

Capitolul 3. Interfața (neprietenoasă) cu serverul: SQL*Plus

Orice server de baze de date, deci și Oracle, impune existența unei aplicații client prin intermediul căreia utilizatorii să gestioneze obiectele din baza de date. În decursul timpului, odată cu sporirea performanțelor noilor versiuni Oracle și a cotei de piață a produsului, mai multe companii au dezvoltat diverse utilitare software comerciale, ce oferă o interfață grafică prietenoasă pentru administrarea bazei de date. Chiar dacă ultimile versiuni Oracle furnizează și ele un asemenea utilitar, sub forma Enterprise Manager Console, bătrânul SQL*Plus rămâne totuși varianta de platformă-client pe care o utilizează majoritatea dezvoltatorilor împătimiți, pentru care nimic nu se compară cu a scrie cod pentru orice operație direct în linie de comandă.

SQL*Plus reprezintă cea mai simplă formă de aplicație-client pentru server-ul Oracle ce se instalează în mod implicit odată cu instalarea clientului Oracle. Orice operație de administrare a bazei de date sau de manipulare a obiectelor din schema proprie a utilizatorului presupune scrierea explicită a unei comenzi SQL (uneori SQL*Plus) sau a unui bloc PL/SQL.

3.1.Elemente generale

Pentru a lansa în execuție SQL*Plus există două posibilități:

- pentru sistemul de operare Windows9X putem utiliza meniul de start, Start→Programs→Oracle-OUIHome→Application Development/SQL Plus
- lansarea în execuție a fișerului corespunzător ce se găsește pe calea: %ORACLE_HOME%\ora81\bin\SQLPLUSW.EXE¹

Înainte de a lansa în execuție acest utilitar trebuie să ne asigurăm că serverul Oracle este pornit. Bineînțeles că primele informații pe care SQL*Plus le va solicita vor fi: numele de utilizator, parola și serviciul NET la care dorim să ne conectăm (figura 3.1).

¹ ORACLE_HOME = adresa pe disc a directorului rădăcină în care s-a instalat server-ul

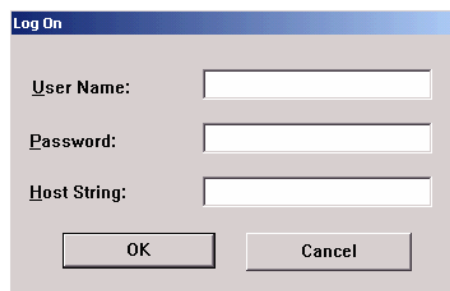


Figura 3.1 Fereastra de "Log On" a SQL*Plus

În fereastra de mai sus trebuie introduse:

- numele de utilizator Oracle și parola *pentru serverul la care dorim să ne conectăm*;
- în căsuța Host String: numele serviciului NET8 configurat pentru un anumit server. Dacă serverul Oracle este instalat local și este pornit, putem ignora această căsuță, SQL*Plus conectându-se implicit la acesta.

Este important să înțelegem faptul că în căsuța Host String trebuie introdus numele serviciului definit pentru un anumit server Oracle și **nu** adresa IP a calculatorului-gazdă. Numele tuturor serviciilor definite pot fi găsite în fișierul ORACLE_HOME\ora81\network\ADMIN\Tnsnames.ora². Spre exemplu, pentru un conținut precum cel din listing-ul 3.1 al acestui fișier, o valoare validă pentru câmpul Host String ar fi: VINZARI

Listing 3.1. Exemplu de fișier tnsnames.ora

```
# TNSNAMES.ORA Network Configuration File: D:\oracle\ora81\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

VINZARI =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = xxx.xxx.xxx.xxx)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = orcl)
    )
  )
BDSTUD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = xxx.xxx.xxx.xxx)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = oracle)
    )
  )
```

² Trebuie acordată atenție deosebită aceluiași fișier! Pentru mai multe informații în acest sens vezi capitolul 2

Dacă nu introducem un nume valid de serviciu NET, server-ul va returna eroarea:

ERROR:

ORA-12154: TNS:could not resolve service name

În cazul în care informațiile de conectare au fost specificate corect, iar SQL*Plus a reușit să se conecteze la server, vom obține fereastra din figura 3.2.

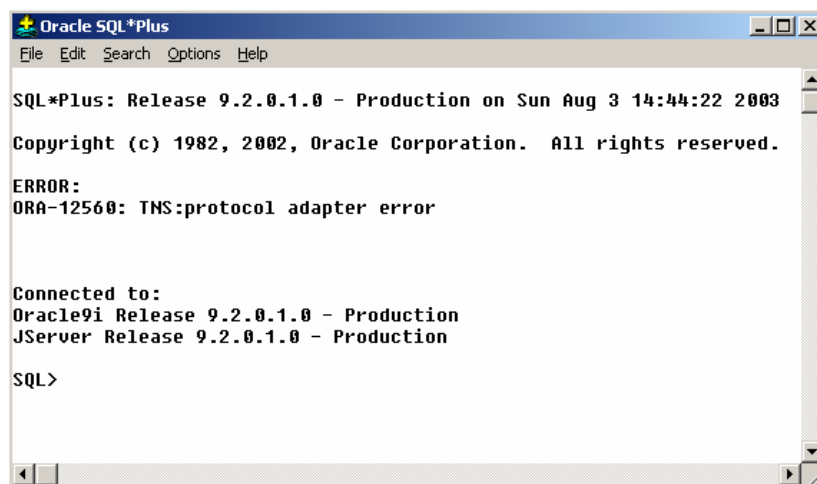
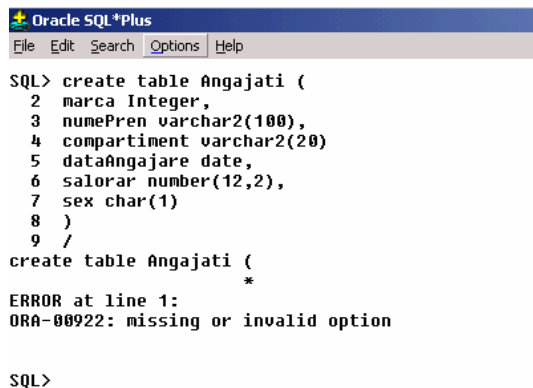


Figura 3.2 Fereastra standard SQL*Plus după două încercări de conectare (prima nereușită)

Acest utilitar nu oferă elemente de interfață GUI (Graphical User Interface) ci doar un prompter la nivelul căruia pot fi introduse: *instrucțiuni SQL* (atât din categoria Data Definition Language cât și Data Manipulation Language) și *comenzi specifice SQL*Plus*. Fiecare instrucțiune SQL va fi trimisă serverului, acesta o va executa și va trimite înapoi un răspuns (rezultatul execuției) ce va fi afișat de SQL*Plus în fereastra curentă de lucru.

După cum observați și în figura 3.2, prompter-ul este poziționat imediat după o directivă `SQL>` așteptând introducerea de instrucțiuni SQL sau comenzi specifice SQL*Plus.

Vom exemplifica maniera de lucru cu acest utilitar pornind de la crearea unei tabele, Angajați, cu definiția din figura 3.3.



```

Oracle SQL*Plus
File Edit Search Options Help

SQL> create table Angajati (
      2  marca Integer,
      3  numePren varchar2(100),
      4  compartiment varchar2(20)
      5  dataAngajare date,
      6  salorar number(12,2),
      7  sex char(1)
      8  )
      9  /
create table Angajati (
                                *
ERROR at line 1:
ORA-00922: missing or invalid option

SQL>

```

Figura 3.3 Primul contact...prima eroare

Figura 3.3 demonstrează câteva din *elementele esențiale* ce trebuie cunoscute atunci când lucrăm cu SQL*Plus:

- o instrucțiune standard SQL poate fi scrisă pe oricâte rânduri (sfârșitul de rând se introduce prin intermediul tastei *Enter*);
- nu pot fi lăsate rânduri libere în cadrul unei instrucțiuni scrise pe mai multe rânduri;
- o instrucțiune SQL se lansează în execuție prin una din comenzile SQL*Plus: */* (slash) sau *;* (punct-și-virgulă);
- putem utiliza comanda */* pentru a lansa de mai multe ori în execuție conținutul buffer-ului SQL*Plus (ultima instrucțiune SQL sau ultimul bloc PL/SQL);
- un rând, odată editat, nu mai poate fi modificat în mod direct, decât prin intermediul tastei *BackSpace* care șterge caracterele din stânga cursorului;
- în cazul unei erori (în exemplul de mai sus lipsește virgula dintre *compartiment* și *dataAngajare*) numărul liniei indicate drept eronată nu este întotdeauna corect.

Odată eroarea detectată, s-ar părea că ar trebui să rescriem întreaga frază SQL. Există însă o posibilitate de a modifica instrucțiunea lansată ultima dată în execuție, dacă apelăm la comanda SQL*Plus *EDIT*, care deschide un editor de texte (Notepad în Windows) ce va conține buffer-ul SQL*Plus (ultima instrucțiune SQL executată). Odată efectuate modificările necesare, editorul se închide cu opțiunea de salvare a modificărilor și se rescrie comanda pentru lansare în execuție: */* sau *;*. În exemplul de mai sus, tastarea comenzii *EDIT* pe linia prompter-ului SQL*Plus va deschide editorul de text cu fraza SQL de creare a tabelului *Angajati*.

Notă. Comanda */* **se scrie întotdeauna pe o linie nouă** (care de altfel va fi și ultima). Comanda *;* (punct-și-virgulă) poate fi scrisă la sfârșitul ultimei linii ce

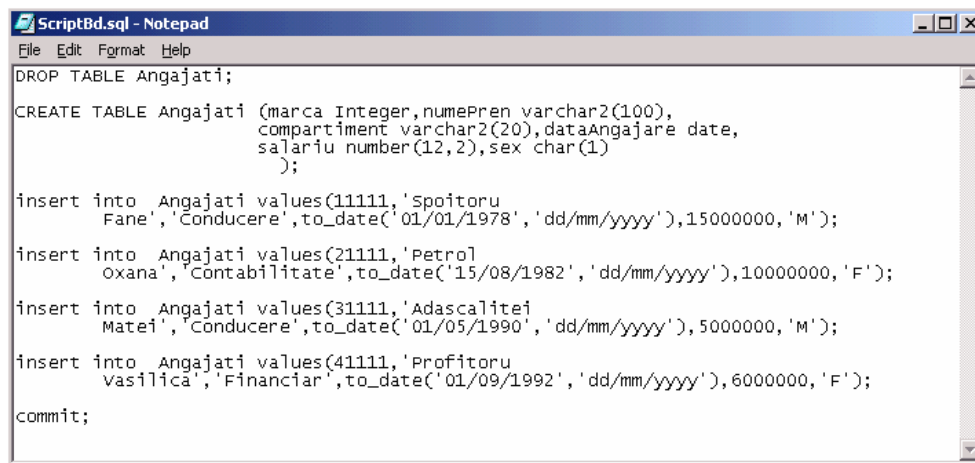
constituie instrucțiunea SQL. De asemenea, este important să reținem că actualizările înregistrărilor din diverse tabele ce au loc în urma unei instrucțiuni SQL trebuie permanentizate explicit prin instrucțiunea SQL `commit` (sau `rollback` pentru anulare – vezi capitolul 5)

Vizualizarea definiției unui obiect din baza de date poate fi realizată prin comanda `DESCRIBE numeObiect` (spre ex. `DESCRIBE Angajati`)

3.2.Scripturi SQL

Buffer-ul SQL*Plus poate conține la un moment dat doar o singură instrucțiune SQL, așa că dezvoltatorii de aplicații recurg, de cele mai multe ori, la scripturi SQL. Acestea nu sunt altceva decât o înșiruire de instrucțiuni SQL și/sau comenzi SQL*Plus scrise cu un editor de texte oarecare și salvate într-un fișier cu extensia `.sql`. Extensia nu este obligatorie (poate fi și `.txt`), însă fișierul va conține doar text conform standardului ANSI (fără nici un element de formatare).

Figura 3.4 prezintă un exemplu de asemenea fișier ce poate fi lansat în execuție din SQL*Plus.



```
File Edit Format Help
DROP TABLE Angajati;

CREATE TABLE Angajati (marca Integer,numePren varchar2(100),
                        compartiment varchar2(20),dataAngajare date,
                        salariu number(12,2),sex char(1)
                        );

insert into Angajati values(11111,'Spoitoru
Fane','Conducere',to_date('01/01/1978','dd/mm/yyyy'),15000000,'M');

insert into Angajati values(21111,'Petrol
oxana','Contabilitate',to_date('15/08/1982','dd/mm/yyyy'),10000000,'F');

insert into Angajati values(31111,'Adascalitei
Matei','Conducere',to_date('01/05/1990','dd/mm/yyyy'),5000000,'M');

insert into Angajati values(41111,'Profitoru
Vasilica','Financiar',to_date('01/09/1992','dd/mm/yyyy'),6000000,'F');

commit;
```

Figura 3.4 Exemplu de script.

Elementele esențiale ce trebuie avute în vedere la redactarea unui script SQL sunt următoarele:

- orice instrucțiune SQL trebuie finalizată cu una din comenzile SQL*Plus pentru execuție (`/` pe o linie nouă sau `;` pe aceeași linie);
- în cadrul unei instrucțiuni SQL nu pot exista linii libere, această restricție nefiind valabilă pentru instrucțiuni diferite;

- pentru șiruri de caractere se utilizează apostrof și nu ghilimele (regulă general valabilă, de altfel, în Oracle);
- comentariile se introduc în maniera: `-- text comentat --` (vezi în acest sens și listing-ul 3.2) sau prin secvența de caractere `/*` la un capăt și `*/` la sârșit;
- nu trebuie uitată „comiterea” eventualelor modificări, ca rezultat al unor fraze SQL-DML (Data Manipulation Language) prin intermediul instrucțiunii COMMIT.

Comanda SQL*Plus pentru lansarea în execuție a unui script SQL este :

`@fiser.ext [lista de parametri]`

Așa cum observați și în figura 3.5 execuția unui script înseamnă execuția frazelor SQL și a comenzilor SQL*Plus în ordinea „naturală” în care au fost editate. Despre parametri vom discuta în cele ce urmează

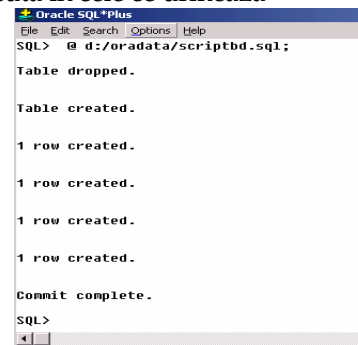


Figura 3.5 Execuția unui script SQL

Notă. Pentru a lansa în execuție un script dintr-un alt script se va folosi comanda

`@@ numeSubScript (dublu “at”)`

3.2.1. Variabile de substituție și parametri în scripturile SQL

Variabilele de substituție sunt utilizate pentru a furniza valori în mod interactiv la execuția unui script SQL. Variabilele se declară în interiorul scriptului astfel (vezi și listing 3.2):

`&numevariabila`

La momentul execuției, SQL*Plus va solicita valori pentru toate variabilele declarate, va înlocui respectivele valori în instrucțiunea SQL (în locul variabilelor), iar, în final, va trimite fraza SQL spre execuție (figura 3.6).

Analizând listing-ul 3.2 și figura 3.6 este important de reținut că :

- numele variabilelor nu trebuie obligatoriu să fie unic;
- dacă utilizăm același nume de variabilă de mai multe ori, SQL*Plus va solicita valori pentru fiecare declarație și nu pentru fiecare nume unic de variabilă (există și o altă posibilitate redată mai jos);
- în cazul variabilelor ce substituie un șir de caractere există două posibilități de declarare a lor (vezi variabila `compart_`):
 - se încadrează declarația variabilei între apostrofuri și, la execuție, se furnizează o valoare fără marcajul de șir de caractere;
 - se renunță la marcaj în script, iar la execuție se specifică un șir de caractere încadrat între apostrofuri ('valoare');
- dacă din script lipsește instrucțiunea `commit`, pentru permanentizarea actualizărilor aceasta va trebui executată din SQL*Plus după execuția scriptului sau la un moment ulterior, în funcție de alte necesități

Listing 3.2. Utilizarea variabilelor de substituție într-un script SQL

```
-- Comentariu: Vom utiliza variabile de substituție compart_ si salariu_vitalie --

insert into Angajati values
(11111 , 'Enea Silvestra', '&compart_', to_date('01/01/1978','dd/mm/yyyy'), 15000000, 'F');

insert into Angajati values
( 21111, 'Vartolomei Vitalie', &compart_ ,to_date('01/01/1992','dd/mm/yyyy'), &salariu_vitalie, 'M');
```

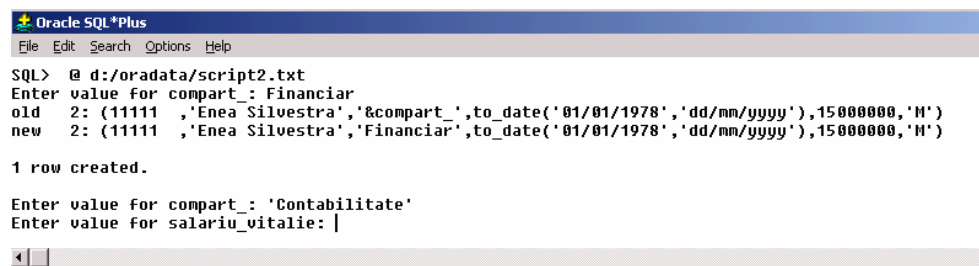


Figura 3.6 Execuția scriptului din listing 3.2

Există și posibilitatea utilizării variabilelor de substituție în combinație cu instrucțiunea `ACCEPT` care *definește* o variabilă și solicită la momentul execuției o valoare pentru aceasta, valoare ce va fi apoi utilizată ori de câte ori este întâlnită variabila respectivă pe parcursul scriptului. Sintaxa comenzii `ACCEPT` este următoarea:

```
ACCEPT variabila [NUMBER|CHAR|DATE] [FORMAT format]
[DEFAULT valImplicita] PROMPT textDeAfisat|[NOPROMPT]
```

Un exemplu în acest sens îl constituie listingul 3.2 Bis și figura 3.6 Bis unde putem observa că, la execuție, SQL*Plus verifică dacă se respectă formatul specificat, în caz contrar afișându-se un mesaj de eroare și solicitându-se din nou o valoare validă.

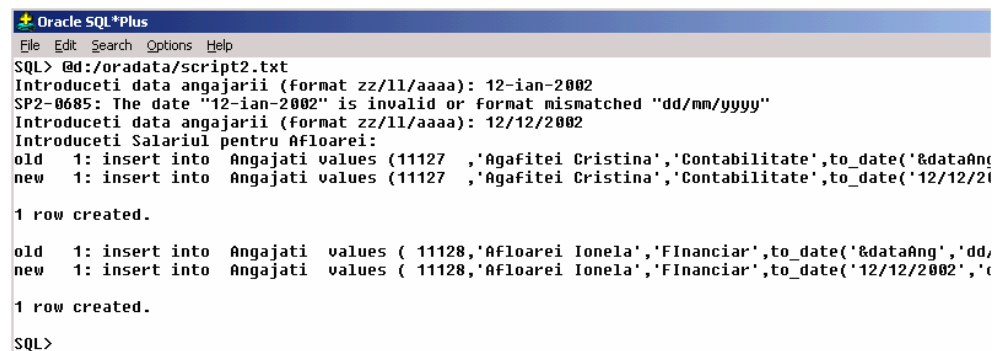
Listing 3.2 Bis Declarația unor variabile cu instrucțiunea ACCEPT

```
ACCEPT dataAng DATE FORMAT 'dd/mm/yyyy' DEFAULT '01/01/2003' PROMPT 'Introduceți data
angajării (format zz/ll/aaaa): '

ACCEPT sal NUMBER DEFAULT '0.00' PROMPT 'Introduceți Salariul pentru Afloarei: '

insert into Angajati values (11127 , 'Agafitei
Cristina', 'Contabilitate', to_date( '&dataAng' , 'dd/mm/yyyy' ), 15000000, 'F');

insert into Angajati values ( 11128, 'Afloarei Ionela', 'Financiar', to_date( '&dataAng' , 'dd/mm/yyyy'),
&sal, 'M');
```



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> @d:/oradata/script2.txt
Introduceți data angajării (format zz/ll/aaaa): 12-ian-2002
SP2-0685: The date "12-ian-2002" is invalid or format mismatched "dd/mm/yyyy"
Introduceți data angajării (format zz/ll/aaaa): 12/12/2002
Introduceți Salariul pentru Afloarei: 15000000
old 1: insert into Angajati values (11127 , 'Agafitei Cristina', 'Contabilitate', to_date('12/12/2002', 'dd/mm/yyyy'), 15000000, 'F');
new 1: insert into Angajati values (11127 , 'Agafitei Cristina', 'Contabilitate', to_date('12/12/2002', 'dd/mm/yyyy'), 15000000, 'F');
1 row created.

old 1: insert into Angajati values ( 11128, 'Afloarei Ionela', 'Financiar', to_date('12/12/2002', 'dd/mm/yyyy'), 15000000, 'M');
new 1: insert into Angajati values ( 11128, 'Afloarei Ionela', 'Financiar', to_date('12/12/2002', 'dd/mm/yyyy'), 15000000, 'M');
1 row created.

SQL>
```

Figura 3.6 Bis. Execuția scriptului 3.2 Bis

Se observă că, deși variabila dataAng este declarată de tip Date aceasta este preluată de la prompter sub formă de șir de caractere, fapt pentru care se recurge, în cadrul scriptului, la o funcție de conversie (TO_DATE('text', 'format'))

Un script SQL poate să conțină declarații de parametri, valorile actuale ale acestora fiind furnizate sub formă de listă (separate prin spațiu) în cadrul comenzii de lansare în execuție (@ .../numescript.ext param1 param2 param3). Declarația parametrilor se realizează prin aceeași comandă &:

&indexParametru

unde indexParametru reprezintă numărul de ordine al valorilor specificate în lista de parametri la execuția scriptului.

Listing 3.3 Script parametrizat

```
-- utilizarea parametrilor--
```

```
insert into Angajati values(11112 , 'Atudorei Violeta', '&1', to_date('01/01/1978', 'dd/mm/yyyy'), &2, 'F');
```



```
insert into Angajati values( 11113,'Creanga Maria','&1',to_date('01/01/1978','dd/mm/yyyy'),&3,'F');

commit;
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> @ d:\oradata\script3.txt Sectia2 500000 230000
old 1: insert into Angajati values(11112 , 'Atudorei Violeta', '&1', to_date('01/01/1978', 'dd/mm/yyyy')
new 1: insert into Angajati values(11112 , 'Atudorei Violeta', 'Sectia2', to_date('01/01/1978', 'dd/mm/

1 row created.

old 1: insert into Angajati values( 11113, 'Creanga Maria', '&1', to_date('01/01/1978', 'dd/mm/yyyy')
new 1: insert into Angajati values( 11113, 'Creanga Maria', 'Sectia2', to_date('01/01/1978', 'dd/mm/

1 row created.

Commit complete.

SQL> |
```

Figura 3.7 Execuția scriptului din listing 3.3 pe baza valorilor furnizate pentru parametri

După cum se observă, spre deosebire de variabilele de substituție, un parametru poate fi utilizat de mai multe ori în cadrul aceluiași script, având aceeași valoare furnizată la momentul execuției.

Notă. Parametrii ce corespund unor valori de tip șir de caractere (&1 în exemplul de mai sus) vor fi încadrați între apostrofuri, iar valoarea va fi furnizată la momentul execuției fără acești marcatori.

3.3.Formatarea rezultatului interogărilor SQL

Rezultatul frazelor Select SQL este afișat de SQL*Plus după o schemă de formatare predefinită care, de cele mai multe ori, nu oferă un aspect tocmai lizibil. Astfel, dacă analizăm figura 3.8 observăm cel puțin câteva neajunsuri:

- dimensiunea implicită a liniei este de 80 de caractere ceea ce determină afișarea unei înregistrări pe mai multe linii;
- dimensiunea implicită a paginii de afișare este de 14 rânduri fapt ce are ca rezultat afișarea antetului (numele coloanelor) de mai multe ori în aceeași fereastră SQL*Plus;
- în antetul rezultatului, titlul coloanelor este același cu numele coloanelor tabelelor din baza de date;
- spațiul alocat fiecărei coloane este egal cu lungimea maximă a câmpurilor tabelelor ceea ce duce la utilizarea inefficientă a spațiului disponibil pe un rând și uneori la afișarea incompletă a titlului coloanelor (vezi coloana Angajati.sex)

```

Oracle SQL*Plus
File Edit Search Options Help
SQL> select * from angajati;

HARCA
NUMEPREN
-----
COMPARTIMENT DATAANGAJ SALARIU $
-----
11111
Spoitoru Fane
Financiar 01-JAN-78 15000000 H
21111
Petrol Oxana
Conatbilitate 01-JAN-92 10000000 F
HARCA
NUMEPREN
-----
COMPARTIMENT DATAANGAJ SALARIU $
-----
31111
Adascalitei Matei
Conducere 01-MAY-90 213 H

```

Figura 3.8 Rezultatul neformatat al unei interogări SQL

Ca urmare, este necesară „pregătirea terenului” înainte de realizarea unei interogări.

3.3.1. Principalele comenzi SQL*Plus de formatare a rezultatului interogărilor

Formatarea rezultatului unei interogări SQL trebuie să se realizeze înainte de execuția frazei SELECT. O parte din comenzile SQL*Plus pentru formatare le regăsim în tabelul 3.1

Tabelul nr. 3.1 Principalele comenzi de formatare a rezultatului interogărilor

Comanda	Observații
SET LINESIZE n	n=numărul caracterelor pe rând
SET PAGESIZE n	n=numărul liniilor pe pagină
COLUMN numecol FORMAT model	model poate fi: A20 – format ASCII de dimensiune 20 9,999.99 – format numeric de dimensiune 4 întregi și 2 zecimale (valorile mai mari decât lungimea specificată vor fi înlocuite cu #) Poate fi specificat și modul de gestionare a valorilor de tip sir de caractere ce depășesc lungimea specificată la formatare: WRAP – afișează pe mai multe rânduri

	WORD_WRAP- afișează pe mai multe rânduri cuvinte întregi TRUNCATED - nu afișează caracterele în afara lungimii specificate
COLUMN numecol HEADING 'titlu'	Specifică un titlu pentru coloana respectivă
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';	Modifică formatul de afișare al datei calendaristice pentru sesiunea curentă SQL*Plus.

Figura 3.9 ilustrează utilizarea opțiunilor de formatare pentru a afișa rezultattul interogării din figura 3.8

```

Oracle SQL*Plus
File Edit Search Options Help
SQL> set linesize 120
SQL> set pagesize 60
SQL> column numepren format A13 WORD_WRAP
SQL> column compartiment format A16
SQL> column sex format A3
SQL> column compartiment format A15
SQL> column salariu format 99,999,999.99
SQL> column numepren heading 'NUME |ANGAJAT'
SQL> column dataangajare heading 'DATA |ANGAJARE'
SQL> select * from angajati;

      NUME
MARCA  ANGAJAT      COMPARTIMENT      DATA
ANGAJARE      SALARIU SEX
-----
21111 Petrol Oxana  Contabilitate  01-01-1992  10,000,000.00 F
31111 Adascalitei  Conducere     01-05-1990    213.00 M
41111 Profitoru    Financiar     01-09-1992   6,000,000.00 F
21111 Uartolomei   Contabilitate  01-01-1992   5,600,000.00 M
11112 Atudorei     Sectia2        01-01-1978   500,000.00 F
11113 Creanga Maria Sectia2        01-01-1978   230,000.00 F

6 rows selected.

SQL>

```

Figura 3.9 Rezultatul formatat al frazei SELECT

Rezultatul unei interogări SQL poate fi salvat într-un fișer ASCII pe disc dacă apelăm la comanda SPOOL [*file_name*[.ext)]|OFF|OUT], unde opțiunea OFF dezactivează „spoolig-ul” iar OUT trimite la imprimantă rezultatul. Comanda trebuie executată înainte de inițierea interogării.