Sveučilište Jurja Dobrile u Puli

**Planiranje ekskurzije**

**Tim 9**

**Mentor:** doc. dr. sc. Goran Oreški

**Predmet:** Baze Podataka 2

**Izradili:**

Alan Bubalo

Thomas Siard

Mihael Blažević

Ernest Dudaković

Alen Orbanić

Luka Brodarič

Pula, 10.01.2022.

SADRŽAJ

[1. UVOD 3](#_Toc73810870)

[2. OPIS POSLOVNOG PROCESA ER DIJAGRAMA 4](#_Toc73810871)

[2.1. OPIS ER DIJAGRAMA 5](#_Toc73810872)

[3. DETALJAN OPIS SVIH TABLICA, ATRIBUTA I DOMENA S KOMENTAROM 7](#_Toc73810873)

[3.1 Tablica – grad kretanja 7](#_Toc73810874)

[3.2 Tablica – nacin putovanja 7](#_Toc73810875)

[3.3. Tablica – grad\_nacin\_putovanja 8](#_Toc73810876)

[3.4. Tablica – drzava 8](#_Toc73810877)

[3.5. Tablica -destinacija 9](#_Toc73810878)

[3.6. Tablica – vodic 9](#_Toc73810879)

[3.7. Tablica – ponuda 10](#_Toc73810880)

[3.8. Tablica – osiguranje 11](#_Toc73810881)

[3.9. Tablica – korisnik 11](#_Toc73810882)

[3.10. Tablica – provjera 12](#_Toc73810883)

[4. PREGLED POSLOVNIH PRAVILA U BAZI 13](#_Toc73810884)

[5. POPIS UPITA I NJIHOVO OBJAŠNJENJE 14](#_Toc73810885)

[1. UPIT 14](#_Toc73810886)

[2. UPIT 15](#_Toc73810887)

[3. UPIT 15](#_Toc73810888)

[4. UPIT 16](#_Toc73810889)

[5. UPIT 17](#_Toc73810890)

[6. UPIT 18](#_Toc73810891)

[7. UPIT 19](#_Toc73810892)

[8. UPIT 20](#_Toc73810893)

[9. UPIT 21](#_Toc73810894)

[10. UPIT 22](#_Toc73810895)

[6. ZAKLJUČAK 23](#_Toc73810896)

SADRŽAJ SLIKA

[Slika 1 ER dijagram 4](#_Toc73810897)

[Slika 2 grad\_kretanja (tablica) 7](#_Toc73810898)

[Slika 3 nacin\_putovanja (tablica) 7](#_Toc73810899)

[Slika 4 grad\_nacin\_putovanja (tablica) 8](#_Toc73810900)

[Slika 5 drzava (tablica) 8](#_Toc73810901)

[Slika 6 destinacija (tablica) 9](#_Toc73810902)

[Slika 7 vodic (tablica) 9](#_Toc73810903)

[Slika 8 ponuda (tablica) 10](#_Toc73810904)

[Slika 9 osiguranje (tablica) 11](#_Toc73810905)

[Slika 10 korisnik (tablica) 11](#_Toc73810906)

[Slika 11 provjera (tablica) 12](#_Toc73810907)

[Slika 12 Umetanje vrijednosti u tablicu "korisnik" 13](#_Toc73810908)

[Slika 13 Upit 1. 14](#_Toc73810909)

[Slika 14 Rezultat 1. Upita 14](#_Toc73810910)

[Slika 15 Upit 2. 15](#_Toc73810911)

[Slika 16 Upit 3. 15](#_Toc73810912)

[Slika 17 Rezultat 3. Upita 15](#_Toc73810913)

[Slika 18 Upit 4. 16](#_Toc73810914)

[Slika 19 Rezultat 4. Upita 16](#_Toc73810915)

[Slika 20 Upit 5. 17](#_Toc73810916)

[Slika 21 Rezultat 5. Upita 17](#_Toc73810917)

[Slika 22 Upit 6. 18](#_Toc73810918)

[Slika 23 Rezultat 6. Upita 18](#_Toc73810919)

[Slika 24 Upit 7. 19](#_Toc73810920)

[Slika 25 Rezultat 7. Upita 19](#_Toc73810921)

[Slika 26 Upit 8. 20](#_Toc73810922)

[Slika 27 Rezultat 8. Upita 20](#_Toc73810923)

[Slika 28 Upit 9. 21](#_Toc73810924)

[Slika 29 Rezultat 9. Upita 21](#_Toc73810925)

[Slika 30 Upit 10. 22](#_Toc73810926)

[Slika 31 Rezultat 10. Upita 22](#_Toc73810927)

# UVOD

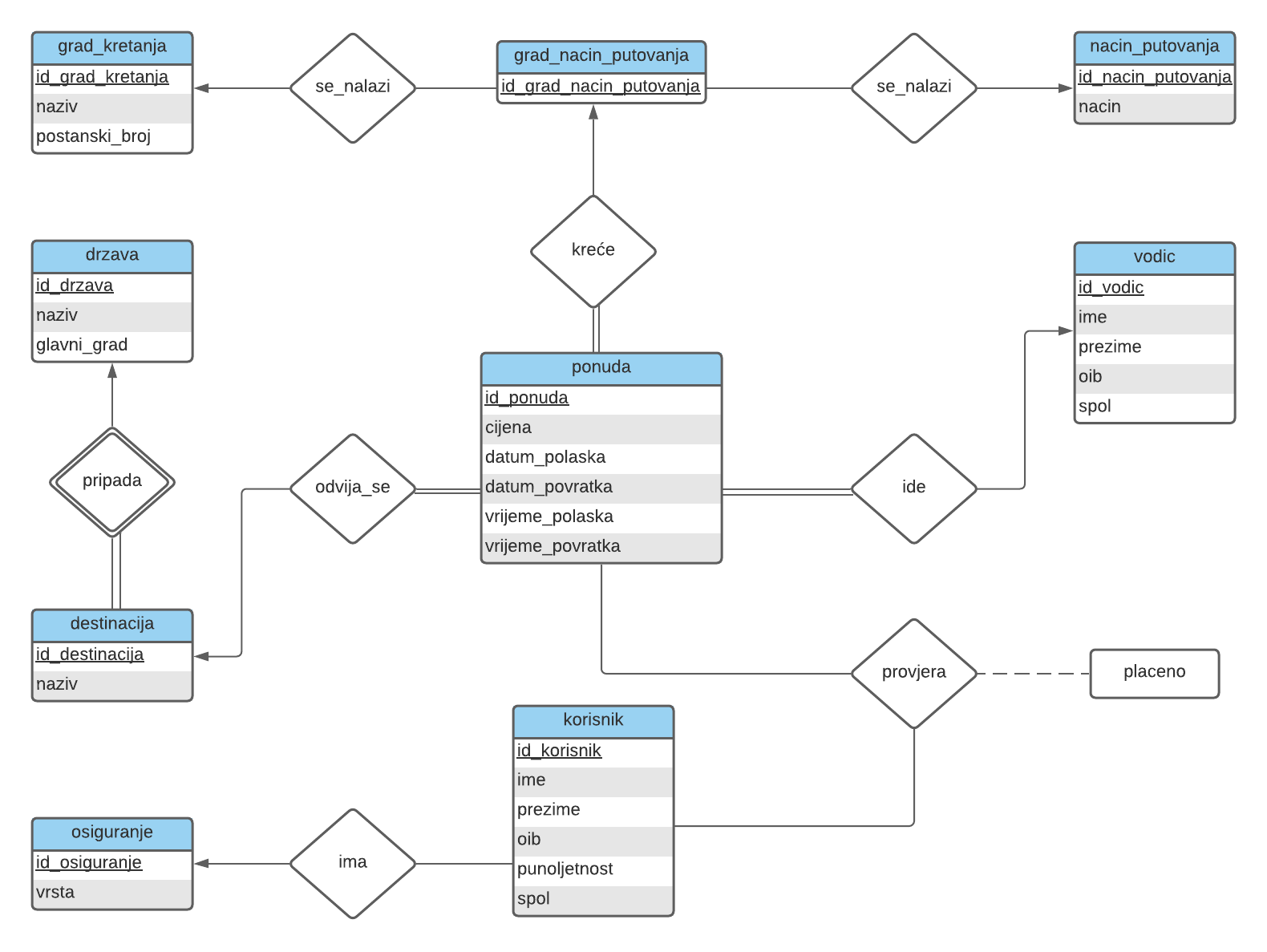
Izabrali smo temu „Sustav za isplatu plaća u auto salonu“ zbog toga što smo očekivali da ta tema neće biti popularna ili zauzeta i moći ćemo ju odraditi bez većih poteškoća. Imali smo članove tima sa iskustvom o ovoj temi pa smo ju odlučili obraditi.

Započeli smo zadavanjem relacija koje opisuju sustav za isplatu plaća u auto salonu . Navedene relacije su: *pozicija, isplata, klasa, automobil, kupac, zaposlenik, racun, praznici, placanje, servis i prisutnost.* Svaka pojedina relacija sadrži primarni ključ, tj. odgovarajući ID radi toga da podatci budu jedinstveni. Neke relacije u sebi imaju strane ključeve koji su povezani sa pripadajućim relacijama. U svakoj tablici naravno možemo promijeniti podatke ako bismo htjeli imati više ili manje opcija korištenjem tih relacija.

Cilj našeg projekta je bio da izradimo sustav za isplatu plaća u auto salonu. Pokušali smo da naša baza podatka bude što realnija i praktičnija tako da veze između relacija imaju više logike. Program koji smo koristili za ovaj projekt je MySQL Workbench, Lucidchart za izradu ER dijagrama.

# OPIS POSLOVNOG PROCESA ER DIJAGRAMA

Ovdje Vam je prikazan naš ER dijagram. Napravili smo ga vrlo jednostavnim i urednim kako biste Vi što bolje razumjeli. Na slici su Vam prikazani svi entiteti koje smo napravili i njihova povezanost. Neki entiteti su povezani drugačije od ostalih jer postoje različite mogućnosti. U nastavku je detaljno opisan svaki korak našeg ER dijagrama.



Slika 1 ER dijagram

## 2.1. OPIS ER DIJAGRAMA

1. ***Grad\_kretanja 🡨🡪 grad\_nacin\_putovanja 🡨🡪 nacin\_putovanja:***

*Grad\_nacin\_putovanja* je kombinacija entiteta *grad\_kretanja* i *nacin\_putovanja*. Veze između oba dva entiteta nazivaju se *se\_nalazi*. Kada imamo običnu ravnu crtu pored entiteta to znači da može biti više njih, a kada imamo crtu sa strelicom, ona nam indicira da može biti samo jedan entitet. U ovom primjeru *grad\_kretanja* i *grad\_nacin\_putovanja* vidimo da samo u jednom gradu se možemo kretati na više načina. Zato nam je kod *grad\_kretanja* strelica, a kod *grad\_nacin\_putovanja* obična crta jer postoji vise načina putovanja u tom jednom gradu. U drugom primjeru kod *grad\_nacin\_putovanja* i *nacin\_putovanja* vidimo da postoji samo jedan način putovanja i on se nalazi u *grad\_nacin\_putovanja*. Na kraju svega *grad\_nacin\_putovanja* je kombinacija ostala dva entiteta.

1. ***Grad\_nacin\_putovanja 🡨🡪 ponuda:***

Veza *krece* spaja *grad\_nacin\_putovanja* i *ponuda*. Ovdje vidimo da *ponuda* ima duple crte to nam znači da ponuda mora postojati za naš entitet *grad\_nacin\_putovanja.* Mora postojati nekakva ponuda da bismo mogli krenuti u jedan grad i onda još način na koji ćemo putovati.

1. ***Ponuda 🡨🡪 vodic:***

Veza *ide* spaja entitete *ponuda* i *vodic*. Ovo nam je isti primjer kao kod prijašnjeg slučaja. Treba postojati više ponuda i u njima se treba nalaziti jedan vodič.

1. ***Ponuda 🡨🡪 destinacija:***

Veza *odvija\_se* spaja entitete *ponuda* i *destinacija*. Uvijek nam ponuda treba postojati i u više ponuda se može pojaviti ta jedna destinacija.

1. ***Destinacija 🡨🡪 drzava:***

Veza *pripada* spaja entitete *destinacija* i *drzava*. Ovdje imamo jednu slabu vezu što znači da više destinacija mora i može pripadati u jednoj državi, obrnuto nije moguće i nema smisla.

1. ***Ponuda 🡨🡪 provjera:***

Ovdje nam je veza zapravo entitet *provjera* i onda se spaja sa ponudom. Htjeli smo zapravo ovdje napraviti tako da sa vezom *provjera* provjeravamo je li sve plaćeno s ponudama koje smo dobili. Zato imamo iscrtkane crte sa komentarom, tj. imenom „placeno“.

1. ***Provjera 🡨🡪 korisnik:***

Opet ovdje nemamo pravu vezu između entiteta *korisnik* i *provjera*. Na oba dvije strane može biti više slučajeva, npr. više korisnika može imati više provjera i obrnuto.

1. ***Korisnik 🡨🡪 osiguranje:***

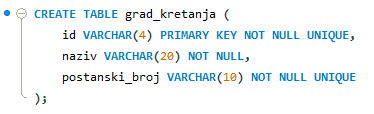
Veza *ima* spaja entitete *korisnik* i *osiguranje*. Više korisnika može imati jedno osiguranje, ali ne više od toga, zato ne može jedno osiguranje imati više korisnika.

# DETALJAN OPIS SVIH TABLICA, ATRIBUTA I DOMENA S KOMENTAROM

## 3.1 Tablica – grad kretanja

Opis:

* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* naziv 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL), služi za ime grada
* postanski\_broj 🡪 VARCHAR (duljina znakova 10), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)

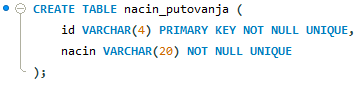


Slika 2 grad\_kretanja (tablica)

## 3.2 Tablica – nacin putovanja

Opis:

* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* nacin 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)

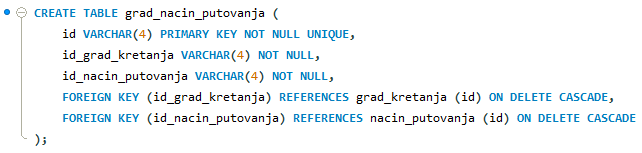


Slika 3 nacin\_putovanja (tablica)

## 3.3. Tablica – grad\_nacin\_putovanja

Opis:

* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* id\_grad\_kretanja 🡪 VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL)
* id\_nacin\_putovanja 🡪 VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL)

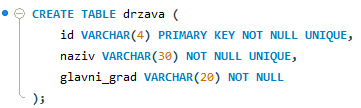


Slika 4 grad\_nacin\_putovanja (tablica)

## 3.4. Tablica – drzava

Opis:

* naziv 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* naziv 🡪 VARCHAR (duljina 30 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* gladni\_grad 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)

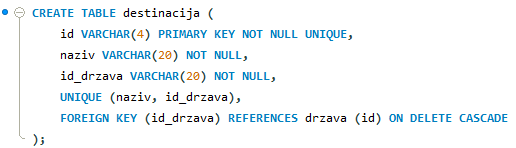


Slika 5 drzava (tablica)

## 3.5. Tablica -destinacija

Opis:

* id 🡪 VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* naziv 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)
* id\_drzava 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)

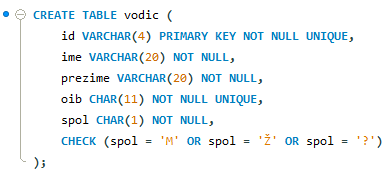


Slika 6 destinacija (tablica)

3.6. Tablica – vodic

Opis:

* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* ime 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)
* prezime 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)
* oib 🡪 VARCHAR (duljina 11 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* spol 🡪 VARCHAR (duljina 1 znaka), ne smije biti prazan (NOT NULL)

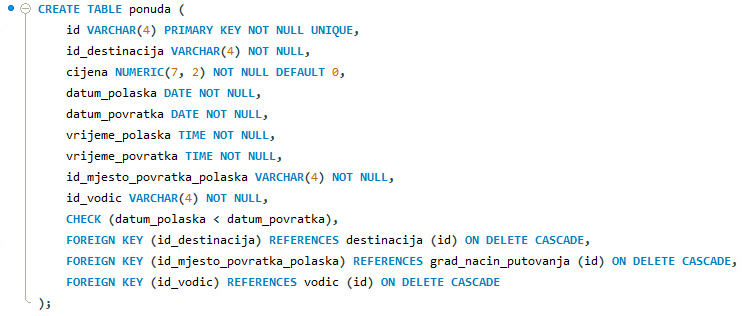


Slika 7 vodic (tablica)

## 3.7. Tablica – ponuda

Opis:

* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* id\_destinacija 🡪 VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL)
* cijena 🡪NUMERIC (7, 2) 7 brojeva i dva su decimalna, ne smije biti prazan (NOT NULL) i počinje od 0
* datum\_polaska 🡪 datum sljedećeg polaska ne smije biti 0 (NOT NULL)
* datum\_povratka 🡪 datum sljedećeg povratka ne smije biti 0 (NOT NULL)
* vrijeme\_polaska 🡪 vrijeme sljedećeg polaska ne smije biti 0 (NOT NULL)
* vrijeme\_povratka 🡪 vrijeme sljedećeg povratka ne smije biti 0 (NOT NULL)
* id\_mjesto\_povratka\_polaska 🡪 VARCHAR (duljina 4 znakova) i ne smije biti prazan (NOT NULL)
* id\_vodic 🡪 VARCHAR (duljina 4 znakova) i ne smije biti prazan (NOT NULL)

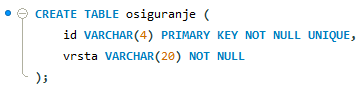


Slika 8 ponuda (tablica)

## 3.8. Tablica – osiguranje

Opis:

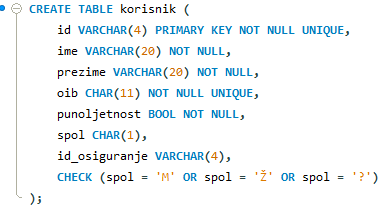
* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* vrsta 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)



Slika 9 osiguranje (tablica)

## 3.9. Tablica – korisnik

Opis:

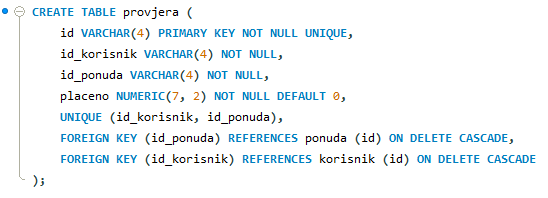
* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* ime 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)
* prezime 🡪 VARCHAR (duljina 20 znakova), ne smije biti prazan (NOT NULL)
* oib 🡪 CHAR (duljina 11 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* punoljetnost 🡪 BOOL (istina ako je punoljetan, laž ako nije), ne smije biti prazan (NOT NULL)
* spol 🡪 CHAR (duljina 1 znaka)
* id\_osiguranje 🡪 VARCHAR (duljina 4 znakova)

Slika 10 korisnik (tablica)

## 3.10. Tablica – provjera

Opis:

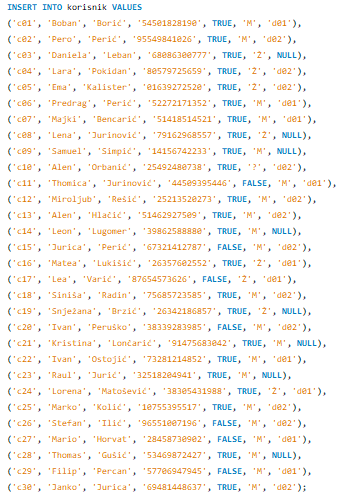
* id 🡪 primarni ključ, VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL) i mora biti unikatan (jedinstven)
* id\_korisnik 🡪 VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL)
* id\_ponuda 🡪 VARCHAR (duljina 4 znakova), ne smije biti prazan (NOT NULL)
* placeno 🡪 NUMERIC (7, 2) 7 brojeva i dva su decimalna, ne smije biti prazan (NOT NULL) i počinje od 0



Slika 11 provjera (tablica)

# PREGLED POSLOVNIH PRAVILA U BAZI

Unos podataka u relacije vrši se tako da umetnemo naredbu „INSERT INTO“, zatim slijedi ime tablice i nakon nje se piše „VALUES“. Nakon šta ovo napišemo možemo ubaciti podatke koje god hoćemo. Na slici su prikazani naši podatci. U zagradi upisujemo vrijednosti i sa zarezom ih odvajamo jedan od drugog i idemo po redoslijedu kao što smo u tablicama napisali atribute.



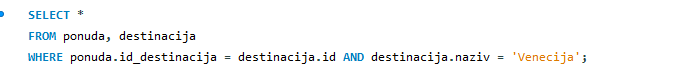
Slika 12 Umetanje vrijednosti u tablicu "korisnik"

# POPIS UPITA I NJIHOVO OBJAŠNJENJE

## 1. UPIT

Pitanje glasi: Ispiši sve id-eve ponuda kojima je destinacija Venecija.

Odgovor:



Slika 13 Upit 1.

Objašnjenje:

* Prvo radimo kartezijev produkt tablica ponuda i destinacija
* Zatim napravimo natural join između te dvije tablice na način da ih povežemo istim atributom (destinacija id)
* te dodamo uvjet da ispiše samo one idjeve čiji je naziv Venecija

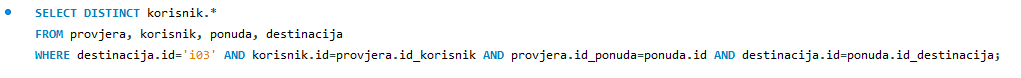


Slika 14 Rezultat 1. Upita

## 2. UPIT

Pitanje glasi: Ispiši sve korisnike koji su putovali u Barcelonu.

Odgovor:



Slika 15 Upit 2.

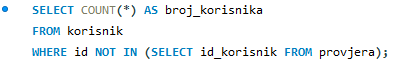
Objašnjenje:

* Odabiremo sve atribute korisnika i stavljamo distinct zbog slucaja da se ne ponavlja neki korisnik koji je otišao više puta u Barcelonu
* Zatim u from dodajemo provjeru, korisnik, ponudu i destinaciju kako bi napravili natural join između korisnika i provjere, ponude i
* provjere, destinacije i ponude te odabiremo da je id destinacije i03 jer je to id od barcelone.

## 3. UPIT

Pitanje glasi: Ispiši broj korisnika koji nisu otišli na ekskurzije.

Odgovor:



Slika 16 Upit 3.

Objašnjenje:

* Funkcijom count u selectu odabiremo da nam se prebroje svi redovi koji ce nam se ispisati iz tablice korisnik
* u kojima se korisnikov id ne nalazi u tablici provjera pod atributom korisnik id, što bi kao uvjet značilo
* da oduzmemo sve korisnike sa onim korisnicima koji su uplaćivali novac za ekskurziju

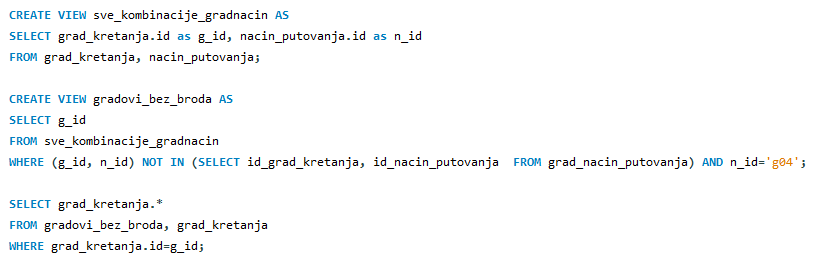


Slika 17 Rezultat 3. Upita

## 4. UPIT

Pitanje glasi: Ispiši sve gradove iz kojih se ne može putovati brodom.

Odgovor:



Slika 18 Upit 4.

Objašnjenje:

* Da bismo pojednostavili zadatak, napravili smo 2 pogleda.
* Prvi pogled je sve\_kombinacije\_gradnacin, spoj svih kombinacije koje mogu postojati kad spojimo
* relacije grad\_kretanja i nacin\_putovanja tj. kartezijev produkt te dvije relacije
* Drugi pogled je gradovi\_bez\_broda.
* U ovom pogledu se odvija oduzimanje našeg prvog pogleda sa relacijom grad\_nacin\_putovanja tako
* da se automatski prikažu samo kombinacije koje ne postoje u relaciji grad\_nacin\_putovanja
* Također moramo napomenut da nam prikaže razliku te dvije relacije s uvjetom da prikaže samo one gdje nema kombinacije s brodom.
* Na kraju spojimo relaciju gradovi\_bez\_broda i grad\_kretanja tako da nam prikaže sve atribute iz grad\_kretanja.

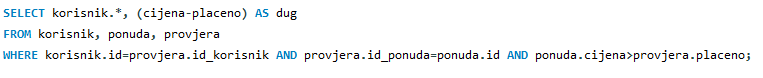


Slika 19 Rezultat 4. Upita

## 5. UPIT

Pitanje glasi: Ispiši sve korisnike koji nisu još u potpunosti platili put i koliko su još dužni.

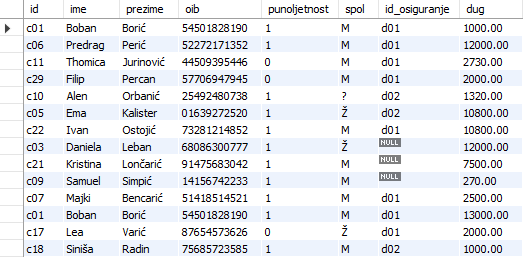
Odgovor:



Slika 20 Upit 5.

Objašnjenje:

* U selectu odabiremo ispisati sve atribute korisnika i dug, dug je razlika izmedju cijene neke ponude i iznosa koji je korisnik platio,
* ako je korisnik platio dovolju ili vecu svotu nego je trebao, nece biti ispisan u tablici.
* Napravili smo kartezijev produkt tablica korisnik, ponuda i provjera.
* Kako bi mogli napraviti razliku atributa cijene i placenog iznosa, trebali smo spojiti tablice provjeru i ponudu,
* zatim smo trebali spojiti korisnika i provjeru kako bi mogli preko provjere spojiti korisnika i ponudu.
* zadnji uvjet nam je da je cijena ponude veća od placenog iznosa korisnika, to znaci da korisnik nije platio dovoljan iznos.

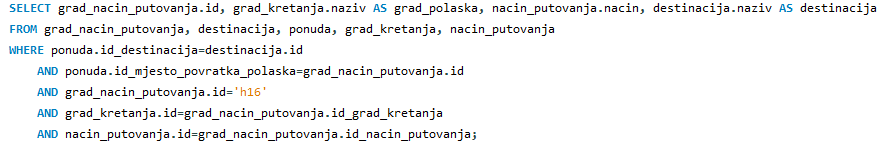


Slika 21 Rezultat 5. Upita

## 6. UPIT

Pitanje glasi: Ispiši id autobusa koji putuje iz Osijeka i kamo putuje.

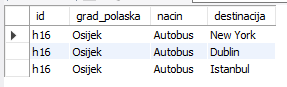
Odgovor:



Slika 22 Upit 6.

Objašnjenje:

* Prikažemo id autobusa koji putuje iz Osijeka, naziv 'Osijek', sam nacin 'Autobus' i sve destinacije gdje taj autobus dolazi
* tako da spojimo relacije grad\_nacin\_putovanja, grad\_kretanja, nacin\_putovanja, destinacija i ponuda sa pripadajućim id-evima.
* Nakon toga postavimo uvjet da prikaže samo id autobusa iz Osijeka tj. 'h16'.

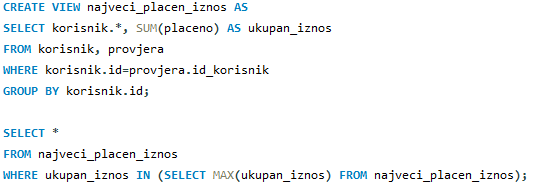


Slika 23 Rezultat 6. Upita

## 7. UPIT

Pitanje glasi: Ispiši koji korisnik je platio najviše.

Odgovor:



Slika 24 Upit 7.

Objašnjenje:

* Prvo radimo pogled tablice nazvan najveci\_placen\_iznos koji prikazuje iznose koje su
* korisnici do sada platili, ako neki korisnik ima više uplata, te uplate mu se zbrajaju
* Odabiremo sve atribute korisnika i novi atribut ukupan\_iznos koji je suma svih placanja
* zatim radimo natural join tablica korisnik i provjera kako bi spojili korisnika sa iznosem novca koji je platio
* te grupiramo tablicu po id-u korisnika kako bi raspoznali koji je korisnik vršio uplatu i koliko je puta vršio uplatu
* zatim radimo novu selekciju tog pogleda u kojoj je uvjet da je dobiven ukupan iznos selekcija najveceg ukupnog iznosa iz tog pogleda.

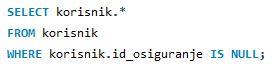


Slika 25 Rezultat 7. Upita

## 8. UPIT

Pitanje glasi: Ispiši korisnike koji nemaju nikakvu vrstu osiguranja.

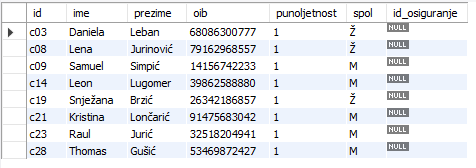
Odgovor:



Slika 26 Upit 8.

Objašnjenje:

* Za ovaj zadatak ćemo prikazati sve korisnike gdje im je vrijednost id\_osiguranje NULL i dobimo korisnike koji nemaju origuranje.

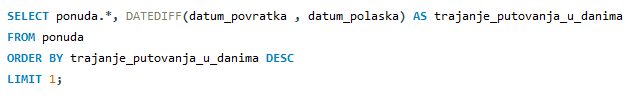


Slika 27 Rezultat 8. Upita

## 9. UPIT

Pitanje glasi: Ispiši putovanje koje je najduže trajalo.

Odgovor:



Slika 28 Upit 9.

Objašnjenje:

* Kako bismo došli do datuma koji je najdulje trajao možemo izvući samo relaciju "ponuda".
* Iskoristimo funkciju DATEDIFF koja nam daje razliku neka 2 datuma u danima.
* Umjesto da koristimo funkciju MAX za dobivanje najveće razlike, možemo samo sortirati ponude po razlikama silazno i označiti samo prvi
* pošto će logično na prvom mjestu biti najveći broj pa onda sve manji i manji na ostalim mjestima.

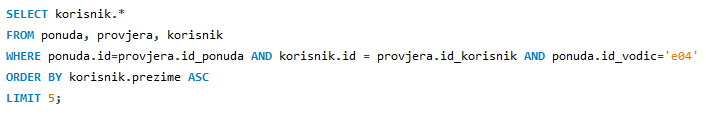


Slika 29 Rezultat 9. Upita

## 10. UPIT

Pitanje glasi: Ispiši prvih pet korisnika sortiranih po prezimenu (silazno) kojima je Thomas Vlakić vodič.

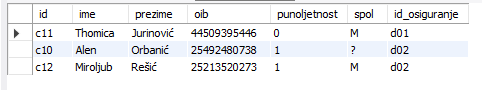
Odgovor:



Slika 30 Upit 10.

Objašnjenje:

* Spojili smo relacije ponuda, provjera i korisnik sa poklapajućim id-evima.
* Onda prikažemo samo provjere gdje se javljao vodić sa id-em 'e04' pošto je taj id Thomasa Vlakića.
* Nakon toga možemo lako sortirati relaciju po prezimenu i pokazati samo prvih pet sa LIMIT-om.



Slika 31 Rezultat 10. Upita

# ZAKLJUČAK

Ovim projektom smo napravili bazu podataka za planiranje ekskurzije. Najteži dio nam je bio posložiti sve dijelove projekta tako da dođemo do kraja. Ovaj projekt nam je definitivno pomogao i proširio znanje. Zbog pandemije nismo mogli se naći uživo, pa smo sve svi čuli preko Interneta i za ostali dio projekta smo koristili aplikacije koje nam je asistent preporučio. Puno toga treba napraviti i dodati da bi baza podataka bilo kojeg projekta ili za posao bila blizu savršenstva. Pokušali smo napraviti što teže i da ima raznih funkcionalnosti naša baza podataka. Treba dosta vremena potrošiti kako bismo napravili još bolju bazu. Moglo bih se još dodati vrijednosti za svaku relaciju i još razne stvari, ali mislimo da smo dovoljno toga napravili i zadovoljni smo sa krajnjim rezultatom.