



Universitatea
Politehnica
București



FACULTATEA DE
**AUTOMATICĂ ȘI
CALCULATOARE**
Universitatea POLITEHNICA din București



Catedra de
Calculatoare

Laborator 3

Rularea cererilor cu variabile substituie în SQL*Plus

Autori

Conf. Dr. Ing. Alexandru Boicea

Ș.L. Dr. Ing. Ciprian-Octavian Truică



Cuprins

- Crearea unei copii a unui tabel
- Variabile substituite
- Variabile substituite cu un singur ampersand(&)
- Variabile substituite cu dublu ampersand(&&)
- Variabile de sistem substituite cu un ampersand(&n)
- Variabile definite cu ACCEPT
- Variabile definite cu DEFINE și resetate cu UNDEFINE
- Exerciții



Crearea unei copii a unui tabel

- Metoda I
 - Se creează tabelul folosind comanda DDL CREATE;
 - Se iau informațiile din tabelul original și se inserează în copie folosind comanda INSERT cu un SELECT.
- Metoda II
 - Se folosește comanda CREATE cu SELECT pentru inserare.



Crearea unei copii a unui tabel

- Crearea unei copii a tabelului EMP

```
1 -- metoda 1
2 create table Angajati (
3     id_ang NUMBER(4),
4     nume VARCHAR2(20),
5     functie VARCHAR2(9),
6     data_ang DATE,
7     salariu NUMBER(7,2),
8     comision NUMBER(7,2),
9     id_dep NUMBER(2)
10 );
11
12 insert into Angajati select * from emp;
13
14 -- metoda 2
15 create table angajati as select * from emp;
```



Crearea unei copii a unui tabel

Observații :

- Niciuna dintre metode nu păstrează constrângerile de Primary Key, Unique, Foreign Key. Excepție este NOT NULL;
- Prima metoda permite schimbarea numelor de coloana, a doua metoda nu permite acest lucru;
- Pentru metoda I, dacă copia nu are același număr de coloane atunci trebuie să se specifice coloanele în care se vor insera date;
- Pentru metoda I, dacă copia nu respectă ordinea coloanelor din tabelul original, atunci trebuie să se specifice coloanele, atât pentru Insert cât și pentru Select.
- Pentru metoda II, coloanele vor avea același nume ca și coloanele din tabela originală.



Variabile substituie

- O cerere SQL poate fi executată folosind anumiți parametri introduși de la tastatură.
- Acești parametri se mai numesc variabile substituie.



Variabile substituite cu un singur ampersand(&)

- O astfel de variabilă se definește folosind **&nume_variabila** și este un parametru care se introduce de la tastatură în timpul execuției comenzii în care este utilizat.
- Parametrul cu un singur ampersand trebuie să fie introdus de fiecare dată, chiar dacă este folosit de mai multe ori în aceeași comandă SQL.



Variabile substituite cu un singur ampersand(&)

- Ex. 1. Să se selecteze id_ang, nume, functie și salariu pentru angajatii care au același sef. Id-ul șefului se va introduce de la tastatură.

```
1 SELECT id_ang, nume, functie, salariu
2     FROM angajati
3     WHERE id_sef = &id_sef;
```




Variabile substituite cu un singur ampersand(&)

- Rezultatul execuției exercițiului 1:

```
SQL> select id_ang, nume, functie, salariu
  2  from angajati
  3  where id_sef = &id_sef;
Enter value for id_sef: 7839
old   3: where id_sef = &id_sef
new   3: where id_sef = 7839
```

| ID_ANG | NUME | FUNCTIE | SALARIU |
|--------|-------|---------|---------|
| 7566 | JONES | MANAGER | 2975 |
| 7698 | BLAKE | MANAGER | 2850 |
| 7782 | CLARK | MANAGER | 2450 |



Variabile substituite cu un singur ampersand(&)

- Dacă nu se dorește afișarea valorilor vechi și noi se poate face următoarea setare: **SET VERIFY OFF**

```
SQL> set verify off
SQL> select id_ang, nume, functie, salariu
  2  from angajati
  3  where id_sef = &id_sef;
Enter value for id_sef: 7839
```

| ID_ANG | NUME | FUNCTIE | SALARIU |
|--------|-------|---------|---------|
| 7566 | JONES | MANAGER | 2975 |
| 7698 | BLAKE | MANAGER | 2850 |
| 7782 | CLARK | MANAGER | 2450 |



Variabile substituite cu un singur ampersand(&)

- Pentru a introduce de la tastatură valori de tip string, sau dată, acestea trebuie să fie introduse cu apostrof;
- Pentru a evita acest lucru, se poate introduce variabila cu apostrof direct în comandă.



Variabile substituite cu un singur ampersand(&)

- Ex. 2. Să se selecteze numele, funcția și salariu pentru angajații care au aceeași funcție. Funcția va fi introdusa de la tastatură.

```
1 SELECT nume, '&functie' functie, salariu
2     FROM angajati
3     WHERE functie = &functie;
```



Variabile substituite cu un singur ampersand(&)

- Rezultatul execuției exercițiului 2:

```
SQL> select nume, '&functie' functie, salariu
  2   from angajati
  3   where functie = &functie;
Enter value for functie: Analist Programator
Enter value for functie: 'ANALYST'
```

| NUME | FUNCTIE | SALARIU |
|-------|---------------------|---------|
| SCOTT | Analist Programator | 3000 |
| FORD | Analist Programator | 3000 |

Observații:

- Atenție unde se pune apostrof;
- Cât timp variabila este *variabilă nedefinită*, valoarea va fi cerută de fiecare dată când se execută cererea.



Variabile substituite cu un singur ampersand(&)

- Este posibil să fie introduse de la tastatură o expresie, numele unei coloane sau chiar numele unei tabele în timpul execuției.
- Ex. 3. Să se selecteze numele, funcția și salariul anual pentru toți angajații din același departament. Salariul anual este calculat după formula $12 * \text{salariu}$. Se va introduce de la tastatură funcția pentru salariul anual, numele tabelului, numele coloanei pe care se pune condiția și id-ul departamentului.

```
1 SELECT nume, functie, &salariu_anual salariu_anual
2     FROM &tabel
3     WHERE &nume_coloana = &valoare_coloana;
```



Variabile substituite cu un singur ampersand(&)

- Rezultatul execuției exercițiului 3:

```
SQL> select nume, functie, &salariu_anual salariu_anual
  2  from &tabel
  3  where &nume_coloana = &valoare_coloana;
Enter value for salariu_anual: salariu*12
Enter value for tabel: angajati
Enter value for nume_coloana: id_dep
Enter value for valoare_coloana: 20
```

| NUME | FUNCTIE | SALARIU_ANUAL |
|-------|---------|---------------|
| SMITH | CLERK | 9600 |
| JONES | MANAGER | 35700 |
| SCOTT | ANALYST | 36000 |
| ADAMS | CLERK | 13200 |
| FORD | ANALYST | 36000 |



Variabile substituite cu dublu ampersand(&&)

- Spre deosebire de variabilele cu un singur ampersand, o variabilă cu dublu ampersand va fi stocată și poate fi apelată pe toată sesiunea de lucru;
- Definirea se face similar **&&nume_variabila** și va fi cerută o singură dată;
- Folosirea ei de mai multe ori în cadrul comenzii se face apelând-o cu **&nume_variabila**.



Variabile substituite cu dublu ampersand(&&)

- Ex. 4. Să se selecteze numele, funcția și venitul lunar pentru toți angajații care au venitul lunar mai mare de 2000. Venitul lunar este o funcție care are formula salariu + nvl(comision, 0).

```
1 SELECT nume, functie, &&venit_lunar venit_lunar  
2 FROM angajati  
3 WHERE &venit_lunar > 2000;
```



Variabile substituite cu dublu ampersand(&&)

- Rezultatul execuției exercițiului 4:

```
SQL> select nume, functie, &&venit_lunar venit_lunar
  2  from angajati
  3  where &venit_lunar > 2000;
Enter value for venit_lunar: salariu+nvl(comision, 0)
```

| NUME | FUNCTIE | VENIT_LUNAR |
|--------|-----------|-------------|
| JONES | MANAGER | 2975 |
| MARTIN | SALESMAN | 2650 |
| BLAKE | MANAGER | 2850 |
| CLARK | MANAGER | 2450 |
| SCOTT | ANALYST | 3000 |
| KING | PRESIDENT | 5000 |
| FORD | ANALYST | 3000 |

7 rows selected.



Variabile substituite cu dublu ampersand(&&)

- Obs. 1. Pentru a reseta o variabilă dată cu dublu ampersant se va utiliza comanda **UNDEFINE nume_variabila**. Pentru exercițiul 4, pentru a reseta variabila venit_lunar, se va da comanda:

SQL> UNDEFINE venit_lunar.

- Obs. 2. Funcția nvl(param1, param2) este o funcție care verifică dacă param1 este NULL și, în cazul în care param1 este NULL, va substitui valoarea parametrului 1 cu valoarea parametrului 2. Această funcție este implementată în toate RDBMS-urile (Sistem de Gestiune a Bazelor de Date Relaționale)



Variabile de sistem substituite cu un ampersand(&n)

- Sunt variabile numerice (1-9) care sunt definite de sistem și funcționează similar cu variabilele cu dublu ampersand;
- Avantajul folosirii acestor variabile este că pot fi apelate direct dintr-un fișier cu comenzi, fără a fi definite în prealabil.



Variabile de sistem substituite cu un ampersand(&n)

- Ex. 5. Să se selecteze id-ul angajatului, numele, funcția și data angajării pentru toți angajații care au aceeași funcție și au venit în firmă după o anumită dată. Funcția și data se vor da de la tastatură în momentul când se execută fișierul de comenzi.
 - Se va scrie următorul select într-un fișier:

```
1 SELECT id_ang, nume, functie, data_ang
2     FROM angajati
3     WHERE functie = '&1' AND data_ang > '&2'
4     ORDER BY data_ang;
```



Variabile de sistem substituite cu un ampersand(&n)

- Execuția fișierului:

```
SQL> @D:\ang.sql ANALYST 15-AUG-1981
```

| ID_ANG | NUME | FUNCTIE | DATA_ANG |
|--------|-------|---------|-----------|
| 7902 | FORD | ANALYST | 03-DEC-81 |
| 7788 | SCOTT | ANALYST | 09-DEC-82 |

```
SQL> start D:\ang.sql CLERK 15-AUG-1981
```

| ID_ANG | NUME | FUNCTIE | DATA_ANG |
|--------|--------|---------|-----------|
| 7900 | JAMES | CLERK | 03-DEC-81 |
| 7934 | MILLER | CLERK | 23-JAN-82 |
| 7876 | ADAMS | CLERK | 12-JAN-83 |



Variabile de sistem substituite cu un ampersand(&n)

- Observatii:
- Cluaza **ORDER BY** este folosită pentru a ordona rezultatul în funcție de **coloanele** specificate;
- Pentru a le ordona ascendent, cuvântul cheie **ASC** poate să fie specificat după numele coloanei după care se face sortarea, sau să lipsească;
- Pentru a sorta descrescător, se va folosi cuvântul cheie **DESC** care va fi specificat după numele coloanei după care se face sortarea.



Variabile definite cu ACCEPT

- Când definim o variabilă cu ampersand, promptul va fi numele variabilei;
- Folosind comanda ACCEPT, se poate redefini promptul și chiar se pot ascunde caracterele introduse de la tastatură.



Variabile definite cu ACCEPT

- Ex. 6. Să se selecteze numele, salariul și funcția angajaților care au aceeași funcție. Funcția se va da de la tastatură.
- Se vor scrie următoarele comenzi într-un fisier (de exemplu ex6.sql)

```
1 accept functie_ang char prompt 'Introduceti functia angajatului:'
2
3 SELECT nume, salariu, comision
4     FROM angajati
5     WHERE functie = '&functie_ang';
6
7 undefine functie_ang
```



Variabile definite cu ACCEPT

- Execuția fișierul:

```
SQL> @d:\ex6.sql
Introduceti functia angajatului:MANAGER

NUME                SALARIU             COMISION
-----
JONES                2975
BLAKE                2850
CLARK                2450
```



Variabile definite cu ACCEPT

- Ex. 7. Să se adauge un nou angajat. La introducerea salariului caracterele să fie ascunse.

```
1 accept id_ang char prompt 'Introduceti ecusonul angajatului:'
2 accept nume char prompt 'Introduceti numele angajatului:'
3 accept functie char prompt 'Introduceti functia angajatului:'
4 accept salariu char prompt 'Introduceti salariul angajatului:' hide
5
6 INSERT INTO angajati(id_ang, nume, functie, salariu)
7     VALUES(&id_ang, '&nume', '&functie', &salariu);
8
9 undefine id_ang
10 undefine functie
11 undefine nume
12 undefine salariu
```



Variabile definite cu ACCEPT

- Execuția fișierului:

```
SQL> set verify off
SQL> @d:\ex7.sql
Introduceti ecusonul angajatului:1111
Introduceti numele angajatului:IONESCU
Introduceti functia angajatului:CLERK
Introduceti salariul angajatului:
1 row created.

SQL> set verify on
SQL> @d:\ex7.sql
Introduceti ecusonul angajatului:1112
Introduceti numele angajatului:POPESCU
Introduceti functia angajatului:ANALYST
Introduceti salariul angajatului:
old 2:          values (&id_ang, '&nume', '&functie', &salariu)
new 2:          values (1112, 'POPESCU', 'ANALYST', 2345)
1 row created.
```



Variabile definite cu DEFINE și resetate cu UNDEFINE

- O variabilă poate să fie setată folosind comanda DEFINE, în acest caz promptul nu va mai cere să se introducă o valoare pentru respectiva variabilă;
- O astfel de variabilă rămâne setată pe tot parcursul sesiunii, până când va fi resetată folosindu-se comanda UNDEFINE.



Variabile definite cu DEFINE și resetate cu UNDEFINE

- Ex. 8. Să se calculeze o primă de 15% pentru toți angajații din departamentul 20.

```
1 define procent_prima = 0.15
2 define id_dep = 20
3
4 SELECT nume, salariu, salariu*&procent_prima prima
5     FROM angajati
6     WHERE id_dep = &id_dep;
7
8 undefine procent_prima
9 undefine id_dep
```



Variabile definite cu DEFINE și resetate cu UNDEFINE

- Rezultatul execuției exercițiului 8:

```
SQL> define procent_prima = 0.15
SQL> define id_dep = 20
SQL> select nume, salariu, salariu*&procent_prima prima
      2  from angajati
      3  where id_dep=&id_dep;
old   1: select nume, salariu, salariu*&procent_prima prima
new   1: select nume, salariu, salariu*0.15 prima
old   3: where id_dep=&id_dep
new   3: where id_dep=20
```

| NUME | SALARIU | PRIMA |
|-------|---------|--------|
| SMITH | 800 | 120 |
| JONES | 2975 | 446.25 |
| SCOTT | 3000 | 450 |
| ADAMS | 1100 | 165 |
| FORD | 3000 | 450 |



Variabile definite cu DEFINE și resetate cu UNDEFINE

- Pentru a vedea setările tuturor variabilelor active, folosiți comanda **DEFINE**.

```
SQL> define
DEFINE _DATE = "02-MAR-14" <CHAR>
DEFINE _CONNECT_IDENTIFIER = "orcl" <CHAR>
DEFINE _USER = "STUDENT" <CHAR>
DEFINE _PRIVILEGE = "" <CHAR>
DEFINE _SQLPLUS_RELEASE = "1102000100" <CHAR>
DEFINE _EDITOR = "Notepad" <CHAR>
DEFINE _O_VERSION = "Oracle Database 11g Enterprise Edition Release 11.2.0.
1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options" <
CHAR>
DEFINE _O_RELEASE = "1102000100" <CHAR>
DEFINE 1 = "ANALYST" <CHAR>
DEFINE 2 = "15-AUG-1981" <CHAR>
DEFINE _RC = "0" <CHAR>
DEFINE PROCENT_PRIMA = "0.15" <CHAR>
DEFINE ID_DEP = "20" <CHAR>
```




Exerciții

- 1. Să se scrie o comandă SQL care listează toți angajații dintr-un departament (introdus ca parametru de la tastatură), care au venitul anual peste un venit mediu anual (introdus tot de la tastatură)
- Dați cel puțin 4 metode de rezolvare.