

## Rapoarte interactive

### Obiective

Prin parcurgerea acestui referat studentul va dobândi cunoștințele necesare:

- scrierea unei interogări care necesită o variabilă de intrare;
- formatarea datelor afișate;
- crearea și execuția unui fișier script.

Exemplele prezentate până acum nu au prezentat nici un grad de interactivitate. Prin utilizarea SQL\*Plus este posibilă crearea de rapoarte care cer utilizatorului să furnizeze propriile valori pentru a limita domeniul datelor returnate. Pentru a crea rapoarte interactive se pot include variabile de substituție într-un fișier de comenzi sau într-o singură instrucțiune SQL.

### Variabile de substituție

În SQL\*Plus se pot folosi variabile de substituție cu ajutorul caracterului '&' pentru a stoca temporar valori. Variabilele pot fi definite cu ajutorul comenzilor **ACCEPT** sau **DEFINE**. **ACCEPT** citește valoarea introdusă de utilizator și o memorează într-o variabilă. **DEFINE** creează și atribuie o valoare unei variabile.

### Utilizarea variabilelor de substituție &

În timpul execuției unui raport, utilizatorii doresc adeseori să limiteze în mod dinamic rezultatele returnate de o interogare. SQL\*Plus oferă această flexibilitate prin intermediul variabilelor utilizator. Pentru a identifica fiecare variabilă din instrucțiunea SQL se va utiliza caracterul '&'.

```
SQL> SELECT empno ,ename ,sal ,deptno
2 FROM emp
3 WHERE empno=&employee_num;
```

Enter value for employee_num:7369			
EMPNO	ENAME	SAL	DEPTNO
-----	-----	-----	-----
7369	SMITH	800	20

NOTAȚIE	DESCRIERE
&variabila_utilizator	Indică o variabilă într-o instrucțiune SQL; dacă variabila nu există, SQL*Plus va crea variabila și va cere utilizatorului să introducă o valoare.

Exemplu de mai sus reprezintă o instrucțiune SQL care va cere utilizatorului să introducă numărul unui angajat în momentul execuției.

**Notă:** în cazul utilizării unui singur & utilizatorul trebuie să introducă valoarea acelei variabile ori de câte ori se execută instrucțiunea, în cazul în care variabila nu există.

### Comanda SET VERIFY

```
SQL> SET VERIFY ON
SQL> SELECT empno ,sal ,deptno
```

```

2 FROM emp
3 WHERE empno=&employee_num;

```

```

Enter value for employee_num:7369
old 3: WHERE empno=&employee_num
new 3: WHERE empno=7369
.....

```

În cazul în care **SET VERIFY** este activă (pe **ON**), SQL\*Plus este forțat să afișeze textul unei instrucțiuni înainte și după ce sunt înlocuite variabilele de substituție cu valorile corespunzătoare. În exemplul de mai sus este afișată atât valoarea veche cât și cea nouă pentru coloana `empno`.

Regula conform căreia într-o clauză WHERE valorile de tip caracter și date calendaristice trebuie încadrate de apostrofuri ( ' ') se aplică și în cazul variabilelor de substituție. Pentru a evita introducerea apostrofulor în momentul execuției se va încadra variabila între apostrofuri. În următorul exemplu este prezentată o interogare pentru a afla numele angajatului, numărul departamentului și salariul anual pe baza funcției introdusă la prompt de utilizator.

```

SQL> SELECT  ename,deptno,sal*12
2 FROM      emp
3 WHERE     job='&job_title';

```

```

Enter value for job_title:ANALYST

```

ENAME	DEPTNO	SAL*12
SCOTT	20	36000
FORD	20	36000

**Notă:** se pot folosi și funcțiile **UPPER** și **LOWER** în conjuncție cu **&**. (`UPPER('&job_title')` va converti caracterele din șirul argument în majuscule; efect opus are funcția `LOWER`)

Variabilele de substituție pot fi utilizate pentru a suplimenta:

- o condiție WHERE;
- o clauză ORDER BY;
- numele unei coloane;
- numele unui tabel;
- o instrucțiune SELECT.

Pentru exemplificare să considerăm cererea de a afișa numărul de înregistrare al angajaților și valorile din coloana introdusă de la tastatură pe baza unei condiții introdusă, de asemenea, de la tastatură.

```

SQL> SELECT  empno, &column_name
2 FROM      emp
3 WHERE     &condition;

```

```

Enter value for column_name: job
Enter value for condition: deptno=10

```

EMPNO	JOB
7839	PRESIDENT
7782	MANAGER

```
7934 CLERK
```

**Notă:** Dacă nu introduceți o valoare pentru variabila de substituție se va genera o eroare când se va executa instrucțiunea de mai sus.

### Utilizarea variabilelor de substituție &&

Se va utiliza ampersand dublu (&&) atunci când se dorește reutilizarea valorii acelei variabile, fără a fi necesară reintroducerea de la tastatură a unei valori pentru variabilă.

```
SQL> SET VERIFY ON
SQL> SELECT empno, ename, job, &&column_name
2 FROM emp
3 ORDER BY &column_name;
```

```
Enter value for column_name: deptno
old 1: select empno,ename,job,&&column_name
new 1: select empno,ename,job,deptno
old 3: order by &column_name
new 3: order by deptno

  EMPNO  ENAME      JOB              DEPTNO
-----  -
7839    KING        PRESIDENT        10
7782    CLARK       MANAGER          10
7934    MILLER     CLERK            10
.....
14 rows selected
```

În exemplul de mai sus, utilizatorului i se cere să furnizeze o valoarea pentru variabila `column_name` o singură dată . Valoarea furnizată de utilizator (`deptno`) va fi utilizată atât pentru afișare cât și pentru ordonarea datelor .

SQL\*Plus memorează valoarea unei variabile cu ajutorul comenzii SQL\*Plus **DEFINE** și va utiliza acea valoare ori de câte ori se face referire la variabilă. Pentru a șterge o variabilă trebuie folosită comanda **UNDEFINE**.

Variabilele utilizator pot fi definite înainte de a executa o instrucțiune **SELECT** cu ajutorul comenzilor **DEFINE** și **ACCEPT**.

Comanda	Descriere
DEFINE <i>variabilă</i> = valoare	Creează o variabilă de tip CHAR și atribuie acelei variabile o valoare.
DEFINE <i>variabilă</i>	Afișează variabila, valoarea și tipul acesteia.
DEFINE	Afișează toate variabilele utilizator, împreună cu valorile asociate și tipurile de date
ACCEPT	Citește o linie introdusă de utilizator și o memorează într-o variabilă.

Sintaxa comenzii ACCEPT este următoarea:

**ACCEPT variabila [tip\_date] [FORMAT format] [PROMPT text] [HIDE]**

unde:

*variabila* este denumirea variabilei. Dacă aceasta nu există, ea va fi creată de SQL\*Plus.

*tip\_date* este tipul de date asociat variabilei: NUMBER, CHAR sau DATE.  
CHAR are o lungime de maxim 240 octeți.

*FOR[MAT] format* precizează modelul de format - de exemplu A10 sau 9.999

*PROMPT text* afișează textul *text* înainte ca utilizatorul să introducă valoarea

*HIDE* ascunde cea ce introduce utilizatorul – de exemplu o parolă

**Notă: nu se va prefixa numele variabilei cu '&' în comanda ACCEPT.**

```
SQL> ACCEPT dept PROMPT 'Introduceti numele departamentului: '  
SQL> SELECT *  
2 FROM scott.dept  
3 WHERE dname=UPPER('&dept');
```

```
Introduceti numele departamentului: Sales  
  
DEPTNO  DNAME      LOC  
-----  -  
          30  SALES      CHICAGO
```

O variabilă va rămâne definită până când :

- se va utiliza comanda UNDEFINE pentru a o anula;
- se va părăsi mediul SQL\*Plus.

Când anulați o variabilă există posibilitatea de a verifica modificările efectuate executând comanda DEFINE.

### Comenzi SQL\*Plus pentru formatarea modului de afișare a rezultatelor

Următoarele comenzi pot fi utilizate pentru a controla anumite caracteristici de afișare ale unui raport:

Comanda	Descriere
COL[UMN] [coloana optiune]	Controlează formatul coloanelor
TTI[TLE] [text   OFF   ON]	Specifică un "header" care va apare în partea superioară a fiecărei pagini
BTI[TLE] [text   OFF   ON]	Specifică un "footer" care va apare în partea de jos a fiecărei pagini
BRE[AK] [ON elem_raport]	Anulează valorile duplicat și împarte liniile din raport. Pentru mai multe detalii executați comanda HELP break.

*Indicații:*

- toate comenzile de formatare rămân valabile până la sfârșitul sesiunii SQL\*Plus sau până în momentul rescrierii sau ștergerii acestor opțiuni.
- după fiecare raport se vor restaura setările SQL\*Plus la valorile implicite.
- dacă se va crea un alias pentru o coloană, referirile ulterioare se vor face la acel alias, nu la numele coloanei.

**Comanda COLUMN**

**COL[UMN] [ {coloana | alias} [optiuni]]**

Optiune	Descriere
CLE[AR]	anulează orice format de coloană
FOR[MAT] format	modifică formatul de afișare al datelor din coloană
HEA[DING] text	setează titlul coloanei. Dacă nu este utilizată opțiunea JUSTIFY, o linie verticală ( ) va forța trecerea pe rândul următor a textului rămas din header.
JUS[TIFY] {aliniere}	aliniază titlul coloanei la stânga, centrat sau la dreapta
NOPRI[NT]	ascunde coloana.
NUL[L] text	indică textul ce va fi afișat pentru valori null
PRI[NT]	arată coloana
TRU[NCATED]	trunchiază șirul la sfârșitul primei linii de afișaj
WRA[PPED]	transferă sfârșitul de șir pe linia următoare

Exemple de utilizare a comenzii COLUMN:

**COLUMN ename HEADING 'Employee|Name' FORMAT A15**

- noua denumire a coloanei ename este Employee Name (scris pe două rânduri), iar pentru valori se alocă un spațiu de 15 caractere

**COLUMN sal JUSTIFY LEFT FORMAT \$99,999.00**

**COLUMN mgr FORMAT 9999999 NULL 'No manager'**

**COLUMN ename** – afișează setarea curentă pentru coloana ENAME

**COLUMN ename CLEAR** – anulează setările pentru coloana ENAME

Modele de format pentru coloane:

Element	Descriere	Exemplu	Rezultat
<i>An</i>	Setează lățimea zonei de afișare la <i>n</i> caractere	N/A	N/A
9	O singură cifră, cu suprimarea valorilor 0	999999	1234
0	prefixează numărul cu un zero	099999	01234
\$	Simbolul pentru dolar	\$9999	\$1234
L	Moneda locală	L9999	L1234
.	Poziția punctului zecimal	9999.99	1234.00
,	Virgulă pentru separarea cifrei zecilor de mii	9,999	1,234

## Comenzile TTITLE și BTITLE

**TTI[TLE] [text | OFF | ON]**

**BTI[TLE] [text | OFF | ON]**

unde: *text* reprezintă titlul raportului. Se introduc apostrofuri dacă textul conține mai multe cuvinte.

Comanda **TTITLE** este utilizată pentru a formata header-ul paginii, iar comanda **BTITLE** pentru footers.

## Procedura de creare a unui fișier script care să execute un raport

1. Se creează instrucțiunea SQL SELECT
2. Se salvează instrucțiunea SELECT într-un fișier script
3. Se încarcă fișier script într-un editor text
4. Se adaugă comenzile de formatare înaintea instrucțiunii SELECT
5. Se verifică dacă după instrucțiunea SELECT urmează caracterul '/' sau ';'.
6. După instrucțiunea SELECT se anulează comenzile de formatare
7. Se salvează fișierul script
8. Se execută comanda *START fișier* pentru a executa script-ul

## Exemplu de raport

Fri Oct 24

page 1

### Employee Report

Job Category	Employee	Salary
CLERK	ADAMS	\$1,100.00
CLERK	JAMES	\$950.00
CLERK	MILLER	\$1,300.00
CLERK	SMITH	\$800.00
MANAGER	BLAKE	\$2,850.00
MANAGER	CLARK	\$2,450.00
MANAGER	JONES	\$2,975.00
SALESMAN	ALLEN	\$1,600.00
SALESMAN	MARTIN	\$1,250.00
SALESMAN	TURNER	\$1,500.00
SALESMAN	WARD	\$1,250.00

Confidential

Să se creeze un fișier script pentru a executa un raport care afișează funcția, numele și salariul fiecărui angajat al cărui salariu este sub 3000\$. Se adaugă ca header șirul "Employee Report", centrat, pe două rânduri și ca footer șirul "Confidential", centrat. Se redenumesc coloana JOB "Job category", despărțită pe

două rânduri. Se redenumeste coloana angajaților "Employee". Se redenumeste coloana pentru salariu "Salary" și se alege formatul corespunzător exemplului \$2,500.00

```
TTITLE 'EmployeeReport'
BTITLE 'Confidential'
COLUMN job HEADING 'Job|Category' Format A15
COLUMN ename HEADING 'Employee' Format A15
COLUMN sal HEADING 'Salary' Format $99,999.99
REM ** Insert SELECT statement
SELECT job,ename,sal
FROM scott.emp
WHERE sal<3000
ORDER BY job,ename
```

**Notă:** REM indică un comentariu în SQL\*Plus

### Probleme:

1. Scrieți o interogare care să afișeze numele, numărul departamentului și salariul oricărui angajat al cărui număr de departament și salariu să se potrivească cu numărul departamentului și salariul oricărui angajat care poate percepe comision.
2. Afișați numele, numele departamentului și salariul oricărui angajat al cărui salariu și comision coincide cu salariul și comisionul oricărui angajat ce lucrează în Dallas.
3. Scrieți o interogare care să afișeze numele, data angajării și salariul pentru toți angajații care au același salariu și comision ca angajatul Scott.
4. Scrieți o interogare care să afișeze angajații care câștigă un salariu mai mare decât al oricărui angajat cu funcția CLERK. Sortați rezultatele descrescător după salariu.
5. Scrieți un fișier script pentru a afișa numele fiecărui angajat, funcția și numele departamentului pentru acei angajați care lucrează în localitatea furnizată la prompt de utilizator. Condiția de căutare a localității trebuie să fie *case-insensitive* (să nu depindă de tipul caracterelor: majuscule sau litere mici). Salvați fișierul script cu numele *p1.sql*.

*Indicație:* pentru a efectua o căutare case-insensitive se va utiliza funcția LOWER (transformă șirul argument în caractere mici) sau UPPER (transformă șirul argument în majuscule).

6. Modificați *p1.sql* pentru a crea un raport ce conține numele departamentului, numele angajatului, data angajării, salariul și salariul anual pentru toți angajații dintr-o anumită localitate, introdusă de utilizator de la tastatură. Se etichetează coloanele DEPARTMENT NAME, EMPLOYEE NAME, START DATE, SALARY și ANNUAL SALARY, plasând etichetele formate din două șiruri pe mai multe linii. Se salvează fișierul script cu numele *p2.sql*
7. Afișați numărul managerilor (pe baza valorilor din coloana mgr) din tabelul emp. Etichetați coloana "Numar Manageri".
8. Scrieți o interogare care să afișeze diferența dintre salariul cel mai mare și cel mai mic din tabelul emp. Etichetați coloana "Diferența".
9. Afișați numărul fiecărui manager și salariul celui mai prost plătit subaltern al său. Exclueți angajații ce nu au manager. Exclueți grupurile care au salariul minim mai mic decât 1000\$. Sortați rezultatele în ordine descrescătoare după salariu.

10. Scrieți o interogare care să afișeze numele fiecărui departament, localitatea, numărul de angajați și salariul mediu al angajaților din acel departament. Etichetați coloanele Dname, Loc, Nr Persoane și Salariu.
11. Să se scrie o interogare care să afișeze numele și data angajării pentru toți angajații din același departament ca Blake (a se exclude Blake).
12. Să se scrie o cerere de interogare pentru a afișa numărul și numele pentru toți angajații care câștigă mai mult decât salariul mediu. Să se sorteze rezultatele în ordinea descrescătoare a salariului.
13. Să se scrie o cerere care va afișa numerele de înregistrare și numele angajaților din departamentele în care lucrează cel puțin un angajat al cărui nume conține litera 'T'.
14. Afișați numele, numărul departamentului și funcția deținută pentru toți angajații al căror departament este situat în Dallas.
15. Afișați numele și salariul tuturor angajaților subordonați lui King.
16. Afișați numărul departamentului, numele și funcția tuturor angajaților din departamentul de vânzări (Sales).
17. Afișați numărul, numele și salariul tuturor angajaților care câștigă mai mult decât salariul mediu și, în plus, lucrează într-un departament care deține cel puțin un angajat al cărui nume conține litera 'T'.