

Capitolul 3

Fisiere: Control, Redo Log

Continut capitol

Ca structura fizica, baza de date contine fisiere de control, de date si de Redo log.

Ca structura logica o baza de date se compune din:

Tablespace \supset Segment \supset Extent \supset Bloc (stocate in fisierele de date)

In acest capitol vom discuta despre:

- Fisierele de control
- Fisierele de Redo Log

In capitolul urmator vor fi detaliate:

- Tablespace (element in structura logica a fisierele de date)
- Fisierele de date

Fisier de control

- ◆ Este un fisier binar, folosit atat la pornirea bazei de date cat si in timpul operarii cu aceasta.
- ◆ Este deschis si citit la montarea bazei de date (vezi cap. precedent) pentru a localiza fisierele de date si fisierele de Redo log.
- ◆ Este actualizat permanent si trebuie sa fie disponibil pe intreaga perioada in care baza de date este montata si deschisa.

Fisier de control - cont

- ◆ Contine informatii de consistenta necesare cand se face restaurarea dupa incident
- ◆ Daca vreunul dintre fisierele de control nu este disponibil functionarea bazei de date este afectata.

Continut fisier de control

- ◆ Numele bazei de date
- ◆ Numele si localizarea fisierele de date si de Redo log
- ◆ Numele pentru tablespace-uri
- ◆ Eticheta timp (timestamp) de la crearea bazei de date
- ◆ Numarul de secventa pentru fisierele Redo log
- ◆ Informatii despre punctele de checkpoint
- ◆ Etc.

Important

- ◆ Montarea bazei de date (si ulterior deschiderea) se poate face doar daca fisierele de control sunt disponibile.
- ◆ In caz contrar, desi toate celelalte fisiere (date, Redo log) pot exista si pot fi consistente, baza de date nu poate trece de pasul MOUNT si sunt necesare scenarii de restaurare (din salvari ale fisierele de control de exemplu – vom vedea tot azi cum).

Important - cont

- ◆ Teoretic este posibil sa avem un singur fisier de control dar aceasta varianta este descurajata de Oracle.
- ◆ Varianta optima este de a avea mai multe copii exploatare in oglinda (terminologia Oracle: multiplexate) chiar si in cazul in care masina gazda are un singur disc fizic
- ◆ In felul acesta este prevenita stergerea accidentala - exemple de acest fel sunt destule

F. Radulescu, Curs: Utilizarea bazelor de date, anul IV CS.

7

Parametrii

- ◆ In fisierul de parametri - init.ora - exista parametrul CONTROL_FILES care primeste ca valoare o multime de nume de fisiere de control (care vor fi exploatare in oglinda):

```
CONTROL_FILES =  
  (/u01/oracle/prod/control01.ctl,  
   /u02/oracle/prod/control02.ctl,  
   /u03/oracle/prod/control03.ctl)
```

- ◆ Aceste fisiere sunt create automat de Oracle la crearea bazei de date

F. Radulescu, Curs: Utilizarea bazelor de date, anul IV CS.

8

Mutare / redenumire

- ◆ Pentru a redenumi/muta un fisier de control trebuie urmati urmatorii pasi:
 1. Oprire (Shutdown) baza de date.
 2. Copierea unui fisier de control existent in noua locatie / sub noul nume folosind comenzile SO
 3. Editarea fisierului de parametri de initializare si schimbarea corespunzatoare a parametrului CONTROL_FILES.
 4. Repornirea bazei de date.

F. Radulescu, Curs: Utilizarea bazelor de date, anul IV CS.

9

Creare nou fisier de control

Crearea unui nou fisier de control poate fi necesara in cazurile:

1. Toate fisierele de control ale BD sunt distruse si nu exista salvari ale lor.
2. Se doreste schimbarea unor parametri permanenti ai BD (specificati la CREATE DATABASE) cum ar fi numele bazei de date sau valoarea parametrilor MAXLOGFILES, MAXLOGMEMBERS, MAXLOGHISTORY, MAXDATAFILES si MAXINSTANCES.

F. Radulescu, Curs: Utilizarea bazelor de date, anul IV CS.

10

Cererea CREATE CONTROLFILE

```
CREATE CONTROLFILE  
SET DATABASE stud  
LOGFILE  
  GROUP 1 ('/dsk1/oracle/stud/redo01_01.log',  
           '/dsk1/oracle/stud/redo01_02.log'),  
  GROUP 2 ('/dsk1/oracle/stud/redo02_01.log',  
           '/dsk1/oracle/stud/redo02_02.log'),  
  GROUP 3 ('/dsk1/oracle/stud/redo03_01.log',  
           '/dsk1/oracle/stud/redo03_02.log')  
RESETLOGS  
DATAFILE '/dsk1/oracle/stud/system01.dbf' SIZE 3M,  
         '/dsk1/oracle/stud/rbs01.dbs' SIZE 5M,  
         '/dsk1/oracle/stud/users01.dbs' SIZE 5M,  
         '/dsk1/oracle/stud/temp01.dbs' SIZE 5M  
MAXLOGFILES 50  
MAXLOGMEMBERS 3  
MAXLOGHISTORY 400  
MAXDATAFILES 200  
MAXINSTANCES 6  
ARCHIVELOG;
```

Se presupune ca baza de date exista si anterior dar cu alt nume.

F. Radulescu, Curs: Utilizarea bazelor de date, anul IV CS.

11

Etape creare nou FC

Pasul 1. Se face o lista cu TOATE fisierele de date si Redo Log ale bazei de date.

Cat timp BD este functionala aceste fisiere se pot obtine si ca rezultat al cererilor:

```
SELECT MEMBER FROM V$LOGFILE;  
SELECT NAME FROM V$DATAFILE;  
SELECT VALUE FROM V$PARAMETER  
  WHERE NAME = 'control_files';
```

Atentie: omiterea vreunui fisier atunci cand se va executa CREATE CONTROLFILE poate duce la pierderi iremediabile de date sau ale intregii baze de date (de exemplu daca nu se specifica fisierul in care se afla tablespace-ul pentru SYSTEM).

F. Radulescu, Curs: Utilizarea bazelor de date, anul IV CS.

12

Etape creare nou FC - cont

- Pasul 2. Oprirea bazei de date. De preferat ca aceasta oprire sa fie una normala (si nu IMMEDIATE sau ABORT)
- Pasul 3. Salvarea tuturor fisierelor de date si de control (cele identificate la pasul 1)
- Pasul 4. Se porneste o instanta dar nu se monteaza si nu se deschide baza de date (STARTUP NOMOUNT)
- Pasul 5. Se creeaza noul fisier de control cu cererea SQL CREATE CONTROLFILE (exemplu in slide anterior). Se foloseste clauza RESETLOGS daca s-a redenumit baza sau s-au pierdut si fisiere de Redo log o data cu cele de control. Altfel se foloseste NORESETLOGS
- Pasul 6. Se salveaza undeva in siguranta fisierul creat (ex.: pe un CD)

Etape creare nou FC - cont

- Pasul 7. Se editeaza fisierul cu parametri de initializare pentru a contine numele noilor fisiere (specificate la pasul 5). Daca s-a redenumit si baza de date, se modifica si parametrul DB_NAME
- Pasul 8. Se recupereaza dupa incident – recovery - (daca este cazul) baza de date
- Pasul 9 – si ultimul – se deschide baza de date cu:
ALTER DATABASE OPEN;
sau cu
ALTER DATABASE OPEN RESETLOGS;
in functie de optiunea RESETLOGS absenta sau prezenta la crearea fisierului de control (pasul 5)

Salvare FC

- Se face cu:
ALTER DATABASE BACKUP CONTROLFILE
Exemple:
1. In cazul in care se doreste salvarea unei copii (fisiere binare) a fisierelor de control putem da comanda:
ALTER DATABASE BACKUP CONTROLFILE TO
'/oracle/backup/control.bkp';
2. In cazul in care se doreste producerea unei secvente de cereri SQL (text deci) care sa poata fi ulterior folosite pentru recrearea fisierului de control:
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
Efectul: in fisierul TRACE vor fi generate comenzile SQL respective

Restaurare din copie

- Cazul 1. In cazul in care exista mai multe copii (multiplexate) si doar una din ele s-a distrus:
- Cu instanta oprita se copiaza fizic (comenzi OS) un fisier bun peste cel care a fost distrus:
- ```
cp /dsk3/oracle/stud/control03.ctl
/dsk2/oracle/stud/control02.ctl
```
2. Se reporneste (STARTUP)

## Restaurare din copie - cont

- Cazul 2: dispozitivul pe care se gaseste FC este avariata permanent
- Cu instanta oprita (!) se copiaza un fisier (bun) de control la o locatie accesibila:  
cp /dsk3/oracle/stud/control03.ctl  
/dsk14/oracle/stud/control02.ctl
  - Se modifica in fisierul de parametri de initializare CONTROL\_FILES astfel incat sa se inlocuiasca locatia defecta cu cea noua (fisierul inaccesibil cu cel obtinut la pasul 1)
  - Se reporneste (STARTUP)

## Stergerea unui FC

- Observatie: orice BD trebuie sa aiba permanent 2 FC.
- Se pot sterge fisiere cand sunt mai multe sau cand se creeaza noi fisiere care le inlocuiesc pe altele mai vechi (de exemplu ca in cazul anterior cand un dispozitiv existent iese din uz).
- Pasii sunt:
  - Oprire baza
  - Editare fisier de parametri pentru a elimina din CONTROL\_FILES fisierul care se sterge
  - Repornire.

## Vederi care contin date despre FC

1. V\$DATABASE – contine date despre baza de date (luate din FC)
2. V\$CONTROLFILE, V\$PARAMETER – contin lista cu numele FC

Mai exista si alte vederi care returneaza date privind continutul FC.

## REDO LOG FILES

- ◆ Li se mai spune si 'fisier jurnal' in cazul altor sisteme de gestiune.
- ◆ In ele se inregistreaza toate modificarile facute in Buffer Cache
- ◆ Se folosesc pentru recuperarea tranzactiilor comise ale caror date nu au fost insa scrise pe disc pana in momentul incidentului (deci se folosesc DOAR pentru recuperarea datelor – recovery).

## Grupuri si membri

- ◆ Fisierele de tip Redo Log folosite la un moment dat de sistem sunt impartite in grupuri, un grup putand contine mai multi membrii.
- ◆ Este de preferat ca membrii unui grup sa fie plasati pe dispozitive diferite pentru a evita pierderi in caz de incident de dispozitiv.
- ◆ Procesul care scrie aceste fisier este LGWR (Log Writer, unul dintre procesele de background ale unei instante)

## Grupuri si membri – cont

- ◆ Toti membrii (=fisier) unui grup au dimensiune identica si sunt scrisi in paralel de LGWR
- ◆ Toti membrii unui grup au acelasi numar de secventa (log sequence number). Acesta este un numar dat de Oracle atunci cand incepe sa scrie intr-un grup.
- ◆ Numarul de secventa curent (este unul singur!) este memorat si in fisierele de control si in antetul fisierele de date.

## Cate grupuri sunt

- ◆ Pentru operarea normala a bazei de date Oracle are nevoie de minimul 2 grupuri de fisier Redo Log.
- ◆ Pot fi maximul 255 de grupuri diferite.

Observatie: Termenul in engleza este 'online redo log file' pentru ca mai pot exista si fisier de acest tip arhivate (vom vedea in continuare).

## PARAMETRI

- ◆ In CREATE DATABASE se folosesc urmatoorii parametri:
  - MAXLOGFILES – numarul maxim de grupuri (asa cum am spus e <=255)
  - MAXLOGMEMBERS – numarul maxim de membrii per grup
- ◆ In fisierul de parametri exista LOG\_FILES care specifica numarul de fisier care sunt deschise la run-time (si care poate fi maxim egal cu produsul celor 2 valori maxime de mai sus).

## Utilizare

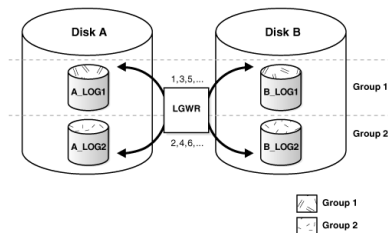
- ◆ Oracle inregistreaza secvential toate schimbarile facute in baza de date in Redo Log Buffer sub forma unor inregistrari de Redo.
- ◆ Bufferul este folosit circular (cand se ajunge la capatul sau se reia cu inceputul).
- ◆ Din Buffer aceste inregistrari sunt scrise in fisierul (=grupul) curent de Redo de catre procesul LGWR in urmatoarele cazuri:

## Utilizare - cont

- ◆ In momentul in care apare un COMMIT (s-a vorbit de asta la un capitol precedent)
- ◆ In momentul in care Redo log buffer este plin intr-o anumita proportie
- ◆ Cand apare un time-out al LGWR (la fiecare 3 secunde)
- ◆ Inainte ca DBWR sa scrie blocurile de date modificate in fisierele de date

## Schimbare grup Redo

- ◆ Cand un fisier (=grup) se umple se trece la urmatorul fisier (sunt cel putin 2)



## Schimbare grup Redo - cont

- ◆ Aceasta schimbare se numeste in documentatia Oracle 'Log switch'
- ◆ DBA-ul poate forta log switch-uri si in cazul in care fisierul curent nu e plin
- ◆ La fiecare switch Oracle asociaza un nou numar de secventa noului fisier (=grup)
- ◆ Cand apare un log switch este de asemenea initiat si un checkpoint

## Checkpoint

- ◆ La aparitia unui checkpoint se executa urmatoarele operatii
  - Toate blocurile de date modificate (dirty buffers) din memorie care sunt monitorizate de fisierul de Redo Log respectiv sunt scrise pe disc de catre DBWR
  - Procesul de checkpoint (CKPT) actualizeaza antetele tuturor fisierele de control si date pentru a reflecta schimbarea produsa.

## Checkpoint - cont

- ◆ Un checkpoint apare:
  - La fiecare log switch
  - La oprirea instantei in modurile normal, tranzactional si imediat
  - In mod forat prin setarea parametrilor: LOG\_CHECKPOINT\_INTERVAL si LOG\_CHECKPOINT\_TIMEOUT
  - Cand e cerut de administratorul bazei de date
- ◆ Informatiile despre fiecare checkpoint sunt stocate in fisierul de alerte (dar doar daca parametrul LOG\_CHECKPOINTS\_TO\_ALERT este setat pe TRUE)

## Clasificare

- ◆ Fisierele de Redo Log se pot clasifica in
  - CURRENT - Curent – cel in care se scrie la un moment dat
  - ACTIVE - Activ – s-a scris in el anterior dar modificarile cuprinse in el nu sunt inca scrise in fisierele de date si deci e necesar pentru recuperare instantata.
  - INACTIVE - Inactiv – s-a scris in el anterior si modificarile s-au inregistrat si in fisierele de date, deci nu mai e necesar pentru recuperare instantata
  - UNUSED – Nou – Nu s-a scris niciodata in el pana acum (probabil un fisier nou adaugat)

## Arhivare

- ◆ Se poate dispune ca fisierele de tip Redo Log File sa fie arhivate de sistemul de gestiune
- ◆ Arhivarea acestor fisiere permite refacerea bazei de date de la 0 pornind de la o salvare la un moment dat si de la fisierele de tip Redo Log care au inregistrat modificarile facute in BD dupa salvarea respectiva.
- ◆ In cazul in care fisierele nu sunt arhivate este posibila in continuare recuperarea instantei dupa incident (pentru asta sunt necesare doar fisierele curente si cele active de Redo Log)

## Arhivare - cont

- ◆ O aceeași baza de date poate fi la un moment dat într-unul din cele 2 moduri:
  - NOARCHIVELOG – în momentul în care ultimul fisier de Redo Log se umple se revine la primul care este rescris (scris-peste). Bineînțeles modificările din acesta au fost scrise și în fisierele de date
  - ARCHIVELOG – după un log switch procesele ARCH (de background) arhivează fisierele de log inactive.

## Vedeti: v\$logdatabase

- ◆ Pentru a vizualiza modul în care este BD: vederea v\$logdatabase

```
select name, log_mode
from v$logdatabase;
```

| NAME | LOG_MODE     |
|------|--------------|
| STUD | NOARCHIVELOG |

## Vedeti: v\$instance

- ◆ Pentru a vizualiza modul în care este BD: vederea v\$instance

```
select archiver
from v$instance;
```

|          |
|----------|
| ARCHIVER |
| STOPPED  |

## Parametru

- ◆ Se poate și astfel (pentru a vedea dacă procesele de arhivare sunt sau nu pornite):

```
SQL> show parameter log_archive_start
```

| NAME              | TYPE    | VALUE |
|-------------------|---------|-------|
| log_archive_start | boolean | FALSE |

## Vederi: v\$thread

- ◆ Pentru a vizualiza care este fisierul (grupul) curent:

```
select groups,current_group#, sequence#
from v$thread;
```

| GROUPS | CURRENT_GR | SEQUENCE# |
|--------|------------|-----------|
| 2      | 1          | 457       |

F. Radulescu. Curs: Utilizarea  
bazelor de date, anul IV C5.

37

## Vederi: v\$log

- ◆ Informatiile returnate sunt din fisierele de control:

```
select group#,sequence#, bytes, members,
status
from v$log;
```

| GROUP# | SEQUENCE# | BYTES   | MEMBERS | STATUS  |
|--------|-----------|---------|---------|---------|
| 2      | 457       | 1023459 | 1       | CURRENT |

F. Radulescu. Curs: Utilizarea  
bazelor de date, anul IV C5.

38

## Vederi: v\$logfile

- ◆ Coloana STATUS are valoarea NULL daca fisierul este utilizat si INVALID, STALE sau DELETED altfel.

```
select *
from v$logfile;
```

| GROUP# | STATUS | MEMBER          |
|--------|--------|-----------------|
| 1      |        | /DSK01/log1.rdo |
| 2      |        | /DSK03/log2.rdo |

F. Radulescu. Curs: Utilizarea  
bazelor de date, anul IV C5.

39

## Fortarea unui log switch

- ◆ Se face cu cererea:

```
ALTER SYSTEM SWITCH LOGFILE;
```

Aceasi operatie se poate efectua si din consola de administrare

F. Radulescu. Curs: Utilizarea  
bazelor de date, anul IV C5.

40

## Fortarea unui checkpoint

- ◆ Se face cu cererea:

```
ALTER SYSTEM CHECKPOINT;
```

Precum si din consola de administrare

- ◆ De asemenea, se poate comanda (in cazul in care baza de date foloseste fisiere de tip Redo Log mari) si efectuarea periodica a checkpointului prin parametrii de care am amintit:

```
LOG_CHECKPOINT_INTERVAL si
LOG_CHECKPOINT_TIMEOUT
```

F. Radulescu. Curs: Utilizarea  
bazelor de date, anul IV C5.

41

## LOG\_CHECKPOINT\_INTERVAL

- ◆ Un checkpoint este initiat dupa ce LGWR a scris un numar de blocuri egal cu acest parametru (blocuri OS!)
- ◆ Cum orice log switch produce de asemenea checkpoint, daca parametrul este mai mare decat fisierul de Log checkpointul se va face doar la log switch
- ◆ Daca parametrul este 0 este ignorat (Oracle 10g). Aceasta este valoarea de default.

F. Radulescu. Curs: Utilizarea  
bazelor de date, anul IV C5.

42

## LOG\_CHECKPOINT\_TIMEOUT

- ◆ Este o valoare specificata in secunde
- ◆ Se masoara de la inceputul precedentului checkpoint
- ◆ Valoarea 0 dezactiveaza declansarea de checkpoint-uri pe baza intervalelor de timp
- ◆ Valoarea de default este de 1800 (in Oracle 10g) deci 30 minute.
- ◆ Garanteaza ca nici un bloc modificat (dirty bloc) nu ramane in memorie mai mult de atatea secunde cat arata parametrul.

## Adaugare grup Redo Log

- ◆ Se face cu ALTER DATABASE:
- ```
ALTER DATABASE ADD LOGFILE GROUP 10
('/dsk01/oracle/dbs/log10a.rdo',
'/dsk04/oracle/dbs/log10b.rdo') SIZE
500K;
```
- ◆ Specificarea numarului de grup este optionala (doar cand dorim sa le creem in alta ordine decat cea normala).
 - ◆ Nu este bine sa creem grupuri ca 10, 20, 30, ... (pe sarite) pentru ca vom consuma inutil spatiu in fisierele de control

Adaugare membrii

- ◆ Se face cu ALTER DATABASE. Exemplu:
- ```
ALTER DATABASE ADD LOGFILE MEMBER
'/dsk01/oracle/dbs/log2b.rdo'
TO GROUP 1,
'/dsk07/oracle/dbs/log2b.rdo'
TO GROUP 3;
```
- ◆ Se poate specifica grupul si prin membrii sai:
- ```
ALTER DATABASE ADD LOGFILE MEMBER
'/dsk06/oracle/dbs/log10c.rdo'
TO
('/dsk01/oracle/dbs/log10a.rdo',
'/dsk04/oracle/dbs/log10b.rdo');
```

Adaugare membrii - cont

- ◆ Daca fisierul adaugat exista deja pe discul respectiv trebuie sa aiba dimensiunea necesara si se va specifica in plus clauza REUSE:

```
ALTER DATABASE ADD LOGFILE MEMBER
'/dsk01/oracle/dbs/log2b.rdo' REUSE
TO GROUP 1;
```

Redenumire / relocare

- ◆ In cazul acesta trebuie sa ne asiguram ca noul fisier (cu noul nume sau din noua locatie) exista.
- ◆ Oracle nu schimba decat informatii din fisierele de control fara sa redenumiasca sau sa creeze fizic fisiere din SO
- ◆ Etapele relocarii unui fisier de Redo Log sunt urmatoarele (valabile si la redenumire):

Redenumire / relocare - cont

1. Oprirea bazei de date (SHUTDOWN)
2. Copierea fisierele Redo Log in noua locatie. Se face cu comenzi OS.

Exemplu:

```
mv /diska/logs/log1a.rdo /diskc/logs/log1c.rdo
mv /diska/logs/log2a.rdo /diskc/logs/log2c.rdo
```

3. Repornire in modul MOUNT:

```
CONNECT / as SYSDBA
```

```
STARTUP MOUNT
```


Redenumire / relocare - cont

4. Redenumirea (in sistem) a fisierelor:

```
ALTER DATABASE RENAME FILE
'/diska/logs/log1a.rdo', '/diska/logs/log2a.rdo'
TO
'/diskc/logs/log1c.rdo', '/diskc/logs/log2c.rdo';
```

5. Deschiderea bazei:

```
ALTER DATABASE OPEN
```

Stergere grup

- ◆ Trebuie sa ramana cel putin 2 grupuri (nu se poate sterge mai mult de atat)
- ◆ Un grup se poate sterge doar daca e INACTIVE.
- ◆ Daca se doreste stergerea grupului curent trebuie fortat un log switch.
- ◆ Grupul trebuie sa fie arhivat (daca arhivarea e pornita).
- ◆ Pentru a vedea starea grupurilor putem utiliza comanda:

Stergere grup - cont

```
SELECT GROUP#, ARCHIVED, STATUS
FROM V$LOG;
```

GROUP#	ARC	STATUS
1	YES	ACTIVE
2	NO	CURRENT
3	YES	INACTIVE
4	YES	INACTIVE

Stergere grup - cont

- ◆ Stergerea efectiva se face cu ALTER DATABASE:

```
ALTER DATABASE DROP LOGFILE GROUP 3;
```
- ◆ Grupul se poate da nu numai ca numar ci si prin lista membrilor sai)
- ◆ Operatia de stergerea nu implica neaparat stergerea fisierelor de pe disc ci doar actualizarea informatiilor interne ale sistemului prin eliminarea grupului respectiv.
- ◆ Dupa terminarea operatiei Oracle, putem sa utilizam comenzi SO pentru stergerea efectiva a fisierelor respective.

Stergere membri

- ◆ Se poate sterge un membru doar daca nu este in grupul curent sau intr-un grup activ.
- ◆ Este bine ca grupul respectiv sa fie in acel moment arhivat
- ◆ Nu putem sterge ultimul membru valid al unui grup daca in felul acesta nu raman cel putin 2 grupuri continand membrii valizi.
- ◆ In cazul in care grupurile au cate doi membrii de exemplu, se poate sterge un membru din unul dintre ele dar este bine ca ulterior grupul sa fie completat (pentru siguranta in functionare)

Stergere membri - cont

- ◆ Comanda este:

```
ALTER DATABASE
DROP LOGFILE MEMBER
'/oracle/dbs/log3c.rdo';
```
- ◆ Operatia de stergerea nu implica neaparat stergerea fisierului de pe disc ci doar actualizarea informatiilor interne ale sistemului
- ◆ Dupa terminarea operatiei Oracle, putem sa utilizam comenzi SO pentru stergerea efectiva a fisierului respective.

CLEAR LOGFILE

- ◆ In cazul in care un grup devine corupt el poate fi reinitializat cu comanda:

```
ALTER DATABASE CLEAR LOGFILE GROUP 3;
```

- ◆ Grupul se poate da ca numar sau ca lista de membri (intre paranteze).

- ◆ In cazul in care fisierul nu a fost arhivat trebuie mentionat in comanda:

```
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 3;
```

- ◆ In acest caz insa recuperarea bazei din salvare+Loguri nu mai e posibila pentru salvarile care ar folosi si logul astfel initializat.

Lecturi obligatorii

1. Oracle Database Administrator's Guide – Cap 5:
Managing Control Files

http://download.oracle.com/docs/cd/B14117_01/server.101/b10739/control.htm

2. Oracle Database Administrator's Guide – Cap 5:
Managing the Redo Log

http://download.oracle.com/docs/cd/B14117_01/server.101/b10739/onlineredo.htm

Sfârșitul capitolului 3