



VISOKA POSLOVNA ŠKOLA STRUKOVNIH STUDIJA  
NOVI SAD

# PROJEKAT

## Predmet: Baze podataka

## Tema: Bolnica

Student: Vranješ Momčilo 2020/004023  
Profesor: dr Danijela Tešendić

Novi Sad, 2020

# Sadržaj

1. Opis informacionog sistema
2. Projektovanje baze podataka
  - 2.1. ER model
  - 2.2. Relacioni model
3. Naredbe za kreiranje šeme baze podataka
4. Unos podataka
5. Upiti nad podacima

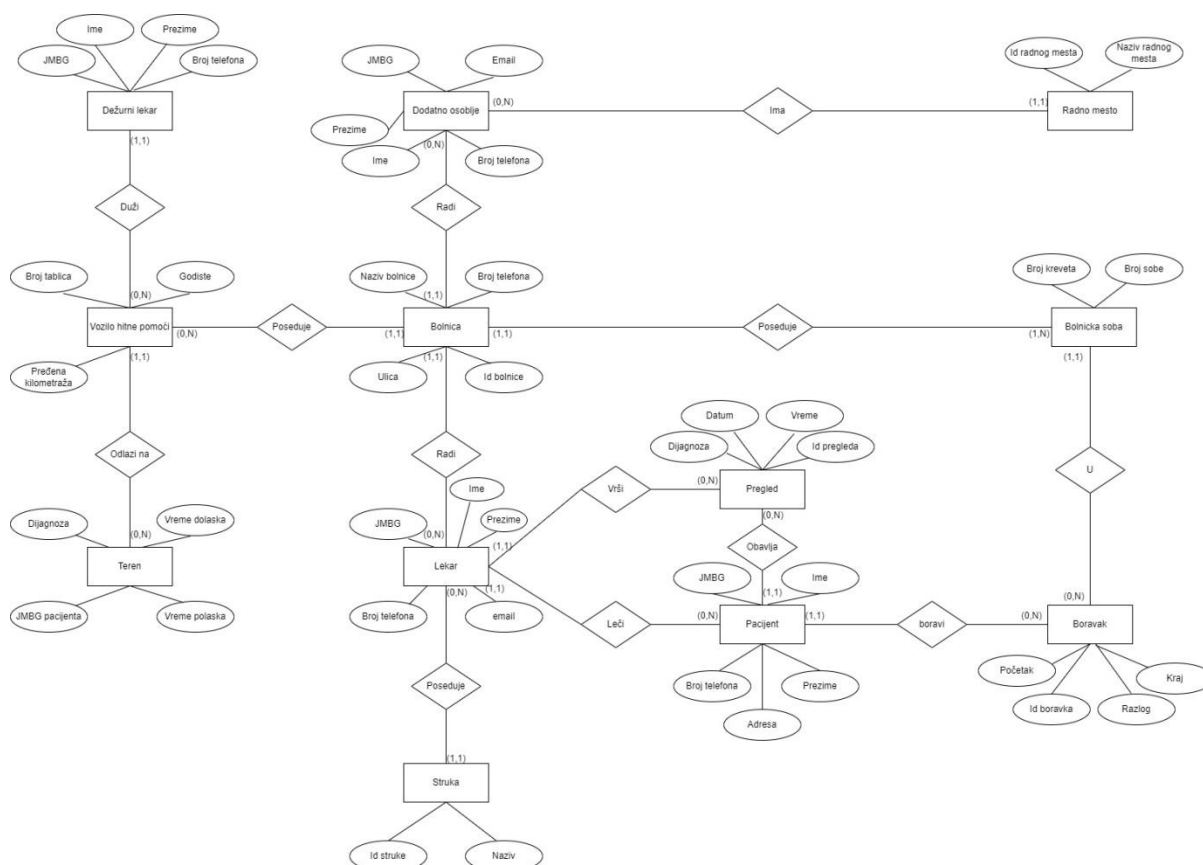
## 1. Opis informacionog sistema

### BAZA BOLNICE

- Bolnica ima id, naziv, ulicu i broj telefona, u bolnici rade lekari koji imaju svoje podatke i struku, struka ima svoj naziv i id.
- Pacijent koji ima svoje podatke i ima svog izabranog lekara, a pored toga može da ima pregled kod specijaliste, pacijent takođe može da boravi u bolničkoj sobi, gde se beleži datum dolaska, odlaska i razlog boravka. Bolnička soba se nalazi u bolnici a ima broj kreveta i broj sobe. Bolnica poseduje i dodatno osoblje koje ima svoje podatke i radno mesto.
- Bolnica takođe poseduje i vozila hitne pomoći, koje duži tim dežurnih lekara koji imaju svoje podatke, a vozilo izlazi na teren pri čemu se zapisuje vreme polaska, dolaska, dijagnoza, ime i prezime pacijenta i njegov JMBG.

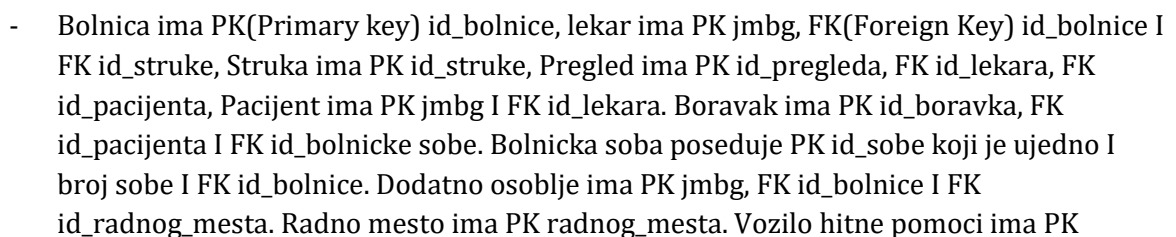
## 2. Projektovanje baze podataka

### 2.1. ER model



- Bolnica je povezana sa bolničkom sobom vezom 1:N, zato što mora postojati barem jedna soba u bolnici a može ih biti i više. Bolnička soba je povezana sa boravkom vezom 0:N, jer u jednoj bolničkoj sobi možemo imati nijedan ili više boravaka. Pacijent je za boravak povezan vezom 0:N jer može boraviti u sobi nijednom ili više puta. Lekar je vezan za pacijenta vezom 0:N jer može biti da nema pacijenta ili može lečiti više pacijenata. Lekar je vezan za pregled vezom 0:N jer može obavljati više pregleda.

## 2.2. Relazioni model



id\_tablice koje su ujedno I tablice vozila I FK id\_bolnice koja ga poseduje. Dezurni lekar ima PK jmbg I FK id\_vozila koje duzi. Teren ima PK id\_terena I id\_vozila koje je otislo na teren kao FK.

### 3. Naredbe za kreiranje šeme baze podataka

```
-- Database generated with pgModeler (PostgreSQL Database
Modeler).
-- pgModeler version: 0.9.2
-- PostgreSQL version: 12.0
-- Project Site: pgmodeler.io
-- Model Author: ---

-- Database creation must be done outside a multicommand
file.
-- These commands were put in this file only as a
convenience.
-- -- object: new_database | type: DATABASE --
-- -- DROP DATABASE IF EXISTS new_database;
-- CREATE DATABASE new_database;
-- -- ddl-end --
--

-- object: bolnica | type: SCHEMA --
-- DROP SCHEMA IF EXISTS bolnica CASCADE;
CREATE SCHEMA bolnica;
-- ddl-end --
-- ALTER SCHEMA bolnica OWNER TO postgres;
-- ddl-end --

SET search_path TO pg_catalog,public,bolnica;
-- ddl-end --

-- object: bolnica.bolnica | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.bolnica CASCADE;
CREATE TABLE bolnica.bolnica (
    id_bolnice integer NOT NULL,
    ime_bolnice text,
    broj_telefona text,
```

```

        ulica text,
        CONSTRAINT "Bolnica_pk" PRIMARY KEY (id_bolnice)
    );
-- ddl-end --

-- object: bolnica.lekar | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.lekar CASCADE;
CREATE TABLE bolnica.lekar (
    jmbg text NOT NULL,
    ime text NOT NULL,
    prezime text NOT NULL,
    email text,
    broj_telefona text,
    id_bolnice integer,
    id_struke integer,
    CONSTRAINT lekar_pk PRIMARY KEY (jmbg)
);
-- ddl-end --

-- object: bolnica.struka | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.struka CASCADE;
CREATE TABLE bolnica.struka (
    id_struke integer NOT NULL,
    naziv text NOT NULL,
    CONSTRAINT struka_pk PRIMARY KEY (id_struke)
);
-- ddl-end --

-- object: bolnica.pregled | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.pregled CASCADE;
CREATE TABLE bolnica.pregled (
    id_pregleda integer NOT NULL,
    datum date NOT NULL,
    dijagnoza text,
    vreme time,

```

```

        id_lekara text,
        id_pacijenta text,
        CONSTRAINT pregled_pk PRIMARY KEY (id_pregleda)
    );
-- ddl-end --

-- object: bolnica.pacijent | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.pacijent CASCADE;
CREATE TABLE bolnica.pacijent (
    jmbg text NOT NULL,
    ime text NOT NULL,
    prezime text NOT NULL,
    broj_telefona integer,
    adresa text,
    id_lekara text,
    CONSTRAINT pacijent_pk PRIMARY KEY (jmbg)
);
-- ddl-end --

-- object: bolnica.boravak | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.boravak CASCADE;
CREATE TABLE bolnica.boravak (
    id_boravka integer NOT NULL,
    razlog text,
    od date,
    "do" date,
    id_pacijenta text,
    id_bolnicke_sobe integer,
    CONSTRAINT odsedanje_pk PRIMARY KEY (id_boravka)
);
-- ddl-end --

-- object: bolnica.bolnicka_soba | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.bolnicka_soba CASCADE;
CREATE TABLE bolnica.bolnicka_soba (

```

```

        id_sobe integer NOT NULL,
        broj_kreveta integer,
        id_bolnice integer,
        CONSTRAINT bolnicka_soba_pk PRIMARY KEY (id_sobe)
    );
-- ddl-end --

-- object: bolnica.dodatno_osoblje | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.dodatno_osoblje CASCADE;
CREATE TABLE bolnica.dodatno_osoblje (
    jmbg text NOT NULL,
    ime text NOT NULL,
    prezime text NOT NULL,
    broj_telefona text,
    email text,
    id_bolnice integer,
    id_radnog_mesta integer,
    CONSTRAINT dodatno_osoblje_pk PRIMARY KEY (jmbg)
);
-- ddl-end --

-- object: bolnica.radno_mesto | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.radno_mesto CASCADE;
CREATE TABLE bolnica.radno_mesto (
    id_radnog_mesta integer NOT NULL,
    ime_radnog_mesta text,
    CONSTRAINT radno_mesto_pk PRIMARY KEY (id_radnog_mesta)
);
-- ddl-end --

-- object: bolnica.vozilo_hitne_pomoci | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.vozilo_hitne_pomoci CASCADE;
CREATE TABLE bolnica.vozilo_hitne_pomoci (
    tablice text NOT NULL,
    godiste integer,

```



```

    predjena_kilometraza integer,
    id_bolnice integer,
    CONSTRAINT vozilo_hitne_pomoci_pk PRIMARY KEY (tablice)
);
-- ddl-end --

-- object: bolnica.dezurni_lekar | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.dezurni_lekar CASCADE;
CREATE TABLE bolnica.dezurni_lekar (
    jmbg integer NOT NULL,
    broj_telefona text,
    ime text NOT NULL,
    prezime text NOT NULL,
    id_vozila text,
    CONSTRAINT dezurni_lekar_pk PRIMARY KEY (jmbg)
);
-- ddl-end --

-- object: bolnica.teren | type: TABLE --
-- DROP TABLE IF EXISTS bolnica.teren CASCADE;
CREATE TABLE bolnica.teren (
    id_terena integer NOT NULL,
    vreme_polaska time,
    vreme_dolaska time,
    dijagnoza text,
    ime_pacijenta text,
    prezime_pacijenta text,
    id_vozila text,
    CONSTRAINT teren_pk PRIMARY KEY (id_terena)
);
-- ddl-end --

-- object: radi | type: CONSTRAINT --
-- ALTER TABLE bolnica.lekar DROP CONSTRAINT IF EXISTS radi
CASCADE;

```

```
ALTER TABLE bolnica.lekar ADD CONSTRAINT radi FOREIGN KEY
(id_bolnice)
REFERENCES bolnica.bolnica (id_bolnice) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: poseduje | type: CONSTRAINT --
-- ALTER TABLE bolnica.lekar DROP CONSTRAINT IF EXISTS
poseduje CASCADE;
ALTER TABLE bolnica.lekar ADD CONSTRAINT poseduje FOREIGN
KEY (id_struke)
REFERENCES bolnica.struka (id_struke) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: vrsi | type: CONSTRAINT --
-- ALTER TABLE bolnica.pregled DROP CONSTRAINT IF EXISTS
vrsi CASCADE;
ALTER TABLE bolnica.pregled ADD CONSTRAINT vrsi FOREIGN KEY
(id_lekara)
REFERENCES bolnica.lekar (jmbg) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: obavlja | type: CONSTRAINT --
-- ALTER TABLE bolnica.pregled DROP CONSTRAINT IF EXISTS
obavlja CASCADE;
ALTER TABLE bolnica.pregled ADD CONSTRAINT obavlja FOREIGN
KEY (id_pacijenta)
REFERENCES bolnica.pacijent (jmbg) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: leci | type: CONSTRAINT --
-- ALTER TABLE bolnica.pacijent DROP CONSTRAINT IF EXISTS
leci CASCADE;
ALTER TABLE bolnica.pacijent ADD CONSTRAINT leci FOREIGN KEY
(id_lekara)
```

```
REFERENCES bolnica.lekar (jmbg) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: odsedao_je | type: CONSTRAINT --
-- ALTER TABLE bolnica.boravak DROP CONSTRAINT IF EXISTS
odsedao_je CASCADE;
ALTER TABLE bolnica.boravak ADD CONSTRAINT odsedao_je
FOREIGN KEY (id_pacijenta)
REFERENCES bolnica.pacijent (jmbg) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: u | type: CONSTRAINT --
-- ALTER TABLE bolnica.boravak DROP CONSTRAINT IF EXISTS u
CASCADE;
ALTER TABLE bolnica.boravak ADD CONSTRAINT u FOREIGN KEY
(id_bolnicke_sobe)
REFERENCES bolnica.bolnicka_soba (id_sobe) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: poseduje | type: CONSTRAINT --
-- ALTER TABLE bolnica.bolnicka_soba DROP CONSTRAINT IF
EXISTS poseduje CASCADE;
ALTER TABLE bolnica.bolnicka_soba ADD CONSTRAINT poseduje
FOREIGN KEY (id_bolnice)
REFERENCES bolnica.bolnica (id_bolnice) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: radi | type: CONSTRAINT --
-- ALTER TABLE bolnica.dodatno_osoblje DROP CONSTRAINT IF
EXISTS radi CASCADE;
ALTER TABLE bolnica.dodatno_osoblje ADD CONSTRAINT radi
FOREIGN KEY (id_bolnice)
REFERENCES bolnica.bolnica (id_bolnice) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
-- ddl-end --

-- object: ima | type: CONSTRAINT --
-- ALTER TABLE bolnica.dodatno_osoblje DROP CONSTRAINT IF
EXISTS ima CASCADE;
ALTER TABLE bolnica.dodatno_osoblje ADD CONSTRAINT ima
FOREIGN KEY (id_radnog_mesta)
REFERENCES bolnica.radno_mesto (id_radnog_mesta) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: poseduje | type: CONSTRAINT --
-- ALTER TABLE bolnica.vozilo_hitne_pomoci DROP CONSTRAINT
IF EXISTS poseduje CASCADE;
ALTER TABLE bolnica.vozilo_hitne_pomoci ADD CONSTRAINT
poseduje FOREIGN KEY (id_bolnice)
REFERENCES bolnica.bolnica (id_bolnice) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: radi | type: CONSTRAINT --
-- ALTER TABLE bolnica.dezurni_lekar DROP CONSTRAINT IF
EXISTS radi CASCADE;
ALTER TABLE bolnica.dezurni_lekar ADD CONSTRAINT radi
FOREIGN KEY (id_vozila)
REFERENCES bolnica.vozilo_hitne_pomoci (tablice) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: izlazi_na | type: CONSTRAINT --
-- ALTER TABLE bolnica.teren DROP CONSTRAINT IF EXISTS
izlazi_na CASCADE;
ALTER TABLE bolnica.teren ADD CONSTRAINT izlazi_na FOREIGN
KEY (id_vozila)
REFERENCES bolnica.vozilo_hitne_pomoci (tablice) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --
```

#### 4. Unos podataka

```
INSERT INTO bolnica.bolnica (id_bolnice, ime_bolnice, broj_telefona, ulica)
VALUES (1, 'Vuk Karadzic', '063123123', 'Vuk Karadzica 22');
```

```
INSERT INTO bolnica.bolnica (id_bolnice, ime_bolnice, broj_telefona, ulica)
VALUES (2, 'Jovan Jovanovic Zmaj', '063222333', 'Jovana Jovanovica 42');
```

```
INSERT INTO bolnica.struka (id_struke, naziv)
VALUES (1, 'Kardiolog');
```

```
INSERT INTO bolnica.struka (id_struke, naziv)
VALUES (2, 'Fizijatar');
```

```
INSERT INTO bolnica.lekar (jmbg, ime, prezime, email, broj_telefona, id_bolnice, id_struke)
VALUES ('1204336500353', 'Marko', 'Markovic', 'markomarkovic@gmail.com', '063556424', 1,
1);
```

```
INSERT INTO bolnica.lekar (jmbg, ime, prezime, email, broj_telefona, id_bolnice, id_struke)
VALUES ('1905246674564', 'Milana', 'Milanovic', 'milanamilanovic@gmail.com', '065363005', 1,
2);
```

```
INSERT INTO bolnica.lekar (jmbg, ime, prezime, email, broj_telefona, id_bolnice, id_struke)
VALUES ('2201246647462', 'Milos', 'Milosevic', 'milosmilosevic@gmail.com', '066342235', 1, 1);
```

```
INSERT INTO bolnica.pacijent (jmbg, ime, prezime, broj_telefona, adresa, id_lekara)
VALUES ('2505241253325', 'Jovan', 'Miletic', '069231412', 'Kralja Petra 5', '1204336500353');
```

```
INSERT INTO bolnica.pacijent (jmbg, ime, prezime, broj_telefona, adresa, id_lekara)
VALUES ('1111352500035', 'Elena', 'Jovic', '066435351', 'Svetojovanska 22', '1905246674564');
```

```
INSERT INTO bolnica.pacijent (jmbg, ime, prezime, broj_telefona, adresa, id_lekara)
VALUES ('1704005346145', 'Kristina', 'Todorovic', '065352553', 'Bulevar Oslobođenja
2', '1905246674564');
```

```
INSERT INTO bolnica.pregled (id_pregleda, datum, dijagnoza, vreme, id_lekara, id_pacijenta)
VALUES (1, '01-10-2023', 'Povremena
tahikardija', '12:10:00', '1204336500353', '2505241253325');
```

```
INSERT INTO bolnica.pregled (id_pregleda, datum, dijagnoza, vreme, id_lekara, id_pacijenta)
VALUES (2, '11-01-2023', 'Pocetna faza
skolioze', '17:22:00', '1905246674564', '1111352500035');
```

```
INSERT INTO bolnica.pregled (id_pregleda, datum, dijagnoza, vreme, id_lekara, id_pacijenta)
VALUES (3,'04-12-2023', 'Bol u sredini grudnog
kosa','08:35:00','2201246647462','1704005346145');
```

```
INSERT INTO bolnica.bolnicka_soba(id_sobe, broj_kreveta, id_bolnice)
VALUES (1, 4,1);
```

```
INSERT INTO bolnica.bolnicka_soba(id_sobe, broj_kreveta, id_bolnice)
VALUES (2, 2,1);
```

```
INSERT INTO bolnica.bolnicka_soba(id_sobe, broj_kreveta, id_bolnice)
VALUES (3, 6,1);
```

```
INSERT INTO bolnica.boravak(id_boravka, razlog, od, "do", id_pacijenta, id_bolnicke_sobe)
VALUES (1, 'Pacijent je zadrzan zbog povremenih nesvestica', '20-10-2023', '22-10-2023',
'2505241253325',1);
```

```
INSERT INTO bolnica.boravak(id_boravka, razlog, od, "do", id_pacijenta, id_bolnicke_sobe)
VALUES (1, 'Pacijent prijavljuje jake bolove u donjem delu ledja', '01-02-2023', '02-02-2023',
'1111352500035',2);
```

```
INSERT INTO bolnica.boravak(id_boravka, razlog, od, "do", id_pacijenta, id_bolnicke_sobe)
VALUES (2, 'Pacijent prijavljuje jake bolove u donjem delu ledja', '01-02-2023', '02-02-2023',
'1111352500035',3);
```

```
INSERT INTO bolnica.radno_mesto(id_radnog_mesta, ime_radnog_mesta)
VALUES (1, 'Portir');
```

```
INSERT INTO bolnica.radno_mesto(id_radnog_mesta, ime_radnog_mesta)
VALUES (2, 'Spremacica');
```

```
INSERT INTO bolnica.radno_mesto(id_radnog_mesta, ime_radnog_mesta)
VALUES (3, 'Domar');
```

```
INSERT INTO bolnica.dodatno_osoblje(jmbg, ime, prezime, broj_telefona, email, id_bolnice,
id_radnog_mesta)
VALUES ('0222535670032', 'Mirko', 'Mirkovic', '064323301','mirkomirkovic@gmail.com',1,3);
```

```
INSERT INTO bolnica.dodatno_osoblje(jmbg, ime, prezime, broj_telefona, email, id_bolnice,
id_radnog_mesta)
VALUES ('1003222144422', 'Nemanja', 'Milutinovic',
'061224251','nemanjamilutinovic@gmail.com',1,1);
```

```
INSERT INTO bolnica.vozilo_hitne_pomoci(tablice, godiste, predjena_kilometraza, id_bolnice)
VALUES ('NS-139-ZX',2015, 130000,1);
```

```
INSERT INTO bolnica.vozilo_hitne_pomoci(tablice, godiste, predjena_kilometraza, id_bolnice)
VALUES ('BG-002-AG',2020, 20000,1);
```

```
INSERT INTO bolnica.dezurni_lekar(jmbg, broj_telefona, ime, prezime, id_vozila)
VALUES (12002301,'061224135', 'Sava','Peric','NS-139-ZX');
```

```
INSERT INTO bolnica.dezurni_lekar(jmbg, broj_telefona, ime, prezime, id_vozila)
VALUES (11124111,'063343533', 'Miona','Maksic','BG-002-AG');
```

```
INSERT INTO bolnica.teren(id_terena, vreme_polaska, vreme_dolaska, dijagnoza, ime_pacijenta,
prezime_pacijenta,id_vozila)
VALUES(1,'11:40:00', '12:00:00', 'Pacijentu je pozlilo', 'Dusan', 'Simic','NS-139-ZX');
```

```
INSERT INTO bolnica.teren(id_terena, vreme_polaska, vreme_dolaska, dijagnoza, ime_pacijenta,
prezime_pacijenta,id_vozila)
VALUES(2,'20:10:00', '20:35:00', 'Prelom ruke', 'Jana', 'Janovic','BG-002-AG');
```

```
INSERT INTO bolnica.teren(id_terena, vreme_polaska, vreme_dolaska, dijagnoza, ime_pacijenta,
prezime_pacijenta,id_vozila)
VALUES(3,'15:15:00', '16:00:00', 'Povisen krvni pritisak', 'Miona', 'Mionic','BG-002-AG');
```

## 5. Upiti nad podacima

Prikazati primere sledećih upita:

### 1) Prikaz svih podataka iz tabele

Query

Query History

```
1 select *
2 from bolnica.pacijent
```

Data Output

Messages

Notifications

	jmbg [PK] text	ime text	prezime text	broj_telefona integer	adresa text	id_lekara text
1	2505241253325	Jovan	Miletic	69231412	Kralja Petra 5	1204336500353
2	1111352500035	Elena	Jovic	66435351	Svetojovanska 22	1905246674564
3	1704005346145	Kristina	Todorovic	65352553	Bulevar Oslobođenja 2	1905246674564

### 2) Filtriranje podataka iz jedne tabele na osnovu nekog kriterijuma

Query Query History ↗ Scratch P:

```

1 select *
2 from bolnica.lekar
3 where ime='Marko'

```

Data Output Messages Notifications

	jmbg [PK] text	ime text	prezime text	email text	broj_telefona text	id_bolnice integer	id_struke integer
1	1204336500353	Marko	Markovic	markomarkovic@gmail.com	063556424	1	1

### 3) Upit sa spajanjem dve tabele i uslovom

```

1 select *
2 from bolnica.dodatno_osoblje o, bolnica.radno_mesto r
3 where o.id_radnog_mesta = r.id_radnog_mesta
4 and r.ime_radnog_mesta = 'Domar'
5

```

Data Output Messages Notifications ↗

	jmbg text	ime text	prezime text	broj_telefona text	email text	id_bolnice integer	id_radnog_mesta integer	ime_radnog text
1	0222535670032	Mirko	Mirkovic	064323301	mirkomirkovic@gmail.com	1	3	Domar

### 4) Upit sa spajanjem tri tabele i uslovom

```

1 select *
2 from bolnica.lekar l, bolnica.pacijent i, bolnica.pregled d
3 where l.jmbg = d.id_lekara
4 and i.jmbg = d.id_pacijenta
5 and l.ime = 'Milos'
6

```

Data Output Messages Notifications ↗

	jmbg text	ime text	prezime text	email text	broj_telefona text	id_bolnice integer	id_struke integer	jmbg text	ime text	prezime text
1	2201246647462	Milos	Milosevic	milosmilosevic@gmail.com	066342235	1	1	1704005346145	Kristina	Todorov

### 5) Group by upit sa spajanjem dve tabele i primenom skupovne funkcije



```

1 select l.jmbg, l.ime, l.prezime, count(j.jmbg)
2 from bolnica.lekar l, bolnica.pacijent j
3 where l.jmbg = j.id_lekara
4 group by l.jmbg, l.ime, l.prezime

```

Data Output Messages Notifications

	jmbg [PK] text	ime text	prezime text	count bigint
1	1905246674564	Milana	Milanovic	2
2	1204336500353	Marko	Markovic	1

#### 6) Ugnježdeni upit

```

1 select *
2 from bolnica.bolnicka_soba
3 where id_sobe = (select id_bolnicke_sobe
4 from bolnica.boravak b
5 where od = '01-02-2023')

```

Data Output Messages Notifications

	id_sobe [PK] integer	broj_kreveta integer	id_bolnice integer
1	3	6	1