

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, apartinand unui fisier de date (fisierul in care se gaseste extensia).
- ❑ In documentatia Oracle se precizeaza:
 - ❑ 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)

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 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 BLE	YES	160	ONLINE
C:\ORACLE11G_DB\ORADATA\BD\UNDOTBS1.DBF	UNDOTBS1	AVAILA BLE	YES	640	ONLINE
C:\ORACLE11G_DB\ORADATA\BD\SYSAUX01.DBF	SYSAUX	AVAILA BLE	YES	1280	ONLINE
C:\ORACLE11G_DB\ORADATA\BD\SYSTEM01.DBF	SYSTEM	AVAILA BLE	YES	1280	SYSTEM
C:\ORACLE11G_DB\ORADATA\BD\EXAMPLE01.DBF	EXAMPLE	AVAILA BLE	YES	80	ONLINE
C:\ORACLE11G_DB\ORADATA\BD\BD_DATA.DBF	BD_DATA	AVAILA BLE	NO	0	ONLINE
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_IDX	CONSTANT_GROW_INDEXES	AVAILA BLE	NO	0	ONLINE
C:\ORACLE11G_DB\PRODUCT\11.2.0\DBHOME_1\DATABASE\REPOS_GR_TAB	CONSTANT_GROW_TABLES	AVAILA 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> 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>

Alte vederi

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

Tablespaces	ts\$ dba_tablespaces user_tablespaces
Tablespace Quotas	tsq\$ dba_ts_quotas user_ts_quotas
Data Files	dba_data_files v_\$backup_datafile v_\$datafile v_\$datafile_copy v_\$datafile_header
Free Space	dba_free_space
Segments	dba_segments 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 dba_undo_extents v_\$rollstat v_\$undostat
Transportable Tablespaces	transport_set_violations
Dictionary Management	fet\$ 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

3. Oracle Concepts - Tablespaces

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

Sfârșitul capitolului 4