii obișnuiți. De asemenea, se mărește disponibilitatea pentru sistemele care trebuie să funcționeze continuu.

Bibliografie

1. Connolly T.M., Begg C.E., *Database Systems: A Practical Approach to Design, Implementation and Management*, 5th edition, Pearson Education, 2005
2. Dollinger R., Andron L., *Baze de date și gestiunea tranzacțiilor*, Editura Albastră, Cluj-Napoca, 2004
3. Oracle and/or its affiliates, *Oracle Database Concepts*, 1993, 2021
4. Oracle and/or its affiliates, *Oracle Database Performance Tuning Guide*, 2013, 2023
5. Oracle and/or its affiliates, *Oracle Database SQL Language Reference*, 1996, 2023
6. Oracle and/or its affiliates, *Oracle Database PL/SQL Language Reference*, 1996, 2023
7. Oracle and/or its affiliates, *Oracle Database Administrator's Guide*, 2001, 2023
8. Oracle and/or its affiliates, *Pro *C/C++ Programmer's Guide*, 1996, 2014
9. Oracle University, *Oracle Database 11g: PL/SQL Fundamentals, Student Guide*, 2009
10. Popescu I., Alecu A., Velcescu L., Florea (Mihai) G., *Programare avansată în Oracle9i*, Ed. Tehnică, 2004