

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

Antun Tkalčec

Bolnički sustav

SEMINARSKI RAD

Varaždin, 2021.

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

Antun Tkalčec

JMBAG: 0016136241

Studij: Informacijski sustavi

BOLNIČKI SUSTAV

SEMINARSKI RAD

Mentor/Mentorica:

Prof. dr. sc. Rabuzin Kornelije

Križevci, siječanj 2021.

Antun Tkalčec

Izjava o izvornosti

Izjavljujem da je moj seminarski rad izvorni rezultat mog rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sadržaj

1. Uvod	1
2. Opis aplikacijske domene	2
3. Prikaz i opis ERA modela	3
3.1. Opisi tablica	4
3.1.1. Opisi triggera nad tablicama u ERA modelu	6
3.1.2. Opisi jednostavnih i složenih upita nad tablicama	7
4. DBeaver universal database tool	10
4.1. Sučelje	12
5. MySQL Workbench i drugi alati	14
5.1. WampServer	15
6. Web aplikacija i sučelje	16
7. Zaključak	22
8. Popis literature	22

1. Uvod

Ovaj projektni zadatak je izraditi bazu podataka nekog bolničkog sustava koristeći MySQL, alat DBeaver i razvijeno sučelje korištenjem PHP jezika.

Kroz projekt bit će prikazan opis aplikacijske domene i ERA model baze podataka bolničkog sustava, opisani će biti alati korišteni za modeliranje i administraciju moje baze podataka te će biti prikazana funkcionalnost sučelja uz priložene *screenshotove*.

Projekt neće za svoj objekt obrade uzimati neku konkretnu bolnicu, već će to biti neka zamišljena bolnica, tako da se imena npr. odjela i djelatnika bolnice mogu, ali i ne moraju podudarati sa stvarnosti. Unatoč tome, nastojat će dostojno i realno prikazati način funkcioniranja bolničkog sustava.

Ovu temu sam izabrao jer je baza podataka bolničkog sustava od izuzetne važnosti svakoj bolnici, te mislim da zbog toga mogu putem projekta sakupiti mnogo kvalitetnog iskustva pošto mi je želja jednog se dana baviti bazama podataka.

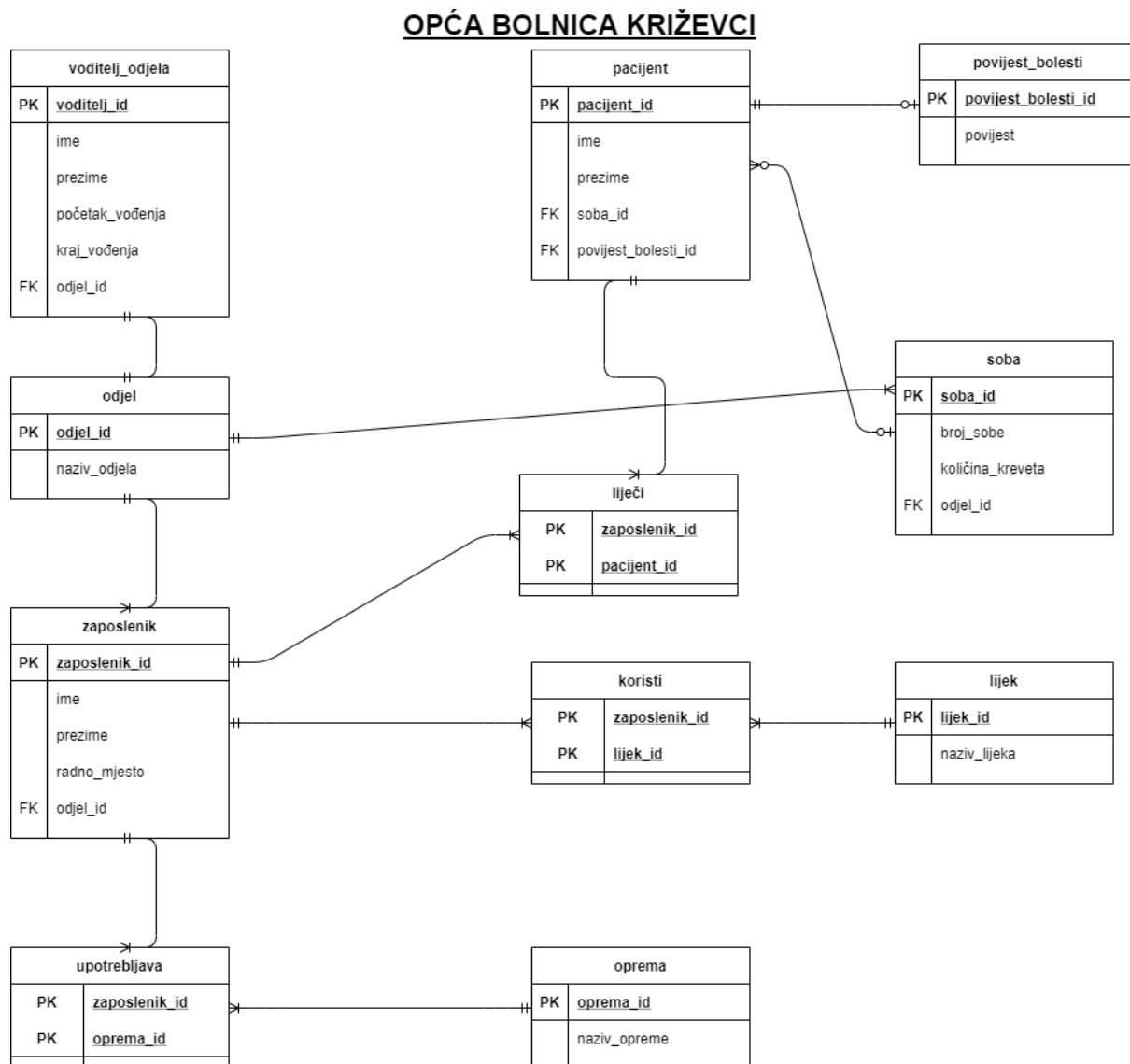
2. Opis aplikacijske domene

Aplikacijska domena kojom će se baviti projekt jest bolnica i njeni pripadajući procesi, kao npr. procesi evidentiranja medicinskih podataka pacijenata koji se liječe u bolnici ili dolaze na kontrole, a ne nalaze se na bolničkom liječenju.

Hipotetska bolnica kojom će se projekt baviti zvat će se Opća bolnica Križevci. Bolnica ima svoje odjele, a svaki odjel ima voditelja/icu te ostale zaposlenike. Isto tako, svaki odjel ima svoje pacijente, sobe, lijekove i ostalu opremu. Zaposlenici mogu biti doktori/ice, medicinske sestre/tehničari, te kuhari/ice. Nadalje, svaki pacijent ima svoju povijest bolesti. Zaposlenici bolnice koriste određenu opremu kako za liječenje pacijenata tako i za spremanje hrane u odjelu kuhinje (ako je taj zaposlenik kuhar), te doktori upotrebljuju lijekove pri tretiranju svojih pacijenata. Pacijenti mogu, ali ne moraju imati povijest bolesti. Mogu se, ali i ne moraju, nalaziti na liječenju u nekoj od soba u nekom od odjela bolnice. Sobe za pacijente kao takve ne postoje u odjelu kuhinje. Svaki odjel može u određenom trenutku imati samo jednog voditelja, a jedan voditelj može u određenom trenutku biti voditelj samo jednog odjela. Međutim, kroz vrijeme određeni odjel može zamijeniti nekoliko voditelja. Zaposlenici koji rade u odjelu kuhinje ne mogu tretirati pacijente.

Dakle, aplikacijska domena je upisivanje odjela, zaposlenika, soba, pacijenata, pacijentove povijesti bolesti, te koji doktor liječi kojeg pacijenta, koje lijekove koristi tijekom liječenja te ostalih podataka o svim navedenim entitetima.

3. Prikaz i opis ERA modela



Slika 1. ERA dijagram sustava

Na slici 1. vidi se ERA dijagram bolničkog sustava po kojem će biti rađena baza podataka. ERA model se sastoji od 11 tablica, od kojih su 3 slabi entiteti i služe za povezivanje tablica s vezama više na više.

3.1. Opisi tablica

Glavna, početna tablica je odjel koja se sastoji od primarnog ključa za odjel, naziv tog odjela i vanjskog ključa na voditelja/icu tog odjela. Naziv odjela će biti varchar duljine, recimo, 30.

Tablica voditelj_odjela se sastoji od primarnog ključa, voditelj_id, njegovog ili njezinog imena, prezimena, koji su oboje tipa varchar(30), zatim početak_vođenja tipa date te kraj_vođenja tipa date koji može biti i null vrijednosti, jer trenutni voditelj očito neće već imati kraj vođenja odjela.

Tablica zaposlenik se odnosi na sve vrste djelatnika bolnice, a to su doktori, medicinske sestre, medicinski tehničari te kuhari ili kuharice. Svaki zaposlenik mora imati upisano ime, prezime, te radno mjesto koje je jedno od prethodno navedenih. Vanjski ključ u tablici zaposlenika spaja na odjel u kojem radi, npr. doktor može raditi u odjelu infektologije, a kuhar može raditi samo u odjelu prehrane. Povezano s atributom radno_mjesto, tu će se kriti jedan od triggera. Naime, pacijenta ne može liječiti kuhar, nego samo doktor ili drugo medicinsko osoblje, stoga će se pri upisu u tablicu liječi, koja se sastoji od složenog primarnog ključa (zaposlenik_id, pacijent_id) provjeravati da li je zaposlenikovo radno mjesto kuhar.

Tablica pacijent ima primarni ključ pacijent_id, ime i prezime pacijenta, vanjski ključ na sobu te vanjski ključ na pacijentovu povijest bolesti, koju on može, ali ne mora imati (ako je, recimo, pacijent mlada osoba i nije imao nikakve teže bolesti).

Tablica soba se sastoji od primarnog ključa soba_id, integera broj_sobe, integera količina_kreveta i vanjskog ključa na odjel u kojem se nalazi.

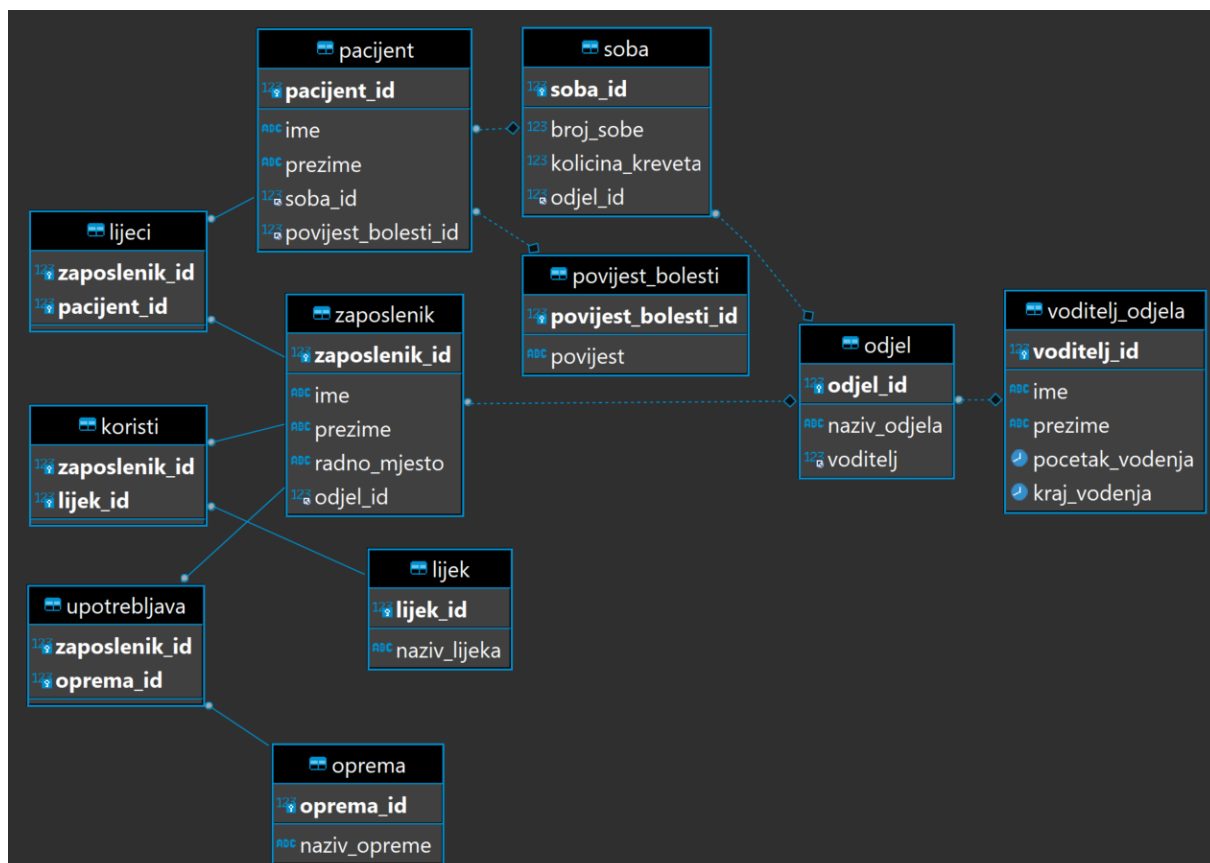
Tablica povijest_bolesti ima primarni ključ povijest_bolesti_id, te atribut povijest koji je varchar duljine 100. U tom atributu može biti zapisano npr. „Prebolio COVID-19“, ili „Alergičan na penicilin“.

Tablica koristi je slabi entitet koji povezuje zaposlenika i lijekove koje koristi, te se sastoji od složenog primarnog ključa (zaposlenik_id, lijek_id). Naime, više zaposlenika može koristiti više lijekova i time liječiti više pacijenata.

Tablica upotrebljava je još jedan slabi entitet koji povezuje zaposlenika i opremu koje koristi, a sastoji se od složenog primarnog ključa (zaposlenik_id, oprema_id).

Tablica oprema se sastoji od primarnog ključa `oprema_id` i tekstualnog atributa `naziv_opreme`. Oprema se razlikuje od lijekova zato što zaposlenik koji je, recimo, doktor može koristiti skalpel, a kuhar može koristiti lonac, što je očito različito od lijekova.

Tablica lijek ima primarni ključ `lijek_id`, i `naziv_lijeka` tipa `varchar` te se odnosi na lijekove koje zaposlenici koriste prilikom tretiranja pacijenata.



Slika 2. ERA dijagram sustava u alatu DBeaver

3.1.1. Opisi triggera nad tablicama u ERA modelu

```
CREATE DEFINER='root'@'localhost' TRIGGER `trigger1` BEFORE INSERT ON `lijeći` FOR EACH ROW begin
  declare uloga varchar(20);
  select zaposlenik.radno_mjesto into uloga from zaposlenik where zaposlenik.zaposlenik_id = new.zaposlenik_id and zaposlenik.radno_mjesto like '%kuhar%';
  if uloga like '%kuhar%' then signal sqlstate value '45000' set message_text = 'Zaposlenik radnog mjesta kuhar ne može liječiti pacijenta.';
  end if;
end
```

Slika 3. Trigger 1 (lijeći)

Na slici 3. vidi se prvi trigger koji sam već spomenuo, a on sprječava da se u tablicu lijeći unese zaposlenik, čije je radno mjesto kuhar. Samo doktori i drugo medicinsko osoblje može liječiti pacijenta. Svrha ovog triggera je očita, a to je da u bazi podataka ne može doći do poprilično velike greške da osoba koja nije medicinski obrazovana, bude postavljena za liječnika nekog pacijenta.

```
CREATE DEFINER='root'@'localhost' TRIGGER `trigger2` BEFORE INSERT ON `soba` FOR EACH ROW begin
  declare naziv varchar(20);
  select odjel.naziv_odjela into naziv from odjel where odjel.odjel_id = new.odjel_id and odjel.naziv_odjela like '%kuh%';
  if naziv like '%kuh%' then signal sqlstate value '45000' set message_text = 'U odjelu kuhinja ne postoje sobe za pacijente.';
  end if;
end
```

Slika 4. Trigger 2 (soba)

Na slici 4. vidi se drugi trigger, koji sprječava da u tablicu soba bude unesena soba koja se nalazi u odjelu za prehranu, tj. u odjelu gdje kuhari spremaju hranu za pacijente. Sobe za pacijente mogu biti samo u odjelima gdje se pacijenti tretiraju. Svrha ovog triggera je ta da se baza podataka u svakom trenutku podudara sa stvarnošću u bolnici, a to je da nema pacijenata niti soba za pacijente u odjelu bolnice gdje se sprema hrana za pacijente.

```
CREATE DEFINER='root'@'localhost' TRIGGER `trigger3` BEFORE INSERT ON `pacijent` FOR EACH ROW begin
  declare counter int;
  declare pomocna int;
  select soba.kolicina_kreveta into pomocna from soba where soba.soba_id = new.soba_id;
  select count(*) into counter from pacijent inner join soba s on pacijent.soba_id = s.soba_id where new.soba_id = s.soba_id;
  if counter >= pomocna then signal sqlstate value '45000' set message_text = 'U toj sobi su već popunjeni svi kreveti.';
  end if;
end
```

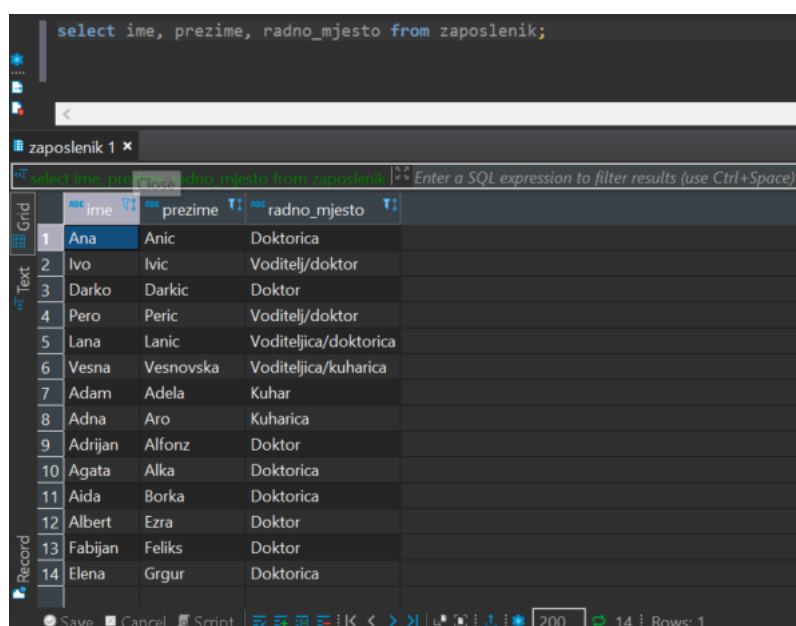
Slika 5. Trigger 3 (pacijent)

Na slici 5. vidi se treći trigger, koji sprječava da u sobi bude smješteno više pacijenata nego ima kreveta. Svrha ovog triggera je da se spriječe situacije gdje bi pacijent dobio nalog da bude smješten u neku sobu, samo da bi netko od osoblja pri vođenju pacijenta u tu sobu uočio da je ta soba već popunjena, što bi uzrokovalo

kašnjenje cijelog procesa, frustraciju medicinskom osoblju te potencijalno jake posljedice za oboljelog pacijenta.

3.1.2. Opisi jednostavnih i složenih upita nad tablicama

Sada ćemo vidjeti nekoliko upita nad popunjenim tablicama ovog bolničkog sustava, njih, recimo, 5. Počet ću sa jednostavnijim upitima i ići prema složenijima, te će svaki biti prikazan i reći ću zašto sam odabrao taj upit.



The screenshot shows a database application interface. At the top, a SQL query is entered in a text field: `select ime, prezime, radno_mjesto from zaposlenik;`. Below the query field, the results are displayed in a grid view. The grid has columns for 'ime' (first name), 'prezime' (last name), and 'radno_mjesto' (job position). The results are listed in 14 rows, numbered 1 to 14. The grid view is titled 'zaposlenik 1'. The status bar at the bottom indicates '200' and '14 Rows: 1'.

	ime	prezime	radno_mjesto
1	Ana	Anic	Doktorica
2	Ivo	Ivic	Voditelj/doktor
3	Darko	Darkic	Doktor
4	Pero	Peric	Voditelj/doktor
5	Lana	Lanic	Voditeljica/doktorica
6	Vesna	Vesnovska	Voditeljica/kuharica
7	Adam	Adela	Kuhar
8	Adna	Aro	Kuharica
9	Adrijan	Alfonz	Doktor
10	Agata	Alka	Doktorica
11	Aida	Borka	Doktorica
12	Albert	Ezra	Doktor
13	Fabijan	Felijs	Doktor
14	Elena	Grgur	Doktorica

Slika 6. Jednostavan upit nad tablicom zaposlenik

Slika 6. prikazuje možda najjednostavniji od najvažnijih upita nad cijelim sustavom, a to je upit koji prikazuje imena i prezimena svih zaposlenika i njihovih radnih mjesta. Ovaj upit je očita polaznica za sve ostale upite nad tablicama u bazi.

```

select ime, prezime, pocetak_vodenja as Početak, kraj_vodenja as Kraj, naziv_odjela
from voditelj_odjela v inner join odjel o on
v.odjel_id = o.odjel_id;

```

	ime	prezime	Početak	Kraj	naziv_odjela
1	Ana	Anic	1998-10-04	1999-04-10	Infektologija
2	Ivo	Ivic	1999-04-10	[NULL]	Infektologija
3	Darko	Darkic	1998-05-02	2004-12-13	Kardiologija
4	Pero	Peric	2004-12-14	[NULL]	Kardiologija
5	Lana	Lanic	1998-03-01	[NULL]	Otorinolaringologija
6	Vesna	Vesnovska	1998-03-01	[NULL]	Kuhinja

Slika 7. Upit nad tablicom voditelj_odjela

Slika 7. prikazuje malo složeniji upit koji iz tablica voditelj_odjela i odjel vadi podatke o voditelju/ici određenog odjela zajedno s datumima od kad do kad ta osoba taj odjel vodi ili je vodio/la. Za osobe koje imaju *NULL* pod „Kraj“ se podrazumijeva da ne postoji datum kraja vodstva odjela, što znači da je ta osoba još uvijek voditelj/ica tog odjela.

```

select concat(ime, ' ', prezime) as Zaposlenik, naziv_odjela as Odjel, naziv_opreme as Oprema
from zaposlenik z inner join odjel o on z.odjel_id = o.odjel_id inner join upotrebljava u
on z.zaposlenik_id = u.zaposlenik_id inner join oprema op on u.oprema_id = op.oprema_id;

```

	Zaposlenik	Odjel	Oprema
1	Ana Anic	Infektologija	Audiometar
2	Ana Anic	Infektologija	Skalpel
3	Ana Anic	Infektologija	EKG
4	Ivo Ivic	Infektologija	Spirometar
5	Adrijan Alfonz	Infektologija	Audiometar
6	Adrijan Alfonz	Infektologija	Skalpel
7	Agata Alka	Infektologija	EKG
8	Darko Darkic	Kardiologija	Skalpel
9	Pero Peric	Kardiologija	Spirometar
10	Aida Borka	Kardiologija	Spirometar
11	Albert Ezra	Kardiologija	Skalpel
12	Lana Lanic	Otorinolaringologija	Audiometar
13	Lana Lanic	Otorinolaringologija	EKG
14	Fabijan Feliks	Otorinolaringologija	Audiometar
15	Elena Grgur	Otorinolaringologija	Audiometar
16	Elena Grgur	Otorinolaringologija	Skalpel
17	Elena Grgur	Otorinolaringologija	EKG
18	Elena Grgur	Otorinolaringologija	Spirometar
19	Vesna Vesnovska	Kuhinja	Lonac
20	Vesna Vesnovska	Kuhinja	Kuhaća
21	Adam Adela	Kuhinja	Lonac
22	Adam Adela	Kuhinja	Kuhaća
23	Adna Aro	Kuhinja	Lonac
24	Adna Aro	Kuhinja	Kuhaća

Slika 8. Upit nad nekoliko tablica

Slika 8. prikazuje složeni upit koji ispisuje zaposlenike, njihove odjele i opremu kojom se koriste. Ovakav upit može biti koristan za pregled opreme koja je potrebna određenom odjelu ili zaposleniku za odrađivanje posla, tj. prikazuje da audiometar nije potreban nijednom zaposleniku u odjelu kardiologije, na primjer.

```

select concat(z.ime, ' ', z.prezime) as Zaposlenik, concat(p.ime, ' ', p.prezime) as Pacijent,
povijest as 'Povijest bolesti' from zaposlenik z inner join lijeци l on z.zaposlenik_id = l.zaposlenik_id
inner join pacijent p on l.pacijent_id = p.pacijent_id left join povijest_bolesti po
on p.povijest_bolesti_id = po.povijest_bolesti_id;

```

	Zaposlenik	Pacijent	Povijest bolesti
1	Ana Anic	Hana Kludije	[NULL]
2	Aida Borka	Hana Kludije	[NULL]
3	Ana Anic	Karlo Henrik	Prebolio COVID-19. Alergičan na prašinu. Prebolio mononukleozu.
4	Albert Ezra	Karlo Henrik	Prebolio COVID-19. Alergičan na prašinu. Prebolio mononukleozu.
5	Ivo Ivic	Ladislav Nedjeljko	Alergičan na penicilin.
6	Fabijan Feliks	Ladislav Nedjeljko	Alergičan na penicilin.
7	Darko Darkic	Nada Lazar	Preboljela šarlah.
8	Elena Grgur	Nada Lazar	Preboljela šarlah.
9	Lana Lanic	Rino Timon	[NULL]
10	Adrijan Alfonz	Žarko Uroš	Alergičan na penicilin.
11	Agata Alka	Zoran Horvat	[NULL]

Slika 9. Složeniji upit nad nekoliko tablica

Slika 9. prikazuje još složeniji upit nad tablicama zaposlenik, liječi, pacijent i povijest_bolesti. Upit ispisuje zaposlenike (doktore), njihove pacijente i povijest bolesti tih pacijenata, ukoliko ju imaju.

```

select concat(z.ime, ' ', z.prezime) as Zaposlenik, concat(p.ime, ' ', p.prezime) as Pacijent,
povijest as 'Povijest bolesti', broj_sobe as Soba, naziv_odjela as Odjel from zaposlenik z
inner join liječi l on z.zaposlenik_id = l.zaposlenik_id inner join pacijent p
on l.pacijent_id = p.pacijent_id left join povijest_bolesti po on
p.povijest_bolesti_id = po.povijest_bolesti_id left join soba s on
p.soba_id = s.soba_id inner join odjel o on s.odjel_id = o.odjel_id;

```

	Zaposlenik	Pacijent	Povijest bolesti	Soba	Odjel
1	Ana Anic	Hana Klaudije	[NULL]	111	Infektologija
2	Aida Borka	Hana Klaudije	[NULL]	111	Infektologija
3	Ana Anic	Karlo Henrik	Prebolio COVID-19. Alergičan na prašinu. Prebolio mononukleozu.	111	Infektologija
4	Albert Ezra	Karlo Henrik	Prebolio COVID-19. Alergičan na prašinu. Prebolio mononukleozu.	111	Infektologija
5	Ivo Ivic	Ladislav Nedjeljko	Alergičan na penicilin.	222	Infektologija
6	Fabijan Feliks	Ladislav Nedjeljko	Alergičan na penicilin.	222	Infektologija
7	Darko Darkic	Nada Lazar	Preboljela šarlah.	333	Kardiologija
8	Elena Grgur	Nada Lazar	Preboljela šarlah.	333	Kardiologija
9	Lana Lanic	Rino Timon	[NULL]	444	Kardiologija
10	Adrijan Alfonz	Žarko Uroš	Alergičan na penicilin.	555	Otorinolaringologija
11	Agata Alka	Zoran Horvat	[NULL]	666	Otorinolaringologija

Slika 10. Složeni upit nad tablicama

Slika 10. prikazuje dosta složeni upit nad nekolicinom tablica koji je zapravo nadogradnja na prošli upit te uz sve što je ispisivao prošli upit, ispisuje i sobu gdje se nalazi pacijent te odjel u kojem se soba nalazi. Upit NE prikazuje pacijente koji nisu na liječenju u bolnici, dakle one pacijente koji tek dolaze redovito na kontrole te se ne nalaze u nijednoj od soba.

4. DBeaver universal database tool

Slike triggera i upita su sve uzete ravno iz alata DBeaver u kojem je ovaj projekt odrađen.



Slika 11. DBeaver logo

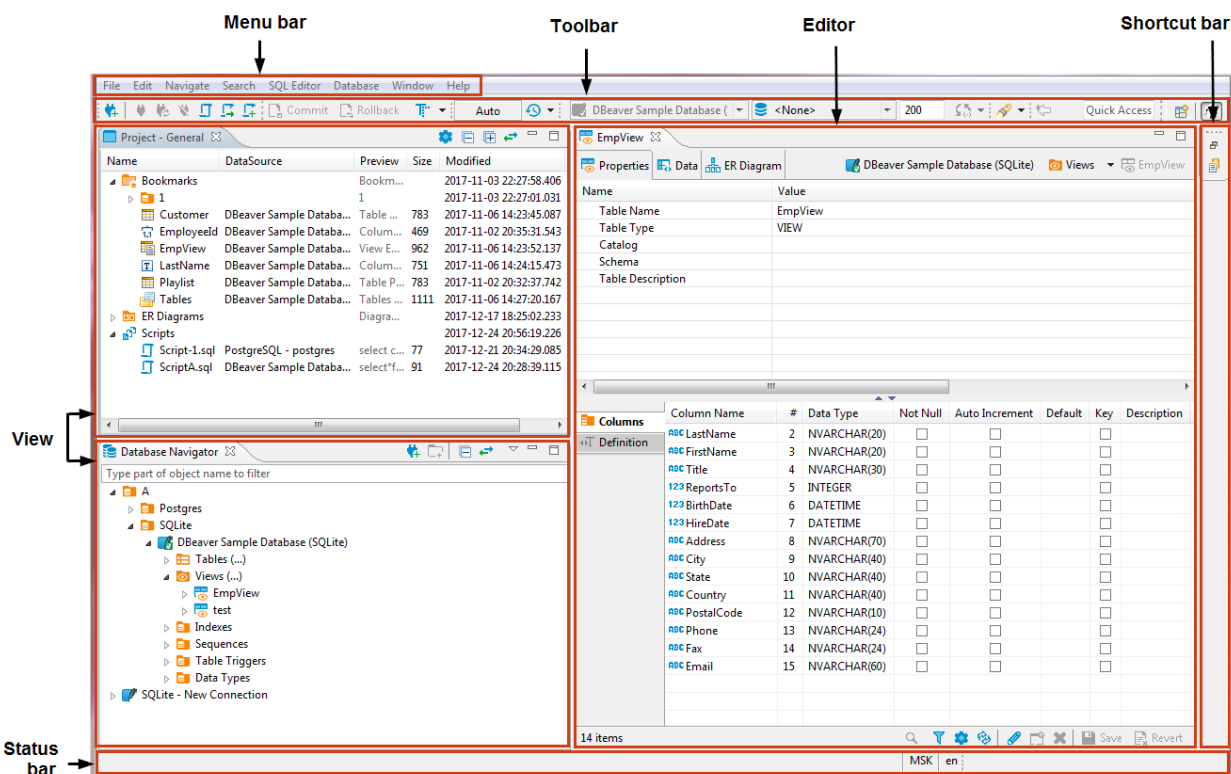
DBeaver je besplatni, *open source* univerzalni alat za developere i administratore baza podataka kojemu je glavni cilj lako korištenje. Alat pruža i korištenje *pluginova* koji korisnicima omogućuju mijenjanje načina rada aplikacije i dodavanje funkcionalnosti koje su neovisne o bazi podataka. Aplikacija je pisana u programskom jeziku Java.

Program je započeo kao hobi originalnog stvaratelja Sergea Ridera u 2010. Nakon službenog izlaska 2011. brzo je postao popularan alat u zajednici.

Program postoji u verzijama Community Edition, Eclipse Plugin Edition i Enterprise Edition. Ja sam koristio Community Edition verziju, koja podržava baze podataka poput MySQL koji sam upravo i koristio, PostgreSQL, Oracle, EXASOL, Firebird i ostale.

DBeaver podržava funkcije poput izvršavanja SQL upita, browser i editor podataka, SQL auto-završavanje, menadžment SQL skripti, browser i editor meta podataka, rendering ER dijagrama, SSH tuneliranje, izvoz i migraciju podataka te druge funkcije. Dostupne funkcije ovise o bazi podataka koju korisnik koristi.

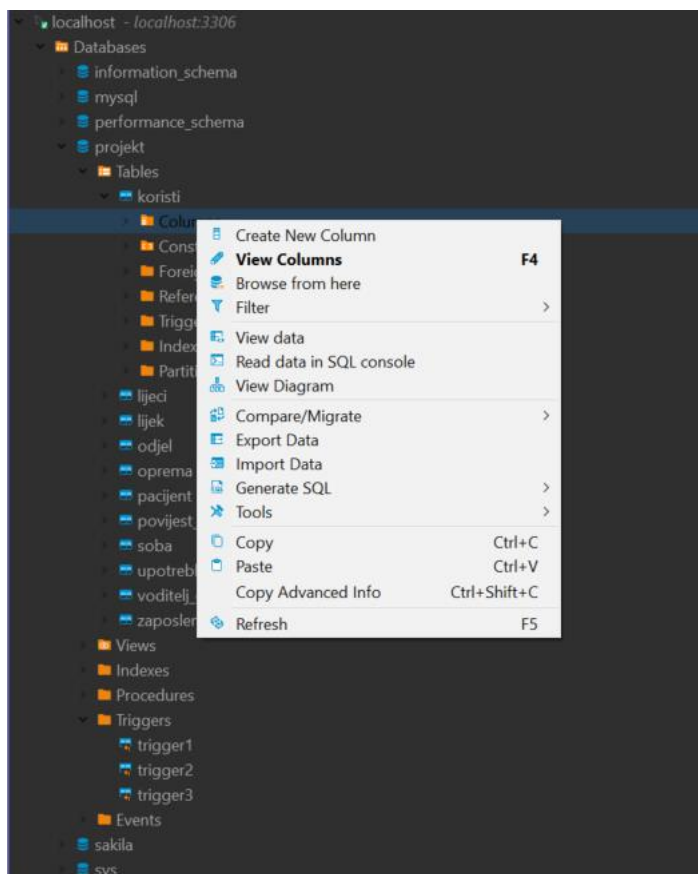
4.1. Sučelje



Slika 12. Sučelje DBeaver (light mode)

Slika 12. prikazuje primjer korisničkog sučelja preuzetog sa github stranice DBeavera. Sučelje se sastoji od menija, alatne trake, editora, trake prečaca, statusne trake i pogleda na projekt i bazu podataka.

U alatnoj traci nalaze se gumbi za spajanje na bazu podataka. Pri izradi projekta, pomoću MySQL Workbench sam stvorio server na koji sam se spajao u DBeaveru. O alatu MySQL Workbench će biti riječi kasnije. Za sada ću prikazati neke dijelove sučelja koji su mi se svidjeli, puno sam ih koristio, ili mislim da su nedostatak u aplikaciji u cijelosti (u usporedbi s npr. Datagripom).

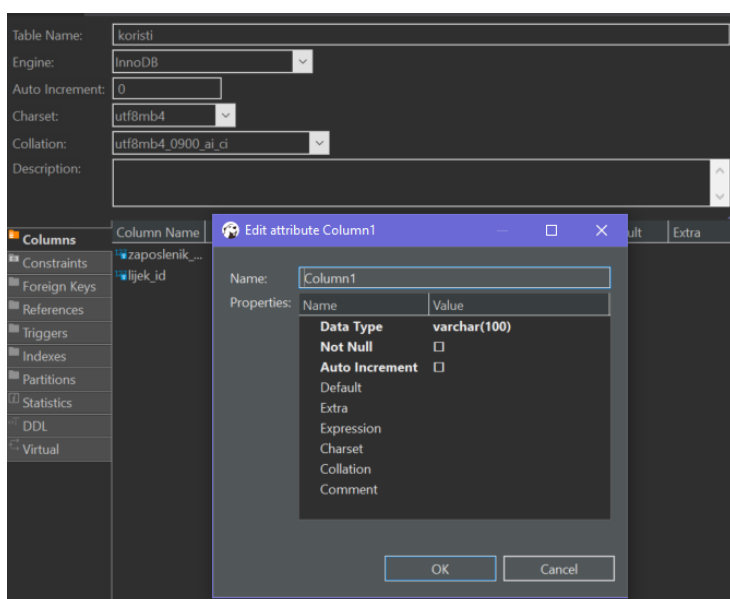


Slika 13. Database Navigator u mojoj sučelju DBeavera

Na slici 13. vidi se shema projekt u mojoj bazi podataka na serveru localhost te tablice koje sam kreirao za projekt, zajedno i sa triggerima koji su ranije prikazani.

U svojem sučelju sam koristio Dark Mode, kao što činim u svakoj drugoj aplikaciji koja ga ima, no ovdje Dark Mode ima probleme sa bojama npr. strelica te otežava korištenje ove aplikacije za koju je lakoća korištenja navedena kao

primarni cilj. Unatoč tome, aplikacija uistinu pruža jednostavno kreiranje primarnih i vanjskih ključeva (čak jednostavnije nego u Datagripu), no kreiranje samih tablica je opet slaba točka.



Slika 14. Kreiranje tablice, odnosno retka tablice

Problem kod kreiranja tablica jest taj što se nakon kreiranja tablice svaki atribut mora upisivati zasebno, što znatno produlji vrijeme stvaranja željenog sustava.

```
*<localhost> Script x
> insert into voditelj_odjela values (2, 'Ivo', 'Ivic', '1999-04-10', null, 1);
> insert into voditelj_odjela values (1, 'Ana', 'Anic', '1998-10-04', '1999-04-10', 1);
> insert into voditelj_odjela values (4, 'Pero', 'Peric', '2004-12-14', null, 2);
> insert into voditelj_odjela values (3, 'Darko', 'Darkic', '1998-05-02', '2004-12-13', 2);
> insert into voditelj_odjela values (5, 'Lana', 'Lanic', '1998-03-01', null, 3);
> insert into voditelj_odjela values (6, 'Vesna', 'Vesnovska', '1998-03-01', null, 4);

insert into odjel values (1, 'Infektologija');
insert into odjel values (2, 'Kardiologija');
insert into odjel values (3, 'Otorinolaringologija');
insert into odjel values (4, 'Kuhinja');
```

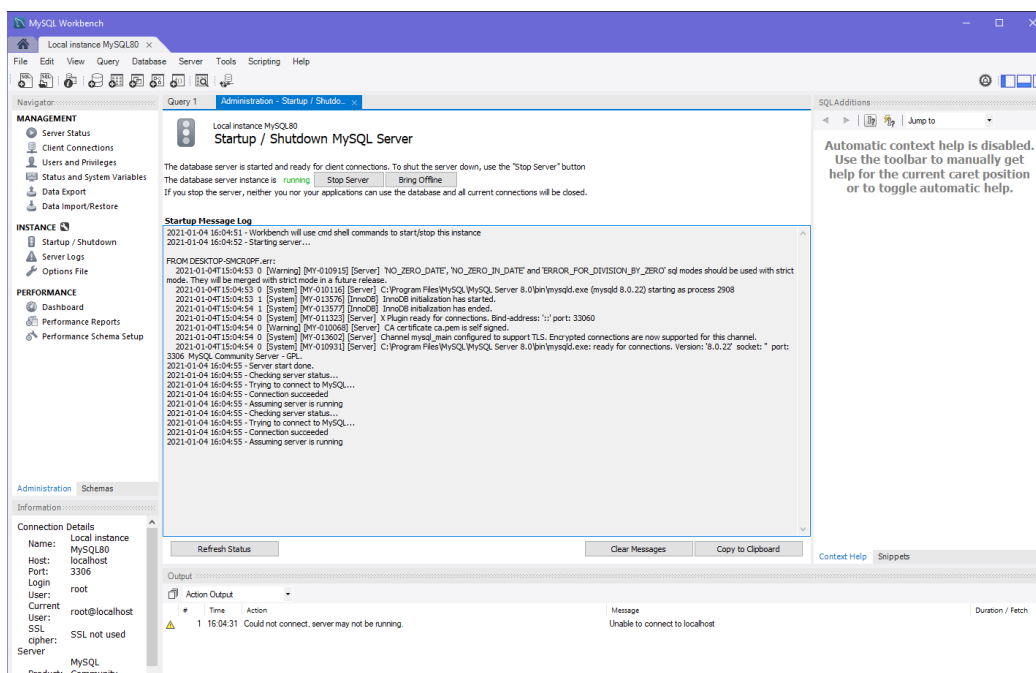
Slika 15. Konzola u sučelju DBeavera

Sljedeći dio sučelja koji sam, naravno, puno koristio je sama konzola gdje se upisuju upiti. U ovom dijelu sučelja ne mogu vidjeti neke znatne razlike od sučelja u Datagripu, osim očitih razlika u pisanju PostgreSQL-a i MySQL-a.

Sve u svemu, DBeaver kao alat mi se isprva činio kao ozbiljan korak unazad od Datagripa kojeg sam pobolje upoznao ranije, no s vremenom mi se njegovo sučelje i funkcionalnosti svidjelo te smatram da je za *open source* program vrlo snažan, koristan i pouzdan alat.

5. MySQL Workbench i drugi alati

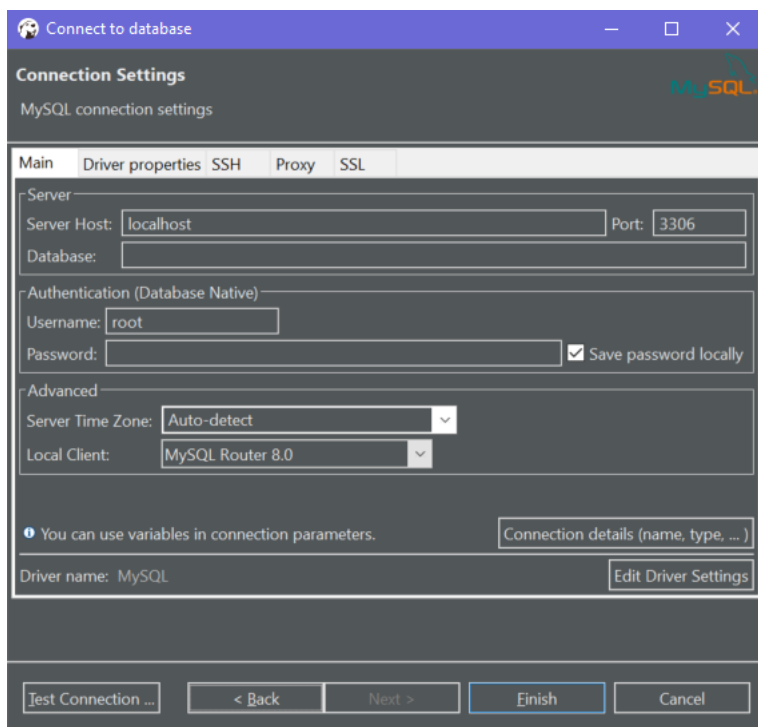
Kao što sam rekao, za izradu ovog projekta koristio sam MySQL Workbench program koji je koristio kao server za MySQL na koji sam se spojio u DBeaveru.



Slika 16. MySQL Server

Slika 16. prikazuje konfiguraciju i pregled servera koji sam lagano stvorio prilikom instalacije programa. Svaki put prilikom pokretanja DBeavera, prvo sam morao u MySQL Workbenchu startati MySQL server.

Dakle, host servera je „localhost“, port je 3306, a ulogiravao sam se putem root računa.



Slika 17. DBeaver konekcija na bazu podataka

Ovako izgleda prozor spajanja na server u DBeaveru.

5.1. WampServer

Slika 18. WampServer



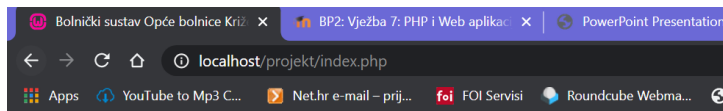
Kao okruženje za izradu web aplikacije koristio sam WampServer. U folder WampServera sam stavio .php datoteke u kojima se nalazio kod pisan u php, te u browseru otišao na domenu svog servera, localhost gdje mogu koristiti spomenute .php datoteke. Kao alat, WampServer u biti stvara server

na mojem računalu, na kojeg se mogu stavljati datoteke i u web okruženju pregledavati i nad njima vršiti druge radnje. Kreirao ga je Romain Bourdon, a zadnja stabilna verzija izašla je 8. svibnja 2020.

6. Web aplikacija i sučelje

Kao što sam već spomenuo, web aplikaciju sam radio u jeziku php čiji kod sam pisao u Visual Studio Code, te uz pomoć alata WampServer.

Web aplikacija ne sadrži sve moguće funkcionalnosti baze podataka, no napravio sam ono što smatram da je dovoljno da se vidi princip na kojem baza funkcionira.



Slika 19. Web aplikacija 1

Unos u tablicu liječi

ID zaposlenika:

ID pacijenta:

Prikaz podataka iz tablice liječi

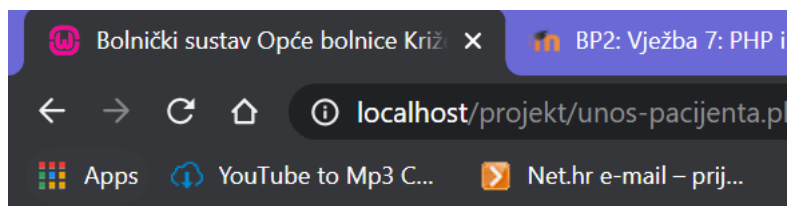
Ana Anic, Hana Klaudije
Aida Borka, Hana Klaudije
Ana Anic, Karlo Henrik
Albert Ezra, Karlo Henrik
Ivo Ivic, Ladislav Nedjeljko
Fabijan Feliks, Ladislav Nedjeljko
Ana Anic, Nada Lazar
Darko Darkic, Nada Lazar
Elena Grgur, Nada Lazar
Lana Lanic, Rino Timon
Adrijan Alfonz, Žarko Uroš
Agata Alka, Zoran Horvat
Agata Alka, Tomislav Tatec

Ispis zaposlenika, njihovih pacijenata, pacijentove povijesti bolesti, sobe i odjela.

na gumb *UPIT*.

Na slici je prikazano sučelje web aplikacije. Izabrao sam u sučelju obraditi i prikazati trigger 1, koji sprječava da se u tablicu liječi unese zaposlenik kojem je radno mjesto kuhar, iz razloga što kuhar ne može liječiti pacijenta. Osim toga, prikazan je i ispis podataka iz te tablice, točnije imena i prezimena medicinskih profesionalaca i njihovih pacijenata. Naposljetku, odlučio sam omogućiti funkcionalnost ispisa onog najzahtjevnijeg upita kojeg sam gore obradio, a taj ispis se događa prilikom pritiska

Što se tiče unosa u tablicu liječi, moguće je upisati ID zaposlenika i ID pacijenta kojeg liječi. Kada se pritisne na gumb *U redu*, ukoliko je upisan ID zaposlenika koji je kuhar, ispisuje se poruka greške iz triggera 1.



Unos voditelja

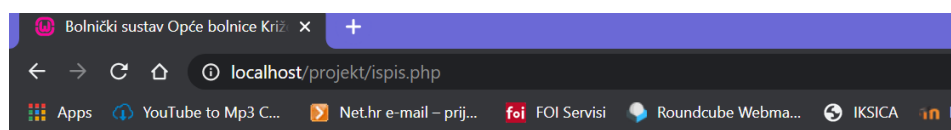
Zaposlenik radnog mjesta kuhar ne može liječiti pacijenta.

Natrag

Slika 20. Web aplikacija 2

Naravno, taj upis se neće izvršiti i korisnik može ići unazad na početnu stranicu aplikacije pritiskom na gumb *Natrag*.

Pritiskom na gumb *UPIT* otvara se sljedeći ekran:



Slika 21. Web aplikacija 3

Ispis upita

Ana Anic, Hana Klaudije, , 111, Infektologija

Aida Borka, Hana Klaudije, , 111, Infektologija

Ana Anic, Karlo Henrik, Prebolio COVID-19. Alergičan na prašinu. Prebolio mononukleozu., 111, Infektologija

Albert Ezra, Karlo Henrik, Prebolio COVID-19. Alergičan na prašinu. Prebolio mononukleozu., 111, Infektologija

Ivo Ivic, Ladislav Nedjeljko, Alergičan na penicilin., 222, Infektologija

Fabijan Feliks, Ladislav Nedjeljko, Alergičan na penicilin., 222, Infektologija

Ana Anic, Nada Lazar, Preboljela šarlah., 333, Kardiologija

Darko Darkic, Nada Lazar, Preboljela šarlah., 333, Kardiologija

Elena Grgur, Nada Lazar, Preboljela šarlah., 333, Kardiologija

Lana Lanic, Rino Timon, , 444, Kardiologija

Adrijan Alfonz, Žarko Uroš, Alergičan na penicilin., 555, Otorinolaringologija

Agata Alka, Zoran Horvat, , 666, Otorinolaringologija

Uspješan ispis podataka.

Natrag

Ispisani su imena i prezimena doktora, pacijenata, povijesti bolesti pacijenata (ukoliko je imaju), broj sobe u kojoj se pacijent nalazi, te odjel u kojem se nalazi soba.

Što se tiče php koda, kod početne datoteke index.php izgleda ovako:

```
<!doctype html>
<html>
  <head>
    <title>Bolnički sustav Opće bolnice Križevci</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <main>
      <header>
        <h1>Unos u tablicu liječi</h1>
      </header>
      <section>
        <article>
          <form action="unos-pacijenta.php" method="post">
            <span>ID zaposlenika:</span><br />
            <input type="text" required name="zaposlenik_id" />
            <br /><br />
            <span>ID pacijenta:</span><br />
            <input type="text" required name="pacijent_id" />
            <input type="submit" value="U redu" />
            <br /><br />
          </form>
        </article>
        <article>
          <h2>Prikaz podataka iz tablice liječi</h2>
          <?php
            $servername = "localhost";
            $database = "projekt";
            $username = "root";
            $password = "qwerty4567";
            $veza = mysqli_connect($servername, $username, $password, $database);
            if($veza->connect_error)
            {
              die("Greška kod povezivanja na bazu podataka: " . $veza->connect_error);
            }
            $upit = "select z.ime, z.prezime, p.ime, p.prezime from zaposlenik z inner join lijeci l on z.zaposlenik_id = l.zaposlenik_id inner j
oin pacijent p on l.pacijent_id = p.pacijent_id";
            $rezultat_ispis = mysqli_query($veza,$upit);
            if(!$rezultat_ispis)
            {
              echo mysqli_error($veza);
            }
            else
            {
              while($red = mysqli_fetch_row($rezultat_ispis))
              {
                echo "<p>" . $red[0] . " " . $red[1] . " , " . $red[2] . " " . $red[3] . "</p>";
              }
            }
          </?php>
        </article>
      </section>
    </main>
  </body>
</html>
```

```

    }
}
?>
</article>
<br /><br />
<br /><br />
<h3>Ispis zaposlenika, njihovih pacijenata, pacijentove povijesti bolesti, sobe i odjela.</h3>
<form action="ispis.php" method="post">
  <input type="submit" value="UPIT" />
</form>
</section>
</main>
</body>
</html>

```

Kod datoteke za unos u tablicu lijeci izgleda ovako:

```

<!doctype html>
<html>
  <head>
    <title>Bolnički sustav Opće bolnice Križevci</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <main>
      <header>
        <h1>Unos voditelja</h1>
      </header>
      <section>
        <?php

          $servername = "localhost";
          $database = "projekt";
          $username = "root";
          $password = "qwerty4567";
          $veza = mysqli_connect($servername, $username, $password, $database);

          if($veza->connect_error)
          {
            die("Greška kod povezivanja na bazu podataka: " . $veza->connect_error);
          }

          $poruka = "";
          if(empty(trim($_POST["zaposlenik_id"])))
          {
            $poruka .= "ID zaposlenika je obavezno za unos!<br />";
          }
        </?php>
      </section>
    </main>
  </body>
</html>

```

```

    }
    else
    {
        $zaposlenik_id = strip_tags($_POST["zaposlenik_id"]);
    }
    if(empty(trim($_POST["pacijent_id"])))
    {
        $poruka .= "ID pacijenta je obavezno za unos!<br />";
    }
    else
    {
        $pacijent_id = strip_tags($_POST["pacijent_id"]);
    }

    if(empty($poruka))
    {
        $unos = "insert into Ijeci values ('$zaposlenik_id', '$pacijent_id')";
        $rezultat_unos = mysqli_query($veza, $unos);
        if(!$rezultat_unos)
        {
            $poruka .= mysqli_error($veza);
        }
        else
        {
            $poruka .= "Uspješan unos podataka.";
        }
    }

    mysqli_close($veza);
?>
<p><?php echo $poruka; ?></p>
<input type="button" value="Natrag" onclick="location.href='index.php'" />
</section>
</main>
</body>
</html>

```

Na kraju, kod datoteke za ispis upita izgleda ovako:

```

<!doctype html>
<html>
<head>
<title>Bolnički sustav Opće bolnice Križevci</title>
<meta charset="utf-8" />
</head>
<body>
<main>

```



```

<header>
  <h1>Ispis upita</h1>
</header>
<section>
  <?php
    $servername = "localhost";
    $database = "projekt";
    $username = "root";
    $password = "qwert4567";
    $veza = mysqli_connect($servername, $username, $password, $database);
    if($veza->connect_error)
    {
      die("Greška kod povezivanja na bazu podataka: " . $veza->connect_error);
    }
    $poruka = "";

    if(empty($poruka))
    {
      $unos = "select concat(z.ime, ' ', z.prezime) as Zaposlenik, concat(p.ime, ' ', p.prezime) as Pacijent,
        povijest as 'Povijest bolesti', broj_sobe as Soba, naziv_odjela as Odjel from zaposlenik z
        inner join lijeci l on z.zaposlenik_id = l.zaposlenik_id inner join pacijent p
        on l.pacijent_id = p.pacijent_id left join povijest_bolesti po on
        p.povijest_bolesti_id = po.povijest_bolesti_id left join soba s on
        p.soba_id = s.soba_id inner join odjel o on s.odjel_id = o.odjel_id;";
      $rezultat_unos = mysqli_query($veza, $unos);
      if(!$rezultat_unos)
      {
        $poruka .= mysqli_error($veza);
      }
      else
      {
        {
          while($red = mysqli_fetch_row($rezultat_unos))
          {
            echo "<p>".$red[0].", ".$red[1].", ".$red[2].", ".$red[3].", ".$red[4]."</p>";
          }
          $poruka .= "Uspješan ispis podataka.";
        }
      }

      mysqli_close($veza);
    }
  ?>
  <p><?php echo $poruka; ?></p>
  <input type="button" value="Natrag" onclick="location.href='index.php'" />
</section>
</main>
</body>
</html>

```

7. Zaključak

Izrada ovog projekta je bio pomalo mukotrpan posao pošto sam se koristio većim brojem alata s kojima nisam dosad imao doticaja. Unatoč tome, mislim da sam uistinu puno toga naučio pošto sam veliku većinu morao raditi samostalno. Nakon ovog projekta, znam kako funkcionira spajanje baza podataka sa serverima, spajanje raznih alata sa tim bazama podataka i sasvim sam utvrdio znanje o ERA modelima i dizajniranju bazi podataka. Najveći izazov u cijelom projektu je bila sintaksna razlika MySQLa i PostgreSQL-a u triggerima i rad sa php jezikom. Mogu odlučno reći da u svojoj budućnosti ne vidim rad sa web stranicama.

8. Popis literature

“About | DBeaver Community.” pristupljeno na 3.1., 2021. <https://dbeaver.io/about/>.

“DBeaver.” In *Wikipedia*, pristupljeno 3.1., 2021.

<https://en.wikipedia.org/w/index.php?title=DBeaver&oldid=998179425>.

GitHub. “Dbeaver/Dbeaver.” pristupljeno 3.1., 2021.

<https://github.com/dbeaver/dbeaver>.

“WampServer.” In *Wikipedia*, pristupljeno 5.1.2021.

<https://en.wikipedia.org/w/index.php?title=WampServer&oldid=970779492>.