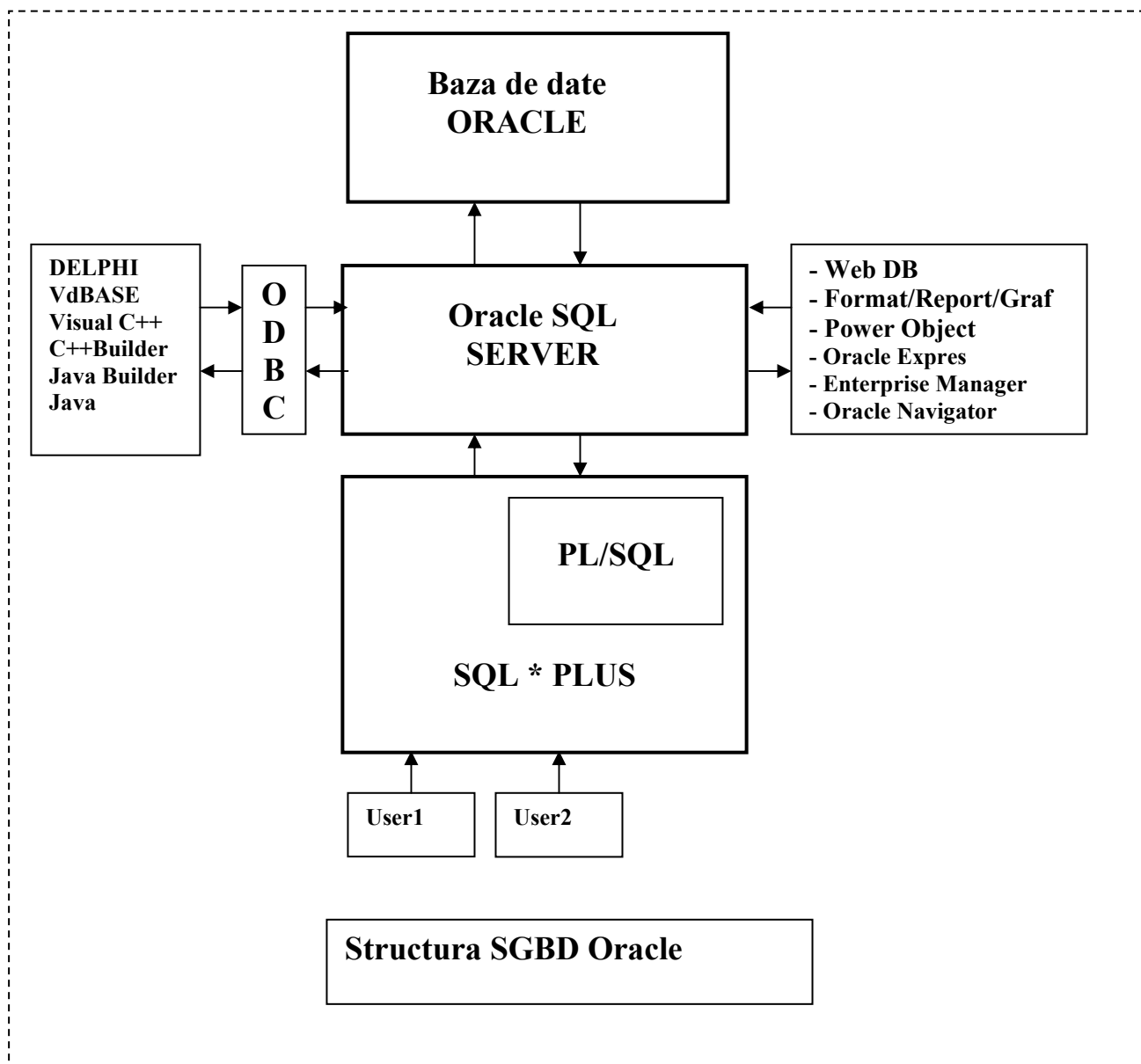


## Lucrarea 4.

### Comenzi SQL in Oracle 8i, 9i, 10g

#### Structura SGBD Oracle



**Baza de date Oracle** este o colecție de informații organizate sub forma de tabele pentru care se asigură stocarea, actualizarea și regăsirea informațiilor prin limbajul SQL. BD conține:

- Tabele sistem de control (Dicționar)
- Tabele utilizator care conțin într-o forma normală toate informațiile
- Vederi ale bazei de date, care sunt machete pentru vizualizarea tabelelor
- Proceduri, funcții și package-uri pentru prelucrarea informațiilor
- Obiecte definite de utilizator cu proprietățile și metodele lor.

**Limbajul SQL (Structured Query Language)** este limbajul de fundamental pentru gestiunea BD și este standardizat prin mai multe norme ISO (International Standard Organization) acceptat total sau parțial de toate Sistemele de Gestiune a Bazelor de Date (SGBD). SQL este un limbaj linie care are un număr redus de comenzi și un număr mare de clauze în Oracle și DB2/IBM .

**Oracle SQL Server** asigura accesul la baza de date pentru toate celelalte procese client:

- Realizează controlul accesului la informațiile din BD cu drepturi diferențiate pentru utilizatori.
- **Asigură execuția pentru toate comenzile SQL** care provin de la utilizatori prin SQL\*Plus sau de la alte programe Oracle sau non Oracle.
- Asigură integritatea informațiilor din BD prin controlul tranzacțiilor și protecția împotriva distrugerilor accidentale prin operații de rollback și arhivare.
- Păstrează toate informațiile de structură a obiectelor BD și asigură organizarea datelor și controlul condițiilor impuse de corectitudine (constrângeri)
- Asigura mecanisme de regăsirea rapidă a informațiilor pe baza conținutului lor folosind tehnici de indexare și randomizare și operații de combinare a datelor din mai multe tabele.
- Asigura accesul partajat în condiții de securitate a mai multor utilizatori la BD

**SQL \* Plus** este un mediu de acces la serviciile SQL Server pentru utilizatorii conectați la BD.

- Asigură editarea comenzilor și programelor SQL și PL/SQL, ce se execută pe Serverul SQL
- Lansează în execuție comenzile SQL și PL/SQL, care se execută pe serverul SQL și preia rezultatul, care este o tabelă asigurând interfața între serverul SQL și utilizatori
- Asigură setarea condițiilor de lucru pentru fiecare sesiune de lucru

**PL/SQL** este un server care asigură executarea programelor PL/SQL care conțin comenzi procedurale și comenzi SQL, într-un limbaj care extinde posibilitățile limbajului SQL.

- Se pot defini **variabile** și structuri de tip **tablouri indexate** și **record** în memorie.
- Permite definirea elementelor de tip **Cursor** pentru prelucrarea individuală și secvențială a înregistrărilor rezultate din interogările SQL
- Asigură definirea unor **programe, proceduri și funcții** stocate în BD
- Permite controlul informațiilor și accesului la nivel superior prin proceduri de tip **TRIGGER** asociate comenzilor de modificare.
- Permite definirea unor **clase de obiecte utilizator**, care permit implementarea conceptului de **baze de date relațional obiectuale**.

Accesul la PL/SQL se face de către utilizatori prin SQL\*Plus și nu direct.

În PL/SQL se execută numai comenzile proprii, iar **comenzile SQL care accesează baza de date se execută pe Serverul SQL**. PL/SQL are câteva sute de funcții și proceduri proprii.

#### **Intefața cu utilizatorii**

**Toate comenzile SQL se execută pe Serverul SQL**, care execută toate controalele de securitate și asigură integritatea datelor. Comenzile SQL pot proveni de la utilizatori conectați la:

- SQL\*Plus care pot transmite comenzi direct sau din programe PL/SQL
- Programe aplicative Oracle (WebDB, Oracle developer server, Oracle Express server,..)
- La module ODBC (Object DataBase Connectivity) din programe scrise în alte Limbaje de programare, care au facilități de conectare și interogare a bazelor de date (Delphi, Borland C++ Builder, Visual dBase, Java, Visual C++, Java Builder,..)

**WebDB** este un pachet de programe Oracle pentru dezvoltare un programe de lucru cu BD Oracle folosind o interfață grafică interactivă.

**Oracle Developer (Format Reports)** conține un set de programe pentru dezvoltare de aplicații interactive de la un terminal local al rețelei sau prin Internet. Se asigură:

- Definirea unor interfețe cu utilizatorul în **Format** grafic Windows pentru actualizarea tabelor și interogarea BD;
- Definirea de rapoarte pentru interogări complexe în aplicații
- Executarea unor grafice bi sau tridimensionale folosind informațiile din tabelele BD

**Oracle Expres Server** este destinat studiilor statistice pe BD multidimensionale (DataWorhouse).

**Oracle Enterprise Manager** este destinat administrării BD folosint o interfață grafică interactivă și poate fi utilizat numai cu drepturi de administrator (DBA).

**Programele scrise în limbajele de programare** clasice au acces la BD Oracle prin module ODBC speciale fiecărui limbaj, dacă au drepturi de acces la un user. Limbajul asigură mediul de dialog interactiv, care permite introducerea datelor ce vor servi la construcția unei interogări. Interogarea se transmite spre Serverul SQL al BD ca text, se execută de serverul SQL, care returnează un rezultat sub formă de tabelă spre programul client. Programul client preia tabela și afișează datele în forma dorită folosind toate posibilitățile de dialog ale limbajului.

**Biblioteca de programe OCI** (Oracle Call Interface) asigură cea mai mare viteză de prelucrare și procedurile pot fi apelate din limbaje ca C++ și PHP. Aceste proceduri apelează direct Serverul SQL evitând modulele ODBC care încetinesc prelucrarea de câteva ori. Dezavantajul utilizării OCI constă în scăderea portabilității programelor.

**Trebuie remarcat faptul că numai Oracle SQL Server are acces la baza de date.**

## Lansare SQL\*Plus

Componentele de baza Oracle sunt:

- Oracle SQL Server - executa toate comenzile SQL
- SQL\*Plus - mediu de dezvoltare,interfata cu utilizatorii
- PL/SQL - Limbaj procedural pentru dezvoltare aplicatii

Oracle SQL Server este singurul component Oracle care are acces la informatiile din Baza de Date, asigurand integritatea si securitatea informatiilor memorate. Serverul executa toate comenzile SQL pentru toti utilizatorii conectati la statii. Pentru fiecare utilizator conectat se deschide o sesiune separata. Fiecare utilizator va avea parola lui, dar mai multe sesiuni se pot lansa pe acelasi user de la statii diferite.

Toata interfata cu utilizatorii este asigurata de catre SQL\*Plus, care are unele comenzi de editare, permite comunicarea cu Oracle SQL Server si lanseaza programe PL/SQL.

Lansarea SQL\*Plus se face:

START | Programs | Oracle OraHome81 | Application Developement | SQL\*Plus

In fereastra deschisa pentru conectare se da:

User Name: **SCOTT** *tab* - user existent cu drepturi limitate  
Password: **TIGER** *tab* - parola utilizatorului  
Host string: **B623server** *enter* - Numele Bazei de Date existente pe server

Se va lansa SQL\*Plus si Oracle8i SQL Server si va apare prompter-ul de SQL.

Se pot introduce comenzi SQL terminate cu ;. Limbajul nu este case sensitive.

O comanda se poate scrie pe mai multe linii, se termina cu; si se lanseaza cu *enter*.

SQL>**Select \* From EMP;** -- afiseaza inregistrările din tabela EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

## DESCRIBE EMP

- comanda SQL\*Plus ce afiseaza structura tabeli EMP de salariati

Name	Null?	Type	
EMPNO	NOT NULL	NUMBER(4)	- tip numeric
ENAME		VARCHAR2(10)	- tip character de lungime variabila
JOB		VARCHAR2(9)	
MGR		NUMBER(4)	
HIREDATE		DATE	- tip data calendaristica
SAL		NUMBER(7,2)	- tip numar cu 10 cifre din care 2 zecimale
COMM		NUMBER(10,2)	
DEPTNO		NUMBER(2)	

Afisarea informatiilor se face conform setarilor SQL+ realizate cu comanda SET:

<b>SET LineSize</b>	<b>120</b>	- specifica numarul de caractere pe rand
<b>SET PageSize</b>	<b>60</b>	- specifica numarul de randuri pe pagina pentru care se pune cap de tabel
<b>SET ColSep</b>	<b>' '</b>	- specifica separatorul folosit intre coloanele tabeli afisate
<b>SET Long</b>	<b>200</b>	- numarul de caractere afisate pentru campurile de tip LONG
<b>SET AutoCommit</b>	<b>ON</b>	- face commit dupa fiecare comanda executata
<b>SHOW ALL</b>		- afisarea setarilor existente in SQL+

**EDIT** - comanda SQL\*Plus care deschide o fereastră de **editare a ultimei comenzi SQL**  
- in fereastra de editare nu se pune ; si se termina cu / pe ultimul rand

**RUN** - comanda SQL\*Plus pentru lansare **ultima comanda SQL editata in buffer**

**EDIT prog** - permite editarea unui script SQL sau program PL/SQL  
**START prog** - lanseaza in executie un program PL/SQL sau un script SQL

## Comenzi SQL

Se poate folosi alias pentru numele de campuri si tabele.

**SELECT** Ename NUME\_PRENUME, Job Functia, Sal Salar, Deptno Nr\_Depart  
**FROM** Emp **WHERE** sal >= 1300;

NUME_PRENU	FUNCTIA	SALAR	NR_DEPART
JONES	MANAGER	2975	20
BLAKE	MANAGER	2850	30
CLARK	MANAGER	2450	10
SCOTT	ANALYST	3000	20
KING	PRESIDENT	5000	10
FORD	ANALYST	3000	20

EDIT - va permite editarea ultimei comenzi modificand-o in **JOIN** intre tabela EMP si DEPT cu afisare Nume departament.

SQL> **RUN**

```
1 select Ename NUME_PRENUME, Job Functia, E.Sal Salar, D.Dname Nume_Depart  
2* FROM Emp E,Dept D WHERE Sal>=2000 AND e.Deptno=d.Deptno;
```

NUME_PRENU	FUNCTIA	SALAR	NUME_DEPART
CLARK	MANAGER	2450	ACCOUNTING
KING	PRESIDENT	5000	ACCOUNTING
JONES	MANAGER	2975	RESEARCH
FORD	ANALYST	3000	RESEARCH
SCOTT	ANALYST	3000	RESEARCH
BLAKE	MANAGER	2850	SALES

SELECT \* FROM Emp ,dept; -- este un **produs cartezian** admis de Oracle (fara conditie)

In lista de campuri afisate se accepta si expresii, care pot contine functii.

Functia **NVL** (Null Value) precizeaza valoarea implicita pentru cazul in care campul este NULL. Toate campurilor la care nu s-au precizat valorile se completeaza cu valoarea NULL care este nedeterminata si nu se admite in calcul

**Like** descrie o masca de cautare pentru camp ( jobul incepe cu 'A')

```
SELECT Ename NUME, Job Functia,  
Sal+NVL(Comm,0) - comisionul NULL inlocuit cu 0  
Venit, sal*0.16 impozit  
FROM emp WHERE job='MANAGER' OR Job Like 'A%';
```

NUME	FUNCTIA	VENIT	IMPOZIT
JONES	MANAGER	2975	595
BLAKE	MANAGER	2850	570
CLARK	MANAGER	2450	490
SCOTT	ANALYST	3000	600
FORD	ANALYST	3000	600

O tabela poate fi creata prin copierea alteia.

```
CREATE TABLE Pers AS SELECT * FROM Emp; - creaza tabela Pers copiind tabela Emp
```

```
SELECT * FROM Pers; -- afiseaza continutul noii tabele  
DESCR Pers -- afiseaza structura tabelei pers
```

**Crearea unei tabele** se face precizand intr-o lista numele si tipul campurilor.

```
CREATE TABLE Stud (Nume Char(20), Adresa Varchar2(15),Bursa Number(7),Datan Date);  
Table created.
```

Tipul Varchar2(20) este sir de caractere de lungime variabila si maxim 20.

**Stergerea unei tabele.**

```
SQL> DROP TABLE stud1; -- nu salveaza inregistrarile in Rollback  
Table dropped.
```

Adaugarea de inregistrari intr-o tabela:

```
INSERT INTO stud1 (nume, adresa,bursa,datan) - lista de campuri
      VALUES ('Radu', 'Lugoj', 1200,'15-jan-77'); - lista de valori
1 row created.
```

Stergere inregistrari:

```
DELETE FROM stud WHERE Adresa Like 'Lugoj%';
% indica un numar de caractere oarecare
_ indica un caracter care poate avea orice valoare
```

Atentie!! DELETE FROM stud; -- sterge toate inregistrarile salvandu-le in Rollback

```
ROLLBACK; -- reface toate inregistrarile sterse sau modificate
```

```
COMMIT; -- sterge fisierul Rollback si modificarile sunt definitive
```

```
TRUNCATE TABLE Stud; sterge numai inregistrarile si tabela ramane goala
```

```
UPDATE stud set bursa=1200 WHERE Nume= 'Radu';
```

```
UPDATE stud set bursa=bursa*1.2; -- mareste toate bursele cu 20%
```

**Afisarea tuturor tabelelor din spatiul utilizator:**

```
SELECT * FROM TAB; - afisare numele tabelelor utilizatorului curent
SELECT * FROM CAT; - afisare catalog utilizator
SELECT Owner,table_name from All_tables; - tabela toate datele despre table
Descr All_tables -- descrie structura tablei All_tables
```

Se admite Select in Select. Afisare salariatii care au aceeaasi functia ca si SCOTT.

```
SELECT * FROM Emp WHERE job= ANY(SELECT job FROM Emp where Ename='SCOTT');
```

Daca SELECT returneaza mai mult de o inregistrare se foloseste ANY sau ALL  
**ANY** – conditie indeplinita daca cel putin o valoare selectata indeplineste conditia.  
**ALL** - conditie indeplinita daca toate valorile selectate indeplinesc conditia.

**SELF JOIN** se admite deschizand de 2 ori aceeaasi tabela cu alias-uri diferite.

Afisare salariati numele si salariul sefului.

```
SELECT E.Ename, E.Sal Salar, S.Ename Nume_sef, S.Sal Salar_sef
FROM EMP E, EMP S WHERE E.Mgr=S.Empno;
```

**Operatii pe grupe de inregistrari** folosind GROUP BY si functii agregat SUM, COUNT, AVG, MIN, MAX. Conditia de grup se verifica cu HAVING.

Afisare pentru fiecare functie suma si media salariilor, salarul maxim, minim si numarul salariatilor.

```
SELECT Job,avg(sal) Salar_mediu,Sum(sal) Suma_sal, Max(Sal) Sal_max,
Min(sal) Min_sal Count(*) FROM EMP Group By job HAVING Avg(sal)>1000;
```