

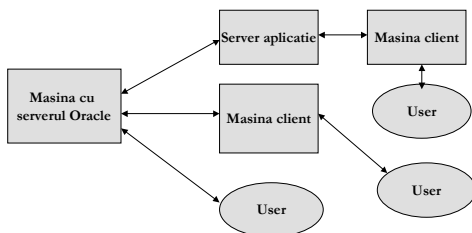
Capitolul 1

Arhitectura Oracle

Serverul ORACLE

- ◆ Este un sistem de gestiune a bazelor de date relationale
- ◆ Userul poate lucra:
 - ♦ Cu un client pe aceeași mașină cu serverul (de exemplu un client SQL*Plus rulant pe aceeași mașină cu serverul Oracle)
 - ♦ Clientul rulează pe o altă mașină decât serverul (two-tiered)
 - ♦ Aplicația userului accesează o altă aplicație iar la rândul ei aceasta e în comunicare cu serverul (three-tiered)
- ◆ Iată o figură:

Serverul ORACLE



USER vs UTILIZATOR

- ◆ Vom numi în cele ce urmează user un cont de utilizator Oracle (exemplu: user 'student' cu parola 'stud1234')
- ◆ Utilizator este o persoană care prin intermediul unui cont de user interacționează cu Oracle
- ◆ Utilizator poate fi numit într-un sens mai larg și un proces, o aplicație care folosește un cont user pentru a interacționa cu Oracle

PROCESE

- ◆ Când userul accesează o aplicație care lucrează cu Oracle sunt create 2 procese:
 - ♦ Proces user pe mașina pe care lucrează utilizatorul. Acesta interacționează cu utilizatorul și asigură comunicarea cu cel de-al doilea proces
 - ♦ Proces server, pe mașina unde este instalat serverul Oracle. Acesta comunică cu serverul Oracle în numele procesului user.

SESIUNE

- ◆ O sesiune de lucru reprezintă o conexiune a unui user cu serverul Oracle.
- ◆ Un user poate avea mai multe sesiuni deschise simultan:
 - ♦ Pe aceeași bază de date
 - ♦ Cu același cont de user
 - ♦ Pe aceeași mașină sau pe mașini diferite

PROCESUL USER

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

PROCESUL SERVER

- ◆ Ruleaza pe masina unde este instalat serverul Oracle
- ◆ Deserveste un singur proces user sau mai multe procese user, in functie de configuratie: Dedicated Server sau Multithreaded server)
- ◆ Foloseste o zona de memorie (PGA – Program Global Area) care va fi descrisa mai jos.
- ◆ Este un intermediar intre procesul user si serverul Oracle, folosind OPI – Oracle Program Interface – pentru a interactiona cu acesta
- ◆ Procesul se incheie cand utilizatorul termina sesiunea.

INSTANTA ORACLE

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

SGA – System Global Area

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

SGA – System Global Area -cont

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

Procese BACKGROUND

- ◆ Database Writer (DBWR) – Este responsabil cu scrierea blocurilor de date modificate/inserate din bufferele de memorie in fisierele de pe disc. In Oracle 10g pot fi maximum 20 de procese de acest fel
- ◆ Log Writer (LGWR) – Scrie datele din Redo Log Buffer pe disc. Scrierea se face secvential intr-un fisier de Redo Log.
- ◆ Checkpoint (CKPT) – Acest proces scrie periodic pe disc toate blocurile de date din buffere care au fost modificate. O astfel de actiune este denumita un checkpoint. Procesul anunta actiunea lui DBWR, actualizeaza fisierele de date si control ale bazei de date si inregistreaza momentul in care s-a facut checkpointul.

Procese BACKGROUND - cont

- ◆ System Monitor (SMON) - Are functia de verificare consistenta date si sa initieze recuperarea dupa incident atunci cand o instanta Oracle care a avut un incident reporneste.
- ◆ Process Monitor (PMON) - Dealloca resursele unui proces care are un incident. Folosit pentru procesele user care au incidente.
- ◆ Archiver (ARCV) - Copiaza fisierele Redo Log in arhiva de pe disc atunci cand acestea sunt pline sau cand acestea se schimba. Actiunea are loc in anumite conditii.

Baza de date

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

Tipuri de fisiere

- ◆ Data files – fisiere de date. Fisierele de date contin:
 - ◆ dictionarul de date al BD (mai stiti ce este acesta?),
 - ◆ obiectele userului (mai stiti care sunt acestea?)
 - ◆ Vechile valori ale datelor modificare de tranzactiile curente (before image)
- ◆ O baza de date are cel putin un astfel de fisier.

Tipuri de fisiere - cont

- ◆ Redo Log Files – Fisiere Redo Log. Contin modificarile facute in baza de date care sunt necesare la reconstructia ei in caz de incident.
- ◆ Fiecare baza de date contine cel putin 2 astfel de fisiere care pot fi tinute pe dispozitive de stocare diferite, prevenind pierderi de date in cazul in care un disc este distrus.

Tipuri de fisiere - cont

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

Alte fisiere

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

Etape de procesare cerere

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

Etape de procesare cerere - cont

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

SHARED POOL

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

Library cache

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

Data Dictionary Cache

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

Database Buffer Cache

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

Database Buffer Cache – cont.

- ◆ Oracle utilizează algoritmul LRU (ce este acesta?) pentru eliberare poziții în buffer,
- ◆ Excepție: operațiile de tip full table scan (când o întreaga tabelă este citită în memorie). În acest caz ultimele blocuri citite pot fi primele dealocate.
- ◆ Algoritmii utilizați sunt complecși, cele de mai sus sunt o prezentare schematică a strategiilor de gestiune a bufferului.
(vezi de exemplu <http://www.adp-gmbh.ch/ora/concepts/cache.html>)

Program Global Area (PGA)

- ◆ Zona de memorie folosită în mod exclusiv de un proces (server sau background) – nu e comună ca în cazul SGA. Ea conține:
 - ♦ Sort area – zona sortării, folosită la operațiile de ordonare (sortare).
 - ♦ Informații sesiune, de exemplu drepturile utilizatorului acelei sesiuni.
 - ♦ Stare cursor continuând starea cursorilor folosiți în sesiunea respectivă
 - ♦ Stivă, continuând diverse variabile de sesiune.

Exemplu: procesare UPDATE

- ◆ Sa luăm exemplul unei cereri de actualizare:

```
UPDATE EMP  
SET SAL = SAL * 1.1  
WHERE EMPNO=1123
```

- ◆ Se execută întâi Pasul 1 - Parse
- ◆ La etapa EXECUTE (Pasul 2) se efectuează operațiile:

Exemplu: procesare UPDATE – cont

1. Procesul server citește blocurile de date și de rollback – valori anterioare modificărilor necomise - din fișierele de date (dacă nu sunt deja în Buffer Cache)
2. Copii ale blocurilor sunt puse în Buffer Cache (dacă nu existau).
3. Procesul server blochează datele respective.
4. Procesul server înregistrează în Redo Log Buffer schimbările de făcut în rollback și în date

Exemplu: procesare UPDATE – cont

5. Procesul server plasează versiunile originale ale blocurilor modificate în Rollback și modifică blocurile de date. Ambele operații se fac în Buffer Cache. Ambele blocuri (rollback și date) sunt marcate ca 'dirty blocks' (blocuri modificate), adică blocuri care nu sunt identice cu cele de pe disc.

Segment de Rollback

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

Segment de Rollback - cont

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

Redo Log Buffer

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

Redo Log Buffer - cont

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

Database Writer

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

Database Writer - cont

- ◆ Scrierea se face cand:
 - ♦ Numarul de 'dirty blocks' in buffer depaseste o anumita valoare
 - ♦ Un proces care cauta locuri libere in buffer nu le gaseste
 - ♦ Timeout
 - ♦ Evenimente care forteaza un checkpoint: exemplu: inchiderea bazei de date.

Log Writer

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

COMMIT

- ◆ Oracle utilizează un mecanism rapid de commit care garantează recuperarea schimbărilor comise în caz de incident.
- ◆ Acest mecanism utilizează un System Change Number (SCN) fiecărei tranzacții care comite. Aceste numere sunt crescătoare și unice la nivelul bazei de date

COMMIT - Pași

- ◆ La apariția unui COMMIT:
 1. Procesul server plasează o înregistrare de commit, împreună cu SCN-ul acesteia în Redo Log Buffer
 2. LGWR scrie o porțiune contigua de înregistrări din buffer până la cea care conține commit în fișierele redo de pe disc. În felul acesta este garantată recuperarea după incident

COMMIT - Pași

3. Utilizatorul este informat că s-a efectuat COMMIT-ul.
4. Procesul server înregistrează că tranzacția este completă și că resursele blocate de ea pot fi eliberate.

Deci:

- ◆ Scrierea efectivă a datelor pe disc este efectuată independent de DBWR.
- ◆ Dimensiunea tranzacției nu contează

Bibliografie generală

1. Colin McGregor - Oracle Database 2 Day DBA, 10g
2. Ulrike Schwinn, Vijayanandan Venkatachalam – Database administration

Lecturi obligatorii

1. Colin McGregor - Oracle Database 2 Day DBA, 10g

Link: http://download-west.oracle.com/docs/cd/B19306_01/server.102/b14196.pdf

- ◆ Capitolul 5 (Managing the Oracle Instance) – integral
- ◆ Capitolul 6 (Managing Database Storage Structure) pag. 6-1 până la 6-5 (fără Tablespaces)

Sfârșitul
primului
capitol