

Zadatak

- Sa tastature uneti broj projekta. Za uneseni broj preuzeti njegove podatke u posebne promenjive i prikazati ih u konzoli.

Zadatak

```
DECLARE
```

```
  V_Spr Projekat.Spr%TYPE := 10;
```

```
  V_Nap Projekat.Nap%TYPE;
```

```
  V_Nar Projekat.Nar%TYPE;
```

```
BEGIN
```

```
  SELECT Spr, Nap, Nar
```

```
  INTO  V_Spr, V_Nap, V_Nar
```

```
  FROM  Projekat
```

```
  WHERE Spr = &V_Spr;
```

```
  DBMS_OUTPUT.PUT_LINE(v_Spr);
```

```
  DBMS_OUTPUT.PUT_LINE(v_Nap);
```

```
  DBMS_OUTPUT.PUT_LINE(v_Nar);
```

```
END;
```

Implicitni SQL kursor

- Sve SQL naredbe se parsiraju i izvršavaju u okviru kursorских područja
- DML naredbama, koje se izvršavaju u PL/SQL bloku, dodeljuju se kursorска područja (kursori), čiji je programski naziv SQL
 - Implicitni SQL kursor
- Moguće je ispitivanje statusa implicitnog SQL kursora, nakon svake izvršene DML naredbe

Implicitni SQL kursor

- Funkcije ispitivanja statusa implicitnog SQL kursora
 - **SQL%FOUND**
 - TRUE, ako je bar jedan red bio predmet poslednje DML operacije, inače FALSE
 - **SQL%NOTFOUND**
 - TRUE, ako ni jedan red nije bio predmet poslednje DML operacije, inače FALSE
 - **SQL%ROWCOUNT**
 - broj redova, koji su bili predmet poslednje DML operacije
 - **SQL%ISOPEN**
 - uvek ima vrednost FALSE.
 - Upravljanje (otvaranje i zatvaranje) implicitnim kursorima je uvek automatsko. Neposredno nakon svake DML operacije, SQL kursorско područje se automatski zatvori.

Primer

```
BEGIN
```

```
  UPDATE Projekat
```

```
  SET Nap = "
```

```
  WHERE 1=2;
```

```
  DBMS_OUTPUT.PUT_LINE('Jedan update sa WHERE  
  USLOVOM 1=2');
```

```
  DBMS_OUTPUT.PUT_LINE(sql%rowcount || ' zapisa');
```

```
END;
```

Zadatak

- U konzolu ispisati radnika (mbr, ime, prezime i platu) čiji je matični broj jednak matičnom broju unetom preko prompta.
- Takođe prikazati jednog radnika čije ime počinje na slovo uneto sa prompta.

Rešenje

```
DECLARE
```

```
    V_MBR RADNIK.MBR%TYPE;
```

```
    V_IME RADNIK.IME%TYPE;
```

```
    V_PRZ RADNIK.PRZ%TYPE;
```

```
    V_PLT RADNIK.PLT%TYPE;
```

```
BEGIN
```

```
    SELECT Mbr, Ime, Prz, Plt
```

```
    INTO  V_MBR, V_IME, V_PRZ, V_PLT
```

```
    FROM  Radnik
```

```
    WHERE mbr = &PR_MBR;
```

```
    DBMS_OUTPUT.PUT_LINE(V_MBR || ' ' || V_IME || ' ' || V_PRZ || ' ' || V_PLT);
```

```
    SELECT Mbr, Ime, Prz, Plt
```

```
    INTO  V_MBR, V_IME, V_PRZ, V_PLT
```

```
    FROM  Radnik
```

```
    WHERE ime like '&PR_IME%' and rownum = 1;
```

```
    DBMS_OUTPUT.PUT_LINE(V_MBR || ' ' || V_IME || ' ' || V_PRZ || ' ' || V_PLT);
```

```
END;
```

DML naredbe

- Normalna upotreba naredbi INSERT, UPDATE i DELETE
- U okviru BEGIN – END sekcije koriste se standardne DML naredbe
 - moguće je zadavati vrednosti preko promenjivih

Naredbe za upravljanje tokom izvođenja programa

- Naredba selekcije
- Naredbe iteracije

Naredba selekcije

```
IF logički_izraz THEN
    blok_izvršnih_naredbi;
[ELSIF logički_izraz THEN
    blok_izvršnih_naredbi;
]...
[
ELSE
    blok_izvršnih_naredbi;
]
END IF;
```

Zadatak

- Preko tastature uneti podatke o radniku i upisati ga u bazu podataka. Za matični broj uneti narednu vrednost iz sekvencera.
- Ukoliko je uspešno uneta nova torka, na konzoli ispisati poruku o uspešnom unosu torke. U protivnom, ispisati poruku o neuspešnom unosu.

Zadatak

```
ACCEPT D_Prz PROMPT 'Unesite prezime: '  
ACCEPT D_Ime PROMPT 'Unesite ime: '
```

```
BEGIN
```

```
    INSERT INTO Radnik (Mbr, Prz, Ime, God)  
    VALUES (SEQ_Mbr.NEXTVAL, '&D_Prz', '&D_Ime',  
    SYSDATE);
```

```
    IF SQL%FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Dodata nova toraka u  
tabelu Radnik.');
```

```
    ELSE
```

```
        DBMS_OUTPUT.PUT_LINE('Unos torke u tabelu  
Radnik nije uspeo.');
```

```
    END IF;
```

```
END;
```

Zadatak

- Obrisati radnika čiji matični broj je unet preko tastature. Na konzoli prikazati koliko je radnika obrisano.

Zadatak

```
DECLARE
```

```
  v_Mbr radnik.mbr%TYPE := 203;
```

```
  broj_del NUMBER;
```

```
BEGIN
```

```
  DELETE FROM radnik
```

```
  WHERE mbr = v_Mbr;
```

```
  broj_del := SQL%ROWCOUNT;
```

```
  DBMS_OUTPUT.PUT_LINE('Obrisano je: '|| broj_del || '  
  radnika');
```

```
END;
```

Naredbe iteracije

- Bezuslovna (beskonačna) iteracija / LOOP

```
LOOP
```

```
    blok_izvršnih_naredbi;
```

```
END LOOP;
```

- Uslovna iteracija, s testom uslova na početku / WHILE LOOP

```
WHILE logički_izraz LOOP
```

```
    blok_izvršnih_naredbi;
```

```
END LOOP;
```

Izlazak iz petlje / EXIT

- EXIT [labela] [WHEN logički_izraz]
- EXIT se, najčešće, koristi u kombinaciji s безусловnom petljom LOOP ... END LOOP
 - Obezbeđenje formiranja uslovne petlje, s mogućnošću testa uslova petlje na bilo kojoj poziciji u petlji

<<labela>>

LOOP

...

EXIT [labela] [WHEN logički_izraz]

...

END LOOP;

Naredbe iteracije

- Brojačka iteracija / FOR LOOP

```
FOR brojač IN [REVERSE] donja_granica..gornja_granica LOOP  
    blok_izvršnih_naredbi;  
END LOOP;
```

NAPOMENA: Brojačku promenljivu *brojač* **nije potrebno deklarirati**. Korak brojača je uvek 1.

Primeri upotrebe konstrukcija za upravljanje tokom izvođenja programa

```
BEGIN
```

```
  FOR i IN REVERSE 1..3 LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Vrednost brojaca i je: '  
    || TO_CHAR(i));
```

```
  END LOOP;
```

```
END;
```

```
BEGIN
```

```
  FOR i IN 1..3 LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Vrednost brojaca i je: '  
    || TO_CHAR(i));
```

```
  END LOOP;
```

```
END;
```

Primeri upotrebe konstrukcija za upravljanje tokom izvođenja programa

```
DECLARE
```

```
    i NUMBER(1) := 1;
```

```
BEGIN
```

```
    WHILE i <= 3 LOOP
```

```
        DBMS_OUTPUT.PUT_LINE('Vrednost brojaca i je: '  
|| TO_CHAR(i));
```

```
        i := i + 1;
```

```
    END LOOP;
```

```
END;
```

Primeri upotrebe konstrukcija za upravljanje tokom izvođenja programa

```
DECLARE
```

```
  i NUMBER(1) := 1;
```

```
BEGIN
```

```
  LOOP
```

```
    EXIT WHEN i > 3;
```

```
    DBMS_OUTPUT.PUT_LINE('Vrednost brojaca i je: '  
|| TO_CHAR(i));
```

```
    i := i + 1;
```

```
  END LOOP;
```

```
END;
```

Primeri upotrebe konstrukcija za upravljanje tokom izvođenja programa

```
DECLARE
```

```
  i NUMBER(1) := 0;
```

```
BEGIN
```

```
  LOOP
```

```
    i := i + 1;
```

```
    DBMS_OUTPUT.PUT_LINE('Vrednost brojaca i je: '  
|| TO_CHAR(i));
```

```
    EXIT WHEN i >= 3;
```

```
  END LOOP;
```

```
END;
```

Zadatak

- Ispisati sve parne i neparne brojeve od 1 do broja unetog sa tastature.

Zadatak

```
ACCEPT N PROMPT 'N: '  
BEGIN  
  FOR i IN 1..&N LOOP  
    IF MOD(i, 2) = 0 THEN  
      DBMS_OUTPUT.PUT_LINE(i || ' je paran  
broj.');
```

broj.');

```
    ELSIF MOD(i, 2) = 1 THEN  
      DBMS_OUTPUT.PUT_LINE(i || ' je neparan  
broj.');
```

broj.');

```
    ELSE  
      DBMS_OUTPUT.PUT_LINE('Nemoguc  
slucaj.');
```

slucaj.');

```
    END IF;  
  END LOOP;  
END;
```

Zadatak

Napisati PL/SQL blok koji će:

- interaktivno prihvatiti vrednosti za Prz, Ime, Sef, Plt i God, (za MBR koristiti sekvencer)
- dodati novu toroku u tabelu Radnik, s prethodno preuzetim podacima i
- angažovati novododatog radnika na projektu sa Spr = 10 i 5 sati rada.

Rešenje

```
BEGIN
```

```
INSERT INTO radnik (Mbr, Prz, Ime, Plt, God)
```

```
VALUES (SEQ_Mbr.NEXTVAL, '&&Prz', '&&Ime', &&Plt,  
'&&God');
```

```
INSERT INTO radproj (Mbr, Spr, Brc)
```

```
VALUES (SEQ_Mbr.CURRVAL, 10, 5);
```

```
COMMIT;
```

```
END;
```

Zadatak

Napisati PL/SQL blok koji će:

- izbrisati angažovanje prethodno dodatog radnika na projektu sa šifrom 10 i obavestiti porukom korisnika da li je brisanje uspešno obavljeno,
- izbrisati prethodno dodatog radnika iz evidencije i obavestiti porukom korisnika da li je brisanje uspešno obavljeno,
- sačuvati vrednost za Mbr izbrisanog radnika u lokalnoj promenljivoj pod nazivom *Del_Mbr*

Rešenje

```
ACCEPT v_Mbr PROMPT 'Mbr = '  
  
DECLARE  
    Del_Mbr radnik.Mbr%TYPE;  
BEGIN  
    DELETE FROM radproj  
    WHERE Mbr = &v_Mbr AND Spr = 10;  
    IF SQL%FOUND THEN  
        DBMS_OUTPUT.PUT_LINE('Brisanje rada na projektu uspesno obavljeno.');    ELSE  
        DBMS_OUTPUT.PUT_LINE('Brisanje rada na projektu nije uspesno obavljeno.');    END IF;  
  
    DELETE FROM radnik  
    WHERE Mbr = &v_Mbr ;  
    IF SQL%FOUND THEN  
        DBMS_OUTPUT.PUT_LINE('Brisanje radnika uspesno obavljeno.');    ELSE  
        DBMS_OUTPUT.PUT_LINE('Brisanje radnika nije uspesno obavljeno.');    END IF;  
    Del_Mbr := &Mbr;  
END;
```

Zadatak

Kreirati tabelu Spisak_zarada, korišćenjem SQL komande:

```
CREATE TABLE Spisak_zarada (Mbr NUMBER(3),  
Plt NUMBER(10, 2), Evri VARCHAR2(10),  
CONSTRAINT Sz_PK PRIMARY KEY (Mbr))
```

Napisati PL/SQL blok koji će:

za svaku torku iz tabele Radnik, za koju je matični broj u intervalu od 10 do 100, izuzimajući radnika s matičnim brojem 90, preneti u tabelu Spisak_zarada matični broj, iznos plate, i inicijalizovati polje Evri sa vrednošću plate u evrima. Ukoliko radnik već postoji u tabeli izvršiti izmenu vrednosti obeležja Plt i Evri. Kurs evra treba da zadaje korisnik iz okruženja.

Rešenje

```
ACCEPT E PROMPT 'Kurs evra je: '  
DECLARE  
    v_Plt Spisak_zarada.Plt%TYPE;  
    broj NUMBER :=0;  
BEGIN  
    FOR i IN 1..10 LOOP  
        IF i != 9 THEN  
            SELECT Plt INTO v_Plt FROM Radnik  
            WHERE Mbr = 10*i;  
            SELECT COUNT(*) INTO broj FROM Spisak_zarada  
            WHERE Mbr = 10*i;  
            IF broj = 0 THEN  
                INSERT INTO Spisak_zarada (Mbr, Plt, Evri)  
                VALUES (10*i, v_Plt, v_Plt/ &E );  
            ELSE  
                UPDATE Spisak_zarada  
                SET Plt = v_Plt,  
                Evri = v_Plt* &E  
                WHERE Mbr = 10*i;  
            END IF;  
        END IF;  
    END LOOP;  
END;
```

Zadatak

Napisati PL/SQL blok koji će:

Proveravati ima li radnika sa platom manjom od zadate. Ako ima povećati premiju za 20% svakom radniku koji ima takvu platu. Ukoliko radnik nema uopste premiju dodeliti mu premiju od 5000. Ako svi radnici imaju platu veću od zadate ispisati poruku o tome.

Rešenje

```
ACCEPT plata PROMPT 'Plata = '  
DECLARE  
    broj_rad NUMBER;  
BEGIN  
    SELECT COUNT(*) INTO broj_rad FROM Radnik  
    WHERE Plt < &plata;  
    IF broj_rad = 0 THEN  
        DBMS_OUTPUT.PUT_LINE('Svi imaju platu vecu od ' ||  
        TO_CHAR(&plata));  
    ELSE  
        UPDATE Radnik  
        SET Pre = NVL(Pre*1.2,5000)  
        WHERE Mbr IN (SELECT Mbr FROM Radnik  
        WHERE Plt < &plata);  
    END IF;  
END;
```