

CAPITOLUL 3

Rularea cererilor standard cu variabile substituie.

In acest capitol sunt descrise variabilele substituie. Variabilele substituie sunt folosite in declaratiile SQL si permite utilizarea valorilor specificate la rulare.

Variabile in SQL *Plus

In plus fata de variabilele pe care le vom vedea in limbajul PL/SQL, SQL*Plus este inezestrat cu doua tipuri de variabile ale sale:

Variabile legate

Utilizate pentru stocarea valorilor individuale ce pot fi asignate si citite in timpul rularii (vor fi discutate mai tarziu in acest curs).

Variabile substituie

Utilizate pentru stocarea partilor de text de comanda ,care sunt "editate" in comenzi inainte de executia lor.

Variabile substituie cu un singur ampersand.

Puteti utiliza 'variabile substituie' intr-un fisier de comanda sau intr-o declaratie SQL pentru a reprezenta valori ce vor fi aflate la executie.

O variabila poate fi vazuta ca un container in care valoarea este stocata temporar.

O variabila substituie este prefixata de un singur ampersand(&) si o valoare care ii este asignata.

Urmatoarea declaratie obliga utilizatorul sa dea un numar de departament la executie:

```
SELECT      EMPNO, ENAME, SAL
FROM        EMP
WHERE       DEPTNO = &DEPARTMENT_NUMBER;
```

```
Enter value for department_number:  10
```

```
EMPNO ENAME          SAL
-----
7782 CLARK            2,450.00
7839 KING             5,000.00
7934 MILLER           1,300.00
```

Exemplul anterior utilizeaza conditia WHERE DEPTNO=10.

Cu un singur ampersand utilizatorul este obligat sa dea o valoare de fiecare data cand comanda este executata, deoarece variabila nu este definita si de fiecare data valoarea introdusa nu este salvata .

O valoare de tip caracter sau data are nevoie sa fie inchisa in ghilimele simple cand sunt introduse la cerere. Pentru a evita ghilimelele simple ce trebuie sa fie introduse la executie ,puteti pune variabilele in ghilimele simple.

In urmatoarea declaratie ,variabila este pusa intre ghilimele simple, asa ca ghilimele nu mai sunt necesare la introducerea valorii respective:

```
SELECT      ENAME,DEPTNO,SAL*12
FROM        EMP
WHERE       JOB = ' <JOB_TITLE';
```

Enter value for job_title:MANAGER

ENAME	DEPTNO	SAL*12
JONES	20	35700
BLAKE	30	34200
CLARK	10	29400

Cat timp variabila este variabila 'nedefinita' ,valoarea va fi ceruta de fiecare data cand declaratia este executata.

Este posibil sa fie cerut numele unei coloane sau chiar numele unei tabele in timpul executiei.

In urmatorul exemplu vi se cere o expresie aritmetica:

```
SELECT      DEPTNO, <ARITHMETIC_EXPRESION
FROM        EMP;
```

Enter value for arithmetic_expression:sal/12

DEPTNO	SAL/12
20	66.6667
30	133.333
30	104.167
20	247.917
30	104.167
30	237.5
10	204.167
20	250
10	416.667
30	125
20	91.6667
30	79.1667
20	250

Variabile substituie cu dublu ampersand.

Daca variabila este prefixata de dublu ampersand(&&),SQL*Plus va cere valoarea pentru variabila doar o singura data .SQL*Plus stocheaza prima valoare furnizata si o foloseste din nou de fiecare data cand declaratia SQL este rulata

```
SELECT      ENAME,DEPTNO,JOB
FROM        EMP
WHERE       DEPTNO = &&DEPTNO_PLEASE;
```

Enter value for deptno_please: 10

```
ENAME      DEPTNO JOB
-----
CLARK       10  MANAGER
KING        10  PRESIDENT
MILLER      10  CLERK
```

Puteti folosi comanda SQL*Plus DEFINE pentru a determina daca variabila este deja definita.Daca variabila este deja definita se afiseaza valoarea asignata.

```
SQL> DEFINE
DEFINE DEPTNO_PLEASE = "10" (CHAR)
```

Comanda DEFINE este de asemenea folosita pentru a crea o variabila utilizator.

Substitutia datelor ca o comanda text

Cand informatia dintr-o comanda substituita este introdusa intr-o comanda inaintea executiei sale,variabila poate fi folosita pentru a inzebra orice parte a structurii comenzii exceptand numele insusi al comenzii.De exemplu:

```
SELECT      * FROM SALGRADE
WHERE       &CONDITION
```

Enter value for CONDITION: losal>2000

Variabila CONDITION ,de mai sus,trece intreaga conditie a clauzei WHERE intr-o comanda.Mai departe intregul cod al comenzii(incepand cu cuvantul SELECT) poate fi trecut intr-o variabila.

```
SELECT &THE_REST;
```

Comanda DEFINE

O valoare poate fi asignata unei variabile folosind comanda DEF[INE] a lui SQL*PLUS. Valoarea definita trebuie referita intr-o constructie SELECT sau un fisier de comenzi , prefixind numele variabilei cu &. Variabilele pot fi golite folosind UNDEF[INE].

In urmatorul exemplu , o variabila a fost definita folosind o expresie aritmetica ce calculeaza remuneratia. In urmatoarele constructii, variabila REM e referita de un numar de ori. Variabila e apoi golita folosind UNDEFINE.

```
SQL> DEFINE REM = 'SAL*12+NVL (COMM,0) '  
SQL>SELECT ENAME, JOB, <REM  
SQL>FROM EMP  
SQL>ORDER BY <REM;
```

```
SQL >undefine REM
```

ENAME	JOB	SAL*12+NVL (COMM,0)
SMITH	CLERK	9600
JAMES	CLERK	11400
ADAMS	CLERK	13200
WARD	SALESMAN	15500
MILLER	CLERK	15600
MARTIN	SALESMAN	16400
TURNER	SALESMAN	18000
ALLEN	SALESMAN	19500
CLARK	MANAGER	29400
BLAKE	MANAGER	34200
JONES	MANAGER	35700
SCOTT	ANALYST	36000
FORD	ANALYST	36000
KING	PRESIDENT	60000

" de la expresii sunt optionale , daca expresia nu contine spatii. Ghilimelele pot fi deasemenea folosite si sunt optionale dar folosite daca expresia contine blancuri sau apostrofi.

Pornirea unui fisier cu comenzi care contine substitutii de variabile

Presupunind ca doriti o serie de rapoarte care sa listeze angajatii cu functii diferite - de exemplu , una pentru SALESMEN , una pentru CLERKS, una pentru MANAGERS si tot asa.Cunoasteti cum sa folositi substitutia de variabile pentru a obtine aceste rapoarte dintr-o singura constructie SELECT.

Cu toate acestea , exista o tehnica alternativa pe care s-o folositi si anume sa scrieti constructia SELECT intr-un fisier si apoi folosind comanda START sa-l executati.

Aceasta tehnica impune folosirea unor substitutii speciale de variabile. Exista 9 asemenea variabile si ele au intregii de la 1 la 9 ca nume. Pentru folosirea acestor variabile , puneti un '&' urmat de un intreg (1-9) in comanda SQL. Aceste variabile pot fi folosite de ori de cite ori doriti si in orice ordine .

De fiecare data cind comanda e rulata, fiecare '&1' din comanda e inlocuit de primul parametru dupa START fisier, fiecare '&2' e inlocuit de al doilea parametru s.a.m.d.

Pentru a crea un fisier de comenzi care ia un parametru specificind functia care trebuie afisata , introduceti:

```
SELECT EMPNO,ENAME,SAL
FROM EMP
WHERE JOB = ' <1' ;
```

```
SQL>SAVE JOB1
```

```
SQL*PLUS intoarce mesajul:
```

```
Created file job1
```

```
Apoi rulati comanda cu parametrul 'CLERK'.
```

```
SQL*PLUS inlocuieste variabila cu valoarea parametrului.
```

```
SQL> START JOB1 CLERK
```

EMPNO	ENAME	SAL
7369	SMITH	800
7876	ADAMS	1100
7900	JAMES	950

Observati ca nu se pot folosi variabile cind executati o comanda cu RUN. Trebuie sa introduceti comanda in fisier si sa o rulati cu START comanda.

Comanda ACCEPT

Comanda ACCEPT permite unei variabile sa fie create si unei valori , care e introdusa sa fie memorata in ea. Aceasta variabila poate fi apoi referita in constructia SQL. ACCEPT este folosita deseori intr-un fisier de comenzi . Sunt citeva avantaje rezultate din folosirea lui ACCEPT pentru definirea substitutiei de variabile:

- Tipurile de date pot fi verificate
- Prompturile pot fi mai explicative
- Valorile de raspuns pot fi ascunse

Sintaxa comenzii este:

```
ACC[EPT] variabila [ NUMBER/CHAR] [PROMPT/NOPROMPT  
'text'] [HIDE]
```

Sintaxa + Descrierea

NUMBER/CHAR

determina tipul variabilei. Dacă valoarea introdusă este nevalidă, va fi afișat un mesaj.

PROMPT 'text'

afișează dacă text este specificat

NOPROMPT

face ACCEPT să sara o linie așteptând intrarea

HIDE

șterge răspunsul utilizatorului. folosit pentru parole

Exemple

```
SQL> ACCEPT SALARY NUMBER PROMPT 'Salary figure :'  
Salary figure : 30000  
SQL>ACCEPT PASSWORD CHAR PROMPT 'Password :' HIDE  
Password :  
SQL>ACCEPT COMM NUMBER NOPROMPT  
500  
SQL>DEFINE  
  
DEFINE SALARY =30000 (NUMBER)  
DEFINE PASSWORD ="FREEBIES (CHAR)  
DEFINE COMM =500 (NUMBER)
```

Rezumat

Când SQL*Plus întâlnește &variabilă:

1. Dacă &variabilă este deja definită, definiția deja existentă este folosită.
2. Dacă & variabilă nu este definită:
 - Utilizatorul este gata pentru definiție
 - Definiția furnizată de utilizator este apoi folosită
 - Definiția furnizată de utilizator este apoi pierdută

Când SQL Plus întâlnește &&variabilă :

La fel ca pașii 1 și 2 de deasupra cu excepția faptului că valoarea variabilei este salvată.

Observație: &1 parametru este tratat ca &&.

Pentru a afla dacă o variabilă este deja definită, folosiți comanda

DEF[INE]:

DEF DEPARTAMENT

daca e definit, da definitia lui DEPARTAMENT daca nu specifica 'nedefinit'

Pentru a defini o variabila , se poate folosi deasemenea comanda

DEF[INE].

DEF COMM =

Ghilimelele nu sunt necesare , dar ajuta daca "NVL (COMM,0)" expresia
contine blankuri imbricate sau apostrofi.

Cit timp ramim variabilele definite? Pina cind le UNDEF[ine] sau le redefinim sau
pina iesim din SQL*PLUS.

Alte 2 modalitati de definire a unei variabile:

```
SQL>ACCEPT variabila [tip] [PROMPT text][HIDE]
SQL>COL[UMN] nume coloana /alias NEW_VALUE variabila (tratata
in
capitolul 11)
```

Cap.3 Exerciții -Rulind cereri cu parametrii.

Acest exercitiu va ofera oportunitatea sa creati fisiere care pot fi rulate interactiv , si
care folosesc substitutii de variabile pentru crearea unor criterii de selectie.

TEME

1. Generati o constructie prin care sa afisati angajatii cu
date de
angajare intre 2 date date. Rulati de 2 ori cererea.
Modificati cererea pentru a folosi &&variabila .Rulati-o de
citeva
ori. Observati diferenta.

2. Obtineti o cerere care sa accepte o functie data. Pentru
testare executati cererea de citeva ori.

ENAME	JOB	SAL	MGR	DEPT	INO
-----	-----	-----	-----	-----	-----
SCOTT	ANALYST	3,000.00	7566		20
FORD	ANALYST	3,000.00	7566		20

3. Definiti o variabila reprezentind expresia folosita pentru
calculul remuneratiei anuale a angajatilor.Folositi
variabile
intr-o constructie care gaseste toti angajatii care
cistiga
30.000 \$ pe an sau mai mult.

ENAME	SAL*12+NUL (COMM,0)
JONES	35700
BLAKE	34200
SCOTT	36000
KING	60000
FORD	36000

Cap.3 Solutii

- ```

1. SELECT ENAME, HIREDATE
 FROM EMP
 WHERE HIREDATE
BETWEEN '&FIRST_DATE' AND '&LAST_DATE';

```
- ```

2.  SELECT      ENAME, JOB, SAL, MGR, DEPTNO
    FROM        EMP
    WHERE       JOB = '&JOB';

```
- ```

3. SELECT ENAME, &REM
 FROM EMP
 WHERE &REM > 30000;

```