Sistemi baze podataka

PL/SQL - Kursori i kompleksni tipovi

Sadržaj

- Kursori - osnovi
- Kompleksni tipovi
- Kursori - kursorska for petlja

Kursori - osnovi

Kursori u jezku PL/SQL

- Implicitni (SQL)
- Eksplicitni
  - Deklariše se programski
  - Njime se upravlja programski

Kursori u jezku PL/SQL

- Deklarisanje kursora
  ```
  CURSOR naziv_kursora [(lista_formalnih_parametara)] IS SELECT ...
  ```
- Otvaranje kursora
  ```
  OPEN naziv_kursora [(lista_stvarnih_parametara)];
  ```
- Preuzimanje torke kursora
  ```
  FETCH naziv_kursora INTO [var1, var2,... | record_var];
  ```
- Zatvaranje kursora
  ```
  CLOSE naziv_kursora;
  ```

Funkcije ispitivanja statusa kursora

- `naziv_kursora%FOUND`
  - TRUE, ako je bar jedan red bio predmet poslednje fetch operacije, inače FALSE
- `naziv_kursora%NOTFOUND`
  - TRUE, ako ni jedan red nije bio predmet poslednje fetch operacije, inače FALSE
- `naziv_kursora%ROWCOUNT`
  - broj redova, koji su bili predmet poslednje fetch operacije
- `naziv_kursora%ISOPEN`
  - TRUE, ako je kursor otvoren, a inače FALSE

Primer eksplicitno deklarisanog kursora

```
DECLARE
        Ukup_plt NUMBER;
    L_Mbr radnik.Mbr%TYPE;
    L_Plt radnik.Plt%TYPE;

    CURSOR spisak_rad IS                    -- eksplicitno deklarisani kursor
    SELECT Mbr, Plt
    FROM radnik
    WHERE Mbr BETWEEN 01 AND 99;
BEGIN
    Ukup_Plt := 0;
    OPEN spisak_rad;                        -- otvoren kursor, izvršava se SELECT

    LOOP
         FETCH spisak_rad INTO L_Mbr, L_Plt;    -- dobavljanje naredne torke iz kursora
         EXIT WHEN spisak_rad%NOTFOUND;    -- uslov izlaska iz petlje
         Ukup_Plt := Ukup_Plt + L_Plt;
    END LOOP;

    CLOSE spisak_rad;               -- zatvoren kursor
        DBMS_OUTPUT.PUT_LINE('Plata je: ' || Ukup_Plt);
END;
```

Primer eksplicitno deklarisanog kursora s parametrima i funkcijom %ROWCOUNT

```
DECLARE
    Ukup_plt NUMBER;
    L_tek_red Radnik%ROWTYPE;

    -- kursor, deklarisan s parametrima
    CURSOR spisak_rad (D_gran Radnik.Mbr%TYPE, G_gran Radnik.Mbr%TYPE) IS
    SELECT *
    FROM radnik
    WHERE Mbr BETWEEN D_gran AND G_gran;

BEGIN
    Ukup_Plt := 0;
    OPEN spisak_rad (01, 99);               -- otvoren kursor, izvršava se SELECT

    LOOP
            FETCH spisak_rad INTO L_tek_red;
            EXIT WHEN (spisak_rad%NOTFOUND) OR (spisak_rad%ROWCOUNT > 5);
            Ukup_Plt := Ukup_Plt + L_tek_red.Plt;
    END LOOP;

    CLOSE spisak_rad;                       -- zatvoren kursor
    DBMS_OUTPUT.PUT_LINE('Plata je: ' || Ukup_Plt);
END;
```

Zadatak

- Napisati PL/SQL blok koji će:
  - Ispisati sve radnike koji rade u sektorima kojima rukovode Pera Perić, Savo Oroz i Đoka Đokić. Ukoliko neko od njih nije šef obavestiti korisnika o tome.
  - Ispis rezultata treba da izgleda ovako:

    Radnici kojima je sef ...
    Ime zaposlenog je ...
    Ime zaposlenog je ...

    Radnici kojima je sef ...
    Ime zaposlenog je ...

Rešenje

```
DECLARE
   CURSOR sefovi IS SELECT * FROM radnik WHERE ime || ' ' || prz IN ('Pera Peric', 'Savo Oroz', 'Djoka Djokic');

   CURSOR radnik_sef (p_sef in number)
   IS SELECT * FROM radnik WHERE Sef = p_sef;

   v_sef Radnik%ROWTYPE;
   v_radnik Radnik%ROWTYPE;
   v_broj NUMBER;
BEGIN
   OPEN  sefovi;
   LOOP
      FETCH sefovi INTO v_sef;
      EXIT WHEN sefovi%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Ime sefa ' || ' - ' || v_sef.ime || ' ' || v_sef.prz);

--  Nastavak na sledećem slajdu
```

Rešenje

```
      SELECT COUNT(*) INTO v_broj FROM radnik WHERE sef=v_sef.mbr;
      IF v_broj = 0 THEN
         DBMS_OUTPUT.PUT_LINE('Nije sef!');
      ELSE
         DBMS_OUTPUT.PUT_LINE('Radnici kojima je sef:');
         OPEN  radnik_sef(v_sef.mbr);
         LOOP
            FETCH radnik_sef INTO v_radnik;
            EXIT WHEN radnik_sef%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('  Ime zaposlenog: ' || v_radnik.ime || ' ' || v_radnik.prz);
                END LOOP;
                CLOSE radnik_sef;
      END IF;
      DBMS_OUTPUT.PUT_LINE('');
      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 5 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 >5;

   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

Složeni tipovi podataka

- PL/SQL tip sloga
  - RECORD
- 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`

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;
```

%ROWTYPE atribut

- Deklariše promenljivu prema kolekciji kolona u tabeli ili pogledu baze podataka
- Ispred %ROWTYPE može da stoji ime tabele, pogleda ili kursora
- 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
   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)
identifier(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(1);
   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';  -- NAPOMENA: Tab1(-1) := 'Cane'; NIJE MOGUĆE! Indeks ugneždenih tabela može ići samo od 1!
   Tab1(5) := 'Darko';
   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;
```

Kursori - kursorska for petlja

Kursorska FOR petlja

```
FOR record_var IN naziv_kursora [(lista_stvarnih_parametara)] LOOP
  statement1;
  statement2;
  . . .
END LOOP;
```

Kursorska FOR petlja

- Obavezna deklaracija kursorskog područja
- **Automatsko otvaranje, preuzimanje torki i zatvaranje kursora**
- Slogovsku promenljivu record_var nije potrebno eksplicitno deklarisati

Primer eksplicitno deklarisanog kursora s parametrima i upotrebe kursorske FOR petlje.

```
DECLARE
        Ukup_Plt NUMBER;

    CURSOR spisak_rad (D_gran radnik.Mbr%TYPE,
          G_gran radnik.Mbr%TYPE)
    IS
    SELECT *
    FROM radnik
    WHERE Mbr BETWEEN D_gran AND G_gran;

BEGIN
    Ukup_Plt := 0;
    FOR p_tek_red IN spisak_rad (01, 99) LOOP
          Ukup_Plt := Ukup_Plt + p_tek_red.Plt;
    END LOOP;
        DBMS_OUTPUT.PUT_LINE('Plata je: ' || Ukup_Plt);
END;
```

Kursorska FOR petlja sa implicitnom deklaracijom kursora

```
FOR record_var IN (SELECT ...) LOOP
  statement1;
  statement2;
  . . .
END LOOP;
```

Kursorska FOR petlja sa implicitnom deklaracijom kursora

- Kursorsko područje se ne deklariše eksplicitno
- Automatsko otvaranje, preuzimanje torki i zatvaranje kursora
- Slogovsku promenljivu record_var nije potrebno eksplicitno deklarisati

Primer upotrebe kursorske FOR petlje, s implicitno deklarisanim kursorom

```
DECLARE
        Ukup_Plt NUMBER;
BEGIN
    Ukup_Plt := 0;
    FOR p_tek_red IN (SELECT * FROM radnik
        WHERE Mbr BETWEEN 01 AND 99) LOOP
                -- otvoren kursor, izvršava se SELECT
         Ukup_Plt := Ukup_Plt + p_tek_red.Plt;
    END LOOP;    -- zatvoren kursor
        DBMS_OUTPUT.PUT_LINE('Plata je: ' || Ukup_Plt);
END;
```

Zadatak

- Napisati PL/SQL blok koji će preuzeti sve torke iz tabele Projekat i prebaciti ih u PL/SQL tabelarnu kolekciju. Zatim će, redom, odštampati sve elemente tako dobijene tabelarne kolekcije.

Rešenje

```
DECLARE
TYPE T_Projekat IS TABLE OF Projekat%ROWTYPE
   INDEX BY BINARY_INTEGER;
Tabela T_Projekat;
i BINARY_INTEGER:=0;
BEGIN
FOR rec IN (SELECT * FROM Projekat) LOOP
  Tabela(i):=rec;
  i:=i+1;
END LOOP;
i:= Tabela.FIRST;
WHILE i<=Tabela.LAST LOOP
  DBMS_OUTPUT.PUT_LINE('Naziv projekta: ' || Tabela(i).nap);
  DBMS_OUTPUT.PUT_LINE('Sifra rukovodioca: ' || Tabela(i).ruk);
  DBMS_OUTPUT.PUT_LINE('Narucilac projekta: ' || Tabela(i).nar);
  i:=Tabela.NEXT(i);
END LOOP;
END;
```

Zadatak

- Napisati PL/SQL blok koji će:
  - preuzeti sve torke iz tabele Projekat, uređene u opadajućem redosledu šifri projekata, i prebaciti ih u PL/SQL tabelarnu kolekciju. Uz svaku preuzetu torku iz tabele Projekat, treba inicijalizovati novu kolekciju koja će sadržati skup svih matičnih brojeva radnika, koji su angažovani na datom projektu.
  - Zatim treba, redom, odštampati sve torke iz kolekcije projekata, a uz svaku torku iz kolekcije projekata treba prikazati matične brojeve svih radnika koji su angažovani na tom projektu.

Rešenje 1

```
DECLARE
   TYPE T_Projekat IS TABLE OF
      Projekat%ROWTYPE INDEX BY BINARY_INTEGER;
   TYPE T_SifRad IS TABLE OF
      Radnik.mbr%TYPE INDEX BY BINARY_INTEGER;
   TYPE T_Radnici IS TABLE OF
      T_SifRad INDEX BY BINARY_INTEGER;
   TabelaP T_Projekat;
   TabelaR T_Radnici;
   i BINARY_INTEGER:=0;
   j BINARY_INTEGER:=1;
BEGIN
   FOR rec IN (SELECT * FROM Projekat ORDER BY spr DESC) LOOP
      TabelaP(i):=rec;
      FOR rec1 IN (SELECT mbr FROM RadProj WHERE spr = rec.spr) LOOP
         TabelaR(i)(j):=rec1.mbr;
         j:=j+1;
      END LOOP;
   i:=i+1;
   END LOOP;
```

Rešenje 1

```
    i:= TabelaP.FIRST;
    WHILE i<=TabelaP.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Naziv projekta: ' || TabelaP(i).nap);
        DBMS_OUTPUT.PUT_LINE('Sifra rukovodioca: ' || TabelaP(i).ruk);
        DBMS_OUTPUT.PUT_LINE('Narucilac projekta: ' || TabelaP(i).nar);
        j:= TabelaR(i).FIRST;
        WHILE j<=TabelaR(i).LAST LOOP
            DBMS_OUTPUT.PUT_LINE('Maticni broj radnika: ' || TabelaR(i)(j));
            j:= TabelaR(i).NEXT(j);
        END LOOP;
    i:=TabelaP.NEXT(i);
    END LOOP;
END;
```

Rešenje 2

```
DECLARE
   TYPE T_SifRad IS TABLE OF
      Radnik.mbr%TYPE INDEX BY BINARY_INTEGER;
   TYPE T_Projekat IS RECORD (
      ProjPodaci Projekat%ROWTYPE,
      Radnici T_SifRad);
   TYPE T_Projekti IS TABLE OF
      T_Projekat INDEX BY BINARY_INTEGER;

   TabelaP T_Projekti;
   i BINARY_INTEGER:=0;
   j BINARY_INTEGER:=1;

BEGIN
   FOR rec IN (SELECT * FROM Projekat ORDER BY spr DESC) LOOP
      TabelaP(i).ProjPodaci:=rec;
      FOR rec1 IN (SELECT mbr FROM RadProj WHERE spr = rec.spr) LOOP
         TabelaP(i).Radnici(j):=rec1.mbr;
         j:=j+1;
      END LOOP;
   i:=i+1;
   END LOOP;
```

Rešenje 2

```
    i:= TabelaP.FIRST;
    WHILE i<=TabelaP.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Naziv projekta: ' || TabelaP(i).ProjPodaci.nap);
        DBMS_OUTPUT.PUT_LINE('Sifra rukovodioca: ' || TabelaP(i).ProjPodaci.ruk);
        DBMS_OUTPUT.PUT_LINE('Narucilac projekta: ' || TabelaP(i).ProjPodaci.nar);
        DBMS_OUTPUT.PUT_LINE('');
        IF TabelaP(i).Radnici.COUNT != 0 THEN
            DBMS_OUTPUT.PUT_LINE('Radnici: ');
            DBMS_OUTPUT.PUT_LINE('');
            j:= TabelaP(i).Radnici.FIRST;
            WHILE j<=TabelaP(i).Radnici.LAST LOOP
                DBMS_OUTPUT.PUT_LINE('Maticni broj radnika: ' || TabelaP(i).Radnici(j));
                j:= TabelaP(i).Radnici.NEXT(j);
            END LOOP;
        ELSE
            DBMS_OUTPUT.PUT_LINE('Nijedan radnik ne radi na projektu "' || TabelaP(i).ProjPodaci.nap || '".');
        END IF;
        i:=TabelaP.NEXT(i);
        DBMS_OUTPUT.PUT_LINE('');
    END LOOP;
END;
```

Zadatak

- Napisati PL/SQL blok koji će:
  - Prikazati mbr, ime i prezime za radnike čija je vrednost matičnog broja između 1 i 100.
  - Za svakog radnika prikazati kumulativnu sumu isplata vršenih u toku 2023. Godine.

Primer ispisa:

```
Radnik: 10 Pera Peric
    Datum isplate - Razlog isplate - Iznos - Kumulativni zbir
    01-JAN-23 - PLATA_DEO1 - 5000 - 5000
    15-JAN-23 - PLATA_DEO2 - 5000 - 10000
    01-FEB-23 - PLATA_DEO1 - 5000 - 15000
                  ...
Radnik: 20 Milan Milic
        ...
```

Rešenje

```
DECLARE
   CURSOR kum_sume_plata(p_mbr IN NUMBER) IS
      SELECT mbr, datum_isplate, razlog_isplate, iznos,
         SUM(iznos) OVER (ORDER BY datum_isplate) AS kumulativni_zbir
      FROM isplate_radnicima
      WHERE mbr = p_mbr AND godina = 2023
      ORDER BY datum_isplate;
BEGIN
   FOR c_radnik IN (SELECT * FROM Radnik WHERE mbr between 01 and 100) LOOP
   DBMS_OUTPUT.PUT_LINE('Radnik: ' || c_radnik.mbr || ' ' || c_radnik.ime || ' ' || c_radnik.prz);
   DBMS_OUTPUT.PUT_LINE('Datum isplate' || ' - ' || 'Razlog isplate' || ' - ' || 'Iznos' || ' - ' || 'Kumulativni zbir');
      FOR c_kum_suma_plate IN kum_sume_plata(c_radnik.mbr) LOOP
         DBMS_OUTPUT.PUT_LINE(c_kum_suma_plate.datum_isplate || ' - ' || c_kum_suma_plate.razlog_isplate || ' - ' ||
c_kum_suma_plate.iznos || ' - ' || c_kum_suma_plate.kumulativni_zbir);
      END LOOP;
      DBMS_OUTPUT.PUT_LINE(' ');
   END LOOP;
END;
```

Kraj!

Hvala na pažnji!