Administrare BD

Stocare (BD fizică Oracle 12c)

Backup/Recovery

Administrarea spațiului BD fizică

- Serverul de baze de date Oracle gestionează automat spațiul pe disc
- Eventualele probleme sunt semnalate sub formă de alerte și sunt recomandate soluții posibile

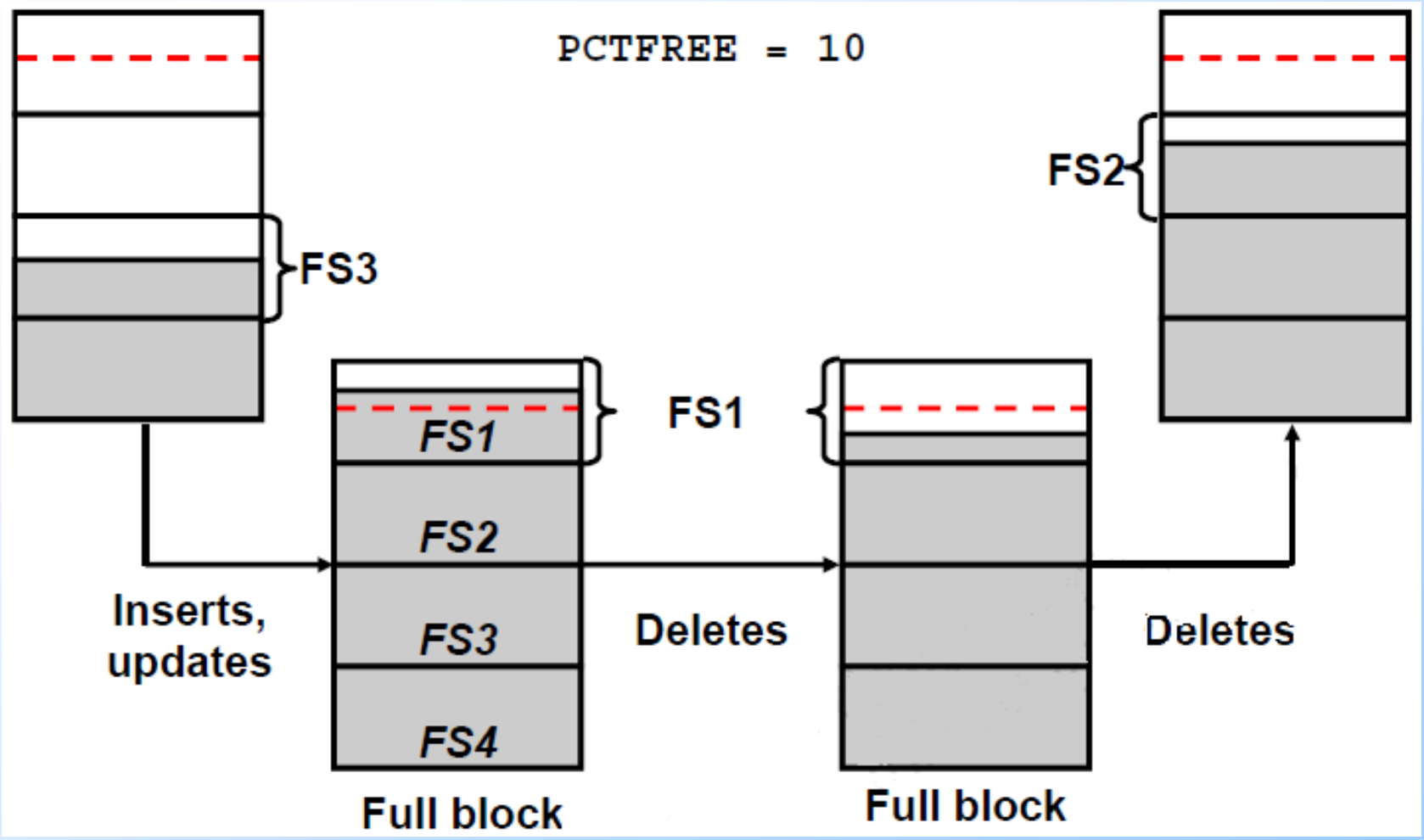
Administrarea spațiului

- Oracle Managed Files (OMF)
  - Operațiile se specifică în termeni obiecte BD în loc de fișiere pe disc
- Pentru un tablespace Oracle folosește bitmaps la gestionarea spațiului (locally managed) – Automatic Segment Space Management
- Se dă posibilitatea ca fișierele să-și mărească dimensiunea automat în funcție de gradul de umplere al tablespace-ului

Administrarea spațiului

- Gestiune proactivă a spațiului
  - Când spațiul liber scade sub un prag specificat, se declanșează alerta
- Planificarea spațiului (capacity planning)
  - Serverul BD estimează spațiul pe baza structurii tabelelor, a cardinalității tabelelor și a tendinței de creștere pe baza istoricului, raport stocat în Automatic Workload Repository (AWR)

Block Space Management



Block Space Management

- Automatic Segment Space Management împarte fiecare bloc de date în 4 secțiuni FS_i, i=1 (75% - 100% spațiu liber) .. 4 (0% - 25% spațiu liber)
- În funcție de lungimea unei tuple inserate se poate cunoaște dacă un bloc satisface tupla inserată
- „full” reprezintă starea în care un bloc nu mai este disponibil pentru operații insert

Block Space Management

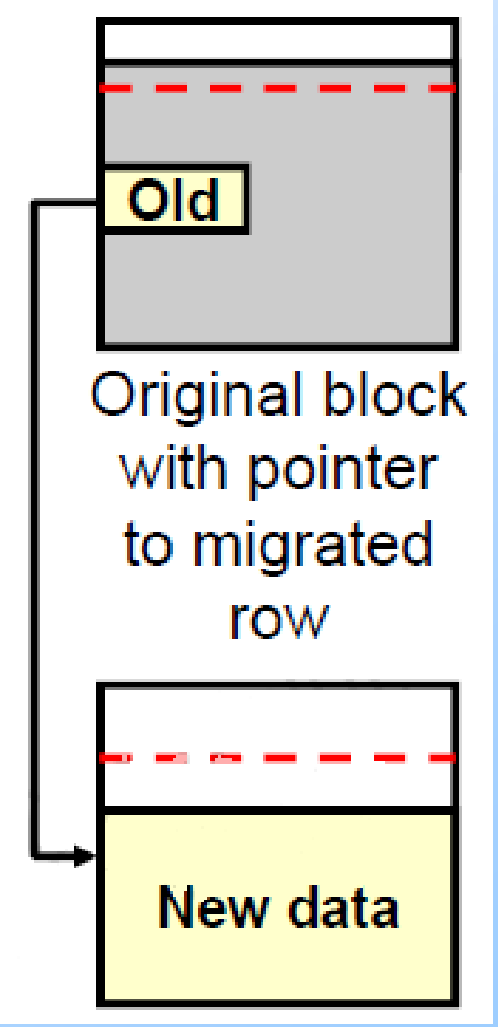
- Coloanele de tip LOB (Large Object): BLOB, CLOB, NCLOB și BFILE nu folosesc parametrul PCTFREE
- Blocurile folosite la sistemele OLTP și blocurile necomprimate au parametrul PCTFREE implicit 10
- Blocurile comprimate au parametrul PCTFREE implicit 0

Block Space Management

- Oracle recomandă Oracle Database block de dimensiuni 2 KB sau 4 KB pentru Online Transaction Processing (OLTP) sau „mixed workload environments” și dimensiuni mai mari, de: 8 KB, 16 KB, sau 32 KB pentru Decision Support System (DSS).

Row chaining and migration

- Să presupunem o tabelă care are o coloană de tip VARCHAR, de exemplu Nume Persoană
- La INSERT se alocă spațiu într-un bloc oarecare
- La UPDATE dimensiunea tuplei poate să crească (inițial erau 3 caractere de exemplu „Pop” și se modifică la noua valoare „Popescu”, +4 octeți) și să depășească spațiul liber din bloc
- Se alocă spațiu într-un alt bloc din același segment
- ROWID se păstrează



Row chaining and migration

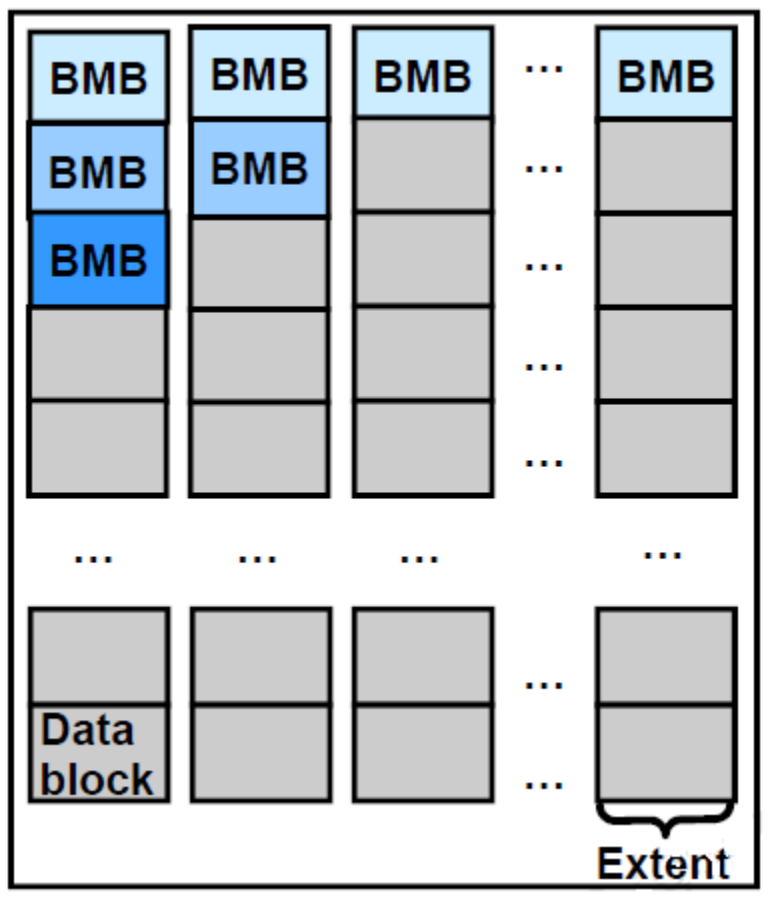
- Înlănțuirea tuplelor nu poate fi evitată la tabele cu coloane de tip LONG sau LONG RAW, în al doilea caz scenariul de bază la UPDATE este ca tupla să fie migrată în întregime într-un nou bloc dacă spațiul liber din blocul curent este ocupat, ROWID se păstrează
- Înlănțuirea tuplelor conduce la scăderea performanței, deoarece serverul BD trebuie să citească două sau mai multe blocuri

Row chaining and migration

- Segmentele ce conțin tuple înlănțuite pot fi descoperite cu Segment Advisor
- Spațiul liber fragmentat din interiorul unui bloc este fuzionat automat atunci când:
  - La INSERT sau UPDATE se încearcă să se folosească un bloc cu suficient spațiu liber
  - Spațiul liber este fragmentat a.î. datele nu pot fi scrise într-o zonă contiguă a unui bloc

Gestiunea spațiului liber din segmente

- Este urmărit prin bitmaps (Automatic Segment Space Management la creare tablespace)
- În zona de început a segmentului există blocuri bitmap (BMB) ce descriu utilizarea spațiului pentru blocurile de date



BMB sunt organizate arborescent cu maxim 3 nivele

Tipuri de segmente

- Set de extent-uri alocate pentru o structură logică de tip:
  - Tabelă și cluster
  - Index
  - Undo
  - Temporar
- Alocarea se face dinamic de serverul BD

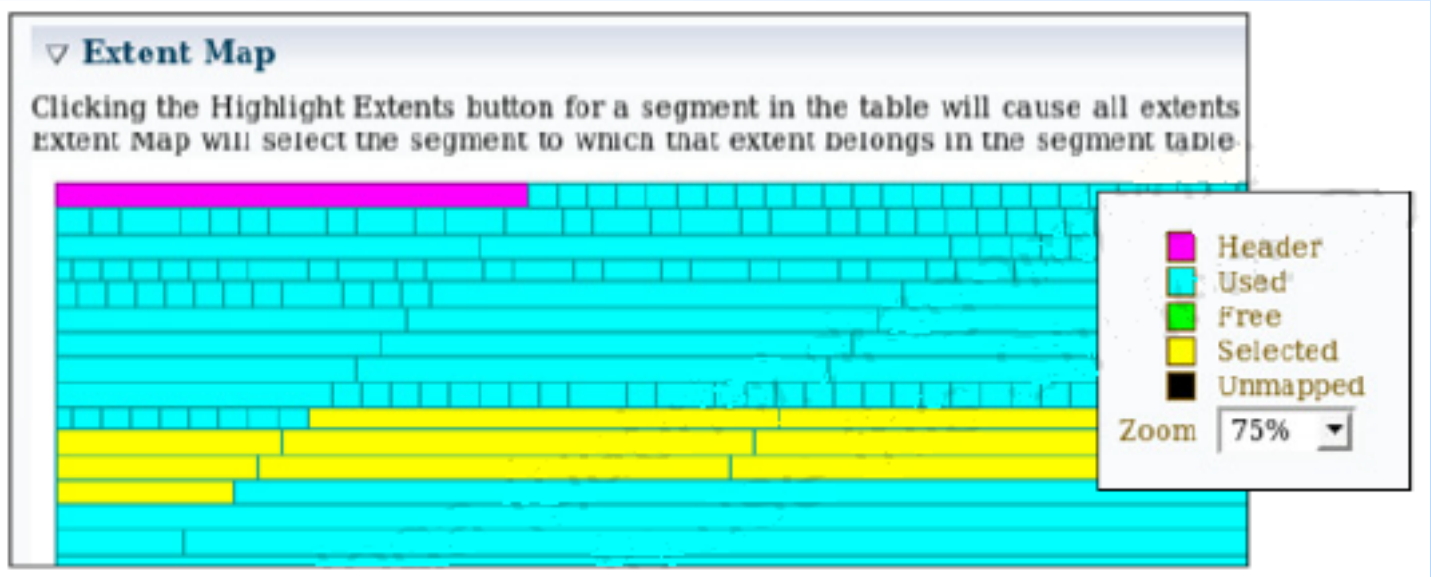
Tipuri de segmente

- Segment de tip Tabelă
  - Fiecare tabel ne-clusterizat are datele stocate în extent-uri ale unui segment tabelă
  - O tabelă partiționată are câte un segment pentru fiecare partiție
- Cluster
  - Fiecare cluster are un segment de date, unde sunt stocate datele fiecărei tabele din cluster

Tipuri de segmente

- Segment Index
  - Fiecare index are un astfel de segment
  - Pentru index partiționat, fiecare partiție are un segment index
- Segment Undo
  - Se folosește pentru a reface starea datelor (rollback)
- Segment Temporar
  - Este creat atunci când o instrucțiune SQL are nevoie de spațiu temporar, la terminarea instrucțiunii spațiul (extent-urile) este redat sistemului

Alocare extent-uri



- Serverul BD determină fișierul de date al tablespace-ului
- Se caută în bitmap-urile fișierului de date blocuri adiacente libere
- Dacă nu există blocuri libere, se caută în alt fișier de date

DEFERRED_SEGMENT_CREATION

- Implicit este TRUE
- Când este creată o nouă tabelă, definiția tabelei se salvează în dicționarul datelor, dar nu este alocat spațiu pentru date
- La primul INSERT în tabelă se alocă spațiu
- Avantaje:
  - Se salvează spațiu la instalarea unei aplicații cu sute de tabele, dar care nu sunt populate cu date
  - Durata de instalare de reduce mult

DEFERRED_SEGMENT_CREATION

- Exemplu
  - SQL> SHOW PARAMETERS deferred_segment_creation;
  - SQL>CREATE TABLE seg_test(c NUMBER, d VARCHAR2(500));
  - SQL>SELECT segment_name FROM user_segments;
    - Obs. Nu afișează nimic
  - SQL>INSERT INTO seg_test VALUES(1, 'aaaaaaaaaa')
  - SQL>SELECT segment_name FROM user_segments;
    - Obs. Afișează SEG_TEST
  - Alte vederi ce pot fi interogate pentru coloana SEGMENT_CREATED: USER_TABLES, USER_INDEXES, USER_LOBS (arată YES când s-a alocat spațiu
  - Tabela SYS.SEG$ spune ce parametri s-au folosit la creare tabelă

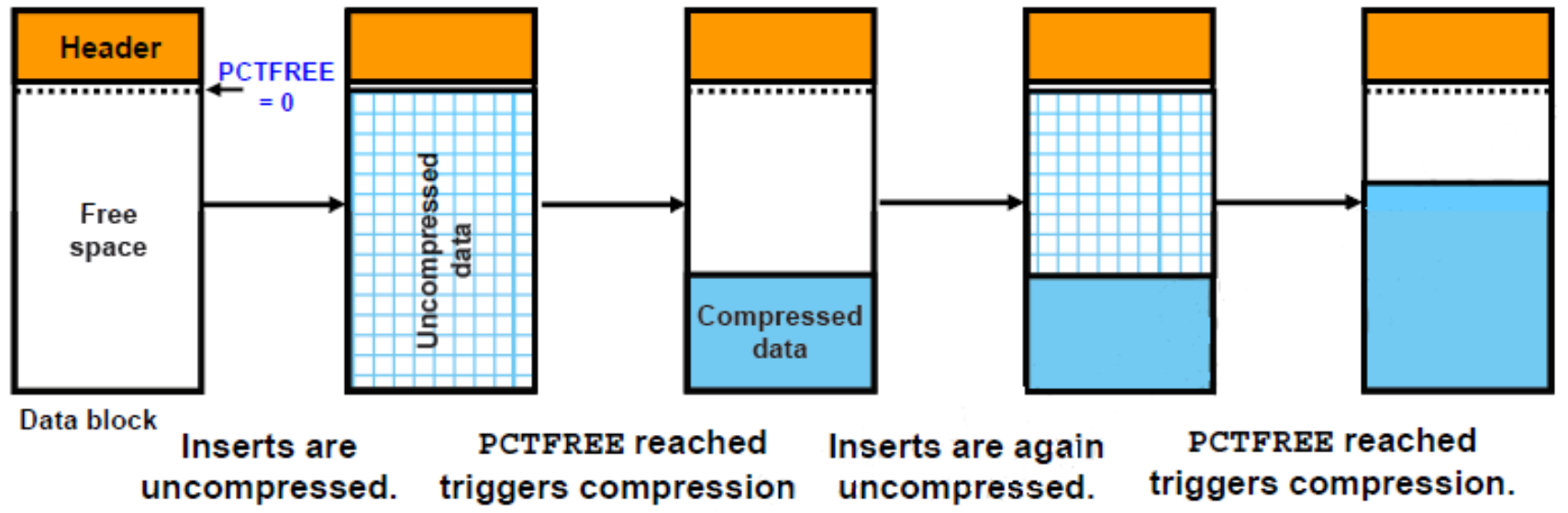
Controlul DEFERRED SEGMENT CREATION

- Cu parametrul DEFERRED_SEGMENT_CREATION ce poate fi setat:
  - În fișierul de inițializare
  - Cu ALTER SESSION
    - SQL>ALTER SESSION SET DEFERRED_SEGMENT_CREATION = TRUE;
  - Cu ALTER SYSTEM
- Cu clauza SEGMENT CREATION la CREATE TABLE
  - IMMEDIATE
  - DEFERRED (implicit)
- Indecșii moștenesc de la tabela de bază

Comprimarea spațiului

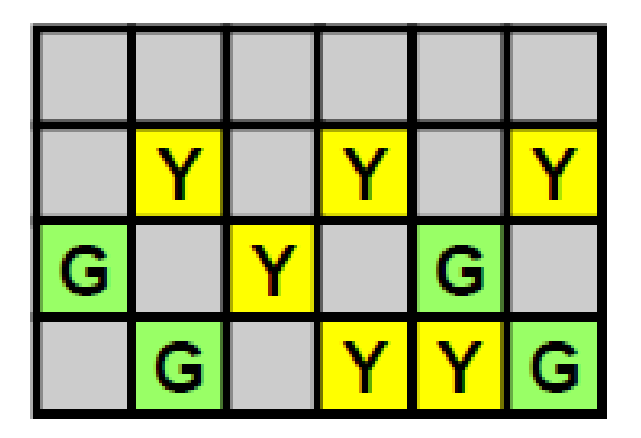
- Trei metode:
  - Basic table compression
  - Advanced row compression
  - Hybrid columnar compression (cu Exadata)
- Cuvântul cheie COMPRESS
- Basic: Operații bulk (de exemplu CREATE TABLE as SELECT ..)
- Advanced: la orice operație DML

Comprimare pentru operații INSERT Direct-Path

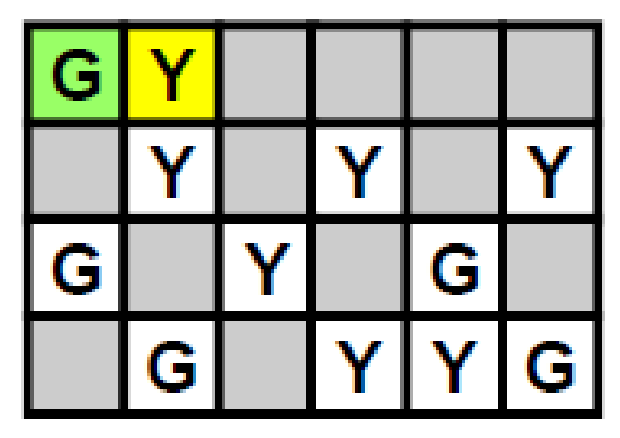


- CREATE TABLE .. COMPRESS BASIC .. ;
- Se folosește la încărcare date bulk în datawarehouse
- Maximizează spațiul liber contiguu în blocuri

Advanced row compression



Bloc necomprimat



Compresie OLTP cu tabelă de simboluri la începutul blocului

- CREATE TABLE .. ROW STORE COMPRESS ADVANCED ..;
- Se recomandă în medii OLTP active
- Valorile duplicate sunt înlocuite cu o referință la tabela de simboluri

Folosirea Compression Advisor

- Compression Advisor analizează obiectele BD și determină ratele de compresie estimate
- Ajută administratorul BD la determinarea nivelelor potrivite de compresie
- Din Enterprise Manager determină compresia OLTP

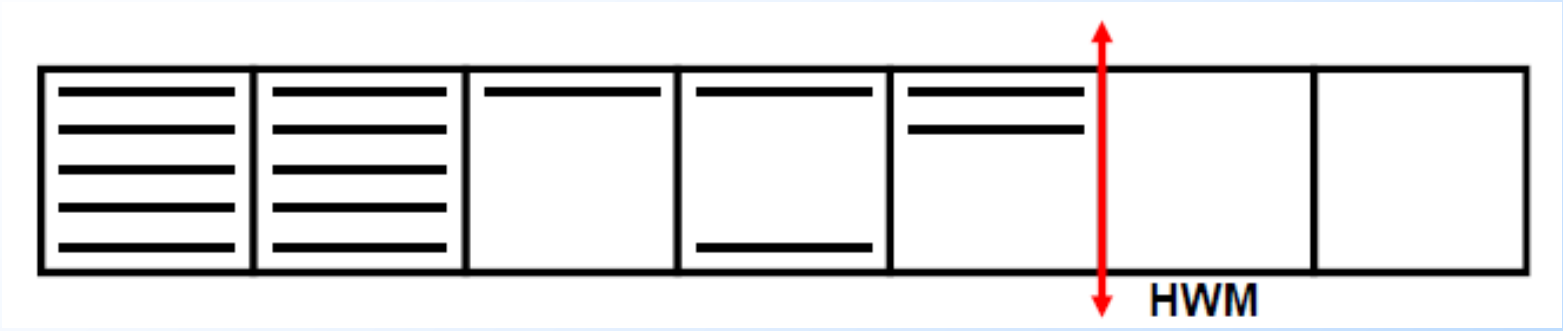
Folosirea Pragurilor

- Se pot fixa praguri fie pentru cât de ocupat este un tablespace, fie pentru spațiul rămas liber, exprimat în procente
  - Critical
  - Warning
- DBMS_SERVER_ALERT este package-ul folosit pentru set și get valori praguri

Rezolvare probleme de spațiu

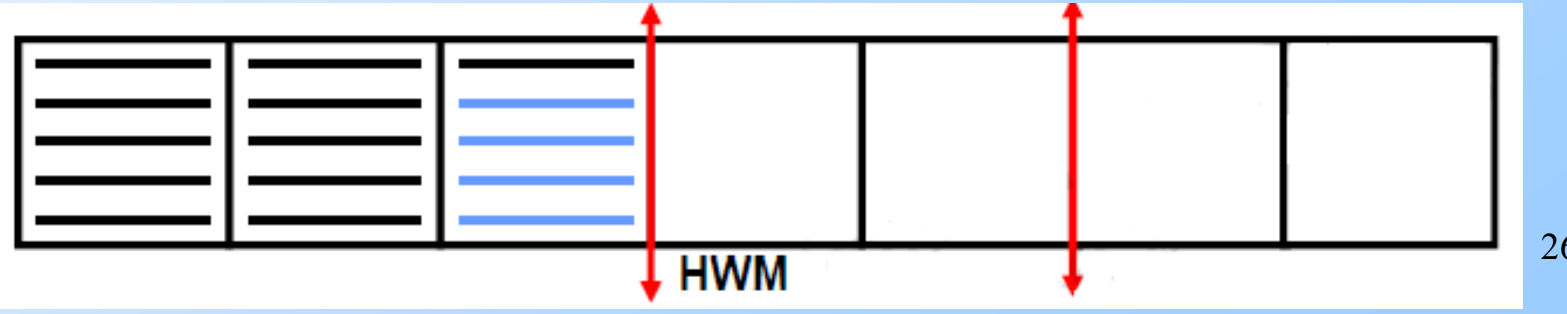
- Se adaugă fișier, sau se redimensionează fișier existent
- Se fixează AUTOEXTEND ON (atenție la spațiu liber pe disc))
- Se efectuează „shrink” la obiecte dispersate
- Se reduce UNDO_RETENTION
- Se verifică interogările cu durată lungă, ce folosesc tablespace-uri temporare

Efectuare „shrink”



ALTER TABLE employees SHRINK SPACE COMPACT;

ALTER TABLE employees SHRINK SPACE;



Efectuare „shrink”

- În prima fază se deplasează rândurile spre stânga (fără a afecta prin „lock” operații DML)
- În faza a doua „Higher-Water Mark” (HVM) este ajustat și spațiul nefolosit este eliberat
- Clauza COMPACT este folositoare la interogări de lungă durată, pentru a nu fi afectate
  - Progresul operației „shrink” este ținut în blocurile bitmap ale segmentului
- Faza a doua se poate efectua fără COMPACT la ore când BD e mai puțin accesată

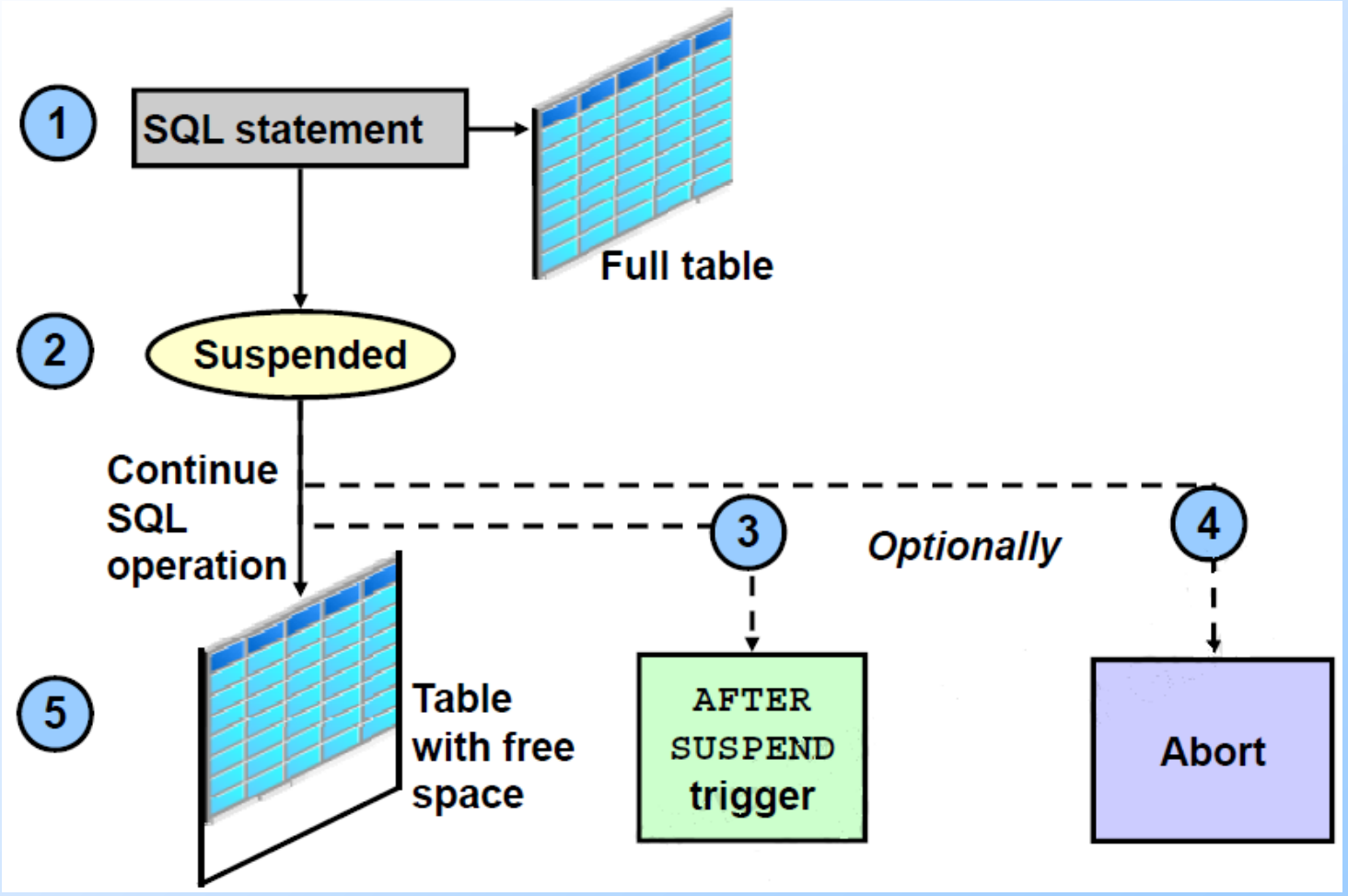
„Automatic Segment Advisor”

- Există un „job” planificat să ruleze în ferestrele:
  - Zile lucrătoare: 22:00 – 02:00
  - Sâmbăta și duminica: începe la 06:00 și durează 20 ore
- Examinează statistici BD, extrage mostre de date de segment și selectează următoarele obiecte pentru evaluare:
  - Tablespace-uri ce au depășit praguri critice
  - Segmente cu activitate intensă sau cu rată de creștere mare

Instrucțiune care poate fi reluată

- Permite oprirea de operații ce consumă mult spațiu înainte de a primi un mesaj de eroare
- Se poate corecta problema în timp ce instrucțiunea este suspendată
- Condiții de suspendare:
  - Lipsă spațiu
  - S-a atins numărul maxim de extent-uri
  - S-a depășit „quota” pentru spațiu (tablespace gestionat de dicționar)
- Poate fi suspendată și reluată de mai multe ori

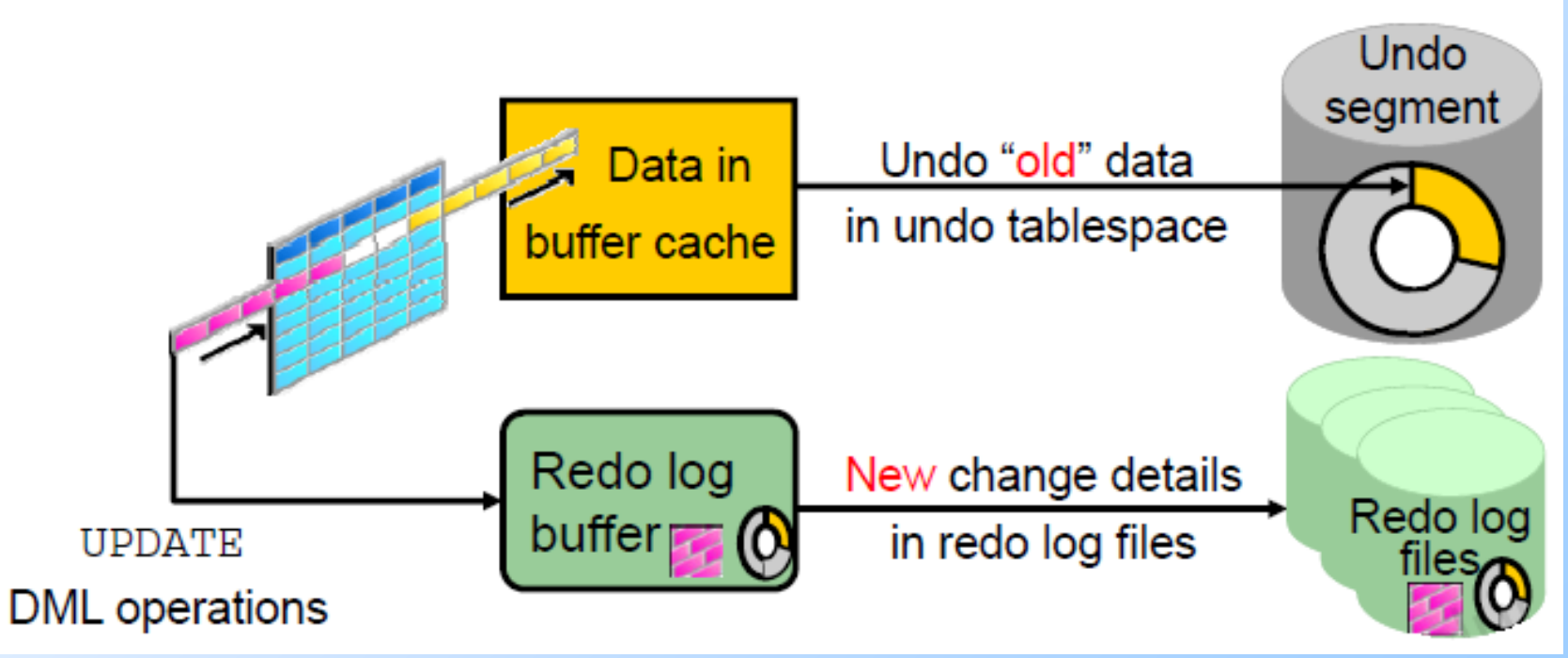
Reluarea instrucțiunilor suspendate



Undo data

- Este o înregistrare corespunzătoare acțiunii unei tranzacții
- Este capturată pentru fiecare tranzacție ce modifică date
- Se păstrează cel puțin până la încheierea tranzacției
- Oferă suport pentru:
  - Acțiuni rollback
  - Interogări consistente la read
  - Oracle Flashback Query, Oracle Flashback Transaction, Oracle Flashback Table
  - Recuperarea de tranzacții eșuate

Tranzacții și Undo Data



Tranzacții și Undo Data

- La începutul unei tranzacții îi este atribuit un segment undo
- Pe durata de viață a tranzacției, înainte de modificarea datelor, datele originale sunt copiate în segmentul undo
- Vederea V$TRANSACTION spune ce tranzacții sunt asignate la care segmente undo
- Operații paralele și DDL pot cauza folosirea mai multor segmente undo de către o tranzacție

Tranzacții și Undo Data

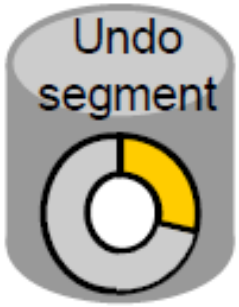
- Segmentele undo
  - Sunt create automat de serverul BD pentru a susține tranzacțiile
  - Sunt constituite din extent-uri care sunt la rândul lor constituite din blocuri de date
  - Acționează ca un buffer circular, crescând sau micșorându-se automat
- Tranzacția scrie într-un extent al segmentului undo până la terminarea tranzacției sau umplerea spațiului extent-ului, caz în care folosește spațiu din extent-ul următor; când toate extent-urile sunt ocupate se reia cu primul extent sau se solicită alocarea unui nou extent pentru segmentul undo

Tranzacții și Undo Data

- Segmentele undo există într-un tablespace specializat numit "undo tablespace"
- Pentru sistemele OLTP cu multe tranzacții concurente de scurtă durată, pot apărea conflicte pe headerul de fișier; soluția este folosirea mai multor fișiere pentru undo tablespace
- Deși o BD poate avea mai multe undo tablespace-uri unul singur poate fi folosit în mod curent pentru oricare instanță a BD
- Fiecare segment undo are cel puțin 2 extent-uri, numărul maxim depinde de dimensiunea blocului (la bloc de 8 kB sunt 32765 blocuri)

Comparație între Undo Data și Redo Data

	Undo	Redo
Record of	How to undo a change	How to reproduce a change
Used for	Rollback, read consistency, flashback	Rolling forward database changes
Stored in	Undo segments	Redo log files



Backup și Recovery

- Administratorul BD este responsabil să asigure că BD este disponibilă utilizatorilor:
  - Anticipează și acționează pentru a evita cauzele principale ce cauzează eșuări
  - Încearcă să crească timpul mediu între eșuări (MTBF)
  - Protejează componentele critice prin redundanță
    - Real Application Clusters (RAC)
    - Oracle Data Guard
  - Încearcă să reducă timpul mediu de recuperare (MTTR)
  - Încearcă să minimizeze pierderile de date
    - Fișiere jurnal (Archive log)
    - Tehnologia flashback
    - Standby DBs și Oracle Data Guard

Tipuri de eșuări

- Eșuare instrucțiune
- Eșuare proces utilizator
- Eșuare rețea
- Eroare utilizator
- Eșuare instanță
- Eșuare mediu de stocare

Eșuare instrucțiune

Probleme tipice	Soluții posibile
Încercare de a introduce date invalide într-o tabelă	Se lucrează cu utilizatorii pentru a valida și corecta datele
Încercare de a efectua operații cu privilegii insuficiente	Se oferă privilegiile potrivite la nivel de sistem sau obiect
Încercare de a aloca spațiu, ce eșuează	• Permite alocare de spațiu pentru reluare instrucțiune • Mărește quota ownerului • Adaugă spațiu la tablespace
Erori logice în aplicații	Se lucrează cu dezvoltatorii pentru a corecta erori de program

Eșuare proces utilizator

Probleme tipice	Soluții posibile
Utilizatorul efectuează o deconectare anormală	Nu este nevoie de acțiunea administratorului BD la rezolvarea de eșuare proces utilizator. Există procese background la nivel de instanță care efectuează rollback la modificările uncommitted și eliberează lock-urile.
Sesiunea unui utilizator se termină anormal	
Utilizatorul întâlnește o eroare în program ce termină sesiunea	

Eșuare rețea

Probleme tipice	Soluții posibile
Eșuare Listener	Se configurează Listener de backup și connect-time failover
Eșuare Network Interface Card (NIC)	Se configurează mai multe plăci de rețea
Eșuare conexiune la rețea	Se configurează o conexiune rețea de rezervă

Eroare utilizator

Probleme tipice	Soluții posibile
Utilizatorul modifică (UPDATE) sau șterge (DELETE) date din greșeală	Se efectuează rollback la tranzacție și la tranzacțiile dependente sau se reface tabela la starea inițială
Utilizatorul efectuează DROP TABLE	Se recuperează tabela din recycle bin. Se recuperează tabela dintr-un backup.

Eroare utilizator

- Se poate utiliza Oracle LogMiner pentru a interoga online redo logs folosind Enterprise Manager sau interfața SQL.
- Datele despre tranzacții sunt persistente în online redo logs mai mult decât în segmentele undo, dacă a fost configurată arhivarea informației redo, atunci există până ce se șterg fișierele arhivă.
- Dacă s- șters din greșeală o tabelă cu DROP, atunci cu tehnologia flashback se poate încerca recuperarea din recycle bin.

Tehnologia flash back

Pentru analiza erorilor
Oracle Flashback Query
Oracle Flashback Versions Query
Oracle Flashback Transaction Query

Pentru recuperare în caz de eroare
Oracle Flashback Transaction Backout
Oracle Flashback Table
Oracle Flashback Drop
Oracle Flashback Database

Tehnologia flash back

- Pentru analiza erorilor:
  - Flashback Query permite a se vizualiza date salvate așa cum existau ele la un moment dat în trecut: SELECT cu clauza AS OF și o marcă de timp sau System Change Number (SCN)
  - Flashback Version Query permite a se vizualiza date istorice pentru un interval de timp specific: SELECT cu clauza VERSIONS BETWEEN
  - Flashback Transaction Query permite a se vizualiza toate modificările efectuate bazei de date la nivel de tranzacție

Tehnologia flash back

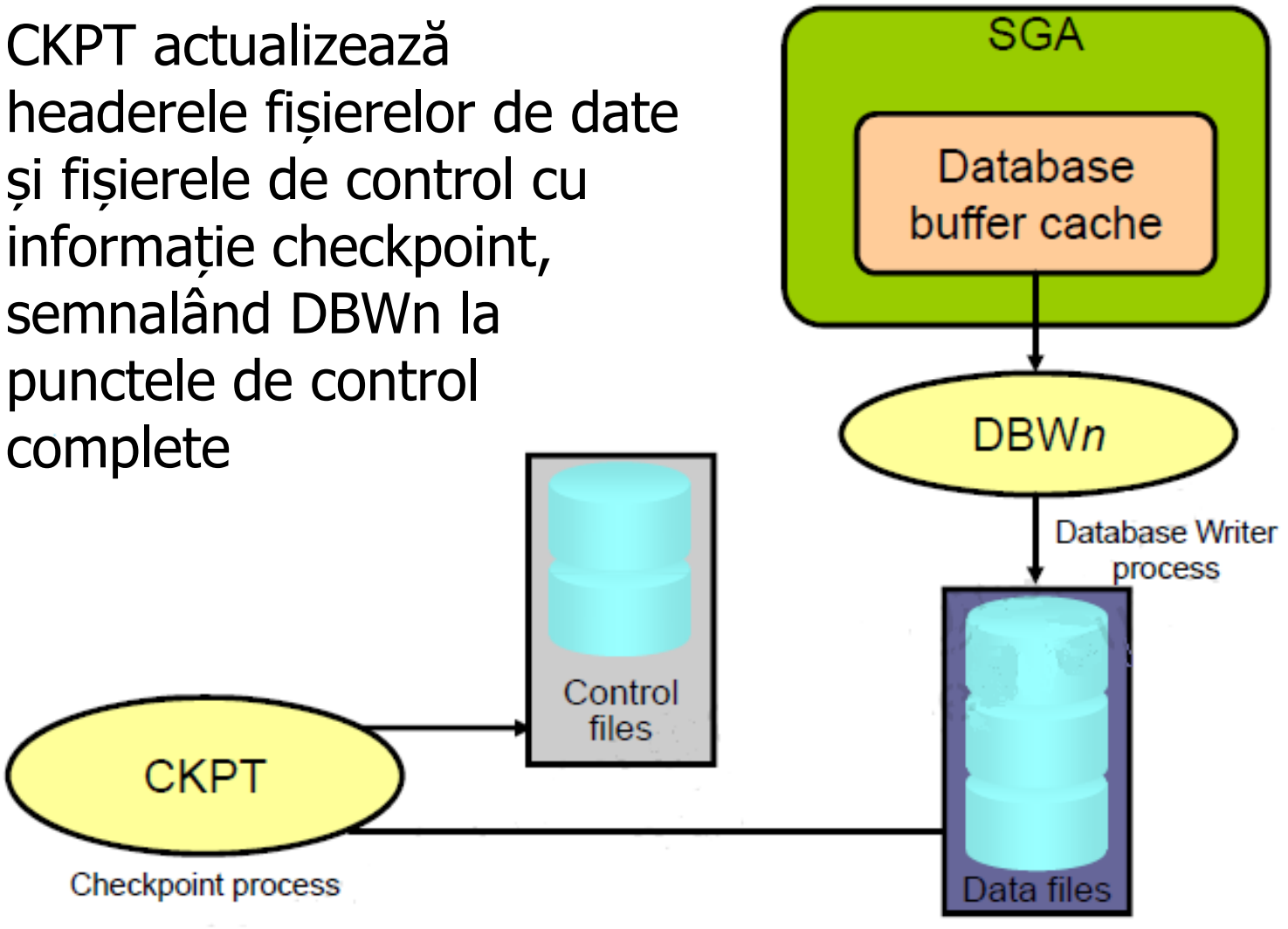
- Pentru recuperare în caz de eroare:
  - Flashback Transaction Backout efectuează rollback la o tranzacție specifică și de asemenea la tranzacțiile dependente
  - Flashback Table reface una sau mai multe tabele la conținutul lor la un moment de timp anterior, fără a afecta alte obiecte
  - Flashback Drop reface efectele unei operații DROP TABLE preluând tabela din recycle bin împreună cu obiectele dependente (indecși și trigere)
  - Flashback Database reface baza de date la un moment dat sau SCN

Eșuare instanță

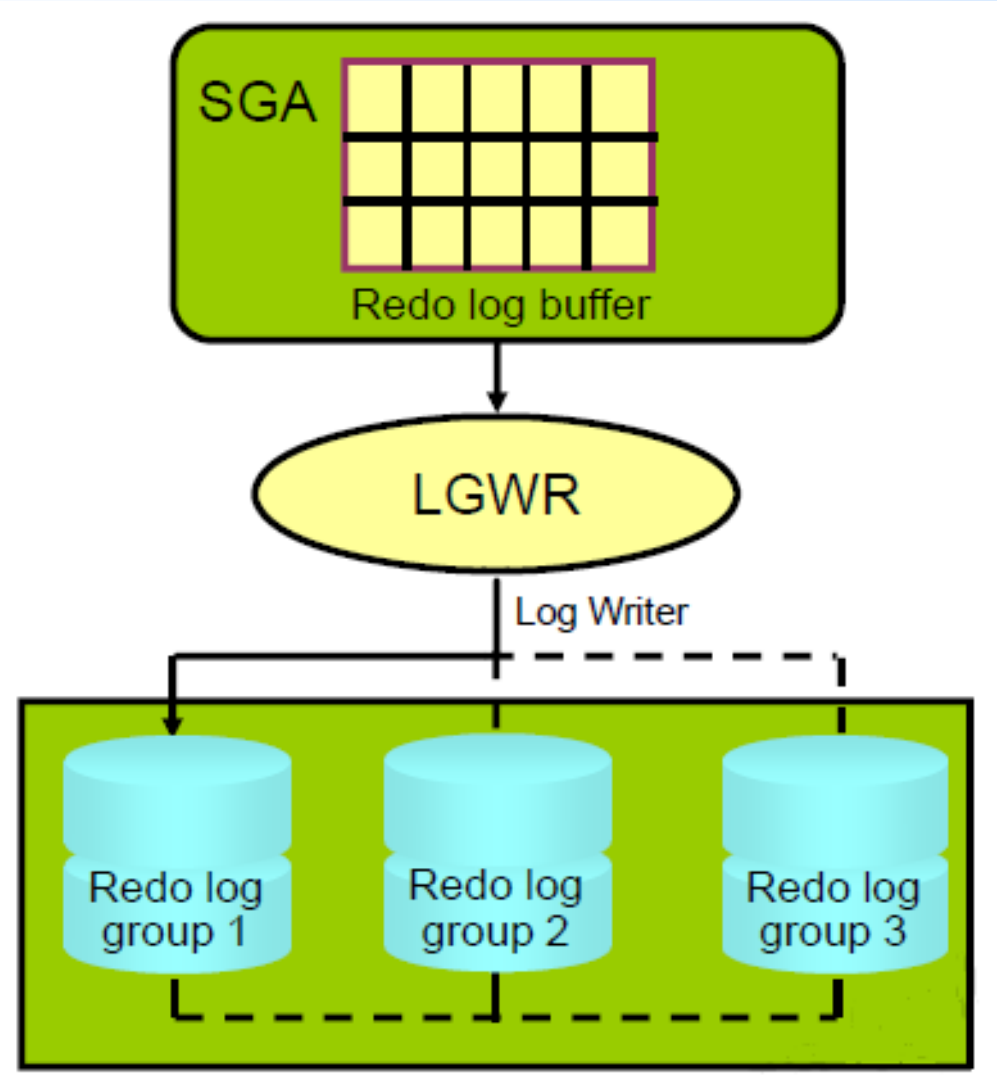
Probleme tipice	Soluții posibile
Întrerupere de energie electrică	Se repornește instanța cu STARTUP. Recuperarea din eșuarea instanței este un proces automat, ce include rularea înainte a modificărilor utilizând redo logs și la sfârșit rularea înapoi a tranzacțiilor ce nu au fost încheiate cu succes (prin COMMIT). Se investighează cauzele eșecului prin consultarea alert log, fișiere trace și Enterprise Manager.
Eșuare hardware	
Eșuare a unui proces background critic	
Proceduri shutdown de urgență	

Procesul Checkpoint (CKPT)

CKPT actualizează headerele fișierelor de date și fișierele de control cu informație checkpoint, semnalând DBWn la punctele de control complete



Fișiere Redo Log și Log Writer

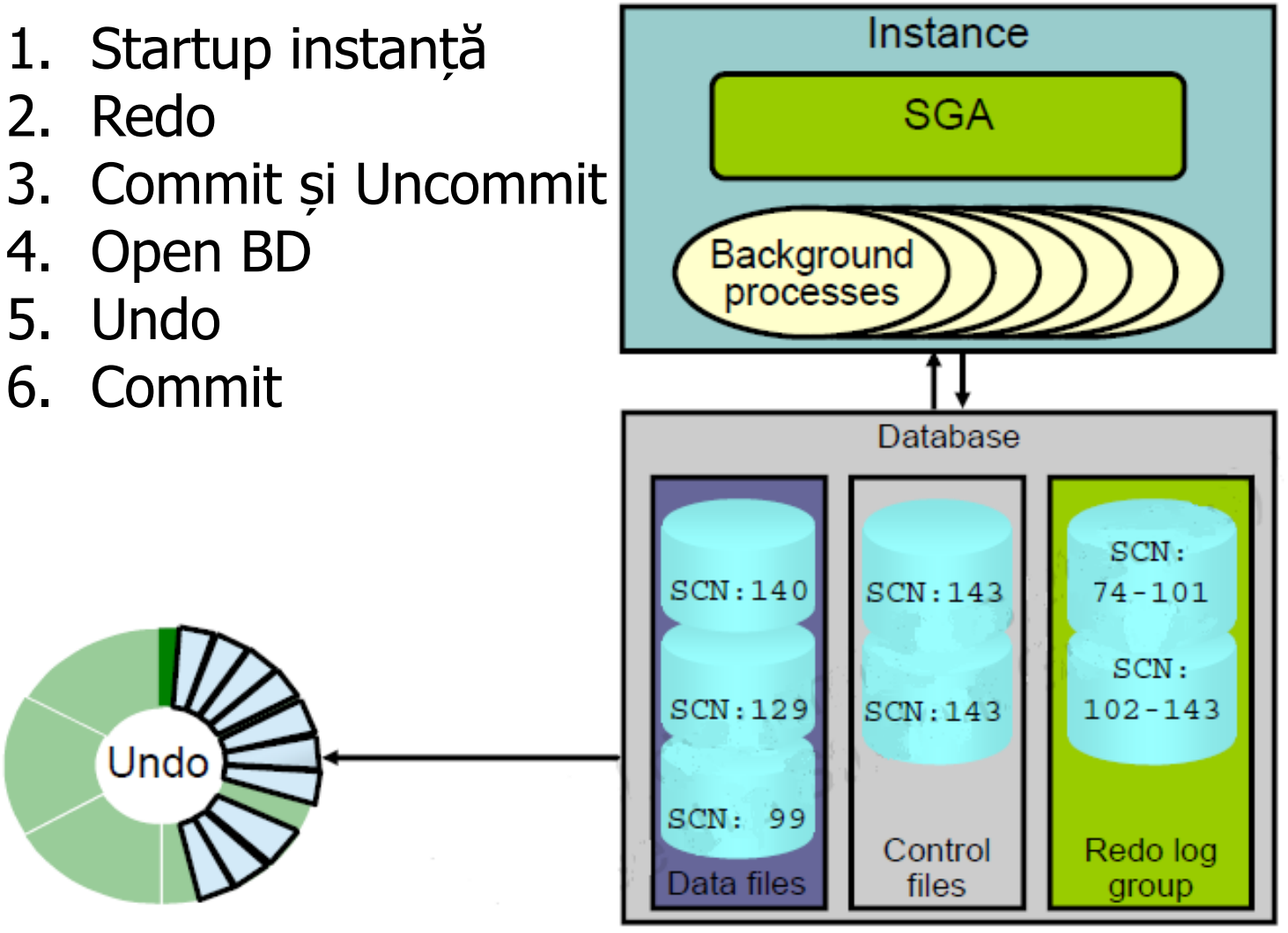


Fișiere Redo Log și Log Writer

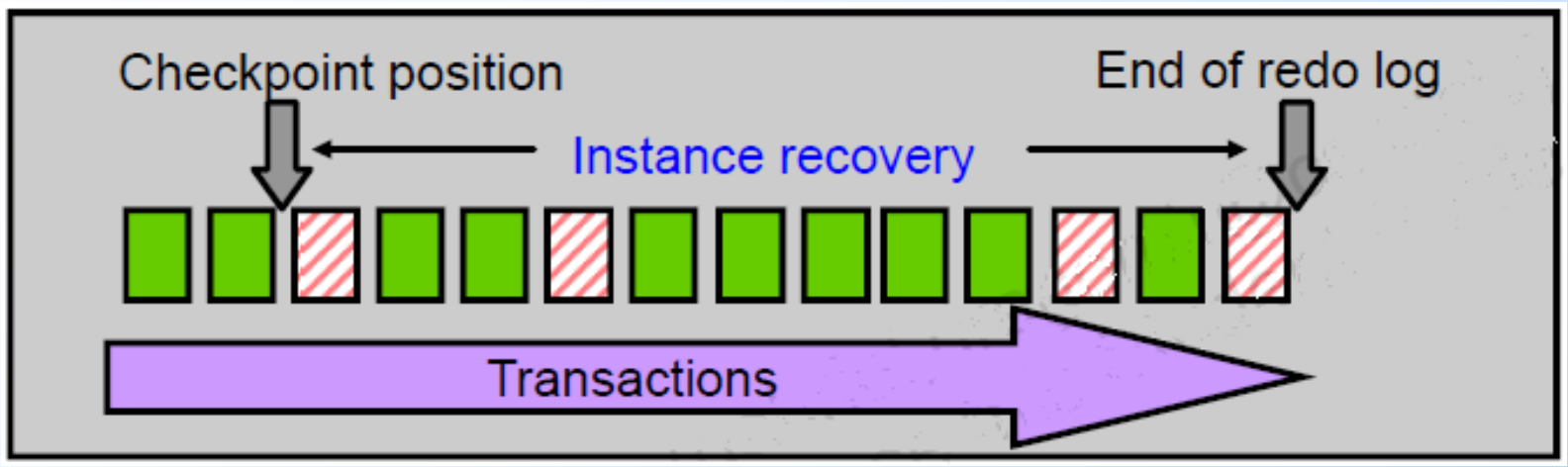
- Fișierele redo log
  - Înregistrează modificările efectuate bazei de date
  - Ar trebui să fie multiplexate pentru a proteja împotriva pierderilor de informație
- Log Writer (LGWR) scrie:
  - La COMMIT
  - Când se umple 1/3 (redo log buffer)
  - La fiecare 3 secunde
  - Înainte să scrie DBWn
  - Înainte de shutdown normal

Faze ale recuperării instanței

1. Startup instanță
2. Redo
3. Commit și Uncommit
4. Open BD
5. Undo
6. Commit



Reglarea recuperării instanței

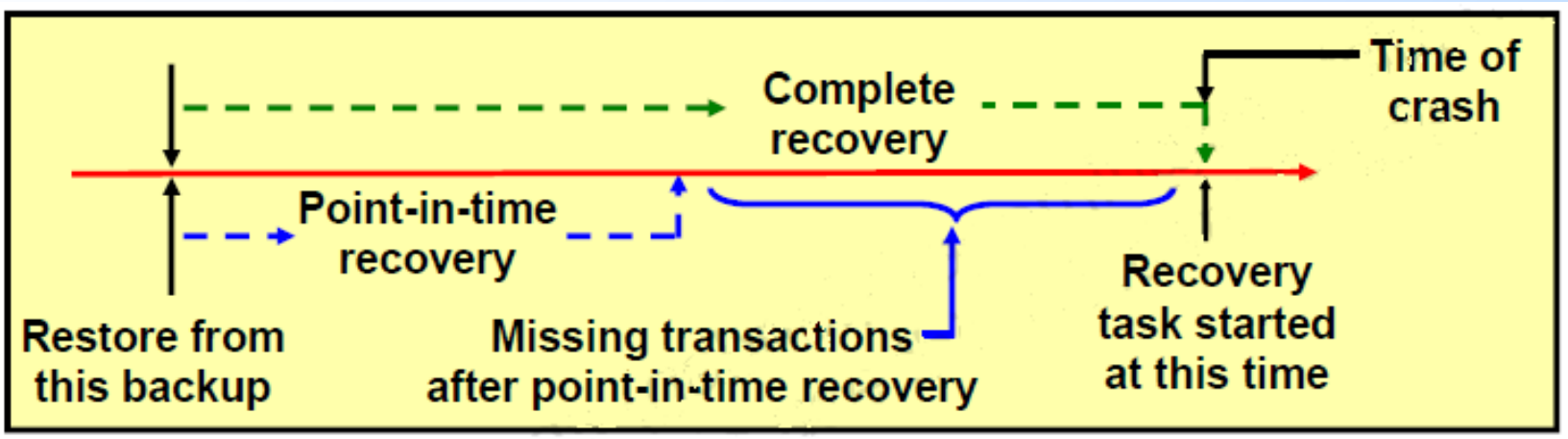


- În timpul recuperării instanței, tranzacțiile între poziția checkpoint și sfârșitul redo log trebuie aplicate fișierelor de date
- Reglarea se face prin controlul diferenței între poziția checkpoint și sfârșitul redo log

Eșuare mediu de stocare

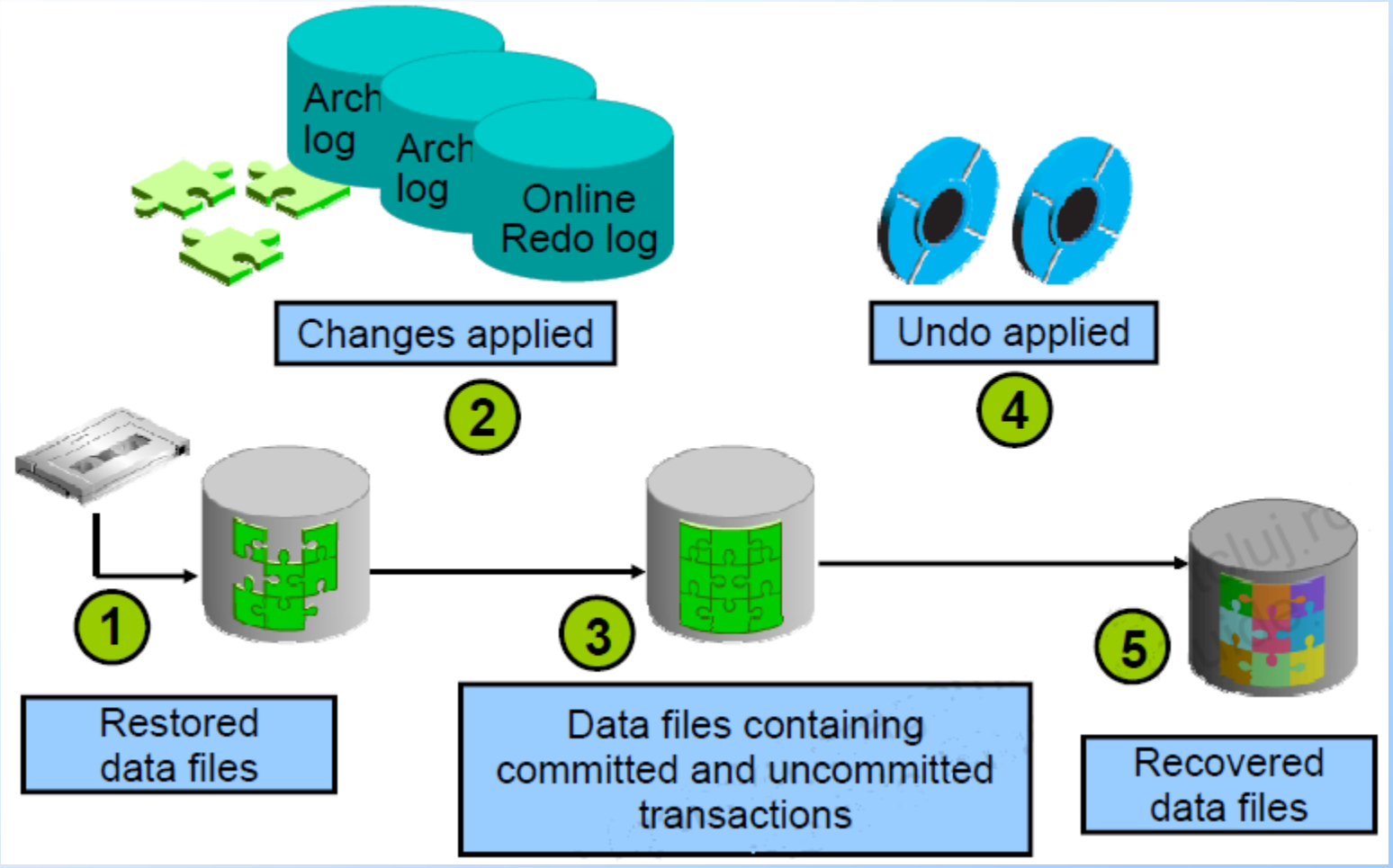
Probleme tipice	Soluții posibile
Eșuare disk drive	1. Se restaurează fișierul afectat din backup 2. Se informează baza de date cu privire la noua locație a fișierului (dacă este cazul) 3. Se recuperează fișierul prin aplicarea de informație redo (dacă este cazul)
Eșuare disk controller	
Ștergerea sau coruperea unui fișier de care are nevoie o operație pe baza de date	

Comparație între recuperare completă și incompletă



- Recuperare completă: aduce BD sau tablespace la zi incluzând toate modificările de date commit efectuate până la momentul de timp solicitat
- Recuperare incompletă (PITR): aduce BD sau tablespace la un moment de timp specificat, în trecut (point in time recovery)

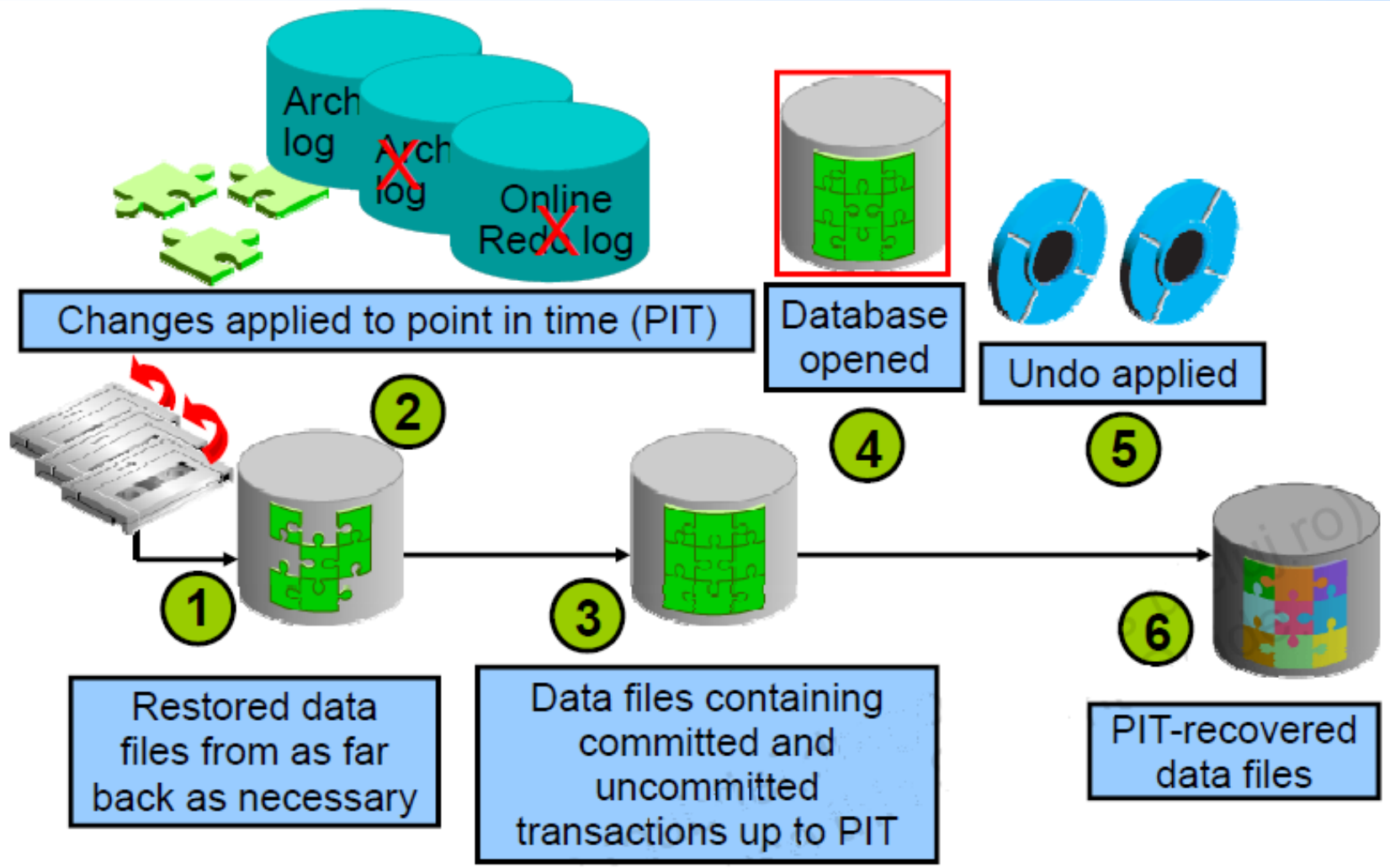
Procesul de recuperare completă



Procesul de recuperare completă

1. Fișierele stricate sau lipsă sunt restaurate din backup
2. Sunt aplicate modificările din backup-uri incrementale, fișiere redo log arhivate și fișiere redo log online
3. Fișierele de date restaurate vor conține acum modificări commited și uncommited
4. Blocurile undo sunt folosite pentru rollback a modificărilor uncommited (transaction recovery)
5. Fișierele de date sunt recuperate și consistente

Procesul de recuperare Point-in-Time



Procesul de recuperare Point-in-Time

1. Se restaurează fișierele de date din backup (fie prin copiere fizică a fișierelor la nivel SO, fie cu comanda RMAN RESTORE)
2. Se folosește comanda RECOVERER – se aplică redo din fișiere redo log arhivate
3. Stare pe parcursul recuperării: fișierele de date conțin unele tranzacții încheiate cu succes și altele nefinalizate
4. Se folosește comanda ALTER DATABASE OPEN – BD se deschide înainte de aplicare undo pentru disponibilitate maximă
5. Se aplică undo data (pentru tranzacții nefinalizate)
6. Procesul se încheie

Configurare pentru recuperare

- Se programează backupuri frecvente
- Se multiplexează fișierele de control
- Se multiplexează grupuri redo log
- Se păstrează copii arhivate ale jurnalelor redo

Configurare Fast Recovery Area

- Este spațiul pe disc unde se păstrează arhivele jurnalelor, backup-uri, jurnale flashback, fișiere de control multiplexate, jurnale redo multiplexate
- Se recomandă să fie pe alt disc față de fișierele BD și fișierele jurnal și de control primare
- Dimensiunea spațiului ar trebui să fie dublu față de dimensiunea BD

Configurare Fast Recovery Area

- Spațiul este gestionat prin politici de retenție
- Serverul BD Oracle gestionează automat spațiul luând decizia când să șteargă fișiere backup de care nu mai este nevoie
- Spațiul este monitorizat continuu cu Enterprise Manager pentru a preîntâmpina umplerea sa
- Periodic se copiază fișierele pe bandă magnetică

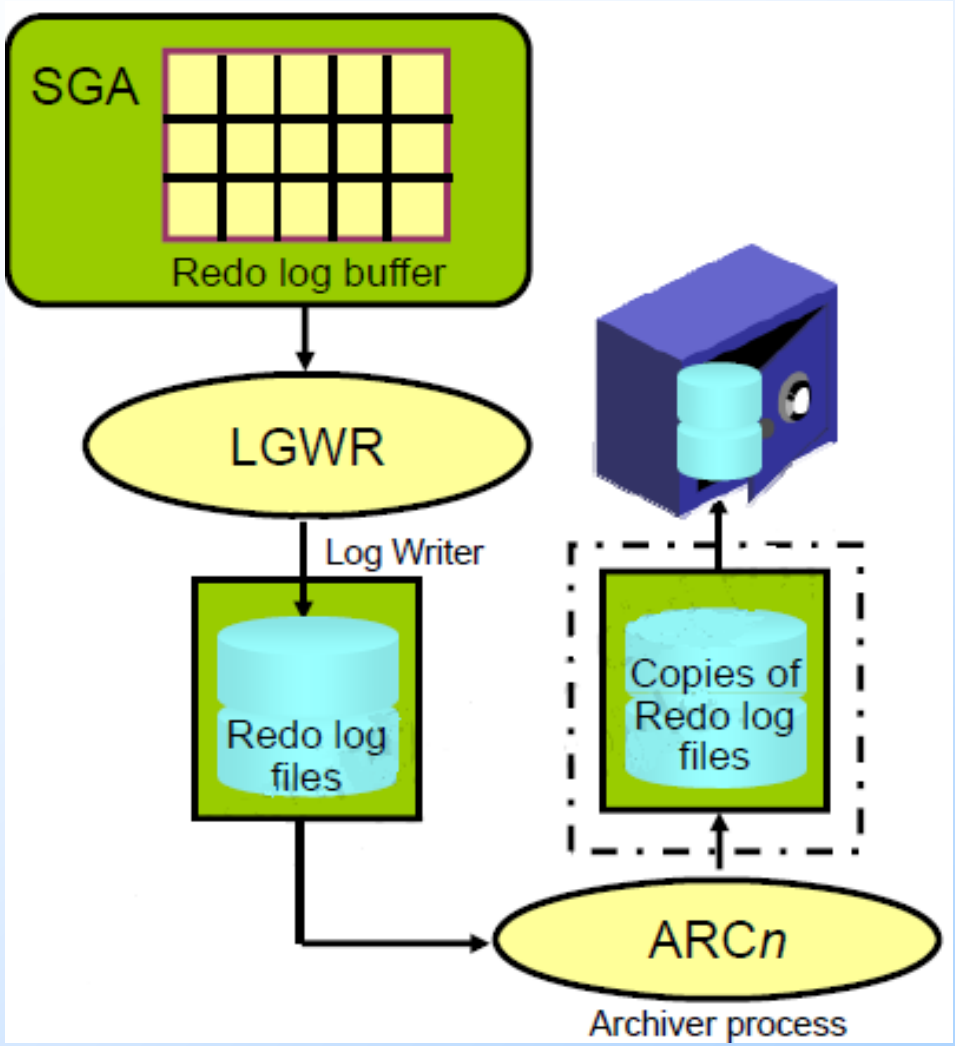
Multiplexarea fișierelor de control

	Stocare ASM	Stocare sistem fișier
Best practice	O copie pe fiecare grup de disc (cum ar fi +DATA și +FRA)	Cel puțin două copii, fiecare pe alt disc (cel puțin unul pe controler de disc separat)
Pași pentru a crea fișiere de control suplimentare	Nu sunt necesare copii suplimentare ale fișierrului de control	1. Se modifică SPFILE cu comanda ALTER SYSTEM SET fișiere_control 2. Se oprește BD 3. Se copiază fișierul de control în altă parte 4. Se pornește BD și se verifică adăugarea noului fișier de control

Fișiere jurnal redo

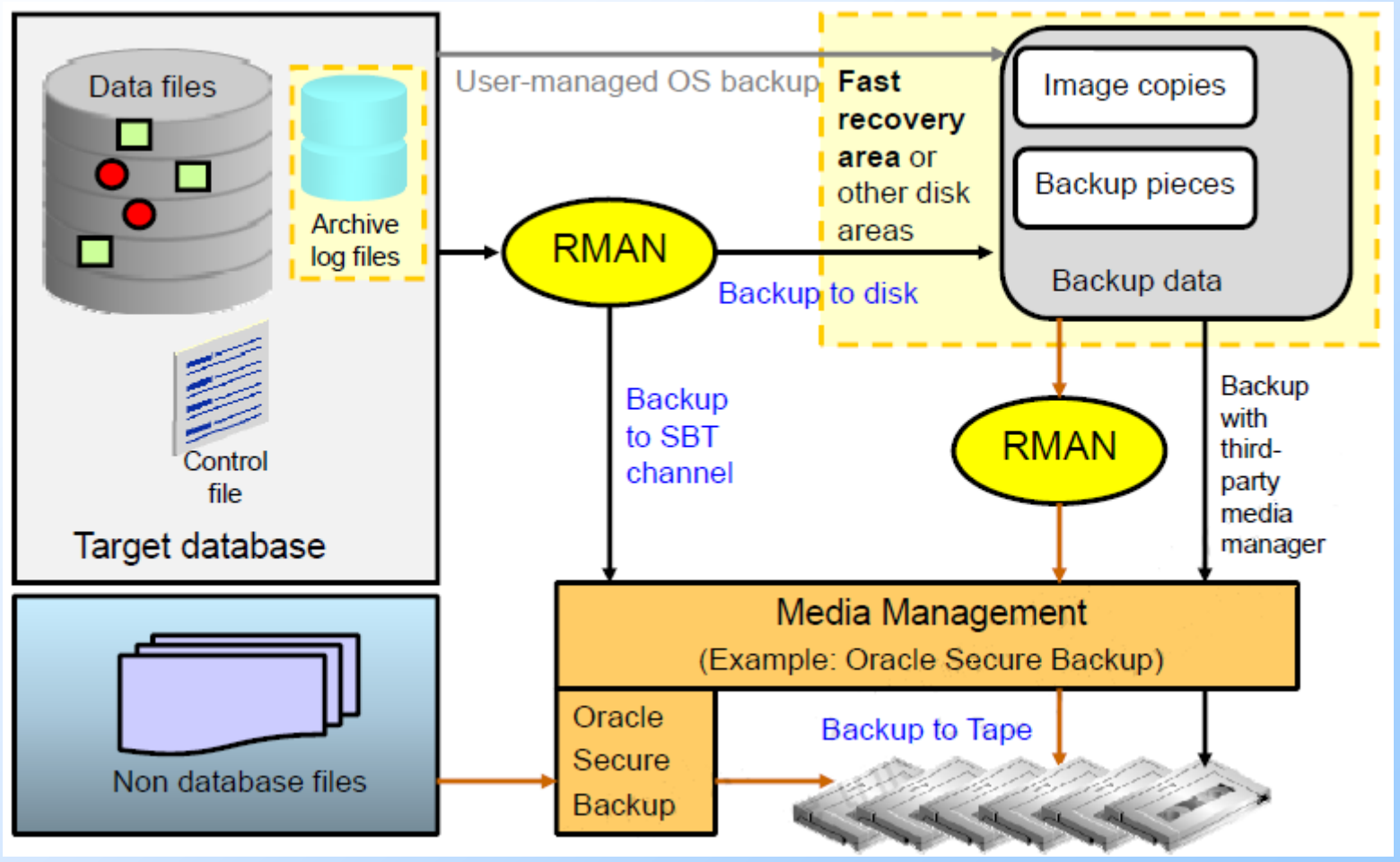
- Oracle recomandă:
  - Fiecare grup să aibă minim două fișiere
  - Fiecare fișier să fie pe un disc separat când se utilizează sistem fișier sau pe un grup de discuri separat când se folosește ASM (+DATA, +FRA)
- Atenție la performanță, deoarece un commit se finalizează doar după ce informația despre tranzacție este scrisă în toate fișierele jurnal

Procesul Archiver (ARCn)



- Este un proces background opțional
- Arhivează fișierele jurnal redo online când BD este în mod ARCHIVELOG
- Conservă modificările asupra BD

Soluții Backup



Oracle Secure Backup

- Oracle Secure Backup și RMAN oferă o soluție end-to-end:
  - Gestiunea centralizată backup pe bandă magnetică
  - Administrare media pentru RMAN
  - Backup date în rețea – servere, clienți, Network Attached Storage (NAS) și alte dispozitive de stocareOracle Secure Backup

Backup manual

- Folosește scripturi scrise de utilizator
- Necesită ca fișierele BD să fie în starea potrivită pentru backup
- Se bazează pe comenzi sistem de operare pentru backup fișiere

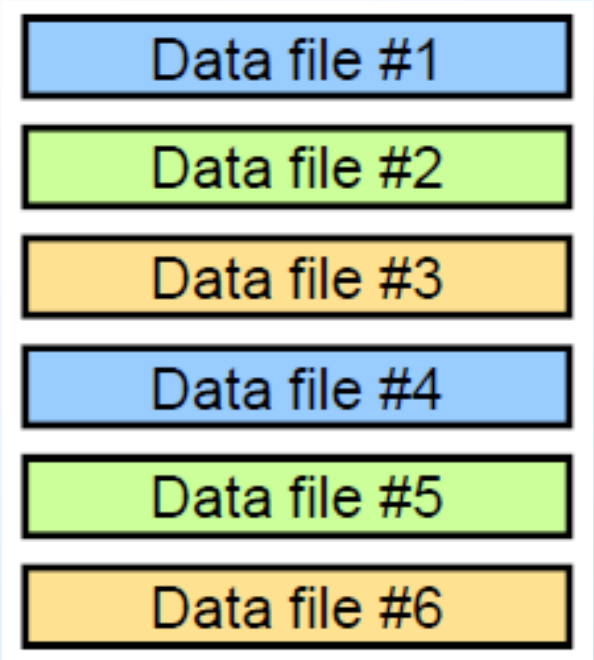
Acțiuni ale scriptului de backup

- Se interoghează V$DATAFILE pentru a determina ce fișiere cu date se salvează
- Se interoghează V$LOGFILE pentru a identifica fișierele online redo log
- Se interoghează V$CONTROLFILE pentru a identifica fișierele de control
- Se pune fiecare tablespace în modul online backup
- Se interoghează V$BACKUP pentru determina ce fișiere sunt atașate unui tablespace aflat în mod online backup
- Se emit comenzi SO de copiere fișiere la locația backup
- Se scoate fiecare tablespace din modul online backup

Terminologie backup

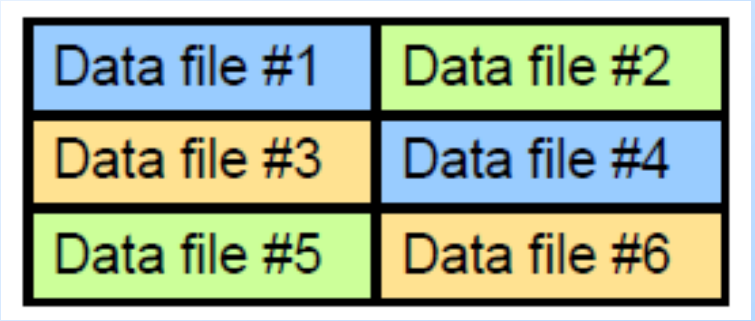
- Strategie backup:
  - Întreagă – toată BD
  - Parțială – porțiune din BD
- Tip backup:
  - Complet – toate blocurile de date
  - Incremental – doar informația modificată de la un backup precedent
    - Cumulativ – modificări de la început
    - Diferențial – modificări de la incrementul precedent
- Mod backup:
  - Offline – consistent
  - Online - inconsistent

Tipuri de backup



Copii imagine

(duplicate ale fișierelor de date și fișierelor jurnal la nivel SO)



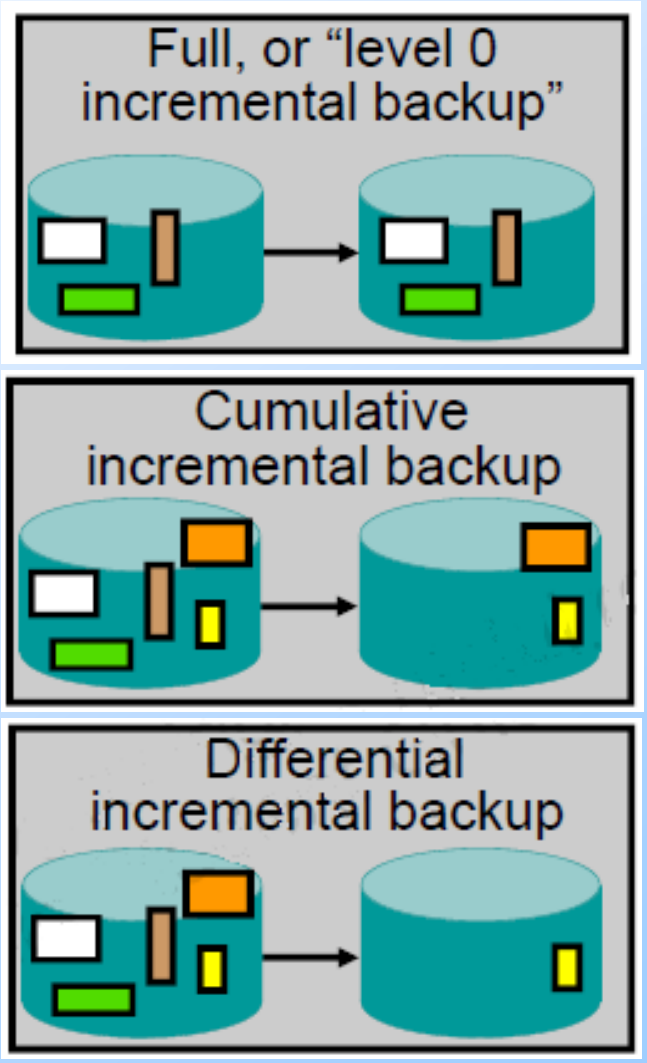
Set backup

(fișiere binare comprimate în format proprietar Oracle)

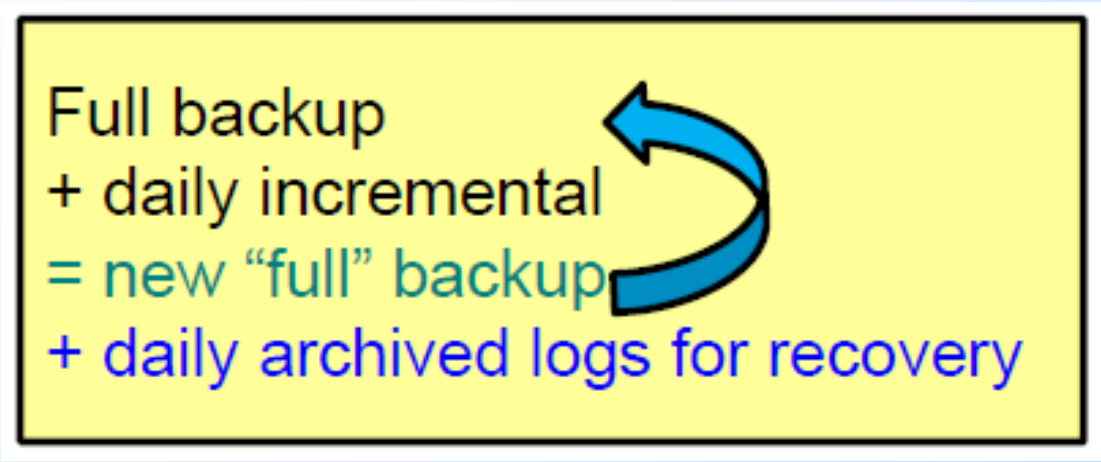
Blocurile goale nu sunt salvate, de obicei 20% din blocuri sunt goale → mai avantajos

Tipuri backup RMAN

- Full backup conține toate blocurile de date, este chivalent cu Incremental backup level 0
- Backup incremental cumulativ level 1 conține doar blocurile modificate de la precedentul backup incremental level 0
- Backup incremental diferențial level 1 conține doar blocurile modificate de la ultimul backup incremental



Recomandări Oracle



- Strategia Oracle pentru backup folosește backup incremental și actualizare incrementală
- Atunci când BD este în modul ARCHIVELOG poate fi restaurată la ultima tranzacție finalizată cu succes

```
RMAN>BACKUP DATABASE PLUS ARCHIVELOG;
```

Pornirea BD

1. Starea NOMOUNT: instanța citește fișierul cu parametri de inițializare
2. Starea MOUNT: instanța verifică dacă toate fișierele de control menționate în fișierul cu parametri de inițializare sunt prezente și sincronizate, dacă un fișier lipsește sau este corupt instanța rămâne în starea NOMOUNT
3. Starea OPEN:
   - Instanța verifică dacă grupurile de fișiere redo log au cel puțin un membru prezent (membri lipsă sunt notați în alert log)
   - Instanța verifică dacă toate fișierele de date sunt prezente (fișierele offline nu sunt verificate) dacă nu există vreun fișier instanța rămâne în starea MOUNT

Pornirea BD

(continuare)

- Instanța verifică dacă toate fișierele de date (care nu sunt offline sau read-only) sunt sincronizate cu fișierul de control, dacă este necesar este executată automat recuperarea, dacă nu se poate recupera din jurnalele redo online este necesară media recovery (de către administratorul BD) și instanța rămâne în starea MOUNT (v$recoverer_file oferă informații asupra fișierelor ce necesită atenție)

- După ce BD este OPEN, instanța poate eșua dacă se pierde:
  - Un fișier de control
  - Un fișier ce aparține tablespace-urilor SYSTEM sau UNDO
  - Un întreg grup redo log (dacă cel puțin un fișier redo log este disponibil, instanța rămâne OPEN)