

SVEUČILIŠTE J. J. STROSSMAYERA U OSIJEKU
ODJEL ZA MATEMATIKU

MODERNI SUSTAVI BAZA PODATAKA

SEMINAR

BAZA ZA KARATE LIGU

Dorotea Osmanović

1.srpanj.2021.

Sadržaj

1. Uvod	1
2. Dijagrami	2
2.1 MEV	2
2.2 Relacijski model	4
3. Kreiranje baze	5
3.1. Zadane vrijednosti.....	5
3.2. Uvjeti.....	6
4. Upiti.....	7
4.1 Jednostavni upiti	7
4.2 Složeni upiti	7
4.3 Upiti s agregirajućim funkcijama.....	8
4.4 Upiti s podupitima i skupovnim operacijama	8
5. Komentari	9
6. Indeksi	9
7. Procedure.....	10
8. Okidači	10

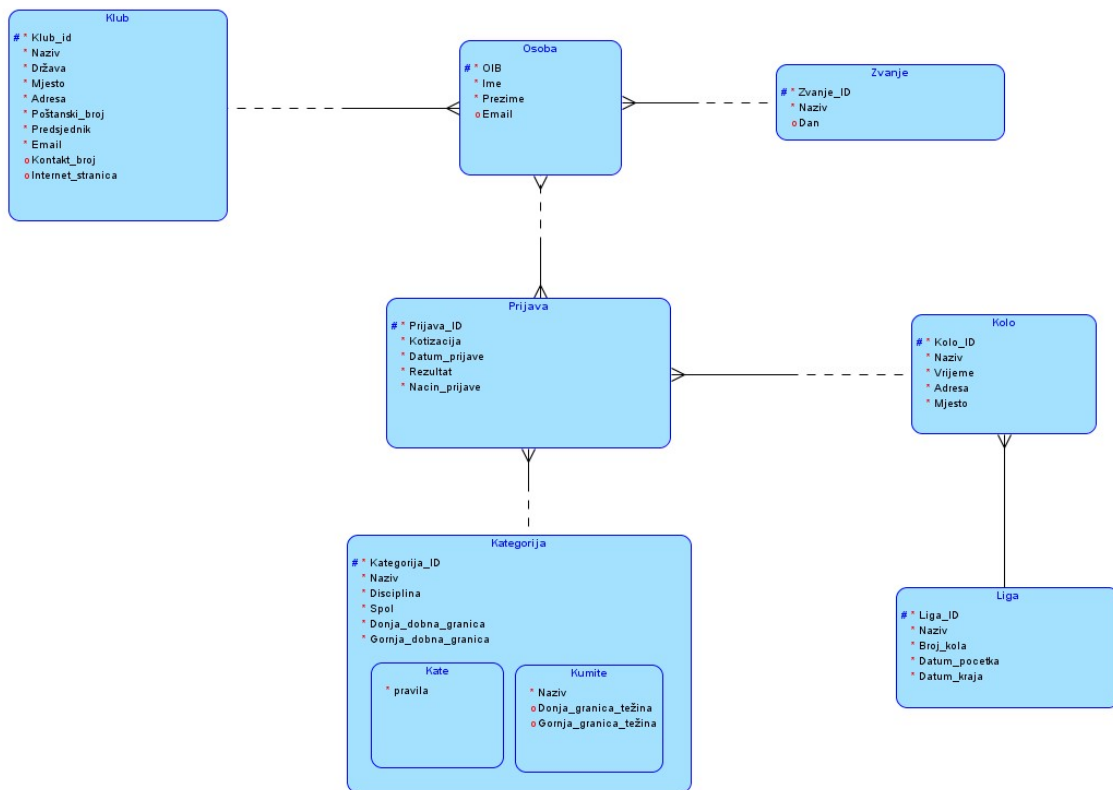
1.Uvod

Karate je drevna istočnjačka borilačka vještina, utemeljena na vrijednostima kao životnoj filozofiji. Prvenstveno je namijenjena samoobrani, a danas se karate trenira kao olimpijski sport.

S obzirom da sam godinama bila članica Karate kluba Podravski sokol, to me motiviralo da izradim karate bazu. Karate natjecanja su najčešće organizirana u tipu lige, stoga je moj cilj bio napraviti bazu za karate ligu, koja sadržava popis klubova, članova te rezultata na pripadnom kolu lige sa dodatnim informacijama o natjecateljima, klubovima i kategorijama.

2. Dijagrami

2.1 MEV



Slika: Model entiteta i veze

Model za bazu podataka karate lige sastoji se od više entiteta koji su povezani smislenim vezama.

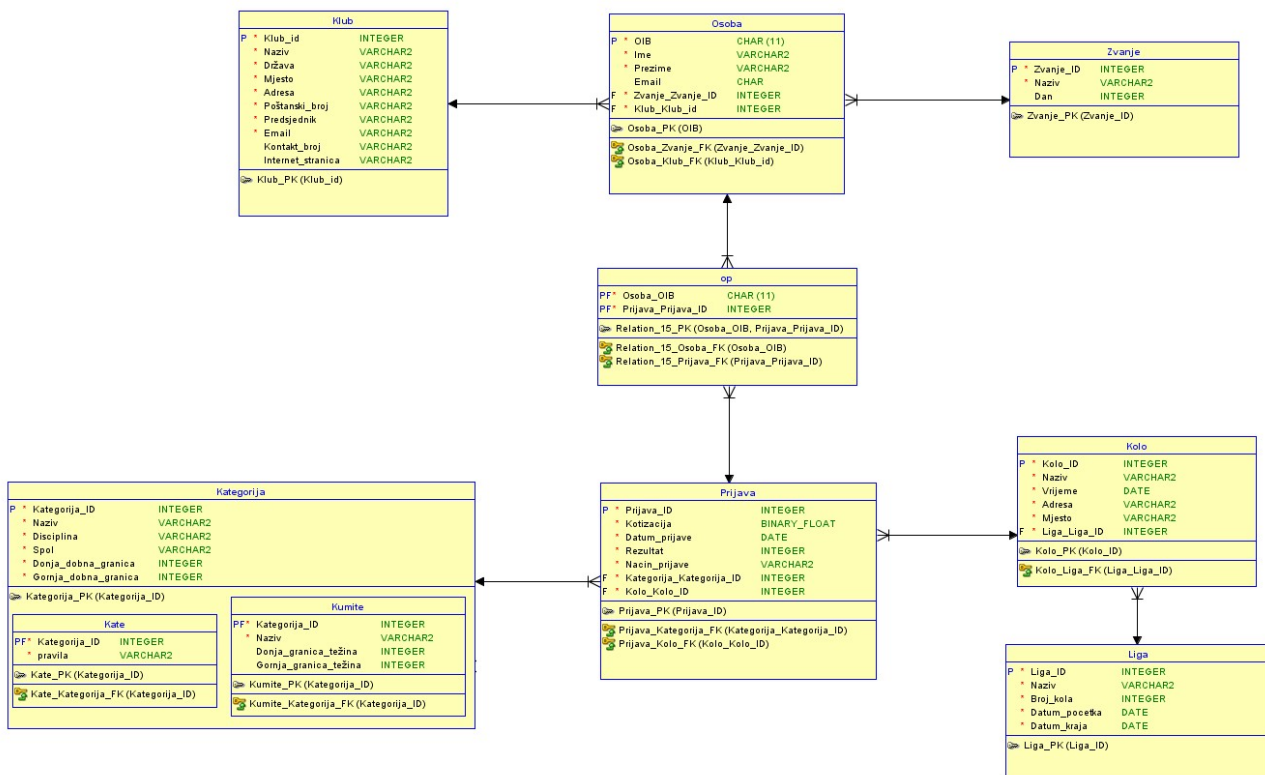
Natjecatelje razlikujemo prema njihovom OIB-u te za njih pratimo osnovne informacije poput imena, prezimena, a usput i klub kojem pripadaju te njihovo zvanje. U prijavi pratimo rezultat natjecatelja, a pratimo i kategoriju, kolo te ligu u kojoj se natjecao.

Veze koje povezuju entitete (sa kratkim objašnjenjem):

- Svaka osoba mora pripadati jednom klubu. (S obzirom da pratimo samo natjecatelje svaka osoba mora pripadati klubu)

- Klub može imati jednog ili više članova.
- Osoba mora imati zvanje. (Kada postanemo članovi kluba, odmah nosimo osnovno zvanje)
- Zvanje može imati jednu ili više osoba. (Više osoba može imati isto zvanje, ali neka zvanja ne mora nitko imati)
- Osoba može imati jednu ili više prijava. (Osoba se može natjecati u više kategorija)
- Prijava mora imati jednu ili više osoba (Ako prijava ima više osoba onda imamo grupu.)
- Prijava mora imati jedno kolo.
- Kolo može imati jednu ili više prijava.
- Liga mora imati jedno ili više kola.
- Kolo mora pripadati jednoj ligi.
- Prijava mora imati jednu kategoriju.
- Kategorija može imati jednu ili više prijava.

2.2 Relacijski model



Slika: relacijski model

U relacijskom modelu su detaljno prikazane sve tablice sa pripadnim atributima.

Tablica osoba prikazuje podatke o natjecatelju, kao što su ime, prezime, te mail koji je opcionalan. Ona sadrži strani ključ Klub_id koji je primarni ključ tablice Klub te tako dohvaćamo kojem klubu osoba pripada. Također na isti način dohvaćamo i zvanje koje je osoba stekla.

Tablica op je nastala kao rezultat više prema više veze između tablice Osoba i tablice Prijava. Stoga, ona sadrži attribute Osoba_OIB i Prijava_Prijava_ID koji su njeni strani, ali ujedno i primarni ključevi.

Tablica prijava sadrži podatke o prijavi (način prijave, datum, kotizacija) te u njoj pratimo rezultat. Ona ima strani ključ Kategorija_Kategorija_ID preko kojeg je povezana s Kategorijom, te strani ključ Kolo_Kolo_ID preko kojeg se povezana s tablicom Kolo. Preko stranih ključeva dohvaćamo kategoriju i kolo za pripadajuću prijavu.

U tablici kolo, svi su atributi obavezni. Ona je preko stranog ključa Liga_Liga_ID povezana sa tablicom Liga, čiji je primarni ključ Liga_ID. Povezana je tako, da bi za svako kolo znali kojoj ligi pripada. Tablica liga sadrži datume početka i kraja natjecanja, naziv te broj kola.

Tablica kategorija ima primarni ključ Kategorija_ID. Ona sadrži informacije o kategorijama. S obzirom da karate ima dvije discipline: kate i kumite, a one imaju različite kategorije no sa ipak nekim istim obilježjima (atributima), tablica Kategorija napravljena je kao nadtip dok su tablica Kate i tablica Kumite njeni podtipovi. Za obje discipline potrebno je pratiti naziv, spol, donju i gornju dobnu granicu pa su one navedene kao atributi u Kategoriji. No, svaka disciplina ima i neka svoja obilježja koja su zasebno stavljena u podtipove. Za Kate je potrebno pratiti dodatna pravila, a za kumite se mora pratiti i donja te gornja granica težine.

3.Kreiranje baze

Prema iznad opisanom relacijsko modelu kreirala sam tablice. Neki primjeri tablica:

```
-----ZVANJE-----
CREATE TABLE zvanje(
  zvanje_id INTEGER CONSTRAINT zvanje_pk PRIMARY KEY,
  naziv VARCHAR2(20) NOT NULL,
  dan INTEGER
);

-----KLUB-----
CREATE TABLE klub(
  klub_id INTEGER CONSTRAINT klub_pk PRIMARY KEY,
  naziv VARCHAR2(60) NOT NULL,
  drzava VARCHAR2(20) NOT NULL,
  mjesto VARCHAR2(30) NOT NULL,
  adresa VARCHAR2(60) NOT NULL,
  postanski_broj VARCHAR2(40) NOT NULL,
  predsjednik VARCHAR2(40) NOT NULL,
  email VARCHAR2(30) NOT NULL,
  kontakt_broj VARCHAR2(60),
  internetska_stranica VARCHAR2(60)
);

-----LIGA-----
CREATE TABLE liga(
  liga_id INTEGER CONSTRAINT liga_pk PRIMARY KEY,
  naziv VARCHAR2(40) NOT NULL,
  broj_kola INTEGER NOT NULL,
  datum_pocetka DATE NOT NULL,
  datum_kraja DATE NOT NULL
);

-----KATEGORIJA-----
CREATE TABLE kategorija(
  kategorija_id INTEGER CONSTRAINT kategorija_pk PRIMARY KEY,
  naziv VARCHAR2(60) NOT NULL,
  disciplina VARCHAR2(7) NOT NULL,
  spol VARCHAR2(2) NOT NULL,
  donja_dobna_granica INTEGER NOT NULL,
  gornja_dobna_granica INTEGER NOT NULL
);

-----KATE-----
CREATE TABLE kate(
  kategorija_id INTEGER CONSTRAINT kate_kategorija_pk
    PRIMARY KEY REFERENCES kategorija(kategorija_id),
  pravila VARCHAR2(120) NOT NULL
);

-----KUMITE-----
CREATE TABLE kumite(
  kategorija_id INTEGER CONSTRAINT kumite_kategorija_pk
    PRIMARY KEY REFERENCES kategorija(kategorija_id),
  naziv VARCHAR2(20) NOT NULL,
  donja_granica_tezina INTEGER,
  gornja_granica_tezina INTEGER
);

-----KOLO-----
CREATE TABLE kolo(
  kolo_id INTEGER CONSTRAINT kolo_pk PRIMARY KEY,
  naziv VARCHAR2(40) NOT NULL,
  vrijeme DATE NOT NULL,
  adresa VARCHAR2(40) NOT NULL,
  mjesto VARCHAR2(20) NOT NULL,
  liga_id INTEGER CONSTRAINT kolo_fk_liga_id REFERENCES liga(liga_id)
);

-----PRIJAVA-----
CREATE TABLE prijava(
  prijava_id INTEGER CONSTRAINT prijava_pk PRIMARY KEY,
  kotizacija BINARY_FLOAT NOT NULL,
  datum_prijave DATE NOT NULL,
  rezultat INTEGER NOT NULL,
  nacin_prijave VARCHAR2(10) NOT NULL,
  kategorija_id INTEGER CONSTRAINT prijava_fk_kategorija
    REFERENCES kategorija(kategorija_id),
  kolo_id INTEGER CONSTRAINT prijava_fk_kolo_id
    REFERENCES kolo(kolo_id)
);

-----OSOBA-----
CREATE TABLE osoba(
  oib CHAR(11) CONSTRAINT osoba_pk PRIMARY KEY,
  ime VARCHAR2(20) NOT NULL,
  prezime VARCHAR2(20) NOT NULL,
  email CHAR(30),
  zvanje_id INTEGER CONSTRAINT osoba_fk_zvanje REFERENCES zvanje(zvanje_id),
  klub_id INTEGER CONSTRAINT osoba_fk_klub REFERENCES klub(klub_id)
);

-----OP-----
CREATE TABLE op(
  oib CHAR(11) NOT NULL CONSTRAINT op_fk_osoba REFERENCES osoba(oib),
  prijava_id INTEGER NOT NULL CONSTRAINT op_fk_prijava REFERENCES prijava(prijava_id),
  CONSTRAINT op_pk PRIMARY KEY(oib,prijava_id)
);
```

3.1. Zadane vrijednosti

Nakon toga, dodala sam neke zadane vrijednosti.

```
ALTER TABLE zvanje  
MODIFY dan DEFAULT 0;
```

Ako nije unesen atribut dan u tablici zvanje onda dan postavi na nula.

```
ALTER TABLE klub  
MODIFY internetska_stranica DEFAULT 'http://karate.hr/web/klubovi.php';
```

Ako nije ništa uneseno u atribut internetska_stranica u tablici klub onda postavi link od HKS-a jer tamo su navedene stranice svih klubova.

3.2. Uvjeti

Kako bi spriječila neke krive unose podataka kod unosa podataka naredbom INSERT dodala sam uvjete:

```
ALTER TABLE kategorija ADD CONSTRAINT kategorija_spol_ck CHECK(spol in ('M', 'Ž'));  
ALTER TABLE kategorija ADD CONSTRAINT kategorija_disciplina_ck CHECK(disciplina in ('kate', 'kumite'));
```

U tablicu kategorija u atribut spol, uvjet omogućava upisati samo M ili Ž, za ostalo izbacuje pogrešku, kako ne bi imali neželjeni unos.

Primjer pogreške kod krivog unosa. Uvjet kategorija_disciplina_ck

```
SQL> INSERT INTO kategorija VALUES (15, 'Seniori kumite', 'komite', 'M', 18, 35);  
  
INSERT INTO kategorija VALUES (15, 'Seniori kumite', 'komite', 'M', 18, 35)  
*  
ERROR at line 149:  
ORA-02290: check constraint (DOSMANOVIC.KATEGORIJA_DISCIPLINA_CK) violated
```

Dodala sam još neke uvjete kako bi unosi bili što točniji.

```
ALTER TABLE prijava ADD CONSTRAINT prijava_nacin_ck CHECK(nacin_prijave in ('online', 'uživo'));  
ALTER TABLE kategorija ADD CONSTRAINT kategorija_donja_dobna_granica_ck CHECK(donja_dobna_granica > 5);
```


Nakon toga započela sam sa unošenjem podataka u tablice. Primjeri unosa.

```
INSERT INTO osoba VALUES ('888888888888', 'Antun', 'Tun', 'tuntun@gmail.com', 6, 1);
INSERT INTO osoba VALUES ('999999999999', 'Grga', 'Čokolin', 'gcoko@hotmail.com', 9, 1);
INSERT INTO osoba VALUES ('12121212121', 'Juraj', 'Milić', 'jmilic@gmail.com', 7, 1);
INSERT INTO osoba VALUES ('13131313131', 'Pavao', 'Pandić', 'ppandic@hotmail.com', 8, 1);
INSERT INTO osoba VALUES ('15151515151', 'Iva', 'Tadić', 'itadic@gmail.com', 10, 1);
```

4. Upiti

Navesti ću primjere svakog upita.

4.1 Jednostavni upiti

Upit: Ispis lokacije održavanja 1.kola natjecanja.

```
SELECT adresa, mjesto AS "Lokacija"
FROM kolo
WHERE kolo_id = 1;
```

Rezultat upita:

ADRESA	Lokacija
Samostanski prilaz 8	Daruvar

4.2 Složeni upiti

Upit: ispisati sve natjecatelje iz kategorije borbi učenici, sa njihovim rezultatima, po prezimenu.

```
SELECT o.ime || ' ' || o.prezime AS "Ime prezime", p.rezultat
FROM osoba o, op, prijava p, kategorija k, kumite ku
WHERE o.oib = op.oib AND op.prijava_id = p.prijava_id AND p.kategorija_id = k.kategorija_id
AND k.kategorija_id = ku.kategorija_id AND k.kategorija_id = 3
ORDER BY o.prezime;
```

Rezultat upita:

Ime prezime	REZULTAT
Nicolas Janssens	6
Ranko Kovač	3
Matija Marić	7
Juraj Milić	8
Robert Petkov	0
Antun Tun	6
Borna Župan	1

4.3 Upiti s agregirajućim funkcijama

Upit: ispisati ukupnu naplaćenu kotizaciju.

```
SELECT SUM(p.kotizacija) AS "Ukupna kotizacija"  
FROM prijava p;
```

Rezultat upita:

Ukupna kotizacija
7700

4.4 Upiti s podupitima i skupovnim operacijama

Upit: ispisati klub_id klubova koji nemaju ni jednog predstavnika u disciplini kate.

```
SELECT klub_id FROM klub  
MINUS  
SELECT DISTINCT klub_id FROM osoba  
WHERE oib = ANY (SELECT oib FROM op  
WHERE prijava_id = ANY(SELECT prijava_id FROM prijava  
WHERE kategorija_id = ANY(1,2,5,6,9,10)));
```

Koristila sam funkciju minus. U prvom dijelu prije funkcije MINUS napravila sam upit koji ispisuje klub_id od svih unesenih klubova, a poslije minusa upit koji ispisuje klub_id svih klubova koji imaju predstavnika u disciplini kate. Kada sam primjenila

funkciju minus, dobila sam upravo id klubova koji nemaju ni jednog predstavnika u disciplini kate.

Rezultat upita:

KLUB_ID
4
5

5.Komentari

Kako bi olakšala razumjevanje same baze, dodala sam komentare na tablice u kojima se nalaze kratki opisi tablica.

```
COMMENT ON TABLE kategorija IS 'Opis svih kategorija koje su dostupne.';
COMMENT ON TABLE kate IS 'Dodatni podatci o svakoj kategoriji koja pripada disciplini kate.';
COMMENT ON TABLE kumite IS 'Dodatni podatci o svakoj kategoriji koja pripada disciplini kumite.';
COMMENT ON TABLE zvanje IS 'Dostupna zvanja/pojasevi.';
COMMENT ON TABLE klub IS 'Podatci informativnog karaktera o svakom klubu.';
COMMENT ON TABLE liga IS 'Informativni podatci o ligi.';
COMMENT ON TABLE kolo IS 'Informativni podatci o kolu';
COMMENT ON TABLE prijava IS 'Podatci o prijavi';
COMMENT ON TABLE osoba IS 'Podatci o osobi';
COMMENT ON TABLE op IS 'Poveznica između osobe (oib) i prijave (prijava_id)';
```

6.Indeksi

Kako bi ubrzala pretragu dodala sam neke indekse.

```
CREATE INDEX i_prijava_rezultat ON prijava(rezultat);
CREATE INDEX i_osoba_naziv ON osoba(ime, prezime);
```

S obzirom da stupac rezultat sadrži velik broj različitih vrijednosti odlučila sam dodati indeks kako bi ubrzala pretraživanje.

```
CREATE BITMAP INDEX i_nacin_prijave ON prijava(nacin_prijave);
CREATE BITMAP INDEX i_spol_kategorija ON kategorija(spol);
```

7.Procedure

Procedura za unaprjeđenje zvanja. Zvanja se ne mogu preskakti, odnosno kod svakog polaganja zvanje se može povećati samo za 1, stoga sam napravila proceduru koja trenutno zvanje samo poveća za 1 (SET zvanje_id = zvanje_id + 1)

```
CREATE OR REPLACE PROCEDURE promjena_zvanja (  
    v_oib IN osoba.oib % TYPE  
    ) AS  
    v_brojac INTEGER;  
BEGIN  
    SELECT COUNT(*) INTO v_brojac FROM osoba WHERE oib = v_oib;  
  
    IF v_brojac = 1 THEN  
        UPDATE osoba  
        SET zvanje_id = zvanje_id + 1  
        WHERE oib = v_oib;  
        COMMIT;  
    ELSE  
        ROLLBACK;  
    END IF;  
END promjena_zvanja;  
/
```

8.Okidači

Okidač koji upozorava ako se na prijava_id koju mi pokušavamo upisati, već neko nalazi. Te ako je broj ljudi u toj prijavi već 3.

```
CREATE OR REPLACE TRIGGER before_insert_op  
    BEFORE INSERT ON op  
    FOR EACH ROW  
DECLARE  
    brojac INTEGER;  
BEGIN  
    SELECT COUNT(*) INTO brojac FROM op WHERE prijava_id= :NEW.prijava_id; --postoji li ta prijava  
  
    IF brojac = 3 THEN  
        RAISE_APPLICATION_ERROR(-20202, 'Ova grupa već ima dovoljno članova, napravi novu prijavu!');  
    ELSIF brojac > 0 THEN  
        dbms_output.put_line('Na ovom prijava_id je već netko osim Vas upisan. Sada je ovo grupna prijava.');    END IF;  
END;  
/
```