

Sistemi baza podataka

Dr Ivan Luković

Dr Slavica Kordić

Vladimir Dimitrieski

Nikola Todorović

Milica Todorović



PL/SQL – KURSORI

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 deklarisan 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:

- Za svakog šefa ispišite njegove radnike

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:

- 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 na sledeći način:

Radnici kojima je sef ...

Ime zaposlenog je ...

Ime zaposlenog je ...

Radnici kojima je sef ...

Ime zaposlenog je ...

Rešenje

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

  v_radnik Radnik%rowtype;
  v_mbr Radnik.mbr%type := 0;
  v_broj NUMBER;
BEGIN
  SELECT mbr INTO v_mbr FROM radnik WHERE ime='Pera' and prz='Peric';
  SELECT COUNT(*) INTO v_broj FROM radnik WHERE sef=v_mbr;
  IF v_broj = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Pera Peric nije sef');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Radnici kojima je sef Pera Peric');
    DBMS_OUTPUT.PUT_LINE("");
    OPEN radnik_sef(v_mbr);
    LOOP
      FETCH radnik_sef INTO v_radnik;
      EXIT WHEN radnik_sef%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Ime zaposlenog ' || ' ' je ' || v_radnik.ime || ' ' || v_radnik.prz);
    END LOOP;
    CLOSE radnik_sef;
  END IF;
  DBMS_OUTPUT.PUT_LINE("");

  -- nastavak na sledecoj strani
```

Rešenje

```
SELECT mbr INTO v_mbr FROM radnik WHERE ime='Savo' and prz='Oroz';
SELECT COUNT(*) INTO v_broj FROM radnik WHERE sef=v_mbr;
IF v_broj = 0 THEN
  DBMS_OUTPUT.PUT_LINE('Savo Oroz nije sef');
ELSE
  DBMS_OUTPUT.PUT_LINE('Radnici kojima je sef Savo Oroz');
  DBMS_OUTPUT.PUT_LINE("");
  OPEN radnik_sef(v_mbr);
  LOOP
    FETCH radnik_sef INTO v_radnik;
    EXIT WHEN radnik_sef%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Ime zaposlenog ' || ' je ' || v_radnik.ime || ' ' || v_radnik.prz);
  END LOOP;
  CLOSE radnik_sef;
END IF;
DBMS_OUTPUT.PUT_LINE("");
SELECT mbr INTO v_mbr FROM radnik WHERE ime='Djoka' and prz='Djokic';
SELECT COUNT(*) INTO v_broj FROM radnik WHERE sef=v_mbr;
IF v_broj = 0 THEN
  DBMS_OUTPUT.PUT_LINE('Djoka Djokic nije sef');
ELSE
  DBMS_OUTPUT.PUT_LINE('Radnici kojima je sef Djoka Djokic');
  DBMS_OUTPUT.PUT_LINE("");
  OPEN radnik_sef(v_mbr);
  LOOP
    FETCH radnik_sef INTO v_radnik;
    EXIT WHEN radnik_sef%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Ime zaposlenog ' || ' je ' || v_radnik.ime || ' ' || v_radnik.prz);
  END LOOP;
  CLOSE radnik_sef;
END IF;
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;
```

PL/SQL – 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';
    -- Tab(-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;
```

Primeri upotrebe promenljivih tipa tabele

```
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 tabele

```
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;
```

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 deklarirati

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 deklarirati

Primer upotrebe kursorske FOR petlje, s implicitno deklaranim 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;
```

Zadaci

- 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 prvog zadatka

```
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;
```

Zadaci

- 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 drugog zadatka

```
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;
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 drugog zadatka

```
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;
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 koliko radnika nema ni najmanju ni najveću platu.

Rešenje

```
declare
    platamin radnik.plt%type;
    platamax radnik.plt%type;
    ukupno NUMBER;

begin
    ukupno:=0;
    select min(plt) into platamin from radnik;
    select max(plt) into platamax from radnik;
    DBMS_OUTPUT.PUT_LINE('Min plata: ' || platamin);
    DBMS_OUTPUT.PUT_LINE('Max plata: ' || platamax);
    for rec in (select * from radnik) loop
        if (rec.plt > platamin and rec.plt < platamax) then
            ukupno:=ukupno+1;
        end if;
    end loop;
    DBMS_OUTPUT.PUT_LINE('Broj radnika: ' || ukupno);
end;
```