

## INTRODUCERE IN SERVERUL ORACLE

### Gestiunea bazelor de date si a informatiilor

Un server de baze de date este cheia rezolvării problemelor de gestiune a informațiilor. În general, un server trebuie să asigure gestiunea unei cantități mari de date într-un mediu utilizator în astă fel ca mai mulți utilizatori să poată accesa concurrent aceleasi date. Această acțiune trebuie să fie insotita de o performanță crescută. Serverul de baze de date trebuie să impiedice accesul neautorizat și furnizeaza solutii eficiente pentru refacerea în caz de defectiune.

Serverul Oracle furnizează solutii efective și eficiente cu următoarele caracteristici:

- mediu client / server (procesare distribuită) pentru a lua toate avantajele unui sistem dat sau a unei retele, Oracle permite ca procesarea să fie împărțita între serverul de baze de date și programul client. Calculatorul care rulează sistemul de gestiune a bazelor de date manipulează toate responsabilitățile serverului de baze de date în timp ce statia de lucru rulează aplicații concentrate pe interpretarea și afisarea datelor
- gestiunea bazelor de date mari și a spațiului în Oracle suportă baze de date mari potențial de dimensiune terabyte. Pentru a face eficientă folosirea echipamentelor hardware costisitoare, permite control complet al folosirii spațiului
- mai mulți utilizatori concurrenti ai bazei de date, Oracle suportă un număr mare de utilizatori concurrenti care executa o varietate de aplicații ce operează cu aceleasi date. Aceasta minimizează conflictul dintre date și garantează concurrenta datelor.
- performanță înaltă la procesarea tranzacțiilor - Oracle menține caracteristicile anterioare cu un mare grad de performanță al întregului sistem. utilizatorii bazei de date nu suferă scaderea performanțelor de procesare
- înaltă disponibilitate - în anumite locații Oracle lucrează 24 ore pe zi fără limitarea utilizării bazei de date sau caderea parțială a sistemului nu intrerupe utilizarea bazei de date
- disponibilitatea controlului - Oracle poate controla selectiv disponibilitatea bazei de date la nivelul bazei de date sau la nivelul sub bazei. De exemplu: un administrator poate dezactiva folosirea unei aplicații specifice astfel ca datele aplicației să poată fi reincarcate fără a afecta alte aplicații
- deschidere, standard industrial - Oracle aloca la standardele industriale acceptate pentru limbajul de acces la date, sistemul de operare, interfața utilizator, protocolele de comunicare în rețea. Este un sistem deschis care protejează investitia cumpărătorului.

Versiunea Oracle server 8.0 este certificată de US National Institute of Standard and Technology 100% împreună cu standardul ANSI / ISO SQL92. Oracle satisfăce toate cerințele standardului US Government IS FIPS / 27-2.

Oracle suportă standardul Simple Network Management Protocol (SNMP) pentru gestiunea sistemului. Acest protocol permite administrarea sistemelor heterogene cu o singură interfață administrator.

- gestionarea securității - pentru protecția împotriva accesului neautorizat la bazele de date și la utilizarea lor, Oracle furnizează o caracteristică de securitate pentru a limita și monitoriza accesul la date. Această caracteristică face gestiunea usoară chiar dacă proiectul de accesare a datelor este complex

- integritatea bazelor de date - Oracle introduce integritatea datelor, "reguli de lucru" care dicteaza standardele pentru datele acceptate. Ca rezultat, costul codificarii in gestiunea cautarii in mai multe baze de date este eliminat
- sistem distribuit - pentru retele, medii distribuite, Oracle combina datele localizate fizic pe diferite calculatoare intr-o baza de date logica ce poate fi accesata de toti utilizatorii retelei. Sistemele distribuite au acelasi grad de transparenta pentru utilizatori si consistenta a datelor ca si a sistemelor distribuite, pe deasupra primeste avantajele gestiunii bazelor de date locale. Oracle ofera optiuni heterogene care permit utilizatorilor sa acceseze transparent date din anumite baze de date non-Oracle
- portabilitate - softul Oracle este portat sa lucreze sub diferite sisteme de operare.

Aplicatiile dezvoltate pentru Oracle pot fi portate pe oricare sistem de operare cu modificari putine sau deloc

- compatibilitate - softul Oracle este compatibil cu standardele industriale, incluzind majoritatea sistemelor de operare standard. Aplicatiile dezvoltate pentru Oracle pot fi folosite virtual in oricare sistem de operare cu modificari putine sau deloc
- conectibilitate - Oracle permite diferitelor tipuri de calculatoare si sisteme de operare sa partajeze informatiile in retea
- mediu replicat - Oracle permite replicarea grupurilor de tabele si a obiectelor in locatii multiple.

## Serverul Oracle

Serverul Oracle este un sistem de gestiune a bazelor de date relationale care furnizeaza o cale de gestionare deschisa, cuprinsa de o structura logica a informatiilor.

Un server Oracle consta dintr-o baza de date Oracle si o instanta Oracle.

## Structured Query Language (SQL)

SQL este limbajul de programare care defineste si manipuleaza bazele de date. Bazele de date SQL sunt baze de date relationale; datele sunt memorate intr-un set de relatii simple. O baza de date poate avea una sau mai multe tabele. Oricare tabela are coloane si rinduri.

Se pot defini date si manipula date in tabele cu comenzi SQL. Se pot folosi comenzi ale limbajului de definire a datelor (Data Definition Language-DDL) pentru a manevra date. Comenzile DDL includ comenzi pentru crearea si modificararea bazelor de date si tabelelor.

Se pot actualiza, sterge, accesa date dintr-o tabela cu comenzi de manipulare a datelor (Data Manipulation Commands DML). Comenzile DML includ comenzi de modificare si extragere a datelor. Cea mai comună comandă SQL este SELECT care permite regasirea datelor din baza de date.

In plus fata de comenzile SQL, serverul Oracle are un limbaj procedural numit PL/SQL. PL/SQL permite programatorilor sa programeze comenzi SQL.

## Structura bazei de date

Baza de date Oracle are o structura fizica si o structura logica. Deoarece structura logica si structura fizica sunt separate, structura fizica poate fi gestionata fara sa afecteze accesul la structura logica.

**Structura fizica :** o baza de date Oracle are o structura fizica determinata de fisierele sistemului de operare care constituie baza de date Oracle. Fiecare baza de date este constituita din trei tipuri de fisiere: unul sau mai multe fisiere de date, doua sau mai multe fisiere redo-log si unul sau mai multe fisiere de control.

Fisierele bazei de date Oracle furnizeaza memorarea fizica a informatiilor bazei de date.

**Structura logica** : incluzind spatiul pentru tabele (tablespace), segmente si extenturi stabileste modul in care este utilizat spatiul fizic al bazei de date. Structura logica a bazei de date este determinata de:

- unul sau mai multe spatii pentru tabele (tablespace). Spatiul pentru tabele (tablespace) este o zona logica de memorare

- obiectele schemei bazei de date

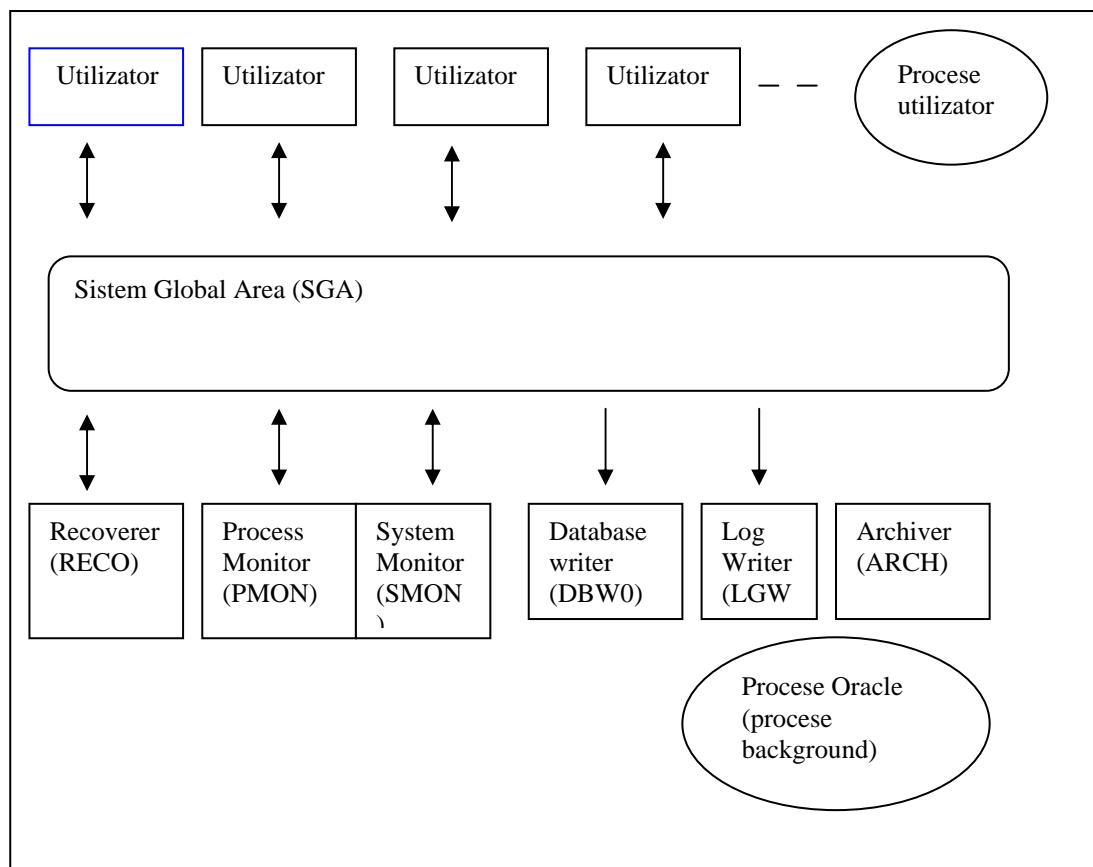
Schema este o colectie de obiecte. Obiectele schemei sint structuri logice care refera direct datele din baza de date. Schema include stucturi cum at fi tabele, view-uri, sevante, proceduri memorate, sinonime, indecesi,clustere si baze legate.

Schema obiectelor si relatiile dintre ele formeaza designul relational al bazei de date.

**Instanta Oracle** - de fiecare data cind este startata o baza de date, este alocata o zona sistem globala (SGA) si sunt startate procesele background. SGA este o zona de memorie folosita pentru partajarea informatiilor bazei de date intre utilizatorii bazei de date. Combinatia dintre procesele background si bufferele de memorie este numita instanta Oracle.

O instanta Oracle are doua tipuri de procese: procese utilizator si procese Oracle.

- un proces utilizator executa codul unui program de aplicatie (cum ar fi Oracle Forms) sau un utilitar Oracle (cum ar fi Oracle Enterprise Manager)
- procesul Oracle este procesul server care efectueaza lucru pentru procesele utilizator si procesele background care efectueaza lucru de intretinere pentru serverul Oracle



## Softul de comunicatie si NET8

Daca procesul utilizator si procesul server sint pe calculatoare diferite in retea sau daca procesul utilizator este conectat la un proces server partajat prin intermediul unui dispecer de procese, procesul utilizator si procesul server, comunica folosind NET8. Dispecerul este un proces background optional, prezent numai cind este folosita o configuratie server multi thread. NET8 este interfata Oracle pentru standardizarea protoocoalelor de comunicatie care permit transmisia corespunzatoare a datelor intre calculatoare.

## Oracle Parallel server: Sisteme cu instante multiple

Anumite arhitecturi hardware (de exemplu sistem de discuri partajat) permit ca mai multe calculatoare sa partajeze accesul la date, soft sau echipamente periferice. Oracle cu optiunea Parallel Server poate lua avantajele unei asemenea arhitecturi prin rularea mai multor instante care partajeaza o singura baza de date fizica. In aplicatiile corespunzatoare, Oracle Parallel Server permite accesarea unei singure baze de date de catre utilizatorii de pe mai multe masini cu cresterea performantelor.

## Bazele de date Oracle

O baza de date Oracle este o colectie de date care sunt tratate ca o unitate. Scopul general al bazei de date este memorarea si regasirea informatiilor.

Baza de date are o structura logica si o structura fizica.

## Deschiderea si inchiderea bazei de date:

O baza de date poate fi deschisa (accesibila) sau inchisa (inaccesibila). In situatii normale baza de date este deschisa si disponibila utilizatorilor. Totusi, o baza de date este uneori inchisa pentru efectuarea unor functii de administrare specifice care cer ca baza de date sa nu fie disponibila utilizatorilor.

## Structura bazei de date si gestiunea spatiului

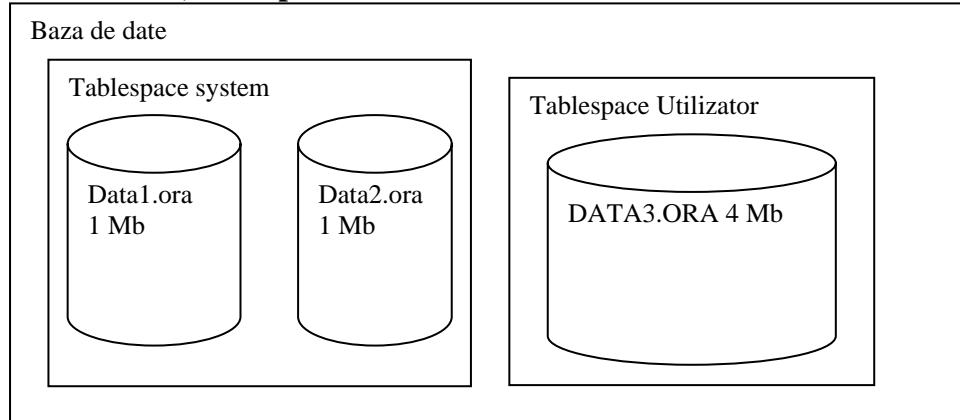
O baza de date Oracle este o colectie de date tratate unitar. Scopul general al bazei de date este memorarea si regasirea informatiilor. Baza de date are o structura logica si o structura fizica.

## Structura logica

### Spatiul pentru table (tablespace)

O baza de date este impartita in unitati logice de memorare numite spatii pentru table (tablespace). Tablespace se foloseste pentru gruparea structurilor logice care au o anumita legatura. De exemplu, tablespace grupeaza toate obiectele aplicatiei pentru a simplifica anumite operatii de administrare.

## Baze de date, tablespace si fisiere de date



- fiecare baza de date este impartita in una sau mai multe tablespace
- unul sau mai multe fisiere de date sunt create explicit pentru fiecare tablespace pentru a memora fizic datele tuturor structurilor logice in tablespace
- marimea combinata a fisierelor de date in totalul capacitatii tablespace (tablespace SYSTEM are 2KB pe cind tablespace USERS sunt 4KB)
- capacitatea de memorare combinata a tuturor tablespace bazei de date este capacitatea totala a bazei de date.

### Tablespace on line si off line

Tablespace poate fi on line (accesibila) sau off line (inaccesibila). Tablespace este on line in mod normal in asa fel ca utilizatorii pot accesa informatiile din tabela. Totusi, uneori tablespace trebuie facuta off line pentru a face o portiune a bazei de date indisponibila in timp ce se permite accesul utilizatorilor la partea ramasa a bazei de date. Aceasta face ca sarcinile administrative sa se desfasoare mai usor.

### Scheme si obiectele schemei

Schema este o colectie de obiecte ale bazei de date. Obiectele schemei sunt o structura logica ce refera direct datele bazei de date.

Obiectele schemei includ asemenea structuri cum ar fi tabele, view-uri, secvente, proceduri memorate, sinonime, indecsi, clustere si baze legate. (Acestea nu sunt relatii intre tablespace si schema; obiectele din aceeasi schema pot fi in tablespace diferite si tablespace poate contine obiecte din scheme diferite).

### Blocuri de date, extenturi si segmente

Oracle controleaza folosirea spatiului pe disc prin structura logica de memorare care include blocuri de date, extenturi si segmente.

**Blocurile de date Oracle** - sunt cele mai mici elemente ale bazei de date. Datele bazei de date Oracle sunt memorate in blocuri de date. Un bloc de date corespunde unui numar specific de bytes al spatiului fizic de pe disc. Marimea unui bloc de date este specificata pentru fiecare baza de date cind aceasta este creata. Baza de date Oracle foloseste si aloca spatiul liber al bazei de date in blocuri de date Oracle.

**Extentul** - urmatorul nivel logic al bazei de date este extentul. Un extent este un numar specific de blocuri de date contigute, obtinute intr-o singura alocare, folosite sa memoreze un tip de informatii specific.

**Segmentul** - nivelul logic peste extent este segmentul. Segmentul este un set de extenturi alocate pentru o anumita structura logica. De exemplu, diferitele tipuri de segmente includ urmatoarele:

**segment de date** - fiecare tabela non-cluster are un segment de date.

Toate datele tablei sunt memorate in extenturile acestui segment. Fiecare cluster are un segment de date. Datele fiecarei table din cluster sunt memorate in segmentul de date al clusterului.

**segment index** - fiecare index are un segment care memoreaza toate datele acestuia.

**segmente rollback** - pentru o baza de date sunt create de catre administratorul bazei de

date una sau mai multe segmente rollback pentru memorarea temporara a informatiilor de refacere. Aceste informatii sunt folosite pentru:

- generarea citirii consistente
- in timpul refacerii bazei de date
- pentru a anula tranzactii nerealizate

**segmente temporare** - un segment temporar este creat de catre Oracle

cind o comanda SQL necesita spatiu temporar de lucru pentru executia completa. Cind comanda termina executia, extenturile segmentului temporar sunt returnate sistemului de operare pentru a fi folosite ulterior.

Oracle aloca dinamic spatiu cind extenturile existente ale unui segment devin pline. Totusi cind extenturile existente ale unui segment sunt pline, Oracle aloca un alt extent pentru acest segment daca este nevoie. Deoarece extenturile sunt alocate la nevoie, extenturile unui segment pot fi contigue sau nu.

## Structura fizica a bazei de date

### Fisierele de date

Oricare baza de date Oracle are fizic una sau mai multe fisiere de date. Un fisier de date al bazei de date contine toate datele bazei de date. Datele structurii logice cum ar fi tabelele si indecsii sunt memorate fizic in fisierele de date alocate pentru baza de date.

Caracteristicile fisierelor de date sunt:

- un fisier de date poate fi asociat numai cu o baza de date
- fisierele de date pot avea anumite caracteristici stabilite care le permit extinderea automata cind baza de date ruleaza in afara spatiului
- unul sau mai multe fisiere de date ale bazei de date formeaza o unitate de memorare a bazei de date numita spatiu pentru tabele.

### Folosirea fisierelor de date

Datele din fisierele de date sunt citite, la nevoie, in timpul operatiilor normale a bazei de date si memorate in memoria cache Oracle. De exemplu, se presupune ca un utilizator doreste sa acceseze anumite date dintr-o tabela a bazei de date. Daca informatia ceruta nu este deja in memoria cache a bazei de date, ea este citita din fisierul de date corespunzator si memorata in memoria cache. Datele modificate sau datele noi nu este necesar sa fie scrise in memorie imediat. Pentru a reduce numarul de accesari ale discului si cresterea performantelor, datele sunt tinute impreuna in memorie si scrise in fisierul de date corespunzator toate odata, la un moment determinat de procesul background Oracle DBWn.

### Fisierele redo-log

Fiecare baza de date are două sau mai multe fisiere redo-log. Setul de fisiere redo-log pentru o baza de date este cunoscut împreună ca redo-logul bazei de date. Prima funcție a redo-log este să înregistreze toate schimbările facute în date. Dacă o eventuală defectiune impiedică ca datele modificate să fie scrise permanent în fisierele de date, modificările pot fi obținute din redo-log și lucru nu este pierdut.

Fisierele redo log sunt critice în protecția bazei de date față de defectiuni. Pentru protecția la defectiuni a fisierelor redo-log însăși Oracle furnizează multiplicarea redo-log astfel încât două sau mai multe copii redo-log pot fi menținute pe discuri diferite.

### **Folosirea fisierelor redo-log**

Informatiile din fisierele redo-log sunt folosite numai pentru refacerea datelor dintr-un sistem sau mediu defect care impiedică datele să fie scrise în fisierele de date ale bazei de date. De exemplu, dacă o cadere de tensiune întrerupe brusc operațiile bazei de date, datele din memorie nu pot fi scrise în fisierele de date și acestea sunt pierdute. Totuși, oricare date pierdute poate fi refacuta cind baza de date este deschisa prin aplicarea informatiilor din cel mai recent fisier redo-log la fisierele de date ale bazei de date, Oracle refac baza de date la momentul caderii tensiunii.

### **Fisierele de control**

Fiecare baza de date Oracle are un fisier de control. Un fisier de control conține intrari sau specifică structura fizică a bazei de date. De exemplu, acesta conține urmatoarele tipuri de informații:

- numele bazei de date
- numele și localizarea fisierelor de date și a fisierelor redo-log ale bazei de date
- eticheta de timp a creerii bazei de date

Asemănător cu redo-log, Oracle permite ca fisierele de control să fie multiple pentru protejare.

### **Folosirea fisierelor de control**

De fiecare dată cind o instanță a bazei de date este startată, fisierul ei de control este folosit pentru identificarea bazei de date și fisierele redo-log care trebuie să fie deschise pentru operațiile următoare ale bazei de date. Dacă starea fizică a bazei de date este modificată (de exemplu un fisier de date nou sau un fisier redo-log se creează) fisierul de control este modificat automat de Oracle pentru a reflecta aceste schimbari. Fisierul de control este folosit și la refacerea bazei de date dacă aceasta este necesară.

### **Structura memoriei și procesele**

Serverul Oracle folosește structurile memoriei și procesele pentru gestionarea și accesarea bazei de date. Toate structurile memoriei există în memoria principală a calculatorului care constituie sistemul bazei de date.

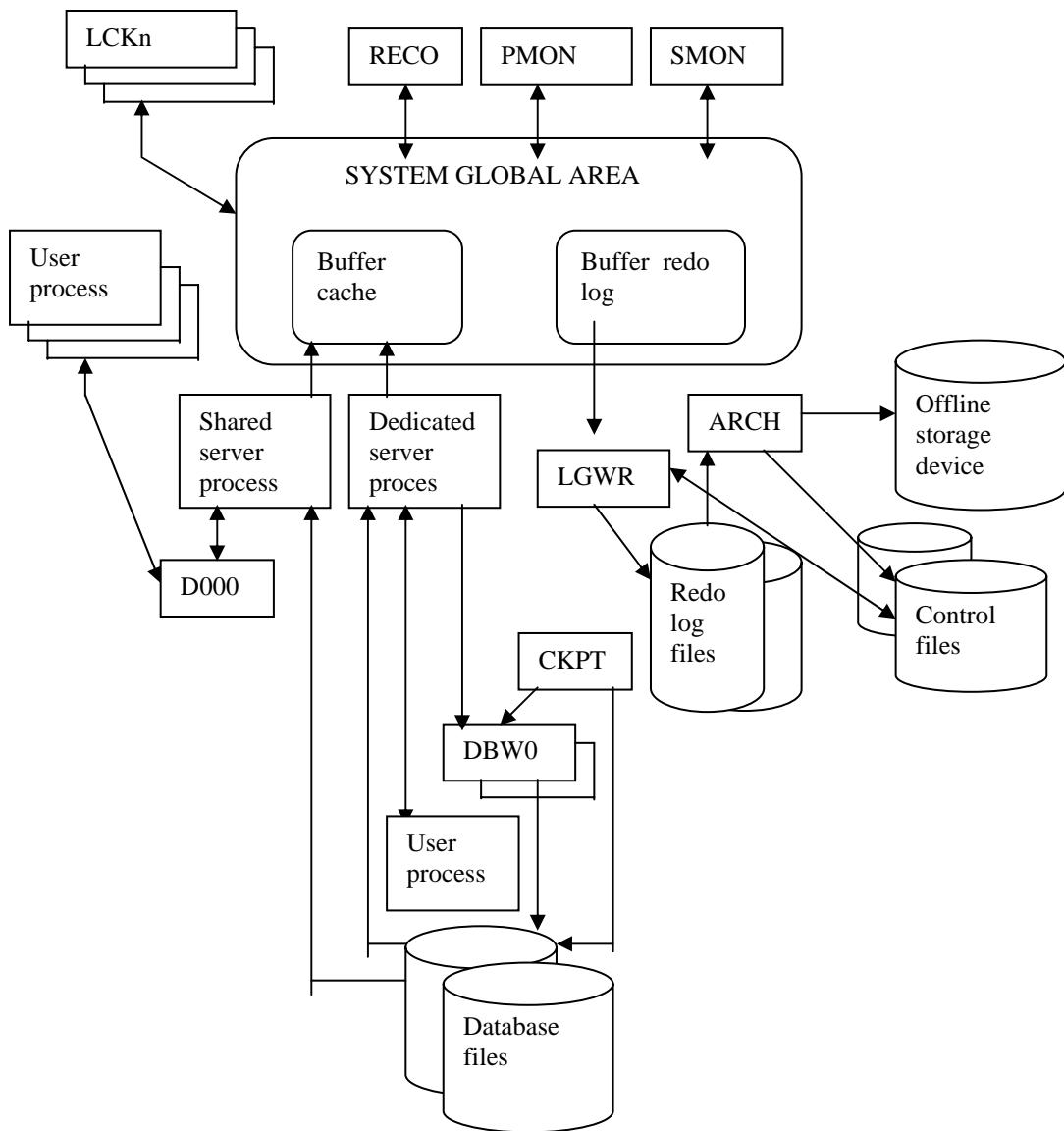
Procesele sunt joburi sau taskuri care lucrează în memoria acestui calculator.

### **Structura memoriei**

Oracle creează și folosește structuri de memorie pentru a efectua anumite joburi. De exemplu, memoria menține codul program care se executa și datele care sunt partajate între

utilizatori. Anumite structuri ale memoriei de baza sunt asociate cu Oracle: SGA (care include bufferele bazei de date, bufferele redo-log și shared pool) și PGA.

### Structura memoriei si procesele Oracle



### SGA - System Global Area

System Global Area (SGA) este o zona de memorie partajata care contine date si informatii de control pentru instanta Oracle. SGA si procesele Oracle background constituie instanta Oracle. Oracle aloca SGA cind o instanta starteaza si o dealoca cind instanta este oprita. Fiecare instanta are propria SGA.

Utilizatorii concurrenti conectati la serverul Oracle partajeaza date in SGA. Pentru performanta optima, intreg SGA trebuie sa fie atit de mare (pina cind incape in memoria reala) incit sa memoreze cit mai multe date posibil si sa minimizeze operatiile I/O cu discul.

Informatiile memorate in SGA sunt impartite in cteva tipuri de structuri de memorie, inclusiv bufferele bazei de date, bufferele redo-log si shared pool. Aceste zone au dimensiune fixa si sunt create in timpul startarii instantei.

### **Bufferul cache al bazei de date**

Bufferele bazei de date sunt o zona SGA care memoreaza cele mai recent utilizate blocuri ale bazei de date; setul de buffere ale bazei de date dintr-o instantă se numesc bufferul cache al bazei de date.

Bufferul cache contine atit blocuri modificate cit si blocuri nemodificate. Deoarece cel mai recent (si adesea cel mai frecvent) folosit bloc de date este tinut in memorie, scad operatiile I/O si cresc performantele.

### **Bufferul redo-log**

Zona SGA care memoreaza intrarile redo - o inregistrare a modificarilor facute in baza de date. Intrarile redo-log se memoreaza in bufferul redo-log si sunt scrise in fisierul redo-log, care este folosit daca este necesara refacerea bazei de date. Are dimensiune statica.

### **Shared pool**

Este portiunea SGA care contine structuri de memorie partajata cum ar fi zona partajata SQL. O zona partajata SQL este ceruta pentru procesarea fiecarei comenzi SQL lansate la o baza de date. Zona SQL contine informatii cum ar fi partea de analiza si planul de executie pentru comanda corespunzatoare. O singura zona SQL partajata este folosita de mai multe aplicatii care lanseaza aceeasi comanda lasind mai multa memorie partajata pentru alti utilizatori.

### **PGA**

Program Global Area este un buffer de memorie care contine date si informatii de control pentru procesul server. PGA este creata de Oracle cind este startat procesul server. Informatiile din PGA depind de configurarea Oracle.

### **Arhitectura proceselor**

Un proces este un mecanism in sistemul de operare care executa o serie de pasi. Anumite sisteme de operare folosesc termenul de job sau task. Un proces are in mod normal propria sa zona de memorie privata in care aceasta ruleaza.

Serverul Oracle are doua tipuri principale de procese: proces utilizator si proces Oracle.

#### **Proces utilizator (client)**

Un proces utilizator este creat si mentinut sa execute codul programului de aplicatii (ca de exemplu, program C++) sau un utilitar Oracle (exemplu, Oracle Enterprise Manager). Procesele utilizator gestioneaza comunicatia cu procesul server. Procesul utilizator comunica cu procesul server prin intermediul interfetei program.

#### **Arhitectura proceselor Oracle**

Procesele Oracle sunt apelate de alte procese sa execute functii pentru procesul invocat.

## Procesele server

Oracle creeaza procese server pentru a manipula cererile proceselor utilizatorilor conectati. Un proces server este insarcinat cu comunicarea cu procesul utilizator si interactioneaza cu Oracle pentru a trimite raspunsul cererii procesului utilizator asociat. De exemplu, daca un utilizator cere anumite date care nu sunt in bufferul bazei de date al SGA, procesul server asociat citeste blocurile de date corespunzatoare din fisierele de date in SGA. Oracle poate fi configurat sa varieze numarul proceselor utilizator per proces server. Intr-o combinatie server dedicat, procesul server manipuleaza cererile unui singur proces utilizator. O configuratie server multithread permite mai multor procese utilizator sa partajeze un numar mai mic de procese server, minimizind numarul proceselor server si maximizind utilizarea resurselor disponibile ale sistemului.

In anumite sisteme, procesul utilizator si server sunt separate, pe cind in altele sunt combinate intr-un singur proces. Daca un sistem foloseste server multithread sau daca procesul utilizator si procesul server ruleaza pe masini diferite, procesul server si procesul utilizator trebuie separate. Sistemul client/server separa procesele utilizator si server si le executa pe masini diferite.

## Procesele background

Oracle creeaza un set de procese background pentru fiecare instanta. Procesele background efectueaza operatii I/O asincrone si monitorizeaza alte procese Oracle care permit cresterea paralelismului pentru o mai buna performanta.

SGA si procesele background constituie instanta Oracle: DBW, LGWR, CKPT, SMON, PMON, ARCH, RECO, DARK, LCK, SNP, QMN.

### DBW ( Database Writer)

Scrie blocurile modificate din bufferul cache in fisierele de date. Cu toate ca un proces DBW este suficient pentru multe sisteme (DBW 0) se pot configura procese suplimentare (DBW1 la DBW0) pentru cresterea performantelor sistemului care modifica date multe sau frecvent. Parametrul de initializare DB\_WRITER\_PROCESSES specifica numarul proceselor DBW.

DBW nu necesita sa scrie blocurile cind se realizeaza o tranzactie. In schimb, DBW este proiectat sa efectueze scrieri cu o mare eficienta. In cele mai obisnuite cazuri, DBW scrie numai cind mai multe date trebuie citite in SGA si sunt prea putine buffere libere. Datele cel mai putin folosite sunt scrise in fisier prima data. DBW efectueaza scrieri pentru alte functii cum ar fi punctele de reluare.

### Log Writer (LGWR)

Scrie intrarile redo-log pe disc. Datele redo log sunt generate in bufferul redo log al SGA. Pe masura ce se realizeaza tranzactiile si bufferele log se umplu, LGWR scrie intr-un fisier redo log online intrarile redo log.

### Checkpoint (CKPT)

La un moment specific toate bufferele bazei de date din SGA sunt scrise in fisierele de date de catre DBW, acest eveniment este numit checkpoint (punct de reluare). Procesul Checkpoint este responsabil cu semnalarea DBW la checkpoint si actualizarea tuturor fisierelor de date si fisierelor de control ale bazei de date pentru a indica cel mai recent checkpoint.

### System Monitor (SMON)

Efectueaza refacerea instantei la startarea instantei. Intr-un sistem cu instante multiple (unul care foloseste Oracle Parallel Server), SMON al unei instante poate efectua si refacerea altor instante care au esuat. SMON mai sterge segmente temporare care nu au fost mult timp utilizate.

Aceste tranzactii sunt eventual refacute de SMON cind tablespace sau fisierul este adus inapoi in starea on line. SMON bricheteaza extenturile libere din baza de date pentru a face spatiu liber contiguu si usor de alocat.

### **Process Monitor (PMON)**

Executa refacerea procesului cind procesul utilizator a esuat. PMON este responsabil de stergerea cache-ului si eliberarea resurselor pe care procesul le-a utilizat. PMON cauta in dispecer si procese server si le restaureaza daca acestea au esuat.

### **Archiver (ARCH)**

Copiază fisierul redo log în memoria de arhivare cind acesta este plin. ARCH este activ numai cind baza de date este în mod ARCHIVELOG.

### **Recoverer (RECO)**

Este folosit să rezolve tranzactii distribuite care sunt nerezolvate datorate retelei sau caderii sistemului într-o baza de date distribuită. La intervale de timp, RECO local încearcă conectarea la bazele de la distanță și completează automat realizarea sau anularea portiunii locale ale oricărei tranzactii distribuite.

### **Dispatcher( Dnnn)**

Este un proces background optional prezent numai cind este folosita o configuratie multi-thread. Cel putin un proces dispecer este creat pentru oricare protocol de comunicatie folosit (D000, ..., Dnnn). Fiecare proces Dispatcher este responsabil cu transferarea cererilor de la procesele utilizatorilor conectati pentru a disponibiliza procesele server partajate si a returna inapoi raspunsul la procesul utilizator corespunzator.

### **Lock (LCK)**

Procesele lock (LCK0, ..., LCK9) sunt folosite pentru blocarea inter-instantelor în Oracle Parallel Server.

### **Job Queue (SNP)**

In configuratii baze de date distribuite, pina la 36 procese job queue (SNP0 ... SNP9, SNPA ... SNPZ) pot reimprospata automat tabelele snapshot (instantane). Procesele se activeaza periodic si reimprospateaza orice tabela snapshot care este programata pentru reimprospatarea automata. Daca sunt folosite mai multe procese job queue, procesele partajeaza sarcinile reimprospatarii snapshoturilor.

### **Queue monitor (QMNN)**

Queue monitor sunt procese background care monitorizeaza sirul de mesaje pentru Oracle Advanced Queing. Se pot configura pina la 10 procese monitor.

## **Interfata program**

Este mecanismul prin care procesul utilizator comunica cu procesul server. Acesta serveste ca o metoda a comunicarii standard intre orice utilitar client sau aplicatie (cum ar fi Oracle Forms) si softul Oracle. Aceste functii:

- actioneaza ca un mecanism de comunicare formatind cererile de date, transmitind date si urmarind si returnind erori
- executa conversia si translatarea datelor, in particular intre diferite tipuri de calculatoare sau tipuri de date la programe utilizator externe

### **Exemplu de lucru Oracle**

Ilustrarea unei configuratii Oracle in care procesul utilizator si procesul server asociat sunt pe masini separate (conectate prin retea).

1. O instanta ruleaza pe calculatorul in care este Oracle (adesea numit host sau server de baza de date).
2. Un calculator care ruleaza o aplicatie (o masina locala sau o statie de lucru client) ruleaza aplicatia intr-un proces utilizator. Aplicatia client incarca sa stabileasca o conexiune cu serverul folosind driverul corespunzator Net8.
3. Serverul ruleaza driverul Net8 corespunzator. Serverul detecteaza cererea de conectare de la aplicatie si creeaza un proces server (dedicat) pentru procesul utilizator.
4. Utilizatorul executa o comanda SQL si realizeaza tranzactia. De exemplu utilizatorul schimba numele unui rind intr-o tabela.
5. Procesul server receptioneaza comanda si cauta in shared pool pentru orice zona partajata SQL care contine o comanda SQL identica. Daca este gasita o zona partajata SQL, procesul server cauta privilegiile de acces ale utilizatorului pentru data ceruta si zona partajata SQL anterioara este folosita pentru procesarea comenzii; daca nu, este alocata o zona partajata SQL noua pentru comanda, in care aceasta va fi analizata si procesata.
6. Procesul server trimite orice valoare a datei necesara din fisierul de date (tabela) sau cea memorata in SGA.
7. Procesul server modifica date in SGA. Procesul DBW scrie blocul modificat permanent pe disc cind aceasta este facuta cu eficienta. Deoarece tranzactia se realizeaza, procesul LGWR inregistreaza imediat tranzactia in fisierul redo log.
8. Daca tranzactia s-a realizat cu succes procesul server trimite un mesaj prin retea catre aplicatie. Daca nu s-a realizat cu succes se transmite un mesaj de eroare corespunzator.
9. Pe parcursul intregii proceduri, ruleaza alte procese background, veghind asupra conditiilor care le cer interventia. In plus, serverul gestioneaza tranzactiile altor utilizatori si impiedica conflictele intre tranzactiile care cer aceleasi date.

### **Concurrenta si consistenta datelor**

#### **Concurrenta**

Principala grija a unui sistem de gestiune a bazelor de date multiuser este modul cum controleaza concurrenta sau accesul simultan la aceeasi data de catre mai multi utilizatori. Fara un control adekvat al concurentei, datele pot fi actualizate si schimbat necorespunzator compromitind integritatea datelor. Daca mai mult utilizatori accesaza aceeasi data, o cale de a gestiona concurrenta este de a face ca fiecare sa-si astepte rindul. Scopul sistemului de gestiune a bazelor de date este de a face aceasta asteptare inexistent sau neglijabila pentru fiecare utilizator. Toate comenziile de manipulare a datelor (DML) vor trebui efectuate astfel incit sa fie prevenita interactiunea distructiva intre tranzactiile concurente. Interactiunea distructiva este oricare interactiune care actualizeaza incorect date sau modifica incorect structura datelor.

Oracle rezolva aceste probleme folosind un tip variat de blocari si un modul de consistenta multiversiune. Aceste caracteristici sunt bazate pe conceptul de tranzactie.

### Citirea consistentă

Citirea consistentă, suportată de Oracle, face urmatoarele:

- garantează ca setul de date consultate de o comandă este consistent cu respectarea unui singur moment de timp și nu se schimbă în cursul executiei comenzi (citire consistentă la nivel de comandă)
- asigură că cititorii datelor nu așteaptă pentru scriere sau după alți cititori ai aceleiași date
- asigură că cei care scriu în baza de date nu așteaptă numai după altii care scriu dacă acestia încearcă să actualizeze rinduri identice în tranzacții concurente.

Modul simplu în care se înțelege implementarea Oracle a citirii consistentă este de a se imagina că fiecare utilizator operează o copie privată a bazei de date, de aici modelul de consistență multiversiune.

### Citirea consistentă, segmentele rollback și tranzacțiile

Pentru a gestiona modelul consistentă multiversiune, Oracle trebuie să creeze un set de date citire consistentă cind o tabelă este interogată (citită) sau actualizată (scrisă) simultan. Cind se întâlnesc o actualizare, valorile originale ale datelor schimbă prin actualizare sunt scrise în segmentul rollback al bazei de date. Atât timp cât aceste actualizări rămân parte a unei tranzacții nerealizate, oricare utilizator care mai tîrziu interoghează datele modificate vede valorile originale ale datelor - Oracle folosește informația curentă din SGA și informația din segmentul rollback pentru a construi o vizualizare consistentă ("read-consistent view") pentru datele tabelei interogate.

Numai cind o tranzacție este realizată schimbarile tranzacției sunt facute permanente. Comenzi lansate după ce tranzacția utilizatorului este realizată vad numai schimbarile facute prin intermediul tranzacției realizate. De notat că tranzacția este cheia strategiei Oracle pentru a furniza citirea consistentă.

Aceasta unitate a comenziilor SQL de realizare (sau nerealizare)

- dictează punctul de start pentru view-urile de citire consistentă generate în beneficiul cititorilor
- controlează cind datele modificate pot fi văzute de alte tranzacții ale bazei de date pentru citire sau actualizare

### Tranzacții read-only

Implicit, Oracle garantează citire consistentă la nivel de comandă. Aceasta stabilește că setul de date returnat de o singură cerere este consistent cu respectarea unui moment de tip. Totuși, în anumite situații, se poate cere citire consistentă la nivel de tranzacție - capacitatea de a rula mai multe cereri într-o singură tranzacție, toate fiind citiri consistente cu

respectarea aceluiasi moment de timp astfel incit cererile acestei tranzactii sa nu vada efectele intervenite prin tranzactiile efectuate.

Daca se doreste rularea unui numar de cereri intre tabele multiple si nu se doreste actualizarea se prefera tranzactiile read-only. Dupa indicarea ca tranzactia este read-only, se pot executa mai multe cereri intre orice tabele, cunoscind ca rezultatul oricarei cereri este consistent cu respectarea aceluiasi moment de timp.

### **Mecanismul de blocare**

Oracle foloseste blocarea pentru a controla accesul concurrent la date. Blocarea este un mecanism cu intentia de a preveni interactiunile distructive intre utilizatorii care acceseaza date Oracle.

Blocarea este utilizata pentru a realiza doua scopuri importante ale bazei de date:

- consistenta - asigura ca, datele unui utilizator sau schimbarile nu sunt efectuate (de alti utilizatori) pina cind utilizatorul nu termina cu datele.
- integritatea - asigura ca datele bazei de date si structurile reflecta toate schimbarile facute in acestea in secventa corecta.

Blocarea garanteaza integritatea datelor prin furnizarea accesului concurrent maxim la date pentru un numar nelimitat de utilizatori.

### **Blocarea automata**

Oracle executa blocarea automata si nu cere actiunea utilizatorului. Blocarea implicita intilnita pentru comenzi SQL la nevoie, depinde de actiunea ceruta.

Managerul de blocare Oracle automat blocheaza datele tabelei la nivel de rind. Prin blocarea datelor la nivel de rind, conflictele pentru aceleasi date sunt minimizate.

Managerul Oracle mentine cteva tipuri de blocare a rindurilor, depinzind de tipul operatiei care stabileste blocarea. In general sunt doua tipuri de blocare: blocare exclusiva si blocare partajata. Numai o blocare exclusiva poate obtine o resursa (cum ar fi un rind sau o tabela); totusi, mai multe blocari partajate pot obtine o singura resursa. Ambele blocari permit interogarea resursei blocate dar interzic alte activitati asupra resursei (ca actualizare sau stergere).

### **Blocarea manuala**

In anumite circumstante, utilizatorul poate modifica blocarea implicita. Oracle permite modificarea manuala a caracteristicilor blocarii automate la nivel de rind (prin prima cerere pentru rindurile care vor fi actualizate in comanda urmatoare) si la nivel de tabela.

### **Procesare distribuita si baze de date distribuite**

Lucrul calculatoarelor in retea devine din ce in ce mai obisnuit in mediile de calcul de zi cu zi de aceea sistemele de gestiune a bazelor de date trebuie sa fie capabile sa ia avantajele proceselor si capacitatilor de memorare distribuite.

### **Arhitectura client / server:Procesare distribuita**

Procesarea distribuita foloseste mai mult de un procesor pentru a imparti procesarea unui set stabilit de lucrari. Procesarea distribuita reduce incarcarea procesului la un singur procesor permitind diferitelor procesoare sa se concentreze la un subset de sarcini (taskuri), prin aceasta crescind capacitatatile si performanta sistemului.

O baza de date Oracle poate lua usor avantajele procesarii distribuite folosind arhitectura client/server. In aceasta arhitectura baza de date sistem este impartita in doua parti: front-end sau portiunea client si back-end sau portiunea server.

Portiunea client - este partea aplicatiei bazei de date care interactioneaza cu tastatura, monitorul si mousele; Nu are responsabilitati de accesare a datelor; concentreaza cererea, procesarea si prezentarea datelor gestionate de portiunea server. Statiile de lucru client pot fi optimizate pentru aceste lucrari. Ele nu necesita discuri de capacitatii mari dar beneficiaza de capacitatii grafice sporite.

Portiunea server - ruleaza softul Oracle si manipuleaza functiunile cerute pentru accesul concurrent si partajat la date. Portiunea server receptioneaza si proceseaza comenzi SQL si PL/SQL generate de aplicatia client. Calculatorul care gestioneaza partea server poate fi optimizat pentru realizarea acestuia: discuri de capacitate mare si procesoare rapide.

### **Baze de date distribuite**

O baza de date distribuita este o gestionare a bazelor de date in retea de catre mai multe servere de baze de date care apare utilizatorului ca o singura baza de date logica. Datele din toate bazele de date distribuite pot fi simultan accesate si modificate. Primul avantaj al bazelor de date distribuite este ca datele bazelor de date fizice separate pot fi combinate logic si potential accesibile tuturor utilizatorilor din retea.

Fiecare calculator care gestioneaza o baza de date dintr-o baza de date distribuita se numeste nod. Baza de date la care utilizatorul este conectat direct se numeste baza de date locala. Oricare baza de date suplimentara accesata de utilizator se numeste baza de date la distanta. Cind o baza de date locala acceseseaza o baza de date la distanta, baza de date locala este client pentru serverul de la distanta (arhitectura client / server).

In timp ce baza de date distribuita permite cresterea accesului la o mare cantitate de date prin retea, trebuie sa furnizeze abilitatea de a ascunde locatia datelor si complexitatea accesului prin retea. DBMS distribuit trebuie sa pastreze avantajele administrarii fiecarei baze de date locale ca si cind nu ar fi fost distribuita.

### **Transparenta locala**

Se intilneste cind locatia fizica a datelor este transparenta utilizatorilor si aplicatiilor. Anumite caracteristici Oracle, cum ar fi view-urile, procedurile si sinonimele pot furniza transparenta locatiei. De exemplu, un view care leaga tabele din anumite baze de date furnizeaza transparenta deoarece utilizatorul view-ului nu trebuie sa cunoasca unde este originea datelor.

### **Autonomia locatiei**

Inseamna ca fiecare baza de date care participa la o baza de date distribuita este administrata separat si independent de alte baze de date ca si cum baza de date nu ar fi intr-o retea. Cu toate ca fiecare baza de date poate lucra cu altele, aceasta este distincta, sistem separat care este intretinut individual.

### **Manipularea bazelor de date distribuite**

Arhitectura Oracle a bazelor distribuite suporta toate operatiile DML, incluzind cereri, inserari, actualizari si stergeri din bazele de date de la distanta. Pentru accesarea unei baze de date de la distanta este facuta o referire care include numele global al obiectului de la distanta - nu este necesara codificarea sau o sintaxa complexa pentru accesarea datelor.

De exemplu, pentru interogarea tablei EMP din baza de date de la distanta SALES, se refera numele global SELECT \* FROM emp @ sales;

### **Realizarea in doua faze**

Oracle furnizeaza aceeasi siguranta a consistentei in mediu distribuit ca si in mediu nedistribuit. Oracle furnizeaza aceasta siguranta folosind modelul tranzactional si mecanismul de realizare in doua faze.

Ca si intr-un sistem nedistribuit, tranzactiile vor fi planificate exact sa includa un set logic de comenzi SQL care se vor succede.

Mecanismul Oracle de realizare in doua faze garanteaza ca indiferent de tipul defectiunii intilnute, sistem sau retea, se realizeaza o tranzactie distribuita sau o anuleaza in toate nodurile implicate pentru mentinerea consistentei datelor intre baza de date distribuita globala.

### **Transparenta completa fata de utilizatori**

Mecanismul de realizare in doua faze este transparent fata de utilizatorii care lanseaza tranzactii distribuite. O comanda simpla COMMIT care denota sfirsitul unei tranzactii automat declanseaza mecanismul de realizare in doua faze pentru realizarea tranzactiei; nu se cere codificarea sau comanda sintactica complexa pentru a include tranzactia distribuita in scopul aplicatiei.

### **Refacere automata fata de defectiuni sistem sau retea**

Procesul background RECO rezolva automat rezultatul tranzactiilor distribuite dubioase (in dubiu), tranzactii distribuite in care realizarea a fost intrerupta de o eroare sistem sau retea. Dupa ce defectiunea a fost reparata si comunicarea restabilita, RECO de pe fiecare server Oracle local realizeaza sau anuleaza oricare tranzactie distribuita in dubiu consistent in toate nodurile implicate.

### **Capacitatea de subscriere optionala normala**

In eventualitatea unei defectiuni pe termen lung, Oracle permite fiecarui administrator local sa realizeze sau sa anuleze orice tranzactie distribuita care este in dubiu ca rezultat al unei erori. Aceasta optiune permite administratorului unei baze de date locale sa elibereze orice resursa locala care este retinuta indefinit ca rezultat al unei defectiuni pe termen lung.

### **Facilitati pentru refacerea distribuita**

Daca o baza de date trebuie refacuta de la un punct din trecut, Oracle faciliteaza administratorului bazei de date din alta locatie sa returneze bazele lor de date la momentul de timp anterior. Aceasta asigura ca baza de date globala ramane consistenta.

### **Replacarea tabelelor**

Sistemul de baze de date distribuite replica adeseori table de la distanta care sunt frecvent interogate de catre utilizatori. Avind aceste copii ale datelor din anumite noduri, baza de date distribuita nu trebuie sa trimita informatiile repetat in retea, prin aceasta ajutand maximizarea performantelor aplicatiilor.

Datele pot fi replicate folosind instantanee (snapshots) sau replicarea tabelelor master.

### **Oracle si Net8**

Net8 este mecanismul Oracle pentru interfata cu protocoale de comunicatii folosite de retea care faciliteaza procesarea distribuita si baze de date distribuite. Protocoalele de

comunicatie difera de transmitere si receptionare a datelor in retea. Intr-un mediu retea, un server Oracle de baze de date comunica cu statile de lucru client si alte servere de baze de date Oracle folosind softul Oracle Net8.

Net8 suporta comunicatiile in toate protocolele importante de retea, de la cele suportate in LAN-urile PC la cele folosite in calculatoarele mainframe. Folosind Net8, dezvoltatorii de aplicatii nu trebuie sa se concentreze asupra suportului comunicatiilor in aplicatie. Daca este folosit un nou protocol, administratorul bazei de date face anumite mici modificari pe cind aplicatia nu cere nici o modificare si continua sa functioneze.

## Operatii de startare si oprire

O baza de date nu este disponibila utilizatorilor pina cind serverul Oracle nu a startat baza de date si nu a deschis-o. Aceste operatii trebuie executate de administratorul bazei de date . Startarea bazei de date si aducerea acesteia in starea in care este disponibila pentru utilizatori se face in trei pasi:

1. Startarea unei instante a serverului Oracle
2. Montarea bazei de date
3. Deschiderea bazei de date

Cind serverul Oracle starteaza acesta foloseste fisierul de parametri de initializare. Acesti parametri specifica numele bazei de date, cantitatea de memorie alocata, numele fisierelor de control, alte limitari si alti parametri sistem.

Oprirea instantei si a bazei de date conectate se face in trei pasi:

1. Inchiderea bazei de date
2. Demontarea bazei de date
3. Oprirea instantei serverului Oracle

Oracle efectueaza automat toti trei pasi cind instanta este oprita.

## Securitatea bazei de date

Sistemele de baze de date multiutilizator, cum este Oracle, includ caracteristici de securitate care controleaza cum acceseaza utilizatori baza de date. De exemplu, mecanismele de securitate:

- previn accesul neautorizat la baza de date
- impiedica accesul neautorizat la obiectele schemei
- controleaza folosirea discului
- controleaza folosirea resurselor sistemului (cum ar fi CPU)
- auditeaza actiunile utilizatorilor

Asociat cu numele fiecarui utilizator al bazei de date este schema cu acelasi nume. Schema este o colectie logica de obiecte ale bazei de date (tabele, view-uri, sechete, sinonime, indecsi, clustere, proceduri, pachete si baze legate). Implicit, fiecare utilizator al bazei de date creeaza si acceseaza toate obiectele din schema orespondenta.

Securitatea bazei de date poate fi clasificata in doua categorii distincte: securitatea sistem si securitatea datelor.

Securitatea sistemului include mecanism care controleaza accesul si folosirea bazei de date la nivel sistem. De exemplu, securitatea sistem include:

- combinatia unui utilizator valid / parola
- cantitatea de spatiu pe disc disponibila schemei obiectelor utilizatorului
- limitele resurselor utilizatorului

Mecanismul de securitate cauta:

- daca utilizatorul este autorizat sa se conecteze la baza de date
- daca auditul bazei de date este activ

- care operatii sistem le poate efectua utilizatorul  
Securitatea datelor include mecanism care controleaza accesul si utilizarea bazei de date la nivelul obiectelor schemei. De exemplu:
- care utilizator are acces la un obiect specific al schemei si tipul de actiuni permise fiecarui utilizator in obiectele schemei (de exemplu utilizatorul SCOTT poate lansa comenzi SELECT si INSERT dar nu poate folosi comanda DELETE in tabela EMP)
- actiunile care sunt audite pentru fiecare obiect al schemei

### Mecanismele de securitate

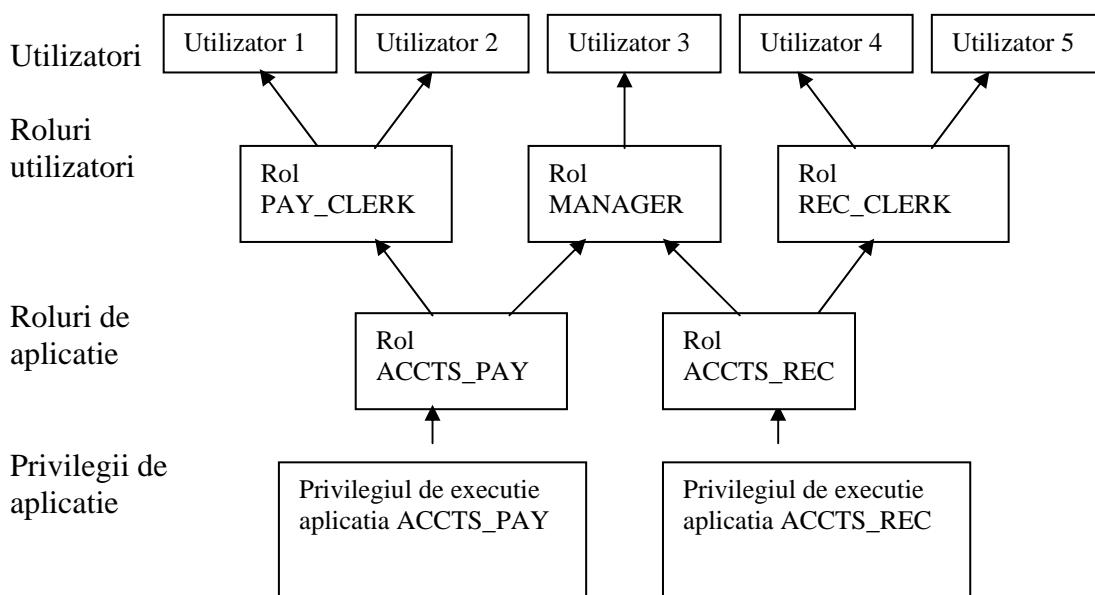
Serverul Oracle furnizeaza controlul accesului discretionar, care este un mod de a restrictiona accesul la informatii bazat pe privilegi. Privilegiile corespunzatoare trebuie asignate unui utilizator in ordinea in care utilizatorul acceseaza obiectele schemei.

Privilegiile corespunzatoare utilizatorului poate sa le acorde altor utilizatori privilegiati la discretionia lor; din acest motiv acest tip de securitate este numit discretionare.

Oracle gestioneaza securitatea bazei de date folosind diferite facilitati:

- utilizatorii bazei de date si schemele
- privilegi
- roluri
- setari de memorare si cote
- limite de resurse
- audit

Figura ilustreaza relatii intre diferite facilitati de securitate Oracle.



### Utilizatorii bazei de date si schemele

Fiecare baza de date Oracle are o lista cu numele utilizatorilor. Pentru a accesa o baza de date, un utilizator trebuie sa foloseasca o aplicatie si sa incerce sa se conecteze cu un nume valid de utilizator al bazei de date. Fiecare nume utilizator are asociata o parola pentru a impiedica accesul neautorizat.

## Securitatea domeniului

Fiecare utilizator are o securitate de domeniu, un set de proprietati care determina:

- actiunile (privilegiile si rolurile) disponibile utilizatorului
- cota de spatiu (spatiul disponibil) pentru utilizator
- limitele resurselor sistemului pentru fiecare utilizator (de exemplu timpul de procesare VC)

## Privilegiile

Un privilegiu este un drept sa execute un tip particular de comanda SQL. Anumite exemple de privilegii includ:

- dreptul de conectare la o baza de date (crearea unei sesiuni)
- dreptul de a crea o tabela in schema sa
- dreptul de a selecta rinduri din tabela altcuiva
- dreptul de a executa procedurile memorate ale altcuiva

Privilegiile bazei de date Oracle pot fi divizate in doua categorii distincte: privilegii sistem si privilegii ale obiectelor schemei.

Privilegiile sistem permit utilizatorilor sa execute o actiune particulara a sistemului sau o actiune particulara intr-un tip particular al schemei. De exemplu, privilegiul de crea un spatiu de tabela sau a sterge rinduri din orice tabela a bazei de date, este privilegiu sistem. Mai multe privilegii sistem sunt disponibile numai administratorului si dezvoltatorilor de aplicatii deoarece aceste privilegii sunt foarte puternice.

Privilegiile obiectelor schemei permit utilizatorilor sa execute o actiune particulara intr-un obiect specific al schemei. De exemplu, privilegiul de a sterge rinduri intr-o tabela specificata este un privilegiu obiect. Privilegiile obiect sunt acordate (asignate) utilizatorului final care poate folosi baza de date cu realizarea sarcinilor (taskurilor) specifice.

## Acordarea privilegiilor

Privilegiile sunt acordate utilizatorilor astfel ca utilizatorii pot accesa si modifica date in baza de date. Un utilizator poate obtine privilegii in doua feluri:

- privilegiile pot fi acordate utilizatorului explicit. De exemplu, privilegiul de a insera articole in tabela EMP poate fi acordat explicit utilizatorului SCOTT.
- privilegiile pot fi acordate rolului (un grup de privilegii) si rolul poate fi acordat la unul sau mai multi utilizatori. De exemplu, privilegiul de a insera articole in tabela EMP poate fi acordat rolului CLERK, care pe rind poate sa-l acorde utilizatorilor SCOTT si BRIAN.

Deoarece rolurile permit o mai usoara si imbunatatita gestiune a privilegiilor, privilegiile sunt normal acordate rolurilor si utilizatorilor specifici.

## Rolurile

Oracle furnizeaza o gestiune si control a privilegiilor prin intermediul rolurilor. Rolurile sunt grupuri de privilegii inrudite care sunt acordate utilizatorilor sau altor roluri. Urmatoarele proprietati ale rolurilor permit o mai usoara gestiune a privilegiilor:

- reducerea acordarii de privilegii - mai degraba decit acordarea aceluiasi set de privilegii la mai multi utilizatori, un administrator al bazei de date poate acorda privilegii pentru un grup de utilizatori inruditi atasati unui rol. Atunci administratorul bazei de date poate acorda rolul fiecarui membru al grupului.
- gestiunea dinamica a privilegiilor - cind privilegiile unui grup trebuie schimbat, numai privilegiul rolului trebuie modificat. Securitatea domeniului tuturor utilizatorilor acordati grupului rol reflecta automat schimbarile facute in rol

- disponibilitatea selectiva a privilegiilor - rolurile acordate utilizatorilor pot fi selectiv activate (disponibile utilizatorilor) sau dezactivate (indisponibile utilizatorilor). Aceasta permite un control specific al controlului privilegiilor utilizatorilor intr-o situatie data.

- constientizarea aplicatiilor - o aplicatie poate fi proiectata sa activeze sau sa dezactiveze selectiv automat rolurile cind un utilizator incerca sa foloseasca aplicatia.

Administratorul bazei de date acceseaza adesea roluri pentru o aplicatie. DBA acorda roluri aplicatiei toate privilegiile pentru rularea aplicatiei. DBA acorda rolul aplicatiei altor roluri sau utilizatori. O aplicatie care are anumite roluri diferite, fiecare acordand un diferit set de privilegii care permit mai usor sau mai greu accesul la date in timpul folosirii aplicatiei.

DBA poate crea un rol cu parola pentru a impiedica folosirea neautorizata a privilegiilor acordate rolului. Obisnuit, o aplicatie este proiectata astfel incit cind starteaza, aceasta activeaza rolul corespunzator. Ca rezultat, utilizatorul aplicatiei nu trebuie sa cunoasca parola pentru rolul aplicatiei.

### **Setari de memorare si cote**

Oracle furnizeaza un mijloc de a limita folosirea spatiului disc alocat bazei de date, incluzind tablespace temporar si cotele pentru tablespace.

#### **Spatiul pentru tabele (tablespace) implicit**

Oricare utilizator este asociat cu tablespace implicit. Cind un utilizator acceseaza o tabela, index sau cluster si nu este specificat tablespace care sa contine fizic obiectele schemei, tablespace implicit este folosit daca utilizatorul are privilegiul sa creeze obiectele schemei si o cota din tablespace implicit. Caracteristica tablespace implicit permite Oracle sa foloseasca direct spatiu in situatia in care locatia obiectelor schemei nu este specificata.

#### **Spatiul pentru tabele (tablespace) temporar**

Fiecare utilizator are tablespace temporar. Cind un utilizator executa o comanda SQL care cere crearea segmentelor temporare (cum ar fi crearea unui index) este folosit tablespace temporar. Prin directionarea tuturor segmentelor temporare ale utilizatorului la tablespace separat, tablespace separat poate reduce activitatilo I/O intre segmente temporare si alte tipuri de segmente.

#### **Cotele de spatiu pentru tabele (tablespace)**

Oracle poate limita cantitatea de spatiu total disponibila obiectelor din schema. Cotele (limitele de spatiu) pot fi stabilite pentru fiecare spatiu de tabela disponibil unui utilizator. Caracteristica de securitate a spatiului de tabela permite controlul selectiv ale cantitatii de spatiu disc care poate fi consumat de un obiect specific al schemei.

#### **Profile si limite de resurse**

Fiecarui utilizator ii este asigurat un profil care specifica limitarile anumitor resurse sistem disponibile utilizatorului, incluzind:

- numarul sesiunilor concurente pe care utilizatorul le poate stabili
- timpul de procesare CPU
  - disponibil utilizatorului
  - disponibil unui singur apel Oracle facut de o comanda SQL
- cantitatea de I/O logice
  - disponibile unei sesiuni utilizator
  - disponibile unui singur apel Oracle facut de o comanda SQL
- cantitatea de timp nefolosita permisa unei sesiuni utilizator

- cantitatea de timp de conectare permisa unei sesiuni utilizator
- restrictii de parola
  - blocarea contului dupa mai multe incercari de conectare nereusite
  - perioada de avizare si de gratie a parolei
  - reutilizarea parolei si restrictia complexa

Pot fi create diferite profile si asigurate individual fiecarui utilizator al bazei de date. Un profil implicit este prezent pentru toti utilizatorii asignati explicit unui profil. Caracteristica de limitare a resurselor impiedica consumul excesiv de resurse ale sistemului.

### Auditarea

Oracle permite auditul selectiv (monitorizarea inregistrata) a actiunilor utilizatorului in ajutorarea investigarii utilizarii suspicioase a bazei de date. Auditul poate fi facut la trei nivele diferite:

- auditul comenzi - este auditul specific comenzi SQL fara sa priveasca un obiect specific numit de schema. (In plus, declarat�orele bazei de date permit DBA sa extinda si sa personalizeze caracteristica intriseca de auditare). Auditul comenzi poate fi extins sa auditeze toti utilizatorii sistemului sau poate fi folosit pentru auditarea numai a utilizatorilor selectati ai sistemului. De exemplu, auditul comenzi pentru utilizator poate audita conectarile de la baza de date a utilizatorilor SCOTT si LORI.
- auditul privilegiului - este auditul celor mai puternice privilegii sistem fara sa priveasca un nume de obiect specific din schema. Poate fi extins sa auditeze toti utilizatorii sau focalizat numai la utilizatori selectati.
- auditul obiectului schemei - este auditul accesului la un obiect specific al schemei fara sa priveasca utilizatorii. Auditul obiectului monitorizeaza comenziile permise prin privilegiile obiectului cum ar fi comenziile SELECT sau DELETE pentru o tabela data.

Pentru toate tipurile de audit, Oracle permite auditarea executiei cu succes a comenzi, executia esuata sau ambele. Aceasta permite monitorizarea comenziilor capricioase indiferent daca utilizatorul care a lansat comanda are privilegiile necesare pentru lansarea comenzi. Rezultatul operatiei de auditare este inregistrat intr-o tabela referita ca audit trail.

View-urile predefinite ale auditului Trail sunt disponibile astfel incit se pot vedea usor inregistrările de auditare.

### Salvarea si restaurarea bazelor de date

- refacerea bazei de date ceruta de diferite tipuri de defectiuni
- operatii flexibile de refacere de efectuat in orice situatie
- disponibilitatea datelor in timpul operatiilor de salvare si restaurare in asa fel ca utilizatorii sistemului sa poata continua lucrul

### Importanta refacerii

In orice sistem de baze de date exista intotdeauna posibilitatea defectiunilor sistem sau hardware. Daca se intilneste o defectiune care afecteaza baza de date, baza de date trebuie refacuta. Scopul, dupa defectiune, este sa se asigure ca efectul tuturor tranzactiilor realizate este reflectat in baza de date refacuta si se revine la operatiile normale cit de repede este posibil.

### Tipuri de defectiuni

Anumite circumstante pot opri operatiile unei baze de date Oracle. Cele mai comune tipuri de defectiuni sunt:

- erori utilizator - erorile utilizator cer ca o baza de date sa fie refacuta la un moment de timp inaintea intilnirii erorii. De exemplu, un utilizator a sters accidental o tabela.

Pentru a permite refacerea fata de erorile utilizatorilor si corelarea cu alte cerinte de refacere, Oracle furnizeaza refacerea la un moment de timp exact. De exemplu, daca un utilizator sterge accidental o baza de date, baza de date poate fi refacuta la momentul de timp dinainte ca baza sa fi fost stearsa.

- defectiuni de comanda si procese - defectiunile de comanda se intilnesc cind se manipuleaza gresit o comanda intr-un program Oracle (de exemplu comanda nu este o constructie SQL valida). Cind se intilneste o comanda Oracle gresita, efectele comenzii sunt automat anulate de Oracle si controlul este returnat utilizatorului.

O defectiune proces este o defectiune a procesului utilizator care acceseara Oracle, cum ar fi o deconectare sau terminare anormala a procesului. Procesul defect nu poate continu sa lucreze, cu toate ca Oracle si alti utilizatori pot. Procesul background PMON detecteaza automat procesul utilizator defect sau este informat despre aceasta de SQL NET. PMON rezolva aceasta problema anulind tranzactia nerealizata a procesului utilizator si eliberind resursa pe care procesul utilizator a folosit-o.

Problemele comune cum ar fi comenzile SQL eronate sau procesele utilizator abortate nu vor opri niciodata baza de date in intregime.

Pe deasupra Oracle efectueaza automat refacerea necesara din tranzactiile nerealizate si blocheaza resursele cu un impact minim asupra sistemului sau altor utilizatori.

- defectiunea instantei - defectiunea instantei se intilneste cind apare o problema care impiedica instanta (SGA si procesele background) sa continue lucru. Defectiunea instantei poate aparea dintr-o problema hardware cum ar fi caderea tensiunii sau dintr-o problema software cum ar fi caderea sistemului de operare. Cind o instanta se defecteaza, datele din bufferele SGA nu sunt scrise in fisierele de date.

Defectiunea instantei cere refacerea instantei. Refacerea instantei este facuta automat de Oracle cind instanta este restartata. Redo log este folosit pentru refacerea datelor din bufferele SGA care au fost piedute cind o instanta s-a defectat.

- defectiunea discului - o eroare care se intilneste cind se incercă citirea sau scrierea unui fisier care este cerut sa opereze baza de date. Aceasta este denumita defectiune disc deoarece este o problema fizica cu citirea sau scrierea fizica a fisierelor pe disc. Cel mai comun exemplu este cind crapa fata discului, ceea ce aduce la pierderea tuturor fisierelor de pe disc. Fisierele diferite pot fi afectate de acest tip de defectiuni ale discului, fisiere de date, fisiere redo log si fisiere de control. De asemenea, deoarece instanta nu poate functiona corespunzator, datele din bufferele SGA nu pot fi scrise permanent pe disc in fisiere de date.

O defectiune disc necesita refacerea suportului. Refacerea suportului refac fisierele de date ale bazei de date in asa fel ca informatiile sa corespunda cu cel mai recent moment dinaintea defectiunii discului, incluzind datele realizate in memorie care au fost pierdute din cauza defectiunii. Pentru completarea refacerii in cazul in defectiunii discului, sunt cerute urmatoarele: salvarea anterioara a bazei de date, toate fisierele redo log si arhivele redo log.

Oracle permite refacerea completa si rapida pentru toate tipurile posibile de defectiuni hardware. Sunt oferite optiuni astfel incit baza de date sa poata fi refacuta complet sau partial la un moment specific. Daca anumite fisiere sunt striccate dintr-o defectiune disc dar restul bazei de date este intacta si operationala, baza de date poate ramane deschisa pina cind tabela ceruta este refacuta individual. Prin urmare, portiunea intacta a bazei de date este disponibila utilizatorilor pentru folosirea normala pina cind portiunea defecta este refacuta.

## Structuri folosite pentru refacerea bazei de date

Oracle foloseste cteva structuri pentru a permite refacerea completa in cazul unei defectiuni disc sau instantă: redo log, segmentele roll back, un fisier de control si daca este necesara salvarea anterioara a bazei de date.

### **Redo-log**

Este un set de fisiere care protejeaza datele modificate, in memorie, care nu au fost scrise pe disc in fisier de date. Redo log poate contine doua parti: redo log online si redo log arhiva.

#### **Redo-log online**

Este un set de doua sau mai multe fisiere redo-log online care inregistreaza toate salvarile realizate facute in baza de date. Ori de cate ori o tranzactie este realizata, intrarile corespunzatoare redo memorate temporar in bufferele redo log ale SGA sunt scrise in fisierele redo log de catre procesul background LGWR.

Fisierele redo log sunt folosite intr-o maniera ciclica; de exemplu, daca redo log este constituit din doua fisiere, primul fisier este umplut, al doilea este umplut, primul este refolosit si umplut, al doilea este refolosit si umplut si asa mai departe. De fiecare data cind un fisier este umplut, i se asigneaza un numar de secventa care identifica setul de intrari redo.

Pentru a evita pierderile bazei de date facute la un singur moment al defectiunii, Oracle poate mentine mai multe seturi redo log. O multiplicare redo log online consta in copii ale fisierelor redo log pe discuri fizice separate; schimburile facute intr-un membru al grupului sunt reflectate in toti membrii.

Daca un disc care contine un fisier redo log se defecteaza, alte copii raman intacte si disponibile pentru Oracle. Operatiile sistemului nu sunt intrerupte si fisierele redo log pierdute pot fi usor refacute folosind o copie intacta.

### **Redo log arhiva**

Optional, fisierele redo log pline pot fi arhivate inainte de a fi reutilizate creand un redo log arhiva. Fisierele redo log arhivate constituie arhiva redo log.

Prezenta sau absenta arhivei redo log este determinata de modul in care este folosit redo log.

ARCHIVELOG - fisierele redo log pline sunt arhivate inainte ca sa fie refoloseite ciclic.

NOARCHIVELOG - fisierele umplute nu sunt arhivate. In modul ARCHIVELOG, baza de date poate fi refacuta complet in ambele cazuri: defectiunea instantei sau a discului. Baza de date poate fi salvata cind este deschisa si disponibila utilizatorilor. Cu toate acestea, operatii administrative suplimentare sunt cerute pentru mentinerea arhivei redo log. Daca baza de date opereaza NOARCHIVE, baza de date poate fi refacuta complet fata de defectiunea instantei dar nu fata de defectiunea discului. In plus, baza de date poate fi salvata numai cind este inchisa complet. Deoarece nu se creeaza arhiva redo log, nu este cerut lucru suplimentar pentru administrator.

### **Fisierele de control**

Fisierele de control mentin, printre altele, informatii despre structura fisierelor bazei de date si secventa redo log curenta scrisa de LGWR. In timpul procedurii de refacere, informatia din fisierul de control este folosita sa coordoneze automat procesul de refacere.

### **Multiplicarea fisierelor de control**

Aceasta caracteristica este similara cu multiplicarea redo log: un numar de fisiere de control identice pot fi mentinute de Oracle, care le actualizeaza pe toate simultan.

### Segmentele rollback

Segmentele rollback inregistreaza informatiile rollback folosite de anumite functii Oracle. In timpul refacerii bazei de date dupa ce toate schimbarile inregistrate in redo log sunt aplicat, Oracle foloseste informatiile segmentelor rollback pentru a reface tranzactiile nerealizate. Deoarece segmentele rollback sunt memorate in bufferele bazei de date, aceste informatii sunt protejate automat prin redo log.

### Salvarile bazei de date

Deoarece unul sau mai multe fisiere pot fi fizic stricate ca rezultat al defectiunii discului, refacerea mediului cere restaurarea fisierelor stricate din cea mai recenta operatie de salvare a bazei de date. Sunt anumite moduri de salvare a fisierelor bazei de date.

#### Salvarea intregii baze de date

O salvare a intregii baze de date este o operatie de salvare sistem a fisierelor de date, fisierelor redo log si a fisierelor de control. Salvarea completa se executa cind baza de date este inchisa si nedisponibila pentru utilizare.

#### Salvare partiala

O salvare partiala este o operatie de salvare a unei parti a bazei de date. Salvarea fisierelor unei tabele de spatiu sau a fisierelor de control este un exemplu de salvare partiala. Salvarea partiala este folosita numai cind baza de date opereaza in mod ARCHIVELOG.

O varietate de salvari partiale pot fi facute in concordanta cu strategia de salvare. De exemplu, se pot salva fisierele de date si fisierele de control cind baza de date este deschisa sau inchisa sau cind o tabela de spatiu specificata este online sau offline. Deoarece redo log opereaza in modul ARCHIVELOG, nu sunt necesare salvari suplimentare ale redo log; arhiva redo log este o salvare a fisierelor redo log pline.

### Pasii de baza pentru refacerea bazei de date

Depinzand de modul in care DBW scrie bufferele bazei de date, la un moment dat, un fisier de date poate contine blocuri modificate prin tranzactii nerealizate si poate sa nu contina blocuri modificate prin tranzactii realizate. Deci, dupa o defectiune pot rezulta doua situatii:

- blocuri continind modificari realizate care nu au fost scrise in fisierele de date astfel incit schimbarile apar numai in redo log. Deci, redo log contine date realizate care trebuie aplicate fisierelor de date.
- deoarece redo log poate contine date care nu au fost realizate, schimbarile tranzactiilor nerealizate aplicate redolog in timpul refacerii trebuie eliminate din fisierele de date.

Pentru rezolvarea acestor situatii, sunt doi pasi disponibili folositi de Oracle in timpul refacerii instantei sau mediului: rolling forward sau rolling back (rulare inainte si rulare inapoi).

### Rulare inainte

Primul pas al refacerii este rularea inainte care consta in refacerea in fisierele de date a tuturor schimbarilor inregistrate in redo log.

Rularea inainte se efectueaza complet astfel ca mai multe fisiere redo log sunt necesare pentru a aduce inapoi fisierile de date la momentul cerut. Daca toate informatiile redo sunt online, Oracle efectueaza refacerea automat cind baza de date starteaza. Dupa rularea inainte, fisierile de date contin toate schimbarile realizate si schimbarile nerealizate care au fost inregistrate in redo log.

### Rulare inapoi

Rularea inapoi este numai o jumitate de refacere. Dupa rularea inainte toate schimbarile care nu au fost realizate trebuie anulate. Dupa ce fisierele redo log au fost aplicate segmentele rollback sint folosite sa identifice si sa anuleze tranzactiile care nu au fost realizate si totusi inregistrate in redo log. Procesul este numit rulare inapoi (rolling back). Oracle completeaza acest pas automat.

### Managerul Recovery

Managerul Recovery este un utilitar Oracle care gestioneaza operatiile de salvare si restaurare.

Manager Recovery mentine un catalog de refacere (recovery catalog) obligatoriu, care contine informatii despre fisierele de backup si fisierele archivelog. Recovery Manager foloseste recovery catalog pentru a automatiza operatiile de restaurare si refacere a suportului de inregistrare.

Recovery catalog contine:

- informatii despre backupurile fisierelor de date si fisierelor archivelog
- informatii despre copiile fisierelor de date
- informatii despre fisierele redo log arhivate si copiile acestora
- informatii despre schema fizica a basei de date destinatie
- sechete de comenzi numite stored scripts

## Modelul Obiect-relational pentru gestionarea bazelor de date

Sistemele de gestiune a bazelor de date au evoluat de la modelul ierarhic la modelul relational. Modelul acceptat in cea mai mare masura este modelul relational. Oracle extinde modelul relational la modelul obiect-relational care face posibila memorarea unui model complex intr-o baza de date relationala.

### Modelul relational

Modelul relational are trei aspecte majore:

- structurile - structurile sunt obiecte definite (tabele, view-uri, indecsi) care memoreaza sau acceseaza datele unei baze de date. Structurile si datele continute in acestea pot fi manipulate prin operatii.
- operatiile - operatiile sunt actiuni definite clar care permit utilizatorilor sa manipuleze datele si structurile bazei de date. Operatiile bazei de date trebuie sa adere la un set predefinit de reguli de intregritate.
- reguli de intregritate - sunt legi care stabilesc ce operatiuni sunt permise datelor si structurilor bazei de date. Regulile de intregritate protejeaza datele si structurile bazei de date.

Sistemul de gestionare a bazelor de date relationale ofera beneficii cum ar fi:

- independenta intre memorarea fizica a datelor si structura logica a bazei de date
- accesarea variata si usoara a datelor
- flexibilitate completa in proiectarea bazei de date
- reducerea memorarii datelor si redundantei

### Modelul relational obiect

Modelul relational obiect permite utilizatorilor sa defineasca tipuri de obiecte, specificind structura datelor si metodele de operare a datelor si folosirea acestor tipuri de date intr-un model relational.

Un obiect are trei componente:

- numele - care serveste la identificarea unica a tipului de obiect
- atributele - care sunt tipuri de date definite implicit sau alte tipuri definite de utilizator
- metodele - care sunt functii sau proceduri scrise in PL/SQL si memorate in baza de date sau scrise in limbaj C si memorate extern.

Metodele implementeaza operatii specifice pe care aplicatiile le pot efectua asupra datelor. Fiecare tip de obiect are o metoda de constructie care face obiectul nou in concordanță cu tipul de date specificat.

### Scheme si obiectele schemei

Schema este o colectie de obiecte ale bazei de date care sunt disponibile unui utilizator. Obiectele schemei sunt structuri logice care referă direct datele bazei de date. Obiectele schemei includ structuri cum ar fi tabele, view-uri, sechete, proceduri memorate, sinonime, indexuri, clustere și tabele legate. (Nu există nici o relație între spațiul pentru tabele și schema; obiecte din aceeași schema pot fi în spații pentru tabele diferite și un spațiu pentru tabele pătează obiecte din scheme diferite).

### Tabla

Tabla este unitatea de bază a memorării datelor în baza de date Oracle. Tabelele bazelor de date contin toate datele accesibile utilizatorilor.

Datele tabelei sunt memorate în rinduri și coloane. Fiecare tabela este definită printr-un nume de tabela și un set de coloane.

Fiecarei coloane i se da un nume de coloană, un tip de date (CHAR, DATE sau NUMBER) și o dimensiune (care poate fi predeterminată de tipul datei, ca de exemplu DATA) sau scală și precizie (numai pentru tipul de date NUMBER). Odată ce o tabela este creată, rindurile valide de date pot fi inserate în tabela. Rindurile tabelei pot fi interogate, stăsește sau actualizate.

### View-uri

Un view este o prezentare personalizată datelor din una sau mai multe tabele. Un view poate fi gindit și ca "interrogare memorată" (stored query).

View-urile nu contin și nu memorează date, ei derivă aceste date din tabele în care își au baza referite ca tabele de bază ale view-ului. Tabelele de bază pot să rindă să fie tabele sau pot fi ele însăși view-uri.

Ca și tabelele, view-urile pot fi interogate, actualizate, inserate în ele, stăsește din ele cu anumite restricții. Toate operațiile efectuate în view-ul curent afectează tabelele de bază ale view-ului.

View-urile sunt folosite adesea astfel:

- furnizează un nivel de securitate suplimentar al tabelei prin restrictionarea accesului la un set predeterminat de rinduri sau coloane ale tabelei. De exemplu, un view al unei tabele poate fi creat în astfel încât coloanele sensibile (ca de exemplu informații despre salar) să nu fie incluse în definiția view-ului.
- ascunde complexitatea datelor. De exemplu, un singur view poate combina tabele de salarii pe 12 luni pentru a furniza date pe un an pentru raportare și analiză. Un singur view poate fi deosebit de creat pentru a crea o imbinare a coloanelor sau rindurilor care au o legătură în mai multe tabele. Totuși, view-ul ascunde faptul că datele curente provin din anumite tabele.
- simplifică comenzi pentru utilizator. De exemplu, view-urile permit utilizatorilor să selecteze informații din mai multe tabele fără să ceară utilizatorilor să cunoască cum se execută subcererile corelate.

- prezinta datele intr-o perspectiva diferita de a tablei de baza. De exemplu, view-urile furnizeaza un mod de redenumire a coloanelor fara a afecta tablele pe care view-ul este bazat.

- memoreaza cereri complexe. De exemplu, o cerere poate efectua calcule extensive cu datele tablei. Prin salvarea acestei cereri cu un view calculele sunt executate numai cind view-ul este cerut.

View-urile care sunt implicate intr-o combinatie (o comanda SELECT care selecteaza date din mai multe table) cu doua sau mai multe table pot fi actualizate numai in anumite conditii.

### Secvente

Secventele genereaza o lista de numere unice pentru coloane numerice ale tabelelor bazei de date. Secventele simplifica programarea aplicatiilor prin generarea automata a unei valori numerice unice pentru rindurile unuia sau mai multor table. De exemplu, se presupune ca doi utilizatori insereaza simultan noi rinduri ale angajatilor din tabela EMP.

Prin folosirea unei secvente care sa genereze un numar unic al angajatului pentru coloana EMPNO, nici unul dintre utilizatori nu va astepta pentru ca celalalt sa introduca urmatorul numar disponibil al angajatului. Secventa genereaza automat valoarea corecta pentru fiecare utilizator.

Numeralele de secventa sunt independente de table asfel ca aceeasi secventa poate fi folosita pentru una sau mai multe table. Dupa creare, o secventa poate fi folosita de numerosi utilizatori ca sa genereze un numar curent de secventa.

### Unitatile de program

Termenul de unitate de program este folosit sa refere o procedura memorata, functie, pachet, declansator si blocuri anomite.

O procedura sau functie este un set de comenzi SQL si PL/SQL grupate impreuna ca o unitate executabila sa execute o sarcina specifica.

Procedurile si functiile permit sa se combine usurinta si flexibilitatea SQL cu functionalitatile procedurale ale limbajului de programare structurat. Folosind PL/SQL, asemenea proceduri si functii pot fi definite si memorate in baza de date pentru folosire continua. Procedurile si functiile sunt identice cu exceptia ca functia returneaza intotdeauna o singura valoare apelantului in timp ce procedura nu returneaza valoare apelantului.

Pachetele furnizeaza o metoda de incapsulare si memorare a procedurilor inrudite, functiilor noi sau altor pachete construite impreuna ca o unitate in baza de date. Pachetele furnizeaza beneficii administratorului bazei de date sau dezvoltatorului de aplicatii in paralele cu cresterea functionalitatii si performantelor bazei de date.

### Sinonime

Sinonimul este un alias pentru o tabela, view, secventa sau unitate de program. Un sinonim nu este un obiect al schemei, dar in schimb este o referire directa la obiectul schemei. Este folosit pentru:

- marcarea numelui real si proprietarului obiectului schemei
- permite accesul public la obiectul schemei
- permite transparenta locatiei pentru table, view-uri sau unitati de programe fata de baza de date de la distanta
- simplifica comenziile SQL pentru utilizatorii bazei de date

Sinonimul poate fi public sau privat. Un utilizator individual poate crea un sinonim privat care este disponibil numai acestui utilizator. Administratorul bazei de date creeaza cel mai

adesea sinonime publice care fac obiectele de baza ale schemei disponibile pentru uzul general al sistemului si al oricarui utilizator al bazei de date.

### **Indecsi, clustere si clustere hash**

Sunt structuri optionale asociate cu tabelele care sunt create pentru a creste performantele regasirii datelor.

Indecsi sunt creati sa creasca performanta regasirii datelor. Indexul ajuta la localizarea informatiei specifice mai repede. Cind se proceseaza o cerere, Oracle cauta anumite sau toate indecsurile disponibile pentru a localiza mai eficient rindurile curente. Indecsi sunt folositi cel mai mult cind aplicatiile interogheaza adesea tabele cu un anumit criteriu al rindurilor (de exemplu, salariatii cu salariul mai mare decat o valoare specificata) sau un rind specific.

Indecsi sunt creati in una sau mai multe coloane ale tabelei. Odata creat, un index este automat mentinut si folosit de Oracle. Schimbrile in datele tabelei (adaugare de noi rinduri, actualizarea sau stergerea rindurilor) sunt automat incorporate in toti indecsii cu transparenta completa fata de utilizator.

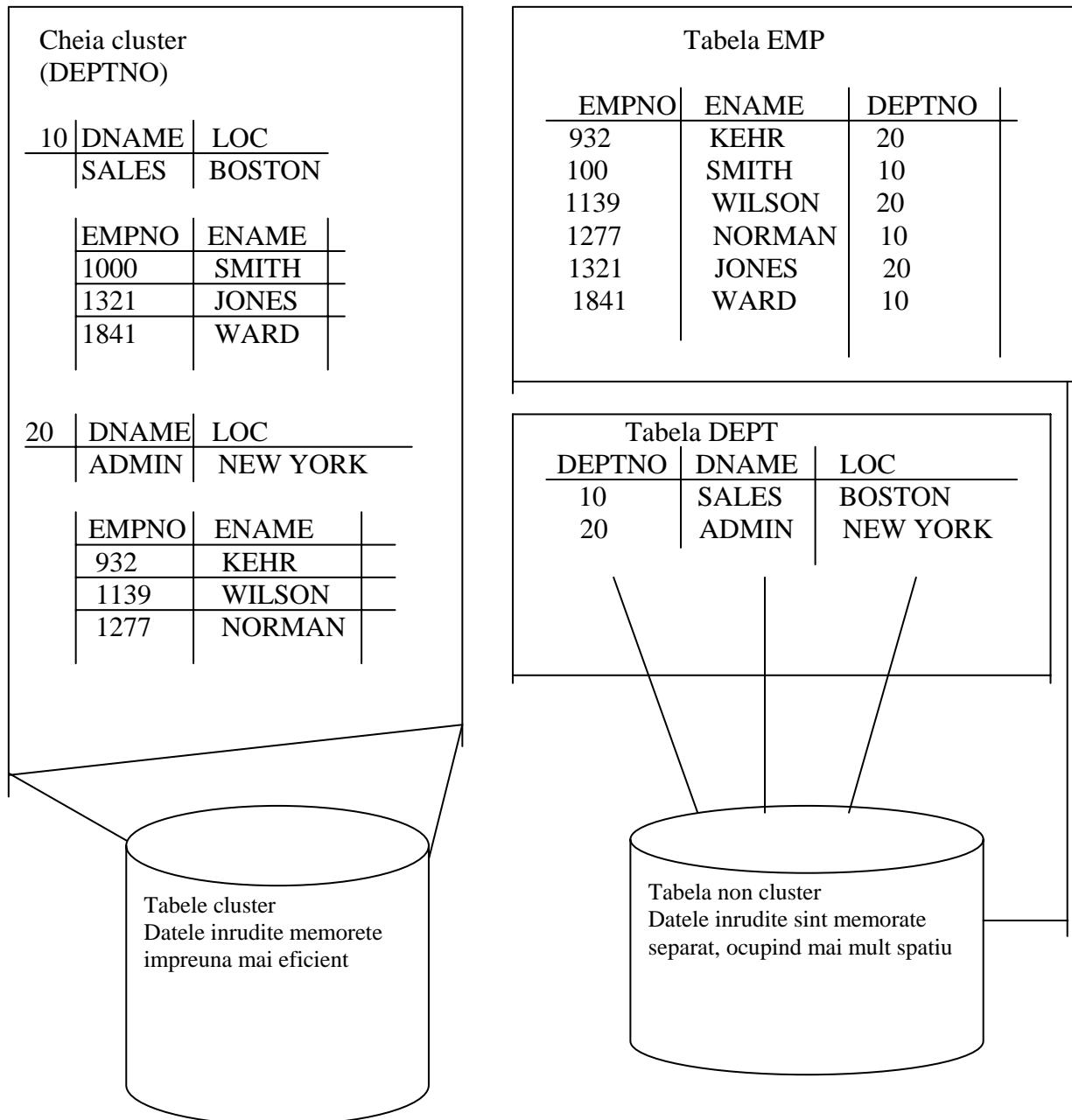
Indecsi sunt logic si fizic independenti de date. Acesteia pot fi stersi sau creati in orice moment fara sa afecteze tabelele sau alti indecsi. Daca un index este sters, aplicatia continua sa functioneze; totusi accesul la datele anterior indexate poate fi mai lent.

Oracle 8 permite partitionarea indecsilor.

Clusterele sunt o metoda optionala de a memora tabele. Clusterul este un grup de una sau mai multe tabele fizic memorate impreuna deoarece acestea partajeaza coloane comune si sunt adesea folosite impreuna. Deoarece rindurile cu legatura intre ele sunt memorate fizic impreuna, timpul de acces la ele este redus.

Coloanele intrudite ale tabelelor din cluster sunt numite cheia clusterului. Cheia clusterului este indicata in asa fel ca scade cantitatea de operatii I/O la minim. Deoarece datele din cheia cluster ale unui index cluster sunt memorate odata pentru mai multe tabele, clusterul memoreaza mai eficient decat daca tabelele ar fi memorate individual.

Tabele cluster si tabele non cluster



Clusterul poate deasemenea creste performanta regasirii datelor in dependenta de distributia datelor si a operatiilor SQL cel mai des efectuate asupra datelor. In particular, tabelele cluster care sunt interogate in legatura beneficiaza de folosirea clusterului deoarece rindurile comune ale tabelelor legate sunt regasite cu aceeasi operatie I/O.

Clusterul nu efectueaza aplicatiile. Faptul ca o tabela este sau nu parte a clusterului este transparenta utilizatorului si aplicatiei. Datele memorate in clustere sunt accesate prin SQL in acelasi mod in care sunt accesate datele memorate in tabele non cluster.

Clusterele hash ofera o cautare mai buna decit folosirea tabelei indexate sau a indecsului cluster cind o tabela este adesea interogata cu interogari de egalitate (de exemplu sa se returneze toate rindurile pentru departamentul 10). Pentru aceste cereri, valoarea specificata a

cheii clusterului este imbunatatita. Valoarea rezultata a cheii puncteaza direct zona din care memoreaza rindurile.

### **Legaturi intre baze de date**

Legaturile intre bazele de date sunt obiecte ale schemei care descriu o cale de la o baza de date la alta. Legaturile intre bazele de date sunt create implicit cind este facuta o referinta la un nume global de obiect intr-o baza de date distribuita.

### **Dictionarul datelor**

Fiecare baza Oracle are un dictionar al datelor. Un dictionar al datelor Oracle este un set de tabele si view-uri care sunt folositi ca referiri read-only asupra bazei de date. De exemplu, un dictionar al datelor memoreaza informatii despre structura logica si fizica a bazei de date. In plus la aceste informatii pretioase, un dictionar poate deasemenea sa memoreze informatii ca:

- utilizatori valizi ai bazei de date Oracle
- informatii despre restrictiile diferite pentru tabelele bazei de date
- cit spatiu este alocat pentru obiectele schemei si cit de mult este in uz

Un dictionar este creat cind este creata baza de date. Pentru a reflecta cu acuratete starea bazei de date in orice moment, dictionarul este automat actualizat de Oracle ca raspuns la actiunile specifice (cum ar fi modificarea structurii bazei de date). Dictionarul este critic pentru operatiile bazei de date. De exemplu, in timpul operarii bazei de date, Oracle citeste dictionarul pentru a verifica daca obiectele schemei exista si daca utilizatorii au drepturi corespunzatoare de acces.

### **Accesul la date**

Cerinte generale pentru DBMS:

- standard acceptat pentru limbajul de acces
- controlul si pastrarea consistentei informatiilor bazei de date in timpul manipularii
- furnizarea unui sistem de detinere a rolurilor pentru mentinerea integritatii informatiilor
- oferirea unei performante inalte

## **SQL**

SQL este un limbaj simplu si puternic de acces si este un limbaj standard pentru un sistem de gestiune a bazelor de date relationale.

O comanda SQL este un sir text care este transmis Oracle pentru a-l executa. O comanda SQL poate fi echivalata cu o propozitie completa SQL ca in

SELECT ename, deptno FROM emp;

Numai o comanda completa SQL poate fi executata, pe cind un fragment de propozitie, cum ar fi SELECT ename, genereaza o eroare indicand ca este nevoie de mai mult text inainte de a putea fi executata comanda.

Comenzile SQL se impart in urmatoarele categorii:

- comenzi ale limbajului de definire a datelor (DDL)
- comenzi de manipulare a datelor (DML)
- comenzi de control a tranzactiilor

- comenzi de control a sesiunii
- comenzi de control a sistemului
- comenzi SQL de incapsulare (embedded)

**DDL** - comenzile DDL definesc, intretin si sterg obiectele schemei cind acestea nu sunt necesare timp lung. Comenzile DDL includ deasemenea comenzi care permit utilizatorului sa acorde drepturi sau privilegii altui utilizator ca sa acceseze baza de date si obiecte specifice din baza de date.

**DML** - comenzile DML manipuleaza datele bazei de date. De exemplu, interogarea, inserarea, actualizarea si stergerea rindurilor unei tabele sunt toate operatii DML; blocarea unei tabele sau view si examinarea planului executiei unei comenzi SQL sunt deasemenea operatii DML.

**Comenzi de control a tranzactiilor** - sunt comenzi care gestioneaza schimbarile efectuate de comenzile DML. Acestea permit dezvoltatorului de aplicatii sa grupeze schimbarile intr-o tranzactie logica. Exemplele includ COMMIT, ROLLBACK si SAVEPOINT.

**Comenzi de control a sesiunii** - permit utilizatorului sa controleze proprietatile sesiunii curente, incluzind activarea si dezactivarea rolurilor si schimbarea setarilor limbajului. Cele doua comenzi de control a sesiunii sunt ALTER SESSION si SET ROLE.

**Comenzi de control al sistemului** - schimba proprietatile instantei server Oracle. Singura comanda este ALTER SYSTEM; aceasta permite schimbari cum ar fi stabilirea numarului minim de servere partajate, distrugerea unei sesiuni sau executia altor sarcini.

**Comenzi SQL de incapsulare** - aceste comenzi incorporeaza comenzi DDL, DML si de control a tranzactiilor intr-un program procedural (cum ar fi cele folosite cu Oracle Precompiler). Exemplele includ OPEN, CLOSE, FETCH, EXECUTE.

## Tranzactiile

Tranzactia este o unitate logica de lucru care cuprinde una sau mai multe comenzi SQL executate de un singur utilizator. In concordanță cu standardul ANSI/ISO SQL, cu care Oracle este compatibil, o tranzactie incepe cu prima comandă SQL executabilă. O tranzactie se termină cind este explicit realizată sau anulată de către utilizator.

Se consideră o bază de date bancară. Cind un cumpărător transferă bani din contul saving în contul cheking, tranzactia trebuie să conste în trei operații separate: scaderea contului saving, creșterea contului cheking și înregistrarea tranzacției în jurnalul de tranzacții.

Oracle trebuie să garanteze că toate trei comenzi sunt executate pentru întreținerea conturilor în balanță corespunzătoare.

```
UPDATE saving_accounts  
    set balance=balance-500  
    where account=3209;
```

```
UPDATE saving_accounts  
    set balance=balance+500  
    where account=3208;
```

```
INSERT INTO journal VALUES  
    (journal_seg.NEXTVAL,'1B'  
    3209,3208,500)
```

```
COMMIT WORK
```

### **Realizarea tranzactiilor si anularea (roll back)**

Schimbarile facute de comenzi SQL care constituie o tranzactie pot fi realizate sau anulate. Dupa ce o tranzactie este realizata sau anulata, urmatoarea tranzactie incepe cu urmatoarea comanda SQL.

Realizarea unei comenzi face permanente schimbarile rezultate din toate comenzi SQL ale tranzactiei. Schimbarile facute de comenzi SQL ale tranzactiei devin vizibile altor tranzactii ale sesiunii utilizator care starteaza numai dupa ce tranzactia este realizata.

Anularea unei tranzactii retrage oricare schimbari rezultate din comenzi tranzactiei. Dupa ce o tranzactie este anulata, date afectate sunt neschimbate ca si cum comanda SQL din tranzactie nu ar fi fost executata niciodata.

### **Puncte de salvare**

Pentru tranzactiile lungi, care contin mai multe comenzi SQL, marcaje intermedie sau puncte de salvare pot fi declarate cu rolul de a imparti o tranzactie in parti mai mici.

Prin utilizarea punctelor de salvare, se poate marca arbitrar lucrul la un punct oarecare intr-o tranzactie lunga. Aceasta permite optiunea unei anulari ulterioare executata de la punctul curent al tranzactiei la punctul de salvare declarat din tranzactie.

De exemplu, se poate folosi un punct de salvare intr-o serie complexa de actualizari, astfel ca daca se intampla o eroare nu trebuie sa se relanseze nici o comanda .

### **Consistenta datelor folosind tranzactiile**

Tranzactiile permit utilizatorului bazei de date sau dezvoltatorului de aplicatii compatibilitatea garantarii consistentei schimbării datelor, atit timp cit comenzi SQL dintr-o tranzactie sunt grupate logic. O tranzactie consta in toate elementele necesare ale unei unitati logice de lucru. Datele din toate tabelele referite sunt intr-o stare consistentă inainte de inceperea tranzactiei si dupa terminarea ei. Tranzactiile sunt consistentе numai daca comenzi SQL pe care le cuprind fac schimbari consistente in date.

De exemplu, operatiunea bancara. Un transfer intre doua conturi (tranzactia) poate include cresterea unui cont (o comanda SQL), descresterea altui cont (o comanda SQL) si inregistrarea in jurnalul tranzactiilor (o comanda SQL). Toate actiunile vor fi esuate sau se vor desfasura impreuna; creditul nu poate fi realizat fara debit. Alte actiuni diferite, cum ar fi un nou depozit in cont, pot sa nu fie incluse in transferul dintre conturi; aceste comenzi pot fi in alte tranzactii.

## PL/SQL

PL/SQL este limbajul procedural Oracle, extensia SQL.

PL/SQL combina usurinta si flexibilitatea SQL cu procedurile functionale ale programarii structurate cum ar fi IF...THEN, WHILE, LOOP.

Cind proiecteaza o aplicatie, dezvoltatorul trebuie sa considere avantajele folosirii PL/SQL memorat:

- deoarece codul PL/SQL poate fi memorat centralizat intr-o baza de date, traficul in retea dintre aplicatii si baza de date este redus, ceea ce duce la cresterea performantelor
- accesul la date poate fi controlat prin cod PL/SQL memorat. In acest caz, utilizatorii PL/SQL pot accesa date numai ca intentie a dezvoltatorului de aplicatie
- blocurile PL/SQL pot fi trimise de aplicatie la o baza de date, executind operatii complexe fara trafic excesiv in retea.

Cind PL/SQL nu este memorat in baza de date, aplicatiile pot trimite blocuri PL/SQL la baza de date mai degraba decit comenzi individuale SQL, prin acestea reducindu-se traficul in retea. Diferite unitati de program pot fi definite si memorate centralizat in baza de date.

## Proceduri si functii

Procedurile si functiile sunt un set de comenzi PL/SQL si SQL care sunt grupate impreuna ca o unitate sa rezolve o problema specifica sau sa execute un set de sarcini inrudite.

O problema este creata si memorata intr-o forma compilata in baza de date si poate fi executata de utilizator sau o aplicatie a bazei de date. Procedurile si functiile sunt identice cu exceptia ca functia returneaza o singura valoare apelantului in timp ce procedurile nu returneaza valori apelantului.

## Pachete

Pachetele furnizeaza o metoda de incapsulare si memorare a procedurilor inrudite, functiilor, variabilelor si alor pachete construite impreuna ca o unitate a bazei de date.

Pe linga faptul ca permit administratorului si dezvoltatorului abilitatea de a organiza asemenea structuri, acestea ofera si cresterea functionalitatii (de exemplu, variabilele globale impachetate pot fi declarate si folosite de orice procedura din pachet) si performantei (de exemplu, toate obiectele pachetului sunt analizate, compilate si incarcate in memorie o data).

## Declansatoarele bazei de date

Oracle permite scrierea procedurilor care sunt executate automat ca rezultat al inserarilor, stergerilor sau actualizarilor tabelei. Aceste proceduri se numesc declansatoare ale bazei de date.

Declansatoarele pot fi folosite intr-o varietate de moduri pentru gestionarea informatiei din baza de date. De exemplu, acestea pot fi folosite pentru generarea automata a datelor, auditarea modificarii datelor, fortarea restrictionarilor complexe.

## Metodele

O metoda este o procedura sau o functie care este parte a definitiei unui tip de date definit de utilizator (tip obiect, tabela imbriicata, tablou de variabile).

Metodele difera de procedurile memorate:

- metoda se invoca prin referire la un obiect al tipului asociat
- metoda are acces complet la atributele obiectului asociat si la informatiile despre acest tip

Fiecare tip de data definit de utilizator are o metoda de construire definita sistem, aceasta fiind o metoda care face un obiect nou in concordanta cu tipul de data specificat. Numele metodei de construire este numele tipului definit de utilizator. In cazul unui tip obiect, parametrii metodei de construire au tipul si metodele tipului obiect.

Metoda de construire este o functie care returneaza un obiect nou ca ca valoarea sa. Tabelele imbricate si tablourile de variabile au deasemenea metode de construire.

Metodele de comparare definesc o relatie de ordine intre obiectele unui tip de obiect dat.

Metoda map foloseste capacitatea Oracle de a compara tipuri implicate.

De exemplu, Oracle poate compara doua dreptunghiuri comparindu-le ariile daca un tip obiect denumit DREPTUNGHI are atributele LUNGIME si LATIME si se defineste o metoda map arie care returneaza un numar numit produsul atributelor LATIME si LUNGIME. O metoda ordin foloseste propria logica interna pentru a compara doua obiecte ale unui tip de obiect dat. Aceasta returneaza o valoare care codifica relatia de ordine. De exemplu, poate returna -1 daca primul este mai mic, 0 daca sunt egale si 1 daca primul este mai mare.

## **Integritatea datelor**

Este foarte important sa se garanteze ca datele adera la anumite reguli ale intreprinderii, determinate de administratorul bazei de date sau dezvoltatorul de aplicatii. De exemplu, se presupune ca regula este ca nici un rind din tabela INVENTORY nu poate contine o valoare mai mare decit 9 in coloana SALE-DISCOUNT. Daca o comanda INSERT sau UPDATE incerca sa incalce aceasta regula, Oracle trebuie sa anuleze comanda invalida si sa returneze o eroare in aplicatie. Oracle furnizeaza restrictii de integritate si declansatoare ca solutii pentru gestiunea regulilor de integritate a bazei de date.

## **Restrictii de integritate (constrangeri)**

O restrictie de integritate este un mod de a defini reguli pentru coloanele unei tabele. O restrictie de integritate este o declaratie asupra datelor tabelei care este intotdeauna adevarata:

- daca se creeaza o restrictie de integritate pentru o tabela si anumite tabele care exista nu satisfac restrictia, restrictia nu se poate aplica
- dupa ce restrictia s-a definit, daca rezultatul oricarei comenzi DML incalca restrictia, comanda este anulata si se returneaza o eroare.

Restrictia de eroare este definita ca o tabela si memorata ca parte a definitiilor tabelei in dictionarul bazei de date, astfel ca toate aplicatiile trebuie sa adere la acelasi set de regula. Daca o regula se schimba, este necesar sa fie schimbata o data la nivelul bazei de date si nu de fiecare data pentru fiecare aplicatie.

### Restrictiile suportate de Oracle

NOT NULL- dezactiveaza intrarile goale coloanele tabelei

UNIQUE - dezactiveaza valorile duble intr-o coloana sau set de coloane

PRIMARY KEY - dezactiveaza valorile duble si nule intr-o coloana sau set de coloane

FOREIGN KEY - cere ca fiecare valoare dintr-o coloana sau set de coloane corespunde unei valori cheie unica sau primara din tabela corespondenta. (Restrictia FOREIGN KEY defineste o actiune de integritate de referinta care dicteaza ce va face Oracle cu datele dependente daca datele acestei referinte sunt alterate)

CHECK - dezactiveaza valorile care nu satisfac expresia logica a restrictiei.

### **CHEI (KEYS)**

Termenul chei (keys) este folosit in definitiile unor anumite tipuri de restrictie de integritate. O cheie este coloana sau setul de coloane incluse in definitia unui anumit tip de restrictie. Cheile descriu relatiile intre diferitele tabele ale bazei de date si coloanele datelor de baze relationale.

Tipuri de chei:

primary key - coloana sau setul de coloane incluse in definitia restrictiei PRIMARY KEY.

O valoare primary key identifica unic rindurile unei tabele. Numai o primary key poate fi definita pentru o tabela

unique key - coloana sau setul de coloane incluse in definitia restrictiei UNIQUE

foreign key - coloana sau setul de coloane inclus in definitia unei restrictii de referire

referenced key - primary key sau unique key ale unei tabele sau a tabelelor diferite care sunt referite prin foreign key.

Valorile individuale ale cheilor se numesc valori cheie.

### **Declansatoarele bazei de date (TRIGGERE)**

Actiunile centralizate pot fi diferite folosind un mod de abordare nondeclarativ (scriere de cod PL/SQL) cu declansatoarele bazei de date. Un declansator este o procedura memorata care este initiată (implicit executată) cînd se lansează o comandă (INSERT, UPDATE sau DELETE) față de tabela asociată. Declansatoarele pot fi folosite să personalizeze sistemul de gestiune a bazelor de date care are caracteristici de auditare și introduce un complex de securitate și reguli de integritate. De exemplu, un declansator poate fi creat să permită modificarea tabelelor numai în timpul orelor normale de lucru ale organizației.

*Nota:* Declansatoarele permit definirea și aplicarea regulilor de integritate, declansatorul nu este identic cu regula de integritate; declansatorul definește și aplica o regula de integritate dar nu verifică datele introduse într-o tabelă. De aceea, este recomandat folosirea declansatoarelor numai cînd regulile de integritate nu pot fi aplicate prin restrictii de integritate.