

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Kristijan Čulina

PROJEKT IZRADE BAZE PODATAKA

BAZE PODATAKA II

Varaždin, 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ź D I N

Kristijan Čulina

Matični broj:

Studij: Informacijski sustavi

PROJEKT IZRADE BAZE PODATAKA

BAZE PODATAKA II

Mentor/Mentorica:

Prof. dr. sc. Kornelije Rabuzin

Varaždin, siječanj 2021.

Sadržaj

1. Uvod	1
2. Opis aplikacijske domene	2
3. ERA Model	3
4. Tablice	5
4.1. Tablica arena	5
4.2. Tablica borac	5
4.3. Tablica borba	6
4.4. Tablica bori_se	6
4.5. Tablica dogadaj	6
4.6. Tablica kamp	7
4.7. Tablica kategorija	7
4.8. Tablica menadzer	7
4.9. Tablica nadimak	7
4.10. Tablica nalazi_se	8
4.11. Tablica sudac	8
4.12. Tablica vodi	8
4.13. Tablica zemlja	9
5. Veze između tablica	10
5.1. Veza borac – nadimak	10
5.2. Veza borac – kategorija	10
5.3. Veza borac – kamp	10
5.4. Veza borac – menadzer	11
5.5. Veza borac – zemlja	11
5.6. Veza menadzer – zemlja	11
5.7. Veza sudac – zemlja	12
5.8. Veza arena – zemlja	12
5.9. Veza kamp – zemlja	12
5.10. Veza borac – borba	13
5.11. Veza borba – dogadaj	13
5.12. Veza borba – arena	13
5.13. Veza borba – sudac	14
6. Okidači	15
6.1. Okidač provjera_potpisivanja_ugovora	15
6.2. Okidač plavi_crveni	16
6.3. Okidač provjera_borbe_f	17

6.4. Okidač provjera_borbe_f2	18
7. Upiti	19
7.1. Upit sudac_zemlja.....	19
7.2. Upit menadzer_zemlja	19
7.3. Upit sudio_borbi	20
7.4. Upit menadzer_borac.....	20
7.5. Upit borac_nadimak_kategorija	21
7.6. Upit borac_nadimak_kategorija_godina	22
8. Alat za modeliranje i administraciju baze podataka	24
9. Opis tehnologija korištenih kod izrade projekta	27
10. Prikaz funkcionalnosti aplikacije.....	30
10.1. Unos	31
10.2. Ažuriranje	34
10.3. Brisanje	36
10.4. Dohvaćanje podataka iz baze	38
11. Zaključak	39
12. Prilozi.....	40
12.1. Slike.....	40
12.2. Tablice	42
13. Literatura	43

1. Uvod

U ovom projektu za zadatak sam imao izraditi bazu podataka te funkcionalno grafičko sučelje sa CRUD operacijama (CREATE, READ, UPDATE i DELETE). Bazu te grafičko sučelje kreirao sam za potrebe evidencije organizacije UFC, koja je najveća svjetska MMA (mixed martial arts) organizacija na čelu s predsjednikom Dana Whiteom.

Osnova svake baze podataka je dobar ERA model, bez njega apsolutno ništa nema smisla. Kreiranjem detaljnog ERA modela s prikazanim entitetima, njegovim atributima te relacijama između entiteta dobivamo širu sliku o samoj aplikaciji. ERA model sam kreirao samostalno u alatu draw.io, koji će te vidjeti odmah u nastavku ove dokumentacije.

Bazu podataka sam radio u MySQL-u, besplatnom sustavu za upravljanje bazom podataka a grafičko sučelje koje komunicira s bazom podataka sam kreirao u C#. Kako sam se na predmetima Baze I i Baze II jako dobro upoznao sa psql-om, prelazak na MySQL bio je lagan zadatak. Kod C# je apsolutno druga priča, nikad se na ovaj način do sad nisam susreo sa OOP, grafičkim sučeljem te mi je trebalo neko vrijeme da se priviknem na rad u takvom okruženju.

U projektnoj dokumentaciji ću prikazati izradu ERA modela uz opis svih entiteta, relacija i atributa. Također ću opisati alat za upravljanje bazom podataka, okidače i upite, tehnologije korištene pri izradi projekta te funkcionalnosti same aplikacije preko grafičkog sučelja.

Motivacija za izbor MySQL-a je bila ta što je jedan on najkorištenijih besplatnih sustava za upravljanje bazom podataka te se u poslovnom svijetu koristi jako često. C# imamo na predmetu (Programsko Inženjerstvo) koji dolazi sljedeći semestar te je ovo bila odlična prilika za upoznat se s Visual Studiom te samim C#-om. Motivacija za kreiranje ove aplikacije je ta što sam cijeli život u nekim borilačkim sportovima te sam veliki MMA fan.

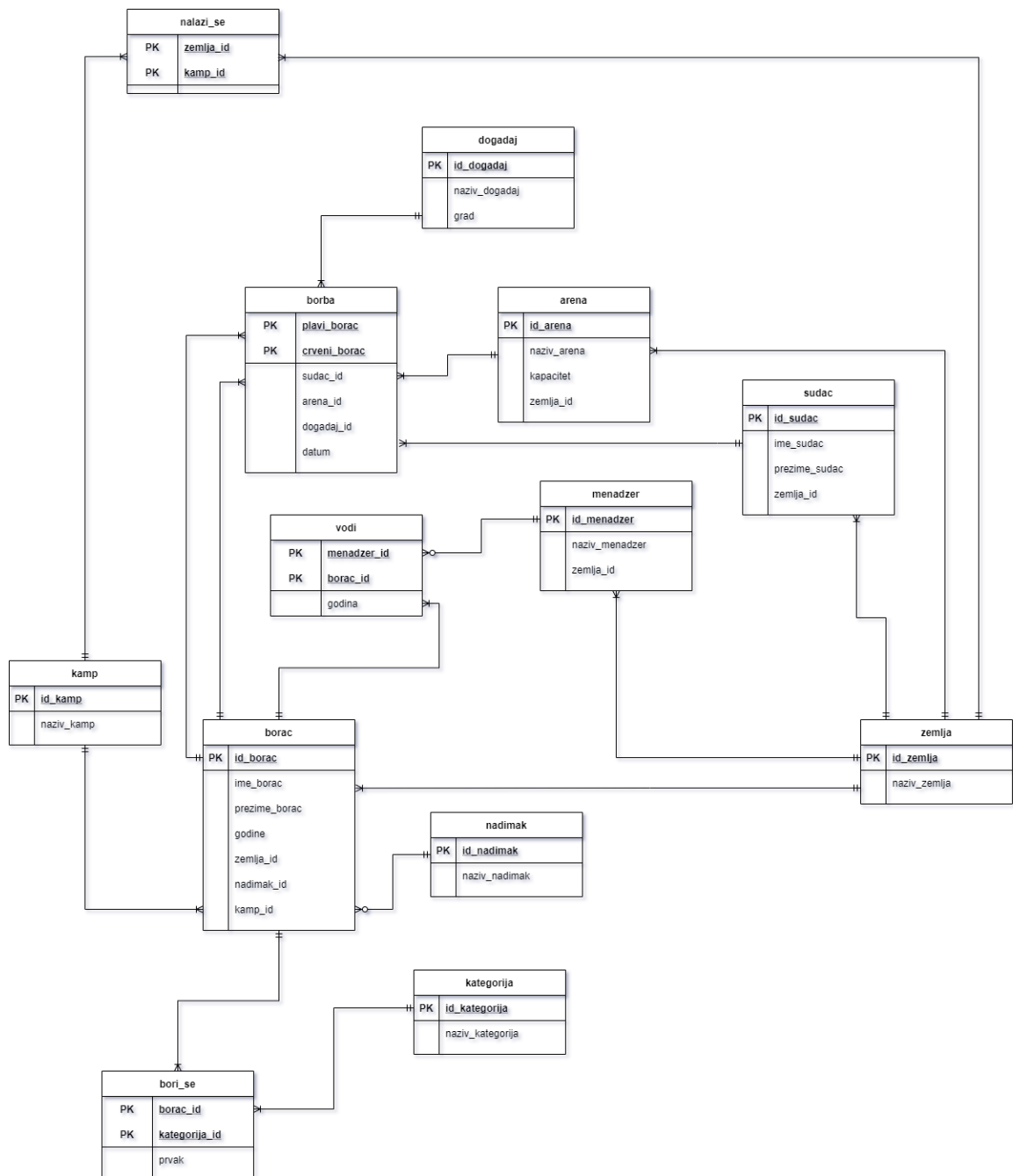
2. Opis aplikacijske domene

U ovom projektu implementirao sam bazu podataka za potrebe evidencije organizacije UFC. Sučelje za upravljanje podacima baze podataka je napravljeno u obliku Windows Form aplikacije. Baza podataka se sastoji od trinaest tablica od kojih je dvanaest jakih entiteta te jedna tablica koja je slabi entitet, koja služi kao implementacija veze više naprema više. Tablice koje su središte ove baze podataka su: *borac*, *borba*, *sudac*.

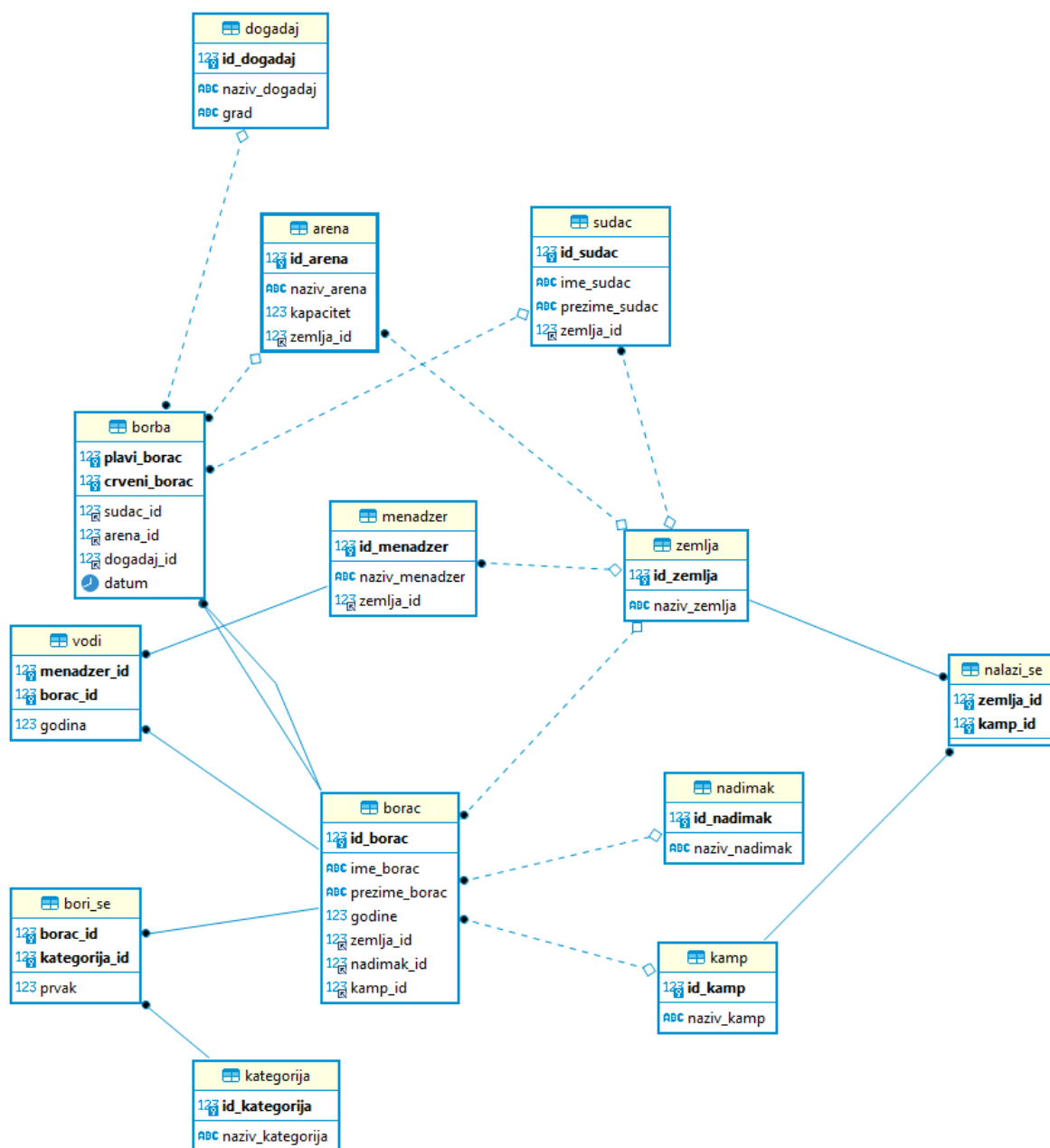
Aplikacija služi za upravljanje evidencijom UFC organizacije. Konkretno za potrebe ovog projekta ćemo moći upravljati tablicom *sudac*. Vrlo lako i brzo možemo to učiniti moguće i za ostalih dvanaest tablica. Osoba koja ima pristup aplikaciji moći će unositi nove sudce, ažurirati i/ili brisati postojeće sudce unutar baze podataka.

Aplikacija je vrlo funkcionalna za potrebe evidencije pojedinih borilačkih organizacija jer obuhvaća jako velik broj entiteta. Aplikaciju je moguće vrlo lako nadograditi i osposobiti za ostale organizacije no za potrebe ovog projekta to nije bilo potrebno.

3. ERA Model



Slika 1 ERA model [vlastoručna izrada, draw.io]



Slika 2 ERA model - generirani diagram iz Dbeavera

Na slici 1 vidimo vlastoručnu izradu ERA modela u alatu draw.io. Možemo vidjeti detaljne prikaze entiteta s njihovim atributima te relacijama. Također možemo detaljno vidjeti koje su središnje tablice te vrstu veza između pojedinih entiteta.

Kod prikaza na slici 2 vidimo generirani diagram iz alata Dbeaver, gdje također možemo vidjeti vrstu veza kao i tipove podataka pojedinih atributa.

4. Tablice

U nastavku su prikazane tablice s pripadajućim atributima te opisima atributa.

4.1. Tablica arena

NAZIV	TIP	OPIS
id_arena	int	primarni ključ
naziv_arena	varchar(100)	
kapacitet	int	
zemlja_id	int	vanjski ključ (zemlja)

Tablica 1 tablica arena

4.2. Tablica borac

NAZIV	TIP	OPIS
id_borac	Int	Primarni ključ
Ime_borac	Varchar(50)	
Prezime_borac	Varchar(50)	
Godine	Int	
Zemlja_id	Int	Vanjski ključ (zemlja)
Nadimak_id	Int	Vanjski ključ (nadimak)
Kamp_id	Int	Vanjski ključ (kamp)

Tablica 2 tablica borac

4.3. Tablica borba

NAZIV	TIP	OPIS
plavi_borac	int	dvokomponentni primarni ključ vanjski ključ (borac)
crveni_borac	int	dvokomponentni primarni ključ vanjski ključ (borac)
sudac_id	int	vanjski ključ (sudac)
arena_id	int	vanjski ključ (arena)
dogadjaj_id	int	vanjski ključ (dogadjaj)
datum	date	

Tablica 3 tablica borba

4.4. Tablica bori_se

NAZIV	TIP	OPIS
borac_id	int	dvokomponentni primarni ključ vanjski ključ (borac)
kategorija_id	int	dvokomponentni primarni ključ vanjski ključ (kategorija)
prvak	bool	

Tablica 4 tablica bori_se

4.5. Tablica dogadjaj

NAZIV	TIP	OPIS
Id_dogadjaj	Int	Primarni ključ
Naziv_dogadjaj	Varchar(50)	
Grad	Varchar(50)	

Tablica 5 tablica dogadjaj

4.6. Tablica kamp

NAZIV	TIP	OPIS
Id_kamp	Int	Primarni ključ
Naziv_kamp	Varchar(100)	

Tablica 6 tablica kamp

4.7. Tablica kategorija

NAZIV	TIP	OPIS
Id_kategorija	Int	Primarni ključ
Naziv_kategorija	Varchar(50)	

Tablica 7 tablica kategorija

4.8. Tablica menadzer

NAZIV	TIP	OPIS
Id_menadzer	Int	Primarni ključ
Naziv_menadzer	Varchar(80)	
Zemlja_id	Int	Vanjski ključ (zemlja)

Tablica 8 tablica menadzer

4.9. Tablica nadimak

NAZIV	TIP	OPIS
Id_nadimak	Int	Primarni ključ
Naziv_nadimak	Varchar(60)	

Tablica 9 tablica nadimak

4.10. Tablica nalazi_se

NAZIV	TIP	OPIS
Zemlja_id	int	dvokomponentni primarni ključ vanjski ključ (zemlja)
Kamp_id	Int	dvokomponentni primarni ključ vanjski ključ (kamp)

Tablica 10 tablica nalazi_se

4.11. Tablica sudac

NAZIV	TIP	OPIS
Id_sudac	Int	Primarni ključ
Ime_sudac	Varchar(100)	
Prezime_sudac	Varchar(100)	
Zemlja_id	Int	Vanjski ključ (zemlja)

Tablica 11 tablica sudac

4.12. Tablica vodi

NAZIV	TIP	OPIS
Menadzer_id	Int	dvokomponentni primarni ključ vanjski ključ (menadzer)
Borac_id	Int	dvokomponentni primarni ključ vanjski ključ (borac)
Godina	Int	

Tablica 12 tablica vodi

4.13. Tablica zemlja

NAZIV	TIP	OPIS
Id_zemlja	Int	Primarni ključ
Naziv_zemlja	Varchar(50)	

Tablica 13 tablica zemlja

5. Veze između tablica

5.1. Veza borac – nadimak

borac	*..1	nadimak
nadimak_id		Id_nadimak

Tablica 14 veza borac – nadimak

Veza između tablice borac i nadimak je više naprema jedan. Jedan borac ima samo jedan nadimak dok jedan nadimak može imati više boraca kao što je to slučaj da dva borca imaju isti nadimak – Cowboy.

5.2. Veza borac – kategorija

borac	1..*	bori_se
id_borac		borac_id

Tablica 15 veza borac - bori_se

kategorija	1..*	bori_se
id_kategorija		kategorija_id

Tablica 16 veza kategorija - bori_se

Veza između tablice borac i tablice kategorija je više naprema više. Dakle dolazimo do zaključka da nam treba još jedna pomoćna tablica. U ovom slučaju je to tablice bori_se koja ih spaja te su veze iz tablice borac i kategorija prema bori_se tipa jedan na prema više. Jedan borac se može boriti u više kategorija, primjer Jose Aldo se bori u Bantamweight kategoriji i Featherweight kategoriji. Naravno u jednoj kategoriji se bori više boraca.

5.3. Veza borac – kamp

borac	*..1	kamp
kamp_id		id_kamp

Tablica 17 veza borac – kamp

Veza između tablice borac i tablice kamp je više naprema jedan. Jedan borac pripada samo jednom kampu dok jednom kampu može pripadati više boraca.

5.4. Veza borac – menadzer

borac	1..*	vodi
id_borac		borac_id
<i>Tablica 18 veza borac – vodi</i>		
menadzer	1..*	vodi
id_menadzer		menadzer_id
<i>Tablica 19 veza menadzer – vodi</i>		

Veza između tablice borac i tablice menadzer je više naprema više te nam treba pomoćna tablica, u ovom slučaju tablica vodi. Veze iz tablice borac i menadzer prema vodi su jedan naprema više. Razlog veze više na više je taj što jedan borac može imati više menadžera a jedan menadžer isto tako može imati više boraca kojima upravlja.

5.5. Veza borac – zemlja

borac	*..1	zemlja
zemlja_id		id_zemlja
<i>Tablica 20 veza borac – zemlja</i>		

Veza između tablice borac i tablice zemlja je više naprema jedan. Jedan borac dolazi iz samo jedne zemlje dok iz jedne zemlje može dolaziti više ljudi. Konkretni primjer, Jose Aldo dolazi iz Brazila ali i Alex Oliveira dolazi isto iz Brazila.

5.6. Veza menadzer – zemlja

menadzer	*..1	zemlja
zemlja_id		id_zemlja
<i>Tablica 21 veza menadzer – zemlja</i>		

Veza između tablice menadzer i tablice zemlje je više naprema jedan. Jedan menadžer dolazi iz samo jedne zemlje, dok iz jedne zemlje može dolaziti više menadžera.

5.7. Veza sudac – zemlja

sudac	*..1	zemlja
zemlja_id		id_zemlja

Tablica 22 veza sudac – zemlja

Veza između tablice sudac i tablice zemlja je više naprema jedan. Jedan sudac dolazi iz jedne zemlje dok jednoj zemlji može pripadati više sudaca.

5.8. Veza arena – zemlja

arena	*..1	zemlja
zemlja_id		id_zemlja

Tablica 23 veza arena – zemlja

Veza između tablice arena i tablice zemlja je više naprema jedan. Jedna arena se nalazi u jednoj zemlji dok se u jednoj zemlji može nalaziti više arena.

5.9. Veza kamp – zemlja

kamp	1..*	nalazi_se
id_kamp		kamp_id

Tablica 24 veza kamp - nalazi_se

zemlja	1..*	nalazi_se
id_zemlja		zemlja_id

Tablica 25 veza zemlja - nalazi_se

Veza između tablice kamp i tablice zemlja je više naprema više. Iz tog razloga trebamo pomoćnu tablicu koju smo nazvali nalazi_se. Jedan kamp se može nalaziti u više zemlji, dok se u jednoj zemlji može nalaziti više kampova. Primjer da se jedan kamp nalazi u više zemlji je primjer American Top Team-a koji se nalazi u Hrvatskoj te u SAD-u, dok se u SAD-u nalazi i kamp Team Oyama.

5.10. Veza borac – borba

borac	2..*	borba
id_borac		plavi_borac
id_borac		crveni_borac

Tablica 26 veza borac – borba

Veza između tablice borac i tablice borba je dva naprema više. Naime jedan borac se može boriti na više borbi, dok se na jednoj borbi mogu boriti samo dva borca. Pod jednom borbom smatramo borbu između 2 borca, da nebi bilo zabune.

5.11. Veza borba – događaj

borba	*..1	Događaj
događaj_id		id_događaj

Tablica 27 veza borba – događaj

Veza između tablice borba i tablice događaj je više naprema jedan. Razlog tome je taj što se jedna borba odvija na samo jednom događaju dok se na jednom događaju može i odvija više borbi. Događaj (event) se obično sastoji od petnaestak borbi koje traju kroz određeni period te je svaka borba unikatna i pripada samo tom događaju.

5.12. Veza borba – arena

borba	*..1	arena
arena_id		id_arena

Tablica 28 veza borba - arena

Veza između tablice borba i tablice arena je više naprema jedan. Jedna borba se odvija u jednoj areni, dok se u jednoj areni može odvijati više borbi.

5.13. Veza borba – sudac

borba	*..1	sudac
sudac_id		id_sudac

Tablica 29 veza borba – sudac

Veza između tablice borba i tablice sudac je više naprema jedan. Jednu borbu sudi samo jedan sudac dok jedan sudac može i obično sudi više različitih borbi.

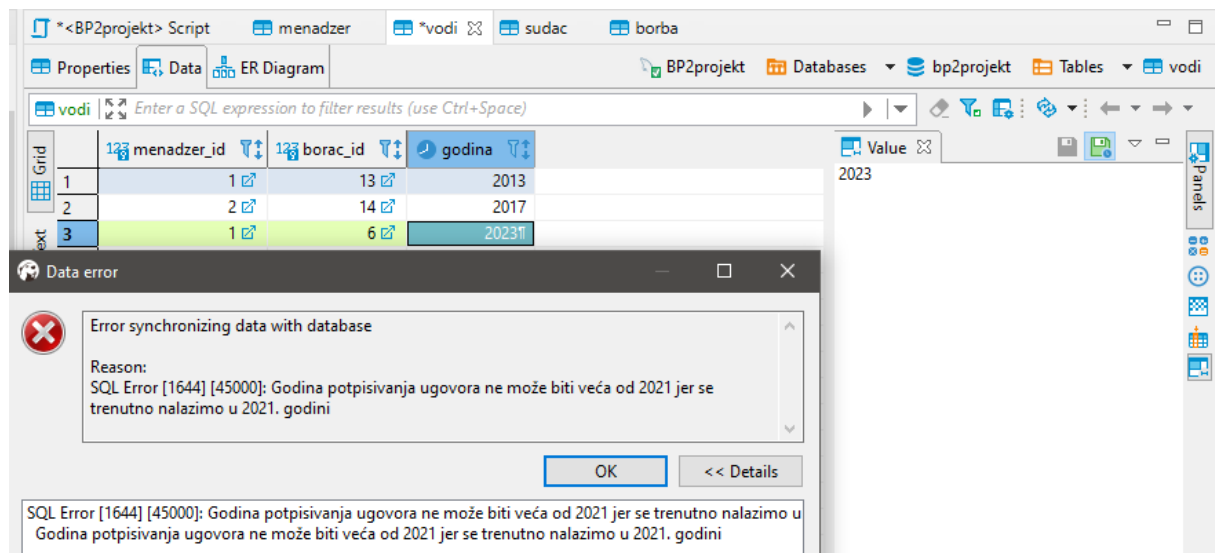
6. Okidači

Prilikom kreiranja baze podataka odlučio sam postaviti neka ograničenja da spriječe korisnika u zlim namjerama kojim bi naudio našoj evidenciji. Ograničenja su jako dobra stvar za očuvanje prvotnog plana aplikacije ali isto tako i za očuvanje živčanog sustava koji bi nam otišao prilikom popravljivanja stvari koje su neki zli ljudi odlučili napraviti.

6.1. Okidač provjera_potpisivanja_ugovora

```
create trigger provjera_potpisivanja_ugovora before insert on vodi for each row
begin
    if new.godina > 2021 then
        signal sqlstate '45000'
        set message_text = 'Godina potpisivanja ugovora ne može biti veća od 2021
jer se trenutno nalazimo u 2021. godini';
    end if;
end;
```

Radi se o jednostavnom okidaču koji sprječava korisnika u zlim namjerama, odnosno sprječava da godina potpisivanja ugovora između borca i menadžera bude veća od trenutne godine, odnosno 2021. Da okidač uspješno radi možemo vidjeti u prilogu koji se nalazi ispod:

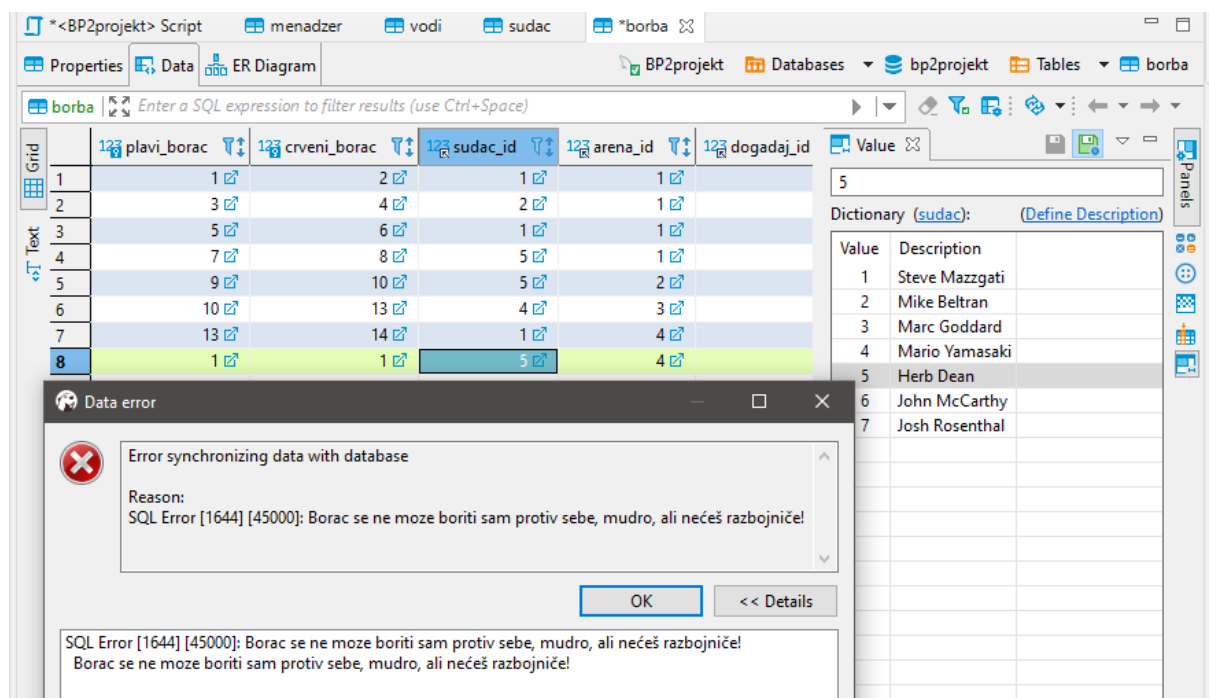


Slika 3 rezultat okidača provjera_potpisivanja_ugovor

6.2. Okidač plavi_crveni

```
create trigger plavi_crveni before insert on borba for each row begin
    if new.crveni_borac = new.plavi_borac then
        signal sqlstate '45000'
        set message_text = 'Borac se ne moze boriti sam protiv sebe, mudro, ali
nećeš razbojniče!';
    end if;
end;
```

Ovaj jako lijep upit nam omogućava sigurnost tako da ne dozvoli korisniku da prilikom unosa borbe unese da se borac bori protiv samog sebe, mislim stvarno, kome bi to palo na pamet... Da okidač uspješno obavlja svoju dužnost možemo vidjeti na slici ispod:

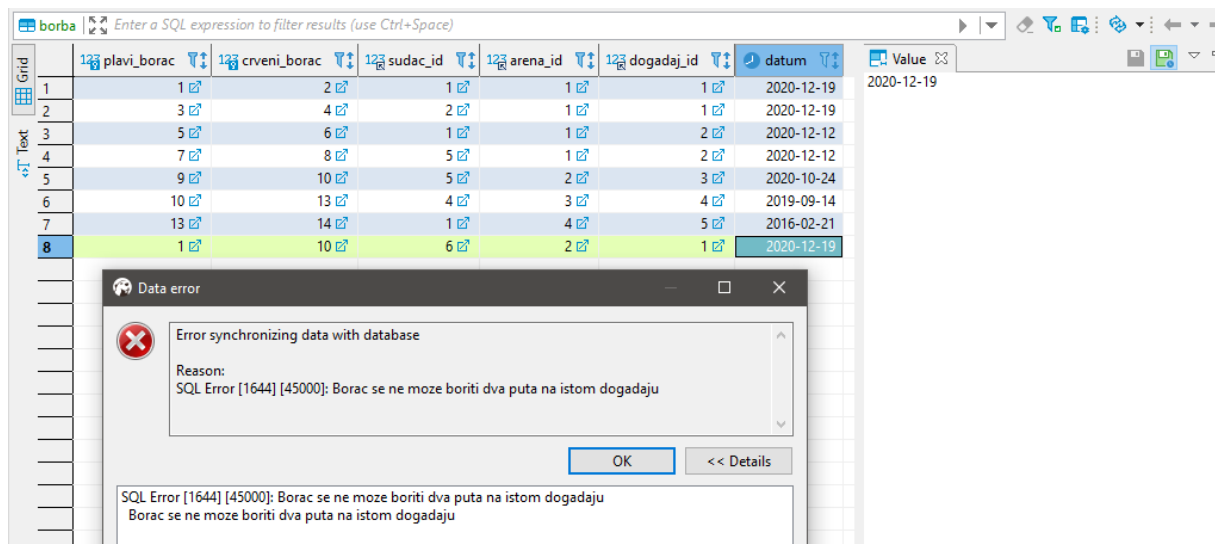


Slika 4 rezultat okidača plavi_crveni

6.3. Okidač provjera_borbe_f

```
create trigger provjera_borbe_f before insert on borba for each row begin
    declare iborac int1;
    select count(plavi_borac) into iborac from borba, događaj, borac where
new.događaj_id = id_događaj
and new.plavi_borac = borac.id_borac;
    if iborac > 1 then
        signal sqlstate '45000'
        set message_text = 'Borac se ne može boriti dva puta na istom događaju ';
    end if;
end;
```

Ovaj okidač sprječava da se borac bori više puta na istom događaju. Da trigger obavlja svoj posao možemo vidjeti na slici koja se nalazi ispod:

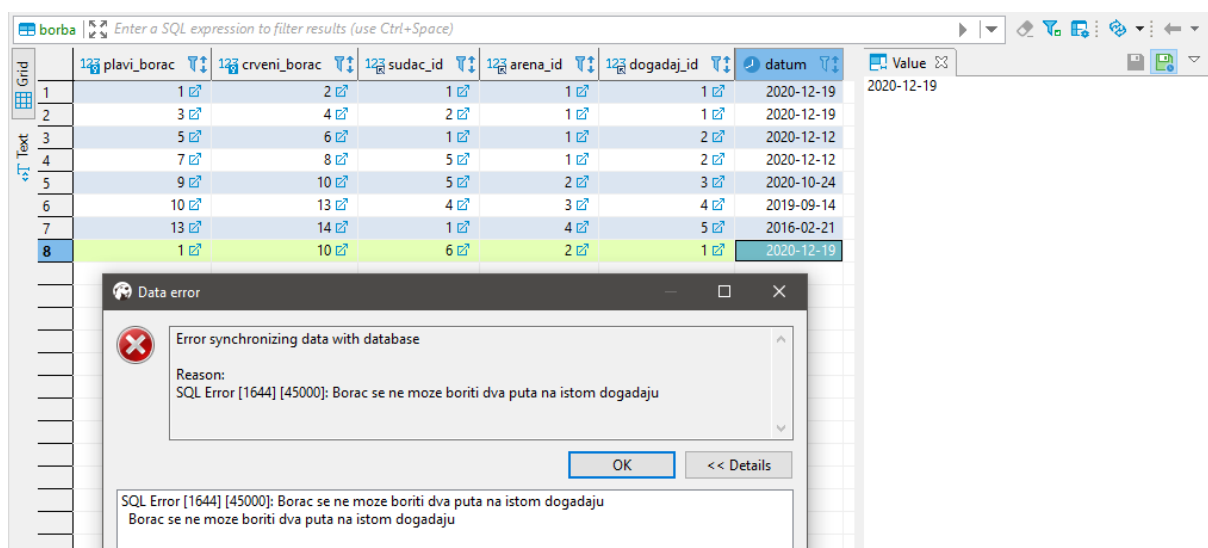


Slika 5 rezultat okidača provjera_borbe_f

6.4. Okidač provjera_borbe_f2

```
create trigger provjera_borbe_f2 before insert on borba for each row begin
    declare iborac int1;
    select count(crveni_borac) into iborac from borba, događaj, borac where
new.događaj_id = id_događaj
and new.crveni_borac = borac.id_borac;
    if iborac > 1 then
        signal sqlstate '45000'
        set message_text = 'Borac se ne može boriti dva puta na istom događaju ';
    end if;
end;
```

Ovaj okidač radi isto to samo za crvenog borca, odnosno drugog borca, da bi kompletno zaštitili aplikaciju. Rezultat možemo vidjeti također dole ispod s istim rezultatima:



Slika 6 rezultat okidača provjera_borbe_f2

7. Upiti

Prilikom kreiranja baze podataka kreirao sam i par upita, nešto jednostavnih i nešto složenijih. Dao sam mašti na volju prilikom kreiranja složenijih upita. U nastavku ću ih priložiti sve te ću ih dodatno objasniti. Također ću uz svaki upit priložiti rezultat upita, tj ono što je upit vratio nama nazad.

7.1. Upit sudac_zemlja

```
SELECT * from sudac ORDER BY `zemlja_id` ASC
```

Ovaj jednostavni upit dohvaća sve podatke iz tablice sudac te ih sortira uzlazno po stupcu zemlja_id.

Rezultat ovog upita možemo vidjeti na sljedećem prikazu:

	123 id_sudac	ABC ime_sudac	ABC prezime_sudac	123 zemlja_id
1	1	Steve	Mazzgati	1
2	2	Mike	Beltran	1
3	5	Herb	Dean	1
4	6	John	McCarthy	1
5	7	Josh	Rosenthal	1
6	3	Marc	Goddard	3
7	4	Mario	Yamasaki	9

Slika 7 rezultat upita sudac_zemlja

7.2. Upit menadzer_zemlja

```
SELECT * from menadzer WHERE zemlja_id = 1
```

Ovaj također jednostavni upit dohvaća sve podatke iz tablice menadzer gdje je zemlja_id = 1, odnosno dohvaća sve menadžere koji dolaze iz SAD-a.

	123 id_menadzer	ABC naziv_menadzer	123 zemlja_id
1	1	Kevin Harvick	1
2	4	Sayif Saud	1
3	8	Lou Di Bono	1

Slika 8 rezultat upita menadzer_zemlja

7.3. Upit sudio_borbi

```
select sudac.ime_sudac, sudac.prezime_sudac, count(borba.sudac_id) as 'sudiodorbi' from sudac
join borba on borba.sudac_id = id_sudac group by 1,2 order by 3 DESC;
```

Ovaj složeni upit dohvaća ime i prezime sudca te broj borbi koje su sudili ukupno i sortira ih po broju suđenih borbi silazno. Rezultat ovog upita možemo vidjeti:

	ABC ime_sudac	ABC prezime_sudac	123 sudio borbi
1	Steve	Mazzgati	3
2	Herb	Dean	2
3	Mario	Yamasaki	1
4	Mike	Beltran	1

Slika 9 rezultat upita sudio_borbi

7.4. Upit menadzer_borac

```
select menadzer.naziv_menadzer, borac.ime_borac, borac.prezime_borac, vodi.godina
from vodi
left join menadzer on menadzer.id_menadzer = vodi.menadzer_id
left join borac on borac.id_borac = vodi.borac_id
order by 4 ASC;
```

Upit menadzer_borac je još jedan složeni upit kojim dohvaćamo naziv menadžera te ime, prezime borca kojeg on vodi te godinu početka suradnje poredanih po godini uzlazno. Rezultat vidimo na sljedećoj slici:

	ABC naziv_menadzer	ABC ime_borac	ABC prezime_borac	123 godina
1	Kevin Harvick	Donald	Cerrone	2,013
2	Ali Abdelaziz	Khabib	Nurmagomedov	2,014
3	Ali Abdelaziz	Justin	Gaethje	2,016
4	Titus Day	Robert	Whittaker	2,016
5	Alex Davis	Alex	Olivera	2,017
6	Wallid Ismail	Deiveson	Figueiredo	2,017
7	Sayif Saud	Geoff	Neal	2,018
8	Lou Di Bono	Jared	Cannonier	2,018
9	Diego Lima	Charles	Olivera	2,019

Slika 10 rezultat upita menadzer_borac

7.5. Upit borac_nadimak_kategorija

```
select borac.ime_borac, borac.prezime_borac, nadimak.naziv_nadimak,
kategorija.naziv_kategorija from bori_se
join borac on borac.id_borac = bori_se.borac_id
join kategorija on kategorija.id_kategorija = bori_se.kategorija_id
join nadimak on borac.nadimak_id = nadimak.id_nadimak
order by 4;
```

Složeni upit borac_nadimak_kategorija vraća ime i prezime te nadimak borca i kategoriju u kojoj se bori, sortirano po nazivu kategorije. Rezultat ovog upita možemo vidjeti na slici ispod:

	ABC ime_borac	ABC prezime_borac	ABC naziv_nadimak	ABC naziv_kategorija
1	Jose	Aldo	Junior	Bantamweight
2	Marlon	Vera	Chito	Bantamweight
3	Jose	Aldo	Junior	Featherweight
4	Marlon	Vera	Chito	Featherweight
5	Charles	Olivera	Do Bronx	Featherweight
6	Deiveson	Figueiredo	God of war	Flyweight
7	Brandon	Moreno	The Assasin Baby	Flyweight
8	Charles	Olivera	Do Bronx	Lightweight
9	Tony	Ferguson	El Cucuy	Lightweight
10	Khabib	Nurmagomedov	The Eagle	Lightweight
11	Justin	Gaethje	The Highlight	Lightweight
12	Donald	Cerrone	Cowboy	Lightweight
13	Alex	Olivera	Cowboy	Lightweight
14	Robert	Whittaker	The Reaper	Middleweight
15	Jared	Cannonier	The Killa Gorilla	Middleweight
16	Stephen	Thompson	Wonderboy	Welterweight
17	Geoff	Neal	Handz of steel	Welterweight
18	Charles	Olivera	Do Bronx	Welterweight
19	Tony	Ferguson	El Cucuy	Welterweight
20	Donald	Cerrone	Cowboy	Welterweight

Slika 11 rezultat upita borac_nadimak_kategorija

7.6. Upit borac_nadimak_kategorija_godina

```

select borac.ime_borac, borac.prezime_borac, nadimak.naziv_nadimak,
kategorija.naziv_kategorija, vodi.godina
from bori_se
join borac on borac.id_borac = bori_se.borac_id
join nadimak on borac.nadimak_id = nadimak.id_nadimak
join kategorija on kategorija.id_kategorija = bori_se.kategorija_id
join vodi on borac.id_borac = vodi.borac_id
where vodi.godina > 2015
order by 5;

```

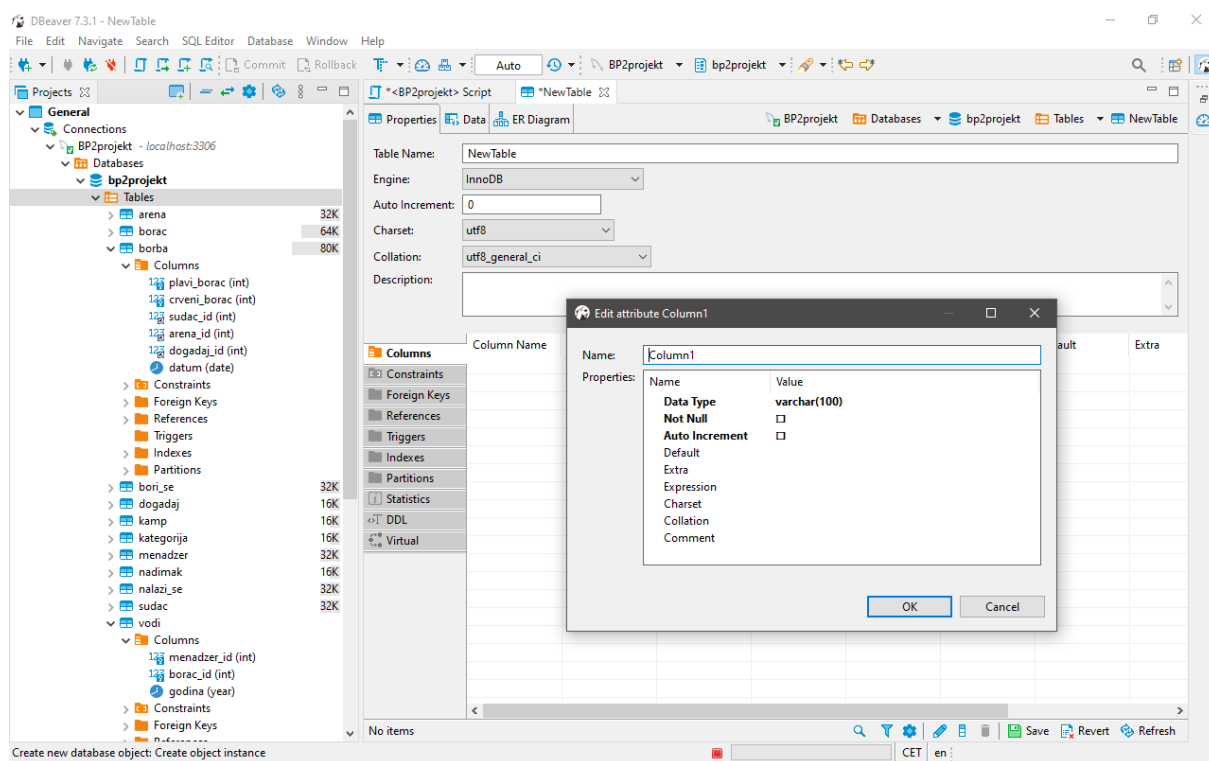
Složeni upit borac_nadimak_kategorija_godina je nadograđena verzija prethodnog upita, malo proširena i još malo zakomplicirana. Upit vraća ime, prezime i nadimak borca, kategoriju kojoj pripada te godinu kad je potpisao ugovor s menadžerom te godina mora biti veća od 2015. Rezultat je prikazan na sljedećoj slici:

	ABC ime_borac 🔼🔽	ABC prezime_borac 🔼🔽	ABC naziv_nadimak 🔼🔽	ABC naziv_kategorija 🔼🔽	🕒 godina 🔼🔽
1	Justin	Gaethje	The Highlight	Lightweight	2016
2	Robert	Whittaker	The Reaper	Middleweight	2016
3	Alex	Olivera	Cowboy	Lightweight	2017
4	Deiveson	Figueiredo	God of war	Flyweight	2017
5	Geoff	Neal	Handz of steel	Welterweight	2018
6	Jared	Cannonier	The Killa Gorilla	Middleweight	2018
7	Charles	Olivera	Do Bronx	Lightweight	2019
8	Charles	Olivera	Do Bronx	Welterweight	2019
9	Charles	Olivera	Do Bronx	Featherweight	2019

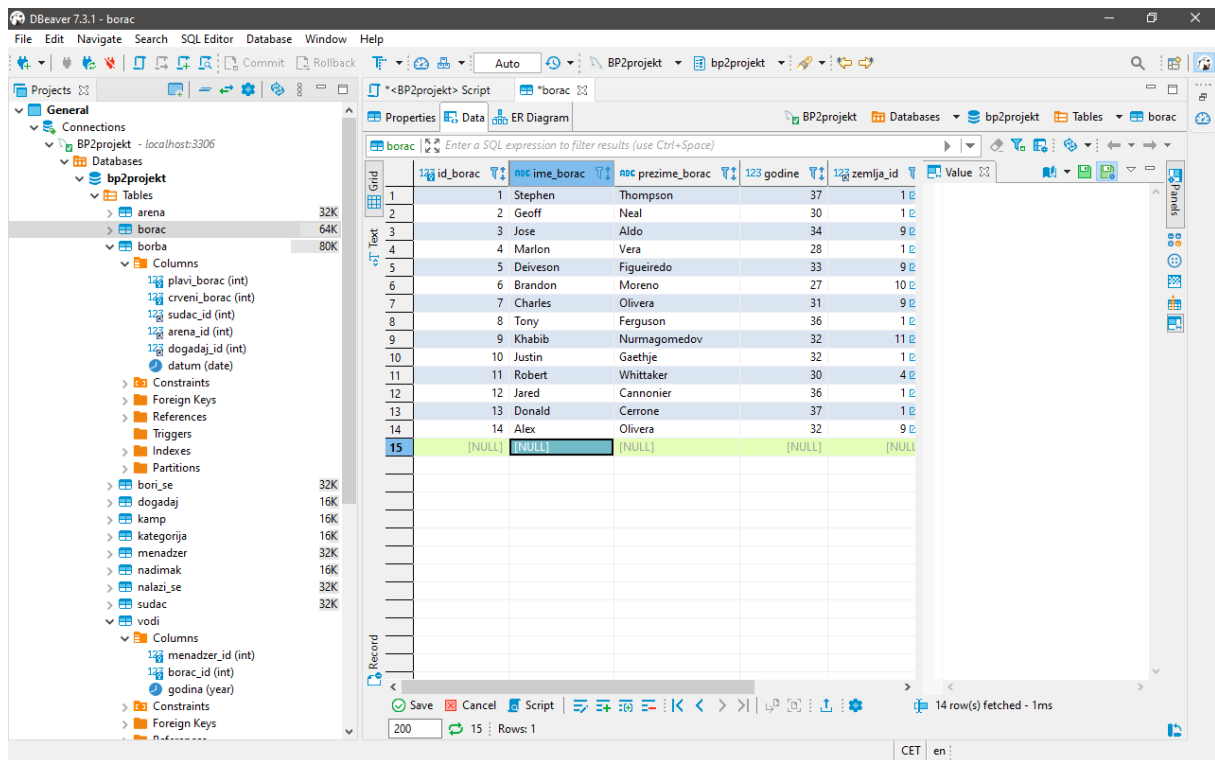
Slika 12 rezultat upita borac_nadimak_kategorija_godina

8. Alat za modeliranje i administraciju baze podataka

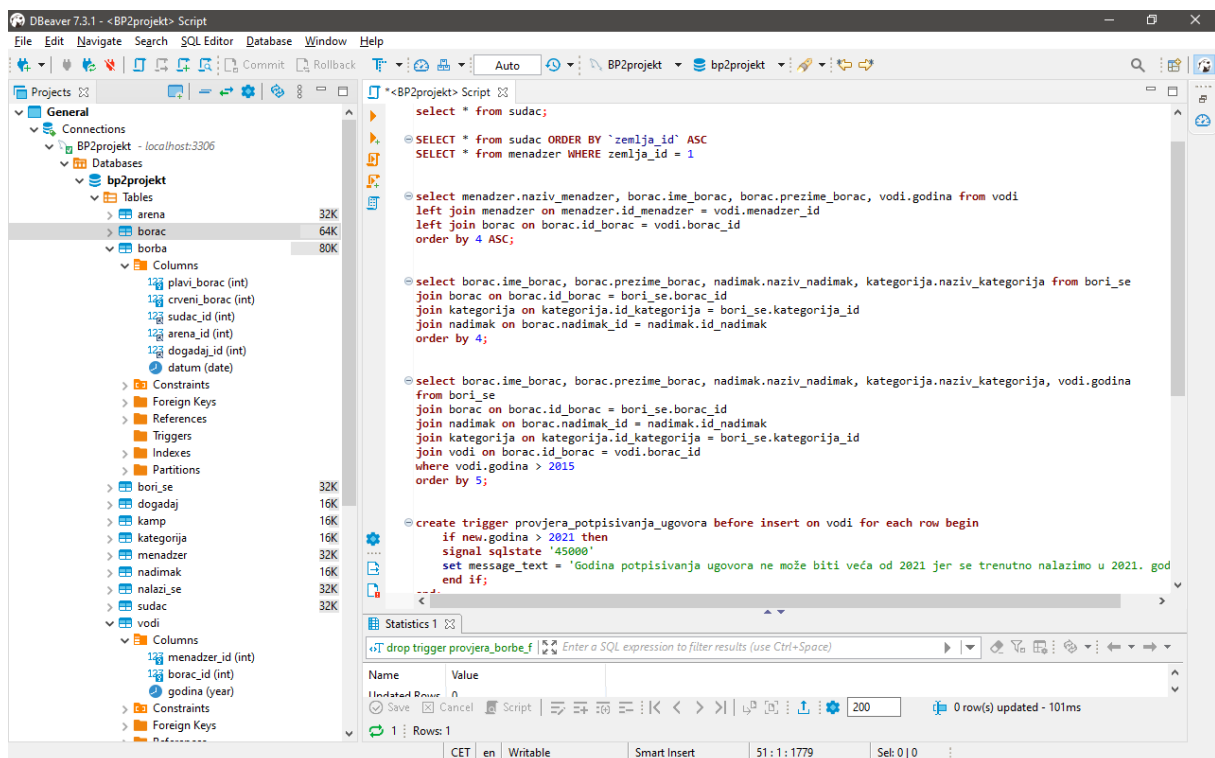
Alat koji sam koristio za modeliranje i administraciju baze podataka je DBeaver. Kako nismo smjeli koristiti već upoznati DataGrip koji smo koristili na labosima, odlučio sam se za također popularan alat DBeaver. Jedan od glavnih razloga zašto sam ga odabrao je taj što je jako light, odnosno ne zauzima previše mjesta na disku. Također alat obavlja sve potrebne funkcije te je naravno besplatan što je jako veliki plus. Preko njega sam se sa svojim korisničkim imenom i lozinkom povezao na lokalnu bazu podataka te u njemu kreirao, uređivao i popunjavao tablice zajedno sa ograničenjima. Upute i okidače sam također radio u samom alatu te nemam nikakvih primjedbi. Alat je vrlo jednostavan za korištenje, vrlo malo vremena mi je bilo potrebno da se u potpunosti stopim s njim. Također, kako sam već naveo, alat ima sve potrebne funkcije koje omogućuju modeliranje i administraciju baze podataka. Radi izvrsno te detaljno i jasno prikazuje ERA model. Zaista sam zadovoljan odabirom alata te ću u nastavku pokazati kako je izgledao rad u DBeaveru kroz par slika.



Slika 13 [DBeaver] kreiranje tablice



Slika 14 [DBeaver] unos podataka



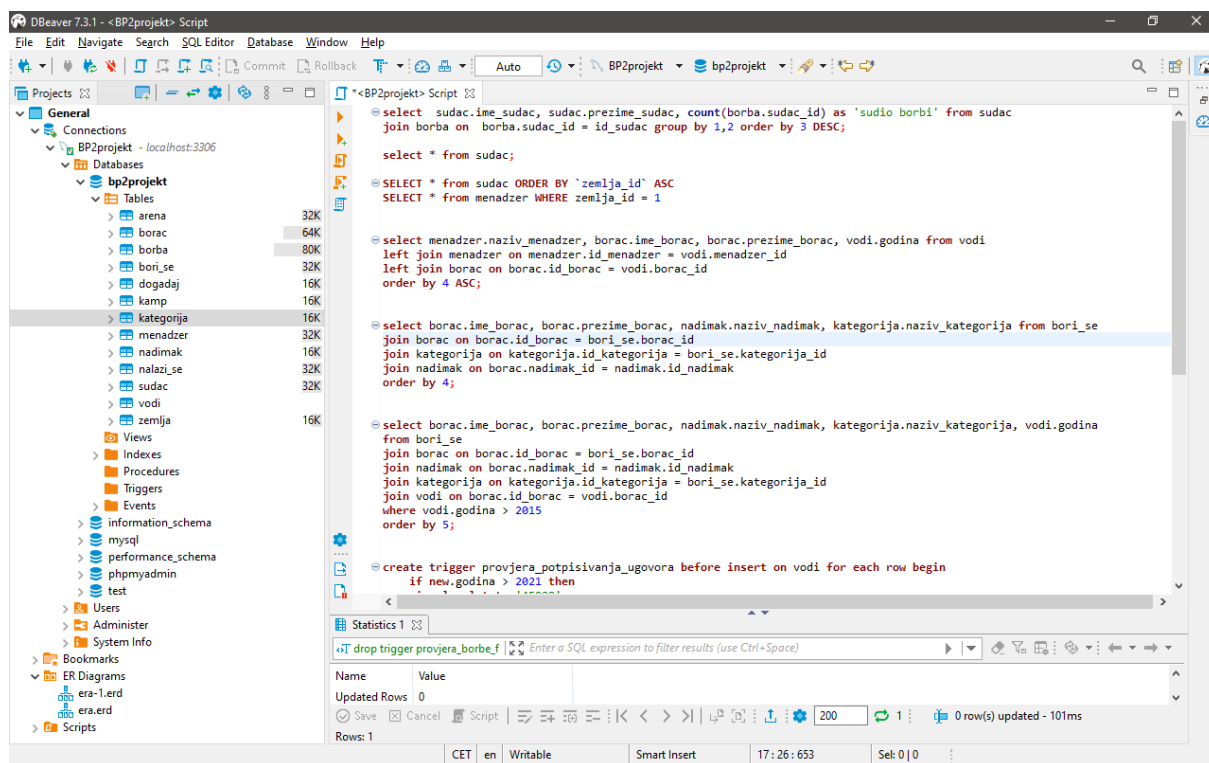
Slika 15 [DBeaver] rad unutar alata



9. Opis tehnologija korištenih kod izrade projekta

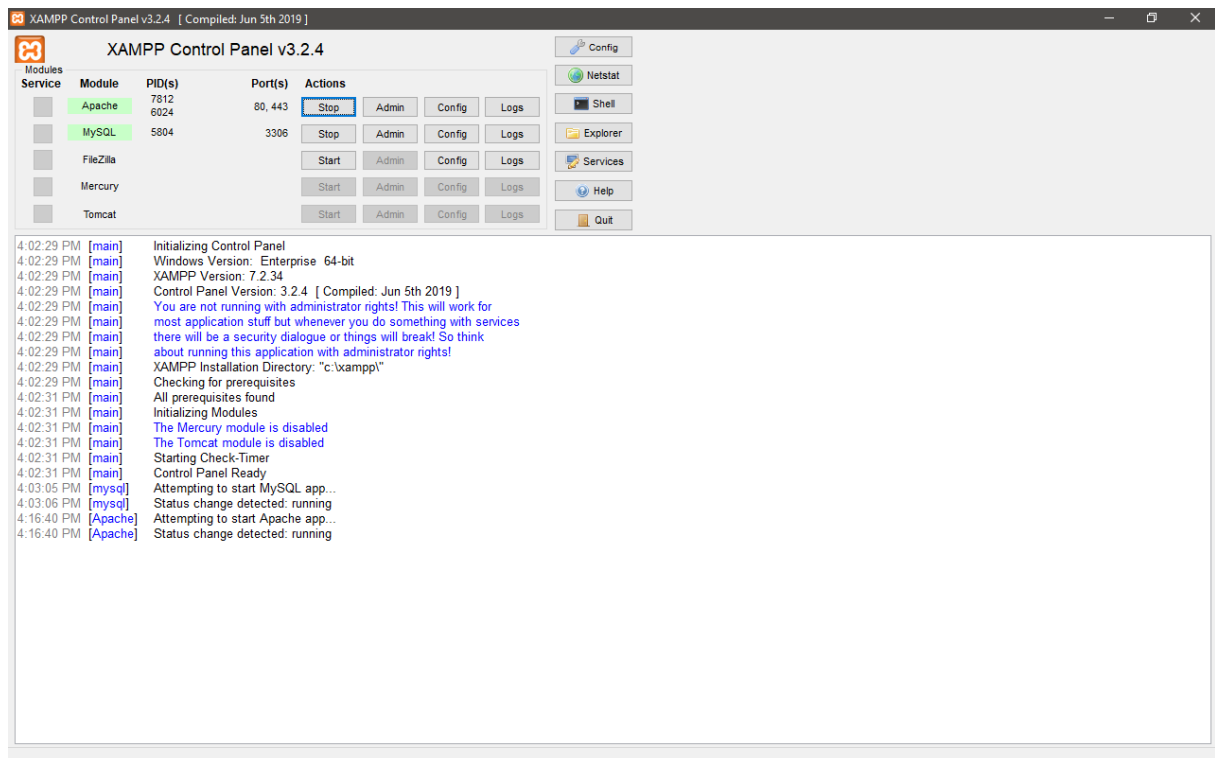
Kod izrade projekta koristio sam razne tehnologije. Koristio sam DBeaver, XAMPP, Microsoft Visual Studio. Također sam koristio Microsoft Word za pisanje ove dokumentacije te Paint za obrezivanje snimki zaslona. U tekstu ću se dotaknuti svake od korištenih tehnologija te ću za svaku napisati par rečenica.

Kao što sam i gore napisao, DBeaver sam koristio kao alat za modeliranje i administraciju bazom podataka.



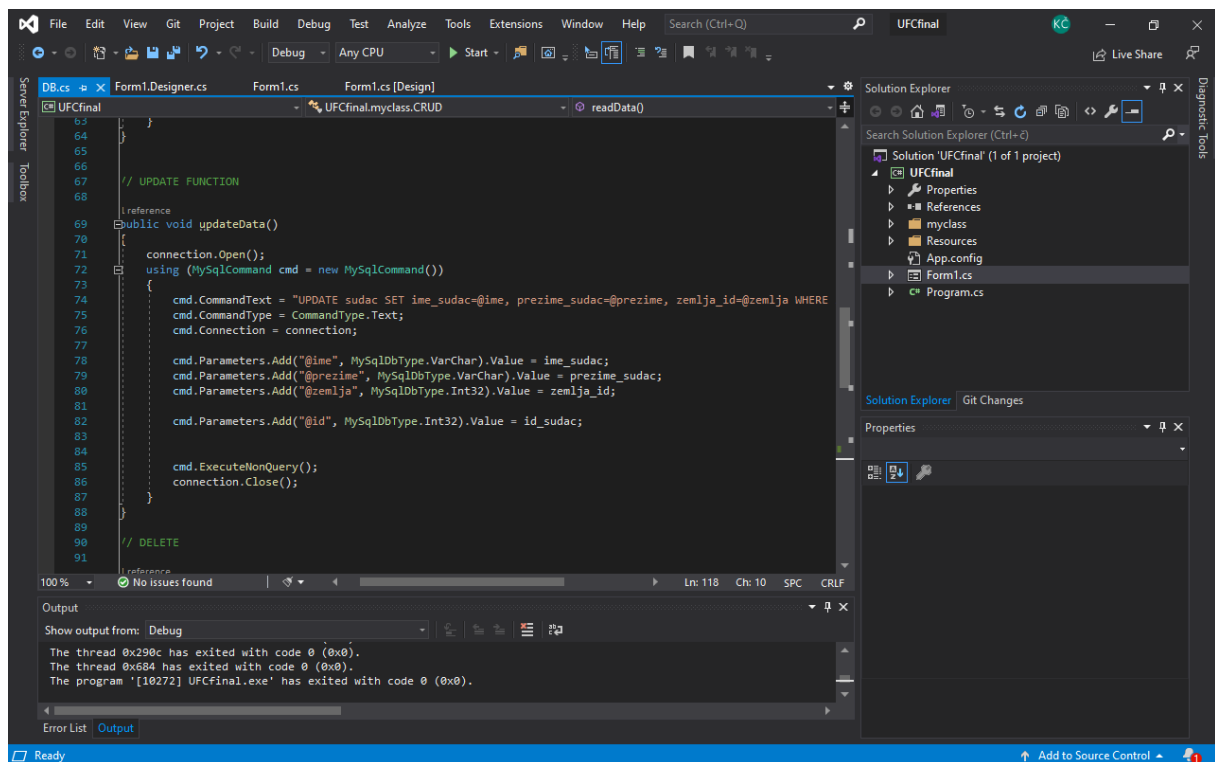
Slika 17 [DBeaver]

Za podizanje i kreiranje MySQL servera bio mi je potreban XAMPP Control Panel. Osim što je besplatan, također je i vrlo jednostavan za korištenje.

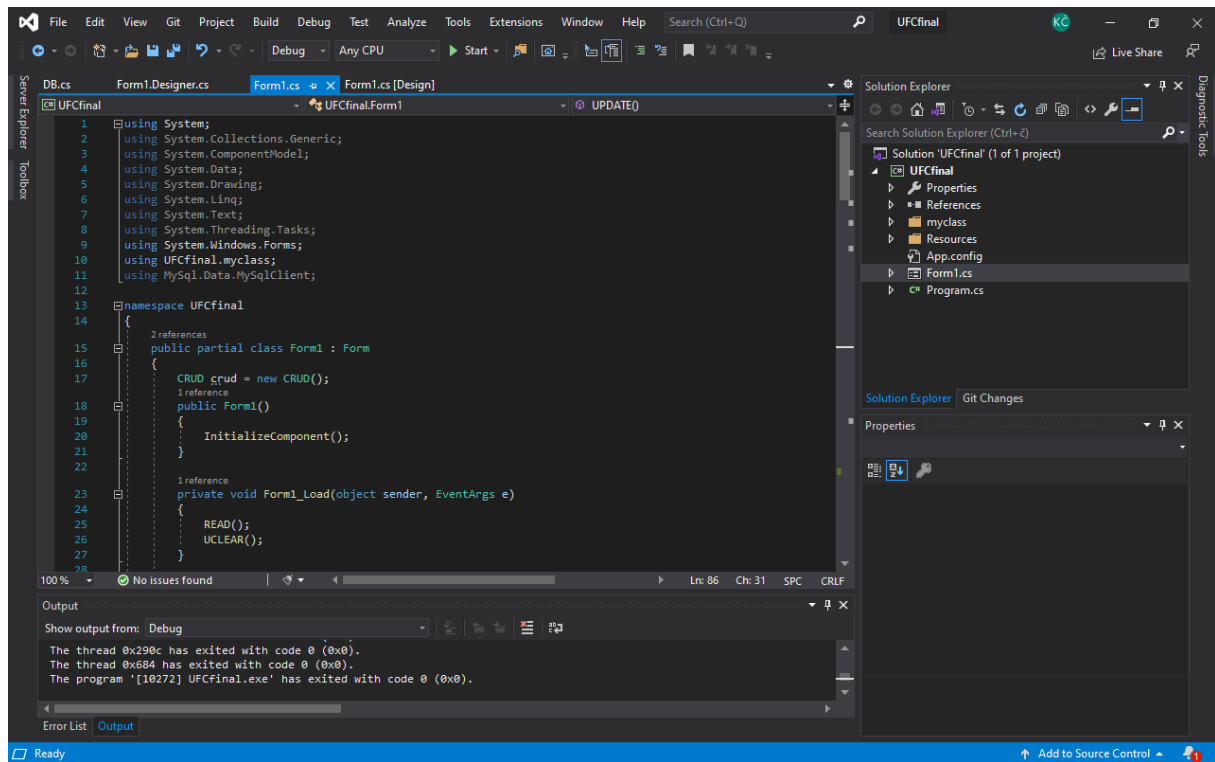


Slika 18 [XAMPP]

Grafičko sučelje je kreirano pomoću C# programskog jezika kao Winows Form Application. Kod je napisan u alatu Microsoft Visual Studio.



Slika 19 [Microsoft Visual Studio]



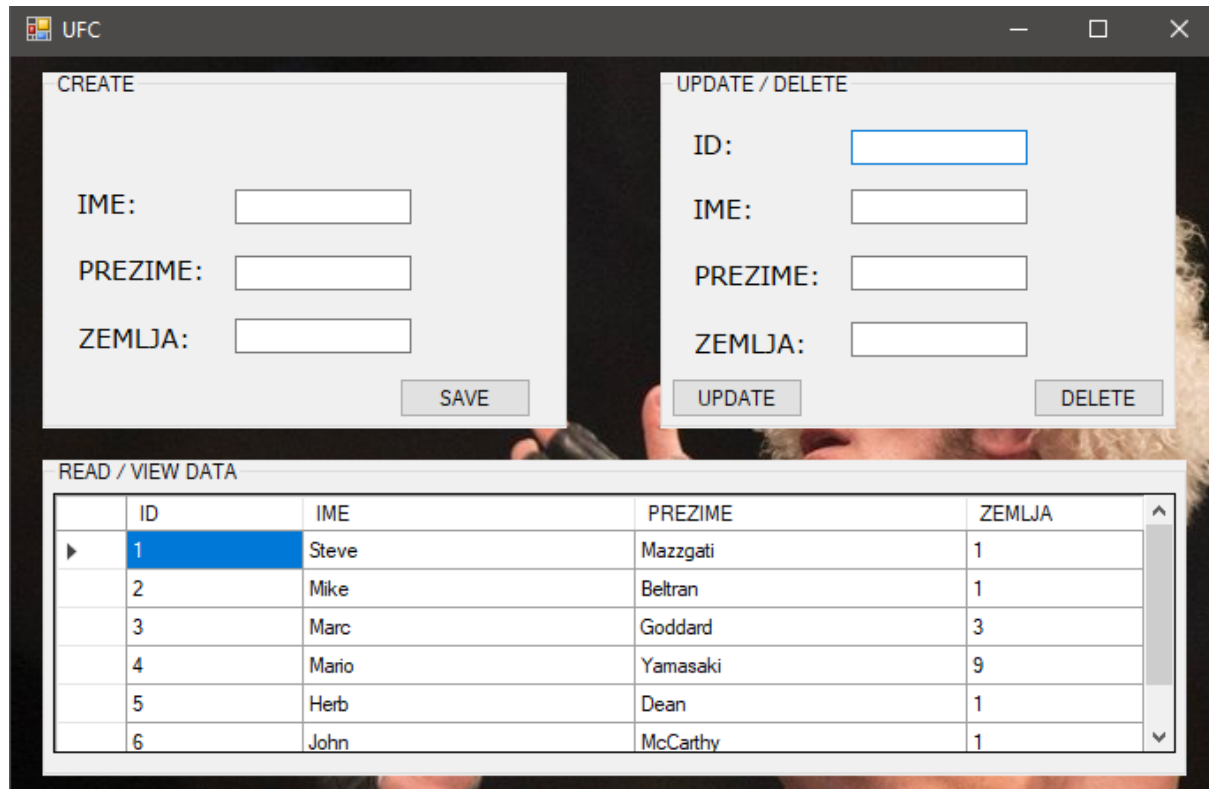
Slika 20 [Microsoft Visual Studio]

10. Prikaz funkcionalnosti aplikacije

Sučelje je kreirano kao Windows Form Application u alatu Microsoft Visual studio a sav kod je pisan u jeziku C#.

Prilikom pokretanja aplikacije, nalazimo se na početnoj stranici, ujedno i jednoj stranici radi jednostavnosti i funkcionalnosti aplikacije. Budući da nam je projektni zadatak bio da imamo funkcionalno grafičko sučelje koje komunicira s bazom podataka odlučio sam izvoditi CRUD operacije nad tablicom sudac. Aplikacija je lako prilagodljiva i za ostale tablice ako je potrebna cjelokupna evidencija no za potrebe ovog projekta sam demonstrirao rad na jednoj tablici.

Operacije koje možemo obavljati nad tablicom sudac su Create, odnosno kreiranje novog sudca, Read, dohvaćanje podataka o svim sudcima koji se nalaze u našoj bazi podataka, Update, ažuriranje vrijednosti iz tablice i Delete, odnosno brisanje podataka iz tablice sudac.



The screenshot shows a Windows application window titled "UFC". It contains three main sections:

- CREATE:** A form with three text boxes labeled "IME:", "PREZIME:", and "ZEMLJA:", and a "SAVE" button.
- UPDATE / DELETE:** A form with three text boxes labeled "ID:", "IME:", "PREZIME:", and "ZEMLJA:", and "UPDATE" and "DELETE" buttons.
- READ / VIEW DATA:** A table displaying a list of judges.

	ID	IME	PREZIME	ZEMLJA
▶	1	Steve	Mazzgati	1
	2	Mike	Beltran	1
	3	Marc	Goddard	3
	4	Mario	Yamasaki	9
	5	Herb	Dean	1
	6	John	McCarthy	1

Slika 21 pokretanje aplikacije

Prilikom pokretanja aplikacije, odmah možemo vidjeti da su dohvaćenje vrijednosti za sve sudce koji se trenutno nalaze u našoj bazi podataka. Na pozadini aplikacije je naravno najveća ikona u povijesti MMA, a to je Khabib Nurmagomedov. Krenuo bih pisati par riječi o njemu no bojim se da će se to pretvoriti u esej, iz tog razloga možda bolje ne.

Lijevo gore vidimo prozor u kojem možemo dodavati nove sudce. Unosimo ime, prezime i zemlju iz koje sudac dolazi. Textbox za ID nam ne treba jer je autoincrement što znači da će se automatski inkrementirati prilikom unosa ostalih podataka.

Desno možemo vidjeti prozor kojim ćemo se služiti ukoliko želimo ažurirati ili brisati vrijednosti iz baze podataka.

Dohvaćanje podataka iz baze podataka prilikom pokretanja aplikacije sam riješio na jako jednostavan način.

```
1 reference
private void Form1_Load(object sender, EventArgs e)
{
    READ();
    UCLEAR();
}
```

Slika 22 form load

Ubacio sam funkciju za dohvaćanje podataka iz baze u load formu. Detaljno ću prikazati funkcije koje sam samostalno kreirao, ali u nešto daljnjem tekstu.

10.1. Unos

Idemo probati unijeti novog sudca u našu bazu podataka. Koristit ćemo se testnim podacima koje ćemo kasnije ažurirati te na kraju izbrisati.

Unosimo testne podatke u odgovarajuća polja za unos novog sudca.

UFC

CREATE

IME:

PREZIME:

ZEMLJA:

SAVE

UPDATE / DELETE

ID:

IME:

PREZIME:

ZEMLJA:

UPDATE DELETE

READ / VIEW DATA

	ID	IME	PREZIME	ZEMLJA
▶	1	Steve	Mazzgati	1
	2	Mike	Beltran	1
	3	Marc	Goddard	3
	4	Mario	Yamasaki	9
	5	Herb	Dean	1
	6	John	McCarthy	1

Slika 23 unos

Klikom na gumb Save potvrđujemo unos i unosimo novog sudca u ovom slučaju Mladog Zengu u našu bazu podataka.

UFC

CREATE

IME:

PREZIME:

ZEMLJA:

SAVE

UPDATE / DELETE

ID:

IME:

PREZIME:

ZEMLJA:

UPDATE DELETE

READ / VIEW DATA

	ID	IME	PREZIME	ZEMLJA
▶	1	Steve	Mazzgati	1
	2	Mike	Beltran	1
	3	Marc	Goddard	3
	4	Mario	Yamasaki	9
	5	Herb	Dean	1
	6	John	McCarthy	1

Slika 24 spremanje

Nakon što smo kliknuli na gumb SAVE, textboxovi se automatski vraćaju na početno stanje, te su spremni na unos novog sudca. To sam također implementirao pomoću funkcije CLEAR() koju ću kasnije pokazati.

Nakon unosa novog sudca, možemo vidjeti da je unos uspio te da se on nalazi unutar naše baze podataka.

ID	IME	PREZIME	ZEMLJA
4	Mario	Yamasaki	9
5	Herb	Dean	1
6	John	McCarthy	1
7	Josh	Rosenthal	1
20	Mladi	Zenga	4

Slika 25 prikaz rezultata

U naredne dvije slike vidite prikaz koda koji nam omogućuje kreiranje novog sudca te zapis u bazu podataka.

```
// CREATE FUNCTION
1reference
public void createData()
{
    connection.Open();
    using (MySqlCommand cmd = new MySqlCommand())
    {
        cmd.CommandText = "INSERT INTO sudac(ime_sudac, prezime_sudac, zemlja_id) VALUES(@ime, @prezime, @zemlja); ";
        cmd.CommandType = CommandType.Text;
        cmd.Connection = connection;

        cmd.Parameters.Add("@ime", MySqlDbType.VarChar).Value = ime_sudac;
        cmd.Parameters.Add("@prezime", MySqlDbType.VarChar).Value = prezime_sudac;
        cmd.Parameters.Add("@zemlja", MySqlDbType.Int32).Value = zemlja_id;

        cmd.ExecuteNonQuery();
        connection.Close();
    }
}
```

Slika 26 [C#] funkcija CreateData

```
// CREATE
1 reference
public void CREATE()
{
    crud.ime_sudac = imetxt.Text;
    crud.prezime_sudac = prezimetxt.Text;
    crud.zemlja_id = int.Parse(zemljatxt.Text);
    crud.createData();
}
```

Slika 27 [C#] funkcija CREAT

10.2. Ažuriranje

Kao što vidimo Mladi Zenga se nalazi unutar naše baze podataka.

Dalje, idemo testirati funkciju update.

Klikom na podatke u DataGridViewu automatski se popunjavaju textboxovi za ažuriranje i brisanje.

The screenshot shows the UFC application window. It contains two main panels: 'CREATE' on the left and 'UPDATE / DELETE' on the right. Below these is a 'READ / VIEW DATA' section with a table.

CREATE Panel:

- IME:
- PREZIME:
- ZEMLJA:
- SAVE button

UPDATE / DELETE Panel:

- ID:
- IME:
- PREZIME:
- ZEMLJA:
- UPDATE button
- DELETE button

READ / VIEW DATA Table:

	ID	IME	PREZIME	ZEMLJA
	4	Mario	Yamasaki	9
	5	Herb	Dean	1
	6	John	McCarthy	1
	7	Josh	Rosenthal	1
▶	20	Mladi	Zenga	4

Slika 28 ažuriranje

Na dalje, možemo ažurirati vrijednost u tablici, na primjer promjeniti ćemo Mladom Zengi zemlju iz 4 u 7. Klikom na gumb UPDATE ažuriraju se vrijednost u bazi podataka koje su i vidljive na DataGridViewu. Kao i kod kreiranja, vrijednosti iz textboxova se automatski vraćaju na početno stanje te su spremne za daljnji rad.

ID	IME	PREZIME	ZEMLJA
4	Mario	Yamasaki	9
5	Herb	Dean	1
6	John	McCarthy	1
7	Josh	Rosenthal	1
20	Mladi	Zenga	7

Slika 29 ažurirani prikaz podataka

U prilogu na dvije slike vidimo kako izgleda kod koji nam omogućuje ažuriranje podataka unutar baze podataka.

```
// UPDATE FUNCTION
1 reference
public void updateData()
{
    connection.Open();
    using (MySQLCommand cmd = new MySQLCommand())
    {
        cmd.CommandText = "UPDATE sudac SET ime_sudac=@ime, prezime_sudac=@prezime, zemlja_id=@zemlja WHERE id_sudac=@id";
        cmd.CommandType = CommandType.Text;
        cmd.Connection = connection;

        cmd.Parameters.Add("@ime", MySQLDbType.VarChar).Value = ime_sudac;
        cmd.Parameters.Add("@prezime", MySQLDbType.VarChar).Value = prezime_sudac;
        cmd.Parameters.Add("@zemlja", MySQLDbType.Int32).Value = zemlja_id;

        cmd.Parameters.Add("@id", MySQLDbType.Int32).Value = id_sudac;

        cmd.ExecuteNonQuery();
        connection.Close();
    }
}
```

Slika 30 [C#] funkcija updateData

```
// UPDATE
1reference
public void UPDATE()
{
    crud.ime_sudac = u_imetxt.Text;
    crud.prezime_sudac = u_prezimetxt.Text;
    crud.zemlja_id = int.Parse(u_zemljatxt.Text);
    crud.id_sudac = int.Parse(u_idtxt.Text);
    crud.updateData();
}
```

Slika 31 [C#] funkcija UPDATE

10.3. Brisanje

Kao što vidimo, ažurirane su vrijednosti na DataGridViewu. Na isti način možemo i obrisati zadani redak. Klikom na njega te klikom na gumb DELETE.

The screenshot shows the UFC application window with the following components:

- CREATE Panel:** Contains input fields for IME, PREZIME, and ZEMLJA, and a SAVE button.
- UPDATE / DELETE Panel:** Contains input fields for ID (20), IME (Mladi), PREZIME (Zenga), and ZEMLJA (7), and buttons for UPDATE and DELETE.
- READ / VIEW DATA Panel:** Contains a table with the following data:

	ID	IME	PREZIME	ZEMLJA
	4	Mario	Yamasaki	9
	5	Herb	Dean	1
	6	John	McCarthy	1
	7	Josh	Rosenthal	1
▶	20	Mladi	Zenga	7

Slika 32 brisanje

The screenshot shows a web application window titled "UFC". It contains two main form sections and a data table.

CREATE Form:

- IME:
- PREZIME:
- ZEMLJA:
-

UPDATE / DELETE Form:

- ID:
- IME:
- PREZIME:
- ZEMLJA:
-
-

READ / VIEW DATA Table:

ID	IME	PREZIME	ZEMLJA
3	Marc	Goddard	3
4	Mario	Yamasaki	9
5	Herb	Dean	1
6	John	McCarthy	1
7	Josh	Rosenthal	1

Slika 33 prikaz vrijednosti nakon brisanja

U prilogu vidimo kako izgleda kod u C# za brisanje podataka iz baze podataka.

```
// DELETE

1reference
public void deleteData()
{
    connection.Open();
    using (MySqlCommand cmd = new MySqlCommand())
    {
        cmd.CommandText = "DELETE FROM sudac WHERE id_sudac=@id";
        cmd.CommandType = CommandType.Text;
        cmd.Connection = connection;

        cmd.Parameters.Add("@id", MySqlDbType.Int32).Value = id_sudac;

        cmd.ExecuteNonQuery();
        connection.Close();
    }
}
```

Slika 34 [C#] funkcija deleteData

```
// DELETE
1 reference
public void DELETE()
{
    crud.id_sudac = int.Parse(u_idtxt.Text);
    crud.deleteData();
}
```

Slika 35 [C#] funkcija DELETE

10.4. Dohvaćanje podataka iz baze

U prilogu možete vidjeti dio koda koji se odnosi na dohvaćanje podataka iz baze podataka.

```
// READ
1 reference
public void readData()
{
    dt.Clear();
    string query = "SELECT * from sudac";
    MySqlDataAdapter mySqlData = new MySqlDataAdapter(query, connection);
    mySqlData.Fill(ds);
    dt = ds.Tables[0];
}
```

Slika 36 [C#] funkcija readData

```
// READ
4 references
public void READ()
{
    dataGridView1.DataSource = null;
    crud.readData();
    dataGridView1.DataSource = crud.dt;
    dataGridView1.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
    dataGridView1.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
    dataGridView1.Columns[0].HeaderText = "ID";
    dataGridView1.Columns[1].HeaderText = "IME";
    dataGridView1.Columns[2].HeaderText = "PREZIME";
    dataGridView1.Columns[3].HeaderText = "ZEMLJA";
}
```

Slika 37 funkcija READ

11. Zaključak

Ovaj projekt služio je da se prikaže kako se kreira baza podataka u svrhe vođenja evidencije organizacije UFC. Baza podataka se sastoji od trinaest tablica. U dokumentaciji je detaljno prikazano kreiranje tablice sa svim svojim atributima i vezama. Također napravljen je dobar broj okidača te upita što jednostavnijih što složenijih. Opisane su tehnologije koje su korištene u ovom projektnom zadatku te detaljan prikaz grafičkog sučelja koje je kreirano u C# unutar alata Microsoft Visual Studio. Modeliranje i administracija baze podataka izvršeno je unutar alata DBeaver. Grafičko sučelje je kreirano u odnosu na projektni zadatak te omogućuje osnovne CRUD operacije nad bazom podataka.

Mogu sa sigurnošću reći da je ovo do sada bio najzahtjevniji, najzanimljiviji i najbolji projektni zadatak do sada. Iako mi je oduzeo dosta vremena, uživao sam svake sekunde praveći ga. Nadam se da će i budući projekti u nastavku mog školovanja biti zanimljivi i edukativni kao i ovaj projekt.

12. Prilozi

12.1. Slike

Slika 1 ERA model [vlastoručna izrada, draw.io].....	3
Slika 2 ERA model - generirani diagram iz Dbeavera	4
Slika 3 rezultat okidača provjera_potpisivanja_ugovor.....	15
Slika 4 rezultat okidača plavi_crveni	16
Slika 5 rezultat okidača provjera_borbe_f.....	17
Slika 6 rezultat okidača provjera_borbe_f2	18
Slika 7 rezultat upita sudac_zemlja.....	19
Slika 8 rezultat upita menadzer_zemlja	20
Slika 9 rezultat upita sudio_borbi	20
Slika 10 rezultat upita menadzer_borac.....	21
Slika 11 rezultat upita borac_nadimak_kategorija	22
Slika 12 rezultat upita borac_nadimak_kategorija_godina	23
Slika 13 [DBeaver] kreiranje tablice	24
Slika 14 [DBeaver] unos podataka.....	25
Slika 15 [DBeaver] rad unutar alata	25
Slika 16 [DBeaver] ERA model.....	26
Slika 17 [DBeaver].....	27
Slika 18 [XAMPP]	28
Slika 19 [Microsoft Visual Studio]	28
Slika 20 [Microsoft Visual Studio]	29
Slika 21 pokretanje aplikacije	30
Slika 22 form load.....	31
Slika 23 unos.....	32
Slika 24 spremanje.....	32
Slika 25 prikaz rezultata	33
Slika 26 [C#] funkcija CreateData	33
Slika 27 [C#] funkcija CREATE.....	34
Slika 28 ažuriranje.....	34
Slika 29 ažurirani prikaz podataka	35
Slika 30 [C#] funkcija updateData.....	35
Slika 31 [C#] funkcija UPDATE.....	36
Slika 32 brisanje	36

Slika 33 prikaz vrijednosti nakon brisanja	37
Slika 34 [C#] funkcija deleteData	37
Slika 35 [C#] funkcija DELETE	38
Slika 36 [C#] funkcija readData	38
Slika 37 funkcija READ.....	38

12.2. **Tablice**

Tablica 1 tablica arena	5
Tablica 2 tablica borac.....	5
Tablica 3 tablica borba	6
Tablica 4 tablica bori_se	6
Tablica 5 tablica dogadaj.....	6
Tablica 6 tablica kamp.....	7
Tablica 7 tablica kategorija	7
Tablica 8 tablica menadzer.....	7
Tablica 9 tablica nadimak	7
Tablica 10 tablica nalazi_se.....	8
Tablica 11 tablica sudac	8
Tablica 12 tablica vodi	8
Tablica 13 tablica zemlja	9
Tablica 14 veza borac - nadimak.....	10
Tablica 15 veza borac - bori_se.....	10
Tablica 16 veza kategorija - bori_se	10
Tablica 17 veza borac - kamp.....	10
Tablica 18 veza borac – vodi.....	11
Tablica 19 veza menadzer - vodi.....	11
Tablica 20 veza borac - zemlja	11
Tablica 21 veza menadzer – zemlja	11
Tablica 22 veza sudac – zemlja.....	12
Tablica 23 veza arena – zemlja	12
Tablica 24 veza kamp - nalazi_se.....	12
Tablica 25 veza zemlja - nalazi_se	12
Tablica 26 veza borac - borba	13
Tablica 27 veza borba - dogadaj	13
Tablica 28 veza borba - arena	13
Tablica 29 veza borba - sudac.....	14

13. Literatura

- [1] <https://www.mysql.com/>, pristupano 03.01.2021.
- [2] SQL-Napredne teme, Kornelije Rabuzin, Varaždin 2014
- [3] <https://visualstudio.microsoft.com/>, pristupano 28.12.2020
- [4] <https://www.apachefriends.org/download.html>, pristupano 26.12.2020.
- [5] <https://dbeaver.io/>, pristupano 26.12.2020.
- [6] <https://app.diagrams.net/>, pristupano 19.12.2020.