

KREIRANJE TABELA

```
CREATE TABLE Lice(lice_id INTEGER PRIMARY KEY NOT NULL,  
  
    ime VARCHAR2(20),  
  
    prezime VARCHAR2(20),  
  
    JMBG VARCHAR2(13),  
  
    email VARCHAR2(20),  
  
    spol VARCHAR2(10),  
  
    CONSTRAINT c_lice_spol_CK CHECK (spol IN ('zensko','musko')));
```

```
CREATE TABLE Zaposleni( zaposlenik_id INTEGER PRIMARY KEY NOT NULL,  
  
    plata NUMBER(20),  
  
    dodatak_na_platu NUMBER(6),  
  
    datum_zaposlenja DATE,  
  
    funkcija_id INTEGER NOT NULL REFERENCES Funkcija_Koju_Obavlja(funkcija_id),  
  
    odjel_id INTEGER NOT NULL REFERENCES Odjeli(odjel_id),  
  
    lice_id INTEGER NOT NULL REFERENCES Lice(lice_id));
```

```
CREATE TABLE Funkcija_Koju_Obavlja(funkcija_id INTEGER PRIMARY KEY NOT NULL,  
  
    naziv VARCHAR2(20));
```

```
CREATE TABLE Odjeli(odjel_id INTEGER PRIMARY KEY NOT NULL,  
  
    naziv NUMBER(20),  
  
    sredstva NUMBER(30),  
  
    nadredjeni_odjel_id INTEGER NOT NULL REFERENCES Odjeli(odjel_id),  
  
    lokacija_id INTEGER NOT NULL REFERENCES Lokacije(lokacija_id));
```

```
CREATE TABLE Lokacije(lokacija_id INTEGER PRIMARY KEY NOT NULL,  
  
    grad varchar2(50),  
  
    adresa VARCHAR2(50),  
  
    drzava_id INTEGER NOT NULL REFERENCES Drzave(drzava_id));
```

```
CREATE TABLE Drzave(drzava_id INTEGER PRIMARY KEY NOT NULL,  
  
    naziv varchar2(50),  
  
    porez NUMBER(7));
```

```
CREATE TABLE Klijenti(klijent_id INTEGER PRIMARY KEY NOT NULL,  
  
    iznos_racuna NUMBER(20),  
  
    pogodnost_id INTEGER REFERENCES Pogodnosti(pogodnost_id),  
  
    opklada_id INTEGER REFERENCES Opklade(opklada_id),  
  
    lice_id INTEGER NOT NULL REFERENCES Lice(lice_id));
```

```
CREATE TABLE Pogodnosti(pogodnost_id INTEGER PRIMARY KEY NOT NULL,  
  
    tip_pogodnosti VARCHAR2(20));
```

```
CREATE TABLE Strategija(strategija_id INTEGER PRIMARY KEY NOT NULL,  
  
    tip_strategije VARCHAR2(30),  
  
CONSTRAINT c_strategija_tip_CK CHECK (tip_strategije IN ('pobjeda','poraz','nerijeseno')));
```

```
CREATE TABLE Opklade(opklada_id INTEGER PRIMARY KEY NOT NULL,  
  
    odjel_id INTEGER REFERENCES Odjeli(odjel_id),  
  
    datum_opklade DATE,  
  
    broj_listica INTEGER);
```

```
CREATE TABLE Opklade_arhiva(opklada_id INTEGER REFERENCES Opklade(opklada_id),  
    strategija_id INTEGER REFERENCES Strategija(strategija_id),  
    sport_lica_id INTEGER REFERENCES Sport_lica(sport_lica_id),  
    trka_id INTEGER REFERENCES Trke(trka_id));
```

```
CREATE TABLE Sport_lica(sport_lica_id INTEGER PRIMARY KEY NOT NULL,  
    naziv VARCHAR2(50),  
    tim_id INTEGER REFERENCES Sport_lica(sport_lica_id),  
    tip_sporta_id INTEGER REFERENCES Sport(sport_id));
```

```
CREATE TABLE Sport(sport_id INTEGER PRIMARY KEY NOT NULL,  
    naziv VARCHAR2(50),  
    liga_id INTEGER REFERENCES Liga(liga_id));
```

```
CREATE TABLE Liga(liga_id INTEGER PRIMARY KEY NOT NULL,  
    naziv VARCHAR2(50),  
    broj_ucesnika INTEGER,  
    bodovi INTEGER);
```

```
CREATE TABLE Trke(trka_id INTEGER PRIMARY KEY NOT NULL,  
    naziv VARCHAR2(50),  
    broj_ucesnika INTEGER,  
    datum_odrzavanja DATE);
```

```
CREATE TABLE Statistika(statistika_id INTEGER PRIMARY KEY NOT NULL,  
    sport_lica_id INTEGER REFERENCES Sport_lica(sport_lica_id),
```

```
trke_id INTEGER REFERENCES Trke(trka_id)

);
```

```
CREATE TABLE Statistika_arhiva(statistika_id INTEGER REFERENCES Statistika(statistika_id),
                                bodovi INTEGER,
                                tip VARCHAR2(20));
```

KREIRANJE SEKVENCI

```
CREATE SEQUENCE lice_id_sekv
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
nomaxvalue
```

```
minvalue 1
```

```
nocache
```

```
nocycle;
```

```
CREATE SEQUENCE zaposleni_id_sekv
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
nomaxvalue
```

```
minvalue 1
```

```
nocache
```

```
nocycle;
```

```
CREATE SEQUENCE klijenti_id_sekv
```

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

nocycle;

CREATE SEQUENCE odjeli_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

nocycle;

CREATE SEQUENCE lokacije_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

nocycle;

CREATE SEQUENCE drzave_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE pogodnosti_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE funkcije_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE strategija_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE opklade_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE trke_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE sport_lica_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE sport_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE liga_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE statistika_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

CREATE SEQUENCE statistika_arhiva_id_sekv

START WITH 1

INCREMENT BY 1

nomaxvalue

minvalue 1

nocache

NOCYCLE;

UNOS VRIJEDNOSTI U TABELE

--Lice

SELECT * FROM USER_objects;

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Edina','Kovac','1234567891123','ekovac2','zensko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Zerina','Ramic','1223567891123','zramic1','zensko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Samira','Mamela','1214567891123','smamela2','zensko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Damir','Pozderac','1213567891123','dpozderac2','musko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Haris','Soljic','1234567891223','hsoljic1','musko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Amina','Kurtovic','1234567891233','akurtovic1','zensko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Mujo','Kozic','1234517891123','mkozic1','musko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Hamo','Pita','1234511291123','hpita1','musko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Selam','Fukic','1234511291123','sfukic1','musko');

INSERT INTO Lice

VALUES(lice_id_sekv.NEXTVAL, 'Sanela','Pumic','1234511121123','spumic3','zensko');

--Zaposlenici

INSERT INTO Zaposleni

VALUES(zaposleni_id_sekv.currval,
700,0.05,To_Date('21.12.2014','dd.mm.yyyy'),zaposleni_id_sekv.currval,zaposleni_id_sekv.currval,zap
posleni_id_sekv.currval);

INSERT INTO Zaposleni

```
VALUES(zaposleni_id_sekv.nextval,  
700,0.05,To_Date('21.12.2014','dd.mm.yyyy'),zaposleni_id_sekv.currval,zaposleni_id_sekv.currval,za  
posleni_id_sekv.currval);
```

```
INSERT INTO Zaposleni
```

```
VALUES(zaposleni_id_sekv.nextval,1200,0.15,To_Date('21.01.2014','dd.mm.yyyy'),zaposleni_id_sekv.  
currval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);
```

```
INSERT INTO Zaposleni
```

```
VALUES(zaposleni_id_sekv.nextval,1000,0.005,To_Date('19.01.2018','dd.mm.yyyy'),zaposleni_id_sek  
v.currval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);
```

```
INSERT INTO Zaposleni
```

```
VALUES(zaposleni_id_sekv.nextval,1400,0.15,To_Date('21.01.2012','dd.mm.yyyy'),zaposleni_id_sekv.  
currval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);
```

```
INSERT INTO Zaposleni
```

```
VALUES(zaposleni_id_sekv.nextval,800,0.1,To_Date('28.07.2012','dd.mm.yyyy'),zaposleni_id_sekv.cu  
rrval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);
```

```
INSERT INTO Zaposleni
```

```
VALUES(zaposleni_id_sekv.nextval,1300,0.09,To_Date('21.01.2017','dd.mm.yyyy'),zaposleni_id_sekv.  
currval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);
```

```
INSERT INTO Zaposleni
```

```
VALUES(zaposleni_id_sekv.nextval,1210,0.15,To_Date('21.01.2015','dd.mm.yyyy'),zaposleni_id_sekv.  
currval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);
```

```
INSERT INTO Zaposleni
```

```
VALUES(zaposleni_id_sekv.nextval,1010,0,To_Date('21.01.2016','dd.mm.yyyy'),zaposleni_id_sekv.cur  
rval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);
```

INSERT INTO Zaposleni

VALUES(zaposleni_id_sekv.nextval,600,0.3,To_Date('21.01.2017','dd.mm.yyyy'),zaposleni_id_sekv.currval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);

INSERT INTO Zaposleni

VALUES(zaposleni_id_sekv.currval,900,0.07,To_Date('21.11.2017','dd.mm.yyyy'),zaposleni_id_sekv.currval,zaposleni_id_sekv.currval,zaposleni_id_sekv.currval);

--Odjeli

INSERT INTO Odjeli

VALUES(odjeli_id_sekv.currval,odjeli_id_sekv.CURRVAL,170000,odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);

INSERT INTO Odjeli

VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,121000,odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);

INSERT INTO Odjeli

VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,1212110,odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);

INSERT INTO Odjeli

VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,11223000,odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);

INSERT INTO Odjeli

```
VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,12100120, odjeli_id_sekv.CURRVAL-5,odjeli_id_sekv.CURRVAL);
```

```
INSERT INTO Odjeli
```

```
VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,12100120, odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);
```

```
INSERT INTO Odjeli
```

```
VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,12120120, odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);
```

```
INSERT INTO Odjeli
```

```
VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,12120120, odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);
```

```
INSERT INTO Odjeli
```

```
VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,121202120, odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);
```

```
INSERT INTO Odjeli
```

```
VALUES(odjeli_id_sekv.NEXTVAL,odjeli_id_sekv.CURRVAL,121202120, odjeli_id_sekv.CURRVAL,odjeli_id_sekv.CURRVAL);
```

```
--Lokacije
```

```
INSERT INTO Lokacije
```

```
VALUES(lokacije_id_sekv.NEXTVAL,'Sarajevo','Sutjeska 1', lokacije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokacije_id_sekv.NEXTVAL,'Tijuana','Tijuana 56', lokacije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Beograd','Kraljice Marije 4', lokalije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Zagreb','Trg Stjepana Radica 1', lokalije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Ljubljana','Ajdovščina 1', lokalije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Sofia','Odrin 57', lokalije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Kopenhagen','Aboulevard 64', lokalije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Berlin','Torstrasse 2', lokalije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Korint','Zaloggou 17', lokalije_id_sekv.currval);
```

```
INSERT INTO Lokacije
```

```
VALUES(lokalije_id_sekv.NEXTVAL,'Tokio','Hiju 17', lokalije_id_sekv.currval);
```

```
--Drzave
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Bosna i Hercegovina',0.17);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Meksiko',0.09);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Srbija',0.15);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Hrvatska',0.18);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Slovenija',0.13);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Bugarska',0.14);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Danska',0.07);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Njemacka',0.2);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.NEXTVAL,'Grcka',0.17);
```

```
INSERT INTO Drzave
```

```
VALUES(drzave_id_sekv.nextval,'Japan',0.17);
```

```
--Klijenti
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,1200,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,1500,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,2200,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,700,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,800,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,120,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```



```
VALUES(klijenti_id_sekv.nextval,1400,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,1010,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,1278,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
INSERT INTO Klijenti
```

```
VALUES(klijenti_id_sekv.nextval,1790,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL,klijenti_id_sekv.CURRVAL );
```

```
--Pogodnosti
```

```
INSERT INTO Pogodnosti
```

```
VALUES(pogodnosti_id_sekv.nextval,'odbijanje poreza');
```

```
INSERT INTO Pogodnosti
```

```
VALUES(pogodnosti_id_sekv.nextval,'stopa dobitka ++');
```

```
INSERT INTO Pogodnosti
```

```
VALUES(pogodnosti_id_sekv.nextval,'stopa dobitka --');
```

```
INSERT INTO Pogodnosti
```

```
VALUES(pogodnosti_id_sekv.nextval,'bonusi');
```

INSERT INTO Pogodnosti

VALUES(pogodnosti_id_sekv.nextval,'+1');

INSERT INTO Pogodnosti

VALUES(pogodnosti_id_sekv.nextval,'bingo');

INSERT INTO Pogodnosti

VALUES(pogodnosti_id_sekv.nextval,'1 besplatno');

INSERT INTO Pogodnosti

VALUES(pogodnosti_id_sekv.nextval,'preko interneta');

INSERT INTO Pogodnosti

VALUES(pogodnosti_id_sekv.nextval,'ekstra listic');

--Opklade

INSERT INTO Opklade

VALUES(opklade_id_sekv.NEXTVAL,opklade_id_sekv.CURRVAL,To_Date('21.12.2014','dd.mm.yyyy'),opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,sysdate,opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,To_Date('21.12.2017','dd.mm.yyyy'),opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,sysdate,opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,sysdate,opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,To_Date('29.12.2017','dd.mm.yyyy'),opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,To_Date('01.01.2018','dd.mm.yyyy'),opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,To_Date('17.01.2018','dd.mm.yyyy'),opklade_id_sekv.CURRVAL+500);

INSERT INTO Opklade

VALUES(opklade_id_sekv.nextVAL,opklade_id_sekv.CURRVAL,To_Date('19.01.2018','dd.mm.yyyy'),opklade_id_sekv.CURRVAL+500);

--Strategija

INSERT INTO strategija

VALUES(strategija_id_sekv.NEXTVAL, 'nerijeseno');

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'nerijeseno');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'nerijeseno');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'pobjeda');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'pobjeda');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'pobjeda');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'poraz');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'pobjeda');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'poraz');
```

```
INSERT INTO strategija  
VALUES(strategija_id_sekv.NEXTVAL, 'poraz');
```

```
SELECT * FROM strategija;
```

```
--Opklade Arhiva
```

```
--Funkcija Koju Obavlja
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'salter');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'zastitar');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'operater');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'cistac');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'blagajnik');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'menadzer');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'direktor');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'sekretar');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'salter-pomoc');
```

```
INSERT INTO funkcija_koju_obavlja
```

```
VALUES(funkcije_id_sekv.nextval,'zastitar 2');
```

```
--Trke
```

```
INSERT INTO Trke
```

```
VALUES(trke_id_sekv.nextval,'psi',23,To_Date('17.02.2018','dd.mm.yyyy'));
```

```
INSERT INTO Trke
```

```
VALUES(trke_id_sekv.nextval,'kornjace',23,To_Date('17.03.2018','dd.mm.yyyy'));
```

```
INSERT INTO Trke
```

```
VALUES(trke_id_sekv.nextval,'konji',23,To_Date('17.03.2018','dd.mm.yyyy'));
```

```
INSERT INTO Trke
```

```
VALUES(trke_id_sekv.nextval,'slonovi',23,To_Date('17.07.2018','dd.mm.yyyy'));
```

```
INSERT INTO Trke
```

```
VALUES(trke_id_sekv.nextval,'bikovi',23,To_Date('17.05.2018','dd.mm.yyyy'));
```

```
INSERT INTO Trke
```

```
VALUES(trke_id_sekv.nextval,'zecevi',23,To_Date('24.02.2018','dd.mm.yyyy'));
```

INSERT INTO Trke

VALUES(trke_id_sekv.nextval,'mravi',23,To_Date('17.02.2019','dd.mm.yyyy'));

INSERT INTO Trke

VALUES(trke_id_sekv.nextval,'magarci',23,To_Date('17.02.2020','dd.mm.yyyy'));

INSERT INTO Trke

VALUES(trke_id_sekv.nextval,'kamile',23,To_Date('17.06.2018','dd.mm.yyyy'));

INSERT INTO Trke

VALUES(trke_id_sekv.nextval,'nojevi',23,To_Date('17.09.2018','dd.mm.yyyy'));

--Sport Lica

INSERT INTO Sport_lica

VALUES(sport_lica_id_sekv.nextval,'Barcelona',NULL, 3);

INSERT INTO Sport_lica

VALUES(sport_lica_id_sekv.nextval,'Real Madrid',NULL, 3);

INSERT INTO Sport_lica

VALUES(sport_lica_id_sekv.nextval,'Mesi',2, 3);

INSERT INTO Sport_lica

VALUES(sport_lica_id_sekv.nextval,'Ronaldo',3, 3);

```
INSERT INTO Sport_lica  
VALUES(sport_lica_id_sekv.nextval,'Roma',null, 3);
```

```
INSERT INTO Sport_lica  
VALUES(sport_lica_id_sekv.nextval,'Dzeko',7, 3);
```

```
INSERT INTO Sport_lica  
VALUES(sport_lica_id_sekv.nextval,'Hirscher',null, 9);
```

```
INSERT INTO Sport_lica  
VALUES(sport_lica_id_sekv.nextval,'Girardelli',null, 9);
```

```
INSERT INTO Sport_lica  
VALUES(sport_lica_id_sekv.nextval,'Phelps',null, 7);
```

```
INSERT INTO Sport_lica  
VALUES(sport_lica_id_sekv.nextval,'De Rossi',7, 3);
```

--Sport

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Fudbal',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Nogomet',sport_id_sekv.CURRVAL);
```



```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Kosarka',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Rukomet',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Kosarka',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Plivanje',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Odbojka',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Skijanje',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Boks',sport_id_sekv.CURRVAL);
```

```
INSERT INTO sport  
VALUES(sport_id_sekv.NEXTVAL,'Hokej',sport_id_sekv.CURRVAL);
```

```
--Statistika
```

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

INSERT INTO statistika

VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);

```
INSERT INTO statistika
```

```
VALUES(statistika_id_sekv.nextval,statistika_id_sekv.CURRVAL,statistika_id_sekv.CURRVAL);
```

```
SELECT * FROM statistika;
```

```
--Statistika Arhiva
```

```
INSERT INTO statistika_arhiva
```

```
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'kos');
```

```
INSERT INTO statistika_arhiva
```

```
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 7,'gol');
```

```
INSERT INTO statistika_arhiva
```

```
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'3 kosa');
```

```
INSERT INTO statistika_arhiva
```

```
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'4 minute');
```

```
INSERT INTO statistika_arhiva
```

```
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 7,'5 puta');
```

```
INSERT INTO statistika_arhiva
```

```
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'trojka');
```

```
INSERT INTO statistika_arhiva
```

```
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'3 gola');
```

```
INSERT INTO statistika_arhiva  
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'kos');
```

```
INSERT INTO statistika_arhiva  
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 0,'prvi na listi');
```

```
INSERT INTO statistika_arhiva  
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'penal');
```

```
INSERT INTO statistika_arhiva  
VALUES(statistika_arhiva_id_sekv.NEXTVAL, 5,'12 kornera');
```

--Liga

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'La Liga',13,7);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Americka liga',13,7);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Italijanska liga',10,4);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Premier liga',10,5);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Engleska liga',10,4);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Turska liga',9,6);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Portugalska liga',12,5);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Ruska liga',10,4);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Francuska liga',10,4);
```

```
INSERT INTO Liga  
VALUES(liga_id_sekv.nextval,'Njemacka liga',10,4);
```

INDEKSI

--Indeksi

```
CREATE INDEX lice_ime_i_prezime_indeks  
ON lice(ime, prezime);
```

--indeks je kreiran zbog cinjenice da kada pretrazujemo podatke kako

--o klijentima tako i o zaposlenicima, uvijek pisemo ove dvije kolone skupa

```
CREATE INDEX drzave_drzave_naziv_indeks
```

```
ON Drzave(naziv);
```

--indeks kreiran iz razloga sto u upitima u where klauzi cesto pristupamo imenu drzave

--jer je nam je cesto za mnoge stvari bitno u kojoj se drzavi nalazi neki odjel, takodjer

--cesto koristena kolona u group by..

```
CREATE INDEX drzave_drzave_porez_indeks
```

```
ON Drzave(porez);
```

--Takodjer jedna od kolona kojoj se cesto pristupa u where klauzi zbog cinjenice da od poreza

--ovisi dobitak klijenta, pogodnosti koje moze dobiti i sl.

```
CREATE INDEX opklade_datum_opklade_indeks
```

```
ON Opklade(datum_opklade);
```

--Indeks kreiran zbog cinjenice da se kolona u koj se pristupa datumu cesto koristi

--u where klauzama koje se cesto naalze i u ugnjezdenim upitima

```
CREATE INDEX odjeli_sredstva_indeks
```

```
ON odjeli(sredstva);
```

--indeks kreiran jer se kolona sredstva cesto koristi u where klauzi

UPITI

*/*10 jednostavnih upita*/*

--1. Napisati upit koji će vratiti osnovne podatke o zaposlenicima i odjel u kojem rade

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime", l.spol "Spol", l.email "Email",  
       z.plata + z.plata*Nvl(z.dodatak_na_platu,0) "Plata(Uracunat dodatak)",  
       o.naziv "Odjel", o.sredstva "Zalihe odjela", o2.odjel_id "Odgovara odjelu: "  
FROM lice l, zaposleni z, odjeli o, odjeli o2  
WHERE l.lice_id=z.lice_id AND z.odjel_id=o.odjel_id AND o.nadredjeni_odjel_id=o2.odjel_id;
```

--2. Napisati upit koji prikazuje sve sifre opklada, te datum opklade koje su se odigrale u odjelu locoranom u Grckoj

```
SELECT o.opklada_id "Sifra", o.datum_opklade "Datum kladjenja"  
FROM opklade o, odjeli od, lokacije l, drzave d  
WHERE o.odjel_id=od.odjel_id AND od.lokacija_id=l.lokacija_id AND l.drzava_id=d.drzava_id  
       AND d.naziv='Grcka';
```

--3. Napisati upit koji će prikazati sve trenutne igrace i njihove timove

```
SELECT s.naziv, tim.naziv  
FROM sport_lica s "Igrac", sport_lica tim "Tim"  
WHERE s.tim_id=tim.sport_lica_id;
```

--4. Napisati upit koji će prikazati sve muske klijente koji imaju bilo kakvu pogodnost

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime", p.tip_pogodnosti "Tip pogodnosti"
FROM lice l, klijenti k, pogodnosti p
WHERE l.lice_id=k.lice_id AND k.pogodnost_id=p.pogodnost_id AND
      k.pogodnost_id<>NULL AND l.spol='musko';
```

--5. Napisati upit koji prikazuje sve drzave i njihov porez, sortirane po porezu

```
SELECT d.naziv "Drzava", d.porez
FROM drzave d
ORDER BY d.porez;
```

--6. Napisati upit koji će prikazati sifre svih klijenata koji su se kladili da ce FC Barcelona igrati nerijeseno,

-- te datum kada su se kladili

```
SELECT k.klijent_id "Sifra klijenta", o.datum_opklade "Datum opklade"
FROM klijenti k, opklade o, opklade_arhiva oa, sport_lica sl, sport s
WHERE k.opklada_id=o.opklada_id AND oa.opklada_id=o.opklada_id AND
      oa.sport_lica_id=sl.sport_lica_id AND sl.tip_sporta_id=s.sport_id
      AND sl.naziv='Barcelona' AND s.naziv='Nogomet';
```

--7. Prikazati ime i prezime klijenta koji su odigrali listic na danasnji datum

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime"
FROM lice l, klijenti k, opklade o
```



```
WHERE l.lice_id=k.lice_id AND k.opklada_id=o.opklada_id AND  
o.datum_opklade=SYSDATE;
```

--8. Ispisati sve zaposlene i platu, onih koji rade u odjelu 8 i obavljaju ulogu zastitara

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime", z.plata + z.plata*Nvl(z.dodatak_na_platu,0)  
"Plata(Uracunat dodatak)"  
FROM lice l, zaposleni z, funkcija_koju_obavlja f  
WHERE l.lice_id=z.lice_id AND f.funkcija_id=z.funkcija_id AND f.naziv='Zastitar' AND z.odjel_id=8;
```

--9. Izlistati sve trke koje su se održale u 2017.godini i koliko je učesnika bilo

```
SELECT t.naziv, t.broj_ucesnika  
FROM trke t  
WHERE To_Number(To_Char(t.datum_odrzavanja,'yyyy')) = 2017;
```

--10. Napisati upit koji će prikazati sve trke koje su se održale u Sarajevu

```
SELECT t.naziv "Utrka"  
FROM trke t, opklade_arhiva oa, opklade opk, odjeli o, lokacije l  
WHERE l.grad='Sarajevo' AND o.lokacija_id=l.lokacija_id AND o.odjel_id=opk.odjel_id AND  
opk.opklada_id=oa.opklada_id AND oa.trka_id=t.trka_id;
```

*/*5 upita sa grupnim funkcijama(2 sa having)*/*

--1.Napisati upit koji će prikazati prosjecan broj radnika po odjelima

```
SELECT o.naziv "Odjel",Avg(z.zaposlenik_id) "Prosjecan broj radnika"
```

```
FROM odjeli o, zaposleni z

WHERE o.odjel_id=z.odjel_id

GROUP BY o.naziv;
```

--2. Napisati upit koji će prikazati koliko puta su se klijenti kladili na nogomet u toku dana

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime", Count(o.opklada_id) "Broj opklada po klijentu"

FROM lice l, klijenti k, opklade o, opklade_arhiva oa, sport_lica sl, sport s

WHERE l.lice_id=k.lice_id AND k.opklada_id=o.opklada_id

      AND oa.opklada_id=o.opklada_id AND oa.sport_lica_id=sl.sport_lica_id AND

      sl.tip_sporta_id=s.sport_id

      AND o.datum_opklade=SYSDATE AND s.naziv='Nogomet'

GROUP BY l.ime || ' ' || l.prezime;
```

--3. Napisati upit koji će prikazati drzave u kojoj imaju najmanje 3 poslovnice

```
SELECT d.naziv "Drzava"

FROM drzave d, lokacije l, odjeli o

WHERE d.drzava_id=l.drzava_id AND l.lokacija_id=o.lokacija_id

GROUP BY d.naziv

HAVING Count(o.odjel_id)>=3;
```

--4. Napisati upit koji će prikazati broj opklada u 2017., te 2018.

-- godine, te ukupan broj opklada u ovim godinama.

```
SELECT Decode(Sum(Decode(To_Char(o.datum_opklade,'yyyy'),'2017',1,0)),

NULL,0,Sum(Decode(To_Char(o.datum_opklade,'yyyy'),'2017',1,0))) "2017",

      Decode(Sum(Decode(To_Char(o.datum_opklade,'yyyy'),'2018',1,0)), NULL,

0,Sum(Decode(To_Char(o.datum_opklade,'yyyy'),'2018',1,0))) "2018",
```

```
Decode(Sum(Decode(To_Char(o.datum_opklade,'yyyy'),'2017',1,'2018',1,0)),NULL,0,Sum(Decode(To_Char(o.datum_opklade,'yyyy'),'2017',1,'2018',1,0))) "Broj klijenata"
```

```
FROM opklade o;
```

--5. Napisati upit koji će ispisati zaposlenika sa maksimalnom platom ali samo ukoliko je ona veća od 2000

```
SELECT l.ime||' '||l.prezime "Ime i prezime"
```

```
FROM lice l, zaposleni z, odjeli o
```

```
WHERE l.lice_id=z.lice_id AND z.odjel_id=o.odjel_id
```

```
GROUP BY l.ime||' '||l.prezime
```

```
HAVING Max(z.plata)>2000;
```

*/*5 upita sa podupitima*/*

--1.Ispisati sve klijente kod kojih je tip pogodnosti jednak kao kod klijenta sa id=4

```
SELECT l.ime||' '||l.prezime "Ime i prezime", p.tip_pogodnosti "Tip pogodnosti"
```

```
FROM lice l, klijenti k, pogodnosti p
```

```
WHERE l.lice_id=k.lice_id AND k.pogodnost_id=p.pogodnost_id AND
```

```
p.tip_pogodnosti = (SELECT p2.tip_pogodnosti
```

```
FROM pogodnosti p2, klijenti k
```

```
WHERE p2.pogodnost_id=k.pogodnost_id AND k.klijent_id=4);
```

--2.Napisati upit koji će prikazati sve sportove koji se nalaze u bilo kojoj ligi čiji naziv sadrži slovo B

--i broj ucesnika joj je veci od ili jednak 6

```

SELECT s.naziv "Sport"

FROM sport s, liga l

WHERE s.liga_id=l.liga_id AND l.liga_id=(SELECT l1.liga_id

FROM liga l1

WHERE Upper(l1.naziv) LIKE '%B%' AND l1.broj_ucesnika>=6);

```

--3. Napisati upit koji će prikazati sve podatke o poslovnica kojima je nadređena poslovnica locirana u Bosni i Hercegovini

```

SELECT o.odjel_id "Sifra", o.naziv "Odjel", o.sredstva "Sredstva"

FROM odjeli o

WHERE o.nadredjeni_odjel_id=(SELECT oo.odjel_id

FROM odjeli oo, lokacije l, drzave d

WHERE oo.lokacija_id=l.lokacija_id AND l.drzava_id=d.drzava_id

AND d.naziv='Bosna i Hercegovina');

```

--4. Prikazati klijente čiji je iznos racuna veći od prosječnog iznosa racuna

-- svih klijenata koji su odigrali listić u poslovnici sa IDem 3

```

SELECT l.ime||' '||l.prezime "Ime i prezime"

FROM lice l, klijenti k, opklade o

WHERE l.lice_id=k.lice_id AND k.opklada_id=o.opklada_id AND

k.iznos_racuna>(SELECT Avg(k1.iznos_racuna)

FROM klijenti k1, opklade oo

WHERE k1.opklada_id=oo.opklada_id AND oo.odjel_id=3 AND oo.opklada_id<>null);

```

--5. Napisati upit koji vraća sve one poslovnice kod kojih su sredstva veća

-- od odjela sa maksimalnim sredstvima ali kad se uz sredstva uracuna porez

--2.Izlistati sifre opklada za timove koji imaju vise od 17 igraca

```
SELECT oa.opklada_id "Sifra opklade"
FROM opklade_arhiva oa, sport_lica sl
WHERE oa.sport_lica_id=sl.sport_lica_id
      AND sl.sport_lica_id=(SELECT sl1.tim_id
                            FROM sport_lica sl1
                            WHERE 17<=(SELECT Count(sl1.sport_lica_id)
                            FROM sport_lica sl2
                            WHERE sl2.tim_id=sl.sport_lica_id));
```

--3.Napisaiti upit koji će azurirati tabelu zaposlenih tako da za svakog zaposlenog koji radi u poslovnic

-- kojem se nadređena poslovница nalazi u Spaniji poveća dodatak na platu za 2% od sredstava u toj
poslovnici

```
UPDATE TABLE zaposlenici z
SET z.dodatak=(SELECT o.sredstva*0.02
FROM odjeli o
WHERE o.odjel_id=z.odjel_id
      AND o.nadređeni_odjel=(SELECT nad.odjel_id
                              FROM odjeli nad, lokacije l, drzave d
                              WHERE nad.lokacija_id=l.lokacija_id AND
                              l.lokacija_id=d.drzava_id AND d.naziv='Spanija'));
```

--4. Napisati upit koji će prikazati sve zaposlene koji rade u odjelima kojima su presušila sredstva i koji
se nalaze u

[illegible]

```

SELECT l.ime || ' ' || l.prezime "Ime i prezime"

FROM lice l, klijenti k

WHERE l.lice_id=k.lice_id AND

2>=(SELECT Count(o.opklada_id)

FROM opklade o, opklade_arhiva oa, sport_lica sl

WHERE o.opklada_id=oa.opklada_id AND oa.sport_lica_id=sl.sport_lica_id

AND sl.tim_id=(SELECT sl2.sport_lica_id

FROM sport_lica sl2

WHERE (SELECT Count(sl3.sport_lica_id)

FROM sport_lica sl3

WHERE sl3.tim_id=sl2.sport_lica_id) > (SELECT Max(Count(sl3.sport_lica_id))

FROM sport_lica sl3, sport_lica sl4

```

```
WHERE sl3.tim_id=sl4.sport_lica_id  
GROUP BY sl3.sport_lica_id));
```

*/*2 upita sa subtotalima*/*

--1.Modifikovati 5. zadatak pri koristenju grupnih fja da se prikazu dodatni redovi sa subtotalima

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime"  
FROM lice l, zaposleni z, odjeli o  
WHERE l.lice_id=z.lice_id AND z.odjel_id=o.odjel_id  
GROUP BY rollup(l.ime || ' ' || l.prezime)  
HAVING Max(z.plata)>2000;
```

--2.Napisati upit koji vraća sve one poslovnice kod kojih su sredstva manja

-- od odjela sa minimalni sredstvima grupisanim po porezu, prikazati i dodatne redove sa subtotalima

```
SELECT o.naziv  
FROM odjeli o  
WHERE o.sredstva<(SELECT Min(oo.sredstva)  
FROM odjeli oo, lokacije l, drzave d  
WHERE oo.lokacija_id=l.lokacija_id AND l.drzava_id=d.drzava_id  
GROUP BY d.porez)  
GROUP BY rollup(o.naziv);
```


*/*2 upita sa top N analizom*/*

--1. Napisati upit koji će prikazati top 5 sportova na koje su se

-- klijenti sa iznosom racuna manjim od 1200 kladili

```
SELECT s.naziv
```

```
FROM sport s
```

```
WHERE s.sport_id=(SELECT *
```

```
FROM (SELECT so.sport_id
```

```
FROM sport so, sport_lica sl,
```

```
opklade_arhiva oa, opklade o, klijenti k
```

```
WHERE so.sport_id=sl.tip_sporta_id AND sl.sport_lica_id=oa.sport_lica_id
```

```
AND o.opklada_id=k.opklada_id AND k.iznos_racuna<1200
```

```
GROUP BY so.sport_id
```

```
ORDER BY Count(o.opklada_id))
```

```
WHERE ROWNUM<=5);
```

--2.Prikaza top 3 klijenta koji su izvršili najviše opklada u toku danasnjeg dana

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime"
```

```
FROM lice l, klijenti k
```

```
WHERE l.lice_id=k.lice_id
```

```
AND k.klijent_id=(SELECT *
```

```
FROM (SELECT k1.klijent_id
```

```
FROM klijenti k1, opklade o
```

```
WHERE k1.opklada_id=o.opklada_id AND o.opklada_id<>NULL
```

```
AND o.datum_opklade=sysdate
```

```
GROUP BY k1.klijent_id
```

```
ORDER BY Count(o.opklada_id))  
WHERE ROWNUM<=3);
```

*/*1 sa union, set*/*

--1. Napisati upit koji ce prikazati klijenata cije prezime pocinje sa M a imaju iznos racuna veci od 300,
-- i klijenata cije ime pocinje sa D a imaju iznos racuna manji od 1000

```
SELECT l.ime || ' ' || l.prezime "Ime i prezime"  
FROM lice l, klijenti k  
WHERE l.lice_id=k.lice_id AND Upper(l.prezime) like 'M%' and  
      k.iznos_racuna > 300  
UNION  
SELECT l.ime || ' ' || l.prezime  
FROM lice l, klijenti k  
WHERE l.lice_id=k.lice_id AND Upper(l.ime) like 'D%' and  
      k.iznos_racuna < 1000;
```

FUNKCIJE

--Funkcije

--1.Funkcija koja provjerava da li je dati odjel ostao bez sredstava,
-- ako jeste vraca 1, u suprotnom vraca 0

```
CREATE OR REPLACE FUNCTION SredstvaOdjela(odjel_id in odjeli.odjel_id%TYPE)
```

RETURN NUMBER IS

propao NUMBER:=1;

BEGIN

SELECT Decode(o.sredstva,0,0,1)

INTO propao

FROM odjeli o

WHERE o.odjel_id=odjel_id;

RETURN(propao);

END;

--2. Kreirati funkciju koja ce da vrati koliko klijenata je odigralo listic na danasnji dan

CREATE OR REPLACE FUNCTION BrojKladjenjaUDanu

RETURN NUMBER IS

broj NUMBER:=0;

BEGIN

SELECT count(k.klijent_id)

INTO broj

FROM klijenti k,opklade o

WHERE k.opklada_id=o.opklada_id AND

o.datum_opklade=SYSDATE AND

o.opklada_id<>NULL;

RETURN(broj);

END;

--3. Napraviti funkciju koja vraća broj zaposlenih koji rade u glavnoj centrali

```
CREATE OR REPLACE FUNCTION BrojZaposlenihCentrala
```

```
RETURN NUMBER IS
```

```
    brojZaposlenih NUMBER:=0;
```

```
BEGIN
```

```
SELECT count(z.zaposlenik_id)
```

```
INTO brojZaposlenih
```

```
FROM zaposleni z, odjeli o, lokacije l, drzave d,odjeli nadredjeni, odjeli centrala
```

```
WHERE d.naziv='Meksiko' AND d.drzava_id=l.drzava_id AND
```

```
    l.lokacija_id=centrala.lokacija_id AND
```

```
    z.odjel_id=o.odjel_id AND
```

```
    nadredjeni.odjel_id=o.nadredjeni_odjel_id AND
```

```
    nadredjeni.nadredjeni_odjel_id=centrala.odjel_id;
```

```
RETURN(brojZaposlenih);
```

```
END;
```

--4.Napisati funkciju koja će vratiti klijenta sa najviše kladjenja u toku dana u nekom odjelu

```
CREATE OR REPLACE FUNCTION NajviseKladjenjaUTokuDana(odjeli_id in odjeli.odjel_id%TYPE)
```

```
RETURN NUMBER IS
```

```
    broj NUMBER:=0;
```

```

BEGIN

SELECT count(oa.opklada_id)

INTO broj

FROM klijenti k,opklade o, opklade_arhiva oa, odjeli oo

WHERE k.opklada_id=o.opklada_id AND oa.opklada_id=o.opklada_id AND

      o.odjel_id=oo.odjel_id AND oo.odjel_id=odjeli_id AND

      o.datum_opklade=SYSDATE AND

      o.opklada_id<>NULL

GROUP BY o.odjel_id;

RETURN(broj);

END;

```

--5.Napisati funkciju koja prima iznos dobitka, i id opklade a

-- vraca koliko ce novca biti isplaceno u odnosu na porez drzave

```

CREATE OR REPLACE FUNCTION Isplata(iznos NUMBER, odjeli_id in odjeli.odjel_id%TYPE)

RETURN NUMBER IS

      dobitak NUMBER:=0;

BEGIN

SELECT iznos-d.porez*iznos

INTO dobitak

FROM opklade o,odjeli oo, lokacije l, drzave d

WHERE o.odjel_id=oo.odjel_id AND oo.lokacija_id=l.lokacija_id

      AND l.drzava_id=d.drzava_id AND o.odjel_id=odjeli_id

GROUP BY d.drzava_id;

```

```
RETURN(dobitak);
```

```
END;
```

--6. Napisati funkciju koja vraca sport na koji se izvrsilo najvise opklada

-- u drzavi proslijedjenoj kao parametar

```
CREATE OR REPLACE FUNCTION KladjenjeSport(nazivi in drzave.drzava_id%TYPE)
```

```
RETURN VARCHAR2 IS
```

```
    sport_ime VARCHAR2(50);
```

```
BEGIN
```

```
SELECT ss.naziv
```

```
INTO sport_ime
```

```
FROM sport ss
```

```
WHERE ss.sport_id=(SELECT *
```

```
    FROM (SELECT s.sport_id
```

```
        FROM opklade o,odjeli oo, lokacije l,drzave d,
```

```
        opklade_arhiva oa, sport_lica sl, sport s
```

```
        WHERE d.naziv=nazivi AND o.odjel_id=oo.odjel_id AND
```

```
        oo.lokacija_id=l.lokacija_id AND l.drzava_id=d.drzava_id AND
```

```
        o.opklada_id=oa.opklada_id AND oa.sport_lica_id=sl.sport_lica_id AND
```

```
        sl.tip_sporta_id=s.sport_id
```

```
        GROUP BY s.sport_id
```

```
        ORDER BY Count(oa.opklada_id))
```

```
    WHERE ROWNUM<=1);
```

```
RETURN(sport_ime);
```

```
END;
```

--7. Napisati funkciju koja ce vratiti koliko ima drzava u kojoj imaju najvise 4 poslovnice

```
CREATE OR REPLACE FUNCTION DrzaveSa4Poslovnice
RETURN VARCHAR2 IS
    broj_drzava VARCHAR2(50);
BEGIN
    SELECT Count(DISTINCT d.naziv)
    INTO broj_drzava
    FROM drzave d, lokacije l, odjeli o
    WHERE d.drzava_id=l.drzava_id AND l.lokacija_id=o.lokacija_id
    GROUP BY d.naziv
    HAVING Count(o.odjel_id)<=4;
    RETURN(broj_drzava);
END;
```

--8. Napisati funkciju koja vraca 1 ako u ligi La Liga unesen FC Real Madrid, u

-- suprotnom vraca 0

```
CREATE OR REPLACE FUNCTION LaLiga
RETURN NUMBER IS
    broj NUMBER:=0;
BEGIN
    SELECT Decode(sl.naziv,'Real Madrid',1,0)
    INTO broj
    FROM sport_lica sl, sport ss, liga l
    WHERE sl.tip_sporta_id=ss.sport_id AND ss.naziv='Nogomet' AND
```

```
ss.liga_id=l.liga_id AND l.naziv='La Liga';  
  
RETURN(broj);  
  
END;
```

--9.Napisati funkciju koja vraca id tima koji ima manje od 15 igraca

```
CREATE OR REPLACE FUNCTION TimSaManjeOd15Igraca  
  
RETURN sport_lica.sport_lica_id%type IS  
  
    id sport_lica.sport_lica_id%type;  
  
BEGIN  
  
SELECT sl.sport_lica_id  
  
INTO id  
  
FROM sport_lica sl  
  
WHERE sl.sport_lica_id=(SELECT sl1.tim_id  
  
                        FROM sport_lica sl1  
  
                        WHERE 17<=(SELECT Count(sl1.sport_lica_id)  
  
                                FROM sport_lica sl2  
  
                                WHERE sl2.tim_id=sl.sport_lica_id));  
  
RETURN(id);  
  
END;
```

--10. Napisati funkciju koja vraca broj odjela kod kojih su sredstva veca
-- od prosjenog broja sredstava svih odjela ne racunajuci taj odjel


```
CREATE OR REPLACE FUNCTION OdjeliSredstva
```

```
RETURN NUMBER IS
```

```
    broj NUMBER:=0;
```

```
BEGIN
```

```
SELECT Count(distinct o.naziv)
```

```
INTO broj
```

```
FROM odjeli o
```

```
WHERE o.sredstva>(SELECT avg(oo.sredstva)
```

```
    FROM odjeli oo
```

```
    WHERE oo.odjel_id<>o.odjel_id);
```

```
RETURN(broj);
```

```
END;
```

PROCEDURE

```
--Procedure
```

```
--1.Napisati proceduru za unos novog klijenta
```

```
CREATE OR REPLACE PROCEDURE KreirajKlijenta( iznos_racunap in klijenti.iznos_racuna%type,
```

```
    pogodnost_idp in klijenti.pogodnost_id%type,
```

```
    opklada_idp in klijenti.opklada_id%type,
```

```
    lice_idp in klijenti.lice_id%type) IS
```

```
BEGIN
```

```
INSERT INTO klijenti
```

```
VALUES (klijenti_id_sekv.nextval,iznos_racunap,pogodnost_idp,opklada_idp, lice_idp);  
  
COMMIT;  
  
END;
```

--2. Napisati proceduru koja uklanja brise zaposlenika sa id prenesenim kao parametrom
-- samo ukoliko odjel u kojem radi vise nema sredstava

```
CREATE OR REPLACE PROCEDURE BrisiZaposlenika(zaposlenik IN zaposleni.zaposlenik_id%type) IS
```

```
  
    sredstva1 NUMBER:=0;  
    postojilizaposlenik NUMBER:=0;  
  
BEGIN  
  
    SELECT Decode(zaposlenik_id,zaposlenik,1,0)  
  
    INTO postojilizaposlenik  
  
    FROM zaposleni;  
  
  
    SELECT o.sredstva  
  
    INTO sredstva1  
  
    FROM odjeli o, zaposleni z  
  
    WHERE o.odjel_id=z.odjel_id AND z.zaposlenik_id=zaposlenik;  
  
    IF postojilizaposlenik=0 THEN  
  
        Raise_Application_Error(-20501,'Ne postoji dati zaposlenik');  
  
    ELSIF sredstva1=0 THEN  
  
        DELETE FROM zaposleni  
  
        WHERE zaposlenik_id=zaposlenik;
```

```
ELSIF sredstva1<>0 THEN
```

```
    Raise_Application_Error(-20502, 'Sredstva odjela nisu jednaka 0');
```

```
END IF;
```

```
END;
```

--3. Kreirati proceduru za pohranjivanje iznosa opklade na racun klijenta

-- pri tome uracunati porez

```
CREATE OR REPLACE PROCEDURE IsplatiKlijenta (iznos IN NUMBER,  
                                             id IN klijenti.klijent_id%type) IS
```

```
    iznos_uplate NUMBER:=0;
```

```
BEGIN
```

```
    SELECT iznos-d.porez*iznos
```

```
    INTO iznos_uplate
```

```
    FROM drzave d, lokacije l,
```

```
         odjeli oo, opklade o, klijenti k
```

```
    WHERE k.opklada_id=o.opklada_id AND d.drzava_id=l.drzava_id AND
```

```
          o.odjel_id=oo.odjel_id AND oo.lokacija_id=l.lokacija_id
```

```
          AND k.klijent_id=id;
```

```
    UPDATE klijenti
```

```
    SET iznos_racuna=iznos_racuna+iznos_uplate
```

```
    WHERE klijent_id=id;
```

END;

--4. Kreirati proceduru za promjenu pogodnosti klijenta

```
CREATE OR REPLACE PROCEDURE PromijeniPogodnost(nova_pogodnost in
pogodnosti.tip_pogodnosti%TYPE,

                id IN klijenti.klijent_id%type) AS

    stara_pogodnost pogodnosti.tip_pogodnosti%type;

BEGIN

    SELECT p.tip_pogodnosti

    INTO stara_pogodnost

    FROM pogodnosti p, klijenti k

    WHERE p.pogodnost_id=k.pogodnost_id AND k.klijent_id=id;

    IF stara_pogodnost = nova_pogodnost THEN

        raise_application_error(-20503, 'Nova pogodnost je ista kao stara!');

    ELSE

        UPDATE pogodnosti p

        SET p.tip_pogodnosti = nova_pogodnost

        WHERE p.pogodnost_id=(SELECT p1.pogodnost_id

                                FROM pogodnosti p1, klijenti k

                                WHERE p1.pogodnost_id=k.pogodnost_id AND k.klijent_id=id);

    END IF;

END;
```

--5. Kreirati proceduru koja omogucava klijentu da promijeni strategiju

-- u opkladi poslanoj kao parametar

```

CREATE OR REPLACE PROCEDURE PromijeniStrategiju(nova_strategija in
strategija.tip_strategije%TYPE,

                id IN klijenti.klijent_id%type) AS

    stara_strategija strategija.tip_strategije%TYPE;

BEGIN

    SELECT p.tip_strategije

    INTO stara_strategija

    FROM strategija p, klijenti k, opklade o, opklade_arhiva oa

    WHERE p.strategija_id=oa.strategija_id AND oa.opklada_id=o.opklada_id

        AND o.opklada_id=k.opklada_id AND k.klijent_id=id;

    IF stara_strategija = nova_strategija THEN

        raise_application_error(-20504, 'Nova strategija je ista kao stara!');

    ELSE

        UPDATE strategija p

        SET p.tip_strategije = nova_strategija

        WHERE p.strategija_id=(SELECT p1.strategija_id

                                FROM strategija p1, klijenti k, opklade o, opklade_arhiva oa

                                WHERE p1.strategija_id=oa.strategija_id AND oa.opklada_id=o.opklada_id

                                    AND o.opklada_id=k.opklada_id AND k.klijent_id=id);

    END IF;

END;

```

--6.Kreirati proceduru za stavljanje datuma održavanje neke trke na null ukoliko je ona već održana

```

CREATE OR REPLACE PROCEDURE PromijeniDatum AS

```

```
datum DATE;
```

```
BEGIN
```

```
SELECT t.datum_odrzavanja
```

```
INTO datum
```

```
FROM trke t;
```

```
IF datum>sysdate THEN
```

```
raise_application_error(-20505, 'Trka se tek treba odrzati');
```

```
ELSE
```

```
UPDATE trke t
```

```
SET t.datum_odrzavanja=null
```

```
WHERE t.datum_odrzavanja<SYSDATE;
```

```
END IF;
```

```
END;
```

--7.Kreirati proceduru koja ce obracunavati platu zaposlenika zajedno sa dodatkom na platu

-- u zavisnosti od poreza drzave

```
CREATE OR REPLACE PROCEDURE ObracunajPlatu( plata out NUMBER,  
                                             zaposlenik in zaposleni.zaposlenik_id%type) AS
```

```
BEGIN
```

```
SELECT (plata+plata*dodatak_na_platu)*(1-d.porez)
```

```
INTO plata
```

```
FROM zaposleni z, odjeli o, lokacije l, drzave d
```

```
WHERE z.zaposlenik_id=zaposlenik AND z.odjel_id=o.odjel_id AND
```

```
o.lokacija_id=l.lokacija_id AND l.drzava_id=d.drzava_id;
```

```
END;
```

--8. Napisati proceduru koja će prikazati maksimalan broj radnika u nekom odjelu

```
CREATE OR REPLACE PROCEDURE ProsjecanBrojRadnika(broj_radnika out NUMBER) AS
```

```
BEGIN
```

```
    SELECT Avg(z.zaposlenik_id)
```

```
    INTO broj_radnika
```

```
    FROM odjeli o, zaposleni z
```

```
    WHERE o.odjel_id=z.odjel_id
```

```
    GROUP BY o.naziv;
```

```
END;
```

--9. Napisati proceduru koja mijenja funkciju radnika

```
CREATE OR REPLACE PROCEDURE PromijeniFunkcijuRadnika( nova_funkcija IN  
funkcija_koju_obavlja.naziv%TYPE,
```

```
id IN zaposleni.zaposlenik_id%type) IS
```

```
stara_funkcija funkcija_koju_obavlja.naziv%TYPE;
```

```
BEGIN
```

```
    SELECT f.naziv
```



```
WHERE oo.lokacija_id=l.lokacija_id AND l.drzava_id =d.drzava_id  
AND d.naziv='Meksiko');
```

```
END;
```

TRIGERI

```
--Trigeri
```

```
--1.Trigger za sredstva u odjelu koja ne smiju biti = 0
```

```
CREATE OR REPLACE TRIGGER OdjeliSredstvaTriger
```

```
BEFORE INSERT OR UPDATE ON odjeli
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    sredstva_validacija number;
```

```
BEGIN
```

```
    SELECT Decode(o.sredstva, 0, 0, 1)
```

```
    INTO sredstva_validacija
```

```
    FROM odjeli o;
```

```
    IF sredstva_validacija = 0 THEN
```

```
        raise_application_error (-20507, 'Dati odjel nema vise dovoljno sredstava da bi funkcionisao!');
```

```
    END IF;
```

```
END;
```

```
--2. Triger za platu zaposlenika, ne smije biti < 0
```

```
CREATE OR REPLACE TRIGGER PlataZaposlenikaTriger
```

```
BEFORE INSERT OR UPDATE ON zaposlenici
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF :new.plata < 0 THEN
```

```
    raise_application_error (-20510, 'Plata ne smije biti manja od 0');
```

```
END IF;
```

```
END;
```

--3. trigger ako se pokušaju unijeti nelegalna sredstva u odjele

```
CREATE OR REPLACE TRIGGER NelegalnaSredstvaTriger
```

```
BEFORE INSERT OR UPDATE ON odjeli
```

```
FOR EACH ROW
```

```
DECLARE
```

```
BEGIN
```

```
IF :new.sredstva <= 0 THEN
```

```
    raise_application_error (-20511, 'Odjel ne može imati sredstva manja od ili jednaka 0');
```

```
END IF;
```

```
END;
```

--4. trigger za provjeru da li je strategija ispravno unijeta

```
CREATE OR REPLACE TRIGGER StrategijIspravnaTriger
```

```
BEFORE INSERT OR UPDATE ON strategija
FOR EACH ROW
DECLARE
BEGIN
    IF Lower(:new.tip_strategije) NOT IN ('pobjeda', 'poraz', 'nerijesen') THEN
        raise_application_error (-20512, 'Neispravan tip strategije');
    END IF;
END;
```

--5. Provjera da li je spol unesen ispravno

```
CREATE OR REPLACE TRIGGER IspravanSpol
BEFORE INSERT OR UPDATE ON lice
FOR EACH ROW
DECLARE
BEGIN
    IF Lower(:new.spol) NOT IN ('musko', 'zensko') THEN
        raise_application_error (-20513, 'Neispravan spol!');
    END IF;
END;
```

--6. Provjera da li je datum vrsenja opklade izvršen prije danasnjeg dana

```
CREATE OR REPLACE TRIGGER IspravanDatumKladjenja
BEFORE INSERT OR UPDATE ON opklade
```

```
FOR EACH ROW
```

```
DECLARE
```

```
BEGIN
```

```
IF :new.datum_opklade>sysdate THEN
```

```
    raise_application_error (-20513, 'Neisparan datum kladjenja!');
```

```
END IF;
```

```
END;
```

--7. Provjera da li je datum održavanja utrke poslije danasnjeg dana

```
CREATE OR REPLACE TRIGGER IspravanDatumTrke
```

```
BEFORE INSERT OR UPDATE ON trke
```

```
FOR EACH ROW
```

```
DECLARE
```

```
BEGIN
```

```
IF :new.datum_odrzavanja<sysdate THEN
```

```
    raise_application_error (-20513, 'Trka vec odrzana');
```

```
END IF;
```

```
END;
```

--8. Napisati trigger koji ce za nadredjenu poslovnicu, nadredjene poslovnice staviti onu koja se nalazi u meksiku

```
CREATE OR REPLACE TRIGGER NadredjenaPoslovnica
```

```
BEFORE INSERT OR UPDATE ON odjeli
```

```
FOR EACH ROW
```

```

DECLARE

BEGIN

UPDATE odjeli o

SET o.nadredjeni_odjel_id=(SELECT oo.odjel_id

                        FROM odjeli oo, lokacije l, drzave d

                        WHERE oo.lokacija_id=l.lokacija_id AND l.drzava_id=d.drzava_id

                        AND d.naziv='Meksiko')

WHERE o.odjel_id=(SELECT oo.nadredjeni_odjel_id

                  FROM odjeli oo);

END;

```

--9. Kreirati triger koji ce omogućiti automatsko dodjeljivanje id tabeli sport

```

CREATE OR REPLACE TRIGGER DodijeliIDSport

BEFORE INSERT OR UPDATE ON sport

FOR EACH ROW

DECLARE

BEGIN

INSERT INTO sport

VALUES (sport_id_sekv.nextval,:new.naziv,:new.liga_id);

END;

```

--10.Kreirati triger nad vašom tabelom trke koji se za promjenu datuma_odrzavanja trke

```
-- stari datum unijeti u novu tabelu trke_arhiva, koja ce sadrzavati pored tog datum
-- id zaposlenika koji je azurirao datum, te id trke
```

```
CREATE TABLE trke_arhiva(stari_datum DATE,

                           zaposlenik_id INTEGER,

                           trka_id INTEGER REFERENCES trke(trka_id));
```

```
CREATE OR REPLACE TRIGGER NadredjenaPoslovnicaTriger

BEFORE INSERT OR UPDATE ON trke

FOR EACH ROW

DECLARE

BEGIN

    INSERT INTO trke_arhiva

    SELECT :old.datum_odrzavanja, z.zaposlenik_id,:old.trka_id

    FROM trke t, opklade_arhiva oa, opklade o, odjeli oo, zaposleni z

    WHERE t.trka_id=oa.trka_id AND oa.opklada_id=o.opklada_id AND

           oo.odjel_id=z.odjel_id;

END;
```

PL/SQL SKRIPTA

```
--Skripta
```

```
/*Skripta služi da ukoliko dodje do toga da neki odjel ostane bez sredstava,
dostavlja upozorenje da je odjel ostao bez sredstava*/
```

```
CREATE OR REPLACE FUNCTION Upozorenje(poruka IN VARCHAR2)
```

```
RETURN VARCHAR2 IS
```

```
    povratna_poruka VARCHAR2(50):=poruka;
```

```
BEGIN
```

```
    RETURN(povratna_poruka);
```

```
END;
```

```
CREATE OR REPLACE PROCEDURE Upozorenje_Poruka(poruka IN VARCHAR2) IS
```

```
    povratna_poruka VARCHAR2(50):=poruka;
```

```
BEGIN
```

```
    SELECT Upozorenje(poruka)
```

```
    INTO povratna_poruka
```

```
    FROM odjeli;
```

```
END;
```

```
DECLARE
```

```
    stanje_racuna odjeli.sredstva%TYPE;
```

```
CURSOR odjeli_i is
```

```
    SELECT odjel_id
```

```
    FROM odjeli;
```

```
BEGIN
```

```
    FOR trenutni_odjel IN odjeli_i LOOP
```

```
        SELECT o.sredstva
```

```
INTO stanje_racuna  
FROM odjeli o  
WHERE o.odjel_id=trenutni_odjel.odjel_id;  
  
IF stanje_racuna<=0 THEN  
    Upozorenje_Poruka('Odjel vise nema sredstava');  
END IF;  
END LOOP;  
END;
```