

Sistemi baza podataka

Slavica Aleksić

slavica@uns.ns.ac.yu

Zadatak – primer ugnježenih kursora

Napisati PL/SQL blok koji će:

- Ispisati sve radnike po sektorima sa određenim šefom

Rešenje

```
DECLARE
  CURSOR sefovi
  IS SELECT DISTINCT sef FROM radnik WHERE Sef IS NOT NULL;
  CURSOR radnici (p_sef in number)
  IS SELECT * FROM radnik WHERE sef = p_sef;
  v_sef Radnik.sef%type;
  v_ime Radnik.ime%type;
  v_prz Radnik.prz%type;
  v_radnik Radnik%rowtype;

BEGIN
  OPEN sefovi;
  LOOP
    FETCH sefovi INTO v_sef;
    EXIT WHEN sefovi%NOTFOUND;
    SELECT ime, prz INTO v_ime, v_prz FROM radnik WHERE Mbr = v_sef;
    DBMS_OUTPUT.PUT_LINE("");
    DBMS_OUTPUT.PUT_LINE('Radnici kojima je sef ' || v_ime || ' ' || v_prz);
    DBMS_OUTPUT.PUT_LINE("");
    OPEN radnici(v_sef);
    LOOP
      FETCH radnici INTO v_radnik;
      EXIT WHEN radnici%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Ime zaposlenog ' || ' je ' || v_radnik.ime || ' ' || v_radnik.prz);
    END LOOP;
    CLOSE radnici;
  END LOOP;
  CLOSE sefovi;
END;
```

Zadatak

Napisati PL/SQL blok koji će:

za zadati naziv projekta, za svakog radnika koji radi na tom projektu i ima broj časova rada veći od jedan povećati premiju za 10 posto. Ako radnik uopšte nema premiju dati mu premiju od 1000.

Rešenje

```
ACCEPT v_NazivProj CHAR PROMPT 'Unesite naziv projekta'
DECLARE
  CURSOR radnici (p_spr in number)
  IS SELECT r.mbr FROM radnik r, radproj rp WHERE r.mbr = rp.mbr
     AND rp.spr = p_spr AND rp.brc >1;

v_mbr Radnik.mbr%type;
v_spr Projekat.spr%type;
BEGIN
  SELECT spr INTO v_spr from projekat where nap = '&v_NazivProj';
  OPEN radnici(v_spr);
  LOOP
    FETCH radnici INTO v_mbr;
    EXIT WHEN radnici%NOTFOUND;
    UPDATE radnik
    SET pre = NVL(pre*1.1,1000)
    WHERE mbr = v_mbr;
  END LOOP;
  CLOSE radnici;
END;
```

Složeni tipovi podataka

- PL/SQL tip sloga
- PL/SQL tip kolekcije
 - INDEX BY tables – indeksirane tabele
 - nested tables – "ugnježdene" tabele
 - VARRAY – nizovi ograničene maksimalne dužine

PL/SQL tip sloga

- Deklarisanje

1. TYPE type_name IS RECORD

(field_declaration[, field_declaration]...);

<field_declaration>:

field_name {field_type | variable%TYPE
| table.column%TYPE | table%ROWTYPE}
[[NOT NULL] {:= | DEFAULT} expr]

2. identifier type_name;

PL/SQL tip sloga

- Referenciranje polja sloga

identifier.field_name

%ROWTYPE atribut

- Deklariše promenljivu prema kolekciji kolona u tabeli ili pogledu baze podataka
- Ispred %ROWTYPE može da stoji ime tabele ili pogleda
- Polja u slogu imaju isti naziv i tip podatka kao i kolone u tabeli ili pogledu

Sintaksa

```
identifier table%ROWTYPE;
```

Primeri upotrebe promenljivih tipa sloga

```
DECLARE
  TYPE T_ProjSlog IS RECORD(
    Spr Projekat.Spr%TYPE := 10,
    Nap Projekat.Nap%TYPE);
  V_Proj T_ProjSlog;
BEGIN
  SELECT Spr, Nap
  INTO   V_Proj
  FROM   Projekat
  WHERE  Spr = V_Proj.Spr;
  DBMS_OUTPUT.PUT_LINE('Naziv projekta je: ' ||
  V_Proj.Nap );
END;
```

Primeri upotrebe promenljivih tipa sloga

```
DECLARE
```

```
    V_Proj Projekat%ROWTYPE;
```

```
BEGIN
```

```
    SELECT *
```

```
    INTO  V_Proj
```

```
    FROM  Projekat
```

```
    WHERE Spr = 10;
```

```
END;
```

PL/SQL tip indeksirane tabele

- Deklarisanje

```
TYPE type_name IS TABLE OF  
    {column_type | variable%TYPE  
    | table.column%TYPE} [NOT NULL]  
    | table%ROWTYPE  
    [INDEX BY BINARY_INTEGER];
```

```
identifier type_name;
```

PL/SQL tip indeksirane tabele

- Referenciranje elementa tabele (niza)

identifier(index)

indetifier(ind1)...(indn) - za višedimenzionalne strukture

- Referenciranje polja sloga, koji predstavlja element tabele (niza)

identifier(index).field_name

Metode (operacije) nad promenljivama tabelarnog tipa

| | |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------|
| COUNT | Ukupan broj elemenata kolekcije |
| EXISTS(n) | Indikacija postojanja n-tog elementa kolekcije |
| EXTEND(n) | Alokacija prostora za novih n članova tabele – obavezno kada se ne koristi INDEX BY deklaracija indeksa tabele. |
| FIRST | Indeks prvog elementa kolekcije |
| LAST | Indeks poslednjeg elementa kolekcije |
| PRIOR(n) | Indeks prethodnog elementa kolekcije, u odnosu na n |
| NEXT(n) | Indeks narednog elementa kolekcije, u odnosu na n |
| DELETE[(n [, m])] | Brisanje svih, ili samo n-tog, ili intervala od n-tog do m-tog elementa iz kolekcije. Oslobađa se memorijski prostor. |
| TRIM[(n)] | Brisanje poslednjeg, ili n poslednjih elemenata iz kolekcije ("odsecanje" kolekcije) i oslobađanje memorijskog prostora. |

Metode (operacije) nad promenljivama tabelarnog tipa

- Referenciranje metode

`identifier.method_name[(parameters)]`

Primeri upotrebe promenljivih tipa tabele

```
DECLARE
  TYPE T_Tab IS TABLE OF VARCHAR2(20) INDEX BY
    BINARY_INTEGER;
  Tab T_Tab;
  i BINARY_INTEGER;
BEGIN
  Tab(1) := 'DEJAN';
  Tab(3) := 'NENAD';
  Tab(-1) := 'MARKO';
  Tab(5) := 'ACA';
  Tab.DELETE(3);
  i := Tab.FIRST;
  WHILE i IS NOT NULL LOOP
    DBMS_OUTPUT.PUT_LINE(i || ' ' || Tab(i));
    i := Tab.NEXT(i);
  END LOOP;
  DBMS_OUTPUT.PUT_LINE(NVL(TO_CHAR(i), 'i ima NULL vrednost.));
END;
```


Primeri upotrebe promenljivih tipa tabele

```
DECLARE
  TYPE T_Slog IS RECORD(
    Naziv VARCHAR2(50),
    BrojStudenata NUMBER := 0);
  TYPE T_Tab IS TABLE OF T_Slog INDEX BY BINARY_INTEGER;
  Tabela T_Tab;
  i BINARY_INTEGER;
BEGIN
  Tabela(1).Naziv := 'Napredno serversko programiranje';
  Tabela(1).BrojStudenata := 12;
  Tabela(2).Naziv := 'Informacioni sistemi';
  Tabela(2).BrojStudenata := 8;
  i := Tabela.FIRST;
  WHILE i <= Tabela.LAST LOOP
    DBMS_OUTPUT.PUT('Broj studenata koji slusa predmet ');
    DBMS_OUTPUT.PUT('"' || Tabela(i).Naziv || '" je ');
    DBMS_OUTPUT.PUT_LINE(Tabela(i).BrojStudenata);
    i := Tabela.NEXT(i);
  END LOOP;
END;
```

Primeri upotrebe promenljivih tipa tabele

```
DECLARE
  TYPE T_Tab IS TABLE OF VARCHAR2(20);
  Tab1 T_Tab := T_Tab();
  Tab2 T_Tab := T_Tab('Janko', 'Jana');
  i BINARY_INTEGER;
BEGIN
  Tab1.EXTEND(5);
  Tab1(1) := 'Ana';
  Tab1(3) := 'Bora';
  -- Tab(-1) := 'C'; STOP! Indeks moze ici samo od 1!
  Tab1(5) := 'Deni';
  i := Tab1.FIRST;
  WHILE i <= Tab1.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(i || '. ' || Tab1(i));
    i := Tab1.NEXT(i);
  END LOOP;
  i:= Tab2.FIRST;
  WHILE i <= Tab2.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(i || '. ' || Tab2(i));
    i := Tab2.NEXT(i);
  END LOOP;
END;
```

Primeri upotrebe promenljivih tipa tabelle

```
DECLARE
  TYPE T_Tab1 IS TABLE OF NUMBER;
  TYPE T_Mat1 IS TABLE OF T_Tab1;
  Mat1 T_Mat1 := T_Mat1(T_Tab1(1, 2), T_Tab1(3, 4), T_Tab1(5, 6));
  Tab1 T_Tab1 := T_Tab1();
BEGIN
  Tab1.EXTEND(2);
  Tab1 := Mat1(2);
  DBMS_OUTPUT.PUT_LINE(Mat1(3)(2));
  DBMS_OUTPUT.PUT_LINE(Tab1(1));
  DBMS_OUTPUT.PUT_LINE(Tab1(2));
  Mat1.EXTEND(2);
  Mat1(4):=T_Tab1(7,8);
  Tab1.EXTEND(1);
  Tab1(3) := 10;
  DBMS_OUTPUT.PUT_LINE(Tab1(3));
  Mat1(3)(2) := 2;
  DBMS_OUTPUT.PUT_LINE(Mat1(3)(2));
END;
```

Primeri upotrebe promenljivih tipa tabelle

```
DECLARE
```

```
    TYPE T_Tab1 IS TABLE OF NUMBER INDEX BY  
    BINARY_INTEGER;
```

```
    TYPE T_Mat1 IS TABLE OF T_Tab1 INDEX BY  
    BINARY_INTEGER;
```

```
    Mat1 T_Mat1;
```

```
    Tab1 T_Tab1;
```

```
BEGIN
```

```
    Mat1(1)(1) := 1;
```

```
    Mat1(1)(2) := 2;
```

```
    Tab1 := Mat1(1);
```

```
    DBMS_OUTPUT.PUT_LINE(Mat1(1)(1));
```

```
END;
```