SISTEME DE GESTIUNE A BAZELOR DE DATE

AN UNIVERSITAR 2019-2020

INFORMATICĂ AN III SEM I

Lect. Univ. Dr. Gabriela Mihai

Recapitulare

1. O variabilă de tip *nume_tabel%rowtype* păstrează restricțiile de *not null*, respectiv valorile *default* definite la nivel de tabel?

```
create table test (id number(2) not null,
                   a  number(2) default 1);
declare
  t test%rowtype;
begin
  dbms_output.put_line(nvl(t.id,0)||' '||nvl(t.id,0));
end;
/
```

- **Output:** 0 0

Recapitulare

2. Lucrul cu tipuri de date colecție

```
create or replace package pachet_test
AS
 TYPE tip_imbricat IS TABLE OF NUMBER(4);
 TYPE tip_indexat  IS TABLE OF NUMBER(4) INDEX BY
                                         PLS_INTEGER;
 x tip_imbricat;
 y tip_indexat;
end;
/
```

Recapitulare

```
declare
    k pls_integer;
begin
    --tablou imbricat
    select employee_id bulk collect into pachet_test.x
    from   employees
    where  employee_id<=105;

    dbms_output.put_line('tablou imbricat:');
    for i in pachet_test.x.first..pachet_test.x.last loop
       dbms_output.put(pachet_test.x(i)|| ' ');
    end loop;
    dbms_output.new_line;
```

Recapitulare

```
   --tablou indexat
   for i in (select employee_id
                from   employees
                where  employee_id<=105) loop
     pachet_test.y(i.employee_id) := i.employee_id;
    end loop;

    dbms_output.put_line('tablou indexat:');
    k := pachet_test.y.first;
    while k <= pachet_test.y.last loop
        dbms_output.put(pachet_test.y(k)|| ' ');
        k := pachet_test.y.next(k);
    end loop;
    dbms_output.new_line;
end;
/
```

Recapitulare

- **Output:**

```
tablou imbricat:
100 101 102 103 104 105

tablou indexat:
100 101 102 103 104 105
```

Recapitulare

```
declare
    k pls_integer;
begin
    --tablou imbricat
    select employee_id bulk collect into pachet_test.x
    from   employees
    where  employee_id between 106 and 110;

    dbms_output.put_line('tablou imbricat:');
    for i in pachet_test.x.first..pachet_test.x.last loop
       dbms_output.put(pachet_test.x(i)|| ' ');
    end loop;
    dbms_output.new_line;
```

Recapitulare

```
   --tablou indexat
   for i in (select employee_id
                from   employees
                where  employee_id between 106 and 110) loop
     pachet_test.y(i.employee_id) := i.employee_id;
    end loop;

    dbms_output.put_line('tablou indexat:');
    k := pachet_test.y.first;
    while k <= pachet_test.y.last loop
        dbms_output.put(pachet_test.y(k)|| ' ');
        k := pachet_test.y.next(k);
    end loop;
    dbms_output.new_line;
end;
/
```

Recapitulare

- **Output:**

```
tablou imbricat:
106 107 108 109 110

tablou indexat:
100 101 102 103 104 105 106 107 108 109 110
```

Recapitulare

3. Exemplu *trigger-i – table mutating.*

   Implementarea restricției conform căreia într-un departament pot lucra maxim 45 de angajați.

```
CREATE OR REPLACE PACKAGE pachet
AS
   TYPE tip_rec IS RECORD
                   (id employees.department_id%TYPE,
                    nr NUMBER(2));
   TYPE tip_imbricat IS TABLE OF tip_rec;
   TYPE tip_indexat IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
   t       tip_imbricat;
   contor tip_indexat;
END;
/
```

Recapitulare

```
CREATE TABLE emp_copie AS
SELECT employee_id, last_name, department_id
FROM   employees;

CREATE OR REPLACE TRIGGER trig_comanda
BEFORE INSERT OR UPDATE OF department_id ON emp_copie
BEGIN
  pachet.contor.delete;

  select nvl(department_id,0), count(*)
  bulk   collect into pachet.t
  from   emp_copie
  group by department_id
  order by 1;
END;
/
```

Recapitulare

```
CREATE OR REPLACE TRIGGER trig_linie
BEFORE INSERT OR UPDATE OF department_id ON emp_copie
FOR EACH ROW
BEGIN
  IF pachet.contor.EXISTS(:NEW.department_id)=FALSE
     THEN pachet.contor(:NEW.department_id) := 0;
  END IF;
  FOR i in 1..pachet.t.last LOOP
    IF pachet.t(i).id = :NEW.department_id
       AND pachet.t(i).nr +
           pachet.contor(:NEW.department_id)>=45 THEN
     RAISE_APPLICATION_ERROR(-20000,'Restrictie incalcata');
    END IF;
  END LOOP;
  pachet.contor(:NEW.department_id) :=
      pachet.contor(:NEW.department_id)+1;
END;
```

Recapitulare

```
-- in dept 40 avem 1 angajat
-- linia este inserata
INSERT INTO emp_copie
VALUES (300, 'Ang 300',40);

-- in dept 50 avem deja 45 de angajati
-- linia nu este inserata
INSERT INTO emp_copie
VALUES (301, 'Ang 301',50);
```

Recapitulare

```
-- creez un tabel sursa pentru insert
create sequence id_emp start with 350;
create table emp_sursa as select * from emp_copie where 0=1;
begin
   for i in 1..12 loop
      insert into emp_sursa
      values (id_emp.nextval,'Ang '||id_emp.currval,80);
   end loop;
   insert into emp_sursa
   values (id_emp.nextval,'Ang '||id_emp.currval,50);
   insert into emp_sursa
   values (id_emp.nextval,'Ang '||id_emp.currval,20);
   commit;
end;
/
```

Recapitulare

```
--in dept 80: 34ang, iar prin insert din emp_sursa aduc 12ang
--in dept 50: 45ang, iar prin insert din emp_sursa aduc 1ang
--in dept 20: 2ang, iar prin insert din emp_sursa aduc 1ang

-- liniile nu sunt inserate
INSERT INTO emp_copie
SELECT * FROM emp_sursa;

-- liniile nu sunt inserate
INSERT INTO emp_copie
SELECT * FROM emp_sursa WHERE department_id <> 50;

-- linia este inserata; pentru dept 20 nu incalc restrictia
INSERT INTO emp_copie
SELECT * FROM emp_sursa WHERE department_id not in (50, 80);
```

Recapitulare

```
-- dept 50 are 45 de ang
-- dept 10 are 1 ang
-- actualizarea nu este permisa
UPDATE emp_copie
SET    department_id = 50
WHERE  department_id = 10;

-- dept 80 are 34 de ang
-- dept 60 are 5 ang
-- dept 70 are 1 ang
-- actualizarea este permisa
UPDATE emp_copie
SET    department_id = 80
WHERE  department_id in (60,70);
```

Recapitulare

```
-- acum dept 80 are 40 de ang
-- dept 20 are 3 ang
-- dept 90 are 3 ang
-- actualizare nepermisa
UPDATE emp_copie
SET    department_id = 80
WHERE  department_id in (20,90);
```

Recapitulare

```
-- acum dept 40 are 2 ang
--      dept 50 are 45 ang
--      dept 60 are 0 ang
--      dept 70 are 0 ang
--      dept 80 are 40 ang
--      dept 100 are 6 ang
-- actualizarea nu este permisa
UPDATE emp_copie
SET    department_id = department_id + 20
where  department_id IN (40,50,80);

--actualizarea este permisa
UPDATE emp_copie
SET    department_id = department_id + 20
where  department_id IN (40,50);
```