

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

Goran Filinić

DB za Kafić

PROJEKTNI RAD ZA KOLEGIJ BAZE PODATAKA 2

Mentor/Mentorica:

Prof. dr. sc. Kornelije Rabuzin

Varaždin, siječanj 2020.

Sadržaj

Sadržaj	ii
1. Uvod.....	1
1.1. Korištene tehnologije i alati.....	1
1.1.1. MySQL	1
1.2. PyCharm.....	2
2. ERA model	3
3. Opisi entiteta.....	5
4. MySQL.....	7
4.1. Koraci korištenja alata.....	7
5. Primjeri upita	10
5.1. Jednostavni upiti	10
5.2. Složeni upiti.....	10
5.3. Okidači	11
6. Aplikacijsko sučelje za rad s bazom podataka	12
7. Zaključak.....	14
Popis literature	15

1. Uvod

Cilj ovog projekta je implementacija jednostavne baze podatka te korištenja nekoliko funkcionalnosti baza u nekom programskom jeziku: u ovom slučaju **Python**. Glavna podatkovna domena za ovaj projekt biti će kafić. Pošto mi je ovo prvi projekt gdje gradim svoju bazu, a nekoliko sezona sam radio kao konobar, ovo je idealna domena jer smatram da je meni najbliža.

Komunikacija i interakcija s bazom podataka ostvarena je pomoću jednostavnog sučelja. Sučelje je realizirano pomoću pythonovog standardnog GUI paketa **Tkinter** [1]. Baza te njeni podaci pohranjeni su u lokalnoj **MySQL** [2] bazi.

1.1. Korištene tehnologije i alati

1.1.1. MySQL



Slika 1. Logotip MySQL

MySQL je jedan od najpoznatijih open source sustava za upravljanje relacijskim bazama podatka podržan od Oracle-a. Neke od najvećih organizacija današnjice koriste ovaj sustav[3]:

- Facebook
- Google
- Adobe
- Alcatel Lucent
- Zappos

Što ne čudi obzirom na jednostavnost naprema svojim konkurentima poput samog Oracle Database-a ili Microsoftovog SQL Server-a. MySQL naravno nudi i plaćene licence s punom podrškom.

1.2. PyCharm



Slika 2. Logotip PyCharm

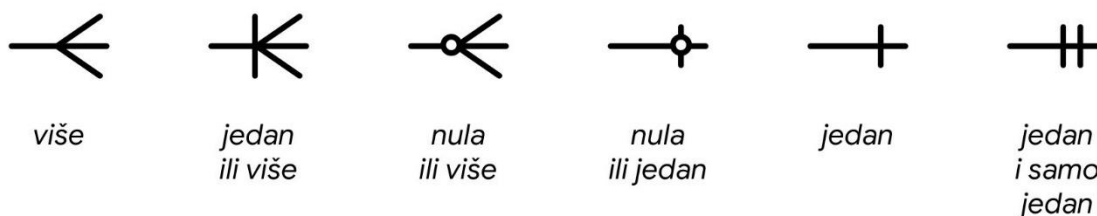
PyCharm je IDE sustav za sve Pythonove alate razvijen od strane Jet Brainsa. Ima mogućnosti poput *Intelligent Coding Assistance and Editor*, *Built-in Developer Tools*, *Web Development* i dr. U ovom projektu se u ovom sučelju razvija GUI.

Za više informacija vidi → [PyCharm](#)

2. ERA model

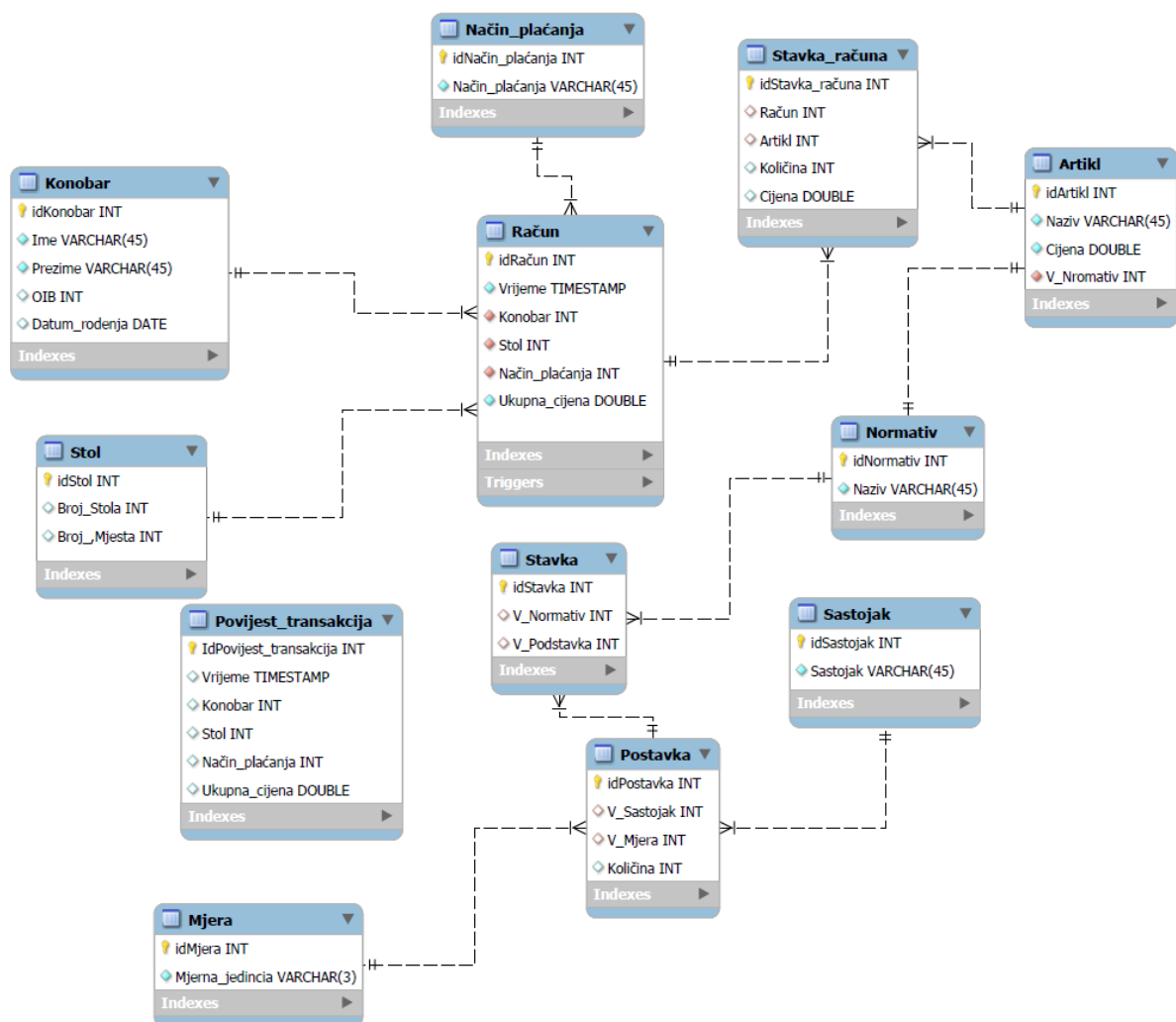
ERA dijagram je dijagram čiji su osnovni elementi entitet, atributi entiteta te veze među entitetima. ERA dijagram najčešće se koristi u razvoju softvera i dizajnu podatkovnih modela, odnosno baza podataka. [4] Veze u ERA modelu imaju svoju **kardinalnost**. Tako veze mogu biti sljedećih kardinalnosti:

- veza jedan na jedan (1:1)
- veza jedan na više (1:N)
- veza više na više (N:M)



Slika 3: Simboli kardinalnosti veza u ERA dijagramu

Kako bi vezu N:M mogli ostvariti potrebno je dodatni tzv. *Slabi entitet*, odnosno dodatni entitet koji sadrži vanjske ključeve entiteta koji su u N:M vezi. U nastavku slijedi ERA model naše baze.



Slika 4. ERA model Kafića

Kako bi ovaj ERA model bio potpun nedostaje još nekoliko tablica, prvenstveno za skladište. No u svrhu ovog projekta nisam ulazio u razvijanju cijelog sustava te sam umjesto toga izabrao razviti sustav za unos novih artikala.

3. Opisi entiteta

Konobar
idKonobar INT
Ime VARCHAR(45)
Prezime VARCHAR(45)
OIB INT
Datum_rodenja DATE

Tablica Konobar sadrži osnovne podatke o konobaru. Atributi su Ime, Preime, OIB, Datum rođenja.

Primarni ključ ove tablice je vanjski ključ na tablicu Račun (veza 1:N) kako bismo mogli vidjeti tko „kuca“ račun.

Račun
idRačun INT
Vrijeme TIMESTAMP
Konobar INT
Stol INT
Način_plaćanja INT
Ukupna_cijena DOUBLE

Tablica Račun glavna je tablica našeg projekta. Ona sadrži atribute Vrijeme, zatim vanjski ključ na tablicu Konobar, Stol, Način_plaćanja, te nakraju Ukupnu cijenu.






Veza s konobarom već je definirana, ali i atributi Stol i Način plaćanja također imaju vezu 1:N. Valja napomenuti kako je sama tablica Račun povezana s tablicom Razrada teme

Stol
idStol INT
Broj_Stola INT
Broj_Mjesta INT

Tablica **Stol** jednostavno sadrži atribute **Broja_Stola** i **Broj_Mjesta** koji stol ima. Povezana je na Račun.

Način_plaćanja
idNačin_plaćanja INT
Način_plaćanja VARCHAR(45)



Tablica **Način_plaćanja** ima jedan istoimeni atribut kako bi mogli imati više načina plaćanja za račun.

Stavka_računa ▼	
	idStavka_računa INT
	Račun INT
	Artikl INT
	Količina INT
	Cijena DOUBLE




Kako bismo povezali tablicu **Artikl** i tablicu **Račun** moramo dodati slabiji entitet pošto je kardinalnost veze **N:M**. Taj slabiji entitet realiziramo kao **Stavka_računa** koja osim vanjskih ključeva sadrži i **Količinu** koja se odnosi na broj artikla koji želimo na račun i **Cijena** u kojoj se zapisuje količina*cijena artikla.

Artikl ▼	
	idArtikl INT
	Naziv VARCHAR(45)
	Cijena DOUBLE
	V_Normativ INT




Tablica **Artikl** ima atribute **Naziv** i **Cijenu** ali i vanjski ključ na **Normativ** koji služi da vidimo što i koliko toga ide u zadani artikl. Vrijedi napomenuti kako je ova veza **1:1** što je jedini primjer takve veze za ovaj projekt. Jedan Artikl može imati samo jedan normativ.

Normativ ▼	
	idNormativ INT
	Naziv VARCHAR(45)

Tablica **Normativ** ima samo naziv za normativ (*primjer: N1, N2...*). Osim s Artiklom povezana je i sa slabijim entitetom **Stavka** vezom **1:N**.

Stavka ▼	
	idStavka INT
	V_Normativ INT
	V_Podstavka INT

Stavka je slabiji entitet za **Normativ** i tablicu **Podstavka**.

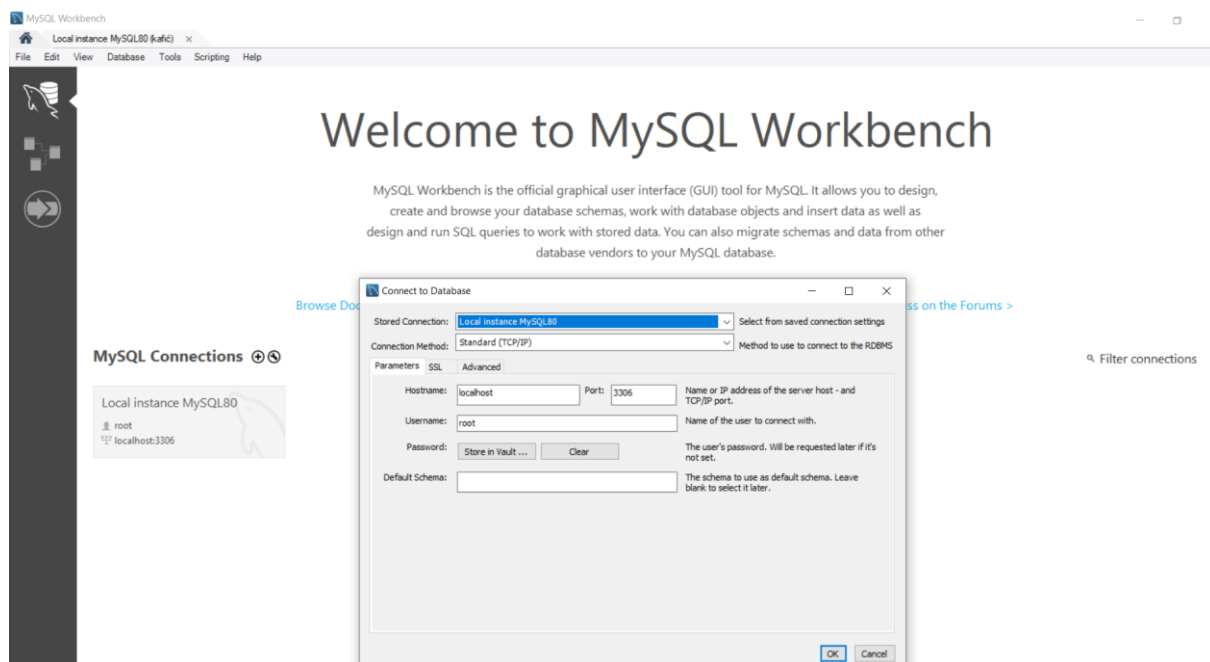
Postavka ▼	
	idPostavka INT
	V_Sastojak INT
	V_Mjera INT
	Količina INT

Kako bismo povezali sve sastojke i dobili punu sliku bez da imamo kopija unosa za artikle tablica **Podstavka** je slabiji entitet za još dva entiteta: **Sastojak** i **Mjera**. Te tablice ne sadrže ništa više osim svog naziva i primarnog ključa. Za kraj nam ostaje još ne povezana tablica **Povijest_transakcija** koju ispunjava okidač. No o tome nešto kasnije.

4. MySQL

Pomoću MySQL alata realizirana je baza podataka. Slijedeće poglavlje će prikazati malo više o radu u njemu,

4.1. Koraci korištenja alata



Slika 5. Login prilikom paljenja programa

Kada upalimo program otvori nam se glavni prozor di se možemo spojiti s bazama s kojima smo već bili spojeni, ili napraviti novu konekciju. No prije no što se spojimo na samo bazu na lijevoj alatnoj traci možemo izabrati **Models** u kojima možemo visualno raditi ERA model te ga kasnije sinkronizirati s našom bazom podataka.

Models + - >

Project_Kafic

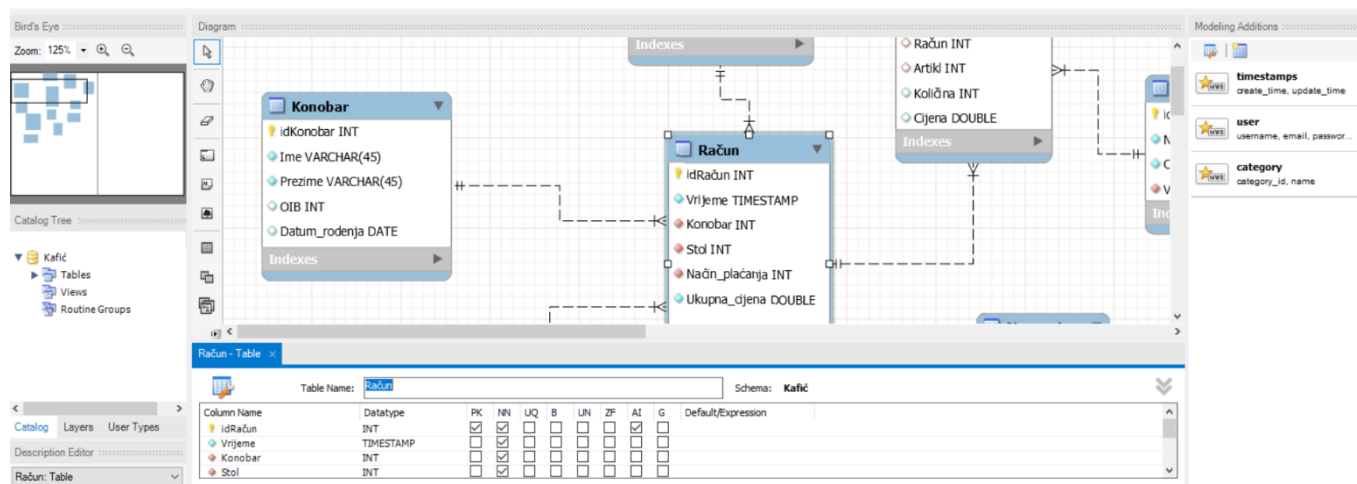


...Faks\BP2\Project_Kafana

Kafic

16 Jan 20, 06:27

(Slika 6.) Vidimo kako izgleda korisničko sučelje dok modeliramo tablicu račun. U ovom sučelju moguće je dodati sve u već unaprijed napravljene forme. To uključuje: stupce, vanjskih ključeve, okidače, indexe pa čak i sam unos.



Slika 6. Primjer modeliranja u Models

Nakon što smo gotovi s modeliranjem, na alatnoj traci imamo *Database* → *Syncronise Models* opciju (Slika 7.) koja nam sinkronizira sve naše promjene s povezanom bazom podataka. Usput nam daje i opciju da spremimo SQL za cijeli model kao zasebni file.

Synchronize Model with Database

Connection Options

- Sync Options
- Connect to DBMS
- Select Schemas
- Retrieve Objects
- Select Changes to Apply
- Review DB Changes
- Synchronize Progress

Set Parameters for Connecting to a DBMS

Stored Connection: Select from saved connection settings

Connection Method: Method to use to connect to the RDBMS

Parameters **SSL** Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

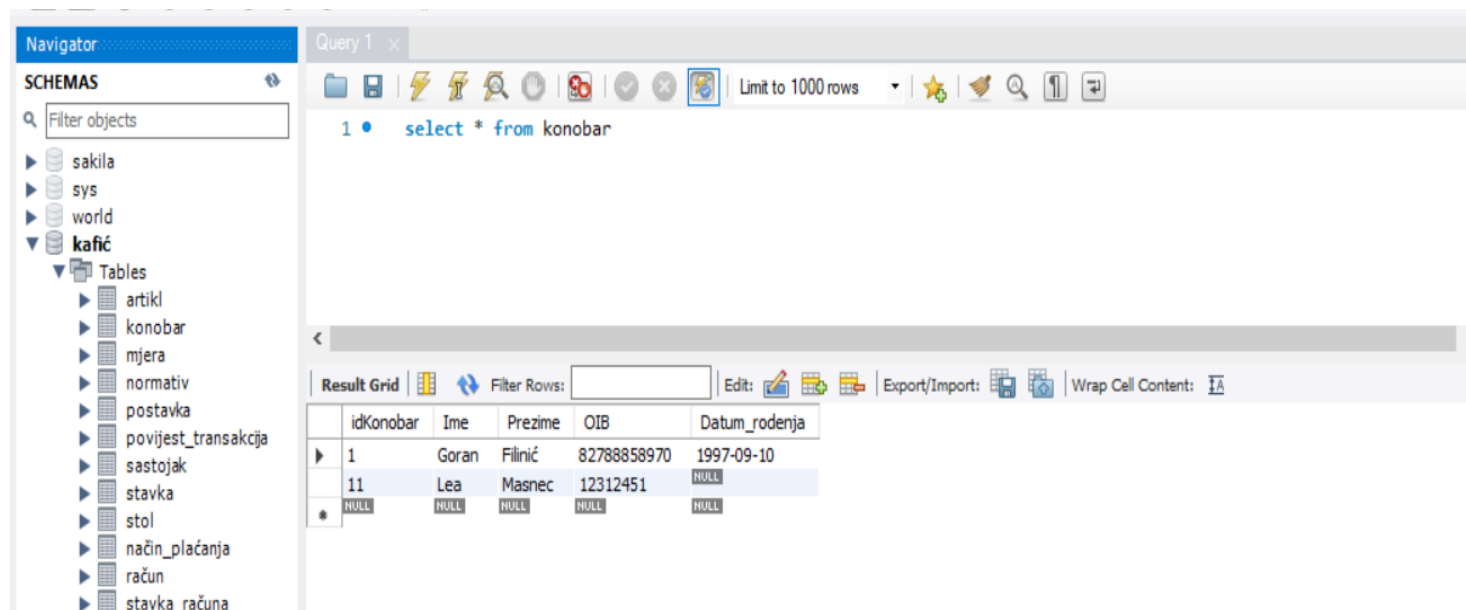
Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Slika 7. Sinkronizacija Modela

Za kraj kada se povežemo na bazu imamo klasičnu SQL konzolu (Slika 8,) kao i objekt s našim shemama ili administracijskim alatima.



Slika 8. Primjer MySQL console

5. Primjeri upita

5.1. Jednostavni upiti

```
SELECT Count(*)
FROM Konobar;
---Vraća ukupan broj konobara u kafiću

SELECT DISTINCT (Cijena)
FROM Artikl;
---Vraća jedinstvene vrijednosti cijene u tablici artikl

SELECT MAX (Ukupna_cijena)
FROM Račun;
---Vraća najskuplji račun
```

5.2. Složeni upiti

```
SELECT (Konobar.ime,Konobar.prezime,
Račun.stol,Račun.vrijeme,Račun.ukupna_cijena)
FROM Konobar,Račun
WHERE Konobar.idKonobar=Račun.konobar;

---Vraća ime i prezime konobara zajedno sa stolom, vremenom i ukupnom
cijenom računa koje je konobar izdao

SELECT Naziv,Cijena
FROM Artikl
WHERE Cijena = (
    SELECT MAX (Cijena)
    FROM Artikl);

---Vraća naziv i cijenu najskupljeg artikla.
```

5.3. Okidači

```
CREATE DEFINER = CURRENT_USER TRIGGER `Kafić`.`Račun_AFTER_INSERT` AFTER
INSERT ON `Račun` FOR EACH ROW

BEGIN

Insert into Povijest_transakcija
(Vrijeme,Konobar,Stol,Način_plaćanja,Ukupna_cijena)

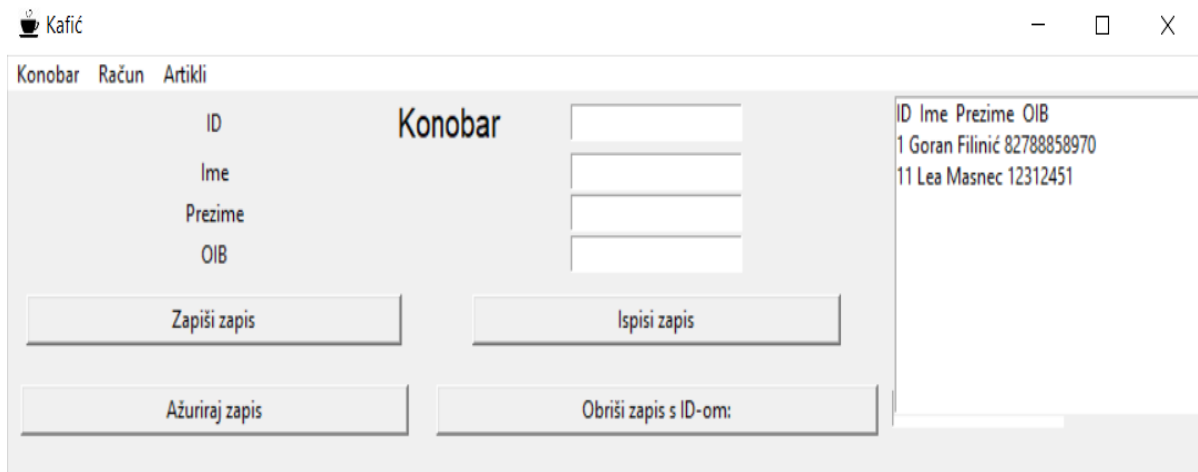
values
(new.Vrijeme,new.Konobar,new.Stol,new.Način_plaćanja,new.Ukupna_cijena);

delete from Račun;

END

---Briše sve zapise računa i sprema ih u tablicu povijest_transakcija
```

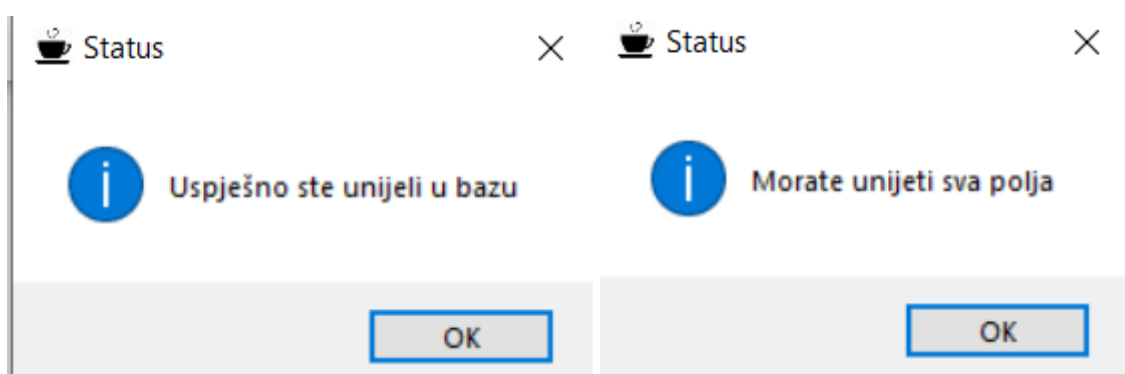
6. Aplikacijsko sučelje za rad s bazom podataka



ID	Ime	Prezime	OIB
1	Goran	Filinić	82788858970
11	Lea	Masnec	12312451

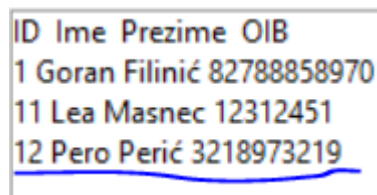
Slika 9. Forma Konobar aplikacije

Na formi Konobar nalazi se grupa labela za unos podataka. Ispod labela imamo osnovne CRUD operacije za upravljanjem sa entitetom Konobar. Na desnoj strani nam je objekt listbox koji se automatski refresha prilikom svakog unosa.



Slika 10. Prozor Statusa

Pritiskom na jednoj od CRUD operacija pojavljuje se statusni prozor koji nas obavještava ako je radnja uspješna odnosno neuspješna (Slika 10.)



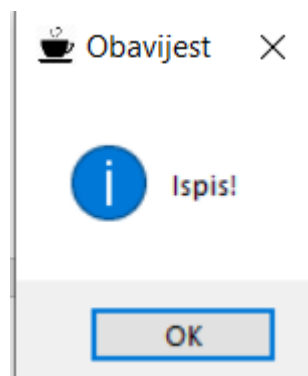
ID	Ime	Prezime	OIB
1	Goran	Filinić	82788858970
11	Lea	Masnec	12312451
12	Pero	Perić	3218973219

Slika 11. Listbox nakon izvršene operacije unosa

Iznad forme imamo mali meni navigacije koji nas vodi na forme Račun ili Artikli.

Slika 12. Forma Račun prije Ispisa

Na formi Račun imamo jednostavnu formu koja osim unosa prikazuje i trenutno vrijeme. Sa strane u ListBoxovima vidljivi su konobari iz Tablice Konobari bez OIB-a i Trenutne Stavke računa. Imamo dva gumba od kojih jedan ispisuje (u našem slučaju samo zapisuje račun) te gumb da dodamo još stavki računa. Nakon Ispisa zbog Triggera se Tablica račun prazni dok se njen zapis zapisuje u tablicu Povijest_transakcija kako bi se jasno vidjele transakcije. To služi kako korisnik ne može sam manipulirati transakcijama. Također nam se javlja statusni prozor nakon pritiska gumba (Slika 13.)



Slika 13. Statusni prozor nakon operacije Ispis

7. Zaključak

Kreiranje baze podataka temeljit je posao koji zahtjeva posebnu pažnju na detalje. Naravno s više iskustva stvari počinju biti znatno lakše, ali sama implementacije i oblikovanje bazom u nekom vanjskom jeziku zahtjeva dobro poznavanje i SQL-a i jezika i biblioteke jezika u kojemu radite. U mom slučaju Python je zahtijevao učenje **Tkinter** biblioteke i njena implementacije kako bih stvorio grafičko sučelje te samog poziva i implementacije naredbi SQL-a u sam Python što je trebalo puno dulje nego sam očekivao.

Za kraj smatram da je ovo jedan od najpraktičnijih projekta koji bilo kakav IT-ovac mora jednostavno proći kako bi koncipirao što zapravo ulazi u arhitekture aplikacija koje svakodnevno koristimo.

Popis literature

- [1] Tkinter; <https://wiki.python.org/moin/TkInter> dostupno 16.01.2020.
- [2] MySQL: <https://www.mysql.com> dostupno 16.01.2020.
- [3] Zašto MySQL: <https://www.mysql.com/why-mysql/> dostupno, 16.01.2020.
- [4] LucidChart, „What is an Entity Relationship Diagram”: <https://www.lucidchart.com/pages/er-diagrams> dostupno 16.01.2019.