

# ADMINISTRAREA BAZELOR DE DATE

Titulari: Florin Radulescu,  
Alexandru Boicea

Email: [florin.radulescu@upb.ro](mailto:florin.radulescu@upb.ro)  
[alexandru.boicea@upb.ro](mailto:alexandru.boicea@upb.ro)

**SITE: [CURS.UPB.RO](http://CURS.UPB.RO)**

# NOTARE

- ❑ 60% IN CURSUL SEMESTRULUI:
  - ❑ PREZENTA CURS 10%
  - ❑ PREZENTA, ACTIVITATE SI TEST LABORATOR 35%
  - ❑ LUCRARE LA MIJLOCUL SEMESTRULUI FARA DEGREVARE (SAPTAMANA 8-10): 15%. ACEASTA LUCRARE NU SE POATE REFACE IN SESIUNEA DE EXAMENE
- ❑ 40% VERIFICARE FINALA (EXAMEN)

# REGULI DE TRECERE

□ 50% din punctajul din timpul semestrului (30 pct. din 60)

ȘI

□ 50% din punctajul de la examen (20 pct. din 40)

# Nelamuriri?

# BIBLIOGRAFIE

- ❑ Va fi indicata la fiecare capitol
- ❑ Pantru fiecare capitol, la bibliografie exista documente care aprofundeaza ceea ce s-a predat la curs si care **fac parte integranta** din materia pentru lucrarea de la mijlocul semestrului si din cea pentru examen.

# Incepem?

# Administrarea bazelor de date DEFINITII

Definitii care se pot gasi in Internet:

- ❑ A technical function that is responsible for
  - ❑ physical database design
  - ❑ security enforcement,
  - ❑ database performance,
  - ❑ backup and recovery.

# DEFINITII (2)

- ❑ An area of IT that
    - ❑ develops,
    - ❑ implements,
    - ❑ updates,
    - ❑ tests, and
    - ❑ repairs
- a company's server database.



# DEFINITII (3)

- ❑ Database Administration involves the overall design and management of the database. Administration tasks include
  - ❑ archiving,
  - ❑ consistency checks,
  - ❑ developing/maintaining indexing and retrieval functionality,
  - ❑ migration,
  - ❑ monitoring,
  - ❑ performance issues,
  - ❑ replication issues, and
  - ❑ database sizing/space management.

# JOB PROFILE

- ❑ A database administrator (DBA) is an IT professional responsible for the integrity, performance and security of an organization databases.
- ❑ The role includes the development and design of database strategies, system monitoring and improving database performance and capacity, and planning for future expansion requirements.
- ❑ They may also plan, co-ordinate and implement security measures to safeguard the database

(sursa: wikipedia)

# JOB PROFILE - cont

## Duties:

- ☐ Installing and upgrading the database server and application tools
- ☐ Allocating system storage and planning future storage requirements for the database system
- ☐ Modifying the database structure, as necessary, from information given by application developers
- ☐ Enrolling users and maintaining system security
- ☐ Ensuring compliance with database vendor license agreement
- ☐ Controlling and monitoring user access to the database
- ☐ Monitoring and optimizing the performance of the database
- ☐ Planning for backup and recovery of database information
- ☐ Maintaining archived data
- ☐ Backing up and restoring databases
- ☐ Contacting database vendor for technical support
- ☐ Generating various reports by querying from database as per need  
(sursa: wikipedia)

# CU CE INCEPEM?

## Administrare ORACLE

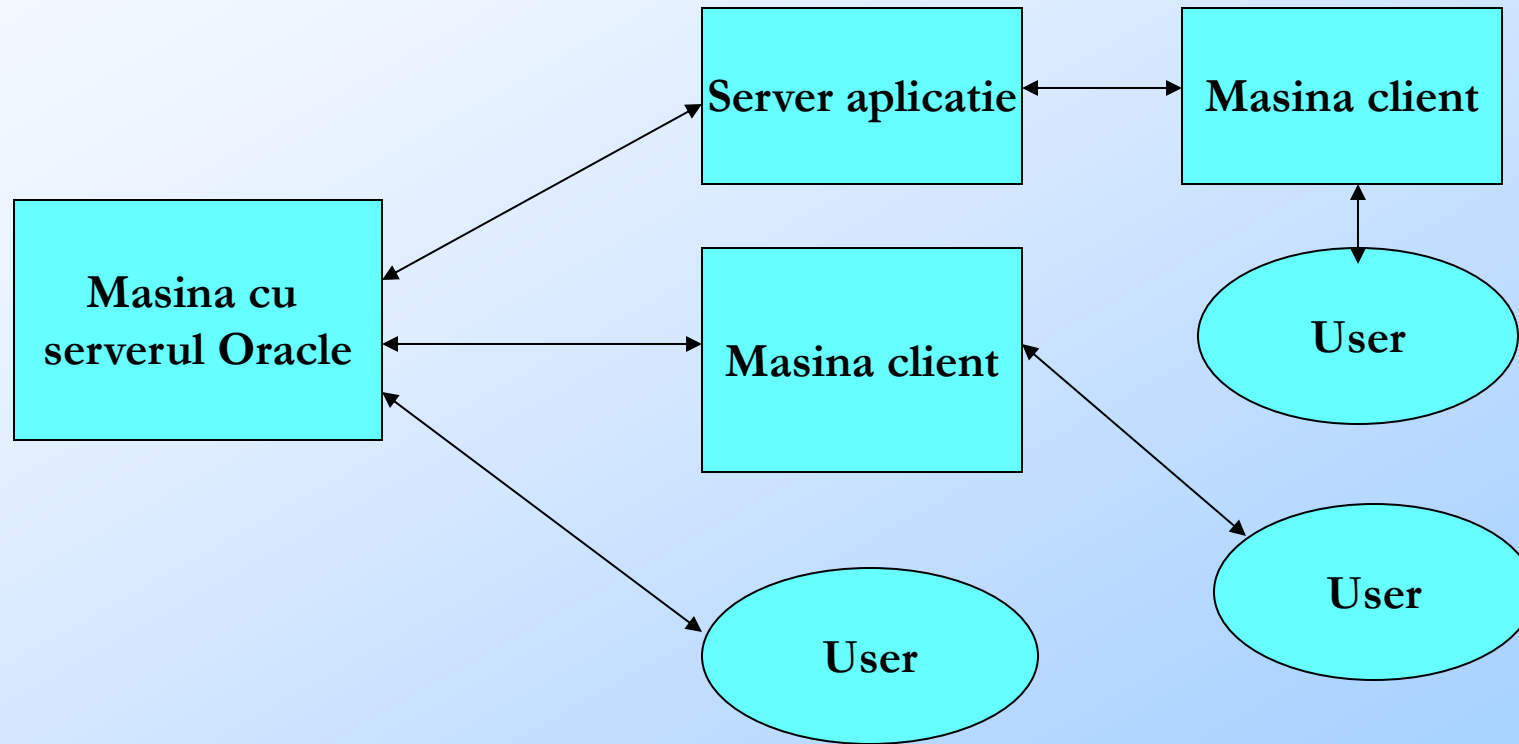
# Capitolul 1

## Arhitectura Oracle

# Serverul ORACLE

- ❑ Este un sistem de gestiune a bazelor de date relationale
- ❑ Userul poate lucra:
  - ❑ Cu un client pe aceeași masina cu serverul (de exemplu un client SQL\*Plus rulant pe aceeași masina cu serverul Oracle)
  - ❑ Clientul ruleaza pe o altă masina decât serverul (two-tiered)
  - ❑ Aplicatia userului acceseaza o altă aplicatie iar la randul ei aceasta e în comunicatie cu serverul (three-tiered)

# Serverul ORACLE



# USER vs UTILIZATOR

- ❑ Vom numi in cele ce urmeaza **user** un cont de utilizator Oracle (exemplu: user 'stud1' cu parola 'student')
- ❑ Utilizator este o persoana care prin intermediul unui cont de user interactioneaza cu Oracle
- ❑ Utilizator poate fi si un proces, o aplicatie care foloseste un cont de user pentru a interactiona cu Oracle



# PROCESE

- ❑ Cand o aplicatie lucreaza cu Oracle sunt create 2 procese:
  - ❑ Un "Proces user" (client) pe masina pe care lucreaza utilizatorul. Acesta interactioneaza cu utilizatorul si asigura comunicatia cu cel de-al doilea proces
  - ❑ Un "Proces server", pe masina unde este instalat serverul Oracle. Acesta comunica cu serverul Oracle in numele procesului user.
  - ❑ Atentie: procesul server **nu este** serverul Oracle!

# SESIUNE

- ❑ O sesiune de lucru reprezinta o conexiune a unui user cu serverul Oracle (prin procesul server).
- ❑ Un user poate avea mai multe sesiuni deschise simultan:
  - ❑ Pe aceeași baza de date
  - ❑ Cu același cont de user
  - ❑ Pe aceeași mașină sau pe mașini diferite

# PROCESUL USER

- ❑ Este creat atunci cand un utilizator incepe o sesiune de lucru folosind fie un client care interactioneaza direct cu Oracle (SQL\*Plus) fie o unealta Oracle (Server Manager, Developer, etc)
- ❑ Ruleaza pe masina utilizatorului
- ❑ Asigura interfata - grafica uneori - pentru acesta (GUI, UPI – User Program Interface)
- ❑ Procesul se termina cand utilizatorul iese din programul folosit
- ❑ Trimite comenzile/cererile utilizatorului catre procesul server (care este altceva decat serverul Oracle) si afiseaza rezultatele (date, stare, eventualele mesaje de eroare)

# PROCESUL SERVER

- ❑ Ruleaza pe masina unde este instalat serverul Oracle
- ❑ Serveste un singur proces user sau mai multe procese user, in functie de configuratie: Dedicated Server sau Multithreaded server)
- ❑ Foloseste o zona de memorie privata (PGA – Program Global Area).
- ❑ Este un intermediar intre procesul user si serverul Oracle
- ❑ Procesul se incheie cand utilizatorul termina sesiunea.

# INSTANTA ORACLE

- ❑ Un server Oracle consta dintr-o **instanta** Oracle si o **baza de date** Oracle.
- ❑ Instanta Oracle este compusa dintr-o zona de memorie numita **SGA** – System Global Area – si **procesele rulate in background** care sunt utilizate de Oracle pentru gestiunea bazei de date.
- ❑ O instanta Oracle deschide o singura baza de date.
- ❑ Este identificata prin ORACLE\_SID (variabila la nivel de SO).  
SID = System ID

# Instance

## System Global Area (SGA)

### Shared Pool

#### Library Cache

##### Shared SQL Area

```
SELECT * FROM  
employees
```

Private  
SQL Area  
(Shared  
Server Only)

Data  
Dictionary  
Cache

Server  
Result  
Cache

Other

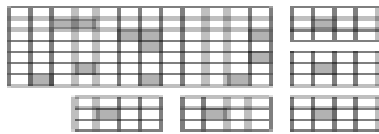
Reserved  
Pool

### Large Pool

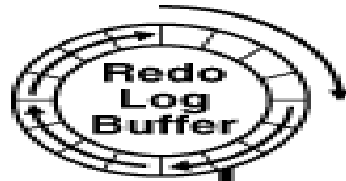
Response  
Queue

Request  
Queue

### Database Buffer Cache



### Redo Log Buffer



Fixed  
SGA

Java  
Pool

Streams  
Pool

### PGA

SQL Work Areas

Session Memory

Private SQL Area

Server  
Process

DBW<sub>n</sub>

CKPT

LGWR

ARC<sub>n</sub>

RVWR

Free Memory  
I/O Buffer Area  
UGA

PMON

SMON

RECO

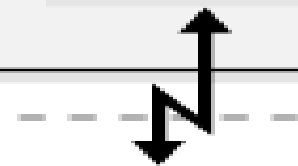
MMON

MMNL

Others

Background  
Processes

Client  
Process



# SGA – System Global Area

- ❑ Este alocata in memoria virtuala a sistemului pe care ruleaza serverul Oracle.
- ❑ Contine date si informatii de control
- ❑ Este de tip partajat - shared – mai multi useri pot folosi datele de aici pentru a se evita accesul repetat la disc
- ❑ Contine mai multe componente dintre care cele mai importante sunt:
  - ❑ Buffer Cache
  - ❑ Redo Log Buffer

# SGA – System Global Area -cont

- ❑ Mai exista de asemenea:
  - ❑ Shared Pool
  - ❑ Java Pool – utilizata pentru cod si date specifice Java de catre Java Virtual Machine (JVM)
  - ❑ Large Pool – optional, pentru date de mari dimensiuni.
  - ❑ Streams Pool – Folosita de produsul Oracle Streams



# Procese BACKGROUND

Sunt de trei tipuri:

A. Obligatorii - apar in toate configuratiile uzuale:

- ❑ **Database Writer** (DBWn) – Este responsabil cu scrierea blocurilor de date modificate/inserate din bufferele de memorie in fisierele de pe disc. In Oracle 10g pot fi maximum 20 de procese de acest fel
- ❑ **Log Writer** (LGWR) – Scribe datele din Redo Log Buffer pe disc. Scrierea se face secvential intr-un fisier de Redo Log.

# Procese BACKGROUND - cont

## A. Obligatorii - continuare

- ❑ **Checkpoint** (CKPT) – Acest proces scrie periodic pe disc toate blocurile de date (din buffere) care au fost modificate. O astfel de actiune este denumita *checkpoint*.
- ❑ Procesul anunta actiunea lui DBWR, actualizeaza fisierele de date si control ale bazei de date si inregistreaza momentul in care s-a facut checkpointul.

# Procese BACKGROUND - cont

## A. Obligatorii - continuare

- ❑ **System Monitor (SMON)** - Face verificarea consistentei datelor si initiaza recuperarea dupa incident atunci cand o instanta Oracle care a avut un incident reporneste.
- ❑ **Process Monitor (PMON)** - Dealoca resursele unui proces care are un incident. Folosit pentru procesele care au incidente.

# Procese BACKGROUND - cont

A.Obligatorii - continuare

## □ Manageability Monitor Processes :

Manageability monitor process (MMON) - efectueaza operatii legate de Automatic Workload Repository (AWR). Acesta contine date istorice despre performantele sistemului: statistici pentru sistem, sesiuni, cereri SQL individuale si servicii. E folosit pentru tuning.

Manageability monitor lite process (MMNL) - scrie statisticile din Active Session History (ASH - buffer in zona SGA) pe disc atunci cand acest buffer se umple.

# Procese BACKGROUND - cont

## A.Obligatorii - continuare

- **Recoverer Process** (RECO) - acesta este folosit in cazul bazelor de date distribuite pentru rezolvarea incidentelor aparute la tranzactiile distribuite.

# Procese BACKGROUND - cont

## B. Optionale

- ❑ **Archiver** (ARCn) – Copiaza fisierele Redo Log in arhiva de pe disc atunci cand acestea sunt pline sau cand acestea se schimba. Actiunea are loc in anumite conditii. Pot fi mai multe procese de acest fel.

# Procese BACKGROUND - cont

## B. Optionale - continuare

- ❑ **Job Queue Processes:** sunt folosite mai ales la lucrul pe loturi pentru executarea unui job la un moment dat sau repetat. Exista:
  - ❑ Job queue coordinator process (CJQ0)
  - ❑ Job queue slave process (Jnnn) - pot fi mai multe
- ❑ **Flashback Data Archiver Process (FBDA)** - pastreaza copii ale liniilor modificate in anumite tabele
- ❑ **Space management Coordinator Process (SMCO)** - pentru gestiunea spatiului

# Procese BACKGROUND - cont

## C. Procese 'slave'

- Sunt lansate de celelalte procese pentru a efectua diverse actiuni.

Nota: O descriere a proceselor de background poate fi gasita de exemplu la adresa:

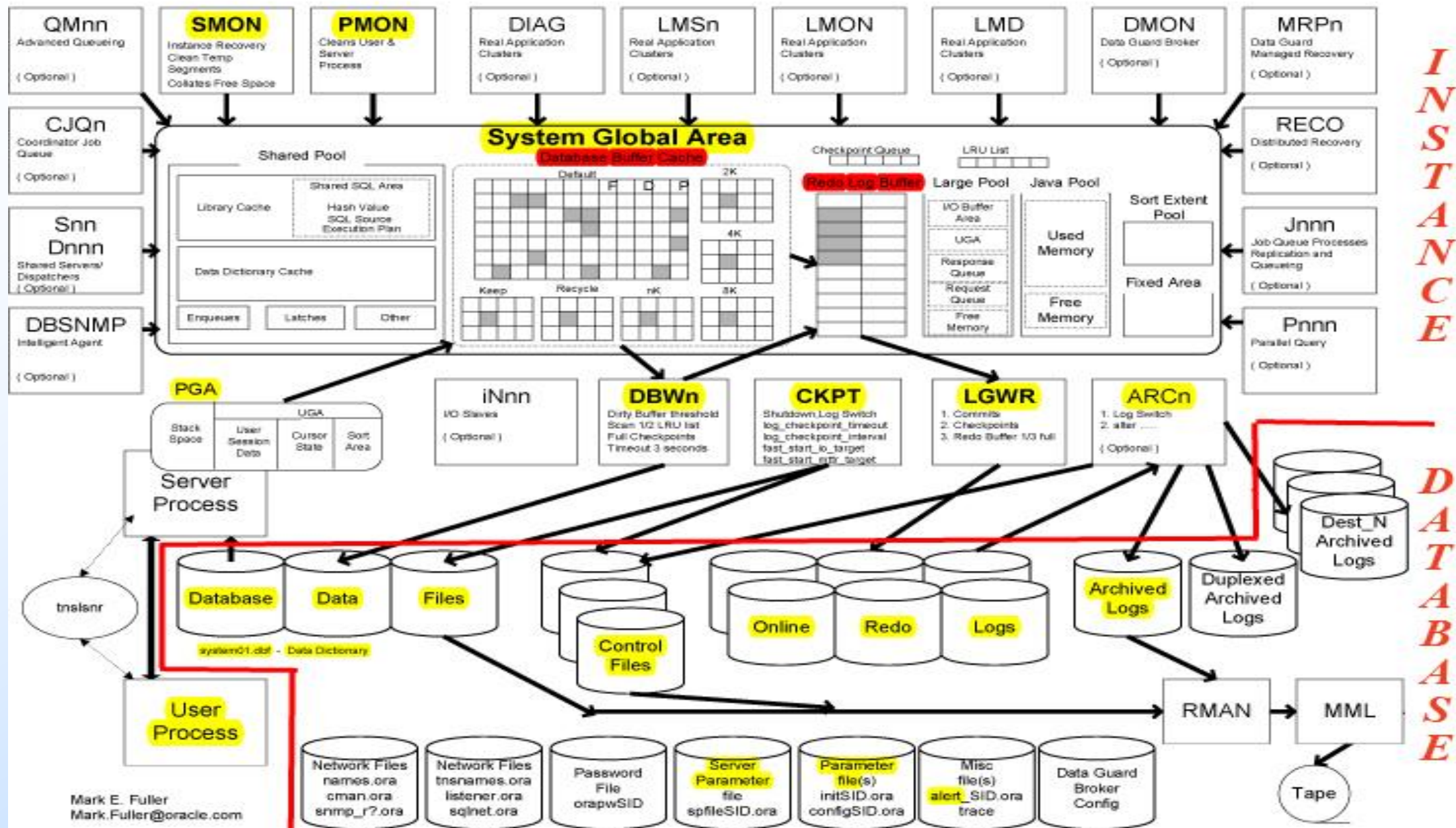
<https://www.oracle.com/webfolder/technetwork/tutorials/architecture-diagrams/19/pdf/db-19c-architecture.pdf>



# Baza de date

- ❑ A doua componenta a serverului (pe langa instanta) este Baza de date
- ❑ Este identificata prin DB\_NAME (variabila SO)
- ❑ Oracle recomanda ca instanta si BD sa aiba acelasi nume (pot fi si diferite) pentru usurinta administrarii
- ❑ Este compusa din fisiere de mai multe tipuri

# ORACLE ARCHITECTURE



# Tipuri de fisiere

- ❑ **Data files** – fisiere de date. Fisiererele de date contin:
  - ❑ Dictionarul de date al BD (mai stiti ce este acesta?),
  - ❑ Obiectele userului (mai stiti care sunt acestea?)
  - ❑ Contin si vechile valori ale datelor modificate de tranzactiile curente ('before image')
- ❑ O baza de date are cel putin un astfel de fisier.

# Tipuri de fisiere - cont

- ❑ **Redo Log Files** – Fisiere Redo Log (jurnal). Contin modificarile facute in baza de date. Sunt necesare la reconstructia ei in caz de incident.
- ❑ Fiecare baza de date contine cel putin 2 astfel de fisiere scrise in paralel si care pot fi tinute pe dispozitive de stocare diferite, prevenind pierderi de date in cazul in care un disc este distrus.

# Tipuri de fisiere - cont

- ❑ **Control files** – fisiere cu date de control. Contin informatii necesare pastrarii integritatii bazei de date
- ❑ Fiecare baza de date are cel putin un astfel de fisier.

# Alte fisiere

- ❑ Pe langa fisierele bazei de date, Oracle mai utilizeaza si alte fisiere, ca de exemplu:
  - ❑ **Fisier de parametri**: contine parametri care caracterizeaza instanta Oracle
  - ❑ **Fisier de parole**: utilizat pentru autentificarea userilor
  - ❑ **Fisiere Redo Log arhivate**: copii offline ale fisierelor Redo Log.



# Etape de procesare cerere

- ❑ Exista mai multe etape parcurse de o cerere de la emiterea ei de catre utilizator pana la executia completa. Exemplu pentru o cerere SELECT:
  - ❑ Pasul 1: **Parse**: cererea primita de la user este analizata sintactic si semantic (compilata). Serverul intoarce o informatie de stare (ok sau eroare). Shared Pool este folosita ca zona de memorie pentru aceasta operatie

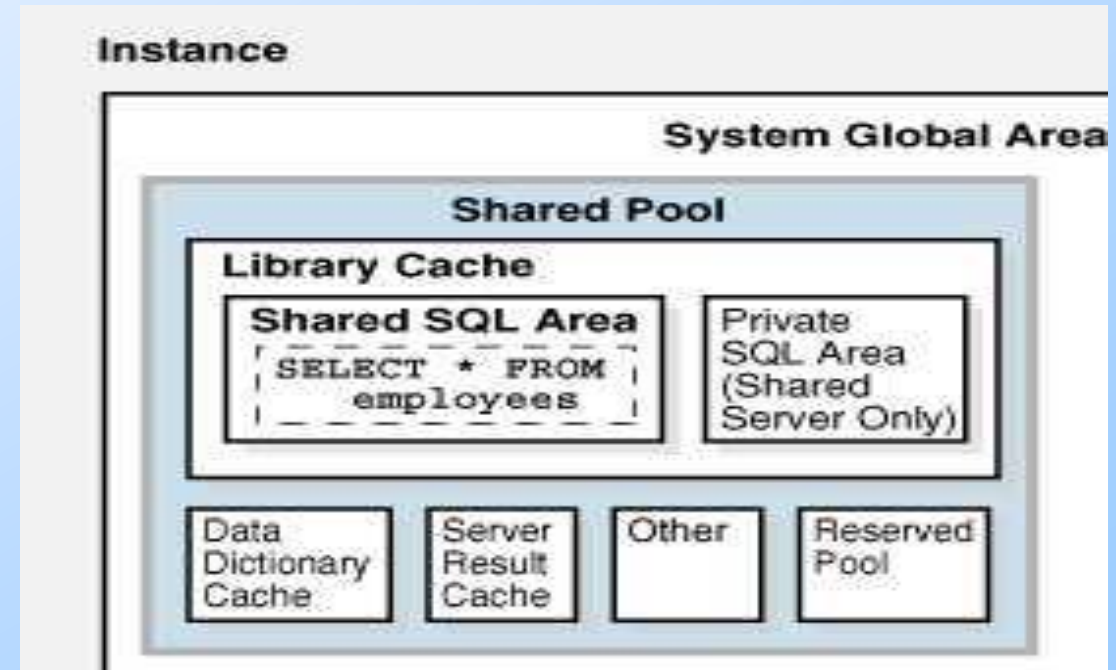
## Etape de procesare cerere - cont

- ❑ Pasul 2: **Execute**. Cererea este executata si datele din tabela rezultat sunt pregatite pentru a fi returnate userului.
- ❑ Pasul 3: **Fetch**. Liniile returnate de cerere sunt trimise userului (procesului user) pentru a fi procesate acolo (afisare sau procesare). In functie de dimensiunea datelor returnate se pot executa una sau mai multe operatii de tip FETCH



# SHARED POOL

- ❑ Este parte a SGA. Este utilizata si in pasul 1 (parse) de executie a cererii.
- ❑ Dimensiunea sa e data de parametrul SHARED\_POOL\_SIZE (din fisierul de parametrii).
- ❑ Pentru Pasul 1 sunt utilizate urmatoarele componente ale Shared Pool:
  - ❑ Library cache
  - ❑ Data dictionary cache



# Library cache

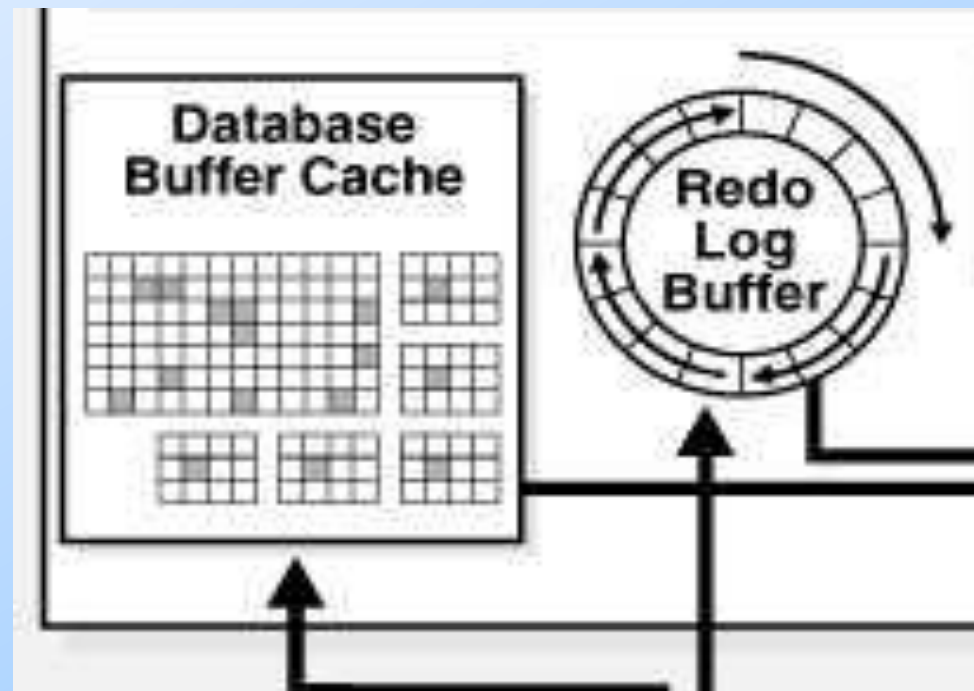
- ❑ Stocheaza informatii despre cele mai recente cereri SQL utilizate:
  - Textul cererii
  - Arborele cererii rezultat in urma etapei Parse (compilare). El este versiunea compilata a textului cererii
  - Planul de executie al cererii rezultat in urma optimizarii.
- ❑ Daca o cerere este re-executata pana sa apara alte cereri care sa afecteze planul de executie etapa Parse nu mai este necesara la re-executie (se foloseste rezultatul existent).

# Data Dictionary Cache

- ❑ Contine cele mai recent folosite informatii din dictionarul de date:
  - Descrieri tabele si coloane
  - Date cont user
  - Drepturi (privilegii)
  - Etc.
- ❑ In etapa Parse acest cache e folosit de procesul server pentru informatiile necesare compilarii cererii (numele folosite in cererea analizata). Daca nu exista sunt incarcate din fisierele de pe disc.

# Database Buffer Cache

- ❑ Parte a SGA. Tine cele mai recent utilizate blocuri de date.
- ❑ Cand o cerere este executata procesul server verifica aici existenta blocurilor necesare. Daca nu exista le citeste si le plaseaza aici.
- ❑ Dimensiunea sa este data de parametrul DB\_BLOCK\_BUFFERS
- ❑ Dimensiunea unui bloc e data de parametrul DB\_BLOCK\_SIZE.



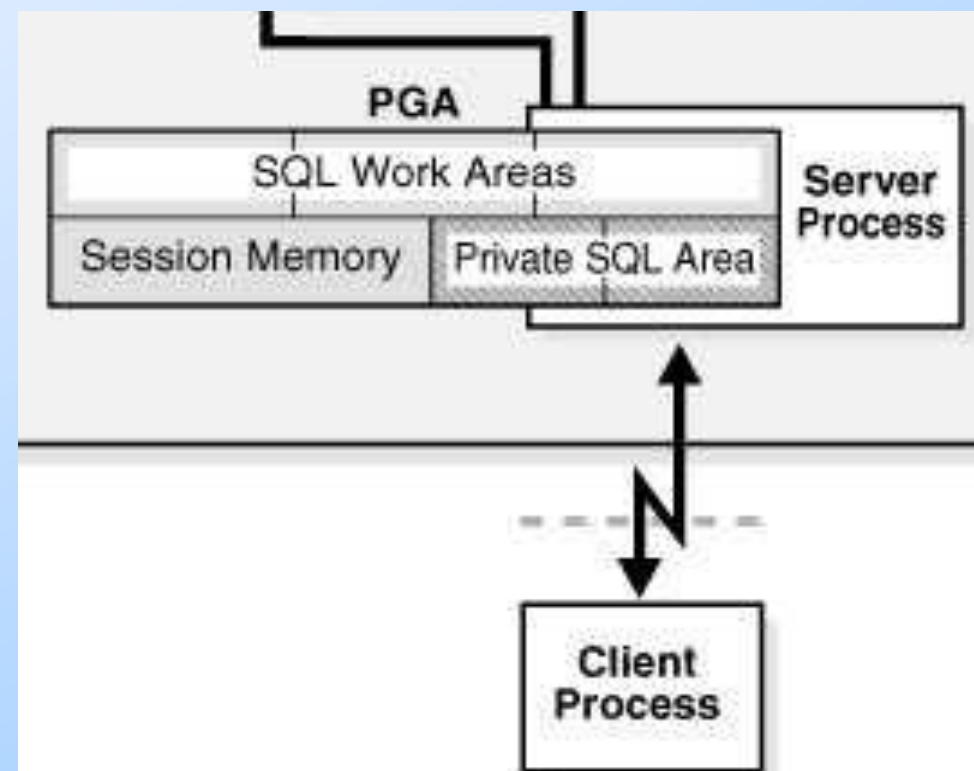
# Database Buffer Cache – cont.

- ❑ Oracle utilizeaza algoritmul LRU (ce este acesta?) pentru eliberare pozitii in buffer,
- ❑ Exceptie: operatiile de tip 'full table scan' (cand se parcurge o intreaga tabela – deci este citita in memorie). In acest caz ultimele blocuri citite pot fi primele dealocate.
- ❑ Algoritmii utilizati sunt complecsi, cele de mai sus sunt o prezentare schematica a strategiilor de gestiune a bufferului.

(vezi de exemplu <http://www.adp-gmbh.ch/ora/concepts/cache.html>)

# Program Global Area (PGA)

- ❑ Zona de memorie folosita in mod exclusiv de un proces (de tip server sau background). Nu este comuna ca in cazul SGA. Ea contine:
  - ❑ Sort area – zona sortari, folosita la operatiile de ordonare (sortare).
  - ❑ Informatii sesiune, de exemplu drepturile userului acelei sesiuni.
  - ❑ Starea cursorilor folositi in sesiunea respectiva (daca exista)
  - ❑ Stiva, continand diverse variabile de sesiune.





# Segment de Rollback

- ❑ Înainte de a se face schimbări, serverul salvează vechile valori de bloc într-un segment de rollback.
- ❑ Aceasta salvare permite anularea tranzacției (operația ROLLBACK, opusă lui COMMIT), asigură ca alte tranzacții pot citi valorile anterioare începutului de tranzacție (read consistency – mai știți ce era asta?) și ne permit de asemenea recuperarea după incident.

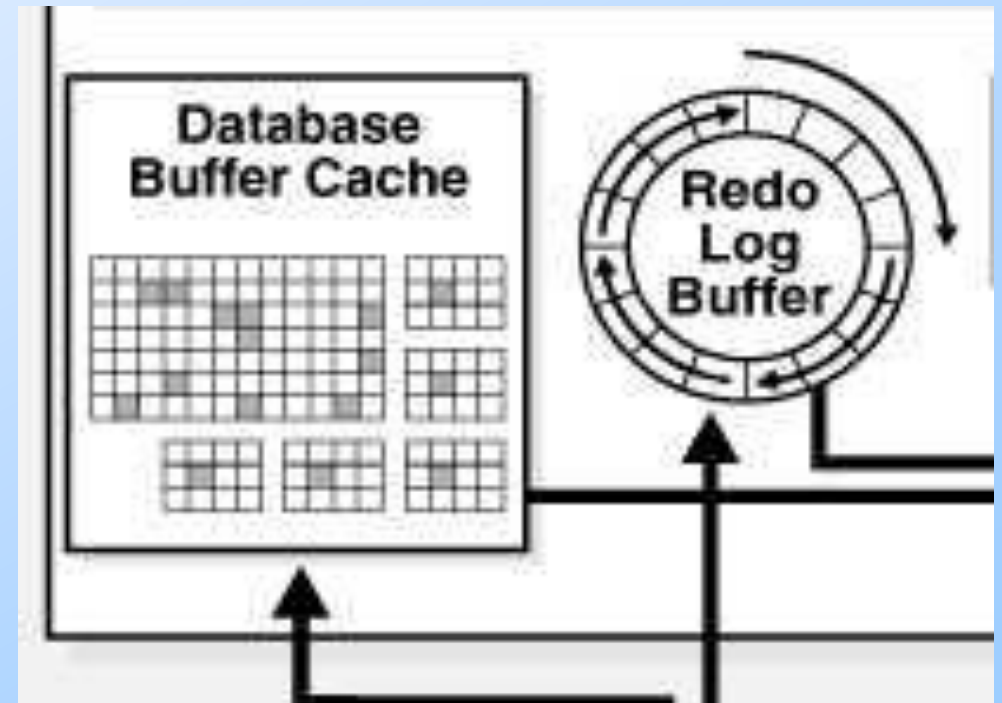
# Segment de Rollback - cont

- Segmentele de rollback sunt stocate in fisierele de date ale bazei de date si sunt aduse in buffer cache la cerere (cand este nevoie de ele).



# Redo Log Buffer

- ❑ Procesul server inregistreaza schimbarile facute de o instanta in Redo Log Buffer.
- ❑ Are o dimensiune data de parametrul LOG\_BUFFER (in bytes).
- ❑ Contine inregistrari Redo: blocul care a fost schimbat, pozitia schimbarii, noua valoare.



# Redo Log Buffer - cont

- ❑ Contine toate schimbarile, atat in date cat si in blocuri de index, rollback, etc.
- ❑ Scrierea acestor inregistrari este secventiala
- ❑ Contine inregistrari privind schimbarile facute de toate tranzactiile.
- ❑ Este folosit prin parcurgere circulara. Cand un bloc e refolosit, el este scris anterior in fisierele de pe disc.

# Database Writer

- ❑ Este un proces de tip background
- ❑ Scrie blocurile modificate - 'dirty blocks' - din Buffer Cache in fisierele de date, astfel incat sa existe suficiente blocuri libere care sa fie folosite de sistem
- ❑ A fost necesar pentru a degreva procesul server de aceasta operatie – imbunatatirea performantelor

# Database Writer - cont

□ Scrierea se face cand:

- Numarul de 'dirty blocks' in buffer depaseste o anumita valoare
- Un proces care cauta locuri libere in buffer nu le gaseste
- Timeout (la fiecare N secunde)
- Evenimente care forteaza un checkpoint: exemplu: inchiderea bazei de date.

# Log Writer

- ❑ Este un proces de tip background
- ❑ Scrie intrari din Redo Log Buffer in fisierele de pe disc.
- ❑ Operatia se efectueaza cand:
  - Redo Log Buffer e aproape plin
  - Timeout (ca mai sus)
  - Inainte ca DBWR sa scrie blocurile modificate pe disc
  - Cand o tranzactie e comisa

# COMMIT

- ❑ Oracle utilizeaza un mecanism rapid de commit care garanteaza recuperarea schimbarilor comise in caz de incident.
- ❑ Oracle asigneaza un "commit System Change Number" (SCN - este de tip timestamp) fiecărei tranzactii care se comite. Aceste numere sunt crescatoare si unice la nivelul bazei de date

# COMMIT - Pasi

- La aparitia unui COMMIT:
  1. Procesul server plaseaza o inregistrare de commit, impreuna cu SCN-ul acesteia in Redo Log Buffer
  2. LGWR scrie o portiune contigua de inregistrari din buffer pana la cea care contine COMMIT-ul in fisierele Redo Log de pe disc. In felul acesta este garantata recuperarea dupa incident

# COMMIT - Pasi

3. Userul este informat ca s-a efectuat COMMIT-ul.
4. Procesul server inregistreaza ca tranzactia e completa si ca resursele blocate de ea pot fi eliberate.

Deci:

- ☐ Scrierea efectiva a datelor pe disc este efectuata independent, de DBWR, ulterior.
- ☐ Dimensiunea tranzactiei nu conteaza



# Sumar

1. Server Oracle = Instanta + Baza de date.
2. Baza de date = Ansamblu de fisiere de diverse tipuri: de date, de redo log, de control, de parametri, samd.
3. Instanta = Zona de memorie SGA + Procese de background
4. SGA contine: Buffer cache, Redo log buffer, Shared Pool si altele; e comuna tuturor proceselor server.
5. Procese de background = 3 categorii fiecare continand mai multe tipuri, putand fi uneori mai multe procese de acelasi tip.

# Sumar - continuare

6. Printre procesele de background avem:
  1. DBWR - Database writer: scrie date din Buffer cache pe disc; sunt blocuri de date si de rollback.
  2. LGWR – Log writer: scrie inregistrari din Redo log buffer pe disc; ele contin modificarile efectuate in date.
7. La momentul unui COMMIT nu se scriu efectiv datele pe disc (DBWR) ci doar inregistrarile de redo log (LGWR). Datele vor fi scrise ulterior de DBWR.

# Bibliografie generala

## 1. Oracle Database 2 Day DBA, 19c

<https://docs.oracle.com/en/database/oracle/oracle-database/19/admq/index.html>

# Sfârșitul primului capitol

# Capitolul 2

## Instantă și bază de date

# DBA

- ❑ Exista doi utilizatori privilegiati care sunt creati inca de la instalarea Oracle (se cere doar parola pentru ei la instalare):
  1. SYS – proprietarul (owner) bazei de date precum si al tuturor tabelelor si vederilor din dictionarul bazei de date (ce este acesta?). Are rol de DBA
- ❑ SYS are privilegiul SYSDBA - vom vedea ce e asta
- ❑ Atentie: Nu creati/modificati niciodata obiecte in schema SYS (ce e o schema?)

# DBA - cont

- 2. SYSTEM – are de asemenea rol de DBA. Este proprietarul (owner) celorlalte tabele si vederi de sistem Oracle, altele decat cele din dictionarul de date (ex: cele folosite de uneltele Oracle)
  - ❑ Este bine sa nu creati obiecte in schema SYSTEM

# DBA vs SYSDBA

- ❑ DBA este **un rol** (= ansamblu/multime de privilegii) si contine majoritatea privilegiilor (drepturilor) de system – de tipul “root” din Unix
- ❑ SYSDBA este **un privilegiu** de sistem
- ❑ DBA nu contine totusi doua privilegii importante: SYSDBA si SYSOPER
- ❑ Acestea sunt privilegii importante care permit administratorului sa execute o serie de operatii de administrare.



# SYSDBA

Poate efectua operatiile:

- ☐ STARTUP si SHUTDOWN
- ☐ ALTER DATABASE: open, mount, backup, sau schimbarea setului de caractere
- ☐ CREATE DATABASE
- ☐ DROP DATABASE
- ☐ CREATE SPFILE
- ☐ ALTER DATABASE ARCHIVELOG
- ☐ ALTER DATABASE RECOVER
- ☐ Include privilegiul RESTRICTED SESSION

# SYSOPER

Poate efectua operatiile:

- ☐ STARTUP si SHUTDOWN
- ☐ ALTER DATABASE OPEN/MOUNT/BACKUP ~~sau schimbarea setului de caractere~~
- ☒ ~~CREATE DATABASE~~
- ☒ ~~DROP DATABASE~~
- ☐ CREATE SPFILE
- ☐ ALTER DATABASE ARCHIVELOG
- ☐ ALTER DATABASE RECOVER (**dar doar restaurare completa**. Restaurarea incompleta - de tip UNTIL TIME | CHANGE| CANCEL| CONTROLFILE necesita privilegiul SYSDBA)
- ☐ Include privilegiul RESTRICTED SESSION

# Etapele pornirii unei BD

1. Pornirea (start) instantă
2. Montarea bazei de date (Mount)
3. Deschiderea bazei de date (Open)

ORACLE instance shut down.

SQL> select \* from v\$instance;

select \* from v\$instance

\*

ERROR at line 1:

ORA-01034: ORACLE not available

ERROR:

ORA-01034: ORACLE not available

SQL> startup

ORACLE instance started.

|                          |           |       |
|--------------------------|-----------|-------|
| Total System Global Area | 167772160 | bytes |
|--------------------------|-----------|-------|

|            |         |       |
|------------|---------|-------|
| Fixed Size | 1218316 | bytes |
|------------|---------|-------|

|               |          |       |
|---------------|----------|-------|
| Variable Size | 62916852 | bytes |
|---------------|----------|-------|

|                  |           |       |
|------------------|-----------|-------|
| Database Buffers | 100663296 | bytes |
|------------------|-----------|-------|

|              |         |       |
|--------------|---------|-------|
| Redo Buffers | 2973696 | bytes |
|--------------|---------|-------|

Database mounted.

Database opened.

SQL> 

# Etapele pornirii unei BD

- ❑ Initial la pornirea instantei, Oracle foloseste un fisier de parametri (init<SID>.ora) care este un fisier text.
- ❑ Daca exista un fisier binar de parametri (SPFILE), parametri sunt cititi de acolo.
- ❑ Se poate specifica citirea din oricare din cele doua (text sau binar).

# Exemplu de continut (fisier text)

- ❑ db\_name=ORE
- ❑ db\_files = 80
- ❑ db\_block\_size = 8192
- ❑ db\_block\_buffers = 100
- ❑ shared\_pool\_size = 3500000
- ❑ log\_checkpoint\_interval = 10000
- ❑ log\_buffer = 32768
- ❑ log\_files = 10
- ❑ processes = 50
- ❑ max\_dump\_file\_size = 10240
- ❑ background\_dump\_dest = (/home/disk1/BDUMP)
- ❑ user\_dump\_dest = (/home/disk1/UDUMP)
- ❑ rollback\_segments = (r01, r02)
- ❑ control\_files = (ora\_control1, ora\_control2)
- ❑ compatible = 8.0.0

# Pasul 1. Pornirea instantei

- ❑ Dupa momentul pornirii instantei (fara montarea si deschiderea bazei de date) se pot executa operatiile:
  - Crearea bazei de date
  - Recrearea fisierelor de control
- ❑ Pornirea instantei presupune:
  - Citirea fisierului de parametri init<SID>.ora sau SPFILE
  - Alocarea SGA
  - Pornirea proceselor de background
  - Deschiderea fisierelor de tip TRACE si ALERT

## Pasul 2. Montarea BD

- ❑ Montarea bazei de date presupune:
  - Asocierea unei baze de date cu o instanta deja pornita
  - Localizarea si deschiderea fisierelor de control specificate in fisierul de parametri
  - Citirea fisierelor de control pentru cunoasterea numelor si locatiilor fisierelor de date si de Redo log (fara a verifica existenta lor fizica)



## Pasul 2. Montarea BD - cont

- In momentul in care instanta este pornita si baza de date montata (dar nu deschisa) se pot executa operatii de mentenanta ca:
  - Redenumirea fisierelor bazei de date (Data files)
  - Activare/dezactivare arhivare fisiere Redo Log
  - Restaurarea bazei de date

## Pasul 3. Deschiderea BD

- ❑ Dupa deschiderea BD se poate opera normal cu baza de date. Userii se pot acum conecta si trimite cereri.
- ❑ Deschiderea presupune:
  - Deschiderea fisierelor de date
  - Deschiderea fisierelor Redo log (jurnale).
  - Verificarea consistentei bazei de date. Daca este necesar, procesul SMON face o recuperare dupa incident.

## Pasul 3. Deschiderea BD

- ❑ Situatiile in care se face recuperarea dupa incident sunt acelea in care instanta nu a reusit sa efectueze toate operatiile (de exemplu in caz de crash de sistem).
- ❑ Recuperare presupune actualizarea fisierelor de date pe baza modificarilor din fisierele Redo log (care sunt actualizate la fiecare COMMIT, deci efectele tuturor tranzactiilor incheiate cu succes sunt inregistrate aici).

# Etapele opririi BD

- ❑ Sunt cele de la pornire, in ordine inversa:
  1. Inchidere BD
  2. Demontare BD
  3. Oprere instanta
- ❑ La inchiderea BD Oracle scrie pe disc blocurile modificate din Buffer Cache si inregistrarile din Redo Log Buffer dupa care inchide fisierele de date si de tip Online Redo Log
- ❑ Fisierele de control sunt inchise la demontarea bazei de date.
- ❑ Dealocarea SGA si oprirea proceselor de background se fac la oprirea instantei.

```
florin@rowlf:~  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL> select status from v$instance;  
  
STATUS  
-----  
OPEN  
  
SQL> alter database close;  
  
Database altered.  
  
SQL> select status from v$instance;  
  
STATUS  
-----  
MOUNTED  
  
SQL>
```

```
florin@rowlf:~  
STATUS  
-----  
OPEN  
  
SQL> alter database close;  
  
Database altered.  
  
SQL> select status from v$instance;  
  
STATUS  
-----  
MOUNTED  
  
SQL> alter database dismount;  
  
Database altered.  
  
SQL> select status from v$instance;  
  
STATUS  
-----  
STARTED  
  
SQL> 
```

```
florin@rowlf:~  
Database altered.  
  
SQL> select status from v$instance;  
  
STATUS  
-----  
STARTED  
  
SQL> shutdown  
ORA-01507: database not mounted  
  
ORACLE instance shut down.  
SQL> select * from v$instance;  
select * from v$instance  
*  
ERROR at line 1:  
ORA-01034: ORACLE not available  
  
ERROR:  
ORA-01034: ORACLE not available  
  
SQL> 
```

# Tipuri de oprire

- Sunt 4 moduri de oprire. Oprirea normala este varianta implicita

|                                               | <b>Normal</b> | <b>Tranzactio<br/>nal</b> | <b>Imediat</b> | <b>Abort</b> |
|-----------------------------------------------|---------------|---------------------------|----------------|--------------|
| Permisiune noi conexiuni                      | NU            | NU                        | NU             | NU           |
| Asteapta pana se termina sesiunile curente    | DA            | NU                        | NU             | NU           |
| Asteapta pana se termina tranzactiile curente | DA            | DA                        | NU             | NU           |
| Forteza un checkpoint si inchide fisierele    | DA            | DA                        | DA             | NU           |



# Oprire normala

- ❑ Nu sunt permise noi conexiuni
- ❑ Oracle asteapta ca toti userii deja conectati sa termine sesiunea de lucru (sa se deconecteze)
- ❑ Inchidere si demontare baza de date si oprire instanta
- ❑ Repornire normala (nu este nevoie de recuperare)

# Oprire tranzactionala

- ❑ Nu sunt permise noi conexiuni si nici noi tranzactii de la userii deja conectati
- ❑ La terminarea tranzactiei curente pentru orice user acesta e deconectat
- ❑ Se executa apoi pasii de la oprirea imediata
- ❑ Repornire normala (nu este nevoie de recuperare)

# Oprire imediata

- ❑ Nu sunt permise noi conexiuni
- ❑ Cererile SQL curente sunt oprite din executie
- ❑ Oracle deconecteaza userii curenti
- ❑ Tranzactiile active sunt revocate (ROLLBACK)
- ❑ Inchidere si demontare baza de date si oprire instanta
- ❑ Repornire normala (nu este nevoie de recuperare)

# Oprire tip ABORT

- ❑ Nu sunt permise noi conexiuni
- ❑ Cererile SQL curente sunt oprite din executie
- ❑ Oracle deconecteaza userii curenti
- ❑ Tranzactiile active sunt revocate (ROLLBACK)
- ❑ Instanta este oprita fara inchiderea fisierelor
- ❑ La repornire este necesara recuperarea dupa incident a instantei (procesul SMON)

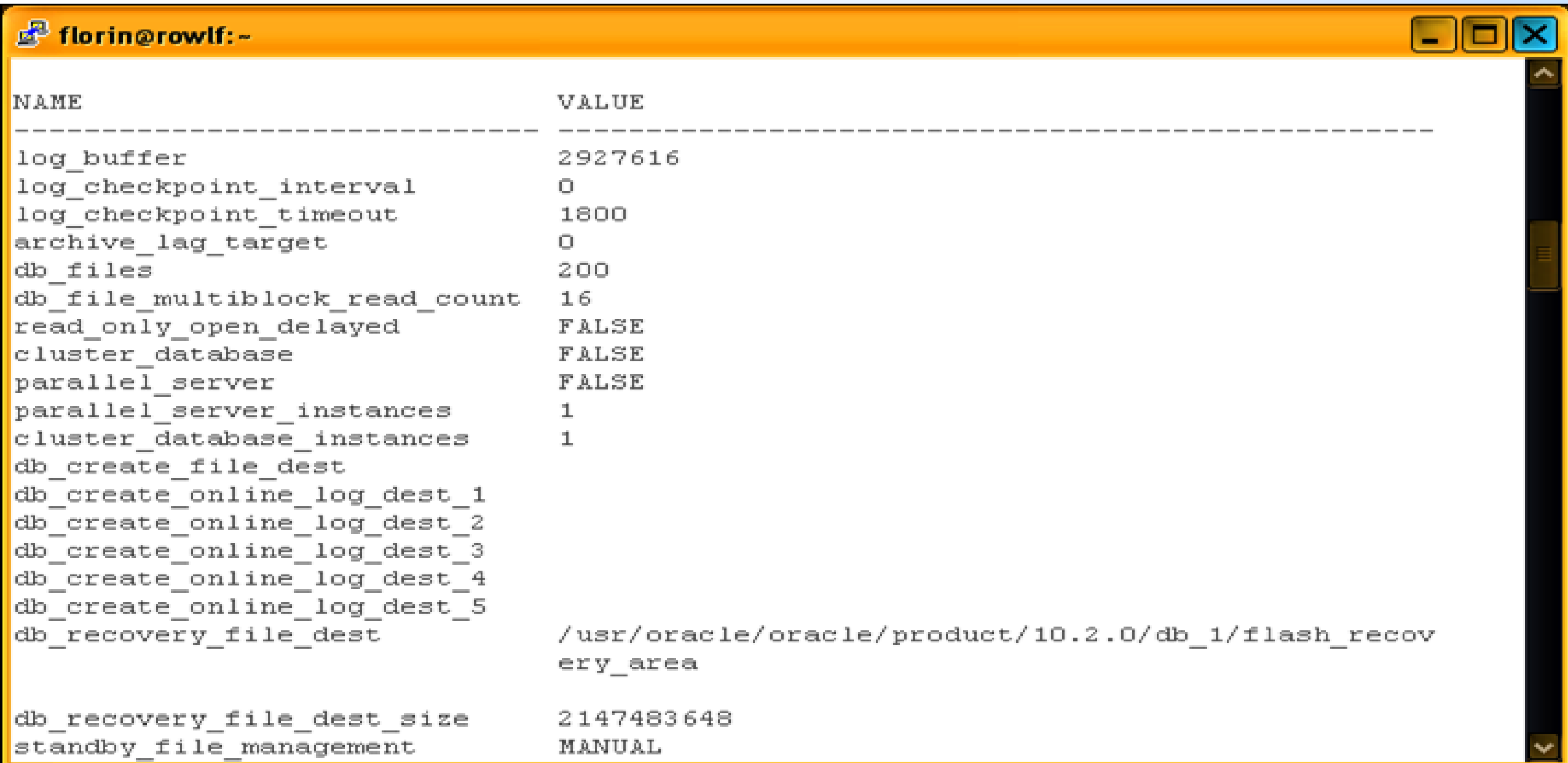
# Vederi dinamice privind performantele

- ❑ Exista tabele de sistem continand date legate de performante care sunt accesibile administratorului prin vederi
- ❑ Aceste vederi au un nume incepand cu V\$
- ❑ Unele sunt accesibile dupa pornirea instantei (BD inca nemontata)
- ❑ Altele sunt accesibile doar dupa montarea BD
- ❑ Vezi de exemplu: <http://www.ss64.com/orav/>

# Exemple de vederi in cazul NOMOUNT

- ❑ V\$PARAMETER - informatii despre parametrii de initializare
- ❑ V\$SGA - informatii despre SGA
- ❑ V\$SESSION - informatii despre sesiunile curente
- ❑ V\$INSTANCE - starea instantei curente
- ❑ V\$OPTION - optiunile de instalare pentru serverul Oracle

# V\$PARAMETER



```
florin@rowlf:~
```

| NAME                          | VALUE                                                      |
|-------------------------------|------------------------------------------------------------|
| log_buffer                    | 2927616                                                    |
| log_checkpoint_interval       | 0                                                          |
| log_checkpoint_timeout        | 1800                                                       |
| archive_lag_target            | 0                                                          |
| db_files                      | 200                                                        |
| db_file_multiblock_read_count | 16                                                         |
| read_only_open_delayed        | FALSE                                                      |
| cluster_database              | FALSE                                                      |
| parallel_server               | FALSE                                                      |
| parallel_server_instances     | 1                                                          |
| cluster_database_instances    | 1                                                          |
| db_create_file_dest           |                                                            |
| db_create_online_log_dest_1   |                                                            |
| db_create_online_log_dest_2   |                                                            |
| db_create_online_log_dest_3   |                                                            |
| db_create_online_log_dest_4   |                                                            |
| db_create_online_log_dest_5   |                                                            |
| db_recovery_file_dest         | /usr/oracle/oracle/product/10.2.0/db_1/flash_recovery_area |
| db_recovery_file_dest_size    | 2147483648                                                 |
| standby_file_management       | MANUAL                                                     |

# V\$SGA

```
florin@rowlf:~  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL> r  
1* select * from v$sga  
  
NAME                                VALUE  
-----  
Fixed Size                          1218316  
Variable Size                       62916852  
Database Buffers                   100663296  
Redo Buffers                       2973696  
  
SQL>
```



# V\$SESSION

```
florin@rowlf:~  
SQL> r  
1* select sid, status, server, schemaname, osuser from v$session
```

| SID | STATUS | SERVER    | SCHEMANAME | OSUSER |
|-----|--------|-----------|------------|--------|
| 149 | ACTIVE | DEDICATED | SYS        | florin |
| 151 | ACTIVE | DEDICATED | SYS        | florin |
| 155 | ACTIVE | DEDICATED | SYS        | florin |
| 159 | ACTIVE | DEDICATED | SYS        | florin |
| 160 | ACTIVE | DEDICATED | SYS        | florin |
| 161 | ACTIVE | DEDICATED | SYS        | florin |
| 162 | ACTIVE | DEDICATED | SYS        | florin |
| 163 | ACTIVE | DEDICATED | SYS        | florin |
| 164 | ACTIVE | DEDICATED | SYS        | florin |
| 165 | ACTIVE | DEDICATED | SYS        | florin |
| 166 | ACTIVE | DEDICATED | SYS        | florin |
| 167 | ACTIVE | DEDICATED | SYS        | florin |
| 168 | ACTIVE | DEDICATED | SYS        | florin |
| 169 | ACTIVE | DEDICATED | SYS        | florin |
| 170 | ACTIVE | DEDICATED | SYS        | florin |

```
15 rows selected.  
SQL>
```

# V\$INSTANCE

```
florin@rowlf:~  
  
File afiedt.buf saved.  
  
1* select instance_name, version, status, archiver, logins from v$instance  
SQL> /  
  
INSTANCE_NAME      VERSION            STATUS            ARCHIVE LOGINS  
-----  
ore                10.2.0.1.0        OPEN              STOPPED ALLOWED  
  
SQL>
```

# V\$OPTION



florin@rowlf: -

```
1* select * from v$option
```

| PARAMETER                          | VALUE |
|------------------------------------|-------|
| Partitioning                       | TRUE  |
| Objects                            | TRUE  |
| Real Application Clusters          | FALSE |
| Advanced replication               | TRUE  |
| Bit-mapped indexes                 | TRUE  |
| Connection multiplexing            | TRUE  |
| Connection pooling                 | TRUE  |
| Database queuing                   | TRUE  |
| Incremental backup and recovery    | TRUE  |
| Instead-of triggers                | TRUE  |
| Parallel backup and recovery       | TRUE  |
| Parallel execution                 | TRUE  |
| Parallel load                      | TRUE  |
| Point-in-time tablespace recovery  | TRUE  |
| Fine-grained access control        | TRUE  |
| Proxy authentication/authorization | TRUE  |
| Change Data Capture                | TRUE  |
| Plan Stability                     | TRUE  |
| Online Index Build                 | TRUE  |
| Coalesce Index                     | TRUE  |
| Managed Standby                    | TRUE  |

# Exemple de vederi in cazul MOUNT

- ❑ V\$CONTROLFILE - numele fisierelor de control
- ❑ V\$DATABASE - informatii despre baza de date
- ❑ V\$DATAFILE - informatii despre fisierele de date luate din fisierele de control
- ❑ V\$LOGFILE - informatii despre fisierele curente de tip Redo log

# V\$CONTROLFILE

- ❑ STATUS: This is null if the name can be determined, INVALID otherwise
- ❑ NAME: The control filename
- ❑ IS\_RECOVERY\_DEST\_FILE: This indicates whether the control file was created in the fast recovery area (YES) or not (NO)
- ❑ BLOCK\_SIZE: The control file block size
- ❑ FILE\_SIZE\_BLKES: The control file size in blocks

# V\$CONTROLFILE

```
florin@rowlf:~  
1* select * from v$controlfile
```

| STATUS | NAME                                                             | IS_ | BLOCK_SIZE | FILE_SIZE | BLKS |
|--------|------------------------------------------------------------------|-----|------------|-----------|------|
|        | /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/control01.ctl | NO  | 16384      |           | 430  |
|        | /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/control02.ctl | NO  | 16384      |           | 430  |
|        | /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/control03.ctl | NO  | 16384      |           | 430  |

```
SQL>
```

# V\$DATABASE

```
florin@rowlf:~  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL> r  
1* select dbid, name, created, log_mode, open_mode from v$database  
  
      DBID NAME                                CREATED      LOG_MODE      OPEN_MODE  
-----  
1404736936 ORE                                09-APR-06 NOARCHIVELOG READ WRITE  
  
SQL>  
SQL>  
SQL>
```

# V\$DATAFILE

```
florin@rowlf:~  
SQL> select file#, name, status, enabled, blocks, block_size from v$datafile;  
  
FILE# NAME STATUS ENABLED BLOCKS BLOCK_SIZ  
-----  
1 /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/system01.dbf SYSTEM READ WRITE 62720 819  
2 /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/undotbs01.dbf ONLINE READ WRITE 19840 819  
3 /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/sysaux01.dbf ONLINE READ WRITE 44800 819  
4 /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/users01.dbf ONLINE READ WRITE 2720 819  
5 /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore/example01.dbf ONLINE READ WRITE 12800 819  
  
SQL>
```



# V\$LOGFILE

GROUP#: The redo log group identification number

STATUS: The status of the log member; it can be  
INVALID, STALE, DELETED, or NULL when the file  
is in use

TYPE: The type of the log file, it can be ONLINE or  
STANDBY

MEMBER: The redo log member name

IS\_RECOVERY\_DEST\_FILE: This indicates whether  
the file was created in the fast recovery area (YES)  
or not (NO)

# V\$LOGFILE

florin@rowlf:~

SQL>

SQL> r

```
1* select * from v$logfile
```

| GROUP# | STATUS | TYPE   | MEMBER                                                            | IS_ |
|--------|--------|--------|-------------------------------------------------------------------|-----|
| 3      | STALE  | ONLINE | /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore<br>/redo03.log | NO  |
| 2      |        | ONLINE | /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore<br>/redo02.log | NO  |
| 1      |        | ONLINE | /usr/oracle/oracle/product/10.2.0/db_1/oradata/ore<br>/redo01.log | NO  |

SQL>

# Parametri dinamici

- ❑ Unii parametri de initializare pot fi alterati dinamic (cand instanta este pornita)
- ❑ Sunt cei care sunt marcati ca modificabili in coloanele  
    ISSES\_MODIFIABLE  
    ISSYS\_MODIFIABLE  
din vederea V\$PARAMETER
- ❑ Comenzile ALTER SYSTEM [DEFERRED] sunt inregistrate in fisierul de alerte (ALERT file)

# Parametri dinamici - cont

Exemplu:

```
ALTER SESSION SET nume_parametru=valoare
```

- modifica parametrul doar pentru sesiunea unde este executata comanda

```
ALTER SYSTEM SET nume_parametru=valoare  
[DEFERRED]
```

- modifica global parametrul.
- optiunea DEFERRED modifica parametrul pentru sesiunile care se deschid dupa executia comenzii (nu si pentru cele deschise)

# Sesiuni RESTRICTED

- ❑ Sunt folosite cand se efectueaza operatii de mentenanta asupra bazei de date.
- ❑ Cand baza de date e pornita in mod RESTRICTED doar userii cu privilegiul RESTRICTED SESSION pot sa se conecteze.
- ❑ La pornire se da STARTUP RESTRICT

# Sesiuni RESTRICTED - cont

- ❑ Daca baza de date este deja pornita se poate trece in mod RESTRICTED cu comanda:

```
ALTER SYSTEM {ENABLE | DISABLE }  
RESTRICTED SESSION
```

- ❑ ENABLE – se permit noi conexiuni doar de la userii cu privilegiul mentionat. Sesiunile existente nu sunt afectate.
- ❑ DISABLE – se permit conexiuni de la orice user

# Sesiuni RESTRICTED - cont

❑ Pentru a vedea modul curent putem lansa cererea:

```
SELECT LOGINS FROM V$INSTANCE;
```

❑ Obtinem ca rezultat o tabela (ca cea de mai jos)

**LOGINS**

-----

**RESTRICTED**

# Inchiderea sesiunilor

- ❑ Dupa trecerea in modul RESTRICTED putem dori sa inchidem anumite sesiuni active.

- ❑ Aflarea datelor despre o sesiune:

```
SELECT SID, SERIAL# FROM V$SESSION  
WHERE USERNAME = 'SCOTT'
```

- ❑ Inchiderea unei sesiuni:

```
ALTER SYSTEM KILL SESSION '5,10'
```

Unde 5, 10 sunt numere returnate pentru SID si SERIAL#



# Efectul comenzilor: in fereastra DBA

```
florin@rowlf:~  
SQL>  
SQL> r  
1* SELECT SID, SERIAL# FROM V$SESSION WHERE USERNAME = 'SCOTT'  
  
      SID      SERIAL#  
-----  
      148         1  
  
SQL> alter system kill session '148,1';  
  
System altered.  
  
SQL> 
```

# Efectul comenzilor: in fereastra lui SCOTT

```
florin@rowlf:~  
SQL> select * from spec;  
  
      CODS NUME      DOMENIU  
-----  
      11  MATEMATICA  STIINTE EXACTE  
      21  GEOGRAFIE   UMANIST  
      24  ISTORIE     UMANIST  
  
SQL>  
SQL>  
SQL> select * from stud;  
select * from stud  
*  
ERROR at line 1:  
ORA-00028: your session has been killed  
  
SQL>
```

# Inchiderea sesiunilor - cont

- ❑ Efectul comenzii (realizator: procesul PMON) este:
  - ❑ Se anuleaza tranzactia curenta din sesiune (rollback)
  - ❑ Se elibereaza toate resursele ocupate de acea sesiune inclusiv linii sau tabele blocate

# Fisiere TRACE

- ❑ Sunt scrise de procesele server si background
- ❑ Oracle inregistreaza in ele informatii despre erorile aparute
- ❑ Operatia se activeaza fie prin ALTER SESSION fie prin parametrul SQL\_TRACE

Exemplu:

```
ALTER SESSION SET SQL_TRACE=TRUE;
```

# Fisiere TRACE - cont

Caracteristicile fisierelor TRACE sunt date de parametrii:

- ❑ MAX\_DUMP\_FILE\_SIZE – specificat in blocuri pe disc
- ❑ BACKGROUND\_DUMP\_DEST – locatia fisierelor trace pentru procesele de background si a fisierelor ALERT
- ❑ USER\_DUMP\_DEST – locatia fisierelor TRACE create la cererea userului.
- ❑ Incepand cu Oracle 11.1, locatia lor e specificata prin parametrul DIAGNOSTIC\_DEST

# FISIERE ALERT

- ❑ Sunt scrise de procesele server si background
- ❑ Oracle inregistreaza in ele cronologic mesajele si erorile
- ❑ Numele fisierului este de obicei ALERT\_<SID>.log sau <SID>alrt.log
- ❑ Contin toate erorile interne Oracle (cod -600) si erori privind coruperea datelor de pe disc (cod -1578) precum si informatii despre STARTUP, SHUTDOWN, ARCHIVE LOG, RECOVER

# CREAREA BD

- ❑ In Oracle se poate crea o baza de date:
  - ❑ Folosind instrumente cum este DBCA - Database Configuration Assistant (asistent de creare a bazei de date)
  - ❑ Manual, prin comenzi SQL
- ❑ La instalarea Oracle de obicei se creaza o prima baza de date
- ❑ Se poate crea de asemenea o baza de date dupa instalare in cazuri ca:
  - ❑ S-a folosit *Oracle Universal Installer* (OUI) doar pentru instalare fara crearea unei baze de date
  - ❑ Crearea unei noi baze de date (si a unei noi instante) pe aceeaasi masina
  - ❑ Crearea unei baze de date care sa fie o copie a uneia existente (clonare)

# Preliminarii

- ❑ Inainte de crearea unei baze de date trebuie sa ne asiguram ca:
- ❑ Oracle este instalat, deci exista inclusiv variabilele de mediu necesare si sunt stabilite directoarele care vor gazdui datele si aplicatiile
- ❑ Exista suficienta memorie interna pe masina in cauza pentru a putea lansa o instanta
- ❑ Exista suficient spatiu pe disc pentru crearea fisierelor necesare bazei de date
- ❑ Utilizatorul care efectueaza operatia are privilegiile necesare (este administrator de sistem de exemplu sau foloseste un fisier de parole pentru autentificare)

(<https://docs.oracle.com/en/database/oracle/oracle-database/19/admq/index.html>)



# Procedeu

Descris in cursul practic:

[https://curs.upb.ro/2024/pluginfile.php/115967/mod\\_resource/content/1/ABD\\_cp1\\_2024.pdf](https://curs.upb.ro/2024/pluginfile.php/115967/mod_resource/content/1/ABD_cp1_2024.pdf)

# Sfârșitul capitolului 2

# 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)

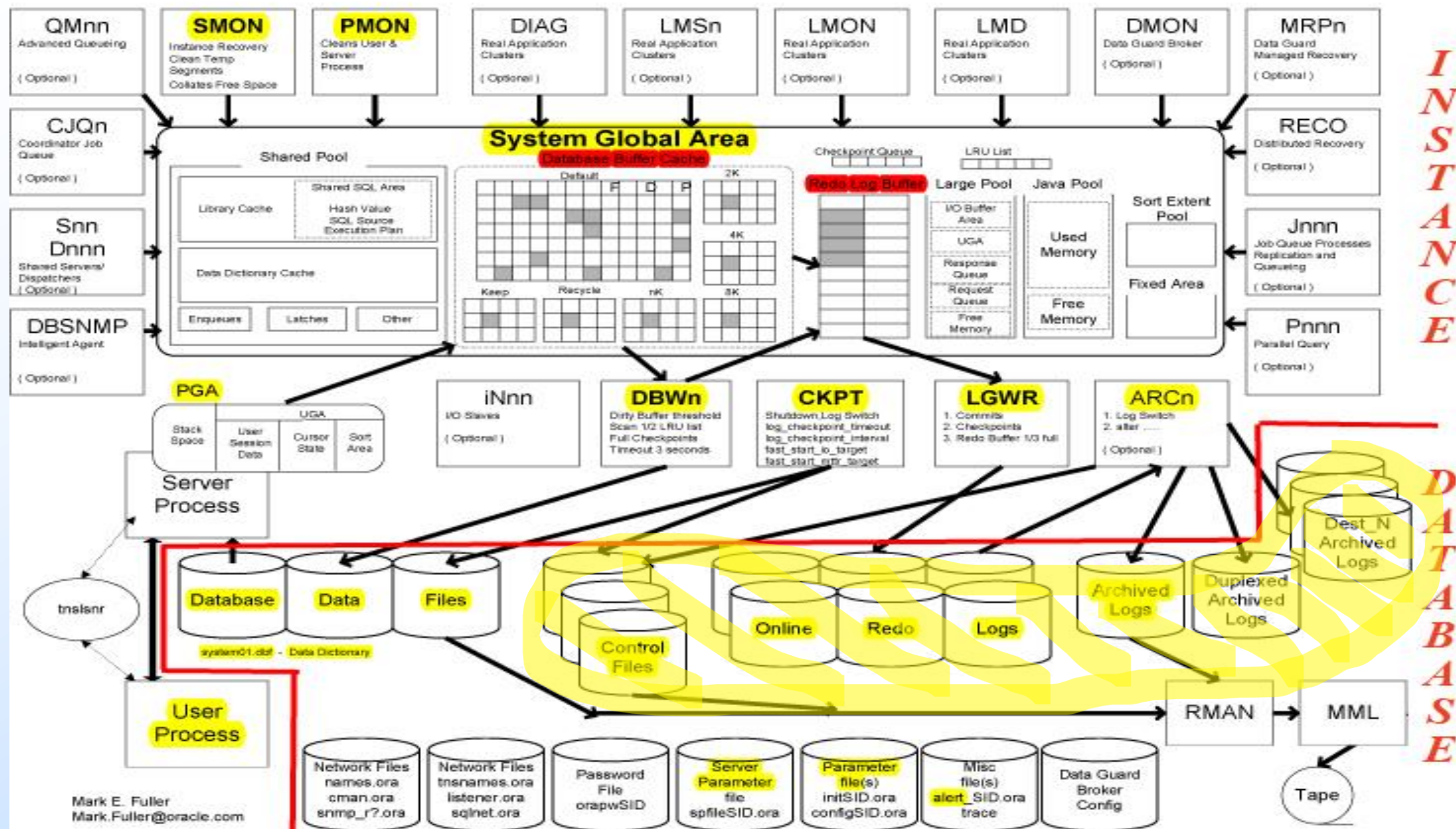
In acest capitol vom discuta despre:

- ☐ Fisierele de control
- ☐ Fisierele de Redo Log

In capitolul urmator vor fi detaliate:

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

# ORACLE ARCHITECTURE



# 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 fisierelor 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 fisierelelor 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 paralel (au acelasi continut - 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

# 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 exploatate in paralel):

```
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

# Mutare / redenumire

- ❑ Pentru a redenumi/muta un fisier de control trebuie urmati urmatorii pasi:
  1. Opreire (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.

# 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.

# 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.***

# 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).

SQL> connect system as sysdba

Introduce mi parola:

Conectat.

SQL> select member from v\$logfile;

MEMBER

---

C:\ORACLE11G\_DB\ORADATA\BD\REDO03.LOG

C:\ORACLE11G\_DB\ORADATA\BD\REDO02.LOG

C:\ORACLE11G\_DB\ORADATA\BD\REDO01.LOG

SQL>



```
SQL>  
SQL> SELECT NAME FROM U$DATAFILE;
```

```
NAME
```

```
-----  
C:\ORACLE11G_DB\ORADATA\BD\SYSTEM01.DBF  
C:\ORACLE11G_DB\ORADATA\BD\SYSAUX01.DBF  
C:\ORACLE11G_DB\ORADATA\BD\UNDOTBS01.DBF  
C:\ORACLE11G_DB\ORADATA\BD\USERS01.DBF  
C:\ORACLE11G_DB\ORADATA\BD\EXAMPLE01.DBF  
C:\ORACLE11G_DB\ORADATA\BD\BD_DATA.DBF  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_IDX  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_TAB  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DEP_IDX  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DEP_TAB  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DIA_IDX  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DIA_TAB  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_LOB_DATA  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\RAP_GR_IDX  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_SYS_META_IDX  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_SYS_META_TAB  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_TEMP_IDX  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_TEMP_TAB  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_UER_IDX  
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_UER_TAB
```

```
20 tnregistrÑri selectate.
```

```
SQL> _
```

```
SQL>  
SQL>  
SQL> SELECT VALUE FROM V$PARAMETER  
2 WHERE NAME='control_files';
```

VALUE

---

C:\ORACLE11G\_DB\ORADATA\BD\CONTROL01.CTL, C:\ORACLE11G\_DB\FLASH\_RECOVERY\_AREA\BD\CONTROL02.CTL

```
SQL>
```

# 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. Fisierul va fi create in locatia indicate de parametrul CONTROL\_FILES.

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 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:

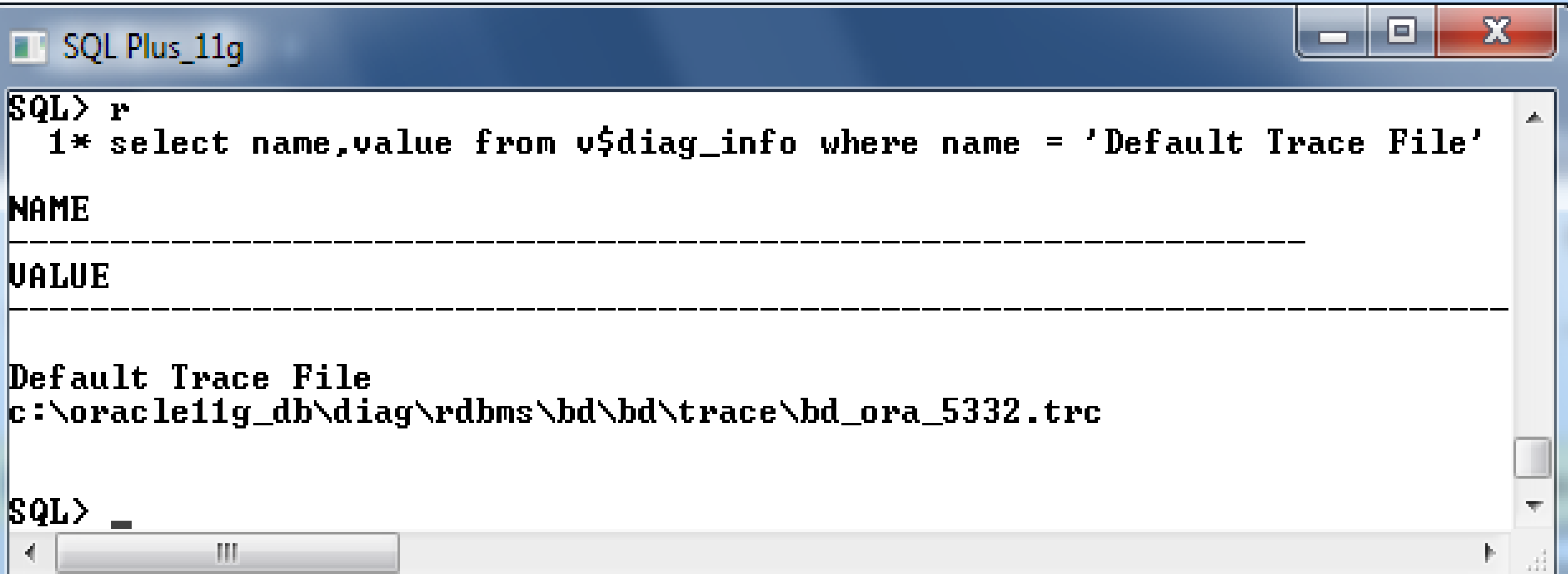
```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

Efectul: in fisierul TRACE vor fi generate comenzile SQL respective

# Salvare FC

- ❑ Locatia fisierului TRACE folosit in sesiunea curenta se poate afla cu cererea:

```
SELECT VALUE FROM V$DIAG_INFO  
WHERE NAME='Default Trace File'
```



The screenshot shows a window titled "SQL Plus\_11g" with a standard Windows-style title bar (minimize, maximize, close buttons). The main area contains the following text:

```
SQL> r  
1* select name,value from v$diag_info where name = 'Default Trace File'
```

| NAME               | VALUE                                                  |
|--------------------|--------------------------------------------------------|
| Default Trace File | c:\oracle11g_db\diag\rdbms\bd\bd\trace\bd_ora_5332.trc |

SQL> \_

The table output is displayed with dashed lines separating the header from the body and between rows. The path shown is c:\oracle11g\_db\diag\rdbms\bd\bd\trace\bd\_ora\_5332.trc.

# Salvare FC

In fisierul TRACE vom gasi:

**CREATE CONTROLFILE REUSE DATABASE "BD" RESETLOGS NOARCHIVELOG**

**MAXLOGFILES 16**

**MAXLOGMEMBERS 3**

**MAXDATAFILES 100**

**MAXINSTANCES 8**

**MAXLOGHISTORY 292**

**LOGFILE**

**GROUP 1 'C:\ORACLE11G\_DB\ORADATA\BD\REDO01.LOG' SIZE 50M BLOCKSIZE 512,**

**GROUP 2 'C:\ORACLE11G\_DB\ORADATA\BD\REDO02.LOG' SIZE 50M BLOCKSIZE 512,**

**GROUP 3 'C:\ORACLE11G\_DB\ORADATA\BD\REDO03.LOG' SIZE 50M BLOCKSIZE 512**

**-- STANDBY LOGFILE**

**DATAFILE**

**'C:\ORACLE11G\_DB\ORADATA\BD\SYSTEM01.DBF',**

**'C:\ORACLE11G\_DB\ORADATA\BD\SYSAUX01.DBF',**

**'C:\ORACLE11G\_DB\ORADATA\BD\UNDOTBS01.DBF',**

**'C:\ORACLE11G\_DB\ORADATA\BD\USERS01.DBF',**

**'C:\ORACLE11G\_DB\ORADATA\BD\EXAMPLE01.DBF',**

**'C:\ORACLE11G\_DB\ORADATA\BD\BD\_DATA.DBF',**

# Salvare FC

```
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_IDX',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_TAB',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DEP_IDX',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DEP_TAB',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DIA_IDX',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_DIA_TAB',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_LOB_DATA',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\RAP_GR_IDX',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_SYS_META_IDX',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_SYS_META_TAB',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_TEMP_IDX',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_TEMP_TAB',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_VER_IDX',  
'C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_VER_TAB'
```

**CHARACTER SET EE8MSWIN1250**

;



# 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 avariat permanent

1. 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**

2. 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)
3. 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:
  1. Oprire baza
  2. Editare fisier de parametri pentru a elimina din CONTROL\_FILES fisierul care se sterge
  3. 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.

# V\$DATABASE

SQL Plus\_11g



SQL>

SQL>

SQL>

SQL> r

```
1 select controlfile_type, controlfile_created,  
2 controlfile_sequence#, controlfile_change#, controlfile_time  
3* from v$database
```

| CONTROL | CONTROLFIL | CONTROLFILE_SEQUENCE# | CONTROLFILE_CHANGE# | CONTROLFIL |
|---------|------------|-----------------------|---------------------|------------|
| CURRENT | 23-03-2015 | 4484                  | 3119322             | 10-10-2019 |

SQL> \_



# V\$CONTROLFILE

SQL Plus\_11g

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

```
NAME  
-----  
C:\ORACLE11G_DB\ORADATA\BD\CONTROL01.CTL  
C:\ORACLE11G_DB\FLASH_RECOVERY_AREA\BD\CONTROL02.CTL
```

```
SQL> _
```

# 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 inca scrise pe disc pana in momentul incidentului (deci se folosesc DOAR pentru recuperarea datelor – recovery).

# Grupuri si membri

- ❑ Fisiererele 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 fisiere este LGWR (Log Writer, unul dintre procesele de background ale unei instante)



# Grupuri si membri – cont

- ❑ Toti membrii (=fisiere) 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 fisierelor de date.

# Cate grupuri sunt

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

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

# PARAMETRI

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

# Utilizare

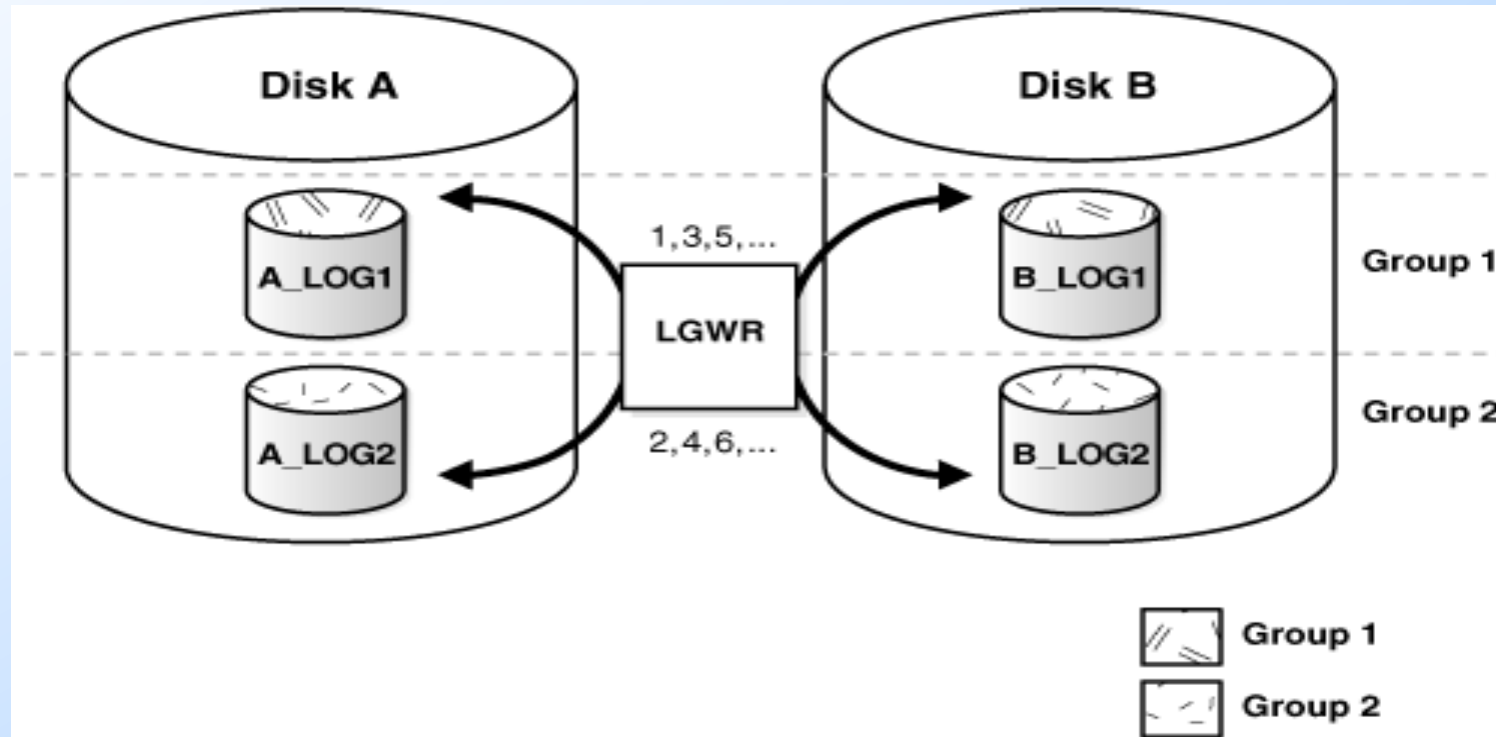
- ❑ Oracle inregistreaza secvential toate schimbarile facute in baza de date in Redo Log Buffer sub forma unor inregistrari de Redo.
- ❑ Bufferul este folosit circular (cand se ajunge la capat se reia cu inceputul).
- ❑ Din Buffer aceste inregistrari sunt scrise in fisierul (=grupul) curent de Redo de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

# Utilizare - cont

1. In momentul in care apare un COMMIT (s-a vorbit de asta la un capitol precedent)
2. In momentul in care Redo log buffer este plin intr-o anumita proportie
3. Cand apare un time-out al LGWR (la fiecare 3 secunde)
4. 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 o astfel de schimbare si in cazul in care fisierul curent nu e plin
- ❑ La fiecare log 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 fisierelor 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 fortat 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

- Fisiererele de Redo Log se pot clasifica in
  1. CURRENT - Curent – cel in care se scrie la un moment dat
  2. ACTIVE - Activ – s-a scris in el anterior dar modificarile cuprinse in el nu sunt inca scrise in fisiererele de date si deci e necesar pentru recuperare instantata.
  3. INACTIVE - Inactiv – s-a scris in el anterior si modificarile s-au inregistrat si in fisiererele de date, deci nu mai e necesar pentru recuperare instantata
  4. 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)

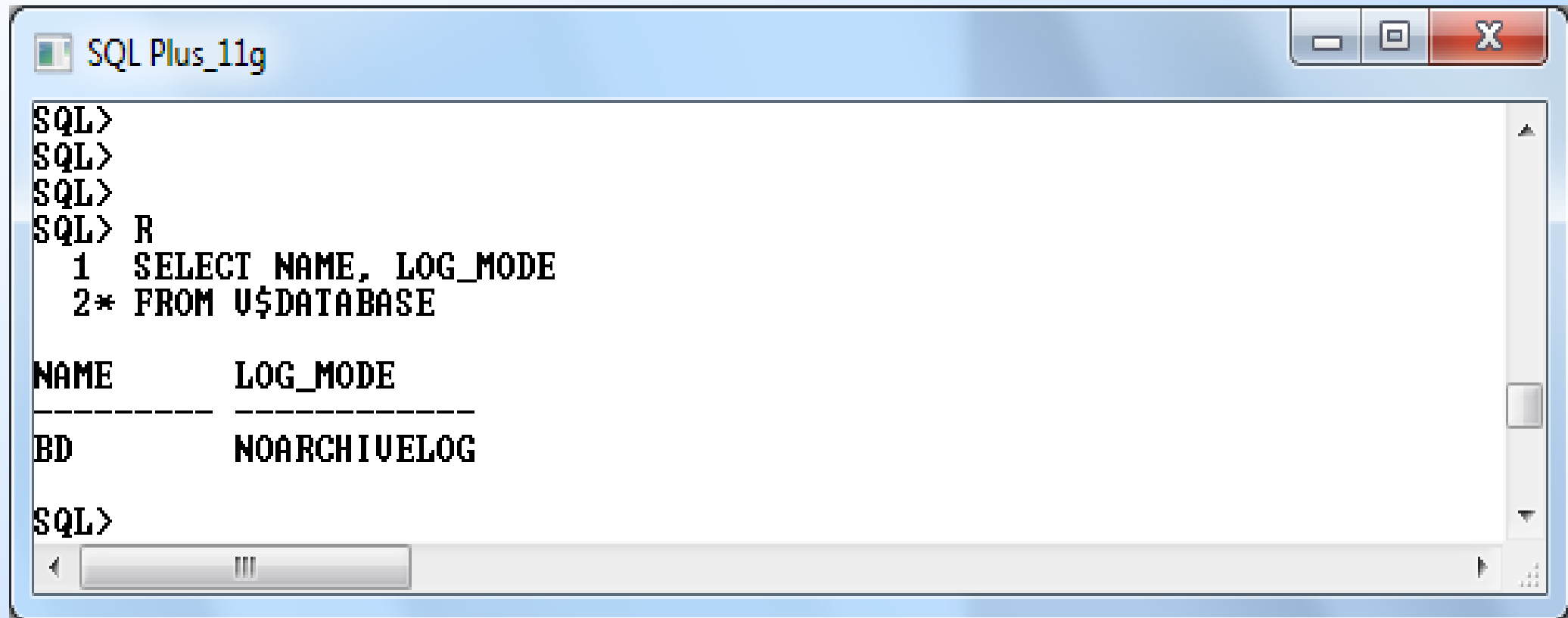
# Arhivare - cont

□ O aceeași bază de date poate fi la un moment dat într-unul din cele 2 moduri:

1. NOARCHIVELOG – în momentul în care ultimul fișier de Redo Log se umple se revine la primul care este rescris (scris-peste). Bineînțeles modificările din acesta au fost scrise și în fișierele de date
2. ARCHIVELOG – după un log switch procesele ARCH (de background) arhivează fișierele de log inactive.

# Vederi: v\$database

- ❑ Pentru a vizualiza modul in care este BD: vederea **v\$database**



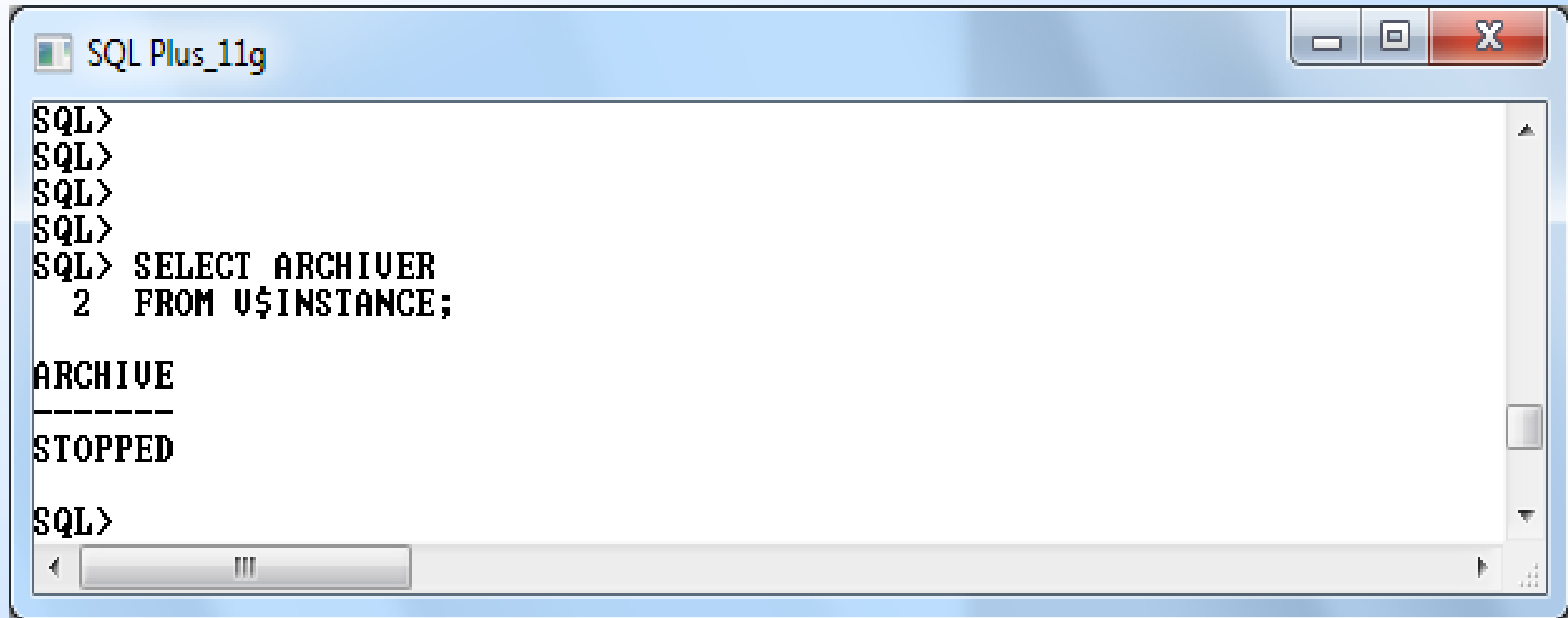
```
SQL>
SQL>
SQL>
SQL> R
  1  SELECT NAME, LOG_MODE
  2*  FROM V$DATABASE

NAME          LOG_MODE
-----
BD            NOARCHIVELOG

SQL>
```

# Vederi: v\$instance

- ❑ Pentru a vizualiza modul in care este BD: vederea **v\$instance**



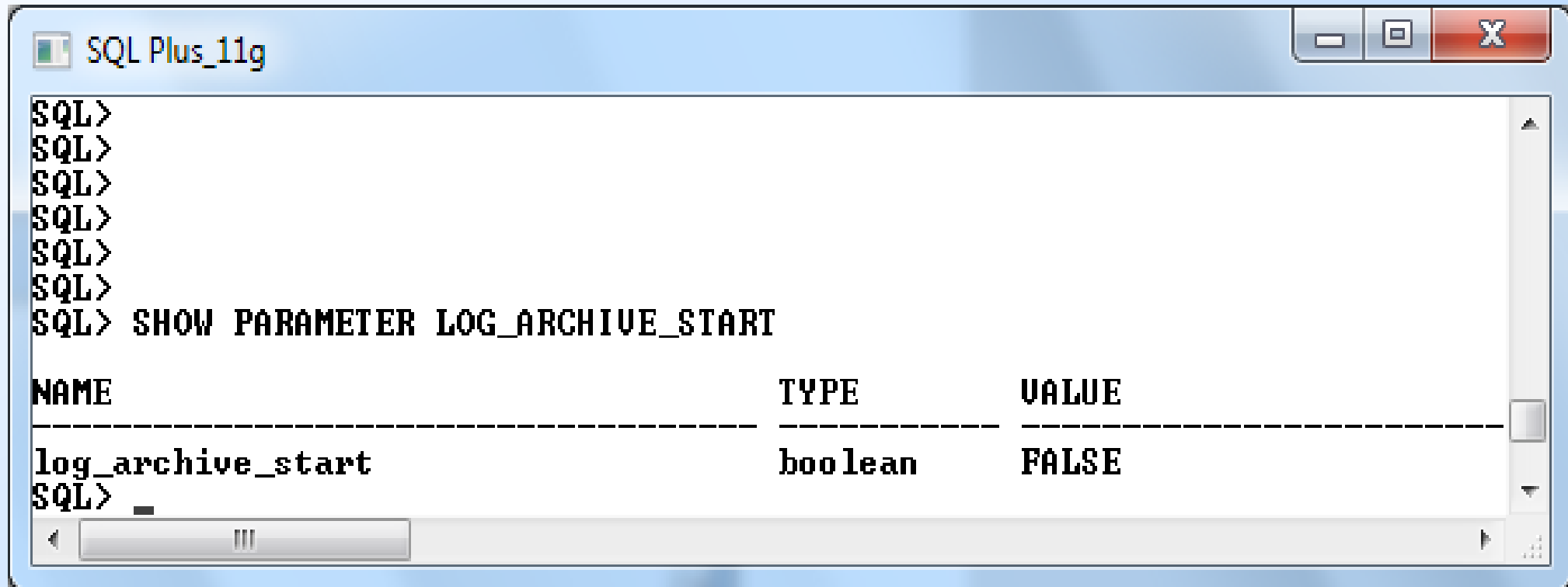
```
SQL>
SQL>
SQL>
SQL>
SQL> SELECT ARCHIVER
      2  FROM V$INSTANCE;

ARCHIVE
-----
STOPPED

SQL>
```

# Parametru

- Se poate si prin examinarea parametrului **log\_archive\_start** (comanda SQL\*Plus):



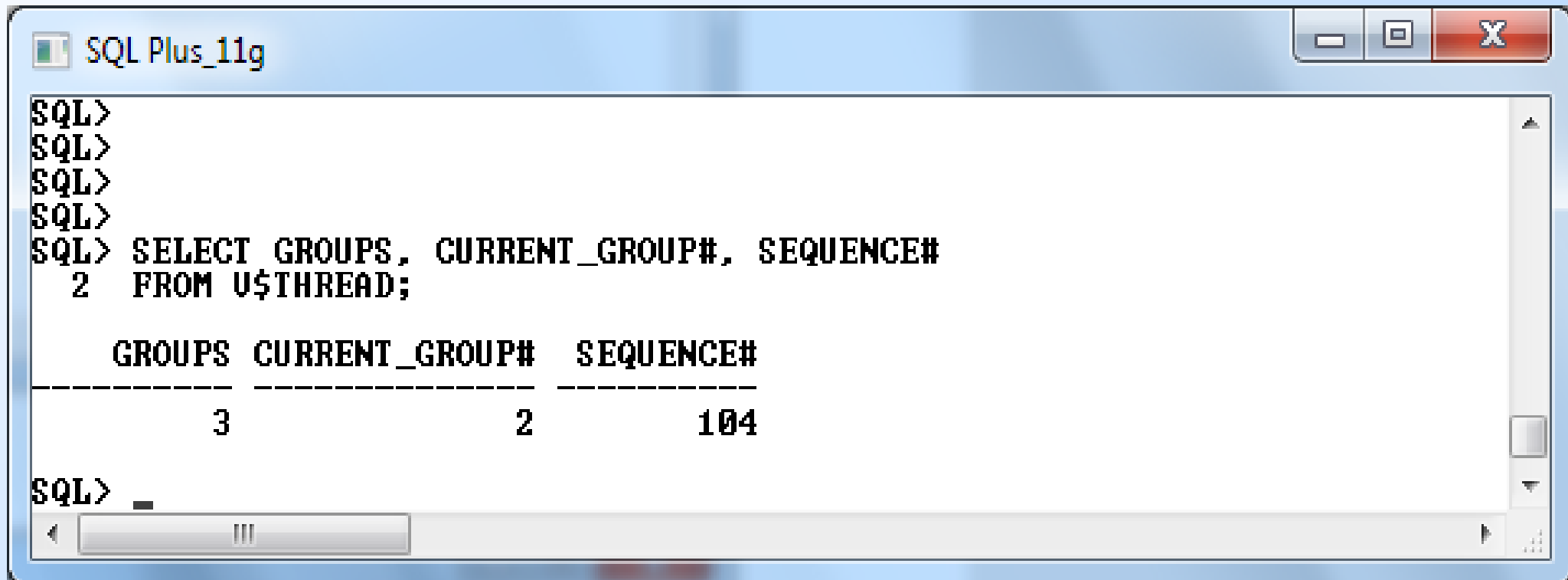
```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> SHOW PARAMETER LOG_ARCHIVE_START
```

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

```
SQL>
```

# Vederi: v\$thread

- ❑ Pentru a vizualiza care este fisierul (grupul) curent se poate interoga **v\$thread** :



```
SQL>
SQL>
SQL>
SQL>
SQL> SELECT GROUPS, CURRENT_GROUP#, SEQUENCE#
      2  FROM V$THREAD;

  GROUPS  CURRENT_GROUP#  SEQUENCE#
-----
        3                2          104

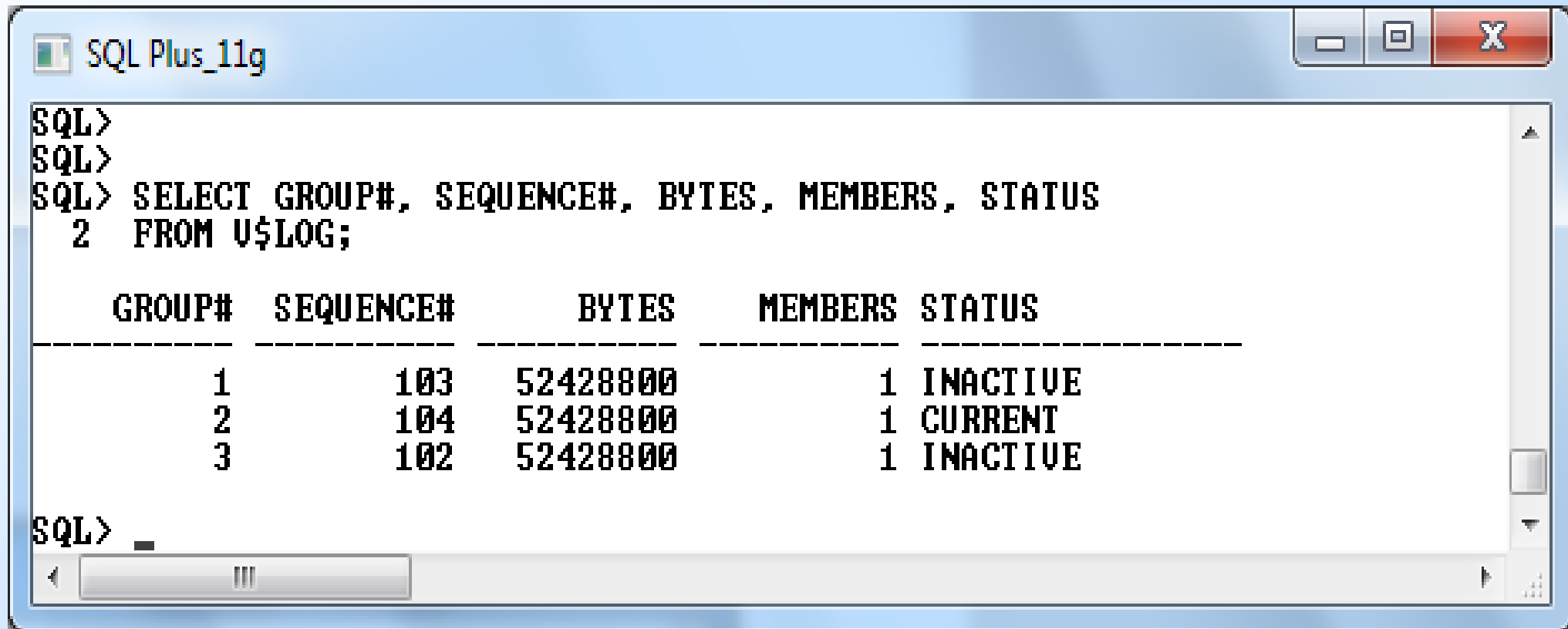
SQL> _
```

| GROUPS | CURRENT_GROUP# | SEQUENCE# |
|--------|----------------|-----------|
| 3      | 2              | 104       |



# Vederi: v\$log

- ❑ Informatiile returnate sunt din fisierele de control:



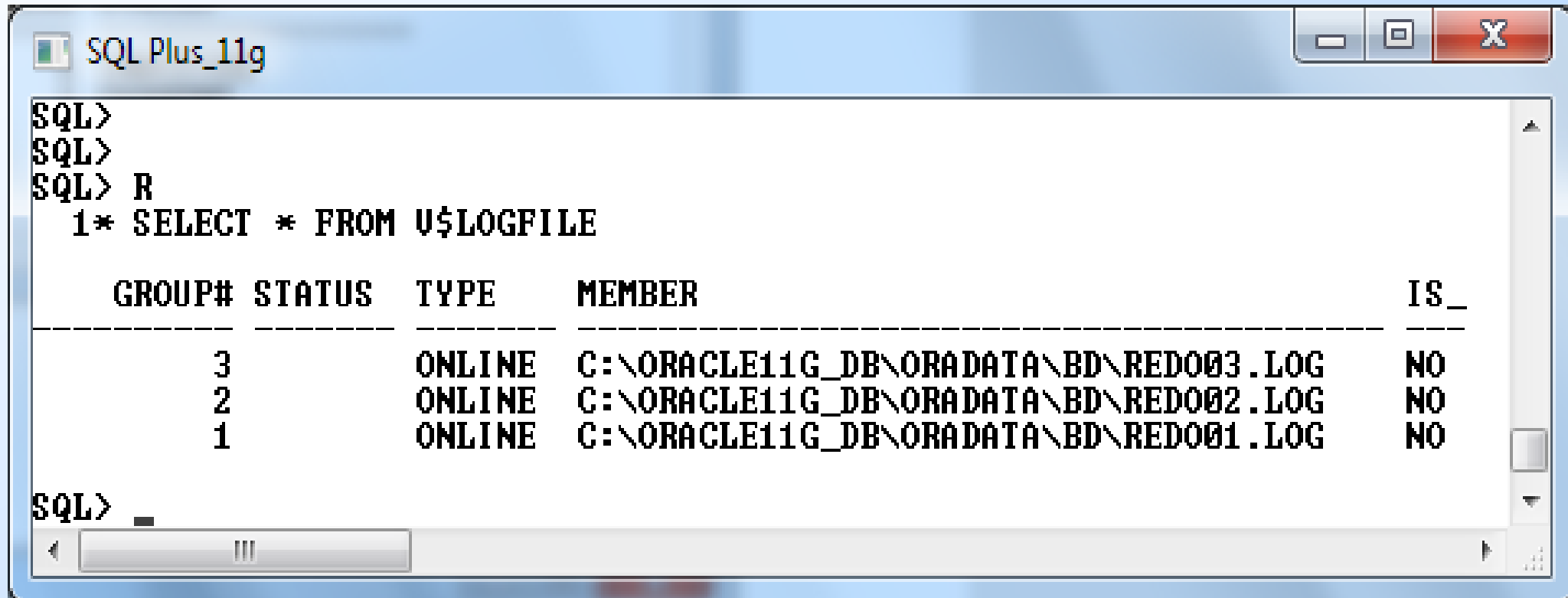
```
SQL>
SQL>
SQL> SELECT GROUP#, SEQUENCE#, BYTES, MEMBERS, STATUS
      2  FROM V$LOG;

  GROUP#  SEQUENCE#      BYTES  MEMBERS  STATUS
-----
         1         103  52428800          1  INACTIVE
         2         104  52428800          1  CURRENT
         3         102  52428800          1  INACTIVE

SQL> _
```

# Vederi: v\$logfile

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



```
SQL>
SQL>
SQL> R
1* SELECT * FROM V$LOGFILE
```

| GROUP# | STATUS | TYPE | MEMBER                                | IS_ |
|--------|--------|------|---------------------------------------|-----|
| 3      | ONLINE |      | C:\ORACLE11G_DB\ORADATA\BD\REDO03.LOG | NO  |
| 2      | ONLINE |      | C:\ORACLE11G_DB\ORADATA\BD\REDO02.LOG | NO  |
| 1      | ONLINE |      | C:\ORACLE11G_DB\ORADATA\BD\REDO01.LOG | NO  |

```
SQL>
```

# Vederi: v\$logfile

- ❑ Un fisier redo log devine INVALID daca sistemul nu il poate accesa.
- ❑ Un fisier redo log devine STALE daca sistemul suspecteaza ca nu este complet sau corect.
- ❑ Un fisier redo log STALE devine valid din nou cand grupul sau devine activ data viitoare.

# Fortarea unui log switch

□ Se face cu cererea:

**ALTER SYSTEM SWITCH LOGFILE;**

Aceeasi operatie se poate efectua si din consola de administrare

# Fortarea unui checkpoint

□ Se face cu cererea:

**ALTER SYSTEM CHECKPOINT;**

Precum si din consola de administrare

□ De asemenea, in cazul in care baza de date foloseste fisiere de tip Redo Log mari, se poate comanda si efectuarea periodica a checkpointului prin parametrii :

**LOG\_CHECKPOINT\_INTERVAL** si  
**LOG\_CHECKPOINT\_TIMEOUT**

# 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.

# 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 fișierelor Redo Log în noua locație. 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 în 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 stergerea fisierelor de pe disc ci doar actualizarea informatiilor interne ale sistemului prin eliminarea grupului respectiv.
- ❑ Putem sa utilizam apoi 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' ;**

❑ Si aici operatia nu implica 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 respectiv.

# 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.

# Bibliografie

- ❑ Oracle Database Administrator's Guide (10g) – Cap 5: Managing Control Files

[http://download.oracle.com/docs/cd/B14117\\_01/server.101/b10739/control.htm](http://download.oracle.com/docs/cd/B14117_01/server.101/b10739/control.htm)

- ❑ Oracle Database Administrator's Guide (10g) – Cap 6: Managing the Redo Log

[http://download.oracle.com/docs/cd/B14117\\_01/server.101/b10739/onlineredo.htm](http://download.oracle.com/docs/cd/B14117_01/server.101/b10739/onlineredo.htm)

Sau:

- ❑ Oracle Database Administrator's Guide (12c) – Cap 10: Managing Control Files si  
Cap 11: Managing the Redo Log în:

<https://docs.oracle.com/en/database/oracle/oracle-database/12.2/admin/database-administrators-guide.pdf>

# Sfârșitul capitolului 3

# Capitolul 4

## Fisiere de date si Tablespace

# 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  $\supset$  Extensie (extent)  $\supset$  Bloc (stocate in fisierele de date)

Din punct de vedere fizic avem:

Tablespace  $\supset$  Fisiere de date

In acest capitol vom discuta despre:

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

# Tablespace

- ❑ O baza de date contine (una sau) mai multe subdiviziuni numite 'tablespace'.
- ❑ Un tablespace apartine unei singure baze de date.
- ❑ Un tablespace poate fi stocat in unul sau mai multe fisiere de date.
- ❑ Un fisier de date apartine unui singur tablespace.
- ❑ Cu unele exceptii (SYSTEM de ex.) un tablespace poate fi trecut intre starile
  - ❑ online ↔ offline si
  - ❑ read-write ↔ read-only



# Fisiere de date

- ❑ Aceste fișiere se creează:
  - la crearea bazei de date (pentru tablespace-urile care sunt create atunci)
  - la crearea unui nou tablespace
  - La adăugarea unui nou fișier de date la un tablespace
- ❑ Dimensiunea fișierelor este specificată la creare.
- ❑ Unui fișier de date existent i se poate modifica dimensiunea ulterior.
- ❑ Unui fișier de date i se poate seta opțiunea 'AUTOEXTEND' pentru a crește automat ca dimensiune când este necesar.

# SEGMENTE

- ❑ Un tablespace contine segmente.
- ❑ Un segment contine un obiect (tabela, index, etc)
- ❑ Sunt de 4 tipuri generice (si 11 tipuri efective):
  - ❑ Segment de tip date (tabele si cluster)
  - ❑ Segment de tip index
  - ❑ Segment temporar
  - ❑ Segment de rollback
- ❑ Un segment se poate intinde pe mai multe fisiere de date care apartin aceluasi tablespace.

# SEGMENTE

□ Cele 11 tipuri de segmente sunt:

1. table
2. table partition
3. index
4. index partition
5. cluster
6. rollback
7. deferred rollback
8. temporary
9. cache
- 10.lobsegment
- 11.lobindex

# SEGMENTE - cont

- ❑ Segmentele temporare sunt in general cele folosite pentru sortari.
- ❑ Urmatoarele cereri SQL au nevoie de segment temporar in cazul in care sortarile nu pot fi efectuate in memorie:

create index

select ... order by

select distinct

select ... group by

select ... union

select ... intersect

select ... minus

analyze table

joinuri care nu folosesc indecsi

anumite subcereri corelate

# SEGMENTE - cont

- ❑ Segmentele temporare se pot stoca in orice tablespace
- ❑ Exista insa posibilitatea de a crea un tablespace temporar (temporary tablespace)
- ❑ Segmentele temporare sunt eliberate dupa folosire. Cine face asta: procesul de background SMON

# EXTENSII

- ❑ Un segment este format din una sau mai multe extensii (eng.: extent).
- ❑ O extensie e formata dintr-o succesiune **contigua** de blocuri de date (database blocks).
- ❑ O extensie se gaseste in intregime intr-un singur fisier de date dintre cele care formeaza tablespace-ul.
- ❑ Faptul ca este contigua este relevant pentru cresterea vitezei de exploatare a datelor (citire – scriere)

# EXTENSII

- ❑ Nota privind contiguitatea blocurilor unei extensii: acestea sunt blocuri logice, aparținând unui fisier de date (fișierul în care se găsește extensia).
- ❑ În documentația Oracle se precizează:
  - ❑ File system extents are not the same as Oracle Database extents.
  - ❑ File system extents are physical contiguous blocks of data written to a device as managed by the file system.
  - ❑ Oracle Database extents are logical structures managed by the database, such as tablespace extents.

([http://docs.oracle.com/cd/B28359\\_01/server.111/b28310/dfiles009.htm](http://docs.oracle.com/cd/B28359_01/server.111/b28310/dfiles009.htm))

# BLOC

- ❑ O extensie e formata din blocuri.
- ❑ Este vorba despre blocuri ale bazei de date (de dimensiune DB\_BLOCK\_SIZE)
- ❑ Un astfel de bloc poate fi format din unul sau mai multe blocuri fizice (de disc)
- ❑ Un bloc este cea mai mica unitate de intrare – iesire pentru SGBD.



# Revenim la TABLESPACE

- Avantajele folosirii mai multor tablespace-uri:
  - Se pot separa datele user de datele de sistem (prin stocarea in tablespace-uri diferite). In felul acesta se micsoreaza si traficul de date pe tablespace-urile de sistem.
  - Se pot separa datele unei aplicatii de ale alteia (prin stocarea in tablespace-uri diferite). In cazul in care un tablespace trece in starea offline – din diverse motive – doar o aplicatie va avea de suferit.
  - Se pot stoca pe discuri diferite, micsorand astfel traficul de date pentru fiecare disc in parte.

# Avantaje - cont

- ❑ Se poate optimiza utilizarea tablespace-urilor prin crearea de tablespace-uri dedicate:
  - Unele pentru aplicatii update-intensive
  - Altele pentru exploatare read-only
  - Altele pentru date temporare (segmente temporare)
- ❑ Se pot efectua operatii de salvare la nivel de tablespace deci se pot astfel salva doar datele aferente unor aplicatii importante care ruleaza in sistem.

# Tablespace-ul SYSTEM

- ❑ La crearea bazei de date se creaza automat tablespace-ul **SYSTEM** care contine printre altele dictionarul de date al sistemului si segmentul de rollback de sistem.
- ❑ Acesta este primul tablespace creat si are caracteristici speciale:
  - Nu poate fi redenumit
  - Nu poate fi sters
  - Nu poate fi trecut in starea offline
  - Necesita privilegii sporite pentru operare

# Tablespace-ul SYSAUX

- ❑ La crearea bazei de date se creaza de asemenea si tablespace-ul SYSAUX care contine informatii despre schemele de date folosite de uneltele Oracle – astfel ele nu vor avea nevoie de un alt tablespace suplimentar.
- ❑ Acesta are de asemenea caracteristici speciale:
  - Nu poate fi redenumit
  - Nu poate fi sters
  - Nu poate fi trecut in starea offline
  - Necesita privilegii sporite pentru operare

# Clasificare

- Un tablespace poate fi – din punct de vedere al datelor continute - de unul dintre urmatoarele tipuri:
  - Permanent – sunt tablespace-urile uzuale, inclusiv cele de sistem
  - Temporar – contin segmente temporare – am vorbit despre ele
  - De tip Undo – introduse incepand cu versiunea 9i – contin segmente de undo (rollback), necesare in cazul revocarii operatiilor de actualizare. Anterior versiunii 9i existau doar segmente de rollback – nu si tablespace.

# Alta clasificare: DMT si LMT

- ❑ Fiecare Tablespace este format dintr-o multime de extensii (continute in segmentele componente).
- ❑ Gestiunea acestora (care sunt libere si care sunt ocupate) se poate face in doua feluri: fie informatiile respective se stocheaza in dictionarul de date fie se memoreaza in tablespace
- ❑ Tablespace-urile pentru care gestiunea se face prin intermediul dictionarului de date (o solutie costisitoare ca timp) se numesc **DMT - dictionary managed tablespaces**

# DMT si LMT - cont

- ❑ Tablespace-urile pentru care gestiunea se face local, prin stocarea datelor privind starea extensiilor in interiorul tablespace-ului se numesc **LMT - locally managed tablespaces**
- ❑ Informatiile se tin in headerul acestuia, de fapt in headerul fiecarui fisier de date component
- ❑ In acest caz headerul contine un bitmap unde fiecare bit este un bloc sau un grup de blocuri. Bitul arata daca zona respectiva este ocupata sau nu.



# Exemplu

- ❑ Pentru a afla informatii despre tablespace-urile existente, se poate interoga vederea dba\_tablespaces din dictionarul de date al sistemului.
- ❑ Un exemplu de cerere este urmatorul:

```
SQL> select tablespace_name,  
        extent_management, allocation_type  
from dba_tablespaces;
```



# Exemplu

| TABLESPACE_NAME   | EXTENT_MAN | ALLOCATIO |
|-------------------|------------|-----------|
| -----             | -----      | -----     |
| SYSTEM            | LOCAL      | SYSTEM    |
| SYSAUX            | LOCAL      | SYSTEM    |
| UNDOTBS1          | LOCAL      | SYSTEM    |
| TEMP              | LOCAL      | UNIFORM   |
| USERS             | LOCAL      | SYSTEM    |
| EXAMPLE           | LOCAL      | SYSTEM    |
| BD_DATA           | LOCAL      | SYSTEM    |
| BD_TEMP           | LOCAL      | UNIFORM   |
| .....             |            |           |
| TEMPORARY_INDEXES | LOCAL      | UNIFORM   |
| TEMPORARY_TABLES  | LOCAL      | UNIFORM   |
| VERSION_INDEXES   | LOCAL      | UNIFORM   |
| VERSION_TABLES    | LOCAL      | UNIFORM   |
| REPOS_TEMP        | LOCAL      | UNIFORM   |

23 înregistrări selectate.

# Sintaxa CREATE TABLESPACE

```
CREATE [TEMPORARY / UNDO] TABLESPACE <tblspc_name>
DATAFILE / TEMPFILE
    '<datafile01_name and Path where file to create>' SIZE <intM>[, '
    <datafile02_name and Path where file to create>' SIZE <intM>[, '
    <datafile0N_name and Path where file to create>' SIZE <intM>[,...]]]
BLOCKSIZE <DB_BLOCK_SIZE parameter /2k/4k/8k/16k/32k >
AUTOEXTEND { [OFF/ON (NEXT <integer K/M > MAXSIZE<integer K/M >) / UNLIMITED] }
LOGGING/NOLOGGING (implicit: logging)
FORCE LOGGING
ONLINE/OFFLINE (implicit: online)
SEGMENT SPACE MANAGEMENT { AUTO | MANUAL }
FLASHBACK ON | OFF
EXTENT MANAGEMENT { [DICTIONARY] /
    [LOCAL (AUTOALLOCATE / UNIFORM <integer K/M >)] }
PERMANENT / TEMPORARY (implicit: permanent)
MINIMUM EXTENT integer
<Clauza DEFAULT STORAGE> -- vezi slide urmator
NOCACHE;
```

# Clauza DEFAULT STORAGE

```
DEFAULT STORAGE { [INITIAL <integer K/M >]  
                  [NEXT <integer K/M >]  
                  [PCTINCREASE <integer K/M >]  
                  [MINEXTENTS <integer>]  
                  [MAXEXTENTS <integer> / UNLIMITED]  
                  [FREELISTS <integer>]  
                  [FREELIST GROUPS <integer>]  
                  [OPTIMAL <integer>/NULL]  
                  [BUFFER_POOL < DEFAULT/KEEP/RECYCLE >] }
```

Are efect in dimensionarea extensiilor.

(pentru crearea unui tablespace vezi si: <http://www.orafaq.com/wiki/Tablespace>)

# EXTENT MANAGEMENT

❑ **EXTENT MANAGEMENT {DICTIONARY |  
LOCAL {AUTOALLOCATE | UNIFORM [SIZE int K | M]} }**

Aceasta optiune arata daca acel tablespace va fi de tip DMT sau LMT

In cazul LMT se poate specifica suplimentar:

- ❑ optiunea AUTOALLOCATE (cand se interogheaza vederile din dictionarul de date se afiseaza in acest caz 'SYSTEM'):
- tablespace-ul va contine extensii de dimensiuni diferite, gestiunea fiind facuta automat de catre sistem.

# EXTENT MANAGEMENT

- optiunea AUTOALLOCATE – continuare
  - Aceasta optiune este buna atunci cand in acel tablespace vor fi stocate obiecte (segmente) de dimensiuni variabile, fiecare putand avea mai multe extensii.
  - Este un mod simplificat de gestiune (pt. ca e facuta de sistem) dar poate duce uneori la imobilizarea unor spatii pe disc.
  - Dimensiunea minima a unei extensii este de 64K. Daca blocul de date al BD este 16K sau mai mare atunci dimensiunea minima a unei extensii este de 1M.

# EXTENT MANAGEMENT

## □ optiunea UNIFORM

- Specifica faptul ca acel tablespace este gestionat folosindu-se extensii de dimensiune fixa.
- Fiecare extensie trebuie sa aiba minim 5 blocuri (blocuri BD!). Deci:
  - Daca blocul este de 8192 octeti (8K) atunci dimensiunea minima pentru UNIFORM este de 40K.
  - Pentru 16384 octeti (16K) minimul pentru UNIFORM este 80K.

# EXTENT MANAGEMENT

- optiunea UNIFORM - continuare
  - UNIFORM nu este o optiune valida pentru tablespace-ul SYSTEM
  - Aceasta optiune permite o alocare mai precisa a spatiului astfel incat sa se minimizeze pierderile de spatiu pe disc.
  - Se foloseste atunci cand avem o estimare asupra spatiului ocupat de fiecare obiect din acel tablespace.



# Exemple

Exemple:

## ❑ Cazul AUTOALLOCATE:

```
CREATE TABLESPACE user  
DATAFILE '/u02/oracle/data/user01.dbf' SIZE 50M  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

## ❑ Cazul UNIFORM:

```
CREATE TABLESPACE user  
DATAFILE '/u02/oracle/data/user01.dbf' SIZE 50M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```



# Sintaxa CREATE TABLESPACE

CREATE [TEMPORARY / UNDO] TABLESPACE <tblspc\_name>  
DATAFILE / TEMPFILE

'<datafile01\_name and Path where file to create>' SIZE <intM>[, '  
<datafile02\_name and Path where file to create>' SIZE <intM>[, '  
<datafile0N\_name and Path where file to create>' SIZE <intM>[,...]]]

**BLOCKSIZE** <DB\_BLOCK\_SIZE parameter /2k/4k/8k/16k/32k >

AUTOEXTEND { [OFF/ON (NEXT <integer K/M > MAXSIZE<integer K/M >) / UNLIMITED] }

**LOGGING/NOLOGGING** (implicit: logging)

**FORCE LOGGING**

**ONLINE/OFFLINE** (implicit: online)

SEGMENT SPACE MANAGEMENT { AUTO | MANUAL }

**FLASHBACK ON | OFF**

EXTENT MANAGEMENT { [DICTIONARY] /  
[LOCAL (AUTOALLOCATE / UNIFORM <integer K/M >)] }

**PERMANENT / TEMPORARY** (implicit: permanent)

**MINIMUM EXTENT** integer

**<Clauza DEFAULT STORAGE>**

**NOCACHE;**

# CREATE TABLESPACE - cont

- ❑ MINIMUM EXTENT *int* {K|M} - arata dimensiunea minima a unei extensii (in KB sau MB dupa cum dupa numar urmeaza K sau M). O extensie este de acea dimensiune sau **multiplu** de acea dimensiune. Se foloseste pentru a impiedica o prea mare fragmentare a spatiului.
- ❑ BLOCKSIZE *int* K – se poate specifica o dimensiune nonstandard a blocului pentru acel tablespace. E legat si de alti parametri care trebuiesc setati.
- ❑ LOGGING | NOLOGGING – anumite operatii (cum ar fi crearea unui index sau incarcarea de date cu loaderul) nu sunt inregistrate in fisierele Redo Log in caz de NOLOGGING. Se aplica obiectelor din acel tablespace. Nu este recomandat!

# CREATE TABLESPACE - cont

- ❑ FORCE LOGGING – Se forteaza inregistrarea in Redo Log a modificarilor pe obiectele din acel tablespace chiar daca ele au fost create cu NOLOGGING.
- ❑ ONLINE | OFFLINE – in cazul OFFLINE acel tablespace nu este disponibil imediat dupa creare (trebuie ca ulterior sa fie adus in starea online)
- ❑ PERMANENT | TEMPORARY – tablespace permanent sau temporar.

# CREATE TABLESPACE - cont

## □ FLASHBACK ON | OFF

Se foloseste in conjunctie cu operatii de tip:

- ALTER DATABASE FLASHBACK ON si
- FLASHBACK DATABASE TO

pentru a readuce baza de date la o stare anterioara.

Un exemplu ilustrativ se gaseste la adresa:

<http://www.orafaq.com/node/1847>

# DEFAULT STORAGE

Clauza Storage arata cum va stoca Oracle fiecare obiect in acel tablespace.  
Optiunile sale sunt:

- **INITIAL *int* K | M**
- **NEXT *int* K | M**
- **MINEXTENTS *int***
- **MAXEXTENTS *int* | UNLIMITED**
- **PCTINCREASE *int***
- **FREELISTS *int***
- **FREELIST GROUPS *int***
- **OPTIMAL *int* | NULL |**
- **BUFFER\_POOL {KEEP | RECYCLE | DEFAULT}**

# DEFAULT STORAGE

Detaliiere:

- ❑ **INITIAL *int* K | M** – definește dimensiunea primei extensii (minim 2 blocuri). Valoarea implicită este 5 blocuri ale BD.
- ❑ **NEXT *int* K | M** – da dimensiunea celei de-a doua extensii. Valoarea minimă este de 1 bloc, valoarea implicită este de asemenea 5 blocuri.
- ❑ **MINEXTENTS *int*** - este numărul de extensii care sunt alocate când segmentul este creat. Valoarea minimă – și implicită – este 1.

# DEFAULT STORAGE

## Clauza Storage - cont

- ❑ **MAXEXTENTS** *int* – determina numarul maxim de extensii pe care le poate avea un segment. Valoarea minima este 1 iar valoarea maxima depinde de dimensiunea blocului.
- ❑ **MAXEXTENTS UNLIMITED** – este echivalenta cu 2G extensii
- ❑ **PCTINCREASE** *int* – este procentul cu care creste dimensiunea extensiilor. Valoarea minima este 0, cea implicita 50.



# DEFAULT STORAGE

- Exista o formula care ne da dimensiunea extensiei cu numarul n:

$$\text{Size}(n) = \text{NEXT} * (1 + \text{PCTINCREASE}/100)^{(n-2)}$$

Deci daca NEXT = 200K iar PCTINCREASE este 50 atunci

- Size(2) = 200K,
- Size(3) = 300K,
- Size(4) = 450K, etc



# DEFAULT STORAGE

- ❑ Optiunile FREELISTS int si FREELIST GROUPS int sunt legate de clauza:  
    SEGMENT SPACE MANAGEMENT {MANUAL | AUTO}
- ❑ Aceasta clauza spune cum este gestionat spatiul liber dintr-un segment:
  - MANUAL
  - AUTO

# Sintaxa CREATE TABLESPACE

```
CREATE [TEMPORARY / UNDO] TABLESPACE <tblspc_name>
DATAFILE / TEMPFILE
    '<datafile01_name and Path where file to create>' SIZE <intM>[, '
    <datafile02_name and Path where file to create>' SIZE <intM>[, '
    <datafile0N_name and Path where file to create>' SIZE <intM>[,...]]]
BLOCKSIZE <DB_BLOCK_SIZE parameter /2k/4k/8k/16k/32k >
AUTOEXTEND { [OFF/ON (NEXT <integer K/M > MAXSIZE<integer K/M >) / UNLIMITED] }
LOGGING/NOLOGGING (implicit: logging)
FORCE LOGGING
ONLINE/OFFLINE (implicit: online)
SEGMENT SPACE MANAGEMENT { AUTO | MANUAL }
FLASHBACK ON | OFF
EXTENT MANAGEMENT { [DICTIONARY] /
    [LOCAL (AUTOALLOCATE / UNIFORM <integer K/M >)] }
PERMANENT / TEMPORARY (implicit: permanent)
MINIMUM EXTENT integer
<Clauza DEFAULT STORAGE>
NOCACHE;
```

# SEGMENT SPACE MANAGEMENT

- ❑ MANUAL: Sunt utilizate liste ale spatiului liber pentru gestiunea acestuia. Acestea sunt liste de blocuri care contin spatiu disponibil pentru noi operatii de INSERT.
- ❑ MANUAL este valoarea implicita pentru aceasta clauza. In acest caz FREELISTS este un parametru care specifica numarul de liste de blocuri care pot primi inregistrari. In aplicatii de tip paralel sau distribuit se folosesc grupuri de liste (cate unul pentru fiecare nod).
- ❑ AUTO: In acest caz sunt utilizate bitmapuri pentru spatiul liber din segmente. Acestea permit o gestiune automata a spatiului disponibil.
- ❑ Optiunea AUTO poate fi lenta insa in cazul in care se fac multe actualizari.

# SEGMENT SPACE MANAGEMENT

- ❑ In cazul LMT nu este nevoie de a specifica in CREATE sau ALTER optiuni de stocare (gestiunea facandu-se automat).
- ❑ Deci nu vor aparea clauzele:
  - next
  - pctincrease
  - minextents
  - maxextents
  - default storage
- ❑ In cazul DMT insa aceste clauze pot sa apara atat la crearea unui tablespace cat si in comanda de modificare ALTER TABLESPACE

# CREATE TABLESPACE - cont

Daca nu exista clauza EXTENT MANAGEMENT atunci pentru determinarea tipului (DMT sau LMT) se folosesc informatiile de compatibilitate (parametru in fisierul init.ora) precum si clauzele MINIMUM EXTENT si DEFAULT clauza\_storage astfel:

1. If compatibil < 9.0.0 se creeaza un tablespace DMT
2. If compatibil >= 9.0.0 si **DEFAULT clauza\_storage** NU a fost specificata se creeaza un LMT cu AUTOALLOCATE.

# CREATE TABLESPACE - cont

3. If compatibil  $\geq 9.0.0$  si clauza ***DEFAULT clauza\_storage*** a fost specificata si
- ❑ MINIMUM EXTENT a fost specificata atunci:
    - a. Daca MINIMUM EXTENT, INITIAL, si NEXT sunt egale intre ele iar PCTINCREASE = 0 atunci se creeaza un LMT cu UNIFORM avand dimensiune extensie = INITIAL
    - b. MINIMUM EXTENT, INITIAL si NEXT nu sunt egale SAU PCTINCREASE nu este 0 atunci se creeaza un LMT cu AUTOALLOCATE.
  - ❑ MINIMUM EXTENT nu a fost specificata atunci:
    - ❑ Daca INITIAL si NEXT sunt egale iar PCTINCREASE = 0 atunci LMT cu UNIFORM
    - ❑ Altfel LMT cu AUTOALLOCATE.

# Exemple

❑ Tablespace permanent:

```
create tablespace TS1
```

```
logging
```

```
datafile '/home/oracle/data/ts1.dbf' size 16m autoextend on next 32m  
maxsize 2048m
```

```
extent management local; -- default: autoallocate
```

**Sau:**

```
create tablespace TSDATE
```

```
datafile '/home/oracle/data/date.dbf' size 10M autoextend on maxsize  
200M
```

```
extent management local uniform size 64K;
```



# Exemple

- ❑ Tablespace permanent cu mai multe fisiere de date:

```
create tablespace TS1
datafile '/home/oracle/data/ts1.dbf' size 16m autoextend on
        next 16m maxsize 2048m,
'/home/oracle/data/ts2.dbf' size 32m autoextend on next 32m
        maxsize 2048m,
'/home/oracle/data/ts3.dbf' size 64m autoextend on next 64m
        maxsize 2048m,
extent management local; -- default: autoallocate
```



# Exemple

- ❑ Tablespace temporar:

```
create temporary tablespace ttemp  
tempfile '/home/oracle/temp/temp01.dbf' size 16m autoextend  
on next 16m maxsize 2048m  
extent management local;
```

- ❑ Tablespace de tip UNDO:

```
create undo tablespace tsundo  
datafile '/home/oracle/data/undo.dbf'  
size 100M;
```

# ALTER TABLESPACE

□ Sintaxa (partiala) este:

```
ALTER TABLESPACE tablespace
{ ADD DATAFILE { filespec
  [AUTOEXTEND [ OFF | ON [NEXT integer [K|M]] [MAXSIZE
  {UNLIMITED | integer[K|M] } [, filespec ...] }
| RENAME DATAFILE 'filename' [, 'filename'] ... TO
  'filename' [, 'filename'] ...
| DEFAULT STORAGE storage_clause
| ONLINE
| OFFLINE [{NORMAL | TEMPORARY | IMMEDIATE}]
| READ ONLY
| READ WRITE
| {BEGIN | END} BACKUP
```

# Adaugare fisier

- ❑ Se adauga un nou fisier de date la un tablespace folosind cereri de tip ALTER TABLESPACE:

```
ALTER TABLESPACE user  
ADD DATAFILE  
    '/u02/oracle/data/user01.dbf' SIZE 50M
```

# Adaugare fisier - cont

- ❑ Se pot adauga mai multe fisiere cu aceeasi comanda:

```
ALTER TABLESPACE user  
ADD DATAFILE '/u02/oracle/data/user01.dbf' SIZE  
50M,  
' /u02/oracle/data/user02.dbf' SIZE 50M,  
' /u02/oracle/data/user03.dbf' SIZE 50M,
```

- ❑ Obs: Daca nu se specifica in comanda calea, Oracle creeaza fisierele in directorul default al serverului.

# AUTOEXTEND

- Clauza AUTOEXTEND permite / inhiba extinderea automata a fisierelor de date:
  - AUTOEXTEND OFF inhiba cresterea automata a acestora in dimensiune
  - In cazul AUTOEXTEND ON fisierele de date se extind automat la nevoie

# AUTOEXTEND - cont

- ❑ NEXT specifica dimensiunea minima a incrementului (in Kb sau Mb) in cazul in care sunt necesare noi extensii (extents) si spatiul disponibil din fisier nu este suficient pentru acestea.

Valoarea implicita pentru NEXT este de 1 bloc al BD

- ❑ MAXSIZE specifica dimensiunea maxima a spatiului care se poate alocata pentru acel fisier de date (pana la ce dimensiune poate creste).
- ❑ UNLIMITED specifica faptul ca nu este setata o dimensiune maxima permisa (se poate extinde oricat, in limita spatiului existent).

# Exemplu

```
ALTER TABLESPACE user  
ADD DATAFILE '/u02/oracle/data/user01.dbf' SIZE 200M  
AUTOEXTEND ON  
NEXT 10M  
MAXSIZE 500M
```

Clauza AUTOEXTEND poate fi prezenta in cererile:

- ☐ CREATE DATABASE
- ☐ ALTER DATABASE
- ☐ CREATE TABLESPACE
- ☐ ALTER TABLESPACE

# Specificare AUTOEXTEND pentru fisier existent:

❑ Se face folosind ALTER DATABASE:

```
ALTER DATABASE ora  
DATAFILE '/u02/oracle/data/user01.dbf '  
AUTOEXTEND ON  
NEXT 10M  
MAXSIZE 500M
```



# RESIZE

- ❑ Pentru schimbarea manuala a dimensiunii unui fisier (marire sau micșorare) se poate folosi ALTER DATABASE. In clauza DATAFILE pot fi prezente mai multe nume de fisiere (sunt toate afectate):

```
ALTER DATABASE ora  
DATAFILE '/u02/oracle/data/user01.dbf'  
RESIZE 500M
```

- ❑ Pentru cazul micșorarii dimensiunii, aceasta se poate face doar cu spatiul liber de la sfarsitul fisierului (daca exista!)

# ONLINE / OFFLINE

- Sintaxa clauzei:

**ONLINE | OFFLINE [{NORMAL | TEMPORARY | IMMEDIATE  
}]**

- Trecerea in modul ONLINE aduce un tablespace care nu era asa in mod online.
- OFFLINE este optiunea inversa, caz in care se inhiba accesul la acel tablespace si la segmentele care se afla in el.

# ONLINE / OFFLINE - cont

- Trecerea OFFLINE se poate face in trei feluri:
  1. NORMAL – se executa checkpoint pentru toate fisierele de date din acel tablespace. Aceste fisiere trebuie sa fie toate online (si fisierele individuale pot fi trecute offline!)
    - In cazul NORMAL, la revenirea online nu este necesar sa se execute operatii de recovery.
    - NORMAL este valoarea implicita (in caz in care la trecerea OFFLINE nu se specifica nici una din cele trei optiuni).

# ONLINE / OFFLINE - cont

2. In cazul TEMPORARY se face checkpoint pentru toate fisierele de date care sunt online dar Oracle nu se asigura ca toate fisierele pot fi scrise.
  - Orice fisier care e in acel moment offline poate avea nevoie de recovery cand revenim online.
3. IMMEDIATE nu face checkpoint si nici nu verifica daca fisierele sunt disponibile sau nu. La revenirea online este nevoie de recovery.

# Exemple

**ALTER TABLESPACE user ONLINE**

**ALTER TABLESPACE user OFFLINE**

**ALTER TABLESPACE user OFFLINE TEMPORARY**

**ALTER TABLESPACE user OFFLINE IMMEDIATE**

# Read Only – Read Write

- ❑ READ ONLY specifica faptul ca nu sunt permise operatii de scriere in acel tablespace.
- ❑ Inainte de a trece un tablespace in acest mod trebuie sa fie indeplinite urmatoarele:
  - Acel tablespace trebuie sa fie online.
  - Nu trebuie sa existe tranzactii active in baza de date respectiva.
  - Acel tablespace nu trebuie sa contina segmente active de rollback.

# Read Only – Read Write

- ❑ Conditii de trecere R/O – cont:
  - Acel tablespace nu trebuie sa fie implicat in acel moment intr-o operatie de salvare (online backup).
  - Parametrul de initializare COMPATIBLE trebuie setat la versiunea 7.1.0 sau la una ulterioara acesteia.
- ❑ READ WRITE specifica faptul ca acel tablespace revine din starea de READ ONLY in starea READ WRITE in care poate fi si scris.
- ❑ In acest caz toate fisierele de date ale acelui tablespace trebuie sa fie online.

# Mutarea fisierelor de date

- Fisierele de date ale unui tablespace pot fi mutate astfel:
  1. Se trece acel tablespace offline.
  2. Cu comenzi SO copiem fisierele in noua locatie.
  3. Se executa ALTER TABLESPACE RENAME.
  4. Se readuce acel tablespace online.
  5. Se pot apoi sterge vechile fisiere de date cu comenzi SO.



# Mutarea fisierelor de date - cont

- Iata un exemplu de comanda:

```
ALTER TABLESPACE user  
RENAME DATAFILE  
    '/u02/oracle/data/user01.dbf' TO  
    '/u15/oracle/data/user01.dbf'
```

- Oracle nu face efectiv vreo redenumire de fisiere ci doar inlocuieste in fisierele de control vechiul nume de fisier cu cel nou.

# Mutarea fisierelor – v2

- Exista si posibilitatea de a muta fisierele de date cu comanda ALTER DATABASE. Pentru aceasta:
  1. Se opreste baza de date.
  2. Se muta fisierele cu comenzi SO.
  3. Se monteaza baza.
  4. Se executa ALTER DATABASE RENAME FILE.
  5. Se deschide baza.

# Mutarea fisierelor de date - cont

- Iata un exemplu de comanda:

```
ALTER DATABASE ora  
RENAME FILE  
    '/u02/oracle/data/user01.dbf' TO  
    '/u15/oracle/data/user01.dbf'
```

- La fel ca inainte, Oracle nu face efectiv vreo redenumire de fisiere ci doar inlocuieste in fisierele de control vechiul nume de fisier cu cel nou.
- In ambele cazuri se pot redenumi cu o singura comanda mai multe fisiere (RENAME FILE lista-old TO lista-new).

# Stergere TABLESPACE

- ❑ Se face cu DROP TABLESPACE.
- ❑ Sintaxa:

```
DROP TABLESPACE nume  
[INCLUDING CONTENTS  
[CASCADE CONSTRAINTS ]  
]
```

# Stergere TABLESPACE

- ❑ INCLUDING CONTENTS specifica faptul ca se sterg inclusiv acele tablespace-uri care contin date (altfel acestea nu pot fi sterse).
- ❑ CASCADE CONSTRAINTS – sterge si constrangerile referentiale aferente obiectelor din acel tablespace.
- ❑ In cazul in care CASCADE CONSTRAINTS este omisa si exista astfel de constrangeri Oracle va returna o eroare si nu va efectua stergerea.

# VEDERI

- ❑ Exista mai multe vederi care pot fi interogate pentru a obtine informatii despre tablespace-uri.
- ❑ Una dintre ele este DBA\_TABLESPACES. Iata un program de vizualizare:

```
set linesize 250
col "%INC" for 9999
col TABLESPACE_NAME for A21
col EXT_MAN for A6
col STATUS for A6
Col IN_EX for 9999
Col NX_EX for 9999
col minext for 9999
col blksize for 99999
```

# VEDERI

```
select tablespace_name, logging, force_logging FLOG,  
       block_size blksz, status, contents,  
       extent_management ext_man,  
       segment_space_management, allocation_type,  
       initial_extent/1024 in_ex, next_extent/1024  
       nx_ex, pct_increase "%INC", min_extents minext,  
       max_extents/1024 max_ext_db, min_extlen  
from DBA_TABLESPACES  
order by 1;
```



SQL> SET PAGESIZE 300  
SQL> /

| TABSPACE_NAME         | LOGGING   | FLO | BLKSZ | STATUS | CONTENTS  | EXT_MA | SEGMENT | ALLOCATION | IN_EX | NX_EX | %INC | MINEXT | MAX_EXT_DB | MIN_EXTLEN |
|-----------------------|-----------|-----|-------|--------|-----------|--------|---------|------------|-------|-------|------|--------|------------|------------|
| BD_DATA               | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | SYSTEM     | 64    |       |      | 1      | 2097152    | 65536      |
| BD_TEMP               | NOLOGGING | NO  | 8192  | ONLINE | TEMPORARY | LOCAL  | MANUAL  | UNIFORM    | 1024  | 1024  | 0    | 1      |            | 1048576    |
| CONSTANT_GROW_INDEXES | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| CONSTANT_GROW_TABLES  | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| DEPENDENCY_INDEXES    | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| DEPENDENCY_TABLES     | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| DIAGRAM_INDEXES       | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| DIAGRAM_TABLES        | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| EXAMPLE               | NOLOGGING | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | SYSTEM     | 64    |       |      | 1      | 2097152    | 65536      |
| LOB_DATA              | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| RAPID_GROW_INDEXES    | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| REPOS_TEMP            | NOLOGGING | NO  | 8192  | ONLINE | TEMPORARY | LOCAL  | MANUAL  | UNIFORM    | 1024  | 1024  | 0    | 1      |            | 1048576    |
| SYS_AUX               | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | SYSTEM     | 64    |       |      | 1      | 2097152    | 65536      |
| SYSTEM                | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | MANUAL  | SYSTEM     | 64    |       |      | 1      | 2097152    | 65536      |
| SYSTEM_META_INDEXES   | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| SYSTEM_META_TABLES    | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| TEMP                  | NOLOGGING | NO  | 8192  | ONLINE | TEMPORARY | LOCAL  | MANUAL  | UNIFORM    | 1024  | 1024  | 0    | 1      |            | 1048576    |
| TEMPORARY_INDEXES     | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| TEMPORARY_TABLES      | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| UNDOTBS1              | LOGGING   | NO  | 8192  | ONLINE | UNDO      | LOCAL  | MANUAL  | SYSTEM     | 64    |       |      | 1      | 2097152    | 65536      |
| USERS                 | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | SYSTEM     | 64    |       |      | 1      | 2097152    | 65536      |
| VERSION_INDEXES       | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |
| VERSION_TABLES        | LOGGING   | NO  | 8192  | ONLINE | PERMANENT | LOCAL  | AUTO    | UNIFORM    | 104   | 104   | 0    | 1      | 2097152    | 106496     |

23 tnregistrNri selectate.

SQL> \_



# VEDERI - cont

- Coloane in DBA\_TABLESPACES:
  - Tablespace\_name – numele acelui tablespace.
  - Contents – daca el contine date permanente, de undo sau temporare.
  - Status – Daca este Online, Offline sau Read Only.
  - De asemenea sunt coloane pentru toti parametrii specificati la creare (pentru a putea vedea valoarea lor): BLOCK\_SIZE , INITIAL\_EXTENT , NEXT\_EXTENT , MIN\_EXTENTS , MAX\_EXTENTS, PCT\_INCREASE, etc.

# VEDERI - cont

- ❑ Vederea DBA\_DATA\_FILES contine date despre fisierele de date aferente fiecarui tablespace.
- ❑ Se pot folosi si vederile V\$DATAFILE si V\$TABLESPACE (legate prin coloana comuna TS# - id-ul de tablespace) pentru a obtine informatii despre fisierele de date ale fiecarui tablespace.

```
SQL> R
1  SELECT FILE_NAME, TABLESPACE_NAME, STATUS, AUTOEXTENSIBLE,
2  INCREMENT_BY, ONLINE_STATUS
3* FROM DBA_DATA_FILES
```

| FILE_NAME                                                     | TABLESPACE_NAME       | STATUS        | AUT | INCREMENT_BY | ONLINE_ |
|---------------------------------------------------------------|-----------------------|---------------|-----|--------------|---------|
| C:\ORACLE11G_DB\ORADATA\BD\USERS01.DBF                        | USERS                 | AVAILA<br>BLE | YES | 160          | ONLINE  |
| C:\ORACLE11G_DB\ORADATA\BD\UNDOTBS1.DBF                       | UNDOTBS1              | AVAILA<br>BLE | YES | 640          | ONLINE  |
| C:\ORACLE11G_DB\ORADATA\BD\SYSAUX01.DBF                       | SYSAUX                | AVAILA<br>BLE | YES | 1280         | ONLINE  |
| C:\ORACLE11G_DB\ORADATA\BD\SYSTEM01.DBF                       | SYSTEM                | AVAILA<br>BLE | YES | 1280         | SYSTEM  |
| C:\ORACLE11G_DB\ORADATA\BD\EXAMPLE01.DBF                      | EXAMPLE               | AVAILA<br>BLE | YES | 80           | ONLINE  |
| C:\ORACLE11G_DB\ORADATA\BD\BD_DATA.DBF                        | BD_DATA               | AVAILA<br>BLE | NO  | 0            | ONLINE  |
| C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_IDX | CONSTANT_GROW_INDEXES | AVAILA<br>BLE | NO  | 0            | ONLINE  |
| C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_TAB | CONSTANT_GROW_TABLES  | AVAILA<br>BLE | NO  | 0            | ONLINE  |
| C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\                      | DEPENDENCY_INDEXES    | AVAILA        | NO  | 0            | ONLINE  |

```
1* SELECT * FROM U$DATAFILE
```

111

SQL&gt; R

1\* SELECT \* FROM U\$TABLESPACE

| TS# | NAME                  | INC | BIG | FLA | ENC |
|-----|-----------------------|-----|-----|-----|-----|
| 0   | SYSTEM                | YES | NO  | YES |     |
| 1   | SYSAUX                | YES | NO  | YES |     |
| 2   | UNDOTBS1              | YES | NO  | YES |     |
| 4   | USERS                 | YES | NO  | YES |     |
| 3   | TEMP                  | NO  | NO  | YES |     |
| 6   | EXAMPLE               | YES | NO  | YES |     |
| 7   | BD_DATA               | YES | NO  | YES |     |
| 8   | BD_TEMP               | NO  | NO  | YES |     |
| 9   | CONSTANT_GROW_INDEXES | YES | NO  | YES |     |
| 10  | CONSTANT_GROW_TABLES  | YES | NO  | YES |     |
| 11  | DEPENDENCY_INDEXES    | YES | NO  | YES |     |
| 12  | DEPENDENCY_TABLES     | YES | NO  | YES |     |
| 13  | DIAGRAM_INDEXES       | YES | NO  | YES |     |
| 14  | DIAGRAM_TABLES        | YES | NO  | YES |     |
| 15  | LOB_DATA              | YES | NO  | YES |     |
| 16  | RAPID_GROW_INDEXES    | YES | NO  | YES |     |
| 18  | SYSTEM_META_INDEXES   | YES | NO  | YES |     |
| 19  | SYSTEM_META_TABLES    | YES | NO  | YES |     |
| 20  | TEMPORARY_INDEXES     | YES | NO  | YES |     |
| 21  | TEMPORARY_TABLES      | YES | NO  | YES |     |
| 22  | VERSION_INDEXES       | YES | NO  | YES |     |
| 23  | VERSION_TABLES        | YES | NO  | YES |     |
| 24  | REPOS_TEMP            | NO  | NO  | YES |     |

23 tnregistrÑri selectate.

SQL&gt;

# Alte vederi

Sursa: <http://psoug.org/reference/tablespaces.html>

|                           |                                                                                                   |
|---------------------------|---------------------------------------------------------------------------------------------------|
| Tablespaces               | ts\$<br>dba_tablespaces<br>user_tablespaces                                                       |
| Tablespace Quotas         | tsq\$<br>dba_ts_quotas<br>user_ts_quotas                                                          |
| Data Files                | dba_data_files<br>v_\$backup_datafile<br>v_\$datafile<br>v_\$datafile_copy<br>v_\$datafile_header |
| Free Space                | dba_free_space                                                                                    |
| Segments                  | dba_segments<br>v_\$segment_statistics                                                            |
| Extents                   | dba_extents                                                                                       |
| Blocks                    | v_\$database_block_corruption                                                                     |
| Groups                    | dba_tablespace_groups                                                                             |
| SYSAUX Tablespace         | v_\$sysaux_occupants                                                                              |
| Temp Tablespace           | dba_temp_files                                                                                    |
| Undo Tablespace           | dba_rollback_segs<br>dba_undo_extents<br>v_\$rollstat<br>v_\$undostat                             |
| Transportable Tablespaces | transport_set_violations                                                                          |
| Dictionary Management     | fet\$<br>uet\$                                                                                    |

1\* SELECT \* FROM DBA\_FREE\_SPACE

| TABLESPACE_NAME | FILE_ID | BLOCK_ID | BYTES   | BLOCKS | RELATIVE_FNO |
|-----------------|---------|----------|---------|--------|--------------|
| SYSTEM          | 1       | 102952   | 720896  | 88     | 1            |
| SYSTEM          | 1       | 104576   | 9437184 | 1152   | 1            |
| SYSAUX          | 2       | 55680    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 55696    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 55712    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 55728    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 55760    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 59024    | 196608  | 24     | 2            |
| SYSAUX          | 2       | 59080    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 59128    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 59144    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 59464    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 59488    | 262144  | 32     | 2            |
| SYSAUX          | 2       | 60184    | 196608  | 24     | 2            |
| SYSAUX          | 2       | 60224    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 60560    | 131072  | 16     | 2            |
| SYSAUX          | 2       | 61008    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 61048    | 262144  | 32     | 2            |
| SYSAUX          | 2       | 61088    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 61128    | 327680  | 40     | 2            |
| SYSAUX          | 2       | 61176    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 61312    | 196608  | 24     | 2            |
| SYSAUX          | 2       | 61360    | 393216  | 48     | 2            |
| SYSAUX          | 2       | 61472    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 62720    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 63024    | 327680  | 40     | 2            |
| SYSAUX          | 2       | 63072    | 131072  | 16     | 2            |
| SYSAUX          | 2       | 63096    | 1179648 | 144    | 2            |
| SYSAUX          | 2       | 63256    | 65536   | 8      | 2            |
| SYSAUX          | 2       | 63288    | 262144  | 32     | 2            |
| SYSAUX          | 2       | 63544    | 65536   | 8      | 2            |

1\* SELECT \* FROM U\$DATAFILE\_HEADER

| FILE#                                    | STATUS     | ERRO               | FORMAT       | REC              | FUZ     | CREATION_CHANGE# | CREATION_T            | TABLESPACE_NAME | TS# | RFILE# |
|------------------------------------------|------------|--------------------|--------------|------------------|---------|------------------|-----------------------|-----------------|-----|--------|
| RESETLOGS_CHANGE#                        | RESETLOGS_ | CHECKPOINT_CHANGE# | CHECKPOINT   | CHECKPOINT_COUNT | BYTES   | BLOCKS           |                       |                 |     |        |
| NAME                                     |            |                    | SPACE_HEADER |                  |         | LAST_DEALLOC_CHA |                       |                 |     |        |
| UNDO_OPT_CURRENT_CHANGE#                 |            |                    |              |                  |         |                  |                       |                 |     |        |
| 1                                        | ONLINE     |                    | 10 NO        | YES              | 7       | 30-03-2010       | SYSTEM                |                 | 0   |        |
| 947455                                   | 23-03-2015 |                    | 3102137      | 09-04-2019       | 637     | 866123776        | 105728                |                 |     |        |
| C:\ORACLE11G_DB\ORADATA\BD\SYSTEM01.DBF  |            |                    | 4194306      |                  |         | 942037           |                       |                 |     |        |
| 2                                        | ONLINE     |                    | 10 NO        | YES              | 2160    | 30-03-2010       | SYSAUX                |                 | 1   |        |
| 947455                                   | 23-03-2015 |                    | 3102137      | 09-04-2019       | 637     | 587202560        | 71680                 |                 |     |        |
| C:\ORACLE11G_DB\ORADATA\BD\SYSAUX01.DBF  |            |                    | 8388610      |                  |         | 947192           |                       |                 |     |        |
| 3                                        | ONLINE     |                    | 10 NO        | YES              | 944668  | 30-03-2010       | UNDOTBS1              |                 | 2   |        |
| 947455                                   | 23-03-2015 |                    | 3102137      | 09-04-2019       | 225     | 256901120        | 31360                 |                 |     |        |
| C:\ORACLE11G_DB\ORADATA\BD\UNDOTBS01.DBF |            |                    | 12582914     |                  |         |                  |                       |                 |     |        |
| 4                                        | ONLINE     |                    | 10 NO        | YES              | 17981   | 30-03-2010       | USERS                 |                 | 4   |        |
| 947455                                   | 23-03-2015 |                    | 3102137      | 09-04-2019       | 634     | 14417920         | 1760                  |                 |     |        |
| C:\ORACLE11G_DB\ORADATA\BD\USERS01.DBF   |            |                    | 16777218     |                  |         |                  |                       |                 |     |        |
| 5                                        | ONLINE     |                    | 10 NO        | YES              | 976602  | 23-03-2015       | EXAMPLE               |                 | 6   |        |
| 947455                                   | 23-03-2015 |                    | 3102137      | 09-04-2019       | 221     | 104857600        | 12800                 |                 |     |        |
| C:\ORACLE11G_DB\ORADATA\BD\EXAMPLE01.DBF |            |                    | 209771522    |                  |         | 967727           |                       |                 |     |        |
| 6                                        | ONLINE     |                    | 10 NO        | YES              | 1068220 | 24-03-2015       | BD_DATA               |                 | 7   |        |
| 947455                                   | 23-03-2015 |                    | 3102137      | 09-04-2019       | 208     | 52428800         | 6400                  |                 |     |        |
| C:\ORACLE11G_DB\ORADATA\BD\BD_DATA.DBF   |            |                    | 25165826     |                  |         |                  |                       |                 |     |        |
| 7                                        | ONLINE     |                    | 10 NO        | YES              | 1078075 | 24-03-2015       | CONSTANT_GROW_INDEXES |                 | 9   |        |
| 947455                                   | 23-03-2015 |                    | 3102137      | 09-04-2019       | 208     | 20977152         | 256                   |                 |     |        |
| C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\ |            |                    | 29360130     |                  |         |                  |                       |                 |     |        |



```
SQL>
SQL>
SQL>
SQL> R
1* SELECT * FROM DBA_TEMP_FILES
```

| FILE_NAME                                                   |                 |            |             |        |        |              |       |            |
|-------------------------------------------------------------|-----------------|------------|-------------|--------|--------|--------------|-------|------------|
| FILE_ID                                                     | TABLESPACE_NAME |            | BYTES       | BLOCKS | STATUS | RELATIVE_FNO | AUT   | MAXBYTES   |
| MAXBLOCKS                                                   | INCREMENT_BY    | USER_BYTES | USER_BLOCKS |        |        |              |       |            |
| C:\ORACLE11G_DB\ORADATA\BD\TEMP01.DBF                       |                 |            |             |        |        |              |       |            |
| 1                                                           | TEMP            |            | 20971520    | 2560   | ONLINE |              | 1 YES | 3,4360E+10 |
| 4194302                                                     | 80              | 19922944   | 2432        |        |        |              |       |            |
| C:\ORACLE11G_DB\ORADATA\BD\BD_TEMP.DBF                      |                 |            |             |        |        |              |       |            |
| 2                                                           | BD_TEMP         |            | 10485760    | 1280   | ONLINE |              | 1 NO  | 0          |
| 0                                                           | 0               | 9437184    | 1152        |        |        |              |       |            |
| C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_TEMP |                 |            |             |        |        |              |       |            |
| 3                                                           | REPOS_TEMP      |            | 15745024    | 1922   | ONLINE |              | 1 NO  | 0          |
| 0                                                           | 0               | 14680064   | 1792        |        |        |              |       |            |

SQL> R  
1\* SELECT \* FROM DBA\_ROLLBACK\_SEGS

| SEGMENT_NAME          |                |             | OWNER        | TABLESPACE_NAME |              | SEGMENT_ID | FILE_ID |
|-----------------------|----------------|-------------|--------------|-----------------|--------------|------------|---------|
| BLOCK_ID              | INITIAL_EXTENT | NEXT_EXTENT | MIN_EXTENTS  | MAX_EXTENTS     | PCT_INCREASE | STATUS     |         |
| INSTANCE_NUM          |                |             | RELATIVE_FNO |                 |              |            |         |
| SYSTEM                |                |             | SYS          | SYSTEM          |              | 0          | 1       |
| 128                   | 114688         | 57344       | 1            | 1               | 32765        | ONLINE     |         |
| _SYSSMU10_378818850\$ |                |             | PUBLIC       | UNDOTBS1        |              | 10         | 3       |
| 272                   | 131072         | 65536       | 2            | 3               | 32765        | ONLINE     |         |
| _SYSSMU9_3186340089\$ |                |             | PUBLIC       | UNDOTBS1        |              | 9          | 3       |
| 256                   | 131072         | 65536       | 2            | 3               | 32765        | ONLINE     |         |
| _SYSSMU8_1682283174\$ |                |             | PUBLIC       | UNDOTBS1        |              | 8          | 3       |
| 240                   | 131072         | 65536       | 2            | 3               | 32765        | ONLINE     |         |
| _SYSSMU7_1101470402\$ |                |             | PUBLIC       | UNDOTBS1        |              | 7          | 3       |
| 224                   | 131072         | 65536       | 2            | 3               | 32765        | ONLINE     |         |

# Lecturi obligatorii

## 1. Locally vs. Dictionary Managed Tablespaces

<http://www.oraFAQ.com/node/3>

## 2. Oracle Database Administrator's Guide – Cap 8: Managing Tablespaces

[http://download.oracle.com/docs/cd/B14117\\_01/server.101/b10739/tspaces.htm](http://download.oracle.com/docs/cd/B14117_01/server.101/b10739/tspaces.htm)

## 3. Oracle Concepts - Tablespaces

<http://www.adp-gmbh.ch/ora/concepts/tablespaces.html>

# Sfârșitul capitolului 4

# Capitolul 5

## Gestiunea tabelelor

### - Partea 1. Crearea tabelelor -

# Tipuri de organizare

- Exista patru tipuri de organizare pentru tabelele unei baze de date:
  1. Tabele uzuale – heap-organized tables – este tipul de baza, uzual. O astfel de tabela reprezinta o multime neorganizata (heap) de linii. Acest tip de tabele este subiectul principal al capitolului de fata.
  2. Tabele partitionate – partitioned tables – in care liniile sunt impartite in mai multe grupuri, numite partitii, fiecare astfel de partitie (sau subpartitie) putand fi gestionata separat.

# Tipuri de organizare - cont

3. Tabele de tip cluster - 'clustered tables'. O astfel de tabela este parte a unui cluster. Un cluster contine mai multe tabele care au in comun blocuri de date - ele au in comun anumite coloane si, de asemenea, sunt folosite frecvent impreuna.
4. Tabele de tip index - 'index organized'. Spre deosebire de primele (heap organized), inregistrările (liniile) unei astfel de tabele sunt organizate sub forma unui arbore B, sortate dupa cheia primara.

# Tipuri de organizare - cont

5. Tabele externe - 'external tables'. O astfel de tabela nu se află în baza de date, ci în afara bazei de date, în fișiere externe, cum ar fi fișierele sistemului de operare sau fișierele de tip Hadoop Distributed File System (HDFS).
6. Tabele hibride partitionate - 'hybrid partitioned table'. O tabela hibrida partiționata este o tabela partiționata în care unele partiții se află în baza de date, iar altele în afara bazei de date, în fișiere externe, cum ar fi fișierele sistemului de operare sau fișierele Hadoop Distributed File System (HDFS). Au aparut in versiunea 19c.



# Tipuri de organizare - cont

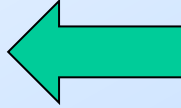
- ❑ In cazul tabelelor partitionate, impartirea liniilor in partitii se face dupa valoarea uneia sau mai multor coloane.
- ❑ O linie a tabeli poate sa apartina unei singure partitii
- ❑ Fiecare partitie are un nume si este stocata intr-un segment. Aceste segmente pot fi in tablespace-uri diferite
- ❑ Partitionarea se practica in cazul tabelelor de mari dimensiuni care sunt accesate concurent.
- ❑ Se pot partitiona si tabelele de tip 'index-organized' cu conditia ca attributele (coloanele) dupa care se face partitionarea sa fie o submultime a cheii primare.

# CREATE TABLE

- Pe langa elementele cunoscute din cursurile anterioare (BD an 3), cererea SQL **CREATE TABLE** poate avea si alte clauze, suplimentare, specificand parametrii de stocare pentru datele tabelului respective.

# Syntaxa CREATE TABLE (9i)

```
CREATE TABLE [schema.]table
  ( coloane_si_constrangeri ) - sintaxa clasica
  [ [PCTFREE integer] [PCTUSED integer]
    [INITRANS integer] [MAXTRANS integer]
    [TABLESPACE tablespace]
    [STORAGE storage_clause]
    [ PARALLEL [integer ] | NOPARALLEL ]
    [ CACHE | NOCACHE ]
    [ LOGGING | NOLOGGING ]
  |
  [CLUSTER cluster (column [, column]...)]
  ]
  [ ENABLE enable_clause | DISABLE disable_clause ] ...
  [AS subquery]
```



# Clauze CREATE – PCTFREE(1)

- ❑ PCTFREE specifica procentul de spatiu din fiecare bloc rezervat cresterii in lungime a inregistrarilor (liniilor) determinata de operatii de tip update.
- ❑ Valoarea trebuie sa fie intre 1 si 99.
- ❑ In cazul specificarii valorii 0 intregul bloc poate fi umplut prin inserare de noi linii.
- ❑ Valoarea implicita a acestui parametru este 10 (deci 10% spatiu disponibil pentru update, 90% spatiu disponibil pentru insert).

# Clauze CREATE – PCTFREE(2)

- ❑ In cazul in care inregistrarile dintr-un bloc cresc in lungime si se depaseste spatiul alocat lor (inclusiv cel initial retinut pentru crestere prin PCTFREE)  
Oracle ia o linie din acel bloc si o muta in alt bloc, lasand in locul ei doar un pointer.
- ❑ Acest proces este numit si 'migrarea liniilor'
- ❑ In acest caz performantele scad, deoarece pentru citirea acelei linii sunt citite doua blocuri.

# Clauze CREATE – PCTFREE(3)

- ❑ In cazul in care o inregistrare este prea lunga pentru a incapa intr-un bloc aceasta inregistrare este sparta in mai multe bucati care sunt stocate in mai multe blocuri, impreuna cu pointerii necesari recuperarii intregii linii. Aceasta situatie se numeste 'row chaining'.
- ❑ Si in acest caz performantele scad, deoarece pentru citirea acelei linii sunt citite mai multe blocuri.
- ❑ Parametrul PCTFREE poate fi prezent in comenzile create/alter si pentru alte obiecte (ex. indecsi).

# Clauze CREATE - PCTUSED

- ❑ PCTUSED specifica procentajul minim de spatiu utilizat din fiecare bloc. Daca spatiul utilizat scade sub acea valoare blocul devine candidat pentru inserarea de noi inregistrari (linii).
- ❑ PCTUSED are valori intre 1 si 99.
- ❑ Valoarea de default este 40
- ❑ Combinatia PCTFREE - PCTUSED duce la directionarea noilor inregistrari fie in blocuri existente fie in blocuri noi (goale la acel moment)



# Clauze CREATE - PCTUSED

- ❑ Acest parametru poate fi prezent si in comenzile de creare pentru alte obiecte (ex. Indecsi)
- ❑ Suma dintre PCTFREE si PCTUSED trebuie sa fie mai mica sau egala cu 100.
- ❑ Cu cat diferenta intre 100 si aceasta suma este mai mica, cu atat este mai eficienta folosirea spatiului pe disc, insa pot sa scada performantele.



# Clauze CREATE - INITRANS

- ❑ INITRANS specifica numarul initial de 'transaction entries' alocate in fiecare bloc. Fiecare tranzactie care actualizeaza un bloc are nevoie de o astfel de intrare la nivelul blocului.
- ❑ Valoarea poate fi de la 1 la 255
- ❑ Valoarea implicita este 1 in cazul tabelelor (2 la indecsi).
- ❑ In general Oracle recomanda sa se pastreze valoarea implicita
- ❑ Acest parametru asigura un numar minim de tranzactii per bloc fara overheadul alocarii dinamice a unei intrari ("transaction entry").

# Clauze CREATE - MAXTRANS

- ❑ MAXTRANS specifica numarul maxim de tranzactii concurente care pot actualiza un bloc al tabelii – deci numarul maxim de 'transaction entries' care pot fi alocate unui bloc.
- ❑ Acestea se alocă dinamic de Oracle după depășirea INITRANS.
- ❑ Valoarea de default este 255. Citat din documentatia pentru 19c:
  - "This parameter has been deprecated. Oracle now automatically allows up to 255 concurrent update transactions for any data block, depending on the available space in the block."
- ❑ MAXTRANS este parametru și în alte operații de creare obiecte ale bazei de date.

# TABLESPACE, STORAGE

- ❑ TABLESPACE specifica unde se va crea tabela respectiva.
- ❑ Daca optiunea lipseste, tabela se creaza in tablespace-ul implicit (default) al userului care detine schema in care se face crearea.
- ❑ STORAGE specifica modul in care extensiile vor fi alocate tabeli.
- ❑ Sintaxa clauzei STORAGE a fost prezentata in capitolul anterior (cel despre tablespace-uri).
- ❑ Aceasta clauza are implicatii in performantele obtinute in cazul tabelor de mari dimensiuni.

# Reamintire:

Clauza Storage are optiuni ca:

- ☐ **INITIAL *int* K | M**
- ☐ **NEXT *int* K | M**
- ☐ **MINEXTENTS *int***
- ☐ **MAXEXTENTS *int***
- ☐ **MAXEXTENTS UNLIMITED**
- ☐ **PCTINCREASE *int***
- ☐ **FREELISTS *int***
- ☐ **FREELIST GROUPS *int***

# Reamintire:

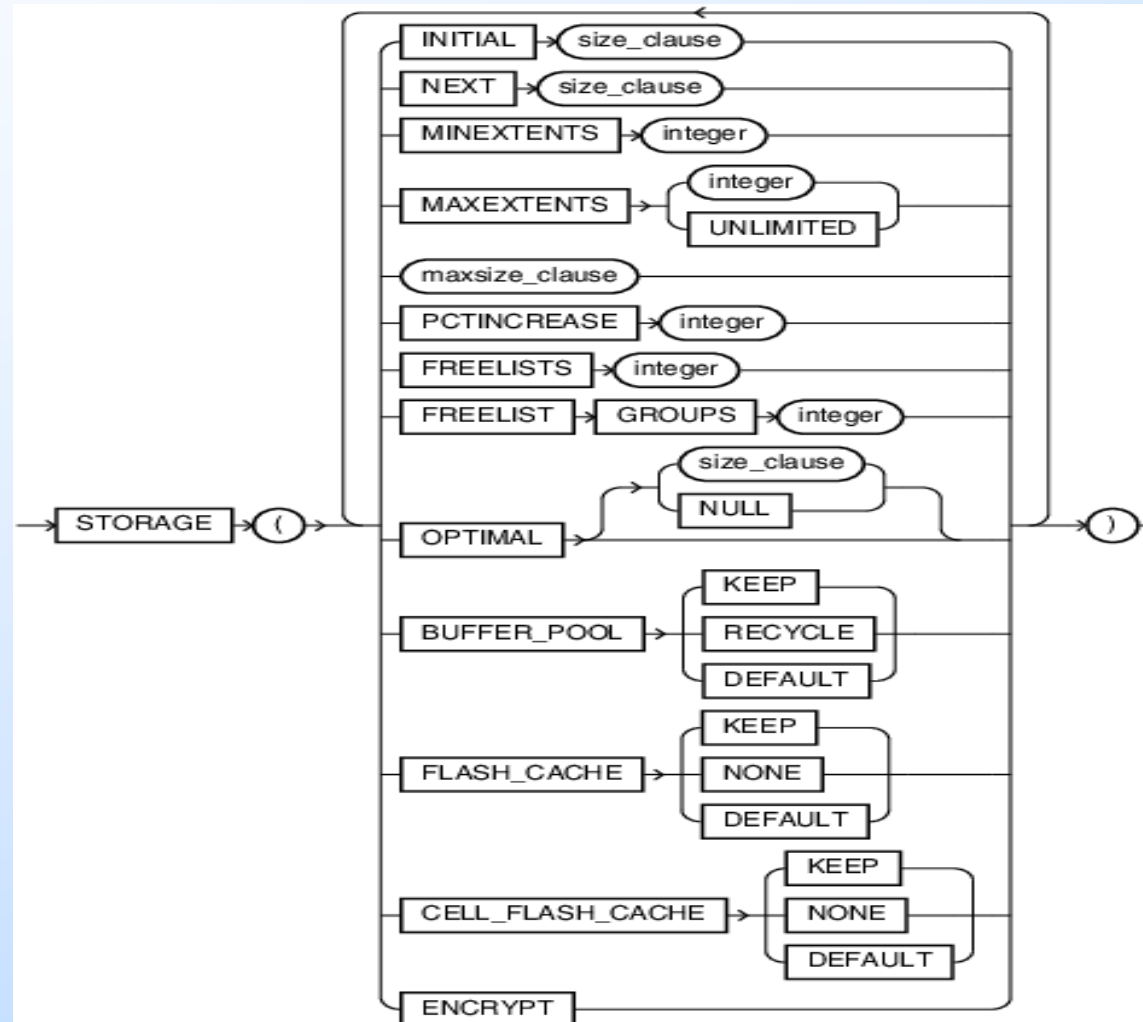
Unde:

- ❑ **INITIAL *int* K | M** – definește dimensiunea primei extensii (minim 2 blocuri). Valoarea implicită este 5 blocuri ale BD.
- ❑ **NEXT *int* K | M** – da dimensiunea celei de-a doua extensii. Valoarea minimă este de 1 bloc, valoarea implicită este de asemenea 5 blocuri.
- ❑ **MINEXTENTS *int*** - este numărul de extensii care sunt alocate când segmentul este creat. Valoarea minimă – și implicită – este 1.

# Reamintire:

- ❑ **MAXEXTENTS *int*** – determina numarul maxim de extensii pe care le poate avea un segment. Valoarea minima este 1 iar valoarea maxima depinde de dimensiunea blocului.
- ❑ **MAXEXTENTS UNLIMITED** – este echivalenta cu 2G extensii
- ❑ **PCTINCREASE *int*** – este procentul cu care creste dimensiunea extensiilor. Valoarea minima este 0, cea implicita 50.

# Sintaxa completa



# Exemplu:

```
create table tabela_mea (  
  2     nume      varchar2(30),  
  3     descriere  varchar2(4000) )  
  4     tablespace users  
  5     storage (  
  6       initial   1M  
  7       next      512K  
  8       pctincrease 0  
  9       minextents 2  
 10     maxextents unlimited )  
 11  /
```



Enter user-name: sys as sysdba

Enter password:

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production

With the Partitioning, Oracle Label Security, OLAP and Data Mining Scoring Engine options

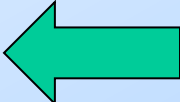
```
SQL> create table tabela_mea(  
  2  nume varchar2(20))  
  3  tablespace users  
  4  storage(  
  5  initial 1M  
  6  next 64K  
  7  pctincrease 0  
  8  minextents 2  
  9  maxextents unlimited) ;
```

Table created.

SQL>

# Syntaxa CREATE TABLE (9i)

```
CREATE TABLE [schema.]table
( coloane_si_constrangeri ) - sintaxa clasica
[ [PCTFREE integer] [PCTUSED integer]
  [INITRANS integer] [MAXTRANS integer]
  [TABLESPACE tablespace]
  [STORAGE storage_clause]
  [ PARALLEL [integer ] | NOPARALLEL ]
  [ CACHE | NOCACHE ]
  [ LOGGING | NOLOGGING ]
  |
  [CLUSTER cluster (column [, column]...)]
]
[ ENABLE enable_clause | DISABLE disable_clause ] ...
[AS subquery]
```



# Clauze CREATE: PARALLEL

- ❑ PARALLEL *int* specifica paralelizarea cererii de creare (de exemplu in cazul CREATE ... AS SELECT) sau gradul de paralelism pentru cereri DML - numarul de procese server care pot scana (parcure) in paralel tabela in cereri SELECT, INSERT, UPDATE, DELETE, MERGE.
- ❑ Se poate specifica: nimic (Oracle alege nr\_CPU x nr\_fire\_executie\_per\_CPU) sau un numar intreg.
- ❑ NOPARALLEL specifica faptul ca pe aceasta tabela cererile nu pot fi executate (in mod obisnuit) prin paralelizare – se pot folosi ‘hint’-uri pentru a forta executia paralela (ca in exemplul urmator).

# Clauze CREATE: PARALLEL - cont

- Cererea urmatoare specifica parcurgerea intregii tabele cu un grad de paralelism egal cu 5. Observam ca daca se defineste un alias de tabela hintul trebuie sa foloseasca acest alias:

```
SELECT /*+ FULL(s)
        PARALLEL(s, 5) */
ename
FROM emp s;
```

# Clauze CREATE: PARALLEL - cont

- ❑ Cererea urmatoare specifica parcurgerea tabelii fara paralelizare. De asemenea trebuie sa se foloseasca aliasul definit:

```
SELECT /*+ NOPARALLEL(s) */  
       ename  
FROM emp s;
```

# Clauze CREATE - CACHE

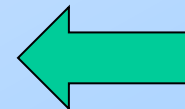
- ❑ CACHE se foloseste mai ales pentru tabele de mici dimensiuni si specifica faptul ca acea tabela va fi pastrata in buferele de memorie (deci nu va fi dealocata) prin plasarea blocurilor sale in zona celor mai recent utilizate chiar si atunci cand se executa o parcurgere completa a tablei.
- ❑ NOCACHE (valoare implicita) specifica faptul ca blocurile tablei din buffer cache se supun algoritmului LRU standard atunci cand se executa o parcurgere completa a tablei (full table scan), si deci se pun in zona celor mai putin recent utilizate.

# Clauze CREATE - LOGGING

- ❑ LOGGING arata ca atat operatia de creare a tabelii cat si operatiile care vor fi facuta apoi asupra acesteia vor fi inregistrate in fisierele Redo Log.
- ❑ NOLOGGING specifica faptul ca operatia de creare a tabelii precum si unele operatii de incarcare cu date (nu insa si operatiile obisnuite de insert) nu vor fi inregistrate in fisierele Redo Log.
- ❑ In lipsa acestor optiuni se folosesc parametrii de la crearea tablespace-ului in care este gazduita tabela (si acolo aveam aceste doua optiuni)

# Syntaxa CREATE TABLE (9i)

```
CREATE TABLE [schema.]table
( coloane_si_constrangeri ) - sintaxa clasica
[ [PCTFREE integer] [PCTUSED integer]
  [INITRANS integer] [MAXTRANS integer]
  [TABLESPACE tablespace]
  [STORAGE storage_clause]
  [ PARALLEL [integer ] |
    NOPARALLEL ]
  [ CACHE | NOCACHE ]
  [ LOGGING | NOLOGGING ]
  |
  [CLUSTER cluster (column [, column]...)]
]
[ ENABLE enable_clause | DISABLE disable_clause ] ...
[AS subquery]
```





# Clauze CREATE - CLUSTER

- ❑ CLUSTER arata ca tabela este parte a unui cluster.
- ❑ Coloanele din clauza sunt coloane ale tablei care corespund cu coloanele clusterului.
- ❑ In general coloanele respective ale tablei sunt parte a cheii primare (sau intreaga cheie primara).

# Clauze CREATE - CLUSTER

- ❑ Trebuie specificata cate o coloana a tablei pentru fiecare coloana a clusterului.
- ❑ Corespondenta este pozitionala (nu prin nume)
- ❑ Deoarece tablele de tip cluster folosesc o alta alocare a spatiului NU se pot folosi in paralel clauzele PCTFREE, PCTUSED, INITRANS, MAXTRANS, TABLESPACE in conjunctie cu clauza CLUSTER

# Exemplu

- ❑ 1. Creare cluster:

```
CREATE CLUSTER pers
```

```
(dept NUMBER(2))
```

```
SIZE 512
```

```
STORAGE (initial 100K next 50K);
```

- ❑ 2. Creare index pentru cheia clusterului:

```
CREATE INDEX idx_pers ON CLUSTER pers;
```

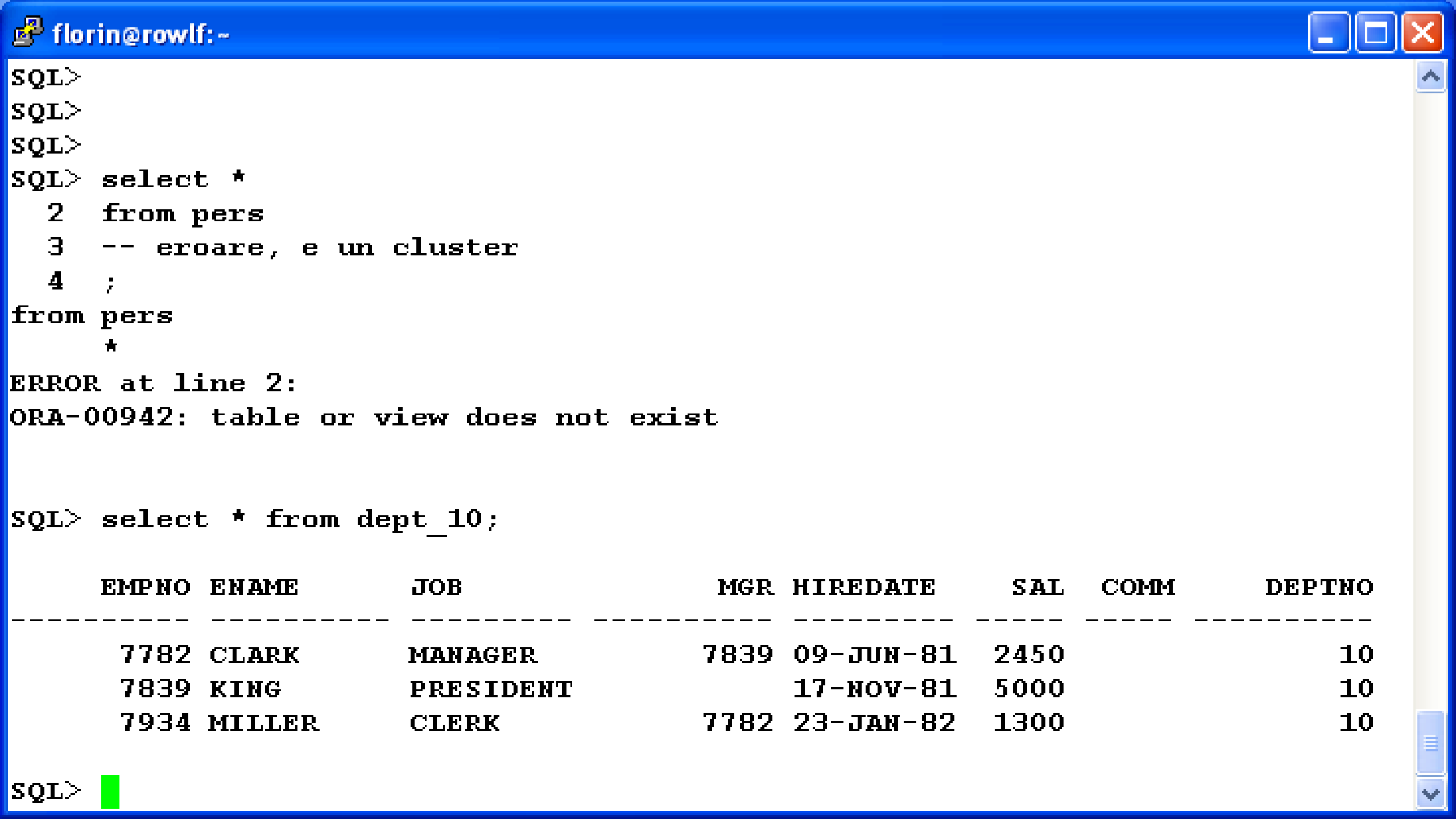
# Exemplu

□ 3. Adaugare tabele la cluster:

```
CREATE TABLE dept_10 CLUSTER pers (deptno)  
AS SELECT * FROM scott.emp  
WHERE deptno = 10;
```

```
CREATE TABLE dept_20 CLUSTER pers (deptno)  
AS SELECT * FROM scott.emp  
WHERE deptno = 20;
```

```
florin@rowlf:~  
SQL>  
SQL>  
SQL>  
SQL>  
SQL> CREATE CLUSTER pers (dept NUMBER(2)) SIZE 512 STORAGE (initial 100K next 50K)  
;  
  
Cluster created.  
  
SQL> CREATE INDEX idx_pers ON CLUSTER pers;  
  
Index created.  
  
SQL> CREATE TABLE dept_10 CLUSTER pers (deptno) AS SELECT * FROM scott.emp WHERE d  
eptno = 10;  
  
Table created.  
  
SQL> CREATE TABLE dept_20 CLUSTER pers (deptno) AS SELECT * FROM scott.emp WHERE d  
eptno = 20;  
  
Table created.  
  
SQL>
```



SQL>

SQL>

SQL>

```
SQL> select *
      2  from pers
      3  -- eroare, e un cluster
      4  ;
from pers
      *
```

ERROR at line 2:  
ORA-00942: table or view does not exist

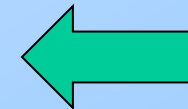
SQL> select \* from dept\_10;

| EMPNO | ENAME  | JOB       | MGR  | HIREDATE  | SAL  | COMM | DEPTNO |
|-------|--------|-----------|------|-----------|------|------|--------|
| 7782  | CLARK  | MANAGER   | 7839 | 09-JUN-81 | 2450 |      | 10     |
| 7839  | KING   | PRESIDENT |      | 17-NOV-81 | 5000 |      | 10     |
| 7934  | MILLER | CLERK     | 7782 | 23-JAN-82 | 1300 |      | 10     |

SQL>

# Syntaxa CREATE TABLE (9i)

```
CREATE TABLE [schema.]table
( coloane_si_constrangeri ) - sintaxa clasica
[ [PCTFREE integer] [PCTUSED integer]
  [INITRANS integer] [MAXTRANS integer]
  [TABLESPACE tablespace]
  [STORAGE storage_clause]
  [ PARALLEL [integer ] |
    NOPARALLEL ]
  [ CACHE | NOCACHE ]
  [ LOGGING | NOLOGGING ]
  |
  [CLUSTER cluster (column [, column]...)]
]
[ ENABLE enable_clause | DISABLE disable_clause ] ...
[AS subquery]
```



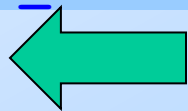
# ENABLE / DISABLE

- ❑ ENABLE si DISABLE activeaza / inhiba o constrangere de integritate.
- ❑ Aceste constrangeri sunt dintre cele create in aceeaasi comanda.
- ❑ In mod implicit, la creare, constrangerile de integritate sunt active (ENABLE)



# Syntaxa CREATE TABLE (9i)

```
CREATE TABLE [schema.]table
( coloane_si_constrangeri ) - sintaxa clasica
[ [PCTFREE integer] [PCTUSED integer]
  [INITRANS integer] [MAXTRANS integer]
  [TABLESPACE tablespace]
  [STORAGE storage_clause]
  [ PARALLEL [integer ] |
    NOPARALLEL ]
  [ CACHE | NOCACHE ]
  [ LOGGING | NOLOGGING ]
  |
  [CLUSTER cluster (column [, column]...)]
]
[ ENABLE enable_clause | DISABLE disable_clause ] ...
[AS subquery]
```



# Clauze CREATE - AS

- ❑ AS specifica faptul ca noua tabela va fi populata cu liniile rezultate din cererea select prezenta in clauza AS.
- ❑ Crearea unei tabele ca rezultat al unei cereri SELECT a fost studiata in semestrele trecute.

# CREATE .. TEMPORARY

- ❑ Se pot crea tabele temporare cu cereri de tipul: `CREATE GLOBAL TEMPORARY TABLE`
- ❑ Definitia acestor tabele este vizibila tuturor sesiunilor active
- ❑ Datele din aceste tabele sunt vizibile doar sesiunii care le insereaza (doar una la un moment dat)

# CREATE .. TEMPORARY

- Liniile din tabela se sterg la sfarsitul fiecărei tranzactii sau la sfarsitul sesiunii, dupa cum se specifica in clauza ON COMMIT:
- ON COMMIT DELETE ROWS – liniile se sterg la sfarsitul fiecărei tranzactii
- ON COMMIT PRESERVE ROWS – se sterg la sfarsit de sesiune.

# Exemplu

```
CREATE GLOBAL TEMPORARY TABLE  
test_temp (nume VARCHAR2(20)) ON  
COMMIT DELETE ROWS;
```

- ❑ Se va crea in acest caz o tabela temporara in care liniile se sterg la sfarsitul fiecărei tranzactii.

```
florin@rowlf:~
SQL> CREATE GLOBAL TEMPORARY TABLE test_temp (nume VARCHAR2(20)) ON COMMIT DE
LETE ROWS;

Table created.

SQL> INSERT INTO test_temp VALUES('ION');

1 row created.

SQL> SELECT * FROM test_temp;

NUME
-----
ION

SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM test_temp;

no rows selected

SQL>
```

# Copierea unei tabele

- ❑ Folosind `CREATE TABLE .. AS SELECT` se poate copia o tabela schimbându-i, la momentul copierii, anumiti parametrii
- ❑ In acest caz se pot specifica noi nume ale coloanelor, noi parametrii de stocare fizica, etc.
- ❑ Atentie: nu sunt copiate in acest caz si constrangerile de integritate (cu exceptia coloanelor definite cu `NOT NULL`, care vor fi la fel si in noua tabela.

# Lecturi obligatorii

1. Oracle Database Administrator's Guide (v19c) – Cap 20: Managing Tables.

<https://docs.oracle.com/en/database/oracle/oracle-database/19/admin/>

2. Sintaxa cereri Oracle de la adresa:

<http://www4.utc.fr/~nf17/DOCS/complement/sqlplus-ref/>

3. On shrinking table sizes:

<http://www.adp-gmbh.ch/blog/2005/july/20.html>



# Sfârșit creare tabele

# Capitolul 5

## Gestiunea tabelelor

### - Partea 2. Modificarea schemei -

# Comanda ALTER TABLE

```
ALTER TABLE [schema.]table
[options ADD, MODIFY, etc - prezentate anul trecut la BD1]
[PCTFREE integer] [PCTUSED integer]
[INITRANS integer] [MAXTRANS integer]
[STORAGE storage_clause]
[DROP drop_clause] ...
[ALLOCATE EXTENT [( [SIZE integer [K|M] ]
                    [DATAFILE 'filename']
                    [INSTANCE integer] )]
[ PARALLEL { integer } |
  NOPARALLEL ]
[ CACHE | NOCACHE ]
[{ ENABLE | DISABLE } { enable_clause | TABLE LOCK }]
[{ ENABLE | DISABLE } ALL TRIGGERS ]]
```

# ALTER TABLE - ALLOCATE

- ❑ `ALLOCATE EXTENT` alocă explicit o nouă extensie pentru acea tabelă.
- ❑ `SIZE` specifică dimensiunea extensiei în octeți (bytes). Se poate folosi K ori M pentru a specifica KB sau MB. În cazul absenței acestei clauze Oracle determină dimensiunea pe baza valorilor de `STORAGE` ale tabelului.
- ❑ `DATAFILE` specifică numele fișierului (aferent tablespace-ului în care se găsește tabelul) în care se va alocă noua extensie. În lipsă, alegerea este făcută de Oracle.

# ALTER TABLE - ALLOCATE

- ❑ INSTANCE face acea extensie disponibilă pentru instanța specificată. Acest parametru este util în conjuncție cu Oracle Real Application Clusters.
- ❑ Alocarea explicită a unei extensii nu afectează dimensiunea următoarei extensii care va fi alocată pe baza parametrilor NEXT și PCTINCREASE (prezentați în capitolul anterior)

# ALTER TABLE - cont

PARALLEL

NOPARALLEL

CACHE

NOCACHE

ENABLE

DISABLE

- ❑ Sunt folosite pentru a schimba setările curente.
- ❑ Aceste clauze au fost prezentate în cursul trecut.

# ALTER TABLE - LOCK

- ❑ **ENABLE TABLE LOCK** - Activează posibilitatea obținerii de blocări asupra tablei de comenzile DDL.
- ❑ Aceste comenzi nu se pot executa dacă blocarea nu este permisă.
- ❑ **DISABLE TABLE LOCK** duce implicit la interzicerea operațiilor DDL asupra tablei.

# ALTER TABLE - TRIGGERS

ENABLE ALL TRIGGERS

DISABLE ALL TRIGGERS

- ❑ Permit activarea / dezactivarea tuturor declanșatorilor asociați unei tabele
- ❑ Pentru activarea / dezactivarea unui singur declanșator se poate folosi comanda ALTER TRIGGER.
- ❑ Clauza DROP – specifica ștergerea unei constrângeri de integritate.



# Comanda ALTER TABLE - cont

- ❑ PCTFREE,
- ❑ PCTUSED,
- ❑ INITRANS,
- ❑ MAXTRANS,
- ❑ STORAGE

Schimbă valorile și opțiunile care există în acel moment. Explicația acestor parametri s-a făcut la descrierea comenzii CREATE TABLE

# Parametrii de stocare

- În cazul lui ALTER TABLE, semnificația noilor valori ale acestor parametri este următoarea:
- NEXT – În momentul alocării unei noi extensii Oracle va folosi noua valoare a lui NEXT după care creșterea se face pornind de la această valoare și cea a lui PCTINCREASE (vezi formula din capitolul precedent pentru dimensiunea extensiei **n**)

# Parametrii de stocare - cont

- PCTINCREASE – schimbarea acestuia va afecta doar dimensiunea noilor extensii care se alocă.
- MINEXTENTS – poate fi schimbată cu orice valoare care e mai mica sau egala decât numărul de extensii din acel moment
- MAXEXTENTS – poate fi schimbată cu orice valoare mai mare sau egală cu numărul de extensii din acel moment

# Parametri utilizare bloc

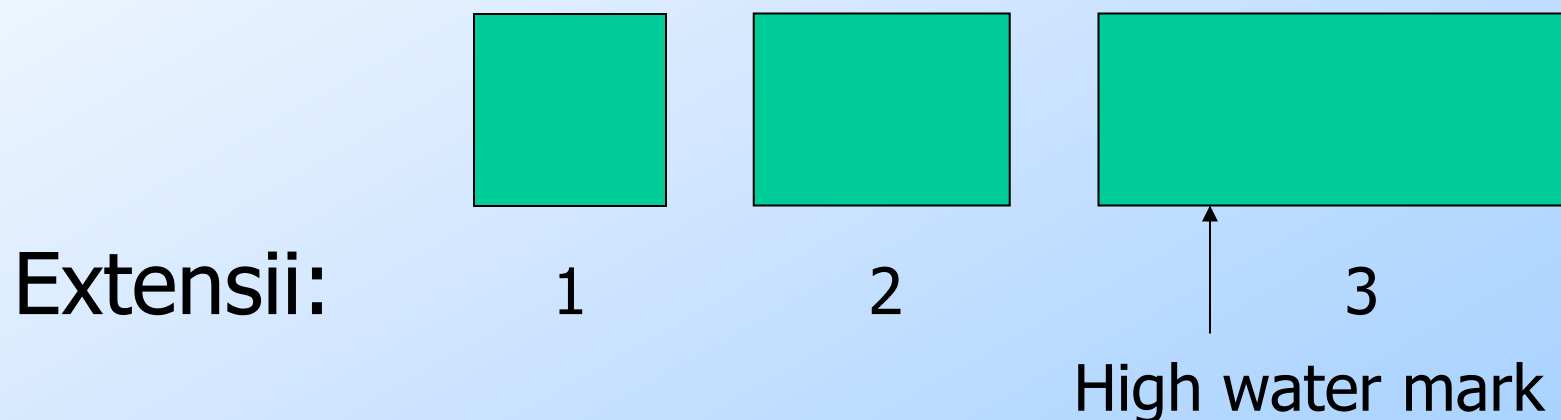
- ❑ PCTFREE – schimbarea sa afectează următoarele operații de inserare. Blocurile 'umplute' după vechea valoare nu sunt afectate decât în momentul în care ele ajung în lista de blocuri cu spațiu liber – FREELIST - deci lista de blocuri în care se pot face operații de inserare.
- ❑ Un bloc ajunge în lista de blocuri cu spațiu liber doar dacă se efectuează ștergeri din el pana sub PCTUSED

# Parametri utilizare bloc - cont

- ❑ PCTUSED – orice modificare a acestui parametru afectează toate blocurile din tabelă. Dacă o linie e actualizată sau ștearsa, blocul care o conține va fi testat dacă poate fi pus în lista de blocuri cu spațiu liber.
- ❑ INITTRANS – schimbarea acestui parametru va afecta doar blocurile noi
- ❑ MAXTRANS – vezi observația din cursul anterior. La schimbare va lua automat valoarea 255.

# High water mark

- Pentru orice segment (inclusiv deci pentru segmentele conținând tabele) exista un marcaj al ultimului bloc care a fost vreodată utilizat.
- Acest marcaj se numește 'high water mark' (HWM)





# De ce se numeste asa?



High water mark

Surse: [http://www.tripper.ro/view.php?ce=gal&gal\\_id=119&id=&limba\\_id=1](http://www.tripper.ro/view.php?ce=gal&gal_id=119&id=&limba_id=1) și <http://www.plaiurimioritice.ro/tag/lac-de-acumulare/>

# High water mark - cont

- ❑ Pe măsură ce datele sunt inserate în tabelă, marcajul HWM este mutat spre blocuri superioare
- ❑ Acest marcaj NU este resetat în momentul în care sunt șterse linii din tabela (resetarea se face doar în cazul în care se executa TRUNCATE)
- ❑ Când Oracle face o parcurgere completă a unei tabele (full table scan) atunci sunt citite toate blocurile pana la HWM, deci inclusiv blocuri golite ca urmare a ștergerilor de înregistrări (linii)!



# High water mark - cont

- Daca se dorește însă 'defragmentarea' tabelii se poate executa ALTER TABLE MOVE care mută o tabelă dintr-un tablespace în altul (cele doua nu sunt neapărat distincte):

**ALTER TABLE emp MOVE tablespace2**

- În acest caz se păstrează definițiile tuturor constrângerilor de integritate și ale indecșilor, dar aceștia din urma trebuie refăcuți (indecșii sunt bazați pe ROWID iar în procesul de compactare acesta se schimbă).

# Exemplu

❑ Creare tabela și umplere cu date:

```
create table table_size_test (  
  a char(100), b number )  
storage (initial 65K next 65K pctincrease 0)  
tablespace ts_01;
```

```
begin  
for i în 1 .. 10000 loop -- PL/SQL block  
  insert into table_size_test values  
    (dbms_random.string('X', 100), i);  
end loop;  
end;
```

```
/
```

```
commit;
```

# Exemplu - cont

❑ Crearea unui index:

```
create index ix_table_size_test on  
table_size_test(a) storage (initial 65K  
next 65K pctincrease 0)  
tablespace ts_02;
```

# Exemplu - cont

□ Vizualizare spațiu utilizat:

```
select substr(segment_name,1,20) segment,  
       bytes / 1024 "Size [KB]"  
from user_segments  
where segment_name în ('TABLE_SIZE_TEST',  
    'IX_TABLE_SIZE_TEST');
```

| SEGMENT            | Size [KB] |
|--------------------|-----------|
| -----              | -----     |
| TABLE_SIZE_TEST    | 1280      |
| IX_TABLE_SIZE_TEST | 1280      |

# Exemplu - cont

❑ Ștergere din tabela

```
delete from table_size_test where  
    mod(b,2)=0;  
commit;
```

❑ Vizualizare spațiu (rezultat)

| SEGMENT            | Size [KB] |
|--------------------|-----------|
| -----              |           |
| TABLE_SIZE_TEST    | 1280      |
| IX_TABLE_SIZE_TEST | 1280      |

# Exemplu - cont

- ❑ Alter table move

```
alter table table_size_test move;
```

- ❑ Vizualizare spațiu (rezultat)

| SEGMENT            | Size [KB] |
|--------------------|-----------|
| -----              |           |
| TABLE_SIZE_TEST    | 640       |
| IX_TABLE_SIZE_TEST | 1280      |

## Exemplu - cont

- Indexul însă a devenit UNUSABLE:

```
select status from user_indexes  
where index_name = 'IX_TABLE_SIZE_TEST';  
  
STATUS
```

-----

UNUSABLE

- Îl refacem:

```
alter index ix_table_size_test rebuild;
```

# Exemplu - cont

□ Date despre indexul refăcut (acum a devenit valid):

```
select status, bytes/1024
from user_indexes
join user_segments on index_name =
    segment_name
where index_name = 'IX_TABLE_SIZE_TEST';
```

| STATUS | BYTES/1024 |
|--------|------------|
|--------|------------|

|       |       |
|-------|-------|
| ----- | ----- |
|-------|-------|

|       |     |
|-------|-----|
| VALID | 704 |
|-------|-----|



# High water mark - cont

- ❑ Începând cu Oracle 10 se mai poate face ajustarea HWM în cazul segmentelor care utilizează ASSM – Automatic Segment Space Management astfel:
  - Permite schimbarea ROWID-ului liniilor:  
`ALTER TABLE emp ENABLE ROW MOVEMENT;`
  - Dăm comanda de shrink:  
`ALTER TABLE emp SHRINK SPACE;`
- ❑ Efectul comenzii de shrink este: mută liniile compactându-le și muta HWM. Pentru asta e nevoie de o blocare a tabelului dar pentru o perioadă scurtă de timp.

# High water mark - cont

## ❑ Variante ale comenzii:

1. Mutare linii și HWM într-o tabelă:

```
ALTER TABLE emp SHRINK SPACE;
```

2. Mutare linii și HWM într-o tabela plus compactare obiecte dependente (ex. indecși):

```
ALTER TABLE emp SHRINK SPACE CASCADE;
```

3. Muta doar liniile fără sa mute HWM:

```
ALTER TABLE emp SHRINK SPACE COMPACT;
```

# High water mark - cont

Restricții pentru SHRINK:

- ☐ Doar în tablespace-uri cu ASSM
- ☐ Nu se pot compacta (lista e mai lungă):
  - Segmente UNDO
  - Segmente temporare
  - Tabele de tip cluster
  - Tabele cu o coloană de tip LONG
  - Indecși de tip LOB
- ☐ Se poate utiliza pachetul de sistem DBMS\_SPACE pentru a vedea informații despre spațiul utilizat.

# DEALLOCARE SPATIU LIBER

- ❑ Spațiul liber ocupat de un segment (cel de după HWM) poate fi dealocat folosind:

```
ALTER TABLE [schema.]tabela
```

```
DEALLOCATE UNUSED [KEEP int [K | M] ]
```

- ❑ În cazul folosirii KEEP se păstrează o parte a acestui spațiu liber (dimensiunea e data în bytes, KB sau MB).
- ❑ Spațiul astfel dealocat poate fi folosit de alte segmente.

# DEALOCARE SPATIU - cont

- ❑ În cazul în care HWM este într-o extensie cu număr mai mic decât MINEXTENTS, se dealocă toate extensiile de după MINEXTENTS.
- ❑ Pentru a dealoca tot spațiul disponibil (pana la HWM), inclusiv în cazul în care HWM e sub MINEXTENTS, se folosește KEEP 0.

# Trunchiere

- ❑ Comanda de trunchiere golește o tabelă și resetează HWM.
- ❑ Spațiul ocupat de tabelă este dealocat în afara cazului când se specifica explicit REUSE STORAGE
- ❑ Sintaxa comenzii este:  

```
TRUNCATE TABLE [schema.]tabela  
[ { DROP | REUSE } STORAGE]
```
- ❑ Comanda TRUNCATE e o comanda DDL deci este comisa automat și nu se poate face rollback (nu poate fi anulată ca în cazul unui DELETE)

# DROP STORAGE

- ❑ În cazul DROP STORAGE:
  - Sunt dealocate toate extensiile superioare lui MINEXTENTS
  - HWM e resetat
  - Valoarea lui NEXT\_EXTENT (dimensiunea următoarei extensii care va fi alocată la nevoie) este resetată la valoarea extensiei cu numărul cel mai mic care a fost dealocată
- ❑ În ambele cazuri (REUSE sau DROP), trunchierea afectează toți indecșii tabelului respective.



# DROP TABLE

- ❑ Ștergerea unei tabele se face cu DROP TABLE
- ❑ Sintaxa este:

**DROP [schema.] tabela**  
**[CASCADE CONSTRAINTS]**

- ❑ Efectul este ștergerea tablei și a tuturor constrângerilor de integritate aferente (inclusiv cele referențiale – FOREIGN KEY)
- ❑ Dacă nu se specifica CASCADE tabela nu se poate șterge dacă exista constrângeri referențiale care o referă.



# Validare structura

- ❑ Se face cu comanda ANALYZE TABLE.
- ❑ Aceasta colectează statistici despre tabelă și le stochează în dicționarul de date.
- ❑ Printre alte opțiuni sunt și cele de:
  - Validare a structurii unei tabele
  - Identificarea liniilor care au migrat sau sunt înlănțuite
- ❑ În cazul validării structurii, toate blocurile tabelului sunt verificate din punct de vedere al integrității.

# VALIDATE STRUCTURE - cont

- Sintaxa este:

**ANALYZE TABLE [schema.]tabela**

**VALIDATE STRUCTURE [CASCADE]**

- În cazul folosirii opțiunii CASCADE, este validată și structura tuturor indecșilor asociați tablei și se face și o verificare încrucișată între conținutul de date al tablei și cel al indecșilor.

# Migrare și înlănțuire

- ❑ ANALYZE TABLE poate fi folosită și pentru detectarea liniilor care au migrat sau a celor înlănțuite (din cauza lui PCTUSED sau pentru că sunt prea voluminoase).
- ❑ Pentru aceasta, întâi se calculează sau se estimează statisticile asupra tabelului respective.
- ❑ Statisticile estimate se fac pe baza unui eșantion de linii (implicit 1064 linii).

# Migrare și înlănțuire - cont

- ❑ Sintaxa comenzii în acest caz este:

```
ANALYZE TABLE [schema.]tabela
```

```
{ COMPUTE STATISTICS
```

```
| ESTIMATE STATISTICS
```

```
[SAMPLE integer { ROWS | PERCENT }]
```

- ❑ COMPUTE va genera statistici pornind de la o parcurgere completă a tablei.
- ❑ La ESTIMATE se poate specifica (în linii sau în procente) dimensiunea eșantionului.

# Migrare și înlănțuire - cont

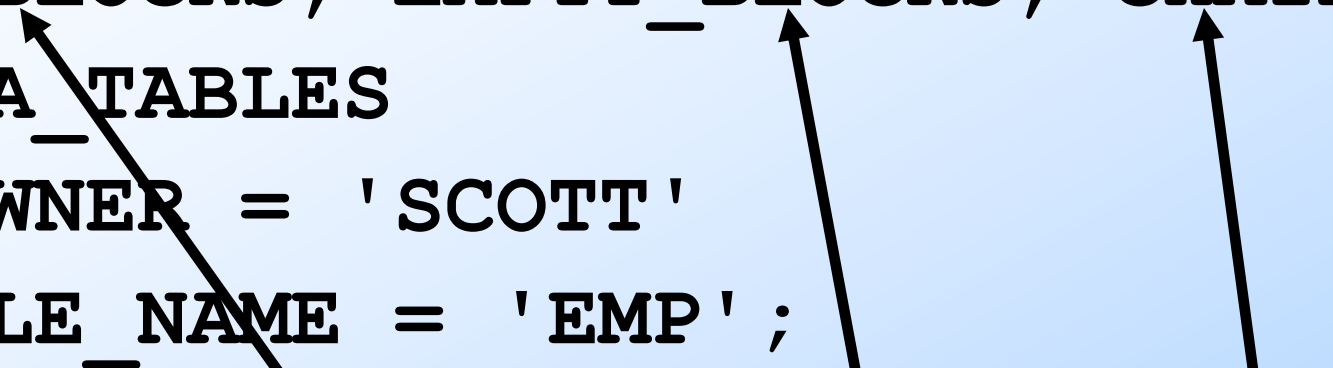
- ❑ După generarea statisticilor, în vederea de dicționar DBA\_TABLES există în coloana CHAIN\_CNT numărul de linii care sunt migrate sau înlănțuite.
- ❑ în cazul în care un număr mare de linii sunt în aceasta situație trebuie ca tabela sa fie reorganizată pentru a remedia această situație (de exemplu prin recrearea tablei folosind CREATE ... AS SELECT ... ORDER BY)

# VEDERI

- ❑ Pe lângă DBA\_TABLES se mai pot folosi și vederile DBA\_OBJECTS și DBA\_SEGMENTS.
- ❑ Toate cele 3 vederi pot fi unite (join) după condiția:  
DBA\_TABLES.OWNER = DBA\_OBJECTS.OWNER= DBA\_SEGMENTS.OWNER  
AND  
DBA\_TABLES.TABLE\_NAME = DBA\_OBJECTS.OBJECT\_NAME=  
DBA\_SEGMENTS.SEGMENT\_NAME

# Exemplu

```
SELECT BLOCKS, EMPTY_BLOCKS, CHAIN_CNT  
FROM DBA_TABLES  
WHERE OWNER = 'SCOTT'  
AND TABLE_NAME = 'EMP';
```



- Obținem în acest caz un rezultat conținând:
  - Prima coloana conține HWM
  - A doua numărul de blocuri de după HWM
  - A treia numărul de linii (înregistrări) migrate sau înlănțuite



```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> analyze table scott.emp compute statistics;
```

Table analyzed.

```
SQL> select blocks, empty_blocks, chain_cnt
2   from dba_tables
3   where owner='SCOTT'
4   and table_name='EMP';
```

| BLOCKS | EMPTY_BLOCKS | CHAIN_CNT |
|--------|--------------|-----------|
| 5      | 3            | 0         |

```
SQL> █
```



# DBA\_EXTENTS

- ❑ Această vedere poate fi folosită pentru a afla numărul de extensii și alte informații despre ele.
- ❑ Printre coloanele vederii sunt: OWNER, SEGMENT\_NAME, EXTENT\_ID, FILE\_ID, BLOCK\_ID, BLOCKS
- ❑ Fiecare linie reprezintă o extensie și în BLOCKS este numărul de blocuri ale acesteia.

# Lecturi obligatorii

1. Oracle Database Administrator's Guide (v19c) – Cap 20: Managing Tables.

<https://docs.oracle.com/en/database/oracle/oracle-database/19/admin/>

2. On shrinking table sizes:

<http://www.adp-gmbh.ch/blog/2005/july/20.html>

# Sfârșitul capitolul 5

# Capitolul 6

## Gestiune utilizatori, profiluri, privilegii, roluri - Partea 1 -

# Setari pentru useri

- ☐ Cota pe diverse tablespace-uri ✓
- ☐ Tablespace implicit (default)
- ☐ Tablespace temporar
- ☐ Blocare cont
- ☐ Creare / modificare user
- ☐ Limitari de resurse (profiluri)
- ☐ Privilegii user
- ☐ Roluri

# Mecanismul de autentificare

- Autentificarea userului se poate face in mai multe feluri:
  1. De catre serverul de BD (**database authentication**) – pe baza unui username si a unei parole (cum lucrati de obicei la laborator).
  2. Prin sistemul de operare (**operating system authentication**) – Oracle foloseste informatiile despre user aflate in sistemul de operare si il autentifica, nemaifiind necesara introducerea unui username si a unei parole.
  3. Prin retea (**network authentication**) – folosind servicii de autentificare third-party.

# Cota pe diverse tablespace-uri

- ❑ La crearea unui nou user se poate specifica spatiul pe care acel user il poate 'consuma' din diversele tablespace-uri care exista la acel moment in sistem.
- ❑ Nu se pot asocia cote pe tablespace-urile temporare
- ❑ Implicit userii nu au cote asociate cu nici un tablespace

SQL&gt;

SQL&gt;

SQL&gt;

SQL&gt; select \* from DBA\_TS\_QUOTAS;

| TABLESPACE_NAME | USERNAME    | BYTES    | MAX_BYTES | BLOCKS | MAX_BLOCKS | DRO |
|-----------------|-------------|----------|-----------|--------|------------|-----|
| BD_DATA         | STUD1       | 196608   | 10485760  | 24     | 1280       | NO  |
| BD_DATA         | UBD1        | 196608   | 10485760  | 24     | 1280       | NO  |
| BD_DATA         | UBD2        | 196608   | 10485760  | 24     | 1280       | NO  |
| SYSAUX          | APPQOSSYS   | 0        | -1        | 0      | -1         | NO  |
| SYSAUX          | FLows_FILES | 0        | -1        | 0      | -1         | NO  |
| SYSAUX          | SYSMAN      | 95092736 | -1        | 11608  | -1         | NO  |
| SYSAUX          | OLAPSYS     | 7667712  | -1        | 936    | -1         | NO  |
| BD_DATA         | DMUSER      | 0        | 10485760  | 0      | 1280       | NO  |
| BD_DATA         | STUD2       | 196608   | 10485760  | 24     | 1280       | NO  |

9 tnregistrÑri selectate.

SQL&gt; \_



SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>  
SQL>

SQL> CONNECT STUD1/student

Conectat.

SQL> SELECT \* FROM USER\_TS\_QUOTAS;

| TABLESPACE_NAME | BYTES  | MAX_BYTES | BLOCKS | MAX_BLOCKS | DRO |
|-----------------|--------|-----------|--------|------------|-----|
| USERS           | 0      | 0         | 0      | 0          | NO  |
| BD_DATA         | 196608 | 10485760  | 24     | 1280       | NO  |

SQL>

# Cota - cont

- ❑ Asignarea unei cote pentru un user intr-un tablespace are urmatoarele efecte:
  - Userii care au privilegiul de a crea obiecte pot crea acele obiecte in tablespace-ul respectiv.
  - Oracle limiteaza spatiul pe care acele obiecte il pot ocupa in tablespace-ul specificat la cat spune cota alocata.
- ❑ Se poate inhiba pentru un user posibilitatea de creare de noi obiecte intr-un anumit tablespace prin setarea unei cote egale cu 0

# Cota - cont

- Cand cota unui user este modificata la o valoare mai mica decat spatiul ocupat la acel moment de acel user in acel tablespace (inclusiv la setarea unei cote egala cu 0) obiectele existente nu se sterg dar:
  - Nu se mai pot crea noi obiecte
  - Obiectele existente nu mai pot creste in dimensiune (dar pot scadea)

# Setari pentru useri

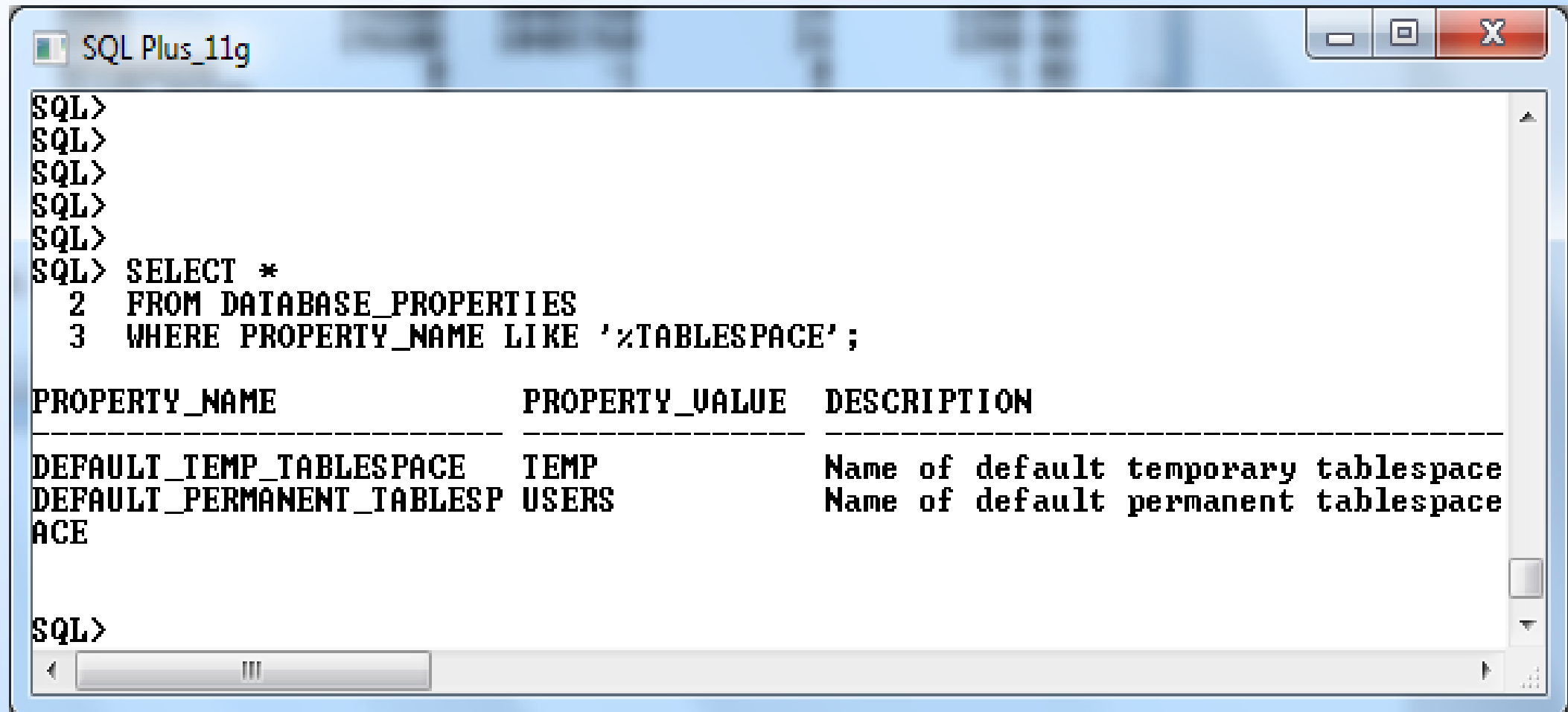
- ☐ Cota pe diverse tablespace-uri ✓
- ☐ Tablespace implicit (default) ✓
- ☐ Tablespace temporar
- ☐ Blocare cont
- ☐ Creare / modificare user
- ☐ Limitari de resurse (profiluri)
- ☐ Privilegii user
- ☐ Roluri

# Tablespace implicit (default)

- ❑ Orice user are un tablespace implicit (default).
- ❑ Acest tablespace definește locația unde sunt create obiectele (segmentele) userului în absența specificării unui tablespace în momentul creării acelui obiect.
- ❑ La crearea unui nou user se poate optional specifica tablespace-ul implicit al userului (cel permanent și cel temporar).
- ❑ Dacă nu se specifică aceste informații userul va moșteni valorile implicite ale bazei de date.

# Tablespace implicit (default)

- ❑ Aflarea tablespace-urilor implicite se poate face din DATABASE\_PROPERTIES:



```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> SELECT *
      2  FROM DATABASE_PROPERTIES
      3  WHERE PROPERTY_NAME LIKE '%TABLESPACE';
```

| PROPERTY_NAME                | PROPERTY_VALUE | DESCRIPTION                          |
|------------------------------|----------------|--------------------------------------|
| DEFAULT_TEMP_TABLESPACE      | TEMP           | Name of default temporary tablespace |
| DEFAULT_PERMANENT_TABLESPACE | USERS          | Name of default permanent tablespace |

```
SQL>
```

# Tablespace implicit (default)

- ❑ In Oracle valoarea de default este tablespace-ul SYSTEM, ceea ce nu este foarte bine in cazul in care userul creaza noi obiecte.
- ❑ Este bine sa se creeze un tablespace permanent si unul temporar iar userii uzuali sa le aiba pe acestea ca implicite.
- ❑ Userii de sistem (SYS, SYSTEM) trebuie insa sa ramana cu tablespace implicit SYSTEM.
- ❑ Tablespace-urile default ale unui user se pot schimba si dupa crearea userului, cu ALTER USER.

# Setari pentru useri

- ☐ Cota pe diverse tablespace-uri ✓
- ☐ Tablespace implicit (default) ✓
- ☐ Tablespace temporar ✓
- ☐ Blocare cont
- ☐ Creare / modificare user
- ☐ Limitari de resurse (profiluri)
- ☐ Privilegii user
- ☐ Roluri



# Tablespace temporar

- ❑ In cazul in care sunt folosite segmente temporare (de exemplu sunt executate cereri care implica sortari de date voluminoase), acestea sunt stocate:
  - In tablespace-ul implicit (default) daca nu s-a specificat un tablespace temporar la crearea userului
  - In tablespace-ul temporar daca acesta a fost specificat
- ❑ Si acest tablespace se poate specifica si ulterior, prin **ALTER USER**

# Aflarea valorilor implicite

- Pentru a afla valorile implicite ale unui user existent se poate interoga si vederea DBA\_USERS:

```
SQL> select USERNAME, DEFAULT_TABLESPACE,  
        TEMPORARY_TABLESPACE  
from DBA_USERS  
where USERNAME='stud1' ;
```

| USERNAME | DEFAULT_TABLESPACE | TEMPORARY_TABLESPACE |
|----------|--------------------|----------------------|
| stud1    | USERS              | TEMP                 |

```
SQL>
```

# Setari pentru useri

- ☐ Cota pe diverse tablespace-uri ✓
- ☐ Tablespace implicit (default) ✓
- ☐ Tablespace temporar ✓
- ☐ Blocare cont ✓
- ☐ Creare / modificare user
- ☐ Limitari de resurse (profiluri)
- ☐ Privilegii user
- ☐ Roluri

# Blocare cont

- ❑ Un cont poate fi configurat sa se blocheze dupa un anumit numar de incercari de intrare fara succes.
- ❑ Contul se poate debloca dupa un anumit interval de timp, specificat, sau de catre administratorul bazei de date.
- ❑ De asemenea, parola de la creare se poate seta ca expirata, fortand astfel schimbarea parolei (de user sau de administratorul bazei de date) inainte de a putea intra in sistem.

# Obiectele unui user

- ❑ Ele formeaza 'schema' acelui user
- ❑ Pot fi:
  - ❑ Tabele (cu declansatori si constrangeri asociate)
  - ❑ Indecsi
  - ❑ Vederi
  - ❑ Secvente
  - ❑ Subprograme stocate
  - ❑ Sinonime
  - ❑ Tipuri definite de user
  - ❑ Legaturi (database links – prin ele se pot accesa obiecte din alte baze de date)

# Setari pentru useri

- ☐ Cota pe diverse tablespace-uri ✓
- ☐ Tablespace implicit (default) ✓
- ☐ Tablespace temporar ✓
- ☐ Blocare cont ✓
- ☐ Creare / modificare user ✓
- ☐ Limitari de resurse (profiluri)
- ☐ Privilegii user
- ☐ Roluri

# Crearea unui nou user

- ❑ La crearea unui nou user se stabilesc mai intai urmatoarele:
  - Numele, parola si metoda de autentificare pentru acel user
  - Tablespace-urile care pot fi utilizate de catre acesta
  - Cota alocata userului pentru fiecare tablespace
  - Tablespace-ul implicit si cel temporar
- ❑ Se emite comanda CREATE USER care foloseste informatiile de mai sus
- ❑ Se adauga apoi privilegii si roluri pentru user.

# Sintaxa

```
CREATE USER username
IDENTIFIED {BY password
           | EXTERNALLY
           | GLOBALLY AS 'external_name' }
[ DEFAULT TABLESPACE tablespace ]
[ TEMPORARY TABLESPACE tablespace ]
[ QUOTA int {K | M} ON tablespace ]
[ QUOTA UNLIMITED ON tablespace ]
[ PROFILE { profile_name | DEFAULT } ]
[ PASSWORD EXPIRE ]
[ ACCOUNT {LOCK|UNLOCK} ]
```



# Detalii

```
IDENTIFIED {BY password  
            | EXTERNALLY ...  
            | GLOBALLY ...}
```

- ❑ Aceasta clauza spune modul de autentificare pentru acest user:
- ❑ BY password arata ca este un user local care trebuie sa specifice username si parola la login,
- ❑ EXTERNALLY indica un user extern, autentificat fie prin sistemul de operare fie prin servicii third party
- ❑ GLOBALLY arata ca este un user global, autentificat prin 'directory services'

# Detalii

[ **DEFAULT TABLESPACE** *tablespace* ]

- ❑ Aceasta clauza specifica tablespace-ul default (implicit)

[ **TEMPORARY TABLESPACE** *tablespace* ]

- ❑ Aceasta clauza specifica tablespace-ul pentru segmente temporare

[ **QUOTA** *int* {**K** | **M**} **ON** *tablespace* ]

- ❑ Aceasta clauza specifica valoarea cotei pe un anumit tablespace in bytes / KB / MB.

[ **QUOTA UNLIMITED** **ON** *tablespace* ]

- ❑ Aceasta clauza specifica faptul ca nu este fixata o limita superioara pentru cota pe acel tablespace (bineinteles segmentele userului nu pot depasi spatiul existent acolo)

# Detalii

[ **PROFILE** { *profile\_name* | **DEFAULT** } ]

- ❑ Specifica profilul asociat cu acel user, acesta aratand limitările privind resursele pe care le poate consuma userul. Dacă nu se specifica, va fi asociat un profil implicit (numit **DEFAULT**).

[ **PASSWORD EXPIRE** ]

- ❑ Specifica faptul ca parola este 'pre-expirata', deci DBA sau userul trebuie sa o schimbe inainte de a putea intra in acel cont

[ **ACCOUNT** { **LOCK** | **UNLOCK** } ]

- ❑ Specifica faptul ca acel cont este blocat (**LOCK**), deci necesita deblocare inainte de a fi utilizat. Implicit contul este deblocat (**UNLOCK**) si se poate lucra.

# Exemplu

## ❑ User autentificat prin parola:

```
CREATE USER mihai341C5  
IDENTIFIED BY ec004  
DEFAULT TABLESPACE users  
QUOTA 100M ON test  
QUOTA 500K ON users  
TEMPORARY TABLESPACE temp  
PROFILE clerk;
```

## ❑ Se adauga si niste privilegii:

```
GRANT create session TO mihai341C5;
```

# Restrictii pentru parola

- ❑ In cazul in care autentificarea se face prin parola, aceasta trebuie sa verifice restrictiile de nume Oracle:
  1. Maximum 30 de caractere (pana in versiunea 11g este case-insensitive.)
  2. Incepe cu o litera
  3. Contine litere, cifre sau caracterele speciale:  
# \_ \$

Motivatia acestor restrictii tine de sintaxa comenzii de creare a unui user cu specificarea parolei sau a modificarii parolei sale - aceasta nu se pune intre apostrofi deci trebuie sa respecte regulile pe care le respecta si numele de obiecte.

Exemplu: `CREATE USER U1 IDENTIFIED BY PAROLA_MEA`

# Restrictii pentru parola

❑ Incepand cu versiunea 11g se pot seta parole 'case sensitive':

Exemplu: creare user test2 cu parola Test2

```
CONN / AS SYSDBA
```

```
ALTER SYSTEM SET SEC_CASE_SENSITIVE_LOGON = TRUE;
```

```
CREATE USER TEST2 IDENTIFIED BY Test2;
```

```
GRANT CONNECT TO test2
```

```
-- autentificare
```

```
SQL> CONNECT TEST2/Test2
```

```
Connected.
```

```
SQL> CONNECT TEST2/test2
```

```
ERROR:
```

```
ORA-01017: invalid username/password; logon denied
```

# Restrictii pentru parola

In cazul in care se seteaza SEC\_CASE\_SENSITIVE\_LOGON pe FALSE literele mari vor fi la fel cu cele mari.

Exemplu: pentru userul anterior:

```
CONN / AS SYSDBA
```

```
ALTER SYSTEM SET SEC_CASE_SENSITIVE_LOGON = FALSE;
```

```
SQL> CONN TEST2/Test2
```

```
Connected.
```

```
SQL> CONN TEST2/test2
```

```
Connected.
```

```
SQL>
```

# Modificare date user

- ❑ Datele privind autentificarea userului:

```
ALTER USER username  
IDENTIFIED {BY password  
            | EXTERNALLY ...  
            | GLOBALLY ...}  
[ PASSWORD EXPIRE ]  
[ ACCOUNT {LOCK|UNLOCK} ]
```

- ❑ In momentul blocarii unui cont (LOCK), daca userul e logat la acel moment nu va fi afectat. Modificarile date de comanda de mai sus sunt valabile **incepand cu urmatoarea sesiune** de lucru.



```
SQL>
SQL> CONNECT UBD1 AS SYSDBA
Introduce■i parola:
Conectat.
SQL> ALTER USER STUD1 ACCOUNT LOCK;
```

Utilizator modificat.

```
SQL> CONNECT STUD1
Introduce■i parola:
ERROR:
ORA-28000: the account is locked
```

Aten■ie: Nu mai sunte■i conectat la ORACLE.

```
SQL> CONNECT UBD1 AS SYSDBA
Introduce■i parola:
Conectat.
SQL> ALTER USER STUD1 ACCOUNT UNLOCK;
```

Utilizator modificat.

```
SQL> CONNECT STUD1
Introduce■i parola:
Conectat.
SQL> _
```

# Modificare date user - cont

- Datele privind tablespace si cote:

```
ALTER USER username
```

```
[ DEFAULT TABLESPACE tablespace ]
```

```
[ TEMPORARY TABLESPACE tablespace ]
```

```
[ QUOTA int {K | M} ON tablespace ]
```

```
[ QUOTA UNLIMITED ON tablespace ]
```

- La trecerea pe 0 a cotei nu se mai pot crea obiecte si cele existente nu mai pot creste. Exemplu:

```
ALTER USER mihai341c5
```

```
QUOTA 0 ON users;
```

# Modificare date user - cont

- ❑ Observatie: trecerea pe 0 a cotei nu are efect daca userul are asignat rolul (colectia de privilegii) **RESOURCE** deoarece aceasta implica o cota nelimitata.

# Exemplu: stud1 are rolul RESOURCE

SQL Plus\_11g

```
SQL>
SQL> connect stud1/student
Conectat.
SQL> create table t(a number);

Tabelul creat.

SQL> connect ubd1/ubd1
Conectat.
SQL> alter user stud1 quota 0 on USERS;

Utilizator modificat.

SQL> connect stud1/student
Conectat.
SQL> create table t1(b number);

Tabelul creat.

SQL>
```

# Stergere user

- ❑ Stergerea unui user se face cu comanda DROP USER:

**DROP USER** *nume* [**CASCADE**]

- ❑ Optiunea CASCADE sterge intai toate obiectele din schema userului respectiv (altfel se obtine un mesaj de eroare).
- ❑ Fara CASCADE se pot sterge doar useri care nu detin nici un obiect in schema proprie.

```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> CONNECT UBD1/ubd1
Conectat.
SQL> DROP USER STUD1;
DROP USER STUD1
*
EROARE la linia 1:
ORA-01922: CASCADE trebuie specificat pentru a elimina 'STUD1'
```

```
SQL>
```

# Vederi care se pot utiliza

| View                              | Description                                                                                                      |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------|
| DBA_USERS                         | Describes all users of the database.                                                                             |
| ALL_USERS                         | Lists users visible to the current user, but does not describe them.                                             |
| USER_USERS                        | Describes only the current user.                                                                                 |
| DBA_TS_QUOTAS ,<br>USER_TS_QUOTAS | Describes tablespace quotas for users.                                                                           |
| USER_PASSWORD_LIMITS              | Describes the password profile parameters that are assigned to the user (vezi partea despre profiluri din curs). |
| USER_RESOURCE_LIMITS              | Displays the resource limits for the current user (vezi partea despre profiluri din curs).                       |
| DBA_PROFILES                      | Displays all profiles and their limits.                                                                          |
| RESOURCE_COST                     | Lists the cost for each resource.                                                                                |
| V\$SESSION                        | Lists session information for each current session. Includes user name.                                          |
| V\$SESSTAT                        | Lists user session statistics.                                                                                   |
| V\$STATNAME                       | Displays decoded statistic names for the statistics shown in the V\$SESSTAT view.                                |
| PROXY_USERS                       | Describes users who can assume the identity of other users.                                                      |

# Exemplu

```
SELECT TABLESPACE_NAME, BLOCKS, MAX_BLOCKS, BYTES,  
       MAX_BYTES  
FROM DBA_TS_QUOTAS  
WHERE USERNAME = 'SCOTT';
```

- ❑ Se obtine un rezultat care contine date despre cota userului:

| TABLESPACE_NAME | BLOCKS | MAX_BLOCKS | BYTES | MAX_BYTES |
|-----------------|--------|------------|-------|-----------|
| DATE            | 10     | -1         | 20480 | -1        |

- ❑ Valoarea -1 reprezinta cota nelimitata. Restul valorilor reprezinta spatiul ocupat la acel moment.



# Alt exemplu

```
SELECT USERNAME, ACCOUNT_STATUS,  
       TEMPORARY_TABLESPACE  
FROM DBA_USERS
```

□ Se obtine o lista cu starea fiecarui cont (si alte date):

| USERNAME | ACCOUNT_STATUS | TEMPORARY_TABLESPACE |
|----------|----------------|----------------------|
| SYS      | OPEN           | TEMP                 |
| SYSTEM   | OPEN           | TEMP                 |
| DBSNMP   | OPEN           | TEMP                 |
| SCOTT    | OPEN           | TEMP                 |

# Setari pentru useri

- ☐ Cota pe diverse tablespace-uri ✓
- ☐ Tablespace implicit (default) ✓
- ☐ Tablespace temporar ✓
- ☐ Blocare cont ✓
- ☐ Creare / modificare user ✓
- ☐ Limitari de resurse (profiluri) ✓
- ☐ Privilegii user
- ☐ Roluri

# PROFIL

- ❑ Profilurile sunt o modalitate prin care se pot limita resursele care pot fi utilizate de un utilizator.
- ❑ Un profil se creaza cu CREATE PROFILE si se asigneaza userului la creare sau ulterior prin comanda ALTER USER.
- ❑ Exista un profil DEFAULT care se asociaza implicit la userii pentru care la creare nu s-a specificat un profil.

# Resurse ale sistemului

Pentru ca aceste limitari de sistem sa fie active trebuie ca parametrul de initializare RESOURCE\_LIMIT sa fie setat pe TRUE – se poate modifica folosind ALTER SYSTEM

- ❑ Numarul maxim de sesiuni concurente pentru user (**SESSIONS\_PER\_USER**)
- ❑ Timp CPU per sesiune (**CPU\_PER\_SESSION**) – masurat in sutimi de secunda.
- ❑ Timp CPU per operatie (**CPU\_PER\_CALL**) – masurat in sutimi de secunda. O operatie este un ciclu parse, execute, fetch.

# Resurse ale sistemului

- ❑ Timpul maxim de conectare masurat in minute (**CONNECT\_TIME**).  
Sesiunile userului sunt inchise de Oracle dupa expirarea acestui timp.
- ❑ Timp maxim de asteptare (**IDLE\_TIME**) – masurat in minute  
- sesiunile vor fi inchise de Oracle dupa expirarea perioadei specificate daca in sesiunea respectiva nu s-a facut nimic (e 'idle'). Atentie: cererile a caror executie este lunga nu intra in aceasta categorie!

# Resurse ale sistemului

- ❑ Numar maxim de blocuri citite per sesiune. Este vorba aici de numarul de blocuri citite de pe disc ***sau*** din memorie. Acest parametru este gandit pentru a limita cererile care fac citiri intensive (**LOGICAL\_READS\_PER\_SESSION**).
- ❑ Numarul maxim de blocuri citite per operatie (call) (**LOGICAL\_READS\_PER\_CALL**).
- ❑ Dimensiunea maxima de memorie ocupata in shared pool – parte a SGA - de o sesiune de lucru – in bytes (**PRIVATE\_SGA**).

# Resurse legate de parola

- ❑ Numarul maxim de incercari eronate de login (**FAILED\_LOGIN\_ATTEMPTS**)
- ❑ Timpul maxim (in zile) cat parola este valida (**PASSWORD\_LIFE\_TIME**)
- ❑ Numarul minim de parole diferite utilizate pana cand o parola poate fi reutilizata (**PASSWORD\_REUSE\_MAX**)
- ❑ Numarul minim de zile dupa care o parola poate fi reutilizata (**PASSWORD\_REUSE\_TIME**)

# Resurse legate de parola

Mai exista si:

- ❑ **PASSWORD\_LOCK\_TIME** : Cate zile se blocheaza contul dupa incercari repetate de login esuate
- ❑ **PASSWORD\_GRACE\_TIME** : Cate zile sunt disponibile pentru a schimba o parola dupa expirarea acesteia
- ❑ **PASSWORD\_VERIFY\_FUNCTION**: bloc (program) PL/SQL utilizat pentru verificarea parolei
- ❑ **SEC\_CASE\_SENSITIVE\_LOGON** : literele mari si cele mici sunt considerate identice sau nu intr-o parola.



# Alte informatii

- ❑ Lista de mai sus nu este exhaustiva.
- ❑ Am dat numele parametrilor pentru ca fiecare in parte se poate modifica ulterior prin comenzi ALTER PROFILE.

# Limitari

- ❑ Daca este atinsa o limita la nivel de sesiune atunci:
  - Fie se afiseaza un mesaj de eroare (de exemplu cand se incearca deschiderea unei noi sesiuni si se depaseste sessions\_per\_user)
  - Fie Oracle deconecteaza userul (sesiunea), de exemplu cand s-a atins durata ei maxima.

# Limitari

□ Daca este atinsa o limita la nivel de operatie (call) atunci:

- Procesarea cererii curente este oprita
- Cererea curenta este revocata (rollback)
- Efectul cererilor anterioare persista
- Userul ramane conectat.

# Lecturi obligatorii

1. Oracle Database Security Guide (v19c) – Capitolele despre gestiune utilizatori, privilegii, profiluri, roluri

<https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/database-security-guide.pdf>

# Sfârșit partea 1