CUPRINS

10. Structura bazei de date

- Baza de date *Oracle*:
  - este o colecţie de date tratate unitar;
  - este creată pentru a asigura stocarea şi extragerea informaţiilor.
- Orice bază de date are o structură fizică şi o structură logică. Deoarece structura fizică este separată de cea logică, stocarea fizică a datelor poate fi administrată fără să afecteze accesul la structurile logice.

10.1. Structura fizică a bazei de date

- Structura fizică a bazei de date *Oracle* presupune existenţa unui set de fişiere binare prin intermediul cărora se realizează stocarea fizică a datelor.
- Aceste fişiere sunt împărţite în următoarele categorii:
  - fişiere de date (*Datafiles*);
  - fişiere de reluare (*Redo Log Files*);
  - fişiere de control (*Control Files*).

10.1.1. Fişiere de date

- Sunt fişiere fizice ale sistemului de operare care stochează:
  - datele tuturor structurilor logice ale bazei (obiecte create de utilizator);
  - informaţia necesară funcţionării sistemului.
    - Primul fişier de date creat este cel care stochează dicţionarul datelor. Dimensiunea acestui fişier este de cel puţin 150 *MB*.
- Caracteristicile fişierelor de date:
  - dimensiunea fişierelor de date poate fi schimbată ulterior creării acestora şi poate fi extinsă automat până la o valoare specificată, atunci când spaţiul alocat este depăşit (alocare dinamică a spaţiului);
  - unul sau mai multe fişiere de date formează o unitate logică numită spaţiu tabel (*tablespace*);
  - un fişier de date poate fi asociat unui singur spaţiu tabel şi unei singure baze de date.

- Informaţiile din fişierele de date pot fi accesate în timpul operaţiilor uzuale asupra bazei şi stocate în memoria *cache*.
- Dacă un utilizator interoghează date dintr-un tabel al bazei şi informaţiile cererii solicitate nu sunt încă în memoria *cache*, atunci acestea vor fi preluate din fişierul de date şi stocate în memorie.

- Alocarea unui fişier de date se face conform sintaxei următoare:

```
DATAFILE ['nume_fişier']
[SIZE întreg [ {K | M} ] ] [REUSE]
[clauza_autoextend]
```

  - În cazul spaţiilor tabel *undo*, pentru fiecare fişier de date asociat trebuie specificată o dimensiune.
  - Pentru celelalte spaţii tabel, clauza *SIZE* poate fi omisă dacă fişierul există deja.
  - Opţiunea *REUSE* permite sistemului să reutilizeze fişierele deja existente. Dacă fişierul specificat nu există, sistemul *Oracle* va ignora această clauză şi îl va crea.
  - Dacă se doreşte ca fişierele de date să fie auto-extensibile, atunci se precizează *clauza_autoextend*.

10.1.2. Fişiere de reluare

- Înregistrează toate modificările care au loc asupra datelor bazei (indiferent dacă au fost permanentizate sau nu) şi care nu au fost scrise încă în fişierele de date.
- O bază de date *Oracle* conţine două sau mai multe fişiere de reluare.
  - Specificarea acestora se face în momentul creării sau modificării bazei.
  - Fişierele de reluare sunt utilizate în manieră circulară. Cele care au fost folosite în întregime, pot fi arhivate până când sistemul le va reutiliza.
- Fişierele de reluare asigură protecţia bazei de date în cazul defecţiunilor.
  - De exemplu, dacă operaţiile bazei sunt întrerupte din cauza unei avarii de curent, datele din memorie nu pot fi scrise în fişierele de date şi sunt pierdute temporar. Atunci când baza de date este repornită, informaţiile vor putea fi recuperate din cele mai recente fişiere de reluare.
  - Deoarece şi aceste fişiere pot fi alterate, sistemul *Oracle* permite utilizarea fişierelor de reluare multiplexate. Aceasta presupune că pentru fiecare fişier de reluare se creează un grup de copii ale sale care sunt localizate pe discuri separate.

Orice modificare făcută asupra unuia dintre membrii grupului se propagă la întregul grup. Dacă un fişier de reluare este pierdut sau afectat, poate fi utilizată o copie a acestuia de pe alt disc.

- Vizualizările *V$LOG* şi *V$LOGFILE* din dicţionarul datelor furnizează informaţii despre grupurile fişierelor de reluare şi membrii acestora.

10.1.3. Fişiere de control

- Sunt fişiere binare de dimensiune redusă, necesare pentru pornirea şi funcţionarea bazei de date.
  - Fiecare fişier de control este asociat unei singure baze de date şi conţine informaţii despre structura fizică a acesteia (numele şi data creării bazei, informaţii despre spaţiile tabel, numele şi locaţia fişierelor de date şi a fişierelor de reluare, informaţii despre operaţiile de *backup* sau despre *checkpoint*-urile efectuate etc.).
- Orice bază de date *Oracle* deţine cel puţin un fişier de control, care este creat odată cu aceasta. Ca şi în cazul fişierelor de reluare, sistemul *Oracle* permite existenţa fişierelor de control multiplexate. În mod implicit, sistemul realizează cel puţin o copie a fişierului de control pe durata creării bazei de date.
- La pornirea unei instanţe *Oracle*, sistemul foloseşte fişierul de control pentru a identifica baza şi a determina dacă aceasta este în stare validă pentru utilizare. De asemenea, sunt identificate fişierele de reluare necesare execuţiei operaţiilor bazei de date.
- Fişierele de control reflectă automat schimbările (creare, redenumire sau ştergere) care au loc la nivelul fişierelor de date sau de reluare. Informaţiile din fişierele de control pot fi modificate doar de *server*-ul *Oracle*.
- Dacă un fişier de control devine inaccesibil, baza de date nu va funcţiona la parametrii corespunzători. Dacă toate copiile fişierelor de control ale bazei sunt pierdute, atunci aceasta trebuie recuperată (*recovery*) înainte de a fi deschisă.
- Interogând anumite vizualizări din dicţionarul datelor se pot obţine următoarele informaţii despre fişierele de control:
  - numele şi starea fişierelor asociate instanţei (*V$CONTROLFILE*);
  - starea şi localizarea tuturor parametrilor (*V$PARAMETER*);
  - înregistrările conţinute (*V$CONTROLFILE_RECORD_SECTION*).

10.2. Structura logică a bazei de date

- Utilizarea adecvată a structurilor logice determină alocarea optimă a spaţiului de pe disc în vederea obţinerii unei baze de date eficiente.
- Componentele structurii logice a unei baze de date *Oracle* sunt:
  - blocurile de date (*data block*);
  - extensiile (*extent*);
  - segmentele (*segment*);
  - spaţiile tabel (*tablespace*);
  - obiectele schemei (*schema object*).
- Gestiunea dinamică a spaţiului de pe disc pe măsura utilizării bazei de date, se realizează prin trei niveluri de granularitate:
  - segmentul;
  - extensia;
  - blocul.
- Spaţiile tabel, segmentele, extensiile şi blocurile permit definirea logică a organizării fizice a bazei de date şi efectuarea legăturii dintre nivelul fizic şi nivelul logic al acesteia.
- Obiectele schemei (tabele, vizualizări, vizualizări materializate, secvenţe, unităţi de program, sinonime, indecşi, grupări, dimensiuni, legături de baze de date) sunt structuri logice care referă în mod direct datele bazei.

10.2.1. Blocuri de date

- Sistemul administrează spaţiul de stocare al fişierelor de date prin unităţi logice numite blocuri de date.
- Blocul reprezintă cea mai mică unitate *I/O* folosită de baza de date, corespunzătoare unui bloc fizic de octeţi de pe disc.
- Dimensiunea blocului de date este definită în momentul creării bazei şi poate fi modificată ulterior.
  - Aceasta trebuie să reprezinte un multiplu al dimensiunii blocurilor fizice de la nivelul sistemului de operare.
  - Modificarea dimensiunii blocurilor logice se face prin intermediul parametrului *DB_BLOCK_SIZE*.

- Din punct de vedere structural, blocul de date *Oracle* cuprinde:
  - un antet (*header*), care conține informaţii generale referitoare la bloc:
    - adresa;
    - tipul segmentului căruia îi aparţine;
    - un catalog al tabelelor (*table directory*);
      - cuprinde informaţii despre tabelele care au date (înregistrări) stocate în blocul respectiv;
    - un catalog al liniilor (*row directory*);
      - conţine informaţii despre liniile situate în bloc (de exemplu, adresa acestora);
      - odată ce s-a alocat spaţiu pentru catalogul liniilor, acesta nu poate fi revendicat de celelalte porţiuni ale blocului de date; sistemul îl va utiliza doar atunci când sunt introduse linii noi în bloc;
  - un spaţiu pentru date (*data space*);
    - este format din linii de date care conţin informațiile stocate în tabele sau indecşi;
    - un bloc poate cuprinde una sau mai multe linii de date; dacă o linie nu încape într-un bloc, atunci aceasta poate fi stocată în două sau mai multe blocuri înlănţuite (de exemplu, dacă tabelul conţine o coloană de tip *LOB***);**
  - un spaţiu liber (*free space*);
    - este alocat pentru inserarea de noi linii sau actualizarea liniilor care necesită spaţiu suplimentar;
    - alegerea blocului în care va fi inserată o linie nouă depinde de spaţiul liber al acestuia şi de valorile parametrilor *PCTFREE* şi *PCTUSED*;
      - parametrul *PCTFREE* reprezintă procentul minim din blocul de date care trebuie păstrat liber pentru actualizările liniilor deja existente în bloc; valoarea implicită a acestui parametru este 10%;
      - parametrul *PCTUSED* reprezintă procentul minim al spaţiului utilizat din bloc care trebuie atins pentru a permite din nou inserarea unor linii de date; valoarea sa implicită este 40%;
    - într-un bloc, se pot introduce date atâta timp cât dimensiunea spaţiului liber este mai mare decât limita fixată de parametrul *PCTFREE*; sistemul *Oracle* va considera acest bloc indisponibil pentru inserarea de noi linii, până când procentajul spaţiului utilizat coboară sub valoarea dată de *PCTUSED*.

- spaţiul liber poate fi administrat automat de către sistem sau manual de către administrator;
  - pentru anumite tipuri de segmente (de date sau index), sistemul menţine una sau mai multe liste de blocuri de date care au dimensiunea spaţiului liber mai mare decât valoarea parametrului *PCTFREE*; acestea se numesc liste libere (*free list*); blocurile de date care compun listele libere sunt folosite pentru comenzile *INSERT*.

10.2.2. Extensii

- Extensia este o unitate logică de alocare a spaţiului bazei de date, compusă dintr-o mulţime contiguă de blocuri de date (din acelaşi fişier de date).
- Una sau mai multe extensii formează un segment.
  - Iniţial, segmentul are o singură extensie (*initial extent*);
    - De exemplu, la crearea unui tabel, sistemul alocă acestuia un segment cu o extensie iniţială care conţine un anumit număr de blocuri de date.
    - Blocurile ce corespund extensiei iniţiale sunt rezervate pentru liniile tabelului.
  - Dacă spaţiul de stocare alocat segmentului este complet folosit, sistemul *Oracle* alocă pentru acesta o nouă extensie (*next extent*).
- O extensie este alocată atunci când este creat sau extins un segment şi este dezalocată când segmentul este șters sau trunchiat.
- Atunci când extensiile sunt eliberate, sistemul modifică fişierele de date (dacă administrarea spaţiilor tabel se face local) sau reactualizează dicţionarul datelor (dacă administrarea spaţiilor tabel se face prin dicţionar). În acest fel, dimensiunea spaţiului de stocare disponibil este cunoscută în orice moment. Eliberarea unei extensii implică ştergerea datelor existente în blocurile de date alocate acesteia. Blocurile respective vor fi reutilizate pentru extensiile nou create.
- Parametrii care permit definirea dimensiunilor şi a limitelor extensiilor în cadrul segmentelor sunt definiţi cu ajutorul clauzei *STORAGE*.
  - Aceasta poate fi specificată în momentul creării sau modificării obiectelor bazei de date (tabele, vizualizări materializate, *cluster*-e, indecşi, partiţii).
  - Clauza *STORAGE* are următoarea sintaxă:

```
STORAGE ( {INITIAL întreg [ {K | M} ]
     | NEXT întreg [ {K | M} ]
     | MINEXTENTS întreg
```

```
      | MAXEXTENTS { întreg | UNLIMITED}
      | PCTINCREASE întreg
      | FREELISTS întreg
      | FREELIST GROUPS întreg
      | OPTIMAL [ { întreg [ {K | M} ] | NULL} ]
      | BUFFER_POOL {KEEP | RECYCLE | DEFAULT});
```

- *INITIAL* specifică dimensiunea în octeţi a extensiei iniţiale. Sistemul alocă spaţiu pentru această extensie atunci când creează un segment.
- *NEXT* precizează dimensiunea în octeţi a următoarei extensii care va fi alocată segmentului. Valoarea maximă a acesteia depinde de sistemul de operare.
- *MINEXTENTS* reprezintă numărul minim de extensii care trebuie alocat atunci când se creează un segment, iar *MAXEXTENTS* numărul total de extensii. Opţiunea *UNLIMITED* determină alocarea automată a extensiilor.
- *PCTINCREASE* specifică procentul de creştere a dimensiunii unei extensii (începând cu a treia extensie), faţă de dimensiunea extensiei anterioare. Valoarea minimă pentru *PCTINCREASE* este 0, ceea ce presupune că, exceptând extensia iniţială, toate extensiile alocate au aceeaşi dimensiune.
- *FREELISTS* specifică numărul de componente ale fiecărui grup de liste libere pentru un tabel, o partiţie, o grupare sau un index (valoarea minimă este 1, iar valoarea maximă depinde de dimensiunea blocurilor de date).
- *FREELIST GROUPS* precizează numărul de grupuri de liste libere pentru obiectele create.
- *OPTIMAL* precizează dimensiunea optimă în octeţi pentru segmentele de revenire, opţiunea fiind relevantă doar pentru aceste tipuri de segmente. Sistemul menţine dimensiunea specificată prin dezalocarea dinamică a extensiilor care nu mai sunt folosite. Opţiunea *NULL* presupune că nu se specifică o dimensiune optimă pentru segmentele de revenire. În acest caz, sistemul nu va dezaloca niciodată extensiile alocate acestor tipuri de segmente.
- Clauza *BUFFER_POOL* permite specificarea unei zone de memorie *cache* (*buffer pool*) implicite pentru un obiect al schemei. Toate blocurile de date alocate obiectului vor fi stocate în această zonă. Opţiunea *KEEP* permite sistemului să menţină obiectul în memorie, evitând astfel operaţiile *I/O*. Opţiunea *RECYCLE* determină depunerea blocurilor de date alocate segmentului într-o zonă de memorie *RECYCLE*. Opţiunea *DEFAULT* indică o zonă *buffer pool* implicită.

10.2.3. Segmente

- Segmentul este un set de extensii care conţine toate datele unei structuri logice dintr-un spaţiu tabel.
  - Un segment conţine cel puţin o extensie.
  - Fiecărui obiect fizic stocat îi corespunde un segment.
- Deoarece nu toate obiectele unei baze de date sunt stocate fizic, există obiecte cărora nu le corespunde niciun segment (de exemplu, vizualizările).
- Segmentul foloseşte blocuri de date care se găsesc în acelaşi spaţiu tabel.
- Sistemul permite administrarea automată a spaţiului liber din interiorul segmentelor bazei de date. În acest fel, spaţiul liber, respectiv cel utilizat al segmentului este monitorizat prin intermediul unor obiecte numite *bitmap*-uri.
- Baza de date *Oracle* conţine diferite tipuri de segmente:
  - segmente de date (*data segment*);
  - segmente index (*index segment*);
  - segmente temporare (*temporary segment*);
  - segmente de revenire (*undo segment*).

Segmentele de date

- Sunt definite atunci când este folosită comanda de creare a unui tabel sau a unei grupări.
- Un singur segment de date este folosit pentru stocarea tuturor datelor dintr-un tabel nepartiţionat care nu face parte din nici o grupare, dintr-o partiţie a unui tabel partiţionat sau dintr-o grupare de tabele.
- Una sau mai multe coloane ale unui tabel pot stoca obiecte de tip *LOB* (de exemplu, documente text, imagini, videoclipuri). În acest caz, *server*-ul *Oracle* stochează datele în segmente separate, numite segmente *LOB*.
- coloană a unui tabel poate fi definită de tip tablou imbricat. Valorile unei coloane de tip tablou imbricat sunt stocate în segmente diferite.
- Parametrii de stocare pentru un tabel sau o grupare, specificaţi prin clauza *STORAGE*, determină modul de alocare a extensiilor segmentului. Aceştia sunt importanţi, deoarece pot afecta eficienţa stocării şi a accesului la date.
- Pentru vizualizările materializate sistemul creează segmentele de date în aceeaşi manieră ca şi în cazul tabelelor sau grupărilor.

Segmentele index

- Sunt folosite pentru a stoca datele unui index. Fiecare index nepartiţionat este conţinut într-un singur segment. În cazul indecşilor partiţionaţi, fiecărei partiţii i se asociază câte un segment index.
- Sistemul creează un segment index de fiecare dată când este folosită comanda *CREATE INDEX*. În această comandă se pot specifica parametrii de stocare pentru extensiile asociate segmentului index şi spaţiul tabel în care este creat acesta. Spaţiul tabel respectiv poate fi diferit de cel care conţine segmentul de date al tabelului asupra căruia a fost creat indexul.

Segmentele temporare

- Sunt utilizate de sistem pentru analiza şi execuţia comenzilor *SQL* care necesită un spaţiu temporar de stocare.
- Sistemul alocă în mod automat segmente temporare atunci când este necesar şi le suprimă după execuţia comenzii *SQL*.
- Segmentele temporare sunt alocate în majoritatea cazurilor de sortare (atunci când operaţia respectivă nu se poate face în memorie sau dacă folosirea indecşilor nu presupune o soluţie mai eficientă).
- Comenzile *CREATE INDEX* şi *SELECT* (cu opţiunile *ORDER BY, DISTINCT, GROUP BY* şi operaţiile *UNION, INTERSECT* sau *MINUS*) pot determina alocarea unui segment temporar. O singură comandă *SQL* poate necesita mai multe segmente temporare (de exemplu, o interogare ce conţine simultan clauzele *DISTINCT, GROUP BY* şi *ORDER BY*).

Segmentele de revenire

- Înregistrează valorile anterioare ale datelor care au fost modificate de către tranzacţii (permanent sau nu), permiţând astfel revenirea la forma iniţială a acestora.
- O bază de date conţine unul sau mai multe segmente de revenire. Ele sunt folosite pentru a oferi consistenţă la citire, a anula acţiunea tranzacţiilor sau a efectua operaţiile de recuperare a bazei de date.
- Segmentele de revenire nu pot fi accesate de către utilizatorii sau administratorii bazei de date. Ele pot fi scrise şi citite doar de către sistem. Atunci când o tranzacţie devine activă, sunt modificate blocurile de date din segmentul de revenire asociat. Sistemul înregistrează în fişierele de reluare toate schimbările care au loc asupra blocurilor de date şi informaţiile pentru revenire. Astfel, în cazul eventualelor defecţiuni, *Oracle*

poate restaura automat baza de date la starea anterioară, folosind segmentele de revenire corespunzătoare tranzacţiilor care erau active în momentul defecţiunii.

- În versiunile anterioare, administrarea segmentelor de revenire se realiza manual. Începând cu versiunea *Oracle9i* se permite administrarea acestora şi în mod automat (*Automatic Undo Management*). *Server*-ul *Oracle* gestionează crearea, alocarea şi optimizarea segmentelor de revenire, menţinând valorile vechi ale datelor în spaţii tabel de revenire.

10.2.4. Spaţii tabel

- O bază de date este compusă dintr-o mulţime de unităţi logice de stocare numite spaţii tabel. Acestea pot fi asociate unei singure baze de date şi sunt formate dintr-unul sau mai multe segmente.
- Spaţiile tabel sunt utilizate pentru a grupa logic o mulţime de obiecte.
  - De exemplu, pentru a simplifica unele operaţii de administrare, spaţiile tabel pot grupa toate obiectele unei anumite aplicaţii.
  - Fiecare obiect al bazei are specificat un spaţiu tabel în care trebuie să fie creat. Datele care alcătuiesc obiectul sunt apoi stocate în fişierele de date alocate spaţiului tabel respectiv. Un fişier de date poate fi alocat unui singur spaţiu tabel.
- În orice bază de date *Oracle*, primul spaţiu tabel creat este *SYSTEM*, căruia îi va fi alocat automat (în timpul creării bazei de date) primul fişier de date. Pe lângă spaţiul tabel *SYSTEM*, baza de date conţine şi alte spaţii tabel (non-*SYSTEM*), pentru a stoca logic obiectele sale.

Spaţiul tabel *SYSTEM*

- Conţine dicţionarul datelor, inclusiv unităţile de program stocate.
- Conţine segmentul de revenire *SYSTEM*.
- Nu trebuie să conţină date ale utilizatorilor, deşi este permis acest lucru.

Spaţiile tabel non-*SYSTEM*

- Permit administrarea flexibilă a bazei de date.
- Separă segmentele de revenire, segmentele temporare, segmentele de date şi segmentele index.
- Separă datele dinamice de cele statice.
- Controlează spaţiul alocat pentru obiectele utilizatorilor.

- Sistemul permite atribuirea implicită a spaţiilor tabel.
- De asemenea, pentru fiecare utilizator se poate aloca explicit un spaţiu tabel, în care vor fi stocate toate obiectele create de acesta. Folosirea mai multor spaţii tabel permite flexibilitate în execuţia operaţiilor bazei de date.
- Spaţiile tabel pot fi *online* (accesibile) sau *offline* (neaccesibile).
  - În mod normal, spaţiile tabel sunt *online* pentru a permite utilizatorilor accesul la informaţiile stocate în ele.
  - Spaţiul tabel *SYSTEM* nu poate fi niciodată *offline*.
- Comanda *CREATE TABLESPACE* permite crearea unui spaţiu tabel:

```
CREATE [UNDO] TABLESPACE nume_spaţiu_tabel
   [DATAFILE specificaţie_fişier [clauza_autoextend]
           [, specificaţie_fişier [clauza_autoextend]…] ]
   [MINIMUM EXTENT întreg [ {K | M} ] ]
   [BLOCKSIZE întreg [K] ]
   [ {LOGGING | NOLOGGING} ]
   [DEFAULT clauza_storage]
   [ {ONLINE | OFFLINE} ]
   [ {PERMANENT | TEMPORARY} ]
   [clauza_administrare_extensie]
   [clauza_administrare_segment];
```

  - Opţiunea *UNDO* permite crearea unui spaţiu tabel de revenire (*undo*). Spaţiile tabel *undo* sunt rezervate sistemului, deci utilizatorii nu pot crea obiecte ale bazei de date în acestea. Dacă sunt definite mai multe spaţii tabel *undo*, numai unul dintre ele poate fi activ la un moment dat. Un spaţiu tabel *undo* poate fi suprimat numai dacă nu este activ.
  - Clauza *MINIMUM EXTENT* specifică dimensiunea minimă a extensiilor care vor fi alocate spaţiului tabel. Această dimensiune se va aplica pentru toate extensiile care vor fi asociate segmentelor din spaţiul tabel respectiv. Clauza nu este relevantă pentru spaţiile tabel temporare administrate prin dicţionarul datelor.
  - Pentru specificarea dimensiunii nonstandard a blocurilor de date alocate spaţiului tabel se foloseşte clauza *BLOCKSIZE*. Spaţiile tabel temporare nu permit blocuri de date cu dimensiune nonstandard.

- Prin opţiunea *LOGGING* se specifică dacă toate schimbările ce au loc asupra tabelelor, indecşilor şi partiţiilor care vor avea alocat spaţiul tabel definit sunt înregistrate implicit în fişierele de reluare. În modul *NOLOGGING*, datele sunt modificate cu jurnalizare minimă adică, doar pentru a marca extensiile noi cu *INVALID* şi pentru a înregistra modificările de la nivelul dicţionarului datelor. Clauza *DEFAULT* permite specificarea parametrilor impliciţi de stocare pentru toate obiectele create în spaţiul tabel definit.

- ❖ Valoarea unui parametru de stocare specificat la nivel de segment are prioritate faţă de cea precizată pentru acesta la nivelul spaţiului tabel, exceptând parametrii *MINIMUM EXTENT* şi *UNIFORM SIZE*.
- ❖ Dacă la nivel de segment parametrii de stocare nu au fost specificaţi explicit, atunci aceştia vor avea valorile specificate pentru spaţiul tabel în care se găsesc segmentele respective.
- ❖ Atunci când parametrii de stocare nu sunt specificaţi explicit pentru un spaţiu tabel, *server*-ul *Oracle* foloseşte valorile sistem implicite.
- ❖ Dacă parametrii de stocare sunt modificaţi, noile opţiuni se aplică doar pentru extensiile care nu au fost deja alocate.
- ❖ Există parametri de stocare ce nu pot fi specificaţi la nivel de spaţiu tabel, ci numai la nivel de segment.

- Dacă se foloseşte opţiunea *ONLINE*, spaţiul tabel devine disponibil imediat după ce a fost creat pentru toţi utilizatorii care au dreptul de a-l accesa. Această opţiune este implicită. *OFFLINE* presupune că spaţiul tabel nu este disponibil după creare, până când opţiunea este modificată la valoarea *ONLINE* prin comanda *ALTER TABLESPACE*.
- Clauza *PERMANENT* este specificată dacă spaţiul tabel va fi folosit pentru a stoca obiecte permanente. Dacă spaţiul tabel trebuie să stocheze obiecte temporare (de exemplu, segmentele folosite pentru sortări) atunci este utilizată clauza *TEMPORARY*. În acest caz nu se pot specifica clauzele *EXTENT MANAGEMENT LOCAL* sau *BLOCKSIZE*. Pentru a crea spaţii tabel temporare gestionate local se utilizează comanda *CREATE TEMPORARY TABLESPACE*.
- Specificarea modului de administrare a extensiilor alocate spaţiului tabel se face prin *clauza_administrare_extensie.*
- Pentru spaţiile tabel permanente, administrate local se poate specifica *clauza_ administrare_segment*.

Aceasta are următoarea sintaxă:

```
SEGMENT SPACE MANAGEMENT {MANUAL | AUTO}
```

- Opţiunea *MANUAL* presupune că sistemul administrează spaţiul liber folosind liste libere. Dacă se specifică opţiunea *AUTO*, sistemul *Oracle* va administra spaţiul liber al segmentelor spaţiului tabel folosind un *bitmap*.
- Spaţiile tabel pot fi modificate sau eliminate. Modificarea unui spaţiu tabel se face prin comanda *ALTER TABLESPACE* care permite adăugarea unor fişiere de date, transferarea fişierelor, schimbarea parametrilor de stocare, schimbarea stării din *online* în *offline* şi reciproc, recuperarea spaţiului tabel respectiv etc.
- Spaţiile tabel pot fi suprimate chiar atunci când conţin date. Comanda prin care se suprimă un spaţiu tabel este *DROP TABLESPACE*. Spaţiul tabel *SYSTEM* nu poate fi eliminat deoarece conţine dicţionarul datelor. De asemenea, nu pot fi eliminate spaţiile tabel care conţin segmente active. Sistemul *Oracle9i* şterge odată cu un spaţiu tabel şi fişierele de date asociate acestuia.
- Informaţii despre spaţiile tabel definite în baza de date se obţin interogând vizualizările *DBA_TABLESPACES* şi *V$TABLESPACE* din dicţionarul datelor.

Relaţia dintre baze de date, spaţii tabel şi fişiere de date presupune că:

- ❖ fiecare bază de date este împărţită din punct de vedere logic în unul sau mai multe spaţii tabel;
- ❖ unul sau mai multe fişiere de date sunt create explicit pentru fiecare spaţiu tabel, cu scopul de a stoca fizic datele din structurile sale logice;
- ❖ suma mărimilor tuturor fişierelor de date asociate unui spaţiu tabel reprezintă capacitatea totală de stocare a spaţiului tabel;
- ❖ suma capacităţilor de stocare a spaţiilor tabel ale unei baze de date reprezintă capacitatea totală de stocare a bazei.

- Vizualizări din dicţionarul datelor care oferă informaţiile despre spaţii tabel, fişiere de date, segmente şi extensii:
  - *DBA_TABLESPACES*;
  - *DBA_DATA_FILES*;
  - *DBA_SEGMENTS*;
  - *DBA_EXTENTS*;
  - *DBA_FREE_SPACE*.

10.2.5. Scheme de obiecte

- O schemă reprezintă mulţimea obiectelor bazei de date, aflate în posesia unui utilizator (fiecare utilizator deţine o singură schemă). Numele schemei este acelaşi cu numele utilizatorului.
- Între spaţiile tabel şi schemele de obiecte nu există o corespondenţă biunivocă. Obiectele aceleiaşi scheme pot fi în spaţii tabel diferite, iar un spaţiu tabel poate conţine obiecte din mai multe scheme.
- Pentru a accesa un obiect din propria schemă, utilizatorul poate folosi doar numele acestuia. Pentru referirea unui obiect din schema altui utilizator, trebuie specificat atât numele obiectului, cât şi schema din care face parte, prin folosirea notaţiei *schema.obiect*.
- Obiecte ale schemei
  - Tabelul (*table*) este unitatea fundamentală pentru stocarea datelor unei baze *Oracle*. Sistemul permite stocarea partiţionată a tabelelor. Dacă un tabel este partiţionat, atunci fiecare partiţie a sa corespunde unui segment.
  - Vizualizarea (*view*) este o reprezentare logică a datelor din unul sau mai multe tabele sau vizualizări. O vizualizare poate fi privită ca o „interogare stocată“. Vizualizarea nu stochează înregistrări, ea doar referă datele unor tabele de bază (*master table*) care, la rândul lor, pot fi tabele sau vizualizări.
  - O vizualizare materializată (*materialized view*) permite accesul indirect la datele unui tabel, prin stocarea rezultatelor unei interogări. Diferenţa faţă de o vizualizare obişnuită, care fiind virtuală nu stochează date, este că vizualizarea materializată necesită alocarea unui spaţiu de memorie pentru stocarea liniilor rezultate în urma interogării asociate.
  - Secvenţa (*sequence*) este un obiect al bazei de date care permite utilizatorilor să genereze automat o listă serială de numere întregi unice. Numerele secvenţei sunt independente de tabele, astfel că aceeaşi secvenţă poate fi folosită pentru mai multe tabele.
  - Unităţile de program (*program unit*) reprezintă subprograme (proceduri şi funcţii stocate), pachete, declanşatori şi clase *Java*.
  - Sinonimele (*synonym*) sunt nume alternative pentru tabele, vizualizări, secvenţe sau unităţi de program.

- Indexul (*index*) este o structură opţională, asociată tabelelor bazei de date, care conţine adresa fizică a fiecărei linii dintr-un tabel.
- Grupările (*cluster*) sunt structuri opţionale pentru stocarea datelor din tabele create cu scopul de a permite acces rapid la acestea. O grupare conţine unul sau mai multe tabele.
  - Din punct de vedere logic, tabelele unei grupări sunt stocate unitar (în acelaşi segment) şi, de obicei, sunt folosite împreună, deoarece au coloane comune. Aceste coloane sunt numite chei *cluster* şi trebuie indexate, astfel încât liniile grupării să fie recuperate cu minimum de operaţii *I*/*O*.
  - Asemenea indecşilor, grupările nu afectează proiectarea aplicaţiilor. Datele stocate într-o grupare pot fi accesate prin comenzi *SQL*, în acelaşi mod ca şi cele dintr-un tabel obişnuit.
- Obiectul dimensiune (*dimension*) defineşte ierarhia („părinte“/„copil“) dintre perechi sau seturi de coloane. Fiecare valoare de pe un nivel „copil“ este asociată cu o valoare unică de pe nivelul „părinte“.
  - Obiectul dimensiune nu conţine date, ci relaţii logice între tabele. Coloanele unui obiect dimensiune pot proveni din acelaşi tabel (denormalizat) sau de la mai multe tabele (în întregime sau parţial normalizate).
- O legătură de baze de date (*database link*) este un obiect al unei scheme dintr-o bază care permite accesarea obiectelor din altă bază. Atunci când se lucrează cu mai multe instanţe, legăturile dintre bazele de date sunt folosite pentru a transmite date între aplicaţii.

10.3. Dicţionarul datelor

- Una dintre cele mai importante componente ale bazei de date *Oracle* este dicţionarul datelor.
- Dicționarul datelor include tabele şi vizualizări *read-only* ce conţin informaţii despre baza de date:
  - definiţiile tuturor obiectelor din schemele bazei;
  - cantitatea de spaţiu alocat pentru obiectele schemelor şi cantitatea de spaţiu utilizat de acestea la momentul curent;
  - valorile implicite ale coloanelor;
  - constrângerile de integritate;

  - numele utilizatorilor *Oracle*;
  - privilegiile şi *role*-urile acordate fiecărui utilizator;
  - informaţii de auditare etc.
- Dicţionarul datelor:
  - este generat automat la crearea bazei de date;
  - este reactualizat de către *server*-ul *Oracle* după fiecare comandă de definire a datelor sau de control al accesului la acestea;
  - reflectă imaginea bazei de date (structura fizică şi logică) la un moment dat;
  - este deţinut de către utilizatorul *SYS*;
  - se află în spaţiul tabel *SYSTEM*.
- Sistemul poate accesa dicţionarul datelor pentru a obţine informaţii despre utilizatori, despre obiecte sau despre structurile de stocare. Orice utilizator poate consulta dicţionarul datelor pentru a afla informaţii despre baza de date (documentare sau administrare). Accesul la informaţiile din dicţionar se face utilizând instrucţiunea *SELECT* din *SQL*.
- Din punct de vedere structural, dicţionarul datelor este compus din tabele de bază şi vizualizări publice asupra acestora.
  - Tabelele de bază stochează informaţiile asociate bazei de date, fiind primele obiecte create. În general, doar sistemul are acces direct la scrierea şi citirea datelor din aceste tabele. Majoritatea datelor din tabelele de bază sunt stocate în format criptat şi sunt dificil de interpretat direct de către utilizatori.
  - Dicţionarul datelor conţine un tabel numit *DUAL*, pe care aplicaţiile sistemului sau cele create de utilizatori îl pot referi pentru a garanta un rezultat cunoscut. Acest tabel este format dintr-o singură coloană (*DUMMY*) şi o singură linie care conţine valoarea *X*.
  - Vizualizările decodifică informaţiile stocate în tabelele de bază şi le sintetizează pentru a fi disponibile utilizatorilor. Vizualizările dicţionarului pot fi relative la:
    - toate obiectele din baza de date (prefix *DBA_*);
    - obiectele accesibile unui utilizator (prefix *ALL_*);
    - obiectele unui utilizator (prefix *USER_*);
    - performanţe (prefix *V$* sau *GV$*).
- Utilizatorii fără privilegii de administrare pot accesa doar vizualizările prefixate de *USER_* sau *ALL_*. Pentru a obţine lista vizualizărilor disponibile se poate interoga vizualizarea *DICTIONARY* care are sinonimul *DICT*.

- Vizualizările prefixate de *DBA_* prezintă o imagine globală asupra bazei de date. Acestea pot fi interogate doar de utilizatorii care au privilegiul *SELECT ANY DICTIONARY*. Pentru a referi o astfel de vizualizare, trebuie utilizată notaţia cu punct şi prefixat numele acesteia cu numele proprietarului său (*SYS*).
- Vizualizările prefixate de *ALL_* reflectă baza de date din perspectiva utilizatorului curent. Acestea oferă informaţii despre toate obiectele schemei la care are acces utilizatorul curent.
- Vizualizările prefixate de *USER_* conţin informaţii despre utilizatorul curent, obiectele create de acesta, privilegiile sistem pe care le deţine, privilegiile pe care le-a acordat altor utilizatori etc. Exceptând coloana *OWNER*, vizualizările prefixate de *USER_* conţin aceleaşi coloane ca şi celelalte vizualizări.
- În funcţie de valorile sufixului (precizat între paranteze), aceste vizualizări furnizează informaţii despre:
  - utilizatori (*USERS*);
  - coloane specificate în constrângeri (*CONS_COLUMNS*);
  - tabele ale bazei (*TABLES*) şi tabele externe (*EXTERNAL_TABLES*);
  - vizualizări (*VIEWS*) şi vizualizări materializate (*MVIEWS*);
  - grupări (*CLUSTERS*) şi indecşi (*INDEXES*);
  - declanşatori (*TRIGGERS*);
  - sinonime (*SYNONIMS*) şi secvenţe (*SEQUENCES*);
  - obiecte (*OBJECTS*);
  - biblioteci (*LIBRARIES*);
  - politici de securitate (*POLICIES*);
  - privilegii obiect asupra tabelelor (*TAB_PRIVS*);
  - tipuri obiect (*TYPES*) şi colecţie (*COLL_TYPES*) etc.
- Vizualizarea *DBA_OBJECTS* este utilă pentru obţinerea de informaţii referitoare la toate categoriile de obiecte ale unei scheme. Coloanele acestei vizualizări sunt următoarele:
  - *OWNER* (proprietarul obiectului);
  - *OBJECT_NAME* (numele obiectului);
  - *SUBOBJECT_NAME* (numele subobiectelor);
  - *OBJECT_ID* (identificatorul obiectului);
  - *DATA_OBJECT_ID* (identificatorul segmentului ce conţine obiectul);
  - *OBJECT_TYPE* (tipul obiectului);

- *CREATED* (data creării obiectului);
- *LAST_DDL_TIME* (data ultimei modificări structurale a obiectului);
- *TIMESTAMP* (specificarea formatului datei);
- *STATUS* (starea obiectului);
- *TEMPORARY* (precizarea că obiectul este temporar);
- *GENERATED* (specificarea modului de generare al obiectului);
- *SECONDARY* (precizarea că obiectul este secundar, creat prin metoda *ODCIIndexCreate* a lui *Oracle Data Cartridge*).

Exemplul 10.1

```
 -- proprietarul, numele şi tipul obiectelor din baza de date.
SELECT  OWNER, OBJECT_NAME, OBJECT_TYPE
FROM    SYS.DBA_OBJECTS;

-- proprietarul, numele şi tipul tuturor obiectelor accesibile
utilizatorului curent.
SELECT  OWNER, OBJECT_NAME, OBJECT_TYPE
FROM    ALL_OBJECTS;

-- numele şi tipul obiectelor create de utilizatorul curent.
SELECT  OBJECT_NAME, OBJECT_TYPE
FROM    USER_OBJECTS;
```

- Vizualizările prefixate de *V$* oferă informaţii despre performanţele bazei de date.
  - Se numesc vizualizări fixe, deoarece administratorii bazei nu le pot modifica sau suprima.
  - Fiecare instanţă are asociat un set propriu de vizualizări prefixate de *V$*.
  - Fiind virtuale, acestea există în memorie doar în timp ce baza de date este pornită.
  - Prin interogarea acestor vizualizări se obţin informaţii despre:
    - sesiunile curente (*V$SESSION*) şi procesele active (*V$PROCESS*);
    - obiectele care sunt blocate la un moment dat şi sesiunile care le accesează (*V$ACCES*);
    - tranzacţiile active (*V$TRANSACTION*);
    - procesele *background* (*V$BGPROCESS*);
    - fişierele de reluare care trebuie arhivate (*V$ARCHIVE*);
    - numele şi numărul tuturor spaţiilor tabel asociate fişierelor de control (*V$TABLESPACE*);
    - numele şi starea fişierelor de control asociate instanţei bazei de date (*V$CONTROLFILE*) etc.

Bibliografie

1. *Programare avansată în Oracle9i*, I. Popescu, A. Alecu, L. Velcescu, G. Florea (Mihai), Ed. Tehnică (2004)
2. *Oracle Database PL/SQL Language Reference 11g Release 2*, Oracle Online Documentation (2012)
3. *Oracle Database SQL Language Reference11g Release 2*, Oracle Online Documentation (2012)
4. *Oracle Database 11g*: *PL/SQL Fundamentals*, *Student Guide*, Oracle University (2009)