

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike u Puli

Tim 31
Servis mobitela

Članovi: Randi Mohorović

Ivan Maurović

Branimir Kuveždić

Filip Milovan

Shana Antonia Lazarić

Kolegij: Baze podataka I

Mentor: doc. dr. sc. Goran Oreški

Pula, svibanj 2022.

SADRŽAJ

<u>1. Uvod</u>	<u>3</u>
<u>2. ER dijagram</u>	<u>3</u>
<u>2.1 Opis ER dijagrama</u>	<u>5</u>
<u>3. Opis tablica</u>	<u>6</u>
<u>4. Opis upita</u>	<u>17</u>
<u>5. Zaključak</u>	<u>21</u>

1. Uvod

U sklopu projekta, članovi tima 31 su na osnovu različitih ideja sklopili dogovor o temi „Servis mobitela“.

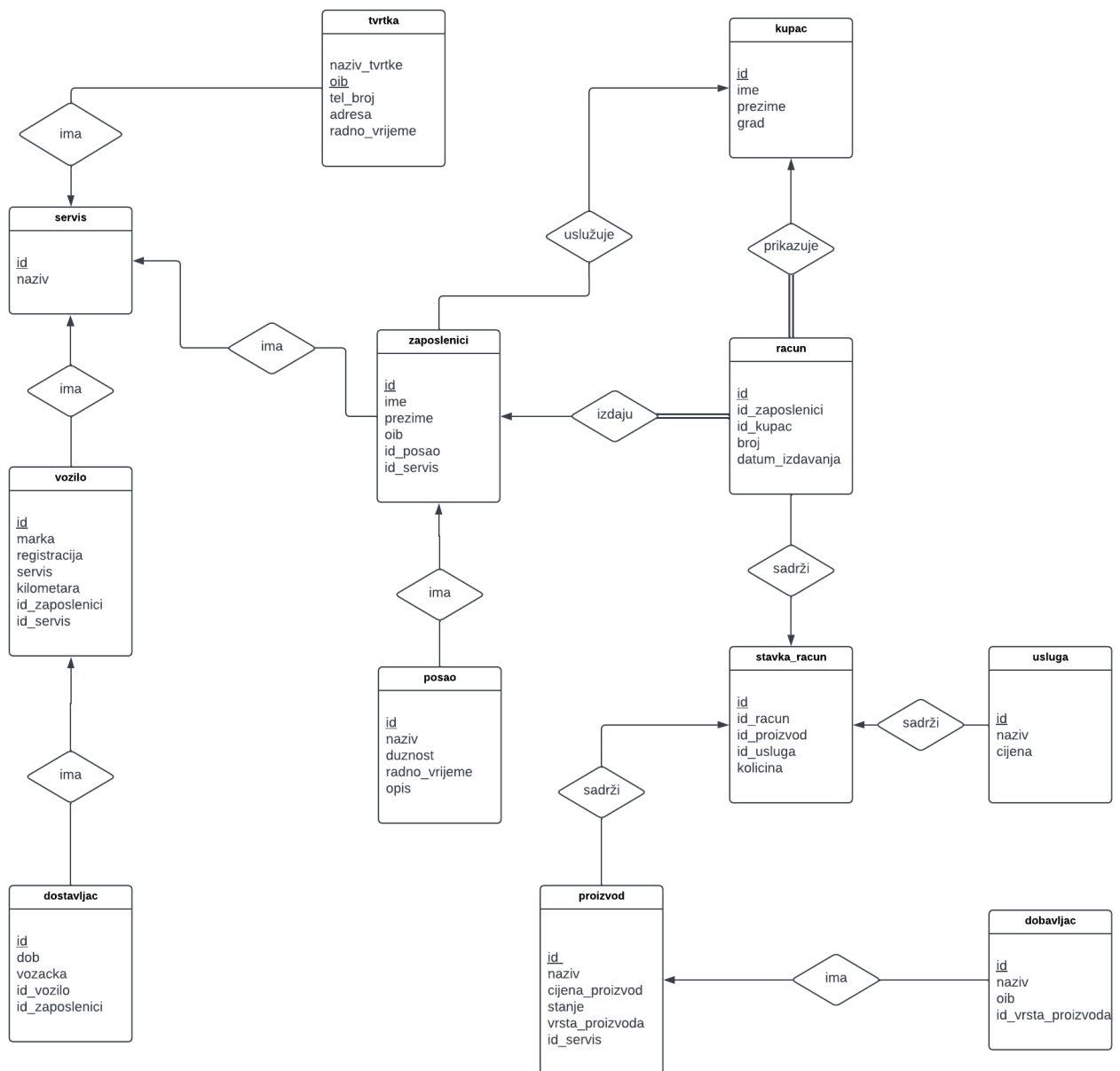
Tijekom prvog sastanka, osim dogovorenih uloga, stvorili smo i koncept samoga projekta.

„Servis mobitela“ je radnja u kojoj se daje usluga, prvobitno za popravak mobitela, ali u sklopu projekta, smo definirali i poslovne odnose među zaposlenicima, te opcija dostave, različitih dostavljačkih vozila i proizvodi.

Ideju smo stvorili na osnovu široke kreativnosti, u smislu da osim uslužne djelatnosti, u trgovačkim obrtima psotoje i računi, djelatnici dr. Kada bi se služili terminologijom u skolu projekta BP1, dobivamo mogućnost stvaranja veza tj. Odnosa između svih faktora ove radnje. Jedan primjer bi bio djelatnik koji izdaje račun. Ovakva vrsta teme za projekt, omogućuje nam da što lakše možemo izmisliti potrebne dijelove projekta.

MySQL Workbench je bio alat za kreiranje same baze, te LucidChart za ER dijagram.

2. ER diagram



2.1 Opis ER dijagrama

Svaki račun je povezan sa zaposlenicima više naprema jedan zato što jedan određeni račun može izdati jedan zaposlenik, a jedan zaposlenik može izdati više računa.

Svaki posao je povezan sa zaposlenicima više naprema jedan iz razloga što jedan zaposlenik može imati više poslova, a jedan posao može imati više zaposlenika.

Svaki dostavljač je povezan sa vozilom jedan naprema jedan zato što jedan dostavljač može imati jedno vozilo.

Zaposlenici su povezani s kupcima jedan naprema više jer jedan kupac može biti poslužen od strane jednog zaposlenika, a jedan zaposlenik može uslužiti više kupaca.

Stavka račun je povezana s računom jedan naprema jedan jer će se stavka račun jednom pojaviti na jednom računu.

Račun je povezan s kupcem jedan naprema više jer jedan određeni račun može dobiti jedan kupac, a više kupaca može dobiti više računa.

Zaposlenici su povezani sa servisom jedan naprema više zato što jedan servis može imati više zaposlenika, a jedan zaposlenik može raditi u jednom servisu.

Servis je povezan sa vozilom više naprema jedan zato što jedan servis može imati više vozila, a jedno vozilo može pripadati jednom servisu.

Tvrtka je povezana sa servisom jedan naprema više jer jedan servis može surađivati sa više tvrtki, a jedna tvrtka može surađivati sa više servisa.

Dobavljač je povezan sa proizvodom jedan naprema više, jer jedan dobavljač može nabaviti više proizvoda, a jedan proizvod može nabaviti jedan dobavljač.

Usluga je povezana sa stavkom račun jedan naprema jedan jer jedna usluga sadrži jednu stavku račun.

3. Opis tablica

Prije samoga stvaranja tablice, DROP DATABASE koristimo kako nebi došlo do neke vrste preklapanja. Također, pošto je bilo puno izmjena i nadodavanja tijekom izrade .sql skripte, DROP DATABASE je također dobrodošao zato što radimo lokalno, te ne postoji rizik da ugrozimo cjelokupnu bazu

```
DROP DATABASE servis_mobitela;  
CREATE DATABASE servis_mobitela;  
USE servis_mobitela;
```

Tablica 1

Početna tablica, s kojom smo započeli osnove. U svakoj tablici, uključujući i ovu, definiran je primarni ključ. U ovom slučaju oib postavljamo kao glavni ključ.

```
CREATE TABLE tvrtka (  
    naziv_tvrtke VARCHAR(50) NOT NULL,  
    oib INTEGER(9) NOT NULL,  
    tel_broj INTEGER NOT NULL,  
    adresa VARCHAR(50) NOT NULL,  
    radno_vrijeme VARCHAR(20) NOT NULL,  
    PRIMARY KEY (oib)  
);
```

Tablica 2

U sklopu tablice „servis“ uneseni su podaci u sklopu samih atributa, gdje se 3 vrijednosti unose, ovisno gdje se servis mobitela nalazi

```
CREATE TABLE servis (  
    id INTEGER NOT NULL,  
    naziv VARCHAR(50) NOT NULL,  
    PRIMARY KEY (id)  
);
```

Tablica 3

Svrha tablice „kupac“ je služila za generično popunjavanje, gdje se nalaze podaci o kupcima te se u njoj nalazi 30 atributa

```
CREATE TABLE kupac (  
    id INTEGER NOT NULL,  
    ime VARCHAR(20) NOT NULL,  
    prezime VARCHAR(20) NOT NULL,  
    grad VARCHAR(50) NOT NULL,  
    PRIMARY KEY (id)  
);
```

Tablica 4

Tablica „zaposlenici“ je slična prethodnoj tablici, zato što također se sastoji o podacima o ljudima. Ovu tablicu ćemo kasnije upotrijebiti kako bi smo stvorili određeni upit.

```
CREATE TABLE zaposlenici (  
    id INTEGER NOT NULL,  
    ime VARCHAR(50) NOT NULL,  
    prezime VARCHAR(50) NOT NULL,  
    oib BIGINT NOT NULL,  
    id_posao INTEGER NOT NULL,  
    id_servis INTEGER NOT NULL,  
    PRIMARY KEY (id)  
);
```

Tablica 5

Kao jedna od glavnih tablica, uneseni su proizvodi koje naša tvrtka ima. U ovoj tablici se također nalazi i jedno drugačije ograničenje, odnosno NUMERIC, koje govori gdje se nalazi decimalna točka prilikom upisa cijene.

```
CREATE TABLE proizvod (  
    id INTEGER NOT NULL,  
    naziv VARCHAR(50) NOT NULL,  
    cijena_proizvod NUMERIC(10,2) NOT NULL,  
    stanje CHAR(2) NOT NULL,  
    vrsta_proizvoda INTEGER NOT NULL,  
    id_servis INTEGER NOT NULL,  
    PRIMARY KEY (id)  
);
```


Tablica 6

Glavna tablica, na koju smo stvarali sve druge, sadrži podatke o uslugama naše tvrtke, gdje se također nalazi ograničenje NUMERIC zato što sadrži cijenu

```
CREATE TABLE usluga (  
    id INTEGER NOT NULL,  
    naziv VARCHAR(50) NOT NULL,  
    cijena NUMERIC(5,2) NOT NULL,  
    PRIMARY KEY (id)  
);
```

Tablice 6 i 7

U ovom opisu tablice, zajedno opisujemo tablice „stavka_racun“ i „racun“ zato što međusobno komuniciraju. U ovim tablicama, nalazi se i strani ključ, uz pomoć kojeg stvaramo „vezu“ između dviju tablica, a točnije to odredimo pomoću naredbe REFERENCES. ON DELETE CASCADE govori kako će se odabrati jedna od različitih vrsta brisanja podataka, odnosno kaskadno. Ograničenje UNIQUE zabranjuje ponavljanje istog id-ja računa i proizvoda.

```
CREATE TABLE racun (  
    id INTEGER NOT NULL,  
    id_zaposlenici INTEGER NOT NULL,  
    id_kupac INTEGER NOT NULL,  
    broj VARCHAR(100) NOT NULL,  
    datum_izdavanja DATETIME NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (id_zaposlenici) REFERENCES zaposlenici (id),  
    FOREIGN KEY (id_kupac) REFERENCES kupac (id)  
);
```

```

CREATE TABLE stavka_racun (
    id INTEGER NOT NULL,
    id_racun INTEGER NOT NULL,
    id_proizvod INTEGER NOT NULL,
    id_usluga INTEGER NOT NULL,
    kolicina INTEGER NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_racun) REFERENCES racun (id) ON DELETE CASCADE,
    FOREIGN KEY (id_proizvod) REFERENCES proizvod (id),
    UNIQUE (id_racun, id_proizvod)
);

```

Tablica 8

Dobavljač je „partner“ naše tvrtke koji dostavlja djelove. Valja napomenuti da smo stvorili i nekoliko pogleda, gdje će se kasnije i dobavljač pronaći kako bi smo stvorili pogled.

```

CREATE TABLE dobavljac (
    id INTEGER NOT NULL,
    naziv VARCHAR(50) NOT NULL,
    oib CHAR(9) NOT NULL,
    id_vrsta_proizvoda INTEGER NOT NULL,
    PRIMARY KEY (id)
);

```

Tablica 8

Dostavljač je također jedna od bitnijih tablica, jer ćemo ju također koristiti u upitima. U ovom slučaju nije potrebno dodavati posebna ograničenja osim NOT NULL, kako bi imali sve upisane podatke o svakom dostavljaču

```
> CREATE TABLE dostavljac (  
    id INTEGER NOT NULL,  
    dob INTEGER NOT NULL,  
    vozacka VARCHAR(50) NOT NULL,  
    id_vozilo INTEGER NOT NULL,  
    id_zaposlenici INTEGER NOT NULL,  
    PRIMARY KEY (id)  
);
```

Tablica 9

Tablica „posao“ nam sadrži vrijednosti titula, odnosno ulogu u našoj poslovnici te sadrži opis njihovog posla

```
CREATE TABLE posao (  
    id INTEGER NOT NULL,  
    naziv VARCHAR(50) NOT NULL,  
    duznost VARCHAR(50) NOT NULL,  
    radno_vrijeme VARCHAR(50) NOT NULL,  
    opis VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id)  
);
```

Tablica 10

Tablica „vozilo“ nastala je nakon „dostavljača“. U našem use case-u, potrebno je i opisati vozila kako bi imali uvid u transport kojim se dostavljači služe.





```
CREATE TABLE vozilo (  
    id INTEGER NOT NULL,  
    marka VARCHAR(50) NOT NULL,  
    registracija DATE NOT NULL,  
    servis DATE NOT NULL,  
    kilometara INTEGER NOT NULL,  
    id_zaposlenici INTEGER NOT NULL,  
    id_servis INTEGER NOT NULL,  
    PRIMARY KEY (id)  
);
```




3. Ograničenja i veze

Tablice tvrtka, servis, zaposlenici, usluga, proizvod, dobavljač, dostavljač, vozilo, posao i kupac sadržavaju primarne ključeve, vrijednosti poput naziva tvrtke, oib, datuma, telefonskog broja, adrese, radnog vremena, grada,...ne smiju bit jednaki nuli.

```
• CREATE TABLE tvrtka (  
    naziv_tvrtke VARCHAR(50) NOT NULL,  
    oib INTEGER(9) NOT NULL,  
    tel_broj INTEGER NOT NULL,  
    adresa VARCHAR(50) NOT NULL,  
    radno_vrijeme VARCHAR(20) NOT NULL,  
    PRIMARY KEY (oib)  
);
```

```
• CREATE TABLE servis (  
    id INTEGER NOT NULL,  
    naziv VARCHAR(50) NOT NULL,  
    PRIMARY KEY (id)  
);
```

-  CREATE TABLE kupac (
 id INTEGER NOT NULL,
 ime VARCHAR(20) NOT NULL,
 prezime VARCHAR(20) NOT NULL,
 grad VARCHAR(50) NOT NULL,
 PRIMARY KEY (id)
);
-  CREATE TABLE zaposlenici (
 id INTEGER NOT NULL,
 ime VARCHAR(50) NOT NULL,
 prezime VARCHAR(50) NOT NULL,
 oib BIGINT NOT NULL,
 id_posao INTEGER NOT NULL,
 id_servis INTEGER NOT NULL,
 PRIMARY KEY (id)
);
-  CREATE TABLE usluga (
 id INTEGER NOT NULL,
 naziv VARCHAR(50) NOT NULL,
 cijena NUMERIC(5,2) NOT NULL,
 PRIMARY KEY (id)
);
-  CREATE TABLE dobavljac (
 id INTEGER NOT NULL,
 naziv VARCHAR(50) NOT NULL,
 oib CHAR(9) NOT NULL,
 id_vrsta_proizvoda INTEGER NOT NULL,
 PRIMARY KEY (id)
);

-  CREATE TABLE dostavljac (
 id INTEGER NOT NULL,
 dob INTEGER NOT NULL,
 vozacka VARCHAR(50) NOT NULL,
 id_vozilo INTEGER NOT NULL,
 id_zaposlenici INTEGER NOT NULL,
 PRIMARY KEY (id)
);
-  CREATE TABLE vozilo (
 id INTEGER NOT NULL,
 marka VARCHAR(50) NOT NULL,
 registracija DATE NOT NULL,
 servis DATE NOT NULL,
 kilometara INTEGER NOT NULL,
 id_zaposlenici INTEGER NOT NULL,
 id_servis INTEGER NOT NULL,
 PRIMARY KEY (id)
);
-  CREATE TABLE posao (
 id INTEGER NOT NULL,
 naziv VARCHAR(50) NOT NULL,
 duznost VARCHAR(50) NOT NULL,
 radno_vrijeme VARCHAR(50) NOT NULL,
 opis VARCHAR(100) NOT NULL,
 PRIMARY KEY (id)
);

U tablici proizvod se prilikom unosa provjerava da je unesena cijena pozitivan broj

```
CREATE TABLE proizvod (  
    id INTEGER NOT NULL,  
    naziv VARCHAR(50) NOT NULL,  
    cijena_proizvod NUMERIC(10,2) NOT NULL,  
    stanje CHAR(2) NOT NULL,  
    vrsta_proizvoda INTEGER NOT NULL,  
    id_servis INTEGER NOT NULL,  
    PRIMARY KEY (id) ,  
    CHECK (cijena_proizvod > 0)  
);
```

U tablici stavka_racun uz vrijednosti koje ne smiju biti jednake nuli i primarni ključ, sadržavaju i strane ključeve :

```
FOREIGN KEY (id_racun) REFERENCES racun (id) ON DELETE CASCADE,  
FOREIGN KEY (id_proizvod) REFERENCES proizvod (id),
```

kao i UNIQUE što znači da u istom stupcu mogu biti iste vrijednosti id_racun i id_proizvod, ali kombinacija tih dviju vrijednosti mora biti jedinstvena

```
UNIQUE (id_racun, id_proizvod)
```

Tablica racun sadržava dva strana ključa za zaposlenike i kupce

```
FOREIGN KEY (id_zaposlenici) REFERENCES zaposlenici (id),  
FOREIGN KEY (id_kupac) REFERENCES kupac (id)
```


Primjer unosa u tablicu vozila:

```
INSERT INTO vozilo values (1, 'Mercedes', STR_TO_DATE('15.5.2021.', '%d.%m.%Y.'), STR_TO_DATE('1.5.2021.', '%d.%m.%Y.'), 150000, 16, 1),
(2, 'BMW', STR_TO_DATE('26.6.2021.', '%d.%m.%Y.'), STR_TO_DATE('12.5.2022.', '%d.%m.%Y.'), 200000, 17, 1),
(3, 'Peugeot', STR_TO_DATE('2.5.2021.', '%d.%m.%Y.'), STR_TO_DATE('2.5.2021.', '%d.%m.%Y.'), 100000, 18, 1),
(4, 'Fiat', STR_TO_DATE('3.5.2021.', '%d.%m.%Y.'), STR_TO_DATE('24.3.2021.', '%d.%m.%Y.'), 250000, 19, 1),
```

4. Opis upita

Prikaz jedinstvenih prezimena zaposlenika

```
SELECT DISTINCT prezime
FROM zaposlenici;
```

Prvi upit prikazuje najjeftiniji proizvod koristeći funkciju ASC za uzlazni poredak i

```
SELECT *
FROM proizvod
ORDER BY cijena_proizvod ASC
LIMIT 1;
```

Drugi upit prikazuje najskuplji proizvod koristeći funkciju DESC za silazni poredak

```
SELECT *  
FROM proizvod  
ORDER BY cijena_proizvod DESC  
LIMIT 1;
```

Treći upit ispisuje račun gdje je više od 1 proizvoda. Vrš se grupiranje po id-evima proizvoda i pomoću funkcije SUM se sumira ukupna količina proizvoda

```
select p.*  
from proizvod as p  
where p.id in (select id_proizvod  
from stavka_racun  
group by id_proizvod  
HAVING SUM(kolicina) > 1);
```

Četvrti upit prikazuje sva vozila marke BMW koji su registrirani zadnjih 12 mjeseci tako što se prvo pronađe marka "BMW" i zatim sva registrirana vozila te marke i pomoću intervala se izračuna količina vozila registriranih 12 mjeseci unazad do sada

```
select * from vozilo  
where marka = 'BMW' and registracija > now() - interval 12 month;
```

Peti upit prikazuje sve proizvode koji su nabavljeni od strane Mobile Parts d.o.o. tvrtke tako što prvo pronađemo id vrste proizvoda i nakon toga dobavljača odnosno tvrtku

```
select *  
from proizvod  
where vrsta_proizvoda in (SELECT id_vrsta_proizvoda  
FROM dobavljac  
WHERE naziv = 'Mobile Parts d.o.o');
```

Šesti upit prikazuje sve radnike u poslovnici Stoja tako da prvo pronađemo zaposlenike koji rade u poslovnici i zatim id poslovnice čiji je naziv "Servis Mobitela-Stoja"

```
select *  
from zaposlenici  
where id_posao in (SELECT id FROM posao WHERE duznost = 'Rad u poslovnici' )  
and id_servis in(select id FROM servis  
WHERE naziv = 'Servis Mobitela-Stoja');
```

Sedmi upit prikazuje zaposlenike koji imaju položenu vozačku za teretna vozila; prvo tražimo id zaposlenika koji rade kao dostavljači i zatim tražimo id dostavljača koji imaju vozačku dozvolu "C1" ili "C" kategorije.

```
SELECT *  
FROM zaposlenici  
WHERE id IN ( SELECT id_zaposlenici FROM dostavljac WHERE vozacka = 'C1' OR vozacka = 'C');
```

Osmi upit prikazuje dostavljače i njihova vozila tako što tražimo id-eve zaposlenika i njihovih vozila

```
SELECT *  
FROM vozilo AS v, zaposlenici AS z  
WHERE v.id_zaposlenici = z.id;
```

Deveti upit prikazuje marke teretnih vozila tako da prvo pronađemo id-eve zaposlenika i njihovih vozila i zatim zaposlenike dostavljače koji imaju vozačku za teretna vozila

```
SELECT *  
FROM vozilo AS v, zaposlenici AS z  
WHERE v.id_zaposlenici = z.id AND  
z.id IN ( SELECT id_zaposlenici FROM dostavljac WHERE vozacka = 'C1' OR vozacka = 'C');
```

Deseti upit prikazuje stavku računa sa najvećom količinom proizvoda

```
-- 10. prikaz stavke racuna sa najvecom kolicinom proizvoda
```

```
SELECT *  
FROM stavka_racun  
WHERE kolicina IN (SELECT MAX(kolicina) FROM stavka_racun);
```

Jedanaesti upit prikazuje datum zadnjeg izdanog racuna koristeći funkciju DESC za silazni poredak

```
-- 11. Prikaz datuma izdavanja zadnjeg racuna
SELECT datum_izdavanja
FROM racun
ORDER BY broj DESC
LIMIT 1;
```

Dvanaesti upit prikazuje prvog kupca koji je kupovao u tvrtci tako da se prvo koristi funkcija ASC za uzlazni poredak te se zatim selektira samo prvi rezultat ograničenjem LIMIT.

```
SELECT *
FROM kupac
WHERE id IN ( SELECT id_kupac FROM racun ORDER BY broj ASC )
LIMIT 1;
```

Trinaesti upit prikazuje servise i njihove menadere tako da se prvo odredi select gdje je dužnost 'Menadžer servisa' te se pomocu LEFT OUTER JOINA tablica spoji sa tablicom zaposlenici i servis

```
-- 13. Prikaz servisa i njihovih menadera
SELECT *
FROM zaposlenici AS z
LEFT OUTER JOIN servis AS s ON z.id_servis = s.id
WHERE id_posao IN ( SELECT id FROM posao WHERE duznost = 'Menadžer servisa');
```

5. Zaključak

U sklopu kolegija Baze Podataka 1 i uz mentorstvo izvršnog profesora i asistenta, znanje stečeno na predavanjima i vježbama prenijeli smo u vlastiti projekt. Odabirom teme „servis mobitela“ imali smo pregršt ideja s kojima smo lako manipulirali i realizirali u projekt. Proveli smo dosta vremena na planiranje i predviđali određene probleme, te se tako prilagodili samome projektu. Naš use case doveo nas je do konkretnog znanja, gdje smo uz pomoć materijala sa vježbi i predavanja, došli do rješenja našega problema. Također, dobro nam je došlo timsko okruženje i snalaženje u programu MySQL Workbench-u. U sklopu kolegija Baze Podataka 1, pojedinačno je sa svatko sa svojim znanjem pridonjeo kvaliteti i zradi ovoga projekta.