Dizajniranje baze podataka za popis stanovništva

Forjan, Matej

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:194987

Rights / Prava: Attribution-ShareAlike 3.0 Unported/Imenovanje-Dijeli pod istim uvjetima 3.0

Download date / Datum preuzimanja: **2024-04-19**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Forjan

DIZAJNIRANJE BAZE PODATAKA ZA POPIS STANOVNIŠTVA

ZAVRŠNI RAD

Varaždin, 2022.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Forjan

JMBAG: 0016142401

Studij: Informacijski sustavi

DIZAJNIRANJE BAZE PODATAKA ZA POPIS STANOVNIŠTVA

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Miroslav Bača

Varaždin, veljača 2022.

Matej Forjan

Izjava o izvornosti

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

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Tema završnog rada je dizajniranje baze podataka za popis stanovništva. U prvom je dijelu rada objašnjen Državni zavod za statistiku te izrađen dijagram kompozicija, dijagram toka podataka na razini konteksta, dijagram toka podataka niže razine te ERA dijagram. Također, opisana je važnost i svrha provođenja popisa stanovništva. Nakon toga su pojašnjeni svi elementi koji su usko povezani s pojmom baza podataka te je dizajnirana baza podataka prema izrađenom ERA dijagramu. Za kraj je implementirano aplikacijsko sučelje koje koristi bazu podataka.

U radu su korišteni sljedeći alati: draw.io (za izradu dijagrama), MySQL Workbench za implementaciju baze podataka te Microsoft Visual Studio 2022 za izradu aplikacijskog sučelja.

Ključne riječi: Državni zavod za statistiku, popis stanovništva, dijagram dekompozicija, dijagram toka podataka, ERA dijagram, baza podataka, MySQL, Microsoft Visual studio 2022

Sadržaj

1. Uvod 1
2. Metode i tehnike rada 2
3. Državni zavod za statistiku 3
3.1. Misija 3
3.2. Vizija 3
4. Dijagram dekompozicija 4
4.1. Dijagram dekompozicija – Državni zavod za statistiku 5
5. Popis stanovništva 7
6. Dijagram toka podataka 8
6.1. Dijagram toka podataka – razina konteksta 11
6.2. Dijagram toka podataka – niža razina 11
7. Baza podataka 13
7.1. Sustav za upravljanje bazom podataka 13
7.2. Entitet 14
7.3. Atribut 15
7.4. Veze 15
7.5. Relacijski model podataka 16
7.5.1. Relacija 16
7.5.2. Primarni ključ 16
7.5.3. Vanjski ključ 17
7.5.4. Relacijska shema 17
8. Model baze podataka 18
8.1. ERA dijagram 18
8.2. Opis aplikacijske domene 19
8.3. Relacije u bazi podataka 19
8.4. Veze u bazi podataka 23
8.5. Relacijska shema 25

9. Izrada baze podataka 27
9.1. MySQL 27
9.2. MySQL Workbench 28
9.3. Kreiranje baze podataka 29
9.3.1. Kreiranje tablica 30
9.3.2. Unos podataka u tablice 32
9.4. Upiti 32
9.4.1. Jednostavni upiti 33
9.4.1.1. Upit 1 33
9.4.1.2. Upit 2 33
9.4.1.3. Upit 3 34
9.4.2. Složeni upiti 34
9.4.2.1. Upit 4 34
9.4.2.2. Upit 5 35
9.4.2.3. Upit 6 35
9.5. Okidači 36
9.5.1. Okidač 1 36
9.5.2. Okidač 2 37
9.5.3. Okidač 3 38
10. Izrada aplikacijskog sučelja baze podataka 40
10.1. Microsoft Visual Studio 2022 40
10.2. Početna forma 40
10.3. Forma – Popisivači 41
10.4. Forma – Popisne jedinice 42
11. Zaključak 44
Popis literature 45
Popis slika 47
Popis tablica 49

1. Uvod

Popis stanovništva je vrlo opsežan proces koji zahtjeva visoku organiziranost kako bi se u što većoj mjeri olakšalo njegovo provođenje. Iako se popis provodi svakih 10 godina, za njegovo provođenje pripreme počinju i nekoliko godina unaprijed. U tu je svrhu izrađen ovaj rad kako bi se cijeli proces popisa stanovništva pojednostavio što je više moguće.

U prvom će dijelu rada biti opisan Državni zavod za statistiku te definirana njegova misija i vizija. U nastavku će biti objašnjena svrha provođenja popisa stanovništva te postupak kojim se provodi.

Također, u radu će biti izrađeno nekoliko dijagrama. Prvi dijagram je dijagram kompozicija koji prikazuje organizacijske jedinice unutar Državnog zavoda za statistiku. Nakon toga slijedi dijagram toka podataka na razini konteksta te dijagram toka podataka niže razine za popis stanovništva. Posljednji dijagram koji će biti izrađen je ERA dijagram. Za svaki će se dijagram objasniti svi elementi od kojih se sastoje.

Nakon dijagrama implementirat će se baza podataka u alatu MySQL Workbench. Definirat će se postupak kojim se stvara baza podataka, relacije unutar nje, a zatim i popunjavanje podacima u pojedinim relacijama. Nakon toga slijede upiti koji omogućavaju lakši pregled unesenih podataka. Potom će biti izrađeni okidači koji onemogućavaju da se izvrše nedozvoljene radnje nad pojedinim podacima.

Na kraju rada implementirat će se aplikacijsko sučelje u alatu Microsoft Visual Studio 2022. Za izradu aplikacije koja će se povezati s izrađenom bazom podataka koristit će se programski jezik C#.

2. Metode i tehnike rada

Za izradu teorijskog dijela korišteni su razni izvori poput materijala s nastave, internetskih stranica te knjiga.

Kod izrade praktičnog dijela rada korišteno je nekoliko programskih alata. Za izradu dijagrama koristio se besplatan online alat draw.io. Ovaj alat omogućava izradu raznih dijagrama poput ERA dijagrama, dijagrama toka podataka, dijagrama slijeda, dijagrama komunikacije itd.. Osim toga, vrlo je jednostavan za korištenje zbog intuitivnog sučelja.

Baza podataka dizajnirana je u alatu MySQL Workbench. Ovaj alat služi za vizualno dizajniranje, modeliranje, generiranje i upravljanje MySQL bazama podataka. U njemu je kreirana baza podataka, uneseni podaci u relacije, izvršavani SQL upiti te okidači.

Za posljednji dio rada koji obuhvaća izradu aplikacijskog sučelja baze podataka koristio se alat Microsoft Visual Studio 2022. Ovaj alat dostupan je u nekoliko verzija koje se razlikuju prema mogućnostima koje nude te cijeni. U ovom je radu korišten Windows Forms alat koji služi za stvaranje desktop aplikacija te za povezivanje s MySQL bazama podataka. Programski jezik koji se koristio prilikom izrade aplikacije je C#.

3. Državni zavod za statistiku

Državni zavod za statistiku (kraće DZS) je prema strateškim i planskim dokumentima središnje tijelo službene statistike Republike Hrvatske te glavni predstavnik nacionalnog statističkog sustava pred europskim i međunarodnim tijelima nadležnim za statistiku. DZS redovita statistička istraživanja provodi na temelju Godišnjeg provedbenog plana statističkih aktivnosti Republike Hrvatske i Programa statističkih aktivnosti Republike Hrvatske [1].

3.1. Misija

Misija DZS-a je provedba statističkih podataka o ekonomskom i društvenom stanju s ciljem usmjeravanja politika povezanih s razvojem hrvatskog društva i gospodarstva te politika Europske unije [1].

3.2. Vizija

Vizija DZS-a je prisutnost u društvenim i ekonomskim pojavama, orijentiranost korisniku, učinkovitost pri rješavanju zadataka, upotreba suvremenih tehnologija i elektroničkih oblika komunikacije [1].

4. Dijagram dekompozicija

Stvarni sustavi, odnosno problemi su preveliki i presloženi da bi se rješavali u isto vrijeme. Osim toga, dijagrami i modeli su za takve sustave vrlo veliki što stvara probleme za korisnike. Kako bi se ti problemi sveli na minimum, sustavi se dijele na podsustave, podsustavi na funkcije itd.. Dijagrami dekompozicije koriste se kao vizualni prikaz te podjele [2].

Pojam dekompozicija može se definirati kao proces dijeljenja opisa sustava na manje komponente. Isto tako, dekompozicija predstavlja razlaganje složenih elemenata na jednostavnije. Dekompozicija se vrši tako što se složeni problem rastavlja na dva ili više potproblema, a zatim se svaki od tih potproblema rješava pojedinačno. Dijagram dekompozicije ima samo jedan simbol koji je pravokutnog oblika te predstavlja organizacijsku jedinicu, proces, problem, cilj ili lokaciju, što ovisi o vrsti dijagrama dekompozicije. Dijagram dekompozicije je u obliku stabla. Korijen stabla predstavlja početni problem od kojeg dekompozicija kreće. Korijen se dijeli na određene podsustave ili potprocese što ovisi o vrsti dijagrama dekompozicije. Svaki se element u dijagramu imenuje kako bi bilo jasno što taj element predstavlja. Listovima se nazivaju oni čvorovi u stablu koji nemaju svojih potomaka odnosno ne dijele se na druge podsustave ili potprocese [2].

Ovisno o tome što dekomponiramo, postoji više vrsta dekompozicijskih dijagrama, a to su:

- Procesa
- Organizacijskih jedinica
- Ciljeva
- Lokacija
- Problema

Dijagrami dekompozicije ne prikazuju usporednost vremenskog tijeka procesa, a to znači da iz ovih dijagrama nije vidljivo odvijaju li se ti procesi u isto vrijeme. Osim toga, redoslijed kojim su definirani procesi na dijagramu ne označava njihov vremenski redoslijed odvijanja. Iz tog je razloga cilj procese poredati u skladu s fazama životnog ciklusa ukoliko je to moguće. To bi značilo da se na jednoj fazi navode procesi planiranja, zatim upravljački, izvršni, obračunski i kontrolni procesi. Važno je da procesi djeca zajedno potpuno opisuju sve aktivnosti procesa roditelja (nadređenog procesa). To znači da proces koji je podijeljen na četiri potprocesa mora i na dijagramu dekompozicije biti podijeljen na četiri procesa jer u tom slučaju nadređeni proces ne bi bio potpuno opisan [2].

Dijagram dekompozicije dijeli se na razine, a broj razina ovisi o veličini sustava koji se dekomponira. Korijen dijagrama naziva se nulta razina. Na prvoj se razini sustav dekomponira na osnovne elemente koji objedinjuju skup srodnih podelemenata. Na drugoj se razini i svakoj sljedećoj složeniji podelementi dijele na jednostavnije podelemente. Na najnižoj se razini nalaze podelementi koji se dalje ne dijele (listovi stabla) [2].

U dijagramu je potrebno numerirati svaki element. Korijen stabla numerira se rednim brojem 0. Elementi na koje se dijeli korijen numeriraju se rednim brojevima od 1 do n. Zatim se elementi dijele na podelemente te se numeriraju kao decimale nadređenog elementa [2].

Prilikom izrade dijagrama dekompozicije, potrebno je pratiti nekoliko koraka kako bi se dijagram mogao kvalitetno izraditi. Za početak je bitno definirati promatrani sustav za kojeg se izrađuje dijagram. Ovisno o promatranom sustavu, definiramo vrstu dijagrama koji se izrađuje (dekompozicijski dijagram organizacijskih jedinica, procesa, ciljeva, lokacija ili problema). Potom se identificiraju elementi prve razine te se raspoređuju po životnom ciklusu. Dijagram je izrađen kada je svaki element dekomponiran do tog trenutka kada se više ne može dekomponirati [2].

Također, postoje i određena pravila kojih se potrebno pridržavati tijekom izrade dijagrama dekompozicije. Prvo pravilo određuje da sustav treba dekomponirati dok svaki od podsustava ne postane dovoljno definiran da ga nije moguće različito interpretirati. Drugo pravilo definira da svaki od navedenih podsustava treba tvoriti relativno nezavisnu i funkcijski zaokruženu cjelinu. Pri tome treba paziti da je unutarnja povezanost što jača. Povezanost je slaba ako podsustavi komuniciraju samo na razini podataka. Posljednje pravilo odnosi se na ograničenje papira, a to je najčešće format A4 [2].

Prednosti dijagrama dekompozicija:

- Korisni su za veće sustave jer daju pregled sustava
- Prikazuju veličinu sustava i broj elemenata
- Uključivanje korisnika u kreiranje sustava

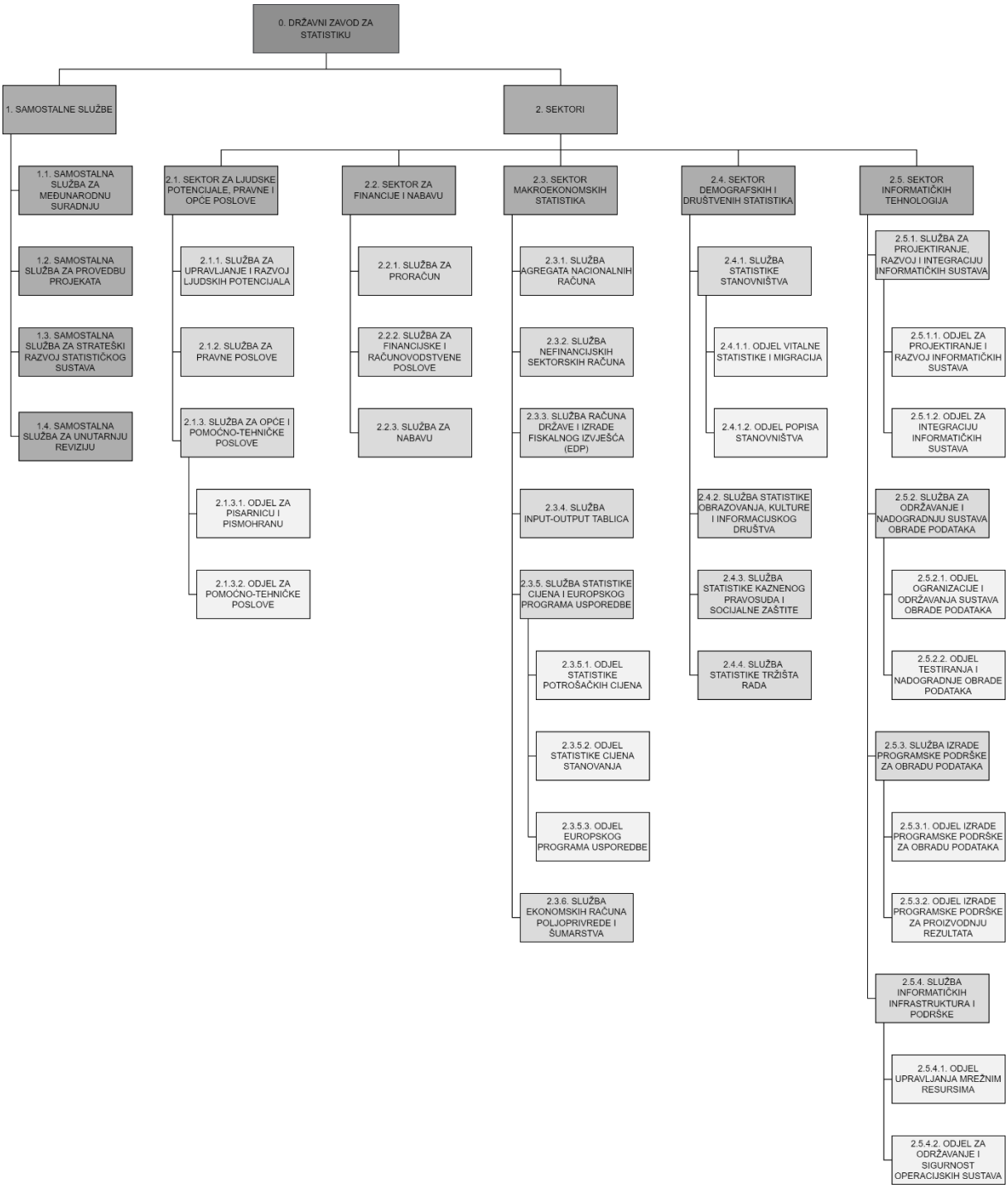
Nedostatak dijagrama dekompozicija:

- Dijagram ne prikazuje vremenski slijed procesa

4.1. Dijagram dekompozicija – Državni zavod za statistiku

U nastavku je prikazan dekompozicijski dijagram organizacijskih jedinica Državnog zavoda za statistiku. Svi elementi su organizacijske jedinice. Korijen dijagrama je Državni

zavod za statistiku numeriran rednim brojem 0. Korijen predstavlja nultu razinu. Korijen se dijeli na samostalne službe i sektore što je prva razina dijagrama. Zatim se sektori dijele na odjele što predstavlja drugu razinu dijagrama. Dijagram je izrađen kao kombinacija vertikalnog i horizontalnog oblika. Osim navedenih sektora, u DZS-u postoji još nekoliko sektora, ali zbog preglednosti i ograničenosti prostora na stranici, oni nisu prikazani.



Slika 1: Dekompozicijski dijagram Državnog zavoda za statistiku [autorski rad]

5. Popis stanovništva

Popis stanovništva, kućanstva i stanova u Republici Hrvatskoj je najopsežnije i najsloženije statističko istraživanje koje se posljednji put u Republici Hrvatskoj provelo 2021. godine. Popis se provodi svakih 10 godina, a u Republici Hrvatskoj se izvodio tri puta (2001., 2011. te 2021. godine) [3].

Svrha Popisa je ustanoviti broj i prostorni raspored stanovništva. Osim toga, utvrđuju se i ekonomska, demografska, migracijska, obrazovna i druga obilježja stanovništva. Rezultati Popisa su vrlo važni kako bi se moglo analizirati trenutno stanje utvrđeno Popisom te usporediti trenutno stanje s prethodno provedenim popisima. Pomoću tih podataka moguće je prognozirati i buduća stanja ispitanih podataka. Rezultati popisa dostupni su u raznim medijima kako bi svaki stanovnik mogao imati uvid u njih [3].

Popis 2021. provodio se na dva načina:

- Koristeći popisni upitnik u elektroničkom obliku koji je bio dostupan na sustavu e-Građani
- Popisivači su pomoću elektroničkih uređaja popisivali sve popisne jedinice koje se nisu samostalno popisale te su provjeravali podatke prikupljene samostalnim popisivanjem [3]

6. Dijagram toka podataka

Dijagram toka podataka (eng. *Data Flow Diagram*) prikazuje tokove podataka i procesa u sustavu. Pomoću dijagrama toka podataka (kraće DTP) korisnik i analitičar mogu precizno definirati zahtjeve korisnika. DTP je grafički prikaz sustava te se on pomoću ovog dijagrama može vrlo lako i brzo shvatiti. Osim za postojeći sustav, DTP se može koristiti i za modeliranje novog sustava. Elementi DTP-a su procesi, ulazni/izlazni tokovi podataka, spremišta podataka i vanjski objekti. Uspjeh informacijskog sustava ovisi o razumljivosti DTP-ova. Kroz povijest razvoja DTP-a pojavljuje se nekoliko stručnjaka koji su doprinijeli njegovom razvoju poput Yourdona, Toma DeMarca, Chrisa Ganea i Trisha Sarsona. Postoje tri vrste notacija koje su dobile ime prema ranije navedenim stručnjacima, a to su Yourdon notacija, DeMarco notacija te Gane i Sarson notacija. Prilikom kreiranja dijagrama potrebno je koristiti jednu odabranu notaciju [4].

Različite razine DTP-a opisuju područje analize, a prema tome ovaj se dijagram dijeli na dvije vrste:

- DTP na kontekstnoj razini
- DTP na nižoj razini

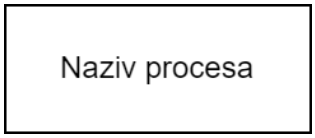
DTP na razini konteksta sadrži samo jedan proces, ulaze u taj proces, izlaze iz tog procesa te vanjske objekte. Svrha dijagrama kontekstne razine je prikazivanje granice između sustava koji se promatra i okoline tog sustava. Ovaj dijagram je nulte razine te definira područje dalje analize. Tokovi podataka otkrivaju veze sustava s okolinom [5].

Na DTP-u niže razine, proces koji se nalazi na dijagramu kontekstne razine, dijeli se na potprocese. Preporuka je da se na jednom dijagramu nalazi najmanje dva, a najviše devet procesa. Na DTP-u niže razine postoje različite vrste procesa koji se razvrstavaju prema podređenosti (potprocesi, nadređeni procesi) i položaju u modelu procesa (procesi dijagrama konteksta, primitivni procesi, procesi srednje razine, temeljni procesi) [5].

Elementi dijagrama toka podataka:

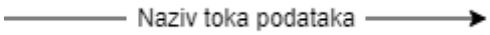
- Proces
- Tok podataka
- Vanjski objekt
- Spremište podataka

Cilj procesa je transformacija ulaznih tokova podataka u izlazne. Procesi koji se crtaju na dijagramu u stvarnosti se mogu izvršavati usporedno. Svakom se procesu dodjeljuje naziv (glagol u infinitivu) koji predstavlja kratak opis značenja procesa. Postoje određena pravila koja su vezana uz procese. Prvo pravilo glasi kako proces na DTP-u može biti povezan samo s tokom podataka. Osim toga, svaki proces mora sadržavati barem jedan ili više ulaznih tokova podataka te barem jedan ili više izlaznih tokova podataka. Prilikom transformiranja ulaza u izlaz, proces može pisati ili čitati podatke iz spremišta podataka. Na slici 2 prikazan je simbol procesa [5].



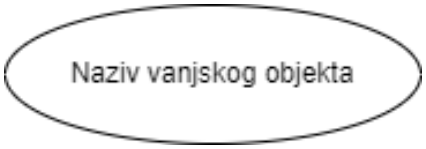
Slika 2: Simbol procesa [autorski rad]

Ulazni i izlazni tokovi podataka su oni tokovi koje sustav dobiva ili daje u okruženje. Tokovi prikazuju smjer kojim se podaci kreću između procesa, spremišta podataka i vanjskih entiteta. Tokovi podataka dijele se u tri skupine, a to su vanjski ulazi, vanjski izlazi i interni tokovi. Vanjski ulazi su oni tokovi podataka koji idu iz izvorišta prema procesima. Vanjski izlazi su oni tokovi podataka koji idu iz sustava prema odredištima. Interni tokovi su svi ostali tokovi podataka u sustavu. Primjer toka podataka može biti dokument unutar sustava, elementarni podatak i slično. Naziv toka podataka je imenica ili kombinacija imenice i pridjeva. Tokovi se grafički prikazuju ravnim crtama, krivim crtama ili lukovima sa ili bez strelice na kraju linije. Strelica predstavlja smjer kojim se kreću podaci. Na slici 3 prikazan je simbol toka podataka [5].



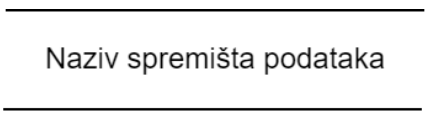
Slika 3: Simbol toka podataka [autorski rad]

Vanjski objekti mogu biti izvorišta ili odredišta informacija koje ulaze ili izlaze iz sustava. Zadatak vanjskih objekata je slanje tokova podataka prema sustavu ili primanje tokova podataka od sustava. Primjer vanjskog objekta može biti osoba, ustanova i slično. Naziv vanjskog objekta je imenica, tj. naziv sustava koji vanjski objekt predstavlja. Na slici 4 prikazan je simbol vanjskog objekta [5].



Slika 4: Simbol vanjskog objekta [autorski rad]

Spremišta podataka predstavljaju memorijska mjesta u kojima se pohranjuju podaci potrebni za izvršenje procesa ili podaci koji su dobiveni kao rezultat rada procesa. Spremišta podataka predstavljaju jedan ili više tokova u mirovanju. Nekoliko je pravila koja su vezana za spremišta podataka. Prvo pravilo glasi da jedan proces koristi podatke iz spremišta i pohranjuje podatke u spremište. Nadalje, svako spremište mora biti punjeno nekim procesom na nekom dijagramu u cjelokupnom modelu. Posljednje pravilo glasi kako nije moguće koristiti podatke iz spremišta koji prije toga nisu bili pohranjeni u njemu. Nad spremištem podataka mogu se izvršavati dvije operacije, a to je pisanje u spremište te čitanje iz spremišta pri čemu pisanje prethodi čitanju. Primjer spremišta podataka može biti dokument, baza podataka i slično. Na slici 5 prikazan je simbol spremišta podataka [5].



Slika 5: Simbol spremišta podataka [autorski rad]

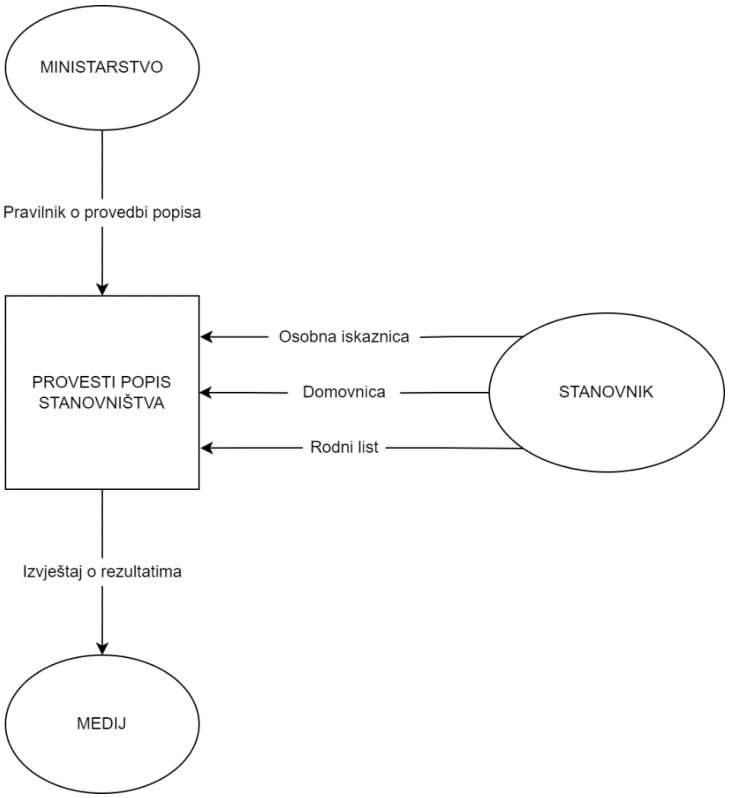
Načelo očuvanja vanjskih tokova definira kako količina tokova podataka koji ulaze i izlaze iz procesa treba biti jednaka na svim razinama dekompozicije. Primjerice, ako se za neki proces definira da na kontekstnoj razini ima četiri ulaza i tri izlaza, bitno je da taj isti proces na nižoj razini ima isti broj ulaza i izlaza [5].

Prilikom izrade dijagrama postoje određena ograničenja:

- Ne smiju se povezivati dva toka
- Ne smiju se povezivati spremišta i vanjski objekti tokom podataka
- Ne smiju se povezivati dva spremišta tokom podataka
- Ne smiju se povezivati procesi i vanjski objekti tokom podataka koji sadrži više imena
- Ne smije postojati proces samo s ulaznim tokom podataka
- Ne smije postojati proces samo s izlaznim tokom podataka
- Ne smije postojati spremište podataka samo s ulaznim tokom podataka
- Ne smije postojati spremište podataka samo s izlaznim tokom podataka
- Ne smije postojati proces, spremište podataka, vanjski objekt ili tok podataka koji nije povezan ni sa jednim elementom u sustavu

6.1. Dijagram toka podataka – razina konteksta

U nastavku je izrađen dijagram toka podataka kontekstne razine. Proces koji je odabran naziva se „Provesti popis stanovništva“. Nakon što je definiran proces, određuju se vanjski objekti, tj. s kim definirani proces komunicira. U ovom su dijagramu vanjski objekti „Ministarstvo“, „Stanovnik“ te „Medij“. Vanjski objekti „Ministarstvo“ i „Stanovnik“ predstavljaju izvorište jer šalju određene podatke prema procesu. Vanjski objekt „Stanovnik“ procesu šalje osobnu iskaznicu, domovnicu i rodni list. Vanjski objekt „Ministarstvo“ procesu šalje pravilnik o provedbi popisa. Vanjski objekt „Medij“ je odredište jer prima podatke iz procesa. Krajnji rezultat ovog procesa je izvještaj o rezultatima koji se šalje u medij. Na slici 6 prikazan je dijagram toka podataka na razini konteksta.

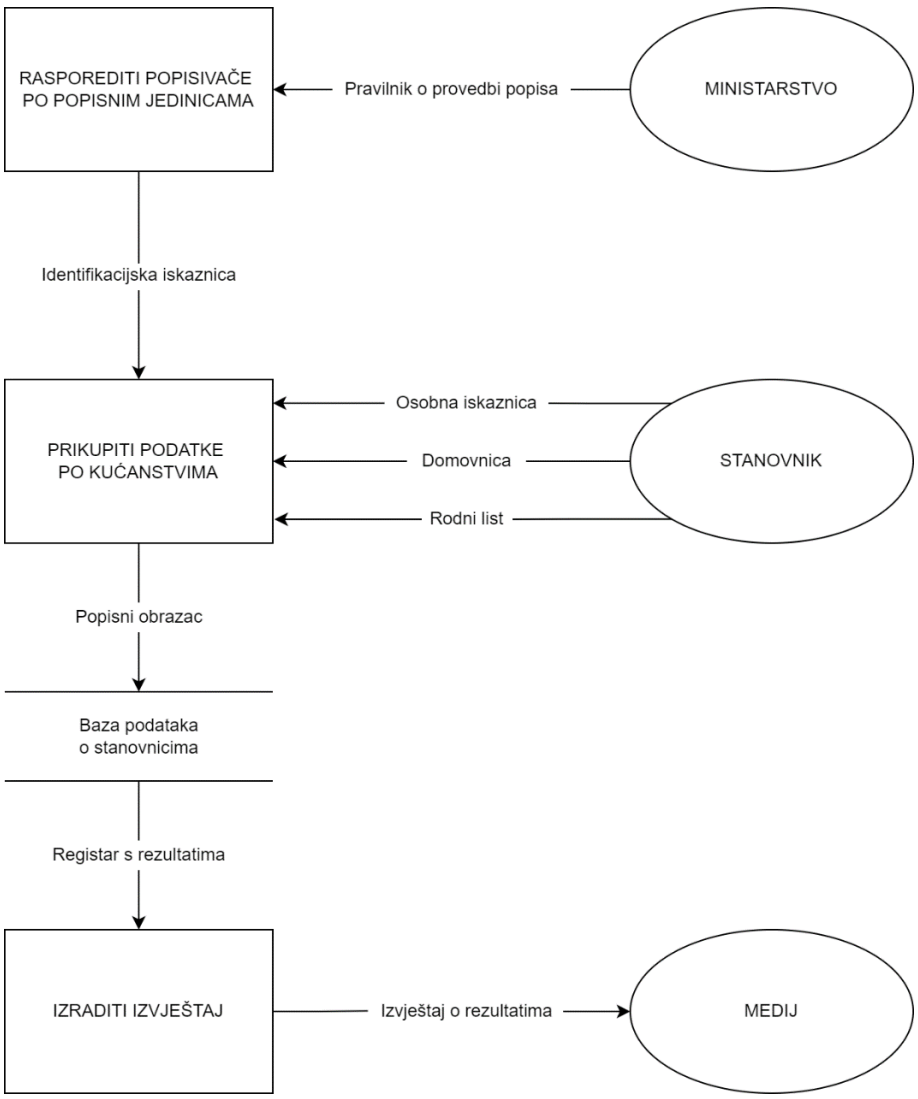


Slika 6: Dijagram toka podataka – razina konteksta [autorski rad]

6.2. Dijagram toka podataka – niža razina

U nastavku je izrađen dijagram toka podataka niže razine. Kod dijagrama toka podataka niže razine, vanjski su objekti ostali isti kao i u dijagramu toka podataka kontekstne razine. Proces u DTP-u kontekste razine ovdje se dijeli na potprocese, a to su „Rasporediti popisivače po popisnim jedinicama“, „Prikupiti podatke po kućanstvima“ te „Izraditi izvještaj“.

Osim potprocesa, ovdje se nalazi i spremište podataka koje se naziva „Baza podataka o stanovnicima“. Tokovi podataka koji su u DTP-u kontekste razine išli prema vanjskim objektima ili iz vanjskih objekata idu i u DTP-u niže razine prema načelu očuvanja vanjskih tokova. Izvorište toka podatka „Pravilnik o provedbi popisa“ jest vanjski objekt „Ministarstvo“, dok mu je odredište potproces „Rasporediti popisivače po popisnim jedinicama“. Nakon što se popisivači rasporede po popisnim jedinicama, pojedini popisivač dobiva identifikacijsku iskaznicu te može prikupljati podatke o stanovnicima. Tokovi podataka kojima je izvorište vanjski objekt „Stanovnik“ (osobna iskaznica, domovnica, rodni list) imaju odredište u potprocesu „Prikupiti podatke po kućanstvima“. Potom se ispunjava popisni obrazac te se sprema u spremište podataka čiji je naziv „Baza podataka o stanovnicima“. Potproces „Izraditi izvještaj“ čita spremljene podatke iz spremišta podataka te se izrađuje izvještaj o rezultatima. Kada je izvještaj gotov, šalje se u vanjski objekt „Medij“. Na slici 7 prikazan je dijagram toka podataka niže razine.



Slika 7: Dijagram toka podataka – niža razina [autorski rad]

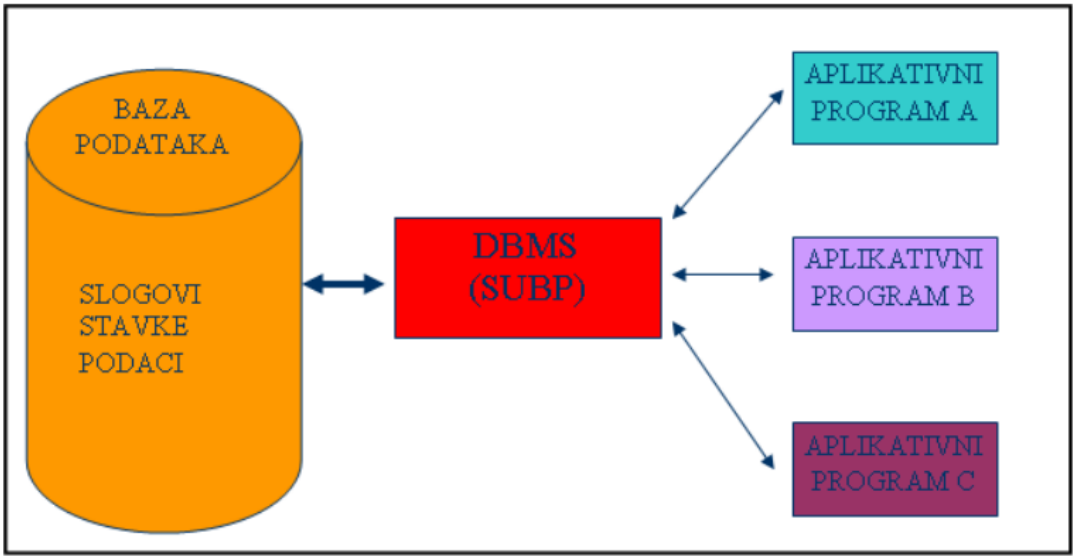
7. Baza podataka

U literaturi se mogu pronaći razne definicije koje opisuju pojam baze podataka. Konkretno, Kaluža [6, str. 5] bazu podataka definira kao organiziranu, povezanu, elektroničku biblioteku datoteka koja se može lako koristiti. Prema tome, podaci se u bazi podataka mogu jednostavno pregledavati, koristiti, sortirati, uspoređivati, pretraživati, dodavati, brisati te mijenjati. Također, redundancija odnosno zalihost je minimalna [6, str. 5].

Nadalje, Manger [7, str. 3] je bazu podataka definirao kao skup međusobno povezanih podataka koji su pohranjeni u vanjskoj memoriji računala. Podaci unutar baze podataka dostupni su raznim korisnicima te aplikacijama [7, str. 3].

7.1. Sustav za upravljanje bazom podataka

Sustav za upravljanje bazom podataka (eng. *Data Base Management System*) jest poslužitelj odnosno server baze podataka. Pomoću sustava za upravljanje bazom podataka (kraće SUBP) kreiraju se podaci te se kontrolira njihov pristup unutar baze podataka. Funkcije koje također obavlja SUBP su: obavljanje svih operacija s podacima u ime klijenta, automatiziranje administrativnih poslova s bazom, briga za sigurnost podataka te oblikovanje fizičkog prikaza baze u skladu s traženom logičkom strukturom. Na slici 8 prikazan je shematski pristup podacima upotrebom SUBP-a [7, str. 3].



Slika 8: Shematski prikaz pristupa podacima upotrebom SUBP-a [6, str. 18]

Prednosti SUBP-a:

- Povećava fleksibilnost sustava
- Smanjuje kompleksnost
- Smanjuje ovisnost programa o podacima
- Povećava dohvat i raspoloživost podataka
- Smanjuje zalihost i nekonzistentnost baze podataka
- Smanjuje troškove razvoja i održavanja baze podataka

Podaci se unutar baze podataka logički organiziraju u skladu s određenim modelom podataka. Model podataka može se definirati kao skup pravila koja određuju izgled logičke strukture baze. Model podataka predstavlja osnovu za projektiranje i implementiranje baze podataka. Osim toga, model prikazuje i odnose između pojedinih tipova podataka [7, str. 3].

SUBP podržava sljedeće modele podataka:

- Relacijski model podataka
- Mrežni model podataka
- Hijerarhijski model podataka
- Objektni model podataka

7.2. Entitet

Entitet predstavlja pojam o kojemu se u informacijskom sustavu prikupljaju i spremaju podaci. Entitet može biti biće (npr. student), objekt (npr. automobil), događaj (npr. promocija). Entiteti se mogu podijeliti u tri vrste s obzirom na ovisnost o drugim tipovima entiteta, a to su jaki, slabi i asocijativni entiteti. Jaki entiteti su neovisni o drugim entitetima te sadrže jedan ili više atributa koji razlikuju svaku pojavu entiteta (primarni ključevi). Slabi entiteti ovise o drugim tipovima entiteta. Asocijativni entiteti povezuju instance jednog ili više tipova entiteta. Instanca predstavlja primjer entiteta (npr. Državni i upravni informacijski sustavi je kolegij). Asocijativni se entiteti najčešće modeliraju kod veze više naprama više [8].

7.3. Atribut

Entiteti se pobliže opisuju pripadajućim atributima. Čubrilo [9, str. 30] je atribut definirao kao varijablu koja može poprimiti svaku od vrijednosti iz zadanog skupa. Primjer atributa može biti „Ime studenta“ koji pripada entitetu „Student“. Entitet se može sastojati od neograničenog broja atributa. Atributi mogu biti kompozitni, multivalued i derived atributi. Kompozitni atributi sadrže svoje specifične atribute. Multivalued atributi poprimaju više vrijednosti. Derived atributi su oni atributi čija se vrijednost izračunava iz povezanih atributa [8], [9, str. 30].

7.4. Veze

Veze se uspostavljaju između dva ili više tipova entiteta. Prema Mangeru [7, str. 9], riječ je o imenovanoj binarnoj ili k-narnoj relaciji između primjeraka entiteta zadanih tipova. Binarne veze su veze između dva tipa entiteta. Ternarna veza je ona veza koja u sebi sadrži tri različita tipa entiteta, a uvodi se kada vezu nije moguće rastaviti na binarne veze. Postoji iznimka kada je entitet u vezi sa samim sobom (unarna veza) [7, str. 9].

Brojnost veze predstavlja broj instanci entiteta iz relacije koji može biti povezan s relacijom, a ona može biti:

- Jedan naprama jedan (1:1)
    - Ova veza podrazumijeva da se jedan primjer tipa entiteta može povezati s najviše jednim primjerom drugog tipa entiteta
    - Primjer veze za entitete „Predsjednik“ i „Država“: jedna država može imati jednog predsjednika dok jedan predsjednik može biti u jednoj državi
- Jedan naprama više (1:N)
    - Ova veza podrazumijeva da se jedan primjer tipa entiteta može povezati s nula, jednim ili više primjera drugog tipa entiteta
    - Primjer veze za entitete „Stanovnik“ i „Županija“: jedan stanovnik može imati prebivalište u jednoj županiji dok prebivalište u jednoj županiji može imati više stanovnika
- Više naprama više (M:N)
    - Ova veza podrazumijeva da se više primjera prvog tipa entiteta može povezati s više primjera drugog tipa entiteta
    - Ovakva se veza ne može implementirati nego ju je potrebno razdvojiti u dvije 1:N veze i dodati asocijativni entitet između njih
    - Primjer veze za entitete „Proizvod“ i „Trgovina“: jedan proizvod može biti u više trgovina dok u jednoj trgovini može biti više proizvoda

7.5. Relacijski model podataka

Relacijski model podataka osnovan je krajem 60-tih godina 20. stoljeća, a osnivač je britanski znanstvenik Edgar Frank Codd. Efikasnost relacijskog modela podataka poboljšavala se razvojem računala. Većina sustava za upravljanje bazom podataka koristi relacijski model podataka. Podaci su, unutar relacijskog modela podataka, prikazani u relacijama (tablicama) [7, str. 13].

Osnova relacijskog modela podataka su:

- Relacija
- Ograničenja nad danom strukturom podataka
- Operacije nad strukturama

7.5.1. Relacija

Relacijski se model podataka zasniva na skupu dvodimenzionalnih tablica koje se nazivaju relacije. Relacije se unutar iste baze podataka međusobno razlikuju po nazivu. Svaka se relacija sastoji od naziva (naziv sheme relacije), stupaca (atributa), redova (slogova) te referenci (veze između pojedinih relacija). Slog označava jedan redak unutar tablice odnosno jedan primjerak entiteta. Unutar jedne relacije ne smiju postojati dva jednaka sloga (retka). Redak se još naziva i n-torka, a broj n-torki predstavlja kardinalnost relacije. Broj atributa definira stupanj relacije. Redoslijed atributa i slogova unutar relacije nije definiran [7, str. 13].

7.5.2. Primarni ključ

Primarni ključ je atribut ili skup atributa koji jednoznačno označavaju jednu instancu entiteta. Njihova se vrijednost razlikuje u svakom slogu u relaciji. Primarni atributi su oni atributi koji čine primarni ključ dok vrijednost primarnog ključa ne smije biti prazna niti u jednoj n-torki. Iz tog je razloga prilikom implementiranja baze podataka potrebno postaviti dodatno ograničenje za primarni ključ, a to je „NOT NULL“ [7, str. 14].

Svojstva primarnog ključa:

- Pojavnost – svaka instanca entiteta mora imati vrijednost primarnog ključa
- Jednoznačnost – vrijednost primarnog ključa za svaku instancu entiteta mora biti različita i jednoznačna
- Minimalnost – primarni ključ mora biti sastavljen od najmanjeg broja atributa koji mogu jednoznačno opisati svaku instancu entiteta [8]

7.5.3. Vanjski ključ

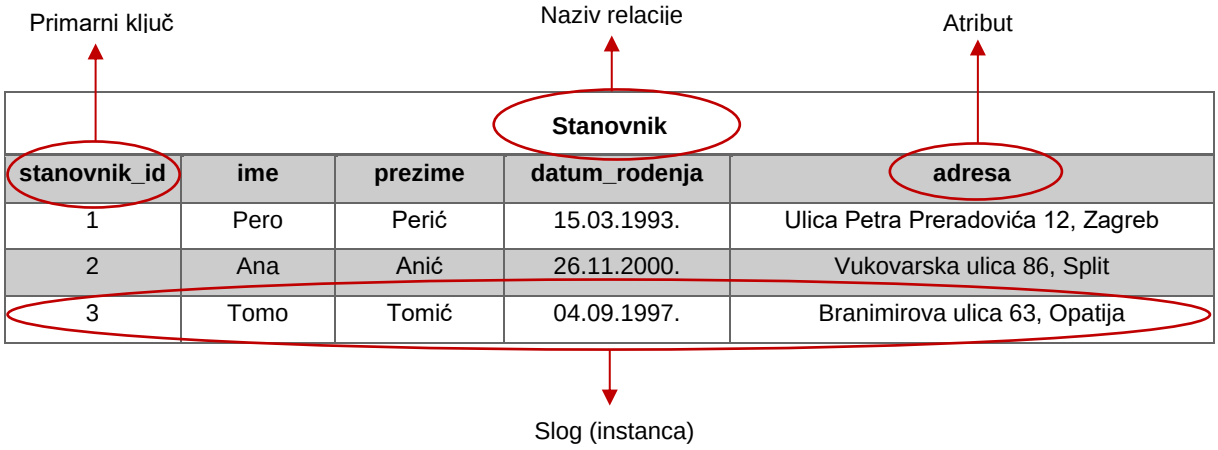
Vanjski ključevi su oni atributi koji služe za spajanje s drugim entitetom odabranom vezom. Kada se povezuju dva entiteta, vanjski ključ se uvijek nalazi u entitetu koji je na strani „više“. Osim toga, vanjski se ključ uvijek spaja s primarnim ključem drugog entiteta. Unutar jednog entiteta, jedan ili više atributa mogu poprimiti ograničenje vanjskog ključa [8].

7.5.4. Relacijska shema

Čubrilo [9, str. 30] relacijsku shemu definira kao skup atributa, ali se svaki skup atributa ne smatra relacijskom shemom. Relacijske se sheme označavaju velikim slovom R, dok se relacije označavaju malim slovom r. Shema se sadrži od imena relacije i naziva svih atributa koji se nalaze unutar pojedine relacije. Na relacijskoj shemi može biti definirano više relacija [9, str. 30].

Primjer relacijske sheme u kojoj se nalazi relacija „Stanovnik“:
STANOVNIK (stanovnik_id, ime, prezime, datum_rodenja, adresa, email, broj_telefona)

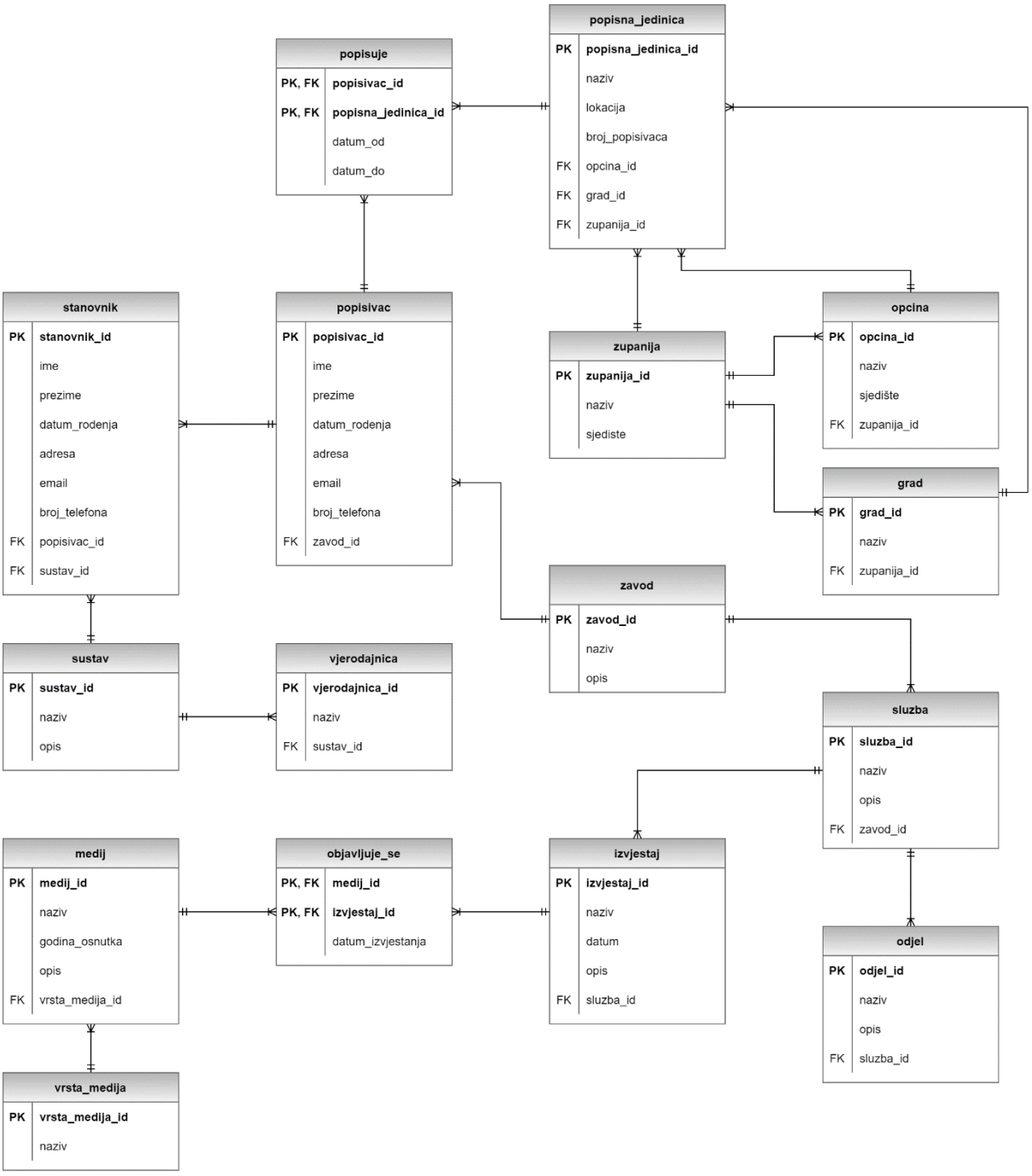
Tablica 1: Prikaz relacije „Stanovnik“ s pripadajućim dijelovima [autorski rad]



8. Model baze podataka

U svrhu izrade aplikacije za popis stanovništva, izrađena je baza podataka. U ovom će poglavlju biti prikazan ERA (eng. *Entity Relationship Attribute*) dijagram, opis aplikacijske domene, pripadajuće relacije unutar baze podataka, veze između relacija te relacijska shema.

8.1. ERA dijagram



Slika 9: ERA dijagram [autorski rad]

8.2. Opis aplikacijske domene

Baza podataka za popis stanovništva izradit će se prema prikazanom ERA dijagramu na slici 9. Kako bi se popis stanovništva odradio bez velikih poteškoća te sa što većom preciznošću, potrebno je izraditi aplikaciju koja se sastoji od svih važnih podataka koji su potrebni za obavljanje popisa. U ERA dijagramu, pravokutnicima su prikazane relacije s pripadajućim atributima. Relacije su međusobno povezane pripadajućim vezama.

Prvi korak koji je potrebno odraditi prije samog obavljanja popisa stanovništva je registracija popisivača koji će provoditi popis te definiranje popisnih jedinica po kojima će se provoditi popis od strane državnog zavoda za statistiku. Nakon što su se registrirali svi popisivači, oni se raspoređuju po popisnim jedinicama.

Svaki stanovnik ima mogućnost biranja hoće li se popisati samostalno putem sustava „e-Građani“ ili će ih popisati popisivač. Stanovnik se na sustav „e-Građani“ prijavljuje odabranom vjerodajnicom. Nakon što popisivač popiše sve stanovnike u popisnim jedinicama koje su mu dane, on svoje rezultate predaje u Državni zavod za statistiku koji zatim te rezultate prosljeđuje u pripadajuće službe i odjele. Služba statistike stanovništva obrađuje rezultate te izrađuje izvještaje prema dobivenim rezultatima. Izvještaji se zatim objavljuju u raznim medijima (tiskani mediji, elektronički mediji i sl.).

8.3. Relacije u bazi podataka

Baza podataka sastoji se od šesnaest relacija koje su međusobno povezane vezama. Svaka relacija ima svoj naziv te atribute. U nastavku će biti prikazane sve relacije s pripadajućim atributima, tipovima podataka i ograničenjima.

Tablica 2: Relacija „zavod“ [autorski rad]

zavod		
Atribut	**Tip podatka**	**Ograničenje**
zavod_id	int	primary key, not null
naziv	varchar(50)	not null
opis	varchar(150)	not null

Tablica 3: Relacija „sluzba“ [autorski rad]

sluzba		
Atribut	**Tip podatka**	**Ograničenje**
sluzba_id	int	primary key, not null
naziv	varchar(50)	not null
opis	varchar(150)	not null
zavod_id	int	foreign key, not null

Tablica 4: Relacija „odjel“ [autorski rad]

odjel		
Atribut	**Tip podatka**	**Ograničenje**
odjel_id	int	primary key, not null
naziv	varchar(50)	not null
opis	varchar(150)	not null
sluzba_id	int	foreign key, not null

Tablica 5: Relacija „popisivac“ [autorski rad]

popisivac		
Atribut	**Tip podatka**	**Ograničenje**
popisivac_id	int	primary key, not null
ime	varchar(30)	not null
prezime	varchar(30)	not null
datum_rodenja	date	not null
adresa	varchar(100)	not null
email	varchar(100)	not null
broj_telefona	varchar(15)	not null
zavod_id	int	foreign key, not null

Tablica 6: Relacija „popisuje“ [autorski rad]

popisuje		
Atribut	**Tip podatka**	**Ograničenje**
popisivac_id	int	primary key, foreign key, not null
popisna_jedinica_id	int	primary key, foreign key, not null
datum_od	date	not null
datum_do	date	not null

Tablica 7: Relacija „popisna_jedinica“ [autorski rad]

popisna_jedinica		
Atribut	**Tip podatka**	**Ograničenje**
popisna_jedinica_id	int	primary key, not null
naziv	varchar(50)	not null
lokacija	varchar(50)	not null
broj_popisivaca	int	not null
opcina_id	int	foreign key
grad_id	int	foreign key
zupanija_id	int	foreign key, not null

Tablica 8: Relacija „zupanija“ [autorski rad]

zupanija		
Atribut	**Tip podatka**	**Ograničenje**
zupanija_id	int	primary key, not null
naziv	varchar(50)	not null
sjediste	varchar(30)	not null

Tablica 9: Relacija „opcina“ [autorski rad]

opcina		
Atribut	**Tip podatka**	**Ograničenje**
opcina_id	int	primary key, not null
naziv	varchar(50)	not null
sjediste	varchar(30)	not null
zupanija_id	int	foreign key, not null

Tablica 10: Relacija „grad“ [autorski rad]

grad		
Atribut	**Tip podatka**	**Ograničenje**
grad_id	int	primary key, not null
naziv	varchar(50)	not null
zupanija_id	int	foreign key, not null

Tablica 11: Relacija „stanovnik“ [autorski rad]

stanovnik		
Atribut	**Tip podatka**	**Ograničenje**
stanovnik_id	int	primary key, not null
ime	varchar(30)	not null
prezime	varchar(30)	not null
datum_rodenja	date	not null
adresa	varchar(100)	not null
email	varchar(100)	not null
broj_telefona	varchar(15)	not null
popisivac_id	int	foreign key
sustav_id	int	foreign key

Tablica 12: Relacija „sustav“ [autorski rad]

sustav		
Atribut	**Tip podatka**	**Ograničenje**
sustav_id	int	primary key, not null
naziv	varchar(50)	not null
opis	varchar(150)	not null

Tablica 13: Relacija „vjerodajnica“ [autorski rad]

vjerodajnica		
Atribut	**Tip podatka**	**Ograničenje**
vjerodajnica_id	int	primary key, not null
naziv	varchar(50)	not null
sustav_id	int	foreign key, not null

Tablica 14: Relacija „izvjestaj“ [autorski rad]

izvjestaj		
Atribut	**Tip podatka**	**Ograničenje**
izvjestaj_id	int	primary key, not null
naziv	varchar(50)	not null
datum	date	not null
opis	varchar(150)	not null
sluzba_id	int	foreign key, not null

Tablica 15: Relacija „objavljuje_se“ [autorski rad]

objavljuje_se		
Atribut	**Tip podatka**	**Ograničenje**
medij_id	int	primary key, foreign key, not null
izvjestaj_id	int	primary key, foreign key, not null
datum_izvjestavanja	date	not null

Tablica 16: Relacija „medij“ [autorski rad]

medij		
Atribut	**Tip podatka**	**Ograničenje**
medij_id	int	primary key, not null
naziv	varchar(50)	not null
godina_osnutka	int	not null
opis	varchar(150)	not null
vrsta_medija_id	int	foreign key, not null

Tablica 17: Relacija „vrsta_medija“ [autorski rad]

vrsta_medija		
Atribut	**Tip podatka**	**Ograničenje**
vrsta_medija_id	int	primary key, not null
naziv	varchar(50)	not null

8.4. Veze u bazi podataka

Relacije su međusobno povezane zadanim vezama. U nastavku će biti navedene veze između pojedinih relacija.

Veza „zavod – sluzba“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se jedan zavod može sastojati od više službi dok jedna služba može biti dio jednog zavoda. Kako bi se takva veza implementirala, potrebno je u relaciji „sluzba“ kao vanjski ključ unijeti primarni ključ iz relacije „zavod“.

Veza „sluzba – odjel“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se jedna služba može sastojati od više odjela dok jedan odjel može biti dio jedne službe. Kako bi se takva veza implementirala, potrebno je u relaciji „odjel“ kao vanjski ključ unijeti primarni ključ iz relacije „sluzba“.

Veza „zavod – popisivac“ je veza jedan naprama više (1:N). Iz tog proizlazi kako jedan zavod može sadržavati više popisivača dok jedan popisivač može biti dio jednog zavoda. Kako bi se takva veza implementirala, potrebno je u relaciji „popisivac“ kao vanjski ključ unijeti primarni ključ iz relacije „zavod“.

Veza „popisivac – popisna_jedinica“ je veza više naprama više (M:N). Iz tog proizlazi kako jedan popisivač popisuje u više popisnih jedinica dok se jedna popisna jedinica sastoji od više popisivača. Kako bi se takva veza implementirala, potrebno je napraviti asocijativni entitet, a u ovom je slučaju to entitet „popisuje“. Relacija „popisuje“ sadrži kao vanjske ključeve primarne ključeve iz relacija „popisivac“ i „popisna_jedinica“.

Veza „popisna_jedinica – zupanija“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se jedna popisna jedinica nalazi u jednoj županiji dok se u jednoj županiji nalazi više popisnih jedinica. Kako bi se takva veza implementirala, potrebno je u relaciji „popisna_jedinica“ kao vanjski ključ unijeti primarni ključ iz relacije „zupanija“.

Veza „popisna_jedinica – grad“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se jedna popisna jedinica nalazi u jednom gradu dok se u jednom gradu nalazi više popisnih jedinica. Kako bi se takva veza implementirala, potrebno je u relaciji „popisna_jedinica“ kao vanjski ključ unijeti primarni ključ iz relacije „grad“.

Veza „popisna_jedinica – opcina“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se jedna popisna jedinica nalazi u jednoj općini dok se u jednoj općini nalazi više popisnih jedinica. Kako bi se takva veza implementirala, potrebno je u relaciji „popisna_jedinica“ kao vanjski ključ unijeti primarni ključ iz relacije „opcina“.

Veza „zupanija – grad“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se u jednoj županiji nalazi više gradova dok se jedan grad nalazi u jednoj županiji. Kako bi se takva veza implementirala, potrebno je u relaciji „grad“ kao vanjski ključ unijeti primarni ključ iz relacije „zupanija“.

Veza „zupanija – opcina“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se u jednoj županiji nalazi više općina dok se jedna općina nalazi u jednoj županiji. Kako bi se takva veza implementirala, potrebno je u relaciji „opcina“ kao vanjski ključ unijeti primarni ključ iz relacije „zupanija“.

Veza „popisivac – stanovnik“ je veza jedan naprama više (1:N). Iz tog proizlazi kako jedan popisivač može popisati više stanovnika dok jedan stanovnik može biti popisan od strane jednog popisivača. Kako bi se takva veza implementirala, potrebno je u relaciji „stanovnik“ kao vanjski ključ unijeti primarni ključ iz relacije „popisivac“.

Veza „stanovnik – sustav“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se jedan stanovnik može prijaviti u jedan sustav dok se na jedan sustav može prijaviti više stanovnika. Kako bi se takva veza implementirala, potrebno je u relaciji „stanovnik“ kao vanjski ključ unijeti primarni ključ iz relacije „sustav“.

Veza „sustav – vjerodajnica“ je veza jedan naprama više (1:N). Iz tog proizlazi kako se na sustav može prijaviti preko više vjerodajnica dok se s jednom vjerodajnicom može prijaviti na jedan sustav. Kako bi se takva veza implementirala, potrebno je u relaciji „vjerodajnica“ kao vanjski ključ unijeti primarni ključ iz relacije „sustav“.

Veza „sluzba – izvjestaj“ je veza jedan naprama više (1:N). Iz tog proizlazi kako jedna služba može izraditi više izvještaja dok jedan izvještaj može biti izrađen od strane jedne službe. Kako bi se takva veza implementirala, potrebno je u relaciji „izvjestaj“ kao vanjski ključ unijeti primarni ključ iz relacije „sluzba“.

Veza „izvjestaj – medij“ je veza više naprama više (M:N). Iz tog proizlazi kako se jedan izvještaj može objaviti u više medija dok se u jednom mediju može objaviti više izvještaja. Kako bi se takva veza implementirala, potrebno je napraviti asocijativni entitet, a u ovom je slučaju to entitet „objavljuje_se“. Relacija „objavljuje_se“ sadrži kao vanjske ključeve primarne ključeve iz relacija „izvjestaj“ i „medij“.

Veza „medij – vrsta_medija“ je veza jedan naprama više (1:N). Iz tog proizlazi kako jedan medij može biti jedne vrste dok jedne vrste medija može biti više medija. Kako bi se takva veza implementirala, potrebno je u relaciji „medij“ kao vanjski ključ unijeti primarni ključ iz relacije „vrsta_medija“.

8.5. Relacijska shema

Relacijska shema izrađena je na temelju ERA dijagrama. Primarni ključevi u relacijskoj shemi prikazani su podcrtano, a vanjski ključevi su **podebljani** i označeni **crvenom** bojom. Dvokomponentni ključevi prikazani su **podebljano**, podcrtano te su označeni **crvenom** bojom. Tablica 18 prikazuje relacijsku shemu za bazu popisa stanovništva.

Tablica 18: Relacijska shema [autorski rad]

RELACIJE	ATRIBUTI
zavod	(zavod_id, naziv, opis)
sluzba	(sluzba_id, naziv, opis, **zavod_id**)
odjel	(odjel_id, naziv, opis, **sluzba_id**)
popisivac	(popisivac_id, ime, prezime, datum_rodenja, adresa, email, broj_telefona, **zavod_id**)
popisuje	(**popisivac_id**, **popisna_jedinica_id**, datum_od, datum_do)
popisna_jedinica	(popisna_jedinica_id, naziv, lokacija, broj_popisivaca, **opcina_id**, **grad_id**, **zupanija_id**)
zupanija	(zupanija_id, naziv, sjediste)
opcina	(opcina_id, naziv, sjediste, **zupanija_id**)
grad	(grad_id, naziv, **zupanija_id**)
stanovnik	(stanovnik_id, ime, prezime, datum_rodenja, adresa, email, broj_telefona, **popisivac_id**, **sustav_id**)
sustav	(sustav_id, naziv, opis)
vjerodajnica	(vjerodajnica_id, naziv, **sustav_id**)
izvjestaj	(izvjestaj_id, naziv, datum, opis, **sluzba_id**)
objavljuje_se	(**medij_id**, **izvjestaj_id**, datum_izvjestavanja)
medij	(medij_id, naziv, godina_osnutka, opis, **vrsta_medija_id**)
vrsta_medija	(vrsta_medija_id, naziv)

9. Izrada baze podataka

U ovom će poglavlju biti opisan standardizirani jezik za pristup bazama podataka SQL (eng. *Structured Query Language*). Osim toga, definirat će se sustav za upravljanje relacijskim bazama podataka u kojemu je razvijena baza podataka za provedbu popisa stanovništva. Zatim će se obraditi program u kojem je izrađena baza podataka te opisati postupak kreiranja baze podataka (tablice, upiti, okidači).

9.1. MySQL

MySQL je sustav za upravljanje relacijskim bazama podataka. Što se povijesnog dijela tiče, MySQL je 1995. godine razvila švedska tvrtka MySQL AB čiji su osnivači Michael Widenius, David Axmark te Allan Larsson. 2000. godine je MySQL postao softver otvorenog koda. MySQL je napisan u C i C++ programskom jeziku te je dostupan za preuzimanje na više operacijskih sustava kao što je Windows, Linux, Mac OS. Osim toga, dostupan je i u dvije verzije koje se odabiru priliku preuzimanja: MySQL Community Server i MySQL Enterprise Server. Community Server je besplatan za preuzimanje, ali je potreban veći period vremena u kojem bi se riješili mogući problemi kada do njih dođe. Enterprise Server nije besplatan, ali je znatno bolja podrška ORACLE-a te se vrlo brzo rješavaju mogući problemi. Na slici 10 prikazan je logo MySQL-a [10], [11], [12].

Slika 10: Logo MySQL-a [13]

SQL služi za unos, pristup te obradu podataka pohranjenih u bazi. Naredbe se u SQL-u mogu podijeliti u 4 kategorije, a to su:

- DDL (eng. *Data Definition Language*)
    - Upravlja shemama baze podataka te strukturom podataka unutar baze
    - Naredbe: CREATE, ALTER, DROP, TRUNCATE, COMMENT, RENAME
- DML (eng. *Data Manipulation Language*)
    - Koristi se za unos, pristup, promjenu i brisanje podataka unutar baze
    - Naredbe: SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, EXPLAIN PLAN, LOCK TABLE
- DCL (eng. *Data Control Language*)
    - Odnosi se na prava, dozvole i druge kontrole sustava baze podataka
    - Naredbe: GRANT, REVOKE
- TCL (eng. *Transaction Control Language*)
    - Upravlja transakcijama unutar baze podataka
    - Naredbe: COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION [14]

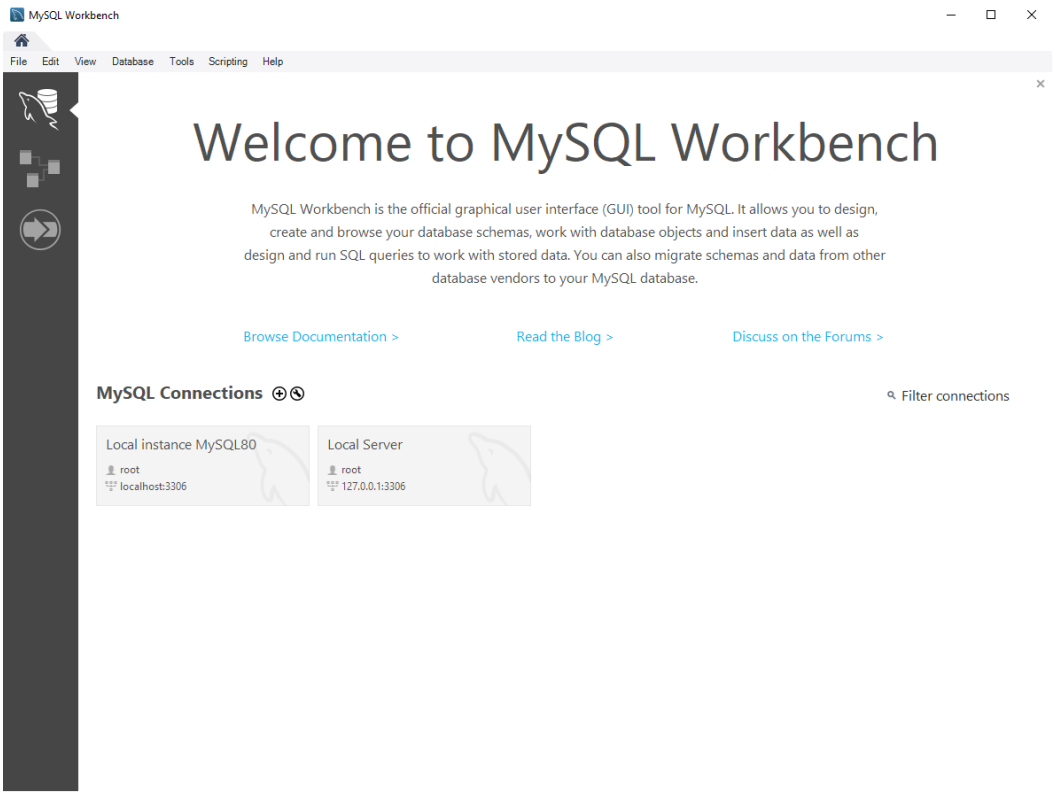
9.2. MySQL Workbench

MySQL Workbench je alat za vizualno dizajniranje, modeliranje, generiranje i upravljanje MySQL baza podataka. Tvrtka Oracle je prvu verziju ovog alata plasirala na tržište 2005. godine. MySQL Workbench je nasljednik alata DBDesigner4 kojeg je razvio austrijski programer Michael G. Zinner. Ovaj alat pomoću SQL editora omogućava kreiranje baze podataka, izvršavanje SQL upita, okidača i još mnogo toga. Također, MySQL Workbench sadrži alate za korisničku administraciju, sigurnosno kopiranje (eng. *backup*) i tako dalje [15], [16].

Dostupan je za preuzimanje na više operacijskih sustava, a to su Windows, Linux te Mac OS. Osim toga, može se preuzeti u dvije verzije, a to su MySQL Workbench Community Edition te Enterprise Edition. Community Edition je besplatna verzija za razliku od Enterprise Edition verzije koja nije, ali Enterprise Edition verzija sadrži neke mogućnosti koje besplatna verzija nema poput validacije modela i sheme, automatske dokumentacije i slično [15].

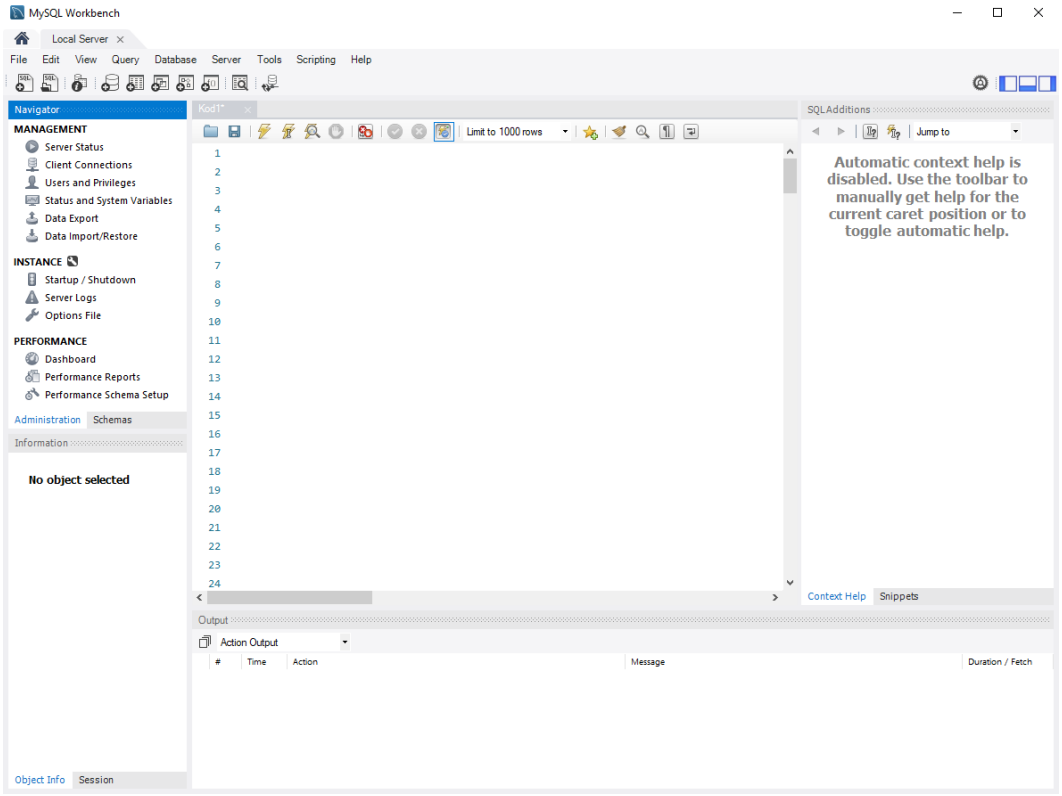
9.3. Kreiranje baze podataka

Slika 11 prikazuje početno sučelje alata MySQL Workbench. Kako bi se uspostavila konekcija, potrebno je odabrati ponuđenu ili izraditi novu konekciju. U ovom je radu izrađena nova konekcija „Local Server“ te je u njoj izrađena baza podataka. S lijeve strane sučelja su vidljiva tri modula koja je moguće odabrati. Prvi modul omogućava kreiranje i upravljanje bazama podataka, drugi modul omogućava modeliranje podataka prema vlastitim potrebama dok posljednji modul omogućava migraciju baza podataka.



Slika 11: Početno sučelje alata MySQL Workbench [autorski rad]

Slika 12 prikazuje sučelje alata nakon što se odabrala željena konekcija. Ovdje su prikazane brojne mogućnosti koje pruža ovaj alat poput kreiranja sheme, kreiranje tablice, kreiranje funkcije, kreiranje pogleda i još mnogo toga. Automatski se otvara prozor za postavljanje upita, prozor za ispis rezultata, prozor s informacijama o odabranom objektu te prozor za administraciju.

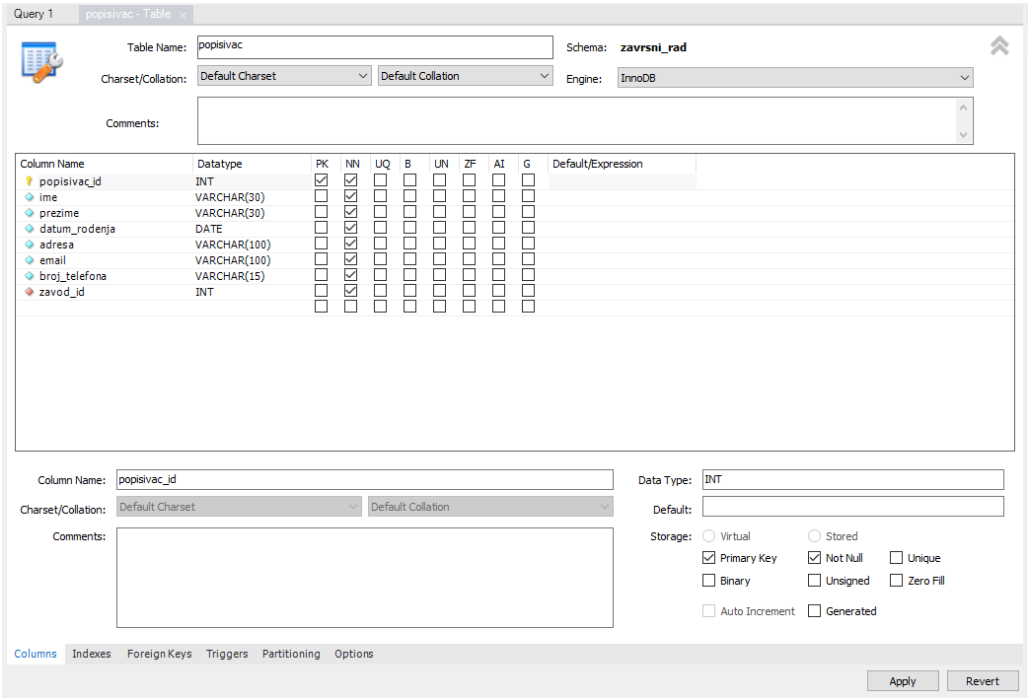


Slika 12: Sučelje alata MySQL Workbench nakon odabrane konekcije [autorski rad]

Kako bi se kreirala nova baza podataka potrebno je koristiti naredbu CREATE DATABASE. Naredba se piše u prozoru za postavljanje upita sljedećom sintaksom „CREATE DATABASE *naziv_baze_podataka*“. Baza je nazvana „zavrsni_rad“. Kako bi se uvjerilo da se baza zaista stvorila, koristi se naredba SHOW DATABASES čijim će se izvršenjem ispisati sve kreirane baze podataka. Za korištenje željene baze, potrebno je koristiti naredbu USE. Sintaksa za naredbu USE je sljedeća: „USE *naziv_baze_podataka*“. Naredbom SHOW TABLES ispisuju se sve tablice unutar korištene baze, ali se u bazi podataka trenutno ne nalazi niti jedna tablica već ih je potrebno u daljnjim koracima kreirati.

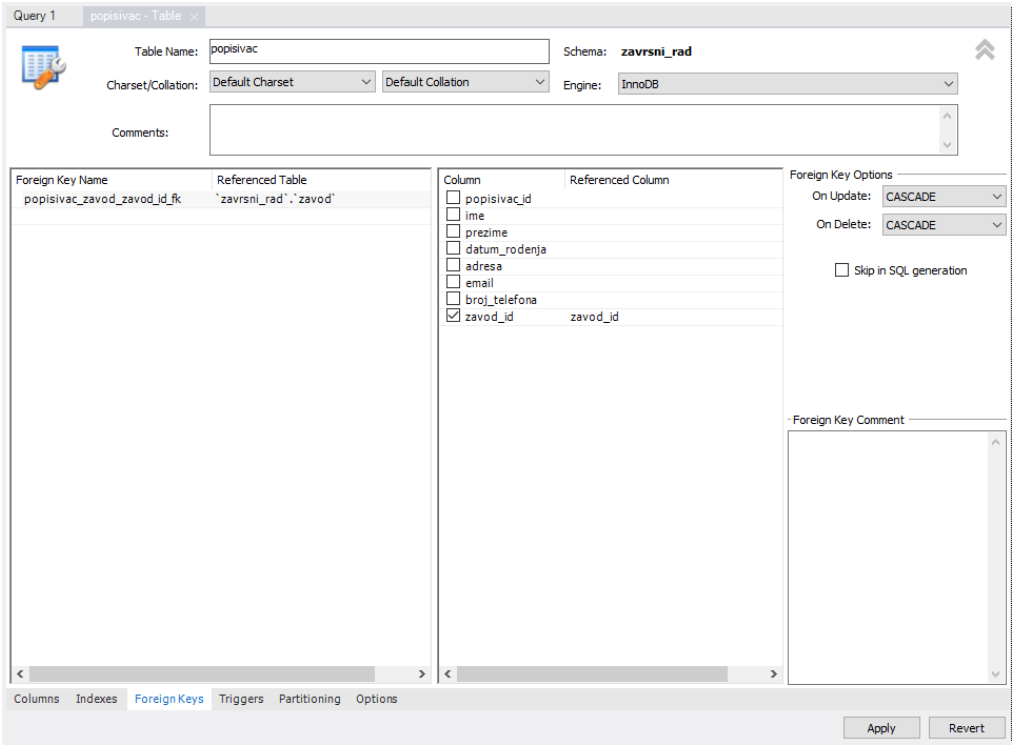
9.3.1. Kreiranje tablica

Tablice se kreiraju na način da se u navigatoru s lijeve strane odabere željena baza te se unutar nje odabere opcija „Create table“. Nakon toga se otvara novi prozor u kojem se unosi naziv tablice, nazivi atributa s pripadajućim tipovima podataka (int, float, varchar, date itd.) i ograničenjima (PK (Primary key), NN (Not null), UQ (Unique) itd.). Osim toga, ovdje se definiraju i vanjski ključevi, indeksi, okidači. Nakon što su odabrane sve željene opcije, tablica se kreira pritiskom na gumb „Apply“. Postoji i drugi način kreiranja tablica, a to je pisanjem SQL koda u prozoru za postavljanje upita. U ovom je radu korišten prvi način kreiranja tablica. Na slici 13 prikazan je prozor u kojem se kreira tablica.



Slika 13: Prozor za kreiranje tablica [autorski rad]

Ukoliko se u tablici nalazi jedan ili više vanjskih ključeva, za njihovo kreiranje koristi se opcija „Foreign Keys“. Prilikom kreiranja vanjskog ključa potrebno je navesti naziv vanjskog ključa i tablicu na koju se vanjski ključ referencira. Odabirom tablice, s desne se strane prikazuju atributi tablice koju kreiramo te atributi iz tablice na koju se želimo referencirati. Potrebno je odabrati iz svake tablice po jedan atribut koji se žele povezati. Posljednje što je potrebno navesti prilikom kreiranja vanjskog ključa je referencijalni integritet. Referencijalni integritet definira što će se dogoditi s vanjskim ključem ukoliko se briše ili mijenja primarni ključ na kojeg se vanjski ključ referencira. Ključne riječi prilikom postavljanja integriteta su ON UPDATE i ON DELETE nakon kojih dolaze 4 mogućnosti, a to su: RESTRICT, CASCADE, SET NULL i NO ACTION. RESTRICT označava da je zabranjena promjena nad referenciranim retkom. CASCADE označava da se promjena događa i nad svim redcima u drugoj tablici koji se referenciraju na redak koji se mijenja. SET NULL označava da se vrijednost vanjskog ključa u referencirajućim redcima postavlja na NULL. NO ACTION označava da je promjena zabranjena ako u tom trenutku postoji redak u drugoj tablici koji se referencira na redak koji se mijenja. Na slici 14 je prikazan prozor u kojem se postavlja vanjski ključ.



Slika 14: Prozor za postavljanje vanjskog ključa [autorski rad]

9.3.2. Unos podataka u tablice

Glavna svrha baze podataka je spremanje podataka, odnosno njihovo dohvaćanje. Iz tog je razloga vrlo važno unijeti podatke u pripadajuće tablice. Jedan će dio podataka biti unesen u alatu MySQL Workbench naredbom INSERT dok će ostatak podataka biti unesen u aplikaciji koja će biti izrađena u nastavku. U nastavku će biti prikazan primjer unosa jedne instance u tablicu „zupanija“ naredbom INSERT.

```
INSERT INTO zupanija(zupanija_id, naziv, sjediste)
VALUES (7, 'Bjelovarsko-bilogorska županija', 'Bjelovar');
```

9.4. Upiti

Svrha upita je filtriranje podataka unutar baze podataka i prikazivanje samo onih podataka koji su definirani pojedinim upitom. Upiti se dijele na jednostavne i složene upite. Jednostavni se upiti izvršavaju nad jednom relacijom dok se složeni izvršavaju kombinacijom više relacija.

9.4.1. Jednostavni upiti

9.4.1.1. Upit 1

Prikazani upit prikazuje sve županije sa svim pripadajućim podacima. Na slici 15 prikazan je rezultat navedenog upita.

```
SELECT * FROM zupanija;
```

zupanija_id	naziv	sjediste
1	Zagrebačka županija	Zagreb
2	Krapinsko-zagorska županija	Krapina
3	Sisačko-moslavačka županija	Sisak
4	Karlovačka županija	Karlovac
5	Varaždinska županija	Varaždin
6	Koprivničko-križevačka županija	Koprivnica
7	Bjelovarsko-bilogorska županija	Bjelovar
8	Primorsko-goranska županija	Rijeka
9	Ličko-senjska županija	Gospić
10	Virovitičko-podravska županija	Virovitica
11	Požeško-slavonska županija	Požega
12	Brodsko-posavska županija	Slavonski Brod
13	Zadarska županija	Zadar

Slika 15: Rezultat prvog upita [autorski rad]

9.4.1.2. Upit 2

Sljedeći upit prikazuje imena, prezimena te datum rođenja onih popisivača koji su rođeni u 2000. godini. Na slici 16 prikazan je rezultat navedenog upita.

```
SELECT concat(popisivac.ime, ' ', popisivac.prezime) AS 'Ime i prezime popisivača', popisivac.datum_rodenja AS 'Datum rođenja' FROM popisivac WHERE popisivac.datum_rodenja BETWEEN cast('2000-1-1' AS DATE) AND cast('2000-12-31' AS DATE);
```

Ime i prezime popisivača	Datum rođenja
Matej Forjan	2000-09-25
Tatjana Kumer	2000-03-12

Slika 16: Rezultat drugog upita [autorski rad]

9.4.1.3. Upit 3

Sljedeći upit prikazuje ID, naziv, godinu osnutka te opis onih medija koji su osnovani u 20. stoljeću ili ranije. Na slici 17 prikazan je rezultat navedenog upita.

```
SELECT medij.medij_id AS 'ID medija', medij.naziv AS 'Naziv medija',
medij.godina_osnutka AS 'Godina osnutka', medij.opis AS 'Opis medija'
FROM medij WHERE godina_osnutka < 2001;
```

ID medija	Naziv medija	Godina osnutka	Opis medija
1	Večernji list	1959	Nacionalne dnevne novine sa sjedištem u Zagrebu.
4	Jutarnji list	1998	Hrvatski dnevnik s nakladom od oko 25 000 primjeraka.

Slika 17: Rezultat trećeg upita

9.4.2. Složeni upiti

9.4.2.1. Upit 4

Sljedeći upit prikazuje naziv županije te broj gradova u njima, ali samo za one županije koje imaju 5 ili manje gradova. Popis je sortiran silazno prema broju gradova. Na slici 18 prikazan je rezultat navedenog upita.

```
SELECT zupanija.naziv AS 'Naziv županije', COUNT(*) AS 'Broj gradova'
FROM zupanija INNER JOIN grad ON zupanija.zupanija_id =
grad.zupanija_id GROUP BY zupanija.zupanija_id HAVING COUNT(*) < 6
ORDER BY 2 DESC;
```

Naziv županije	Broj gradova
Bjelovarsko-bilogorska županija	5
Vukovarsko-srijemska županija	5
Dubrovačko-neretvanska županija	5
Ličko-senjska županija	4
Koprivničko-križevačka županija	3
Virovitičko-podravska županija	3
Međimurska županija	3
Brodsko-posavska županija	2

Slika 18: Rezultat četvrtog upita [autorski rad]

9.4.2.2. Upit 5

Sljedeći upit prikazuje popis službi s njezinim pripadajućim odjelima. Na slici 19 prikazan je rezultat navedenog upita.

```
SELECT sluzba.naziv AS 'Naziv službe', odjel.naziv AS 'Naziv odjela'
FROM sluzba, odjel WHERE sluzba.sluzba_id = odjel.sluzba_id;
```

Naziv službe	Naziv odjela
Služba statistike stanovništva	Odjel popisa stanovništva
Služba statistike stanovništva	Odjel vitalne statistike i migracija
Služba statistike životnih uvjeta	Odjel statistike životnih uvjeta stanovništva

Slika 19: Rezultat petog upita [autorski rad]

9.4.2.3. Upit 6

Sljedeći upit prikazuje popisivače te popisne jedinice za koje su zaduženi. Na slici 20 prikazan je rezultat navedenog upita.

```
SELECT concat(popisivac.ime, ' ', popisivac.prezime) AS 'Ime i prezime
popisivača', popisna_jedinica.naziv AS 'Naziv popisne jedinice' FROM
popisivac INNER JOIN popisuje ON popisivac.popisivac_id =
popisuje.popisivac_id INNER JOIN popisna_jedinica ON
popisuje.popisna_jedinica_id = popisna_jedinica.popisna_jedinica_id;
```

Ime i prezime popisivača	Naziv popisne jedinice
Matej Forjan	Općina Štefanje
Tatjana Kumer	Općina Štefanje
Dijana Grubić	Općina Ivanska

Slika 20: Rezultat šestog upita [autorski rad]

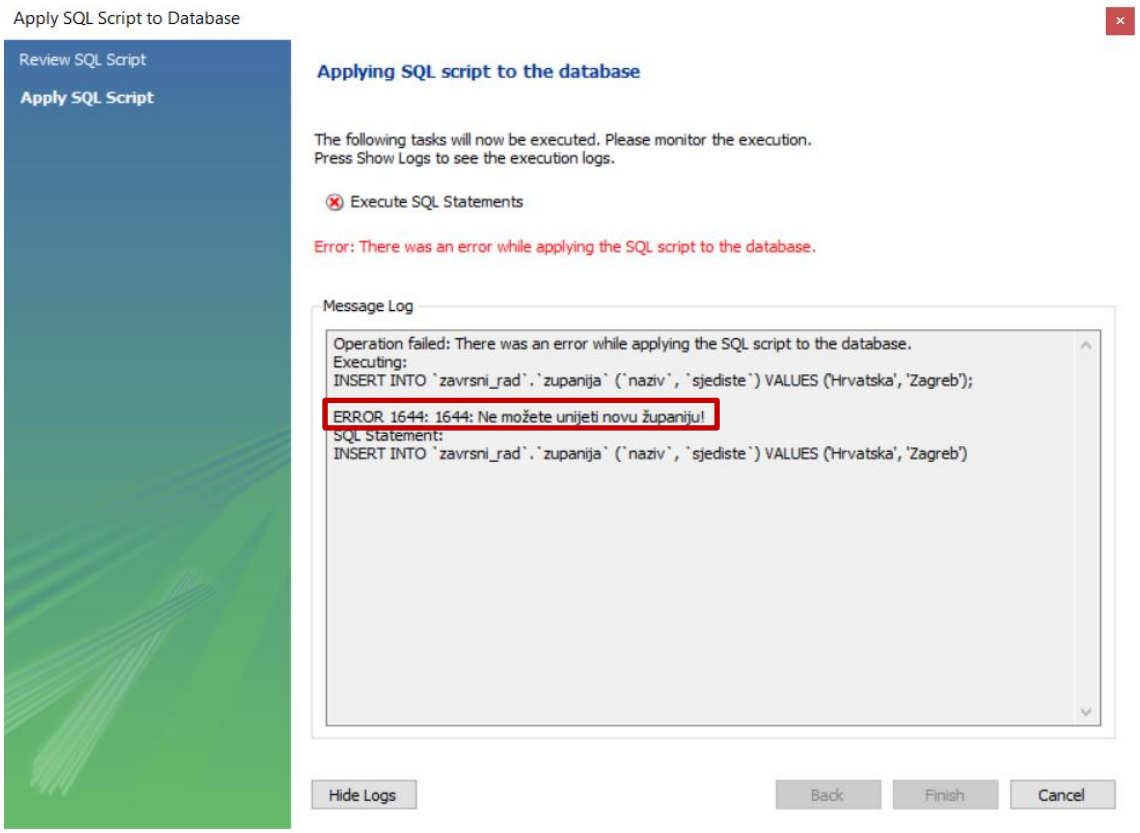
9.5. Okidači

Aktivne baze podataka su one baze u kojima se podaci automatski ažuriraju ili se pojedine stavke brišu ili mijenjaju nakon određenog događaja. Sve to navedeno moguće je implementirati pomoću okidača (eng. *Trigger*). U ovom će radu biti prikazana tri okidača. Okidač reagira na radnje koje se izvršavaju nad relacijama unutar baze podataka. Okidač se može koristit prije ili nakon unosa, brisanja ili ažuriranja podataka. Okidač se kreira naredbom CREATE TRIGGER nakon koje definiramo naziv okidača, vrijeme kada se okidač aktivira (BEFORE ili AFTER), radnju nakon koje se okidač aktivira (INSERT, DELETE ili UPDATE), relaciju nad kojom želimo izvoditi okidač te funkciju koja se izvršava unutar okidača.

9.5.1. Okidač 1

Prvi okidač pod nazivom “onemoguci_unos_zupanije“ onemogućava unos novih županija u bazu podataka. Republika Hrvatska podijeljena je u 21 županiju koje su unesene u bazu te je iz tog razloga zabranjen unos novih županija koje ne postoje. U nastavku će biti prikazan programski kod navedenog okidača i poruka koju okidač prikazuje nakon što se pokuša izvesti zabranjena radnja. Na slici 21 prikazana je poruka navedenog okidača.

```
CREATE TRIGGER `onemoguci_unos_zupanije`
BEFORE INSERT
ON `zupanija` FOR EACH ROW
BEGIN
    DECLARE brojac INT DEFAULT ROW_COUNT();
    DECLARE upozorenje VARCHAR(500);
    IF brojac <> 1 THEN
    SET upozorenje = 'Ne možete unijeti novu županiju!';
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = upozorenje;
    END IF;
END
```

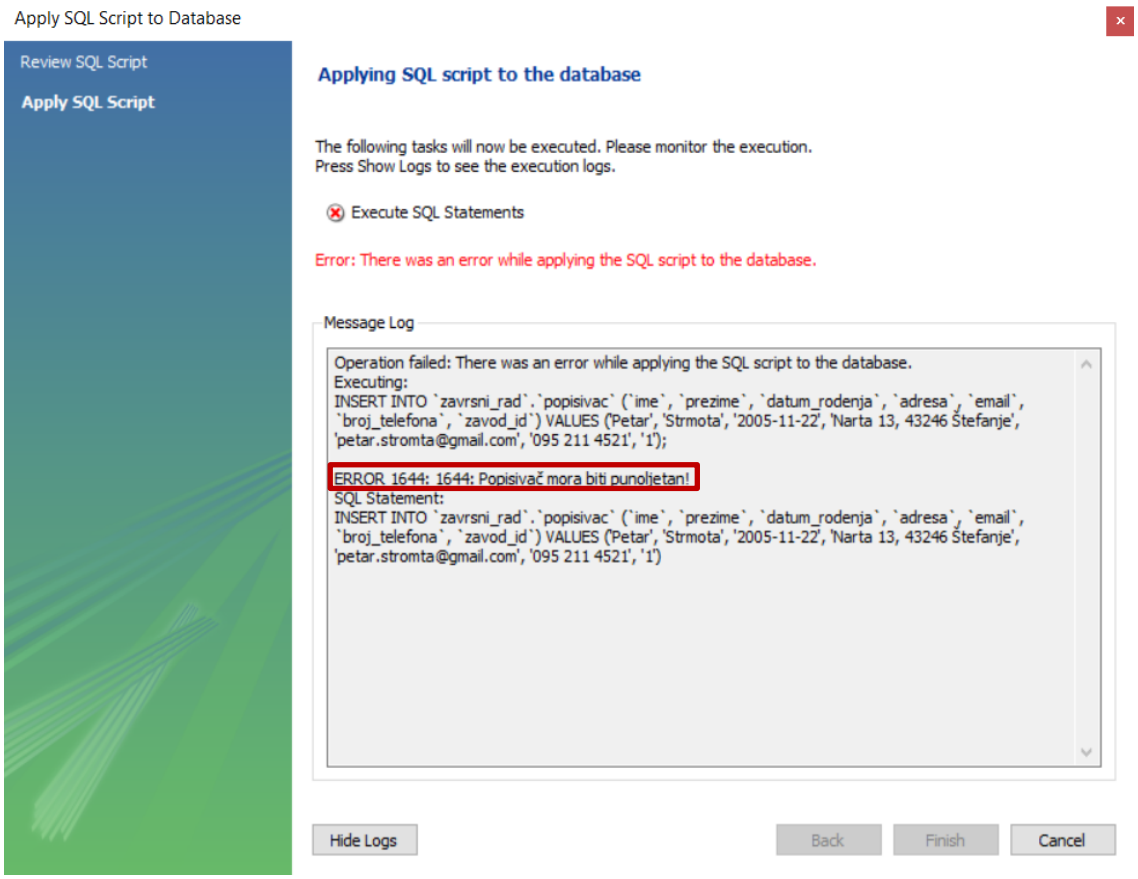


Slika 21: Poruka prvog okidača [autorski rad]

9.5.2. Okidač 2

Drugi okidač pod nazivom „provjera_starosti_popisivaca“ onemogućava unos popisivača koji nije punoljetan. Na slici 22 prikazana je poruka navedenog okidača.

```
CREATE TRIGGER `provjera_starosti_popisivaca`
BEFORE INSERT
ON `popisivac` FOR EACH ROW
BEGIN
     IF (DATEDIFF(CURRENT_DATE(), NEW.datum_rodenja) < 6574) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Popisivač mora biti
punoljetan!';
    END IF;
END
```



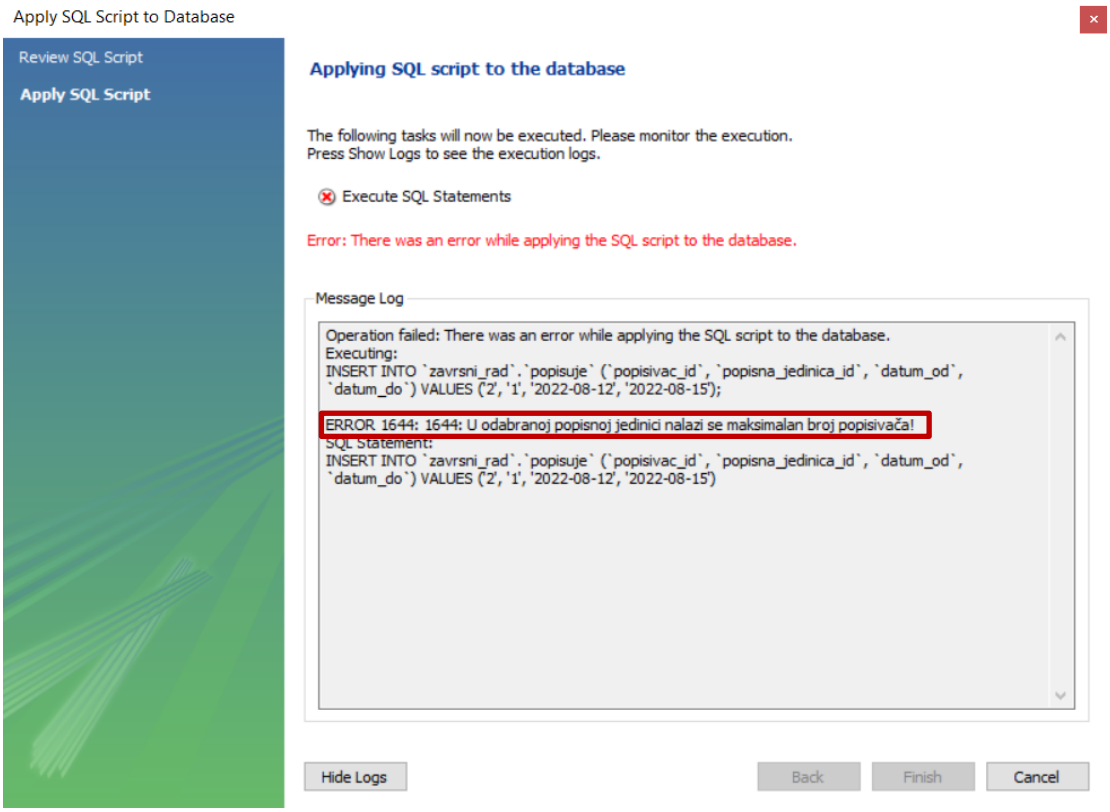
Slika 22: Poruka drugog okidača [autorski rad]

9.5.3. Okidač 3

Treći okidač pod nazivom „provjera_broja_popisivaca_u_popisnoj_jedinici“ provjerava broj popisivača koji su dodijeljeni pojedinoj popisnoj jedinici. Svaka popisna jedinica ima definiran maksimalan broj popisivača koji se nalazi u njoj te nije moguće pridružiti više popisivača u popisnu jedinicu od tog definiranog broja. Na slici 23 prikazana je poruka navedenog okidača.

```
CREATE TRIGGER `provjera_broja_popisivaca_u_popisnoj_jedinici`
BEFORE INSERT
ON `popisuje`
FOR EACH ROW
BEGIN
     DECLARE preostalo INT;
     SELECT popisna_jedinica.broj_popisivaca INTO preostalo FROM
popisna_jedinica
```

```
      WHERE popisna_jedinica.popisna_jedinica_id =
NEW.popisna_jedinica_id;
      IF preostalo = 0 THEN
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "U odabranoj
popisnoj jedinici nalazi se maksimalan broj popisivača!";
      ELSE
            UPDATE popisna_jedinica SET broj_popisivaca = preostalo -
1
         WHERE popisna_jedinica_id = NEW.popisna_jedinica_id;
      END IF;
END
```



Slika 23: Poruka trećeg okidača [autorski rad]

10. Izrada aplikacijskog sučelja baze podataka

10.1. Microsoft Visual Studio 2022

Microsoft Visual Studio koristi se za razvoj računalnih programa, web aplikacija, web stranica te mobilnih aplikacija. Visual Studio podržava velik broj različitih programskih jezika (ukupno 36), a neki od njih su C, C++, JavaScript, C#, HTML, VSS, XML. Visual Studio dolazi u nekoliko verzija od kojih je izdanje Community potpuno besplatno. Najnovija verzija ovog programa je Microsoft Visual Studio 2022. U ovom će radu biti izrađena Windows Forms aplikacija u programskom jeziku C# [17].

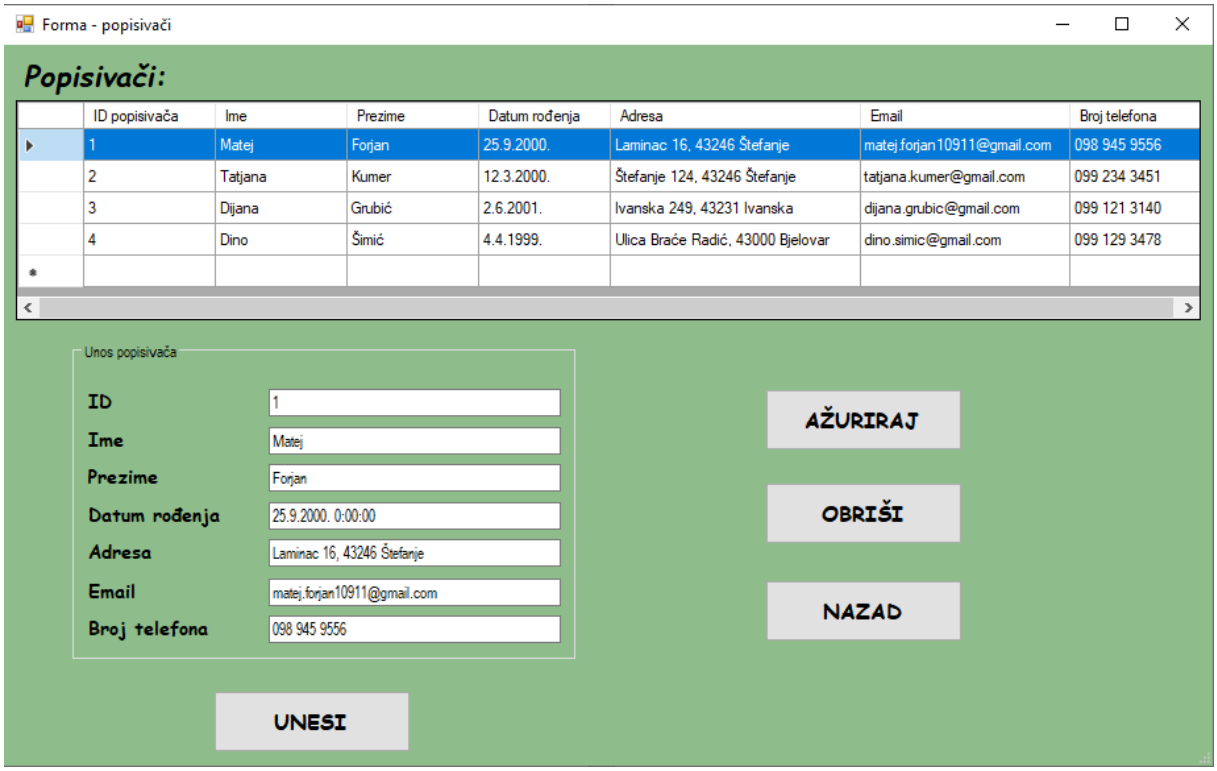
10.2. Početna forma

Prilikom pokretanja aplikacije, korisniku se otvara početna forma. Korisnik ima mogućnost odabira na gumb „POPISIVAČI“ gdje se otvara forma o popisivačima te na gumb „POPISNE JEDINICE“ gdje se otvara forma za pridruživanje popisivača popisnoj jedinici. Na slici 24 prikazana je početna forma.

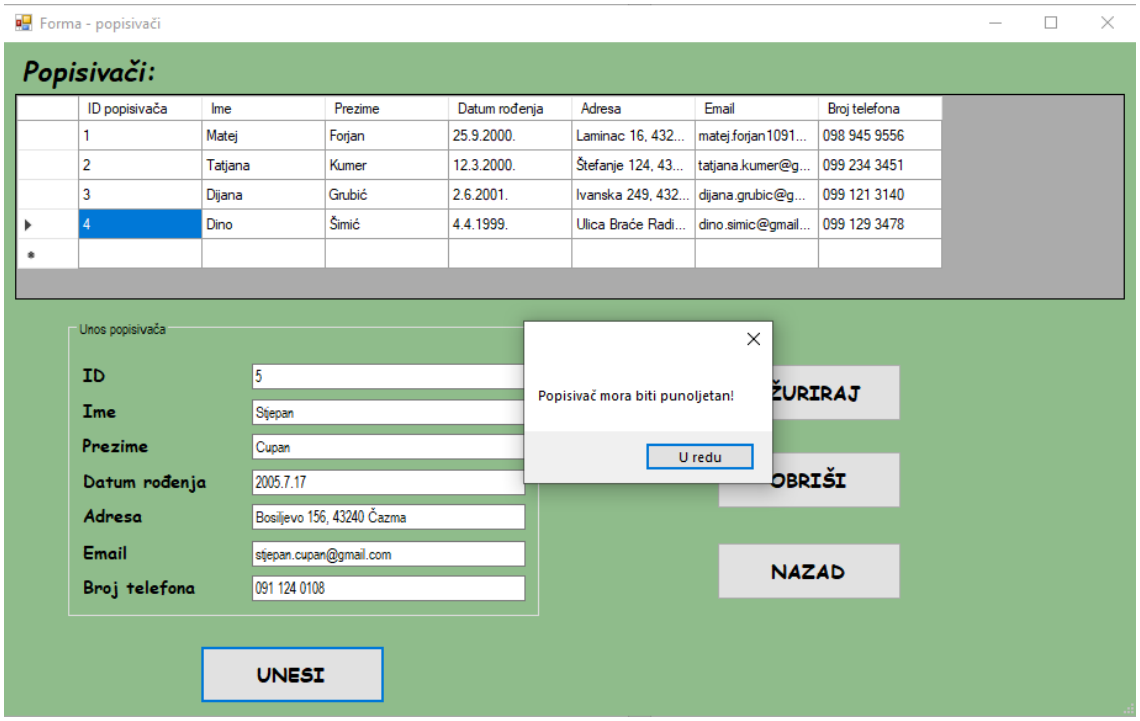
Slika 24: Početna forma [autorski rad]

10.3. Forma – Popisivači

Na formi „Popisivači“ prikazanoj na slici 25, korisnik ima mogućnost pregleda svih registriranih popisivača s njihovim osobnim podacima poput ID-a, imena, prezimena, datuma rođenja, adrese, email adrese te broja telefona. Osim toga, korisnik može unijeti novog popisivača, ažurirati podatke o unesenom popisivaču te obrisati popisivača iz baze podataka. Pritiskom na gumb „NAZAD“, otvara se početna forma. Prilikom unošenja novog popisivača nije dozvoljen unos popisivača koji je maloljetan, a ukoliko korisnik pokuša unijeti maloljetnog popisivača, aplikacija će ispisati grešku te se unos neće izvršiti. Na slici 26 prikazana je poruka koju aplikacija prikazuje prilikom pokušaja unosa maloljetnog popisivača.



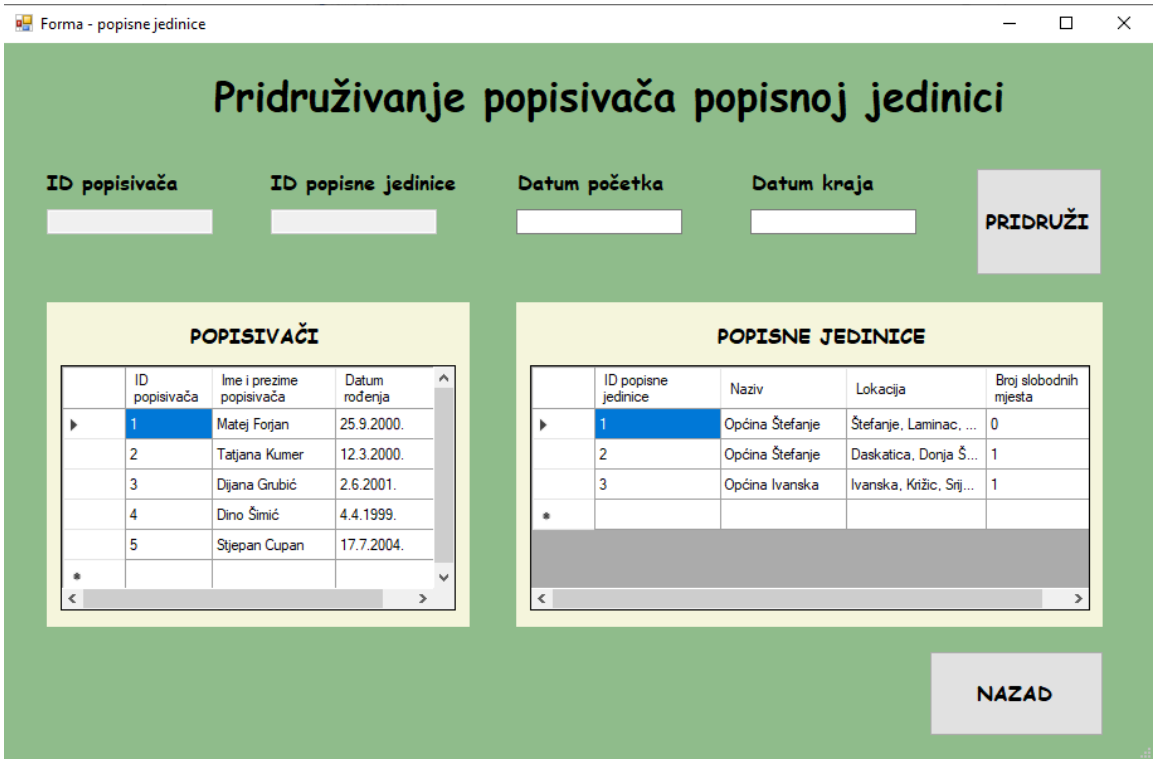
Slika 25: Forma – Popisivači [autorski rad]



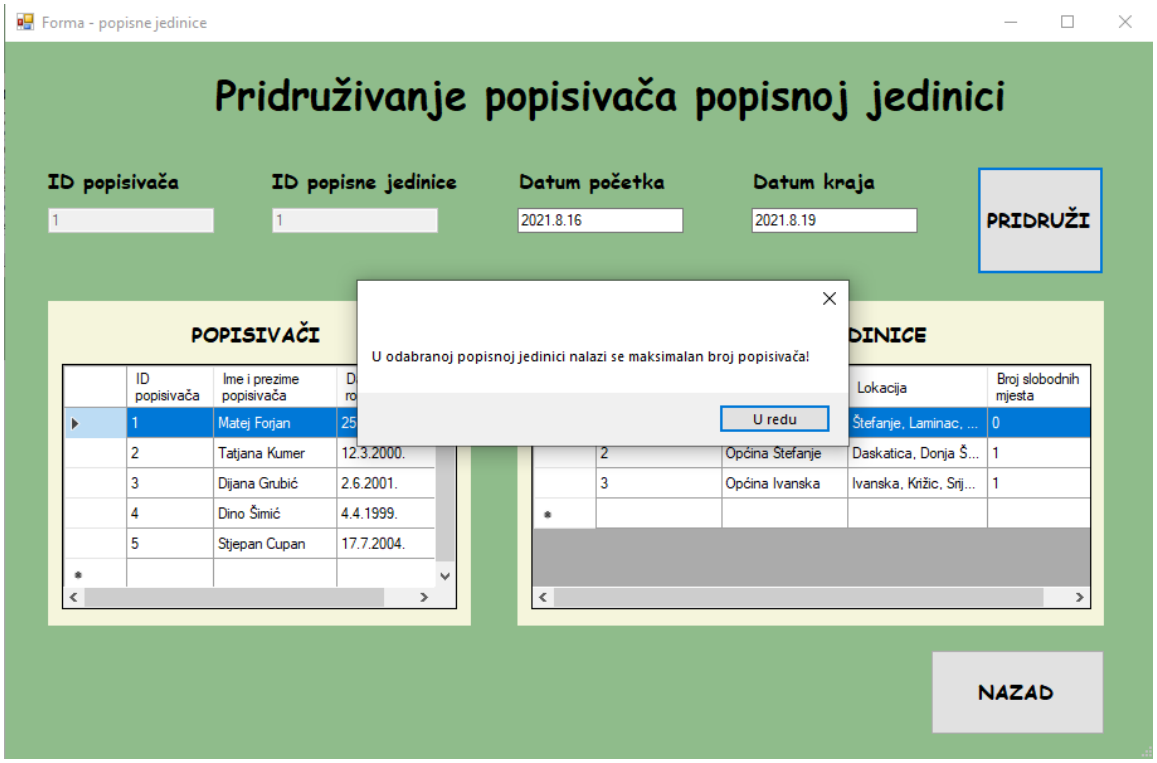
Slika 26: Pokušaj unosa maloljetnog popisivača [autorski rad]

10.4. Forma – Popisne jedinice

Na formi „Popisne jedinice“ prikazanoj na slici 27 korisnik ima mogućnost pridruživanja odabranog popisivača odabranoj popisnoj jedinici. Osim toga, unosi vremenski period u kojem će popisivač izvršiti popisivanje u popisnoj jedinici. Pritiskom na gumb „NAZAD“, otvara se početna forma. Prilikom pridruživanja popisivača popisnoj jedinici, potrebno je paziti na broj slobodnih mjesta u popisnoj jedinici. Ukoliko u popisnoj jedinici više nema mjesta, popisivač se neće pridružiti toj popisnoj jedinici te će aplikacija prikazati pripadajuću poruku. Na slici 28 prikazana je poruka koju aplikacija prikazuje prilikom pokušaja pridruživanja popisivača u popisnu jedinicu u kojoj više nema slobodnih mjesta.



Slika 27: Forma – Popisne jedinice [autorski rad]



Slika 28: Pokušaj pridruživanja popisivača u popisnu jedinicu u kojoj nema slobodnih mjesta [autorski rad]

11. Zaključak

Tema ovog rada bila je dizajniranje baze podataka za popis stanovništva. U prvom je dijelu rada opisan Državni zavod za statistiku koji je zadužen za organizaciju i provedbu popisa stanovništva. Definirana je misija i vizija DZS-a, a dijagramom kompozicija prikazano unutarnje ustrojstvo istog. U dijagramu dekompozicija svi su elementi organizacijske jedinice.

Dijagramom toka podataka kontekstne razine i dijagramom toka podataka niže razine detaljnije je prikazan postupak provedbe popisa stanovništva. U radu su opisani svi elementi koji se nalaze u izrađenim dijagramima. ERA dijagramom je također prikazan postupak provedbe popisa stanovništva. Opisane su sve relacije sa svim pripadajućim atributima, primarnim i vanjskim ključevima te veze između pojedinih relacija. Navedeni dijagrami izrađenu su u besplatnom online alatu draw.io.

Baze podataka imaju veliku važnost u svakodnevnom životu jer se sve temelji na njima. Iz tog je razloga veći dio rada temeljen upravo na bazi podataka. Definirana je baza podataka te svi važniji pojmovi koji su povezani uz nju. Nakon teorijskog dijela o bazi podataka, dizajnirana je baza podataka temeljena na izrađenom ERA dijagramu. Za dizajniranje baze podataka koristio se alat MySQL Workbench. Nakon što su se izradile relacije i veze između njih, u alatu je napisano nekoliko jednostavnih i složenih upita koji pojednostavljuju prikaz podataka. Osim toga, kreirana su tri okidača koji zabranjuju da se unutar baze podataka pokušaju izvršiti nedozvoljene radnje.

Na kraju je rada izrađeno aplikacijsko sučelje za bazu podataka u alatu Microsoft Visual Studio 2022. Za potrebe ovog rada napravljena je Windows Forms aplikacija napisana u programskom jeziku C#.

Popis literature

[1] Državni zavod za statistiku (bez dat.) „*O Zavodu*“ [Na internetu] Dostupno: https://dzs.gov.hr/o-zavodu/9 [pristupano 06.08.2022.].

[2] P. Grd, „Dijagram dekompozicije“ nastavni materijal na predmetu Državni i upravni informacijski sustavi [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2020.

[3] Popis 2021. (bez dat.) „*Popis 2021.*“ [Na internetu] Dostupno: https://popis2021.hr/ [pristupano 06.08.2022.].

[4] Lucid (bez dat.) „*What is a Dana Flow Diagram?“* [Na internetu]. Dostupno: https://www.lucidchart.com/pages/data-flow-diagram [pristupano 10.02.2022.].

[5] P. Grd, „Dijagram toka podataka“ nastavni materijal na predmetu Državni i upravni informacijski sustavi [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2020.

[6] M. Kaluža, *Sustavi baza podataka*. Veleučilište u Rijeci, Rijeka, 2008.

[7] R. Manger, *Baze podataka*. Sveučilište u Zagrebu, Prirodoslovno matematički fakultet, Zagreb, 2003.

[8] P. Grd, „ER dijagram“ nastavni materijal na predmetu Državni i upravni informacijski sustavi [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2020.

[9] M. Čubrilo, *Pravilo rezolucije i relacijske baze podataka.* Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 1988. [Na internetu]. Dostupno: Hrčak, https://hrcak.srce.hr/. [pristupano 21.02.2022.].

[10] Oracle (bez dat.) „*What is MySQL?*“ [Na internetu] Dostupno: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html [pristupano 24.02.2022.].

[11] Talend (bez dat.) „*What is MySQL?*“ [Na internetu] Dostupno: https://www.talend.com/resources/what-is-mysql/ [pristupano 24.02.2022.]

[12] Oracle (bez dat.) „*Supported Platforms: MySQL Database?*“ [Na internetu] Dostupno: https://www.mysql.com/support/supportedplatforms/database.html [pristupano 24.02.2022.].

[13] Oracle [Slika] (bez dat.) Dostupno: https://www.mysql.com/ [pristupano 24.02.2022.].

[14] W3Schools (bez dat.) „*MySQL What is DDL, DML and DCL?*“ [Na internetu] Dostupno: https://www.w3schools.in/mysql/ddl-dml-dcl/ [pristupano 25.02.2022.].

[15] Oracle (bez dat.) „*MySQL Workbench*“ [Na internetu] Dostupno: https://www.mysql.com/products/workbench/ [pristupano 25.02.2022.].

[16] „MySQL Workbench“ (bez dat.). u *Wikipedia, the Free Encyclopedia*. Dostupno: https://en.wikipedia.org/wiki/MySQL_Workbench [pristupano 25.02.2022.]

[17] Microsoft Visual Studio (bez dat.) „*It's How You Make Software*“ [Na internetu] Dostupno: https://visualstudio.microsoft.com/ [pristupano 13.08.2022.].

Popis slika

Slika 1: Dekompozicijski dijagram Državnog zavoda za statistiku [autorski rad] 6
Slika 2: Simbol procesa [autorski rad].....9
Slika 3: Simbol toka podataka [autorski rad].....9
Slika 4: Simbol vanjskog objekta [autorski rad].....10
Slika 5: Simbol spremišta podataka [autorski rad] 10
Slika 6: Dijagram toka podataka – razina konteksta [autorski rad] 11
Slika 7: Dijagram toka podataka – niža razina [autorski rad].....12
Slika 8: Shematski prikaz pristupa podacima upotrebom SUBP-a [6, str. 18] 13
Slika 9: ERA dijagram [autorski rad] 18
Slika 10: Logo MySQL-a [13].....27
Slika 11: Početno sučelje alata MySQL Workbench [autorski rad].....29
Slika 12: Sučelje alata MySQL Workbench nakon odabrane konekcije [autorski rad] 30
Slika 13: Prozor za kreiranje tablica [autorski rad] 31
Slika 14: Prozor za postavljanje vanjskog ključa [autorski rad] 32
Slika 15: Rezultat prvog upita [autorski rad].....33
Slika 16: Rezultat drugog upita [autorski rad] 33
Slika 17: Rezultat trećeg upita 34
Slika 18: Rezultat četvrtog upita [autorski rad].....34
Slika 19: Rezultat petog upita [autorski rad].....35
Slika 20: Rezultat šestog upita [autorski rad] 35
Slika 21: Poruka prvog okidača [autorski rad].....37
Slika 22: Poruka drugog okidača [autorski rad].....38
Slika 23: Poruka trećeg okidača [autorski rad].....39
Slika 24: Početna forma [autorski rad].....40
Slika 25: Forma – Popisivači [autorski rad].....41
Slika 26: Pokušaj unosa maloljetnog popisivača [autorski rad] 42
Slika 27: Forma – Popisne jedinice [autorski rad] 43

Slika 28: Pokušaj pridruživanja popisivača u popisnu jedinicu u kojoj nema slobodnih mjesta [autorski rad] 43

Popis tablica

Tablica 1: Prikaz relacije „Stanovnik“ s pripadajućim dijelovima [autorski rad] 17
Tablica 2: Relacija „zavod“ [autorski rad] 19
Tablica 3: Relacija „sluzba“ [autorski rad] 20
Tablica 4: Relacija „odjel“ [autorski rad] 20
Tablica 5: Relacija „popisivac“ [autorski rad] 20
Tablica 6: Relacija „popisuje“ [autorski rad] 20
Tablica 7: Relacija „popisna_jedinica“ [autorski rad] 21
Tablica 8: Relacija „zupanija“ [autorski rad] 21
Tablica 9: Relacija „opcina“ [autorski rad] 21
Tablica 10: Relacija „grad“ [autorski rad] 21
Tablica 11: Relacija „stanovnik“ [autorski rad] 22
Tablica 12: Relacija „sustav“ [autorski rad] 22
Tablica 13: Relacija „vjerodajnica“ [autorski rad] 22
Tablica 14: Relacija „izvjestaj“ [autorski rad] 22
Tablica 15: Relacija „objavljuje_se“ [autorski rad] 23
Tablica 16: Relacija „medij“ [autorski rad] 23
Tablica 17: Relacija „vrsta_medija“ [autorski rad] 23
Tablica 18: Relacijska shema [autorski rad] 26