

```
-- OBP PROJEKAT- KLADIONICA - DEDIC MIRELA --
```

```
SELECT * FROM odjel;  
SELECT * FROM funkcija;  
SELECT * FROM Poslovnica;  
SELECT * FROM Tip_Poslovnice;  
SELECT * FROM Lokacija;  
SELECT * FROM Drzava;  
SELECT * FROM Regija;  
SELECT * FROM Uposlenik;
```

```
SELECT * FROM Korisnik;  
SELECT * FROM Racun;  
SELECT * FROM Korisnik_Racun;  
SELECT * FROM Opklade;  
SELECT * FROM Pogodnosti;  
SELECT * FROM Tip_Opklade;  
SELECT * FROM Subjekt;  
SELECT * FROM Par;  
SELECT * FROM Listic;
```

```
-- UPITI --
```

```
--1.
```

```
SELECT p.naziv AS "Poslovnica",  
       l.adresa AS "adresa",  
       l.postanski_kod AS "pk",  
       l.grad AS "grad",  
       d.ime AS "drzava",  
       r.ime AS "regija"  
FROM Poslovnica p, Lokacija l, Drzava d, Regija r  
WHERE p.lokacija_id = l.lokacija_id AND l.drzava_id = d.drzava_id AND  
d.regija_id = r.regija_id;
```

```
--2.
```

```
SELECT *  
FROM Uposlenik  
WHERE plata BETWEEN 1000 AND 5000;
```

```
--3.
```

```
SELECT Next_Day(datum_zaposlenja, 'SATURDAY')  
FROM Uposlenik;
```

```

--4.
SELECT ime || ' ' || prezime AS "ime_prezime"
FROM Korisnik
WHERE ime LIKE '%A%' OR ime LIKE '%a%'
ORDER BY prezime DESC;

--5.
SELECT grad AS "grad",
       adresa AS "adresa",
       postanski_kod AS "postanski_kod"
FROM Poslovnica p, Lokacija l
WHERE p.lokacija_id = l.lokacija_id;

--6.
SELECT ime || ' ' || prezime AS "ime_prezime",
       Trunc(Months_Between(datum_zaposlenja, SYSDATE)) AS "datum",
       Add_Months(datum_zaposlenja, 15) AS "mjeseci_plus"
FROM Uposlenik u, Funkcija f
WHERE u.funkcija_id = f.funkcija_id AND f.naziv LIKE 'Direktor';

--7.
SELECT k.ime || ' ' || k.prezime AS "naziv",
       To_Char(Round(o.novca_uplaceno), '9999') AS "novca_uplaceno",
       To_Char(Trunc(Ceil(o.porez_na_dobitak)), '9.') AS "porez_na_dobitak"
FROM Opklade o, Korisnik k
WHERE o.korisnik_id = k.korisnik_id AND k.ime LIKE '%a%' ;

--8.
SELECT DISTINCT RPad(p.naziv, 15, '.') AS "tackica",
       RPad(k.prezime , 15, '*') AS "zvjezdica",
       RPad(u.prezime, 15, '?') AS "upitnik"
FROM Poslovnica p, Uposlenik u, Korisnik k
WHERE k.poslovnica_id = p.poslovnica_id AND u.poslovnica_id =
p.poslovnica_id;

--9.
SELECT DISTINCT UPPER(s.naziv) AS "VelikaSlova_Subjekt"
FROM Subjekt s;

--10.
SELECT REPLACE(l.grad , 'a', ' ') AS "Zamijeni_Prazno"
FROM Lokacija l;

```

```

-- 5 GRUPNE FUNKCIJA + 2 SA HAVING
--1.
SELECT u.prezime AS "uposlenik",
       o.naziv AS "odjel_uposlenika",
       f.naziv AS "funkcija_uposlenika",
       Avg(u.plata) AS "prosjecna_plata_uposlenika"
FROM Uposlenik u,Odjel o,Funkcija f
WHERE u.odjel_id = o.odjel_id AND u.funkcija_id = f.funkcija_id
GROUP BY u.prezime,o.naziv,f.naziv;
--2.
SELECT k.prezime AS "Korisnik",
       Sum(o.novca_uplaceno) AS "Novca_uplatio_na_opklade"
FROM korisnik k, Opklade o
WHERE o.korisnik_id = k.korisnik_id
GROUP BY k.prezime;
--3.
SELECT k.prezime AS "Korisnik",
       Max(r.stanje) AS "Najbogatiji"
FROM Racun r, Korisnik k , Korisnik_Racun kr
WHERE kr.korisnik_id = k.korisnik_id AND kr.racun_id = r.racun_id
GROUP BY k.prezime
HAVING Max(r.stanje) >= ALL (SELECT r1.stanje
                             FROM Racun r1) ;
--4.
SELECT p.naziv AS "poslovnice",
       tp.tip AS "tip_poslovnice" ,
       Min(p.Sredstva)AS "stanje_"
FROM Poslovnica p ,Tip_Poslovnice tp
WHERE p.tip_id = tp.tip_id
GROUP BY p.naziv,tp.tip
HAVING Min(p.sredstva) <= ALL (SELECT p1.sredstva
                              FROM Poslovnica p1);
--5.
SELECT o.naziv AS "odjel",
       Count(u.prezime) AS "Broj_uposlenih_u_odjelu"
FROM Uposlenik u , Odjel o
WHERE u.odjel_id = o.odjel_id
GROUP BY o.naziv;

```

--5 UPITA SA KORISTENJEM PODUPITA--

--1.

```
SELECT u.ime || ' ' || u.prezime AS "Uposlenik",
       o.naziv AS "odjel",
       f.naziv AS "funkcija"
FROM Uposlenik u, Odjel o, Funkcija f
WHERE u.odjel_id = o.odjel_id AND u.funkcija_id = f.funkcija_id AND u.ime
NOT LIKE 'Angelina' AND o.odjel_id = (SELECT o1.odjel_id

FROM Uposlenik u1, Odjel o1

WHERE u1.odjel_id = o1.odjel_id

AND u1.ime LIKE 'Angelina');
```

--BEZ Angeline

--ODJELI U KOJIMA RADI Angelina

--2. -- SVI UPOSLENICI KOD KOJIH PLATA NIJE JEDNAKA MINIMALNOJ PLATI

```
SELECT u.ime || ' ' || u.prezime AS "Uposlenik",
       o.naziv AS "odjel",
       u.plata AS "plata"
FROM Uposlenik u, Odjel o
WHERE u.odjel_id = o.odjel_id AND u.plata <> (SELECT Min(u1.plata)
                                              FROM Uposlenik u1
                                              WHERE u1.odjel_id =
u.odjel_id)
ORDER BY u.plata;
```

--3.

```
SELECT p.naziv AS "Poslovnica",
       l.grad AS "Grad"
FROM Poslovnica p, Lokacija l
WHERE p.lokacija_id = l.lokacija_id AND l.grad = (SELECT l1.grad
                                                    FROM Lokacija l1
                                                    WHERE l1.grad LIKE
'%y%' ) ;
```

--4. -- daj mi sve opklade za koje je vise novca uplaceno od prosjecne vrijednosti novca uplacenih za sve opklade

```
SELECT DISTINCT s.naziv AS "Subjekt",
                tp.naziv AS "Tip_Opklade",
                Nvl(s.liga, 0) AS "Liga",
                o.novca_uplaceno AS "uplaceno"
FROM Opklade o, Listic l, Tip_Opklade tp, Subjekt s
```

```
WHERE l.opklada_id = o.opklade_id AND l.tip_id = tp.tip_id AND s.tip_id =  
tp.tip_id AND o.novca_uplaceno > (SELECT Avg(o1.novca_uplaceno)
```

```
FROM Opklade o1);
```

```
--Provjera
```

```
SELECT Avg (o.novca_uplaceno)
```

```
FROM Opklade o ;
```

```
--5. dobio na listicu
```

```
SELECT k.prezime AS "Korisnik",  
       o.novca_uplaceno AS "Uplatio",  
       p.datum AS "Datum_Uplate",  
       p.vrijeme_odrzavanja AS "Vrijeme"
```

```
FROM Korisnik k, Opklade o,Par p,Listic l
```

```
WHERE o.korisnik_id = k.korisnik_id AND l.opklada_id = o.opklade_id AND  
p.listic_id = l.listic_id AND p.ishod = (SELECT p1.realni_ishod
```

```
FROM Par p1,Korisnik k1
```

```
WHERE k.prezime = k1.prezime AND p1.Ishod LIKE 'poredak');
```

```
--5 UPITA SA VISE OD JEDNOG NIVOA PODUPITA--
```

```
--1.  daj mi sve one odjele , u kojima radi uposlenik koji ima %a%
```

```
SELECT *
```

```
FROM Odjel o
```

```
WHERE o.odjel_id = ANY (SELECT o1.odjel_id  
                        FROM Odjel o1,Upislenik u1  
                        WHERE u1.odjel_id = o1.odjel_id AND u1.ime LIKE  
'%a%');
```

```
--2. prikazati sve uposlene koji imaju platu vecu os prosjecne plate bilo  
kojeg odjela
```

```
SELECT u.ime || ' ' || u.prezime AS "Upislenik",  
       o.naziv AS "Odjel",  
       u.plata AS "Plata"
```

```
FROM Upislenik u,Odjel o
```

```
WHERE u.odjel_id = o.odjel_id AND u.plata >ANY (SELECT Avg(u1.plata)  
                                                FROM Upislenik u1,Odjel o1  
                                                WHERE u1.odjel_id =
```

```
o1.odjel_id
```

```
GROUP BY o1.odjel_id);
```

```

--3.
SELECT *
FROM Subjekt s,Tip_Opklade tp
WHERE s.tip_id = tp.tip_id AND broj_subjekata =ALL( SELECT
tp1.broj_subjekata
FROM Tip_Opklade tp1
WHERE
tp1.broj_subjekata = '2');
--4.
SELECT k.ime|| ' ' || k.prezime AS "Korisnik",
       k.starost AS "godine",
       p.naziv AS "Poslovnica"
FROM Korisnik k,Poslovnica p
WHERE k.poslovnica_id = p.poslovnica_id AND p.naziv IN (SELECT p1.naziv
FROM Poslovnica p1
WHERE
p.poslovnica_id = p1.poslovnica_id AND p1.naziv = 'Millennium'
) ;
--5.
SELECT u.ime || ' ' || u.prezime AS "Uposlenik",
       o.naziv AS "Odjel",
       f.naziv AS "Funkcija",
       p.naziv AS "Poslovnica",
       u.plata AS "min_plata"
FROM Uposlenik u ,Odjel o,Funkcija f,Poslovnica p
WHERE u.odjel_id = o.odjel_id AND u.funkcija_id = f.funkcija_id AND
u.poslovnica_id = p.poslovnica_id AND u.plata <= ALL ( SELECT u1.plata

FROM Uposlenik u1);

--2 upita sa subtotalima (ROLLUP, CUBE, GROUPING SETS)
--1.
SELECT p.naziv AS "Poslovnica",
       Nvl(tp.tip,'0') AS "Tip_Poslovnice",
       Sum(p.sredstva) AS "Sredstva"
FROM Poslovnica p,Tip_Poslovnice tp
WHERE p.tip_id = tp.tip_id
GROUP BY ROLLUP(p.naziv,tp.tip);
--2.
SELECT u.prezime AS "Uposlenik",
       Round(Sum(u.plata),'99999') AS "Plata"
FROM Uposlenik u

```

```
GROUP BY Cube(u.prezime);
```

```
--3. koristenje Grouping Sets
```

```
SELECT k.ime AS "Ime",  
       k.prezime AS "Prezime",  
       Avg(r.stanje) AS "Prosjecno_Stanje_Racuna"  
FROM Korisnik k,Racun r, Korisnik_Racun kr  
WHERE kr.korisnik_id = k.korisnik_id AND kr.racun_id = r.racun_id  
GROUP BY Grouping SETS((k.ime,k.prezime));
```

```
--2 upita sa TOP N analizom   Poslovnica->Korisnik Korisnik->Poslovnica ->  
POZADINSKA LOGIKA
```

```
--TOP DOWN -- PARENT -> CHILD
```

```
SELECT LPad(p.naziv,Length(p.naziv) + (LEVEL * 2) - 2,'_') AS "Org_Chart"  
FROM Poslovnica p, Korisnik k  
START WITH p.naziv = 'Millennium'  
CONNECT BY NOCYCLE PRIOR p.poslovnica_id = k.korisnik_id;
```

```
--BOTTOM UP -- CHILD -> PARENT
```

```
SELECT LPad(p.naziv,Length(p.naziv) + (LEVEL * 2) - 2,'_') AS "Org_Chart"  
FROM Poslovnica p, Korisnik k  
START WITH p.naziv = 'Millennium'  
CONNECT BY NOCYCLE PRIOR p.poslovnica_id = k.korisnik_id;
```

```
--UNION,INTERSPECT,MINUS,UNION ALL
```

```
SELECT p.naziv  
FROM Poslovnica p  
UNION
```

```
SELECT l.grad  
FROM Lokacija l;
```

```
-- Indexi --
```

```
--1.
```

```
CREATE INDEX poslovnica_naziv_idx  
ON Poslovnica(naziv);
```

```
--2.
```

```
CREATE INDEX uposlenik_ime_prezime_idx  
ON Uposlenik(ime,prezime);
```

```
--3
```

```
CREATE INDEX korisnik_ime_prezime_idx  
ON Korisnik(ime,prezime);
```

```

--4.
CREATE INDEX odjel_naziv_idx
ON Odjel(naziv);

--5.
CREATE INDEX Uposlenik_plata_idx
ON Uposlenik(plata);

--DROP INDEX index;

-- Osnovna namjena INDEX-a je da ubrza proces vracanja slogova iz baze
koristenjem pointera.
-- Nad kolonama koje najcesce koristim pri kreiranju upita kreirala sam
INDEX :
-- (Naziv Poslovnice -> poslovnica_naziv_idx,

-- Ime i Prezime Uposlenika -> uposlenik_ime_prezime_idx,
-- Ime i Prezime Korisnika -> korisnik_ime_prezime_idx,
-- Naziv Odjela -> odjel_naziv_idx,
-- Plata Uposlenika -> Uposlenik_plata_idx).
-- Tako sam ubrzala proces dohvacanja slogova iz baze podataka nad ovim
kolonama.

-- 10. PROCEDURA
--1.
CREATE OR REPLACE PROCEDURE ObrisiKorisnika(uv_prezime VARCHAR2 )
IS
BEGIN
    DELETE FROM Korisnik
    WHERE prezime = uv_prezime;
COMMIT;
END;

SELECT * FROM Korisnik;
EXECUTE ObrisiKorisnika('Tolkien');

CREATE OR REPLACE PROCEDURE test_obrisi IS
BEGIN
    Dbms_Output.ENABLE(100000);
    dbms_Output.ObrisiKorisnika('Tolkien');
END;

```



```

BEGIN
test_obrisi;
END;

EXEC sys.test_obrisi;

EXECUTE ObrisiKorisnika 'Tolkien';

--ObrisiKorisnika();
call ObrisiKorisnika('Tolkien');

--2.
CREATE OR REPLACE PROCEDURE ObrisiUposlenika(uv_prezime IN
Uposlenik.Prezime%TYPE )
IS
BEGIN
DELETE FROM Uposlenik
WHERE prezime = uv_prezime;
COMMIT;
END;

--3.
CREATE OR REPLACE PROCEDURE ObrisiPoslovnica(uv_naziv IN
Poslovnica.Naziv%TYPE )
IS
BEGIN
DELETE FROM Poslovnica
WHERE naziv = uv_naziv;
COMMIT;
END;

--4.
CREATE OR REPLACE PROCEDURE UnesiUposlenika(up_ime Uposlenik.Ime%TYPE
,up_prezime Uposlenik.Prezime%TYPE )
IS
BEGIN
INSERT INTO Uposlenik
VALUES (uposlenik_seqv.NEXTVAL,up_ime,up_prezime
,'123456789',SYSDATE,
```

```

(SELECT funkcija_id
FROM Funkcija f
WHERE f.naziv = 'Direktor'),9850.56,

(SELECT odjel_id
FROM Odjel o
WHERE o.naziv = 'Administracija'),

(SELECT poslovnica_id
FROM Poslovnica p
WHERE p.naziv = 'Meridian')

);

COMMIT;
END;

EXECUTE UnesiUposlenika('Tom','Cruise');
SELECT * FROM Uposlenik;

```

--5.

```

CREATE OR REPLACE PROCEDURE UnesiKorisnika(v_ime Korisnik.Ime%TYPE
,v_prezime Korisnik.Prezime%TYPE ,
                                v_starost
Korisnik.Starost%TYPE ,v_poslovnica Poslovnica.Naziv%TYPE )
IS
BEGIN
    INSERT INTO Korisnik
    VALUES( korisnik_seqv.NEXTVAL ,
                                (SELECT poslovnica_id
                                FROM Poslovnica p
                                WHERE p.naziv = v_poslovnica)
    ,v_ime,v_prezime, v_starost);

```

```

COMMIT;
END;
    SELECT * FROM Korisnik;
EXECUTE UnesiKorisnika('Wil','Smith','25','Meridian');
--6.
CREATE OR REPLACE PROCEDURE PromijeniAdresuIGrad(v_adresa
Lokacija.Adresa%TYPE , v_grad Lokacija.Grad%TYPE )
IS
p_naziv Poslovnica.Naziv%TYPE;

CURSOR Naziv IS
SELECT p.naziv
FROM Poslovnica p, Lokacija l
WHERE p.lokacija_id = l.lokacija_id AND p.naziv = 'Play';

BEGIN
    OPEN Naziv;
    FETCH Naziv INTO p_naziv;
    CLOSE Naziv;

    IF(p_naziv = 'Play')
    THEN
        UPDATE Lokacija
        SET adresa= v_adresa , grad = v_grad;
    END IF;
COMMIT;
END;

SELECT p.naziv,l.adresa,l.grad
FROM Poslovnica p, Lokacija l
WHERE p.lokacija_id = l.lokacija_id AND p.naziv = 'Play';

EXECUTE PromijeniAdresuIGrad('Torggatan','GothenBurg');
SELECT * FROM Lokacija;

--7.
ALTER TRIGGER NoviUposlenik_triger DISABLE;
SELECT * FROM Uposlenik;
UPDATE Uposlenik u
SET u.dodatak_na_platu =0
WHERE u.ime = 'Sarah Jessica';
ALTER TRIGGER NoviUposlenik_triger ENABLE;

```

```

ALTER TABLE Uposlenik ADD dodatak_na_platu NUMBER ;

CREATE OR REPLACE PROCEDURE PromijeniPlatu
IS
BEGIN
UPDATE Uposlenik u
      SET      (plata,dodatak_na_platu) = (SELECT
-- UKOLIKO IMAJU DODATAK PLATA IM JE
UVECANNA ZA TAJ DODATAK
-- UKOLIKO NEMAJU DODATAK SMANJIMO
DODATAK ZA 10 % , A PLATU UVECAMO ZA 15 %

Decode(Nvl(dodatak_na_platu,0),
-- AKO NEMAJU DODATAK SMANJIMO PLATU ZA
10 POSTO

0, plata * 0.9,
-- U SUPROTNOM POVECAMO PLATU
plata + dodatak_na_platu),
--DECODE ZA DODATAK
Decode(Nvl(plata,0),
-- DODATAK SE UVECA ZA 15
0, 0.15 *dodatak_na_platu,
dodatak_na_platu)

FROM Uposlenik u1
WHERE u1.uposlenik_id = u.uposlenik_id)

WHERE
-- NAPRAVIM PODUPIT KOJI CE MI VRATITI SVE ONE KOJI ZIVE I RADE U NEW YORKU
( SELECT l.grad
FROM Uposlenik u2 ,Odjel o ,Lokacija l ,Poslovnica p
WHERE u2.odjel_id = o.odjel_id AND u2.poslovnica_id =
p.poslovnica_id AND p.lokacija_id = l.lokacija_id AND u.uposlenik_id =
u.uposlenik_id) LIKE 'Sarajevo';
COMMIT;
END;

--8.
CREATE OR REPLACE PROCEDURE UnesiSubjekta(uv_naziv IN
Tip_Opklade.Naziv%TYPE,uv_brojSubjekata IN Tip_Opklade.Broj_Subjekata%TYPE,
vv_naziv IN

```

```

Subjekt.naziv%TYPE, uv_liga IN Subjekt.Liga%TYPE,uv_igraci
Subjekt.Igraci%TYPE,uv_statistika Subjekt.Statistika%TYPE )
IS
BEGIN
INSERT INTO Subjekt
VALUES (subjekt_seqv.NEXTVAL,(SELECT tip_id
FROM Tip_Opklade
WHERE naziv = uv_naziv AND
broj_subjekata = uv_brojSubjekata),
vv_naziv,uv_liga,uv_igraci,uv_statistika);

COMMIT;
END;

```

--9.

```

CREATE OR REPLACE PROCEDURE ObrisiOpkladu
IS
BEGIN
DELETE FROM Opklade
WHERE novca_uplaceno < 2;
COMMIT;
END;

```

--10.

```

CREATE OR REPLACE PROCEDURE UpdateNazivOdjela
IS
BEGIN
UPDATE Odjel o
SET o.naziv = (SELECT
Decode(naziv,
'Americki','US%',
'OS%')
FROM Uposlenik u, Odjel o1
WHERE u.odjel_id =o1.odjel_id AND o1.odjel_id =
o.odjel_id)
WHERE (SELECT r.ime
FROM Poslovnica p, Uposlenik u,Odjel o2,Lokacija l,Drzava d,Regija
r
WHERE o2.odjel_id = o.odjel_id AND u.odjel_id = o2.odjel_id AND
u.poslovnica_id = p.poslovnica_id
AND p.lokacija_id = l.lokacija_id AND l.drzava_id =
d.drzava_id AND d.regija_id = r.regija_id ) LIKE 'Americki';
COMMIT;

```

```

        END;

-- 10. FUNKCIJA
--1.
    CREATE OR REPLACE FUNCTION dajPlatu_func(v_prezime IN
Uposlenik.prezime%TYPE)
    RETURN NUMBER
    IS
        v_plata Uposlenik.plata %TYPE := 0;
    BEGIN
        SELECT u.plata
        INTO   v_plata
        FROM   Uposlenik u
        WHERE  u.prezime = v_prezime;

        RETURN(v_plata);

    END;

    SELECT dajPlatu_func('Dedic') AS "Funkcija"
    FROM dual ;

--2.
    CREATE OR REPLACE FUNCTION dajPoslovniciKorisnika_func(v_prezime
Korisnik.prezime%TYPE)
    RETURN VARCHAR2
    IS
        v_poslovnica Poslovnica.naziv%TYPE;
    BEGIN
        SELECT p.naziv
        INTO   v_poslovnica
        FROM   Poslovnica p,Korisnik k
        WHERE  k.poslovnica_id = p.poslovnica_id AND k.prezime = v_prezime;

        RETURN(v_poslovnica);
    END;

    SELECT * FROM Korisnik;
    SELECT * FROM Poslovnica;
    SELECT dajPoslovniciKorisnika_func('Defoe') AS "Funkcija"
    FROM dual;

--3.
    CREATE OR REPLACE FUNCTION dajStanjeRacunaKorisnika_func(v_prezime IN

```

```

Korisnik.prezime%TYPE)
    RETURN NUMBER
IS
    v_stanjeRacuna Racun.stanje%TYPE := 0;
BEGIN
    SELECT r.stanje
    INTO v_stanjeRacuna
    FROM Korisnik k,Racun r,Korisnik_Racun kr
    WHERE kr.korisnik_id = k.korisnik_id AND kr.racun_id = r.racun_id AND
k.prezime = v_prezime;

    RETURN(v_stanjeRacuna);
END;

SELECT * FROM Korisnik;
SELECT dajStanjeRacunaKorisnika_func('Tolkien') AS "Funkcija"
FROM dual;
--4.
CREATE OR REPLACE FUNCTION dajUplatu(v_prezime IN
Korisnik.prezime%TYPE)
    RETURN NUMBER
IS
    v_uplatio Opklade.Novca_Uplaceno%TYPE := 0;
BEGIN
    SELECT o.novca_uplaceno
    INTO v_uplatio
    FROM Korisnik k, Opklade o
    WHERE o.korisnik_id = k.korisnik_id AND k.prezime = v_prezime;

    RETURN (v_uplatio);
END;

SELECT * FROM Korisnik;
SELECT dajUplatu('Tolkien') AS "funkcija"
FROM dual;
--5.
CREATE OR REPLACE FUNCTION AdresaPoslovnice_func(v_naziv
Poslovnica.naziv%TYPE)
    RETURN VARCHAR2
IS
    uv_adresa Lokacija.adresa%TYPE;
BEGIN

```

```

SELECT l.adresa
INTO uv_adresa
FROM Poslovnica p,Lokacija l
WHERE p.lokacija_id = l.lokacija_id AND p.naziv = v_naziv;

RETURN (uv_adresa);
END;

SELECT AdresaPoslovnice_func('Meridian') AS "funkcija"
FROM dual;

```

--6.

```

CREATE OR REPLACE FUNCTION VратиTipPoslovnice_func(v_naziv IN
Poslovnica.naziv%TYPE)
RETURN varchar2
IS
uv_tipPoslovnice Tip_Poslovnice.tip%TYPE;
BEGIN
SELECT tp.tip
INTO uv_tipPoslovnice
FROM Poslovnica p,Tip_Poslovnice tp
WHERE p.tip_id = tp.tip_id AND p.naziv = v_naziv;

RETURN(uv_tipPoslovnice);
END;

SELECT VратиTipPoslovnice_func('Millennium') AS "Funkcija"
FROM dual;

```

--7.

```

CREATE OR REPLACE TYPE dvije_vrijednosti AS object (uplaceno
NUMBER,porez NUMBER);
CREATE OR REPLACE FUNCTION KorisnikOpklada_func( v_prezime
Korisnik.prezime%TYPE,uv_porez OUT NUMBER )
RETURN NUMBER
--dvije_vrijednosti
IS
uv_uplaceno Opklade.novca_uplaceno%TYPE ;
-- uv_porez      Opklade.porez_na_dobitak%TYPE;

```



```

CURSOR KorisnikUplatioKursor IS
    SELECT o.novca_uplaceno
    FROM Korisnik k,Opklade o
    WHERE o.korisnik_id = k.korisnik_id;

CURSOR KorisnikPorezKursor IS
    SELECT o.porez_na_dobitak
    FROM Korisnik k, Opklade o
    WHERE o.korisnik_id = k.korisnik_id;

BEGIN
    OPEN KorisnikUplatioKursor;
    FETCH KorisnikUplatioKursor INTO uv_uplaceno;
    CLOSE KorisnikUplatioKursor;

    OPEN KorisnikPorezKursor;
    FETCH KorisnikPorezKursor INTO uv_porez;
    CLOSE KorisnikPorezKursor;

    --RETURN dvije_vrijednosti(uv_uplaceno,uv_porez);
    RETURN uv_uplaceno;

END;

--SELECT KorisnikOpklada_func('Tolkien',o.porez_na_dobitak)AS "funkcija"
--FROM dual,Opklade o;

--8.
CREATE OR REPLACE FUNCTION VратиNaziveKorisnika
RETURN VARCHAR2
IS
    v_naziv VARCHAR2(100) := ' ';
BEGIN
    SELECT k.ime || ' ' || k.prezime
    INTO v_naziv
    FROM Korisnik k ;

    RETURN(v_naziv);
END;

SELECT * FROM Uposlenik;

```

```

CREATE OR REPLACE FUNCTION NazivUposlenika(v_prezime
Korisnik.prezime%TYPE)
RETURN VARCHAR2
IS
v_nazivUposlenika VARCHAR2(50);

CURSOR Uposlenik IS
SELECT up.ime || ' ' || up.prezime
FROM Uposlenik up, Korisnik k, Poslovnica p
WHERE k.poslovnica_id = p.poslovnica_id AND up.poslovnica_id =
p.poslovnica_id AND k.prezime = v_prezime;

BEGIN
OPEN Uposlenik;
FETCH Uposlenik INTO v_nazivUposlenika;
CLOSE Uposlenik;
RETURN(v_nazivUposlenika);
END;

SELECT * FROM Korisnik;
SELECT NazivUposlenika('Tolkien') AS "Funkcija"
FROM dual;

--9.
CREATE OR REPLACE FUNCTION DajFunkcijuUposlenika(v_prezime IN
Uposlenik.prezime%TYPE)
RETURN varchar2
IS
v_funkcija VARCHAR2(50);

CURSOR FunkcijaUposlenika IS
SELECT f.naziv
FROM Funkcija f, Uposlenik u
WHERE u.funkcija_id = f.funkcija_id AND u.prezime = v_prezime;

BEGIN
OPEN FunkcijaUposlenika;
FETCH FunkcijaUposlenika INTO v_funkcija;
CLOSE FunkcijaUposlenika;
RETURN(v_funkcija);
END;

```

```

SELECT * FROM Uposlenik;
SELECT DajFunkcijuUposlenika('English') AS "Funkcija"
FROM dual;
--10.
CREATE OR REPLACE FUNCTION DatumOdigranogListica(uv_prezime IN
Korisnik.Prezime%TYPE )
RETURN DATE
IS
v_datum DATE;

CURSOR DatumListica IS
SELECT p.datum
FROM Par p, Listic l,Opklade o, Korisnik k
WHERE p.listic_id = l.listic_id AND l.opklada_id = o.opklade_id AND
o.korisnik_id = k.korisnik_id AND k.prezime = uv_prezime;

BEGIN
OPEN DatumListica;
FETCH DatumListica INTO v_datum;
CLOSE DatumListica;

RETURN(v_datum);
END;

SELECT * FROM Korisnik;
SELECT DatumOdigranogListica('Defoe') AS "Funkcija"
FROM dual;

-- 10. TRIGERA
--1.
CREATE OR REPLACE TRIGGER UnesiOpkladu_triger
BEFORE UPDATE OF datum_opklade ON Opklade
FOR EACH ROW
BEGIN
IF(To_Char(SYSDATE,'DY') IN ('MON','TUE','WED','THU','FRI'))
THEN
IF(To_Date(To_Char(SYSDATE,'HH24-MI'),'HH24-MI') BETWEEN
To_Date('19-00','HH24-MI') AND To_Date('08-30','HH24-MI'))
THEN Raise_Application_Error('-20501','Radni dan (08,30 -
19,00),Update Opklade');
END IF;
END IF;

```

```

IF (To_Char(SYSDATE, 'DY') IN ('SAT', 'SUN'))
THEN
    IF (To_Date(To_Char(SYSDATE, 'HH24-MI'), 'HH24-MI') BETWEEN
To_Date('15-30', 'HH24-MI') AND To_Date('09-00', 'HH24-MI'))
    THEN Raise_Application_Error('-20502', 'Vikend (09,00 - 15,30),Update
Opklade');
    END IF;
END IF;
END;

```

```

UPDATE Opklade
SET datum_opklade = To_Date('12.03.2017 21-45', 'DD.MM.YYYY
HH24-MI');

```

```

ALTER TRIGGER UnesiOpkladu_triger DISABLE;

```

```

DROP TRIGGER UnesiOpkladu_triger;

```

```

SELECT * FROM Opklade;
ALTER TABLE Opklade ADD datum_opklade DATE ;
UPDATE Opklade SET datum_opklade = SYSDATE;

```

--2.

```

CREATE OR REPLACE TRIGGER PredefVrijUposlenik_Triger
BEFORE INSERT ON Uposlenik
FOR EACH ROW
BEGIN
IF (:new.uposlenik_id IS NULL) THEN
    :NEW.uposlenik_id := uposlenik_seqv.NEXTVAL;
END IF;
END;
SELECT * FROM Uposlenik;

```

```

        DELETE FROM Uposlenik
        WHERE ime = 'Mirela';
INSERT INTO Uposlenik
VALUES (NULL, 'Mirela', 'Dedic', '34566', SYSDATE,
(SELECT funkcija_id

```

```

FROM Funkcija f

WHERE f.naziv = 'Racunovodja'),2350.56,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Racunovodstvo'),

(SELECT poslovnic_id

FROM Poslovnica p

WHERE p.naziv = 'Play')

);
--3.
--NEMA SMISLA
CREATE OR REPLACE TRIGGER PlataUposlenik_trigger
BEFORE INSERT OR UPDATE OF plata ON Uposlenik
FOR EACH ROW
BEGIN
    IF(:new.plata = 0 ) THEN
        Raise_Application_Error('-20503','Uposlenik radi za dzabe');
    END IF;
END;

UPDATE Uposlenik
SET plata = 0;

--4.
CREATE OR REPLACE TRIGGER TipPoslovnice_triger
BEFORE INSERT OR UPDATE OR DELETE OF tip ON Tip_Poslovnice
FOR EACH ROW
BEGIN
    IF(:new.tip NOT IN ('Obicna','Regionalna-Centralna','Glavna'))
    THEN Raise_Application_Error('-20504','Ne mozete dodati,mijenjati ni
    brisati tip poslovnice');

```

```

        END IF;
    END;

SELECT * FROM Tip_Poslovnice;
DELETE FROM Tip_Poslovnice
WHERE tip = 'Nova';
INSERT INTO Tip_Poslovnice
VALUES (Tip_Poslovnice_seqv.NEXTVAL , 'Nova');
--5.
CREATE OR REPLACE TRIGGER Tip_Opklade_triger
BEFORE INSERT ON Tip_Opklade
FOR EACH ROW
BEGIN
    IF (:new.naziv = 'Sportske') THEN :new.broj_subjekata := 2;
    ELSIF (:new.naziv = 'Ostale') THEN :new.broj_subjekata := 1;
    ELSE
        Raise_Application_Error('-20505', 'Tip->Sportska->Opklada->brojSubjekata->2
        ili Tip->Ostale->Opklade->brojSubjekata->1 ');
    END IF;
END;

ALTER TABLE Pogodnosti RENAME COLUMN pogodnosti TO pogodnosti_kolona;

--6.
CREATE OR REPLACE TRIGGER NoviUposlenik_triger
BEFORE INSERT OR UPDATE ON Uposlenik
FOR EACH ROW
DECLARE
    c_nt INTEGER := 0;
BEGIN
    SELECT Count(*) INTO c_nt FROM uposlenik WHERE ime = :new.ime AND prezime =
    :new.prezime AND telefon = :new.telefon ;
    IF (c_nt > 0)
    THEN Raise_Application_Error('-20506', 'Uposlenik sa tim imenom i prezimenom
    vec postoji ');
    END IF;
END;

DELETE FROM Uposlenik
WHERE prezime = 'English';

INSERT INTO Uposlenik

```

```

VALUES
(uposlenik_seqv.NEXTVAL, 'Johnny', 'English', '09867455872', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Direktor'), 9850.56,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Administracija'),

(SELECT poslovnica_id

FROM Poslovnica p

WHERE p.naziv = 'Millennium')

);

```

--7.

```

CREATE OR REPLACE TRIGGER SredstvaPoslovnice_triger
BEFORE INSERT OR UPDATE OR DELETE OF sredstva ON Poslovnica
FOR EACH ROW
BEGIN
    IF (:old.sredstva = 0 OR :new.sredstva = 0)
    THEN
        Raise_Application_Error('-20507', 'Poslovnica probada, nema
sredstava');
    END IF;
END;

SELECT * FROM Poslovnica;

UPDATE Poslovnica
SET sredstva = 0;

```

```
--8.
CREATE OR REPLACE TRIGGER Korisnik_Poslovnica_triger
BEFORE INSERT OR UPDATE OF korisnik_id ON Korisnik
FOR EACH ROW
BEGIN
    IF (:new.korisnik_id = 0 OR :new.korisnik_id IS NULL ) THEN
        Raise_Application_Error('-20507','Korisnik_id je nula ili null ');
    END IF;
END;
```

```
SELECT * FROM Korisnik;
```

```
INSERT INTO Korisnik
VALUES( NULL ,
        (SELECT poslovница_id
         FROM Poslovnica p
         WHERE p.naziv = 'Millennium')
        , 'Mirela', 'Dedic', 38);
```

```
--9.
CREATE OR REPLACE TRIGGER AzurirajStanjeRacuna_triger
BEFORE UPDATE OF stanje ON Racun
FOR EACH ROW
BEGIN
    -- UPDATE Racun
    -- SET stanje = :old.stanje;
    IF (:old.stanje <> :new.stanje) THEN
        Raise_Application_Error('-20508','Pokusali ste promijeniti stanje racuna
korisnika,to nije dozvoljeno,stoga je zadržano prethodno stanje');
    END IF;
END;
```

```
UPDATE Racun
SET stanje = 1;
```

```
--10.
```



```

CREATE OR REPLACE TRIGGER NazivPoslovnica_triger
BEFORE UPDATE OF naziv ON Poslovnica
FOR EACH ROW
BEGIN
    IF(:old.naziv LIKE :new.naziv) THEN
Raise_Application_Error('-20509','Nije dozvoljeno mijenjati ime
poslovnice');
    END IF;
END;

UPDATE Poslovnica
SET naziv = 'Millennium';

--Skripta--
call dbms_output.enable();
begin
    dbms_output.put_line('Statistika poslovnica: ');
    dbms_output.put_line('-----');
end;

CREATE GLOBAL TEMPORARY TABLE statistika_poslovnica ON COMMIT DELETE ROWS
AS (SELECT
    poslovnica.POSLOVNICA_ID,
    POSLOVNICA.NAZIV AS "poslovnica",
    SUM(OPKLADE.NOVCA_UPLACENO) AS "uplate",
    SUM(NVL(R.STANJE, 0.0)) AS "racuni_stanje_ukupno",
    COUNT(R.RACUN_ID) AS "broj_racuna",
    COUNT(KORISNIK.KORISNIK_ID) AS "broj_korisnika"
FROM
    KORISNIK
    LEFT OUTER JOIN KORISNIK_RACUN KR
        ON KORISNIK.KORISNIK_ID = KR.KORISNIK_ID
    LEFT OUTER JOIN RACUN R
        ON KR.RACUN_ID = R.RACUN_ID,
    OPKLADE,
    POSLOVNICA
WHERE
    KORISNIK.POSLOVNICA_ID = POSLOVNICA.POSLOVNICA_ID AND
    OPKLADE.KORISNIK_ID = KORISNIK.KORISNIK_ID
GROUP BY POSLOVNICA.POSLOVNICA_ID, POSLOVNICA.NAZIV);

begin

```

```

for cur in (select * FROM statistika_poslovnica)
loop
    dbms_output.put_line('Poslovnica: ' || cur."poslovnica" || '(' ||
cur.POSLOVNICA_ID || ')');
    dbms_output.put_line('Uplate: ' || cur."uplate" || ' KM');
    dbms_output.put_line('Stanje racuna ukupno: ' ||
cur."racuni_stanje_ukupno" || ' KM');
    dbms_output.put_line('Broj racuna: ' || cur."racuni_stanje_ukupno" || '
KM');
    dbms_output.put_line('Broj korisnika: ' || cur."broj_korisnika");
    dbms_output.put_line('_____');
end loop;
end;

begin

    dbms_output.put_line('Uposlenici sa najvećim doprinosom(po poslovnicama):
');
for cur in (select * FROM statistika_poslovnica)
loop

    dbms_output.put_line('Poslovnica: ' || cur."poslovnica" || '(' ||
cur.POSLOVNICA_ID || ')');
    for up in (SELECT UPOSLENIK_ID, IME, PREZIME, TELEFON FROM UPOSLENIK
WHERE UPOSLENIK.POSLOVNICA_ID = cur.POSLOVNICA_ID) LOOP
        dbms_output.put_line('Uposlenik: ' || up.IME || ' ' || up.PREZIME ||
' (' || up.UPOSLENIK_ID || ')');
        dbms_output.put_line('Kontakt: ' || up.TELEFON);
        dbms_output.put_line('_____');
    END LOOP;
end loop;
end;

begin

    dbms_output.put_line('Klijenti sa najvećim uplatama(po poslovnicama): ');
for cur in (select * FROM statistika_poslovnica)
loop

    dbms_output.put_line('Poslovnica: ' || cur."poslovnica" || '(' ||
cur.POSLOVNICA_ID || ')');
    for k in (SELECT KORISNIK.KORISNIK_ID, IME, PREZIME, SUM(RACUN.STANJE)

```

AS stanje

```
        FROM KORISNIK
        LEFT OUTER JOIN KORISNIK_RACUN KR
        ON KORISNIK.KORISNIK_ID = KR.KORISNIK_ID
        LEFT OUTER JOIN RACUN
        ON KR.RACUN_ID = RACUN.RACUN_ID
        WHERE KORISNIK.POSLOVNICA_ID = cur.POSLOVNICA_ID) LOOP
    dbms_output.put_line('Klijent: ' || k.IME || ' ' || k.PREZIME || ' '
(' || k.KORISNIK_ID || '));
    dbms_output.put_line('Stanje: ' || k.stanje);
    dbms_output.put_line('_____');
END LOOP;
end loop;
end;
TRUNCATE TABLE statistika_poslovnica;
DROP TABLE statistika_poslovnica;
call dbms_output.disable();
```

-- kreiranje tabela --

```
CREATE TABLE Uposlenik ( uposlenik_id INT NOT NULL,
    ime VARCHAR2(20) NOT NULL,
    prezime VARCHAR2(20) NOT NULL,
    telefon VARCHAR2(20) NOT NULL,
    datum_zaposlenja DATE NOT NULL,
    funkcija_id INT NOT NULL,
    plata NUMBER NOT NULL,
    odjel_id INT NOT NULL,
    poslovnica_id INT NOT NULL);

SELECT * FROM uposlenik;
ALTER TABLE Uposlenik ADD CONSTRAINT uposlenik_id_pk PRIMARY KEY
(uposlenik_id);
ALTER TABLE Funkcija ADD CONSTRAINT funkcija_id_pk PRIMARY KEY
(funkcija_id);

ALTER TABLE Uposlenik ADD CONSTRAINT funkcija_id_fk FOREIGN KEY
(funkcija_id) REFERENCES Funkcija(funkcija_id);
ALTER TABLE Uposlenik ADD CONSTRAINT odjel_id_fk FOREIGN KEY (odjel_id)
REFERENCES Odjel(odjel_id);
ALTER TABLE Uposlenik ADD CONSTRAINT poslovnica_id_fk FOREIGN KEY
(poslovnica_id) REFERENCES Poslovnica(poslovnica_id);
```

```
CREATE SEQUENCE uposlenik_seqv
INCREMENT BY 1
START WITH 1
MAXVALUE 100000
NOCACHE
NOCYCLE;
```

```
SELECT funkcija_seqv.NEXTVAL FROM USER_SEQUENCES ;
SELECT uposlenik_seqv.NEXTVAL FROM USER_SEQUENCES;
```

```
SELECT * FROM funkcija;
SELECT * FROM odjel;
SELECT * FROM Poslovnica;
SELECT * FROM Uposlenik;
```

```
INSERT INTO Uposlenik
VALUES
(uposlenik_seqv.NEXTVAL, 'Johnny', 'English', '09867455872', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Direktor'), 9850.56,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Administracija'),

(SELECT poslovnica_id

FROM Poslovnica p

WHERE p.naziv = 'Millennium')

);
INSERT INTO Uposlenik
```

```

VALUES (uposlenik_seqv.NEXTVAL, 'Marshall
', 'Mathers', '063789521', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Racunovodja'), 2150.26,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Racunovodstvo'),

(SELECT poslovnica_id

FROM Poslovnica p

WHERE p.naziv = 'Premier')

);
INSERT INTO Uposlenik
VALUES (uposlenik_seqv.NEXTVAL, 'Mitchell ', 'J.
Marr', '063789521', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Zastitar'), 2150.26,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Odrzavanje'),

```

```
(SELECT poslovnica_id  
  
FROM Poslovnica p  
  
WHERE p.naziv = 'Mozzart')
```

```
);  
INSERT INTO Uposlenik  
VALUES (uposlenik_seqv.NEXTVAL, 'Sandy', 'J.  
McKenzie', '063789521', SYSDATE,
```

```
(SELECT funkcija_id  
  
FROM Funkcija f  
  
WHERE f.naziv = 'Striptizeta'), 2150.26,
```

```
(SELECT odjel_id  
  
FROM Odjel o  
  
WHERE o.naziv = 'Tehnicka Podrska'),
```

```
(SELECT poslovnica_id  
  
FROM Poslovnica p  
  
WHERE p.naziv = 'Play')
```

```
);  
INSERT INTO Uposlenik  
VALUES  
(uposlenik_seqv.NEXTVAL, 'Angelina', 'Jolie', '0613987526', SYSDATE,  
  
(SELECT funkcija_id  
  
FROM Funkcija f
```

```
WHERE f.naziv = 'Striptizeta'),2150.26,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Tehnicka Podrska'),

(SELECT poslovnica_id

FROM Poslovnica p

WHERE p.naziv = 'Millennium')

);
INSERT INTO Uposlenik
VALUES
(uposlenik_seqv.NEXTVAL, 'Samantha', 'Jones', '0338756241', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Kaser'),1236.56,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Kadrovska Sluzba'),

(SELECT poslovnica_id

FROM Poslovnica p

WHERE p.naziv = 'Millennium')
```

```

);
INSERT INTO Uposlenik
    VALUES (uposlenik_seqv.NEXTVAL, 'Sarah
Jessica', 'Parker', '0983256647', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Racunovodja'), 2350.56,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Racunovodstvo'),

(SELECT poslovnica_id

FROM Poslovnica p

WHERE p.naziv = 'Play')

);
INSERT INTO Uposlenik
    VALUES (uposlenik_seqv.NEXTVAL, 'Leonardo', 'Di
Caprio', '033698122', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Direktor'), 9850.56,

(SELECT odjel_id

```



```

FROM Odjel o

WHERE o.naziv = 'Administracija'),

(SELECT poslovnic_id

FROM Poslovnica p

WHERE p.naziv = 'Lutrija BIH')

);
INSERT INTO Uposlenik
VALUES (uposlenik_seqv.NEXTVAL, 'Will', 'Smith', '0785448544', SYSDATE,

(SELECT funkcija_id

FROM Funkcija f

WHERE f.naziv = 'Cistacica'), 9850.56,

(SELECT odjel_id

FROM Odjel o

WHERE o.naziv = 'Odrzavanje'),

(SELECT poslovnic_id

FROM Poslovnica p

WHERE p.naziv = 'Millennium')

);
INSERT INTO Uposlenik
VALUES
(uposlenik_seqv.NEXTVAL, 'Dakota', 'Johnson', '011255244', SYSDATE,

```

```

(SELECT funkcija_id
FROM Funkcija f
WHERE f.naziv = 'Kaser'),9850.56,

(SELECT odjel_id
FROM Odjel o
WHERE o.naziv = 'Kadrovska Sluzba'),

(SELECT poslovnica_id
FROM Poslovnica p
WHERE p.naziv = 'Meridian')

);
INSERT INTO Uposlenik
VALUES
(uposlenik_seqv.NEXTVAL,'Emma','Stone','125586214455000',SYSDATE,

(SELECT funkcija_id
FROM Funkcija f
WHERE f.naziv = 'Kaser'),9850.56,

(SELECT odjel_id
FROM Odjel o
WHERE o.naziv = 'Kadrovska Sluzba'),

(SELECT poslovnica_id

```



```
CREATE SEQUENCE funkcija_seqv
INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 1000000
NOCYCLE
NOCACHE;
```

```
INSERT INTO Funkcija
VALUES (funkcija_seqv.NEXTVAL, 'Direktor');
```

```
INSERT INTO Funkcija
VALUES (funkcija_seqv.NEXTVAL, 'Racunovodja');
```

```
INSERT INTO Funkcija
VALUES (funkcija_seqv.NEXTVAL, 'Cistacica');
```

```
INSERT INTO Funkcija
VALUES (funkcija_seqv.NEXTVAL, 'Zastitar');
```

```
INSERT INTO Funkcija
VALUES (funkcija_seqv.NEXTVAL, 'Kaser');
```

```
INSERT INTO Funkcija
VALUES (funkcija_seqv.NEXTVAL, 'Striptizeta');
```

```
CREATE TABLE Poslovnica( poslovnica_id INT NOT NULL ,
                           naziv VARCHAR2(30) NOT NULL ,
                           lokacija_id INT NOT NULL ,
                           tip_id INT NOT NULL ,
                           sredstva NUMBER,
                           CONSTRAINT poslovnica_id_pf PRIMARY KEY
(poslovnica_id),
                           CONSTRAINT posl_tip_id_fk FOREIGN KEY (tip_id)
REFERENCES Tip_Poslovnice(tip_id));
```

```
ALTER TABLE Poslovnica ADD CONSTRAINT lokacija_id_fk FOREIGN KEY
(lokacija_id) REFERENCES Lokacija(lokacija_id);
```

```
CREATE SEQUENCE poslovnica_seqv
```

```

INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 100000
NOCACHE
NOCYCLE;

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'Premier',(SELECT lokacija_id
FROM Lokacija l,Drzava d
WHERE l.drzava_id =
d.drzava_id AND d.ime = 'Bosna i Hercegovina'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip = 'Obicna'),

1540000.98);

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'Lutrija BIH',(SELECT lokacija_id
FROM Lokacija l,Drzava
d
WHERE l.drzava_id =
d.drzava_id AND d.ime = 'Bosna i Hercegovina'),

(SELECT tip_id
FROM Tip_Poslovnice
tp
WHERE tp.tip =
'Regionalna-Centralna'),

152354.98);

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'Mozzart',(SELECT lokacija_id
FROM lokacija l,drzava d
WHERE l.drzava_id =
d.drzava_id AND d.ime = 'Hrvatska'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip =
'Regionalna-Centralna'),

```

```

87234.34);

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'BetLive', (SELECT lokacija_id
FROM lokacija l,drzava d
WHERE l.drzava_id =
d.drzava_id AND d.ime = 'Srbija'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip = 'Obicna'),

14897.92);

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'Millennium', (SELECT lokacija_id
FROM lokacija l,drzava
d
WHERE l.drzava_id =
d.drzava_id AND d.ime = 'Meksiko'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip =
'Glavna'),

12345623.50);

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'Play', (SELECT lokacija_id
FROM lokacija l,drzava d
WHERE l.drzava_id =
d.drzava_id AND d.ime = 'Ujedinjene drzave'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip =
'Regionalna-Centralna'),

98540.23);

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'Meridian', (SELECT lokacija_id
FROM lokacija l,drzava d
WHERE l.drzava_id =

```

```

d.drzava_id AND d.ime = 'Japan'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip =

'Obicna'),

13580.54);

/*
SELECT l.grad,d.ime
FROM lokacija l,drzava d
WHERE l.drzava_id = d.drzava_id AND d.ime = 'Kina';
*/

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'SportPlus',(SELECT lokacija_id
FROM lokacija l,drzava d
WHERE l.drzava_id =

d.drzava_id AND d.ime = 'Kina'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip =

'Regionalna-Centralna'),

342000.90);

INSERT INTO Poslovnica
VALUES(poslovnica_seqv.NEXTVAL, 'Soccer',(SELECT lokacija_id
FROM lokacija l,drzava d
WHERE l.drzava_id =

d.drzava_id AND d.ime = 'Kanada'),

(SELECT tip_id
FROM Tip_Poslovnice tp
WHERE tp.tip = 'Obicna'),

56900.20);

INSERT INTO Poslovnica
VALUES (poslovnica_seqv.NEXTVAL,'Games', (SELECT lokacija_id
FROM lokacija l,drzava d
WHERE l.drzava_id =

d.drzava_id AND d.ime = 'Kongo'),

```

```

        (SELECT tip_id
         FROM Tip_Poslovnice tp
         WHERE tp.tip = 'Obicna'),

        1500);

INSERT INTO Poslovnica
        VALUES(Poslovnica_seqv.NEXTVAL, 'KladionicaBF', (SELECT lokacija_id
        FROM lokacija

        WHERE l.drzava_id =

        d.drzava_id AND d.ime = 'Burkina Faso'),

        tp

        (SELECT tip_id
         FROM Tip_Poslovnice

        WHERE tp.tip =

        'Obicna'),

        89564.34);

CREATE TABLE Tip_Poslovnice( tip_id INT NOT NULL ,
        tip VARCHAR2(50) NOT NULL );

ALTER TABLE Tip_Poslovnice ADD CONSTRAINT posl_tip_id_pf PRIMARY KEY
(tip_id);

CREATE SEQUENCE tip_poslovnice_seqv
INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 99999999
NOCYCLE
NOCACHE;

INSERT INTO Tip_Poslovnice
        VALUES (tip_poslovnice_seqv.NEXTVAL, 'Obicna');

INSERT INTO Tip_Poslovnice
        VALUES (tip_poslovnice_seqv.NEXTVAL, 'Regionalna-Centralna');

INSERT INTO Tip_Poslovnice

```



```
VALUES (tip_poslovnice_seqv.NEXTVAL, 'Glavna');
```

```
CREATE TABLE Lokacija ( lokacija_id INT,  
                          adresa VARCHAR2(50),  
                          postanski_kod INT,  
                          grad VARCHAR2(30),  
                          drzava_id INT);
```

```
ALTER TABLE Lokacija ADD CONSTRAINT lokacija_pk PRIMARY KEY (lokacija_id);  
ALTER TABLE Lokacija ADD CONSTRAINT drzava_id_fk FOREIGN KEY (drzava_id)  
REFERENCES Drzava(drzava_id);
```

```
SELECT * FROM lokacija;
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Ibrahima Ljubovica  
45', 71210, 'Sarajevo',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Bosna i  
Hercegovina'));
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Knez Mihailova', 11000, 'Beograd',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Srbija'));
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Rudolf Fizir', 10110, 'Zagreb',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Hrvatska'));
```

```
UPDATE Lokacija  
SET adresa = 'Rudolf Fizir', grad = 'Zagreb'  
WHERE postanski_kod = 10110;
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Av Vicente Guarrero', 06600, 'Ciudad  
de Juarez ',
```

```
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Meksiko'));
```

```
UPDATE Lokacija  
SET adresa = 'Av Vicente Guarrero', grad = 'Ciudad de Juarez'  
WHERE postanski_kod = 06600;
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Pennsylvania Ave  
NW', 20001, 'Washington',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Ujedinjene drzave'));
```

```
UPDATE Lokacija  
SET adresa = 'Pennsylvania Ave NW', grad = 'Washington'  
WHERE postanski_kod = 20001;
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'QuensWay', 01200, 'Ottawa',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Ottawa'));
```

```
UPDATE Lokacija  
SET adresa = 'QuensWay', grad = 'Washington'  
WHERE postanski_kod = 01200;
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Mita Dori', 000010, 'Tokyo',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Japan'));
```

```
UPDATE Lokacija  
SET adresa = 'Mita Dori', grad = 'Tokyo'  
WHERE postanski_kod = 000010;
```

```
DELETE FROM Lokacija l  
WHERE l.adresa = 'Mita Dori';
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Toshima-ku', 12300, 'Beijing',
```

```
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Kina'));
```

```
UPDATE Lokacija  
SET adresa = 'Toshima-ku', grad = 'Beijing'  
WHERE postanski_kod = 12300;
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Av De La ', 198547, 'Ouagadougou',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Burkina Faso'));
```

```
UPDATE Lokacija  
SET adresa = 'Av De La', grad = 'Ouagadougou'  
WHERE postanski_kod = 198547;
```

```
INSERT INTO Lokacija  
VALUES (lokacija_seqv.NEXTVAL, 'Boulevard SendWe', 111111, 'Kinshasa',  
(SELECT drzava_id FROM drzava d WHERE d.ime = 'Kongo'));
```

```
UPDATE Lokacija  
SET adresa = 'Boulevard SendWe', grad = 'Kinshasa'  
WHERE postanski_kod = 111111;
```

```
SELECT * FROM lokacija;  
SELECT * FROM drzava;
```

```
CREATE SEQUENCE lokacija_seqv  
INCREMENT BY 1  
START WITH 1  
MINVALUE 1  
MAXVALUE 999999  
NOCACHE  
NOCYCLE;
```

```
CREATE TABLE Drzava ( drzava_id INT,  
ime VARCHAR2(50) NOT NULL ,  
regija_id INT);
```

```
ALTER TABLE Drzava ADD CONSTRAINT drzava_pk PRIMARY KEY(drzava_id);  
ALTER TABLE Drzava ADD CONSTRAINT regija_id_fk FOREIGN KEY (regija_id)
```

```
REFERENCES Regija(regija_id);
```

```
CREATE SEQUENCE drzava_seqv  
INCREMENT BY 1  
START WITH 1  
MINVALUE 1  
MAXVALUE 9999999  
NOCACHE  
NOCYCLE;
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Bosna i Hercegovina', (SELECT regija_id  
FROM regija r WHERE r.ime = 'Europa'));
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Hrvatska', (SELECT regija_id FROM regija  
r WHERE r.ime = 'Europa'));
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Srbija', (SELECT regija_id FROM regija r  
WHERE r.ime = 'Europa'));
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Kanada', (SELECT regija_id FROM regija r  
WHERE r.ime = 'Amerika'));
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Japan', (SELECT regija_id FROM regija r  
WHERE r.ime = 'Azija'));
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Burkina Faso', (SELECT regija_id FROM  
regija r WHERE r.ime = 'Srednji Istok i Afrika'));
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Kongo', (SELECT regija_id FROM regija r  
WHERE r.ime = 'Srednji Istok i Afrika'));
```

```
INSERT INTO Drzava  
VALUES (drzava_seqv.NEXTVAL, 'Kina', (SELECT regija_id FROM regija r  
WHERE r.ime = 'Azija'));
```

```
INSERT INTO Drzava
VALUES (drzava_seqv.NEXTVAL, 'Meksiko', (SELECT regija_id FROM regija r
WHERE r.ime = 'Amerika'));
```

```
INSERT INTO Drzava
VALUES (drzava_seqv.NEXTVAL, 'Ujedinjene drzave', (SELECT regija_id
FROM regija r WHERE r.ime = 'Amerika'));
```

```
DELETE FROM drzava
WHERE ime = 'Kanada';
```

```
DELETE FROM drzava
WHERE ime = 'Burkina Faso';
```

```
SELECT * FROM regija;
SELECT * FROM drzava;
```

```
CREATE SEQUENCE regija_seqv
INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 9999999
NOCYCLE
NOCACHE ;
```

```
CREATE TABLE Regija ( regija_id INT ,
ime VARCHAR2(50) NOT NULL,
CONSTRAINT regija_id_pk PRIMARY KEY (regija_id));
```

```
INSERT INTO Regija
VALUES (regija_seqv.NEXTVAL, 'Europa');
```

```
INSERT INTO Regija
VALUES (regija_seqv.NEXTVAL, 'Amerika');
```

```
INSERT INTO Regija
VALUES (regija_seqv.NEXTVAL, 'Azija');
```

```
INSERT INTO Regija
VALUES (regija_seqv.NEXTVAL, 'Srednji Istok i Afrika');
```

```
-- DELERE ROW --
```

```
DELETE FROM Regija
WHERE ime = 'Europa';
```

```
CREATE TABLE Korisnik(korisnik_id INT,
                        poslovница_id INT,
                        ime VARCHAR2(20) NOT NULL ,
                        prezime VARCHAR2(30) NOT NULL ,
                        starost INT NOT NULL ,
                        CONSTRAINT klijent_id_pk PRIMARY KEY (korisnik_id));
```

```
ALTER TABLE Korisnik ADD CONSTRAINT korisnik_poslovnica_id_fk FOREIGN
KEY(poslovnica_id) REFERENCES Poslovnica(poslovnica_id);
```

```
INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovница_id
       FROM Poslovnica p
       WHERE p.naziv = 'Play')
      , 'Dan', 'Brown', 74);
```

```
INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovница_id
       FROM Poslovnica p
       WHERE p.naziv = 'Millennium')
      , 'Jane', 'Austen', 38);
```

```
INSERT INTO Korisnik
VALUES ( korisnik_seqv.NEXTVAL ,
      (SELECT poslovница_id
       FROM Poslovnica p
       WHERE p.naziv = 'Millennium'),
      'Ernest', 'HamingWay', 23);
```

```
INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovница_id
       FROM Poslovnica p
       WHERE p.naziv = 'Premier')
      , 'J.K.', 'Rowling', 19);
```

```

INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovnic_id
       FROM Poslovnica p
       WHERE p.naziv = 'Meridian')
      , 'John R.R.', 'Tolkien', 45);
DELETE FROM Korisnik
WHERE ime = ' Tolkien';
INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovnic_id
       FROM Poslovnica p
       WHERE p.naziv = 'Soccer')
      , 'Miguel', 'De Cervantes', 54);

INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovnic_id
       FROM Poslovnica p
       WHERE p.naziv = 'Meridian')
      , 'Daniel', 'Defoe', 25 );

INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovnic_id
       FROM Poslovnica p
       WHERE p.naziv = 'Play')
      , 'Jonathan', 'Swift', 21 );

INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovnic_id
       FROM Poslovnica p
       WHERE p.naziv = 'Mozzart')
      , 'Charles', 'Dickens', 63 );
INSERT INTO Korisnik
VALUES( korisnik_seqv.NEXTVAL ,
      (SELECT poslovnic_id
       FROM Poslovnica p
       WHERE p.naziv = 'Games')
      , 'Leo', 'Tolstoy', 29 );
INSERT INTO Korisnik

```

```

VALUES( korisnik_seqv.NEXTVAL ,
        (SELECT poslovnica_id
         FROM Poslovnica p
         WHERE p.naziv = 'BetLive')
        , 'Fyodor', 'Dostoevsky', 55 );

/*
(SELECT last_number
 FROM user_sequences
 WHERE sequence_name LIKE 'POSLOVNICA_SEQV')
CREATE SEQUENCE korisnik_seqv
*/
INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 999999
NOCACHE
NOCYCLE;

CREATE TABLE Racun( racun_id INT,
                     stanje NUMBER NOT NULL ,
                     CONSTRAINT racun_id_pk PRIMARY KEY (racun_id));

CREATE SEQUENCE racun_seqv
INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 9999999
NOCACHE
NOCYCLE;

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL, 3200.56);

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL, 125.56);

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL, 3298.56);

INSERT INTO Racun

```



```

VALUES (racun_seqv.NEXTVAL,21900.33);

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL,3);

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL,0 );

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL,1269.41);

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL,100.56);

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL,159000.23);

INSERT INTO Racun
VALUES (racun_seqv.NEXTVAL,320.76);


CREATE TABLE korisnik_racun(korisnik_id INT,
                             racun_id INT );

INSERT INTO korisnik_racun
VALUES( (SELECT korisnik_id
        FROM Korisnik k
        WHERE k.prezime ='Tolkien' ),

        (SELECT racun_id
        FROM Racun r
        WHERE r.stanje = 21900.33));

SELECT * FROM Korisnik;
SELECT * FROM Racun;

INSERT INTO korisnik_racun
VALUES( (SELECT korisnik_id
        FROM Korisnik k
        WHERE k.prezime ='De Cervantes' ),

```

```

        (SELECT racun_id
         FROM Racun r
         WHERE r.stanje = 3298.56));

INSERT INTO korisnik_racun
VALUES( (SELECT korisnik_id
        FROM Korisnik k
        WHERE k.prezime = 'Brown' ),

        (SELECT racun_id
         FROM Racun r
         WHERE r.stanje = 159000.23));

INSERT INTO korisnik_racun
VALUES( (SELECT korisnik_id
        FROM Korisnik k
        WHERE k.prezime = 'Austen' ),

        (SELECT racun_id
         FROM Racun r
         WHERE r.stanje = 320.76));

ALTER TABLE korisnik_racun ADD CONSTRAINT korisnik_id_fk FOREIGN KEY
(korisnik_id) REFERENCES Korisnik(korisnik_id);

ALTER TABLE korisnik_racun ADD CONSTRAINT racun_id_fk FOREIGN KEY
(racun_id) REFERENCES Racun(racun_id);

CREATE TABLE Opklade (opklade_id INT,
                      korisnik_id INT,
                      novca_uplaceno NUMBER,
                      porez_na_dobitak NUMBER,
                      CONSTRAINT opklade_id_pk PRIMARY KEY (opklade_id),
                      CONSTRAINT opklade_korisnik_fk FOREIGN
KEY(korisnik_id) REFERENCES Korisnik(korisnik_id));
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Dostoevsky'),

```

```

523.80,0.23);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Tolstoy'),
45.30,0.21);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Dickens'),
100.20,0.18);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Swift'), 2 ,0.01);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Defoe'),
1498.50,0.62);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Tolkien'),
34.80,0.23);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'De Cervantes'),
8.23,0.05);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Rowling'),
1054.90,0.5);

```

```

INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'HamingWay'),
1024.10,0.06);
INSERT INTO Opklade
VALUES(opklade_seqv.NEXTVAL,
      (SELECT korisnik_id
       FROM Korisnik k
       WHERE k.prezime = 'Austen'),
2408.80,0.35);

CREATE SEQUENCE opklade_seqv
INCREMENT BY 1
START WITH 0
MINVALUE 0
MAXVALUE 9999999
NOCACHE
NOCYCLE;

ALTER TABLE Opklade
DROP COLUMN datum_opklade;

CREATE TABLE Pogodnosti( pogodnosti_id INT,
                          opklade_id INT,
                          pogodnosti VARCHAR2(50),
                          CONSTRAINT pogodnosti_id_pk PRIMARY KEY
(pogodnosti_id),
                          CONSTRAINT pogodnosti_opklada_fk FOREIGN KEY
(opklade_id) REFERENCES Opklade(opklade_id) );

CREATE SEQUENCE pogodnosti_seqv
INCREMENT BY 1
START WITH 0
MINVALUE 0
MAXVALUE 9999999
NOCACHE
NOCYCLE;
INSERT INTO Pogodnosti
VALUES (pogodnosti_seqv.NEXTVAL,

```



```

'LISTIC_SEQV'),

                                (SELECT tip_id
                                   FROM Tip_Opklade tp
                                   WHERE tp.naziv = 'Sportske'));

INSERT INTO Listic
      VALUES (listic_seqv.NEXTVAL,
                                (SELECT last_number
                                   FROM user_sequences
                                   WHERE sequence_name LIKE

'LISTIC_SEQV'),

                                (SELECT tip_id
                                   FROM Tip_Opklade tp
                                   WHERE tp.naziv = 'Ostale'));

INSERT INTO Listic
      VALUES (listic_seqv.NEXTVAL,
                                (SELECT last_number
                                   FROM user_sequences
                                   WHERE sequence_name LIKE

'LISTIC_SEQV'),

                                (SELECT tip_id
                                   FROM Tip_Opklade tp
                                   WHERE tp.naziv = 'Sportske'));

INSERT INTO Listic
      VALUES (listic_seqv.NEXTVAL,
                                (SELECT last_number
                                   FROM user_sequences
                                   WHERE sequence_name LIKE

'LISTIC_SEQV'),

                                (SELECT tip_id
                                   FROM Tip_Opklade tp
                                   WHERE tp.naziv = 'Ostale'));

INSERT INTO Listic
      VALUES (listic_seqv.NEXTVAL,
                                (SELECT last_number
                                   FROM user_sequences
                                   WHERE sequence_name LIKE

'LISTIC_SEQV'),

```

```
(SELECT tip_id
FROM Tip_Opklade tp
WHERE tp.naziv = 'Sportske'));
```

```
CREATE SEQUENCE listic_seqv
INCREMENT BY 1
START WITH 0
MINVALUE 0
MAXVALUE 99999999
NOCACHE
NOCYCLE;
```

```
CREATE TABLE Par (par_id INT,
listic_id INT,
datum DATE NOT NULL,
vrijeme_odrzavanja DATE NOT NULL,
ishod VARCHAR2(20) NOT NULL,
realni_ishod VARCHAR2(20) NOT NULL,
listic_ishod VARCHAR2(20) NOT NULL,
CONSTRAINT par_id_pk PRIMARY KEY (par_id),
CONSTRAINT par_listic_fk FOREIGN KEY (listic_id)
REFERENCES Listic(listic_id)
);
```

```
DELETE FROM Par
WHERE listic_id = NULL;
INSERT INTO Par
VALUES(par_seqv.NEXTVAL,
(SELECT listic_id
FROM Listic l,Opklade o
WHERE l.opklada_id= o.opklade_id and
o.NOVCA_UPLACENO =45.3), SYSDATE,SYSDATE ,'pobjeda','nerjeseno','gubitak'
);
INSERT INTO Par
VALUES(par_seqv.NEXTVAL,
(SELECT listic_id
FROM Listic l,Opklade o
WHERE l.opklada_id= o.opklade_id and
o.NOVCA_UPLACENO = 2408.8), SYSDATE,SYSDATE ,'pobjeda','pobjeda','dobitak'
);
```

```

INSERT INTO Par
    VALUES(par_seqv.NEXTVAL,

            (SELECT listic_id
             FROM Listic l,Opklade o
             WHERE l.opklada_id= o.opklade_id and

o.NOVCA_UPLACENO = 2) ,
To_Date('08.11.2013', 'DD.MM.YYYY'),To_Date('08.11.2013.
20-00-27', 'DD.MM.YYYY HH24-MI-SS')

, 'nerjeseno', 'pobjeda', 'gubitak');
INSERT INTO Par
    VALUES(par_seqv.NEXTVAL,

            (SELECT listic_id
             FROM Listic l,Opklade o
             WHERE l.opklada_id= o.opklade_id and

o.NOVCA_UPLACENO = 523.8 ) ,
To_Date('23.11.2014', 'DD.MM.YYYY'),To_Date('23.11.2014.
21-45-27', 'DD.MM.YYYY HH24-MI-SS'),

'pobjeda', 'pobjeda', 'dobitak');

INSERT INTO Par
    VALUES(par_seqv.NEXTVAL,

            (SELECT listic_id
             FROM Listic l,Opklade o
             WHERE l.opklada_id= o.opklade_id and

o.NOVCA_UPLACENO = 1024.1 ),
To_Date('18.07.2016', 'DD.MM.YYYY'),To_Date('18.07.2016.
07-30-57', 'DD.MM.YYYY HH24-MI-SS'),

'gubitak', 'gubitak', 'gubitak');

INSERT INTO Par
    VALUES(par_seqv.NEXTVAL,

            (SELECT listic_id
             FROM Listic l,Opklade o
             WHERE l.opklada_id= o.opklade_id and

o.NOVCA_UPLACENO = 100.2 ),
To_Date('02.08.2015', 'DD.MM.YYYY'),To_Date('02.08.2015.
20-32-45', 'DD.MM.YYYY HH24-MI-SS'),

```



```
'poredak','poredak','gubitak');
```

```
INSERT INTO Par
VALUES(par_seqv.NEXTVAL,
      (SELECT listic_id
       FROM Listic l,Opklade o
       WHERE l.opklada_id= o.opklade_id and
o.NOVCA_UPLACENO = 1498.5 ), SYSDATE,SYSDATE
,'pobjeda','nerjeseno','gubitak' );
CREATE SEQUENCE par_seqv
INCREMENT BY 1
START WITH 0
MINVALUE 0
MAXVALUE 99999999
NOCACHE
NOCYCLE;
```

```
CREATE TABLE Tip_Opklade(tip_id INT,
                          naziv VARCHAR2(50) NOT NULL,
                          broj_subjekata INT NOT NULL,
                          CONSTRAINT tip_id_pk PRIMARY KEY (tip_id));
```

```
CREATE SEQUENCE tip_opklade_seqv
INCREMENT BY 1
START WITH 0
MINVALUE 0
MAXVALUE 99999999
NOCACHE
NOCYCLE;
```

```
INSERT INTO Tip_Opklade
VALUES (tip_opklade_seqv.NEXTVAL,'Sportske',2);
```

```
INSERT INTO Tip_Opklade
VALUES (tip_opklade_seqv.NEXTVAL,'Ostale',1);
```

```
SELECT * FROM Tip_Opklade;
CREATE TABLE Subjekt (subjekt_id INT,
                      tip_id INT,
                      naziv VARCHAR2(20) NOT NULL,
```

```

        liga VARCHAR2(30),
        igraci VARCHAR(100),
        statistika VARCHAR2(100),
        CONSTRAINT subjekt_id_pk PRIMARY KEY (subjekt_id),
        CONSTRAINT subjekt_tip_id_fk FOREIGN KEY (tip_id)
REFERENCES Tip_Opklade(tip_id));

CREATE SEQUENCE subjekt_seqv
INCREMENT BY 1
START WITH 0
MINVALUE 0
MAXVALUE 9999999
NOCACHE
NOCYCLE;
INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
        (SELECT tip_id
        FROM Tip_Opklade tp
        WHERE tp.naziv = 'Sportske'),

        'Roma','Italijanska','Dzeko, Totti, Perotti,Kolarov
','Statistika Roma : Broj_Golova 3, Broj_Kornera: 7,Broj_Pokusaja 5' );

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
        (SELECT tip_id
        FROM Tip_Opklade tp
        WHERE tp.naziv = 'Sportske'),

        'Real Madrid','Spanska','Cristiano Ronaldo , Benzema , Tonny
Kros , Modric ','Statistika Real Madrid : Broj_Golova 8, Broj_Kornera:
18,Broj_Pokusaja 23');

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
        (SELECT tip_id
        FROM Tip_Opklade tp
        WHERE tp.naziv = 'Sportske'),

        'Manchester City ','Engleska','Aguero, Sterling ,Silva
','Statistika Manchester City : Broj_Golova 1, Broj_Kornera:
3,Broj_Pokusaja 90' );

```

```

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
      (SELECT tip_id
       FROM Tip_Opklade tp
       WHERE tp.naziv = 'Ostale'),

      'Formula1',NULL ,'    Lewis Hamilton','Brz 254 km/h' );

```

```

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
      (SELECT tip_id
       FROM Tip_Opklade tp
       WHERE tp.naziv = 'Ostale'),

      'Pas',NULL ,'    Pujdo ','Bio Prvi 2 Puta' );

```

```

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
      (SELECT tip_id
       FROM Tip_Opklade tp
       WHERE tp.naziv = 'Ostale'),

      'Macka',NULL ,'Cici','Pobjegla' );

```

```

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
      (SELECT tip_id
       FROM Tip_Opklade tp
       WHERE tp.naziv = 'Ostale'),

      'Kornjaca',NULL ,'Mimi','Brzina 2 km/h' );

```

```

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,
      (SELECT tip_id
       FROM Tip_Opklade tp
       WHERE tp.naziv = 'Ostale'),

      'Skijanje',NULL ,'Ivica Kostelic ','Osvojio Zlatni Globus' );

```

```

INSERT INTO Subjekt
VALUES(subjekt_seqv.NEXTVAL,

```

```
(SELECT tip_id
FROM Tip_Opklade tp
WHERE tp.naziv = 'Ostale'),

'Skijanje',NULL , 'Janica Kostelic ', 'Osvojila Zlatni Globus'
);

SELECT * FROM Subjekt;
DELETE FROM Subjekt
WHERE tip_id IS NULL;
```