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 ⊃ Segment ⊃ Extensie (extent) ⊃ Bloc (stocate in fisierele 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 (unul 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 $\leftrightarrow$ offline si
  - read-write $\leftrightarrow$ read-only

Fisiere de date

- Aceste fisiere se creaza:
  - la crearea bazei de date (pentru tablespace-urile care sunt create atunci)
  - la crearea unui nou tablespace
  - La adaugarea unui nou fisier de date la un tablespace
- Dimensiunea fisierelor este specificata la creare.
- Unui fisier de date existent i se poate modifica dimensiunea ulterior.
- Unui fisier de date i se poate seta optiunea 'AUTOEXTEND' pentru a creste automat ca dimensiune cand este necesar.

SEGMENTE

- Un tablespace contine segmente.
- Un segment este un ‘container’ pentru un obiect (tabela, index, etc)
- Sunt de 4 tipuri generice (si 11 tipuri efective):
  - Segment de tip tabela (Table segment)
  - Segment de tip index (Index segment)
  - Segment temporar (Temporary segment)
  - Segment de rollback (Rollback segment)
- Un segment se poate intinde pe mai multe fisiere de date care apartin aceluiasi tablespace.

SEGMENTE - cont

- Segmentele temporare sunt in general cele folosite pentru sortari.
- Urmatoarele cereri SQL au nevoie de segment temporar in cazul in care nu pot fi efectuate in memorie:
  - **`create index`**
  - **`select order by`**
  - **`select distinct`**
  - **`select group by`**
  - **`select union`**
  - **`select intersect`**
  - **`select minus by`**
  - **`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 de catre procesul de background SMON

EXTENSII

- Un segment este format din unul sau mai multe extensii (eng.: extent).
- O extensie e formata dintr-o succesiune ***contigua*** de blocuri pe disc.
- 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)

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 penstru sistemul de gestiune de baze de date.

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
  - Temporare – contin segmente temporare – am vorbit despre ele
  - De tip Undo – introduse incepand cu versiunea 9i – contin segmente de undo, necesare in cazul revocarii operatiilor de actualizare.

Alta clasificare: DMT si LMT

- Fiecare Tablespace este format dintr-o multime de extensii.
- 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
- In cazul in care gestiunea se face prin intermediul dictionarului de date (o solutie costisitoare ca timp) se numesc DMT - dictionary managed tablespaces

DMT si LMT - cont

- In cazul in care gestiunea se face local, prin stocarea datelor privind starea extensiilor in interiorul tablespace-ului (in headerul sau) atunci se numesc LMT - locally managed tablespaces
- In acest caz headerul unui tablespace contine un bitmap unde fiecare bit este un bloc sau un grup de blocuri. Bitul arata daca zona respectiva este ocupata sau nu.

Exemplu

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

TABLESPACE_NAME    EXTENT_MAN ALLOCATIO
------------------ ---------- ---------
SYSTEM             DICTIONARY USER
TEMP               LOCAL      UNIFORM
```

Sintaxa CREATE TABLESPACE

```
CREATE [UNDO] TABLESPACE tablespace_name
DATAFILE Datafile_Options
Storage_Options ;
```

Unde *Datafile_Options* este format dintr-o lista cu elemente de tipul:
'filespec' [AUTOEXTEND OFF]
'filespec' [AUTOEXTEND ON [NEXT *int* K | M] [MAXSIZE *int* K | M]]

Iar Storage_Options:
DEFAULT [COMPRESS|NOCOMPRESS] STORAGE clauza_storage
MINIMUM EXTENT *int* {K|M}
BLOCKSIZE *int* K
LOGGING | NOLOGGING
FORCE LOGGING
ONLINE | OFFLINE {NORMAL | TEMPORARY | IMMEDIATE }
PERMANENT | TEMPORARY
EXTENT MANAGEMENT {DICTIONARY |
LOCAL {AUTOALLOCATE | UNIFORM [SIZE *int* K | M]} }
SEGMENT SPACE MANAGEMENT {MANUAL | AUTO}

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!
- 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 logate 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

- 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
  - tablespace-ul va contine extensii de dimensiuni diferite, gestiunea fiind facuta automat de catre sistem.

CREATE TABLESPACE - cont

- optiunea AUTOALLOCATE – cont.
  - 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 da date al BD este 16K sau mai mare atunci dimensiunea minima a unei extensii este de 1M.

CREATE TABLESPACE - cont

- optiunea UNIFORM
  - Specifica faptul ca acel tablespace este gestionat folosindu-se extensii de dimensiune fixa.
  - Valoarea implicita a dimensiunii este 1M
  - 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.

CREATE TABLESPACE - cont

- optiunea UNIFORM - cont
  - 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 dn acel tablespace.

CREATE TABLESPACE - cont

- Daca nu exista clauza EXTENT MANAGEMENT atunci pentru determinarea tipului 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 >= 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.

CREATE TABLESPACE - cont

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***

CREATE TABLESPACE - cont

Clauza Storage are urmatoarele optiuni:

- **INITIAL** ***int*** **K** | **M** – defineste dimensiunea primei extensii (minim 2 blocuri). Valoarea implicita este 5 blocuri ale BD.
- **NEXT** ***int*** **K** | **M** – da dimensiunea celui de-a doua extensii. Valoarea minima este de 1 bloc, valoarea implicita este de asemenea 5 blocuri.
- **MINEXTENTS** ***int*** - este numarul de extensii care sunt alocate cand segmentul este creat. Valoarea minima – si implicita – este 1.

CREATE TABLESPACE - cont

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.

CREATE TABLESPACE - cont

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

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

Deci daca NEXT = 200K iar PCTINCREASE este 50 atunci

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

CREATE TABLESPACE - cont

- 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

CREATE TABLESPACE - cont

- 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. 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.

Creare TABLESPACE

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;
```

Creare TABLESPACE - cont

- 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

ALTER TABLESPACE

- Sintaxa 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 OFF, NEXT si MAXSIZE sunt automat egale cu 0 si trebuiesc respecificate – daca se doreste – cu o alta comanda ALTER TABLESPACE AUTOEXTEND
- 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 si nu mai exista spatiu pentru acestea.
- Valoarea implicita pentru NEXT este de 1 bloc al BD
- MAXSIZE specifica dimensiunea maxima a spatiului care se poate aloca pentru acel fisier de date (pana la ce dimensiune poate creste).
- UNLIMITED specifica faptul ca nu exista 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 micsorare) 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 micsorarii 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:
- NORMAL – se executa checkpoint pentru toate fisierele de date din acel tablespace (aceste fisiere trebuie sa fie toate online – si ele 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
- Daca baza de date este in modul NOARCHIVELOG, NORMAL este optiunea care trebuie aleasa (pentru ca in acest caz nu se poate face recuperarea).

ONLINE / OFFLINE - cont

- 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.
- IMMEDIATE nu face checkpoint si nici nu verifica daca fisierele sunt disponibile sau nu. La revenirea online este nevoie de revovery.

Exemple

```
ALTER TABLESPACE user ONLINE

ALTER TABLESPACE user OFFLINE
```
(implicit e NORMAL)

```
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 ulterioeara 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.

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 150
col "%INC" for 9999
col minext for 9999
col blksz for 99999
select tablespace_name, logging, force_logging FLOG,
block_size blksz, status, contents, extent_management,
segment_space_management, allocation_type,
initial_extent/1024 init_ext_kb, next_extent/1024
next_ext_kb, pct_increase "%INC", min_extents minext,
max_extents/1024 max_ext_db, min_extlen
from DBA_TABLESPACES
order by 1;
```

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.

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

Sfârşitul capitolului 4