

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

**Vatroslav Mileusnić**

# **MODELIRANJE I DIZAJN BAZE PODATAKA**

**ZAVRŠNI RAD**

**Varaždin, 2011.**

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

**Vatroslav Mileusnić**

**Redoviti student**

**Broj indeksa: 35978/07-R**

**Smjer: Informacijski sustavi**

**Preddiplomski studij**

**MODELIRANJE I DIZAJN BAZE PODATAKA**

**ZAVRŠNI RAD**

**Mentor:**

prof. dr. sc. Mirko Maleković, redoviti  
profesor

**Varaždin, lipanj 2011.**

# Sadržaj

<b>1. UVOD.....</b>	<b>1</b>
<b>2. SEMANTIČKO MODELIRANJE BAZE PODATAKA.....</b>	<b>2</b>
2.1. OSNOVNI ER KONSTRUKTI .....	2
2.1.1. <i>Entiteti</i> .....	2
2.1.2. <i>Atributi</i> .....	3
2.1.3. <i>Veze</i> .....	4
2.2. NAPREDNI ER KONSTRUKTI .....	6
2.2.1. <i>Generalizacija: nadtipovi i podtipovi</i> .....	6
2.2.2. <i>Agregacija</i> .....	7
2.2.3. <i>Referencijalni integritet</i> .....	7
2.3. ER MODEL.....	7
2.3.1. <i>Informacije za bazu podataka</i> .....	7
2.3.2. <i>Izrada ER modela</i> .....	8
2.3.3. <i>ER model baze podataka</i> .....	10
2.3.4. <i>Pojašnjenje ER modela</i> .....	11
<b>3. LOGIČKO MODELIRANJE BAZE PODATAKA.....</b>	<b>15</b>
3.1. KREIRANJE TABLICA.....	15
3.1.1. <i>Reprezentiranje veza</i> .....	15
3.1.2. <i>Primarni ključevi</i> .....	15
3.2. ZAVISNOSTI .....	16
3.2.1. <i>Funkcionalne zavisnosti</i> .....	16
3.2.2. <i>Višeznačne zavisnosti</i> .....	17
3.2.3. <i>Zavisnosti spoja</i> .....	17
3.3. NORMALNE FORME.....	18
3.3.1. <i>Prva normalna forma</i> .....	19
3.3.2. <i>Druga normalna forma</i> .....	19
3.3.3. <i>Treća normalna forma</i> .....	20
3.3.4. <i>Boyce-Coddova normalna forma</i> .....	20
3.3.5. <i>Četvrta normalna forma</i> .....	20
3.3.6. <i>Peta normalna forma</i> .....	21
3.3.7. <i>Šesta normalna forma</i> .....	21
3.4. NORMALIZACIJA.....	21
3.4.1. <i>Popis tablica i njihovih zavisnosti</i> .....	22
<b>4. SQL DIZAJN BAZE PODATAKA .....</b>	<b>24</b>
<b>5. ZAKLJUČAK .....</b>	<b>28</b>
<b>6. LITERATURA.....</b>	<b>29</b>

# 1. Uvod

Dizajn relacijskih baza podataka je evoluirao od umjetnosti u znanost, tvrde autori knjige „Modeliranje i dizajn baze podataka“<sup>1</sup>, a ja se s njima potpuno slažem, te ću kroz ovaj rad prikazati i opisati elemente semantičkog i logičkog modeliranja, te SQL dizajna baze podataka na primjeru baze podataka web stranice koja se bavi glazbom.

Na samome početku bih napomenuo razliku između semantičkog i logičkog modeliranja. Prilikom semantičkog modeliranja govorimo o entitetima, atributima i vezama, te u njega uvrštavamo tehnike za definiranje značenja podataka u kontekstu svojih veza s drugim podacima, gradeći konceptualni model podataka koje koristi neka organizacija. Semantičko se modeliranje još naziva i konceptualnim oblikovanjem. Prilikom logičkog modeliranja (oblikovanja) govorimo o tablicama i stupcima, uz njihov opis, dakle opis strukture domene informacija, što uključuje prepoznavanje funkcijskih zavisnosti i zavisnosti spoja, te provjeru normalnih formi, gradeći logičku reprezentaciju baze podataka.

U praksi, za početak je potrebno dobiti informacije od naručitelja baze podataka, no u ovom ću slučaju taj korak preskočiti, odnosno, točnije, informacije ću pružati sam sebi, budući da sam ja naručitelj. Sljedeći korak je semantičko modeliranje, koje je opisano u drugom poglavlju, dok je u trećem poglavlju opisan sljedeći korak, logičko modeliranje. Četvrto poglavlje sadrži SQL dizajn baze podataka, koristeći PostgreSQL sustav za upravljanje relacijskim bazama podataka, a slijedi ga zaključak.

Moram napomenuti kako u ovome radu nisu spomenuti svi aspekti izgradnje baze podataka, poput fizičkog dizajna baze podataka, što uključuje odabir indeksa i način rada s podacima, budući da to nije dio teme ovoga rada, iako to nikako ne znači da je taj dio imalo manje važan. Također, nisu objašnjeni neki od osnovnih konstrukata relacijskih baza podataka, pa se od čitaoca očekuje da pozna i razumije osnovne ideje relacijskih baza podataka i relacijske algebre.

---

<sup>1</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), *Database Modeling and Design: Logical Design (fourth edition)*, Morgan Kaufmann, stranica 1.

## 2. Semantičko modeliranje baze podataka

Korištenje tehnika za definiranje značenja podataka u kontekstu svojih veza s drugim podacima.<sup>2</sup> Ova, naizgled zbunjujuća, rečenica dobro sažima o čemu se zapravo radi u semantičkom modeliranju baze podataka. Nije pogrešno reći kako se semantičko modeliranje baze podataka odvija u isto vrijeme kao i prikupljanje podataka potrebnih za izgradnju baze podataka, jer dok prikupljamo podatke moramo znati koja je njihova uloga, odnosno njihovo značenje, u stvarnom svijetu, kako bi mogli znati koja je njihova uloga u budućoj bazi. Podaci su međusobno povezani, bilo direktno ili indirektno, te se međusobno nadopunjavaju. Ukoliko to nije istina, odnosno, ukoliko za neke podatke ne definiramo nikakvu vezu s drugim podacima, onda ti podaci možda ne spadaju u istu bazu podataka. Za točno definiranje samih podataka i veza među njima koristimo tehnike poput ER (engl. *entity-relationship*, u prijevodu: entiteti-veze), koji je objašnjen u ovome radu, ili UML (engl. *unified modeling language*, u prijevodu: jedinstveni jezik modeliranja) modeliranja.

Sada nam je početna rečenica ovog poglavlja puno jasnija: koristimo modeliranje kako bismo objasnili podatke i njihovu povezanost.

### 2.1. Osnovni ER konstrukti

Prilikom prikupljanja podataka potrebnih za izgradnju baze podataka, raspravljamo o tri klase objekata: entitetima, njihovim atributima i vezama između entiteta. Iako jednostavno, bitno je znati raspoznati što je entitet, što atribut, a što veza, a to se lako postiže kada znamo što zapravo svaka od navedenih klasa predstavlja.

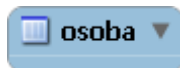
#### 2.1.1. Entiteti

Entiteti su osnovni elementi o kojima prikupljamo informacije i za koje možemo odrediti neke karakteristike, oni su objekti naše baze, koji dijele iste karakteristike. Primjeri entiteta su osoba, mjesto, stvar, događaj ili bilo što drugo što se može opisati nekim podacima; bilo što za što nam ima smisla napraviti tablicu u kojoj bi svaki red predstavljao jednu instancu tog entiteta, kao što bi u tablici osoba svaki red predstavljao jednu osobu.

U ER modelu, entitet se označuje pravokutnikom, a primjer, izrađen u alatu MySQL Workbench, se može vidjeti na slici 2.1.

---

<sup>2</sup> Federal Information Processing Standards Publication 184. (1993). Announcing the Standard for Integration definition for information modeling (IDEF1X), stranica 56



Slika 2.1

### 2.1.2. Atributi

Nakon što znamo što su entiteti, potrebno je shvatiti što su atributi. Primjetimo kako se o entitetima skupljaju informacije u vidu različitih karakteristika, kao na primjer dob i spol za osobe. U semantičkom modeliranju, takve karakteristike nazivamo atributima, i oni služe za detaljan opis entiteta. Ako su entiteti tablice, tada su atributi stupci u tablicama i njihova vrijednost opisuje instancu entiteta.

Postoje dvije vrste atributa: identifikatori i deskriptori<sup>3</sup>. Identifikator se koristi kako bi se instanca entiteta jedinstveno identificirala, te se još naziva ključnim atributom. Deskriptori se koriste kako bi se odredila vrijednost atributa koji mogu biti zajednički većem broju instanci entiteta, te se još nazivaju neključnim atributima. Primjeri ključnih atributa su OIB i šifra, dok su primjeri neključnih atributa ime, prezime, dob i spol.

Atributi mogu biti jednostavni, što znači da se ne mogu podijeliti na više manjih atributa, poput atributa OIB, ili složeni, što znači da se mogu podijeliti u više jednostavnijih atributa, poput atributa datum rođenja, koji se može podijeliti u dan, mjesec i godinu<sup>4</sup>. Vršenje ovakve podjele može biti korisno zbog statističke analize, ali je u nekim slučajevima nepotrebno.

Bitno je znati odrediti što su atributi kojeg entiteta. Kao primjer nam može poslužiti broj stanovnika u državi, što zaista možemo staviti kao jedan od mnogih atributa entitetu osoba, ali je zgodnije ako entitetu osoba dodijelimo samo atribut država, dok atribut broj stanovnika u državi pridamo entitetu država.

Atributi koji imaju višeznačne vrijednosti bi se trebali klasificirati kao entiteti, tj. ako više od jedne vrijednosti neključnog atributa odgovara jednoj vrijednosti ključnog atributa, neključni atribut bi se trebao odrediti kao entitet, iako tada takav entitet možda neće imati dodatnih neključnih atributa.<sup>5</sup> Kao primjer uzmimo filmove i njima pripadajuće žanrove; jednom filmu pripada više žanrova, ali tada ne popisujemo sve žanrove unutar jednog atributa, već izrađujemo novi entitet za žanrove, pa povezujemo entitet film i entitet žanr.

---

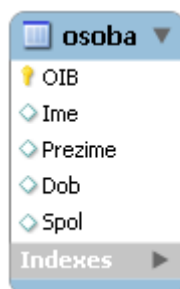
<sup>3</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), *Database Modeling and Design: Logical Design (fourth edition)*, Morgan Kaufmann, stranica 15.

<sup>4</sup> Thomas Conolly, Carolyn Begg, Richard Holowczak, (2008), *Business Database Systems*, Addison-Wesley, an imprint of Pearson Education, stranica 135.

<sup>5</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), *Database Modeling and Design: Logical Design (fourth edition)*, Morgan Kaufmann, stranica 57.

Neki se atributi mogu izvesti pomoću drugih atributa, poput starosti, koja se može izvesti iz datuma rođenja i trenutnog datuma. Takvi atributi nazivaju se izvedenim atributima<sup>6</sup> i nisu nužno uključeni u isti entitet kao atributi iz kojih su izvedeni.

U notaciji koju ću koristiti, zvanoj „vranino stopalo“<sup>7</sup>, atributi se popisuju unutar entiteta, dodatno označavajući atribut identifikator, kao što je vidljivo na slici 2.2. Obratite pažnju na ključ pokraj atributa identifikatora. U nastavku rada, za atribut indentifikator, korišten je izraz primarni ključ.



**Slika 2.2**

### 2.1.3. Veze

Veze predstavljaju asocijacije iz stvarnog svijeta između entiteta. Postoji nekoliko podjela veza: po povezivosti između entiteta razlikujemo vezu jedan-na-jedan, vezu jedan-na-više i vezu više-na-više; po opcionalnosti uključenosti entiteta razlikujemo opcionalnu i obaveznu uključenost; po stupnju konekcije, odnosno brojnosti entiteta u vezi, razlikujemo unarne, binarne i veze višeg stupnja.

Povezivost označava brojčanu ovisnost između entiteta, odnosno koliko se puta entitet s jedne strane veze pojavljuje u entitetu s druge strane veze. Kao primjer uzmimo entitete osoba i vozilo, i provedimo taj primjer kroz sva tri stupnja konekcije. Veza jedan-na-jedan označavala bi kako jedna osoba može imati samo jedno vozilo, te obrnuto, da je jedno vozilo u vlasništvu samo jedne osobe; veza jedan-na-više označavala bi kako jedna osoba može imati više vozila, ali da jedno vozilo pripada samo jednoj osobi; veza više-na-više značila bi kako jedna osoba može imati više vozila, ali i kako vozilo može biti u vlasništvu više osoba. Dakle, vidljivo je da uvijek gledamo koliko jedna instanca entiteta, dakle jedan red u tablici, ima veza s drugim entitetom. Za vezu jedan-na-više nam je, također, bitno u kojem smjeru gledamo, tj. da li jedna osoba može posjedovati više vozila, a vozilo pripada jednoj osobi ili jedno vozilo može pripadati više osoba, dok osoba može imati samo jedno vozilo.

<sup>6</sup> Thomas Conolly, Carolyn Begg, Richard Holowczak, (2008), Business Database Systems, Addison-Wesley, an imprint of Pearson Education, stranica 136.

<sup>7</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), *Database Modeling and Design: Logical Design (fourth edition)*, Morgan Kaufmann, stranica 20.

Kako bi se veze ostvarile, potrebno je učiniti odgovarajuću stvar s primarnim ključem. U slučaju veze jedan-na-jedan, primarni se ključ jednog entiteta uključuje u drugi entitet i naziva se vanjski ključ, jer je ključ određenog entiteta, ali se trenutno nalazi u drugome entitetu. U ovom slučaju nije bitno iz kojeg entiteta vadimo ključ. U slučaju veze jedan-na-više, primarni ključ se uvijek vadi iz entiteta na strani jedan, i uvrštava kao vanjski ključ na stranu više. U slučaju veze više-na-više, potrebno je izvaditi primarne ključeve oba entiteta i uvrstiti ih u novostvorenu tablicu, tablicu veze. Kad god stvaramo tablicu veze, moguće je u nju uvrstiti i dodatne attribute, potrebne toj vezi.

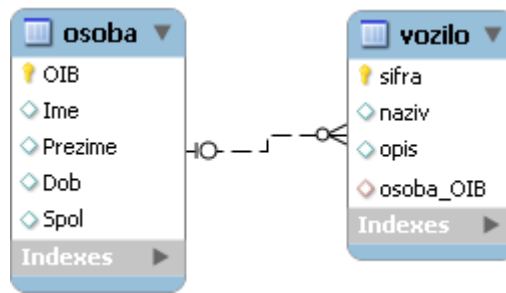
Opcionalnost uključenosti entiteta označava da li entitet mora biti uključen u vezu ili ne. Povezujući se na prethodni primjer, uzmimo povezivost jedan-na-više za entitete osoba i vozilo. Do sad smo utvrdili da jedna osoba može imati više vozila, ali da vozilo pripada samo jednoj osobi. Zapravo smo opcionalnost već i sami dodali, bez da smo to shvatili, čim smo rekli da „osoba može imati“, umjesto da „osoba mora imati“, što označava da osoba nije obvezana imati vozilo. Opcionalnost nam određuje razliku između minimalno nula i minimalno jedan entitet u vezi.

Stupanj konekcije određuje brojnost entiteta u vezi, tj. koliko se različitih entiteta spaja određenom vezom. Razlikujemo unarni stupanj, što je spajanje entiteta samog sa sobom, binarni stupanj, što je spajanje dva različita entiteta, te stupnjeve višeg reda, poput ternarnog, što je spajanje tri entiteta u jednu vezu. Spajanje entiteta samog sa sobom naziva se i binarnom rekurzivnom vezom, budući da se takva veza može prikazati uobičajenom binarnom vezom, koja na obje strane ima isti entitet.

Lako je primjetiti kako smo u dosadašnjim primjerima veza koristili binarnu vezu, budući da je ona veoma intuitivna. Unarna veza također nije komplicirana za razumjeti, pogotovo ako kao primjer uzmemo babuške (ruske lutke koje idu jedna u drugu, po određenoj veličini) i kažemo kako je unarna veza povezivosti jedan-na-jedan zapravo veza između trenutno promatrane babuške i babuške veće razine. Ternarna veza se može opisati primjerom poput atletičara, utrke i natjecanja, gdje zaista moramo uključiti sva tri entiteta kako bi imali točnu informaciju o konkretnom nastupu. Veze višeg stupnja su rijetke, ali se temelje na istom principu kao ternarna veza, te je za svaku takvu vezu potrebno stvoriti novu tablicu s primarnim ključevima svih uključenih entiteta.

Već sam spomenuo način crtanja ER dijagrama zvan „vranino stopalo“, pa je vrijeme da vidimo zašto takvo ime, na slici 2.3. Vidljivo je kako osoba može imati nula ili više vozila, a vozilo može pripadati jednoj ili nijednoj osobi.





Slika 2.3

## 2.2. Napredni ER konstrukti

Ponekad nam osnovne tri klase nisu dovoljne, te tada imamo potrebu uključiti i generalizaciju i agregaciju.

### 2.2.1. Generalizacija: nadtipovi i podtipovi

Generalizacija određuje poopćenje više entiteta sa određenim zajedničkim atributima u entitet više razine, zvan nadtip. Eniteti niše razine, zvani podtipovi, mogu biti ili disjunktni ili preklapajući podskupovi entiteta više razine. Kao primjer nam mogu poslužiti entiteti osoba i zaposlenik. Kako bi ovo jednostavno razumijeli, dovoljno je shvaćati kako ne mora svaka osoba biti zaposlenik, ali kako svaki zaposlenik mora biti osoba. Stoga, entitet osoba i zaposlenik imaju neke zajedničke attribute, poput imena i prezimena, ali ako bi ih ponavljali u obje tablice, imali bi redundanciju podataka, što uvijek pokušavamo izbjeći. Dakle, entitet zaposlenik je podtip entiteta osoba, što zapravo znači kako entitet zaposlenik proširuje entitet osoba dodatnim atributima za one instance za koje je to potrebno. Vidljivo je kako je veza između nadtipa i bilo kojeg od njegovih podtipova uvijek oblika jedan-na-jedan.<sup>8</sup>

Postoje snažni i slabi entiteti. Snažni entiteti ne ovise o postojanju drugog entiteta, dok slabi entiteti ovise o postojanju drugog entiteta, te ih se može prepoznati po tome što nemaju vlastiti primarni ključ, već koriste primarni ključ snažnog entiteta.<sup>9</sup> Možemo reći kako je nadtip uvijek snažni entitet, dok je podtip uvijek slabi entitet.

Suprotno od generalizacije je specijalizacija, što označava proces povećanja razlika između instanci entiteta dodavajući nove karakteristike, odnosno attribute.<sup>10</sup> Dakle, generalizacija je proces koji ide od nižih razina, prema višim, a specijalizacija proces koji ide od viših razina, prema nižim.

<sup>8</sup> Thomas Conolly, Carolyn Begg, Richard Holowczak, (2008), Business Database Systems, Addison-Wesley, an imprint of Pearson Education, stranica 153.

<sup>9</sup> Thomas Conolly, Carolyn Begg, Richard Holowczak, (2008), Business Database Systems, Addison-Wesley, an imprint of Pearson Education, stranica 138.

<sup>10</sup> Thomas Conolly, Carolyn Begg, Richard Holowczak, (2008), Business Database Systems, Addison-Wesley, an imprint of Pearson Education, stranica 155.

### 2.2.2. Agregacija

Agregacija je vrsta apstrakcije između entiteta više i niže razine koja je značajno drugačija od generalizacije. Dok generalizacija opisuje kako je, na primjer, zaposlenik ujedno i osoba, agregacija opisuje vezu između dva entiteta kao da je jedan sastavni dio drugoga. Dobar primjer su dijelovi u vozilu. Svaki dio vozila može biti zaseban entitet, ako su nam potrebni atributi o tom entitetu, ali su svi ti entiteti i dalje dio entiteta vozila, i bez njega nemaju smisla. Dodatna razlika između agregacije i generalizacije je u samim atributima. Napomenuo sam kako u generalizaciji entitet niže razine proširuje entitet više razine dodatnim atributima, ali u agregaciji to nije slučaj; entiteti niže i više razine imaju svoje attribute koji ne moraju biti povezani, a ni isti.

### 2.2.3. Referencijalni integritet

Referencijalni integritet je, po mom mišljenju, ključni dio relacijskih baza podataka. Prilikom objašnjavanja atributa sam napomenuo kako atribut može biti ključni ili neključni, te koja je razlika između te dvije vrste. Sad ću pojasniti kako ta razlika nije zanemariva, već je izrazito bitna i ključni se atributi moraju posebno naznačiti unutar entiteta. Uobičajeno je u entitetu imati samo jedan ključni atribut i nazvati ga primarnim ključem; što činiti ako imamo više ključnih atributa objašnjeno je u trećem poglavlju. Kada želimo koristiti podatke određene instance entiteta u nekoj drugoj tablici, koristimo vrijednost primarnog ključa kako bismo jedinstveno identificirali instancu. Korištenje primarnog ključa drugog entiteta u trenutnoj tablici naziva se vanjskim ključem, i primjer se može vidjeti na slici 2.3., gdje je primarni ključ iz entiteta osoba iskorišten kao vanjski ključ u entitetu vozilo.

Referencijalni integritet nalaže da za svaki vanjski ključ koji postoji u tablici, mora postojati instanca u entitetu čiji primarni ključ koristimo, odnosno da vrijednost vanjskog ključa u tablici mora postojati i kao vrijednost primarnog ključa u entitetu.

## 2.3. ER model

Upoznali smo osnovne elemente svakog ER modela, pa je vrijeme da se napravi ER model baze podataka sustava spomenutog u uvodu. Kako bi znali ispravno napraviti model, potrebno je znati entitete, attribute i veze, pa ću sada u proznom obliku opisati konačan izgled baze podataka, iz kojeg ću zatim izlučiti točne elemente za izradu ER modela.

### 2.3.1. Informacije za bazu podataka

Želimo napraviti sustav koji će moći čuvati podatke o raznim glazbenim grupama i izvođačima. Podaci koje želimo čuvati uključuju naziv grupe, godinu osnutka i, ukoliko postoji, godinu

raspada grupe, izdane albume, trenutne i bivše članove, izdavačke kuće s kojima su imali ugovor, te s kojom ga imaju trenutno, zemlju podrijetla i lokaciju, status aktivnosti, žanrove kojima pripadaju, teme o kojima pjevaju, mjesto za tekst koji bi služio kao opis grupe, te poveznicu na njihovu web stranicu.

Željeli bi, također, imati detaljne podatke o svakom članu grupe i solo izvođaču, jer se često neke osobe pojavljuju u više različitih grupa. Ti podaci bi bili ime, prezime i umjetničko ime izvođača, datum rođenja i trenutni broj godina, datum smrti, državu i grad rođenja, spol, popis grupa u kojima je bio ili čiji član još uvijek jest, te koji instrument izvodi u kojoj grupi.

Želimo imati detaljne podatke o albumima koje glazbenici izdaju, kao i o njihovim izdavačkim kućama. Podaci bi bili naslov albuma, popis pjesama na albumu, trajanje svake pjesme, riječi pjesme, grupa i dodatni izvođači koji su napravili album, vrsta albuma, izdavačka kuća, žanr albuma, datum izdavanja, te tekst sa dodatnim informacijama o albumu.

Želimo znati koja grupa pripada u koji žanr glazbe tako da se to zaključi iz žanrova glazbe njihovih albuma. Žanrovi bi bili preddefinirani, pa bi korisnici naše web stranice povezivali albume i grupe sa žanrovima.

Pored svega toga, želimo skupljati podatke o korisnicima naše web stranice, kako bi korisnici mogli bilježiti koji žanrovi ih zanimaju, koje albume imaju, a koje žele imati, te koje grupe slušaju. Također, shvaćamo kako će neki korisnici također biti članovi neke glazbene grupe u našoj bazi, pa bi htjeli i to učiniti jasnim, tj. želimo omogućiti povezivanje podataka izvođača i korisnika. Podaci o korisnicima bi uključivali ime i prezime, korisničko ime, lozinku, e-mail, spol, datum, mjesto i državu rođenja, trenutno boravište, te tekst u koji svaki korisnik može pisati što želi. Želimo dati članu priliku unosa datuma nabave albuma, trenutnog stanja albuma, osobnu ocjenu albuma, osobnu ocjenu grupe, osobnu ocjenu žanra, te popis najdražih pjesama s mogućnošću komentiranja.

### 2.3.2. Izrada ER modela

Prethodni tekst je sadržavao mnoge korisne informacije, i vjerojatno je puno čišći i razumljiviji od onih kakvi se susreću u praksi, ali je ipak dovoljno zakompliciran kako bi poslužio kao primjer. Sada je potrebno iz tog teksta izlučiti što naručitelj zapravo želi.

Već je opisano kako se prepoznaju entiteti, atributi i veze, pa ću ih stoga direktno implementirati u ER model, vidljiv na slici 2.4, te nakon toga pojasniti svaki element u tablici 2.1. Bitno je znati kako najprije prepoznamo entitete, pa veze, te zatim attribute koje pridjeljujemo entitetima ili vezama. Nakon toga određujemo domenu atributa, tj. odredimo kakvu vrstu vrijednosti atribut

može poprimiti. Zatim slijedi određivanje primarnog ključa, tj. identificirajućeg atributa, pa pregled postojanja generaliziranih entiteta. Nakon toga je korisno provjeriti da li imamo redundantnih podataka, te, ako ih imamo, ispraviti ih.<sup>11</sup>

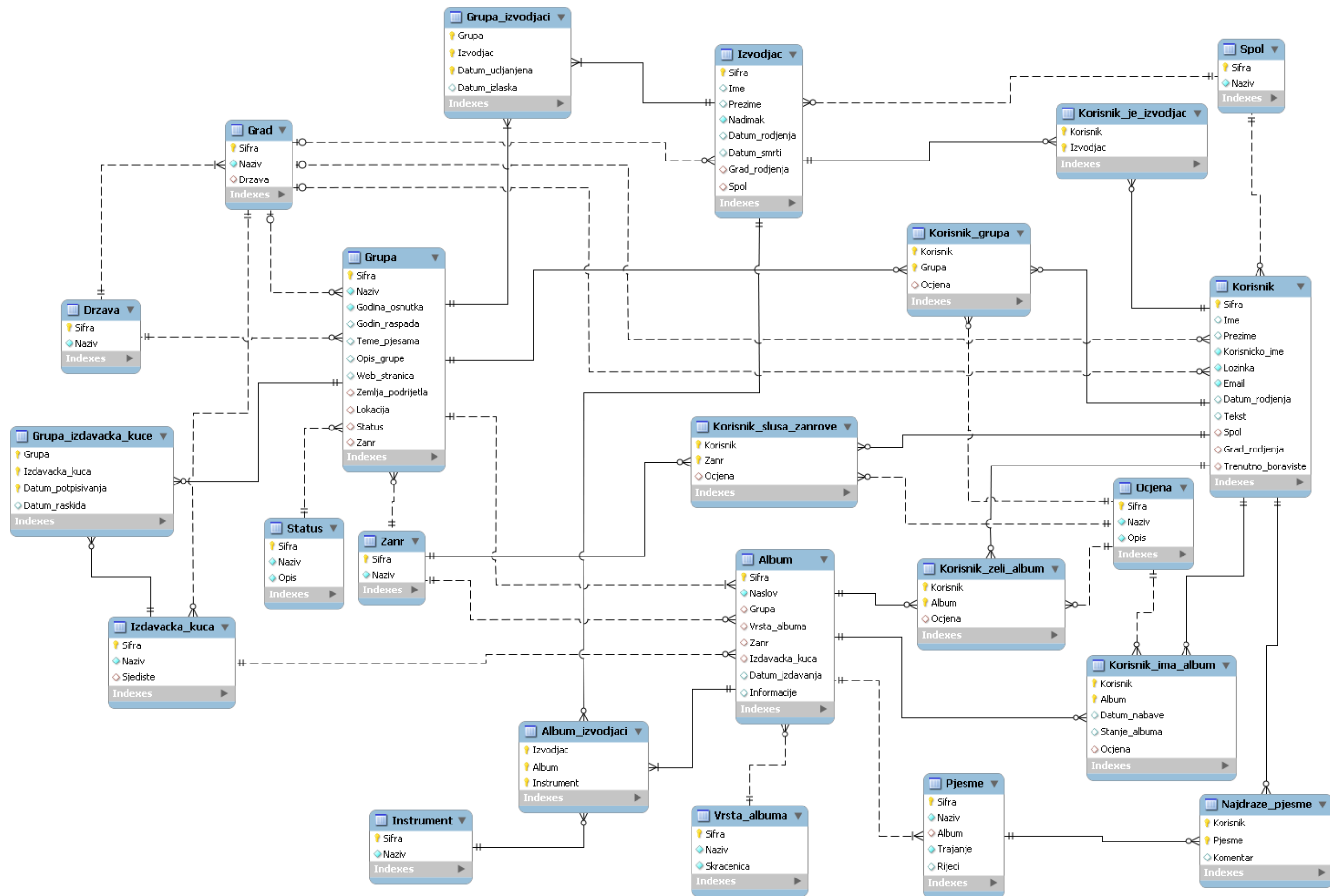
Napomenuo bih kako pune linije označavaju identifikacijsku vezu, a isprekidane neidentifikacijsku. Razlika između ova dva tipa veze je što je identifikacijska veza, veza u kojoj se tablica dijete ne može jedinstveno identificirati bez svog roditelja.

Oprez, ovaj ER model nije strogo na konceptualnoj razini, jer su vidljivi vanjski ključevi unutar tablica (atributi sa crvenim rombom prije naziva atributa). To je stoga što alat MySQL Workbench automatski upisuje vanjske ključeve povezanih tablica, radi lakše preglednosti.

---

<sup>11</sup> Thomas Conolly, Carolyn Begg, Richard Holowczak, (2008), Business Database Systems, Addison-Wesley, an imprint of Pearson Education, stranica 181.

### 2.3.3. ER model baze podataka



Slika 2.4

### 2.3.4. Pojašnjenje ER modela

U tablici 2.1 pojašnjen je svaki element ER modela. Tablica 2.1 sadrži sve tablice iz ER modela<sup>12</sup>, njihove atribute i osobine svakog atributa.

**Tablica 2.1**

Naziv tablice	Naziv atributa	Kratak opis
<b>objašnjenje tablice</b>		
Drzava	Sifra	šifra države
	Naziv	naziv države
Tablica sadržava sve države svijeta.		
Grad	Sifra	šifra grada
	Naziv	naziv grada
	Drzava	država u kojoj se nalazi
Tablica sadržava gradove; označeno u kojoj se državi grad nalazi.		
Status	Sifra	šifra statusa aktivnosti
	Naziv	naziv statusa
	Opis	opis statusa
Tablica sadržava statuse aktivnosti glazbenih grupa, uz opis statusa.		
Zanr	Sifra	šifra žanra
	Naziv	naziv žanra
Tablica sadržava sve žanrove.		
Grupa	Sifra	šifra grupe
	Naziv	naziv grupe
	Godina_osnutka	godina kada je grupa osnovana
	Godina_raspada	godina kada je grupa prekinula postojanje
	Teme_pjesama	kratak opis o čemu grupa sklada i pjeva
	Opis_grupe	opis grupe
	Web_stranica	poveznica na web stranicu grupe
	Zemlja_podrijetla	iz koje države je grupa
	Lokacija	koje je sjedište grupe
	Status	trenutni status aktivnosti grupe
	Zanr	trenutni žanr grupe
Tablica sadržava sve glazbene grupe.		
Izdavacka_kuca	Sifra	šifra izdavačke kuće
	Naziv	naziv izdavačke kuće
	Sjediste	sjedište izdavačke kuće
Tablica sadržava sve izdavačke kuće koje su povezane s grupama u bazi.		
Grupa_izdavacka_kuca	Grupa	šifra grupe

<sup>12</sup> zašto se koristi izraz tablica, a ne entitet, pogledajte u poglavlju 3.1.

Naziv tablice	Naziv atributa	Kratak opis
<b>objašnjenje tablice</b>		
	Izdavacka_kuca	šifra izdavačke kuće
	Datum_potpisivanja	datum potpisivanja ugovora
	Datum_raskida	datum raskida ugovora
Tablica sadržava ugovore između grupa i izdavačkih kuća.		
Spol	Sifra	šifra spola
	Naziv	naziv spola
Tablica sadržava sve spolove. Postavljena kako bi unos ovisio o bazi, a ne o web programeru.		
Izvodjac	Sifra	šifra izvođača
	Ime	ime izvođača
	Prezime	prezime izvođača
	Nadimak	umjetničko ime izvođača
	Datum_rodjenja	datum rođenja
	Datum_smrti	datum smrti
	Grad_rodjenja	grad u kojem se izvođač rodio
	Spol	spol izvođača
Tablica sadržava sve članove grupa.		
Grupa_izvodjaci	Grupa	šifra grupe
	Izvodjac	šifra izvođača
	Datum_uclanjenja	datum kad se izvođač učlanio
	Datum_izlaska	datum kad je izvođač izašao iz grupe
Tablica povezuje izvođače i grupe. Aktivni članovi grupe nemaju unešen datum izlaska.		
Vrsta_albuma	Sifra	šifra vrste albuma
	Naziv	naziv vrste albuma
	Skracenica	skraćeni naziv vrste albuma
Tablica sadržava sve vrste albuma, sa nazivima i skraćenicama koje se koriste.		
Album	Sifra	šifra albuma
	Naslov	naslov albuma
	Grupa	šifra grupe
	Vrsta_albuma	šifra vrste albuma
	Zanr	šifra žanra
	Izdavacka_kuca	šifra izdavačke kuće
	Datum_izdavanja	datum izdavanja albuma
	Informacije	dodatne informacije
Tablica sadrži albume.		
Instrument	Sifra	šifra instrumenta
	Naziv	naziv instrumenta
Tablica sadržava instrumente.		
Album_izvodjaci	Izvodjac	šifra izvođača
	Album	šifra albuma

Naziv tablice	Naziv atributa	Kratak opis
<b>objašnjenje tablice</b>		
	Instrument	šifra instrumenta
Tablica povezuje izvođače i njihove instrumente sa albumima.		
Pjesme	Sifra	šifra pjesme
	Naziv	naslov pjesme
	Album	šifra albuma
	Trajanje	trajanje pjesme
	Rijeci	riječi pjesme
Tablica sadržava sve pjesme sa albuma.		
Korisnik	Sifra	šifra korisnika
	Ime	ime korisnika
	Prezime	prezime korisnika
	Korisnicko_ime	korisničko ime korisnika
	Lozinka	kriptirana lozinka
	Email	e-mail korisnika
	Datum_rođenja	datum rođenja korisnika
	Tekst	slobodni tekst
	Spol	šifra spola
	Grad_rođenja	šifra grada
	Trenutno_boraviste	šifra grada
Tablica sadržava sve korisnike web stranice.		
Najdraze_pjesme	Korisnik	šifra korisnika
	Pjesma	šifra pjesme
	Komentar	komentar korisnika
Tablica sadržava najdraže pjesme svakog korisnika, uz njihov komentar.		
Korisnik_je_izvodjac	Korisnik	šifra korisnika
	Izvodjac	šifra izvođača
Tablica povezuje korisnike koji su istovremeno i izvođači u bazi. Ovo je moglo biti riješeno na način da imamo nadentitet Osoba i podentitete Korisnik i Izvodjac, ali se očekuje maleni broj Izvodjaca koji su ujedno i Korisnici, a brže je dohvaćati podatke iz jedne tablice (Korisnik) nego iz dvije (Osoba i Korisnik) kad god želimo popis korisnika, što se očekuje kao čest zahtjev.		
Ocjena	Sifra	šifra ocjene
	Naziv	naziv ocjene
	Opis	opis ocjene
Tablica sadrži sve ocjene, uz njihov opis.		
Korisnik_grupa	Korisnik	šifra korisnika
	Grupa	šifra grupe
	Ocjena	šifra ocjene
Tablica sadrži ocjene grupa koje unosi korisnik.		
Korisnik_ima_album	Korisnik	šifra korisnika



Naziv tablice	Naziv atributa	Kratak opis
<b>objašnjenje tablice</b>		
	Album	šifra albuma
	Datum_nabave	datum nabave albuma
	Stanje_albuma	stanje albuma
	Ocjena	šifra ocjene
Tablica sadrži albume koje korisnik posjeduje.		
Korisnik_zeli_album	Korisnik	šifra korisnika
	Album	šifra albuma
	Ocjena	šifra ocjene
Tablica sadrži albume koje korisnik želi.		
Korisnik_slusa_zanrove	Korisnik	šifra korisnika
	Zanr	šifra žanra
	Ocjena	šifra ocjene
Tablica sadrži ocjene žanrova koje unosi korisnik.		

### 3. Logičko modeliranje baze podataka

Potrebno je prevesti konceptualni dizajn baze podataka u logički dizajn baze podataka koristeći ER model, kreiran u semantičkom modeliranju, kao izvor informacija za logičko modeliranje baze podataka.<sup>13</sup> Već sam u uvodu ovog rada spomenuo kako u logičkom modeliranju koristimo izraze tablice i stupci, umjesto entiteta i atributa, te da se radi o opisu strukture domene informacija. U ovome poglavlju su objašnjenje tablice, stupci, funkcijske zavisnosti, zavisnosti spoja i normalne forme.

#### 3.1. Kreiranje tablica

Prvi korak u logičkom modeliranju baze podataka je iz postojećeg ER modela kreirati skup tablica za bazu podataka. Možda ste primjetili kako je u tablici 2.1 već korišten izraz tablica, umjesto izraza entitet. To je stoga što nisu sve tablice entiteti, ali je svaki entitet jedna ili više tablica. Iako se pojmovi entitet i tablica mogu smatrati sinonimima, samo konceptualni model sadrži entitete s pripadajućim vezama, dok dizajn konceptualnog modela sadrži pripadajuće tablice. Implementacija takvog dizajna dovodi do fizičke baze podataka. Iz tog razloga ER model na slici 2.4 sadržava tablice i veze između njih, radije nego entitete i veze između njih. Preostali su još samo atributi i njihovi tipovi, koji su zapravo stupci sa definiranom domenom.

##### 3.1.1. Reprezentiranje veza

Prilikom semantičkog modeliranja razmišljali smo o vezama između entiteta, te su te veze ili niz tablica prikazane na ER modelu. Reprezentiranje tih veza u logičkom modeliranju se vrši pomoću mehanizma primarnog i vanjskog ključa, na način koji je već objašnjen u poglavlju 2.1.3. Vidljivo je kako je kreiranje ER modela u spomenutom alatu rezultiralo već gotovim vezama i korektno implementiranim vanjskim ključevima.

##### 3.1.2. Primarni ključevi

Za nastavak rada potrebno je razumijeti koncept primarnog ključa. U poglavlju 2.1.2. spominju se dvije vrste atributa: identifikatori i deskriptori, odnosno ključni i neključni atributi. Ključni atribut jedinstveno određuje instancu entiteta, tj. redak u tablici. Takav atribut nazivamo jednostavnim primarnim ključem. Moguće je da se primarni ključ sastoji od više atributa, te se naziva kompozitnim primarnim ključem. Ako planiramo primarni ključ tablice koristiti kao vanjski ključ u nekoj drugoj tablici, korisno je da taj primarni ključ bude jednostavan. Na slici

---

<sup>13</sup> Thomas Conolly, Carolyn Begg, Richard Holowczak, (2008), Business Database Systems, Addison-Wesley, an imprint of Pearson Education, stranica 208.

2.4. vidljivi su mnogi kompozitni primarni ključevi, ali se oni definiraju u tablicama iz kojih se primarni ključ ne koristi u drugim tablicama kao vanjski, već koje svoje primarne ključeve dobijaju kao vanjske ključeve drugih tablica. U nekim slučajevima korisno je dodati i dodatni atribut u kompozitni primarni ključ, atribut koji nije vanjski ključ, kako bi dobili više mogućnosti unutar tablice.

Za primjer pogledajmo tablicu Grupa\_izvodjaci koja ima kompozitni primarni ključ od tri atributa: Grupa, Izvodjac, te Datum\_uclanjenja. Iako su prva dva atributa primarnog ključa ujedno i vanjski ključevi istoimenih tablica, treći atribut nije vanjski ključ, ali je dodan u kompozitni primarni ključ kako bi bilo moguće da se isti izvođač pridruži u istu grupu na različite datume. Da se primarni ključ sastoji samo od prva dva atributa, svaki bi se izvođač mogao u jednu grupu pridružiti samo jednom, jer se kombinacija ta dva atributa više ne smije ponoviti.

## 3.2. Zavisnosti

### 3.2.1. Funkcionalne zavisnosti

Svojstvo jednog ili više atributa da jedinstveno određuju vrijednost jednog ili više preostalih atributa naziva se funkcionalnom zavisnošću. Za danu tablicu (T), set atributa (Y) je funkcionalno ovisan o drugom setu atributa (X) ako je, u svakom danom trenutku, svaka Y vrijednost asocirana sa samo jednom Y vrijednosti.<sup>14</sup> Takva se funkcionalna zavisnost označava sa  $X \rightarrow Y$ .

Postoje trivijalne, netrivijalne i nestandardne funkcijske zavisnosti. Trivijalne funkcijske zavisnosti se, po dogovoru, ne bilježe prilikom logičkog modeliranja jer ne uzrokuju anomalije i uvijek vrijede. Primjer trivijalne funkcijske zavisnosti je  $X \rightarrow X$ . Netrivijalne funkcijske zavisnosti su zavisnosti koje promatramo u daljnjem radu, a nestandardne funkcijske zavisnosti su zavisnosti gdje je atribut ili skup atributa (Y) ovisan o praznom skupu ( $\emptyset$ ), što znači da je Y u svakom redu jednake vrijednosti. Nestandardne zavisnosti ne proučavamo u ovome radu.

Na skupu netrivijalnih funkcijskih zavisnosti za danu relacijsku shemu možemo klasificirati nekoliko dodatnih tipova zavisnosti.

Funkcionalna zavisnost  $X \rightarrow Y$  je parcijalna ako postoji pravi podskup (X1) lijeve strane zavisnosti (X), za koji vrijedi da X1 povlači desnu stranu zavisnosti (Y), odnosno:  $X1 \rightarrow Y$ .

---

<sup>14</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 111.

Funkcionalna zavisnost  $X \rightarrow Z$  je tranzitivna ako postoji podskup atributa (Y) tablice, za koji vrijedi da atribut ili skup atributa X povlači Y, a Y netrivialno povlači drugi atribut ili skup atributa (Z), ali ne povlači X. Odnosno vrijedi da  $X \rightarrow Y$ , a  $Y \rightarrow Z$ .

### 3.2.2. Višeznačne zavisnosti

Osim funkcijskih zavisnosti, tablica može imati još i višeznačne zavisnosti, te zavisnosti spoja, iako je to rijetko slučaj, jer većina tablica ima samo funkcionalne zavisnosti.<sup>15</sup> Ipak, višeznačne se zavisnosti ne mogu izvesti iz promatranja podataka, već je potrebno znati i semantiku tablice.

Višeznačnu zavisnost označavamo s dvije uzastopne strelice, odnosno  $X \twoheadrightarrow Y$ . Ova zavisnost vrijedi za tablicu T ako i samo ako za valjanu instancu tablice T atributi (X, Y, Z) sadrže par redova s jednakim vrijednostima za X, za koje instanca sadrži i par redova dobivenih izmjenjivanjem Y vrijednosti u originalnom paru. Ovo uključuje situacije gdje postoji samo jedan par redova.

Ovu ćemo definiciju lakše shvatiti na primjeru tablice Restoran, s atributima Restoran, Obrok, Kuhar. Ako pretpostavimo kako restoran ima obroke i kuhare, ali ni kuhar ni obrok ne ovise jedan o drugome, tablica sadrži višeznačne zavisnosti  $\text{Restoran} \twoheadrightarrow \text{Obrok}$  i  $\text{Restoran} \twoheadrightarrow \text{Kuhar}$ . Ako su mogući obroci čevapčići, pizza i lazanje, a kuhari Anić, Ivić i Perić, tada za isti restoran moramo imati ukupno devet redova, kako bi obje višeznačne zavisnosti vrijedile. Te zavisnosti možemo shvatiti kako u tom restoranu određeni obrok može spremati bilo koji kuhar, ali i određeni kuhar može spremati bilo koji obrok. Ukoliko želimo dodati još jedno jelo, na primjer tortilje, tada moramo dodati jedan novi red za svakog kuhara, kako bi višeznačna zavisnost i dalje vrijedila.

Trivijalne višeznačne zavisnosti su zavisnosti za koje vrijedi da je Y podskup od X ili da je unija X i Y jednaka skupu svih atributa u tablici.

### 3.2.3. Zavisnosti spoja

Tablica T zadovoljava zavisnost spoja preko proizvoljno odabranih podskupova atributa tablice ako i samo ako je T jednaka prirodnom spoju projekcija tablice na te attribute. Zavisnost spoja je trivijalna ako je jedan od podskupova cijela tablica.<sup>16</sup> Nijedan od podskupova ne smije biti prazan skup, te unija svih skupova mora sadržavati sve attribute tablice T. Zavisnost spoja

---

<sup>15</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 127.

<sup>16</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 133.

označavamo nabranjem podskupa unutar zagrada kojima prethodi znak prirodnog spoja, odnosno  $\infty(\text{podskup\_1}, \text{podskup\_2}, \dots)$ .

Drugim riječima, dekompozicija skupa atributa  $R$ , u oznaci  $d(R)$ , je skup  $d(R) = \{R_1, \dots, R_k\}$ , gdje je  $R_i$  podskup od  $R$  za svaki  $i$  od 1 do  $k$ , gdje nijedan  $R_i$  nije prazan skup, te gdje unija svih  $R_i$  čini  $R$ . Izraz  $\infty(R_1, \dots, R_k)$  zovemo zavisnost spoja nad  $R$ .

Neke tablice zadovoljavaju samo trivijalne višeznačne zavisnosti, pa se nad njima ne može izvršiti dekompozicija na dvije tablice koja bi sačuvala sve informacije. To se lako može provjeriti tako da se tablice dobivene dekompozicijom spoje pomoću prirodnog spoja. Takve se tablice trebaju dekompozirati na tri tablice, kako bi sve informacije ostale sačuvane. Vidljivo je kako su višeznačne zavisnosti specijalni slučaj zavisnosti spoja.<sup>17</sup>

Tablica  $(T)$  zadovoljava višeznačnu zavisnost  $X \twoheadrightarrow Y$  ako i samo ako se ona dekompozira bez gubitaka, na tablice s atributima  $XY$  i  $XZ$ , gdje su  $Z$  svi preostali atributi, osim  $X$  i  $Y$ . Isti taj uvjet, iskazan kao zavisnost spoja, je  $\infty(XY, XZ)$ .

Ako kao primjer uzmemo tablicu Predavanja s atributima Predavač, Tema i Dvorana, te ako napravimo dekompoziciju na tablice s atributima Predavač, Tema i Predavač, Dvorana, pa dobivene tablice spojimo prirodnim spojem, jasno je kako dobijemo više redova nego u početnoj tablici Predavanja. Stoga je potrebna dekompozicija na tri tablice, dvije već navedene, ali s dodatnom, trećom, s atributima Tema, Dvorana. Prirodni spoj ove tri tablice nam daje tablicu jednaku početnoj. Zavisnost spoja koja tada vrijedi je  $\infty((\text{Predavač}, \text{Tema}), (\text{Predavač}, \text{Dvorana}), (\text{Tema}, \text{Dvorana}))$ .

### 3.3. Normalne forme

U tablicama dobivenim iz ER modela se ponekad pojavljuju problemi, koji utječu na performanse i integritet, te stvaraju anomalije prilikom izmjene podataka u tablici ili njihovog brisanja. Kako bi izbjegli takve neugodnosti, koristimo normalne forme.<sup>18</sup> Proces provođenja normalnih formi nad tablicom, odnosno dovođenje tablice u stanje koje određena normalna forma zahtijeva se naziva normalizacijom. To se postiže analizirajući zavisnosti između atributa tablice te, po potrebi, razdvajanja jedne veće tablice na više manjih. Postoji sedam normalnih formi, od kojih se prve četiri redovito koriste u praksi, a posljednje tri se nazivaju višim

---

<sup>17</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 136.

<sup>18</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 108.

normalnim formama, i nisu toliko često korištene. Međutim, u ovom radu je objašnjenje svih sedam normalnih formi.

Kroz primjere je vidljivo da dobar, promišljen dizajn konceptualnog modela rezultira bazom podataka koja je ili već normalizirana ili ju je moguće normalizirati uz sitne izmjene.<sup>19</sup>

### 3.3.1. Prva normalna forma

Tablica je u prvoj normalnoj formi (1NF) ako i samo ako svi njeni stupci sadrže samo atomske vrijednosti, tj. ako svaki stupac može imati samo jednu vrijednost za svaki red u tablici.<sup>20</sup>

Uzmimo kao primjer tablicu Osoba sa atributima Šifra, Ime, Prezime, Djeca. Atributi Ime i Prezime mogu sadržavati samo jednu vrijednost u redu, ali atribut Djeca je vrsta atributa u koju bi morali spremiti više vrijednosti u jednu ćeliju (mjesto križanja stupca i reda). To krši prvu normalnu formu, pa je potrebno razdvojiti trenutnu tablicu Osoba na dvije manje tablice, tablicu Osoba, s atributima Šifra, Ime, Prezime, i tablicu Djeca, s atributima Šifra\_roditelja i Šifra\_djeteta, gdje su oba atributa vanjski ključ, a zajedno čine kompozitni primarni ključ.

### 3.3.2. Druga normalna forma

Tablica je u drugoj normalnoj formi ako i samo ako je u prvoj normalnoj formi, te ne postoji trivijalna parcijalna zavisnost neključnog atributa od ključa.

Uzmemo li za primjer tablicu Autor\_knjige s kompozitnim primarnim ključem Autor, Knjiga i neključnim atributima Ime, Prezime i Naslov, te zavisnosti  $Autor, Knjiga \rightarrow Ime, Prezime, Naslov$ ,  $Autor \rightarrow Ime, Prezime$  i  $Knjiga \rightarrow Naslov$ , uviđamo postojanje parcijalnih zavisnosti, te zbog toga zaključujemo kako tablica nije u drugoj normalnoj formi. Kako bismo tablicu doveli u 2NF, potrebno je napraviti dekompoziciju tablice na više manjih tablica po jednoj od zavisnosti.

Dekompozicija se vrši na način da u prvu novu tablicu prepisemo sve attribute iz zavisnosti u pitanju, a u drugu tablicu sve attribute osim onih koji se nalaze na desnoj strani zavisnosti. Odnosno, funkcijsku zavisnost  $X \rightarrow Y$ , za relacijsku shemu R, dekomponiramo na tablicu s atributima XY i tablicu s atributima  $X(R \setminus XY)$ .

Ovim postupkom dobijamo tablice Autor s atributima Šifra, Ime i Prezime, te tablicu Autor\_knjige sa atributima Autor, što je vanjski ključ tablice Autor, Knjiga i Naslov. U posljednjoj tablici primarni ključ je Autor, Knjiga, pa zavisnost  $Knjiga \rightarrow Naslov$ , koja još uvijek

---

<sup>19</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 107.

<sup>20</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 109.

vrijedi, krši 2NF. Stoga je potrebno ponovno napraviti dekompoziciju po istim pravilima, ali drugoj zavisnosti.

Konačno dobijemo tri tablice: Autor, Knjiga i Autor\_knjige, od kojih su sve tri u 2NF.

### 3.3.3. Treća normalna forma

Tablica je u trećoj normalnoj formi (3NF) ako i samo ako ne postoji tranzitivna zavisnost neključnog atributa od ključa.

Ako kao primjer uzmemo tablicu Poslovi s atributima Posao, Struka i Školovanje, gdje je Posao primarni ključ, te vrijede zavisnosti  $\text{Posao} \rightarrow \text{Struka}$  i  $\text{Struka} \rightarrow \text{Školovanje}$ , vidljivo je kako vrijedi tranzitivna zavisnost  $\text{Posao} \rightarrow \text{Školovanje}$ , pa je potrebno izvršiti dekompoziciju po zavisnosti  $\text{Struka} \rightarrow \text{Školovanje}$ . Dobivene su tablice sa atributima Posao, Struka i Struka, Školovanje, od kojih su obje u 3NF.

Zanimljivo je primjetiti kako je tablica koja je u 3NF također i u 2NF. Ovo vrijedi za sve daljnje normalne forme, odnosno, ako je tablica u određenoj normalnoj formi, tada je ona i u svim nižim normalnim formama.

### 3.3.4. Boyce-Coddova normalna forma

Tablica je u Boyce-Coddovoj normalnoj formi (BCNF) ako i samo ako za svaku netrivialnu funkcijsku zavisnost vrijedi da lijeva strana zavisnosti sadrži ključ te tablice.

Neka nam za primjer posluži tablica Zaposlenici, s atributima Zaposlenik, Odjel i Menadžer, gdje primarni ključ čine Zaposlenik i Odjel, te vrijede zavisnosti  $\text{Zaposlenik, Odjel} \rightarrow \text{Menadžer}$  i  $\text{Menadžer} \rightarrow \text{Odjel}$ . Ova je tablica u 3NF, ali vidimo da nije u BCNF. Stoga vršimo dekompoziciju po zavisnosti koja krši BCNF, odnosno po  $\text{Menadžer} \rightarrow \text{Odjel}$ , na tablice s atributima Menadžer, Odjel i Menadžer, Zaposlenik, od kojih su obje u BCNF, a zbog toga i u nižim normalnim formama.

### 3.3.5. Četvrta normalna forma

Tablica je u četvrtoj normalnoj formi (4NF) ako i samo ako za svaku netrivialnu višeznačnu zavisnost vrijedi da lijeva strana zavisnosti sadrži ključ te tablice.

Uzmemo li za primjer istu tablicu s kojom smo objasnili višeznačnu zavisnost, shvaćamo kako tablica Restorani nema određen ključ, izuzevši cijele sheme, te stoga ona krši uvjete četvrte normalne forme. Kako bi taj problem riješili, vrši se dekompozicija po višeznačnoj zavisnosti koja krši 4NF, odnosno, po bilo kojoj od navedenih:  $\text{Restoran} \twoheadrightarrow \text{Obrok}$  ili  $\text{Restoran} \twoheadrightarrow \text{Kuhar}$ ; rezultati će biti jednaki.

Dekompozicijom se dobiju tablice s atributima Restoran, Obrok i Restoran, Kuhar, za koje možemo definirati kompozitne primarne ključeve od oba atributa. Dobivene tablice zadovoljavaju uvjete 4NF.

Ukoliko tablica nema netrivialnih višeznačnih zavisnosti, ona je u 4NF.<sup>21</sup>

### 3.3.6. Peta normalna forma

Tablica je petoj normalnoj formi (5NF) ako i samo ako svaka netrivialna zavisnost spoja ima svojstvo da sve njene komponente sadrže ključ.

Ako ponovno uzmemo zavisnost spoja  $\infty((\text{Predavač, Tema}), (\text{Predavač, Dvorana}), (\text{Tema, Dvorana}))$ , koja nam je pomogla prilikom objašnjavanja same zavisnosti spoja, jasno je kako nijedna od komponenata ne sadrži ključ, te je potrebno izvršiti spomenutu dekompoziciju. Tri dobivene tablice su u 5NF.

### 3.3.7. Šesta normalna forma

Tablica je u šestoj normalnoj formi (6NF) ako i samo ako ne postoji netrivialna zavisnost spoja.

Šesta normalna forma se u većini slučajeva smatra suvišnom u praksi, osim ako radimo sa vremenskim, odnosno temporalnim bazama podataka. Stoga su u ovom radu sve tablice dovedene samo do 5NF, budući da bi pretvaranje u 6NF značilo da moramo imati samo tablice od dva atributa, što nam ometa pregledan rad.

## 3.4. Normalizacija

Normalizacija tablica dobivenih iz ER modela se postiže analiziranjem zavisnosti koje se asociraju s tim tablicama, bilo da smo zavisnosti dobili iz ER modela, analizom zahtjeva ili intuitivno.<sup>22</sup>

Stečenim znanjem, lako je zaključiti kako tablica koja ima primarni ključ, bio on jednostavan ili složen, i jednu funkcionalnu zavisnost kojom ključ implicira ostale attribute, zadovoljava petu normalnu formu, a time i sve niže normalne forme, neovisno o broju stupaca.

---

<sup>21</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 130.

<sup>22</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 118.



### 3.4.1. Popis tablica i njihovih zavisnosti

Sve je tablice moguće prikazati jednostavnim popisom u obliku: Ime\_tablice (stupac\_1, ... ). Primarni ključevi su podcrtani su punom crtom, a vanjski ključevi točkicama. Stupac koji je istovremeno vanjski i primarni ključ podcrtan je crtom isprekidanom točkicama.

Uz svaku tablicu popisane su sve njene netrivialne funkcijske i višeznačne zavisnosti, te zavisnosti spoja koje krše petu normalnu formu. Neke tablice, čiji se kompozitni primarni ključ sastoji od različitih vanjskih ključeva, nemaju zavisnosti spoja, jer postoji mogućnost da je nemoguće izvršiti dekompoziju bez gubitaka informacije, pa je to posebno naznačeno kod takvih tablica.<sup>23</sup>

- Drzava (Sifra, Naziv)
  - Sifra → Naziv
- Grad (Sifra, Naziv, Drzava)
  - Sifra → Naziv, Drzava
- Status (Sifra, Naziv, Opis)
  - Sifra → Naziv, Opis
- Zavr (Sifra, Naziv)
  - Sifra → Naziv
- Grupa (Sifra, Naziv, Godina\_osnutka, Godina\_raspada, Teme\_pjesama, Opis\_grupe, Web\_stranica, Zemlja\_podrijetla, Lokacija, Status, Zavr)
  - Sifra → Naziv, Godina\_osnutka, Godina\_raspada, Teme\_pjesama, Opis\_grupe, Web\_stranica, Zemlja\_podrijetla, Lokacija, Status, Zavr
- Izdavacka\_kuca (Sifra, Naziv, Sjediste)
  - Sifra → Naziv, Sjediste
- Grupa\_izdavacka\_kuca (Grupa, Izdavacka\_kuca, Datum\_potpisivanja, Datum\_raskida)
  - Grupa, Izdavacka\_kuca, Datum\_potpisivanja → Datum\_raskida
- Spol (Sifra, Naziv)
  - Sifra → Naziv
- Izvodjac (Sifra, Ime, Prezime, Nadimak, Datum\_rodjenja, Datum\_smrti, Grad\_rodjenja, Spol)
  - Sifra → Ime, Prezime, Nadimak, Datum\_rodjenja, Datum\_smrti, Grad\_rodjenja, Spol
- Grupa\_izvodjaci (Grupa, Izvodjac, Datum\_uclanjenja, Datum\_izlaska)
  - Grupa, Izvodjac, Datum\_uclanjenja → Datum\_izlaska

---

<sup>23</sup> Toby Teorey, Sam Lightstone, Tom Nadeau, (2006), Database Modeling and Design: Logical Design (fourth edition), Morgan Kaufmann, stranica 135.

- Vrsta\_albuma (Sifra, Naziv, Skracenica)
  - Sifra → Naziv, Skracenica
- Album (Sifra, Naslov, Grupa, Vrsta\_albuma, Zanr, Izdavacka\_kuca, Datum\_izdavanja, Informacije)
  - Sifra → Naslov, Grupa, Vrsta\_albuma, Zanr, Izdavacka\_kuca, Datum\_izdavanja, Informacije
- Instrument (Sifra, Naziv)
  - Sifra → Naziv
- Album\_izvodjaci (Izvodjac, Album, Instrument)
  - ne postoji dekompozicija koja čuva informaciju
- Pjesme (Sifra, Naziv, Album, Trajanje, Rijeci)
  - Sifra → Naziv, Album, Trajanje, Rijeci
- Korisnik (Sifra, Ime, Prezime, Korisnicko\_ime, Lozinka, Email, Datum\_rodjenja, Tekst, Spol, Grad\_rodjenja, Trenutno\_boraviste)
  - Sifra → Ime, Prezime, Korisnicko\_ime, Lozinka, Email, Datum\_rodjenja, Tekst, Spol, Grad\_rodjenja, Trenutno\_boraviste
- Najdraze\_pjesme (Korisnik, Pjesma, Komentar)
  - Korisnik, Pjesma → Komentar
- Korisnik\_je\_izvodjac (Korisnik, Izvodjac)
  - ne postoji dekompozicija koja čuva informaciju
- Ocjena (Sifra, Naziv, Opis)
  - Sifra → Naziv, Opis
- Korisnik\_grupa (Korisnik, Grupa, Ocjena)
  - ne postoji dekompozicija koja čuva informaciju
- Korisnik\_ima\_album (Korisnik, Album, Datum\_nabave, Stanje\_albuma, Ocjena)
  - Korisnik, Album → Datum\_nabave, Stanje\_albuma, Ocjena
- Korisnik\_zeli\_album (Korisnik, Album, Ocjena)
  - ne postoji dekompozicija koja čuva informaciju
- Korisnik\_slusa\_zanrove (Korisnik, Zanr, Ocjena)
  - ne postoji dekompozicija koja čuva informaciju

Zaključujemo kako sve tablice zadovoljavaju prvu, drugu, treću, Boyce-Coddovu, četvrtu i petu normalnu formu, a poneka i šestu.

## 4. SQL dizajn baze podataka

Dobivene tablice potrebno je implementirati u sustav za upravljanje relacijskim bazama podataka pomoću SQL koda. Izabrao sam PostgreSQL sustav, pa je slijedeći kod prilagođen njemu.

```
CREATE TABLE „Drzava“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL);

CREATE TABLE „Grad“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL,
    „Drzava“ INTEGER NOT NULL REFERENCES „Drzava“ ON DELETE
    RESTRICT ON UPDATE CASCADE);

CREATE TABLE „Status“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL,
    „Opis“ VARCHAR(200) NOT NULL);

CREATE TABLE „Zanr“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL);

CREATE TABLE „Grupa“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL,
    „Godina_osnutka“ SMALLINT NOT NULL,
    „Godina_raspada“ SMALLINT,
    „Teme_pjesama“ VARCHAR(100),
    „Opis_grupe“ TEXT,
    „Web_stranica“ VARCHAR(100),
    „Zemlja_podrijetla“ INTEGER NOT NULL REFERENCES „Drzava“ ON
    DELETE RESTRICT ON UPDATE CASCADE,
    „Lokacija“ INTEGER REFERENCES „Grad“ ON DELETE RESTRICT ON
    UPDATE CASCADE,
    „Status“ INTEGER NOT NULL REFERENCES „Status“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    „Zanr“ INTEGER NOT NULL REFERENCES „Zanr“ ON DELETE RESTRICT
    ON UPDATE CASCADE);

CREATE TABLE „Izdavacka_kuca“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL,
    „Sjediste“ INTEGER NOT NULL REFERENCES „Grad“ ON DELETE
    RESTRICT ON UPDATE CASCADE);

CREATE TABLE „Grupa_izdavacka_kuca“ (
    „Grupa“ INTEGER REFERENCES „Grupa“ ON DELETE RESTRICT ON
    UPDATE CASCADE,
    „Izdavacka_kuca“ INTEGER REFERENCES „Izdavacka_kuca“ ON
    DELETE RESTRICT ON UPDATE CASCADE,
    „Datum_potpisivanja“ DATE,
    „Datum_raskida“ DATE,
```

```

        PRIMARY KEY („Grupa“, „Izdavacka_kuca“,
        „Datum_potpisivanja"));

CREATE TABLE „Spol“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL);

CREATE TABLE „Izvodjac“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Ime“ VARCHAR(100),
    „Prezime“ VARCHAR(100),
    „Nadimak“ VARCHAR(100) NOT NULL,
    „Datum_rodjenja“ DATE,
    „Datum_smrti“ DATE,
    „Grad_rodjenja“ INTEGER REFERENCES „Grad“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Spol“ INTEGER REFERENCES „Spol“ ON DELETE RESTRICT ON UPDATE
    CASCADE);

CREATE TABLE „Grupa_izvodjaci“ (
    „Grupa“ INTEGER REFERENCES „Grupa“ ON DELETE RESTRICT ON
    UPDATE CASCADE,
    „Izvodjac“ INTEGER REFERENCES „Izvodjac“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Datum_uclanjenja“ DATE,
    „Datum_izlaska“ DATE,
    PRIMARY KEY („Grupa“, „Izvodjac“, „Datum_uclanjenja"));

CREATE TABLE „Vrsta_albuma“(
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL,
    „Skracenica“ VARCHAR(10) NOT NULL);

CREATE TABLE „Album“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL,
    „Grupa“ INTEGER NOT NULL REFERENCES „Grupa“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    „Vrsta_albuma“ INTEGER NOT NULL REFERENCES „Vrsta_albuma“ ON
    DELETE RESTRICT ON UPDATE CASCADE,
    „Zanr“ INTEGER NOT NULL REFERENCES „Zanr“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Izdavacka_kuca“ INTEGER NOT NULL REFERENCES „Izdavacka_kuca“
    ON DELETE RESTRICT ON UPDATE CASCADE,
    „Datum_izdavanja“ DATE,
    „Informacije“ TEXT);

CREATE TABLE „Instrument“ (
    „Sifra“ SERIAL PRIMARY KEY,
    „Naziv“ VARCHAR(100) NOT NULL);

CREATE TABLE „Album_izvodjaci“ (
    „Izvodjac“ INTEGER REFERENCES „Izvodjac“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Album“ INTEGER NOT NULL REFERENCES „Album“ ON DELETE

```

```

RESTRICT ON UPDATE CASCADE,
„Instrument“ INTEGER NOT NULL REFERENCES „Instrument“ ON
DELETE RESTRICT ON UPDATE CASCADE,
PRIMARY KEY („Izvodjac“, „Album“, „Instrument“));

CREATE TABLE „Pjesme“ (
„Sifra“ SERIAL PRIMARY KEY,
„Naziv“ VARCHAR(100) NOT NULL,
„Album“ INTEGER NOT NULL REFERENCES „Album“ ON DELETE
RESTRICT ON UPDATE CASCADE,
„Trajanje“ SMALLINT NOT NULL,
„Rijeci“ TEXT);

CREATE TABLE „Korisnik“ (
„Sifra“ SERIAL PRIMARY KEY,
„Ime“ VARCHAR(100),
„Prezime“ VARCHAR(100),
„Korisnicko_ime“ VARCHAR(100) NOT NULL UNIQUE,
„Lozinka“ VARCHAR(100) NOT NULL,
„Email“ VARCHAR(100) NOT NULL,
„Datum_rodjenja“ DATE,
„Tekst“ TEXT,
„Spol“ INTEGER REFERENCES „Spol“ ON DELETE RESTRICT ON UPDATE
CASCADE,
„Grad_rodjenja“ INTEGER REFERENCES „Grad“ ON DELETE RESTRICT
ON UPDATE CASCADE,
„Trenutno_boraviste“ INTEGER REFERENCES „Grad“ ON DELETE
RESTRICT ON UPDATE CASCADE);

CREATE UNIQUE INDEX „iKorisnici“ ON „Korisnik“
(„Korisnicko_ime“);

CREATE TABLE „Najdraze_pjesme“ (
„Korisnik“ INTEGER REFERENCES „Korisnik“ ON DELETE RESTRICT
ON UPDATE CASCADE,
„Pjesma“ INTEGER REFERENCES „Pjesme“ ON DELETE RESTRICT ON
UPDATE CASCADE,
„Komentar“ TEXT,
PRIMARY KEY („Korisnik“, „Pjesma“));

CREATE TABLE „Korisnik_je_izvodjac“ (
„Korisnik“ INTEGER REFERENCES „Korisnik“ ON DELETE RESTRICT
ON UPDATE CASCADE,
„Izvodjac“ INTEGER REFERENCES „Izvodjac“ ON DELETE RESTRICT
ON UPDATE CASCADE,
PRIMARY KEY („Korisnik“, „Izvodjac“));

CREATE TABLE „Ocjena“ (
„Sifra“ SERIAL PRIMARY KEY,
„Naziv“ VARCHAR(100) NOT NULL,
„Opis“ VARCHAR(200) NOT NULL);

CREATE TABLE „Korisnik_grupa“ (
„Korisnik“ INTEGER REFERENCES „Korisnik“ ON DELETE RESTRICT
ON UPDATE CASCADE,

```

```

    „Grupa“ INTEGER NOT NULL REFERENCES „Grupa“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    „Ocjena“ INTEGER NOT NULL REFERENCES „Ocjena“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    PRIMARY KEY („Korisnik“, „Grupa“));

CREATE TABLE „Korisnik_ima_album“ (
    „Korisnik“ INTEGER REFERENCES „Korisnik“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Album“ INTEGER NOT NULL REFERENCES „Album“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    „Datum_nabave“ DATE,
    „Stanje_albuma“ VARCHAR(100),
    „Ocjena“ INTEGER NOT NULL REFERENCES „Ocjena“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    PRIMARY KEY („Korisnik“, „Album“));

CREATE TABLE „Korisnik_zeli_album“ (
    „Korisnik“ INTEGER REFERENCES „Korisnik“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Album“ INTEGER NOT NULL REFERENCES „Album“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    „Ocjena“ INTEGER NOT NULL REFERENCES „Ocjena“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    PRIMARY KEY („Korisnik“, „Album“));

CREATE TABLE „Korisnik_slusa_zanrove“ (
    „Korisnik“ INTEGER REFERENCES „Korisnik“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Zanr“ INTEGER NOT NULL REFERENCES „Zanr“ ON DELETE RESTRICT
    ON UPDATE CASCADE,
    „Ocjena“ INTEGER NOT NULL REFERENCES „Ocjena“ ON DELETE
    RESTRICT ON UPDATE CASCADE,
    PRIMARY KEY („Korisnik“, „Zanr“));

```

## 5. Zaključak

Počevši sa saznavanjem informacija i podataka potrebnih za izradu baze podataka, izradivši ER model uz pomoć alata MySQL Workbench, normaliziravši nastale tablice, te implementiravši gotov SQL kod u PostgreSQL, dobili smo gotovu bazu podataka, spremnu za unos podataka i povezivanje s web aplikacijom koja će korisnicima dozvoljavati unos podataka.

Ali, ovo nikako ne završava posao modeliranja i dizajna baze podataka, nego je ovo samo vrh sante leda. U ovom radu nije opisano fizičko oblikovanje baze podataka, kao ni sigurnost ili održavanje. Svakom od ovih tema se bave mnoge knjige, koje vrijedi pročitati ako želimo biti kompetentni u tom području.

## 6. Literatura

1. Toby Teorey, Sam Lightstone, Tom Nadeau. (2006). *Database Modeling and Design: Logical Design (fourth edition)*. Morgan Kaufmann publishers.
2. Thomas Conolly, Carolyn Begg, Richard Holowczak. (2008). *Business Database Systems*. Addison-Wesley, an imprint of Pearson Education.
3. Federal Information Processing Standards Publication 184. (1993). Announcing the Standard for Integration definition for information modeling (IDEF1X). S interneta skinuto 30. kolovoza 2011. <http://www.itl.nist.gov/fipspubs/idef1x.doc>
4. Prezentacije, nadopunjene bilješkama, s predavanja profesora dr.sc. Mirka Malekovića