

Nogometni klub

Dokumentacija

Matej Kolak

Moderni sustavi baza podataka

Sadržaj

1. Uvod
 - a. Motivacija
 - b. Kratki opis
2. Model baze podataka
 - a. Model entiteta i veza
 - b. Relacijski model
 - c. Tablice
3. Funkcionalnosti
 - a. Okidači
 - b. Procedure
 - c. Indeksi

1. Uvod

a. Motivacija

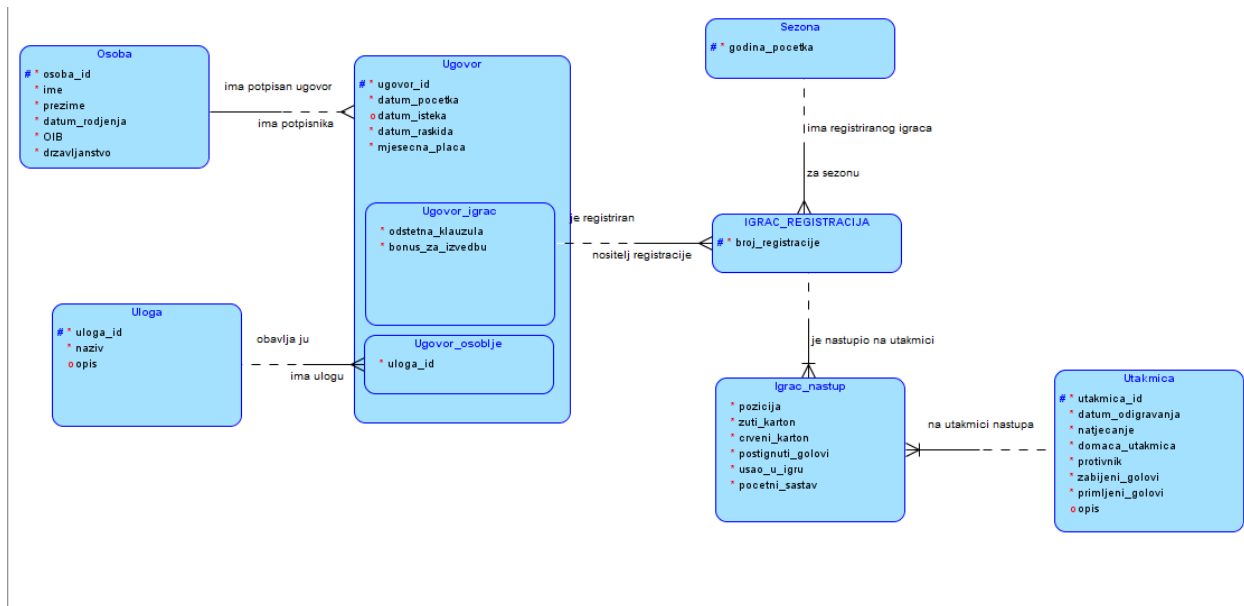
Od trenutka kada sam prvi puta kročio na zeleni travnjak pa sve do današnjeg dana, nogomet je bio važan dio mog života. S obzirom da su se i moj otac i moja starija braća bavila nogometom, tako sam i ja krenio njihovim stopama. Od malena sam igrao, pratio utakmice i kretao se u krugu ljudi uključenih u nogomet. Godine opažanja i razumijevanja kako stvari funkcioniraju iznutra navele su me da za svoj projekt odaberem izradu baze za nogometni klub.

b. Kratki opis

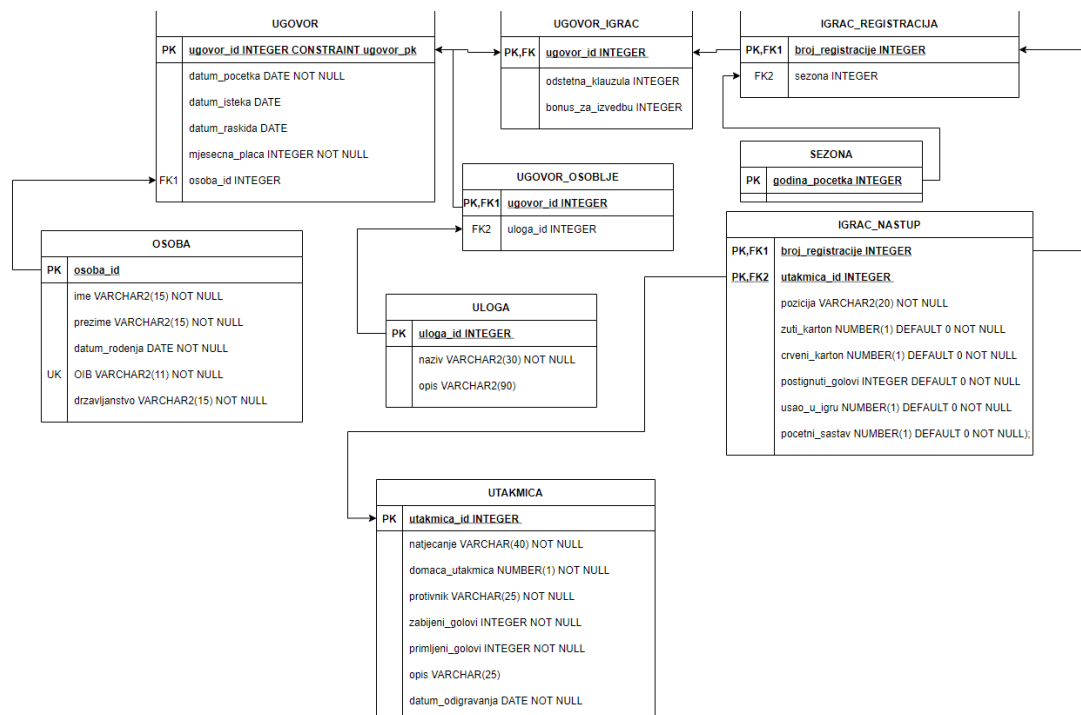
Nogometni klub je baza podataka koja služi kako bi olakšala upravljanje nogometnim klubom. Djelovanje kluba sastoji se od administrativnog i sportskog segmenta. U administrativnom dijelu, klub zapošljava osoblje i igrače, dijeli im uloge i potpisuje ugovore s njima. U sportskom dijelu, klub je dužan voditi računa o utakmicama koje se odigravaju, igračima koji nastupaju na utakmicama i o njihovoj statistici na tim utakmicama.

2. Model baze podataka

a. Model entiteta i veza



b. Relacijski model



c. Tablice

OSOBA je tablica u kojoj se nalaze zapisi o svim osobama koje su u bilo kojem trenutku sudjelovale u radu kluba. U njoj se nalaze osnovni podaci nalik onima koje bismo pronašli na osobnoj iskaznici. Iako bi OIB mogao poslužiti kao primarni ključ, tablica ima umjetni primarni ključ zbog tehničkih razloga.

UGOVOR je tablica u koju unosimo podatke o ugovorima koje neka osoba potpisuje sa klubom. Osoba može potpisati više ugovora, pa prema tome osoba može obavljati i više funkcija. Naprimjer, osoba koja je nekoć u klubu bila igrač, kasnije može postati trener. Obavezna polja u ugovoru su umjetno generirani primarni ključ, datum početka ugovora, mjesečna plaća i strani ključ koji povezuje **UGOVOR** s **OSOBA**. Polja koja nisu obavezna, ali su jednako važna su datum isteka i datum raskida. Na prvu, činilo bi se da ta dva polja jednaka, no ona imaju različito značenje i pomoću njih omogućeno je potpisivanje dviju različitih vrsta ugovora. Prva vrsta je ugovor na određeni period. U tom slučaju, prilikom unosa u tablicu ugovor, točno moramo odrediti datum isteka ugovora. Druga vrsta je ugovor na neodređeni period i u tom slučaju datum isteka je **null**. U oba slučaja, polje datum raskida je po defaultu **null** sve do onog trenutka kada:

1. Raskidamo ugovor prijevremeno(Datum raskida < Datum isteka)
2. Prošao je datum isteka ugovora(Datum raskida = Datum isteka)
3. Raskidamo ugovor na neodređeno

Postojanje polja datum raskida omogućuje nam prema tome da u svakom trenutku možemo pronaći ugovore koji trenutno traju(nisu raskinuti).

UGOVOR_IGRAC koristimo za zapis svih ugovora igrača u klubu i u njoj možemo pronaći neke dodatne stavke poput odštetne klauzule i bonusa za izvedbu.

UGOVOR_OSOBLJE i ULOGA služe za zapis svih ugovora onih osoba koje u klubu obavljaju neku dužnost, a nisu igrači. Te dužnosti detaljno su opisane u tablici **ULOGA**, dok se u tablici **UGOVOR_OSOBLJE** nalazi strani ključ koji ju povezuje s njom.

SEZONA sadrži zapise svih sezona u kojima je klub odigravao svoje utakmice. Unosi se samo godina početka sezone, dok je naredna godina uvijek godina poslije godine početka, pa da bi uštedili na prostoru ne unosimo godinu završetka. Također, podrazumijeva se da svaka sezona počinje 1. lipnja.

IGRAC_REGISTRACIJA sadrži umjetno generirani primarni ključ koji nazivamo broj registracije, te dva strana ključa koji spajaju tablicu s tablicama **UGOVOR_IGRAC** te **SEZONA**. Zapis u tablici možemo promatrati kao registraciju za nekog igrača koji ima ugovor s klubom i ona vrijedi za jednu sezonu. Svaku iduću sezonu, isti igrač mora imati novi zapis u tablici. Na razini razumijevanja, registracija se može promatrati poput članske iskaznice koja traje točno godinu dana i svake godine se mora nanovo izraditi.

UTAKMICA služi za unos podataka o utakmicama koje klub igra. Tablica ima umjetno generirani primarni ključ. Zapis ima datum održavanja, natjecanje u čijem sklopu se utakmica održava, boolean varijablu kojom utvrđujemo igra li se utakmica na domaćem terenu, naziv protivnika, te rezultat utakmice koji unosimo u polja zabijenih, te primljenih golova.

IGRAC_NASTUP sadrži zapise igrača koji su nastupili na nekoj utakmici. Primarni ključ u ovoj tablici je uređeni par stranih ključeva za tablice **IGRAC_REGISTRACIJA** i **UTAKMICA**. Zapis u tablici sadrži poziciju koju je igrač igrao, statističke podatke o kartonima koje je igrač zaradio, golovima koje je postigao, te da li je igrač započeo utakmicu od početka, ušao u igru, ili je ostao na klupi. Kako bi se osiguralo da igrač ne može istovremeno i igrati početni sastav i ući sa klupe, postavljeno je dodatno ograničenje koje će to provjeravati prilikom unosa i ažuriranja. Također, sva statistička polja imaju zadane vrijednosti 0.

3. Funkcionalnosti

a. Okidači

Validacija ugovora prilikom unosa registracije IGRAC_REGISTRACIJA

```
CREATE OR REPLACE TRIGGER valid_contract_season_registration
BEFORE INSERT OR UPDATE ON IGRAC_REGISTRACIJA
FOR EACH ROW
DECLARE
    ugovor_pocetak UGOVOR.datum_pocetka%type;
    ugovor_raskid UGOVOR.datum_raskida%type;
BEGIN
    SELECT datum_pocetka, datum_raskida INTO ugovor_pocetak, ugovor_raskid
    FROM UGOVOR WHERE ugovor_id = :new.ugovor_id;

    IF (TO_DATE('01-Jul-' || :new.sezona) <= ugovor_raskid OR TO_DATE('01-Jul-' || (:new.sezona + 1)) <= ugovor_pocetak) THEN
        RAISE_APPLICATION_ERROR(-20222, 'Igrač nema važeći ugovor za tu sezonu');
    END IF;
END;
/
```

Ovaj okidač služi kako bismo provjerili ima li igrač valjani ugovor za sezonu u kojoj ga želimo registrirati.

Primjer: Ukoliko igrač ima ugovor koji ističe 31.5.2020. , a sezona za koju ga želimo registrirati počinje 1.6.2020., okidač će obaciti pokušaj unosa.

Validacija broja igrača u početnom sastavu IGRAC_NASTUP

```
CREATE OR REPLACE TRIGGER eleven_players_in_field
BEFORE INSERT OR UPDATE ON IGRAC_NASTUP
FOR EACH ROW
WHEN (new.pocetni_sastav != 0)
DECLARE
    broj_igraca INTEGER;
BEGIN
    SELECT COUNT(*) INTO broj_igraca
    FROM IGRAC_NASTUP
    WHERE utakmica_id = :new.utakmica_id AND pocetni_sastav != 0;

    IF (broj_igraca = 11) THEN
        RAISE_APPLICATION_ERROR(-20224, 'Utakmicu ne može započeti više od 11 igrača.');
```

Okidač služi kako bismo spriječili korisnika da unese više od 11 igrača u početni sastav

Podudaranje broja postignutih golova IGRAC_NASTUP i UTAKMICA

```
CREATE OR REPLACE TRIGGER matching_goal_count
BEFORE INSERT OR UPDATE ON IGRAC_NASTUP
FOR EACH ROW
DECLARE
    broj_golova_nastup INTEGER;
    broj_golova_utakmica INTEGER;
BEGIN
    SELECT SUM(postignuti_golovi) + :new.postignuti_golovi INTO broj_golova_nastup
    FROM IGRAC_NASTUP
    WHERE utakmica_id = :new.utakmica_id;

    SELECT zabijeni_golovi INTO broj_golova_utakmica
    FROM UTAKMICA
    WHERE utakmica_id = :new.utakmica_id;
    IF (broj_golova_nastup > broj_golova_utakmica) THEN
        RAISE_APPLICATION_ERROR(-20223, 'Različit broj postignutih golova u tablicama utakmica i igrač nastup');
    END IF;
END;
/
```

S obzirom da u tablici utakmica unosimo podatak o broju golova koje je klub postigao na utakmici, suma golova koje su pojedini igrači postigli na utakmici mora biti manja ili jednaka broju golova koje je klub postigao. Napomena: uvjet manje ili jednako je tu zbog potencijalnih autogolova.

Primjer: Klub je odigrao utakmicu s rezultatom 3:0. Zapis u tablici utakmica, pod zabijeni golovi je 3. U tablici igrač nastup imamo slučaj da je zapisano kako je 'Igrač 1' zabio 2 gola, 'Igrač 2' zabio 1 gol te 'Igrač 3' zabio 1 gol. U sumi to su 4 gola i ne podudaraju se s podatkom da je klub na utakmici zabio 3 gola.

Validacija registriranih igrača na utakmici i datuma odigravanja

IGRAC_NASTUP i IGRAC_REGISTRACIJA

```
CREATE OR REPLACE TRIGGER matching_season_and_matchdate
BEFORE INSERT OR UPDATE ON IGRAC_NASTUP
FOR EACH ROW
DECLARE
    are_registrations_equal INTEGER;
    new_insert_season INTEGER;
    reg_season INTEGER;
    matchdate DATE;
BEGIN
    SELECT COUNT(*) INTO are_registrations_equal
    FROM (
        SELECT SEZONA
        FROM IGRAC_NASTUP
        INNER JOIN IGRAC_REGISTRACIJA
        USING(broj_registracije)
        WHERE :new.utakmica_id = utakmica_id
        GROUP BY SEZONA
    );

    SELECT SEZONA INTO new_insert_season
    FROM IGRAC_REGISTRACIJA
    WHERE :new.broj_registracije = broj_registracije;

    IF (are_registrations_equal = 1) THEN
        SELECT DISTINCT SEZONA INTO reg_season
        FROM IGRAC_NASTUP
        INNER JOIN IGRAC_REGISTRACIJA
        USING(broj_registracije)
        WHERE :new.utakmica_id = utakmica_id;
        SELECT datum_odigravanja INTO matchdate
        FROM UTAKMICA
        WHERE :new.utakmica_id = utakmica_id;
        IF (new_insert_season != reg_season) THEN
            RAISE_APPLICATION_ERROR(-20228, 'Unesena registracija nema valjanu sezonu');
        END IF;
        IF (TO_DATE('01-Jun-' || reg_season) >= matchdate OR matchdate >= TO_DATE('01-Jun-' || (reg_season + 1))) THEN
            RAISE_APPLICATION_ERROR(-20226, 'Registracije ne odgovaraju sezoni u kojoj se utakmica odigrava');
        END IF;
    ELSIF(are_registrations_equal > 1) THEN
        RAISE_APPLICATION_ERROR(-20227, 'U zapisniku se nalaze registracije za različite sezone');
    END IF;
END;
/
```

Okidač spriječava situacije u kojima bi se moglo dogoditi da se u tablici igrač nastup na istoj utakmici pojave igrači sa registracijama iz različitih sezona. Prema tome, okidač osigurava da svi igrači koji su prijavljeni na utakmici imaju registraciju u istoj sezoni. Podatak o toj istoj sezoni, koristimo kako bi dodatno provjerili je li datum odigrane utakmice unutar te sezone.

b. Procedure

Procedura za ažuriranje datuma raskida UGOVOR

```
CREATE OR REPLACE PROCEDURE update_datum_raskid AS
BEGIN
    UPDATE UGOVOR uout
    SET datum_raskida = datum_isteka
    WHERE datum_isteka <= CURRENT_DATE;
END;
/
```

Ovo je procedura pomoću koje će se ažurirati datum raskida u tablici ugovor, za one ugovore kojima je prošao datum isteka. U trenutku kada pozivamo ovu proceduru, uspoređujemo datum isteka sa trenutnim datumom, pozivajući pritom CURRENT_DATE. Ovu proceduru administrator baze treba pozvati periodički, kako bi zapisi u tablici UGOVOR bili validni.

Procedura za ispis početnog sastava s utakmice IGRAC_NASTUP

```
CREATE OR REPLACE PROCEDURE starting_lineup(  
    p_utakmica_id IN UTAKMICA.UTAKMICA_ID%TYPE  
) AS  
    utakmica_exists INTEGER;  
    v_ime OSOBA.ime%TYPE;  
    v_prezime OSOBA.ime%TYPE;  
    CURSOR v_osoba_kursor IS  
        SELECT ime, prezime  
        FROM IGRAC_NASTUP  
        INNER JOIN IGRAC_REGISTRACIJA  
        USING (broj_registracije)  
        INNER JOIN UGOVOR  
        USING (ugovor_id)  
        INNER JOIN OSOBA  
        USING (osoba_id)  
        WHERE utakmica_id = p_utakmica_id AND POCETNI_SASTAV != 0;  
BEGIN  
    SELECT COUNT(*) INTO utakmica_exists  
    FROM UTAKMICA  
    WHERE p_utakmica_id = utakmica_id;  
  
    IF (utakmica_exists > 0) THEN  
        OPEN v_osoba_kursor;  
  
        LOOP  
            FETCH v_osoba_kursor  
            INTO v_ime, v_prezime;  
            DBMS_OUTPUT.PUT_LINE(v_ime || ' ' || v_prezime);  
            EXIT WHEN v_osoba_kursor%NOTFOUND;  
        END LOOP;  
        CLOSE v_osoba_kursor;  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('Utakmica ne postoji');  
    END IF;  
END;  
/
```

Prilikom poziva procedure, kao argument predajemo id utakmice za koju želimo pronaći početni sastav kluba. Procedura prvo provjerava postoji li utakmica s tim id-jem, a zatim koristi kursor kako bi povukla podatke iz složenog upita koji je napravljen tako da vraća ime i prezime onih igrača koji su bili u početnom sastavu kluba na toj utakmici.

c. Indeksi

```
CREATE INDEX i_osoba_ime_prezime ON OSOBA(ime, prezime);  
  
CREATE INDEX i_utakmica_protivnik ON UTAKMICA(protivnik);  
  
CREATE BITMAP INDEX i_igrac_nastup_pozicija ON IGRAC_NASTUP(pozicija);
```

Index OSOBA(ime, prezime)

Budući da ćemo često pretraživati osobe po njihovom imenu i prezimenu, koja će se većinu vremena razlikovati, postavljen je index koji će ubrzati pretragu.

Index UTAKMICA(protivnik)

Protivnika s kojima se klub natječe je mnogo. Kako bi mogli analizirati statistiku u međusobnim susretima, često ćemo pretraživati utakmice po protivnicima.

Bitmap index IGRAC_NASTUP(pozicija)

Igrač na utakmici može nastupiti na jednoj od 4 moguće pozicije: napadač, veznjak, branič i golman. Kako je broj pozicija mali, bitmap index će nam pomoći prilikom pretrage.