

Elektrotehnički fakultet
Univerzitet Sarajevo

Projekat
Osnove Baza Podataka

Student: Vedad Fejzagić
Broj indeksa: 17336
Asistent: Haris Hasić

Projekat je rađen u JetBrains DataGrip programu. Postavljeni upiti nisu nužno istim redoslijedom izvršeni.

- a) ERD je u prilogu
- b) Kreiranje tabela, sekvenci i ograničenja

```
CREATE TABLE Poslovnice (  
  id INTEGER PRIMARY KEY NOT NULL,  
  id_nadredjene_poslovnice INTEGER,  
  tip_poslovnice VARCHAR(255) NOT NULL ,  
  sjediste VARCHAR(255) NOT NULL ,  
  raspoloziva_sredstva FLOAT NOT NULL ,  
  porez_na_dobitak FLOAT NOT NULL  
);
```

```
ALTER TABLE Poslovnice ADD CONSTRAINT strani_kljuc_poslovnice FOREIGN  
KEY(id_nadredjene_poslovnice) REFERENCES Poslovnice(id);
```

```
CREATE TABLE Klijenti (  
  id   INTEGER PRIMARY KEY NOT NULL,  
  ime  VARCHAR(255),  
  prezime VARCHAR(255),  
  dostupna_sredstva FLOAT,  
  povecana_stopa_dobitka FLOAT,  
  smanjenje_poreza FLOAT,  
  poslovnica_id INTEGER NOT NULL  
);
```

```
ALTER TABLE Klijenti ADD CONSTRAINT sk_poslovnica FOREIGN KEY(poslovnica_id)  
REFERENCES poslovnice(id);
```

```
CREATE TABLE Odjeli(  
  id INTEGER PRIMARY KEY NOT NULL,  
  naziv_odjela VARCHAR(255) NOT NULL ,  
  poslovnica_id INTEGER NOT NULL  
);
```

```
ALTER TABLE Odjeli ADD CONSTRAINT sk_poslovnica_id FOREIGN KEY(poslovnica_id)  
REFERENCES Poslovnice(id);
```

```
CREATE TABLE Poslovi(  
  id INTEGER PRIMARY KEY NOT NULL ,  
  naziv_posla VARCHAR(255) UNIQUE NOT NULL ,  
  plata FLOAT NOT NULL  
);
```

```
CREATE TABLE Zaposleni(  
  id INTEGER PRIMARY KEY NOT NULL,  
  ime VARCHAR(255) NOT NULL,  
  prezime VARCHAR(255) NOT NULL,  
  odjel_id INTEGER NOT NULL,  
  posao_id INTEGER NOT NULL,  
  datum_zaposlenja date NOT NULL  
);
```

```
ALTER TABLE Zaposleni add CONSTRAINT sk_zaposleni_odjel FOREIGN KEY(odjel_id)  
REFERENCES Odjeli(id);  
ALTER TABLE Zaposleni add CONSTRAINT sk_zaposleni_posao FOREIGN KEY(posao_id)  
REFERENCES Poslovi(id);
```

```
CREATE TABLE Opkladelgraci(  
  id INTEGER PRIMARY KEY NOT NULL,  
  uplaceno_novca FLOAT NOT NULL,  
  datum_uplate DATE NOT NULL,  
  odabrani_igrac_id INTEGER NOT NULL,  
  odabrana_utakmica_id INTEGER NOT NULL,  
  broj_poena INTEGER NOT NULL,  
  poslovnica_id INTEGER NOT NULL,  
  klijent_id INTEGER NOT NULL  
);
```

```
ALTER TABLE Opkladelgraci ADD CONSTRAINT fk_poslovnica_id FOREIGN  
KEY(poslovnica_id) REFERENCES Poslovnice(id);  
ALTER TABLE Opkladelgraci ADD CONSTRAINT fk_klijent_id FOREIGN KEY (klijent_id)  
REFERENCES Klijenti(id);  
ALTER TABLE Opkladelgraci ADD CONSTRAINT fk_odabrani_igrac FOREIGN KEY  
(odabrani_igrac_id) REFERENCES Igraci(id);  
ALTER TABLE Opkladelgraci ADD CONSTRAINT fk_odabrana_utakmica FOREIGN KEY  
(odabrana_utakmica_id) REFERENCES Utakmice(id);
```

```
CREATE TABLE OpkladeUtakmice(  
  id INTEGER PRIMARY KEY NOT NULL ,  
  uplaceno_novca FLOAT NOT NULL ,  
  datum_update DATE NOT NULL ,  
  odabrana_utakmica_id INTEGER NOT NULL ,  
  poeni_domaci INTEGER NOT NULL ,  
  poeni_gost INTEGER NOT NULL ,  
  poslovnica_id INTEGER NOT NULL ,  
  klijent_id INTEGER NOT NULL  
);
```

```
ALTER TABLE OpkladeUtakmice ADD CONSTRAINT fk_poslovnica2_id FOREIGN  
KEY(poslovnica_id) REFERENCES Poslovnice(id);  
ALTER TABLE OpkladeUtakmice ADD CONSTRAINT fk_klijent2_id FOREIGN KEY (klijent_id)  
REFERENCES Klijenti(id);  
ALTER TABLE OpkladeUtakmice ADD CONSTRAINT fk_odabrana2_utakmica FOREIGN KEY  
(odabrana_utakmica_id) REFERENCES Utakmice(id);
```

```
CREATE TABLE OpkladeMecevi(  
  id INTEGER PRIMARY KEY NOT NULL ,  
  uplaceno_novca FLOAT NOT NULL ,  
  datum_update DATE NOT NULL ,  
  odabrani_mec_id INTEGER NOT NULL ,  
  poeni_domaci INTEGER NOT NULL ,  
  poeni_gost INTEGER NOT NULL ,  
  poslovnica_id INTEGER NOT NULL ,  
  klijent_id INTEGER NOT NULL  
);
```

```
ALTER TABLE OpkladeMecevi ADD CONSTRAINT fk_poslovnica3_id FOREIGN  
KEY(poslovnica_id) REFERENCES Poslovnice(id);  
ALTER TABLE OpkladeMecevi ADD CONSTRAINT fk_klijent3_id FOREIGN KEY (klijent_id)  
REFERENCES Klijenti(id);  
ALTER TABLE OpkladeMecevi ADD CONSTRAINT fk_odabrani_mec FOREIGN KEY  
(odabrani_mec_id) REFERENCES Mecevi(id);
```

```
CREATE TABLE IgraciEfikasnost(  
  id INTEGER PRIMARY KEY NOT NULL ,  
  igrac_id INTEGER NOT NULL ,  
  utakmica_id INTEGER NOT NULL ,  
  broj_poena INTEGER NOT NULL
```

);

ALTER TABLE IgraciEfikasnost ADD CONSTRAINT fk_igrac_id FOREIGN KEY(igrac_id)
REFERENCES Igraci(id);

ALTER TABLE IgraciEfikasnost ADD CONSTRAINT fk_utakmica_id FOREIGN
KEY(utakmica_id) REFERENCES Utakmice(id);

CREATE TABLE Igraci(
id INTEGER PRIMARY KEY NOT NULL ,
ime VARCHAR(100) NOT NULL ,
prezime VARCHAR(100) NOT NULL ,
poeni_ove_sezone INTEGER NOT NULL ,
tim_id INTEGER
);

ALTER TABLE Igraci ADD CONSTRAINT fk_tim_id FOREIGN KEY(tim_id)
REFERENCES Timovi(id);

CREATE TABLE Mecevi(
id INTEGER PRIMARY KEY NOT NULL ,
igrac_domaci_id INTEGER NOT NULL ,
igrac_gost_id INTEGER NOT NULL ,
poeni_domaci INTEGER NOT NULL ,
poeni_gost INTEGER NOT NULL
);

ALTER TABLE Mecevi ADD CONSTRAINT fk_igrac_domaci_id FOREIGN
KEY(igrac_domaci_id) REFERENCES Igraci(id);
ALTER TABLE Mecevi ADD CONSTRAINT fk_igrac_gost_id FOREIGN KEY(igrac_gost_id)
REFERENCES Igraci(id);

CREATE TABLE Timovi(
id INTEGER PRIMARY KEY NOT NULL ,
naziv VARCHAR(100) NOT NULL ,
poeni_ove_sezone VARCHAR(100) NOT NULL ,
trener VARCHAR(100) NOT NULL ,
liga_id INTEGER NOT NULL
);

```
ALTER TABLE Timovi ADD CONSTRAINT fk_liga_id FOREIGN KEY(liga_id)
REFERENCES Lige(id);
ALTER TABLE Timovi ADD CONSTRAINT uniq_naziv UNIQUE(naziv);
```

```
CREATE TABLE Utakmice(
  id INTEGER PRIMARY KEY NOT NULL ,
  tim_domaci_id INTEGER NOT NULL ,
  tim_gost_id INTEGER NOT NULL ,
  poeni_domacin INTEGER NOT NULL ,
  poeni_gost INTEGER NOT NULL
);
```

```
ALTER TABLE Utakmice ADD CONSTRAINT fk_domacin_id FOREIGN
KEY(tim_domaci_id) REFERENCES Timovi(id);
ALTER TABLE Utakmice ADD CONSTRAINT fk_gost_id FOREIGN KEY(tim_gost_id)
REFERENCES Timovi(id);
```

```
CREATE TABLE Lige(
  id INTEGER PRIMARY KEY NOT NULL ,
  ime VARCHAR(100) NOT NULL ,
  sport VARCHAR(100) NOT NULL
);
```

c) Popunjavanje tabela

Koristit ćemo sekvencu inc koju ćemo ‘drop’ i ‘create’ iznova svaki put kada unosimo u novu tabelu. Nećemo pisati create i drop upite nakon svakog ubacivanja u tabelu, već je dovoljno na početku.

```
CREATE SEQUENCE inc INCREMENT BY 1 START WITH 0 MINVALUE 0 NOCACHE  
NOCYCLE;
```

```
DROP SEQUENCE inc;
```

Insert upiti:

```
INSERT INTO Poslovnice(id, tip_poslovnice, sjediste, raspoloziva_sredstva, porez_na_dobitak)  
VALUES
```

```
(inc.nextval, 'glavna', 'Ciudad de Juarez', 999999999.15, 0.025);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```
(inc.nextval, 0, 'regionalna centralna', 'Kosovo', 554687.12, 0.03);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```
(inc.nextval, 0, 'regionalna centralna', 'Berlin', 9455123.18, 0.11);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```
(inc.nextval, 0, 'regionalna centralna', 'Cuba', 10465123.25, 0.22);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```
(inc.nextval, 2, 'obicna', 'Amsterdam', 645123.18, 0.11);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```
(inc.nextval, 2, 'obicna', 'Paris', 765123.22, 0.11);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```
(inc.nextval, 1, 'obicna', 'Sarajevo', 210135.22, 0.03);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```
(inc.nextval, 1, 'obicna', 'Beograd', 546123.22, 0.03);
```

```
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,  
raspoloziva_sredstva, porez_na_dobitak) VALUES
```

```

    (inc.nextval, 2, 'obicna', 'Paris', 765123.22, 0.11);
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,
raspoloziva_sredstva, porez_na_dobitak) VALUES
    (inc.nextval, 2, 'obicna', 'London', 664123.22, 0.11);
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,
raspoloziva_sredstva, porez_na_dobitak) VALUES
    (inc.nextval, 3, 'obicna', 'Kingston', 312123.22, 0.22);
INSERT INTO Poslovnice(id, id_nadredjene_poslovnice, tip_poslovnice, sjediste,
raspoloziva_sredstva, porez_na_dobitak) VALUES
    (inc.nextval, 3, 'obicna', 'Guatemala', 212123.22, 0.22);

```

```

INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, poslovница_id) VALUES
    (inc22.nextval, 'Reynaldo', 'Arata', 1204.4, 0);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, poslovница_id) VALUES
    (inc22.nextval, 'Samantha', 'Swords', 4564.4, 2);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, poslovница_id) VALUES
    (inc22.nextval, 'Faith', 'Dalton', 1354.65, 8);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, poslovница_id) VALUES
    (inc22.nextval, 'Vernia', 'Rylander', 2354.564, 1);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, poslovница_id) VALUES
    (inc22.nextval, 'Lakia', 'Allender', 4512.75, 8);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, povecana_stopa_dobitka,
smanjenje_poreza, poslovница_id) VALUES
    (inc22.nextval, 'Jessie', 'Constable', 10000.0, 0.02, 0.05, 8);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, povecana_stopa_dobitka,
poslovnica_id) VALUES
    (inc22.nextval, 'Tuan', 'Madera', 1134.11, 0.12, 3);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, smanjenje_poreza, poslovница_id)
VALUES
    (inc22.nextval, 'Verdell', 'Flom', 9945, 0.31, 3);
INSERT INTO Klijeti (id, ime, prezime, dostupna_sredstva, smanjenje_poreza, poslovница_id)
VALUES
    (inc22.nextval, 'Nanci', 'Lowe', 11111, 0.11, 5);

```

```

INSERT INTO Odjeli (id, naziv_odjela, poslovница_id) VALUES
    (inc.nextval, 'javni odnosi', 0);
INSERT INTO Odjeli (id, naziv_odjela, poslovница_id) VALUES

```



```

    (inc.nextval, 'menadzment', 0);
INSERT INTO Odjeli (id, naziv_odjela, poslovnica_id) VALUES
    (inc.nextval, 'javni odnosi', 1);
INSERT INTO Odjeli (id, naziv_odjela, poslovnica_id) VALUES
    (inc.nextval, 'javni odnosi', 2);
INSERT INTO Odjeli (id, naziv_odjela, poslovnica_id) VALUES
    (inc.nextval, 'javni odnosi', 3);
INSERT INTO Odjeli (id, naziv_odjela, poslovnica_id) VALUES
    (inc.nextval, 'podrska', inc22.nextval);
INSERT INTO Odjeli (id, naziv_odjela, poslovnica_id) VALUES
    (inc.nextval, 'glavni odbor', 0);

```

```

INSERT INTO Poslovi(id, naziv_posla, plata) VALUES (inc.nextval, 'online podrska', 2000);
INSERT INTO Poslovi(id, naziv_posla, plata) VALUES (inc.nextval, 'terenska podrska', 2500);
INSERT INTO Poslovi(id, naziv_posla, plata) VALUES (inc.nextval, 'sef', 7000);
INSERT INTO Poslovi(id, naziv_posla, plata) VALUES (inc.nextval, 'ekonomski menadzer',
3000);
INSERT INTO Poslovi(id, naziv_posla, plata) VALUES (inc.nextval, 'menadzer osoblja', 3300);
INSERT INTO Poslovi(id, naziv_posla, plata) VALUES (inc.nextval, 'zamjenik sefa', 4500);

```

```

INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Vedad', 'Fejzagic', 26, 3, TO_DATE('11-11-2016', 'dd-mm-yyyy'));
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Abena', 'Pasternack', 11, 3, current_date);
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Agung', 'Rahman', 11, 4, current_date);
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Aditya', 'Bagus', 11, 5, current_date);
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Raja', 'Guntur', 23, 1, current_date);
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Wira', 'Abdullah', 23, 2, current_date);
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Taufik', 'Dewi', 26, 6, current_date);
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Cahaya', 'Widya', 14, 4, current_date);
INSERT INTO Zaposleni(id, ime, prezime, odjel_id, posao_id, datum_zaposlenja) VALUES
(inc.nextval, 'Vina', 'Ibrahim', 15, 4, current_date);

```

```
INSERT INTO OpkladeIgraci (id, uplaceno_novca, datum_uplate, odabrani_igrac_id,
odabrana_utakmica_id, broj_poena, poslovnica_id, klijent_id)
```

```
VALUES (inc.nextval, 600, current_date, 1, 1, 11, 0, 1);
```

```
INSERT INTO OpkladeIgraci (id, uplaceno_novca, datum_uplate, odabrani_igrac_id,
odabrana_utakmica_id, broj_poena, poslovnica_id, klijent_id)
```

```
VALUES (inc.nextval, 400, current_date, 8, 4, 8, 0, 1);
```

```
INSERT INTO OpkladeIgraci (id, uplaceno_novca, datum_uplate, odabrani_igrac_id,
odabrana_utakmica_id, broj_poena, poslovnica_id, klijent_id)
```

```
VALUES (inc.nextval, 600, current_date, 15, 11, 1, 0, 2);
```

```
INSERT INTO OpkladeUtakmice (id, uplaceno_novca, datum_uplate, odabrana_utakmica_id,
poeni_domaci, poeni_gost, poslovnica_id, klijent_id) VALUES
```

```
(inc.nextval, 5000, current_date, 3, 2, 3, 8, 6);
```

```
INSERT INTO OpkladeUtakmice (id, uplaceno_novca, datum_uplate, odabrana_utakmica_id,
poeni_domaci, poeni_gost, poslovnica_id, klijent_id) VALUES
```

```
(inc.nextval, 200, current_date, 7, 8, 10, 3, 7);
```

```
INSERT INTO OpkladeMecevi (id, uplaceno_novca, datum_uplate, odabrani_mec_id,
poeni_domaci, poeni_gost, poslovnica_id, klijent_id) VALUES
```

```
(inc.nextval, 1200, current_date, 1, 1, 0, 5, 9);
```

```
INSERT INTO OpkladeMecevi (id, uplaceno_novca, datum_uplate, odabrani_mec_id,
poeni_domaci, poeni_gost, poslovnica_id, klijent_id) VALUES
```

```
(inc.nextval, 1200, current_date, 2, 1, 2, 8, 6);
```

```
INSERT INTO IgraciEfikasnost (id, igrac_id, utakmica_id, broj_poena) VALUES (inc.nextval, 1,
1, 3);
```

```
INSERT INTO IgraciEfikasnost (id, igrac_id, utakmica_id, broj_poena) VALUES (inc.nextval, 8,
4, 1);
```

```
INSERT INTO IgraciEfikasnost (id, igrac_id, utakmica_id, broj_poena) VALUES (inc.nextval, 15,
11, 0);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Joel',
'Buchanan', 12, 1);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Pat',
'Grant', 5, 1);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone) VALUES (inc.nextval, 'Tyrone',
'Reyes', 10);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone) VALUES (inc.nextval, 'Chris', 'Hall',
13);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Flora', 'Hale', 1, 2);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Debbie', 'Wise', 11, 2);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Susie', 'Wells', 51, 3);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Flora', 'Hale', 32, 4);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Joey', 'Ball', 22, 5);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Gilbert', 'Moran', 18, 6);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Ignacio', 'Blake', 11, 9);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Chris', 'Scott', 12, 10);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Agnes', 'Lawson', 13, 11);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Jeremy', 'Lindsey', 14, 12);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Maurice', 'May', 15, 13);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Beatry', 'Luna', 16, 14);
```

```
INSERT INTO Igraci (id, ime, prezime, poeni_ove_sezone, tim_id) VALUES (inc.nextval, 'Milton', 'Williamson', 17, 15);
```

```
INSERT INTO Mecevi (id, igrac_domaci_id, igrac_gost_id, poeni_domaci, poeni_gost) VALUES (inc.nextval, 3, 4, 1, 1);
```

```
INSERT INTO Mecevi (id, igrac_domaci_id, igrac_gost_id, poeni_domaci, poeni_gost) VALUES (inc.nextval, 4, 3, 2, 1);
```

```
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Real Madrid', 22, 'Zinedine Zidane', 1);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Barcelona', 23, 'Roberto Carlos', 1);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Inter', 18, 'Luis Enrique', 1);
```

```
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Sarajevo', 11, 'Husen Kapetan', 2);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Crvena Zvezda', 13, 'Vladan Milojevic', 2);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Hajduk', 11, 'Zeljko Kopic', 2);
```

```
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'LA Lakers', 23, 'Leona Castillo', 4);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Golden State Warriors', 28, 'Roberto Carlos', 4);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Boston Celtics', 20, 'Margarita Ross', 4);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Oklahoma City Thunder', 31, 'Casey Stone', 4);
```

```
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'New York Knicks', 16, 'Terrance King', 5);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Chicago Bulls', 17, 'Annete Lawson', 5);
INSERT INTO Timovi (id, naziv, poeni_ove_sezone, trener, liga_id) VALUES (inc.nextval, 'Toronto Raptors', 18, 'Patty Drake', 5);
```

```
INSERT INTO Utakmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES (inc.nextval, 1, 2, 2, 2);
INSERT INTO Utakmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES (inc.nextval, 1, 3, 1, 2);
INSERT INTO Utakmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES (inc.nextval, 2, 3, 2, 3);
```

```
INSERT INTO Utakmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES (inc.nextval, 4, 5, 1, 1);
INSERT INTO Utakmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES (inc.nextval, 4, 6, 2, 1);
INSERT INTO Utakmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES
```

(inc.nextval, 5, 6, 4, 2);

INSERT INTO UtaKmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES
(inc.nextval, 9, 10, 22, 36);

INSERT INTO UtaKmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES
(inc.nextval, 9, 11, 44, 42);

INSERT INTO UtaKmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES
(inc.nextval, 12, 9, 51, 34);

INSERT INTO UtaKmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES
(inc.nextval, 11, 12, 47, 44);

INSERT INTO UtaKmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES
(inc.nextval, 13, 14, 1, 1);

INSERT INTO UtaKmice (id, tim_domaci_id, tim_gost_id, poeni_domacin, poeni_gost) VALUES
(inc.nextval, 14, 13, 1, 1);

INSERT INTO Lige (id, ime, sport) VALUES (inc.nextval, 'Liga A', 'Fudbal');

INSERT INTO Lige (id, ime, sport) VALUES (inc.nextval, 'Liga B', 'Fudbal');

INSERT INTO Lige (id, ime, sport) VALUES (inc.nextval, 'Liga C', 'Fudbal');

INSERT INTO Lige (id, ime, sport) VALUES (inc.nextval, 'A Liga', 'Kosarka');

INSERT INTO Lige (id, ime, sport) VALUES (inc.nextval, 'B Liga', 'Kosarka');

d) 30 SQL upita

1. 10 jednostavnijih:

- Odjeli u glavnoj poslovnici:

```
SELECT o.naziv_odjela from Odjeli o, Poslovnice p where o.poslovnica_id = p.id  
AND p.sjediste = 'Ciudad de Juarez';
```

- Klijenti koji nemaju smanjenje poreza sortirani prema opadajućem redoslijedu

```
SELECT k.ime || ' ' || k.prezime "Naziv klijenta" , k.smanjenje_poreza "Procenat  
smanjenja poreza" FROM Klijenti k, Poslovnice p  
WHERE k.poslovnica_id = p.id  
AND k.smanjenje_poreza IS NOT NULL  
ORDER BY k.smanjenje_poreza DESC;
```

- Prosječna vrijednost dostupnih sredstava klijenata u poslovnici u Havani

```
SELECT ROUND(AVG(k.dostupna_sredstva), 2) "Dostupna sredstva prosjek"  
FROM Klijenti k, Poslovnice p  
WHERE k.poslovnica_id = p.id  
AND p.sjediste LIKE '%Havana%';
```

- Naziv sefa glavne poslovnice

```
SELECT z.ime || ' ' || z.prezime "Naziv sefa" FROM Zaposleni z, Poslovi p, Odjeli  
o, Poslovnice l  
WHERE p.id = z.posao_id  
AND o.id = z.odjel_id  
AND o.poslovnica_id = l.id  
AND l.tip_poslovnice LIKE 'glavna'
```

AND p.naziv_posla LIKE 'sef';

- Svi zaposleni, njihovi poslovi i plate koji rade u odjelu podrška u Sarajevu

```
SELECT z.ime || ' ' || z.prezime "Naziv" , p.naziv_posla, p.plata FROM Zaposleni
z, Poslovi p, Poslovnice l, Odjeli o
WHERE l.id = o.poslovnica_id
AND o.id = z.odjel_id
AND o.naziv_odjela LIKE 'podrska'
AND l.sjediste LIKE 'Sarajevo'
AND z.posao_id = p.id;
```

- Igraci koji nemaju tim

```
SELECT i.ime || ' ' || i.prezime "Naziv igraca", i.poeni_ove_sezone "Poeni" FROM
Igraci i WHERE i.tim_id IS NULL ;
```

- Prikaz tima igrača i njihovih imena kojima se pratila efikasnost

```
SELECT t.naziv "Naziv tima", i.ime || ' ' || i.prezime "Naziv igraca"
FROM Timovi t, Igraci i, IgraciEfikasnost ie
WHERE ie.igrac_id = i.id
AND i.tim_id = t.id;
```

- Prikaz klijenata koji se nisu kladili

```
SELECT DISTINCT k.ime || ' ' || k.prezime "Naziv klijenta", p.sjediste "Sjediste
poslovnice"
FROM Klijenti k, Poslovnice p, OpkladeIgraci oi
WHERE oi.klijent_id <> k.id
AND k.poslovnica_id = p.id;
```

- Klijenti koji su pogodili ishod opklade na utakmice

```
SELECT k.ime || ' ' || k.prezime "Naziv klijenta", ou.uplaceno_novca "Uplaceno  
novca"  
FROM Klijenti k, OpkladeUtakmice ou, Utakmice u  
WHERE k.id = ou.klijent_id  
AND ou.odabrana_utakmica_id = u.id  
AND ou.poeni_domaci = u.poeni_domacin  
AND ou.poeni_gost = u.poeni_gost;
```

- Naziv klijenta i njihov potencijalni dobitak sa uracunatom povecanom stopom
dobitka i smanjenja poreza ako su uplatili 4000 pod uslovom da imaju toliko
dostupnih sredstava

```
SELECT k.ime || ' ' || k.prezime "Naziv klijenta", 4000 + (1000 *  
Nvl(k.povecana_stopa_dobitka,0)) + (1000 * Nvl(k.smanjenje_poreza,0)) Dobitak  
FROM Klijenti k  
WHERE k.dostupna_sredstva >= 4000;
```


2. 5 upita sa grupnim funkcijama (minimalno 2 sa HAVING)

1. suma raspolozivih sredstava svih poslovnica

```
SELECT Sum(raspoloziva_sredstva) "Suma raspolozivih sredstava" FROM  
Poslovnice;
```

2. Srednja vrijednost i suma dostupnih sredstava klijenata na 2 decimale

```
SELECT Round(Avg(k.dostupna_sredstva), 2) "Srednja vrijednost",  
Round(Sum(k.dostupna_sredstva), 2) "Suma dostupnih sredstava"  
FROM Klijenti k;
```

3. Broj zaposlenih po odjelima pri cemu je broj zaposlenih manji od srednje vrijednosti svih zaposlenih

```
SELECT Count(z.id) "Broj zaposlenih", o.naziv_odjela "Naziv odjela"  
FROM Zaposleni z, Odjeli o  
WHERE z.odjel_id = o.id  
GROUP BY o.naziv_odjela  
HAVING Avg(z.id) > Count(z.id);
```

4. Sjediste Poslovnica koje imaju raspolozivih sredstava vise od zbira suma opklada igraca i opklada utakmica

```
SELECT p.sjediste, Sum(ou.uplaceno_novca) + Sum(oi.uplaceno_novca) +  
Sum(om.uplaceno_novca) "Suma"  
FROM OpkladeUtakmice ou, OpkladeIgraci oi, OpkladeMecevi om, Poslovnice p  
HAVING p.raspoloziva_sredstva > Sum(ou.uplaceno_novca) +  
Sum(oi.uplaceno_novca) + Sum(om.uplaceno_novca)  
GROUP BY p.sjediste, p.raspoloziva_sredstva;
```

5. Najviše odjela u poslovnici

```
SELECT Max(Count(o.id)) "Broj odjela"  
  FROM Poslovnice p, Odjeli o  
 WHERE p.id = o.poslovnica_id  
 GROUP BY o.poslovnica_id;
```

3. 5 upita sa korištenjem podupita:

/*1. Zaposleni koji zaradjuju platu vecu od prosjecne plate zaposlenih u odjelima 23 i 11 sortirano u opadajucem redoslijedu*/

```
SELECT z.ime || ' ' || z.prezime "Naziv zaposlenog", p.plata "Plata"
FROM Zaposleni z, Poslovi p
WHERE z.posao_id = p.id
AND p.plata > (SELECT Avg(p.plata)
               FROM Zaposleni z, Poslovi p
               WHERE z.posao_id = p.id
               AND (z.odjel_id = 23 OR z.odjel_id = 11)
              )
ORDER BY p.plata DESC;
```

/*2. Zaposleni koji rade u istom odjelu kao i neki od zaposlenih cije ime pocinje sa 'A'*/

```
SELECT z.ime || ' ' || z.prezime "Naziv zaposlenog", z.odjel_id "Odjel"
FROM Zaposleni z
WHERE z.odjel_id = ANY (SELECT o.id
                       FROM Odjeli o, Zaposleni z
                       WHERE z.odjel_id = o.id
                       AND z.ime LIKE 'A%');
```

/*3. Zaposleni koji primaju platu manju od plate Aditya Bagus*/

```
SELECT z.ime || ' ' || z.prezime "Naziv zaposlenog", z.odjel_id "Odjel",
p.naziv_posla "Posao", p.plata "Plata"
FROM Zaposleni z, Poslovi p
WHERE z.posao_id = p.id
AND p.plata < (SELECT pp.plata
               FROM Poslovi pp, Zaposleni zz
```

```
WHERE pp.id = zz.posao_id
AND zz.ime LIKE '%Aditya%'
AND zz.prezime LIKE '%Bagus%');
```

/*4. Zaposleni koji ne primaju najveću i najmanju platu*/

```
SELECT z.ime || ' ' || z.prezime "Naziv zaposlenog", p.plata "Plata"
FROM Zaposleni z, Poslovi p
WHERE z.posao_id = p.id
AND z.posao_id = p.id
AND p.plata <> (SELECT Max(pp.plata) FROM Poslovi pp)
AND p.plata <> (SELECT Min(pp.plata) FROM poslovi pp)
ORDER BY p.plata DESC;
```

/*5. Klijenti čija su dostupna sredstva veća od dostupnih sredstava klijenata koji imaju povećanu stopu dobitka*/

```
SELECT k.ime || ' ' || k.prezime "Naziv klijenta" , k.dostupna_sredstva "Dostupna
sredstva"
FROM Klijenti k
WHERE k.dostupna_sredstva > ALL (SELECT kk.dostupna_sredstva
FROM Klijenti kk
WHERE kk.povecana_stopa_dobitka IS NOT NULL);
```

4. 5 upita sa više od jednog nivoa podupita

/*1. Poslovnice sa raspolozivim sredstvima vecim od srednje vrijednosti raspolozivih sredstava svih poslovnica koje imaju vise od jednog odjela*/

```
SELECT p.sjediste "Sjediste", p.raspoloziva_sredstva "Raspoloziva sredstva"
FROM Poslovnice p
WHERE p.raspoloziva_sredstva > (SELECT Avg(pp.raspoloziva_sredstva)
FROM Poslovnice pp
WHERE (SELECT Count(o.id)
FROM Odjeli o
WHERE pp.id = o.poslovnica_id) > 1);
```

/*2. Utakmice u kojima su domaćini imali više poena od prosječne vrijednosti poena gostiju svih utakmica čiji poeni su također veći od srednje vrijednosti poena gostiju u mečevima*/

[illegible]

5. 2 upita sa subtotalima (ROLLUP, CUBE, GROUPING SETS)

/*1. Ukupna dostupna sredstva klijenata po svakoj poslovnici kao i zbir ukupnih sredstava svih poslovnica */

```
SELECT p.sjediste "Poslovnica", Sum(k.dostupna_sredstva) "Dostupna sredstva"  
FROM Poslovnice p, Klijenti k  
WHERE p.id = k.poslovnica_id  
GROUP BY ROLLUP (p.sjediste);
```

/*2. Ukupna plata zaposlenog grupisano po odjelu i nazivu posla*/

```
SELECT Sum(p.plata)"Plata", z.odjel_id "Odjel", p.naziv_posla "Naziv posla"  
FROM Zaposleni z, Poslovi p  
WHERE z.posao_id = p.id  
GROUP BY CUBE (z.odjel_id, p.naziv_posla);
```

6. Upit sa korištenjem UNION, SET

```
SELECT Poslovnice.id, sjediste, raspoloziva_sredstva
FROM Poslovnice
LEFT JOIN Odjeli
ON Poslovnice.id = Odjeli.poslovnica_id
UNION ALL
SELECT Poslovnice.id, sjediste, raspoloziva_sredstva
FROM Poslovnice
RIGHT JOIN Odjeli
ON Poslovnice.id = Odjeli.poslovnica_id;
```

e) 5 Indeksa

```
CREATE INDEX naziv_klijenta_index ON Klijenti(ime, prezime);
CREATE INDEX informacija_posl_index ON Poslovnice(sjediste,
raspoloziva_sredstva);
CREATE INDEX svi_poslovi_index ON Poslovi(naziv_posla, plata);
CREATE INDEX zaposleni_index ON Zaposleni(ime, prezime,
datum_zaposlenja);
CREATE INDEX igraci_index ON Igraci(ime, prezime, poeni_ove_sezone);
```

f) 10 Funkcija

/*1. funkcija koja vraca vecu vrijednost varijable tipa float*/

CREATE OR REPLACE FUNCTION je_veci(

 a in float,

 b in float

)

return FLOAT

IS

BEGIN

 IF (a > b) THEN return a;

 ELSE return b;

 END IF;

END je_veci;

/*2. Funkcija za provjeru postojanja klijenta koja vraca id klijenta ako je on pronadjen, ako ne onda vraca -1*/

CREATE OR REPLACE FUNCTION klijent_postoji(

 imee in VARCHAR,

 prezimee in VARCHAR

)

return INTEGER

IS

 broj INTEGER;

BEGIN

 SELECT k.id INTO broj

 FROM Klijenti k

 WHERE k.ime LIKE imee

 AND k.prezime LIKE prezimee;


```
if(Nvl(broj, 0) != 0) then return broj;  
else return -1;
```

```
END IF;  
END klijent_postoji;
```

/*3. Funkcija za dobavljanje prosjecne vrijednosti dostupnih sredstava klijenata u odabranoj poslovnici*/

```
CREATE OR REPLACE FUNCTION daj_dostupna_sredstva(  
  poslovnica INTEGER  
)  
  return FLOAT  
IS  
  rez FLOAT;  
BEGIN  
  SELECT Avg(k.dostupna_sredstva) INTO rez FROM Klijenti k  
  WHERE k.poslovnica_id = poslovnica;  
  Nvl(rez, -1);  
  return rez;  
END;
```

/*4. Funkcija koja vraca naziv zaposlenog i njegovu platu*/

```
CREATE OR REPLACE FUNCTION daj_zaposleni_platu(  
  id_zaposlenog INTEGER  
)  
  RETURN VARCHAR  
IS  
  lv_rez VARCHAR(255);  
  lv_ime VARCHAR(255);  
  lv_prezime VARCHAR(255);  
  lv_naziv VARCHAR(255);
```

```
lv_plata float;  
lv_broj_zap INTEGER;
```

```
CURSOR c_broj_zap IS  
  SELECT Count(id) FROM Zaposleni  
  WHERE id = id_zaposlenog;
```

```
BEGIN
```

```
  OPEN c_broj_zap;  
  FETCH c_broj_zap INTO lv_broj_zap;  
  CLOSE c_broj_zap;
```

```
  if(lv_broj_zap = 1) then  
    SELECT z.ime, z.prezime, p.plata INTO lv_ime, lv_prezime, lv_plata  
    FROM Zaposleni z, Poslovi p  
    WHERE z.id = id_zaposlenog AND z.posao_id = p.id;  
    lv_rez := CONCAT(CONCAT(lv_ime, lv_prezime), TO_CHAR(lv_plata));  
    return lv_rez;  
  else  
    RAISE_APPLICATION_ERROR(-1001, 'Zaposleni ne postoji');  
  end IF;
```

```
END daj_zaposleni_platu;
```

```
/*5. Funkcija koja vraca broj timova u ligi*/
```

```
CREATE OR REPLACE FUNCTION daj_broj_timova(uv_liga_id INTEGER)  
  RETURN INTEGER  
  IS  
    lv_rez INTEGER;  
    CURSOR c_broj_timova IS  
      SELECT Count(id) FROM Timovi WHERE liga_id = uv_liga_id;
```

```
BEGIN
  OPEN c_broj_timova;
  FETCH c_broj_timova INTO lv_rez;
  CLOSE c_broj_timova;

  RETURN lv_rez;
END daj_broj_timova;
```

/*6. Funkcija koja vraca broj igraca u timu, ili izuzetak ako tim ne postoji.*/

```
CREATE OR REPLACE FUNCTION daj_broj_igraca(uv_tim_id INTEGER)
  RETURN INTEGER
IS
  lv_broj_igraca INTEGER;
  lv_tim_postoji INTEGER;

  CURSOR c_postoji IS
    SELECT Count(id) FROM Timovi
  WHERE uv_tim_id = id;

BEGIN

  OPEN c_postoji;
  FETCH c_postoji INTO lv_tim_postoji;
  CLOSE c_postoji;

  if(lv_tim_postoji > 0) THEN
    SELECT Count(i.id) INTO lv_broj_igraca FROM Igraci i, Timovi t
    WHERE i.tim_id = t.id;
    RETURN lv_broj_igraca;
  else
    RAISE_APPLICATION_ERROR(-13512, 'Tim ne postoji');
  END IF;
```

END;

/*7. Funkcija koja vraca broj poena igraca u nekoj utakmici*/

CREATE OR REPLACE FUNCTION daj_broj_poena_igraca_utakmice(

uv_igrac_id INTEGER,

uv_utakmica_id INTEGER

)

RETURN INTEGER IS

lv_broj_poena INTEGER;

lv_utakmica_postoji INTEGER;

lv_igrac_postoji INTEGER;

CURSOR c_utakmica IS

SELECT Count(id)

FROM Utakmice WHERE uv_utakmica_id = id;

CURSOR c_igrac IS

SELECT Count(id)

FROM Igraci WHERE uv_igrac_id = id;

BEGIN

OPEN c_utakmica;

FETCH c_utakmica INTO lv_utakmica_postoji;

CLOSE c_utakmica;

OPEN c_igrac;

FETCH c_igrac INTO lv_igrac_postoji;

CLOSE c_igrac;

if(lv_igrac_postoji > 0 AND lv_utakmica_postoji > 0) THEN

```

SELECT ie.broj_poena INTO lv_broj_poena FROM Utakmice u, Igraci i,
IgraciEfikasnost ie
WHERE u.id = ie.utakmica_id AND i.id = ie.igrac_id
AND u.id = uv_utakmica_id AND i.id = uv_igrac_id;
return lv_broj_poena;
ELSE
RAISE_APPLICATION_ERROR(-1345, 'Igrac i/ili utakmica ne postoje');
END IF;

END;

```

/*8. Funkcija koja vraca ukupan broj opklada u poslovnici*/

```

CREATE OR REPLACE FUNCTION
daj_ukupan_broj_opklada(uv_poslovnica_id INTEGER)
RETURN INTEGER IS
lv_ukupno INTEGER;

CURSOR c_o IS
SELECT Nvl(Count(om.id), 0) + Nvl(Count(oi.id), 0) + Nvl(Count(ou.id), 0)
FROM OpkladeUtakmice ou, OpkladeMecevi om, OpkladeIgraci oi
WHERE ou.poslovnica_id = uv_poslovnica_id
AND om.poslovnica_id = uv_poslovnica_id
AND oi.poslovnica_id = uv_poslovnica_id;
BEGIN

OPEN c_o;
FETCH c_o INTO lv_ukupno;
CLOSE c_o;

return lv_ukupno;
END;

```

/*9. Funkcija koja vraca ukupno uplaceno novca za igraca u odredjenoj utakmici (tabela opkladeigraci)*/

```
CREATE OR REPLACE FUNCTION daj_ukupno_novca(  
  uv_igrac_id INTEGER,  
  uv_utakmica_id INTEGER  
)
```

```
RETURN FLOAT IS
```

```
lv_rez FLOAT;
```

```
lv_utakmica_postoji INTEGER;
```

```
lv_igrac_postoji INTEGER;
```

```
CURSOR c_utakmica IS
```

```
  SELECT Count(odabrana_utakmica_id)
```

```
  FROM OpkladeIgraci WHERE uv_utakmica_id = odabrana_utakmica_id;
```

```
CURSOR c_igrac IS
```

```
  SELECT Count(odabrani_igrac_id)
```

```
  FROM OpkladeIgraci WHERE uv_igrac_id = odabrana_utakmica_id;
```

```
BEGIN
```

```
  OPEN c_utakmica;
```

```
  FETCH c_utakmica INTO lv_utakmica_postoji;
```

```
  CLOSE c_utakmica;
```

```
  OPEN c_igrac;
```

```
  FETCH c_igrac INTO lv_igrac_postoji;
```

```
  CLOSE c_igrac;
```

```
  if(lv_igrac_postoji > 0 AND lv_utakmica_postoji > 0) THEN
```

```
    SELECT Sum(uplaceno_novca) INTO lv_rez FROM OpkladeIgraci
```

```
    WHERE uv_igrac_id = odabrani_igrac_id
```

```
    AND uv_utakmica_id = odabrana_utakmica_id;
```

```
  return lv_rez;
```

```
ELSE
    RAISE_APPLICATION_ERROR(-1325, 'Igrac i/ili utakmica ne postoje u
opkladama ili ne postoje nikako');
end If;
END;
```

```
/*10. Funkcija koja vraca prosjek raspolozivih sredstava svih poslovnica */
CREATE OR REPLACE FUNCTION daj_prosjek_rs_posl()
RETURN FLOAT IS
lv_rez FLOAT;
BEGIN
    SELECT Avg(raspoloziva_sredstva) INTO lv_rez FROM Poslovnice;
    return lv_rez;
END;
```

g) 10 Procedura

/*Procedura za azuriranje pogodnosti klijentu*/

```
CREATE OR REPLACE PROCEDURE azuriraj_pogodnosti(  
    uv_klijent_id INTEGER,  
    uv_povecana_stopa_dobitka FLOAT,  
    uv_smanjenje_poreza FLOAT  
)  
IS
```

```
    lv_id_klijenta INTEGER := -1;
```

```
    CURSOR c_klijent_postoji IS  
    SELECT k.id FROM Klijenti k  
    WHERE k.id = uv_klijent_id;
```

```
BEGIN
```

```
    OPEN c_klijent_postoji;  
    FETCH c_klijent_postoji INTO lv_id_klijenta;  
    CLOSE c_klijent_postoji;
```

```
    if(lv_id_klijenta <> -1) THEN  
        if(uv_povecana_stopa_dobitka < 0 OR uv_povecana_stopa_dobitka > 1 OR  
uv_smanjenje_poreza < 0 OR uv_smanjenje_poreza > 1) THEN  
            RAISE_APPLICATION_ERROR(-4650, 'Parametri moraju biti u opsegu [0,  
1]');  
        else  
            UPDATE Klijenti SET  
                povecana_stopa_dobitka = uv_povecana_stopa_dobitka,  
                smanjenje_poreza = uv_smanjenje_poreza  
            WHERE id = uv_klijent_id;
```



```
    END IF;  
else  
    RAISE_APPLICATION_ERROR(-4651, 'Klijent ne postoji');  
END IF;  
  
END azuriraj_pogodnosti;  
  
CALL azuriraj_pogodnosti(1, 0.1, 0.02);
```