Együttes hozzárendelés

Együttes hozzárendelés

Hozzárendelés: az a tevékenységet, amikor egy PL/SQL-változónak egy SQL-utasításban adunk értéket.

Együttes hozzárendelés: Egy kollekció minden elemének egyszerre történő hozzárendelése.

FORALL: PL/SQL-oldali együttes hozzárendelés eszköze

BULK COLLECT: SQL-oldali együttes hozzárendelés eszköze.

BULK COLLECT

- A SELECT, INSERT, DELETE, UPDATE, FETCH utasítások INTO utasításrészében használható
- Az SQL-motortól az együttes hozzárendelést kéri
- Alakja:

```
BULK COLLECT INTO
  kollekciónév [, kollekciónév]…
```

- A kollekciókat nem kell inicializálni.
- Az adatokat a kollekcióban az 1-es indextől kezdve helyezi el folyamatosan, felülírva a korábbi elemeket.
- Csak szerveroldali programokban alkalmazható.

```
declare
type t_nt is table of orszagok%rowtype;
v_nt t_nt;

begin
select *
bulk collect into v_nt
from orszagok
where foldresz='Európa';
for i in 1..v_nt.count
    loop

  dbms_output.put_line(v_nt(i).orszag);
    end loop;
end;
```

```
create table kis_orszagok
  (nev varchar2(50),
   terulet number(10),
   foldresz varchar2(50));

insert into kis_orszagok
select orszag, terulet, foldresz
from orszagok
where terulet<10000;

commit;
…
```

```
…
declare
type t_at_nev is table of orszagok.orszag%type
   index by pls_integer;
type t_at_ter is table of orszagok.terulet%type
   index by pls_integer;
v_at_nev t_at_nev;
v_at_ter t_at_ter;
begin
 update kis_orszagok
 set terulet=terulet*1.1
 where foldresz='Európa'
 returning nev, terulet
 bulk collect into v_at_nev,v_at_ter;
commit;
for i in 1..v_at_nev.count
    loop
    dbms_output.put_line(v_at_nev(i)||v_at_ter(i));
    end loop;
end;
```

```
create or replace
type t_dt_emp is varray(10) of varchar2(50);

create or replace
FUNCTION reszleg_dolgozoi(p_reszleg varchar2)
   RETURN t_dt_emp IS
  v_dt_emp      t_dt_emp;
-- BULK COLLECT-nél nem kell inicializálni
BEGIN
  SELECT first_name||' '||last_name nev
    BULK COLLECT INTO v_dt_emp
    FROM hr.employees
    WHERE department_id = (select department_id
                    from hr.departments
                    where department_name=p_reszleg)
   ORDER BY nev;
  RETURN v_dt_emp;
END;
/
SELECT * FROM TABLE(reszleg_dolgozoi('IT'));
```

```
declare
  type t_dt_emp is varray(10) of hr.employees%rowtype;
  v_dt_emp      t_dt_emp;
  cursor c_emp (p_reszleg hr.departments.department_name%type)
    is  select * FROM hr.employees
        WHERE department_id = (select department_id
                               from hr.departments
                               where department_name=p_reszleg)
        ORDER BY last_name;
BEGIN
  open c_emp('Sales');
  fetch c_emp bulk collect into v_dt_emp;
  close c_emp;
  for i in 1..v_dt_emp.count
    loop
    dbms_output.put_line(
       v_dt_emp(i).first_name||' '||v_dt_emp(i).last_name);
    end loop;
END;
/
```

```
declare
  type t_dt_emp is varray(10) of hr.employees%rowtype;
  v_dt_emp      t_dt_emp;
  cursor c_emp (p_reszleg hr.departments.department_name%type) is
                  select *
                  FROM hr.employees
                  WHERE department_id = (select department_id
                                         from hr.departments
                                         where
   department_name=p_reszleg)
                  ORDER BY last_name;
BEGIN
  open c_emp('Sales');
  loop
  fetch c_emp bulk collect into v_dt_emp limit 10;
  exit when v_dt_emp.count=0;
  for i in 1..v_dt_emp.count
    loop
    dbms_output.put_line(
       v_dt_emp(i).first_name||' '||v_dt_emp(i).last_name);
    end loop;
  end loop;
  close c_emp;
END;
/
```

Probléma

```
CREATE TABLE employees2 AS
   SELECT * FROM HR.employees;

DECLARE
TYPE NumList IS VARRAY(20) OF NUMBER;
depts NumList := NumList(10, 30, 70);
   --részlegszámok
BEGIN
FOR i IN depts.FIRST..depts.LAST
   LOOP
    DELETE FROM employees2
    WHERE department_id = depts(i);
   END LOOP;
COMMIT;
END;
```

FORALL: PL/SQL-oldali együttes hozzárendelés eszköze

FORALL *index* IN {*alsó_határ*..*felső_határ*
| INDICES OF *kollekció*
[BETWEEN *alsó_határ* AND *felső_határ*]
| VALUES OF *indexkollekció_név*}
[SAVE EXCEPTIONS]
sql_utasítás;

- Az SQL-motor az SQL-utasítást a megadott indextartomány minden értéke mellett egyszer végrehajtja.
- Csak szerverotldali programokban alkalmazható.

FORALL: PL/SQL-oldali együttes hozzárendelés eszköze

- A FORALL implicit módon a kollekció indexének megfelelő típusúnak deklarálja az indexet
- INDICES OF kollekció: a megadott kollekció elemeinek az indexeit veszi fel
  - BETWEEN-nel korlátozhatjuk
- VALUES OF *indexkollekció_név*: az index által felveendő értékekekt az indexkollekció_név által megnevezett kollekció tartalmazza

```
CREATE TABLE employees2 AS
  SELECT * FROM HR.employees;

DECLARE
TYPE NumList IS VARRAY(20) OF NUMBER;
depts NumList := NumList(10, 30, 70);
  --részlegszámok
BEGIN
FORALL i IN depts.FIRST..depts.LAST
    DELETE FROM employees2
    WHERE department_id = depts(i);
COMMIT;
END;
```

```
CREATE TABLE employees2 AS
   SELECT * FROM employees;

DECLARE
TYPE NumList IS VARRAY(10) OF NUMBER;
depts NumList :=
  NumList(5,10,20,30,50,55,57,60,70,75);
BEGIN
FORALL j IN 4..7 – varray bizonyos elemeire
DELETE FROM employees2
  WHERE department_id = depts(j);
COMMIT;
END;
```

```
CREATE TABLE t1
(id number(5),
adat varchar2(20));
insert into t1 values(1,'x');
insert into t1 values(2,'x');
insert into t1 values(3,'x');
insert into t1 values(4,'x');
commit;

SELECT * FROM t1;
/*
ID                      ADAT
----------------------- --------------------
1                       x
2                       x
3                       x
4                       x
*/
```

…

```
…
DECLARE
  TYPE T_Id_lista IS TABLE OF NUMBER;
  v_Id_lista T_Id_lista := T_Id_lista(3,1);
BEGIN
  -- Törüljük az 1. elemet, 1 elem marad
  v_Id_lista.DELETE(1);

  FORALL i IN INDICES OF v_Id_lista  -- i: {2}
    UPDATE t1 SET adat = 'INDICES OF'
     WHERE id   = v_Id_lista(i);
    -- T_Id_lista(2) = 1 -> 1-es Id-re fut le az UPDATE
END;
/

SELECT * FROM t1;
/*
ID                     ADAT
---------------------- --------------------
1                      INDICES OF
2                      x
3                      x
4                      x
*/
…
```

```
…
DECLARE
  TYPE T_Id_lista    IS TABLE OF NUMBER;
  TYPE T_Index_lista IS TABLE OF BINARY_INTEGER;

  v_Id_lista    T_Id_lista    := T_Id_lista(5,1,4,3,2);
  v_Index_lista T_Index_lista := T_Index_lista(4,5);
BEGIN
  FORALL i IN VALUES OF v_Index_lista  -- i: {4, 5}
    UPDATE t1
       SET adat = 'VALUES OF'
     WHERE id   = v_Id_lista(i);
  -- p_Id_lista(4) = 3
  -- p_Id_lista(5) = 2
  --   -> 3-as és 2-es Id-kre fut le az UPDATE
END;
/
SELECT * FROM t1;
/*
ID                     ADAT
---------------------- --------------------
1                      INDICES OF
2                      VALUES OF
3                      VALUES OF
4                      x
*/

DROP TABLE t1;
```

```
drop table forall_kivetel;
create table forall_kivetel
(szam number(1));

declare
type t_nt is table of number(5);
v_nt t_nt:=t_nt(1,30,5);

begin
forall i in indices of v_nt
insert into forall_kivetel
values (v_nt(i));
/*exception
  when others
  then null;*/
end;
/

select *
from forall_kivetel;
```

```
CREATE TABLE emp2 AS SELECT * FROM hr.employees;

DECLARE
TYPE NumList IS TABLE OF NUMBER;
depts NumList := NumList(30, 50, 60);
BEGIN
FORALL j IN depts.FIRST..depts.LAST
DELETE FROM emp2
   WHERE department_id = depts(j);
FOR i IN depts.FIRST..depts.LAST
  LOOP
  dbms_output.put_line('Iteration #'
    || i || ' deleted '
    || SQL%BULK_ROWCOUNT(i)
    || ' rows.');
   END LOOP;
END;
```

```
CREATE TABLE emp_by_dept AS SELECT employee_id, department_id
  FROM hr.employees WHERE 1 = 0;

DECLARE
TYPE dept_tab IS TABLE OF hr.departments.department_id%TYPE;
deptnums dept_tab;
BEGIN
SELECT department_id BULK COLLECT INTO deptnums
FROM hr.departments;
FORALL i IN 1..deptnums.COUNT
INSERT INTO emp_by_dept
SELECT employee_id, department_id
FROM hr.employees
WHERE department_id = deptnums(i);
FOR i IN 1..deptnums.COUNT LOOP
  dbms_output.put_line('Dept '||deptnums(i)
    || ': inserted '|| SQL%BULK_ROWCOUNT(i)||' records');
END LOOP;
dbms_output.put_line('Total records inserted ='
  || SQL%ROWCOUNT);
END;
```

```
CREATE TABLE emp2 AS SELECT * FROM employees;

DECLARE
TYPE NumList IS TABLE OF NUMBER;
num_tab NumList := NumList(10,0,11,12,30,0,20,199,2,0,9,1);
errors NUMBER;
dml_errors EXCEPTION;
PRAGMA exception_init(dml_errors, -24381);

BEGIN
 FORALL i IN num_tab.FIRST..num_tab.LAST SAVE EXCEPTIONS
 DELETE FROM emp2 WHERE salary > 500000/num_tab(i);
EXCEPTION
 WHEN dml_errors THEN
  errors := SQL%BULK_EXCEPTIONS.COUNT;
  dbms_output.put_line(
     'Number of DELETE statements that failed: ' || errors);
  FOR i IN 1..errors LOOP
      dbms_output.put_line('Error #' || i ||
       ' occurred during '||
       'iteration #' || SQL%BULK_EXCEPTIONS(i).ERROR_INDEX);
      dbms_output.put_line('Error message is ' ||
      SQLERRM(-SQL%BULK_EXCEPTIONS(i).ERROR_CODE));
  END LOOP;
END;
```

```
drop table forall_kivetel;
create table forall_kivetel(szam number(1));

declare
 type t_nt is table of number(5);
 v_nt t_nt:=t_nt(1,30,5,40,6,10,1,3);
begin
forall i in indices of v_nt save exceptions
insert into forall_kivetel values (v_nt(i)) ;
exception when others
  then for i in 1..sql%bulk_exceptions.count loop
         dbms_output.put_line(i|| '; ‘
         || sql%bulk_exceptions(i).error_index
         || '; ‘ || sql%bulk_exceptions(i).error_code);
         end loop;
end;
/
select *
from forall_kivetel;
```

```
CREATE TABLE emp2 AS SELECT * FROM employees;
DECLARE
TYPE NumList IS TABLE OF NUMBER;
depts NumList := NumList(10,20,30);
TYPE enum_t IS TABLE OF employees.employee_id%TYPE;
TYPE dept_t IS TABLE OF employees.department_id%TYPE;
e_ids enum_t;
d_ids dept_t;
BEGIN
FORALL j IN depts.FIRST..depts.LAST
DELETE FROM emp2 WHERE department_id = depts(j)
RETURNING employee_id, department_id
  BULK COLLECT INTO e_ids, d_ids;
dbms_output.put_line('Deleted ' ||
   SQL%ROWCOUNT || ' rows:');
FOR i IN e_ids.FIRST .. e_ids.LAST
  LOOP
   dbms_output.put_line('Employee #'
    || e_ids(i) ||' from dept #' || d_ids(i));
  END LOOP;
END;
```