

# Model baze podataka za računalni servis

---

Kukučka, Luka

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Economics in Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:145:061587>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-10-04**



Repository / Repozitorij:

[EFOS REPOSITORY - Repository of the Faculty of Economics in Osijek](#)



Sveučilište Josipa Jurja Strossmayera u Osijeku  
Ekonomski fakultet u Osijeku  
Sveučilišni prijediplomski studij (*Poslovna informatika*)

Luka Kukučka

**Model baze podataka za računalni servis**

Završni rad

Osijek, 2023.

Sveučilište Josipa Jurja Strossmayera u Osijeku  
Ekonomski fakultet u Osijeku  
Sveučilišni prijediplomski studij (*Poslovna informatika*)

Luka Kukučka

## **Model baze podataka za računalni servis**

Završni rad

**Kolegij: Baze podataka i poslovni procesi**

JMBAG: 0152211212

e-mail: [lkukucka@efos.hr](mailto:lkukucka@efos.hr)

Mentor: prof.dr.sc. Branimir Dukić

Komentor: Saša Mitrović, mag.oec

Osijek, 2023.

Josip Juraj Strossmayer University of Osijek  
Faculty of Economics and Business in Osijek  
Undergraduate Study (Business informatics)

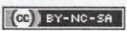
Luka Kukučka

**Database model for computer service**

Final paper

Osijek, 2023.

**IZJAVA**  
**O AKADEMSKOJ ČESTITOSTI,**  
**PRAVU PRIJENOSA INTELKTUALNOG VLASNIŠTVA,**  
**SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA**  
**I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA**

1. Kojom izjavljujem i svojim potpisom potvrđujem da je završni (navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska*. 
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o visokom obrazovanju i znanstvenoj djelatnosti, NN 119/2022).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

**Ime i prezime studenta/studentice:** Luka Kukučka

**JMBAG:** 0152211212

**OIB:** 79899625175

**e-mail za kontakt:** luka.kukucka1@gmail.com

**Naziv studija:** preddiplomski studij; poslovna informatika

**Naslov rada:** Model baze podataka za računalni servis

**Mentor/mentorica rada:** prof.dr.sc. Branimir Dukić

U Osijeku, 20.9.2023. godine

Potpis Luka Kukučka

## Sažetak

Računalni servisi nude široki spektar usluga koje uključuju popravak računala, prodaju gotovih računala ili pak sastavljanje računala od komponenti koje klijent zatraži. U ovakvim procesima često sudjeluje dokumentacija poput radnog naloga, potvrde o zaprimanju opreme, izvještaja o dijagnostici, ponude za popravak, potvrdi o popravku, računa, te potvrde o preuzimanju popravljene robe. Pet računalnih servisa u Osijeku bilo je posjećeno u sklopu primarnog istraživanja, koje je uključivalo prikupljanje radnih naloga iz svakog od servisa, te se nakon obavljenog kratkog intervjua sa svakim od zaposlenika određenog servisa došlo do zaključaka o strukturi procesa koji se odvijaju u računalnim servisima u Republici Hrvatskoj te o obuhvatu podataka koji treba sadržavati optimalna relacijska baza podataka računalnog servisa. U tu svrhu su prikupljeni i analizirani obrasci radnih naloga koji se koriste u računalnim servisima. Temeljem sagledanog podatkovnog obuhvata definirani su entiteti i pripadajući atributi. Relacijska baza podataka modelirana je metodom sinteze, a uočene nekontrolirane redundancije otklonjene su upotrebom vertikalne normalizacije. Optimalan model relacijske baze podataka za računalni servis kreiran je i testiran upotrebom Oracle-ovog relacijsko sustava za upravljanje bazama podataka MySQL Server u trenutnoj verziji 8.0.33. Uz teorijsku vrijednost kreirani model ima i praktičnu vrijednost jer može poslužiti poslovnim subjektima koji se bave razvoje programskih aplikacija kao osnova u razvoju programskih rješenja namijenjenih radu računalnih servisa.

**Ključne riječi:** računalni servis, radni nalog, baza podataka, relacija, MySQL

## Summary

Computer services offer a wide range of services that include computer repair, sale of ready-made computers or assembly of computers from components requested by the client. Such processes often involve documentation such as a work order, receipt of equipment, diagnostic report, repair offer, repair certificate, invoice, and receipt of repaired goods. Five computer services in Osijek were visited as part of the primary research, which included the collection of work orders from each of the services, and after a short interview with each of the employees of a particular service, conclusions were reached about the structure of the processes that take place in computer services in the Republic of Croatia and about the scope of data that should be contained in the optimal relational database of the computer service. For this purpose, work order forms used in computer services were collected and analyzed. Entities and associated attributes are defined based on the observed data scope. The relational database was modeled using the synthesis method, and the observed uncontrolled redundancies were removed using vertical normalization. The optimal relational database model for computer service was created and tested using Oracle's relational database management system MySQL Server in the current version 8.0.33. In addition to the theoretical value, the created model also has a practical value because it can be used by business entities involved in the development of software applications as a basis in the development of software solutions intended for the operation of computer services.

**Keywords:** computer service, work order, database, relation, MySQL

# Sadržaj

1. Uvod .....	1
2. Metodologija istraživanja .....	3
3. Baze podataka s osvrtom na relacijske baze podataka .....	5
3.1. Pojam baze podataka .....	5
3.2. Uobičajeno korišteni modeli baza podataka u poslovnoj praksi .....	9
3.3. Uloga relacijskog modela podataka u poslovanju .....	12
4. Relacijski model podataka .....	13
4.1. Nastanak i povijest relacijskog sustava za upravljanje relacijskim modelom .....	13
4.2. Temeljni pojmovi vezani uz relacijski model podataka .....	16
4.2.1. Jedinka i relacija .....	16
4.2.2. Atributi i domene vrijednosti .....	18
4.2.3. Relacije unutar tablice i između tablica .....	20
4.2.4. Vertikalna normalizacija u funkciji oblikovanja i eliminacija nekontrolirane redundancije .....	23
4.2.5. SQL .....	25
5. Računalni servis .....	28
5.1. Računalni servisi na području istočnog dijela Republike Hrvatske .....	28
5.2. Dokumentacija koja se koristi u računalnim servisima kao temelj za modeliranje baze podataka .....	29
6. Model baze podataka za računalni servis .....	31
6.1. Model baze podataka za računalni servis u pseudo kodu .....	31
6.2. Model baze podataka računalnog servisa u SQL-u .....	34
6.3. Primjeri testiranja ispravnosti modela baze podataka računalnog servisa .....	39
7. Zaključak .....	45
8. Popis literature .....	47



# 1. Uvod

Zahvaljujući razvoju mikroprocesorske tehnologije, odnosno nastanku super visoko integriranih procesora (engl. Very Large-Scale Integration - VLSI) kao i razvitku drugih elektroničkih komponenti digitalnih računalnih uređaja u 80-tim godinama prošlog stoljeća, gotovo istovremeno, pojavila su se osobna (engl. Personal Computer – PC) i kućna, odnosno mala računala. Relativno prihvatljiva cijena tih računalni učinila ih je dostupnim i popularnim kako u poslovnom svijetu tako i kod kućanstava. Potražnja za osobnim i kućnim računalima potakla je potrebu za širenjem kako mreže distributera, odnosno prodavača ovih uređaja, tako i mreže računalnih servisa. Kako su osobna računala u to vrijeme bila modularne građe distribucija te servisi, osim prodaje samih računala, koja su se često naručivala kao skup odabranih komponenti, prodavali su i obavljali zamjenu strojnih komponenti (engl. Hardware), ali prodaju i ugradnju modularnog programskog podsustava kao i prodaju i priključivanje perifernih uređaja kao što su pisači, crtači, čitači i drugo. Računalni servisi, kao mjesta gdje su se računala popravljala u slučaju strojnih kvarova, obavljali su uobičajeno i održavanja računala i računalne periferne opreme, u vidu održavanja i programske i podatkovne komponente (engl. Software), kao i osnovnih strojnih te perifernih uređaja osiguravajući korisnicima na taj način tehničku podršku. U devedesetim godinama prošlog stoljeća, osim samim računalima, trgovine se i servisi s računalnom opremom bave i mrežnom infrastrukturom, posebice u domeni žičanih lokalnih mreža (engl. Local Area network – LAN) te povezivanja tih mreža na globalnu računalnu mrežu Internet, da bi s novim milenijem fokus stavljen na bežične mreže, multifunkcionalne digitalne uređaje kao što su multimedijски uređaji, mobilni telefoni, prijenosna računala (Notebook i tablet računala), poslužiteljski sustavi i drugo. Zbog konvergencija digitalnih tehnologija danas je u domeni rada trgovina i servisa s računalnom opremom čitav niz različitih digitalnih naprava i uređaja, kao i čitav niz programskih komponenti.

Nastanak i razvoj malih, posebice osobnih računala pokrenuo je američki div u računalnoj tehnologiji IBM (engl. International Business Machines), no ubrzo su ga počeli, kopiranjem tehnologije, slijediti kako drugi proizvođači računala u Americi, tako i u svijetu. Ubrzo su Azijski proizvođači komponenti cijenom, a nerijetko i kvalitetom, nadmašili IBM i druge američke proizvođače, što je rezultiralo poplavom tzv. PC klonova. U takvim uvjetima uz mrežu distributera

i servisa velikih američkih proizvođača osobnih računala pojavila se i mreža distributer i servisa računala PC klonova. Ovakva se situacija zadržala i u današnje doba. S obzirom da su mrežu distributera i servisera klonova često činili i čine mali i srednje veliki poslovni subjekti kod njih se javljala i javlja potreba za malim i neovisnim programskim rješenjima koja omogućavaju učinkovito upravljanje poslovnim operacijama, praćenje zalihe gotovih proizvoda i rezervnih dijelova, praćenje trenutnih i potencijalnih klijentima, uspješno pružanje usluga i drugo. Temelj su takvih programskih rješenja baze podataka u kojima se bilježe, kroz programske aplikacije, prikupljeni podaci. Sukladno navedenom provedeno se istraživanje fokusiralo na izradu optimalnog modela relacijske baze podataka za servise računala i računalne opreme.

## 2. Metodologija istraživanja

Računalni servisi zaprimaju neispravnu računalnu opremu, obavljaju preglede, popravak i povrat te iste opreme komitentima. Ovaj proces prati odgovarajuća dokumentacija. Da bi se načinila programska aplikacija koja bi osigurala praćenje procesa unutar računalnih servisa, te omogućila izradu i izdavanje računa koji se odnose na popravak računalne opreme, potrebno je prvo modelirati, te zatim izraditi odgovarajuću bazu podataka. Takvu bazu podataka potrebno je testirati i tek nakon uspješnog testiranja, takva se baza podataka može pustiti u produkciju. Prema tome, problem koji se nameće je s jedne strane istražiti procese vezane za rad računalnih servisa kroz analizu dokumentacije koja prati te procese, a s druge strane istražiti aktualne modele baza podataka te odabrati optimalan model za kreiranje baze podataka koja će omogućiti kroz evidentiranje poslovne strukture i poslovnih procesa praćenje rada i podršku računalnim servisima. U tom je smislu potrebno modelirati bazu podataka na temelju koje će se razviti odgovarajuća programska aplikacija za evidentiranje procesa u računalnim servisima.

S obzirom na navedeni problem istraživanja, postavljaju se sljedeći ciljevi istraživanja :

1. Istražiti što su to baze podataka, koji su aktualni modeli, te sagledati zbog čega je relacijski model najprisutniji u poslovnoj praksi
2. Istražiti relacijski model baze podataka, te vidjeti koje komponente sudjeluju u modeliranju baze podataka namijenjene računalnom servisu
3. Istražiti koji su računalni servisi aktualni na lokalnom tržištu, zatim sagledati procese vezane uz računalni servis, te dokumentaciju koja prati te procese
4. Izraditi relacijski model baze podataka namijenjen računalnom servisu, te testirati isti

Kako bi navedeni ciljevi istraživanja bili ostvareni, potrebno je koristiti sljedeće znanstvene metode :

- Metoda analize i sinteze
- Metoda deskripcije
- Metoda klasifikacije i generalizacije
- Metoda apstrakcije
- Metoda agregacije

- Metoda testiranja
- Metoda eksperimenta
- Metoda intervjua
- Povijesna metoda
- Statistička metoda

Istraživački je proces obuhvatio primarno istraživanje u okviru kojega je posjećeno i analizirano pet različitih računalnih servisa iz grada Osijeku. U okviru primarnog istraživanja sagledani su poslovni procesi i poslovna dokumentacija te su utvrđene ekvivalentnosti u podatkovnom obuhvatu između pet različitih servisnih naloga svakog od posjećениh servisa. Tijekom primarnih istraživanja intervjuirani su zaposlenici servisa kako bi se dobio uvid u obuhvat usluga koje servis nudi i te procesa koji se u okviru poslovne aktivnosti odvijaju u svakom od servisa. Prikupljeni podaci tijekom primarnih istraživanja, kao i prikupljene informacije i znanja u okviru sekundarnih desk istraživanja poslužili su kao temelj za sintezu rezultata istraživanja u vidu modela baze podataka primjenjivog u poslovnim aplikacijama namijenjenim praćenju procesa u računalnim serviserima. Takav je model potom testiran i predstavljen u ovom završnom radu.

Izvori sekundarnih i tercijarnih informacija i znanja uključuju dostupnu tiskanu i digitalnu literaturu. Prilikom sastavljanja izvješća o rezultatima istraživanja, odnosno prilikom pisanja ovog završnog rada korišten je programski alat za obradu teksta Microsoft Word kao i web programski alat za izradu dijagrama i tablica Lucidchart, a za modeliranje i testiranje modela baze podataka besplatni sustav za upravljanje relacijskom bazom podataka, pod licencom Oracle-a, MySQL Server 8.0.33. Sukladno zakonu o zaštiti osobnih podataka, testiranje funkcionalnosti izrađene baze podataka provedeno je nad izmišljenim podacima.

### 3. Baze podataka s osvrtom na relacijske baze podataka

Postoji velik broj različitih vrsta baza podataka, no u ovom je istraživanju korišten relacijski model baze podataka koji je široko rasprostranjen zbog svoje jednostavnosti, fleksibilnosti i mogućnosti normalizacije koja smanjuje redundanciju podataka. Takva se vrsta baze podataka temelji na četiri glavna načela koja se skraćeno nazivaju ACID. Slovo „A“ predstavlja „Atomicitet“ (engl. Atomicity), što znači da svaka operacija nad podacima može imati samo dva ishoda: ili uspjeh ili grešku. Nadalje, slovo „C“ predstavlja „Konzistentnost“ (engl. Consistency). Konzistentnost podrazumijeva da se nakon izvršene operacije nad podacima moraju sačuvati i njihovo stanje prije i poslije operacije. Slovo „I“ predstavlja „Izolaciju“ (engl. Isolation). Izolacija podrazumijeva da ako postoje dvije paralelne istovremene radnje nad podacima, one ne bi trebale biti vidljive jedna drugoj. Konačno, slovo „D“ predstavlja „Izdržljivost“ (engl. Durability). Izdržljivost podrazumijeva da nakon odrađene operacije nad bazom podataka načinjene promjene ostaju trajne.<sup>1</sup>

#### 3.1. Pojam baze podataka

Za razumijevanje pojma baze podatka ključno je razumjeti sam pojam podatka. Prema tome podatak je: „(...) poznata ili pretpostavljena činjenica na osnovi koje se oblikuje informacija. Sastoji se od skupa kvantitativnih parametara koji se mogu zapisati kao nizovi znakova ili nizovi brojeva.“<sup>2</sup> Također za podatak se može navesti i sljedeće: „Bez podataka bi bilo nemoguće poslovati. U poslovanju se evidentiraju podaci o nizu elemenata poslovanja, kao što su nazivi i adrese poslovnih partnera ili klijenata, stanja gotovih proizvoda i sirovina u skladištima, količina naručene i prodane robe, izdani i primljeni računi itd. U specifičnim je primjenama vođenje poslovnih podataka i zakonski uređeno. Primjerice zakonom je određen način vođenja matičnih podataka građana ili trgovačkih društava. U poslovanju susrećemo niz manje ili više unificiranih dokumenata na kojima su zapisani podaci o obavljenoj prodaji, odnosno kupnji (račun), primitku

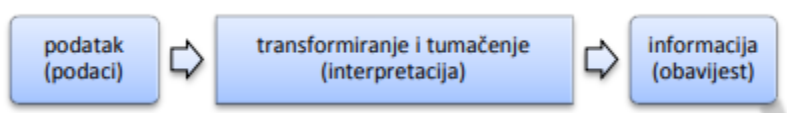
---

<sup>1</sup> Licul, M.: Modeliranje i izgradnja baze podataka za web aplikaciju, Završni rad, Veleučilište u Rijeci, Rijeka, 2022., <https://urn.nsk.hr/urn:nbn:hr:125:343314> [29.6.2023.]

<sup>2</sup> Podatak, enciklopedija.hr, <https://www.enciklopedija.hr/natuknica.aspx?id=48887> [26.6.2023.]

robe u skladište (primka), ulasku na predstavu (ulaznica), dosad položenim ispitima (indeks), potvrdi o završenoj školi (svjedodžba), popisu jela koji se nude u restoranu (jelovnik), itd. Lako je zaključiti da podaci obavještavaju (informiraju) o nekom konkretnom aspektu poslovanja.”<sup>3</sup>

Iz podatka se derivira informacija, a nju se može definirati i kao skup podataka s pripisanim značenjem. U biti informacija je osnovni element komunikacije koji, primljen u određenoj situaciji, povećava čovjekovo znanje.<sup>4</sup> Slika 1. prikazuje transformiranje, odnosno interpretaciju podataka kao informaciju



**Slika 1.** Transformiranje/interpretacija podataka u informacije.<sup>5</sup>

Poslovne aktivnosti prati dokumentacija koja sadrži podatke. U poslovanju vrijedi načelo po kojem se što podacima nije zabilježeno, to se nije ni dogodilo. Posljedično, rezultati se poslovnih događaja u suvremenim uvjetima poslovanja zapisuju zapisima (podataka) u nekoj digitalnoj bazi podataka. Tako svaki posao postaje ujedno i podatkovni (informacijski) posao.<sup>6</sup>

Dukić definira baze podataka na sljedeći način: „Baze podataka, kao deskriptivni model realnog svijeta predstavljaju podatkovni repozitorij u kojem se bilježe elementi strukture objekata realnog svijeta, kao i događaji koji utječu na promjene u strukturi tih objekata.“<sup>7</sup> Nadalje, prema Cariću i Buntiću, baze podataka se također može definirati kao zbirku zapisa pohranjenih u računalu na sustavan način, tako da joj se računalni program može obratiti prilikom odgovaranja na problem. Također, prema Careviću i Buntiću takva zbirka zapisa na računalu predstavlja bazu podataka jedino ako ima određena svojstva, primjerice, ako se podacima upravlja uz osiguranje referencijalnog i domenskog entiteta, ako je omogućen zaštićeni zajednički pristup podacima određenoj grupi korisnika, ako postoji jasna podatkovna shema, ako je podržan standardni upitni

<sup>3</sup> Podaci, element.hr, <https://element.hr/wp-content/uploads/2020/06/unutra-13639.pdf>, [26.6.2023.], str. 1.

<sup>4</sup> Informacija, enciklopedija.hr, <https://www.enciklopedija.hr/natuknica.aspx?id=27405>, [26.6.2023.]

<sup>5</sup> Podatci idem, str 3.

<sup>6</sup> Idem, str. 2

<sup>7</sup> Dukić, B.: Baze podataka i poslovni procesi, Ekonomski fakultet u Osijeku, Osijek, 2010, str. 2.

jezik, itd.<sup>8</sup> Sukladno prethodno navedenom, Carić i Buntić tvrde da dogovorena definicija prethodno navedenih svojstava ne postoji, već mogućnosti koje baze podataka pružaju su predmet tržišne utakmice u industriji programske opreme, te da upravo to osigurava stalan tehnološki napredak baza podataka.<sup>9</sup> Temeljem prethodnih navoda, može se zaključiti kako baze podataka predstavljaju analitički potencijal, te su temelj za racionalno odlučivanje. No, kada se govori o samom poslovanju, Dukić navodi kako baze podataka u kojima se bilježe događaji vezani za poslovanje i koje se zbog toga nazivaju proizvodne ili produkcijske baze podataka, često nisu dovoljan podatkovni izvor za taktičko i strateško odlučivanje.<sup>10</sup> Vezano za prethodni navod, Dukić dodaje sljedeće: „Da bi se mogla formirati produkcijska baza podataka i/ili skladište podataka, potrebno je izgraditi modele baza podataka koji se temelje na raspoloživoj teoriji vezanoj za modeliranje podataka.“<sup>11</sup>

Prema Dukiću, modeliranje je u širem smislu uspostava analogije između segmenata realnog svijeta-originala i simboličkog ili deskriptivnog opisa u kojem je taj segment realnog svijeta predstavljen.<sup>12</sup>

Modeliranje se baze podataka obavlja na tri razine:<sup>13</sup>

- konceptualnoj,
- logičkoj i
- fizičkoj.

Korake i poslovi pri oblikovanju baze podataka prikazuje slika 2.

---

<sup>8</sup> Carić, T., Buntić M.: Uvod u relacijske baze podataka, Sveučilište u Zagrebu, Zagreb, 2015., str.2.

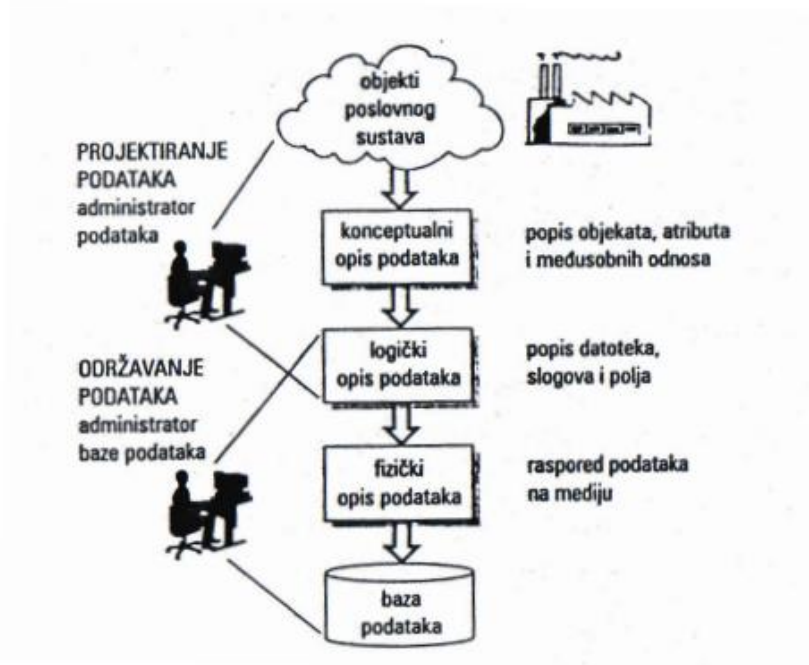
<sup>9</sup> Idem.

<sup>10</sup> Dukić, B.: idem.

<sup>11</sup> Idem.

<sup>12</sup> Idem., str. 35.

<sup>13</sup> Kramberger, T., Duk, S., Kovačević, R.: Baze podataka, Tehničko veleučilište u Zagrebu, Zagreb, 2018., str. 10.



**Slika 2.** Koraci i poslovi pri oblikovanju baze podataka.<sup>14</sup>

Kako je iz slike 2. vidljivo rezultat je prve faze oblikovanja baze podataka, odnosno konceptualnog modeliranja podataka, konceptualna shema cijele baze podataka, a takav je model sastavljen od entiteta, atributa i veza. Prema tome konceptualna shema baze podatka treba opisati sadržaj baze podataka i načine povezivanja podataka u njoj. Prikaz je u konceptualnoj shemi baze podataka jezgrovit, neformalan i pogodan ljudima za razumijevanje, no još je nedovoljno razrađen da bi omogućio izravnu implementaciju u digitalnom okruženju. Rezultat je druge faze oblikovanja baze podataka logička shema baze podataka koja je u slučaju relacijskog modela sastavljena od relacija, odnosno veza među podacima, a prikaz veza među podacima u relacijskom modelu podataka omogućuje tablica kao svojevrsna alternativa matematičkom prikazu podataka pomoću skupa. Sastavni je dio logičkog oblikovanja relacijske baze podataka i vertikalna normalizacija, gdje se primjenom posebnih pravila nastoji popraviti logička struktura samog modela podataka u težnji da se iz modela izbace anomalije među podacima koje producira nekontrolirana redundancija. U konačnici, neovisno od konceptualnog i logičkog modeliranja kojim se bavi osoba koja želi načiniti model realnog svijeta u vidu relacijske baze podataka, egzistira i fizički model koji je interpretiran u programski sustav za upravljanje bazom podataka i koji ima zadatak da logičku

<sup>14</sup> Vukmirović, S.: Modeliranje i analiza podataka u poslovanju, Ekonomski fakultet u Rijeci, Rijeka, 2013., str. 6.



shemu baze podataka koju mu je predao korisnik transformira u odgovarajući fizički zapis na digitalnom pohradbenom uređaju. Tom se prilikom podaci preoblikuju prema fizičkoj shemi baze podataka. U slučaju uporabe DBMS-a zasnovanog na jeziku SQL, pojam fizičke razine treba shvatiti kao nešto korisniku nevidljivo i nedodirljivo. Fizička shema je niz zapisa u vidu slogova i blokova podataka koji se uz pomoć SQL instrukcija interpretiraju na logičkoj razini u tablične prikaze. U fizički model ulaze i pridodane mogućnosti sustavu za upravljanje bazom podataka koji omogućava kreiranje pomoćnih struktura i koji sadrži mehanizme za ostvarenje traženih performansi baze podataka te čuvanje integriteta i sigurnosti podataka.<sup>15</sup>

### 3.2. Uobičajeno korišteni modeli baza podataka u poslovnoj praksi

Logički i konceptualni modeli baza podataka koji se najčešće koriste u poslovnoj praksi su:<sup>16</sup>

- hijerarhijski model,
- objektno-orijentirani model,
- mrežni model,
- metoda entiteti-veza (engl. Entity - relationship),
- relacijski model.

Hijerarhijski model jedan je od najstarijih modela u podatkovnom modeliranju koji je razvio veliki američki proizvođač računalne opreme IBM u pedesetim godinama prošlog stoljeća. U hijerarhijskom modelu podaci se promatraju kao skup segmenata koji tvore hijerarhijski odnos. Pri tome su podaci organizirani u strukturu poput stabla gdje se svaki zapis sastoji od jednog roditeljskog zapisa i mnogo djece. Čak i ako su segmenti povezani kao lančana struktura logičkim asocijacijama, tada trenutna struktura može biti lepezasta struktura s više grana. Nelogične asocijacije nazivamo usmjerenim asocijacijama.<sup>17</sup>

---

<sup>15</sup> Idem.

<sup>16</sup> What is a Database Model, Lucidchart, <https://www.lucidchart.com/pages/database-diagram/database-models> [20.6.2023.]

<sup>17</sup> Hierarchical Model in DBMS, geeksforgeeks, <https://www.geeksforgeeks.org/hierarchical-model-in-dbms/> [28.6.2023.]

Objektno-orijentirani model u DBMS-u ili OODM je podatkovni model u kojem se podaci pohranjuju u obliku objekata. Ovaj se model koristi za predstavljanje entiteta iz stvarnog svijeta. Podaci i odnos podataka pohranjuju se zajedno u jednom entitetu poznatom kao objekt u objektno-orijentiranom modelu. Objektno-orijentirani sustav upravljanja bazom podataka izgrađen je na objektno-orijentiranom modelu. Osnovna svojstva bez kojih se model ne može proglasiti objektno-orijentiranim su: apstrakcija, enkapsulacija, modularnost, nasljeđivanje i polimorfizam.<sup>18</sup>

Za mrežni model baze podataka Carić i Buntić navode sljedeće: „Mrežni model se pojavio krajem 60-tih godina. Prvi standard na području baza podataka nastao je na osnovu mrežnog model 1971. godine. Mrežni model je sličan hijerarhijskom modelu, samo što za razliku od hijerarhijskog, svako dijete može imati više roditelja.“<sup>19</sup>

Prema Pavliću se metodu entiteti-vezan (engl. Entity-relationship) može definirati na sljedeći način: „Metoda entiteti-veze (skraćeno EV) je grafički prikaz međusobno povezanih grupa podataka promatranoga sustava. EV je semantički bogata metoda za modeliranje podataka jer raspolaže ljudski bliskim konceptima“<sup>20</sup>

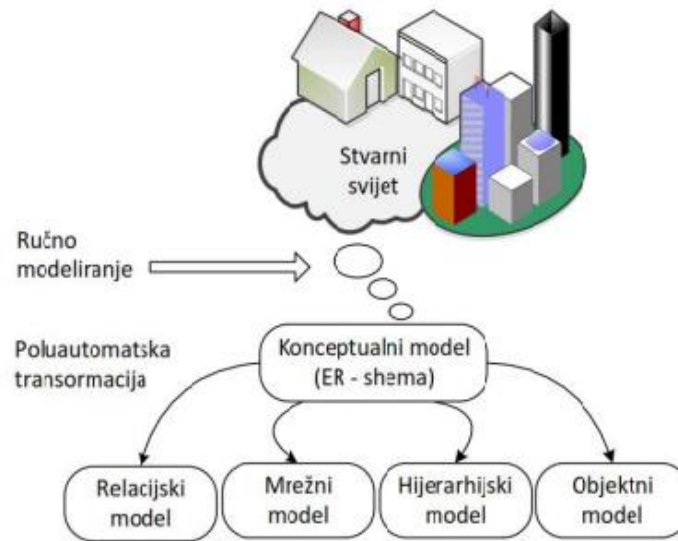
Odnos u modeliranju između konceptualnog modela i logičkih modela baze podataka prikazuje slika 3.

---

<sup>18</sup> What is Object Oriented Model in DBMS?, scaler, <https://www.scaler.com/topics/object-oriented-model-in-dbms/> [28.6.2023.]

<sup>19</sup> Carić, T., Buntić M.: Uvod u relacijske baze podataka, Zagreb, 2015., str.6

<sup>20</sup> Pavlić, M.: Oblikovanje baza podataka, Odjel za informatiku, Rijeka, 2011, str. 35.



**Slika 3.** Odnos u modeliranju između konceptualnog modela i logičkih modela baze podataka.<sup>21</sup>

Ipak, najzastupljeniji model u poslovnoj praksi je svakako relacijski model. Pavlić ukratko opisuje relacijski model na sljedeći način: „Relacijska baza podataka prikazuje pojmove koji čine strukturu relacijske metode i pojmove ograničenja u relacijskome modelu iz kojih slijedi pojam ključnog atributa.“<sup>22</sup>

Nadalje, prema Vukmiroviću, relacijske baze podataka jesu baze kod kojih su podaci smješteni u jednu tablicu ili više tablica koje su međusobno povezane. Takve baze mogu biti na jednom računalu za samo jednog korisnika ili na jednom od računala u računalnoj mreži.<sup>23</sup> Uz relacijski model podataka veže se i pojam RDBMS (engl. Relational database management system), odnosno relacijsku sustav za upravljanje bazama podataka. Prema Dukiću, RDMBS-i su razvijeni na temelju relacijskog modela podataka, te je uz njegovu pomoć moguće kreiranje, ažuriranje, pretraživanje i održavanje transakcijske baze podataka čija je struktura opisana relacijskim modelom podataka.<sup>24</sup>

<sup>21</sup> Carić, T., Buntić M.: Idem, str. 15.

<sup>22</sup> Pavlić, M.: Oblikovanje baza podataka, Odjel za informatiku, Rijeka, 2011, str. 7.

<sup>23</sup> Vukmirović, S.: Modeliranje i analiza podataka u poslovanju, Ekonomski fakultet u Rijeci, 2013. , str. 12.

<sup>24</sup> Dukić, B.: idem, str. 56.

### 3.3. Uloga relacijskog modela podataka u poslovanju

„Relacijski je model podataka najrasprostranjeniji logički podatkovni model i implementiran je gotovo u sve suvremene sustave za upravljanje bazama podataka. Zbog svojih temeljnih svojstava dinamičkog (logičkog) umjesto statičkog (fizičkog) ostvarivanja veza među entitetima i organizacijskog oblikovanja koje omogućava uklanjanje nekontroliranih redundancija i anomalija održavanja, relacijski model podataka možemo shvaćati temeljnim modelom za oblikovanje transakcijskih sustava, odnosno sustava dizajniranih i optimiziranih za ažuriranje podataka.“<sup>25</sup>

Nadalje, prema Dukiću, transakcijske baze podataka se koriste za neposredno prikupljanje i bilježenje podataka, te zato predstavljaju izvorište za kreiranje skladišta podataka. Također, prema Dukiću, skladišta podataka koja su optimizirana za pretraživanje, predstavljaju rudnike podataka u procesu proizvodnje informacija. Skladištima podataka (engl. Data Warehouse - DW) uobičajeno se koriste sustavi za izvješćivanje, sustavi za online analize podataka (engl. OnLine Analize Processing - OLAP), informacijski sustavi za rukovoditelje (EIS), sustavi za potporu „rudarenju“ informacija (engl. Data-Mining) i drugo. Iako relacijske baze podataka osiguravaju vrlo složena pretraživanja, one nisu za tu svrhu optimizirane jer izvršavanje složenih upita uzima značajno procesorsko vrijeme i smanjuje funkcionalnost samog sustava za upravljanje bazom podataka.<sup>26</sup>

---

<sup>25</sup> Idem

<sup>26</sup> Idem

## 4. Relacijski model podataka

Relacijski model podataka je podatkovni model za pohranjivanje i dohvaćanje podataka potrebnih za dobivanje informacija. To je najrašireniji model baze podataka na svijetu, s obuhvatom od više od 90% baza podataka koje se koriste u komercijalne svrhe. Izraz "relacijski" odnosi se na način na koji su podaci u tablicama povezani jedni s drugima putem zajedničkih naziva atributa (polja) odnosno ekvivalentnih domena vrijednosti atributa te povezanosti svih atributa u redovima s atributom koji se naziva primarni ključ i koji je jedinstven za svaku jedinku, odnosno n-torku koja je predstavljena redom u tablici u tabličnom prikazu relacije. Relacijske baze podataka se sastoje od relacija koje se pak sastoje od stupaca (ili atributa) i redaka (koji se nazivaju i n-torke ili zapisi). U modelu relacijske baze podataka, svaki dio podataka pohranjen je u retku (ili zapisu) i stupcima (ili poljima). Kako je već navedeno, svaki je stupac u retku povezan s drugim stupcima u retku putem jedinstvenih vrijednosti ključnog atributa, a s obzirom da ti atributi jedinstveno identificiraju skup vrijednosti koje su dodijeljene jedinki kroz red tablice, takve su vrijednosti identifikatori (ID). Ovo eliminira dupliciranje podataka. Suvremene baze podataka radi očuvanja integriteta imaju uvedenu automatsku brigu o konzistentnosti vrijednosti identifikacijskih atributa kroz sustav referencijalnog integriteta. Sustav za upravljanje bazom podataka u slučaju uključenosti referencijalnog integriteta obavlja automatski očuvanje integriteta, tako npr. ako se ID promijeni u bilo kojem trenutku (npr. zbog ažuriranja), veza će se ažurirati u skladu s tim.<sup>27</sup>

### 4.1. Nastanak i povijest relacijskog sustava za upravljanje relacijskim modelom

“Relacijski podatkovni model razvio je engleski matematičar Edgar Frank Codd-a. Prezentiran je 1970 u njegovom članku 'A Relational Model of Data for Large Shared Data Banks'. Edgar Frank Codd-a u to vrijeme radio je u IBM-u. Njegova očekivanja su bila da će oni prvi implementirati relacijski model spremanja podataka, što se nije dogodilo zbog ustrajnosti IBM-a na korištenju već

---

<sup>27</sup> Relational Database (Model, Operations & Constraints), databaseTown, <https://databasetown.com/relational-database-model-operations-constraints/> [29.6.2023].

provjerenog, ali zastarjelog hijerarhijskog modela u sustavu IMS. Prva tvrtka koji je implementirala relacijski model podataka bila je Oracle koja je i danas sinonim za baze podataka. Codd-ov relacijski upitni jezik Alpha nije prihvaćen već je prihvaćen nerelacijski jezik SEQUEL. Zanimljivo je da je prvotno ime SEQUEL preimenovano u SQL jer je jedna od zrakoplovnih kompanija u to vrijeme imala zaštićen naziv SEQUEL.”<sup>28</sup>

Nadalje prema Cariću Buntiću, Codd se dugi niz godina borio s raznim proizvođačima sustava za upravljanje bazama podataka - SUBP- (engl. Database management system - DBMS), jer su krivo ili djelomično implementirali relacijski podatkovni model, a nazivali su ga relacijskim. Rezultat njegove borbe je objava dvanaest pravila koja SUBP mora poštivati kako bi bio relacijski: <sup>29</sup>

1. Predstavljanje informacija (engl. The information rule) - podaci se reprezentiraju na jedinstven način kao vrijednosti u relacijama, tj. tablicama, jednostavno i dosljedno.
2. Pravilo pristupa (engl. The guaranteed access rule) - svaki podatak u tablici mora biti logički dostupan preko kombinacije imena tablica, vrijednosti primarnog ključa i imena atributa.
3. Tretiranje nepoznatih vrijednosti (engl. Systematic treatment of null values) - vrijednost NULL se tretira kao nepoznata vrijednost neovisno o tipu podatka. Nepoznata vrijednost nije isto što i prazan znak ili broj 0 (nula) ili varijabla.
4. Dinamički online katalog (engl. Active online catalog based on the relational model) - Rječnik baze podataka u kojem se nalaze informacije o samoj relacijskoj shemi tablica mora biti pohranjen kao i svi ostali podaci u bazi. Nad tim podacima autorizirani korisnici mogu postavljati upite koristeći (SQL) upitni jezik.
5. Pravilo sveobuhvatnog jezika (engl. The comprehensive data sublanguage rule) - mora postojati jezik za komunikaciju sa bazom podataka koji podržava relacijske operatore koji se odnose na modifikaciju podataka, definiciju podataka i administraciju.
6. Pravilo pogleda (engl. The view updating rule) - ovo pravilo definira takozvane Tablice pogleda (engl. View table) koje se sastoje od jedne SELECT naredbe koja dohvaća podatke iz jedne ili više tablica. Sve poglede sustav mora moći ažurirati.

---

<sup>28</sup> Carić, T., Buntić M.: idem, str.8.

<sup>29</sup> Idem.: str. 9 i 10.

7. Pravila ažuriranja skupova (engl. High-level insert, update, and delete) - ovo pravilo kaže da podaci iz relacijske baze podataka mogu biti preuzeti u skupovima podataka iz jedne ili više tablica. Ovo pravilo također zahtjeva da operacije umetanja, ažuriranja i brisanja moraju biti podržane za skupove podataka, a ne samo za jedan redak jedne tablice.
8. Nezavisnost fizičkih podataka (engl. Physical data independence) - aplikacije koje pristupaju podacima u relacijskoj bazi podataka ne smiju biti ovisne o promjenama u fizičkom načinu spremanja podataka.
9. Nezavisnost logičkih podataka (engl. Logical data independence) - logička nezavisnost znači da se odnosi između tablica mogu mijenjati, a da se istovremeno ne utječe na funkcije aplikacija koje se spajaju na tablice. Dakle promjena strukture baze podataka ne smije uzrokovati ponovnu izradu baze podataka ili aplikacije.
10. Nezavisnost integriteta podataka (engl. Integrity independence) - sustav za upravljanje bazama podataka mora se brinuti o integritetu baze podataka, a ne aplikacije izvana.
11. Distribuirana nezavisnost (engl. Distribution independence) - aplikacija mora nastaviti operativno raditi kada se uvede distribuirana verzija SUBP-a ili kada se distribuirana verzija centralizira.
12. Pravilo o nenarušavanju (engl. The nonsubversion rule) - integritet podataka ne smije biti narušen zaobilaznjem pravila integriteta i ograničenja.
13. Nulto pravilo (engl. Rule 0 ) - da bi sustav za upravljanje bazama podataka bio relacijski mora koristiti isključivo relacijske mogućnosti baze podataka kod upravljanja.

Prema Cariću i Buntiću, relacijski model je najsporiji, ali u isto vrijeme i najfleksibilniji što ga čini najzastupljenijim u današnje vrijeme.<sup>30</sup> Danas, također prema Cariću, najpopularnije implementacije sustava za upravljanje relacijskim bazama podataka su:<sup>31</sup>

- SQL Server,
- Oracle,
- Microsoft Access,
- PostgreSQL,
- MySQL i dr.

---

<sup>30</sup> Idem, str.10.

<sup>31</sup> Idem.

## 4.2. Temeljni pojmovi vezani uz relacijski model podataka

Temeljni pojmovi vezani uz relacijski model podataka su:<sup>32</sup>

- Tablica: relacijski model koristi format tablice za prikaz spremanih podataka i odnosa među njima. Svaka tablica ima dva svojstva: retke i stupce. Redovi predstavljaju zapise, dok stupci predstavljaju attribute.
- n-torka (engl. Tuple): jedan redak u tablici koji sadrži zapis.
- Kardinalnost: broj redaka (n-torki) tablice.
- Atribut: stupac u tablici naziva se atribut. Ovo su svojstva koja definiraju odnos.
- Stupanj: broj atributa u tablici.
- Shema relacije: shema relacije predstavlja naziv, attribute i kardinalnost jednog odnosa između dva ili više entiteta.
- Instanca relacije: instanca relacije je konačan skup n-torki u RDBMS sustavu. Instance relacije nikada nemaju duple n-torke.
- Ključ relacije: svaki red ima jedan, dva ili više atributa, što se naziva ključ relacije.
- Domena atributa: svaki atribut ima neku unaprijed definiranu vrijednost i opseg koji je poznat kao domena atributa.

### 4.2.1. Jedinka i relacija

Prema Dukiću, entitet (jedinka) je svaki element koji možemo jednoznačno odrediti i jednoznačno ga izdvojiti iz skupine. Jednoznačna određenost je bitna za određivanje entiteta, a to znači da skupine objekata koje nemaju svojstva koja ih međusobno razlikuju ne možemo podrazumijevati entitetima (npr. zrnca pijeska). Nadalje, prema Dukiću, svaki entitet posjeduje svojstva (attribute), a dva ili više entiteta mogu biti u različitim odnosima i onda ti odnosi također čine entitete, koji se nazivaju i izvedenim entitetima i koji imaju svoja svojstva.<sup>33</sup>

---

<sup>32</sup> Relational Database (Model, Operations & Constraints), idem.

<sup>33</sup> Dukić, B.: idem, str. 36



„Svojstva omogućavaju grupiranje entiteta prema zajedničkim svojstvima, tako npr. možemo izabrati sve kupce koji imaju rođendan isti dan i slično. U pravilu postoje svojstva ili skupine svojstava koje su jednoznačne za određeni entitet i omogućavaju njegovo jednoznačno prepoznavanje u skupini. Takva svojstva nazivaju se karakterističnim svojstvima, a vrijednost atributa karakterističnog svojstva je identifikator entiteta u skupini entiteta. Primjer karakterističnog svojstva je jedinstveni matični broj građana ili članski broj(broj s kartice) i slično.“<sup>34</sup>

„Podaci se u relacijskoj bazi podataka nalaze u relacijama, odnosno tablicama. Definicija relacije naziva se relacijskom shemom. Relacija se sastoji od njezina naziva i popisa atributa (njezinih obilježja) koji ulaze u sustav relacije. Relacija ili tablica sastoji se od zapisa prikazanih u redcima i atributa prikazanih u stupcima tablice.“<sup>35</sup>

Osnovne su značajke relacije da: <sup>36</sup>

- ne postoje dva jednaka retka,
- redoslijed redaka i stupaca nije bitan. Tablica (entitet) jest osnovni objekt u relacijskoj bazi podataka. Relacija se može sastojati od jedne ili više tablica.

Operacije nad relacijama tablica su temeljene na relacijskoj algebri. Prema Dukiću, relacijska algebra se definira kao standardni upitni jezik za relacijske baze podataka. Također, prema Dukiću, pojam relacija u relacijskom modelu podataka je ekvivalentna matematičkom pojmu skupa, te je relacijska algebra šira od algebre skupova.<sup>37</sup>

Razlikujemo upite, odnosno operacije koje su pogodne za ažuriranje baze podataka i operacije koje su pogodne za kreiranje izvješća iz baza podataka. Operacije pogodne za ažuriranje su :<sup>38</sup>

- unija i
- razlika (diferencija)

---

<sup>34</sup> Idem.

<sup>35</sup> Vukmirović, S.: idem , str. 12.

<sup>36</sup> Idem.

<sup>37</sup> Dukić, B.: idem, str. 63.

<sup>38</sup> Idem.: str. 64.

Operacije pogodne za izvješćivanje su:<sup>39</sup>

- presjek,
- projekcija,
- selekcija,
- Kartezijev umnožak,
- spajanje i
- dijeljenje

„U relacijskoj algebri operandi ili objekti operacije su relacije opisane operacijskom shemom. U relacijskoj algebri nisu propisana ograničenja glede izbora relacije nad kojima se operacije odvijaju. Zbog ove slobode mogu se pojaviti i problemi. Naime, ako relacije nisu usporedive (unijski kompatibilne), onda se binarne operacije unije, razlike i presjeka neće ispravno obaviti.“<sup>40</sup>

Unijska kompatibilnost – usporedivost je ostvarena:<sup>41</sup>

- ako dvije relacije imaju jednaki stupanj i
- ako se svaka domena jedne relacije preslikava na unijski kompatibilnu i semantički istoznačnu domenu druge relacije pri čemu, redosljed stupaca u relacijama, tj. redosljed jednostavnih domena, nije bitan.

#### 4.2.2. Atributi i domene vrijednosti

Prema Vukmiroviću, atribut je opis entiteta koji služi za određivanje, identificiranje, klasificiranje, kvantificiranje ili izražavanje stanja entiteta (tablice). U tablici je svaki atribut prikazan stupcem.<sup>42</sup>

Prema Dukiću, u jednom vremenskom trenutku pojedina pojava entiteta ima jednu vrijednost (engl. Instance) jednog tipa atributa. Također, prema Dukiću, attribute klasificiramo u :<sup>43</sup>

---

<sup>39</sup> Idem.

<sup>40</sup> Dukić, B.: idem, str. 64.

<sup>41</sup> Idem.

<sup>42</sup> Vukmirović, S.: idem, str. 1.

<sup>43</sup> Dukić, B.: Idem, str. 47.

- identifikacijske,
- opisne i
- izvedbene

„Identifikacijski atribut naziva se još i identifikator, jer se njime pojava jednoznačno i nedvosmisleno identificira. Riječ identitet predstavlja svojstvo trajno pridruženo entitetu, bez obzira na promjenu njegove strukture ili stanja. Primjer takvog svojstva je jedinstveni matični broj građana.(...) Značajno je da se identifikacijski atribut ne može mijenjati bez rušenja integriteta entiteta. Opisni atribut ili deskriptor opisuje kvalitativna i kvantitativna svojstva entiteta. Za razliku od identifikatora, njegove se vrijednosti mogu mijenjati sukladno promjeni stanja i svojstva entiteta. Izvedeni atribut nastaje djelovanjem aritmetičkih ili logičkih operacija kojima se djeluje na druge atribute.“<sup>44</sup>

Nadalje, prema Dukiću, vrijednosti atributa se izvode iz domene vrijednosti, te se njome utvrđuju ograničenja vrijednosti atributa. Domena je skup svih vrijednosti svojstava koje svojstvo (atribut) može poprimiti. Tako su, primjerice, kao domena za atribut „DOB GLASAČA“ vrijednosti između 18 i 100 godina.<sup>45</sup>

Primjeri domena atributa :

- Char – polje znakova fiksne dužine (npr. OIB),
- Varchar – polje znakova varijabilne dužine (npr. Ime i prezime),
- Int – cijeli broj,
- Numeric (Decimal) – decimalni broj,
- Date, Time, Datetime – pohrana vremenskih podataka

Nadalje, prema Cariću i Buntiću: „Pojedinačna vrijednost atributa se smatra najmanjom nedjeljivom semantičkom jedinicom podataka. Za svaki atribut definira se domena i predstavlja podatke koji pripadaju istom tipu podataka. Miješanje više tipova podataka unutar jedne domene nije dopušteno. Osim toga za atribute su obično definirana dodatna ograničenja primjerice entitet

---

<sup>44</sup> Dukić, B.: idem, str. 47

<sup>45</sup> Idem

(tablica) 'Račun' i njegov atribut 'DatumIzdavanja' daju nam informaciju kada je račun izdan. Pretpostavka je da atribut treba biti tipa datetime no pitanje je osigurava li se time realnost podataka, tj. ako je trenutno datum 2016-01-01, a u atribut je unesena vrijednost 2017-11-11 zaključuje se da je unesena vrijednost pogrešna. Dakle nije dovoljno definirati format podataka već treba definirati dodatna ograničenja koja opisuju specifičnost atributa. U ovom slučaju uneseni datum ne smije biti veći od trenutnog.<sup>46</sup>

### 4.2.3. Relacije unutar tablice i između tablica

Kreiranje se relacija u relacijskom modelu baza podataka odvija metodama relacijskog modeliranja baze podataka te postupkom vertikalne normalizacije kojom se eliminiraju nekontrolirane redundancije među podacima. No, postupak vertikalne normalizacije generira kontrolirane redundancije kojima se povezuju početne normalizirane relacije (tablice) s relacijama (tablicama) koje su nastale postupkom dekompozicije polazne relacije (tablice). Na taj se način uspostavljaju vanjske veze, odnosno veze između primarnih ključeva u relacijama (tablicama) nastalim dekompozicijom i vanjskih ključeva u početnoj tablici (relaciji). Prema tome, nadređene tablice nastale dekompozicijom i podređene tablice (polazne tablice) se spajaju preko atributa koji imaju istu vrijednost i tip podatka, no nazivi atributa u različitim tablicama ne moraju biti jednaki. Prema Dukiću, primarni ključ možemo promatrati sa stajališta njegovog karaktera. Tako statički karakter osigurava jedinstvenost relacije u strukturi baze podataka. Dinamički karakter, zbog pravilnog izvršavanja operacija nad relacijama pretpostavlja definiranje dodatnih ograničenja u svrhu očuvanja integriteta baze podataka.<sup>47</sup>

Nadalje, prema Dukiću, vanjski se ključ definira na sljedeći način: „Vanjski ključ je skup atributa u primarnoj relaciji koji nije ključ te relacije, ali je ključ neke druge relacije u modelu. Preko vanjskog ključa uspostavlja se dinamička veza između relacija gdje je relacija s primarnim ključem nadređena, a relacija s vanjskim ključem podređena. Uspostavljanje referencijalnog integriteta

---

<sup>46</sup> Carić, T., Buntić M.: idem, str. 31

<sup>47</sup> Dukić, B.: idem, str. 60

preko vanjskog ključa podrazumijeva kaskadno ažuriranje podataka na relaciji nadređena-podređena relacija.<sup>48</sup>

Dukić definira dva pravila vezana za integritet, a to su:<sup>49</sup>

- pravilo integriteta entiteta i
- pravilo referencijalnog integriteta

„Pravilo integriteta entiteta traži da vrijednost primarnog ključa relacije ne bude nepoznata. To znači da vrijednost primarnog ključa relacije ne smije biti null-vrijednost. Ako se radi o primarnom ključu složenom od više atributa, vrijednost niti jednog atributa ne smije biti nepoznata i mora postojati“<sup>50</sup>

Referencijalni integritet se prema Cariću i Butiću može definirati kao sljedeće: „Referencijalni integritet osigurava logičnu vezu i pravila odnosa među podacima u tablicama koje su relacijski povezane. U tablici ne može postojati vrijednost stranog ključa za koju ne postoji ista vrijednost primarnog ključa u osnovnoj tablici. Referencijalni integritet definira pravila unosa, brisanja i mijenjanja (ažuriranja) podataka s ciljem osiguravanja konzistentnosti podataka u bazi.“<sup>51</sup>

Svaki je entitet u relaciji (vezi) s drugim entitetima iz svog okruženja, stoga se može reći da je relacija nešto što povezuje dva ili više entiteta u određenom okruženju.<sup>52</sup>

Kod veze 1:1, svaki element skupa R može biti povezan samo s jednim ili nijednim elementom skupa S, te vrijedi i obratno za skup S.<sup>53</sup> Prikaz veze jedan prema jedan dan je slikom 4.

---

<sup>48</sup> Idem.

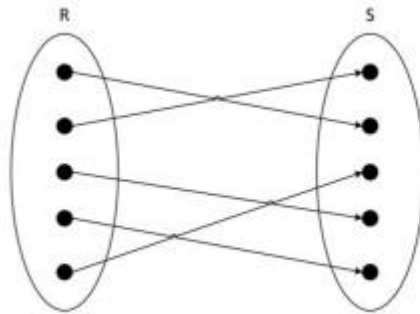
<sup>49</sup> Idem.

<sup>50</sup> Idem, str. 60.

<sup>51</sup> Carić, T., Buntić M.: idem, str.10.

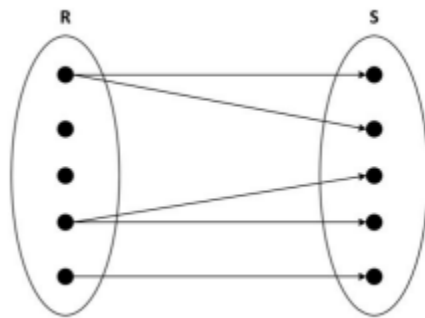
<sup>52</sup> Idem, str. 16.

<sup>53</sup> Idem, str. 17.



**Slika 4.** Prikaz veze jedan prema jedan.<sup>54</sup>

Veza 1:N nam govori da svaki element skupa R može biti povezan sa jednim ili više elemenata skupa S, dok svaki element skupa S može biti povezan sa samo jednim elementom skupa R.<sup>55</sup> Prikaz veze jedan prema više dan je slikom 5.



**Slika 5.** Prikaz veze jedan prema više.<sup>56</sup>

Veza M:N je veza gdje svaki element skupa R može biti povezan sa jednim ili više elemenata skupa S, te vrijedi i obrnuto za skup S.<sup>57</sup> Prikaz veze više prema više dan je slikom 6.

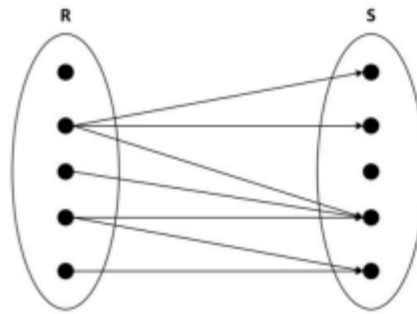
---

<sup>54</sup> Idem.

<sup>55</sup> Idem, str.17.

<sup>56</sup> Idem

<sup>57</sup> Idem



Slika 6. Prikaz veze više prema više.<sup>58</sup>

#### 4.2.4. Vertikalna normalizacija u funkciji oblikovanja i eliminacija nekontrolirane redundancije

Prema Dukiću, postoji horizontalna i vertikalna normalizacija. Horizontalna normalizacija rješava problem distribucije podataka na više nositelja podataka, dok je vertikalna normalizacija, u pragmatičkom smislu važnija, jer rješava organizacijske probleme koji se mogu pojaviti u vidu:<sup>59</sup>

- nekontrolirane redundancije,
- anomalija održavanja podataka,
- varijabilne dužine n-torke u relaciji.

„Redundantni podaci znače da se isti podaci pohranjuju na više od jednog mjesta unutar baze podataka. Ako su podaci redundantni, iste je moguće ukloniti bez gubitka informacije. Redundancija može dovesti do anomalije podataka koja se može dogoditi prilikom unosa, promjene i brisanja podataka. Redundancija može uzrokovati probleme prilikom normalnog rada baze podataka. Na primjer, prilikom unosa podatka u bazu podataka, podatak se mora duplicirati bez obzira postoji li isti u bazi podataka ili ne. Također, prilikom promjene podatka, isti je potrebno promijeniti na više mjesta kako bi se pohranile željene promjene“<sup>60</sup>

<sup>58</sup> Idem, str.18.

<sup>59</sup> Dukić, B.: idem., str. 69.

<sup>60</sup> Kramberger, T., Duk, S., Kovačević, R.: idem, str. 20.

Nadalje, Dukić navodi da se redundanciju postupkom vertikalne normalizacije teži svesti na minimum kako bi se izbjegla anomalija održavanja podataka. Anomalija održavanja podataka podrazumijeva tri moguća problema:<sup>61</sup>

- inkonzistencija,
- anomalija dodavanja i
- anomalija brisanja podataka.

„Inkonzistencija je posljedica nekontrolirane redundancije, a nastaje kada u bazi podataka memoriramo isti podatak u više kopija i nakon ažuriranja neka od kopija ostane neažurirana.(...) Anomalija dodavanja, ili kako se još naziva anomalija upisa, pojavljuje se kada podatke o nekom entitetu nije moguće dodati zbog nepostojanja opisa drugog entiteta, a to se događa u slučajevima kada su podaci o svojstvima jednog entiteta modelirani kao dio opisa nekog drugog entiteta.“<sup>62</sup>

„Anomalija brisanja je inverza anomalije dodavanja(...). Problem je istovjetan kao i kod anomalije dodavanja; kada su uz svojstva jednog entiteta modelirana svojstva drugog entiteta, pa brisanjem jednog entiteta gubimo informacije i o drugom entitetu.“<sup>63</sup>

Dakle, vertikalna normalizacija predstavlja tehniku koja iz relacijske sheme izdvaja attribute koji stoje u nedozvoljenim odnosima s ostalim atributima dane sheme, te se izdvajanjem oblikuju nova ili nove relacijske sheme koje sadrže izdvojene attribute. Bitno je da prilikom izdvajanja ne dođe do gubitka informacija. Dukić razlikuje dvije tehnike vertikalne normalizacije .<sup>64</sup>

- normalizacija dekompozicijom i
- normalizacija sintezom

„(...) normalizacijom sintezom konstruira se relacijska shema ili skup relacijskih shema izravno u traženoj normalnoj formi; to znači da se ova vrsta vertikalne normalizacije ne provodi u koracima niti se polazi od relacijske sheme. Kod normalizacije dekompozicijom slučaj je upravo suprotan.

---

<sup>61</sup> Dukić, B.: Idem.

<sup>62</sup> Idem.

<sup>63</sup> Idem, str. 70.

<sup>64</sup> Idem.



Postupak normalizacije provodi se na nenormaliziranoj relacijskoj shemi i izvodi se u koracima. Svakim korakom relacijska shema se provodi u višu normalnu formu.<sup>65</sup>

#### 4.2.5.SQL

Strukturirani upitni jezik (engl. Structured Query Language - SQL) je programski jezik četvrte generacije za pohranu i pristup podacima u relacijskoj bazi podataka. Relacijska baza podataka pohranjuje podatke u tabličnom obliku, s redovima i stupcima koji predstavljaju različite attribute podataka i različite odnose između vrijednosti podataka. SQL se sastoji od naredbi za pohranjivanje, ažuriranje, uklanjanje, pretraživanje i dohvaćanje podataka iz baze podataka. Također, SQL se koristi za održavanje i optimiziranje performansi baze podataka.<sup>66</sup>

SQL je prvi razvio ranih 1970-ih u IBM-u tim koji su vodili Donald D. Chamberlin i Raymond F. Boyce. Izvorni naziv jezika bio je SEQUEL, što je skraćenica za "Structured English Query Language". SEQUEL je dizajniran da bude jednostavan i intuitivan način za pristup i manipuliranje podacima pohranjenim u IBM-ovom sustavu upravljanja relacijskom bazom podataka System R. SEQUEL je kasnije preimenovan u SQL, a IBM je 1981. izdao prvu komercijalnu implementaciju jezika kao dio svog System R sustava za upravljanje relacijskim bazama podataka (RDBMS). SQL je brzo postao standardni jezik za interakciju s relacijskim bazama podataka, a ubrzo su ga prihvatili i drugi proizvođači RDBMS-a kao što su Oracle, Sybase i Microsoft. Američki nacionalni institut za standarde (ANSI) i Međunarodna organizacija za standarde (ISO) objavili su službene SQL standarde 1986. odnosno 1987. godine.<sup>67</sup>

SQL-om je moguće izvršavati sljedeće operacije:<sup>68</sup>

- izvršavanje upita prema bazi podataka,
- dohvaćanje podataka iz baze podataka,
- umetanje zapisa u bazu podataka,

---

<sup>65</sup> Idem

<sup>66</sup> What Is SQL (Structured Query Language)?, aws Amazon, <https://aws.amazon.com/what-is/sql/> [28.6.2023.]

<sup>67</sup> SQL history, w3schools, <https://www.w3schools.in/sql/history> [28.6.2023.]

<sup>68</sup> SQL intro, w3schools, [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp) [28.6.2023.]

- ažuriranje zapisa u bazi podataka,
- brisanje zapisa u bazi podataka,
- kreiranje nove baze podataka,
- kreiranje novih tablica u bazi podataka,
- pohranjivanje procedura u bazi podataka,
- stvaranje novih prikaza u bazi podataka i
- postavljanje dopuštenja za tablice, procedure i prikaze.

SQL naredbe podijeljene su u nekoliko različitih vrsta, uključujući sljedeće: <sup>69</sup>

- Naredbe jezika za definiranje podataka (engl. Data Definition Language - DDL) nazivaju se i naredbama za definiranje podataka jer se koriste za definiranje podatkovnih tablica.
- Naredbe jezika za manipulaciju s podacima (engl. Data Manipulation Language - DML) koriste se za manipulaciju podacima u postojećim tablicama dodavanjem, mijenjanjem ili uklanjanjem podataka. Za razliku od DDL naredbi koje definiraju kako se podaci pohranjuju, DML naredbe djeluju u tablicama definiranim DDL naredbama.
- Naredbe za upite (engl. Data Query Language – DQL) sastoji se od samo jedne naredbe, SELECT, koja se koristi za dobivanje određenih podataka iz tablica. Ova se naredba ponekad grupira s DML naredbama.
- Naredbe kontrolnog jezika (engl. Data Control Language – DCL) koriste se za dodjelu ili opoziv korisničkih povlastica pristupa.
- Naredbe za kontrolu transakcija (engl. Transaction Control Language – TCL) naredbe se koriste za promjenu stanja nekih podataka -- na primjer, za COMMIT transakcijske promjene ili za ROLLBACK transakcijske promjene.

Temeljna pravila sintakse SQL-a su, prema Dukiću, sljedeća: <sup>70</sup>

- svaki sustav za upravljanje bazom podataka dolazi s alatom koji korisniku dopušta unošenje i izvršavanje SQL naredbi (query = upit),

<sup>69</sup> Structured Query Language (SQL), techtarget, <https://www.techtarget.com/searchdatamanagement/definition/SQL> [28.6.2023.]

<sup>70</sup> Dukić, B.: idem, str. 74

- svaka naredba završava interpunkcijskim znakom točka-zarez (;), a ukoliko se instrukcija prostire kroz više redova, znak ';' stavlja se na kraj posljednjeg reda,
- kada se SQL naredba uključuje u drugi programski jezik (Embedded SQL), ona završava na način svojstven tom jeziku,
- iako je SQL no-case-sensitive, odnosno ne razlikuje mala i velika slova, uobičajeno je da se ključne riječi pišu velikim slovima.

## **5. Računalni servis**

Za svaki tehnički proizvod postoji mogućnost nastanka kvara. Sukladno iskustvu može se reći kako organizacije koje pružaju usluge popravaka tehničkih proizvoda na našem se govornom području uobičajeno nazivaju servisima (engl. Service = hrv. Usluga). Prema tome u biti računalni se servisi primarno bave popravljanjem računala i računalne opreme. S obzirom da računalnu opremu čine strojni, programski, a u suvremeno doba i mrežni dio, računalni se servisi bave prije svega održavanjem, instaliranjem, popravkom i nadogradnjom strojnog djela računalne opreme, programskog dijela računalne opreme i mrežnom opremom. S obzirom da su računalni sustavi hibridni sustavi i oprema dolazi od različitih uglavnom svjetskih proizvođača, iskustvo pokazuje kako su računalni servisi na području Republike Hrvatske uglavnom mali ili srednje veliki poslovni subjekti koji često koriste programska rješenja razvijena od strane poslovnih subjekata s područja Republike Hrvatske. Prema tome, objektivno gledano, postoji permanentna potreba za razvitkom programskih sustava za računalne servise kojima je u podlozi relacijska baza podataka.

Kako bi računalni servisi učinkovito upravljali svojim operacijama, pratili zalihe rezervnih dijelova i opreme, upravljali potencijalnim i stvarnim klijentima, uspješno pružali usluge i drugo, potrebno je u podlozi programskih aplikacija za računalne servise stoje optimalno dizajnirane baze podataka. Upravo je fokus provedenog istraživanja bio na definiranju optimalnog modela baze podataka za računalne servise koji je proizvod praćenja procesa koji su prisutni u računalnim servisima u Republici Hrvatskoj.

### **5.1. Računalni servisi na području istočnog dijela Republike Hrvatske**

Radi izrade optimalnog modela relacijske baze podataka za potrebe praćenja procesa koji se odvijaju u računalnim servisima u Republici Hrvatskoj provedeno je primarno istraživanje kroz posjete u pet različitih računalnih servisa u Gradu Osijeku. Analizirani su iz navedenih servisa preuzeti radni nalozi koji se izdaje klijentima kada donose računalu opremu na servis ili ga pak preuzimaju nakon gotovog servisa. Također, u okviru je primarnih istraživanja obavljen kratak intervju sa zaposlenicima servisa kako bi se dobio uvid u vrste usluge koje servisi nude i procese

koji se odvijaju u tim servisima. Računalni servisi koji su bili posjećeni u sklopu ovog primarnog istraživanja su: Centar tehnike, Melior Computer Shop, HGSpot, Vacom računala, te Restart. Nakon kratkog intervjua sa svakim od zaposlenika pojedinog servisa, usluge koje svi nude su:

- zamjena ili popravak hardverskih komponenti,
- ažuriranje programske opreme (operacijski sustavi, antivirusni programi),
- prodaja strojnih komponenti i programskih licenci,
- sklapanje računala na zahtjev klijenta,
- detekcija i rješavanje zlonamjernih programa (virusi, crvi) na računalu te
- konzultacije i pomoć klijentima.

## **5.2. Dokumentacija koja se koristi u računalnim servisima kao temelj za modeliranje baze podataka**

Kako bi se načinio optimalan model baze podataka namijenjen računalnom servisu, potrebno je detaljno analizirati dokumentaciju koja sudjeluje u procesu predaje računala na servis od strane klijenta, te povratak servisiranog računala klijentu od strane servisa. Dokumentacija se može sastojati od sljedećih dokumenata: radni nalog, potvrda o zaprimanju, izvještaj o dijagnostici, ponuda za popravak, potvrda o popravku, račun, te potvrda o preuzimanju popravljene robe. Dokument koji je primarno poslužio kao temelj za modeliranje optimalne baze podataka za računalni servis je radni nalog.

Radni nalog je osnovni dokument koji se koristi za praćenje procesa unutar servisa. On sadrži podatke o klijentu, opis problema, datum zaprimanja, detalje o izvršenim radnjama, predmet servisa, datum povrata opreme i drugo. Nakon prikupljenih radnih naloga iz ranije spomenutih servisa, bilo je potrebno svaki nalog detaljno sagledati, te ustanoviti koji su zajednički nazivnici, odnosno koje komponente su jednake u svim nalogima. Nakon izdvajanja zajedničkih nazivnika iz svih pet radnih naloga, bilo je potrebno kreirati jedinstvene tablice koje su kasnije poslužile kao temelj za izradu optimalnog modela baze podataka. Navedene su tablice prikazane na slici 7.

Poslovni subjekt					
OIB	Naziv	Adresa	Telefon	Email	Vlasnik

Ziro-racun		
Id	Poslovni subjekt	Broj

Klijent				
OIB	Prezime i ime	Adresa	Kontakt telefon	Email

Serviser	
Šifra	Prezime i ime

Nalog									
Id	Oib poslovnog subjekta	Predmet servisa	Klijent	Jamstvo	Vrsta naloga	Status naloga	Datum zaprimanja	Datum isporuke	Napomena

Predmet servisa				
Id	Klijent	Serijski broj	Opis kvara	Serviser

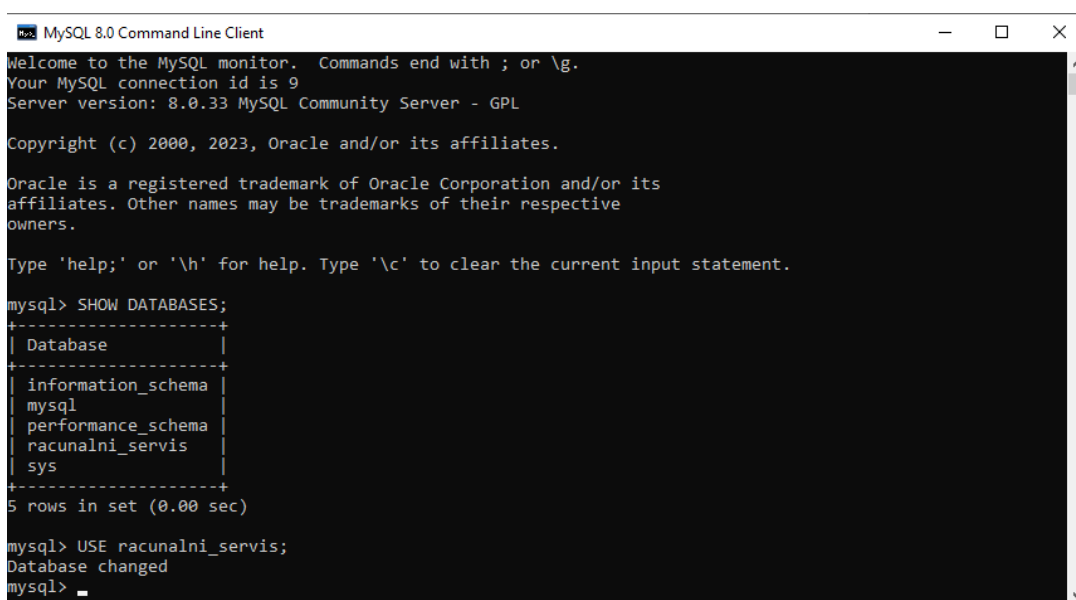
Ugrađena komponenta					
Id	Predmet servisa	Naziv usluge	Jedinica mjere	Količina	Cijena

**Slika 7.** Tablice s elementima koji se pojavljuju u radnim nalogima istraženih računalnih servisa.

Nadalje, svaka od tablica predstavlja entitet u relacijskoj bazi podataka, te će se na temelju ovog koncepta prvo napraviti optimalni model baze podataka za računalne servise u pseudo kodu, a potom u SQL upitnom jeziku.

## 6. Model baze podataka za računalni servis

Za izradu optimalnog modela baze podataka za računalni servis korišten je besplatni programski alat pod licencom Oracle-a, MySQL server 8.0.33. Radi zakona o zaštiti osobnih podataka, testiranje funkcionalnosti izrađene baze podataka provedeno je nad izmišljenim podacima. MySQL server je sustav za upravljanje relacijskim bazama podataka s mogućnostima postavljanja upita i povezivanja, kao i mogućnost integracije s mnogo različitih programskih platformi. MySQL može pouzdano i brzo rukovati velikim bazama podataka u visoko zahtjevnim proizvodnim okruženjima. Izgled sučelja aplikacije MySQL server prikazuje slika 8.



```
MySQL 8.0 Command Line Client
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| racunalni_servis |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> USE racunalni_servis;
Database changed
mysql>
```

Slika 8. Izgled sučelja aplikacije MySQL server.

### 6.1. Model baze podataka za računalni servis u pseudo kodu

Kako bi se pristupilo izradi modela baze podataka pomoću MySQL servera, izabran je put po kojem se prvo izrađuje logički model baze podataka u pseudo koda na temelju kojega će se pisati SQL kod. U pseudo kodu definiraju se entiteti (tablice), njihova svojstva (atributi), vrijednosti svakog atributa, primarni ključevi, strani ključevi te u konačnici veze između tablica. Pseudo kod često koristi strukturalne konvencije normalnog programskog jezika, ali je namijenjen ljudskom

čitanju, a ne strojnom. Pseudo kod za model baze podataka za računalni servis prikazan je na slici 9.

Iz pseudo koda vidljivo je da se model sastoji od osam tablica:

- poslovni\_subjekt,
- ziro\_racun,
- klijent,
- serviser,
- nalog,
- status\_naloga,
- predmetservisa, te
- ugradena\_komponenta.

Također, tablice su povezane na sljedeći način:

- poslovni\_subjekt i ziro\_racun (jedan poslovni subjekt može imati više žiro-računa, odnosno, više žiro-računa može pripadati jednom poslovnom subjektu),
- klijent i predmet\_servisa (jedan klijent može donijeti više predmeta na servis, odnosno, više predmeta servisa može biti doneseno od strane jednog klijenta),
- serviser i predmet\_servisa (jedan serviser može raditi na više predmeta servisa, odnosno, više predmeta servisa može biti servisirano od strane jednog serviser),
- predmet\_servisa i ugradena\_komponenta (jedan predmet servisa može imati više ugrađenih komponenti, odnosno, više ugrađenih komponenti može biti ugrađeno u jedan predmet servisa)



```

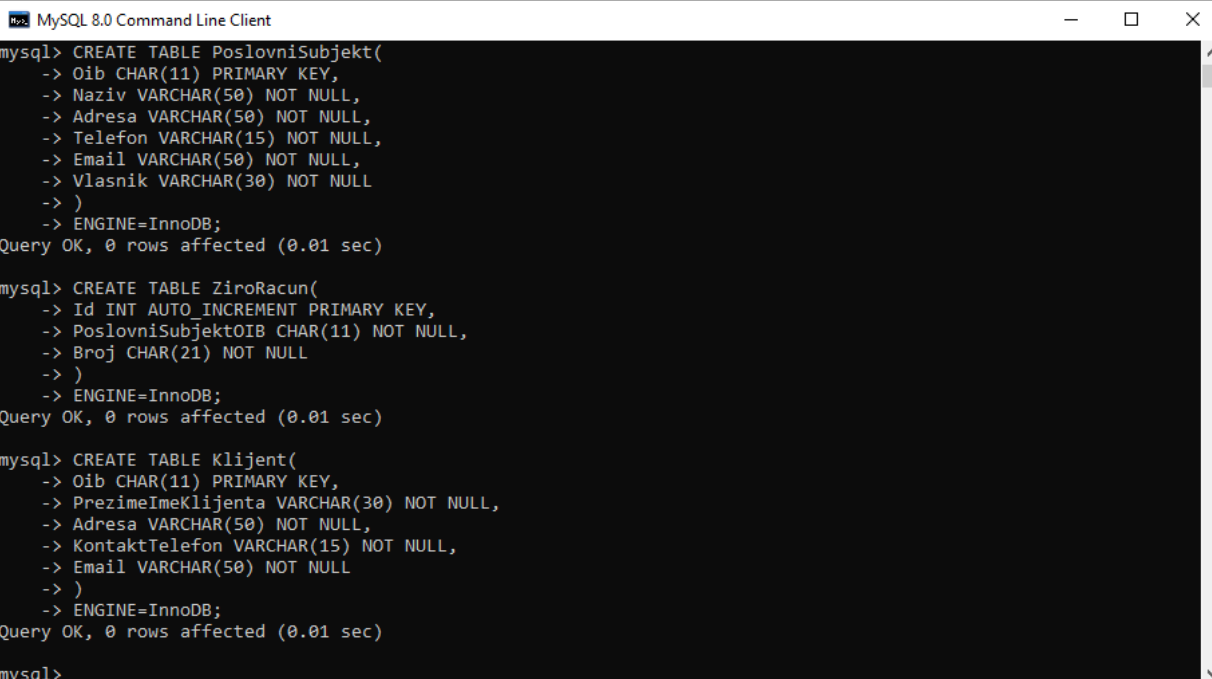
1  POSLOVNI_SUBJEKT
2  (
3      Oib : CHAR(11) PK,
4      Naziv : VARCHAR(50),
5      Adresa : VARCHAR(50),
6      Telefon : VARCHAR(15),
7      Email : VARCHAR(50),
8      Vlasnik : VARCHAR(30)
9  );
10
11
12  ŽIRO_RAČUN
13  (
14      Id : INT AK PK,
15      POSLOVNI_SUBJEKT.Oib : CHAR(11),
16      Broj : CHAR(21)
17  );
18
19
20  KLIJENT
21  (
22      Oib: CHAR(11) PK,
23      Prezime_i_ime_klijenta : VARCHAR(30),
24      Adresa : VARCHAR(50),
25      Kontakt_telefon : VARCHAR(15),
26      Email : VARCHAR(50)
27  );
28
29
30  SERVISER
31  (
32      Sifra: CHAR(5) PK,
33      Prezime_i_ime_servisera : VARCHAR(30)
34  );
35
36
37  NALOG
38  (
39      Id : INT AK PK,
40      POSLOVNI_SUBJEKT.Oib_poslovnog_subjekta : CHAR(11),
41      PREDMET_SERVISA.Predmet_servisa : INT,
42      KLIJENT.Prezime_i_ime : CHAR(11),
43      Jamstvo : CHAR(2),
44      Vrsta_naloga : CHAR(1),
45      STATUS_NALOGA.Status_naloga : CHAR(2),
46      Datum_zaprimanja: DATE,
47      Datum_isporuke : DATE,
48      Napomena : TEXT
49  );
50
51  STATUS_NALOGA
52  (
53      Sifra: CHAR(2) PK,
54      Status: VARCHAR(15)
55  );
56
57  PREDMET_SERVISA
58  (
59      Id : INT AK PK,
60      KLIJENT.PrezimeImeKlijenta: CHAR(11),
61      Serijski_broj : CHAR(13),
62      Opis_Kvara : TEXT,
63      SERVISER.PrezimeImeServisera CHAR(5)
64  );
65
66
67  Ugrađena_komponenta
68  (
69      Id: INT AK PK,
70      PREDMET_SERVISA.Predmet_Servisa: INT
71      Naziv_usluge : VARCHAR(50),
72      Jedinica_mjere : CHAR(3),
73      Količina : INT,
74      Cijena : DECIMAL(10,2)
75  );|

```

Slika 9. Model baze podataka u pseudo kodu.

## 6.2. Model baze podataka računalnog servisa u SQL-u

Kreiranje tablica u MySQL serveru izvodi se na način upisivanja SQL instrukcija u sučelju SQL server programske aplikacije. Na slici 10 vidljivo je kreiranje tablica PoslovniSubjekt, ZiroRacun i Klijent.



```
mysql> CREATE TABLE PoslovniSubjekt(  
-> Oib CHAR(11) PRIMARY KEY,  
-> Naziv VARCHAR(50) NOT NULL,  
-> Adresa VARCHAR(50) NOT NULL,  
-> Telefon VARCHAR(15) NOT NULL,  
-> Email VARCHAR(50) NOT NULL,  
-> Vlasnik VARCHAR(30) NOT NULL  
-> )  
-> ENGINE=InnoDB;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> CREATE TABLE ZiroRacun(  
-> Id INT AUTO_INCREMENT PRIMARY KEY,  
-> PoslovniSubjektOIB CHAR(11) NOT NULL,  
-> Broj CHAR(21) NOT NULL  
-> )  
-> ENGINE=InnoDB;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> CREATE TABLE Klijent(  
-> Oib CHAR(11) PRIMARY KEY,  
-> PrezimeImeKlijenta VARCHAR(30) NOT NULL,  
-> Adresa VARCHAR(50) NOT NULL,  
-> KontaktTelefon VARCHAR(15) NOT NULL,  
-> Email VARCHAR(50) NOT NULL  
-> )  
-> ENGINE=InnoDB;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>
```

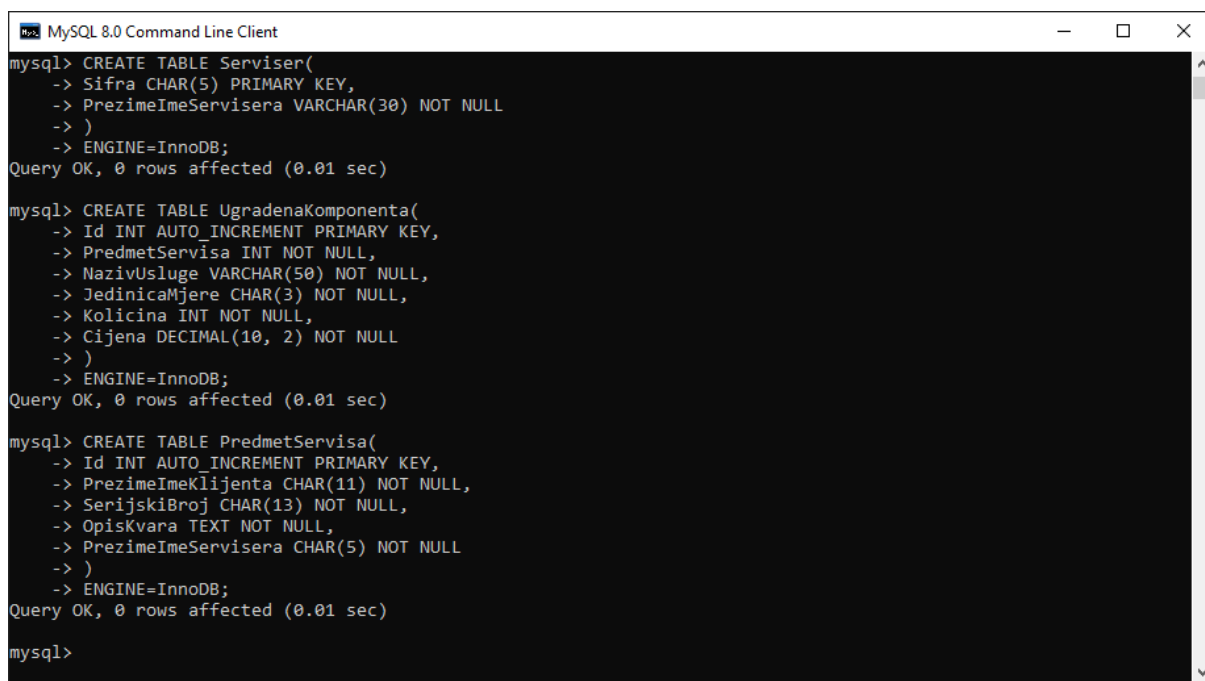
Slika 10. Kreiranje tablica PoslovniSubjekt, ZiroRacun i Klijent.

U kreiranoj tablici PoslovniSubjekt, atribut “Oib” predstavlja primarni ključ te tablice, te zahtjeva jedanaest znakova pri unosu. Atribut “Naziv” predstavlja naziv unesenog poslovnog subjekta, te je ograničen na varijabilnih pedeset znakova, odnosno, moguće je unijeti do maksimalno pedeset znakova. Atribut “Adresa”, kao i “Naziv”, ograničen je do pedeset znakova, te se u to polje unosi adresa poslovnog subjekta. Atribut “Telefon” je ograničen do maksimalno petnaest znakova, te se ovdje unosi kontakt telefon poslovnog subjekta. Atribut “Email” ograničen je do pedeset znakova, te se ovdje unosi kontakt email poslovnog subjekta. Konačno, atribut “Vlasnik” ograničen je do trideset znakova, te predstavlja prezime i ime vlasnika poslovnog subjekta. Sve vrijednosti atributa koje su označene sa “NOT NULL” označavaju da polje ne smije biti prazno prilikom unosa novog

zapisa. Sve vrijednosti atributa koje su označene kao “VARCHAR” ili “CHAR” pripadaju tekstualnom tipu podataka (stringu).

U kreiranoj tablici ZiroRacun, atribut “Id” predstavlja primarni ključ, te je on tip podatka “INT(integer)”, odnosno cijeli broj. Svojstvo “AUTO\_INCREMENT” označava automatsko generiranje novog Id-a prilikom svakog unosa nove n-torke, te prilikom unosa n-torki u tablicu, nije potrebno stavljati taj atribut kao parametar. Atribut “PoslovniSubjektOIB” predstavlja strani ključ, te njegova vrijednost i tip podatka moraju biti jednaki primarnom ključu tablice s kojom se povezuje, u ovom slučaju, to je tablica PoslovniSubjekt. Atribut “Broj” označava broj žiro-računa, te on mora imati dvadeset i jedan znak.

U kreiranoj tablici Klijent, atribut “Oib” je primarni ključ, te mora imati jedanaest znakova. Atribut “PrezimeImeKlijenta” označava polje gdje se upisuje prezime i ime klijenta, te je ograničen do trideset znakova. Atribut “Adresa” ograničen je do pedeset znakova. Atribut “KontaktTelefon” ograničen je do petnaest znakova. Atribut “Email” ograničen je do pedeset znakova. Na slici 11. vidljive su instrukcije za kreiranje tablica Serviser, UgradenaKomponenta i PredmetServisa.



```
MySQL 8.0 Command Line Client
mysql> CREATE TABLE Serviser(
  -> Sifra CHAR(5) PRIMARY KEY,
  -> PrezimeImeServisera VARCHAR(30) NOT NULL
  -> )
  -> ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE UgradenaKomponenta(
  -> Id INT AUTO_INCREMENT PRIMARY KEY,
  -> PredmetServisa INT NOT NULL,
  -> NazivUsluge VARCHAR(50) NOT NULL,
  -> JedinicaMjere CHAR(3) NOT NULL,
  -> Kolicina INT NOT NULL,
  -> Cijena DECIMAL(10, 2) NOT NULL
  -> )
  -> ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE PredmetServisa(
  -> Id INT AUTO_INCREMENT PRIMARY KEY,
  -> PrezimeImeKlijenta CHAR(11) NOT NULL,
  -> SerijskiBroj CHAR(13) NOT NULL,
  -> OpisKvara TEXT NOT NULL,
  -> PrezimeImeServisera CHAR(5) NOT NULL
  -> )
  -> ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

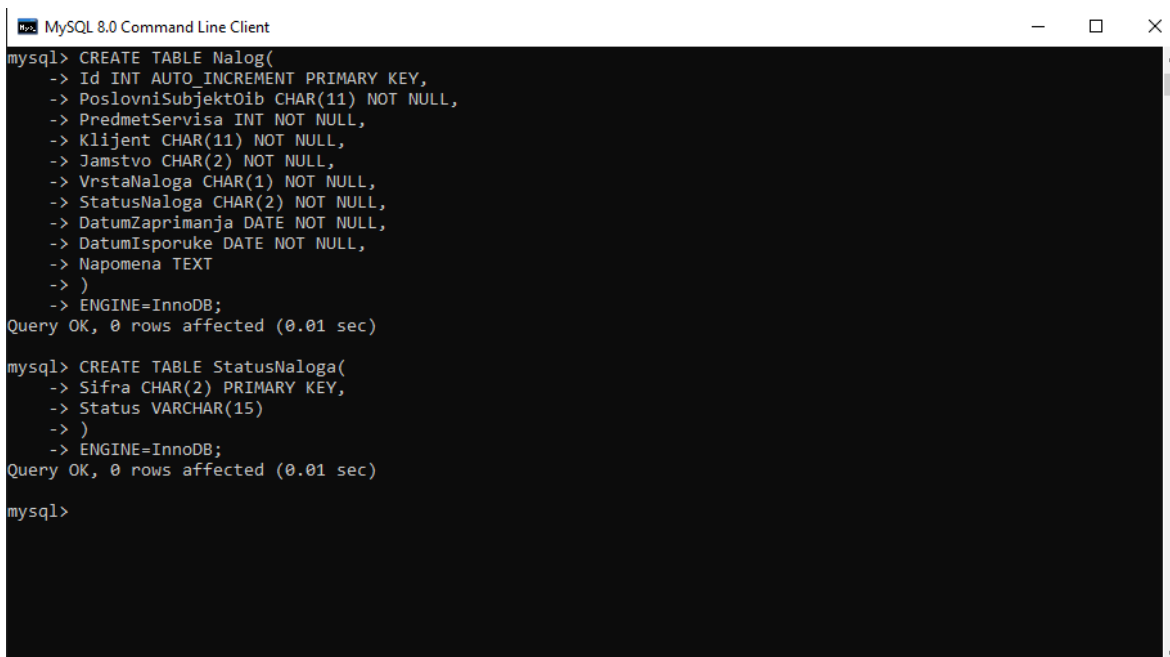
Slika 11. Kreiranje tablica Serviser, UgradenaKomponenta i PredmetServisa.

U kreiranoj tablici Serviser, atribut "Sifra" predstavlja primarni ključ, te mora imati pet znakova. Atribut "PrezimeImeServisera" ograničen je do maksimalno trideset znakova.

U tablici UgradenaKomponenta, "Id" predstavlja primarni ključ, te se inkrementira, odnosno, povećava za jedan prilikom unosa novog zapisa u tablicu. Atribut "PredmetServisa" je tip podatka "INT", te predstavlja strani ključ pomoću kojeg će se tablica povezati s tablicom PredmetServisa. Atribut "NazivUsluge" ograničen je do maksimalno pedeset znakova, te je ovdje predviđeno upisivanje usluga koje su se izvršile za određeni predmet servisa. Atribut "JedinicaMjere" mora imati tri znaka, te predstavlja jedinicu mjere odrađene usluge. Atribut "Kolicina" predstavlja količinu za određenu uslugu. Atribut "Cijena" je tip podatka "DECIMAL(10, 2)", odnosno, prihvaća unos decimalne vrijednosti do maksimalno deset znamenki koje uključuju i dva decimalna mjesta.

U tablici PredmetServisa, atribut "Id" predstavlja primarni ključ. Atribut "PrezimeImeKlijenta" zahtjeva jedanaest znakova pri unosu, te predstavlja strani ključ pomoću kojeg će se tablica povezati s tablicom Klijent. Atribut "SerijskiBroj" zahtjeva trinaest znakova pri unosu, te se ovdje upisuje serijski broj predmeta servisa. Atribut "OpisKvara" je tip podatka "TEXT", odnosno u njega je moguće pohraniti duže tekstualne vrijednosti. Atribut "PrezimeImeServisera" zahtjeva pet znakova pri unosu, te predstavlja strani ključ s kojim će se tablica povezati s tablicom Serviser.

Na slici 12. je vidljiv proces kreiranje tablica Nalog i StatusNaloga



```
mysql> CREATE TABLE Nalog(  
  -> Id INT AUTO_INCREMENT PRIMARY KEY,  
  -> PoslovniSubjektOib CHAR(11) NOT NULL,  
  -> PredmetServisa INT NOT NULL,  
  -> Klijent CHAR(11) NOT NULL,  
  -> Jamstvo CHAR(2) NOT NULL,  
  -> VrstaNaloga CHAR(1) NOT NULL,  
  -> StatusNaloga CHAR(2) NOT NULL,  
  -> DatumZaprimanja DATE NOT NULL,  
  -> DatumIsporuke DATE NOT NULL,  
  -> Napomena TEXT  
  -> )  
  -> ENGINE=InnoDB;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> CREATE TABLE StatusNaloga(  
  -> Sifra CHAR(2) PRIMARY KEY,  
  -> Status VARCHAR(15)  
  -> )  
  -> ENGINE=InnoDB;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>
```

**Slika 12.** Kreiranje tablica Nalog i StatusNaloga

U tablici Nalog, atribut “Id” predstavlja primarni ključ. Atribut “PoslovniSubjektOib” zahtjeva jedanaest znakova pri unosu, te predstavlja strani ključ pomoću kojeg će se tablica povezati s tablicom PoslovniSubjekt. Atribut “PredmetServisa” je tip podatka “INT”, te predstavlja strani ključ pomoću kojeg će se tablica povezati s tablicom PredmetServisa. Atribut “Klijent”. Kao i “PoslovniSubjektOib” zahtjeva jedanaest znakova i predstavlja strani ključ pomoću kojeg će se tablica povezati s tablicom Klijent. Atribut “Jamstvo” zahtjeva dva znaka pri unosu, te se ovdje unosi je li klijent ima jamstvo za određeni predmet servisa. Atribut “VrstaNaloga” označava je li nalog redovan ili izvanredan, te zahtjeva samo jedan znak pri unosu. Atribut “StatusNaloga” zahtjeva dva znaka pri unosu, te predstavlja strani ključ pomoću kojeg će se tablica povezati s tablicom StatusNaloga. Atributi “DatumZaprimanja” i “DatumIsporuke” su tip podatka “DATE”, te predstavljaju kada je predmet servisa donesen na servis i kada je nalog isporučen klijentu. Atribut “Napomena” je tip podatka “TEXT”, te je predviđen za upis napomene klijentu.

Tablica StatusNaloga sastoji se od dva atributa: “Sifra” i “Status”. Atribut “Sifra” zahtjeva dva znaka pri unosu, te predstavlja primarni ključ tablice. Atribut “Status” označava status naloga, odnosno, je li nalog preuzet ili ne.

Na slikama 13. i 14., vidljivo je povezivanje prethodno kreiranih tablica, odnosno, kreiranje relacija između tablica.

```
MySQL 8.0 Command Line Client
mysql> ALTER TABLE ziororacun
  -> ADD FOREIGN KEY (PoslovniSubjektOIB) REFERENCES poslovnisubjekt(Oib)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE ugradenakomponenta
  -> ADD FOREIGN KEY (PredmetServisa) REFERENCES predmetservisa(Id)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE nalog
  -> ADD FOREIGN KEY (PoslovniSubjektOIB) REFERENCES poslovnisubjekt(Oib)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE nalog
  -> ADD FOREIGN KEY (PredmetServisa) REFERENCES predmetservisa(Id)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

Slika 13. Kreiranje relacija između tablica.

```
MySQL 8.0 Command Line Client
mysql> ALTER TABLE nalog
  -> ADD FOREIGN KEY (Klijent) REFERENCES klijent(Oib)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE predmetservisa
  -> ADD FOREIGN KEY (PrezimeImeKlijenta) REFERENCES klijent(Oib)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE predmetservisa
  -> ADD FOREIGN KEY (PrezimeImeServisera) REFERENCES serviser(Sifra)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE nalog
  -> ADD FOREIGN KEY (StatusNaloga) REFERENCES statusnaloga(Sifra)
  -> ON DELETE CASCADE
  -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

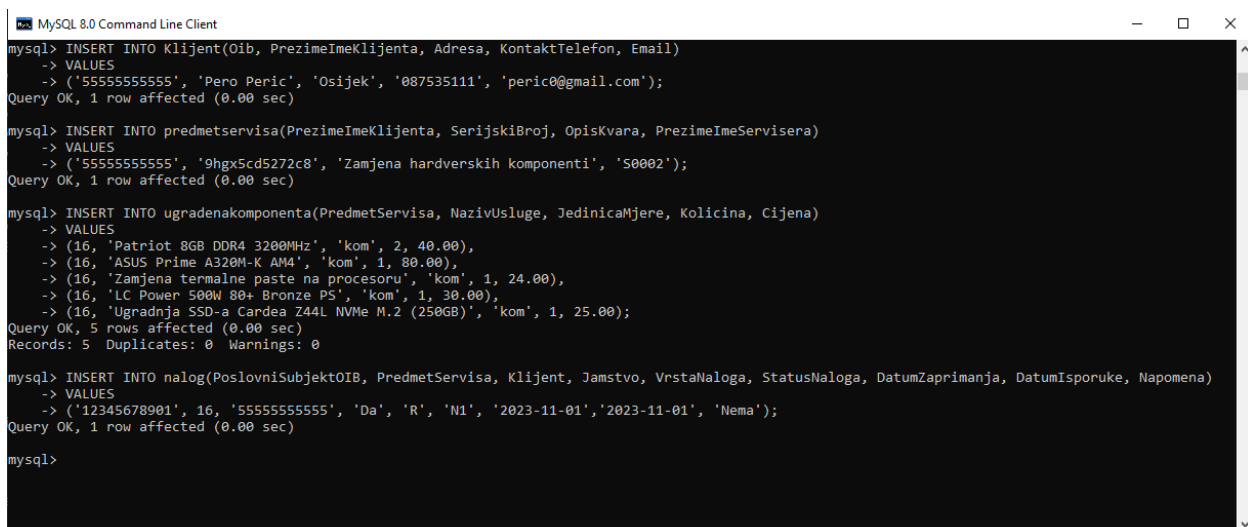
Slika 14. Kreiranje relacija između tablica.

Kreiranje je relacija između tablica obavljeno naknadnim dodavanjem stranog ključa koristeći se naredbom SQL-a kojom je moguće izmijeniti određenu tablicu: “ALTER TABLE”. Naredba “ADD FOREIGN KEY” označava dodavanje stranog ključa, gdje je prvi parametar atribut tablice koji označava strani ključ, dok je drugi parametar tablica s kojom se povezuje, njezin primarni ključ.

### 6.3. Primjeri testiranja ispravnosti modela baze podataka računalnog servisa

Nakon kreiranja tablica baze podataka, te uspostave veza između tih tablica, provedeno je testiranje ispravnosti kreiranog modela baze podataka računalnog servisa upotrebom naredbi za unos, projekcije, ažuriranje i brisanje zapisa u relacijama baze podataka. Unos zapisa se obavio upotrebom SQL naredbi INSERT. Naredba se SELECT koristila jednostavne i složene upite, te kreiranje projekcija na temelju upita. Naredbom su UPDATE obavljene izmjene nad zapisima, dok su se naredbom DELETE brisali zapisi iz kreirane baze podataka.

Na slici 15. prikazan je unos novih zapisa u bazu podataka naredbom INSERT.



```
mysql> INSERT INTO Klijent(Oib, PrezimeImeKlijenta, Adresa, KontaktTelefon, Email)
-> VALUES
-> ('5555555555', 'Pero Peric', 'Osijek', '087535111', 'peric0@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO predmetservisa(PrezimeImeKlijenta, SerijskiBroj, OpisKvara, PrezimeImeServisera)
-> VALUES
-> ('5555555555', '9hgx5cd5272c8', 'Zamjena hardverskih komponenti', 'S0002');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO ugradenakomponenta(PredmetServisa, NazivUsluge, JedinicaMjere, Kolicina, Cijena)
-> VALUES
-> (16, 'Patriot 8GB DDR4 3200MHz', 'kom', 2, 40.00),
-> (16, 'ASUS Prime A320M-K AM4', 'kom', 1, 80.00),
-> (16, 'Zamjena termalne paste na procesoru', 'kom', 1, 24.00),
-> (16, 'LC Power 500W 80+ Bronze PS', 'kom', 1, 30.00),
-> (16, 'Ugradnja SSD-a Cardea Z44L NVMe M.2 (250GB)', 'kom', 1, 25.00);
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO nalog(PoslovniSubjektOIB, PredmetServisa, Klijent, Jamstvo, VrstaNaloga, StatusNaloga, DatumZaprimanja, DatumIsporuke, Napomena)
-> VALUES
-> ('12345678901', 16, '5555555555', 'Da', 'R', 'N1', '2023-11-01', '2023-11-01', 'Nema');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Slika 15. Unos novog zapisa u tablicu relacijske baze podataka

Kao što se vidi iz slike 15., u tablicu “Klijent” unesen je jedan novi zapis. U tablicu “PredmetServisa” također je unesen jedan novi zapis. U tablicu “UgradenaKomponenta” uneseno je pet novih zapisa, te je u tablicu “Nalog” unesen jedan novi zapis. Na slici 16., prikazani su novi zapisi dobiveni upotrebom SQL naredbe SELECT.

```
mysql> SELECT klijent.PrezimeImeKlijenta, klijent.Adresa, predmetservisa.OpisKvara, ugradenakomponenta.NazivUsluge, ugradenakomponenta.Cijena
-> FROM klijent
-> INNER JOIN predmetservisa
-> ON klijent.Oib=predmetservisa.PrezimeImeKlijenta
-> INNER JOIN ugradenakomponenta
-> ON predmetservisa.Id=ugradenakomponenta.PredmetServisa
-> WHERE klijent.Oib="5555555555"
-> ORDER BY
-> ugradenakomponenta.Cijena;
```

PrezimeImeKlijenta	Adresa	OpisKvara	NazivUsluge	Cijena
Pero Peric	Osijek	Zamjena hardverskih komponenti	Zamjena termalne paste na procesoru	24.00
Pero Peric	Osijek	Zamjena hardverskih komponenti	Ugradnja SSD-a Cardea Z44L NVMe M.2 (250GB)	25.00
Pero Peric	Osijek	Zamjena hardverskih komponenti	LC Power 500W 80+ Bronze PS	30.00
Pero Peric	Osijek	Zamjena hardverskih komponenti	Patriot 8GB DDR4 3200MHz	40.00
Pero Peric	Osijek	Zamjena hardverskih komponenti	ASUS Prime A320M-K AM4	80.00

```
5 rows in set (0.00 sec)

mysql> _
```

**Slika 16.** Projekcija novih zapisa upotrebom SQL naredbe SELECT.

Naredba SELECT omogućava projekciju željenih atributa iz određenih tablica koji će se prikazati kao rezultat prilikom realizacije upita. Na primjeru iz slike 16, odabrano je pet atributa: “PrezimeImeKlijenta”, “Adresa”, “OpisKvara”, “NazivUsluge” i “Cijena” iz tri tablice: “Klijent”, “PredmetServisa” i “UgradenaKomponenta”. Svrha ovog upita je prikazati novo unesene zapise sa slike 15. na način da se kombiniraju redovi prethodno navedenih tablica uz određene uvjete. Kombiniranje redova omogućeno je naredbom INNER JOIN, dok naredba ON definira uvjet pod kojim će se određene tablice spojiti. Zatim, naredba WHERE također predstavlja uvjet pod kojim će se određeni upit izvršiti, gdje je konkretno definirano da se prikažu rezultati samo za određenog klijenta, u ovom slučaju, potrebno je prosljediti oib klijenta u upit. U konačnici, naredbom ORDER BY, rezultat prethodno definiranih upita poredan je po cijeni ugrađene komponente, počevši od najmanje prema najvećoj. Na slikama 17. i 18. prikazano je još dva primjera upotrebe SQL naredbe SELECT u kombinaciji s nekim od prethodno navedenih naredbi.



```

MySQL 8.0 Command Line Client
mysql> SELECT * FROM nalog
-> WHERE
-> Jamstvo != "Ne" && VrstaNaloga="R" && StatusNaloga = "N1"
-> ORDER BY
-> DatumZaprimanja;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | PoslovniSubjektOib | PredmetServisa | Klijent | Jamstvo | VrstaNaloga | StatusNaloga | DatumZaprimanja | DatumIsporuke | Napomena |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 12345678901 | 1 | 11122233344 | Da | R | N1 | 2023-01-01 | 2023-01-15 | Uplata na ziro-racun neuspjesna! |
| 3 | 12345678901 | 3 | 44433322211 | Da | R | N1 | 2023-02-10 | 2023-02-20 | Nema |
| 10 | 12345678901 | 10 | 11122233344 | Da | R | N1 | 2023-05-15 | 2023-06-01 | Nema |
| 12 | 12345678901 | 12 | 66699944422 | Da | R | N1 | 2023-06-01 | 2023-06-16 | Nema |
| 16 | 12345678901 | 16 | 55555555555 | Da | R | N1 | 2023-11-01 | 2023-11-01 | Nema |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set, 2 warnings (0.00 sec)

mysql>

```

Slika 17. Prvi primjer upotrebe naredbe SELECT.

Na slici 17., napravljena je jednostavnija projekcija nego u prethodnom primjeru. Razlika je što se ovdje uzimaju svi atributi iz određene tablice, te je naredbom WHERE postavljen uvjet pod kojim je određeni zapis u tablici prikazan. Konkretnije, sva tri postavljena uvjeta moraju biti zadovoljena, a to su: “Jamstvo” ne smije biti “Ne”, “VrstaNaloga” mora biti “R”, te “StatusNaloga” mora biti “N1”. Kako bi se postavio ovakav uvjet, potrebno je koristiti logički operator AND (&&). Naredbom ORDER BY, zapisi su posloženi po atributu “DatumZaprimanja” počevši od najmanjeg datuma.

```

MySQL 8.0 Command Line Client
mysql> SELECT klijent.PrezimeImeKlijenta, predmetservisa.OpisKvara, ugradenakomponenta.NazivUsluge, ugradenakomponenta.Cijena, serviser.PrezimeImeServisera
-> FROM klijent
-> INNER JOIN predmetservisa
-> ON
-> klijent.Oib=predmetservisa.PrezimeImeKlijenta
-> INNER JOIN ugradenakomponenta
-> ON
-> predmetservisa.Id=ugradenakomponenta.PredmetServisa
-> INNER JOIN serviser
-> ON
-> predmetservisa.PrezimeImeServisera=serviser.Sifra
-> WHERE
-> klijent.PrezimeImeKlijenta='Sara Saric' || klijent.PrezimeImeKlijenta='Nikola Nikolic' || klijent.PrezimeImeKlijenta='Teo Teic'
-> ORDER BY
-> klijent.PrezimeImeKlijenta;
+-----+-----+-----+-----+-----+
| PrezimeImeKlijenta | OpisKvara | NazivUsluge | Cijena | PrezimeImeServisera |
+-----+-----+-----+-----+-----+
| Nikola Nikolic | Racunalo se ne ukljucuje | Reinstalacija operacijskog sustava | 20.99 | Ivan Ivanic |
| Nikola Nikolic | Driveri za graficku karticu se ne zeze azurirati | Potpuno brisanje drivera, te ponovna instalacija | 21.99 | Marin Maric |
| Nikola Nikolic | Ventilatori proizvode cudne zvukove | Ciscenje hardvera racunala | 21.00 | Marin Maric |
| Sara Saric | Ciscenje od virusa | Instalacija antivirusnog softvera | 20.00 | Niko Nikic |
| Sara Saric | Racunalo se previse sporo ukljucuje | Reinstalacija operacijskog sustava | 25.00 | Ivan Ivanic |
| Teo Teic | Driveri se sami brisu | Reinstalacija operacijskog sustava | 20.99 | Niko Nikic |
| Teo Teic | Problemi sa zvukom | Azuriranje drivera | 16.99 | Marko Markic |
+-----+-----+-----+-----+-----+
7 rows in set, 2 warnings (0.00 sec)

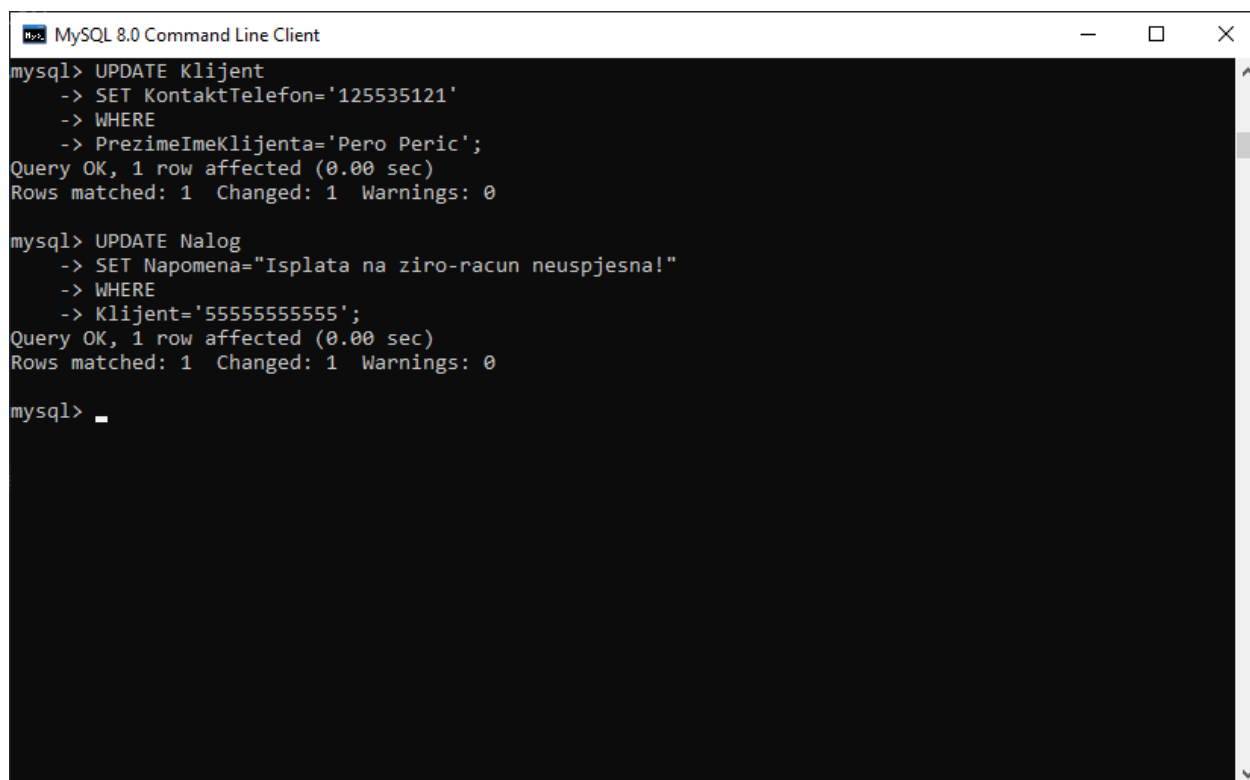
mysql>

```

Slika 18. Drugi primjer upotrebe naredbe SELECT.

Na slici 18., napravljena je složenija projekcija kao na slici 16., s time da se ovdje prikazuju rezultati upita za tri klijenta istovremeno. Ovakva projekcija postigla se upotrebom logičkog sklopa OR (||) u naredbi WHERE, gdje je bilo potrebno proslijediti imena i prezimena klijenata. Naredbom ORDER BY, klijenti su poredani abecedno po imenu i prezimenu.

Na slici 19., prikazani su primjeri upotrebe SQL naredbe UPDATE.



```
mysql> UPDATE Klijent
-> SET KontaktTelefon='125535121'
-> WHERE
-> PrezimeImeKlijenta='Pero Peric';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE Nalog
-> SET Napomena="Isplata na ziro-racun neuspjesna!"
-> WHERE
-> Klijent='5555555555';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> _
```

**Slika 19.** Primjer upotrebe SQL naredbe UPDATE.

Kao što se vidi iz slike 19., naredbom UPDATE ažuriran je atribut “KontaktTelefon” u tablici “Klijent”, te su naredbom SET postavljene nove vrijednost navedenog atributa. Naredbom WHERE definirano je točno koji klijent u tablici se ažurira. Zatim, ažuriran je i atribut “Napomena” u tablici “Nalog” upotrebom prethodno opisanog principa.

Na slici 20. prikazane su tablice “Klijent” i “Nalog” nakon ažuriranja za određeni zapis.

```

MySQL 8.0 Command Line Client
mysql> SELECT * FROM klijent
-> WHERE PrezimeImeKlijenta="Pero Peric";
+-----+-----+-----+-----+-----+
| Oib      | PrezimeImeKlijenta | Adresa | KontaktTelefon | Email      |
+-----+-----+-----+-----+-----+
| 5555555555 | Pero Peric      | Osijek | 125535121      | peric@gmail.com |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM nalog
-> WHERE Klijent="5555555555";
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | PoslovniSubjektOib | PredmetServisa | Klijent | Jamstvo | VrstaNaloga | StatusNaloga | DatumZaprimanja | DatumIsporuke | Napomena |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 16 | 12345678901      |                | 16      | Da      | R          | N1          | 2023-11-01      | 2023-11-01      | Isplata na ziro-racun neuspjesna! |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _

```

Slika 20. Rezultat upotrebe SQL naredbe UPDATE

Na slici 21. prikazan je primjer upotrebe SQL naredbe DELETE.

```

MySQL 8.0 Command Line Client
mysql> DELETE FROM klijent
-> WHERE
-> Oib="55511133365" || Oib="55555555555" || Oib="11122233344";
Query OK, 3 rows affected, 2 warnings (0.00 sec)

mysql> _

```

Slika 21. Primjer upotrebe SQL naredbe DELETE.

Kao što se vidi na slici 21, obrisano je ukupno tri zapisa iz tablice “Klijent”, gdje je u naredbu WHERE trebalo proslijediti OIB-e klijenata koji se brišu, no s obzirom da je tablica “Klijent” povezana s tablicama “Nalog” i “PredmetServisa”, iz ove dvije tablice su također obrisani svi zapisi koji imaju veze s prethodna tri zapisa iz tablice “Klijent”. Razlog toga je naredba ON CASCADE DELETE koja se provodi prilikom kreiranja relacija između tablica.

Na slici 22. moguće je vidjeti tablice “Klijent”, “PredmetServisa” i “Nalog” nakon izvršene SQL naredbe DELETE.

```

mysql> SELECT * FROM Klijent;
+----+-----+-----+-----+-----+
| Oib | PrezimeImeKlijenta | Adresa | KontaktTelefon | Email |
+----+-----+-----+-----+-----+
| 11199966633 | Teo Teic | Nasice | 146130486 | teic@gmail.com |
| 22211133355 | Sonja Sonjic | Valpovo | 205550242 | lukic5@gmail.com |
| 44433322211 | Ana Anic | Petrinja | 804761641 | anic5@gmail.com |
| 66622255500 | Sara Saric | Donji Miholjac | 652037202 | saric0@gmail.com |
| 66699944422 | Hrvoje Hrvojcic | Vinkovci | 742557668 | hrvojcic@gmail.com |
| 77700022266 | Anja Anjic | Osijek | 257352817 | anjic1@gmail.com |
| 77722211177 | Lea Leic | Slatina | 321713017 | leic0@gmail.com |
| 88877711199 | Marina Marinic | Grubisno Polje | 400565806 | marinic0@gmail.com |
+----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> SELECT * FROM PredmetServisa;
+----+-----+-----+-----+-----+
| Id | PrezimeImeKlijenta | SerijskiBroj | OpisKvara | PrezimeImeServisera |
+----+-----+-----+-----+-----+
| 3 | 44433322211 | e5z6cpk5mo881 | "Kernel panic" greska | S0002 |
| 4 | 22211133355 | gw4336kgz2z2jr | "Blue Screen Of Death" greska prilikom ukljucivanja aplikacije | S0001 |
| 5 | 11199966633 | yap68w1260mzf | Driveri se sami brisu | S0001 |
| 6 | 11199966633 | cl0dj8426j6ph | Problem sa zvukom | S0004 |
| 7 | 88877711199 | bk7765mml2zi | Problem sa usb utinom | S0001 |
| 8 | 77700022266 | wl510wc6p80me | Pregrijavanje procesora | S0003 |
| 9 | 44433322211 | xh8r6j8va11n4 | Pregrijavanje napajanja | S0002 |
| 11 | 66699944422 | 7kg362e6288ab | Sustav ne prepoznaje sve RAME | S0004 |
| 12 | 66699944422 | 72r5kcv10h61j | Problem sa wifi-em | S0002 |
| 13 | 66622255500 | 676nb8fx5o01p | Ciscenje od virusa | S0001 |
| 14 | 66622255500 | 2dlpy7p81zcx | Racunalo se previse sporo ukljucuje | S0003 |
+----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> SELECT * FROM Nalog;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | PoslovniSubjektOib | PredmetServisa | Klijent | Jamstvo | VrstaNaloga | StatusNaloga | DatumZaprimanja | DatumIsporuke | Napomena |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | 12345678901 | 3 | 44433322211 | Da | R | N1 | 2023-02-10 | 2023-02-20 | Nema |
| 4 | 12345678901 | 4 | 22211133355 | Da | I | N0 | 2023-02-25 | 2023-03-10 | Nalog nije preuzet! |
| 5 | 12345678901 | 5 | 11199966633 | Ne | I | N0 | 2023-03-20 | 2023-03-25 | Nadoplatiti 20 eura zbog servisa. |
| 6 | 12345678901 | 6 | 11199966633 | Ne | I | N0 | 2023-03-21 | 2023-03-30 | Nadoplatiti 100 eura za nove dijelove. |
| 7 | 12345678901 | 7 | 88877711199 | Da | R | N0 | 2023-04-05 | 2023-04-15 | Nema |
| 8 | 12345678901 | 8 | 77700022266 | Da | I | N1 | 2023-04-20 | 2023-05-10 | Uplata na ziro racun neuspjesna! |
| 9 | 12345678901 | 9 | 44433322211 | Da | R | N0 | 2023-05-02 | 2023-05-10 | Nalog nije preuzet! |
| 11 | 12345678901 | 11 | 66699944422 | Ne | R | N0 | 2023-05-29 | 2023-06-08 | Potrebna nadoplata za novu termalnu pastu. |
| 12 | 12345678901 | 12 | 66699944422 | Da | R | N1 | 2023-06-01 | 2023-06-16 | Nema |
| 13 | 12345678901 | 13 | 66622255500 | Ne | I | N0 | 2023-06-25 | 2023-07-01 | Nema |
| 14 | 12345678901 | 14 | 66622255500 | Ne | I | N0 | 2023-06-26 | 2023-07-10 | Nema |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>

```

Slika 22. Rezultat upotrebe SQL naredbe DELETE.

## 7. Zaključak

Računalni servisi nude široki spektar usluga koje uključuju između ostalog i popravak računala, prodaju gotovih računala ili pak sastavljanje računala od komponenti koje klijent zatraži. U ovakvim procesima sudjeluje dokumentacija koju prije svega čine radni nalozi, potvrde o zaprimanju opreme, izvještaji o dijagnostici, ponude za popravak, potvrde o popravku, računi te potvrde o preuzimanju popravljene robe. Iz dokumentacije proizlazi podatkovni obuhvat koji sudjeluje u determiniranju sadržaja baze podataka koja je u temelju svake programske aplikacije koja se koristi u praćenju procesa koji se odvijaju u računalnom servisu. Kako je fokus predmetnog istraživanja stavljen na iznalaženje optimalnog modela relacijske baze podataka za računalne servise u Republici Hrvatskoj, prije pristupanja dizajniranju navedenog modela bilo je potrebno istražiti u okviru primarnih istraživanja procese vezane za rad računalnih servisa kroz analizu navedene dokumentacije, kao i u okviru sekundarnih istraživanja istražiti aktualne modele baza podataka, te odabrati optimalan model za kreiranje baze podataka koja će omogućiti kroz evidentiranje poslovne strukture i poslovnih procesa praćenje rada i podršku računalnim servisima. U konačnici je modelirana baza podataka koja je namijenjena računalnom servisu, te je ista testirana.

Temeljem analize aktualne domaće i svjetske znanstvene i stručne literature odabran je relacijski model podataka kao aktualan model koji se mahom koristi u programskim aplikacijama koje su namijenjene poslovanju. Razlozi su zbog kojih se koristi relacijski model u poslovnoj praksi: jednostavnosti, fleksibilnosti i mogućnosti normalizacije koja smanjuje redundancije podataka. Da bi se odredio sadržaj optimalnog modela relacijske baze podataka istraženi su računalni servisi s područja Grada Osijeka, te se temeljem prikupljene dokumentacije iz tih servisa sagledao zajednički podatkovni obuhvat koji je nužan za praćenje procesa koji se odvijaju u računalnim servisima u Republici Hrvatskoj. U istraživanju je sudjelovalo pet računalnih servisa iz Grada Osijeka. Navedeni su servisi za potrebe ovog istraživanja ustupili obrasce dokumenata među kojima je zasigurno najvažniji dokument radni nalog. Radni nalog je osnovni dokument koji se koristi za praćenje procesa servisa, te sadrži podatke o klijentu, opis problema, datum zaprimanja, detalje o izvršenim radnjama, predmet servisa, datum povrata opreme i drugo. Kako bi se utvrdile ranije spomenute komponente koje sudjeluju u procesu modeliranja baze podataka, metodom

analize svaki radni nalog je detaljno sagledan, te su izdvojeni zajednički nazivnici, odnosno, sve komponente naloga koje su jednake u svakom od istraženih računalni servisa. Zatim, metodom apstrakcije izdvojene su samo bitne komponente koje su se dalje koristile za modeliranje baze podataka.

Temeljem dobivenih je spoznaja kroz provedena primarna i sekundarna istraživanja modelirana bazu podataka koja je potom testirana. Prvo je metodom modeliranja relacijske baze podataka sintezom načinjen model baze podataka koji je testiran i na kojem su načinjene dodatne vertikalne normalizacije kako bi se izbacile iz modela sve nekontrolirane redundancije. Za testiranje kreiranog modela baze podataka za računalni servis upotrijebljen je MySQL server, odnosno besplatni relacijski sustav za upravljanje bazama podataka koji je široko popularan u programerskim krugovima u Republici Hrvatskoj. Konkretno je alat koji je upotrebljavan za izradu baze podataka, te njezino testiranje Oracle-ov MySQL Server u trenutnoj verziji 8.0.33. U konačnici, proveden je eksperiment simulacijom poslovanja računalnog servisa kreiranjem, pregledom, ažuriranjem i brisanjem podataka u modeliranoj bazi podataka.

Nakon sinteze rezultata istraživanja je vidljivo kako su svi postavljeni ciljevi istraživanja realizirani, a konkretno je rezultat ovog istraživanja optimalan relacijski model baze podataka koji se može koristiti kao predložak u razvoju programskih aplikacija za računalne servise u Republici Hrvatskoj. Ovim istraživanjem otvoren je i niz novih problema, među kojima se ističe potreba izrade modela programske aplikacije koja bi se koristila navedenom bazom podataka. Osim u vertikali, daljnja istraživanja mogu ići i u širinu u vidu definiranja optimalnih modela relacijskih baza podataka za ostale segmente koji su sadržani u poslovanju računalnog servisa, kao što je model baze podataka za robno-materijalno knjigovodstvo, za CRM sustav, za financijsko računovodstvo i drugo.

## 8. Popis literature

1. Carić, T., Buntić M.: Uvod u relacijske baze podataka, Zagreb, 2015
2. Dukić, B.: Baze podataka i poslovni procesi, Ekonomski fakultet u Osijeku, Osijek, 2010
3. Hierarchical Model in DBMS, geeksforgeeks, <https://www.geeksforgeeks.org/hierarchical-model-in-dbms/> [28.6.2023.]
4. Informacija, enciklopedija.hr, <https://www.enciklopedija.hr/natuknica.aspx?id=27405>, [26.6.2023.]
5. Kramberger, T., Duk, S., Kovačević, R.: Baze podataka, Tehničko veleučilište u Zagrebu, Zagreb, 2018
6. Licul, M.: Modeliranje i izgradnja baze podataka za web aplikaciju, Završni rad, Veleučilište u Rijeci, Rijeka, 2022., <https://urn.nsk.hr/urn:nbn:hr:125:343314> [29.6.2023.]
7. Pavlić, M.: Oblikovanje baza podataka, Odjel za informatiku, Rijeka, 2011
8. Podaci, element.hr, <https://element.hr/wp-content/uploads/2020/06/unutra-13639.pdf>, str.1 [26.6.2023.]
9. Podatak, enciklopedija.hr, <https://www.enciklopedija.hr/natuknica.aspx?id=48887> [26.6.2023.]
10. Relational Database (Model, Operations & Constraints), databasetown, <https://databasetown.com/relational-database-model-operations-constraints/> [29.6.2023.]
11. Structured Query Language (SQL), techtarget, <https://www.techtarget.com/searchdatamanagement/definition/SQL> [28.6.2023.]
12. SQL history, w3schools, <https://www.w3schools.in/sql/history> [28.6.2023.]
13. SQL intro, w3schools, [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp) [28.6.2023.]
14. Vukmirović, S.: Modeliranje i analiza podataka u poslovanju, Ekonomski fakultet u Rijeci, 2013
15. What is a Database Model, Lucidchart, <https://www.lucidchart.com/pages/database-diagram/database-models> [20.6.2023.]
16. What is Object Oriented Model in DBMS?, scaler, <https://www.scaler.com/topics/object-oriented-model-in-dbms/> [28.6.2023.]
17. What Is SQL (Structured Query Language)?, aws Amazon, <https://aws.amazon.com/what-is/sql/> [28.6.2023.]

## Popis slika

<b>Slika 1.</b> Transformiranje/interpretacija podataka u informacije. ....	6
<b>Slika 2.</b> Koraci i poslovi pri oblikovanju baze podataka. ....	8
<b>Slika 3.</b> Odnos u modeliranju između konceptualnog modela i logičkih modela baze podataka. ....	11
<b>Slika 4.</b> Prikaz veze jedan prema jedan. ....	22
<b>Slika 5.</b> Prikaz veze jedan prema više. ....	22
<b>Slika 6.</b> Prikaz veze više prema više. ....	23
<b>Slika 7.</b> Tablice s elementima koji se pojavljuju u radnim nalogima istraženih računalnih servisa. ....	30
<b>Slika 8.</b> Izgled sučelja aplikacije MySQL server. ....	31
<b>Slika 9.</b> Model baze podataka u pseudo kodu. ....	33
<b>Slika 10.</b> Kreiranje tablica PoslovniSubjekt, ZiroRacun i Klijent. ....	34
<b>Slika 11.</b> Kreiranje tablica Serviser, UgradenaKomponenta i PredmetServisa. ....	35
<b>Slika 12.</b> Kreiranje tablica Nalog i StatusNaloga. ....	37
<b>Slika 13.</b> Kreiranje relacija između tablica. ....	38
<b>Slika 14.</b> Kreiranje relacija između tablica. ....	38
<b>Slika 15.</b> Unos novog zapisa u tablicu relacijske baze podataka. ....	39
<b>Slika 16.</b> Projekcija novih zapisa upotrebom SQL naredbe SELECT. ....	40
<b>Slika 17.</b> Prvi primjer upotrebe naredbe SELECT. ....	41
<b>Slika 18.</b> Drugi primjer upotrebe naredbe SELECT. ....	41
<b>Slika 19.</b> Primjer upotrebe SQL naredbe UPDATE. ....	42
<b>Slika 20.</b> Rezultat upotrebe SQL naredbe UPDATE. ....	43
<b>Slika 21.</b> Primjer upotrebe SQL naredbe DELETE. ....	43
<b>Slika 22.</b> Rezultat upotrebe SQL naredbe DELETE. ....	44